



PUBLIC

SAP Adaptive Server Enterprise 16.0 SP04

Document Version: 1.0 – 2022-04-15

Managing SAP ASE from the Command Line

Content

- 1 **Managing SAP ASE from the Command Line Guide.** **11****
- 2 **Get Started with SAP ASE Cockpit.** **12****
- 2.1 SAP ASE Cockpit Installation. 12
- 2.2 Start and Stop the SAP ASE Cockpit Server. 12
 - Starting and Stopping the SAP ASE Cockpit Server in Windows. 13
 - Starting and Stopping the SAP ASE Cockpit Server in UNIX. 14
 - Configuring SAP ASE Cockpit Server as a Windows Service. 15
 - Configuring SAP ASE Cockpit Server as a UNIX Service. 17
 - cockpit Command. 18
- 2.3 Technical User. 20
 - Creating the Technical User from Scripts. 20
- 3 **Executing SQL Statements.** **21****
- 4 **Configure SAP ASE Cockpit.** **22****
- 4.1 Configuring the E-mail Server. 22
- 4.2 Repository. 24
 - Connect Using the SQL Anywhere Client. 24
 - Performing a Full Repository Database Backup. 25
 - Performing an Incremental Repository Database Backup. 25
 - Restoring a Repository Database from Backup. 26
- 4.3 Logging. 26
- 4.4 Configuring Memory Usage. 26
 - Changing a Memory Option on the Command Line. 27
 - Changing a Memory Option for an SAP ASE Cockpit Windows Service. 28
 - Changing a Memory Option for an SAP ASE Cockpit UNIX Service. 29
- 4.5 SAP ASE Cockpit Console. 30
 - Console Commands. 30
- 4.6 Setting Server Configuration Parameters. 33
 - Viewing the Parameter Table. 33
 - Configuring Procedure Cache Size. 35
- 5 **Security.** **37****
- 5.1 Managing Permissions. 37
 - Enabling Granular Permissions. 39
- 5.2 Manage Logins. 39
 - Creating a Login. 40

	Modifying Login Properties.	41
	Mapping Users to Logins.	41
	Assigning Roles to a Login Account.	42
	Configuring Login Password Properties.	42
	Changing a Login Password.	43
	Making a Login Profile the Default for a Login.	44
	Deleting a Login.	44
	Generating DDL for a Login.	44
5.3	Manage Groups.	45
	Creating a Group.	45
	Adding and Removing Users to a Group.	45
	Granting and Revoking Object Permissions to a Group.	46
	Granting and Revoke Command Privileges to a Group.	47
	Deleting a Group.	48
	Generating DDL for a Group.	48
5.4	Manage Roles.	49
	Creating a Role.	50
	Activating and Deactivating a Role.	50
	Expiring Role Passwords.	51
	Restoring System Roles.	51
	Creating Role Hierarchy.	52
	Assigning Login Profiles to a Role.	52
	Removing Login Profiles Assigned to a Role.	52
	Assigning Logins to a Role.	53
	Removing Logins Assigned to a Role.	53
	Granting and Revoking Object Permissions to a Role.	54
	Granting and Revoking Command Privileges to a Role.	55
	Deleting a Role.	56
	Generating DDL for a Role.	56
5.5	Manage Users.	57
	Creating a User.	57
	Transferring Ownership of a Database Object.	57
	Changing Groups for a User.	58
	Granting and Revoking Object Permissions to a User.	59
	Granting and Revoking Command Privileges to a User.	60
	Deleting a User.	61
	Generating DDL for a User.	61
5.6	Manage Encryption Keys.	62
	Creating a System Encryption Password.	64
	Modifying a System Encryption Password.	65
	Creating a Master Key.	65

	Modifying a Master Key Password.	66
	Creating a Column Encryption Key.	67
	Modifying a Column Encryption Key Password.	68
	Creating a Database Encryption Key.	68
	Granting or Revoke Encryption Permissions to a Role, User, or Group.	69
	Deleting an Encryption Key.	70
	Generating DDL for an Encryption Key.	71
5.7	Manage Login Profiles.	71
	Create a Login Profile.	72
	Modifying a Login Profile Properties.	72
	Add and Remove Roles to a Login Profile.	73
	Transferring Login Attributes to a Login Profile.	73
	Deleting a Login Profile.	74
	Generating DDL for a Login Profile.	74
6	Alerts in SAP ASE.	75
6.1	Monitor Alerts.	75
	Viewing Alert Profile.	75
	Viewing Alert Threshold.	77
	Viewing the Latest Alert Data for a Managed ASE Resource.	78
6.2	Managing Alert Notification Settings.	88
6.3	SAP ASE Alert and Data Collections Summary.	90
7	Monitor SAP ASE.	96
7.1	Manage Monitor Access Permission.	96
7.2	Viewing the Performance Overview.	96
	Viewing Overall Server Status.	96
	Viewing Engine CPU Utilization.	97
	Viewing Device IO/Sec.	98
	Viewing Memory.	98
	Viewing Processes.	99
	Viewing Details.	100
	Viewing Configured Resources.	102
	Viewing Wait Events.	103
	Viewing Licenses	104
7.3	Monitor Databases.	105
	Viewing Monitor Statistics for All Open Databases.	105
	Viewing Information About Space Usage.	106
	Viewing Combined Data and Log Space	107
	Viewing Information About Processes Currently Using this Database.	107
	Viewing Information About Devices on Which This Database Stores Data.	108
	Viewing Information About Segments Used by This Database.	110

	Viewing Indexes in this Database Not Used Since the Last Server Restart.	111
	Viewing Information about Tables in this Database Used Since the Last Server Restart.	112
	Viewing Database Statistics.	113
7.4	Monitor Devices.	115
	Viewing Device Statistics and Determine I/O Response Time.	115
	Viewing Used and Unused Space.	116
	Viewing Device IO Response Time and Device APF Reads/Sec.	117
	Viewing the Logical Name List.	118
7.5	Monitor Engines.	119
	Viewing All Engines.	119
	Viewing Engine Utilization - IO Processing & Garbage Collection.	120
	Viewing Engine Utilization - Engine CPU Utilization.	121
7.6	Monitor Threads.	121
	Viewing Thread Pools.	121
	Viewing Thread Pool Details.	123
	Viewing Thread Pool CPU Utilization.	124
	Viewing Thread Pool Tasks.	124
7.7	Monitor Processes.	126
	Identifying Resource-Intensive Processes.	126
	Identifying and Terminating Blocked and Blocking Processes.	134
	Viewing Wait Events Associated with a Process.	136
	Viewing the SQL Query Associated with a Process.	137
7.8	Monitor Segments	137
	Viewing a List of Segments.	137
	Viewing Devices Used in a Segment.	138
	Viewing Tables Used in a Segment.	140
	Viewing Indexes Used in a Segment	144
	Viewing Used and Unused Space in a Segment	145
7.9	Monitor Statistics.	146
	Finding the Target Resource Name for Monitoring Statistics.	159
	Finding KPAName and StatName for Monitoring Statistics.	161
	Key Performance Indicators for SAP ASE.	170
7.10	Monitor Transactions.	174
7.11	Monitor SQL Queries.	175
7.12	Monitor The HADR Environment	182
	Replication Paths.	182
	Service Components.	190
	Log Records.	192
	Viewing Throughput.	193
	Viewing the Latency.	194
	Replication Agent.	195

7.13	Monitor Caches.	197
	Data Cache Statistics and Details.	197
	Procedure Cache Statistics and Details.	201
	Statement Cache Statistics and Details.	207
	In-memory Storage Statistics and Details.	217
8	Manage SAP ASE.	223
8.1	Replacing Compiled Object Definitions.	223
8.2	Manage Tables and Indexes.	224
	Restoring Table Data.	224
	Checking Table Consistency.	225
	Placing a Table on a Segment.	225
	Setting a Table Locking Scheme.	226
	Reorganize Tables and Table Objects.	226
	Managing Table or Column Permissions.	230
	Creating a Table.	233
	Creating a Proxy Table.	233
	Creating a Column.	233
	Creating an Index.	234
	Deleting a Table.	234
	Deleting an Index.	235
	Table Level Properties.	235
	Column Level Properties.	254
	Index Level Properties.	260
8.3	Manage Devices.	268
	Viewing a Device Object.	268
	Creating a Database Device.	270
	Modifying Device Sizes.	270
	Creating an In-Memory Device.	271
	Creating a Dump Device.	271
	Database Device Properties.	272
	Viewing In-Memory Device Properties.	274
	Generating DDL for a Device.	274
	Deleting a Database, In-Memory, or Dump Device.	275
8.4	Manage Databases.	276
	Creating a Database.	276
	Creating an Archive Database.	276
	Encrypting a Database.	277
	Creating a Temporary Database Group.	278
	Viewing Database Properties.	278
	Modifying Database Properties.	285
	Database Administration.	288

	Deleting a Database.	292
	Generating DDL for a Database.	293
8.5	Manage Statistics.	294
	Updating Statistics on a Table.	295
	Updating Statistics on Specific Columns.	295
	Updating Statistics on an Index.	296
	Updating Statistics for a Data Partition.	296
	Updating Statistics on an Index Partition.	296
	Deleting Statistics from a Table.	297
	Deleting Statistics from a Column.	297
	Deleting Statistics from a Data Partition.	297
8.6	Manage Compression.	298
	Identifying Table Compression Candidates.	298
	Initiating a Compression Estimate.	299
	Applying Compression Settings.	300
	Managing Index Compression.	301
8.7	Manage Partitions.	302
	Enabling Partition Locking.	304
	Enabling Semantic-based Partitioning.	304
	Using a Hash Partition.	304
	Using a Range Partition.	305
	Using a Partition List.	306
	Using a Round-Robin Partition.	306
	Splitting a Partition.	307
	Merging Partitions.	307
	Moving a Partition.	308
	Deleting a Partition.	309
	Managing Partition Properties.	310
8.8	Manage Remote Servers.	310
	Configuring a Server for Remote Procedure Calls.	310
	Adding a Remote Server.	311
	Testing a Remote Server Connection.	311
	Setting Options for a Remote Server.	312
	Managing Remote Server Login Mappings.	312
	Managing CIS Roles and Logins Mappings.	312
	Deleting a Remote Server.	313
	Viewing Remote Server Properties.	313
8.9	Manage Always-On (HADR) Option.	314
	Viewing the HADR Properties.	314
	Viewing the HADR System Status.	316
	Viewing HADR States.	319

	Performing a Planned Failover.	322
	Suspending Replication.	324
	Resuming Replication.	324
	Rematerializing a Database.	325
	Adding an Existing Database to the HADR System Send Feedback.	325
	Modifying the Replication Agent Configuration Parameters.	325
	HADR KPIs.	326
8.10	Manage Backup and Restore.	332
	Backing Up (Dumping) a Database.	332
	Scheduling a Database Backup.	339
	Restoring a Database.	343
8.11	Manage Execution Classes.	345
	Creating Execution Classes.	345
	Viewing General Execution Classes Properties.	345
	Binding Execution Classes	346
	Unbinding Execution Classes.	346
	Modifying Bindings to Execution Classes.	346
	Viewing Binding Object.	347
8.12	Manage Engine.	347
	Creating Engine Groups.	347
	Adding an Engine to an Engine Group.	348
	Removing an Engine from an Engine Group.	348
	Deleting an Engine Group.	348
	Generating DDL for an Engine Group.	349
8.13	Manage Segments.	350
	Viewing Segments.	350
	Extend a Segment.	351
	Creating a Segment.	351
	Managing Segment Properties.	352
	Removing a Threshold from a Segment.	355
	Adding a Threshold to a Segment.	356
	Removing a Database Device from a Segment.	356
	Adding a Database Device to a Segment.	357
	Deleting a Segment.	357
	Generating DDL for a Segment.	357
8.14	Manage Thread Pools.	358
	Creating a Thread Pool.	358
	Viewing General Thread Pool Properties.	359
	Viewing Thread Pool Execution Classes Properties.	360
8.15	Manage User-Defined Datatypes.	360
	Adding a User-Defined Datatype	360

	User-Defined Datatypes Properties.	361
	Deleting a User-Defined Datatype.	364
8.16	Manage Incrementally Transferring Data.	364
	Enabling Incremental Transfer.	364
	Incrementally Transferring Data In.	365
	Incrementally Transferring Data Out.	365
8.17	Manage Bulk Copying Data.	366
	Bulk Copying Data Into or Out of Tables.	367
8.18	Manage Views.	367
	Creating a View.	367
	Replacing a View Definition.	367
	Managing View Properties.	368
	Managing View Permissions.	373
	Deleting a View.	374
8.19	Manage Triggers.	374
	Creating a Trigger.	374
	Viewing Trigger Properties.	375
	Viewing Objects Referenced by a Trigger.	376
	Modifying a Trigger Definition.	376
8.20	Manage Caches.	377
	Creating a Cache.	377
	Viewing Cache Properties.	377
	Managing Binding Options.	383
	Managing Buffer Pools.	384
	Modifying Data Cache Sizes.	384
	Generating DDL for a Cache.	385
	Deleting a Cache.	386
	Setting the Statement Cache Size.	386
8.21	Manage Constraints.	387
	Creating a Foreign Key.	387
	Foreign Key Properties.	387
	Creating a Check Constraint.	389
	Viewing General Check Constraint Properties.	390
	Creating a Unique Constraint or Primary Key.	390
	Binding Defaults and Rules to a Column.	391
8.22	Manage Rules.	391
	Creating a Rule.	391
	Replacing a Rule Definition.	392
	Manage Rule properties.	392
	Deleting a Rule.	394
	Generating DDL for a Rule.	395

8.23	Manage Precomputed Result Sets.	396
	Creating a Precomputed Result Set	396
	Deleting a Precomputed Result Set	397
	Altering a Precomputed Result Set	397
	Refreshing Precomputed Result Sets	397
	Truncating a Precomputed Result Set.	398
	Managing Permissions on Precomputed Results Sets.	398
	Precomputed Result Sets Properties.	399
	Generating DDL for a Precomputed Result Set.	405
8.24	Manage Defaults.	407
	Creating a Default.	407
	Manage Defaults Properties.	408
	Replacing a Default Definition.	410
	Deleting a Default.	410
8.25	Manage Stored Procedures.	411
	Manage Stored Procedures.	411
	Manage SQLJ Procedures.	424
8.26	Manage Extended Stored Procedures.	432
	Creating an Extended Stored Procedure.	432
	Replacing an Extended Stored Procedure Definition.	433
	Extended Stored Procedure Properties.	433
	Managing Extended Stored Procedure Permissions.	435
	Deleting an Extended Stored Procedure.	436
	Generating DDL for an Extended Stored Procedure.	437
8.27	Manage Functions.	438
	Scalar Functions.	438
	SQLJ Functions.	453
8.28	Manage Data Store Access Management (DSAM).	460
	DSAM Configuration.	461
9	Capturing, Analyzing, and Replaying Workloads.	475
10	Collect Diagnostic Data.	476
11	Troubleshoot SAP ASE Cockpit Server.	477
11.1	Authentication and Access Problems.	477
	Cockpit Server Fails to Start.	477
	SAP Adaptive Server and Replication Server Fails to Start.	478
	Authentication Removed After Host Name Change.	478
11.2	Performance Problems.	479
	Out of Memory Errors.	479
	Memory Warnings at Startup.	479

1 Managing SAP ASE from the Command Line Guide

With the retirement of Adobe Flash Player, this guide is designed to provide documentation on how to accomplish SAP Adaptive Server Enterprise (ASE) Cockpit tasks using existing, alternative functionality in SAP ASE.

This guide describes how to configure, monitor and manage, and administer the SAP Adaptive Server Enterprise cockpit server, using a SQL console.

For up-to-date information on SAP's mitigation plan, see [2905660](#) -SAP Adaptive Server Enterprise Cockpit End of Life: December 2020.

2 Get Started with SAP ASE Cockpit

Start and configure the SAP ASE Cockpit server and launch the SAP ASE Cockpit console.

2.1 SAP ASE Cockpit Installation

SAP ASE Cockpit server is installed as part of an SAP ASE server installation and initial configuration of an ASE.

SAP ASE cockpit is licensed free of charge to customers who have a paid license for SAP ASE.

For detailed installation information, see the *Installation and Upgrade Guide for SAP Adaptive Server Enterprise* for your platform.

To manually add an ASE instance for ASE Cockpit server, please refer to SAP Note [2133600](#) .

Related Information

[Installation and Upgrade Guide for HP-UX](#)

[Installation and Upgrade Guide for Windows](#)

2.2 Start and Stop the SAP ASE Cockpit Server

You can start SAP ASE Cockpit server manually, or set the service to start automatically and to restart in case of failure.

If you elect to run the SAP ASE Cockpit server manually, you must issue a command every time you want to start or shut down the server. If you elect to run as a service (recommended), you can configure the service to start and restart automatically.

Installation of SAP ASE Cockpit server prompts you for your preferred method.

Regardless of the start method used, wait at least five minutes after startup to log in to the SAP ASE Cockpit console. Startup triggers automatic discovery, technical user verification, and collection jobs. Competing with these processes may cause unexpected behavior.

2.2.1 Starting and Stopping the SAP ASE Cockpit Server in Windows

When you run SAP ASE Cockpit server manually, issue a command every time you start or shut down.

Prerequisites

Ensure that your home directory—that is, the home directory of the user who starts SAP ASE Cockpit—is writable from the SAP ASE Cockpit host. If the starting user cannot write to the home directory, SAP ASE Cockpit logs an error and fails to launch. Resetting the user's HOME environment variable to a writable directory does not solve the problem.

Context

i Note

If you are starting SAP ASE Cockpit server for the first time in Windows 2008, Windows 7, or Windows 8, use the *Run as Administrator* option so that SAP ASE Cockpit can register its ODBC driver. (This is necessary even if you are logged in as an administrator.)

Procedure

1. To start the SAP ASE Cockpit server, execute:

```
%SYBASE%\COCKPIT-4\bin\cockpit.bat
```

2. To stop SAP ASE Cockpit, at the cockpit> prompt, execute:

```
shutdown
```

3. At the prompt, enter a text reason for the shutdown.

⚠ Caution

Do not enter `shutdown` at a Windows prompt; it shuts down the operating system.

Related Information

[cockpit Command \[page 18\]](#)

2.2.2 Starting and Stopping the SAP ASE Cockpit Server in UNIX

When you run SAP ASE Cockpit server manually, issue a command every time you start or shut down.

Prerequisites

Ensure that your home directory—that is, the home directory of the user who starts SAP ASE Cockpit—is writable from the SAP ASE Cockpit host. If the starting user cannot write to the home directory, SAP ASE Cockpit logs an error and fails to launch. Resetting the user's HOME environment variable to a writable directory does not solve the problem.

Context

You can start the SAP ASE Cockpit server in the foreground or background. When you run SAP ASE Cockpit server in the background, you can use `nohup`, `&`, and `>` to redirect output and system error to a file, and suppress the SAP ASE Cockpit server console.

Procedure

1. Set the environment variables. Do this only once.
 - a. Change to the parent of the SAP ASE Cockpit installation directory.
 - b. Execute one of the following to set environment variables.

Bourne shell:

```
. SYBASE.sh
```

C shell:

```
source SYBASE.csh
```

2. To start the SAP ASE Cockpit server, do one of:

Option	Description
In the foreground	Execute: <pre>\$\$SYBASE/COCKPIT-4/bin/cockpit.sh</pre>
In the background	Execute a command similar to the sample provided that matches your shell. The sample command directs output to the file <code>cockpit-console.out</code> . If the output file already exists, you might

Option	Description
	need to use additional shell operators to append to or truncate the file. To check which shell is being used, execute <code>echo \$SHELL</code> .
	Bourne shell (sh) or Bash
	<pre>nohup ./cockpit.sh 2>&1 > cockpit-console.out</pre>
	C shell
	<pre>nohup ./cockpit.sh >& cockpit-console.out</pre>

3. To stop the SAP ASE Cockpit server, do one of:

Option	Description
Running in the foreground	At the <code>cockpit></code> prompt, execute:
	<pre>shutdown</pre>
	At the prompt, enter a text reason for the shutdown.
	⚠ Caution Do not enter <code>shutdown</code> at a UNIX prompt; it shuts down the operating system.
Running in the background	At the UNIX command line, execute:
	<pre>\$\$SYBASE/COCKPIT-4/bin/cockpit.sh --stop</pre>

Related Information

[cockpit Command \[page 18\]](#)

2.2.3 Configuring SAP ASE Cockpit Server as a Windows Service

You can set the service to start automatically and to restart in case of failure. SAP recommends running SAP ASE Cockpit as a service.

Prerequisites

Ensure that your home directory—that is, the home directory of the user who starts SAP ASE Cockpit—is writable from the SAP ASE Cockpit host. If the starting user cannot write to the home directory, SAP ASE

Cockpit logs an error and fails to launch. Resetting the user's HOME environment variable to a writable directory does not solve the problem.

Context

If you run SAP ASE Cockpit server as a service, you can still manually start and stop the service, as needed. By default, if you install SAP ASE Cockpit to run as a service, the service is configured to automatically start, but not automatically restart. You must configure this manually.

Procedure

1. To configure automatic restart:
 - a. In Windows Control Panel, open Services with Administrative Tools.
 - b. Click the *Recovery* tab and change the First, Second, and Subsequent failures to *Restart the Service*
 - c. Click *OK*.
2. To manually start the service:

Option	Description
In the Services window	Click <i>Start</i>
At a command line	Execute: <pre>net start "Cockpit 4.0"</pre> <p>The Cockpit 4.0 service is starting..... The Cockpit 4.0 service was started successfully.</p>

3. To manually stop the service:

Option	Description
In the Services window	Click <i>Stop</i>
At a command line	Execute: <pre>net stop "Cockpit 4.0"</pre> <p>The Cockpit 4.0 service is stopping..... The Cockpit 4.0 service was stopped successfully.</p>

2.2.4 Configuring SAP ASE Cockpit Server as a UNIX Service

You can set the service to start automatically and to restart in case of failure. SAP recommends running SAP ASE Cockpit as a service.

Prerequisites

Ensure that your home directory—that is, the home directory of the user who starts SAP ASE Cockpit—is writable from the SAP ASE Cockpit host. If the starting user cannot write to the home directory, SAP ASE Cockpit logs an error and fails to launch. Resetting the user's HOME environment variable to a writable directory does not solve the problem.

Context

A UNIX service is a daemon process that starts automatically after the machine is started and runs in the background. UNIX installations of SAP ASE Cockpit server include a shell script, `cockpitd`, which you can use to configure the SAP ASE Cockpit service. (Some UNIX platforms supply tools that make service configuration easier; Linux `chkconfig` is an example.)

i Note

SAP recommends that if you are not familiar with setting up services in UNIX, you delegate this task to a system administrator or consult the system administration documentation for your UNIX platform.

Procedure

1. Open `cockpitd` and make these changes:
 - a. Change the line that sets the SYBASE variable to the location of your SAP installation (that is, the parent of COCKPIT-4, the SAP ASE Cockpit installation directory). By default, this directory is called `/opt/sybase` if you installed SAP ASE Cockpit on a machine with an existing SAP product or environment variable; otherwise the default parent directory is `/opt/sap`.
 - b. If you are using shared-disk mode to run a single instance whose name is not the host name, or to run multiple instances on the same host, add the instance name to the script name.

```
SCRIPT_NAME=cockpit.sh
```

to:

```
SCRIPT_NAME="cockpit.sh -instance <instance-name>"
```

2. In LINUX, configure the service to run in run levels 2, 3, 4, and 5. Execute:

```
/usr/sbin/chkconfig --add cockpitd
```

```
/usr/sbin/chkconfig --level 2345 cockpitd
```

You can test the `cockpitd` script with `/usr/sbin/service cockpitd status`. (The `service` command accepts these options: `start` | `stop` | `status` | `restart`.)

3. Use the `S90cockpitd` and `K10cockpitd` links to test starting and stopping the service. The links are called automatically when the machine is started or shut down.

2.2.5 cockpit Command

Use `cockpit.bat` (Windows) or `cockpit.sh` (UNIX) to manually start and stop SAP ASE Cockpit servers and to perform administrative tasks like configuring ports and enabling and disabling services.

A single cockpit server can manage other SAP database products if they reside on the same host. The cockpit server shares ports and other resources, and the `cockpit` command enables all installed plugins.

Use the `instance` command to manage multiple server instances from the same Cockpit installation directory.

Syntax

```
cockpit[.bat | .sh] [-a | --address <RMI-service-address>]
[-b | --bitwidth]
[--dbpassword]
[-disable | --disable <service-name,service-name...>]
[-enable | --enable <service-name,service-name...>]
[-h | --help]
[-instance [<instance-name>]]
[-m | --message <message-level>]
[-p | --port {<port-name>=<port-number> |
<service-name>:<property-name>=<port-number>}]
```

Parameters

- | | |
|--|---|
| <code>-a --address <RMI-service-address></code> | the address for the RMI service to use; must be an IP address on this machine or the name of this machine (which is the default). |
| <code>-b --bitwidth</code> | returns an informational only string identifying the bit width (32 or 64) of the underlying platform. |
| <code>--dbpassword</code> | changes the password of the default dba account provided for the repository database. It prompts you for the new password, validates it, and starts the SAP ASE Cockpit server. |
| <code>-disable --disable <service-name,service-name...></code> | disable the specified services. |

-enable | --enable <service-name, service-name...> enable the specified services. Separate each service from the next with a comma.

-h | --help display help and usage information for the `cockpit` command.

-instance [<instance-name>] use with other options (`-start` and `-stop`, for example) to specify an instance in a shared disk deployment. If you do not enter a name for the instance, it defaults to the host name.

-m | --message <message-level> set the amount of detail recorded in system logs; also known as the logging level. Valid values are OFF, FATAL, ERROR, WARN, INFO, DEBUG, and ALL. WARN is the default.

-p | --port {<port-name>=<port-number> | <service-name>:<property-name>=<port-number>} configure the specified service to run on the specified port. Changing ports is useful if you discover a port conflict with other software on the same system.

Valid port names, service names and property names are:

Port Name	Description	Service Names	Property Names	Default Port
db	Database port	SccSADataserver Messaging Alert	com.syb-ase.asa.server.port messaging.db.port alert.database.port	4638
msg	Messaging port	Messaging	messaging.port	4993
rmi	RMI port	RMI	port	4992
tds	Tabular Data Stream™ port (used to communicate with other SAP database products)	Tds	tdsPort	4998

You can also execute `cockpit --info ports` to display service names and associated property names; they appear in the first two columns of the output.

Related Information

[Starting and Stopping the SAP ASE Cockpit Server in Windows \[page 13\]](#)

[Starting and Stopping the SAP ASE Cockpit Server in UNIX \[page 14\]](#)

2.3 Technical User

The Cockpit Technical User account is a dedicated login used for creating and scheduling collection jobs, monitoring, and triggering alerts.

i Note

Do not use the CREATE USER statement to create the account.

The technical user is created during installation of SAP ASE.

2.3.1 Creating the Technical User from Scripts

Create the account under which collections and alerts run.

Procedure

1. Set the environment variables. Do this only once.
 - a. Change to the parent of the SAP ASE Cockpit installation directory.
 - b. Execute one of the following to set environment variables.

Bourne shell:

```
. SYBASE.sh
```

C shell:

```
source SYBASE.csh
```

2. Prepare the technical user resource file.
 - a. Change to the ASEMAP plugin directory under SAP ASE Cockpit installation.
 - b. Prepare a techuser.res file with following entries

```
RESOURCE_NAME=<SAPASE_SERVER_NAME>  
MACHINE=<HOST>  
PORT=<PORT>  
LOGIN_NAME=sa  
LOGIN_CREDENTIALS=<SA_PASSWD>  
TECHUSER_NAME=<TECH_USER_NAME>  
TECHUSER_CREDENTIALS=<TECH_USER_PASSWD>  
PLUGIN_DIRECTORY=$SAPASE/COCKPIT-4/plugins/ASEMAP
```

3. Create the technical user:
 - a. Navigate to the ASEMAP plugin directory under SAP ASE Cockpit installation.
 - b. Execute the shell script:

```
TechnicalUserCreate.sh techuser.res
```

3 Executing SQL Statements

Execute SQL statements on one or more servers.

You can use Interactive SQL or the SQL Anywhere client to run any valid SQL statement, including queries and stored procedures. Anyone can launch a query; no permissions are required. However, if you do not have authority to perform the actions in the query, SAP ASE server displays an error.

Related Information

[Using Interactive SQL from the Command Line](#)

[SAP SQL Anywhere Database Client Download](#) 

4 Configure SAP ASE Cockpit

There are several options available to configure SAP ASE Cockpit.

i Note

These options are over and above the minimum configuration described in [Get Started with SAP ASE Cockpit \[page 12\]](#).

4.1 Configuring the E-mail Server

Specify the e-mail server for SAP ASE Cockpit to send alert notifications.

Context

Check if email server already set with the following statement. Make sure you really want to modify them.

Procedure

1. Connect to the Cockpit repository database.
2. Take a full backup of the Cockpit repository database.
3. Review the current e-mail server settings. Execute:

```
SELECT
  T.TARGET_TYPE_NAME,
  DRIVER_CLASS_NAME,
  TARGET_NAME,
  PARAM_NAME,
  PARAM_VALUE
FROM
  UAS_TARG_TYPE T,
  UAS_TARG_INST TAR,
  UAS_TARG_INST_PRMS P
WHERE
  T.TARGET_TYPE_NAME = TAR.TARGET_TYPE_NAME
  AND P.TARGET_ID = TAR.TARGET_ID
  AND TAR.TARGET_TYPE_NAME='JavaMail'
  AND P.PARAM_NAME IN ('SMTP_SERVER_NAME', 'SMTP_SERVER_PORT')
ORDER BY
  T.TARGET_TYPE_NAME,
  TAR.TARGET_NAME
```

Note

If values have not been set, the query returns nothing and nothing can be changed.

Result Set

TARGET_TYPE_	NAME	DRIVER_CLASS_NAME	TARGET_NAME	PARAM_NAME	PARAM_VALUE
	JavaMail	com.sybase.alerts.drivers.JavaMail.JMailAlertDriverNotifier	ScEmailTarget	SMTP_SERVER_NAME	mail07.xxx.xxx.xxx
	JavaMail	com.sybase.alerts.drivers.JavaMail.JMailAlertDriverNotifier	ScEmailTarget	SMTP_SERVER_NAME	25

- Set connection authentication to allow modification of the database. Execute:

```
SET TEMPORARY OPTION
CONNECTION_AUTHENTICATION='Company=SAP;Application=Cockpit;
Signature=000fa55157edb8e14d818eb4fe3db41447146f1571g6eb2a6d2966db36d82642c745
d1d1bde020b45b6'
```

- Update the configuration setting. Execute:

Note

Execute the statements as a single transaction.

```
UPDATE UAS_TARG_INST_PRMS SET PARAM_VALUE='<new_server_name>'
WHERE param_name = 'SMTP_SERVER_NAME' and param_value=
'<current_server_name>'
UPDATE UAS_TARG_INST_PRMS SET PARAM_VALUE='<new_server_port>'
WHERE param_name = 'SMTP_SERVER_PORT' and
param_value='<current_server_port>'
```

- Repeat step 3. Verify that the result set reflects the new server name and port values.
- Restart the cockpit server. See [Start and Stop the SAP ASE Cockpit Server \[page 12\]](#)

Related Information

[Connect Using the SQL Anywhere Client \[page 24\]](#)

[Performing a Full Repository Database Backup \[page 25\]](#)

4.2 Repository

The SAP ASE Cockpit embedded repository stores information related to managed systems, as well as user preference data, operational data, statistics, and alert configuration.

You can back up and restore the repository from backups. Full and incremental backups are available. A full backup copies the entire repository. An incremental backup copies the transaction log, capturing any changes since the last full or incremental backup.

By default, backups are saved as follows:

- Each full backup is stored in its own subdirectory in %SYBASE%\backup.
- Each incremental backup is stored in a file in %SYBASE%\backup\incremental.

SAP recommends that you periodically move backup files to a secondary storage location to prevent the installation directory from becoming too large.

4.2.1 Connect Using the SQL Anywhere Client

Connect to the repository database using the SQL Anywhere client.

Prerequisites

- You have installed the SQL Anywhere 17.0 client.

Context

Procedure

1. Start ASE cockpit. Execute `cockpit.bat` (for Windows) or `cockpit.sh` (for UNIX).
2. Start the SQL Anywhere client.
3. Specify the following credentials:

Field	Value
Authentication	Database
User ID	DBA

Field	Value
Password	Specify the dbpassword you entered during installation or updated using the cockpit.sh (UNIX) or cockpit.bat (Windows) --dbpassword command.
Action	Connect to a running database on another computer
Host	Specify the host or IP address of the machine cockpit is running on.
Port	4638

Related Information

[SAP SQL Anywhere Database Client Download](#)

4.2.2 Performing a Full Repository Database Backup

Take a full backup of the repository database.

Procedure

1. Connect to the repository database. See [Connect Using the SQL Anywhere Client \[page 24\]](#).
2. Execute:

```
BACKUP DATABASE DIRECTORY '<directory_absolute_path>'
```

<directory_absolute_path>

Specifies the absolute path of the backup directory

4.2.3 Performing an Incremental Repository Database Backup

Take an incremental backup of the repository database.

Procedure

1. Connect to the repository database. See [Connect Using the SQL Anywhere Client \[page 24\]](#).

2. Execute:

```
BACKUP DATABASE DIRECTORY '<directory_absolute_path>'
TRANSACTION LOG ONLY TRANSACTION LOG TRUNCATE TRANSACTION LOG RENAME
```

<directory_absolute_path>

Specifies the absolute path of the backup directory

4.2.4 Restoring a Repository Database from Backup

Load backup files into the repository database to revert undesirable changes or to recover from a catastrophic failure.

Procedure

1. Stop the cockpit server. See [Start and Stop the SAP ASE Cockpit Server \[page 12\]](#).
2. Copy the backup images to the database path. By default, this is \$SYBASE/COCKPIT-4/services/Repository/db. If your backup is stored in a different location, adjust the path accordingly.
3. Restart the cockpit server.

4.3 Logging

Error messages from SAP ASE are sent to the user's screen and to the error log file.

For information on logs, see [SAP ASE error logging](#) in the [System Administration Guide: Volume 1](#)

4.4 Configuring Memory Usage

Determine whether you need to configure how much memory the SAP ASE Cockpit uses, and if so which configuration method to use.

Context

It is not usually necessary to configure memory usage for SAP ASE Cockpit. This table lists memory options you can set and circumstances under which you should consider changing them.

Value	Modify When	Guidelines
<p>Set maximum memory available for Cockpit.</p> <p><code>COCKPIT_MEM_MAX</code> – if you are running SAP ASE Cockpit as a UNIX service or starting SAP ASE Cockpit from the command line.</p>	<ul style="list-style-type: none"> You need to prevent SAP ASE Cockpit from using more than a given amount of memory. SAP ASE Cockpit fails to start and may display an error: <code>Could not create the Java Virtual machine.</code> An <code>OutOfMemory</code> error says SAP ASE Cockpit is out of heap space. A warning message about system memory appears during the start process. The machine where SAP ASE Cockpit is installed has less than 4GB of memory. (Starting SAP ASE Cockpit on a machine with less than 4GB of memory triggers the startup warning message about system memory.) 	<p>On machines with less than 4GB of memory, set maximum memory to 256MB or more.</p> <p>Default value: none. (On machines with 4GB or more of memory, maximum memory is set dynamically and is effectively limited only by the amount of system memory available.)</p>
<p>Set initial memory available for Cockpit.</p> <p><code>COCKPIT_MEM_INIT</code> – if you are running SAP ASE Cockpit as a UNIX service or starting SAP ASE Cockpit from the command line.</p>	<p>Increase the initial memory used by Cockpit when the default value is insufficient.</p>	<p>When set, the value for <code>COCKPIT_MEM_INIT</code> overrides the default value.</p> <p>When not set, the default value for initial memory is 128MB.</p>

You can change memory options in three ways:

- For the SAP ASE Cockpit service – modify a file used by the SAP ASE Cockpit service. When you use this method, your changes to the memory options persist—SAP ASE Cockpit uses them every time it starts as a service.
- If `COCKPIT_MEM_INIT` or `COCKPIT_MEM_MAX` are set as environment variables before executing `cockpit.sh` or `cockpitd`, they will be used for setting the memory usage when starting the Cockpit server.

4.4.1 Changing a Memory Option on the Command Line

Before you start SAP ASE Cockpit server from the command line, you can issue a command to change the value of a memory option temporarily.

Context

Changes made using this method last only as long as the current login session. This method is useful for testing new option values.

Procedure

1. If SAP ASE Cockpit server is running, shut it down.
2. Set the environment variable. Specify a size in megabytes, but do not indicate the units in the command.

UNIX example:

```
bash$ export COCKPIT_MEM_MAX=512
```

3. Use the `cockpit` command to start SAP ASE Cockpit server.

Related Information

[Start and Stop the SAP ASE Cockpit Server \[page 12\]](#)

4.4.2 Changing a Memory Option for an SAP ASE Cockpit Windows Service

Add a `jvmopt` command to the `cockpit.properties` file to change a memory option (`-Xmx` or `-XX:MaxPermSize`) for an SAP ASE Cockpit Windows service.

Context

When you use this method to set memory options, your changes are permanent—SAP ASE Cockpit uses them every time it starts as a service.

Procedure

1. If SAP ASE Cockpit server is running, shut it down.
2. Navigate to `%SYBASE%\COCKPIT-4\bin` and open `cockpit.properties`.
3. Add (or modify, if it already exists) a `jvmopt` line specifying the memory size in Java format. Use `m` for megabytes or `g` for gigabytes.

For example:

```
jvmopt=-Xmx512m
```

4. Save the file and start the SAP ASE Cockpit Windows service.

Related Information

[Start and Stop the SAP ASE Cockpit Server \[page 12\]](#)

4.4.3 Changing a Memory Option for an SAP ASE Cockpit UNIX Service

To change a memory setting for an SAP ASE Cockpit UNIX service, add the appropriate environment variable (<COCKPIT_MEM_MAX> or <COCKPIT_MEM_PERM>) to the cockpit.sh script.

Context

When you use this method to set memory options, your changes are permanent—SAP ASE Cockpit uses them every time it starts as a service.

Procedure

1. If the SAP ASE Cockpit is running, shut it down.
2. Navigate to `$SYBASE/COCKPIT-4/bin` and open `cockpitd`.
3. Add the environment variable at the top of the file (after the comments). Specify a size in megabytes but do not indicate the units in the command.

For example:

```
COCKPIT_MEM_MAX=512
export COCKPIT_MEM_MAX
```

4. Save the file and start the SAP ASE Cockpit UNIX service.

Related Information

[Start and Stop the SAP ASE Cockpit Server \[page 12\]](#)

4.5 SAP ASE Cockpit Console

The console is a command-line interface for displaying details about the status of the SAP ASE Cockpit server, its ports, plug-ins, and services.

When you use the cockpit command to start SAP ASE Cockpit, it displays start-up messages and then displays the console prompt.

i Note

The console prompt does not appear if you start SAP ASE Cockpit as a service, if you direct the output of cockpit to a file.

4.5.1 Console Commands

Use the SAP ASE Cockpit console to get status information on SAP ASE Cockpit and its ports, plug-ins, and services.

4.5.1.1 help Command

Display syntax information for one or more SAP ASE Cockpit console commands.

≡ Syntax

```
help [<command_name>]
```

Parameters

<command_name>

(optional) Use with `status`, `info`, or `shutdown`. If you omit `<command_name>`, `help` returns information on all the console commands.

Examples:

Returns information on the `status` command:

```
help status
```

4.5.1.2 info Command

Display information about specified parts of the SAP ASE Cockpit.

If you enter `info` with no parameters, it returns information for every parameter.

≡ Syntax

```
info [-a | --sys]
[-D | --sysprop [<system-property>]]
[-e | --env [<environment-variable>]]
[-h | --help]
[-m | --mem]
[-p | --ports]
[-s | --services]
```

Parameters

-a | --sys

(Optional) List all the services known to SAP ASE Cockpit, indicate whether each service is enabled, and list other services on which each service depends.

-D | --sysprop [<system-property>]

(Optional) Display information about the specified Java system property. Omit the system-property argument to return a list of all Java system properties and their values.

-e | --env [<environment-variable>]

(Optional) List all the environment variables in the SAP ASE Cockpit Java VM process environment. Omit the environment-variable argument to return a list of environment variables and their values.

-h | --help

(Optional) Display information about the `info` command.

-m | --mem

(Optional) Display information about the server's memory resources.

-p | --ports

(Optional) List all the ports on which the cockpit services listen, indicate whether each port is in use, and show the service running on each port.

-s | --services

(Optional) List all cockpit services, indicate whether each service is enabled, and list other services on which each service depends.

4.5.1.3 shutdown command

Stop the cockpit server if it is running.

☞ Syntax

```
shutdown
```

Examples:

Shuts down the cockpit server:

```
shutdown
```

4.5.1.4 status Command

View information on the status of the service components of cockpit.

☞ Syntax

```
status  
[-h | --help]  
[-s | --service [<service-name>]]
```

Parameters:

-h | --help

Display information about the `info` command.

-s | --service [<service-name>]

Display the status of the specified SAP ASE Cockpit service (for example, the Alert service or the Messaging service). Omit the service-name argument to return a list of services.

Examples:

Displays status information on the Repository service:

```
status --service Repository
```

4.6 Setting Server Configuration Parameters

View or change the details of configuration parameters.

4.6.1 Viewing the Parameter Table

View the parameter table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    convert(char(35), a.name) 'Name',
    b.value 'Value',
    a.value 'PendingValue',
    b.defvalue 'DefaultValue',
    b.minimum_value 'MinimumValue',
    b.maximum_value 'MaximumValue',
    1 - convert(bit, a.status & 8) 'Restart',
    b.datatype,
    b.value2 'ValueText', (SELECT m.description
        FROM
            master.dbo.sysmessages m
        WHERE
            m.error = b.message_num
            and m.langid = <lang_id>) 'Explanation'
FROM
    master.dbo.sysconfigures a,
    master.dbo.syscurconfigs b
WHERE
    a.config = b.config
    AND b.display_level <> null
    AND a.parent != 19
```

<lang_id>

Specifies the language id. Use SELECT @@langid. <lang_id> should be set to NULL if the value of @@langid.

Output

Name	Value	PendingValue	DefaultValue	MinimumValue
AQT baseline optgoal	0	0	allrows_mix	0
AQT baseline optlevel	999999	999999	999999	1

Name	Value	PendingValue	DefaultValue	MinimumValue
Application Functionality	0	0	0	0
Backup/Recovery	0	0	0	0
CIPC large message pool size	2048	2048	2048	512
CIPC regular message pool size	32768	32768	32768	2048
Cache Manager	0	0	0	0
Component Integration Services	0	0	0	0
Configuration Options	0	0	0	0
DMA object pool size	16384	16384	16384	2048
DTM Administration	0	0	0	0
Diagnostics	0	0	0	0
Disk I/O	0	0	0	0
Error Log	0	0	0	0
Extended Stored Procedure	0	0	0	0
FIPS login password encryption	0	0	0	0

MaximumValue	Restart	datatype	ValueText	Explanation
0	0	5	allows_mix	It specifies the optimization goal when creating the initial plan and initializing the baseline learning object for auto query tuning.
999999	0	4	999999	It specifies the optimizer version when creating the initial plan and initializing the baseline learning object for auto query tuning.
0	1	0	NULL	NULL
0	1	0	NULL	NULL
2147483647	1	4	2048	CIPC large message pool size : The parameter 'CIPC large message pool size' indicates number of

MaximumValue	Restart	datatype	ValueText	Explanation
				free messages configured in 'CIPC large message pool
2147483647	1	4	32768	CIPC regular message pool size : The parameter 'CIPC regular message pool size' indicates number of free messages in CIPC regular message pool.
0	1	0	NULL	NULL
0	1	0	NULL	NULL
0	1	0	NULL	NULL
2147483647	1	4	16384	DMA object pool size : The parameter 'DMA object pool size' indicates number of DMA messages in CIPC DMA object pool.
0	1	0	NULL	NULL
0	1	0	NULL	NULL
0	1	0	NULL	NULL
0	1	0	NULL	NULL
0	1	0	NULL	NULL
1	1	1	0	Enable FIPS 140-2 certified module for login password encryption. Default is disabled.

4.6.2 Configuring Procedure Cache Size

Set the size of the procedure cache.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_configure](#).

Syntax

```
EXEC sp_configure 'procedure cache size', <size>
```

Output

Parameter	Name	Default	Memory	Used	Config	Value	Run	Value	Unit
procedure	cache	size	14000	35669	15000	15000	memory	pages(2k)	dynamic

Configuration option changed. ASE need not be rebooted since the option is dynamic.

Changing the value of 'procedure cache size' to '15000' increases the amount of memory ASE uses by 2344 K.

5 Security

Manage permissions and security features such as encryption, users, logins, login profiles, roles, and groups.

5.1 Managing Permissions

Grant command permissions or object permissions to or revoke them from users, groups, or roles.

Permission	Grantee or Object	Links
Create Command Permissions	<p>You can grant permission to use these commands to or revoke it from users, groups, or roles:</p> <ul style="list-style-type: none">• <code>create default</code>• <code>create function</code>• <code>create procedure</code>• <code>create role</code>• <code>create table</code>• <code>create view</code><code>create encryption key</code> (shown only if 'enable encrypted columns' configuration is on for server)• <code>create encryption key</code> (shown when the <code>enable encrypted columns</code> configuration parameter is enabled)	<p>Granting and Revoke Command Privileges to a Group [page 47]</p> <p>Granting and Revoking Command Privileges to a Role [page 55]</p> <p>Granting and Revoking Command Privileges to a User [page 60]</p>
Object Permissions – Owners	<p>You can grant object permissions to or revoke them from a user, group, or role.</p>	<p>Granting and Revoking Object Permissions to a Group [page 46]</p> <p>Granting and Revoking Object Permissions to a Role [page 54]</p> <p>Granting and Revoking Object Permissions to a User [page 59]</p>

Permission	Grantee or Object	Links
Object Permissions – Objects	For tables, you can set these permissions:	Granting and Revoking Table Permissions [page 232]
	<ul style="list-style-type: none"> • <code>select</code> • <code>insert</code> • <code>delete</code> • <code>update</code> • <code>references</code> • <code>transfer</code> • <code>truncate table</code> • <code>update statistics</code> • <code>delete statistics</code> • <code>decrypt</code> (shown when the <code>enable encrypted columns</code> configuration parameter is enabled) 	Granting and Revoking Column Permissions [page 232]
	For stored procedures, scalar functions and extended stored procedures, you can grant or revoke execute permission.	Granting and Revoking Stored Procedure Permissions [page 423] Granting and Revoking Scalar Function Permissions [page 451] Granting and Revoking SQLJ Procedure Permissions [page 430] Granting and Revoking Extended Stored Procedure Permissions [page 436]
	For column encryption key and database encryption keys, you can grant or revoke select permission.	Granting or Revoke Encryption Permissions to a Role, User, or Group [page 69].
	For views, you can set these permissions:	Granting and Revoking View Permissions [page 373]
	<ul style="list-style-type: none"> • <code>select</code> • <code>insert</code> • <code>delete</code> • <code>update</code> 	
	For precomputed results, you can grant or revoke execute permission.	Granting and Revoking Permissions on Precomputed Results Sets [page 399]

5.1.1 Enabling Granular Permissions

Granular permissions enable you to grant system privileges; allowing you to construct site-specific roles with privileges to match your requirements, and let you restrict system administrators and database owners from accessing user data.

Prerequisites

- The ASE_SECDIRS license is required.
- You have SSO_ROLE privileges to turn on granular permissions, and the manage security configuration system privilege to turn off granular permissions.
- The system privilege MANAGE SERVER PERMISSIONS is required to grant the following permissions:
 - checkpoint
 - dump database
 - load database
 - online database
 - own database
 - use database
- The system privilege MANAGE SECURITY PERMISSIONS is required to access the sybsecurity database.
- To generate DDL for encryption keys, logins, and roles:
 - You have the SELECT ANY SYSTEM catalog privilege on the master database to generate DDL for logins or roles.
 - For encryption keys, you have SELECT ANY SYSTEM catalog privilege on the database where the encryption key resides.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_configure](#).

Syntax

```
EXEC sp_configure "enable granular permissions", 1
```

5.2 Manage Logins

Each SAP ASE user must have a login account that is identified by a unique login name and a password.

To access a server, users must have a login account with a unique name and password. When a login account is added to one or more servers, the account is given a unique system user ID, which identifies the users regardless of the server being used. Once a login account is created, a user account is created for users to

access individual database. Login profiles can be associated with a login account to manage attributes such as the default database, default language, authentication mechanism, tracking the login, setting inactivity periods, and invoking login scripts.

The options for managing login accounts are:

- Grant roles to logins
- Map users to logins
- Map client users to logins
- Assign login profiles to login accounts
- Lock login accounts
- Expire login accounts
- Set the number of failed logins
- Configure passwords parameters at the server level
- Change the password for a specific login

5.2.1 Creating a Login

Create a login account for each user.

Prerequisites

Ensure that you have the appropriate privileges.

- Granular permissions enabled – you have the `MANAGE ANY LOGIN` privilege to alter login accounts in general.
 - To modify a login account's password, you have the `CHANGE PASSWORD` privilege or are the account owner.
- Granular permissions disabled – you are a user with `SSO_ROLE`. The account owner is allowed to modify the account's password and full name.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create login](#).

Syntax

```
CREATE LOGIN <login_name> WITH PASSWORD <password> DEFAULT DATABASE  
<default_database_name>
```

<login_name>

Specifies the name of the login account to be created; it must start with an alphabetic character and cannot exceed 30 characters in length.

`<default_database_name>` Specifies a database to be the default. Default is `master`.

5.2.2 Modifying Login Properties

Change login properties, such as set password parameters, add users to a login, and grant logins to a role.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter login](#).

Syntax

```
ALTER LOGIN <login_name> MODIFY DEFAULT DATABASE <default_database_name>
```

`<login_name>` Specifies the name of the login account to be created; it must start with an alphabetic character and cannot exceed 30 characters in length.

`<default_database_name>` Specifies a database to be the default. Default is `master`.

5.2.3 Mapping Users to Logins

Map user accounts to logins.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addalias](#).

Syntax

```
EXEC sp_addalias <login_name>, <name_in_db>
```

`<login_name>` Specifies the `master.dbo.syslogins` name of the user who wants an alternate identity in the current database.

`<name_in_db>` Specifies the database user name to alias `<login_name>` to. The name must exist in both `master.dbo.syslogins` and in the `sysusers` table of the current database.

5.2.4 Assigning Roles to a Login Account

Add a role to a login account.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant role](#).

Syntax

```
GRANT ROLE <role_name> TO <login_profile_name>
```

<role_name> Specifies the name of a system- or user-defined role that the system security officer is granting to a user or a role.

<login_profile_name> Identifies the name of the login profile to which you are granting the role.

5.2.5 Configuring Login Password Properties

Manage password properties for login accounts.

Prerequisites

Ensure that you have the appropriate privileges.

- Granular permissions enabled – you have the MANAGE ANY LOGIN privilege to alter login accounts in general.
 - To modify a login account's password, you have the CHANGE PASSWORD privilege or are the account owner.
- Granular permissions disabled – you are a user with SSO_ROLE. The account owner is allowed to modify the account's password and full name.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_passwordpolicy](#).

To set login passwords:

Syntax

```
EXEC sp_passwordpolicy 'set',<policy_option>, <option_value>
```

To expire login passwords:

Syntax

```
EXEC sp_passwordpolicy 'expire login passwords', <login_name>
```

5.2.6 Changing a Login Password

Change passwords and parameters for login accounts.

Prerequisites

Ensure that you have the appropriate privileges.

- Granular permissions enabled – you have the MANAGE ANY LOGIN privilege to alter login accounts in general.
 - To modify a login account's password, you have the CHANGE PASSWORD privilege or are the account owner.
- Granular permissions disabled – you are a user with SSO_ROLE. The account owner is allowed to modify the account's password and full name.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter login](#).

Syntax

```
ALTER LOGIN <login_name> WITH PASSWORD <caller_password> MODIFY PASSWORD  
<new_login_name_password>
```

<caller_password>

Specifies the current password.

`<new_login_name_password>`

Specifies the new password.

5.2.7 Making a Login Profile the Default for a Login

Manage attributes of login accounts by assigning a login profile to an individual login, a subset of logins, or all logins.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter login profile](#).

Syntax

```
ALTER LOGIN PROFILE <login_profile_name> AS DEFAULT
```

5.2.8 Deleting a Login

Delete a login to remove it from the database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop login](#).

Syntax

```
DROP LOGIN <login_name>
```

5.2.9 Generating DDL for a Login

Generate a DDL script for a login.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<host_name>:<port_number> -TL -N<login_name>
```

5.3 Manage Groups

You can grant permissions to groups to access database objects.

The database owner grants and revokes group encryption permissions.

When SAP ASE is configured to restrict decrypt permission, only the system security officer can grant decrypt permission on tables, columns, and views. When restricted decrypt permission is turned off, the system security officer or the database owner can grant decrypt permission.

Command permissions allow the group to execute create commands. Database owners can assign command permissions to groups in the databases they own.

5.3.1 Creating a Group

A group can have members with specific permissions. You grant and revoke permissions and authorities for a group in the same manner as you do for users.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addgroup](#).

Syntax

```
EXEC sp_addgroup <grpname>
```

5.3.2 Adding and Removing Users to a Group

Add and remove users to a group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_changegroup](#).

Syntax

To add a user to a group:

```
EXEC sp_changegroup <grpname>, <username>
```

To remove a user from a group, assign the user to the PUBLIC group

```
EXEC sp_changegroup 'PUBLIC', <username>
```

<grpname> Specifies the name of the group. The group must already exist in the current database. If you use PUBLIC as the **<grpname>**, enclose it in quotes, because it is a keyword.

5.3.3 Granting and Revoking Object Permissions to a Group

Grant and revoke object privileges to a group.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT {ALL [PRIVILEGES] | <permission_list> ON <object_name> TO <grpname>  
[GRANTED BY <grantor>]}
```

To revoke:

```
REVOKE {ALL [PRIVILEGES] | <permission_list> ON <object_name> FROM <grpname>  
[GRANTED BY <grantor>]}
```

all

When used to assign permission to access database objects, **all** specifies that all permissions, except `decryptgrant` **all**.

When granular permissions is not enabled, a system administrator or the database owner can use **grant all** to assign privileges to create database objects (see syntax for "Grants system privileges to execute certain commands"). When used by a system administrator, **grant all** assigns all create privileges (create database,

create default, create procedure, create rule, create table, create function, and create view). When the database owner uses grant all , or executes grant all outside the master database, the SAP ASE server grants all create privileges except create database and prints an informational message.

Granting all create privileges using grant all is not supported when granular permissions is enabled. For more information, see Using Granular Permissions in the Security Administration Guide.

all cannot be used for a grant statement that includes a where clause.

<permission_list> Specifies a list of object access permissions granted. If more than one permission is listed, separate them with commas. The access permissions that can be granted on each type of object include:, that are applicable to the specified object are granted.

**granted by
<grantor>** Indicates the grantor as a user in the database different from the user executing the command.

5.3.4 Granting and Revoke Command Privileges to a Group

Grant or revoke command privileges to a group.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.
- To grant or revoke privileges, set the configuration parameter enable granular permissions to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

```
GRANT {ALL [PRIVILEGES] | <privilege_list> TO <grpname>
      [GRANTED BY <grantor>]
```

```
REVOKE {ALL [PRIVILEGES] | <privilege_list> FROM <grpname>
      [GRANTED BY <grantor>]
```

all When used to assign permission to access database objects, all specifies that all permissions, except decryptgrant all.

When granular permissions is not enabled, a system administrator or the database owner can use `grant all` to assign privileges to create database objects (see syntax for "Grants system privileges to execute certain commands"). When used by a system administrator, `grant all` assigns all create privileges (create database, create default, create procedure, create rule, create table, create function, and create view). When the database owner uses `grant all`, or executes `grant all` outside the master database, the SAP ASE server grants all create privileges except create database and prints an informational message.

Granting all create privileges using `grant all` is not supported when granular permissions is enabled. For more information, see [Using Granular Permissions in the Security Administration Guide](#).

`all` cannot be used for a grant statement that includes a `where` clause.

<code><privilege_list></code>	Specifies a list of system privileges that can be granted. System privileges include server-wide and database-wide privileges.
<code>granted by <grantor></code>	Indicates the grantor as a user in the database different from the user executing the command.

5.3.5 Deleting a Group

Delete a group to remove it from the database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropgroup](#).

Syntax

```
EXEC sp_dropgroup <grpname>
```

5.3.6 Generating DDL for a Group

Generate a DDL script for a group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<host_name>:<port_number> -TGRP -N<object_name>
```

5.4 Manage Roles

Manage permissions to multiple login accounts by creating roles and granting roles to logins.

Ensure that you have the appropriate privileges.

- Granular permissions enabled – you have the MANAGE ROLES privilege.
- Granular permissions disabled – you are a user with SSO_ROLE.

Create roles as a convenient way to grant and revoke permissions to several logins. A role can be granted to a login account or another role.

When creating or modifying roles, these options are available:

- Choose permission access for object types or command type
- Expire passwords
- Set mutually exclusive roles
- Set role hierarchy
- Assign logins to roles
- Set passwords

Permissions

Permissions granted to roles override permissions granted to users or groups. For example, if John is granted the role of system security officer and individual permissions of sales accounts, he can still access the sales accounts if his individual permissions are revoked because his role permissions override his user permissions.

Hierarchical Roles

A system security officer can define role hierarchies such that a role can be assigned to another role. For example, the chief financial officer role might contain both the financial analyst and the salary administrator roles.

Mutually Exclusive Roles

Roles can be defined to be mutually exclusive. The supported exclusive types are:

- Membership – one user cannot be granted two different roles. For example, the system administrator and system security officer roles can be defined as mutually exclusive for membership; that is, one user cannot be granted both roles.
- Activation – one user cannot activate, or enable, two different roles. For example, a user might be granted both the senior auditor and the equipment buyer roles, but is not permitted to have both roles enabled simultaneously.

5.4.1 Creating a Role

Create a user-defined role.

Prerequisites

Ensure that you have the appropriate privileges.

- Granular permissions enabled – you have the MANAGE ROLES privilege.
- Granular permissions disabled – you are a user with SSO_ROLE.

Grant login to one or more roles, or grant roles to other roles.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create role](#).

Syntax

```
CREATE ROLE <role_name> WITH PASSWD "<password>", MIN PASSWD LENGTH  
<length_value>
```

5.4.2 Activating and Deactivating a Role

The system security officer or user assigned the role can determine whether to activate roles granted by default at login.

Roles may or may not be active at login, depending on the default set for a role.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [Roles and set Options](#).

Syntax

```
SET ROLE <role_name> {on | off}
```

5.4.3 Expiring Role Passwords

Expire the password for a role.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_passwordpolicy](#).

Syntax

```
EXEC sp_passwordpolicy 'expire role passwords', <role_name>
```

5.4.4 Restoring System Roles

Return system role privilege to the default system-defined values.

Prerequisites

- With granular permissions enabled, you have `MANAGE SECURITY PERMISSIONS` system privilege to restore system roles and `MANAGE SERVER PERMISSIONS` to restore `SA_ROLE` privileges.

These system roles can be restored: `SA_ROLE`, `SSO_ROLE`, `OPER_ROLE`, `REPLICAION_ROLE`, `KEYCUSTODIAN_ROLE`.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_restore_system_role](#).

Syntax

```
EXEC sp_restore_system_role <role_name>
```

5.4.5 Creating Role Hierarchy

To manage permissions or privileges for one or more logins, you can grant roles hierarchically.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant role](#).

Syntax

```
GRANT ROLE <role_name> TO <rolename>
```

<role_name> Specifies the name of a system- or user-defined role that is being granted to another role.

<rolename> Specifies the name of the role to which you are granting the role.

5.4.6 Assigning Login Profiles to a Role

Assign a login profile to a role.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant role](#).

Syntax

```
GRANT ROLE <role_name> WHERE @@authmech = 'ldap' TO <login_profile_name>
```

5.4.7 Removing Login Profiles Assigned to a Role

Remove login profiles currently assigned to a role.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [revoke role](#).

Syntax

```
REVOKE ROLE <role_name> FROM <login_profile_name>
```

5.4.8 Assigning Logins to a Role

Assign one or more logins to a role.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant role](#).

Syntax

```
GRANT ROLE <role_name> TO <username>
```

5.4.9 Removing Logins Assigned to a Role

Remove login accounts assigned to roles.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [revoke role](#).

Syntax

```
REVOKE ROLE <role_name> FROM <username>
```

5.4.10 Granting and Revoking Object Permissions to a Role

Grant or revoke object permission to a role.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter `enable predicate privileges` to 1.
- To grant or revoke privileges, set the configuration parameter `enable granular permissions` to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT {ALL [PRIVILEGES] | <permission_list> ON <object_name> TO <role>
      [GRANTED BY <grantor>]
```

To revoke:

```
REVOKE {ALL [PRIVILEGES] | <permission_list> ON <object_name> FROM <role>
       [GRANTED BY <grantor>]
```

all

When used to assign permission to access database objects, `all` specifies that all permissions, except `decrypt` grant all.

When granular permissions is not enabled, a system administrator or the database owner can use `grant all` to assign privileges to create database objects (see syntax for "Grants system privileges to execute certain commands"). When used by a system administrator, `grant all` assigns all create privileges (create database, create default, create procedure, create rule, create table, create function, and create view). When the database owner uses `grant all`, or executes `grant all` outside the master database, the SAP ASE server grants all create privileges except create database and prints an informational message.

Granting all create privileges using `grant all` is not supported when granular permissions is enabled. For more information, see [Using Granular Permissions in the Security Administration Guide](#).

`all` cannot be used for a grant statement that includes a `where` clause.

<permission_list>

Specifies a list of object access permissions granted. If more than one permission is listed, separate them with commas. The access permissions that can be granted on each type of object include:., that are applicable to the specified object are granted.

granted by
<grantor>

Indicates the grantor as a user in the database different from the user executing the command.

5.4.11 Granting and Revoking Command Privileges to a Role

Grant or revoke command privileges to a role.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter `enable predicate privileges` to 1.
- To grant or revoke privileges, set the configuration parameter `enable granular permissions` to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT {ALL [PRIVILEGES] | <privilege_list> TO <role>
      [GRANTED BY <grantor>]
```

To revoke:

```
REVOKE {ALL [PRIVILEGES] | <privilege_list> FROM <role>
       [GRANTED BY <grantor>]
```

all

When used to assign permission to access database objects, `all` specifies that all permissions, except `decrypt` grant all.

When granular permissions is not enabled, a system administrator or the database owner can use `grant all` to assign privileges to create database objects (see syntax for "Grants system privileges to execute certain commands"). When used by a system administrator, `grant all` assigns all create privileges (create database, create default, create procedure, create rule, create table, create function, and create view). When the database owner uses `grant all`, or executes `grant all` outside the master database, the SAP ASE server grants all create privileges except create database and prints an informational message.

Granting all create privileges using `grant all` is not supported when granular permissions is enabled. For more information, see [Using Granular Permissions in the Security Administration Guide](#).

`all` cannot be used for a grant statement that includes a where clause.

<code><privilege_list></code>	Specifies a list of system privileges that can be granted. System privileges include server-wide and database-wide privileges.
<code>granted by</code> <code><grantor></code>	Indicates the grantor as a user in the database different from the user executing the command.

5.4.12 Deleting a Role

Delete a role to remove it from the database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop role](#).

Syntax

```
DROP ROLE <role_name>
```

5.4.13 Generating DDL for a Role

Generate a DDL script for a role.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<host_name>:<port_number> -TRO -N<role_name>
```

5.5 Manage Users

Add a new user to a SAP ASE database and manage database object access and ownership of database objects.

5.5.1 Creating a User

Create a new user on a database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_adduser](#).

Syntax

```
EXEC sp_adduser <login_name> , <name_in_db>,<grpname>
```

<login_name>	Specifies the user's name in master.dbo.syslogins.
<name_in_db>	Specifies a new name for the user in the current database.
<grpname>	Specifies an existing group in the database to add the user to.

5.5.2 Transferring Ownership of a Database Object

Change ownership of database objects.

You can also search for referencing objects in the current, or other databases, that will be affected if the selected object is transferred to a different owner. If referencing objects exist, you can generate the SQL scripts to create these objects with the new owner. You can also compare the scripts to create the object with the old and new owners.

To transfer object ownership with referencing objects, first save the script that creates referencing objects with the new owner, then transfer the database object ownership. When the transfer is complete, run the script to modify ownership of the referencing objects.

Prerequisites

- Ownership of dbo-owned objects can be transferred only by users with an SSO_ROLE. Users cannot also have SA_ROLE and users cannot be a database owner (dbo).

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table...modify owner](#).

Syntax

```
ALTER {<object_type> | ALL} <owner>.{<object_name> | *} MODIFY OWNER <name_in_db>
```

- <object_type>**
- table – user tables and proxy tables
 - view – views
 - procedure – stored procedures
 - function – user-defined functions
 - default – defaults defined separately from the creation of tables
 - rule – rules
 - type – user-defined datatypes
 - encryption key – encryption keys
- all** Specifies all permitted object types. When specified as `ALL <owner>.*`, the ownership of all permitted objects owned by the specified owner are transferred. When specified as `all <owner>.<object_name>`, the ownership of permitted objects with name `<object_name>` owned by the specified owner are transferred.
- <owner>** Specifies the current owner of the object, which is determined by the owner's database user ID (`uid`). Specifying `<owner>` is optional when the user is transferring the ownership of objects owned by themselves. Ownership of objects in the `sysobjects` table are associated with the owner's login name and `uid`.
- <object_name>** Specifies the name of the object for which the ownership is to be transferred. An attempt to transfer the ownership of an object with `<object_name>` to the same owner results in an error message.
- *** Specifies all objects owned by `<owner>` and specified by `<object_name>`. When `<object_name>` is `all`, all objects owned by `<owner>` are transferred. When `<owner>` is the database owner, `* granular permissions settings` is not allowed.
- <name_in_db>** Specifies the database user name of the new owner to whom the ownership transferred. The user specified by `<name_in_db>` must be an existing user and cannot be a guest, role, group, or an alias.

5.5.3 Changing Groups for a User

Changes a user's group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_changegroup](#).

Syntax

```
EXEC sp_changegroup <grpname>, <username>
```

5.5.4 Granting and Revoking Object Permissions to a User

Grant or revoke database object permissions to a user.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT {ALL [PRIVILEGES] | <permission_list> ON <object_name> TO <username>  
[WITH GRANT OPTION] [GRANTED BY <grantor>]}
```

To revoke:

```
REVOKE {ALL [PRIVILEGES] | <permission_list> ON <object_name> FROM <username>  
[WITH GRANT OPTION] [GRANTED BY <grantor>]}
```

all

When used to assign permission to access database objects, all specifies that all permissions, except `decrypt` grant all.

When granular permissions is not enabled, a system administrator or the database owner can use grant all to assign privileges to create database objects (see syntax for "Grants system privileges to execute certain commands"). When used by a system administrator, grant all assigns all create privileges (create database, create default, create procedure, create rule, create table, create function, and create view). When the database owner uses grant all, or executes grant all outside the master database, the SAP ASE server grants all create privileges except create database and prints an informational message.

Granting all create privileges using grant all is not supported when granular permissions is enabled. For more information, see Using Granular Permissions in the Security Administration Guide.

all cannot be used for a grant statement that includes a where clause.

<code><permission_list></code>	Specifies a list of object access permissions granted. If more than one permission is listed, separate them with commas. The access permissions that can be granted on each type of object include:, that are applicable to the specified object are granted.
<code>granted by <grantor></code>	Indicates the grantor as a user in the database different from the user executing the command.

5.5.5 Granting and Revoking Command Privileges to a User

Grant or revoke command privileges to a user.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT {ALL [PRIVILEGES] | <privilege_list> TO <username>
      [WITH GRANT OPTION] [GRANTED BY <grantor>]
```

To revoke:

```
REVOKE {ALL [PRIVILEGES] | <privilege_list> FROM <username>
      [WITH GRANT OPTION] [GRANTED BY <grantor>]
```

all	<p>When used to assign permission to access database objects, all specifies that all permissions, except <code>decrypt</code> grant all.</p> <p>When granular permissions is not enabled, a system administrator or the database owner can use <code>grant all</code> to assign privileges to create database objects (see syntax for "Grants system privileges to execute certain commands"). When used by a system administrator, <code>grant all</code> assigns all create privileges (create database, create default, create procedure, create rule, create table, create function, and create view). When the database owner uses <code>grant all</code>, or executes <code>grant all</code> outside the master database, the SAP ASE server grants all create privileges except create database and prints an informational message.</p>
------------	---

Granting all create privileges using grant all is not supported when granular permissions is enabled. For more information, see Using Granular Permissions in the Security Administration Guide.

all cannot be used for a grant statement that includes a where clause.

<privilege_list> Specifies a list of system privileges that can be granted. System privileges include server-wide and database-wide privileges.

granted by <grantor> Indicates the grantor as a user in the database different from the user executing the command.

5.5.6 Deleting a User

Delete a user to remove it from the database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropuser](#).

Syntax

```
exec sp_dropuser <username>
```

5.5.7 Generating DDL for a User

Generate a DDL script for a user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<host_name>:<port_number> -TUSR -  
N<db_name>.<owner_name>
```

5.6 Manage Encryption Keys

You can encrypt database columns using keys that are created with user-defined or login passwords.

Encryption Keys

In each database, you can create a key that encrypts columns. Creating a key on each database minimizes cross-database key integrity problems. Such key problems can happen in distributed systems, particularly when you are dumping and loading, or mounting and unmounting databases.

To create or modify encryption keys:

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the following privilege or privileges based on the encryption key type:
 - column encryption key – CREATE ENCRYPTION KEY or MANAGE COLUMN ENCRYPTION KEY
 - master key – MANAGE MASTER KEY
 - database encryption key – MANAGE DATABASE ENCRYPTION KEY
 - create an encryption key for another user – MANAGE ANY ENCRYPTION KEY
 - Granular permissions disabled – you have the following roles or permissions:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENCRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

If you are a key owner, allow other users to access encryption keys by either:

- Creating an encryption key with a user-defined password and sharing it with each user who accesses key-encrypted data, or
- Giving each user a copy of the base encryption key, and allowing him or her to change the key-copy password.

Encryption Keys with User-Defined Passwords

Using encryption keys with user-defined passwords creates a highly secure system in which even database owners and system administrators cannot access encrypted data. You can also require that the key encryption method itself use a user-defined password.

SAP ASE provides recovery for lost base-key passwords.

When data is encrypted, system security officers, key-custodians, and users with permission to create encryption keys can also create base keys. System security officers can also grant base key creation permission to users who have no other permissions.

The creator of the base key is the "key owner." To control access to encrypted data, only key owners and system security officers can change the base-key password.

Encryption Keys with Login Passwords

To prevent users from having to keep multiple passwords, you can authorize users to access encrypted data using their login password. Using login passwords to access key-encrypted data:

- Gives users access to encrypted data without requiring them to explicitly supply passwords.
- Involves fewer passwords for users to track.
- Reduces the need for the key custodian to replace lost passwords.

Key Copies

Key owners can allow data access to other users by making copies of the base key—called key copies. A key copy is an additional password for the base key that can be changed as soon as it is assigned to a user, or key-copy owner. Only the key-copy owner can change the key-copy password.

You can make key copies for designated users if you are the base-key owner or a system security officer. Key copies of the base key are not new keys themselves; they are additional passwords for the base key. Key-copy assignees should change their user-defined password for as soon as the key copy is assigned to them.

The key copy is encrypted with the login password as soon as the assignee logs in and accesses the key copy.

i Note

The base key can be encrypted by the system encryption password or a user-defined password. Key copies can be encrypted by a login password or by a user-defined password. The recovery key copy can be encrypted only by a user-defined password. Keys that are encrypted with the system encryption password cannot have key copies.

Key recovery requires you to create a special key copy, called the recovery key, that is designated for the recovery of the base key. If you lose your password, use the recovery key to access the base key.

5.6.1 Creating a System Encryption Password

SAP ASE encrypts keys using the Advanced Encryption Standard (AES) algorithm. The system encryption password is encrypted and stored in the database.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the MANAGE DATABASE ENCRYPTION KEY privilege.
 - To modify or create an encryption key for another user – You have the MANAGE ANY ENCRYPTON KEY privilege.
 - Granular permissions disabled – you have the following roles or permission:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

The security of encryption keys might be compromised if the system encryption password is too short or easy to guess.

i Note

Keys that are encrypted using the system encryption password cannot have key copies.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_encryption](#).

Syntax

```
EXEC sp_encryption system_encr_passwd, <password>
```

5.6.2 Modifying a System Encryption Password

Change or delete the system encryption password.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the MANAGE DATABASE ENCRYPTION KEY privilege.
 - To modify or create an encryption key for another user – You have the MANAGE ANY ENCRYPTON KEY privilege.
 - Granular permissions disabled – you have the following roles or permission:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_encryption](#).

Syntax

```
EXEC sp_encryption system_encr_passwd, <newpassword>, <oldpassword>
```

5.6.3 Creating a Master Key

Create the master key for the database.

SAP ASE encrypts keys using the Advanced Encryption Standard (AES) algorithm. The system encryption password is encrypted and stored in the database.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the MANAGE DATABASE ENCRYPTION KEY privilege.

- To modify or create an encryption key for another user – You have the MANAGE ANY ENCRYPTON KEY privilege.
- Granular permissions disabled – you have the following roles or permission:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create encryption key](#).

Syntax

```
CREATE ENCRYPTION KEY master with passwd <password>
```

5.6.4 Modifying a Master Key Password

Modify existing passwords or regenerate the master key.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the MANAGE DATABASE ENCRYPTION KEY privilege.
 - To modify or create an encryption key for another user – You have the MANAGE ANY ENCRYPTON KEY privilege.
 - Granular permissions disabled – you have the following roles or permission:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.
- Use Change Password when a password is compromised.
- Use Regenerate to periodically change key encryption keys (KEKs), which is recommended as good key management. The master or dual-master key is replaced with a new value and all column encryption keys that are encrypted by the master or dual-master keys are reencrypted

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter encryption key](#).

Syntax

```
ALTER ENCRYPTION KEY MASTER WITH PASSWD <oldpasswd> MODIFY ENCRYPTION WITH  
PASSWD <newpasswd>
```

5.6.5 Creating a Column Encryption Key

Create a column encryption key using a specified encryption method.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the MANAGE DATABASE ENCRYPTION KEY privilege.
 - To modify or create an encryption key for another user – You have the MANAGE ANY ENCRYPTON KEY privilege.
 - Granular permissions disabled – you have the following roles or permission:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create encryption key](#).

Syntax

```
CREATE ENCRYPTION KEY <database>.<keyname> FOR AES WITH KEYLENGTH <num_bits>  
PASSWD <password> INIT_VECTOR RANDOM PAD NULL
```

5.6.6 Modifying a Column Encryption Key Password

Change the encryption key, with the option of adding dual control.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the MANAGE DATABASE ENCRYPTION KEY privilege.
 - To modify or create an encryption key for another user – You have the MANAGE ANY ENCRYPTON KEY privilege.
 - Granular permissions disabled – you have the following roles or permission:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter encryption key](#).

Syntax

```
ALTER ENCRYPTION KEY <database>.<keyname> WITH PASSWD <oldpasswd> MODIFY  
ENCRYPTION WITH PASSWD <newpasswd>
```

5.6.7 Creating a Database Encryption Key

Create a database encryption key using a specified encryption method.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the MANAGE DATABASE ENCRYPTION KEY privilege.
 - To modify or create an encryption key for another user – You have the MANAGE ANY ENCRYPTON KEY privilege.

- Granular permissions disabled – you have the following roles or permission:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

Note

Before you create a database encryption key (DEK), create a key encryption key (KEK). The KEK can be a master key or dual master key; these both protect the database encryption key (DEK). See Using Database-Level Master and Dual Master Keys in the Encrypted Columns Users Guide.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create encryption key](#).

Syntax

```
CREATE ENCRYPTION KEY <keyname> FOR DATABASE ENCRYPTION WITH MASTER KEY
```

5.6.8 Granting or Revoke Encryption Permissions to a Role, User, or Group

Grant or revoke permission to access the encryption key.

Prerequisites

You have the MANAGE ANY ENCRYPTION KEY privilege to create an encryption key for another user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

```
GRANT CREATE ENCRYPTION KEY TO {<name_list> | <role_list>} [WITH GRANT OPTION]
```

```
REVOKE CREATE ENCRYPTION KEY FROM {<name_list> | <role_list>}
```

<name_list> Specifies a list of users' and group names, separated by commas.

<code><role_list></code>	Specifies a list of roles — either system-defined or user-defined — to which you are granting the permission.
WITH GRANT OPTION	Allows the users specified in <code><name_list></code> to grant object access permissions to other users. You can grant permissions with grant option only to individual users, not to "public" or to a group or role. Predicated privileges cannot be granted with the WITH GRANT OPTION.

5.6.9 Deleting an Encryption Key

Delete encryption keys.

Prerequisites

- Verify that you have a valid SAP ASE encryption feature license (ASE_ENCRYPTION).
- Set the ENABLE ENCRYPTED COLUMNS parameter to 1.
- Ensure that you have the appropriate privileges.
 - Granular permissions enabled – you have the following privilege or privileges based on the encryption key type:
 - column encryption key – CREATE ENCRYPTION KEY or MANAGE COLUMN ENCRYPTION KEY
 - master key – MANAGE MASTER KEY
 - database encryption key – MANAGE DATABASE ENCRYPTION KEY
 - create an encryption key for another user – MANAGE ANY ENCRYPTION KEY
 - Granular permissions disabled – you have the following roles or permissions:
 - SSO_ROLE, KEYCUSTODIAN_ROLE, or EXECUTE permission on the CREATE ENRYPTION KEY command.
 - SSO_ROLE to create an encryption key for another user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop encryption key](#).

Syntax

```
DROP ENCRYPTION KEY <keyname>
```

5.6.10 Generating DDL for an Encryption Key

Generate a DDL script for encryption keys.

Prerequisites

- When granular permissions is enabled, you have SELECT ANY SYSTEM catalog privilege on the database where the encryption key resides.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<host_name>:<port_number> -TEK -  
N<database_name>.<owner>.<keyname>
```

5.7 Manage Login Profiles

You can manage login accounts with login profiles that define attributes for individual logins, a subset of logins, or all logins.

A login profile is a collection of attributes that are specific to login accounts. You can manage login accounts attributes by creating login profiles and associating the profile with a login account. You can manage attributes for many login accounts by defining a login profile as: the default for all login accounts, a subset of login accounts, or individual login accounts.

When you create or modify a login profile, you can:

- Assign a default database and default language.
- Assign an authentication mechanism..
- Track the last login.
- Define a stale login inactivity period.
- Execute a login script.

Login profiles attributes are associated with login accounts using this precedence:

1. Attribute values from a login profile that is bound to the login.
2. Attribute values from a default login profile.
3. Values that have been specified using `sp_passwordpolicy` under these circumstances:
 - A default login profile does not exist.
 - A login profile has not been defined and bound to the account.

- The login profile is set to be ignored.
4. The default value for the attribute.

5.7.1 Create a Login Profile

Create a login profile to manage attributes of login accounts.

Prerequisites

Ensure that you have the appropriate privileges:

- Granular permissions enabled – you have the MANAGE ANY LOGIN PROFILE profile privilege.
- Granular permissions disabled – you are a user with SSO_ROLE.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create login profile](#).

Syntax

```
CREATE LOGIN PROFILE <login_profile_name>
```

5.7.2 Modifying a Login Profile Properties

Display or modify login profile properties.

Prerequisites

Ensure that you have the appropriate privileges:

- Granular permissions enabled – you have the MANAGE ANY LOGIN PROFILE profile privilege.
- Granular permissions disabled – you are a user with SSO_ROLE.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter login profile](#).

Syntax

```
ALTER LOGIN PROFILE <login_profile_name> DROP AUTO ACTIVATED ROLES {ALL |  
<role_name>[, <role_name> [...]]}
```

5.7.3 Add and Remove Roles to a Login Profile

Add and remove roles to a login profile.

Prerequisites

- To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

```
GRANT ROLE <role_name> TO <login_profile_name>
```

```
REVOKE ROLE <role_name> FROM <login_profile_name>
```

5.7.4 Transferring Login Attributes to a Login Profile

Use attributes of an existing login account to create a login profile.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create login profile](#).

Syntax

```
CREATE LOGIN PROFILE <login_profile_name> WITH ATTRIBUTES FROM <login_name>
```

5.7.5 Deleting a Login Profile

Delete a login profile to drop it from the database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop login profile](#).

Syntax

```
DROP LOGIN PROFILE <login_profile_name>
```

5.7.6 Generating DDL for a Login Profile

Generate a DDL script for a login profile.

Prerequisites

- You have the SELECT ANY SYSTEM catalog privilege on the master database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<host_name>:<port_number> -TLP -  
N<login_profile_name>
```

6 Alerts in SAP ASE

Monitor and manage alerts in a SAP ASE environment.

Alerts are categorized into three key performance areas (KPA): Availability, Performance, and Capacity. Each KPA is composed of several key performance indicators (KPI). Data is collected at defined intervals and alerts triggered when the KPI exceeds the predefined thresholds or state. By default, all alerts are enabled. You can customize the threshold of each alert and disable alerts not required. You cannot add additional alerts to a KPA.

- Data for KPIs is collected using these collection jobs:
- Availability Statistics (APCA default collection)
- Performance Statistics (APCA default collection)
- Capacity Statistics (APCA default collection)

6.1 Monitor Alerts

View information on all alert profiles, thresholds, and the latest alert data for a managed ASE resource.

6.1.1 Viewing Alert Profile

View information on all alert profiles, including the alert profile id (alp_id), which is used to get the corresponding alert thresholds, resource name, and alert KPA/KPI.

Context

The existence of an alert profile means that a given alert is enabled for a given resource. By default, the system enables all of the alerts for a resource. If you disable an alert, the alert profile is deleted.

Procedure

1. Connect to the repository database.
2. Execute:

```
SELECT
    ALP_ID,
    RT_NAME,
    RT_DESC,
```

```

RES_NAME,
RES_DESC,
KPA_TYPE,
KPA_GROUP,
KPA_DESC,
KPA_NAME,
KPA_DISPLAY_NAME,
KPI_GROUP,
KPI_DISPLAY_NAME,
KPI_DESC,
ADF_ALERT_TYPE_NAME,
ADF_THRS_TYPE,
ALERT_TYPE_NAME,
SUPPRESS_PERIOD
FROM
ALT_ALERT_PROF P,
(SELECT * FROM
RES_RESOURCE D R,
RES_RESOURCE_TYPE_M RT
WHERE
R.RES_RT_ID = RT.RT_ID) R,
(SELECT * FROM
ALT_ALERT_DEF D,
MON_KPI_DEF_D I,
MON_KPA_DEF_D A
WHERE
I.KPI_KPA_ID = A.KPA_ID
AND D.ADF_KPI_ID = I.KPI_ID) D,
UAS_ALRT_TYPE UAT
WHERE
P.ALP_RES_ID = R.RES_ID
AND D.ADF_ID = P.ALP_ADF_ID
AND UAT.ALERT_TYPE_ID = P.ALP_ALERT_TYPE_ID
ORDER BY
RT_NAME,
KPA_TYPE,
KPI_GROUP,
ADF_ALERT_TYPE_NAME

```

Results

ALP_ID	RT_NAME	RT_DESC	RES_NAME	RES_DESC	KPA_TYPE	KPA_GROUP	KPA_DESC	KPA_NAME
1	'ASE Server'	'ASEMap ASE Managed Object Type'	'SAPASE'	'ASEMap ASE Managed Object Type'	'AVAILABILI TY'	'ASE Statistics'	'ASE availability statistics (APCA)'	'ASE Availability Statistics (APCA)'
3	'ASE Server'	'ASEMap ASE Managed Object Type'	'SAPASE'	'ASEMap ASE Managed Object Type'	'AVAILABILI TY'	'ASE Statistics'	'ASE availability statistics (APCA)'	'ASE Availability Statistics (APCA)'
7	'ASE Server'	'ASEMap ASE Managed Object Type'	'SAPASE'	'ASEMap ASE Managed Object Type'	'AVAILABILI TY'	'ASE Statistics'	'ASE availability statistics (APCA)'	'ASE Availability Statistics (APCA)'

KPA_DISPLAY_NAME	KPI_GROUP	KPI_DISPLAY_NAME	KPI_DESC	ADF_ALERT_TYPE_NAME
'ASE Availability Statistics (APCA)'	'AVAILABILITY'	'Available Connections'	'Number of concurrent connections available'	'ASE Monitor:ASE Server/15.0.2.0/kpi_ase_connections_available/Available Connections'
'ASE Availability Statistics (APCA)'	'AVAILABILITY'	'Number of Blocked Processes'	'Current number of processes blocked more than 5 seconds'	'ASE Monitor:ASE Server/15.0.2.0/kpi_ase_num_blocked_processes/Number of Blocked Processes'
'ASE Availability Statistics (APCA)'	'AVAILABILITY'	'Number of Database Dump Failures'	'Number of Database Dump Failures'	'ASE Monitor:ASE Server/15.0.2.0/kpi_ase_num_db_dump_failure/Number of Database Dump Failures'

ADF_THRS_TYPE	ALERT_TYPE_NAME	SUPPRESS_PERIOD
'RANGE'	'7F5CA84021A611EA8000D096226C9 805_Available Connections'	600
'RANGE'	'7F5CA84021A611EA8000D096226C9 805_Number of Blocked Processes'	600
'RANGE'	'7F5CA84021A611EA8000D096226C9 805_Number of Database Dump Failures'	600

Related Information

[Connect Using the SQL Anywhere Client \[page 24\]](#)

6.1.2 Viewing Alert Threshold

View information on the alert thresholds for a given alert profile.

Procedure

1. Connect to the repository database.
2. Execute:

```
SELECT * FROM
```

```

ALT_THRESHOLD_PROF P
WHERE
P.TH_PALP_ID = <alert_profile_id>
ORDER BY
P.TH_PSTART

```

Results

THP_ID	THP_ALP_ID	THP_THD_ID	THP_SEV_LEVEL	THP_START	THP_END
262	59	148	'NORMAL'	0.000000	60.000000
259	59	145	'LOW'	61.000000	100.000000
260	59	146	'MEDIUM'	101.000000	1000.000000
261	59	147	'HIGH'	1001.000000	2147483648.000000

Related Information

[Connect Using the SQL Anywhere Client \[page 24\]](#)

6.1.3 Viewing the Latest Alert Data for a Managed ASE Resource

Show alert data for a database.

Procedure

1. Connect to the repository database.
2. Execute:

```

SELECT
  R.RES_NAME,
  R.ALERT_TYPE_NAME,
  R.ALERT_ID,
  R.ALERT_TYPE_ID,
  R.ALERT_STATE,
  R.ALERT_XML,
  R.ESCALATE_CNT,
  R.RECEIVE_DTM,
  R.SUPPRESS_BY_ID
FROM
  (SELECT RANK() OVER (PARTITION BY A.ALERT_TYPE_ID
                     ORDER BY A.ALERT_ID DESC) RANK,

```

```

        A.ALERT_ID,
        A.ALERT_TYPE_ID,
        A.ALERT_STATE,
        A.ALERT_XML,
        A.ESCALATE_CNT,
        A.RECEIVE_DTM,
        A.SUPPRESS_BY_ID,
        T.ALERT_TYPE_NAME,
        RES.RES_NAME
FROM
    UAS_ALRT A,
    UAS_ALERT_TYPE T,
    ALP_ALERT_PROF P,
    RES_RESOURCE_D RES
WHERE
    A.ALERT_TYPE_ID = T.ALERT_TYPE_ID
    AND P.ALP_ALERT_TYPE_ID = T.ALERT_TYPE_ID
    AND P.ALP_RES_ID = RES.RES_ID) R
WHERE
    R.RANK = 1

```

Results

RES_NAME	ALERT_TYPE_NAME	ALERT_ID	ALERT_TYPE_ID	ALERT_STATE
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Thread System CPU Utilization'	37460	171	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Average Blocked Process Wait Time'	69067	169	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Cache Hit Ratio'	41685	168	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Engine CPU Utilization'	88470	165	'SUPPRESSED'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Thread User CPU Utilization'	88472	162	'SUPPRESSED'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Procedure Cache Hit Ratio'	41666	161	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ sp_who Response Time'	88304	158	'CANCEL'

RES_NAME	ALERT_TYPE_NAME	ALERT_ID	ALERT_TYPE_ID	ALERT_STATE
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Number of Packets Received in Network IO'	38462	157	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Number of Packets Sent in Network IO'	38461	156	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Thread Total CPU Utilization'	88471	155	'SUPPRESSED'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Long Running Transaction Execution Time'	69068	152	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Number of Server Transactions'	52905	150	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Statement Cache Hit Ratio'	41669	149	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Configured Resource Number of Open Objects'	88452	134	'ACTIVE'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Configured Resource Number of Open Indexes'	37467	131	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Segment Space Free'	88456	130	'ACTIVE'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Configured Resource Number of Open Partitions'	37466	129	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Device Space Free'	88455	127	'ACTIVE'

RES_NAME	ALERT_TYPE_NAME	ALERT_ID	ALERT_TYPE_ID	ALERT_STATE
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Configured Resource Procedure Cache Size'	88454	122	'ACTIVE'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Configured Resource Number of User Connections'	88453	119	'ACTIVE'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Number of Database Dump Failures'	88483	107	'SUPPRESSED'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Server Device IO Rate'	38456	106	'CANCEL'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Server Percent CPU Utilization'	88482	104	'SUPPRESSED'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Server Availability Status'	107930	102	'SUPPRESSED'
'SAPASE'	'7F5CA84021A611EA8 000D096226C9805_ Available Connections'	88475	101	'ACTIVE'

ALERT_XML	ESCALATE_CNT	RECEIVE_DTM	SUPPRESS_BY_ID
<?xml version="1.0" encoding="UTF-8"? ><PersonalAlerts><Version> 5.0</ Version><PersonalAlert><D ateTimeStamp>2020-03-07 T00:31:10.069-0800</ DateTimeStamp><SourceA pplication>ASE Monitor</ SourceApplication><AlertID >NORMAL</ AlertID><AlertType>7F5CA8 4021A611EA8000	0	2020-03-07 00:31:10.069	
<?xml version="1.0" encoding="UTF-8"? ><PersonalAlerts><Version> 5.0</ Version><PersonalAlert><D ateTimeStamp>2020-03-27 T05:31:10.389-0700</	0	2020-03-27 05:31:10.389	

ALERT_XML	ESCALATE_CNT	RECEIVE_DTM	SUPPRESS_BY_ID
DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000			
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-16T00:06:10.311-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-16 00:06:10.311	
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T12:01:11.131-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	0	2020-03-31 12:01:11.131	88438
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T12:01:12.096-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	0	2020-03-31 12:01:12.096	88440
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-15T21:01:15.397-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-15 21:01:15.397	

ALERT_XML	ESCALATE_CNT	RECEIVE_DTM	SUPPRESS_BY_ID
DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000			
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T11:01:12.019-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-31 11:01:12.019	88290
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-09T21:46:10.946-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-09 21:46:10.946	
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-09T21:46:10.406-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-09 21:46:10.406	
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T12:01:11.546-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-31 12:01:11.546	88439

ALERT_XML	ESCALATE_CNT	RECEIVE_DTM	SUPPRESS_BY_ID
DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0			
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-27T05:31:11.450-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-27 05:31:11.45	
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-23T02:31:09.973-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-23 02:31:09.973	52879
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-15T21:16:09.876-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	0	2020-03-15 21:16:09.876	
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T11:56:08.066-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000	1	2020-03-31 11:56:08.066	

ALERT_XML	ESCALATE_CNT	RECEIVE_DTM	SUPPRESS_BY_ID
<pre> DateTimeStamp><SourceA pplication>ASE Monitor</ SourceApplication><AlertID >HIGH</ AlertID><AlertType>7F5CA8 4021A611EA8000D0 </pre>			
<pre> <?xml version="1.0" encoding="UTF-8"? ><PersonalAlerts><Version> 5.0</ Version><PersonalAlert><D ateTimeStamp>2020-03-07 T00:56:13.088-0800</ DateTimeStamp><SourceA pplication>ASE Monitor</ SourceApplication><AlertID >NORMAL</ AlertID><AlertType>7F5CA8 4021A611EA8000 </pre>	0	2020-03-07 00:56:13.088	
<pre> <?xml version="1.0" encoding="UTF-8"? ><PersonalAlerts><Version> 5.0</ Version><PersonalAlert><D ateTimeStamp>2020-03-31 T11:56:08.161-0700</ DateTimeStamp><SourceA pplication>ASE Monitor</ SourceApplication><AlertID >HIGH</ AlertID><AlertType>7F5CA8 4021A611EA8000D0 </pre>	1	2020-03-31 11:56:08.161	
<pre> <?xml version="1.0" encoding="UTF-8"? ><PersonalAlerts><Version> 5.0</ Version><PersonalAlert><D ateTimeStamp>2020-03-07 T00:56:11.748-0800</ DateTimeStamp><SourceA pplication>ASE Monitor</ SourceApplication><AlertID >NORMAL</ AlertID><AlertType>7F5CA8 4021A611EA8000 </pre>	0	2020-03-07 00:56:11.748	
<pre> <?xml version="1.0" encoding="UTF-8"? ><PersonalAlerts><Version> 5.0</ Version><PersonalAlert><D ateTimeStamp>2020-03-31 T11:56:08.136-0700</ </pre>	1	2020-03-31 11:56:08.136	

ALERT_XML	ESCALATE_CNT	RECEIVE_DTM	SUPPRESS_BY_ID
DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0			
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T11:56:08.108-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	1	2020-03-31 11:56:08.108	
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T11:56:08.096-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	1	2020-03-31 11:56:08.096	
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T12:06:02.468-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	0	2020-03-31 12:06:02.468	88461
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-09T21:40:02.269-0700</DateTimeStamp>	0	2020-03-09 21:40:02.269	38455

ALERT_XML	ESCALATE_CNT	RECEIVE_DTM	SUPPRESS_BY_ID
DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>NORMAL</AlertID><AlertType>7F5CA84021A611EA8000			
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T12:06:02.459-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	0	2020-03-31 12:06:02.459	88466
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-04-14T00:41:02.120-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	0	2020-04-14 00:41:02.12	107921
<?xml version="1.0" encoding="UTF-8"?><PersonalAlerts><Version>5.0</Version><PersonalAlert><DateTimeStamp>2020-03-31T12:03:02.169-0700</DateTimeStamp><SourceApplication>ASE Monitor</SourceApplication><AlertID>HIGH</AlertID><AlertType>7F5CA84021A611EA8000D0	1	2020-03-31 12:03:02.169	

Related Information

[Connect Using the SQL Anywhere Client \[page 24\]](#)

6.2 Managing Alert Notification Settings

Set email address for alert notifications.

Prerequisites

- The email server and port must be configured to enable email notification. See Configuring the E-mail Server.
- A technical user account exists.

with the following statement. Make sure you really want to change:

Procedure

1. Connect to the Cockpit repository database.
2. Take a full backup of Cockpit repository database.
3. To review the current and escalation email address values, execute:

```
SELECT
  P.CHANNEL_SUB_ID CS_ID,
  P.PARAM_NAME EMAIL,
  P.PARAM_VALUE ADDRESS,
  P_E.CHANNEL_SUB_ID CS_ESC_ID,
  P_E.PARAM_NAME ESC_EMAIL,
  P_E.PARAM_VALUE ESC_ADDRESS
FROM
  UAS_CHNL_SUB_PRMS P,
  UAS_CHNL_SUB_PRMS P_E,
  UAS_CHNL_SUB S,
  UAS_CHNL_SUB S_E,
  UAS_CHNL_ESC ESC
WHERE
  ESC.CHANNEL_ID = S.CHANNEL_ID
  AND ESC.ESCALATE_CHNL_ID = S_E.CHANNEL_ID
  AND P.CHANNEL_SUB_ID = S.CHANNEL_SUB_ID
  AND P_E.CHANNEL_SUB_ID = S_E.CHANNEL_SUB_ID
  AND P.PARAM_NAME = 'EMAIL_ADDRESS'
  AND P.PARAM_NAME = P_E.PARAM_NAME
```

Results Set

CS_ID	EMAIL	ADDRESS	CS_ESC_ID	ESC_EMAIL	ESC_ADDRESS
101	EMAIL_ADDRESS S	NAME1@COMPA NY.COM	102	EMAIL_ADDRESS S	NAME1@COMPA NY.COM
103	EMAIL_ADDRESS S	NAME2@COMPA NY.COM	104	EMAIL_ADDRESS S	NAME2@COMPA NY.COM

CS_ID	EMAIL	ADDRESS	CS_ESC_ID	ESC_EMAIL	ESC_ADDRESS
105	EMAIL_ADDRESS S	NAME3@COMPA NY.COM	106	EMAIL_ADDRESS S	NAME3@COMPA NY.COM

4. Set connection authentication to allow modification to the database with the following statement.

```
SET TEMPORARY OPTION
CONNECTION_AUTHENTICATION='Company=SAP;Application=Cockpit;

Signature=000fa55157edb8e14d818eb4fe3db41447146f1571g6eb2a6d2966db36d82642c745
d1d1bde020b45b6'
```

5. Update the settings accordingly. All updates must be within a single transaction. Separate values in `<email_list>` with a comma.

- a. To update an email address, execute:

```
UPDATE UAS_CHNL_SUB_PRMS P
  SET PARAM_VALUE='<email_list>'
  WHERE P.CHANNEL_SUB_ID IN
    (SELECT
      S.CHANNEL_SUB_ID
    FROM
      UAS_CHNL_ESC ESC,
      UAS_CHNL_SUB S
    WHERE
      ESC.CHANNEL_ID = S.CHANNEL_ID)
  AND P.PARAM_NAME = 'EMAIL_ADDRESS'
```

- b. To update escalation email addresses, execute:

```
UPDATE UAS_CHNL_SUB_PRMS P
  SET PARAM_VALUE='<email_list>'
  WHERE P.CHANNEL_SUB_ID IN
    (SELECT
      S.CHANNEL_SUB_ID
    FROM
      UAS_CHNL_ESC ESC,
      UAS_CHNL_SUB S
    WHERE
      ESC.ESCALATE_CHNL_ID = S.CHANNEL_ID)
  AND P.PARAM_NAME = 'EMAIL_ADDRESS'
```

6. Verify the changes using the syntax from step 3 [page 88].
7. Restart Cockpit server.

Related Information

[Configuring the E-mail Server \[page 22\]](#)

[Connect Using the SQL Anywhere Client \[page 24\]](#)

6.3 SAP ASE Alert and Data Collections Summary

Lists and describes alerts you can configure for SAP ASE data collection.

Alerts are based on the same key performance indicators (KPIs) that are collected for APCA monitor displays and the Statistics Chart.

Table 1: Availability (KPI: ASE Availability Statistics)

Name	Unit	Description	Alert
Available Connections	Count	Available connections on the server.	Yes
Number of Blocked Processes	Count	Number of currently blocked processes that have been blocked for more than 5 seconds. The heat chart uses this metric to display server status.	Yes
Number of Database Dump Failures	Count	Number of failed database dumps.	Yes
Number of Suspended Processes	Count	Number of processes that are currently suspended. The heat chart uses this metric to display server status.	Yes
Server Availability Status	Status	Status of the SAP ASE server. Values of most interest are stopped and running.	Yes
Server Device IO Rate	I/O per second	Total number of I/O operations performed by all devices on the server during the current collection cycle.	Yes
Server Percent CPU Utilization	Percentage	Average CPU utilization percentage across all active engines on the server.	Yes

Table 2: HADR Availability (KPI: HADR Availability Statistics - applies only to HADR enabled servers)

Name	Unit	Description	Alert
State of primary Replication Server	Status	Status of the primary Replication server.	Yes
State of Fault Manager	Status	Status of the fault manager: running or stopped.	Yes
State of RMA	Status	Status of the replication management agent (for primary and remote site respectively): running or stopped.	Yes
State of remote Replication Server	Status	Status of the remote Replication server.	Yes
Failover Initiated	Status	Status of an initiated failover.	Yes
Synchronous to Asynchronous Replication	Status	Status of the change from synchronous to asynchronous replication	Yes
Suspended Replication	Status	Status of suspended replication	Yes
Down Replication	Status	Status of replication	Yes
Fault Manager Hibernate	Status	Status of the hibernating Fault Manager	Yes
No Contact for Primary Host Agent	Status	Primary SAP Host Agent cannot be contacted	Yes

Name	Unit	Description	Alert
No Contact for Standby Host Agent	Status	Standby SAP Host Agent cannot be contacted	Yes

Table 3: Performance (KPI: ASE Performance Statistics)

Name	Unit	Description	Alert
Average Blocked Process Wait Time	MS	Average time, in milliseconds, that the current blocked processes have waited.	Yes
Cache Hit Ratio	Percentage	Hit ratio in the data cache during the current collection cycle.	Yes
Device APF Reads	Per second	Rate of asynchronous prefetch read operations per second on the selected device.	No
Device IO Rate	I/O per second	Rate of I/O operations per second on this device.	Yes
Device IO Response Time	MS	Response time, in milliseconds, for I/O operations performed on this device.	Yes
Engine CPU Utilization	Percentage	CPU utilization for this engine as a percentage.	Yes
Long Running Transaction Execution Time	Seconds	Execution time of longest running transaction	Yes
Number of Address Locks	Count	Number of address-level locks server-wide.	Yes
Number of Cache Misses	Count	Number of times that a needed page was not found in a cache and had to be read from disk.	No
Number of Cache Searches	Count	Number of cache searches, including hits and misses for all caches combined.	No
Number of Critical Flags Last Collection	Count	Number of critical flags received during the collection cycle.	Yes
Number of Deadlocks	Count	Number of deadlocks on the server since the most recent execution of the collection.	Yes
Number of Error Flags Last Collection	Count	Number of error flags received during the collection cycle.	Yes
Number of Information Flags Last Collection	Count	Number of information flags received during the collection cycle.	Yes
Number of Locks	Count	Total number of active locks of all types on the server.	Yes
Number of Packets Received in Network IO	Count	Number of packets received during the current collection cycle.	Yes
Number of Packets Sent in Network IO	Count	Number of packets sent during the current collection cycle.	Yes
Number of Page Locks	Count	Number of page-level locks server-wide.	Yes
Number of Row Locks	Count	Number of row-level locks server-wide.	Yes

Name	Unit	Description	Alert
Number of Server Transactions	Count	Total number of transactions during the current collection cycle.	Yes
Number of Table Locks	Count	Number of table-level locks server-wide.	Yes
Number of Warning Flags Last Collection	Count	Number of warning flags received during the collection cycle.	Yes
Procedure Cache Hit Ratio	Percentage	Hit ratio in the procedure cache.	Yes
Statement Cache Hit Ratio	Percentage	Hit ratio in the statement cache during the current collection cycle.	Yes
Segment Space Usage	MB	Change in megabytes in the amount of space used by this segment since the last refresh.	No
sp_who Response Time	MS	Time, in milliseconds, the sp_who stored procedure takes to return a response. sp_who is called each time collection_ase_histmon is executed to collect performance statistics.	Yes
Thread System CPU Utilization	Percentage	CPU utilization percentage in handling system level operations for each thread.	Yes
Thread Total CPU Utilization	Percentage	Total CPU utilization obtained by adding Thread User CPU Utilization and Thread System CPU Utilization.	Yes
Thread User CPU Utilization	Percentage	CPU utilization percentage in handling user committed queries for each thread.	Yes

Table 4: HADR Performance (KPI: HADR Statistics - applies only to HADR enabled servers)

Name	Unit	Description	Alert
Bytes written by RSI thread	KB per minute	Number of bytes written by RSI thread per minute.	No
Bytes Received by EXEC thread	KB per minute	Number of bytes received by EXEC thread per minute.	No
DIST to DSI Latency	MS	Latency time from DIST thread to DSI thread.	No
DSI Commands	Minute	Number of commands per minute in DSI thread.	No
DSI to RDB Latency	MS	Latency time from DSI thread to RDB thread.	No
DSI Transactions	Minute	Number of transaction per minutes in DSI thread.	No
EXEC to DIST Latency	MS	Latency time from EXEC thread to DIST thread.	No
Free Transaction Log Space	Percentage	Percentage of free transaction log space.	No
Log Records processed by Rep Agent	Minute	Number of log records processed by Rep Agent per minute.	No

Name	Unit	Description	Alert
Log Records scanned by Rep Agent	Minute	Number of log records scanned by Rep Agent per minute.	No
PDB to EXEC Latency	MS	Latency time from PDB thread to EXEC thread.	No
Primary ASE Transaction Log Backlog	MB	Primary backlog of the transaction log.	Yes
Percentage of Free Transaction Log Space	Percentage	Percentage of free transaction log space.	Yes
Overall Ticket Latency	MS	The latency times at various points in the path between the primary database and the replicate database.	Yes
Percentage of Device Usage in Remote RS	Percentage	Percentage of device usage in the remote Replication server.	Yes
Percentage of Device Usage in Primary RS	Percentage	Percentage of device usage in the primary Replication server.	Yes
State of a replication path	Status	The state of the replication path between a primary and a replicate database.	Yes
Primary Replication Queue Backlog	MB	Primary Replication Server queue backlog statistics.	Yes
Remote Replication Queue Backlog	MB	Remote Replication Server queue backlog statistics.	Yes

Table 5: Capacity (KPI: ASE Capacity Statistics)

Name	Unit	Description	Alert
Configured Resource Additional Network Memory	Percentage	Maximum size of additional memory that can be used for network packets that are larger than the default packet size.	Yes
Configured Resource Audit Queue Size	Percentage	The audit queue holds audit records generated by user processes until the records can be processed and written to the audit trail.	Yes
Configured Resource Compression Info Pool Size	Percentage	Size of the memory pool used for compression.	Yes
Configured Resource Disk IO Structures	Percentage	Initial number of disk I/O control blocks SAP ASE allocates at start-up.	Yes
Configured Resource Memory Per Worker Process	Percentage	Amount of memory used by worker processes.	Yes
Configured Resource Number of Aux Scan Descriptors	Percentage	Number of auxiliary scan descriptors available in a pool shared by all users on a server.	Yes

Name	Unit	Description	Alert
Configured Resource Number of Locks	Percentage	Total number of available locks for all users on SAP ASE.	Yes
Configured Resource Number of Mailboxes	Percentage	Number of mailbox structures allocated by SAP ASE.	Yes
Configured Resource Number of Messages	Percentage	Number of message structures allocated by SAP ASE	Yes
Configured Resource Number of Open Indexes	Percentage	Maximum number of indexes that can be used simultaneously on SAP ASE.	Yes
Configured Resource Number of Open Objects	Percentage	Maximum number of objects that can be open simultaneously on SAP ASE	Yes
Configured Resource Number of Open Partitions	Percentage	Number of partitions that SAP ASE can access at one time.	Yes
Configured Resource Number of Sort Buffers	Percentage	Amount of memory allocated for buffers used to hold pages read from input tables and perform index merges during sorts.	Yes
Configured Resource Number of User Connections	Percentage	Maximum number of user connections that can simultaneously be connected to SAP ASE.	Yes
Configured Resource Number of Worker Processes	Percentage	Maximum number of worker processes that SAP ASE can use at any one time for all simultaneously running parallel queries.	Yes
Configured Resource Permission Cache Entries	Percentage	Number of cache protectors per task.	Yes
Configured Resource Procedure Cache Size	Percentage	Size of the procedure cache. The procedure cache is used while running stored procedures.	Yes
Device Space Free	Percentage	Total amount of free space, in megabytes, on this device.	Yes
Segment Space Free	Percentage	Amount, in megabytes, of free space in the segment; collected separately for each segment.	Yes
Server tempdb Free Space	Percentage	Amount, in megabytes, of free space in the tempdb database.	Yes

Note

The configured resource alerts are based on the percentage of the configured resource that is in use rather than a fixed quantity of the resource. These alerts can be configured to fire when the percent utilization of

any of the configurable resources exceeds a configured threshold that may affect server capacity or performance. For example, you can configure an alert whenever the number of locks in use is 80% or more of the configured number of locks. This alert definition does not need to be changed if you increase or decrease the number of locks configured on the server.

7 Monitor SAP ASE

You can monitor the availability and performance of a local server.

Once an application is running, the system administrator should monitor performance and may choose to customize and fine-tune the system by changing server configuration parameters.

7.1 Manage Monitor Access Permission

Grant and revoke the MON_ROLE role to a user or role.

The monitoring data in the SAP ASE Cockpit is retrieved from SAP ASE monitoring tables, but access to these tables is restricted to users with the MON_ROLE role. Only users granted this role can execute queries on the monitoring tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant role](#) and [revoke role](#).

Syntax

```
GRANT ROLE MON_ROLE TO { <username> | <rolename> }
```

```
REVOKE ROLE MON_ROLE TO { <username> | <rolename> }
```

7.2 Viewing the Performance Overview

View the server status, details about memory usage, CPU utilization, recent alerts, and performance status.

7.2.1 Viewing Overall Server Status

View server status information including the server's name, software product and version, its hardware platform, and an indication of whether the server is running.

The following syntax provides basic guidance for viewing this information.

Syntax

To display version and platform information, execute:

```
SELECT @@version
```

Output

```
Adaptive Server Enterprise/16.0 <SP#> <PL#>/EBF 28923 SMP/P/x86_64/SLES 11.1/ase160sp04x/3676/64-bit/FBO/Wed Nov 6 09:29:55 2020
```

To verify the server is running ping database:

```
SELECT 1
```

7.2.2 Viewing Engine CPU Utilization

View aggregate CPU utilization for all engines on this server.

Since all I/O for a process goes through one engine, CPU usage is not always evenly distributed across engines.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  idletime = @@idle,
  cputime = (CONVERT(numeric(11), @@idle) + @@io_busy + @@cpu_busy)
```

Output

idletime	cputime
65589621	65697354

7.2.3 Viewing Device IO/Sec

View device I/O per second, aggregated across all devices on this server.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  Reads,
  Writes,
  LogicalName
FROM
  master..monDeviceIO
```

Output

Reads	Writes	LogicalName
260010	1514735	master
2115	6416	mydev
15225	755346	sysprocsdev
0	0	IMDEV
18	0	systemdbdev
542	430002	tempdbdev
5925	762680	sybmgmtdev

7.2.4 Viewing Memory

View memory usage statistics.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  procedureCacheSize =(SELECT value
    FROM master..syscurconfigs
    WHERE config = 146),
  stalls =(SELECT sum(Stalls)
```

```

FROM master..monProcedureCache),
statementCachesize =(SELECT value
FROM master..syscurconfigs
WHERE config = 414),
cachesize =(SELECT sum(b.value) / 1024.0
FROM master.dbo.syscurconfigs b,
master.dbo.sysconfigures a
WHERE b.config = a.config
AND b.comment = a.name
AND a.config = 19),
hitpercent =(SELECT case
WHEN CacheSearches != 0
THEN convert(numeric(4, 1), 100 - (convert(numeric(12, 2),
PhysicalReads) / convert(numeric(12, 2), CacheSearches)) * 100))
ELSE 0
END
FROM master..monDataCache
WHERE PhysicalReads <= CacheSearches
AND CacheName = 'default data cache'),
totalPhysicalMemory =(SELECT value
FROM master..syscurconfigs
WHERE config = 402),
totalLogicalMemory =(SELECT value
FROM master..syscurconfigs
WHERE config = 104)

```

Output

procedureCache Size	stalls	statementCache size	cachesize	hitpercent	totalPhysicalMe mory	totalLogicalMem ory
14000	0	0	19.000000	99.9	185344	140061

7.2.5 Viewing Processes

Views process statistics.

Including:

- Max User Processes – number of processes for which this server is configured.
- High Water Mark (hwp)– highest number of processes that ran concurrently since this server started.
- Active – processes currently running.
- Blocked Processes – processes that are waiting for a resource, or for another process to finish.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
maxprocesses =(SELECT b.value

```

```

        FROM master..sysconfigures a,
             master..syscurconfigs b
        WHERE a.config = b.config
             AND a.config = 103),
activeprocesses = (SELECT count(*)
                   FROM master..monProcessActivity
                   WHERE ServerUserID != 0),
hwp = (SELECT config_admin(22, 103, 3, 0, 'hwm_connections', NULL)),
blockedprocesses = (SELECT count(*)
                   FROM master..monProcess
                   WHERE BlockingSPID != null)

```

Output

maxprocesses	activeprocesses	hwp	blockedprocesses
25	16	25	0

7.2.6 Viewing Details

View information about this server, including number of days it has been running, number of deadlocks, data cache hit rate, procedure cache stalls, page and device sizes, maximum number of online engines, number of open databases, and the dates and times of the server's most recent restart and of the clearing of these counters.

The following syntax provides basic guidance for viewing this information.

Syntax

To display Running days, number of deadlocks, the dates and times of the server's most recent restart and of the clearing of these counters, execute:

```

SELECT
    runningdays = DaysRunning,
    deadlocks = NumDeadlocks,
    startdate = StartDate,
    counterscleared = CountersCleared
FROM
    master..monState

```

Output

runningdays	deadlocks	startdate	counterscleared
77	0	2019-12-18 22:52:50.503	NULL

Syntax

To display the maximum number of online engines, execute:

```
EXEC sp_configure 'number of open databases'
```

Syntax

To display maximum number of open databases, execute:

```
EXEC sp_configure 'max online engines'
```

Syntax

To display data cache hit rate, procedure cache stalls, and device sizes, execute:

```
SELECT
    procedure_cache_stalls=(SELECT sum(Stalls)
        FROM master..monProcedureCache),
    datacache_hitrate=(SELECT CASE WHEN CacheSearches !=0 THEN
        convert(numeric(4,1),100-((convert(numeric(12,2),PhysicalReads)/
convert(numeric(12,2),CacheSearches))*100))
        ELSE 0
        END
        FROM master..monDataCache
    WHERE PhysicalReads <= CacheSearches
        AND CacheName = 'default data cache'),
    devicesize=ltrim(str(sum((1. + (d.high - d.low)) / (1048576. / @@pagesize)),
10,2))
FROM
    master.dbo.sysdevices d
WHERE
    d.cntrltype = 0
```

Output

procedure_cache_stalls	datacache_hitrate	devicesize
0	99.9	584.00

7.2.7 Viewing Configured Resources

View usage statistics for many of the configured resources for this server.

Including:

- Current – amount of this resource the server is currently using.
- Run value – configured maximum for this resource.
- Percentage – percentage of the configured maximum represented by the current use of this resource.
- High Water Mark – maximum amount of this resource that has been used since this server started.

Use the Percentage and High Water Mark columns to identify resources that might be over- or under-configured.

The following syntax provides basic guidance for viewing this information.

Syntax

```
EXEC sp_monitorconfig 'all', null, full
```

Output

Name	Configure Value	System Value	Run Value	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
additional network memory	0	2056192	2056192	2013877	42315	2.06	42315	0	(Null)
audit queue size	100	0	100	100	0	0.00	0	0	(Null)
compression info pool size	4096	0	4096	4096	0	0.00	0	0	(Null)
disk i/o structures	256	0	256	256	0	0.00	56	0	(Null)

Name	Configure Value	System Value	Run Value	Num_free	Num_active	Pct_act	Max_Used	Reuse_cnt	Instance_Name
heap memory per user	4096	563	4096	3985	111	2.71	130	0	(Null)
kernel resource memory	6396	0	6396	4660	1736	27.14	1736	0	(Null)
max cis remote connections	0	100	100	100	0	0.00	0	0	(Null)

7.2.8 Viewing Wait Events

View a list of server-wide wait events, which can be very useful in performance tuning. Information about the wait events includes the number of waits, wait time, average wait time, and wait description.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    waits = sw.Waits,
    waittime = sw.WaitTime,
    avgwaittime = (sw.WaitTime / sw.Waits),
    description = i.Description
FROM
    master..monSysWaits sw,
    master..monWaitEventInfo i
WHERE
    sw.WaitEventID = i.WaitEventID
```

Output

waits	waittime	avgwaittime	description
109463	6567747	59	xact coord: pause during idle loop
269507	147	0	waiting for regular buffer read to complete
2350378	731	0	waiting for buf write to complete before writing

waits	waittime	avgwaittime	description
20	5	0	waiting for MASS to finish writing before changing
36279	4	0	waiting for last i/o on MASS to complete
1	0	0	waiting for write of the last log page to complete
686999	126	0	wait for i/o to finish after writing last log page
114197	6566965	57	checkpoint process idle loop
2669592	19702858	7	hk: pause for some time
13	9	0	wait for flusher to queue full DFLPIECE
70	0	0	waiting for disk buffer manager i/o to complete
40	0	0	waiting for synchronous disk buffer manager i/o
212	0	0	wait for data from client

7.2.9 Viewing Licenses

View information about software licenses on this server or cluster instance.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  licensename = Name,
  licensetype = Type,
  licensequantity = Quantity,
  licensestatus = Status,
  licenseexpiry = LicenseExpiry
FROM
  master..monLicense
```

Output

license name	license type	license quantity	license status	license expiry
ASE_CORE	CPU license	12	expirable	2020-04-01 00:00:00.0
ASE_IMDB	CPU license	12	expirable	2020-04-01 00:00:00.0
ASE_ENCRYPTIO N	CPU license	12	expirable	2020-04-01 00:00:00.0
ASE_RDDB	CPU license	12	expirable	2020-04-01 00:00:00.0

7.3 Monitor Databases

Display database statistics and monitor database activity.

7.3.1 Viewing Monitor Statistics for All Open Databases

View information on statistics for all open databases.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  DBID = s.dbid,
  DBName = s.name,
  m.BackupStartTime,
  BackupInProgress = isnull(m.BackupInProgress, 0),
  LastBackupFailed = isnull(m.LastBackupFailed, 0),
  TransactionLogFull = isnull(m.TransactionLogFull, 0),
  SuspendedProcesses = isnull(m.SuspendedProcesses, 0),
  s.status,
  s.status2,
  s.status3,
  s.status4,
  s.def_remote_loc
FROM
  master..monOpenDatabases m,
  master..sysdatabases s
WHERE
  m.DBID=*s.dbid
```

Output

DBID	DBName	BackupStartTime	BackupInProgress	LastBackupFailed	TransactionLogFull
1	master	(Null)	0	0	0
2	tempdb	(Null)	0	0	0
3	model	(Null)	0	0	0
4	pubs2	(Null)	0	0	0
5	IMDB	(Null)	0	0	0
31513	sybssystemdb	(Null)	0	0	0
31514	sybssystemprocs	(Null)	0	0	0
31515	sybmgmtldb	(Null)	0	0	0

SuspendedProcesses	status	status2	status3	status4	def_remote_loc
0	0	-32768	131072	16384	(Null)
0	12	-32768	131072	540672	(Null)
0	0	-32768	131072	16384	(Null)
0	0	-32767	131072	536887296	(Null)
0	0	-32768	131072	20480	(Null)
0	0	-32768	131072	16384	(Null)
0	8	-32768	131072	16384	(Null)
0	12	-32768	131072	16384	(Null)

7.3.2 Viewing Information About Space Usage

View information about space usage.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT spaceusage (<db_id>)
```

db_id Specify a numeric expression that is the ID for a database. This value is stored in the dbid column of sysdatabases.

Output

reserved pages=2884, used pages=1799, data pages=1665, index pages=306, oam pages=134, allocation units=162, row count=22788, tables=59, LOB pages=16, syslogs pages=0

reserved pages=2884, used pages=1799, data pages=1665, index pages=306, oam pages=134, allocation units=162, row count=22788, tables=59, LOB pages=16, syslogs pages=0

7.3.3 Viewing Combined Data and Log Space

View information about data segments and log segments.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_spaceused](#).

Syntax

```
USE <database_name>  
EXEC sp_spaceused syslogs
```

Output

name	total_pages	free_pages	used_pages	reserved_pages
syslogs	20480	17452	109	0

7.3.4 Viewing Information About Processes Currently Using this Database

View information about processes that are currently using this database, including the process ID, login, host, command, and transaction name.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
```

```

SPID = p.SPID,
Login = p.Login,
ClientHost = pl.ClientHost,
Command = p.Command,
MasterTransactionID = p.MasterTransactionID,
Application = pl.Application
FROM
  master.dbo.monProcess p,
  master.dbo.monProcessLookup pl
WHERE
  DBName = '<database_name>'
  and p.SPID = pl.SPID

```

Output

SPID	Login	ClientHost	Command	MasterTransactionID	Application
23	(Null)	(Null)	MAINTENANCE TOKEN	(Null)	(Null)
27	(Null)	(Null)	MAINTENANCE TOKEN	(Null)	(Null)
31	sa	Inxpw04	AWAITING COMMAND	(Null)	SCC_ASE
32	sa	Inxpw04	SELECT	(Null)	SCC_ASE
36	tech_user	Inxpw04	AWAITING COMMAND	(Null)	SCC_ASE
52	sa	Inxpw04	AWAITING COMMAND	(Null)	ASEAgentPlugin

7.3.5 Viewing Information About Devices on Which This Database Stores Data

View information about devices on which this database stores its data, including the device name, the amount of space used on that device, AND the usage allocation (data or log).

The following syntax provides basic guidance for viewing this information.

For regular databases:

Syntax

```

SELECT
  DeviceName = v.name,

```

```

Allocation = m.description,
SizeMB = sum(convert(float, u.size)*@@maxpagesize / 1048576)
FROM
  master.dbo.sysusages u,
  master.dbo.sysdevices v,
  master.dbo.spt_values b,
  master.dbo.sysmessages m
WHERE
  u.vdevno = v.vdevno
  AND v.status & 2 = 2
  AND u.dbid = db_id('<database_name>')
  AND b.type = 'S'
  AND u.segmap & 7 = b.number
  AND b.msgnum = m.error
  AND m.langid = <lang_id>
GROUP BY
  v.name,
  m.description
ORDER BY
  1

```

lang_id Specify the local language ID of the language currently in use. It can be retrieved from global variable @@langid. If the @@langid is 0 and then need to be replaced as 'NULL'.

Output

DeviceName	AllocationSizeMB	SizeMB
master	data and log	70.000000

For in-memory database:

Syntax

```

SELECT
  DeviceName = v.name,
  Allocation = m.description,
  SizeMB = sum(convert(float, u.size)*@@maxpagesize / 1048576)
FROM
  master.dbo.sysusages u,
  master.dbo.sysdevices v,
  master.dbo.spt_values b,
  master.dbo.sysmessages m
WHERE
  u.vdevno = v.vdevno
  AND u.dbid = db_id('<database_name>')
  AND b.type = 'S'
  AND u.segmap & 7 = b.number
  AND b.msgnum = m.error
  AND m.langid = <lang_id>
GROUP BY
  v.name,
  m.description
ORDER BY
  1

```

lang_id Specify the local language ID of the language currently in use. It can be retrieved from global variable @@langid. If the @@langid is 0 and then need to be replaced as 'NULL'.

Output

DeviceName	Allocation	SizeMB
master	data and log	70.000000
tapedump1	data and log	70.000000
tapedump2	data and log	70.000000

7.3.6 Viewing Information About Segments Used by This Database

View information about segments used by this database, including the segment name, the size of the segment in megabytes, and the amount of free space in the segment.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  SegmentName = s.name,
  SegmentID = s.segment,
  SizeMB = convert(numeric(18, 3), convert(float, sum(u.size)) *
@@maxpagesize / 1048576),
  UnusedMB = convert(numeric(18, 3), (convert(float, sum(u.size)) - sum(
    CASE
      WHEN (convert(float, u.size) > curunreservedpgs(u.dbid,
u.lstart, u.unreservedpgs))
      THEN convert(float, u.size) - curunreservedpgs(u.dbid,
u.lstart, u.unreservedpgs)
      ELSE 0
    END)) * @@maxpagesize / 1048576)
FROM
  <database_name>.dbo.syssegments s,
  master.dbo.sysusages u
WHERE
  1 = 1 and ((u.segmap / ((s.segment & 1) + 1)) / power(2, (s.segment & 30)))
& 1 = 1
  AND u.dbid = db_id('<database_name>')
GROUP BY
  s.name
ORDER BY
  UnusedMB desc
```

Output

SegmentName	SegmentID	SizeMB	UnusedMB
system	0	80.000	68.172
default	1	80.000	68.172
logsegment	2	80.000	68.172

7.3.7 Viewing Indexes in this Database Not Used Since the Last Server Restart

View indexes in this database that have not been used since the server was last restarted.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    IndexName = i.name,
    DBName = convert(char(20), '<database_name>'),
    TableName = convert(char(20), object_name(i.id, db_id('<database_name>')))
from
    master.dbo.monOpenObjectActivity a,
    <database_name>.dbo.sysindexes i
where
    a.ObjectID = i.id
    AND a.IndexID = i.indid
    AND (a.UsedCount = 0 OR a.UsedCount is NULL)
    AND i.indid > 0
    AND object_name(i.id, db_id('<database_name>')) not like 'sys%'
```

Output

IndexName	DBName	TableName
spt_mda_ind	master	spt_mda
spt_ijdbc_mda_ind	master	spt_ijdbc_mda
tspt_jtext	master	spt_jtext

7.3.8 Viewing Information about Tables in this Database Used Since the Last Server Restart

View information on tables used since the last time the server was restarted.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    TableName = object_name(ObjectID, DBID),
    IndexID,
    LogicalReads,
    PhysicalReads,
    Operations,
    LockWaits,
    LockRequests,
    UsedCount,
    ContentionPerQry = case
        WHEN Operations != 0 then convert(numeric(4, 1), LockWaits * 5.0 /
Operations)
        ELSE 0.0
    END,
    ContentionPercent = case
        WHEN LockRequests != 0 then convert(numeric(4, 1), LockWaits * 100.0 /
LockRequests)
        ELSE 0.0
    END
FROM
    master..monOpenObjectActivity
WHERE
    DBName = '<database_name>'
    and IndexID = 0
ORDER BY
    UsedCount desc
```

Output

TableName	IndexID	LogicalReads	PhysicalReads	Operations
monState	0	0	0	177242
monDeviceIO	0	0	0	150776
monOpenDatabases	0	0	0	133233
monThread	0	0	0	97179

LockWaits	LockRequests	UsedCount	ContentionPerQry	ContentionPercent
0	0	177242	0.0	0.0
0	0	150776	0.0	0.0

LockWaits	LockRequests	UsedCount	ContentionPerQry	ContentionPercent
0	0	133233	0.0	0.0
0	0	97179	0.0	0.0

7.3.9 Viewing Database Statistics

View database statistic details such as data and empty page counts, space utilization, and other statistics for a selected database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [optdiag](#).

Syntax

```
$SYBASE/ASE-16_0/bin/optdiag statistics <database_name> -U<user_name> -P<password> -S<server>
```

Output

ASE optdiag/16.0 SP03 PL08/EBF 28923 SMP/P/x86_64/SLES 11.1/ase160sp03pl08x/3676/64-bit/OPT/Wed Nov 6 07:56:19 2019

Adaptive Server Enterprise/16.0 SP03 PL08/EBF 28923 SMP/P/x86_64/SLES 11.1/ase160sp03pl08x/3676/64-bit/FBO/Wed Nov 6 09:29:55 2019

Server name: ""SAPASE""

Specified database: ""pubs2""

Specified table owner: not specified

Specified table: not specified

Specified column: not specified

Table owner: ""dbo""

Table name: ""table1""

Statistics for table

Data page count: 1

Empty data page count: 0

Data row count: 0.0000000000000000

Forwarded row count: 0.0000000000000000
Deleted row count: 0.0000000000000000
Data page CR count: 0.0000000000000000
OAM + allocation page count: 0
First extent data pages: 0
Data row size: 6.0000000000000000
Parallel join degree: 0.0000000000000000
Unused page count: 6
OAM page count: 1
PIOC for large data mass: 0.0000000000000000
PIOC for small data mass: 0.0000000000000000
PIOC for large index mass: 0.0000000000000000
PIOC for small index mass: 0.0000000000000000
Pages in cache(all masses): 0
Table/Index status: 0
Derived statistics:
Data page cluster ratio: 1.0000000000000000
Space utilization: 0.0000000000000000
Large I/O efficiency: 1.0000000000000000
PIO compensation Adjustment is DEFAULT.
No statistics for remaining columns: ""i""
(default values used)
Optdiag succeeded.

7.4 Monitor Devices

Display information about all devices that store databases server. A device can be an entire disk drive, or any part of a disk or file system.

7.4.1 Viewing Device Statistics and Determine I/O Response Time

View information on device statistics and determine how long it takes a device to respond to I/O requests and what its I/O rate is.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    LogicalName = d.name,
    PhysicalName = d.phyname,
    Status2 = d.status2,
    Size = ltrim( str( (1.+ (d.high - d.low)) / (1048576./ @@pagesize),10, 2)),
    UsedSize = (SELECT ( (sum(u.size) / 512.0) *(@@maxpagesize / @@pagesize) )
                FROM
                    master.dbo.sysusages u
                WHERE
                    u.vdevno = d.vdevno ),
    Reads, Writes, APFReads, IOTime, DevSemaphoreRequests,
    DevSemaphoreWaits
FROM
    master.dbo.sysdevices d,
    master..monDeviceIO m
WHERE
    d.cntrltype = 0
    AND m.LogicalName =* d.name
```

Output

LogicalName	PhysicalName	Status2	Size	UsedSize	Reads
IMDEV	IMCache	8	6.00	6.000000	0
master	/Inxpw04_tst1/ liangm/ASE160/ data/master.dat	0	100.00	98.000000	222025

LogicalName	PhysicalName	Status2	Size	UsedSize	Reads
mydev	/Inxpw04_tst1/ liangm/ASE160/ data/mydev.dat	65536	100.00	6.000000	1544
sybmgmtdev	/Inxpw04_tst1/ liangm/ASE160/ data/ sybmgmtdb.dat	1	76.00	76.000000	4821
sysprocsdev	/Inxpw04_tst1/ liangm/ASE160/ data/sysprocs.dat	1	196.00	196.000000	9796
systemdbdev	/Inxpw04_tst1/ liangm/ASE160/ data/sybsysdb.dat	1	6.00	6.000000	18
tempdbdev	/Inxpw04_tst1/ liangm/ASE160/ data/ tempdbdev.dat	1	100.00	100.000000	343

Writes	APFReads	IOTime	DevSemaphoreRequests	DevSemaphoreWaits
0	0	0	0	0
1462733	1231	553510	3	0
76	20	442	0	0
741214	1317	146546	0	0
732996	4785	102138	2	0
0	0	56	2	0
395945	161	409513	2	0

7.4.2 Viewing Used and Unused Space

View space usage on the selected device. Includes used and unused space, in megabytes, and as percentages of all the available space on the device. The title above the chart indicates the total available space.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  LogicalName=name,
  Size=ltrim(str((1. + (d.high - d.low)) / (1048576. / @@pagesize),10,2)),
  UsedSize=(SELECT ((sum(u.size)/512.0) * (@@maxpagesize/@@pagesize))
    FROM .dbo.sysusages u
```

```

WHERE u.vdevno=d.vdevno)
FROM
  master.dbo.sysdevices d
WHERE
  d.cntrltype=0

```

Output

LogicalName	Size	UsedSize
master	100.00	98.000000
sysprocsdev	196.00	196.000000
systemdbdev	6.00	6.000000
tempdbdev	100.00	100.000000
sybmgmtdev	76.00	76.000000
mydev	100.00	6.000000
IMDEV	6.00	6.000000

7.4.3 Viewing Device IO Response Time and Device APF Reads/Sec

View the response time, in milliseconds, for I/O operations performed on the selected device the rate of asynchronous prefetch read operations, per second, on the selected device.

APF reads indicate that table scans are taking place.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  Reads,
  Writes,
  IOTime,
  APFReads
FROM
  master.monDeviceIO
WHERE
  LogicalName = '<logical name>'

```

Output

Reads	Writes	IOTime	APFReads
222025	1462771	553984	1231

7.4.4 Viewing the Logical Name List

View a list of devices.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  name
FROM
  master..sysdevices
WHERE
  cntrltype = 0
```

Output

name

master

sysprocsdev

systemdbdev

tempdbdev

sybmgmtdev

mydev

IMDEV

7.5 Monitor Engines

Display the current condition of the database and track performance as conditions change.

7.5.1 Viewing All Engines

View the operating system process identifier (OS PID) for each engine.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  EngineNumber,
  SystemCPUtime,
  UserCPUtime,
  IdleCPUtime,
  CPUtime = (SystemCPUtime + UserCPUtime + IdleCPUtime),
  Status,
  StartTime,
  Connections,
  OSPID,
  DiskIOChecks,
  DiskIOPolled,
  DiskIOCompleted,
  HkgcMaxQSize,
  HkgcPendingItems,
  HkgcHWMItems,
  HkgcOverflows
FROM
  master..monEngine
```

Output

EngineNumber	SystemCPUtime	UserCPUtime	IdleCPUtime	CPUtime	Status
0	306	9765	6564307	6574378	online

StartTime	Connections	OSPID	DiskIOChecks	DiskIOPolled
Dec 18 2019 10:52PM	20	8171	348166883	23957006

DiskIOCompleted	HkgcMaxQSize	HkgcPendingItems	HkgcHWMItems	HkgcOverflows
3758654	11150	0	9	0

7.5.2 Viewing Engine Utilization - IO Processing & Garbage Collection

View I/O processing, number of connections, and garbage collector's maximum queue size and counts.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  EngineNumber,
  SystemCPUTime,
  UserCPUTime,
  IdleCPUTime,
  CPUTime = (SystemCPUTime + UserCPUTime + IdleCPUTime),
  Status,
  StartTime,
  Connections,
  OSPID,
  DiskIOChecks,
  DiskIOPolled,
  DiskIOCompleted,
  HkgcMaxQSize,
  HkgcPendingItems,
  HkgcHWMItems,
  HkgcOverflows
FROM
  master..monEngine
WHERE
  EngineNumber = <engine_number>
```

Output

EngineNumber	SystemCPUTime	UserCPUTime	IdleCPUTime	CPUTime	Status
306	9770	6564887	6574963	online	Dec 18 2019 10:52PM

StartTime	Connections	OSPID	DiskIOChecks	DiskIOPolled	DiskIOCompleted
20	8171	0	348207842	23959460	3759130

StartTime	HkgcMaxQSize	HkgcPendingItems	HkgcHWMItems	HkgcOverflows
20	11150	0	9	0

7.5.3 Viewing Engine Utilization - Engine CPU Utilization

View CPU utilization.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SEKLECT
  BusyCPUTime = SystemCPUTime + UserCPUTime,
  TotalCPUTime = SystemCPUTime + UserCPUTime + IdleCPUTime
FROM
  master..monEngine
WHERE
  EngineNumber = <engine_number>
```

Output

BusyCPUTime	TotalCPUTime
10078	6575158

7.6 Monitor Threads

Monitor threads for resource planning.

SAP ASE assigns tasks to thread pools, and all thread pools have threads. Each thread pool includes a scheduler that assigns tasks to threads.

SAP ASE contains both system-created and user-created thread pools. All system-defined thread pools start with the syb_ prefix (for example syb_default_pool). Names for user-defined thread pools cannot start with the syb_ prefix and must follow the naming conventions for objects.

7.6.1 Viewing Thread Pools

View information on thread pools for the current database.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT * FROM master..monThread
```

Output

InstanceID	ThreadID	KTID	OSThreadID	AltOSThreadID	ThreadPoolID
0	6	675283078	14073726680038 4	8171	1
0	8	917511	14073720801049 6	8179	2
0	7	786438	140737216444160	8178	2
0	1	655365	14073730876390 4	8165	2
0	5	131073	140737275193088	8170	3
0	4	262146	140737283585792	8169	3
0	3	524292	140737291978496	8168	3
0	2	393219	140737300371200	8167	3

TaskRuns	TotalTicks	IdleTicks	SleepTicks	BusyTicks	UserTime
28646908	65702977	65593220	0	109757	16992880
1	65702977	65702959	0	18	84150
1	65702977	65697055	0	5922	381770
1	65702977	65694217	0	8760	939740
1	65702977	65702778	0	199	13870
1	65702977	65702729	0	248	15680
1	65702977	65702754	0	223	17130
1	65702977	65702751	0	226	16990

SystemTime	MinorFaults	MajorFaults	VoluntaryCtxtSwitches	NonVoluntaryCtxtSwitches
3903030	261841	1637	356724435	2308806
138710	10	0	9809968	621
457010	12	8	9435868	38416
902720	5	0	65702980	2
10520	1039	5	1400	2339
10360	1142	11	1345	2503

SystemTime	MinorFaults	MajorFaults	VoluntaryCtxtSwitches	NonVoluntaryCtxtSwitches
10460	980	0	1342	2668
10830	975	0	1421	2773

State	ThreadPoolName	ThreadAffinity
ACTIVE	syb_default_pool	NULL
IDLE	syb_system_pool	NULL
IDLE	syb_system_pool	NULL
IDLE	syb_system_pool	NULL
IDLE	syb_blocking_pool	NULL
IDLE	syb_blocking_pool	NULL
IDLE	syb_blocking_pool	NULL
IDLE	syb_blocking_pool	NULL

7.6.2 Viewing Thread Pool Details

View information about affinity and number of ticks in the selected thread and operating system context switches with the current thread.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT * FROM master..monThread WHERE ThreadID = <thread_id>
```

Output

InstanceID	ThreadID	KTID	OSThreadID	AltOSThreadID	ThreadPoolID
0	6	675283078	14073726680038 4	8171	1

TaskRuns	TotalTicks	IdleTicks	SleepTicks	BusyTicks	UserTime
28657949	65720687	65610756	0	109931	17012970

SystemTime	MinorFaults	MajorFaults	VoluntaryCtxt Switches	NonVoluntary CtxtSwitches	State	ThreadPoolName	ThreadAffinity
3904480	263071	1653	356828848	2312422	ACTIVE	syb_default_ pool	NULL
ThreadPoolName		ThreadAffinity					
syb_default_pool		NULL					

7.6.3 Viewing Thread Pool CPU Utilization

View information on thread pool CPU utilization.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    UserTime,
    SystemTime
FROM
    master..monThread
WHERE
    ThreadID = <thread_id>
```

Output

UserTime	SystemTime
17027280	3905550

7.6.4 Viewing Thread Pool Tasks

View information on thread pool tasks.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
```

```

Name,
KTID,
ThreadPoolName
FROM
master..monTask
WHERE
ThreadID = <thread_id>

```

Output

Name	KTID	ThreadPoolName
KPID 65537	1048584	syb_default_pool
KPID 327683	1310730	syb_default_pool
KPID 458756	1441803	syb_default_pool
KPID 589829	1572876	syb_default_pool
KPID 720902	1703949	syb_default_pool
KPID 851975	1835022	syb_default_pool
KPID 983048	1966095	syb_default_pool
KPID 1114121	2097168	syb_default_pool
KPID 1245194	2228241	syb_default_pool
KPID 1376267	2359314	syb_default_pool
KPID 1507340	2490387	syb_default_pool
KPID 1638413	2621460	syb_default_pool
KPID 1769486	2752533	syb_default_pool
KPID 693043289	697040920	syb_default_pool
KPID 652935254	656932901	syb_default_pool
KPID 650313834	654311467	syb_default_pool
KPID 5308457	6291504	syb_default_pool
KPID 5439530	6422577	syb_default_pool
KPID 5570603	6553650	syb_default_pool
KPID 5701676	6684723	syb_default_pool
KPID 5832749	6815796	syb_default_pool
KPID 5963822	6946869	syb_default_pool
KPID 6094895	7077942	syb_default_pool
KPID 6225968	7209015	syb_default_pool
KPID 6357041	7340088	syb_default_pool
KPID 97189928	99352634	syb_default_pool
KPID 728563794	732561470	syb_default_pool

Name	KTID	ThreadPoolName
KPID 653066342	657064002	syb_default_pool
KPID 381091907	383647829	syb_default_pool
KPID 765657142	769654875	syb_default_pool
KPID 765919319	769917022	syb_default_pool
KPID 692912150	696909948	syb_default_pool
KPID 671285277	675283078	syb_default_pool
KPID 765788210	769785999	syb_default_pool
KPID 96534588	98697367	syb_default_pool

7.7 Monitor Processes

Identify resource-intensive process, blocking process, the lead blocker in a chain, and wait events and SQL queries associated with a process.

7.7.1 Identifying Resource-Intensive Processes

Find the user processes that are consuming the most system resources on the selected SAP ASE server.

You can choose a system resource (CPU, disk I/O, incoming network traffic, or outgoing network traffic) and display information about the user processes that are using the chosen resource most intensively.

7.7.1.1 CPU Cumulative

Show cumulative CPU activity since the server started or the counter wrapped.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    spid = SPID
FROM
    master..monProcessActivity
WHERE
    ServerUserID != 0
ORDER BY
    CPUTime desc
```

Output

spid
3465
2456
1789

7.7.1.2 Disk I/O Cumulative

Show cumulative disk I/O since the server started or the counter wrapped.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    spid = SPID
FROM
    master..monProcessActivity
WHERE
    ServerUserID != 0
ORDER BY
    (PhysicalWrites + PhysicalReads) desc
```

Output

spid
3465
2456
1789

7.7.1.3 Incoming Network Traffic Cumulative

Show cumulative incoming network traffic since the server started or the counter wrapped.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    spid = IO.SPID
FROM
    master..monProcessNetIO IO,
    master..monProcessActivity PA
WHERE
    PA.ServerUserID != 0
    and PA.SPID = IO.SPID
    and PA.KPID = IO.KPID
ORDER BY
    IO.PacketsReceived desc
```

Output

spid

3465

2456

1789

7.7.1.4 Outgoing Network Traffic Cumulative

Show cumulative outgoing network traffic since the server started or the counter wrapped.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    spid = IO.SPID
FROM
    master..monProcessNetIO IO,
    master..monProcessActivity PA
WHERE
    PA.ServerUserID != 0
    AND PA.SPID = IO.SPID
    AND PA.KPID = IO.KPID
ORDER BY
    IO.PacketsSent desc
```

Output

spid

3465

2456

1789

7.7.1.5 All Processes

Show all processes

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  spid = P.SPID,
  familyID = P.FamilyID,
  kpid = P.KPID,
  status = SP.status,
  login = P.Login,
  command = P.Command,
  dbname = P.DBName,
  blockingspid = P.BlockingSPID,
  timeblocked = L.WaitTime,
  clienthost = PL.ClientHost,
  program = P.Application,
  memusagekb = PA.MemUsageKB,
  priority = P.Priority,
  clientip = PL.ClientIP,
  enginenum = P.EngineNumber,
  enginegroupname = P.EngineGroupName,
  executionclass = P.ExecutionClass,
  mastertransactionid = P.MasterTransactionID,
  transactionname = T.xactname,
  transactiondate = T.starttime,
  physicalreads = PA.PhysicalReads,
  logicalreads = PA.LogicalReads,
  physicalwrites = PA.PhysicalWrites,
  cputime = PA.CPUTime,
  loggedindatetime = SP.loggedindatetime,
  networkpacketsize = PIO.NetworkPacketSize,
  packetsent = PIO.PacketsSent,
  packetsreceived = PIO.PacketsReceived,
  bytesent = PIO.BytesSent,
  bytesreceived = PIO.BytesReceived,
  runphysicalreads = PS.PhysicalReads,
  runlogicalreads = PS.LogicalReads,
  pagesmodified = PS.PagesModified,
  starttime = PS.StartTime
FROM
  master..monProcess P,
  master..dbo.sysprocesses SP,
```

```

master..monProcessLookup PL,
master..monProcessActivity PA,
master..monProcessNetIO PIO,
master..monProcessStatement PS,
master..systransactions T,
master..monLocks L
where
  P.SPID *= SP.spid
  AND P.KPID *= SP.kpid
  AND P.SPID *= PL.SPID
  AND P.KPID *= PL.KPID
  AND P.SPID *= PA.SPID
  AND P.KPID *= PA.KPID
  AND P.SPID *= PIO.SPID
  AND P.KPID *= PIO.KPID
  AND P.SPID *= PS.SPID
  AND P.KPID *= PS.KPID
  AND P.SPID *= T.spid
  AND L.LockState='Requested'
  AND P.SPID *= L.SPID
  AND P.KPID *= L.KPID

```

Output

SPID	FamilyID	KPID	status	Login	Command	DBName
1	<<null>>	327683	sleeping	<null>	MEMORY TUNE	<null>
3	<null>	458756	sleeping	<null>	DEADLOCK TUNE	master
4	<null>	589829	sleeping	<null>	SHUTDOWN HANDLER	master
5	<null>	720902	sleeping	<null>	KPP HANDLER	master
6	<null>	851975	sleeping	probe	ASTC HANDLER	master
7	<null>	983048	sleeping	<null>	CHECKPOINT SLEEP	master
8	<null>	1114121	sleeping	<null>	HK WASH	master
9	<null>	1245194	sleeping	<null>	HK GC	master
10	<null>	1376267	sleeping	<null>	HK CHORES	master
11	<null>	1507340	sleeping	<null>	PORT MANAGER	master
12	<null>	1638413	sleeping	<null>	NETWORK HANDLER	<null>
13	<null>	1769486	sleeping	<null>	JOB SCHEDULER	sybmgmtdb
14	<null>	1900559	sleeping	<null>	LICENSE HEARTBEAT	master
15	<null>	2031632	recv sleep	jstask	AWAITING COMMAND	sybmgmtdb

SPID	FamilyID	KPID	status	Login	Command	DBName
16	<null>	2162705	recv sleep	jstask	AWAITING COMMAND	sybmgmtdb
17	<null>	2293778	recv sleep	jstask	AWAITING COMMAND	sybmgmtdb
18	<null>	2424851	recv sleep	jstask	AWAITING COMMAND	sybmgmtdb

BlockingSPID	WaitTime	ClientHost	Application	MemUsageKB	Priority	ClientIP
<null>	<null>	<null>	<null>	0	0	<null>
<null>	<null>	<null>	<null>	0	2	<null>
<null>	<null>	<null>	<null>	0	5	<null>
<null>	<null>	<null>	<null>	0	6	<null>
<null>	<null>	<null>	<astc>	0	5	<null>
<null>	<null>	<null>	<null>	0	5	<null>
<null>	<null>	<null>	<null>	0	7	<null>
<null>	<null>	<null>	<null>	0	5	<null>
<null>	<null>	<null>	<null>	0	7	<null>
<null>	<null>	<null>	<null>	0	0	<null>
<null>	<null>	<null>	<null>	0	0	<null>
<null>	<null>	<null>	<null>	0	5	<null>
<null>	<null>	<null>	<null>	0	5	<null>
0	<null>	<null>	<null>	<null>	<null>	0
<null>	<null>	<null>	JS Agent	4	5	10.173.160.47
<null>	<null>	<null>	JS Agent	4	5	10.173.160.47
<null>	<null>	<null>	JS Agent	4	5	10.173.160.47
<null>	<null>	<null>	JS Agent	4	5	10.173.160.47

EngineNumber	EngineGroupName	ExecutionClass	MasterTransactionID	xactname
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>

EngineNumber	EngineGroupName	ExecutionClass	MasterTransactionID	xactname
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	<null>	<null>	<null>	<null>
0	syb_default_pool	EC2	<null>	<null>
0	syb_default_pool	EC2	<null>	<null>
0	syb_default_pool	EC2	<null>	<null>
0	syb_default_pool	EC2	<null>	<null>

transactiondate	PhysicalReads	LogicalReads	PhysicalWrites	CPUTime	loggedindatetime
<null>	0	0	0	0	2020-03-09 21:38:23.663
<null>	0	0	0	0	2020-03-09 21:38:24.663
<null>	0	0	0	0	2020-03-09 21:38:24.663
<null>	0	0	0	29100	2020-03-09 21:38:24.663
<null>	0	0	0	0	2020-03-09 21:38:25.663
<null>	0	0	43261	0	2020-03-09 21:38:25.663
<null>	0	0	29619	0	2020-03-09 21:38:25.663
<null>	0	0	0	0	2020-03-09 21:38:25.663
<null>	0	0	6	0	2020-03-09 21:38:25.663
<null>	0	0	0	0	2020-03-09 21:38:26.663
<null>	0	0	0	1800	2020-03-09 21:38:26.663
<null>	11	284	13	100	2020-03-09 21:38:28.663
<null>	0	0	0	0	2020-03-09 21:38:29.663
<null>	0	33	2	200	2020-03-09 21:38:29.663
<null>	1	33	3	0	2020-03-09 21:38:30.663
<null>	0	33	3	0	2020-03-09 21:38:30.663

transactiondate	PhysicalReads	LogicalReads	PhysicalWrites	CPUTime	loggedindatetime
		25	2	0	2020-03-09 21:38:30.663

NetworkPacketSize					
ize	PacketsSent	PacketsReceived	BytesSent	BytesReceived	PhysicalReads
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	0	0	0	0	<null>
2048	7	8	3982	1730	<null>
2048	7	8	3937	1653	<null>
2048	7	8	3910	1651	<null>
2048	7	8	4684	1650	<null>

LogicalReads	PagesModified	StartTime
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>

LogicalReads	PagesModified	StartTime
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>
<null>	<null>	<null>

7.7.2 Identifying and Terminating Blocked and Blocking Processes

Find user processes that are blocked and the processes that are blocking them.

7.7.2.1 Blocking Processes

Find user processes that are blocking them.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    spid = SPID,
    blockingspid = BlockingSPID
FROM
    master..monProcess
WHERE
    BlockingSPID != null
```

Output

spid	blockingspid
101	277
320	436

7.7.2.2 Blocked Processes

Find user processes that are blocked.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    spid = P.SPID,
    login = P.Login,
    host = S.hostname
FROM
    master..monProcess P,
    master..sysprocesses S
WHERE
    P.BlockingSPID = <blocking_spid>
    AND P.SPID = S.spid
    AND P.KPID = S.kpid
```

Output

spid	login	host
101	usr01	pc01

7.7.2.3 Terminate process

Terminate user processes that are blocked or are blocking them.

The following syntax provides basic guidance for viewing this information.

Syntax

```
kill <process_id>
```

Output

kill 277

7.7.3 Viewing Wait Events Associated with a Process

View a list of all events a for which a process is waiting. Wait events are internal states that represent conditions that cause a process to stop.

The following syntax provides basic guidance for viewing this information.

Syntax

Sample Code

```
SELECT
    waits = PW.Waits,
    waittime = PW.WaitTime,
    description = I.Description,
    avgwaittime = case
        WHEN PW.Waits != 0 then ((PW.WaitTime) / PW.Waits)
        ELSE 0
    END
FROM
    master..monWaitEventInfo I,
    master..monProcess P,
    master..monProcessWaits PW
WHERE
    PW.KPID = P.KPID
    AND PW.SPID = P.SPID
    AND I.WaitEventID = PW.WaitEventID
    AND P.SPID = <process_id>
```

Output

waits	waittime	description	avgwaittime
6427	46	waiting on run queue after sleep	0
6427	131621026	Bucket pool manager consolidator sleep	20479

7.7.4 Viewing the SQL Query Associated with a Process

View information on the SQL statement and query plan for a user process.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    sqltext = SQLText
FROM
    master..monProcessSQLText
WHERE
    SPID = <process_id>
```

Output

sqltext

```
select name, id from mytable
```

7.8 Monitor Segments

Monitor the segments used by the server. View the name of the database that uses the segment, the database's size and unused space on the segment, and the number of thresholds.

7.8.1 Viewing a List of Segments

View information on the used and unused size and threshold of segments.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    SegmentName=s.name,
    SizeMB=convert(numeric(18,3), convert(float, sum(u.size))*@@maxpagesize/
1048576),
    UnusedSizeMB=convert(numeric(18,), (convert(float, sum(u.size))- sum(case
```

```

        WHEN (convert(float,u.size) > curunreservedpgs(u.dbid, u.lstart,
u.unreservedpgs))
        THEN convert(float,u.size) - curunreservedpgs(u.dbid, u.lstart,
u.unreservedpgs)
        ELSE 0 end))*@@maxpagesize/1048576),
    Threshold=(SELECT count(*)
                FROM <database_name>.dbo.syssegments ss,
                <database_name>.dbo.systhresholds t
                WHERE ss.segment=t.segment AND ss.name=s.name)
FROM
    <database_name>.dbo.syssegments s,
    master.dbo.sysusages u
WHERE 1 = 1
    AND ((u.segmap / ((s.segment & 1) + 1)) / power(2, (s.segment & 30))) & 1 = 1
    AND u.dbid = db_id('<database_name>')
GROUP BY
    s.name
ORDER BY
    1

```

Output

SegmentName	SizeMB	UnusedSizeMB	Threshold
default	80.000	68.141	0
logsegment	80.000	68.141	1

7.8.2 Viewing Devices Used in a Segment

View information on the devices included in the selected segment and the size of each device, in megabytes.

The following syntax provides basic guidance for viewing this information.

For regular databases:

Syntax

For a regular database:

```

SELECT
    DeviceName=v.name,
    SizeMB=sum(u.size)/(SELECT 1048576/(SELECT low
    FROM master.dbo.spt_values
    WHERE number = 1 AND type = 'E'))
FROM
    master.dbo.sysusages u,
    <database_name>.dbo.syssegments s,
    master.dbo.sysdatabases d,
    master.dbo.sysdevices v
WHERE

```

```

u.dbid=d.dbid
AND d.name='<database_name>'
AND s.name='<segm_name>'
AND v.status & 2 = 2
AND v.vdevno = u.vdevno
AND ((u.segmap/((s.segment&1)+1))/power(2,(s.segment&30)))&1=1
AND v.cntrltype=0
GROUP BY
v.name

```

Output

DeviceName	SizeMB
master	80

For in-memory database:

Syntax

```

SELECT
DeviceName=v.name,
SizeMB=sum(u.size)/(SELECT 1048576/ (SELECT low
FROM master.dbo.spt_values
WHERE number = 1 AND type = 'E'))
FROM
master.dbo.sysusages u,
<database_name>.dbo.syssegments s,
master.dbo.sysdatabases d,
master.dbo.sysdevices v
WHERE
u.dbid=d.dbid
AND d.name='<database_name>'
AND s.name='<segm_name>'
AND v.vdevno = u.vdevno
AND ((u.segmap/((s.segment&1)+1))/power(2,(s.segment&30)))&1=1
AND v.cntrltype=0
GROUP BY
v.name

```

Output

DeviceName	SizeMB
master	80

7.8.3 Viewing Tables Used in a Segment

View information on the tables allocated on the selected segment and the reserved size of each table, in kilobytes.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    TableName=o.name,
    SpaceReservedKB=reserved_pages(db_id('<database_name>'),
o.id)*@@maxpagesize/1024
FROM
    <database_name>.dbo.sysindexes i,
    <database_name>.dbo.syssegments s,
    <database_name>.dbo.sysobjects o,
    <database_name>.dbo.sysusers u
WHERE
    i.indid = 0
    AND o.id = i.id
    AND u.uid = o.uid
    AND s.name = '<segm_name>'
    AND s.segment = i.segment
    AND i.indid != 255
ORDER BY
    1
```

Output

TableName	SpaceReservedKB
ijdbc_function_escapes	32
jdbc_function_escapes	32
jdbc_ss_function_escapes	32
monBucketPool	0
monCIPC	0
monCIPCEndpoints	0
monCIPCLinks	0
monCIPCMesh	0
monCLMObjectActivity	0
monCMSFailover	0
monCachePool	0
monCachedObject	0

TableName	SpaceReservedKB
monCachedProcedures	0
monCachedStatement	0
monClusterCacheManager	0
monDBRecovery	0
monDBRecoveryLRTypes	0
monDataCache	0
monDeadLock	0
monDeviceIO	0
monDeviceSegmentIO	0
monDeviceSegmentUsage	0
monDeviceSpaceUsage	0
monDropIndexActivity	0
monEngine	0
monErrorLog	0
monFailoverRecovery	0
monHADRMembers	0
monHANANonPushdown	0
monHCBGCTasks	0
monHCBPartitionActivity	0
monHCBTuningActivity	0
monIMRSCache	0
monIMRSCacheActivity	0
monIMRSGCTasks	0
monIMRSPartitionActivity	0
monIOController	0
monIOQueue	0
monImrslogRecovery	0
monInmemoryStorage	0
monLatchFreeIndex	0
monLicense	0
monLockTimeout	0
monLocks	0
monLogicalCluster	0
monLogicalClusterAction	0

TableName	SpaceReservedKB
monLogicalClusterInstance	0
monLogicalClusterRoute	0
monMemoryUsage	0
monNVCache	0
monNetworkIO	0
monOpenDatabases	0
monOpenObjectActivity	0
monOpenPartitionActivity	0
monPCIBridge	0
monPCIEngine	0
monPCISlots	0
monPCM	0
monProcELC	0
monProcedureCache	0
monProcedureCacheMemoryUsage	0
monProcedureCacheModuleUsage	0
monProcess	0
monProcessActivity	0
monProcessLookup	0
monProcessMigration	0
monProcessNetIO	0
monProcessObject	0
monProcessProcedures	0
monProcessSQLText	0
monProcessStatement	0
monProcessWaits	0
monProcessWorkerThread	0
monRepCoordinator	0
monRepLogActivity	0
monRepMemoryStatistics	0
monRepScanners	0
monRepScannersTotalTime	0
monRepSchemaCache	0
monRepSenders	0

TableName	SpaceReservedKB
monRepStreamStatistics	0
monRepSyncTaskStatistics	0
monRepTruncationPoint	0
monSQLRepActivity	0
monSQLRepMisses	0
monSSLCertInfo	0
monServiceTask	0
monSpinlockActivity	0
monState	0
monStatementCache	0
monSysExecutionTime	0
monSysLoad	0
monSysPlanText	0
monSysSQLText	0
monSysStatement	0
monSysWaits	0
monSysWorkerThread	0
monTableColumns	0
monTableCompression	0
monTableParameters	0
monTableTransfer	0
monTables	0
monTask	0
monTempdbActivity	0
monThread	0
monThreadPool	0
monThresholdEvent	0
monWaitClassInfo	0
monWaitEventInfo	0
monWorkQueue	0
monWorkload	0
monWorkloadPreview	0
monWorkloadProfile	0
monWorkloadRaw	0

TableName	SpaceReservedKB
spt_ijdbc_conversion	32
spt_ijdbc_mda	64
spt_ijdbc_table_types	32
spt_jdbc_conversion	32
spt_jdbc_table_types	32
spt_jtext	64
spt_limit_types	32
spt_mda	64
syblicenseslog	32
syscharsets	224
syscomments	288
sysconfigures	256
syslanguages	220
sysreferences	224
sysremotelogins	96
syssservers	160
systransactions	32

7.8.4 Viewing Indexes Used in a Segment

View information on the indexes allocated on the selected segment and the table associated with each index.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
    TableName=o.name,
    IndexName=i.name
FROM
    <db_name>.dbo.sysindexes i,
    <db_name>.dbo.syssegments s,
    <db_name>.dbo.sysobjects o,
    <db_name>.dbo.sysusers u
WHERE
    i.indid != 0
    AND o.id = i.id
    AND u.uid = o.uid
    AND s.name = '<segm_name>'
    AND s.segment = i.segment
    AND i.indid != 255

```

```
ORDER BY  
2, 1
```

Output

TableName	IndexName
syscharsets	csyscharsets
syscomments	csyscomments
sysconfigures	csysconfigures
syslanguages	csyslanguages
sysreferences	csysreferences
sysremotelogins	csysremotelogins
syssservers	csyssservers
sysusermessages	csysusermessages
sysconfigures	nc1sysconfigures
syslanguages	nc1syslanguages
sysconfigures	nc2sysconfigures
syslanguages	nc2syslanguages
sysreferences	nc2sysreferences
syscharsets	ncsyscharsets
sysmessages	ncsysmessages
sysreferences	ncsysreferences
syssservers	ncsyssservers
sysusermessages	ncsysusermessages
spt_ijdbc_mda	spt_ijdbc_mda_ind
spt_mda	spt_mda_ind

7.8.5 Viewing Used and Unused Space in a Segment

Show used and unused segment space.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    SizeMB=convert(numeric(18,3), convert(float, sum(u.size))*@@maxpagesize/
1048576),
    UnusedSizeMB=convert(numeric(18,3), (convert(float, sum(u.size))- sum(case
        WHEN(convert(float,u.size) > curunreservedpgs(u.dbid, u.lstart,
u.unreservedpgs)
        THEN convert(float,u.size) - curunreservedpgs(u.dbid, u.lstart,
u.unreservedpgs)
        ELSE 0
        END))*@@maxpagesize/1048576)
FROM
    <database_name>.dbo.syssegments s,
    master.dbo.sysusages u
WHERE 1 = 1
    AND ((u.segmap / ((s.segment & 1) + 1)) / power(2, (s.segment & 30))) & 1 = 1
    AND u.dbid = db_id('<database_name>')
    AND s.name = '<segm_name>'
```

Output

SizeMB	UnusedSizeMB
80.000	68.141

7.9 Monitor Statistics

Show information on statistics filtered by resource name, statistics name and time range.

Procedure

1. Connect to the repository database. See [Connect Using the SQL Anywhere Client \[page 24\]](#).
2. Execute:

```
SELECT
    collectionName,
    collectionStartTime,
    collectionEndTime,
    resourceName,
    resourceTypeName,
    resourceTypeVersion,
    kpaName,
    statName,
    statNumericValue,
    statMeasurementUnit,
    statDisplayName
FROM
```

```

STATISTICS_FV
WHERE kpaName='<kpa_name>'
AND statName = '<stat_name>'
AND resourceName = '<resource_name>'
AND collectionEndTime between '<time_start>'
AND '<time_end>'

```

<resource_name> For information on finding this value, see [Finding the Target Resource Name for Monitoring Statistics \[page 159\]](#).

<kpa_name> and <stat_name> For information on finding these values, see [Finding KPAName and StatName for Monitoring Statistics \[page 161\]](#).

<time_start> / <time_end> Use ISO date time format for the time range of the statistics.

Results

collectionName	collectionStartTime	collectionEndTime	resourceName	resourceTypeName	resourceTypeVersion
'ASE Performance Statistics (APCA)'	2020-02-13 13:01:06.492	2020-02-13 13:01:08.48	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:06:06.492	2020-02-13 13:06:08.351	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:11:06.495	2020-02-13 13:11:08.441	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:16:06.486	2020-02-13 13:16:08.337	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:21:06.493	2020-02-13 13:21:08.422	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:26:06.491	2020-02-13 13:26:08.45	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:31:06.493	2020-02-13 13:31:08.444	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:36:06.496	2020-02-13 13:36:08.368	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:41:06.493	2020-02-13 13:41:08.382	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:46:06.499	2020-02-13 13:46:08.463	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:51:06.496	2020-02-13 13:51:08.431	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 13:56:06.496	2020-02-13 13:56:08.424	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:01:06.497	2020-02-13 14:01:08.498	'master'	'ASEMap Device MO'	'15.0.2.0'

collectionName	collectionStartTime	collectionEndTime	resourceName	resourceTypeName	resourceTypeVersion
'ASE Performance Statistics (APCA)'	2020-02-13 14:06:06.492	2020-02-13 14:06:08.444	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:11:06.495	2020-02-13 14:11:08.391	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:16:06.494	2020-02-13 14:16:08.36	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:21:06.493	2020-02-13 14:21:08.442	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:26:06.495	2020-02-13 14:26:08.396	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:31:06.496	2020-02-13 14:31:08.446	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:36:06.495	2020-02-13 14:36:08.382	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:41:06.495	2020-02-13 14:41:08.517	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:46:06.495	2020-02-13 14:46:08.463	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:51:06.497	2020-02-13 14:51:08.409	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 14:56:06.496	2020-02-13 14:56:08.475	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:01:06.494	2020-02-13 15:01:08.377	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:06:06.495	2020-02-13 15:06:08.515	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:11:06.491	2020-02-13 15:11:08.447	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:16:06.493	2020-02-13 15:16:08.421	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:21:06.488	2020-02-13 15:21:08.473	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:26:06.495	2020-02-13 15:26:08.368	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:31:06.494	2020-02-13 15:31:08.384	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:36:06.495	2020-02-13 15:36:08.48	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:41:06.485	2020-02-13 15:41:08.59	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:46:06.496	2020-02-13 15:46:08.585	'master'	'ASEMap Device MO'	'15.0.2.0'

collectionName	collectionStartTime	collectionEndTime	resourceName	resourceTypeName	resourceTypeVersion
'ASE Performance Statistics (APCA)'	2020-02-13 15:51:06.494	2020-02-13 15:51:08.578	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 15:56:06.496	2020-02-13 15:56:08.479	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:01:06.492	2020-02-13 16:01:08.365	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:06:06.493	2020-02-13 16:06:08.361	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:11:06.493	2020-02-13 16:11:08.447	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:16:06.492	2020-02-13 16:16:08.403	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:21:06.491	2020-02-13 16:21:08.482	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:26:06.496	2020-02-13 16:26:08.45	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:31:06.494	2020-02-13 16:31:08.387	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:36:06.497	2020-02-13 16:36:08.387	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:41:06.485	2020-02-13 16:41:08.23	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:46:06.487	2020-02-13 16:46:08.518	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:51:06.489	2020-02-13 16:51:08.405	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 16:56:06.494	2020-02-13 16:56:08.489	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:01:06.493	2020-02-13 17:01:08.406	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:06:06.49	2020-02-13 17:06:08.423	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:11:06.492	2020-02-13 17:11:08.411	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:16:06.496	2020-02-13 17:16:08.451	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:21:06.497	2020-02-13 17:21:08.441	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:26:06.492	2020-02-13 17:26:08.393	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:31:06.497	2020-02-13 17:31:08.404	'master'	'ASEMap Device MO'	'15.0.2.0'

collectionName	collectionStartTime	collectionEndTime	resourceName	resourceTypeName	resourceTypeVersion
'ASE Performance Statistics (APCA)'	2020-02-13 17:36:06.49	2020-02-13 17:36:08.495	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:41:06.488	2020-02-13 17:41:08.095	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:46:06.493	2020-02-13 17:46:08.353	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:51:06.491	2020-02-13 17:51:08.545	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 17:56:06.495	2020-02-13 17:56:08.451	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:01:06.496	2020-02-13 18:01:08.42	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:06:06.488	2020-02-13 18:06:08.473	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:11:06.492	2020-02-13 18:11:08.393	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:16:06.493	2020-02-13 18:16:08.479	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:21:06.494	2020-02-13 18:21:08.399	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:26:06.495	2020-02-13 18:26:08.461	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:31:06.493	2020-02-13 18:31:08.417	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:36:06.495	2020-02-13 18:36:08.319	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:41:06.496	2020-02-13 18:41:08.399	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:46:06.487	2020-02-13 18:46:08.347	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:51:06.486	2020-02-13 18:51:08.325	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 18:56:06.491	2020-02-13 18:56:08.451	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:01:06.488	2020-02-13 19:01:08.264	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:06:06.486	2020-02-13 19:06:08.305	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:11:06.486	2020-02-13 19:11:08.258	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:16:06.491	2020-02-13 19:16:08.493	'master'	'ASEMap Device MO'	'15.0.2.0'

collectionName	collectionStartTime	collectionEndTime	resourceName	resourceTypeName	resourceTypeVersion
'ASE Performance Statistics (APCA)'	2020-02-13 19:21:06.486	2020-02-13 19:21:08.53	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:26:06.494	2020-02-13 19:26:08.431	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:31:06.493	2020-02-13 19:31:08.419	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:36:06.495	2020-02-13 19:36:08.421	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:41:06.495	2020-02-13 19:41:08.436	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:46:06.492	2020-02-13 19:46:08.461	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:51:06.495	2020-02-13 19:51:08.457	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 19:56:06.491	2020-02-13 19:56:08.549	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:01:06.493	2020-02-13 20:01:08.4	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:06:06.493	2020-02-13 20:06:08.437	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:11:06.492	2020-02-13 20:11:08.465	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:16:06.495	2020-02-13 20:16:08.434	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:21:06.495	2020-02-13 20:21:08.407	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:26:06.485	2020-02-13 20:26:08.458	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:31:06.494	2020-02-13 20:31:08.471	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:36:06.496	2020-02-13 20:36:08.384	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:41:06.495	2020-02-13 20:41:08.406	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:46:06.494	2020-02-13 20:46:08.418	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:51:06.491	2020-02-13 20:51:08.453	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 20:56:06.495	2020-02-13 20:56:08.431	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:01:06.493	2020-02-13 21:01:08.448	'master'	'ASEMap Device MO'	'15.0.2.0'

collectionName	collectionStartTime	collectionEndTime	resourceName	resourceTypeName	resourceTypeVersion
'ASE Performance Statistics (APCA)'	2020-02-13 21:06:06.492	2020-02-13 21:06:08.5	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:11:06.496	2020-02-13 21:11:08.47	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:16:06.495	2020-02-13 21:16:08.426	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:21:06.495	2020-02-13 21:21:08.431	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:26:06.487	2020-02-13 21:26:08.389	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:31:06.494	2020-02-13 21:31:08.501	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:36:06.497	2020-02-13 21:36:08.586	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:41:06.486	2020-02-13 21:41:08.157	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:46:06.492	2020-02-13 21:46:08.461	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:51:06.533	2020-02-13 21:51:08.503	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 21:56:06.488	2020-02-13 21:56:08.256	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:01:06.493	2020-02-13 22:01:08.457	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:06:06.497	2020-02-13 22:06:08.412	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:11:06.495	2020-02-13 22:11:08.381	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:16:06.491	2020-02-13 22:16:08.524	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:21:06.495	2020-02-13 22:21:08.461	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:26:06.491	2020-02-13 22:26:08.458	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:31:06.494	2020-02-13 22:31:08.438	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:36:06.495	2020-02-13 22:36:08.366	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:41:06.489	2020-02-13 22:41:08.595	'master'	'ASEMap Device MO'	'15.0.2.0'
'ASE Performance Statistics (APCA)'	2020-02-13 22:46:06.492	2020-02-13 22:46:08.674	'master'	'ASEMap Device MO'	'15.0.2.0'

7.9.1 Finding the Target Resource Name for Monitoring Statistics

View information on the names of target resources. This information is needed to monitor statistics.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    DV.V_RES_ID PARENT_RES_ID,
    DV.V_RES_NM PARENT_RES_NAME,
    D.RES_ID CHILD_RES_ID,
    D.RES_NAME CHILD_RES_NAME
FROM
    RES_DV DV,RES_RESOURCE_D D
WHERE
    DV.V_RES_PRNT_ID = D.RES_ID
ORDER BY
    1,3
```

Output

The resource_name is found in the CHILD_RES_NAME column.

PARENT_RES_ID	PARENT_RES_NAME	CHILD_RES_ID	CHILD_RES_NAME
1	'SCC Service'	2	'Cockpit Management Agent Plugin'
1	'SCC Service'	3	'ASE Management Agent Plugin'
5	'SAPASE'	6	'Overview'
5	'SAPASE'	7	'Process'
5	'SAPASE'	8	'Database'
5	'SAPASE'	9	'DevicesCtr'
5	'SAPASE'	10	'Transaction'
5	'SAPASE'	11	'EnginesCtr'
5	'SAPASE'	12	'SegmentsCtr'
5	'SAPASE'	13	'Config'
5	'SAPASE'	14	'SQLActivity'
5	'SAPASE'	15	'DataCachesCtr'
5	'SAPASE'	16	'ProcedureCache'

PARENT_RES_ID	PARENT_RES_NAME	CHILD_RES_ID	CHILD_RES_NAME
5	'SAPASE'	17	'StatementCache'
5	'SAPASE'	18	'InMemoryCache'
5	'SAPASE'	19	'ConfigResourcesCtr'
5	'SAPASE'	20	'Overview'
5	'SAPASE'	21	'ThreadCtr'
9	'DevicesCtr'	22	'master'
9	'DevicesCtr'	23	'sysprocsdev'
9	'DevicesCtr'	24	'systemdbdev'
9	'DevicesCtr'	25	'tempdbdev'
9	'DevicesCtr'	26	'sybmgmtdev'
9	'DevicesCtr'	49	'mydev'
9	'DevicesCtr'	59	'IMDEV'
9	'DevicesCtr'	100	'devForSeg'
11	'EnginesCtr'	27	'0'
12	'SegmentsCtr'	28	'sybssystemdb.default'
12	'SegmentsCtr'	29	'sybssystemdb.logsegment'
12	'SegmentsCtr'	30	'sybssystemdb.system'
12	'SegmentsCtr'	31	'tempdb.default'
12	'SegmentsCtr'	32	'tempdb.logsegment'
12	'SegmentsCtr'	33	'tempdb.system'
12	'SegmentsCtr'	34	'sybssystemprocs.default'
12	'SegmentsCtr'	35	'sybssystemprocs.logsegment'
12	'SegmentsCtr'	36	'sybssystemprocs.system'
12	'SegmentsCtr'	37	'master.default'
12	'SegmentsCtr'	38	'master.logsegment'
12	'SegmentsCtr'	39	'master.system'
12	'SegmentsCtr'	78	'mytempdb.default'
12	'SegmentsCtr'	79	'mytempdb.logsegment'
12	'SegmentsCtr'	80	'mytempdb.system'
15	'DataCachesCtr'	40	'default data cache'
15	'DataCachesCtr'	87	'myCache'
21	'ThreadCtr'	41	'6'
21	'ThreadCtr'	42	'8'
21	'ThreadCtr'	43	'7'

PARENT_RES_ID	PARENT_RES_NAME	CHILD_RES_ID	CHILD_RES_NAME
21	'ThreadCtr'	44	'1'
21	'ThreadCtr'	45	'5'
21	'ThreadCtr'	46	'4'
21	'ThreadCtr'	47	'3'
21	'ThreadCtr'	48	'2'

7.9.2 Finding KPAName and StatName for Monitoring Statistics

View information on KPA and statistics names. This information is needed to monitor statistics.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    kpaName,
    kpaDescription,
    statName,
    StatDisplayName
FROM
    MON_CLCTN_KPA_STAT_DEFS_V
ORDER BY
    1, 3
```

Output

kpaName	kpaDescription
'ASE Availability Statistics (APCA)'	'ASE availability statistics (APCA)'
'ASE Availability Statistics (APCA)'	'ASE availability statistics (APCA)'
'ASE Availability Statistics (APCA)'	'ASE availability statistics (APCA)'
'ASE Availability Statistics (APCA)'	'ASE availability statistics (APCA)'
'ASE Availability Statistics (APCA)'	'ASE availability statistics (APCA)'
'ASE Availability Statistics (APCA)'	'ASE availability statistics (APCA)'
'ASE Availability Statistics (APCA)'	'ASE availability statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'

kpaName	kpaDescription
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE Device Historical Statistics'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE Segment Historical Statistics'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Capacity Statistics (APCA)'	'ASE capacity statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE DataCache Historical Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Device Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Device Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Device Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Device Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Engine Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'

kpaName	kpaDescription
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE DataCache Historical Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE DataCache Historical Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Segment Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Thread Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Thread Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'ASE Performance Statistics (APCA)'	'ASE Thread Statistics'
'ASE Performance Statistics (APCA)'	'ASE performance statistics (APCA)'
'Cockpit Availability Statistics'	'Cockpit Server Availability statistics'
'Cockpit Availability Statistics'	'Cockpit Server Availability statistics'
'Cockpit Availability Statistics'	'Cockpit Server Availability statistics'
'Cockpit Availability Statistics'	'Cockpit Server Availability statistics'
'Cockpit Availability Statistics'	'Cockpit Server Availability statistics'
'Cockpit Availability Statistics'	'Cockpit Server Availability statistics'
'Cockpit Availability Statistics'	'Cockpit Server Availability statistics'

kpaName	kpaDescription
'SCC Configuration'	'SCC MO Configuration'
'SCC Performance Statistics'	'Cockpit Server Performance statistics'
statName	StatDisplayName
'kpi_ase_connections_available'	'RES_ASEMAP_KPI_ASE_CONNECTIONS_AVAILABLE'
'kpi_ase_num_blocked_processes'	'RES_ASEMAP_KPI_ASE_NUM_BLOCKED_PROCESSES'
'kpi_ase_num_db_dump_failure'	'RES_ASEMAP_KPI_ASE_NUM_DB_DUMP_FAILURE'
'kpi_ase_num_suspended_processes'	'RES_ASEMAP_KPI_ASE_NUM_SUSPENDED_PROCESSES'
'kpi_ase_overview_cpu'	'RES_ASEMAP_KPI_ASE_OVERVIEW_CPU'
'kpi_ase_overview_device'	'RES_ASEMAP_KPI_ASE_OVERVIEW_DEVICE'
'kpi_scc_mostate_primary'	'RES_ASEMAP_KPI_SCC_MOSTATE_PRIMARY'
'kpi_ase_configresource_addit_netwk_mem'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_ADDIT_NETWK_MEM'
'kpi_ase_configresource_audit_queue_size'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_AUDIT_QUEUE_SIZE'
'kpi_ase_configresource_comp_info_pool_size'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_COMP_INFO_POOL_SIZE'
'kpi_ase_configresource_disk_io_structures'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_DISK_IO_STRUCTURES_'
'kpi_ase_configresource_mem_per_worker_process'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_MEM_PER_WORKER_PROCESS'
'kpi_ase_configresource_num_aux_scan_desc'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_AUX_SCAN_DESC'
'kpi_ase_configresource_num_locks'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_LOCKS'
'kpi_ase_configresource_num_mailboxes'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_MAILBOXES'
'kpi_ase_configresource_num_messages'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_MESSAGES'
'kpi_ase_configresource_num_open_indexes'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_OPEN_INDEXES'
'kpi_ase_configresource_num_open_objects'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_OPEN_OBJECTS'
'kpi_ase_configresource_num_open_partitions'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_OPEN_PARTITIONS'
'kpi_ase_configresource_num_sort_buffers'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_SORT_BUFFERS'
'kpi_ase_configresource_num_user_conn'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_USER_CONN'
'kpi_ase_configresource_num_worker_processes'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_NUM_WORKER_PROCESSES'

statName	StatDisplayName
'kpi_ase_configresource_perm_cache_entries'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_PERM_CACHE_ENTRIES'
'kpi_ase_configresource_proc_cache_size'	'RES_ASEMAP_KPI_ASE_CONFIGRESOURCE_PROC_CACHE_SIZE'
'kpi_ase_device_space_free'	'RES_ASEMAP_KPI_ASE_DEVICE_SPACE_FREE'
'kpi_ase_device_space_free'	'RES_ASEMAP_KPI_ASE_DEVICE_SPACE_FREE_TOP'
'kpi_ase_segment_space_free'	'RES_ASEMAP_KPI_ASE_SEGMENT_SPACE_FREE'
'kpi_ase_segment_space_free'	'RES_ASEMAP_KPI_ASE_SEGMENT_SPACE_FREE_TOP'
'kpi_ase_tempdb_free'	'RES_ASEMAP_KPI_ASE_TEMPDB_FREE'
'kpi_ase_blocked_process_average_wait_time'	'RES_ASEMAP_KPI_ASE_BLOCKED_PROCESS_AVERAGE_WAIT_TIME'
'kpi_ase_cache_hit_ratio'	'RES_ASEMAP_KPI_ASE_CACHE_HIT_RATIO'
'kpi_ase_cache_hit_ratio'	'RES_ASEMAP_KPI_ASE_CACHE_HIT_RATIO_TOP'
'kpi_ase_device_apf_reads'	'RES_ASEMAP_KPI_ASE_DEVICE_APF_READS'
'kpi_ase_device_apf_reads'	'RES_ASEMAP_KPI_ASE_DEVICE_APF_READS_TOP'
'kpi_ase_device_io'	'RES_ASEMAP_KPI_ASE_DEVICE_IO'
'kpi_ase_device_io'	'RES_ASEMAP_KPI_ASE_DEVICE_IO_TOP'
'kpi_ase_device_io_response_time'	'RES_ASEMAP_KPI_ASE_DEVICE_IO_RESPONSE_TIME'
'kpi_ase_device_io_response_time'	'RES_ASEMAP_KPI_ASE_DEVICE_IO_RESPONSE_TIME_TOP'
'kpi_ase_engine_cpu_utilization'	'RES_ASEMAP_KPI_ASE_ENGINE_CPU_UTILIZATION'
'kpi_ase_engine_cpu_utilization'	'RES_ASEMAP_KPI_ASE_ENGINE_CPU_UTILIZATION_TOP'
'kpi_ase_long_running_transc_execution_time'	'RES_ASEMAP_KPI_ASE_LONG_RUNNING_TRANSC_EXECUTION_TIME'
'kpi_ase_network_io_packets_received'	'RES_ASEMAP_KPI_ASE_NETWORK_IO_PACKETS_RECEIVED'
'kpi_ase_network_io_packets_sent'	'RES_ASEMAP_KPI_ASE_NETWORK_IO_PACKETS_SENT'
'kpi_ase_num_cache_misses'	'RES_ASEMAP_KPI_ASE_NUM_CACHE_MISSES'
'kpi_ase_num_cache_misses'	'RES_ASEMAP_KPI_ASE_NUM_CACHE_MISSES_TOP'
'kpi_ase_num_cache_searches'	'RES_ASEMAP_KPI_ASE_NUM_CACHE_SEARCHES'
'kpi_ase_num_cache_searches'	'RES_ASEMAP_KPI_ASE_NUM_CACHE_SEARCHES_TOP'
'kpi_ase_num_deadlocks'	'RES_ASEMAP_KPI_ASE_NUM_DEADLOCKS'
'kpi_ase_num_errorlog_critical'	'RES_ASEMAP_KPI_ASE_NUM_ERRORLOG_CRITICAL'
'kpi_ase_num_errorlog_error'	'RES_ASEMAP_KPI_ASE_NUM_ERRORLOG_ERROR'
'kpi_ase_num_errorlog_info'	'RES_ASEMAP_KPI_ASE_NUM_ERRORLOG_INFO'
'kpi_ase_num_errorlog_warning'	'RES_ASEMAP_KPI_ASE_NUM_ERRORLOG_WARNING'

statName	StatDisplayName
'kpi_ase_num_lock_by_address_level_lock'	'RES_ASEMAP_KPI_ASE_NUM_LOCK_BY_ADDRESS_LEVEL_LOCK'
'kpi_ase_num_lock_by_page_level_lock'	'RES_ASEMAP_KPI_ASE_NUM_LOCK_BY_PAGE_LEVEL_LOCK'
'kpi_ase_num_lock_by_row_level_lock'	'RES_ASEMAP_KPI_ASE_NUM_LOCK_BY_ROW_LEVEL_LOCK'
'kpi_ase_num_lock_by_table_level_lock'	'RES_ASEMAP_KPI_ASE_NUM_LOCK_BY_TABLE_LEVEL_LOCK'
'kpi_ase_num_lock_for_all_levels'	'RES_ASEMAP_KPI_ASE_NUM_LOCK_FOR_ALL_LEVELS'
'kpi_ase_procedure_hit_percent'	'RES_ASEMAP_KPI_ASE_PROCEDURE_HIT_PERCENT'
'kpi_ase_segment_spaceusage'	'RES_ASEMAP_KPI_ASE_SEGMENT_SPACEUSAGE'
'kpi_ase_segment_spaceusage'	'RES_ASEMAP_KPI_ASE_SEGMENT_SPACEUSAGE_TOP'
'kpi_ase_server_transactions'	'RES_ASEMAP_KPI_ASE_SERVER_TRANSACTIONS'
'kpi_ase_sp_who_response_time'	'RES_ASEMAP_KPI_ASE_SP_WHO_RESPONSE_TIME'
'kpi_ase_stmt_cache_hit_percent'	'RES_ASEMAP_KPI_ASE_STMT_CACHE_HIT_PERCENT'
'kpi_ase_thread_cpu'	'RES_ASEMAP_KPI_ASE_THREAD_CPU'
'kpi_ase_thread_cpu'	'RES_ASEMAP_KPI_ASE_THREAD_CPU_TOP'
'kpi_ase_thread_system_cpu'	'RES_ASEMAP_KPI_ASE_THREAD_SYSTEM_CPU'
'kpi_ase_thread_system_cpu'	'RES_ASEMAP_KPI_ASE_THREAD_SYSTEM_CPU_TOP'
'kpi_ase_thread_user_cpu'	'RES_ASEMAP_KPI_ASE_THREAD_USER_CPU'
'kpi_ase_thread_user_cpu'	'RES_ASEMAP_KPI_ASE_THREAD_USER_CPU_TOP'
'kpi_scc_availability_repository_mtbf'	'RES_KPI_SCC_AVAILABILITY_REPOSITORY_MTBF'
'kpi_scc_availability_repository_mtbf'	'RES_KPI_SCC_AVAILABILITY_REPOSITORY_MTBF'
'kpi_scc_availability_repository_state'	'RES_KPI_SCC_AVAILABILITY_REPOSITORY_STATE'
'kpi_scc_availability_repository_state'	'RES_KPI_SCC_AVAILABILITY_REPOSITORY_STATE'
'kpi_scc_availability_repository_uptime'	'RES_KPI_SCC_AVAILABILITY_REPOSITORY_UPTIME'
'kpi_scc_availability_repository_uptime'	'RES_KPI_SCC_AVAILABILITY_REPOSITORY_UPTIME'
'kpi_scc_mostate_primary'	'RES_KPI_SCC_STATE_PRIMARY'
'kpi_scc_mostate_primary'	'RES_KPI_SCC_STATE_PRIMARY'
'kpi_fault_manager_state'	'RES_ASEMAP_KPI_FAULT_MANAGER_STATE'
'kpi_fault_manager_state'	'RES_ASEMAP_KPI_FAULT_MANAGER_STATE_TOP'
'kpi_fm_down_rep'	'RES_ASEMAP_KPI_FM_DOWN_REP'
'kpi_fm_down_rep'	'RES_ASEMAP_KPI_FM_DOWN_REP_TOP'
'kpi_fm_failover_init'	'RES_ASEMAP_KPI_FM_FAILOVER_INIT'
'kpi_fm_failover_init'	'RES_ASEMAP_KPI_FM_FAILOVER_INIT_TOP'
'kpi_fm_hibernate'	'RES_ASEMAP_KPI_FM_HIBERNATE'

statName	StatDisplayName
'kpi_fm_hibernate'	'RES_ASEMAP_KPI_FM_HIBERNATE_TOP'
'kpi_fm_no_contact_primary_host_agent'	'RES_ASEMAP_KPI_FM_NO_CONTACT_PRIMARY_HOST_AGENT'
'kpi_fm_no_contact_primary_host_agent'	'RES_ASEMAP_KPI_FM_NO_CONTACT_PRIMARY_HOST_AGENT_TOP'
'kpi_fm_no_contact_standby_host_agent'	'RES_ASEMAP_KPI_FM_NO_CONTACT_STANDBY_HOST_AGENT'
'kpi_fm_no_contact_standby_host_agent'	'RES_ASEMAP_KPI_FM_NO_CONTACT_STANDBY_HOST_AGENT_TOP'
'kpi_fm_suspended_rep'	'RES_ASEMAP_KPI_FM_SUSPENDED_REP'
'kpi_fm_suspended_rep'	'RES_ASEMAP_KPI_FM_SUSPENDED_REP_TOP'
'kpi_fm_sync_to_async_rep'	'RES_ASEMAP_KPI_FM_SYNC_TO_ASYNC_REP'
'kpi_fm_sync_to_async_rep'	'RES_ASEMAP_KPI_FM_SYNC_TO_ASYNC_REP_TOP'
'kpi_rat_number_of_pkgs_flushed'	'RES_ASEMAP_KPI_RAT_NUMBER_OF_PKGS_FLUSHED_TOP'
'kpi_rma_state'	'RES_ASEMAP_KPI_RMA_STATE'
'kpi_rma_state'	'RES_ASEMAP_KPI_RMA_STATE_TOP'
'kpi_rs_prim_state'	'RES_ASEMAP_KPI_RS_PRIM_STATE'
'kpi_rs_prim_state'	'RES_ASEMAP_KPI_RS_PRIM_STATE_TOP'
'kpi_rs_remo_state'	'RES_ASEMAP_KPI_RS_REMO_STATE'
'kpi_rs_remo_state'	'RES_ASEMAP_KPI_RS_REMO_STATE_TOP'
'kpi_ase_free_xact_log'	'RES_ASEMAP_KPI_ASE_FREE_XACT_LOG'
'kpi_ase_free_xact_log_percent'	'RES_ASEMAP_KPI_ASE_FREE_XACT_LOG_PERCENT'
'kpi_ase_free_xact_log_percent'	'RES_ASEMAP_KPI_ASE_FREE_XACT_LOG_PERCENT_TO_P'
'kpi_rat_log_record_processed'	'RES_ASEMAP_KPI_RAT_LOG_RECORD_PROCESSED'
'kpi_rat_log_record_scanned'	'RES_ASEMAP_KPI_RAT_LOG_RECORD_SCANNED'
'kpi_rat_number_of_pkgs_flushed'	'RES_ASEMAP_KPI_RAT_NUMBER_OF_PKGS_FLUSHED'
'kpi_rs_path_dist_dsi_latency'	'RES_ASEMAP_KPI_RS_PATH_DIST_DSI_LATENCY'
'kpi_rs_path_dsi_cmds'	'RES_ASEMAP_KPI_RS_PATH_DSI_CMDS'
'kpi_rs_path_dsi_rdb_latency'	'RES_ASEMAP_KPI_RS_PATH_DSI_RDB_LATENCY'
'kpi_rs_path_dsi_xact'	'RES_ASEMAP_KPI_RS_PATH_DSI_XACT'
'kpi_rs_path_exec_b'	'RES_ASEMAP_KPI_RS_PATH_EXEC_B'
'kpi_rs_path_exec_dist_latency'	'RES_ASEMAP_KPI_RS_PATH_EXEC_DIST_LATENCY'
'kpi_rs_path_overall_tick_latency'	'RES_ASEMAP_KPI_RS_PATH_OVERALL_TICK_LATENCY'
'kpi_rs_path_overall_tick_latency'	'RES_ASEMAP_KPI_RS_PATH_OVERALL_TICK_LATENCY_TOP'

statName	StatDisplayName
'kpi_rs_path_pda_exec_latency'	'RES_ASEMAP_KPI_RS_PATH_PDB_EXEC_LATENCY'
'kpi_rs_path_rsi_b'	'RES_ASEMAP_KPI_RS_PATH_RSI_B'
'kpi_rs_prim_device_usage_percent'	'RES_ASEMAP_KPI_RS_PRIM_DEVICE_USAGE_PERCENT'
'kpi_rs_prim_device_usage_percent'	'RES_ASEMAP_KPI_RS_PRIM_DEVICE_USAGE_PERCENT_TOP'
'kpi_rs_prim_queue_backlog'	'RES_ASEMAP_KPI_RS_PRIM_QUEUE_BACKLOG'
'kpi_rs_prim_queue_backlog'	'RES_ASEMAP_KPI_RS_PRIM_QUEUE_BACKLOG_TOP'
'kpi_rs_prim_xact_log_backlog'	'RES_ASEMAP_KPI_RS_PRIM_XACT_LOG_BACKLOG'
'kpi_rs_prim_xact_log_backlog'	'RES_ASEMAP_KPI_RS_PRIM_XACT_LOG_BACKLOG_TOP'
'kpi_rs_remo_device_usage_percent'	'RES_ASEMAP_KPI_RS_REMO_DEVICE_USAGE_PERCENT'
'kpi_rs_remo_device_usage_percent'	'RES_ASEMAP_KPI_RS_REMO_DEVICE_USAGE_PERCENT_TOP'
'kpi_rs_remo_queue_backlog'	'RES_ASEMAP_KPI_RS_REMO_QUEUE_BACKLOG'
'kpi_rs_remo_queue_backlog'	'RES_ASEMAP_KPI_RS_REMO_QUEUE_BACKLOG_TOP'
'kpi_rs_replication_path_state'	'RES_ASEMAP_KPI_RS_REPLICATION_PATH_STATE'
'kpi_rs_replication_path_state'	'RES_ASEMAP_KPI_RS_REPLICATION_PATH_STATE_TOP'
'kpi_scc_capacity_repository_disk_usage_percent'	'RES_KPI_SCC_CAPACITY_REPOSITORY_DISK_USAGE_PERCENT'
'kpi_scc_capacity_repository_disk_used'	'RES_KPI_SCC_CAPACITY_REPOSITORY_DISK_USED'
'kpi_scc_configuration_repository_disk_size'	'RES_KPI_SCC_CONFIGURATION_REPOSITORY_DISK_SIZE'
'kpi_scc_performance_repository_crud_response_time'	'RES_KPI_SCC_PERFORMANCE_REPOSITORY_CRUD_RESPONSE_TIME'

7.9.3 Key Performance Indicators for SAP ASE

View information on the key performance indicators (KPIs) that show performance trends.

For information about the always-on licensed option performance indicators, see [HADR KPIs \[page 326\]](#).

SAP ASE Statistics	KPI Name	Description	Data Collection Name
ASE Availability Statistics	Available Connections	Available Connections	ASE Availability Statistics
	Number of Blocked Processes	Number of currently blocked processes that have been blocked for more than 5 seconds. The heat chart uses this metric to display server status.	ASE Availability Statistics
	Number of Database Dump Failures	Number of failed database dumps.	ASE Availability Statistics

SAP ASE Statistics	KPI Name	Description	Data Collection Name
	Number of Suspended Processes	Number of processes that are currently suspended. The heat chart uses this metric to display server status.	ASE Availability Statistics
	Server Availability Status	Status of the SAP ASE server. Values of most interest are stopped and running.	ASE Availability Statistics
	Server Device IO Rate	Total number of I/O operations performed by all devices on the server during the current collection cycle.	ASE Availability Statistics
	Server Percent CPU Utilization	Average CPU utilization percentage across all active engines on the server.	ASE Availability Statistics
ASE Capacity Statistics	Configured Resource Additional Network Memory	Maximum size of additional memory that can be used for network packets that are larger than the default packet size.	ASE Capacity Statistics
	Configured Resource Audit Queue Size	The audit queue holds audit records generated by user processes until the records can be processed and written to the audit trail.	ASE Capacity Statistics
	Configured Resource Compression Info Pool Size	Size of the memory pool used for compression.	ASE Capacity Statistics
	Configured Resource Disk IO Structures	Initial number of disk I/O control blocks SAP ASE allocates at start-up.	ASE Capacity Statistics
	Configured Resource Memory Per Worker Process	Amount of memory used by worker processes.	ASE Capacity Statistics
	Configured Resource Number of Aux Scan Descriptors	Number of auxiliary scan descriptors available in a pool shared by all users on a server.	ASE Capacity Statistics
	Configured Resource Number of Locks	Total number of available locks for all users on SAP ASE.	ASE Capacity Statistics
	Configured Resource Number of Mailboxes	Number of mailbox structures allocated by SAP ASE.	ASE Capacity Statistics
	Configured Resource Number of Messages	Number of message structures allocated by SAP ASE	ASE Capacity Statistics
	Configured Resource Number of Open Indexes	Maximum number of indexes that can be used simultaneously on SAP ASE.	ASE Capacity Statistics
	Configured Resource Number of Open Objects	Maximum number of objects that can be open simultaneously on SAP ASE	ASE Capacity Statistics
	Configured Resource Number of Open Partitions	Number of partitions that SAP ASE can access at one time.	ASE Capacity Statistics

SAP ASE Statistics	KPI Name	Description	Data Collection Name
	Configured Resource Number of Sort Buffers	Amount of memory allocated for buffers used to hold pages read from input tables and perform index merges during sorts.	ASE Capacity Statistics
	Configured Resource Number of User Connections	Maximum number of user connections that can simultaneously be connected to SAP ASE.	ASE Capacity Statistics
	Configured Resource Number of Worker Processes	Maximum number of worker processes that SAP ASE can use at any one time for all simultaneously running parallel queries.	ASE Capacity Statistics
	Configured Resource Permission Cache Entries	Number of cache protectors per task.	ASE Capacity Statistics
	Configured Resource Procedure Cache Size	Size of the procedure cache. The procedure cache is used while running stored procedures.	ASE Capacity Statistics
	Device Space Free	Total amount of free space, in megabytes, on this device.	ASE Capacity Statistics
	Segment Space Free	Amount, in megabytes, of free space in the segment; collected separately for each segment.	ASE Capacity Statistics
	Server tempdb Free Space	Amount, in megabytes, of free space in the tempdb database.	ASE Capacity Statistics
ASE Performance Statistics	Average Blocked Process Wait Time	Average time, in milliseconds, that the current blocked processes have waited.	ASE Performance Statistics
	Cache Hit Ratio	Hit ratio in the data cache during the current collection cycle.	ASE Performance Statistics
	Device APF Reads	Rate of asynchronous prefetch read operations per second on the selected device.	ASE Performance Statistics
	Device IO Rate	Rate of I/O operations per second on this device.	ASE Performance Statistics
	Device IO Response Time	Response time, in milliseconds, for I/O operations performed on this device.	ASE Performance Statistics
	Engine CPU Utilization	CPU utilization for this engine as a percentage.	ASE Performance Statistics
	Long Running Transaction Execution Time	Execution time of longest running transaction	ASE Performance Statistics
	Number of Address Locks	Number of address-level locks server-wide.	ASE Performance Statistics
	Number of Cache Misses	Number of times that a needed page was not found in a cache and had to be read from disk.	ASE Performance Statistics

**SAP ASE
Statistics**

KPI Name	Description	Data Collection Name
Number of Cache Searches	Number of cache searches, including hits and misses for all caches combined.	ASE Performance Statistics
Number of Critical Flags Last Collection	Number of critical flags received during the collection cycle.	ASE Performance Statistics
Number of Deadlocks	Number of deadlocks on the server since the most recent execution of the collection.	ASE Performance Statistics
Number of Error Flags Last Collection	Number of error flags received during the collection cycle.	ASE Performance Statistics
Number of Information Flags Last Collection	Number of information flags received during the collection cycle.	ASE Performance Statistics
Number of Locks	Total number of active locks of all types on the server.	ASE Performance Statistics
Number of Packets Received in Network IO	Number of packets received during the current collection cycle.	ASE Performance Statistics
Number of Packets Sent in Network IO	Number of packets sent during the current collection cycle.	ASE Performance Statistics
Number of Page Locks	Number of page-level locks server-wide.	ASE Performance Statistics
Number of Row Locks	Number of row-level locks server-wide.	ASE Performance Statistics
Number of Server Transactions	Total number of transactions during the current collection cycle.	ASE Performance Statistics
Number of Table Locks	Number of table-level locks server-wide.	ASE Performance Statistics
Number of Warning Flags Last Collection	Number of warning flags received during the collection cycle.	ASE Performance Statistics
Procedure Cache Hit Ratio	Hit ratio in the procedure cache.	ASE Performance Statistics
Statement Cache Hit Ratio	Hit ratio in the statement cache during the current collection cycle.	ASE Performance Statistics
Segment Space Usage	Change in megabytes in the amount of space used by this segment since the last refresh.	ASE Performance Statistics
sp_who Response Time	Time, in milliseconds, the sp_who stored procedure takes to return a response. sp_who is called each time collection_ase_histmon is executed to collect performance statistics.	ASE Performance Statistics
Thread System CPU Utilization	CPU utilization percentage in handling system level operations for each thread.	ASE Performance Statistics
Thread Total CPU Utilization	Total CPU utilization obtained by adding Thread User CPU Utilization and Thread System CPU Utilization.	ASE Performance Statistics
Thread User CPU Utilization	Thread User CPU Utilization	ASE Performance Statistics

SAP ASE Statistics	KPI Name	Description	Data Collection Name
Data Caches	Cache Hit Ratio	Hit ratio in the data cache during the current collection cycle.	ASE Performance Statistics
	Number of Cache Searches	Number of cache searches, including hits and misses for all caches combined.	ASE Performance Statistics
	Number of Cache Misses	Number of times that a needed page was not found in a cache and had to be read from disk.	ASE Performance Statistics
Devices	Device APF Reads	Rate per second of asynchronous pre-fetch read operations on this device.	ASE Performance Statistics
	Device IO Rate	Rate of I/O operations per second on this device.	ASE Performance Statistics
	Device IO Response Time	Response time, in milliseconds, for I/O operations performed on this device.	ASE Performance Statistics
	Device Space Free	Total amount of free space, in megabytes, on this device.	ASE Capacity Statistics
Engines	Engine CPU Utilization	Percentage of CPU cycles used by this SAP ASE engine.	ASE Performance Statistics
Segments	Segment Space Usage	Change, in megabytes, in the amount of space used by this segment since the last refresh.	ASE Performance Statistics
Threads	Thread User CPU Utilization	CPU utilization percentage in handling user committed queries for each thread.	ASE Performance Statistics
	Thread System CPU Utilization	CPU utilization percentage in handling system level operations for each thread.	ASE Performance Statistics
	Thread Total CPU Utilization	Total CPU utilization obtained by adding Thread User CPU Utilization and Thread System CPU Utilization.	ASE Performance Statistics

7.10 Monitor Transactions

Display information about a currently running transaction, including the process that initiated the transaction.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  UserName=pl.Login,
  p.Application,
```

```

TransactionName=t.xactname,
SPID=t.spid,
StartTime=t.starttime,
Host=pl.ClientHost,
DBName=p.DBName,
pa.ULCBytesWritten,
pa.ULCFlushes,
pa.ULCFlushFull,
pa.ULCMaxUsage,
pa.ULCCurrentUsage
FROM
  master.dbo.systransactions t,
  master.dbo.monProcess p,
  master.dbo.monProcessLookup pl,
  master.dbo.monProcessActivity pa
WHERE
  p.SPID=t.spid
  AND pl.SPID=t.spid
  AND pa.SPID=t.spid

```

Output

UserName	Application	TransactionName	SPID	StartTime	Host
sa	isql	\$user_transaction	33	2020-03-01 19:04:49.853	Inxpw04

DBName	ULCBytesWritten	ULCFlushes	ULCFlushFull	ULCMaxUsage	ULCCurrentUsage
pubs2	0	0	0	0	272

7.11 Monitor SQL Queries

Display details about recently executed SQL queries.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  sq.SPID,
  pl.Login,
  sq.KPID,
  sq.BatchID,
  pl.Application,
  CpuTime=sum(st.CpuTime),
  MemUsageKB=sum(st.MemUsageKB),
  WaitTime=sum(st.WaitTime),
  LogicalReads=sum(st.LogicalReads),
  PhysicalReads=sum(st.PhysicalReads),
  StartTime=min(st.StartTime),

```

```

        EndTime=max(st.EndTime),
        sq.SQLText
FROM
    master.dbo.monSysSQLText sq,
    master.dbo.monSysStatement st,
    master..monProcessLookup pl
WHERE
    sq.SPID != @@spid
    AND sq.SPID=st.SPID
    AND sq.KPID=st.KPID
    AND sq.BatchID=st.BatchID
    AND sq.SPID=pl.SPID
    AND sq.KPID=pl.KPID
    AND sq.SequenceInBatch=1
GROUP BY
    sq.SPID,
    pl.Login,
    sq.KPID,
    sq.BatchID,
    pl.Application,
    sq.SQLText
ORDER BY
    11,1,2,3,4,5

```

Output

SPID	Login	KPID	BatchID	Application	CpuTime	MemUsageKB
31	sa	381091907	227493	SCC_ASE	2	2260
31	sa	381091907	227494	SCC_ASE	0	174
31	sa	381091907	227495	SCC_ASE	0	142
31	sa	381091907	227496	SCC_ASE	4	694
31	sa	381091907	227497	SCC_ASE	5	694
31	sa	381091907	227498	SCC_ASE	1	438
31	sa	381091907	227499	SCC_ASE	2	438
31	sa	381091907	227500	SCC_ASE	4	694
31	sa	381091907	227501	SCC_ASE	2	1566
31	sa	381091907	227502	SCC_ASE	1	438
31	sa	381091907	227503	SCC_ASE	4	694
31	sa	381091907	227504	SCC_ASE	4	694
31	sa	381091907	227505	SCC_ASE	1	438
31	sa	381091907	227506	SCC_ASE	2	438
31	sa	381091907	227507	SCC_ASE	4	694
31	sa	381091907	227508	SCC_ASE	1	438
31	sa	381091907	227509	SCC_ASE	0	56
31	sa	381091907	227510	SCC_ASE	13	2790

WaitTime	LogicalReads	PhysicalReads	StartTime	EndTime	SQLText
0	68	0	2020-03-02 19:33:07.1	2020-03-02 19:33:07.126	sp_configure 'number of open databases'
0	3	0	2020-03-02 19:33:07.126	2020-03-02 19:33:07.126	select size=ltrim(str(su m((1. + (d.high - d.low)) / (1048576. / @@pagesize)), 10,2)) from master.dbo.sysdev ices d where d.cntrtype = 0
0	0	0	2020-03-02 19:33:07.13	2020-03-02 19:33:07.13	select runningdays = DaysRunning, deadlocks = NumDeadlocks, startdate = StartDate, counterscleared = CountersCleared from master..monState
0	2546	0	2020-03-02 19:33:10.013	2020-03-02 19:33:10.016	select procedureCacheSi ze=(select value from master..syscurconf igs where config=146), stalls=(select sum(Stalls) from master..monProce dureCache), statementCachesi ze=(select value from master..syscurconf igs where config=414), cachesize=(select sum(b.value)
0	2546	0	2020-03-02 19:33:10.086	2020-03-02 19:33:10.09	select procedureCacheSi ze=(select value from master..syscurconf igs where config=146), stalls=(select sum(Stalls) from

WaitTime	LogicalReads	PhysicalReads	StartTime	EndTime	SQLText
					master..monProcedureCache), statementCacheSize=(select value from master..syscurconfigs where config=414), cachesize=(select sum(b.value)
0	508	0	2020-03-02 19:33:10.403	2020-03-02 19:33:10.403	select maxprocesses=(select b.value from master..sysconfigures a , master..syscurconfigs b where a.config = b.config and a.config = 103), activeprocesses=(select count(*) from master..monProcessActivity where ServerUserID !=0), hwp=(select config_admin(22
0	508	0	2020-03-02 19:33:10.72	2020-03-02 19:33:10.72	select maxprocesses=(select b.value from master..sysconfigures a , master..syscurconfigs b where a.config = b.config and a.config = 103), activeprocesses=(select count(*) from master..monProcessActivity where ServerUserID !=0), hwp=(select config_admin(22
0	2546	0	2020-03-02 19:33:15.956	2020-03-02 19:33:15.96	select procedureCacheSize=(select value from master..syscurconfigs where

WaitTime	LogicalReads	PhysicalReads	StartTime	EndTime	SQLText
					config=146), stalls=(select sum(Stalls) from master..monProce dureCache), statementCachesi ze=(select value from master..syscurconf igs where config=414), cachesize=(select sum(b.value)
0	51	0	2020-03-02 19:33:16.416	2020-03-02 19:33:16.416	select spid = P.SPID, familyID = P.FamilyID, kpid = P.KPID, status = P.status, login = P.Login, command = P.Command , dbname = P.DBName, blockingspid = P.BlockingSPID, timeblocked = L.WaitTime, clienthost = PL.ClientHost, program = P.Application, memusag
0	508	0	2020-03-02 19:33:16.463	2020-03-02 19:33:16.463	select maxprocesses=(s elect b.value from master..sysconfig ures a , master..syscurconf igs b where a.config = b.config and a.config = 103), activeprocesses=(select count(*) from master..monProce ssActivity where ServerUserID ! =0), hwp=(select config_admin(22
0	2546	0	2020-03-02 19:33:18.336	2020-03-02 19:33:18.34	select procedureCacheSi ze=(select value

WaitTime	LogicalReads	PhysicalReads	StartTime	EndTime	SQLText
					from master..syscurconf igs where config=146), stalls=(select sum(Stalls) from master..monProce dureCache), statementCachesi ze=(select value from master..syscurconf igs where config=414), cachesize=(select sum(b.value)
0	2546	0	2020-03-02 19:33:18.426	2020-03-02 19:33:18.43	select procedureCacheSi ze=(select value from master..syscurconf igs where config=146), stalls=(select sum(Stalls) from master..monProce dureCache), statementCachesi ze=(select value from master..syscurconf igs where config=414), cachesize=(select sum(b.value)
0	508	0	2020-03-02 19:33:18.89	2020-03-02 19:33:18.89	select maxprocesses=(s elect b.value from master..sysconfig ures a , master..syscurconf igs b where a.config = b.config and a.config = 103), activeprocesses=(select count(*) from master..monProce ssActivity where ServerUserID != 0), hwp=(select config_admin(22

WaitTime	LogicalReads	PhysicalReads	StartTime	EndTime	SQLText
0	508	0	2020-03-02 19:33:18.943	2020-03-02 19:33:18.943	select maxprocesses=(s elect b.value from master..sysconfig ures a , master..syscurconf igs b where a.config = b.config and a.config = 103), activeprocesses=(select count(*) from master..monProce ssActivity where ServerUserID ! =0), hwp=(select config_admin(22
0	2546	0	2020-03-02 19:33:23.303	2020-03-02 19:33:23.306	select procedureCacheSi ze=(select value from master..syscurconf igs where config=146), stalls=(select sum(Stalls) from master..monProce dureCache), statementCachesi ze=(select value from master..syscurconf igs where config=414), cachesize=(select sum(b.value)
0	508	0	2020-03-02 19:33:23.803	2020-03-02 19:33:23.803	select maxprocesses=(s elect b.value from master..sysconfig ures a , master..syscurconf igs b where a.config = b.config and a.config = 103), activeprocesses=(select count(*) from master..monProce ssActivity where ServerUserID !

WaitTime	LogicalReads	PhysicalReads	StartTime	EndTime	SQLText
					=0), hwp=(select config_admin(22
0	0	0	2020-03-02 19:33:37.096	2020-03-02 19:33:37.096	select 1
0	4528	0	2020-03-02 19:33:37.1	2020-03-02 19:33:37.126	sp_configure 'number of open databases'

7.12 Monitor The HADR Environment

Display primary and standby servers, HADR system status, synchronization mode, and synchronization state.

7.12.1 Replication Paths

Show information on the replication paths.

7.12.1.1 Viewing Database Path Information

View information on the replication modes you have configured, the current replication states in the HADR with DR node environment, distribution mode and path, Replication Server status, and latency.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_status path](#).

Syntax

```
EXEC sap_status path
```

Output

PATH	NAME	VALUE	INFO
	Start Time	2016-04-27 22:33:04.026	Time command started executing.
	Elapsed Time	00:00:03	Command execution time.
DR	Hostname	site2	Logical host name.
DR	HADR Status	DR Standby : Inactive	Identify the primary and standby sites.
DR	Synchronization Mode	Asynchronous	The configured Synchronization Mode value.
DR	Synchronization State	Inactive	Synchronization Mode in which replication is currently operating.
DR	Distribution Mode	Local	Configured value for the distribution_mode replication model property.
DR	Replication Server Status	Active	The status of Replication Server.
HA	Hostname	site1	Logical host name.
HA	HADR Status	Standby : Inactive	Identify the primary and standby sites.
HA	Synchronization Mode	Synchronous	The configured Synchronization Mode value.
HA	Synchronization State	Inactive	Synchronization Mode in which replication is currently operating.
HA	Distribution Mode	Remote	Configured value for the distribution_mode replication model property.
HA	Replication Server Status	Active	The status of Replication Server.
PR	Hostname	site0	Logical host name.
PR	HADR Status	Primary : Active	Identify the primary and standby sites.
PR	Synchronization Mode	Synchronous	The configured Synchronization Mode value.
PR	Synchronization State	Synchronous	Synchronization Mode in which replication is currently operating.
PR	Distribution Mode	Remote	Configured value for the distribution_mode replication model property.

PATH	NAME	VALUE	INFO
PR	Replication Server Status	Active	The status of Replication Server.
HA.DR.PI2	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
HA.DR.PI2	Latency Time	Unknown	No latency information for database 'PI2'.
HA.DR.PI2	Latency	Unknown	No latency information for database 'PI2'.
HA.DR.PI2	Commit Time	Unknown	No last commit time for the database 'PI2'.
HA.DR.PI2	Distribution Path	PR	The path of Replication Server through which transactions travel.
HA.DR.PI2	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
HA.DR.db1	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
HA.DR.db1	Latency Time	Unknown	No latency information for database 'db1'.
HA.DR.db1	Latency	Unknown	No latency information for database 'db1'.
HA.DR.db1	Commit Time	Unknown	No last commit time for the database 'db1'.
HA.DR.db1	Distribution Path	PR	The path of Replication Server through which transactions travel.
HA.DR.db1	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
HA.DR.master	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
HA.DR.master	Latency Time	Unknown	No latency information for database 'master'.
HA.DR.master	Latency	Unknown	No latency information for database 'master'.
HA.DR.master	Commit Time	Unknown	No last commit time for the database 'master'.

PATH	NAME	VALUE	INFO
HA.DR.master	Distribution Path	PR	The path of Replication Server through which transactions travel.
HA.DR.master	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
HA.PR.PI2	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
HA.PR.PI2	Latency Time	Unknown	No latency information for database 'PI2'.
HA.PR.PI2	Latency	Unknown	No latency information for database 'PI2'.
HA.PR.PI2	Commit Time	Unknown	No last commit time for the database 'PI2'.
HA.PR.PI2	Distribution Path	PR	The path of Replication Server through which transactions travel.
HA.PR.PI2	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
HA.PR.db1	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
HA.PR.db1	Latency Time	Unknown	No latency information for database 'db1'.
HA.PR.db1	Latency	Unknown	No latency information for database 'db1'.
HA.PR.db1	Commit Time	Unknown	No last commit time for the database 'db1'.
HA.PR.db1	Distribution Path	PR	The path of Replication Server through which transactions travel.
HA.PR.db1	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
HA.PR.master	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
HA.PR.master	Latency Time	Unknown	No latency information for database 'master'.

PATH	NAME	VALUE	INFO
HA.PR.master	Latency	Unknown	No latency information for database 'master'.
HA.PR.master	Commit Time	Unknown	No last commit time for the database 'master'.
HA.PR.master	Distribution Path	PR	The path of Replication Server through which transactions travel.
HA.PR.master	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
PR.DR.PI2	State	Active	Path is active and replication can occur.
PR.DR.PI2	Latency Time	2016-04-27 22:33:00.840	Time latency last calculated
PR.DR.PI2	Latency	707	Latency (ms)
PR.DR.PI2	Commit Time	2016-04-27 22:33:00.840	Time last commit replicated
PR.DR.PI2	Distribution Path	HA	The path of Replication Server through which transactions travel.
PR.DR.PI2	Drain Status	Not Drained	The drain status of the transaction logs of the primary database server.
PR.DR.db1	State	Active	Path is active and replication can occur.
PR.DR.db1	Latency Time	2016-04-27 22:33:00.832	Time latency last calculated
PR.DR.db1	Latency	703	Latency (ms)
PR.DR.db1	Commit Time	2016-04-27 22:33:00.832	Time last commit replicated
PR.DR.db1	Distribution Path	HA	The path of Replication Server through which transactions travel.
PR.DR.db1	Drain Status	Not Drained	The drain status of the transaction logs of the primary database server.
PR.DR.master	State	Active	Path is active and replication can occur.
PR.DR.master	Latency Time	2016-04-27 22:33:00.832	Time latency last calculated
PR.DR.master	Latency	703	Latency (ms)
PR.DR.master	Commit Time	2016-04-27 22:33:00.832	Time last commit replicated
PR.DR.master	Distribution Path	HA	The path of Replication Server through which transactions travel.

PATH	NAME	VALUE	INFO
PR.DR.master	Drain Status	Not Drained	The drain status of the transaction logs of the primary database server.
PR.HA.PI2	State	Active	Path is active and replication can occur.
PR.HA.PI2	Latency Time	2016-04-27 22:33:00.640	Time latency last calculated
PR.HA.PI2	Latency	607	Latency (ms)
PR.HA.PI2	Commit Time	2016-04-27 22:33:00.646	Time last commit replicated
PR.HA.PI2	Distribution Path	HA	The path of Replication Server through which transactions travel.
PR.HA.PI2	Drain Status	Not Drained	The drain status of the transaction logs of the primary database server.
PR.HA.db1	State	Active	Path is active and replication can occur.
PR.HA.db1	Latency Time	2016-04-27 22:33:00.646	Time latency last calculated
PR.HA.db1	Latency	610	Latency (ms)
PR.HA.db1	Commit Time	2016-04-27 22:33:00.646	Time last commit replicated
PR.HA.db1	Distribution Path	HA	The path of Replication Server through which transactions travel.
PR.HA.db1	Drain Status	Not Drained	The drain status of the transaction logs of the primary database server.
PR.HA.master	State	Active	Path is active and replication can occur.
PR.HA.master	Latency Time	2016-04-27 22:33:00.632	Time latency last calculated
PR.HA.master	Latency	603	Latency (ms)
PR.HA.maste	Commit Time	2016-04-27 22:33:00.632	Time last commit replicated
PR.HA.master	Distribution Path	HA	The path of Replication Server through which transactions travel.
PR.HA.master	Drain Status	Not Drained	The drain status of the transaction logs of the primary database server.

7.12.1.2 Viewing the Replication Agent

View information on the replication agent.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT rep_agent_admin('is running', db_id('<database_id>'))
```

Output

A value of 1 means the agent is running; otherwise stopping.

7.12.1.3 Viewing the Backlog

View information on the estimated failover time, Replication Server device size, simple persistent queue (SPQ) size, usage, backlog, replication truncation backlog (inbound queue and outbound queue), replication route queue truncation backlog, SAP ASE transaction log size and backlog, as well as stable queue backlogs.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_status resource](#).

Syntax

```
EXEC sap_status resource
```

Output

NAME	TYPE	VALUE
	Start Time	2016-06-02 22:38:20.891
	Elapsed Time	00:00:02
	Estimated Failover Time	-1
PR	Replication device size (MB)	256
PR	Replication device usage (MB)	208
DR	Replication device size (MB)	512
DR	Replication device usage (MB)	64
HA	Replication device size (MB)	256
HA	Replication device usage (MB)	208

NAME	TYPE	VALUE
HA.master	Replication simple persistent queue size (MB)	500
HA.PI2	Replication simple persistent queue size (MB)	2000
HA.db1	Replication simple persistent queue size (MB)	2000
PR.master	ASE transaction log size (MB)	100
PR.master	ASE transaction log backlog (MB)	0
PR.db1	ASE transaction log size (MB)	20
PR.db1	ASE transaction log backlog (MB)	0
PR.PI2	ASE transaction log size (MB)	10
PR.PI2	ASE transaction log backlog (MB)	0
HA.master.SPQ	Replication simple persistent queue backlog (MB)	0
HA.master.IBQ	Replication inbound queue backlog (MB)	0
HA.master.IBQ	Replication inbound queue truncation backlog (MB)	0
HA.DR.RQ	Replication route queue backlog (MB)	0
HA.DR.RQ	Replication route queue truncation backlog (MB)	0
DR.master.OBQ	Replication outbound queue backlog (MB)	0
DR.master.OBQ	Replication outbound queue truncation backlog (MB)	0
HA.db1.SPQ	Replication simple persistent queue backlog (MB)	0
HA.db1.IBQ	Replication inbound queue backlog (MB)	0
HA.db1.IBQ	Replication inbound queue truncation backlog (MB)	0
DR.db1.OBQ	Replication outbound queue backlog (MB)	0
DR.db1.OBQ	Replication outbound queue truncation backlog (MB)	0
HA.PI2.SPQ	Replication simple persistent queue backlog (MB)	0
HA.PI2.IBQ	Replication inbound queue backlog (MB)	0
HA.PI2.IBQ	Replication inbound queue truncation backlog (MB)	0

NAME	TYPE	VALUE
DR.PI2.OBQ	Replication outbound queue backlog (MB)	0
DR.PI2.OBQ	Replication outbound queue truncation backlog (MB)	0
HA.master.OBQ	Replication outbound queue backlog (MB)	0
HA.master.OBQ	Replication outbound queue truncation backlog (MB)	0
HA.db1.OBQ	Replication outbound queue backlog (MB)	0
HA.db1.OBQ	Replication outbound queue truncation backlog (MB)	0
HA.PI2.OBQ	Replication outbound queue backlog (MB)	0
HA.PI2.OBQ	Replication outbound queue truncation backlog (MB)	0
DR.PI2	Replication queue backlog (MB)	0
HA.PI2	Replication queue backlog (MB)	0
HA.master	Replication queue backlog (MB)	0
DR.db1	Replication queue backlog (MB)	0
DR.master	Replication queue backlog (MB)	0
HA.db1	Replication queue backlog (MB)	0

7.12.2 Service Components

Summarize the status of all components.

7.12.2.1 Viewing the RMA

View information on the version of the DR agent plug-in.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_version](#).

Syntax

```
EXEC sap_version
```

7.12.2.2 Viewing the Local and Remote Replication Servers

View information on the status of the Replication Server.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [admin health](#).

Syntax

```
EXEC admin health
```

Output

Mode	Quiesce	Status	Loss Status
NORMAL	TRUE	HEALTHY	SUSPECT

Column descriptions for admin health output

Mode

The state of the Replication Server with regard to recovery. It is one of these values:

- NORMAL – Replication Server is operating normally.
- REBUILDING – This is a transient state while Replication Server executes the rebuild queues command.
- RECOVERY – The Replication Server is in stand-alone mode and the rebuild queues command has been executed.
- STANDALONE – Replication Server is not accepting or starting any connections. You can only enter this state by starting Replication Server with the -M flag. Exit from stand-alone mode by shutting down the Replication Server and restarting it without the -M flag.

Quiesce

Indicates if the Replication Server is quiesced. It is one of:

- TRUE – Replication Server is quiesced, that is, all messages have been flushed.
- FALSE – Replication Server is not quiesced.

Status

Overall status of the Replication Server. It is either:

- HEALTHY – All threads are executing as expected.
- SUSPECT – A thread is down and the Replication Server expected it to be running. Or, a thread is in a “Connecting” state. The “Connecting” state means that either the server to which Replication Server is connecting is unavailable and a problem

exists, or the Replication Server will connect successfully in a moment and the suspect status is transitory.

You can see threads that are not running by executing `admin who_is_down`.

Loss Status

Data loss status:

- SUSPECT – Replication Server suspects possible data loss in the queues.
- DETECTING – Replication Server is checking for data loss in the queues.
- IGNORING – Replication Server is ignoring any data loss in the queues because you executed the `ignore loss` command.
- NO LOSS – Replication Server does not detect any data loss in the queues.

7.12.3 Log Records

Display information on the free transaction log and the log records scanned.

7.12.3.1 Viewing the Free Transaction Log

View information on the free transaction log.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  FreeLogSizeMB = lct_admin('logsegment_freepages',
    db_id('<database id>')) / 512.0 *(@@maxpagesize / @@pagesize),
  XactLogSizeMB = sum(size) / 512.0 *(@@maxpagesize / @@pagesize)
FROM
  master..sysusages
WHERE
  dbid = db_id('<database id>')
  AND (segmap & 4 != 0)
```

Output

FreeLogSizeMB	XactLogSizeMB
1986.578120	2000.000000

The calculation to determine the percentage of free transaction log is $(\text{FreeLogSizeMB} / \text{XactLogSizeMB}) * 100$.

7.12.3.2 Viewing the Log Records Scanned and Processed

View information on log records scanned and processed.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    LogRecordsScanned,
    LogRecordsProcessed
FROM
    master..monRepLogActivity
WHERE
    DBID = db_id('<database_name>')
```

Output

LogRecordsScanned	LogRecordsProcessed
42913	45446

7.12.4 Viewing Throughput

View information on multiple ticket rows (one ticket per row) where `pdb_t` is bigger than a user specified start date. `pdb_t` indicates the time stamp when the ticket was injected in corresponding primary ASE database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [hadrgettickethistory](#).

Syntax

```
EXEC hadrgettickethistory <logical_host>, <database_name>, '<start_time>'
```

<logical_host> Specifies the name of logical ASE host, to query ticket information.

<database_name> Specifies the name of the database.

<start_time> Specifies the start date and time since when ticket information must be returned, in the format 'YYYY-MM-DD HH:MM:SS'.

Output

CNT	H1	H2	H3	H4	PDB	PRS			
1089	send_trace	DR	Agent	GetLatency	db1				
RRS	RDB	PDB_T	EXEC_T	DIST_T	RSI_T	DSI_T			
			2020-06-01 07:08:56.616	1900-01-01 00:00:00.000	1900-01-01 00:00:00.000	1900-01-01 00:00:00.000			
RSI_T	DSI_T	RDB_T	EXEC_B	RSI_B	DSI_TNX	DSI_CMD	TICKET	CONN_ID	
1900-01-01 00:00:00.0 00	1900-01-01 00:00:00.0 00	1900-01-01 00:00:00.0 00	1900-01-01 00:00:00.0 00 0	0	0	0	-	0	

7.12.5 Viewing the Latency

View information on multiple ticket rows (one ticket per row) where `pdb_t` is bigger than a user specified start date. `pdb_t` indicates the time stamp when the ticket was injected in corresponding primary ASE database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [hadrtickethistory](#).

Syntax

```
EXEC hadrtickethistory <logical_host>, <database_name>, '<start_time>'
```

<logical_host> Specifies the name of logical ASE host, to query ticket information.

<database_name> Specifies the name of the database.

<start_time> Specifies the start date and time since when ticket information must be returned, in the format 'YYYY-MM-DD HH:MM:SS'.

Output

CNT	H1	H2	H3	H4	PDB	PRS			
1089	send_trace	DR	Agent	GetLatency	db1				

RRS	RDB	PDB_T	EXEC_T	DIST_T	RSI_T	DSI_T
			2020-06-01 07:08:56.616	1900-01-01 00:00:00.000	1900-01-01 00:00:00.000	1900-01-01 00:00:00.000
RDB_T	EXEC_B	RSI_B	DSI_TNX	DSI_CMD	TICKET	CONN_ID
1900-01-01 00:00:00.000	1900-01-01 00:00:00.000 0	0	0	0	-	0

7.12.6 Replication Agent

7.12.6.1 Starting the Replication Agent

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_start_rep_agent](#).

Syntax

```
EXEC sp_start_rep_agent <dbname>
```

dbname Specifies the name of the database for which you want to start a RepAgent.

7.12.6.2 Stopping the Replication Agent

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_stop_rep_agent](#).

Syntax

```
EXEC sp_stop_rep_agent <dbname>
```

dbname Specifies the name of the database for which you want to shut down a RepAgent.

7.12.6.3 Changing the Replication Agent Configuration Parameters

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_config_rep_agent](#).

Syntax

```
EXEC sp_config_rep_agent <dbname>, '<parameter>', '<value>'
```

<dbname>	Specifies the name of a database.
<parameter>	Specifies the parameter name.
<value>	Specifies the new value to be applied.

7.12.6.4 Viewing the Current Context of the Replication Agent

View information on the current context of the replication agent.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    TruncpointsProcessed,
    NumberOfWakeups,
    TruncpointsRequested,
    TruncpointsConfirmed
FROM
    master..monRepTruncationPoint
WHERE
    DBID = db_id('<database_name>')
```

Output

TruncpointsProcessed	NumberOfWakeup	TruncpointsRequested	TruncpointsConfirmed
322	3652	1	3652

7.13 Monitor Caches

Monitor cache size, statistics, and cache activity.

7.13.1 Data Cache Statistics and Details

Show information on the data cache.

7.13.1.1 Viewing Caches

View information about the size and level of activity in each data cache.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    CacheID,
    CacheName,
    CacheSearches,
    CachePartitions,
    RelaxedReplacement,
    LogicalReads,
    PhysicalReads,
    PhysicalWrites,
    SizeMB =(b.value / 1024)
FROM
    master.dbo.monDataCache m,
    master.dbo.sysconfigures a,
    master.dbo.syscurconfigs b
WHERE
    b.config = a.config
    AND b.comment = a.name
    AND a.parent = 19
    AND a.config = a.parent
    AND a.name = m.CacheName
    AND a.status & 196640 != 196640
ORDER BY
    CacheName
```

Output

CacheID	CacheName	CacheSearches	CachePartitions	RelaxedReplacement
0	default data cache	24193046	1	0
2	myCache	96090	1	0

LogicalReads	PhysicalReads	PhysicalWrites	SizeMB
24193046	15701	115481	4
96090	57	0	2

7.13.1.2 Viewing Pool Information

View information about the pools of different sizes that optimize I/O in the selected cache. Details include size, usage, reread ratio, physical and dirty reads, pages touched, buffers to MRU (most recently used), and buffers to LRU (least recently used). The Buffers to MRU and Buffers to LRU columns show buffers added to the ends of the buffer list. The oldest buffers (the least recently used) are flushed first.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    PoolSizeKB = cp.IOBufferSize / 1024,
    SizeMB = convert(numeric(9, 2), cp.AllocatedKB / 1024.0),
    UsagePercent = convert(numeric(9, 2), 100.0 * cp.PagesTouched / AllocatedKB
*
    @@maxpagesize / 1024.0),
    cp.PagesRead,
    cp.PhysicalReads,
    cp.Stalls,
    cp.PagesTouched,
    cp.BuffersToMRU,
    cp.BuffersToLRU
FROM
    master.dbo.monCachePool cp,
    master.dbo.monDataCache dc
WHERE
    cp.CacheID = dc.CacheID
    AND cp.CacheName = '<cache_name>'
```

Output

PoolSizeKB	SizeMB	UsagePercent	PagesRead	PhysicalReads	Stalls
8	1.00	0.39	2	1	0
4	3.00	7.81	60	56	0

PagesTouched	BuffersToMRU	BuffersToLRU
1	2	0
60	60	0

7.13.1.3 Viewing Cached Objects

View information on the tables in the selected data cache and their size, in kilobytes.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    TableIndex =case
        WHEN IndexID = 0 then convert( char(60), DBName +'.'+ isnull(OwnerName,
'dbo') +'.'+ ObjectName)
        ELSE convert( char(60), DBName +'.'+ isnull(OwnerName, 'dbo') +'.'+
object_name(ObjectID, DBID) +'.'+ ObjectName)
        END,
    CachedSizeKB = CachedKB
FROM
    master.dbo.monCachedObject
WHERE
    CacheName = '<cache_name>'
    AND DBID = <database_id>
ORDER BY
    1
```

<cache_name> Specifies the name of the cache.

<database_id> Specifies the ID for a database if the object is not in the current database. Database IDs are stored in the db_id column of sysdatabases.

Output

TableIndex	CachedSizeKB
pubs2.dbo.sysalternates.csysalternate	8
pubs2.dbo.sysattributes.csysattribute	8
pubs2.dbo.sysattributes.ncsysattribute	8
pubs2.dbo.syscolumn	20
pubs2.dbo.syscolumns.csyscolumn	16
pubs2.dbo.syscomments	4
pubs2.dbo.syscomments.csyscomment	8
pubs2.dbo.sysencryptkeys.csysencryptkey	4
pubs2.dbo.sysindexe	12
pubs2.dbo.sysindexes.csysindexe	4
pubs2.dbo.sysobject	16
pubs2.dbo.sysobjects.csysobject	8
pubs2.dbo.sysobjects.ncsysobject	8
pubs2.dbo.sysprocedure	4
pubs2.dbo.sysprocedures.csysprocedure	12
pubs2.dbo.sysprotect	4
pubs2.dbo.sysprotects.csysprotect	8
pubs2.dbo.sysrole	4
pubs2.dbo.sysroles.csysrole	8
pubs2.dbo.syssegment	8
pubs2.dbo.sysstatistics.csysstatistic	8
pubs2.dbo.systhreshold	8
pubs2.dbo.systypes.ncsystype	8
pubs2.dbo.sysusermessage	4
pubs2.dbo.sysuser	4
pubs2.dbo.sysusers.csysuser	8
pubs2.dbo.sysusers.ncsysusers1	8
pubs2.dbo.sysusers.ncsysusers	2

7.13.2 Procedure Cache Statistics and Details

Display information about the contents of the procedure cache.

7.13.2.1 Viewing Procedure Cache Summary

View information about the contents of the procedure cache, which is a memory pool used for stored procedures and a variety of other objects.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    HitRatio =case
        WHEN Requests = 0 then 0.00
        ELSE convert(numeric(5, 2), convert(float, (Requests - Loads)) / Requests
    * 100)
    END,
    Stalls,
    MemoryUsageMB = convert(numeric(8, 2), convert(float, (SELECT sum(MemUsageKB)
        FROM master.dbo.monCachedProcedures)) / 1024.0)
FROM
    master.dbo.monProcedureCache
```

Output

HitRatio	Stalls	MemoryUsageMB
98.55	0	20.44

7.13.2.2 Viewing Top Procedure Cache Module Users

View information on the modules that use the cache most heavily.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
```

```

ActiveMB = convert(numeric(10, 3), Active / 512.0),
ModuleName
FROM
  master.dbo.monProcedureCacheModuleUsage
ORDER BY
  1 desc

```

Output

ActiveMB	ModuleName
21.699	Procedural Objects
0.142	Statement Cache
0.121	Procmem Control
0.109	Execution
0.017	Data change
0.017	Partition Conditions
0.015	Dynamic SQL
0.009	Sort
0.003	CIS
0.001	HK GC
0.001	Pdes Local HashTab
0.001	Multiple Temporary Database
0.000	Access
0.000	Backup
0.000	Parser
0.000	Network
0.000	Recovery
0.000	HK Chores
0.000	Optimizer
0.000	Utilities
0.000	Diagnostics
0.000	Replication
0.000	Transaction
0.000	BLOB Management
0.000	Frame Management
0.000	Workload Manager
0.000	AuxBuf Management

ActiveMB	ModuleName
0.000	Cluster Threshold Manager
0.000	Cluster Membership Service

7.13.2.3 Viewing Cached Procedures

View information on the stored procedures in the cache (in the Procedural Objects Module). For each stored procedure, it gives the name, database name, cached size, owner's name, compile date, and plan ID.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    DBName,
    ObjectName,
    CachedSizeMB = convert(numeric(7, 3), MemUsageKB / 1024.0),
    OwnerName = isnull(OwnerName, 'dbo'),
    CompileDate,
    PlanID
FROM
    master.dbo.monCachedProcedures
```

Output

DBName	ObjectName	CachedSizeMB	OwnerName	CompileDate	PlanID
sybmgmtdb	check_jsc_state	0.011	dbo	Mar 6 2020 11:56PM	8
sybmgmtdb	js_history_jsh_st_1 520005415	0.007	dbo	Mar 6 2020 11:56PM	11
sybmgmtdb	js_history_jsh_ex_ 1536005472	0.007	dbo	Mar 6 2020 11:56PM	12
sybmgmtdb	js_history_jsh_us_ 1522005529	0.007	dbo	Mar 6 2020 11:56PM	13
sybmgmtdb	js_history_jsh_at_ 1568005586	0.007	dbo	Mar 6 2020 11:56PM	14
sybmgmtdb	js_history_jsh_os_ 1584005643	0.007	dbo	Mar 6 2020 11:56PM	15
sybmgmtdb	js_history_jsh_si_1 616005757	0.007	dbo	Mar 6 2020 11:56PM	17

DBName	ObjectName	CachedSizeMB	OwnerName	CompileDate	PlanID
sybmgmtdb	js_history_jsh_up_1632005814	0.007	dbo	Mar 6 2020 11:56PM	18
sybmgmtdb	js_history_jsh_sp_1648005871	0.007	dbo	Mar 6 2020 11:56PM	19
sybmgmtdb	check_jsh_state	0.011	dbo	Mar 6 2020 11:56PM	20
sybmgmtdb	check_jsh_exit_code	0.009	dbo	Mar 6 2020 11:56PM	21
sybmgmtdb	js_schedul_sched_880003135	0.007	dbo	Mar 6 2020 11:56PM	27
sybmgmtdb	js_schedul_sched_896003192	0.007	dbo	Mar 6 2020 11:56PM	28
sybmgmtdb	js_schedul_sched_912003249	0.007	dbo	Mar 6 2020 11:56PM	29
sybmgmtdb	js_schedul_sched_928003306	0.007	dbo	Mar 6 2020 11:56PM	30
sybmgmtdb	js_schedul_sched_944003363	0.007	dbo	Mar 6 2020 11:56PM	31
sybmgmtdb	js_schedul_sched_960003420	0.007	dbo	Mar 6 2020 11:56PM	32
sybmgmtdb	js_schedul_sched_976003477	0.007	dbo	Mar 6 2020 11:56PM	33
sybmgmtdb	check_sched_inter_val	0.007	dbo	Mar 6 2020 11:56PM	34
sybmgmtdb	check_sched_inter_val_units	0.007	dbo	Mar 6 2020 11:56PM	35
sybmgmtdb	check_sched_days	0.007	dbo	Mar 6 2020 11:56PM	36
sybmgmtdb	js_schedul_sjob_u_1104003933	0.007	dbo	Mar 6 2020 11:56PM	37
sybmgmtdb	js_schedul_sjob_e_1120003990	0.007	dbo	Mar 6 2020 11:56PM	38
sybmgmtdb	js_schedul_sjob_p_1136004047	0.007	dbo	Mar 6 2020 11:56PM	39
sybmgmtdb	js_schedul_sjob_t_1152004104	0.007	dbo	Mar 6 2020 11:56PM	40
sybmgmtdb	check_sjob_enable	0.007	dbo	Mar 6 2020 11:56PM	41
sybssystemprocs	sp_mda	0.171	dbo	Mar 6 2020 11:56PM	43
sybssystemprocs	sp_drv_getdbtype	0.031	dbo	Mar 6 2020 11:56PM	45

DBName	ObjectName	CachedSizeMB	OwnerName	CompileDate	PlanID
sybsystemprocs	sp_sql_type_name	0.140	dbo	Mar 6 2020 11:56PM	47
sybsystemprocs	sp_drv_iscasesensitive	0.015	dbo	Mar 6 2020 11:56PM	49
sybmgmtdb	sp_js_agent	0.716	dbo	Mar 7 2020 12:00AM	51
sybmgmtdb	sp_js_check_id	0.611	dbo	Mar 7 2020 12:00AM	53
sybmgmtdb	sp_listjobtext	0.058	dbo	Mar 7 2020 12:00AM	55
sybsystemprocs	sp_setpsexex	0.265	dbo	Mar 7 2020 12:00AM	57
master	sp_aux_checkroleperm	0.097	dbo	Mar 7 2020 12:00AM	59
sybsystemprocs	sp_jst_svr_update_statistics	0.085	dbo	Mar 7 2020 12:00AM	61
sybsystemprocs	sp_jst_db_update_statistics	0.109	dbo	Mar 7 2020 12:00AM	63
sybsystemprocs	sp_jst_get_object_quoted_name	0.062	dbo	Mar 7 2020 12:00AM	65
sybmgmtdb	sp_sjobsetstatus	0.269	dbo	Mar 7 2020 12:00AM	67
sybmgmtdb	sp_writejoboutput	0.255	dbo	Mar 7 2020 12:00AM	69
master	sp_configure	1.939	dbo	Mar 7 2020 12:26AM	71
sybsystemprocs	sp_monitorconfig	1.011	dbo	Mar 7 2020 12:56AM	101
master	sp_getmessage	0.154	dbo	Mar 7 2020 12:56AM	103
sybsystemprocs	sp_namecrack	0.058	dbo	Mar 8 2020 6:25PM	113
sybsystemprocs	sp_spaceused_syslogs	0.046	dbo	Mar 8 2020 6:25PM	115
sybsystemprocs	sp_f_getuint	0.093	dbo	Mar 8 2020 6:25PM	119
sybsystemprocs	sp_f_getval	0.076	dbo	Mar 8 2020 6:25PM	121
sybsystemprocs	sp_helprotect	1.550	dbo	Mar 8 2020 6:30PM	173
sybsystemprocs	sp_server_info	0.052	dbo	Mar 8 2020 6:31PM	185

DBName	ObjectName	CachedSizeMB	OwnerName	CompileDate	PlanID
sybsystemprocs	sp_getmessage	0.154	dbo	Mar 8 2020 6:40PM	269
sybsystemprocs	sp_ddlgen	0.078	dbo	Mar 8 2020 7:04PM	507
sybsystemprocs	sp_ddlgen_permissions	0.154	dbo	Mar 8 2020 7:04PM	509
sybsystemprocs	sp_ddlgen_database	0.873	dbo	Mar 8 2020 7:04PM	514
sybsystemprocs	sp_ddlgen_db_priority_alterdb	0.062	dbo	Mar 8 2020 7:04PM	516
sybsystemprocs	sp_ddlgen_db_priority_with_clause	0.125	dbo	Mar 8 2020 7:04PM	518
sybsystemprocs	sp_help	1.916	dbo	Mar 8 2020 8:06PM	685
sybsystemprocs	sp_help_allobjs	0.146	dbo	Mar 8 2020 8:06PM	687
sybsystemprocs	sp_config_dump	0.626	dbo	Mar 8 2020 8:14PM	694
sybsystemprocs	sp_dump_history	0.937	dbo	Mar 8 2020 8:14PM	696
master	sp_namecrack	0.058	dbo	Mar 8 2020 8:14PM	698
sybsystemprocs	sp_configure	2.308	dbo	Mar 9 2020 12:20AM	941
sybsystemprocs	sp_autoformat	0.726	dbo	Mar 9 2020 12:20AM	943
master	sp_configure	1.955	dbo	Mar 9 2020 12:21AM	963
tempdb	*sq0914971455_0960892451ss*	0.007	dbo	Mar 9 2020 12:36AM	1053
tempdb	*sq0316017326_0960892451ss*	0.007	dbo	Mar 9 2020 1:01AM	1079
sybsystemprocs	sp_spaceused	1.480	dbo	Mar 9 2020 1:05AM	1085
master	sp_autoformat	0.734	dbo	Mar 9 2020 1:16AM	1099

7.13.3 Statement Cache Statistics and Details

Display information about SQL queries and query plans stored in the statement cache.

7.13.3.1 Viewing Statement Cache Summary

View information about the size, hit count, and traffic in the cache.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    TotalSizeMB = convert(numeric(8, 3), TotalSizeKB / 1024.0),
    UsedSizeMB = convert(numeric(8, 3), UsedSizeKB / 1024.0),
    NumStatements,
    RatioOpenObjToStmt = case
        WHEN NumStatements = 0 then 0.0
        ELSE (convert(numeric(4, 1), convert(float, (SELECT cur.value
            FROM
                master.dbo.sysconfigures con,
                master.dbo.syscurconfigs cur
            WHERE
                con.name = 'number of open objects'
                AND cur.config = con.config)) / NumStatements))
    END,
    NumSearches,
    HitCount,
    NumInserts,
    NumRemovals,
    NumRecompilesSchemaChanges,
    NumRecompilesPlanFlushes
FROM
    master.dbo.monStatementCache
```

Output

TotalSizeMB	UsedSizeMB	NumStatements	RatioOpenObjToStmt	NumSearches
9.765	4.230	140	3.6	24

HitCount	NumInserts	NumRemovals	NumRecompilesSchemaChanges	NumRecompilesPlanFlushes
24	0	0	0	0

7.13.3.2 Viewing Cached Statement Texts

View information on the query selected in the Cached Statements table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT Text = show_cached_text(<statement_id>)
```

Output

Text

```
select O.name, U.name, O.type from master.dbo.sysdepends D, master.dbo.sysobjects O, master.dbo.sysusers U where  
O.uid = U.uid and ( O.type = 'S' or O.type = 'U' ) and O.id = D.depid and D.id =1328004731
```

7.13.3.3 Viewing Cached Statement

View information on SQL statements by statement ID (SSQLID), and displays the owner name, use count, CPU time, elapsed time to execute, and logical and physical I/O figures for each query.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT  
    OwnerName = suser_name(SUserID),  
    SSQLID,  
    UseCount,  
    MaxCpuTime,  
    MinCpuTime,  
    AvgCpuTime,  
    MaxElapsedTime,  
    MinElapsedTime,  
    AvgElapsedTime,  
    MaxLIO,  
    MinLIO,  
    AvgLIO,  
    MaxPIO,  
    MinPIO,  
    AvgPIO  
FROM  
    master.dbo.monCachedStatement
```

Output

OwnerName	SSQLID	UseCount	MaxCpuTime	MinCpuTime	AvgCpuTime	MaxElapsedTime
sa	970639670	1	0	0	0	1
tech_user	1198116478	6	0	0	0	1
tech_user	558114198	6	0	0	0	1
sa	435685769	1	0	0	0	1
sa	1767166510	1	0	0	0	1
sa	922639499	31	0	0	0	1
sa	1425593286	30	0	0	0	1
sa	499685997	4	0	0	0	1
sa	51684401	1	0	0	0	1
sa	250637105	1	0	0	0	1
tech_user	718114768	7	0	0	0	1
tech_user	942115566	7	0	0	0	1
tech_user	1873594882	58	0	0	0	1
sa	394637618	1	0	0	0	1
sa	1585593856	1	0	0	0	1
sa	727162805	1	0	0	0	1
sa	243685085	1	0	0	0	1
sa	359161494	1	0	0	0	1
tech_user	1006115794	7	0	0	0	1
tech_user	878115338	7	0	0	0	1
sa	314637333	1	0	0	0	1
sa	1601593913	1	0	0	0	1
sa	810639100	60	0	0	0	1
sa	2001595338	56	1	1	1	1
sa	1345593001	1	0	0	0	1
tech_user	974115680	7	0	0	0	1
sa	410637675	1	0	0	0	1
sa	1895166966	2	0	0	0	1
tech_user	1953595167	29	0	0	0	1
sa	330637390	1	0	0	0	1
sa	1847166795	2	0	0	0	1
sa	515686054	1	0	0	0	1

OwnerName	SSQLID	UseCount	MaxCpuTime	MinCpuTime	AvgCpuTime	MaxElapsedTime
tech_user	1969595224	29	0	0	0	1
sa	954639613	67	0	0	0	1
sa	1457593400	100	0	0	0	1
tech_user	686114654	6	0	0	0	1
sa	275685199	1	0	0	0	1
tech_user	990115737	7	0	0	0	1
tech_user	910115452	7	0	0	0	1
sa	378637561	1	0	0	0	1
sa	362637504	1	0	0	0	1
sa	179684857	1	0	0	0	1
sa	467685883	1	0	0	0	1
sa	595686339	1	0	0	0	1
sa	1617593970	7	0	0	0	1
tech_user	574114255	6	0	0	0	1
sa	682638644	68	0	0	0	1
sa	1473593457	99	0	0	0	1
sa	346637447	1	0	0	0	1
tech_user	846115224	7	1	0	1	1
sa	563686225	1	0	0	0	1
sa	1555689759	1	0	0	0	1
sa	842639214	30	0	0	0	1
sa	2033595452	27	0	0	0	1
tech_user	782114996	7	0	0	0	1
sa	115684629	1	0	0	0	1
tech_user	814115110	6	0	0	0	1
tech_user	590114312	6	0	0	0	1
sa	202636934	1	0	0	0	1
sa	259685142	1	0	0	0	1
sa	1319164914	2	0	0	0	1
sa	826639157	30	0	0	0	1
sa	2017595395	27	0	0	0	1
sa	426637732	1	0	0	0	1
sa	211684971	1	0	0	0	1
sa	403685655	1	0	0	0	1

OwnerName	SSQLID	UseCount	MaxCpuTime	MinCpuTime	AvgCpuTime	MaxElapsedTime
sa	1879166909	2	0	0	0	1
tech_user	1022115851	21	0	0	0	1
sa	1751166453	1	0	0	0	1
sa	1857594825	2	3	3	3	3
tech_user	926115509	7	0	0	0	1
sa	218636991	1	0	0	0	1
sa	1863166852	2	0	0	0	1
sa	547686168	1	0	0	0	1
sa	586638302	1	0	0	0	1
sa	147684743	1	0	0	0	1
sa	1489593514	4	0	0	0	1
sa	67684458	1	0	0	0	1
tech_user	798115053	7	0	0	0	1
tech_user	750114882	7	0	0	0	1
sa	531686111	1	0	0	0	1
sa	387685598	1	0	0	0	1
sa	291685256	1	0	0	0	1
tech_user	1054115965	6	0	0	0	1
tech_user	1214116535	6	0	0	0	1
tech_user	1150116307	6	0	0	0	1
tech_user	1086116079	6	0	0	0	1
sa	627686453	1	0	0	0	1
tech_user	1134116250	6	0	0	0	1
tech_user	1230116592	6	0	0	0	1
tech_user	1182116421	6	0	0	0	1
tech_user	1246116649	6	0	0	0	1
sa	163684800	1	0	0	0	1
sa	1050639955	1	0	0	0	1
sa	1633594027	7	2	2	2	2
sa	451685826	1	0	0	0	1
tech_user	1985595281	29	0	0	0	1
tech_user	542114141	6	0	0	0	1
sa	266637162	1	0	0	0	1
tech_user	670114597	6	0	0	0	1

OwnerName	SSQLID	UseCount	MaxCpuTime	MinCpuTime	AvgCpuTime	MaxElapsedTime
tech_user	734114825	7	0	0	0	1
sa	611686396	1	0	0	0	1
tech_user	1102116136	7	0	0	0	1
sa	131684686	1	0	0	0	1
sa	666638587	1	0	0	0	1
tech_user	894115395	7	1	0	1	1
sa	186636877	1	0	0	0	1
sa	227685028	1	0	0	0	1
sa	938639556	31	0	0	0	1
sa	1441593343	30	0	0	0	1
tech_user	606114369	6	0	0	0	1
sa	371685541	1	0	0	0	1
sa	307685313	1	0	0	0	1
tech_user	766114939	7	0	0	0	1
sa	282637219	1	0	0	0	1
sa	1713594312	58	0	0	0	1
sa	234637048	1	0	0	0	1
tech_user	1443689360	1	0	0	0	1
sa	506638017	1	0	0	0	1
sa	195684914	1	0	0	0	1
sa	83684515	1	0	0	0	1
tech_user	958115623	7	0	0	0	1
sa	339685427	1	0	0	0	1
tech_user	1166116364	6	0	0	0	1
sa	579686282	1	0	0	0	1
sa	483685940	1	0	0	0	1
sa	419685712	1	0	0	0	1
sa	1569593799	3	0	0	0	1
sa	355685484	1	0	0	0	1
sa	323685370	1	0	0	0	1
tech_user	1118116193	7	0	0	0	1
tech_user	702114711	7	0	0	0	1
sa	99684572	1	0	0	0	1
sa	298637276	1	0	0	0	1

OwnerName	SSQLID	UseCount	MaxCpuTime	MinCpuTime	AvgCpuTime	MaxElapsedTime
tech_user	1038115908	7	0	0	0	1
tech_user	862115281	7	0	0	0	1
sa	2039167479	1	0	0	0	1
sa	2055167536	1	0	0	0	1
sa	170636820	1	0	0	0	1
tech_user	622114426	6	0	0	0	1
tech_user	1070116022	7	0	0	0	1
tech_user	830115167	7	1	0	1	1

MinElapsedTime	AvgElapsedTime	MaxLIO	MinLIO	AvgLIO	MaxPIO	MinPIO	AvgPIO
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	3	3	3	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	9	9	9	0	0	0
1	1	9	9	9	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	517	517	517	0	0	0

MinElapsedTime	AvgElapsedTime	MaxLIO	MinLIO	AvgLIO	MaxPIO	MinPIO	AvgPIO
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
2	2	1551	1551	1551	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	6	6	6	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	9	9	9	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	5	5	5	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	6	6	6	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	6	6	6	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0

MinElapsedTime	AvgElapsedTime	MaxLIO	MinLIO	AvgLIO	MaxPIO	MinPIO	AvgPIO
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	6	6	6	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	9	9	9	0	0	0
1	1	9	9	9	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	6	6	6	0	0	0
1	1	9	9	9	0	0	0

7.13.4 In-memory Storage Statistics and Details

Displays information about server in-memory caches, the devices that are created from this cache, and the databases on these devices.

7.13.4.1 Viewing In-memory Storage

View information about the devices that are created from in-memory storage. Details include name, size, space used, start page and number of pages of memory usage.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  name = c.name,
  status = c.status,
  size = c.value,
```

```

unusedSize = s.SizeKB,
CachePartitions = m.CachePartitions
FROM
  master..sysconfigures c,
  master..monDataCache m,
  master..monInmemoryStorage s
WHERE
  c.parent = 19
  and c.config = c.parent
  and c.status & 196640 = 196640
  and c.name = m.CacheName
  and c.name * = s.Name

```

Output

name	status	size	unusedSize	CachePartitions
IMCache	196640	10240	14336	1

7.13.4.2 Viewing In-memory Devices

View information about the devices that are created from in-memory storage. Details include name, size, space used, start page and number of pages of memory usage.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  NAME = d.name,
  SIZE = ltrim(str((1. + (d.high - d.low)) / (1048576. / @@pagesize), 10, 2)),
  USED_SIZE = (SELECT (sum(u.size) / 512.0) * (@@maxpagesize / @@pagesize)
    FROM master.dbo.sysusages u
    WHERE u.vdevno = d.vdevno),
  StartPage = d.low,
  NumPages = (high - low + 1)
FROM
  master.dbo.sysdevices d
WHERE
  d.status2 & 8 = 8
  AND d.phyname = '<cache_name>'

```

Output

NAME	SIZE	USED SIZE	StartPage	NumPages
IMDEV	6.00	6.000000	0	3072

7.13.4.3 Viewing In-memory Database

View information such as name and size of databases that are created on in-memory cache.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    d.dbid,
    DBNAME = d.name,
    SIZE = sum(v.low * (convert(float, u.size) / 1048576))
FROM
    master.dbo.sysdevices e,
    master.dbo.sysdatabases d,
    master.dbo.sysusages u,
    master.dbo.spt_values v
WHERE
    u.vdevno = e.vdevno
    and u.dbid = d.dbid
    and v.type = 'E'
    and v.number = 1
    and e.phyname = '<cache_name>'
GROUP BY
    d.name
```

Output

dbid	DBNAME	SIZE
5	IMDB	6.000000

7.13.4.4 Viewing Cached Objects

View information about tables and table indexes in the selected data cache, and their cached size, in kilobytes.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    TableIndex =case
        WHEN IndexID = 0 then convert(char(60),
            DBName + '.' + isnull(OwnerName, 'dbo') + '.' + ObjectName)
        ELSE convert(char(60), DBName + '.' + isnull(OwnerName, 'dbo') + '.' +
            object_name(ObjectID, DBID) + '.' + ObjectName)
        END,
    CachedSizeKB = CachedKB
FROM
    master.dbo.monCachedObject
WHERE
    CacheName = '<cache_name>'
    AND DBID = <database_id>
ORDER BY
    1
```

Output

TableIndex	CachedSizeKB
IMDB.dbo.sysalternates	8
IMDB.dbo.sysalternates.csysalternates	12
IMDB.dbo.sysattributes	8
IMDB.dbo.sysattributes.csysattributes	12
IMDB.dbo.sysattributes.ncsysattributes	12
IMDB.dbo.sysattributes.tsysattributes	8
IMDB.dbo.syscolumns	28
IMDB.dbo.syscolumns.csyscolumns	20
IMDB.dbo.syscomments	8
IMDB.dbo.syscomments.csyscomments	12
IMDB.dbo.sysconstraints	8
IMDB.dbo.sysconstraints.csysconstraints	12
IMDB.dbo.sysconstraints.ncsysconstraints	12
IMDB.dbo.sysdepends	8
IMDB.dbo.sysdepends.csysdepends	12
IMDB.dbo.sysencryptkeys	8
IMDB.dbo.sysencryptkeys.csysencryptkeys	12
IMDB.dbo.sysindexes	16
IMDB.dbo.sysindexes.csysindexes	12
IMDB.dbo.sysjars	8

TableIndex	CachedSizeKB
IMDB.dbo.sysjars.csysjars	12
IMDB.dbo.sysjars.ncsysjars	12
IMDB.dbo.sysjars.tsysjars	8
IMDB.dbo.syskeys	8
IMDB.dbo.syskeys.csyskeys	12
IMDB.dbo.syslogs	792
IMDB.dbo.sysobjects	12
IMDB.dbo.sysobjects.csysobjects	12
IMDB.dbo.sysobjects.ncsysobjects	12
IMDB.dbo.syspartitionkeys	8
IMDB.dbo.syspartitionkeys.csyspartitionkeys	12
IMDB.dbo.syspartitions	12
IMDB.dbo.syspartitions.csyspartitions	12
IMDB.dbo.syspartitions.nc2syspartitions	12
IMDB.dbo.syspartitions.ncsyspartitions	12
IMDB.dbo.sysprocedures	52
IMDB.dbo.sysprocedures.csysprocedures	16
IMDB.dbo.sysprotects	8
IMDB.dbo.sysprotects.csysprotects	12
IMDB.dbo.sysqueryplans	8
IMDB.dbo.sysqueryplans.csysqueryplans	12
IMDB.dbo.sysqueryplans.ncsysqueryplans	12
IMDB.dbo.sysreferences	8
IMDB.dbo.sysreferences.csysreferences	12
IMDB.dbo.sysreferences.nc2sysreferences	12
IMDB.dbo.sysreferences.ncsysreferences	12
IMDB.dbo.sysroles	8
IMDB.dbo.sysroles.csysroles	12
IMDB.dbo.syssegments	8
IMDB.dbo.syssequences	8
IMDB.dbo.syssequences.csyssequences	12
IMDB.dbo.sysslices	8
IMDB.dbo.sysslices.csysslices	8
IMDB.dbo.sysstatistics	8

TableIndex	CachedSizeKB
IMDB.dbo.sysstatistics.csystatistics	12
IMDB.dbo.systabstats	24
IMDB.dbo.systabstats.csystabstats	12
IMDB.dbo.systhresholds	8
IMDB.dbo.systypes	12
IMDB.dbo.systypes.csystypes	12
IMDB.dbo.systypes.ncsystypes	12
IMDB.dbo.sysusermessages	8
IMDB.dbo.sysusermessages.csysusermessages	8
IMDB.dbo.sysusermessages.ncsysusermessages	8
IMDB.dbo.sysusers	8
IMDB.dbo.sysusers.csysusers	12
IMDB.dbo.sysusers.ncsysusers1	12
IMDB.dbo.sysusers.ncsysusers2	12
IMDB.dbo.sysxtypes	8
IMDB.dbo.sysxtypes.csysxtypes	12
IMDB.dbo.sysxtypes.ncsysxtypes	12
IMDB.dbo.sysxtypes.tsysxtypes	8

8 Manage SAP ASE

Performing system administration tasks using SAP ASE Cockpit.

SAP ASE Cockpit allows you to manage an SAP ASE server and helps you perform complex administration tasks without the need to remember the syntax of Transact-SQL commands or system stored procedures.

8.1 Replacing Compiled Object Definitions

Replace existing compiled objects with a new definition while preserving the original name, object ID, auditing options, and permissions.

A compiled object is any object that requires entries in the `sysprocedures` table.

Compiled objects include:

- Check constraints
- Defaults
- Rules
- Stored procedures
- Extended stored procedures
- Triggers
- Views
- Functions
- Computed columns
- Partition conditions

When granular permissions is enabled or disabled, you must be the object owner to replace a compiled object. You cannot impersonate a user through an alias or by using `setuser`. However, if you are the owner through `set proxy`, you can replace a compiled object.

Replacing compiled objects is achieved by either using the Replace option from an object's context menu, or adding an object and specifying a new definition for an existing object.

When adding an object to replace a compiled object definition, you must specify the same compiled object name, owner, and database location in the Add wizard.

These objects are supported by the Replace option:

- Stored procedures
- SQLJ procedures
- Extended stored procedures
- Views
- Triggers

- Scalar user-defined functions
- SQLJ functions
- Rules
- Defaults

8.2 Manage Tables and Indexes

Create or modify tables and table objects.

Tables consist of columns and rows that contain data on a database. SAP ASE uses the following types of tables:

- A system table stores information that allows the database to perform its services.
- A user table stores and provides access to user data.
- A proxy table accesses data on remote servers.

i Note

System table definitions are not usually updated.

To plan a table's design:

- Decide what columns you need in the table, and the datatype, length, precision, and scale, for each.
- Create any new user-defined datatypes before you define the table where they are to be used.
- Decide which column, if any, should be the IDENTITY column.
- Decide which columns should and which should not accept null values.
- Decide what integrity constraints or column defaults, if any, you need to add to the columns in the table.
- Decide whether you need defaults and rules, and if so, where and what kind.
- Consider the relationship between the NULL and NOT NULL status of a column and defaults and rules.
- Decide what kind of indexes you need and where.

8.2.1 Restoring Table Data

Restore table data from an archive or stand-by database.

The following syntax provides basic guidance for using the SELECT statement with the `<into_clause>` to restore table data. For the full syntax and required permissions, see [select](#) and [using select into](#).

Syntax

```
SELECT * INTO EXISTING TABLE <target_table_name> FROM <source_table_name>
```

- `<target_table_name>` Specify the target table to hold the data.
- `<source_table_name>` Specify the source table containing the data to be restored.

8.2.2 Checking Table Consistency

Check and repair the logical and physical consistency of a table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dbcc](#).

Syntax

```
DBCC CHECKTABLE ('<table_name>')
```

```
DBCC FIX_TEXT ('<table_name>')
```

```
DBCC REINDEX ('<table_name>')
```

```
DBCC TABLEALLOC ('<table_name>', <type>, {fix | nofix} )
```

`<table_name>` Specifies the name of the table on which to perform allocation checks.

`<type>` Specifies the type of report.

```
<type> ::= {FULL | OPTIMIZED | FAST }
```

full Reports all types of allocation errors.

optimized Reports the allocation pages listed in the object allocation map (OAM) pages for the index. It does not report and cannot fix unreferenced extents on allocation pages that are not listed in the OAM pages. The default is optimized.

fast Produces an exception report of pages that are referenced but not allocated in the extent (2521-level errors).

`fix | nofix` Specifies whether or not to fix the allocation errors found in the table.

8.2.3 Placing a Table on a Segment

Using a segment to put a table on a specific database device can improve performance and give increased control over placement, size, and space usage of database objects.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_placeobject](#).

Syntax

```
EXEC sp_placeobject <segment_name>, <table_name>
```

<segment_name> Specifies the name of the segment on which to locate the table.

8.2.4 Setting a Table Locking Scheme

Choose or alter a locking scheme based on required performance.

Conversions between allpages locking and data-only locking schemes can be expensive in time and I/O and require sufficient free space. Convert the locking scheme by creating copies of the tables and re-creating indexes. You must also dump the affected databases, and update statistics before changing between allpages locking and data-only locking schemes.

For the full syntax and required permissions, see [Conversions between data page and data row locking](#) are quick and inexpensive, and implemented by updates to system tables. The data page and data row schemes are collectively called data-only locking.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> LOCK { datarows | datapages | allpages }
```

8.2.5 Reorganize Tables and Table Objects

Reorganize tables, table partitions, indexes, and index partitions to improve performance by reclaiming unused page space, removing row forwarding, or rewriting all table rows to new pages, depending on the option used.

For additional information about reorganization, see *Using the reorg Command* in the *System Administration Guide: Volume 2*.

Related Information

[Using the reorg Command](#)

8.2.5.1 Reorganizing Tables

Reorganize one or more tables to improve performance. By default, reorganization reorganizes all indexes within the selected table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [reorg](#).

Syntax

```
REORG <reorg_type> <table_name>
```

<reorg_type> Specifies the type of reorganization to perform.

```
<reorg_type> ::=  
compact  
| forwarded_rows  
| rebuild  
| reclaim_space  
| defrag
```

- | | |
|-----------------------|---|
| compact | Combines the functions of reorg reclaim_space and reorg forwarded_rows to both reclaim space and undo row forwarding in the same pass. |
| forwarded_rows | Removes row forwarding. While unforwarding or reinserting a data row on a table configured for compression, the row is compressed according to the table's compression level. |
| rebuild | <p>If a table name is specified, rewrites all rows in a table to new pages, so that the table is arranged according to its clustered index (if one exists), with all pages conforming to current space management settings and with no forwarded rows and no gaps between rows on a page. If the table has an index, all indexes are dropped and re-created. If an index name is specified, reorg rebuilds that index while leaving the table accessible for read and update activities.</p> <p>New rows follow the compression level of the partition or table, regardless of the compression level of the data in the original table or partition.</p> <p>Reorg rebuild is not supported for system catalogs.</p> |
| reclaim_space | <p>Reclaims unused space left by deletes and updates. For each data page in a table, if there is unused space resulting from committed deletes or row-shortening updates, reorg reclaim_space rewrites the current rows contiguously, leaving all unused space at the end of the page. If there are no rows on the page, the page is deallocated.</p> <p>If a table is marked for compression, reclaim_space compresses the data.</p> |

reorg reclaim_space only affects tables with variable-length rows, and only frees up space within a page. To reduce the number of pages used, use the reorg rebuild command.

defrag Reorganizes each partition list or partition in the table while allowing concurrent reads or writes on the data being reorganized. See *System Administration Guide: Volume 2 > Using the reorg Command > Running reorg defrag*.

8.2.5.2 Reorganizing Table Partitions

Reorganize a table partition to improve performance.

Reorganization is not available for all-pages-locked tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [reorg](#).

Syntax

```
REORG <reorg_type> <table_name> PARTITION <partition_name>
```

<reorg_type> Specifies the type of reorganization to perform.

```
<reorg_type> ::=  
compact  
| forwarded_rows  
| rebuild  
| reclaim_space  
| defrag
```

- compact** Combines the functions of reorg reclaim_space and reorg forwarded_rows to both reclaim space and undo row forwarding in the same pass.
- forwarded_rows** Removes row forwarding. While unforwarding or reinserting a data row on a table configured for compression, the row is compressed according to the table's compression level.
- rebuild** If a table name is specified, rewrites all rows in a table to new pages, so that the table is arranged according to its clustered index (if one exists), with all pages conforming to current space management settings and with no forwarded rows and no gaps between rows on a page. If the table has an index, all indexes are dropped and re-created. If an index name is specified, reorg rebuilds that index while leaving the table accessible for read and update activities.

New rows follow the compression level of the partition or table, regardless of the compression level of the data in the original table or partition.

Reorg rebuild is not supported for system catalogs.

reclaim_space Reclaims unused space left by deletes and updates. For each data page in a table, if there is unused space resulting from committed deletes or row-shortening updates, reorg reclaim_space rewrites the current rows contiguously, leaving all unused space at the end of the page. If there are no rows on the page, the page is deallocated.

If a table is marked for compression, reclaim_space compresses the data.

reorg reclaim_space only affects tables with variable-length rows, and only frees up space within a page. To reduce the number of pages used, use the reorg rebuild command.

defrag Reorganizes each partition list or partition in the table while allowing concurrent reads or writes on the data being reorganized. See *System Administration Guide: Volume 2 > Using the reorg Command > Running reorg defrag*.

<partition_name> Specifies the name of the partition to be reorganized.

8.2.5.3 Reorganizing Indexes

Reorganize an index to improve performance.

Reorganization is not available for all-pages-locked tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [reorg](#).

Syntax

```
REORG <reorg_type> <table_name> <index_name>
```

<reorg_type> Specifies the type of reorganization to perform.

```
<reorg_type> ::= {rebuild | reclaim_space}
```

rebuild If a table name is specified, rewrites all rows in a table to new pages, so that the table is arranged according to its clustered index (if one exists), with all pages conforming to current space management settings and with

no forwarded rows and no gaps between rows on a page. If the table has an index, all indexes are dropped and re-created. If an index name is specified, reorg rebuilds that index while leaving the table accessible for read and update activities.

New rows follow the compression level of the partition or table, regardless of the compression level of the data in the original table or partition.

Reorg rebuild is not supported for system catalogs.

reclaim_space Reclaims unused space left by deletes and updates. For each data page in a table, if there is unused space resulting from committed deletes or row-shortening updates, reorg reclaim_space rewrites the current rows contiguously, leaving all unused space at the end of the page. If there are no rows on the page, the page is deallocated.

If a table is marked for compression, reclaim_space compresses the data.

reorg reclaim_space only affects tables with variable-length rows, and only frees up space within a page. To reduce the number of pages used, use the reorg rebuild command.

<index_name> Specifies the name of the index to be reorganized.

8.2.5.4 Reorganizing Index Partitions

Reorganize an index partition to improve performance.

Reorganization is not available for all-pages-locked tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [reorg](#).

Syntax

```
REORG REBUILD <table_name> <index_name> PARTITION <partition_name>
```

8.2.6 Managing Table or Column Permissions

Grant or revoke permissions on tables or columns for users, groups, and roles. Grant or revoke permissions on tables or columns for users, groups, and roles.

You can grant and revoke permissions on a table based on the grantee type; users, groups, or roles, then select a specific grantee. You can grant or revoke permission for specific columns belonging to a table.

You can also grant permission with predicated privileges, which are privileges subject to conditions that are evaluated when data is accessed. Row-level access control can be granted on a given object based on conditions expressed through a general SQL where clause.

As an example of the where clause in a SQL statement, the following describes how to grant access to a group of engineers, allowing each member to see only his or her own salary and the salary of any direct reports.

```
GRANT SELECT ON emp (eng_salary)
  WHERE eng_name = USER OR engr = USER
  TO eng_role
```

i Note

To grant or revoke predicated privileges, set the configuration parameter enable predicate privileges to 1.

8.2.6.1 Viewing Tables and Columns Permissions

View the permissions on a column and tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helprotect](#).

Syntax

```
EXEC sp_helprotect <table_name>
```

Output

grantor	grantee	type	action	object	column	predicate	grantable
dbo	ha_role	Grant	References	dbo.mysales detail	ord_num	NULL	FALSE
dbo	ha_role	Grant	References	dbo.mysales detail	store_id	NULL	FALSE
dbo	ha_role	Grant	Select	dbo.mysales detail	qty	NULL	FALSE
dbo	ha_role	Grant	Update	dbo.mysales detail	qty	NULL	FALSE

8.2.6.2 Granting and Revoking Table Permissions

Grant and revoke table access permissions to users, groups, or roles.

Table owners and database owners can grant and revoke database object permissions on a table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT { ALL | <permission_list> } ON <table_name>
  TO { <name_list> | <role_list> } [ WITH GRANT OPTION ]
<permission_list> ::= SELECT, INISERT, DELTE, UPDATE, UPDATE STATISTICS,
  DELETE STATISTICS, REFERNCES, TRUNCATE TABLE, DECRYPT, TRANSFER TABLE
```

To revoke:

```
REVOKE { ALL | <permission_list> } ON <table_name>
  FROM { <name_list> | <role_list> }
```

8.2.6.3 Granting and Revoking Column Permissions

Grant and revoke column access permissions to users, groups, or roles.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT { ALL | <permission_list> } ON <table_name> ( <column_list> )
  TO { <name_list> | <role_list> } [ WITH GRANT OPTION ]
<permission_list> ::= SELECT, UPDATE, REFERENCES, DECRYPT
```

To revoke:

```
REVOKE { ALL | <permission_list> } ON <table_name> ( <column_list> )
  FROM { <name_list> | <role_list> }
```

8.2.7 Creating a Table

Create a user table to store data.

Only a database owner or a user with create table permission can create a table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create table](#).

Syntax

```
CREATE TABLE <table_name> ( <column_name> <datatype> [,...] )
```

8.2.8 Creating a Proxy Table

Create a proxy table to store data.

A proxy table is a user table that allows you to access data in a remote table, view, remote procedure call, directory, or file. A proxy table has all the attributes of a user table, such as columns, indexes, and triggers, but it does not contain any data locally. Only a database owner or a user with create table permission can create a table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create proxy_table](#).

Syntax

```
CREATE PROXY_TABLE <table_name> AT <path_name>
```

<path_name> Specifies the location of the remote object.

```
<path_name> ::= <server_name>[.<dbname>]  
[.<owner_name>].<object_name>
```

8.2.9 Creating a Column

Add a column to an existing table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> ADD <column_name> <datatype>
```

8.2.10 Creating an Index

An index provides quick access to data in a table, based on the values in specified columns.

An index is created on one or more table columns and points to the place where the column data is stored on disk. Indexes speed data retrieval and are useful for enforcing referential integrity. A table can have more than one index.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create index](#).

Syntax

```
CREATE INDEX <index_name> ON <table_name> (<column_expression>)
```

<column_expression> Specifies is a valid Transact-SQL expression that references at least one base column, and does not contain columns from other tables, local and global variables, aggregate functions, or subqueries.

8.2.11 Deleting a Table

Remove a table definition and all of its data, indexes, partition properties, triggers, encryption properties, and permissions from the database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop table](#).

Syntax

```
DROP TABLE <table_name>
```

8.2.12 Deleting an Index

Delete indexes created on user tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop index](#).

Syntax

```
DROP INDEX <table_name>.<index_name>
```

8.2.13 Table Level Properties

Display or modify device usage, compression, permissions, cache, and the locking scheme.

8.2.13.1 General Table Properties

Show or modify basic information on a table.

8.2.13.1.1 Viewing General Table Properties

Show general information about table properties such as name, owner, and type.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  table_name = A.name,
  table_owner = B.name,
  create_date = A.crdate,
  table_type = A.type,
  lock_scheme = A.sysstat2 & 57344,
  cache_name = (SELECT char_value
                FROM
                  <database_name>..sysattributes b,
                  <database_name>..sysobjects c
                WHERE
                  c.id = b.object
                  AND c.id = A.id
```

```

        AND object_type = 'T'),
    segment_name = (SELECT s.name
                    FROM
                        <database_name>..syssegments s,
                        <database_name>..sysindexes i
                    WHERE
                        i.segment = s.segment
                        AND i.id = A.id
                        AND i.indid < 2),
    maxrows_per_page = (SELECT i.maxrowsperpage
                       FROM
                           <database_name>..syssegments s,
                           <database_name>..sysindexes i
                       WHERE
                           i.segment = s.segment
                           AND i.id = A.id
                           AND i.indid < 2)
FROM
    <database_name>..sysobjects A,
    <database_name>..sysusers B
WHERE
    A.uid = B.uid
    AND type in ('S', 'U')
    AND A.id = <table_id>

```

Output

table_name	table_owner	create_date	table_type
mysalesdetail	dbo	Mar 11 2020 6:51PM	U

lock_scheme	cache_name	segment_name	maxrows_per_page
8192	NULL	default	0

8.2.13.1.2 Modifying a Table Name

Specify a different table name.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_rename](#).

Syntax

```
EXEC sp_rename <objectname>, <newname>
```

8.2.13.1.3 Modifying Table Bind to Cache

Select the cache to bind to the table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_bindcache](#).

Syntax

```
EXEC sp_bindcache <cache_name>, <database_name>, <table_name>
```

8.2.13.1.4 Modifying Table Identity Gap

Specify how ID numbers are allocated in memory.

For example, a value of 10 indicates ID numbers are allocated in memory in blocks of 10.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_chgattribute](#).

Syntax

```
EXEC sp_chgattribute <table_name>, "identity_gap", <set_number>
```

<set_number> Specifies the new size of the identity gap.

8.2.13.1.5 Modifying Table Data Compression

Specify the type of data compression.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SET COMPRESSION = <compression_level>
```

<compression_level> Specifies if the data in the row is compressed and to what level.

The levels are:

- 0 – the row is not compressed.
- 1 through 9 – the SAP ASE server uses ZLib compression. Generally, the higher the compression number, the more the SAP ASE server compresses the LOB data, and the greater the ratio between compressed and uncompressed data (that is the greater the amount of space savings, in bytes, for the compressed data versus the size of the uncompressed data).
However, the amount of compression depends on the LOB content, and the higher the compression level, the more CPU-intensive the process. That is, level 9 provides the highest compression ratio but also the heaviest CPU usage.
- 100 – the SAP ASE server uses FastLZ compression. The compression ratio that uses the least CPU usage; generally used for shorter data.
- 101 – the SAP ASE server uses FastLZ compression. A value of 101 uses slightly more CPU than a value of 100, but uses a better compression ratio than a value of 100.

The compression algorithm ignores rows that do not use LOB data.

8.2.13.1.6 Modifying Table Index Compression

Specify whether to apply index compression.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SET index_compression = <value>
```

<value> Specifies the index compression to be enabled or disabled to the table, index, or the local index partition. If the table is modified to be index compressed, newly created indexes are compressed.
<value> can be:

- NONE – indexes on the specified table are not compressed.
- PAGE – all indexes on the specified table are compressed.

8.2.13.17 Modifying Table LOB Compression

Specify the level of compression.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SET lob_compression = <compression_level>
```

<compression_level> Specifies if the data in the row is compressed and to what level.

The levels are:

- 0 – the row is not compressed.
- 1 through 9 – the SAP ASE server uses ZLib compression. Generally, the higher the compression number, the more the SAP ASE server compresses the LOB data, and the greater the ratio between compressed and uncompressed data (that is the greater the amount of space savings, in bytes, for the compressed data versus the size of the uncompressed data). However, the amount of compression depends on the LOB content, and the higher the compression level, the more CPU-intensive the process. That is, level 9 provides the highest compression ratio but also the heaviest CPU usage.
- 100 – the SAP ASE server uses FastLZ compression. The compression ratio that uses the least CPU usage; generally used for shorter data.
- 101 – the SAP ASE server uses FastLZ compression. A value of 101 uses slightly more CPU than a value of 100, but uses a better compression ratio than a value of 100.

The compression algorithm ignores rows that do not use LOB data.

8.2.13.18 Modifying Table Enable Incremental Transfer

Allows you to transfer data incrementally, and, if required, to a different product.

The incremental transfer feature must be available on the selected server.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SET TRANSFER TABLE { ON | OFF}
```

8.2.13.1.9 Modifying Table Erase Residual Data

Specify whether to remove residual data that is left over after some database operations.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SET ERASE RESIDUAL DATA { ON | OFF}
```

8.2.13.2 Usage Table Properties

Show or modify usage information for a table.

8.2.13.2.1 Viewing Table Usage Properties

Show space allocation for a table on a particular segment and segments to a device. Objects cannot grow beyond the space available in the segment's device.

The following syntax provides basic guidance for viewing this information.

Syntax

```
EXEC sp_spaceused <table_name>
```

Output

name	rowtotal	reserved	data	index_size	unused
mysalesdetail	0	128 KB	12 KB	4 KB	96 KB

8.2.13.2.2 Modifying Table Usage Properties

Assign space allocation for a table on a particular segment and segments to a device.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_placeobject](#).

Syntax

```
EXEC sp_placeobject <segment_name>, <table_name>
```

<segment_name> Specifies the name of the segment on which to locate the table.

8.2.13.3 Lock Scheme Properties

Specify the locking scheme to set how much data is locked at one time.

8.2.13.3.1 Viewing Table Lock Scheme Properties

Show the locking scheme on a table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    SpaceUtil = ltrim(CONVERT(varchar(32),str(ROUND(CONVERT(
        double precision,
        derived_stat(i.id, i.indid, i.partitionid, 'sput'),16),
32,16))),
    lock_schema_status = o.sysstat2 & 57344,
    o.type,
    indexNum = (SELECT count(I.name)
```

```

FROM
    <database_name>.dbo.sysindexes I,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    U.uid = O.uid
    AND I.indid between 1
    AND 254
    AND (I.status2 & 2) != 2
    AND I.id = O.id
    AND I.id = o.id)
FROM
    <database_name>..sysobjects o,
    <database_name>..systabstats i
WHERE
    o.type in ('S', 'U')
    AND i.id = o.id
    AND (((o.sysstat2 & 57344) in (0, 8192))
        AND i.indid = 1)
        OR i.indid = 0)
    AND o.id = <table_id>

```

Output

SpaceUtil	lock_schema_status	type	indexNum
0.0000000000000000	8192	U	1
0.0000000000000000	8192	U	1
0.0000000000000000	8192	U	1

8.2.13.3.2 Modifying a Table Lock Scheme Property

Specify the locking scheme to set how much data is locked at one time.

For more information about locking scheme, see *Granularity of locks and locking schemes* in *Performance and Tuning Series: Locking and Concurrency Control*.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> LOCK { datarows | datapages | allpages }
```

Related Information

[Granularity of locks and locking schemes](#)

8.2.13.3 Modifying the Amount of Data to Lock Property

Set how much data is locked at one time.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_chgattribute](#).

Syntax

To specify the row size:

```
EXEC sp_chgattribute <table_name>, "max_rows_per_page", <value>
```

To reserve a specified amount of space for the rows in data-only locked tables:

```
EXEC sp_chgattribute <table_name>, "exp_row_size", <value>
```

8.2.13.4 Viewing Table Data

View information on the table data or table contents.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT * FROM <table_name>
```

Output

DUMMY

X

8.2.13.5 Viewing Objects Referenced by a Table

Show the name, object type, AND owner of objects that reference the specified table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    object_name = A.name,
    owner_name = B.name,
    object_type = A.type
FROM
    <database_name>.dbo.sysobjects A,
    <database_name>.dbo.sysusers B
WHERE
    A.uid = B.uid
    AND type in ('S', 'U')
    AND A.id in (SELECT tableid
        FROM
            <database_name>.dbo.sysreferences
        WHERE
            reftabid = <table_id>
            AND constrid NOT IN (SELECT constrid
                FROM
                    <database_name>.dbo.sysreferences
                WHERE
                    reftabid = <table_id>
                    AND tableid = <table_id>
                    AND frgndbname = null))
UNION
SELECT
    object_name = O.name,
    owner_name = U.name,
    object_type = O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND (O.type = 'P'
        OR O.type = 'TR'
        OR O.type = 'V')
    AND O.id = D.id
    AND D.depid = <table_id>
```

Output

object_name	owner_name	object_type
js_commands	dbo	U
js_scheduledjobs	dbo	U

object_name	owner_name	object_type
sp_createjob	dbo	P
sp_bindscheduledjob	dbo	P
sp_addjobtext	dbo	P
sp_dropjobtext	dbo	P
sp_sjobmodify	dbo	P
sp_sjobcreate	dbo	P
sp_sjobcmd	dbo	P
sp_sjobcopy	dbo	P
sp_sjobcontrol	dbo	P
sp_sjobdrop	dbo	P
sp_sjobhelp	dbo	P
sp_sjobhistory	dbo	P
sp_modifyscheduledjob	dbo	P
sp_listscheduledjob	dbo	P
sp_listjob	dbo	P
sp_js_agent	dbo	P
sp_js_check_id	dbo	P

8.2.13.6 Viewing Tables Referencing Objects

View information on the table name AND owner name of objects of the specified reference.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
    distinct table_name = O.name,
    owner_name = U.name
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U,
    <database_name>.dbo.syscolumns C,
    <database_name>.dbo.sysprocedures P
WHERE
    O.uid = U.uid
    AND ((O.type = 'D' AND O.id = C.cdefault)
        OR ( O.type = 'R' AND O.id = C.domain))
    AND O.id = P.id
    AND P.sequence = 0
    AND P.status & 4096 != 4096

```

```

        AND C.id = <table_id>
UNION
SELECT
    table_name = O.name,
    owner_name = U.name
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysreferences R,
    <database_name>.dbo.sysusers U
WHERE
    O.type = 'RI'
    AND O.uid = U.uid
    AND O.id = R.constrid
    AND R.tableid = <table_id>
UNION
SELECT
    table_name = T.name,
    owner_name = U.name
FROM
    <database_name>.dbo.systypes T,
    <database_name>.dbo.sysusers U,
    <database_name>.dbo.syscolumns C,
    <database_name>.dbo.systypes P
WHERE
    T.uid = U.uid
    AND T.usertype = C.usertype
    AND T.usertype >= 100
    AND T.type = P.type
    AND P.usertype = (SELECT
                        min(usertype)
                        FROM
                            <database_name>.dbo.systypes
                        WHERE
                            type = T.type)
    AND C.id = <table_id>

```

Output

table_name	owner_name
js_command_jcmd_j_816002907	dbo

8.2.13.7 Viewing Table Columns

View information on each column belonging to the table. Clicking the column name opens the properties window for the selected column.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
```

```

column_name = C.name,
column_type = isnull(
    X.xtname,
    ISNULL(get_xtypename(C.xtype, C.xdbid), T.name)),
allows_null =case
    WHEN status & 8 = 8 THEN 'Y'
    ELSE 'N'
END,
length = C.length,
C.prec,
scale = C.scale,
rule_id = C.domain,
default_id = C.cdefault,
status = C.status,
user_type = T.usertype,
table_id = C.id,
column_id = C.colid,
char_size = isnull(@@ncharsize, 1),
xstatus = C.xstatus,
status2 = isnull(C.status2, 0),
status3 = isnull(C.status3, 0),
C.encrkeyid
FROM
    <database_name>.dbo.syscolumns C,
    <database_name>.dbo.systypes T,
    <database_name>.dbo.sysxtypes X
WHERE
    C.id = <table_id>
    AND C.usertype *= T.usertype
    AND C.xtype *= X.xtid
    AND C.name LIKE '%'

```

Output

column_name	column_type	allows_null	length	prec	scale	rule_id	default_id	status
store_id	char	N	4	NULL	NULL	0	0	0
ord_num	varchar	N	20	NULL	NULL	0	0	0
qty	smallint	N	2	NULL	NULL	0	0	0
discount	float	N	8	NULL	NULL	0	0	0

user_type	table_id	column_id	char_size	xstatus	status2	status3	encrkeyid
1	1945054934	1	1	NULL	0	0	NULL
2	1945054934	2	1	NULL	0	0	NULL
6	1945054934	3	1	NULL	0	0	NULL
8	1945054934	4	1	NULL	0	0	NULL

8.2.13.8 Table Index Properties

Displays each index belonging to the table. Clicking the index name opens the properties window for the selected index.

8.2.13.8.1 Viewing Table Indexes

Show the indexes on a table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    index_name = I.name,
    index_id = I.indid,
    I.status,
    I.status2,
    column_key_count = I.keycnt,
    I.maxrowsperpage,
    cache = (
        SELECT
            char_value
        FROM
            <database_name>.dbo.sysattributes b,
            <database_name>.dbo.sysobjects c
        WHERE
            c.id = b.object
            AND b.object_info1 = I.indid
            AND c.id = I.id
            AND object_type = 'I'
        ),
    owner = U.name
FROM
    <database_name>.dbo.sysindexes I,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    U.uid = O.uid
    AND I.id = O.id
    AND I.id = <table_id>
    AND I.indid between 1
    AND 254
    AND (I.status2 & 2) != 2
    AND I.name like '%'
```

Output

index_name	index_id	status	status2	column_key_c ount	maxrowsperp age	cache	owner
new_idx	2	0	48	3	0	default data cache	dbo

8.2.13.8.2 Viewing Table Index Unique Constraints

Show index unique constraints.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  NAME = I.name,
  OWNER = U.name,
  INDEX_ID = I.indid,
  COLUMN_COUNT = I.keycnt,
  STATUS = I.status,
  STATUS2 = I.status2,
  MAX_ROW_PER_PAGE = I.maxrowsperpage,
  FILL_FACTOR = I.fill_factor,
  RES_PAGE_GAP = I.res_page_gap,
  SEGMENT_NAME = S.name,
  TABLE_NAME = O.name
FROM
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysindexes I,
  <database_name>.dbo.sysusers U,
  <database_name>.dbo.syssegments S
WHERE
  I.id = O.id
  AND U.uid = O.uid
  AND O.id = <table_id>
  AND I.segment = S.segment
  AND I.indid between 1
  AND 254
  AND (I.status2 & 2) = 2
  AND I.name like '%'
```

Output

NAME	OWNER	INDEX_ID	COLUMN_COUNT	STATUS	STATUS2
js_command_768 0027362	dbo	2	3	2050	522

MAX_ROW_PER_PAGE	FILL_FACTOR	RES_PAGE_GAP	SEGMENT_NAME	TABLE_NAME
0	0	0	default	js_commands

8.2.13.9 Viewing Table Trigger Properties

Display each trigger belonging to the table. Clicking the trigger name opens the properties window for the selected trigger.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    trigger_name = ta.name,
    creator_name = u.name,
    creation_date = ta.crdate,
    trigger_id = ta.id,
    delete_id = tr.deltrig,
    insert_id = tr.instrig,
    update_id = tr.updtrig
FROM
    <database_name>.dbo.sysobjects ta,
    <database_name>.dbo.sysobjects tr,
    <database_name>.dbo.sysusers u
WHERE
    (ta.id = tr.deltrig
     OR ta.id = tr.instrig
     OR ta.id = tr.updtrig)
    AND ta.type = 'TR'
    AND tr.type = 'U'
    AND ta.uid = u.uid
    AND tr.id = <table_id>
    AND u.name like '%T'
    AND ta.name like '%'
```

Output

trigger_name	creator_name	creation_date	trigger_id	delete_id	insert_id	update_id
reminder	dbo	Apr 29 2020 7:50PM	764526726	0	764526726	764526726

8.2.13.10 Viewing Table Foreign Key Properties

View information on each foreign key belonging to the table. Clicking the foreign key name opens the foreign key window for the selected index.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    fk_name = O.name,
    creator_name = U.name,
    ref_db_name = isnull(R.pmarydbname, '<database_name>'),
    ref_table_id = R.reftabid,
    ref_table_name = object_name(R.reftabid, M.dbid),
    creation_date = O.crdate,
    object_id = O.id
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysreferences R,
    <database_name>.dbo.sysusers U,
    master.dbo.sysdatabases M
WHERE
    O.type = 'RI'
    AND O.uid = U.uid
    AND O.id = R.constrid
    AND R.tableid = <table_id>
    AND M.name = isnull(R.pmarydbname, '<database_name>')
```

Output

fk_name	creator_name	ref_db_name	ref_table_id	ref_table_name	creation_date	object_id
js_command_jc md_j_	dbo	sybmngmtdb	624002223	js_jobs	Mar 16 2020 2:36AM	816002907

8.2.13.11 Viewing Check Constraint Properties

View information on each check constraint belonging to the table. Clicking the check constraint name opens the properties window for the selected check constraint.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    NAME = O.name,
    OWNER = U.name,
    TEXT = C.text
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U,
    <database_name>.dbo.sysprocedures P,
    <database_name>.dbo.sysconstraints CON,
    <database_name>.dbo.syscomments C
WHERE
    C.id = O.id
    AND O.uid = U.uid
    AND O.id = P.id
    AND O.id = CON.constrid
    AND CON.tableid = <table_id>
    AND CON.colid = <column_id>
    AND P.sequence = 0
    AND P.status & 4096 = 4096
    AND O.type = 'R'
```

Output

CHECKCONSTRAINT_NAME	CHECKCONSTRAINT_OWNER
check_job_type	dbo
check_job_timeout	dbo

CHECKCONSTRAINT_ID	CHECKCONSTRAINT_CREATION_DATE
704002508	Mar 16 2020 2:36AM
720002565	Mar 16 2020 2:36AM

8.2.13.12 Viewing Partition Properties

View information on each partition for the table. Clicking the partition name opens the properties window for the selected partition.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  partition_name = case
    WHEN p.indid = 1 then p.cdaptptnname
    ELSE p.name
  END,
  s.name,
  partition_type = isnull(i.partitiontype,
    (SELECT number
     FROM
       master.dbo.spt_values
     WHERE
       name = 'roundrobin' AND type = 'PN' )),
  partition_id = isnull(p.partitionid, 0),
  i.conditionid,
  create_date = p.crdate
FROM
  <database_name>.dbo.syspartitions p,
  <database_name>.dbo.sysindexes i,
  <database_name>.dbo.syssegments s
WHERE
  i.id = p.id
  AND i.id = <table_id>
  AND p.indid < 2
  AND i.indid = p.indid
  AND s.segment = p.segment
  AND p.name LIKE '%'
```

Output

partition_name	name	partition_type	partition_id	conditioned	create_date
p1	default	2	1961054991	2009055162	Mar 11 2020 6:51PM
p2	default	2	1977055048	2009055162	Mar 11 2020 6:51PM
p3	default	2	1993055105	2009055162	Mar 11 2020 6:51PM

8.2.14 Column Level Properties

Change permissions, create check constraints, specify encryption keys, and bind rules and defaults to columns.

8.2.14.1 General Column Properties

Show or modify basic information on the column properties of a table.

8.2.14.1.1 Viewing General Column Properties

Show general information column properties.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  column_name = C.name,
  column_type = isnull(
    X.xtname,
    isnull(get_xtypename(C.xtype, C.xdbid), T.name)),
  allows_null = case
    WHEN status & 8 = 8 THEN 'Y'
    ELSE 'N'
  END,
  length = C.length,
  C.prec,
  scale = C.scale,
  status = C.status,
  user_type = T.usertype,
  table_id = C.id,
  column_id = C.colid,
  char_size = isnull(@@ncharsize, 1),
  xstatus = C.xstatus,
  status2 = isnull(C.status2, 0),
  status3 = isnull(C.status3, 0),
  C.encrkeyid
FROM
  <database_name>.dbo.syscolumns C,
  <database_name>.dbo.systypes T,
  <database_name>.dbo.sysxtypes X
WHERE
  C.id = <table_id>
  AND C.usertype *= T.usertype
  AND C.xtype *= X.xtid
  AND C.name LIKE '<column_name>'
```

Output

column_name	column_type	allows_null	length	prec	scale	status	user_type
store_id	char	N	4	NULL	NULL	0	1

table_id	column_id	char_size	xstatus	status2	status3	enckeyid
1945054934	1	1	NULL	0	0	NULL

8.2.14.1.2 Modifying General Column Properties

Change the general properties on a column.

Modify Column Name

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_rename](#).

Syntax

```
EXEC sp_rename "<table_name>.<column_name>", <new_name>
```

Modify Other Column Properties

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Task	Syntax
Change Data type	<pre>ALTER TABLE <table_name> MODIFY <column_name> <datatype></pre>
Set Primary Key	<pre>ALTER TABLE <table_name> ADD PRIMARY KEY (<column_name>)</pre>
Set Allow Nulls	<pre>ALTER TABLE <table_name> MODIFY { NULL NOT NULL }</pre>
Set Data compression	<pre>ALTER TALBE <table_name></pre>

```
MODIFY { compressed | not compressed }=
{ <compression_level> | not compressed }
```

<compression_level> Specifies if the data in the row is compressed and to what level.

The levels are:

- 0 – the row is not compressed.
- 1 through 9 – the SAP ASE server uses ZLib compression. Generally, the higher the compression number, the more the SAP ASE server compresses the LOB data, and the greater the ratio between compressed and uncompressed data (that is the greater the amount of space savings, in bytes, for the compressed data versus the size of the uncompressed data). However, the amount of compression depends on the LOB content, and the higher the compression level, the more CPU-intensive the process. That is, level 9 provides the highest compression ratio but also the heaviest CPU usage.
- 100 – the SAP ASE server uses FastLZ compression. The compression ratio that uses the least CPU usage; generally used for shorter data.
- 101 – the SAP ASE server uses FastLZ compression. A value of 101 uses slightly more CPU than a value of 100, but uses a better compression ratio than a value of 100.

The compression algorithm ignores rows that do not use LOB data.

8.2.14.2 Column Rules and Defaults

Manage column binding rules and default values.

8.2.14.2.1 Viewing Column Rules and Defaults

View the default value that appears in the column if no value is entered for an insertion or update. View the bind rules to columns to provided criteria against which is SAP ASE checks data entered for an insertion or update.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    column_name = C.name
```

```

column_type = isnull(X.xtname,
    isnull(get_xtypename(C.xtype, C.xdbid), T.name)),
allows_null = Case
    WHEN status & 8 = 8 THEN 'Y'
    ELSE 'N'
    END,
rule_id = C.domain,
default_id = C.cdefault,
FROM
    <database_name>.dbo.syscolumns C,
    <database_name>.dbo.systypes T,
    <database_name>.dbo.sysxtypes X
WHERE
    C.id = <table_id>
    AND C.usertype *= T.usertype
    AND C.xtype *= X.xtid
    AND C.name LIKE '<column_name>'

```

Output

column_name	column_type	allows_null	length	prec	scale	rule_id	default_id
store_id	char	N	4	NULL	NULL	0	0

status	user_type	table_id	column_id	char_size	xstatus	status2	status3	encrkeyid
0	1	194505493	1	1	NULL	0	0	NULL
		4						

8.2.14.2.2 Modifying Column Rules and Defaults

Change the default value that appears in the column if no value is entered for an insertion or update. Change the bind rules to columns to provided criteria against which is SAP ASE checks data entered for an insertion or update.

Set Rules

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_bindrule](#).

Syntax

```
EXEC sp_bindrule <rule_name>, "<table_name>.<column_name>"
```

Set Defaults

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> REPLACE <column_name> DEFUALT <constant_expression>
```

<constant_expression> Specifies is a constant expression to use as a default value for a column. It cannot include global variables, the name of any columns, or other database objects, but can include built-in functions. This default value must be compatible with the datatype of the column.

8.2.14.3 Viewing Check Column Constraints

View the filters that data must pass through before the data can be inserted into a table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    NAME = O.name,
    OWNER = U.name,
    TEXT = C.text
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U,
    <database_name>.dbo.sysprocedures P,
    <database_name>.dbo.sysconstraints CON,
    <database_name>.dbo.syscomments C
WHERE
    C.id = O.id
    AND O.uid = U.uid
    AND O.id = P.id
    AND O.id = CON.constrid
    AND CON.tableid = <table_id>
    AND CON.colid = <column_id>
    AND P.sequence = 0
    AND P.status & 4096 = 4096
    AND O.type = 'R'
```

Output

NAME	OWNER	TEXT
check_syscalls_enabled	dbo	CONSTRAINT check_syscalls_enabled CHECK (syscall_enabled in (0, 1))

8.2.14.4 Column Encryption

Show or modify encryption information for a column.

8.2.14.4.1 Viewing Column Encryption

View the encryption key for column encryption AND optionally a default value when you do not have decrypt permission. See the Encrypted Columns Users Guide.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  o.name,
  u.name,
  o.id,
  o.crdate
FROM
  <database_name>.dbo.sysobjects o,
  <database_name>.dbo.sysusers u,
  <database_name>.dbo.sysencryptkeys e
WHERE
  u.uid = o.uid
  AND o.type = 'EK'
  AND o.id = e.id
  AND (e.type & 4096) = 0
```

Output

encryption_key_name	owner_name	id	crdate
sybencrmasterkey	dbo	1545053509	Mar 2 2020 2:00AM

8.2.14.4.2 Modifying Column Encryption

Specify an encryption key for column encryption and optionally a default value when you do not have decrypt permission.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> MODIFY SSN ENCRYPT  
    WITH <key_name> decrypt_default <constant_expression>
```

<key_name> Identifies a key created using create encryption key.

<constant_expression> Specifies the value the SAP ASE server returns on select statements instead of the decrypted value.

8.2.15 Index Level Properties

Modify cache bindings, specify a device segment, or change index values.

8.2.15.1 General Index Properties

Show or modify general index properties for a table.

8.2.15.1.1 Viewing General Index Properties

View information on general index properties.

These properties include:

- Index name
- Index uniqueness
- Clustered
- Integrity of the index
- Current cache binding
- Cache binding
- Index compression

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  index_name = I.name,
  index_id = I.indid,
  I.status,
  I.status2,
  column_key_count = I.keycnt,
  I.maxrowsperpage,
  cache = (
    SELECT
      char_value
    FROM
      <database_name>.dbo.sysattributes b,
      <database_name>.dbo.sysobjects c
    WHERE
      c.id = b.object
      AND b.object_info1 = I.indid
      AND c.id = I.id
      AND object_type = 'I'),
  owner = U.name
FROM
  <database_name>.dbo.sysindexes I,
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysusers U
WHERE
  U.uid = O.uid
  AND I.id = O.id
  AND I.id = <table_id>
  AND I.indid between 1
  AND 254
  AND (I.status2 & 2) != 2
  AND I.name like '<index_name>'
```

Output

index_name	index_id	status	status2	column_key_c ount	maxrowsperp age	cache	owner
new_idx	2	0	48	3	0	default data cache	dbo

8.2.15.1.2 Viewing Index Compression Properties

Show index compression on a table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    status3
FROM
    <database_name>.dbo.sysindexes
WHERE
    id = <table_id>
    AND name like '<index_name>'
```

Output

status3

0

8.2.15.1.3 Modifying an Index Name Property

Specify a different index name.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_renaame](#).

Syntax

```
EXEC sp_rename "<table_name>.<index_name>", <new_name>
```

8.2.15.1.4 Modifying a Bind To Property

Change the cache binding.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_bindcache](#).

Syntax

```
EXEC sp_bindcache <cache_name>, <database_name>, <index_name>
```

8.2.15.2 Viewing Index Column Properties

View information on the columns used in the index.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  COL_NAME = index_col('<table_name>', <index_id>, <key_#>),
  TYPE = isnull(X.xtname, isnull(get_xtypename(C.xtype, C.xdbid), T.name)),
  ALLOWS_NULL = case
    WHEN status & 8 = 8 then 'Y'
    ELSE 'N'
  END,
  LEN = C.length,
  PREC = C.prec,
  SCALE = C.scale,
  RULE_ID = C.domain,
  DEFAULT_ID = C.cdefault,
  STATUS = C.status,
  USER_TYPE = T.usertype,
  TABLE_ID = C.id,
  ID = C.colid,
  CHAR_SIZE = isnull(@@charsize, 1),
  xstatus = C.xstatus,
  blankplaceholder1 = 0,
  blankplaceholder2 = 0,
  INDEX_COL_ORDER = index_colorder('<table_name>', <index_id>, <key_#> )
FROM
  <database_name>.dbo.syscolumns C,
  <database_name>.dbo.systypes T,
  <database_name>.dbo.sysxtypes X
WHERE
  C.id = <table_id>
  AND C.name = index_col('<table_name>', <index_id>, <key_#> )
  AND C.usertype *= T.usertype
  AND C.xtype *= X.xtid
ORDER BY
  colid
```

- <key_#>** Specifies a key in the index. This value is between:
- 1 and sysindexes.keycnt for a clustered index
 - 1 and sysindexes.keycnt+1 for a nonclustered index.

Output

COL_NAME	TYPE	ALLOWS_NULL	LEN	PREC	SCALE	RULE_ID
ord_num	varchar	N	20	NULL	NULL	0

DEFAULT_ID	STATUS	USER_TYPE	TABLE_ID	ID	CHAR_SIZE
0	0	2	1945054934	2	1
xstatus	blankplaceholder1	blankplaceholder2	INDEX_COL_ORDER		
NULL	0	0	DESC		

8.2.15.3 Miscellaneous Index Properties

Show modify miscellaneous information on the indexes of a table.

8.2.15.3.1 Viewing Miscellaneous Index Properties

Show information about index properties.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  index_name = I.name,
  index_id = I.indid,
  I.status,
  I.status2,
  column_key_count = I.keycnt,
  I.maxrowsperpage,
  cache = (SELECT char_value
            FROM
              <database_name>.dbo.sysattributes b,
              <database_name>.dbo.sysobjects c
            WHERE
              c.id = b.object
              AND b.object_info1 = I.indid
              AND c.id = I.id
              AND object_type = 'I'),
  owner = U.name,
  duplicate_keys = case
    WHEN I.status & 0x0002 != 0 AND I.status & 0x0001 != 0 THEN 'Ignore'
    WHEN I.status & 0x0002 != 0 AND I.status & 0x0001 = 0 THEN 'Do not
ignore'
    ELSE 'N/A' END,
  duplicate_rows = case
    WHEN I.status & 0x0002 = 0 AND I.status & 0x0040 != 0 THEN 'Allow'
    WHEN I.status & 0x0002 = 0 AND I.status & 0x0040 = 0
      AND I.status & 0x0004 != 0 THEN 'Ignore'
    WHEN I.status & 0x0002 = 0 AND I.status & 0x0040 = 0
      AND I.status & 0x0004 = 0 then 'Not allow'
    ELSE 'N/A' END,
  data_presorted = case
    WHEN I.status & 0x0200 != 0 then 'true'
    ELSE 'false' END,

```

```

most_recently_used_replacement = case
    WHEN I.status2 & 0x0020 = 0 then 'true'
    ELSE 'false' END,
large_buffer_prefetch = case
    WHEN I.status2 & 0x0010 = 0 then 'true'
    ELSE 'false' END
FROM
    <database_name>.dbo.sysindexes I,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    U.uid = O.uid
    AND I.id = O.id
    AND I.id = <table_id>
    AND I.indid between 1
    AND 254
    AND (I.status2 & 2) != 2
    AND I.name like '<index_name>'

```

Output

index_name	index_id	status	status2	column_key_count	maxrowsperpage
new_idx	2	0	48	3	0

cache	owner	duplicate_keys	duplicate_rows
default data cache	dbo	N/A	Not allow

data_presorted	most_recently_used_replacement	large_buffer_prefetch
false	true	true

8.2.15.3.2 Viewing Segment Properties

Show general segment properties.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
    I.res_page_gap,
    I.fill_factor,
    S.name
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysindexes I,
    <database_name>.dbo.sysusers U,

```

```

<database_name>.dbo.syssegments S
WHERE
  U.uid = O.uid
  and I.id = <table_id>
  and I.id = O.id
  and I.segment = S.segment
  and I.indid = <index_id>

```

Output

res_page_gap	fill_factor	name
0	0	default

8.2.15.3.3 Modifying Index Segment Properties

Change the segment on which the index is placed.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_placeobject](#).

Syntax

```
EXEC sp_placeobject <segment_name>, <index_name>
```

<segment_name> Specifies the name of the segment on which to locate the table.

8.2.15.3.4 Modifying a Cache Strategy

Specify the strategy for determining where in cache to place data pages when reading in new data. You can also choose to prefetch index pages by performing large I/Os of up to eight data pages simultaneously.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_cachestrategy](#).

Syntax

```
EXEC sp_cachestrategy <database_name>, <table_name>, <index_name>, <setting_name>
```

<setting_name> Use the mru strategy in all caches, regardless of available I/O size. Setting prefetch "on" has no effect on tables or indexes that are read into a cache that allows only 2K I/O.

```
<setting_name> ::= {prefetch | mru}, {"on" | "off"}
```

8.2.15.3.5 Modifying Index Attribute Properties

Changes the max_rows_per_page, fillfactor, or reservepagegap value for future space allocations of a table or an index.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_chgattribute](#).

Task	Syntax
Set rows per page	<pre>EXEC sp_chgattribute <index_name>, "max_rows_per_page", <value></pre>
Set reserve page gap	<pre>EXEC sp_chgattribute <index_name>,"reservepagegap", <value></pre>
Set fill factor	<pre>EXEC sp_chgattribute <index_name>, "fillfactor", <value></pre>

8.2.15.4 Viewing Index Partition Properties

Show information on indexed partitions.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  partition_name = case
    When p.indid = 1 then p.cdaptname
    ELSE p.name
  END,
  s.name,
  partitiontype = isnull(
    i.partitiontype,
    (SELECT number
     FROM
       master.dbo.spt_values
     WHERE
       name = 'roundrobin'
       AND type = 'PN')),
  partitionid = isnull(p.partitionid, 0),
  i.conditionid,
```

```

p.crdate,
status3 = isnull(i.status3, 0)
FROM
  <database_name>.dbo.syspartitions p,
  <database_name>.dbo.sysindexes i,
  <database_name>.dbo.syssegments s
WHERE
  i.id = p.id
  AND i.id = <table_id>
  AND p.indid > 0
  AND i.indid < 255
  AND i.indid = p.indid
  AND s.segment = p.segment
  AND p.name like '%'
  AND i.name = '<index_name>'

```

Output

partition_name	name	partitiontype	partitionid	conditionid	crdate	status3
new_idx_20730 55390	default	3	2073055390	0	Mar 12 2020 1:17AM	0

8.3 Manage Devices

Create, delete, and generate data definition language for database, device, and dump devices.

8.3.1 Viewing a Device Object

View information on database and in-memory device objects.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  LogicalName = d.name,
  PhysicalName = d.phyname,
  Status2 = d.status2,
  Size = ltrim(str((1.+ (d.high - d.low)) / (1048576./ @@pagesize),10,2)),
  UsedSize = (SELECT
    ((SUM(u.size) / 512.0) * (@@maxpagesize / @@pagesize))
  FROM
    master.dbo.sysusages u
  WHERE
    u.vdevno = d.vdevno),

```

```

Reads,
Writes,
APFReads,
IOTime,
DevSemaphoreRequests,
DevSemaphoreWaits
FROM
  master.dbo.sysdevices d,
  master..monDeviceIO m
WHERE
  d.cntrltype = 0
  AND m.LogicalName =* d.name

```

Output

LogicalName	PhysicalName	Status2	Size	UsedSize
IMDEV	IMCache	8	6.00	6.000000
master	/Inxpw04_tst1/ liangm/ASE160/data/ master.dat	0	100.00	98.000000
mydev	/Inxpw04_tst1/ liangm/ASE160/data/ mydev.dat	65536	100.00	6.000000
sybmgmtdev	/Inxpw04_tst1/ liangm/ASE160/data/ sybmgmtdb.dat	1	76.00	76.000000
sysprocsdev	/Inxpw04_tst1/ liangm/ASE160/data/ sysprocs.dat	1	196.00	196.000000
systemdbdev	/Inxpw04_tst1/ liangm/ASE160/data/ sybsysdb.dat	1	6.00	6.000000

Reads	Writes	APFReads	IOTime	DevSemaphoreRequests	DevSemaphoreWaits
0	0	0	0	0	0
220615	1413427	1227	553042	3	0
1540	76	20	434	0	0
4783	715590	1317	146518	0	0
9293	706475	4435	101581	2	0
18	0	0	56	2	0

8.3.2 Creating a Database Device

Create new database devices.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [disk init](#).

Syntax

```
DISK INIT name="<device_name>", physname='<physical_name>',  
size=<number_of_blocks>, skip_alloc={true | false}
```

- <device_name>** Specifies the name of the database device.
- <physical_name>** Specifies the full specification of the database device.
- skip_alloc** Specifies a Boolean parameter for the disk init command. It is supported for devices created on non-Windows file systems and on Windows raw systems. When skip_alloc is set to true, users can avoid initialization of pages with zeros. The default of skip_alloc is false.
- <number_of_blocks>** Specifies the amount of space to allocate to the new device. The following are example unit specifiers, using uppercase, lowercase, and single and double quotes interchangeably: 'k' or "K" (kilobytes), "m" or 'M' (megabytes), "g" or "G" (gigabytes), and 't' or 'T' (terabytes). You should always include a unit specifier. Quotes are optional if you do not include a unit specifier. However, you must use quotes if you include a unit specifier. Acceptable values are:
- 5120 = 10 MB
 - "5120" = 10 MB
 - "10M" = 10 MB

8.3.3 Modifying Device Sizes

Increase the size of devices.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [disk resize](#).

Syntax

```
DISK RESIZE name = "<device_name>", size = <additional_space>
```

- <device_name>** Specifies the name of the device for which you are increasing the size.
- <additional_space>** Specifies the amount of additional space you are adding to the device.

8.3.4 Creating an In-Memory Device

Create an in-memory device, or a cache device, in a cache created for an in-memory database. This device resides on an in-memory storage cache, and allows you to create in-memory databases.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [disk init](#).

Syntax

```
DISKK INIT name ="<device_name>", physname='<physical_name>',  
size = <number_of_blocks>, type = 'inmemory'
```

- <device_name>** Specifies the name of the database device.
- <physical_name>** Specifies the full specification of the database device.
- <number_of_blocks>** Specifies the amount of space to allocate to the new device. The following are example unit specifiers, using uppercase, lowercase, and single and double quotes interchangeably: 'k' or "K" (kilobytes), "m" or 'M' (megabytes), "g" or "G" (gigabytes), and 't' or "T" (terabytes). You should always include a unit specifier. Quotes are optional if you do not include a unit specifier. However, you must use quotes if you include a unit specifier. Acceptable values are:
- 5120 = 10 MB
 - "5120" = 10 MB
 - "10M" = 10 MB

8.3.5 Creating a Dump Device

Create a dump device on a server. A dump device is a tape, partition, or file used for database or transaction dumps.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addumpdevice](#).

Syntax

```
EXEC sp_addumpdevice 'disk', '<dump_name>', '<dump_path>'
```

<code><dump_name></code>	Specifies the name of dump device.
<code><dump_path></code>	Specifies the path of the dump device.

8.3.6 Database Device Properties

Display or modify device's, as well as analyze its mirror status, databases that occupy the space on the device, and segments contained in the device.

8.3.6.1 Modifying a Device Write Option Property

Change the device write option.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_diskdefault](#) and [sp_deviceattr](#).

Syntax

Specify whether or not a database device can be used for database storage if the user does not specify a database device or specifies default with the create database or alter database commands.

```
EXEC sp_diskdefault <device_name>, {defaulton | defaultoff}
```

Syntax

(UNIX platforms only) Changes the device parameter settings of an existing database device file.

```
EXEC sp_deviceattr <device_name>, dsync, {true | false}
```

8.3.6.2 Viewing Device Name and Number

Show the name and type of device.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name,
    vdevno
FROM
    master.dbo.sysdevices
WHERE
    cntrltype = 0
ORDER BY
    vdevno
```

Output

Name	vdevno
master	0
sysprocsdev	1
systemdbdev	2
tempdbdev	3
sybmgmtdev	4
mydev	5
IMDEV	6

8.3.6.3 Modifying Mirror and Disable Disk Mirror Properties

Change whether disk mirroring is enabled.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [disk mirror](#) and [sp_configure](#).

Syntax

To enable mirroring:

```
DISK MIRROR name = "<device_name>", mirror = "<physical_name>"
```

<physical_name> Specifies the full path name of the database mirror device that is to be your secondary device.

To disable mirroring:

```
EXEC sp_configure 'disable disk mirroring'
```

8.3.7 Viewing In-Memory Device Properties

Display or modify in-memory devices, or analyze the list of databases and segments contained in the in-memory device.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    NAME=d.name,
    SIZE=ltrim(str((1. + (d.high - d.low)) / (1048576. / @@pagesize), 10, 2))
FROM
    master.dbo.sysdevices d
WHERE
    d.phyname like '<cache_name>'
GROUP BY
    d.name
ORDER BY
    1
```

Output

NAME	SIZE
IMDB	6

8.3.8 Generating DDL for a Device

Generate a DDL script for devices.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server>:<port> -TDBD -N%
```

Output

DDL for DatabaseDevice 'master'

```
print '<<<<< CREATING DatabaseDevice - ""master"" >>>>>'
go
use master
go
IF EXISTS (SELECT 1 FROM master.dbo.sysdevices WHERE name = 'master')
exec sp_dropdevice master
IF (@@error != 0)
BEGIN
PRINT 'Error CREATING DatabaseDevice master'
SELECT syb_quit()
END
go
disk init name = 'master',
physname = '/Inxpw06_tst2/liangm/SAPASE/data/master.dat',
size = '100M', skip_alloc = true
go
```

8.3.9 Deleting a Database, In-Memory, or Dump Device

Delete database, in-memory, or dump devices.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropdevice](#).

Syntax

```
EXEC sp_dropdevice '<device_name>'
```

8.4 Manage Databases

Create databases, modify their properties, and perform other administrative tasks.

8.4.1 Creating a Database

Create one of several database types.

The following syntax provides basic guidance for using the CREATE DATABASE statement to create various types of databases. For the full syntax and required permissions, see [create database](#).

Database Type	Syntax
User Database	<pre>CREATE DATABASE <database_name> ON <database_device> = '<size>' LOG ON <database_device> = '<size>'</pre>
Proxy Database	<pre>CREATE DATABASE <database_name> ON DEFAULT = "<size>" WITH DEFAULT_LOCATION "<pathname>" FOR PROXY_UPDATE</pre>
Temporary Database	<pre>CREATE TEMPORARY DATABASE <database_name> ON <database_device> = <size> WITH OVERRIDE, LOB_COMPRESSION = <compression_level>, INROW_LOB_LENGTH = <value></pre>
Archive Database	<pre>CREATE ARCHIVE DATABASE <database_name> ON <database_device> = <size> WITH SCRATCH_DATABASE = <database_name></pre>
In-Memory Temporary Database	<pre>CREATE INMEMORY TEMPORARY DATABASE <database_name> ON <database_device> = "<size>" WITH DURABILITY = NO_RECOVERY</pre>

8.4.2 Creating an Archive Database

Create an archive database.

The following syntax provides basic guidance for creating an archive database. For the full syntax and required permissions, see [create archive database](#).

Syntax

```
CREATE ARCHIVE DATABASE <database_name> ON <database_device> = <size>
WITH SCRATCH_DATABASE = <database_name>
```

8.4.3 Encrypting a Database

Encrypt and decrypt a database.

The following syntax provides basic guidance for using the ALTER DATABASE statement to manage database encryption. For the full syntax and required permissions, see [alter database](#).

Encrypt

Action	Syntax
Encrypt an Existing Database	<pre>ALTER DATABASE <database_name> ENCRYPT WITH <key_name></pre>
Suspend the Encryption Process	<pre>ALTER DATABASE <database_name> SUSPEND ENCRYPTION</pre>
Resume the Encryption Process	<pre>ALTER DATABASE <database_name> RESUME ENCRYPTION</pre>

Decrypt

Action	Sample Syntax
Decrypt an Encrypted Database	<pre>ALTER DATABASE <database_name> DECRYPT WITH <key_name></pre>
Suspend the Decryption Process	<pre>ALTER DATABASE <database_name> SUSPEND DECRYPTION</pre>
Resume the Decryption Process	<pre>ALTER DATABASE <database_name> RESUME DECRYPTION</pre>

8.4.4 Creating a Temporary Database Group

Create a temporary database group.

The following syntax provides basic guidance for using the `sp_tempdb` procedure to create temporary database groups. For the full syntax and required permissions, see [sp_tempdb](#).

Action	Syntax
Create a temporary database group.	<pre>EXEC SP_TEMPDB 'CREATE', '<groupname>'</pre>
Add a database to a database group.	<pre>EXEC SP_TEMPDB 'ADD', '<tempdbname>', '<groupname>'</pre>
Bind an application to a database group.	<pre>EXEC SP_TEMPDB 'BIND', 'AP', '<objectname>', 'GR', '<groupname>'</pre>
Bind a login to a database group.	<pre>EXEC SP_TEMPDB 'BIND', 'LG', '<objectname>', 'GR', '<groupname>'</pre>

8.4.5 Viewing Database Properties

Display current database properties.

8.4.5.1 Viewing General Database Properties

View general database properties.

8.4.5.1.1 Viewing Owner, Data Cache, Location, Guest User Properties

Show the owner, data cache, location and guest user on a database.

The following syntax provides basic guidance for using a `SELECT` statement to view the owner, data cache, default location and database guest user properties for the database.

Syntax

```
SELECT  
  dbid,
```

```

owner =
  (SELECT
    name
  FROM
    master.dbo.syslogins
  WHERE
    suid = a.suid ),
creation_date = CONVERT(datetime, crdate),
last_dump_date = CONVERT(datetime, dumptrdate),
bound_cache =
  (SELECT
    char_value
  FROM
    master.dbo.sysattributes b
  WHERE
    object_type = 'D'
    and object = a.dbid
    and b.class = 3),
status,
status2,
guest =
  (SELECT name
  FROM
    <database_name>.dbo.sysusers
  WHERE
    name = 'guest'),
log_file_cache =
  (SELECT char_value
  FROM
    <database_name>.dbo.sysattributes b,
    <database_name>.dbo.sysobjects c
  WHERE
    c.id = b.object
    and object_type = 'T'
    and char_value != null
    and name = 'syslogs'),
log_io_size =
  (SELECT int_value
  FROM
    <database_name>.dbo.sysattributes b,
    <database_name>.dbo.sysobjects c
  WHERE
    c.id = b.object
    and object_type = 'T'
    and char_value = null
    and name = 'syslogs'),
last_chance = (SELECT name
  FROM
    <database_name>.dbo.syssegments s,
    <database_name>.dbo.systhresholds t
  WHERE
    s.segment = t.segment
    and s.segment = 2
    and CONVERT(bit, t.status & 1) = 1),
remote_location = a.def_remote_loc,
status4
FROM
  master.dbo.sysdatabases a
WHERE
  a.name = '<database_name>'

```

Result Set:

dbid	owner	creation_date	last_dump_date	bound_cache	status	status2
1	sa	Mar 16 2020 2:32AM	Apr 1 2020 4:00AM	NULL	0	-32768

guest	log_file_cache	log_io_size	last_chance	remote_location	status4
guest	NULL	NULL	logsegment	NULL	16384

8.4.5.1.2 Viewing the Durability Level Property

Show the durability level of the database.

Value could be:

- NO_RECOVERY – there is no guarantee that, at runtime, committed transactions are written to the disk.
- AT_SHUTDOWN – all committed transactions are written to disk during a normal server shutdown.
- FULL – a complete recovery of committed transactions is possible after a system failure.

The following syntax provides basic guidance for using a SELECT statement to view the durability level property for the database.

Syntax

```
SELECT
    durability
FROM
    master.dbo.sysdatabases
WHERE
    name = '<database_name>'
```

Result Set:

durability
1

8.4.5.1.3 Viewing the DML Logging Property

Show the DML logging.

The following syntax provides basic guidance for using a SELECT statement to view the DML logging property for the database.

Syntax

```
SELECT
    status4
FROM
    master.sysdatabases
WHERE
    name LIKE '<database_name>'
```

Result Set:

status4

536887296

8.4.5.2 Viewing Device Database Properties

View device database properties.

8.4.5.2.1 Viewing the In-memory Database Property

The following syntax provides basic guidance for using a SELECT statement to view the in-memory properties for the database.

Syntax

```
SELECT
    d.name,
    total_size = convert( int, round(a.low * (convert(float, u.size) / 1048576),
0)),
    for_log = (u.segmap & 4) / 4,
    for_data = sign(u.segmap & (~ 4)),
```

```

    d.status2,
    u.size
FROM
    master.dbo.sysusages u,
    master.dbo.spt_values a,
    master.dbo.sysdevices d
WHERE
    d.vdevno = u.vdevno
    AND u.dbid = db_id('<database_name>')
    AND d.status2 & 8 = 8
    AND a.type = 'E'
    AND a.number = 1
ORDER BY
    1

```

Result Set:

name	total_size	for_log	for_data	status2	size
IMDEV	1536	6	1	8	1536

8.4.5.2.2 Viewing the Not In-memory Database Property

The following syntax provides basic guidance for using a SELECT statement to view the not in-memory properties for the database.

Syntax

```

SELECT
    d.name,
    total_size = convert( int, round(a.low * (convert(float, u.size) / 1048576),
0)),
    for_log = (u.segmap & 4) / 4,
    for_data = sign(u.segmap & (~ 4)),
    d.status2,
    u.size
FROM
    master.dbo.sysusages u,
    master.dbo.spt_values a,
    master.dbo.sysdevices d
WHERE
    d.vdevno = u.vdevno
    AND u.dbid = db_id('<database_name>')
    AND d.status & 2 = 2
    AND a.type = 'E'
    AND a.number = 1
ORDER BY
    u.lstart

```

Result Set:

name	total_size	for_log	for_data	status2	size
master	26	1	1	0	6656
master	44	1	1	0	11264

8.4.5.3 Viewing Usage Database Properties

View usage database properties.

8.4.5.3.1 Viewing the Data Size Property

Show the space used by the tables and indexes of your database.

Use these values to determine if you have enough unreserved space to accommodate new database objects.

The following syntax provides basic guidance for viewing this information.

Syntax

To find the dbid value:

```
SELECT dbid FROM sysdatabases WHERE name = '<database_name>'
```

To return the data size:

```
SELECT SPACEUSAGE ( <dbid>)
```

Result Set:

dbid
1

reserved pages=2940, used pages=1819, data pages=1679, index pages=308, oam pages=140, allocation units=169, row count=22890, tables=65, LOB pages=16, syslogs pages=0

(1 row affected)

8.4.5.3.2 Viewing the Data and Log Space Properties

Show the space used by the tables and indexes of your database.

Use these values to determine if you have enough unreserved space to accommodate new database objects.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_spaceused](#).

Syntax

```
USE <database_name>
EXEC sp_spaceused syslogs
```

Result Set:

Name	total_pages	free_pages	used_pages	reserved_pages
syslogs	20480	17365	139	0

8.4.5.3.3 Viewing the Largest User Table Property

Show the largest user tables based on space reserved, used space, and row count.

The following examples provide basic guidance for using a SELECT statements to view the largest user table property for the database.

Syntax

```
SELECT
  TOP <top_number> table_name = O.name,
  table_id = O.id,
  db_id = db_id(),
  owner_name = U.name,
  row_count = row_count(db_id(), O.id),
  space_reserved = reserved_pages(db_id(), O.id),
  space_used = data_pages(db_id(), O.id),
  space_used_by_data = data_pages(db_id(), O.id, 0)
```

```

FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    and O.type = 'U'
    and sysstat2 & 1024 != 1024
ORDER BY
    <order_by> desc;

```

<top_number> Specifies a number that indicates the top N largest tables should be fetched.

<order_by> Specifies a string expression, by which the result set is ordered. Values can be:

- row_count
- space_reserved
- space_used

Result Set:

table_name	table_id	db_id	owner_name	row_count	space_reserved	space_used	space_used_by_data
table1	592002109	1	dbo	0	0	0	0
DBtest1	624002223	1	dbo	0	0	0	0
DBtest2	640002280	1	dbo	0	0	0	0

8.4.6 Modifying Database Properties

Modify current database properties.

8.4.6.1 Modifying the Data Cache Property

Select the cache to which you want to bind the database.

The following syntax provides basic guidance for modifying the data cache property. For the full syntax and required permissions, see [sp_bindcache](#).

Syntax

```
EXEC sp_bindcache '<cachename>', '<dbname>'
```

8.4.6.2 Modifying the Durability Level Property

Specify the durability level.

Select one of:

- NO_RECOVERY – there is no guarantee that, at runtime, committed transactions are written to the disk.
- AT_SHUTDOWN – all committed transactions are written to disk during a normal server shutdown.
- FULL – a complete recovery of committed transactions is possible after a system failure.

The following syntax provides basic guidance for modifying the durability level property. For the full syntax and required permissions, see [alter database](#).

Syntax

```
ALTER DATABASE <dbname> SET durability = NO_RECOVERY
```

8.4.6.3 Modifying the Default Location Property

Specify the default storage location for remote tables if no storage location is provided via `sp_addobjectdef`.

See the section on `sysdatabases` in the *Reference Manual: Tables*.

The following syntax provides basic guidance for modifying the default location property. For the full syntax and required permissions, see [sp_defaultloc](#).

Syntax

```
EXEC sp_defaultloc <dbname>, '<defaultloc>', 'SAPASE.master.dbo.'
```

Related Information

[sysdatabases](#)

8.4.6.4 Modifying the DML Logging Property

Enable DML logging.

The following syntax provides basic guidance for modifying the DML logging property. For the full syntax and required permissions, see [alter database](#).

Syntax

```
ALTER DATABASE <dbname> SET dml_logging = { minimal | default }
```

8.4.6.5 Modifying the Database Guest User Property

Change the guest users on the database.

The following syntax provides basic guidance for modifying the guest user property. For the full syntax and required permissions, see [sp_adduser](#).

Syntax

```
EXEC sp_adduser '<loginname>'
```

8.4.6.6 Resynchronizing Proxy Tables

Force resynchronization of proxy tables in the proxy databases.

The following syntax provides basic guidance for resynchronizing proxy tables. For the full syntax and required permissions, see [alter database](#).

Syntax

```
ALTER DATABASE <database_name> FOR proxy_update
```

8.4.7 Database Administration

Perform database administrative task such as mount or unmount a database, modify the size of a database, shrink a database, check database consistency, or place a database in quiesce hold.

8.4.7.1 Shrinking a Database

Shrink databases to freeing unused space for reuse or deletion.

The following syntax provides basic guidance for resizing a database. For the full syntax and required permissions, see [alter database](#).

Syntax

```
ALTER DATABASE <database_name> OFF <database_device> = '<size>'
```

8.4.7.2 Checking Database Consistency

Use the database consistency check to check the logical and physical consistency of a database.

The following syntax provides basic guidance for checking database consistency. For the full SQL syntax and required permissions, see [dbcc checkdb](#).

Type	Syntax
Overall	<pre>dbcc checkdb ('<database_name>') dbcc checkdb (<database_name>, skip_ncindex)</pre>
Allocation	<pre>dbcc checkalloc ('<database_name>', fix) dbcc checkalloc ('<database_name>', nofix)</pre>
Check System Catalogs	<pre>dbcc checkcatalog('<database_name>')</pre>

8.4.7.3 Placing a Database in Quiesce

Use quiesce hold to block updates to a database during a copy operation. Use quiesce release to resume database updates that were suspended by a quiesce hold command.

The following syntax provides basic guidance for adding or releasing a hold. For the full syntax and required permissions, see [quiesce database](#).

Type	Syntax
Hold	<code>QUIESCE DATABASE <tag_name> HOLD <database_list></code>
Release	<code>QUIESCE DATABASE <tag_name> RELEASE</code>

8.4.7.4 Mounting or Unmounting a Database

Mount or unmount a user database on a server.

The following syntax provides basic guidance for mounting or unmounting a database. For the full syntax and required permissions, see [mount](#) or [unmount](#).

Type	Syntax
Mount	<code>MOUNT DATABASE ALL FROM '<manifest_file>'</code>
Unmount	<code>UNMOUNT DATABASE <database_list> TO '<manifest_file>'</code>

8.4.7.5 Checkpointing Databases

Issuing a checkpoint forces SAP ASE to write modified data pages from memory to disk.

The following syntax provides basic guidance for checkpointing. For the full syntax and required permissions, see [checkpoint](#).

Syntax

```
USE <database_name>
CHECKPOINT
```

8.4.7.6 Resizing a Database

Increase the size of databases.

The following syntax provides basic guidance for resizing a database. For the full syntax and required permissions, see [alter database](#).

Syntax

```
ALTER DATABASE <database_name> ON <database_device> = <size>
```

8.4.7.7 Changing Database Ownership

Change the owner of a user database.

The following syntax provides basic guidance for changing a database's owner. For the full syntax and required permissions, see [sp_changedbowner](#).

Syntax

```
EXEC sp_changedbowner '<loginame>'
```

8.4.7.8 Adding Database Storage

Add space from a different device for your database

The following syntax provides basic guidance for adding storage space. For the full syntax and required permissions, see [alter database](#).

Syntax

```
ALTER DATABASE <database_name> ON <database_device> = <size> WITH OVERRIDE
```

8.4.7.9 Moving the Transaction Log

Move the transaction log of a database with log and data on the same device to a separate database device.

The following syntax provides basic guidance for moving the transaction log. For the full syntax and required permissions, see [sp_logdevice](#).

Syntax

```
EXEC sp_logdevice <dbname>, <devname>
```

8.4.7.10 Modifying the Transaction Log Cache

Modify the transaction log cache.

The following syntax provides basic guidance for changing the transaction log cache. For the full syntax and required permissions, see [sp_bindcache](#).

Syntax

```
EXEC sp_bindcache <cachename>, <dbname>, <tablename>
```

8.4.7.11 Modifying the Transaction Log I/O Buffer Size

Changes the log I/O size used by the SAP ASE server to a different memory pool when doing I/O for the transaction log of the current database.

The following syntax provides basic guidance for changing the buffer size. For the full syntax and required permissions, see [sp_logiosize](#).

Syntax

```
EXEC sp_logiosize '<size>'
```

8.4.7.12 Changing Database Options

Change database options.

The following syntax provides basic guidance for changing database options. For the full syntax and required permissions, see [sp_dboption](#).

Syntax

```
EXEC sp_dboption <dbname>, '<optname>', <optvalue>
<optname> ::=
    allow db suspect on rollback error
    allow incremental dumps
    allow nulls by default
    allow page signing
    allow wide dol rows
    async log service
    auto identity
    auto imrs partition tuning
    dbo use only
    ddl in tran
    deallocate first text page
    deferred table allocation
    delayed commit
    enforce dump tran sequence
    erase residual data
    full logging for all
    full logging for alter table
    full logging for reorg rebuild
    full logging for select into
    identity in nonunique index
    no chkpt on recovery
    no free space acctg
    read only
    scratch database
    select into/bulkcopy/pllsort
    single user
    trunc log on chkpt
    unique auto_identity index
<optvalue> ::= {TRUE | FALSE}
```

8.4.8 Deleting a Database

Delete database objects, or the database itself.

The following syntax provides basic guidance for deleting a database. For the full SQL syntax and required permissions, see [drop database](#).

Syntax

```
DROP DATABASE <database_name>
```

8.4.9 Generating DDL for a Database

Generate definitions for server- and database-level objects in an SAP ASE server.

The following syntax provides basic guidance for generating definitions. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<sa> -P<password> -S<server> -TDB -N<object_name> -F%
```

Output

Dependent DDL for Object(s)

```
USE master
go
PRINT "<<<< CREATE DATABASE pubs2>>>>"
go
CREATE DATABASE pubs2
ON mydev = '6M' -- 1536 pages
WITH DURABILITY = NO_RECOVERY
go
ALTER DATABASE pubs2
ON master = '1M' -- 256 pages
go
ALTER DATABASE pubs2
LOG ON devForSeg = '1M' -- 256 pages
go
sp_bindcache 'myCache', 'pubs2'
go
use pubs2
go
exec sp_changedbowner 'sa', true
```

Dependent DDL for Object(s)

```
go
exec master.dbo.sp_dboption pubs2, 'abort tran on log full', true
go
exec master.dbo.sp_dboption pubs2, 'allow incremental dumps', true
go
checkpoint
go
```

8.5 Manage Statistics

Understand the scope and freshness of server data.

Each server statistic presented has a scope:

- Delta – the number of occurrences since the last screen refresh. For example, the user log cache statistics on the Transactions screen are delta values.
- Rate – the number of occurrences over the given period of time. Device I/O is given as a rate.
- Percentage or ratio – an amount, number, or rate stated as a proportion to a whole. Percentage statistics include CPU utilization, space usage on devices and segments, and cache hit rates. Ratios include cache volatility.
- Count – a simple value; for example, the size of a database or cache in megabytes, or the number of partitions in a cache.
- Cumulative – the number of occurrences since the server started, or since the counter wrapped. On the Processes screen, you can set the Top 5 User Processes chart to display the five processes that use the most CPU, disk I/O, or network resources as rates or as cumulative values.

i Note

When a server has been running for a long time, its statistical counters might wrap, which means they restart from zero. This most often affects cumulative statistics. Information about when or how many times a counter has wrapped is not available.

Most server statistics not otherwise labeled are presented as deltas since the last screen refresh; rates, percentages, and cumulative numbers are labeled as such.

Statistics are displayed promptly. However, there are several factors that affect the freshness of the data on the screens:

- The screen refresh interval, which you can set on the Settings screen.
- The collection repeat interval, which you can set in the scheduler for each server collection
- Network latency.

For more information about interpreting server statistics, see:

- Performance and Tuning Series: Improving Performance with Statistical Analysis
- Performance and Tuning Series: Monitoring Adaptive Server with sp_sysmon
- Performance and Tuning Series: Basics

8.5.1 Updating Statistics on a Table

Update column-related statistics, such as histograms and densities.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [update statistics](#).

Syntax

```
UPDATE STATISTICS <table_name>
```

8.5.2 Updating Statistics on Specific Columns

Creating statistics on unindexed columns can improve the performance of many queries. Adding statistics for the minor columns of indexes and for unindexed columns that are frequently used in search arguments can greatly improve the optimizer's estimates.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [update statistics](#).

Syntax

```
UPDATE STATISTICS <table_name> (<column1>, <column2>, ...)
```

<table_name>	Specifies the name of the table with which the index is associated.
<column1> , <column2>	Creates the same statistics that are created if a local index is created and included on this column tuple. That is, creates a histogram on column1, and create multiattribute distributions on all column prefixes (column1, column2), (column1, column2, column3).

8.5.3 Updating Statistics on an Index

Maintaining indexes statistics can improve the performance of many queries.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [update statistics](#).

Syntax

```
UPDATE STATISTICS <table_name> <index_name>
```

8.5.4 Updating Statistics for a Data Partition

Maintaining data partition statistics can improve the performance of many queries.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [update statistics](#).

Syntax

```
UPDATE STATISTICS <table_name> PARTITION <data_partition_name>
```

<data_partition_name> Specifies the name of the partition to be updated. Statistics for each local index partition on the data partition is updated. Does not update statistics for global indexes.

8.5.5 Updating Statistics on an Index Partition

Maintaining index partition statistics can improve the performance of many queries.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [update statistics](#).

Syntax

```
UPDATE STATISTICS <index_name> PARTITION <index_partition_name>
```

`<index_partition_name>`

Specifies the name of the index partition to be updated.

8.5.6 Deleting Statistics from a Table

Drop statistics for tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [delete statistics](#).

Syntax

```
DELETE STATISTICS <table_name>
```

8.5.7 Deleting Statistics from a Column

Drop statistics for specific columns.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [delete statistics](#).

Syntax

```
DELETE STATISTICS <table_name> (<column_name>)
```

8.5.8 Deleting Statistics from a Data Partition

Drop statistics for data partitions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [delete statistics](#).

Syntax

```
DELETE STATISTICS <table_name> PARTITION <data_partition_name>
```

8.6 Manage Compression

The compression advisor provides an estimate of the percentage of space you can save by compressing a user table or index.

Compression reduces database storage space and improves system performance, especially in I/O-bound systems. Queries against compressed data can be performed with fewer I/O operations, because each read from the disk retrieves more data.

Use the database usage properties to identify tables that might benefit from compression. The compression advisor then provides recommendations for which compression attributes to apply, and a compression estimation that is based on an analysis of sampled data. The estimate is then compared to the selected table in an uncompressed state and the table in its current state, which may or may not be compressed.

Compression advisor requires an ASE_COMPRESSION license and the system-wide configuration parameter enable compression to be set to 1.

8.6.1 Identifying Table Compression Candidates

Identify the largest user tables based on space reserved, used space, or row count.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT TOP <largest_num> table_name=O.name,  
    table_id=O.id,  
    db_id=db_id(),  
    owner_name=U.name,  
    row_count=row_count(db_id(), O.id),  
    space_reserved=reserved_pages(db_id(), O.id),  
    space_used=data_pages(db_id(), O.id),  
    space_used_by_data=data_pages(db_id(),O.id, 0)  
FROM  
    <database_name>.dbo.sysobjects O,  
    <database_name>.dbo.sysusers U  
WHERE  
    O.uid = U.uid  
    AND O.type = 'U'  
    AND sysstat2 & 1024 != 1024  
ORDER BY  
    <order_para> desc
```

<largest_num>

Specifies an integer from 1 to 999.

<order_para>

Specifies the sort order. Values are:

- "row_count"
- "space_reserved"

- "space_used"

Output

table_name	table_id	db_id	owner_name
CFGCHK_RESULTS	992003534	1	35
DBH_ERROR_LOG	1184004218	1	dbo
DBH_DEADLOCKS	1152004104	1	dbo
CFGCHK_RULEBOOK	1024003648	1	dbo
ATM_WINDOWS	864003078	1	dbo
DBH_SNAP_ASERESOURCE S	1440005130	1	dbo
DBH_SNAP_IOCONTROLLE RS	1728006156	1	dbo
DBH_SNAP_TEMPDBACTIVI TY	2016007182	1	dbo
DBH_STG_FILESYSTEMS	156524560	1	dbo
DBH_STG_STMTCACHE	444525586	1	dbo

row_count	space_reserved	space_used	space_used_by_data
dbo	3426	48	36
175	16	7	4
3	8	1	1
1	8	1	1
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

8.6.2 Initiating a Compression Estimate

Apply compression attributes to a selected tables to reduce the size and improve performance for a database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [reorg](#).

Syntax

```
SET quoted_identifier on
go
USE <database_name1>
go
SELECT TOP <sampling_value> * INTO <temp_table_name> FROM <target_table>
go
ALTER TABLE <temp_table_name> SET compression = page
go
REORG REBUILD <temp_table_name>_CA_Sample
go
SET quoted_identifier off
go
```

<database_name1> Specifies the database to use for the estimate.

Prerequisites:

- The sp_dboption SELECT INTO must be enabled.
- You have permission to access the specified database.
- You have the CREATE TABLE permission.
- There is enough space to create the temporary table.

<sampling_value> Specifies the rows number loaded to the sample table.

<temp_table_name> Creates a temporary target table in which sample data is loaded from a source table. `<temp_table_name>_CA_Sample` is the default name of the sample table. its name is derived from the `<temp_table_name>` name. If the `<temp_table_name>` is *table1*, the sample table name is named *table1_CA_Sample*. The temporary tables need to be dropped after compression; however, they will consume space while Compression Advisor is running.

<target_table> The full path of the target table name on which you want to initiate a compression estimate.

```
<target_table> ::= <database_name2>.<owner_name>.<table_name>
```

8.6.3 Applying Compression Settings

The compression advisor provides compression estimates and recommendations. Based on the results of these estimates, you can apply the attributes and compress the selected table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [reorg](#).

Syntax

```
SET quoted_identifier on
```

```
go
ALTER TABLE <database_name>.<owner_name>.<table_name> SET compression = page
go
REORG REBUILD <database_name>.<owner_name>.<table_name>
go
SET quoted_identifier off
go
```

8.6.4 Managing Index Compression

Use the CREATE TABLE and CREATE INDEX to set index compression. You can subsequently turn index compression on or off on the table, index, or local index properties.

Set index compression on:

- Tables
- Indexes
- Local index partitions

Any table, index, or local index partition can be designated for index compression, except system catalogs and work tables.

The value of compression is page-level compression, which compresses redundant information on a page by storing repeated values in a single place and uses a symbol on the data page to refer to them.

8.6.4.1 Setting Index Compression on an Index

Specify index compression on an index.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter index](#).

Syntax

```
ALTER INDEX <table_name>.<index_name> SET index_compression = {page | none}
```

8.6.4.2 Setting Index Compression on a Local Index Partition

Specify index compression on a local index partition.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> MODIFY PARTITION <partition_name> SET compression =  
{page | none}
```

8.6.4.3 Setting Index Compression on a Table

Specify index compression on a table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SET compression = {page | none}
```

8.7 Manage Partitions

Use partitioning to divide large tables and indexes into smaller, more manageable pieces.

Partitions

Partitions are database objects that have unique IDs and can be managed independently. Each partition can reside on a separate segment.

Horizontal partitioning is supported, which means you can distribute a selection of table rows among storage devices. Assign individual table or index rows to a partition according to a partitioning strategy. By default, every table and index is created on a single, round-robin partition. You can also choose a semantics-based strategy that assigns rows to partitions.

Semantics-based partitioning is a separately licensed feature.

Hash Partitioning

With hash partitioning, a hash function is used to specify the partition assignment for each row. You select the partitioning key columns, but SAP ASE chooses the hash function that controls the partition assignment. Hash partitioning is a good choice for:

- Large tables with many partitions, particularly in decision-support environments
- Efficient equality searches on hash key columns
- Data that has no particular order, for example, alphanumeric product code keys

If you choose an appropriate partition key, hash partitioning distributes data evenly across all partitions. However, if you choose an inappropriate key, for example, a key that has the same value for many rows—the result may be skewed data, with an unbalanced distribution of rows among the partitions.

Range Partitioning

Rows in a range-partitioned table or index are distributed among partitions according to values in the partitioning key columns. The partitioning column values of each row are compared with a set of upper and lower bounds that determine the partition to which the row belongs.

Every partition has an inclusive upper bound and every partition except the first has a noninclusive lower bound.

Range partitioning is particularly useful for high-performance applications in both OLTP and decision-support environments. Select ranges carefully so that rows are assigned equally to all partitions—knowledge of the data distribution of the partition key columns is crucial to balancing the load evenly among the partitions. Range partitions are ordered; that is, each succeeding partition must have a higher bound than the previous partition.

List Partitioning

As with range partitioning, list partitioning distributes rows semantically; that is, according to the actual value in the partitioning key column. A list partition has only one key column. The value in the partitioning key column is compared with sets of user-supplied values to determine the partition to which each row belongs. The partition key must match exactly one of the values specified for a partition.

The value list for each partition must contain at least one value, and value lists must be unique across all partitions. You can specify as many as 250 values in each list partition. List partitions are not ordered.

Round-Robin Partitioning

In round-robin partitioning, partitioning criteria is not used. Round-robin-partitioned tables have no partition key. Rows are assigned in a round-robin manner to each partition so that each partition contains a more or less equal number of rows and load balancing is achieved. Because there is no partition key, rows are distributed randomly across all partitions.

8.7.1 Enabling Partition Locking

Partition locking improves concurrency and data availability.

Enable or disable partition locking for user tables. By default, partition locking is disabled.

By enabling partition locking, you are locking a partition of interest and therefore allowing access to other partitions for concurrent DDL and DML access options. Concurrent access is allowed for tables during the merge, move, split, and drop partition operations.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_chgattribute](#).

Syntax

```
EXEC sp_chgattribute <objname>, "ptn_locking", 1
```

8.7.2 Enabling Semantic-based Partitioning

Enable semantic-based partitions to use hash, list, or range partition strategies.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_configure](#).

Syntax

```
EXEC sp_configure 'enable semantic partitioning', 1
```

8.7.3 Using a Hash Partition

Create a new partition or change an existing partition using a strategy of a system-generated hashing function.

i Note

To create hash, list, or range partitions, turn on the server configuration parameter enable semantic partitioning.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> PARTITION BY HASH (<column_name>) <number_of_partitions>
```

- <column_name>** Specifies a partition key column. A partition key column cannot be an encrypted column.
- <number_of_partitions>** Specifies the number of partitions. Adds (<number_of_partitions> -1) empty partitions to an unpartitioned table (round-robin-partitioned table with a single partition).

8.7.4 Using a Range Partition

Create a new partition or change an existing partition according to whether one or more values in a row fall within a range of predefined values for the partition.

Note

To create hash, list, or range partitions, turn on the server configuration parameter `enable semantic partitioning`.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> PARTITION BY RANGE (<column_name>) (values <= (<constant> | MAX))
```

- <column_name>** Specifies a partition key column. A partition key column cannot be an encrypted column.
- {<constant>} | MAX** Specifies the inclusive upper bound of values for a named partition. Specifying a constant value for the highest partition bound imposes an implicit integrity constraint on the table. The keyword `MAX` specifies the maximum value in a given datatype.

8.7.5 Using a Partition List

Create a new partition or change an existing partition according to whether one value in the row matches one of a set of predefined values unique for each partition.

i Note

To create hash, list, or range partitions, turn on the server configuration parameter `enable semantic partitioning`.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> PARTITION BY LIST (<column_name>) (values <=
({<constant>}| MAX})
```

- <column_name>** Specifies that records are to be partitioned according to literal values specified in the named column. The partition key contains only one column. You can list as many as 250 constants as the partition values for each list partition.
- <constant>)| MAX** Specifies a constant value for the highest partition bound imposes an implicit integrity constraint on the table. The keyword MAX specifies the maximum value in a given datatype.

8.7.6 Using a Round-Robin Partition

Create a new partition or change an existing partition using the round-robin strategy so that each partition contains an approximately equal number of rows.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> PARTITION BY ROUNDROBIN (<number_of_partitions>)
```

- <number_of_partitions>** Specifies the number of partitions. Adds (<number_of_partitions> -1) empty partitions to an unpartitioned table (round-robin-partitioned table with a single partition).

8.7.7 Splitting a Partition

Splitting partitions can improve performance on partitions that include frequent queries and updates.

You must set the `select into/bulkcopy/pllsort` database option to `true` before you split a partition.

You may split only list and range partitions.

See the *Transact-SQL Users Guide* for information about partitions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SPLIT PARTITION <partition_name>
```

<partition_name> Specifies the name of a new partition on which table records are to stored. Partition names must be unique within the set of partitions on a table or index. Partition names can be delimited identifiers if set `quoted_identifier` is on. Otherwise, they must be valid identifiers.

Related Information

[Transact-SQL Users Guide](#)

8.7.8 Merging Partitions

Merging partitions consolidates data from partitions that are accessed infrequently into a single partition. You can merge only list and range partitions.

You can merge any two list partitions, but only adjacent range partitions. The partitions selected to be merged must be on same segment.

i Note

You must set the `select into/bulkcopy/pllsort` database option to `true` before you merge partitions.

See the *Transact-SQL Users Guide* for information about partitions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> MERGE PARTITION <partition_name> INTO  
<destination_partition_name>
```

- <partition_name>** Specifies the name of a new partition on which table records are to stored. Partition names must be unique within the set of partitions on a table or index. Partition names can be delimited identifiers if set `quoted_identifier` is on. Otherwise, they must be valid identifiers.
- <destination_partition_name>** Specifies new or existing partition. If `<destination_partition_name>` is an existing partition, it cannot be any of the source partitions you are merging. If you do not specify a destination partition name, a system-generated name is picked.

Related Information

[Transact-SQL Users Guide](#)

8.7.9 Moving a Partition

Table owners, database owners, or system administrators can move partitions to another segment.

i Note

You must set the `select into/bulkcopy/pllsort` database option to true before you split a partition.

See the [Transact-SQL Users Guide](#) for information about partitions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> MOVE PARTITION <partition_name> TO  
<destination_segment_name>
```

- <partition_name>** Specifies the name of a new partition on which table records are to stored. Partition names must be unique within the set of partitions on a table or index. Partition names can be delimited identifiers if set `quoted_identifier` is on. Otherwise, they must be valid identifiers.

`<destination_partition_name>` Specifies new or existing partition. If `<destination_partition_name>` is an existing partition, it cannot be any of the source partitions you are merging. If you do not specify a destination partition name, a system-generated name is picked.

Related Information

[Transact-SQL Users Guide](#)

8.7.10 Deleting a Partition

Drop a user or proxy table partition.

i Note

You must set the `select into/bulkcopy/pllsort` database option to true before you delete a partition.

See the [Transact-SQL Users Guide](#) for information about partitions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> DROP PARTITION <partition_name>
```

`<partition_name>` Specifies the name of a new partition on which table records are to stored. Partition names must be unique within the set of partitions on a table or index. Partition names can be delimited identifiers if set `quoted_identifier` is on. Otherwise, they must be valid identifiers.

Related Information

[Transact-SQL Users Guide](#)

8.7.11 Managing Partition Properties

Display or modify partition properties, such as the partition name, strategy, and type.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helppartition](#).

Syntax

```
EXEC sp_helppartition <table_name>
```

Output

name	type	partition_type	partitions	partition_keys	partition_name
spt_mda	base table	roundrobin	1	NULL	spt_mda_1209052312

partition_id	compression_level	pages	row_count	segment	create_date
1209052312	none	4	270	default	Mar 16 2020 2:33AM

Partition_Condition	Avg_pages	Max_pages	Min_pages	Ratio(Max/Avg)	Ratio(Min/Avg)
NULL	4	4	4	1.0	1.0

8.8 Manage Remote Servers

Add, delete, or configure remote servers using SAP ASE Cockpit.

8.8.1 Configuring a Server for Remote Procedure Calls

Configure server installations to allow request for execution of stored procedures on a remote server from a local server. The result of this request is called a remote procedure call (RPC).

Your choice of RPC handling method affects SAP ASE configuration and login mapping for remote servers. To handle RPCs, you can use either a site handler, or Component Integration Services (CIS).

The default method for handling interaction between local and remote servers is through a site handler, which creates a physical connection between the local server and the remote server. It then creates a logical

connection for each RPC to the remote server. SAP ASE creates a site handler for each remote server it connects to. A site handler is used only for connections between two server installations.

You can enable CIS for a server to request execution of stored procedures and access data on a remote server as if it were on the local server. CIS RPC handling is always used for connections involving proxy tables.

The principal difference between the two methods of handling RPCs is how the remote server views the RPC:

- If you use site handler, the remote server detects that the logical connection is made by another remote server and performs remote server verification through `sysremotelogins`.
- If you use CIS RPC handling, the remote server sees the RPC as an ordinary client connection. There is no verification using `sysremotelogins`. Therefore, connections must have a valid server login account established prior to the connection request. You cannot use trusted mode. Use of CIS RPC handling allows you to include RPCs in a transaction. Work done by an RPC can be committed or rolled back along with the other work performed in the transaction.

8.8.2 Adding a Remote Server

Add servers to your network.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addserver](#).

Syntax

```
EXEC sp_addserver <lname>
```

8.8.3 Testing a Remote Server Connection

Verify that a connection can be established between the local server and the remote server.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT 1 from master.dbo.sysservers WHERE srvname = '<server_name>'
```

8.8.4 Setting Options for a Remote Server

View or change remote server options.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_serveroption](#).

Syntax

```
EXEC sp_serveroption <server>, <optname>, <optvalue>
```

8.8.5 Managing Remote Server Login Mappings

Add, remove, and configure remote server login mappings.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addremotelogin](#).

Syntax

```
EXEC sp_addremotelogin <remoteserver>
```

<remoteserver> Specifies the name of the remote server to which the remote login applies. This server must be known to the local server by an entry in the master.dbo.sysservers table, which was created with sp_addserver.

8.8.6 Managing CIS Roles and Logins Mappings

Add, remove, and configure remote server CIS roles and logins mappings.

Logins and roles for CIS RPC handling are mapped on the local server level. By default, your local login is used as the remote login.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addexternlogin](#).

Syntax

```
EXEC sp_addexternlogin <server>, <loginame>, <externname>
```

- <server>** Specifies the name of the remote server. The <remote_server> must be known to the local server by an entry in the master.dbo.sys.servers table.
- <loginame>** Specifies an account known to the local server. <loginame> must be represented by an entry in the master.dbo.syslogins table. The sa account, the ss account, and the <loginame> account are the only users authorized to modify remote access for a given local user.
- <externname>** Specifies an account known to the <server> and must be a valid account on the node where the <server> runs. This is the account used for logging into the <server>.

8.8.7 Deleting a Remote Server

Delete a remote server from the SAP ASE system tables.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropserver](#).

Syntax

```
EXEC sp_dropserver <server>
```

8.8.8 Viewing Remote Server Properties

Display or modify remote server properties, such as change the server class, map local and remote logins, and change configuration options.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helpserver](#).

Syntax

```
EXEC sp_helpserver <server>
```

Output

name	network_name	security_mechanism	server_principal	class	status	id	cost
SYB_JSTASK	SAPASE	NULL	NULL	ASEnterprise timeouts, no net password encryption,	writable	4	1000

8.9 Manage Always-On (HADR) Option

The always-on option includes the high availability disaster recovery (HADR) feature. Administering HADR includes a number of tasks.

8.9.1 Viewing the HADR Properties

View information on properties such as the server name, the HADR mode, and host name, and the name of the replication user.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_set](#).

Syntax

```
sap_set
```

Output

PROPERTY	VALUE
maintenance_user	DR_maint
sap_sid	db1
installation_mode	nonBS
participating_databases	[master,db2,db1]
connection_timeout	5

PROPERTY	VALUE
connection_alloc_once	true
memory_size	unset
sp03st2, sybase_dir	D:\adong\sp03pl09a\DM
sp03st2, hadrflag	ha
sp03st2, ase_port	8000
sp03st2, ase_user	DR_admin
sp03st2, ase_backup_server_port	8001
sp03st2, ase_backup_server_host_override	null
sp03st2, ase_backup_server_port_override	0
sp03st2, ase_hostname	nsvxc0005.sjc.sap.corp
sp03st2, db_dump_dir	d:\adong\sp03pl09a
sp03st2, delay	0
sp03st2, ase_instance	db1_sp03st2
sp03st2, synchronization_mode	sync
sp03st2, distribution_mode	remote
sp03st2, distribution_target	sp03st1
sp03st2, replication_port	8050
sp03st2, rssid_port	8051
sp03st2, rssid_ra_port	8052
sp03st2, replication_user	DR_admin
sp03st2, device_buffer_dir	d:\adong\sp03pl09a
sp03st2, rs_hostname	nsvxc0005.sjc.sap.corp
sp03st2, device_buffer_size	2048
sp03st2, simple_persistent_queue_dir	d:\adong\sp03pl09a
sp03st2, simple_persistent_queue_size	2048
sp03st2, simple_persistent_queue_max_size	2048
sp03st2, rs_instance	db1_REP_sp03st2
sp03st2, internal_name	sp03st2
sp03st2, dr_plugin_port	8091
sp03st1, sybase_dir	D:\adong\sp03pl09a\DM
sp03st1, hadrflag	ha
sp03st1, ase_port	8000
sp03st1, ase_user	DR_admin
sp03st1, ase_backup_server_port	8001

PROPERTY	VALUE
sp03st1, ase_backup_server_host_override	null
sp03st1, ase_backup_server_port_override	0
sp03st1, ase_hostname	nsvxc0003.sjc.sap.corp
sp03st1, db_dump_dir	d:\adong\sp03pl09a
sp03st1, delay	0
sp03st1, ase_instance	db1_sp03st1
sp03st1, synchronization_mode	sync
sp03st1, distribution_mode	remote
sp03st1, distribution_target	sp03st2
sp03st1, replication_port	8050
sp03st1, rssid_port	8051
sp03st1, rssid_ra_port	8052
sp03st1, replication_user	DR_admin
sp03st1, device_buffer_dir	d:\adong\sp03pl09a
sp03st1, rs_hostname	nsvxc0003.sjc.sap.corp
sp03st1, device_buffer_size	2048
sp03st1, simple_persistent_queue_dir	d:\adong\sp03pl09a
sp03st1, simple_persistent_queue_size	2048
sp03st1, simple_persistent_queue_max_size	2048
sp03st1, rs_instance	db1_REP_sp03st1
sp03st1, internal_name	sp03st1
sp03st1, dr_plugin_port	8091

8.9.2 Viewing the HADR System Status

View information on the health and status of the HADR system.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_status](#).

Syntax

```
sap_status path
```

Output

PATH	NAME	VALUE	INFO
	Start Time	2020-06-01 01:04:57.990	Time command started executing.
	Elapsed Time	00:00:01	Command execution time.
sp03st2	Hostname	nsvxc0005.sjc.sap.corp	Logical host name.
sp03st2	HADR Status	Standby : Inactive	Identify the primary and standby sites.
sp03st2	Synchronization Mode	Synchronous	The configured Synchronization Mode value.
sp03st2	Synchronization State	Inactive	Synchronization Mode in which replication is currently operating.
sp03st2	Distribution Mode	Remote	Configured value for the distribution_mode replication model property.
sp03st2	Replication Server Status	Active	The status of Replication Server.
sp03st1	Hostname	nsvxc0003.sjc.sap.corp	Logical host name.
sp03st1	HADR Status	Primary : Active	Identify the primary and standby sites.
sp03st1	Synchronization Mode	Synchronous	The configured Synchronization Mode value.
sp03st1	Synchronization State	Synchronous	Synchronization Mode in which replication is currently operating.
sp03st1	Distribution Mode	Remote	Configured value for the distribution_mode replication model property.
sp03st1	Replication Server Status	Active	The status of Replication Server.
sp03st1.sp03st2.db1	State	Active	Path is active and replication can occur.
sp03st1.sp03st2.db1	Latency Time	2020-05-22 03:38:58.506	Time latency last calculated
sp03st1.sp03st2.db1	Latency	1643	Latency (ms)
sp03st1.sp03st2.db1	Commit Time	2020-05-22 03:38:58.506	Time last commit replicated
sp03st1.sp03st2.db1	Distribution Path	sp03st2	The path of Replication Server through which transactions travel.
sp03st1.sp03st2.db1	Drain Status	Not Applicable	The drain status of the transaction logs of the primary database server.

PATH	NAME	VALUE	INFO
sp03st1.sp03st2.db2	State	Active	Path is active and replication can occur.
sp03st1.sp03st2.db2	Latency Time	2020-05-22 03:38:58.500	Time latency last calculated
sp03st1.sp03st2.db2	Latency	1620	Latency (ms)
sp03st1.sp03st2.db2	Commit Time	2020-05-22 03:38:58.500	Time last commit replicated
sp03st1.sp03st2.db2	Distribution Path	sp03st2	The path of Replication Server through which transactions travel.
sp03st1.sp03st2.db2	Drain Status	Not Applicable	The drain status of the transaction logs of the primary database server.
sp03st1.sp03st2.master	State	Active	Path is active and replication can occur.
sp03st1.sp03st2.master	Latency Time	2020-05-22 03:39:00.126	Time latency last calculated
sp03st1.sp03st2.master	Latency	1223	Latency (ms)
sp03st1.sp03st2.master	Commit Time	2020-05-22 03:39:00.126	Time last commit replicated
sp03st1.sp03st2.master	Distribution Path	sp03st2	The path of Replication Server through which transactions travel.
sp03st1.sp03st2.master	Drain Status	Not Applicable	The drain status of the transaction logs of the primary database server.
sp03st2.sp03st1.db1	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
sp03st2.sp03st1.db1	Latency Time	Unknown	No latency information for database 'db1'.
sp03st2.sp03st1.db1	Latency	Unknown	No latency information for database 'db1'.
sp03st2.sp03st1.db1	Commit Time	Unknown	No last commit time for the database 'db1'.
sp03st2.sp03st1.db1	Distribution Path	sp03st1	The path of Replication Server through which transactions travel.
sp03st2.sp03st1.db1	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
sp03st2.sp03st1.db2	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.

PATH	NAME	VALUE	INFO
sp03st2.sp03st1.db2	Latency Time	Unknown	No latency information for database 'db2'.
sp03st2.sp03st1.db2	Latency	Unknown	No latency information for database 'db2'.
sp03st2.sp03st1.db2	Commit Time	Unknown	No last commit time for the database 'db2'.
sp03st2.sp03st1.db2	Distribution Path	sp03st1	The path of Replication Server through which transactions travel.
sp03st2.sp03st1.db2	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
sp03st2.sp03st1.master	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
sp03st2.sp03st1.master	Latency Time	Unknown	No latency information for database 'master'.
sp03st2.sp03st1.master	Latency	Unknown	No latency information for database 'master'.
sp03st2.sp03st1.master	Commit Time	Unknown	No last commit time for the database 'master'.
sp03st2.sp03st1.master	Distribution Path	sp03st1	The path of Replication Server through which transactions travel.
sp03st2.sp03st1.master	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.

8.9.3 Viewing HADR States

View information on the health and status of the HADR system.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_status](#).

Syntax

```
sap_status path
```

Output

PATH	NAME	VALUE	INFO
	Start Time	2020-06-01 01:04:57.990	Time command started executing.
	Elapsed Time	00:00:01	Command execution time.
sp03st2	Hostname	nsvxc0005.sjc.sap.corp	Logical host name.
sp03st2	HADR Status	Standby : Inactive	Identify the primary and standby sites.
sp03st2	Synchronization Mode	Synchronous	The configured Synchronization Mode value.
sp03st2	Synchronization State	Inactive	Synchronization Mode in which replication is currently operating.
sp03st2	Distribution Mode	Remote	Configured value for the distribution_mode replication model property.
sp03st2	Replication Server Status	Active	The status of Replication Server.
sp03st1	Hostname	nsvxc0003.sjc.sap.corp	Logical host name.
sp03st1	HADR Status	Primary : Active	Identify the primary and standby sites.
sp03st1	Synchronization Mode	Synchronous	The configured Synchronization Mode value.
sp03st1	Synchronization State	Synchronous	Synchronization Mode in which replication is currently operating.
sp03st1	Distribution Mode	Remote	Configured value for the distribution_mode replication model property.
sp03st1	Replication Server Status	Active	The status of Replication Server.
sp03st1.sp03st2.db1	State	Active	Path is active and replication can occur.
sp03st1.sp03st2.db1	Latency Time	2020-05-22 03:38:58.506	Time latency last calculated
sp03st1.sp03st2.db1	Latency	1643	Latency (ms)
sp03st1.sp03st2.db1	Commit Time	2020-05-22 03:38:58.506	Time last commit replicated
sp03st1.sp03st2.db1	Distribution Path	sp03st2	The path of Replication Server through which transactions travel.
sp03st1.sp03st2.db1	Drain Status	Not Applicable	The drain status of the transaction logs of the primary database server.

PATH	NAME	VALUE	INFO
sp03st1.sp03st2.db2	State	Active	Path is active and replication can occur.
sp03st1.sp03st2.db2	Latency Time	2020-05-22 03:38:58.500	Time latency last calculated
sp03st1.sp03st2.db2	Latency	1620	Latency (ms)
sp03st1.sp03st2.db2	Commit Time	2020-05-22 03:38:58.500	Time last commit replicated
sp03st1.sp03st2.db2	Distribution Path	sp03st2	The path of Replication Server through which transactions travel.
sp03st1.sp03st2.db2	Drain Status	Not Applicable	The drain status of the transaction logs of the primary database server.
sp03st1.sp03st2.master	State	Active	Path is active and replication can occur.
sp03st1.sp03st2.master	Latency Time	2020-05-22 03:39:00.126	Time latency last calculated
sp03st1.sp03st2.master	Latency	1223	Latency (ms)
sp03st1.sp03st2.master	Commit Time	2020-05-22 03:39:00.126	Time last commit replicated
sp03st1.sp03st2.master	Distribution Path	sp03st2	The path of Replication Server through which transactions travel.
sp03st1.sp03st2.master	Drain Status	Not Applicable	The drain status of the transaction logs of the primary database server.
sp03st2.sp03st1.db1	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
sp03st2.sp03st1.db1	Latency Time	Unknown	No latency information for database 'db1'.
sp03st2.sp03st1.db1	Latency	Unknown	No latency information for database 'db1'.
sp03st2.sp03st1.db1	Commit Time	Unknown	No last commit time for the database 'db1'.
sp03st2.sp03st1.db1	Distribution Path	sp03st1	The path of Replication Server through which transactions travel.
sp03st2.sp03st1.db1	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
sp03st2.sp03st1.db2	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.

PATH	NAME	VALUE	INFO
sp03st2.sp03st1.db2	Latency Time	Unknown	No latency information for database 'db2'.
sp03st2.sp03st1.db2	Latency	Unknown	No latency information for database 'db2'.
sp03st2.sp03st1.db2	Commit Time	Unknown	No last commit time for the database 'db2'.
sp03st2.sp03st1.db2	Distribution Path	sp03st1	The path of Replication Server through which transactions travel.
sp03st2.sp03st1.db2	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.
sp03st2.sp03st1.master	State	Suspended	Path is suspended (Replication Agent Thread). Transactions are not being replicated.
sp03st2.sp03st1.master	Latency Time	Unknown	No latency information for database 'master'.
sp03st2.sp03st1.master	Latency	Unknown	No latency information for database 'master'.
sp03st2.sp03st1.master	Commit Time	Unknown	No last commit time for the database 'master'.
sp03st2.sp03st1.master	Distribution Path	sp03st1	The path of Replication Server through which transactions travel.
sp03st2.sp03st1.master	Drain Status	Unknown	The drain status of the transaction logs of the primary database server.

8.9.4 Performing a Planned Failover

After a failover, the original primary server becomes the standby, and the original standby server becomes the new primary server.

You can perform failover if:

- The primary server is in an active state.
- The standby server is in an inactive state.
- All database replication to the standby server is active.
- All database replication to the primary server is suspended (without errors).

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_failover](#).

Syntax

```
EXEC sap_failover <primary_logical_host_name>, <standby_logical_host_name>,
<deactivate_timeout>
```

<primary_logical_host_name> Specifies the name of the logical host that identifies the primary site.

<standby_logical_host_name> Specifies the name of the logical host that identifies the standby site.

<deactivate_timeout> Specifies the number of seconds the process will wait while deactivating the primary data server. If the timeout reached, the failover process terminates.

Output

TASKNAME	TYPE	VALUE
Failover	Start Time	Mon Jun 01 06:26:25 PDT 2020
Failover	Elapsed Time	00:00:02
DRExecutorImpl	Task Name	Failover
DRExecutorImpl	Task State	Running
DRExecutorImpl	Short Description	Failover makes the current standby ASE as the primary server.
DRExecutorImpl	Long Description	Started task 'Failover' asynchronously.
DRExecutorImpl	Additional Info	Please execute command 'sap_status task' to determine when task 'Failover' is complete.
Failover	Task Name	Failover
Failover	Task State	Running
Failover	Short Description	Failover makes the current standby ASE as the primary server.
Failover	Long Description	Deactivate the ASE on the source host 'sp03st1'.
Failover	Current Task Number	4
Failover	Total Number of Tasks	18
Failover	Task Start	Mon Jun 01 06:26:25 PDT 2020
Failover	Hostname	nsvxc0003.sjc.sap.corp
HADRDeactivateASE	Task Name	Deactivate the ASE
HADRDeactivateASE	Task State	Running
HADRDeactivateASE	Short Description	Deactivate the primary ASE.

TASKNAME	TYPE	VALUE
HADRDeactivateASE	Long Description	Deactivate the primary, active ASE 'db1_sp03st1' on host 'sp03st1'.
HADRDeactivateASE	Hostname	nsvxc0003.sjc.sap.corp

8.9.5 Suspending Replication

You can suspend replication to an individual database or to all databases

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_suspend_replication](#).

Syntax

```
EXEC sap_suspend_replication <standby_logical_host_name>,<database_name>
```

- <standby_logical_host_name>** Specifies the standby logical host name.
- <database_name>** Suspends replication to the specified database.

8.9.6 Resuming Replication

You can resume replication to an individual database or to all databases.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_resume_replication](#).

Syntax

```
EXEC sap_resume_replication <standby_logical_host_name>, <database_name>
```

- <standby_logical_host_name>** Specifies the standby logical host name.
- <database_name>** Suspends replication to the specified database.

8.9.7 Rematerializing a Database

You can rematerialize an individual database or all databases.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_disable_replication](#).

Syntax

```
EXEC sap_disable_replication <primary_logical_host_name>
```

<primary_logical_host_name> Specifies the name of the logical host that identifies the primary site.

8.9.8 Adding an Existing Database to the HADR System Send Feedback

You can add existing databases to the HADR system.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sap_update_replication](#).

Syntax

```
EXEC sap_update_replication add_db <database_name>
```

8.9.9 Modifying the Replication Agent Configuration Parameters

Modify the configuration parameters for Rep Agent.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_config_rep_agent](#).

Syntax

```
EXEC sp_config_rep_agent <database_name>, '<RepAgent_parameter>', '<value>'
```

<RepAgent_parameter> The configuration parameter that affects RepAgent. See *RepAgent Configuration Parameters*.

Related Information

[RepAgent Configuration Parameters](#)

8.9.10 HADR KPIs

The HADR feature includes a number of KPIs.

8.9.10.1 KPIs - Latency

KPIs pertaining to latency.

KPI NAME	Description	Unit	Alert Threshold	Calculation
KPI_RS_PATH_OVERALL_TICKET_LATENCY	Overall ticket latency	Milliseconds	N:0-10000 L:10001-20000 M:20001-30000 H:30001-217484648	RDB_T - PDB_T
KPI_RS_PATH_PDB_EXEC_LATENCY	PDB to EXEC latency	Milliseconds		EXEC_T - PDB_T
KPI_RS_PATH_EXEC_DIST_LATENCY	EXEC to DIST latency	Milliseconds		DIST_T - EXEC_T
KPI_RS_PATH_DIST_DSI_LATENCY	DIST to DSI latency	Milliseconds		DSI_T - DIST_T
KPI_RS_PATH_DSI_RDB_LATENCY	DSI to RDB latency	Milliseconds		RDB_T - DSI_T
KPI_RS_PATH_DSI_CMD	DSI commands	Counts per minute		DSI_CMD
KPI_RS_PATH_DSI_TX	DSI transactions	Counts per minute		DSI_TNX
KPI_RS_PATH_EXEC_B	Bytes Received by EXEC thread	Kilobytes per minute		EXEC_B
KPI_RS_PATH_RSI_B	Bytes written by RSI thread	Kilobytes per minute		RSI_B

The following syntax provides basic guidance to return information on these KPIs. For the full syntax and required permissions, see [hadrtickethistory](#).

Syntax

```
EXEC hadrgettickethistory <logical_host>, <database_name>, '<start_time>'
```

<logical_host> Specify the name of logical ASE host.

<start_time> Specify the start date since the ticket information must be returned, the format is YYYY-MM-DD HH:MM:SS.

Output

CNT	H1	H2	H3	H4	PDB	PRS	RRS	RDB	PDB_T
1089	send_trac e	DR	Agent	GetLatenc y	dbl				
EXEC_T	DIST_T		RSI_T	DSI_T	RDB_T				
2020-06-01 07:08:56.616	1900-01-01 00:00:00.000		1900-01-01 00:00:00.000	1900-01-01 00:00:00.000	1900-01-01 00:00:00.000				
EXEC_B	RSI_B	DSI_TNX	DSI_CMD	TICKET	CONN_ID				
1900-01-01 00:00:00.000 0	0	0	0	-	0				

8.9.10.2 KPIs - FREE_XACT_LOG

KPIs pertaining to free transaction log space.

KPI Name	Description	Unit	Alert Threshold	Calculation
KPI_ASE_FREE_XACT_LOG	Amount of free transaction log space	Megabytes		FreeLogSizeMB
KPI_ASE_FREE_XACT_LOG_PERCENT	Percentage of free transaction log space	Percentage	N:31-100 L:21-30 M:11-20 H:0-10	(FreeLogSizeMB / XactLogSizeMB) * 100

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
```

```

FreeLogSizeMB = lct_admin('logsegment_freepages', db_id('<database_id>')) /
512.0 *(@@maxpagesize / @@pagesize),
XactLogSizeMB = sum(size) / 512.0 *(@@maxpagesize / @@pagesize)
FROM
master..sysusages
WHERE
dbid = db_id('<database_name>')
AND (segmap & 4 != 0)

```

Output

FreeLogSizeMB	XactLogSizeMB
NULL	2000.000000

8.9.10.3 KPIs - BACKLOG and USAGE

KPIs pertaining to backlog and usage.

KPI	Description	Unit	Alert Threshold	Calculation
KPI_RS_PRIM_QUEUE_BACKLOG	Primary replication queue backlog	Megabytes	N:active L:not active M:not active H:not active	The value in the VALUE column for each row where the TYPE column is Replication queue backlog (MB) and the NAME column is the primary site's database name.
KPI_RS_REMO_QUEUE_BACKLOG	Remote replication queue backlog	Megabytes	N:0-50 L:51-100 M:101-200 H:201-217484648	The value in the VALUE column for each row where the TYPE column is Replication queue backlog (MB) and the NAME column is the remote site's database name.
KPI_RS_PRIM_XACT_LOG_BACKLOG	Primary SAP ASE transaction log backlog	Megabytes	N:0-50 L:51-100 M:101-200 H:201-217484648	The value in the VALUE column for each row where the TYPE is ASE transaction log backlog (MB) and NAME is the primary site's database name.
KPI_RS_PRIM_DEVICE_USAGE_PERCENT	Percentage of device usage in primary Replication Server	Percentage	N:0-70 L:71-80 M:81-90 H:91-100	(primary_device_usage / primary_device_size) * 100 primary_device_usage = The value in the VALUE column for each row

KPI	Description	Unit	Alert Threshold	Calculation
				<p>where TYPE is <i>Replication device usage (MB)</i> and NAME is the name of the primary site.</p> <p>primary_device_size = The value in the VALUE column for each row where TYPE is <i>Replication device size (MB)</i> and NAME is the name of the primary site.</p>
KPI_RS_REMO_DEVI CE_USAGE_PERCEN T	Percentage of device usage in remote Replication Server	Percentage	N:0-70 L:71-80 M:81-90 H:91-100	<p>$(\text{remote_device_usage} / \text{remote_device_size}) * 100$</p> <p>remote_device_usage = The value in the VALUE column for each row where TYPE is <i>Replication device usage (MB)</i> and NAME is the name of the remote site.</p> <p>primary_device_size = The value in the VALUE column for each row where TYPE is <i>Replication device usage (MB)</i> and NAME is the name of the remote site.</p>

The following syntax provides basic guidance to return values for the following KPIs. For the full syntax and required permissions, see [sap_status resource](#).

Syntax

```
EXEC sap_status resource
```

Output

NAME	TYPE	VALUE
	Start Time	2020-06-01 06:35:54.876
	Elapsed Time	00:00:00

NAME	TYPE	VALUE
	Estimated Failover Time	-1
sp03st2	Replication device size (MB)	2048
sp03st2	Replication device usage (MB)	112
sp03st1	Replication device size (MB)	2048
sp03st1	Replication device usage (MB)	192
sp03st1.master	Replication simple persistent queue size (MB)	500
sp03st1.db2	Replication simple persistent queue size (MB)	2048
sp03st1.db1	Replication simple persistent queue size (MB)	2048
sp03st2.master	ASE transaction log size (MB)	500
sp03st2.master	ASE transaction log backlog (MB)	0
sp03st2.db2	ASE transaction log size (MB)	2000
sp03st2.db2	ASE transaction log backlog (MB)	0
sp03st2.db1	ASE transaction log size (MB)	2000
sp03st2.db1	ASE transaction log backlog (MB)	0
sp03st1.master.SPQ	Replication simple persistent queue backlog (MB)	0
sp03st1.master.IBQ	Replication inbound queue backlog (MB)	0
sp03st1.master.IBQ	Replication inbound queue truncation backlog (MB)	0
sp03st1.master.OBQ	Replication outbound queue backlog (MB)	0
sp03st1.master.OBQ	Replication outbound queue truncation backlog (MB)	0
sp03st1.db2.SPQ	Replication simple persistent queue backlog (MB)	0
sp03st1.db2.IBQ	Replication inbound queue backlog (MB)	0
sp03st1.db2.IBQ	Replication inbound queue truncation backlog (MB)	0
sp03st1.db2.OBQ	Replication outbound queue backlog (MB)	0
sp03st1.db2.OBQ	Replication outbound queue truncation backlog (MB)	0
sp03st1.db1.SPQ	Replication simple persistent queue backlog (MB)	0

NAME	TYPE	VALUE
sp03st1.db1.IBQ	Replication inbound queue backlog (MB)	0
sp03st1.db1.IBQ	Replication inbound queue truncation backlog (MB)	0
sp03st1.db1.OBQ	Replication outbound queue backlog (MB)	0
sp03st1.db1.OBQ	Replication outbound queue truncation backlog (MB)	0
sp03st1.db2	Replication queue backlog (MB)	0
sp03st1.db1	Replication queue backlog (MB)	0
sp03st1.master	Replication queue backlog (MB)	0

8.9.10.4 KPIs - LOG_RECORD

KPIs pertaining to log records scanned and processes by the Rep Agent.

KPI Name	Description	Unit	Alert Threshold	Calculation
KPI_RAT_LOG_RECORDER_SCANNED	Log Records scanned by the Rep Agent	count / minute		LogRecordsScanned
KPI_RAT_LOG_RECORDER_PROCESSED	Log Records processed by the Rep Agent	count / minute		LogRecordsProcessed

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    LogRecordsScanned,
    LogRecordsProcessed
FROM
    master..monRepLogActivity
WHERE
    DBID = db_id('<database_name>')
```

Output

LogRecordsScanned	LogRecordsProcessed
140	38

8.10 Manage Backup and Restore

Regular and frequent backups are your only protection against database damage that results from database-device failure.

8.10.1 Backing Up (Dumping) a Database

Back up a database and its transaction log.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dump database](#).

Syntax

```
DUMP DATABASE <database_name> TO '<stripe_device>'
  WITH retaindays = <number_days>,
  compression = <compress_level>,
  blocksize = <number_bytes>,
  passwd = '<password>'
```

- <stripe_device>** Specifies the device to copy the data to.
- <number_days>** A positive integer that specifies the number of days before you can overwrite the dump file or 0 to overwrite immediately.
- <compress_level>** Specifies a number between 0 and 9, 100, or 101. For single-digit compression levels, 0 indicates no compression, and 9 provides the highest level of compression. Compression levels of 100 and 101 provide a faster, more less efficient compression mode, with 100 providing faster compression and 101 providing better compression. If you do not specify **<compress_level>**, the SAP ASE server does not compress the dump.
- <number_bytes>** Overrides the default block size for a dump device. The block size must be at least one database page (2048 bytes for most systems) and must be an exact multiple of the database page size. For optimal performance, specify the blocksize as a power of 2, for example, 65536, 131072, or 262144.

8.10.1.1 Creating a Dump Configuration

System administrators can create a dump configuration.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_config_dump](#).

Syntax

```
EXEC sp_config_dump '<configuration_name>',  
    @stripe_dir = '<stripe_dir_name>',  
    @backup_svr_name = <backup_server_name>
```

<stripe_dir_name> Specifies a file system directory in which files are archived during the dump operation.

8.10.1.2 Backing Up a Database Using a Dump Configuration

Dump (back up) a database using a dump configuration.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dump database](#).

Syntax

```
DUMP DATABASE <database_name> USING config = <configuration_name>
```

8.10.1.3 Viewing Dump Configurations

View available dump configurations, which define the options used to create a database dump.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_config_dump](#).

Syntax

```
EXEC sp_config_dump
```

Output

dumpconfig

dump_config_0303

mydumpConfig

8.10.1.4 Viewing Dump Records

View available dump records.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dump_history](#).

Syntax

```
EXEC sp_dump_history 'list'
```

Output

Dump_Type	Dbid	Database_name	Stripes	Dump_instant
DATABASE	4	pubs2	1	Mar 2 2020 11:31:20:856PM
DATABASE	4	pubs2	2	Mar 3 2020 7:27:32:680PM
DATABASE	4	pubs2	2	Mar 3 2020 7:27:32:680PM
DATABASE	31513	syssystemdb	0	Mar 3 2020 1:21:02:003AM

File	Server_name	Compression_lvl	Password
/Inxpw04_tst1/liangm/ ASE160/pubs2.DB. 20200302.233119.1.000	*	0	no
/Inxpw04_tst1/liangm/ ASE160/pubs2.DB. 20200303.192732.2.000	*	2	no

File	Server_name	Compression_lvl	Password
/Inxpw04_tst1/liangm/ ASE160/pubs2.DB. 20200303.192732.2.001	*	2	no
dump.error	*	0	no

Status	Label	Dump_begin_time	Dump_end_time	Dump_size
Success	*	Mar 2 2020 11:31:19:156PM	Mar 2 2020 11:31:21:196PM	1.101563 MB
Success	*	Mar 3 2020 7:27:32:603PM	Mar 3 2020 7:27:33:570PM	1.109375 MB
Success	*	Mar 3 2020 7:27:32:603PM	Mar 3 2020 7:27:33:570PM	1.109375 MB
Error_8009	*	Mar 3 2020 1:21:01:990AM	Mar 3 2020 1:21:02:003AM	0.000000 KB

8.10.1.5 Viewing Dump Configuration Properties

Display or modify dump configuration options.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_config_dump](#).

Syntax

```
EXEC sp_config_dump <configuration_name>
```

Output

optionname	optionvalue
stripe_dir	/Inxpw04_tst1/liangm/ASE160
num_stripes	2
retry	1
blocksize	8192
compression	2
verify	header

optionname	optionvalue
notify	client

8.10.1.6 Deleting a Dump Configuration

Delete a dump configuration, database objects, or the database itself.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_config_dump](#).

Syntax

```
EXEC sp_config_dump '<configuration_name>', 'delete'
```

8.10.1.7 Backing Up a Database or Transaction Log to Multiple Stripes

Dump (back up) a database or transaction log to multiple stripes.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dump database](#) and [dump transaction](#).

Syntax

To dump a database:

```
DUMP DATABASE <database_name> TO "<stripe_device>" STRIPE ON "<stripe_device>"
```

To dump a transaction log:

```
DUMP TRANSACTION <database_name> TO <backup_tran_path1>" STRIPE ON  
"<backup_tran_path2>"
```

8.10.1.8 Backing Up a Database Incrementally

Perform a cumulative dump, which is a type of incremental dump (backup) in which only the changes since the last full database dump are backed up.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dump database](#).

Syntax

```
DUMP DATABASE <database_name> CUMULATIVE TO '<stripe_device>'
```

8.10.1.9 Backing Up a Database Incrementally Using a Dump Configuration

Using a dump configuration file, perform a cumulative dump, which is a type of incremental dump (backup) in which only changes since the last full database dump are backed up.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dump database](#).

Syntax

```
DUMP DATABASE <database_name> CUMULATIVE USING config = '<config_name>'
```

8.10.1.10 Purging Dump Records from Dump History

Purge dump records from the dump history file.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dump_history](#).

Syntax

```
EXEC sp_dump_history 'purge'
```

8.10.1.11 Generating Database-Creation SQL for a Target Database

Generate SQL and load sequences to create target databases that differ from the source databases you back up, into which you can load database dumps.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [load database](#).

Syntax

```
LOAD DATABASE <database_name> AS <target_database> WITH listonly = create_sql
```

8.10.1.12 Generating Database-Creation SQL from a Dump Image

Generate database-creation SQL for a target database from a dump image.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [load database](#).

Syntax

```
LOAD DATABASE <database_name> FROM "<dump_device>" WITH listonly = create_sql
```

<dump_device> Specifies the name of the disk database dump from which you want to load the dump.

8.10.1.13 Backing Up Server Configuration Files

Back up server configuration files.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dump configuration](#).

Syntax

```
DUMP CONFIGURATION TO '<dump_dir>' WITH file = (<file_option> [, <file_option>
[, ...]])
```

<dump_dir> Specifies path to the directory in which to save the configuration file.

<file_option> Allows you to make the backup of one or more files based on the options you specify. Valid **<file_option>** values include:

- server_config[uration] – server configuration file.
- dump_history – dump history file.
- cluster configuration file. In cluster configurations in SAP ASE Cluster Edition, each instance can have its own server configuration file. The dump configuration command dumps the server configuration files of the instance on which it was executed. The cluster configuration file is generated from the quorum device, and is named cluster.cfg with the current timestamp appended.
- 'all' – (default) all listed (known) configuration files.

If you omit **<file_option>**, all existing configuration files are backed up.

8.10.2 Scheduling a Database Backup

Database backups are an important part of system management and disaster recovery, and are typically performed by the database administrator or database owner.

By using the features of Job Scheduler, a database administrator or owner can schedule predefined database backup jobs into a prearranged sequence that includes database backups, transaction backups, and cumulative backups, which can then be automatically executed at planned times.

8.10.2.1 Requirements and Permissions for Scheduling Backups

Before you schedule backups, install the Job Scheduler database, templates, and XML files.

Make sure:

- You install sybmgmtldb, the Job Scheduler database.
- Run the installjsdb Job Scheduler installation script.
- You install the JSTemplate procedures and XML files.
- The user has either js_user_role or js_admin_role permissions.
- Ensure that you can connect to the Backup Server from each SAP ASE server you administer.
- Ensure that the login of the person starting the Backup Server has write permissions for the physical backup dump device, and that the dump device is available.

Related Information

[disk init](#)

[Installing Job Scheduler Manually](#)

[Installing Templates](#)

[Setting Up Users for Job Scheduler](#)

8.10.2.2 Creating a Schedule for Backups

Create a new schedule into which you can add a backup job.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobcreate](#).

Syntax

```
EXEC sybmgmtadb..sp_sjobcreate @name = 'sname=<sname>',  
    @option = 'repeats=<repeats>, starttime=<starttime>, endtime=<endtime>'
```

<sname> Specifies the name of the schedule.

<repeats> Specifies the interval at which the schedule repeats. 0 (zero) or NULL means it does not repeat.

8.10.2.3 Deleting a Schedule

Delete a schedule.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobdrop](#).

Syntax

```
EXEC sybmgmtadb..sp_sjobdrop 'sname=<sname>'
```

8.10.2.4 Creating a Scheduled Backup Job

Create a new scheduled backup job.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobcreate](#).

Syntax

```
EXEC sybmgmtdb..sp_sjobcreate @name = 'jname=<jname>'
GO
EXEC sybmgmtdb..sp_sjobmodify @name = 'jname=<jname>', @option = 'jcmd =
'<jcmd>'
GO
EXEC sybmgmtdb..sp_sjobcreate @name = '<sjname>', @option = 'server = '<server>
',timeout = <timeout>'
GO
```

- <jname>** Specifies the name of the job. When you create a name for a job, the name must begin with a letter. If you create a name beginning with a digit, an error occurs.
- <jcmd>** Specifies the SQL text, used for simple jobs when it is easy to provide the text directly.
- <timeout>** Specifies the maximum amount of time allowed for the execution of a scheduled job.
- <sjname>** Specifies the name of the scheduled job.
When you create a scheduled job, the name must begin with a letter. If you create a name beginning with a digit, an error occurs.

8.10.2.5 Managing a Scheduled Backup Job

Manually run an idle scheduled job from the job's context menu, or terminate, disable, or enable a running scheduled job.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobcontrol](#).

Syntax

```
EXEC sybmgmtdb..sp_sjobcontrol @name = '<name>', @option = '<option>'
```

- <name>** Specifies the name of scheduled backup job.
- <option>** Specifies a comma-separated list of options and option_name=option_value pairs that define the action to perform.

- terminate – instances of the job with the given name, ID, or runid are terminated.
- enable – instances of the scheduled job with the given name or ID are enabled.
- disable – instances of the scheduled job with the given name or ID are disabled. Any currently running jobs are unaffected.
- run_now – schedules the job with the given name or ID to run immediately. However, there must be a nonconflicting scheduled job created for the corresponding job or sp_sjobcontrol returns an error.

8.10.2.6 Rescheduling a Backup Job

Choose a different schedule for the backup job.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobdrop](#) and [sp_sjobcreate](#).

Syntax

```
EXEC sybmgmtdb.sp_sjobdrop '<name>'
GO
EXEC sybmgmtdb..sp_sjobcreate @name = '<name>', @option = ' server =
<server>,timeout = <timeout>';
GO
```

8.10.2.7 Deleting a Scheduled Backup Job

Deletes the scheduled job.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobdrop](#).

Syntax

```
EXEC sybmgmtdb..sp_sjobdrop '<name>'
GO
EXEC sybmgmtdb..sp_sjobdrop 'jname=<jname>'
GO
```

8.10.2.8 Managing the Execution History of a Scheduled Backup Job

View the execution history log and properties for all scheduled backup jobs, or delete the record of a backup job's execution history.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobhistory](#).

Syntax

```
EXEC sybmgmtdb..sp_sjobhistory @name = '<name>', @option = 'drop'
```

<name> Specifies the name or ID of the scheduled job, a job, or a runid.

8.10.3 Restoring a Database

Restore a database backup.

8.10.3.1 Restoring (Loading) a Database

Restore a database backup and its transaction log.

Decide whether to load the backup into a new database with the for load option, or into a preexisting database.

You cannot load a database backup that was created on a different operating system, or with an earlier version of SAP ASE.

i Note

When loading an archive database, the block size and compression level are set to the system default value for each stripe.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [load database](#).

Syntax

```
LOAD DATABASE <database_name> FROM '<stripe_device>'
```

8.10.3.2 Generating a Database Load Sequence

Generate a load sequence from the dump history file. You can choose to generating a load sequence for a target database that is different from the database you backed up.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [load database](#).

Syntax

```
LOAD DATABASE <database_name> WITH listonly = load_sql
```

8.10.3.3 Restoring a Database from a Cumulative Dump

Restore a database backup and its transaction log from a cumulative backup, in which only the changes since the last full database dump are backed up.

Decide whether to load the backup into a new database with the for load option, or into a preexisting database.

You cannot load a database backup that was created on a different operating system, or with an earlier version of SAP ASE.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [load database](#).

Syntax

```
LOAD DATABASE <database_name> CUMULATIVE FROM '<stripe_device>'
```

8.10.3.4 Restoring a Database from a Point in Time

Restore a database backup and its transaction log.

You cannot load a database backup that was created on a different operating system, or with an earlier version of SAP ASE.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [load transaction](#).

Syntax

```
LOAD TRANSACTION <database_name> FROM "<stripe_device>" WITH until_time =
"<datetime>"
```

8.11 Manage Execution Classes

Create, delete, and modify execution classes.

8.11.1 Creating Execution Classes

Create execution classes that can be bound to logins or tasks.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addexeclass](#).

Syntax

```
EXEC sp_addexeclass <classname>, <priority>, <timeslice>, <engine_group>
```

8.11.2 Viewing General Execution Classes Properties

Display information on an execution class.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_showexeclass](#).

Syntax

```
EXEC sp_showexeclass '<execution_class>'
```

Output

classname	priority	threadpool
EC1	HIGH	syb_default_pool

8.11.3 Binding Execution Classes

Change the scope and bindings of execution classes.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_bindindexclass](#).

Syntax

```
EXEC sp_bindindexclass '<object_name>', '<object_type>', <scope>,  
'<execution_class>'
```

8.11.4 Unbinding Execution Classes

Remove the execution class attribute previously associated with an client application, login, stored procedure, or default execution class for the specified scope.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_unbindindexclass](#).

Syntax

```
EXEC sp_unbindindexclass '<object_name>', '<object_type>', <scope>
```

8.11.5 Modifying Bindings to Execution Classes

Change the scope and bindings of execution classes.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_bindindexclass](#).

Syntax

```
EXEC sp_bindexeclass '<object_name>', '<object_type>', <scope>,  
'<execution_class>'
```

8.11.6 Viewing Binding Object

Show binding objects.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT  
    ExecutionClass = '<execution_name>',  
    BindingObject = o.name  
FROM  
    master.dbo.sysattributes e,  
    master.dbo.syslogins o  
WHERE  
    e.char_value = '<execution_name>'  
    AND e.object = o.suid
```

Output

ExecutionClass	BindingObject
EC1	EC1

8.12 Manage Engine

Manage SAP ASE engine groups.

8.12.1 Creating Engine Groups

Create groups of SAP ASE engines or processes that run in parallel.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addengine](#).

Syntax

```
EXEC sp_addengine <engine_number>, <engine_group>
```

8.12.2 Adding an Engine to an Engine Group

Add the engine to your engine group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addengine](#).

Syntax

```
EXEC sp_addengine <engine_number>, <engine_group>
```

8.12.3 Removing an Engine from an Engine Group

Remove the engine from an engine group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropengine](#).

Syntax

```
EXEC sp_dropengine <engine_number>, <engine_group>
```

8.12.4 Deleting an Engine Group

Delete an engine group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropengine](#).

Syntax

```
EXEC sp_dropengine <engine_number>, <engine_group>
```

8.12.5 Generating DDL for an Engine Group

Generate a DDL script for engine groups.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server> -TEG -N<engine_group>
```

Output

DDL for EngineGroup 'engine_group'

```
print '<<<<< CREATING EngineGroup - "engine_group" >>>>>'
go
use master
go
IF EXISTS (SELECT 1 FROM master.dbo.sysattributes WHERE class = 6 AND attribute = 1 AND object_type = 'EG' AND
object_cinfo = 'engine_group')
BEGIN
exec sp_dropengine 0, 'engine_group'
END
go
IF (@@error != 0)
BEGIN
PRINT 'Error CREATING EngineGroup engine_group'
SELECT syb_quit()
END
```

DDL for EngineGroup 'engine_group'

```
go
exec sp_addengine 0, 'engine_group'
go
```

8.13 Manage Segments

Use segments to improve performance and provide the System Administrator or Database Owner increased control over the placement, size, and space usage of database objects.

8.13.1 Viewing Segments

View information on a summary of available segments, which are labels that point to one or more database devices in your databases.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    SegmentName = s.name,
    SegmentID = s.segment,
    SizeMB = convert(numeric(18,3), convert(float, sum(u.size)) *
@@maxpagesize / 1048576),
    UnusedMB = convert( numeric(18, 3), ( convert(float, sum(u.size)) - sum( case
        WHEN (convert(float, u.size) > curunreservedpgs(u.dbid,
u.lstart, u.unreservedpgs))
        THEN convert(float, u.size) - curunreservedpgs(u.dbid,
u.lstart, u.unreservedpgs)
        ELSE 0
        END )) * @ @maxpagesize / 1048576)
FROM
    <database_name>.dbo.syssegments s,
    master.dbo.sysusages u
WHERE
    1 = 1
    AND ((u.segmap / ((s.segment & 1) + 1)) / power(2, (s.segment & 30))) & 1 = 1
    AND u.dbid = db_id('<database_name>')
GROUP BY
    s.name
ORDER BY
    UnusedMB desc;
```

Output

SegmentName	SegmentID	SizeMB	UnusedMB
system	0	12.000	8.344
default	1	12.000	8.344
logsegment	2	12.000	8.344

8.13.2 Extend a Segment

Extend a segment on a specific device.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [xp_extendsegment](#).

Syntax

```
EXEC sp_extendsegment <segname>, <dbname>, <devname>
```

- <segname>** Specifies the name of the existing segment previously defined with `sp_addsegment`.
- <dbname>** Specifies the name of the database on which to extend the segment.
- <devname>** Specifies the name of the database device to be added to the current database device range already included in `<segname>`.

8.13.3 Creating a Segment

Create a new segment, which is a label that points to one or more database devices in a database.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addsegment](#).

Syntax

```
EXEC sp_addsegment <segname>, <dbname>, <devname>
```

- <segname>** Specifies the name of the new segment to add to the `syssegments` table of the database.
- <dbname>** Specifies the name of the database in which to define the segment.

<devname> Specifies the name of the database device in which to locate **<segname>**.

8.13.4 Managing Segment Properties

Display or modify database devices, tables, and thresholds.

8.13.4.1 Viewing General Segment Properties

View information on the segment's summary information.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  *
FROM
  <database_name>.dbo.syssegments s,
  master.dbo.sysusages u
WHERE
  u.dbid = db_id('<database_name>')
  AND s.name = 'system'
```

Output

segment	name	status	dbid	segmap	lstart
0	system	0	6	7	0
size	vstart	location	unreservedpgs	crdate	vdevno
1536	3072	0	654	2020-03-02 17:56:46.883	5

8.13.4.2 Viewing Device Segment Properties

Show a list of segments contained in the device.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    device = v.name,
    segment = s.name
FROM
    <database_name>.dbo.syssegments s,
    master.dbo.sysdatabases d,
    master.dbo.sysusages u,
    master.dbo.sysdevices v
WHERE
    u.dbid = d.dbid
    AND d.name = '<db_name>'
    AND v.status & 2 = 2
    AND v.vdevno = u.vdevno
    AND ((u.segmap / ((s.segment & 1) + 1)) / power(2, (s.segment & 30))) & 1 = 1
    AND v.cntrltype = 0
    AND s.name = '<segname>'
```

Output

device	segment
master	system
systemdbdev	system

8.13.4.3 Viewing Tables that Use a Segment

Show tables that use a segment.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    Name = o.name,
    Owner = u.name
FROM
    sybmgmtdb.dbo.sysindexes i,
    sybmgmtdb.dbo.syssegments s,
    sybmgmtdb.dbo.sysobjects o,
    sybmgmtdb.dbo.sysusers u
where
    o.id = i.id
    AND u.uid = o.uid
    AND s.name = '<segname>'
    AND s.segment = i.segment
    AND i.indid != 255
order by
```

```
1,  
2
```

Output

Name	Owner
syslogs	dbo

8.13.4.4 Viewing Indexes that Use a Segment

Show the index and table name that an index uses.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT  
    name = i.name,  
    table_name = o.name  
FROM  
    sybmgmtdb.dbo.sysindexes i,  
    sybmgmtdb.dbo.syssegments s,  
    sybmgmtdb.dbo.sysobjects o,  
    sybmgmtdb.dbo.sysusers u  
WHERE  
    o.id = i.id  
    AND u.uid = o.uid  
    AND s.name = '<segname>'  
    AND s.segment = i.segment  
    AND i.indid != 255;
```

Output

name	table_name
sysobjects	sysobjects
ncsysobjects	sysobjects

8.13.4.5 Viewing Thresholds of a Segment

Show thresholds that are added to the segment in the form of system procedures and their owners.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  Free = ltrim(str(t.free_space)),
  Stored_Procedure = t.proc_name,
  Owner = user_name(t.suid)
FROM
  <database_name>.dbo.syssegments s,
  <database_name>.dbo.systhresholds t
WHERE
  s.segment = t.segment
  AND s.name = '<segname>'
```

Output

Free	Stored_Procedure	Owner
0	sp_addjobtext	dbo

8.13.5 Removing a Threshold from a Segment

Remove a threshold associated with a segment.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_droptreshold](#).

Syntax

```
EXEC sp_droptreshold <dbname>, <segname>, <free_space>
```

- <dbname>** Specifies the database from which you are dropping the threshold.
- <segname>** Specifies the segment with free space that is monitored by the threshold.
- <free_space>** Specifies the number of free pages at which the threshold is crossed.

8.13.6 Adding a Threshold to a Segment

Create a threshold to monitor space on a database segment.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addthreshold](#).

Syntax

```
EXEC sp_addthreshold <dbname>, <segname>, <free_space>, <proc_name>
```

- <dbname>** Specifies the database for which to add the threshold. This must be the name of the current database.
- <segname>** Specifies the segment for which to monitor free space. Use quotes when specifying the "default" segment.
- <free_space>** Specifies the number of free pages at which the threshold is crossed. When free space in the segment falls below this level, the SAP ASE server executes the associated stored procedure.
- <proc_name>** Specifies the stored procedure to be executed when the amount of free space on <segname> drops below <free_space>.

8.13.7 Removing a Database Device from a Segment

Remove a database device that is associated with a segment.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropsegment](#).

Syntax

```
EXEC sp_dropsegment <segname>, <dbname>, <device>
```

- <segname>** Specifies the name of the segment to be dropped.
- <dbname>** Specifies the name of the database from which the segment is to be dropped.
- <device>** Specifies the name of the database device from which the segment <segname> is to be dropped. This parameter is optional, except when the system segment system, default, or logsegment is being dropped from a database device.

8.13.8 Adding a Database Device to a Segment

Use the Segment Properties wizard to add an existing database device to a segment.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_extendsegment](#).

Syntax

```
EXEC sp_extendsegment <segname>, <dbname>, <devname>
```

- <segname>** Specifies the name of the existing segment previously defined with `sp_addsegment`.
- <dbname><dbname>** Specifies the name of the database on which to extend the segment. `<dbname>` must be the name of the current database.
- <devname>** Specifies the name of the database device to be added to the current database device range already included in `<segname>`.

8.13.9 Deleting a Segment

Delete segments.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_dropsegment](#).

Syntax

```
EXEC sp_dropsegment <segname>, <dbname>, <device>
```

8.13.10 Generating DDL for a Segment

Generate a DDL script for segments.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<username> -P<password> -S<host_name>:<port_number> -TSGM -N<object_name>.<segname>
```

Output

Dependent DDL for Object(s)

```
use pubs2
```

```
go
```

```
sp_addthreshold pubs2, 'seg_name', 16, sp_thresholdaction
```

```
go
```

```
sp_addthreshold pubs2, 'seg_name', 145, sp_js_logthreshold
```

```
go
```

8.14 Manage Thread Pools

Use thread pools to group CPU resources and execute SAP ASE tasks associated with that thread pool.

8.14.1 Creating a Thread Pool

Group SAP ASE engines into thread pools.

Thread pools are groups of resources, such as engines, that execute user tasks, run specific jobs such as signal handling, and process requests from a work queue. Both system-defined and user-defined thread pools are supported.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create thread pool](#).

Syntax

```
CREATE THREAD POOL <pool_name> WITH thread count = <count>, idle timeout = <time_period>
```

8.14.2 Viewing General Thread Pool Properties

View information on name, type, thread count, idle time out and description of thread pool.

Two types of threads are supported: Engine (or multiplexed) or Run to completion (RTC) threads. User created thread pools are always multiplexed.

You can increase the thread count up to a maximum value of max online engines configuration parameter.

The idle time out can be set :

- 0 – threads change to sleep mode if no work is available.
- -1 – threads never change to sleep mode even if no work is available.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helpthread](#).

Syntax

Displays information about all pools:

```
EXEC sp_helpthread
```

Output

Name	Type	CurrentSize	TargetSize	IdleTimeout	Description	InstanceName
syb_default_pool	Engine Multiplexed)	1	1	100	The default pool to run query sessions	(Null)

Syntax

Displays information about a specific pool:

```
EXEC sp_helpthread <thread_pool_name>
```

Output

thread_id	osthread_id	state	affinity	instance_id
0	6	140737258407680	ACTIVE	NULL

8.14.3 Viewing Thread Pool Execution Classes Properties

View the execution classes that are associated with each user or system thread pool are displayed.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_showexeclass](#).

Syntax

```
EXEC sp_showexeclass
```

Output

classname	priority	threadpool
EC1	HIGH	syb_default_pool
EC2	MEDIUM	syb_default_pool
EC3	LOW	syb_default_pool

8.15 Manage User-Defined Datatypes

Add, delete, or modify user-defined datatypes.

8.15.1 Adding a User-Defined Datatype

Name and design your own datatypes to supplement system datatypes.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_addtype](#).

Syntax

```
EXEC sp_addtype <typename>, '<phystype>', <>nulltype>
```

- <typename>** Specifies is the name of the user-defined datatype.
- <phystype>** Specifies is the physical or SAP ASE server-supplied datatype on which to base the user-defined datatype. You can specify any SAP ASE datatype except timestamp.
- <>nulltype>** Indicates how the user-defined datatype handles null value entries (null, NULL, nonull, NONULL, "not null", "NOT NULL")

8.15.2 User-Defined Datatypes Properties

Display or modify user-defined datatypes, including bound rules and defaults.

8.15.2.1 Viewing General User-Defined Datatypes Properties

View the name, type, database, AND owner of the user-defined datatype.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    t.name,
    b.name,
    p.name,
    t.usertype
FROM
    <database_name>.dbo.systypes t,
    <database_name>.dbo.sysusers b,
    <database_name>.dbo.systypes p
WHERE
    t.uid = b.uid
    AND t.usertype >= 100
    AND t.type = p.type
    AND p.usertype = (SELECT min(usertype)
        FROM
            <database_name>.dbo.systypes
        WHERE
            type = t.type)
    AND b.name like '%'
    AND t.name like '<typename>'
ORDER BY
    1
```

Output

name	name	name	usertype
mytype	dbo	bigdatetime	101

8.15.2.2 Viewing Advanced Options User-Defined Datatypes Properties

View the settings for a user-defined datatype.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    T.name,
    PHY.name,
    T.allownulls,
    T.length,
    T.prec,
    T.scale,
    T.ident,
    T.usertype
FROM
    <database_name>.dbo.systypes T,
    <database_name>.dbo.sysusers U,
    <database_name>.dbo.systypes PHY
WHERE
    T.uid = U.uid
    AND PHY.usertype = (SELECT
        min(usertype)
        FROM
            <database_name>.dbo.systypes
        WHERE
            type = T.type)
    AND T.type = PHY.type
    AND T.name = '<typename>'
```

Output

name	name	allownulls	length	prec	scale	ident	usertype
mydatatype	bigint	true	8	(Nul	(Null	0	102

8.15.2.3 Viewing Objects Referenced by a User-defined Datatype

View the name, type, owner, AND properties of objects referenced by this user-defined datatype.

The following syntax provides basic guidance for viewing this information.

Syntax

```
DELECT
  objectName = distinct O.name,
  owner = U.name,
  type = O.type
FROM
  <database_name>.dbo.syscolumns C,
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysusers U
WHERE
  (O.type = 'U'
   OR O.type = 'S')
  AND O.uid = U.uid
  AND C.id = O.id
  AND C.usertype = <typename_id>
```

Output

objectName	owner	type
Products	dbo	U

8.15.2.4 Viewing Objects a User-Defined Datatype References

View the name, type, owner, AND properties of objects that this user-defined datatype references.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  objectName = O.name,
  owner = U.name,
  type = O.type
FROM
  <database_name>.dbo.sysobjects O,
```

```
<database_name>.dbo.sysusers U,  
<database_name>.dbo.systypes T  
WHERE  
O.uid = U.uid  
AND T.domain = O.id  
AND T.usertype = <typename_id>
```

Output

objectName	owner	type
myrule	dbo	R

8.15.3 Deleting a User-Defined Datatype

Delete user-defined datatypes.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_droptype](#).

Syntax

```
EXEC sp_droptype '<typename>'
```

8.16 Manage Incrementally Transferring Data

Incremental data transfer lets you transfer data to SAP ASE or other products.

8.16.1 Enabling Incremental Transfer

You can designate incremental transfer eligibility on an existing table, or when you create a table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> SET TRANSFER TABLE ON
```

8.16.2 Incrementally Transferring Data In

Use incremental transfer to read data files into SAP ASE.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [transfer table](#).

Syntax

```
TRANSFER TABLE <table_name> FROM '<destination_file>' FOR ASE
```

<destination_file> Specifies any file or path name that is valid for the operating system, and that the SAP ASE server can access.

If the file is a relative file path, the SAP ASE server provides its absolute path. If the SAP ASE server cannot open **<destination_file>**, it issues an error message.

The SAP ASE server removes a destination file if all these conditions apply:

- The file is a regular file, not a named pipe, symbolic link, or other special file.
- The SAP ASE server opens the file as part of this transfer.
- The transfer fails, or sends no rows.

8.16.3 Incrementally Transferring Data Out

Transfer table data that has changed since a prior transmission from tables that are marked for incremental transfer.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [transfer table](#).

Syntax

```
TRANSFER TABLE <table_name> TO '<destination_file>' FOR ASE  
    WITH column_order = 'offset', encryption = '<option>', tracking_id = <nnn>,  
    resend = <id>
```

<destination_file> Specifies any file or path name that is valid for the operating system, and that the SAP ASE server can access.

If the file is a relative file path, the SAP ASE server provides its absolute path. If the SAP ASE server cannot open **<destination_file>**, it issues an error message.

The SAP ASE server removes a destination file if all these conditions apply:

- The file is a regular file, not a named pipe, symbolic link, or other special file.
- The SAP ASE server opens the file as part of this transfer.
- The transfer fails, or sends no rows.

<option> Specifies how the command handles encrypted columns. Options are:

- true – decrypt columns before transmission. This is the default value. The user must have permission to decrypt any encrypted columns.
- false – transmit columns encrypted exactly as they appear in the data row.

<nnn> Specifies an optional integer identifier that helps track a given transfer. Use the `spt_TableTransfer.tracking_id` column to determine the value for **<nnn>** and use the value in queries. This example returns the ending status and sequence ID of tracking ID number 123, along with the complete path to the output data file (returns NULL if these values do not exist):

```
SELECT end_code, sequence_id, pathname FROM spt_TableTransfer
WHERE id = object_id('mytable') AND tracking_id = 123
```

The SAP ASE server does not control the `tracking_id` or require that it be unique.

<id> Identifies a history entry from a table whose sequence ID column obtains the starting timestamp for this data transfer. This option resends previously sent data. `resend = <id>` is ignored unless the table named in the command is marked for incremental transfer. If the specified sequence ID does not exist for this table, the SAP ASE server resends the entire table.

The SAP ASE server selects the indicated entry's starting timestamp as the starting timestamp for this transfer, and the indicated entry's destination type (`ase`, `bcp`, and so on) as the transfer's default destination type.

Negative values for `id` retrieve history entries for previously completed successful transfers of the specified table. `-1` names the most recently complete successful transfer, `-2` the next most recent, and so forth. The transfer history table stores entries for both successful and failed transfers.

8.17 Manage Bulk Copying Data

You can use bulk copy to copy data into or out of a table.

Bulk copying data in or out of a table provides a convenient, high-speed method for transferring data between a database table or view and an operating system file. When copying in from a file, bulk copy inserts data into an existing database table; when copying out to a file, bulk copy overwrites any previous contents of the file.

8.17.1 Bulk Copying Data Into or Out of Tables

Use bulk copy to insert data into an existing database table or to copy table data to an external file.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [bcp](#).

Syntax

```
.../ASE160/OCS-16_0/bin/bcp mytable out  
.../ASE160/bcp.out -c -t , -r \n -U sa -P<password> -SSAPASE
```

8.18 Manage Views

Views are an alternative way of looking at the data in one or more tables.

A view is the results of a predefined SQL query. Use views to structure table data, restrict access to data, or combine data from various tables, which can be used to generate reports.

8.18.1 Creating a View

Create an SAP ASE view.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create view](#).

Syntax

```
CREATE VIEW <view_name> AS SELECT <select_statement>
```

8.18.2 Replacing a View Definition

You can replace the SQL definition of an existing view.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create view](#).

Syntax

```
CREATE OR REPLACE VIEW <view_name> AS SELECT <select_statement>
```

8.18.3 Managing View Properties

Display or modify column datatype and permissions, and on database objects that reference, and are referenced by, the view.

8.18.3.1 Viewing General View Properties

View the name, type, database, owner, AND creation date of the view.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    viewName = O.name,
    creator = U.name,
    createDate = O.crdate
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND O.type = 'V'
    AND U.name like '%'
    AND lower(O.name) like '%'
    AND O.id = <view_ID>
ORDER BY
    1
```

Output

viewName	creator	createDate
myview	dbo	2020-03-04 22:56:28.856

8.18.3.2 Viewing the SQL for a View

View the SQL statements for creating the view.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    text
FROM
    <database_name>.dbo.syscomments
WHERE
    id = <view_ID>
ORDER BY
    colid2,
    colid;
```

Output

text

create view myview as

select ProductID, ProductName

from Products

where Discontinued = 'No'

(1 row affected)

8.18.3.3 Viewing the Column Properties for a View

View the name AND type of all columns in the view.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    column_name=C.name,
    column_type=isnull(get_xtypename(C xtype, C.xdbid), T.name)
FROM
    <database_name>..syscolumns C,
```

```

<database_name>..systypes T
WHERE
  C.id = <view_ID>
  AND C.usertype *= T.usertype
ORDER BY
  colid;

```

Output

column_name	column_type
ProductID	int
ProductName	varchar

8.18.3.4 Viewing the Data in a View

Show the data for each row in the view.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT * FROM <owner>.<view_name>

```

Output

cache_name	io_size	config_size	run_size	apf_percent	wash_size	cacheid	instanceid	scope
myCache	4	3.0	3.0	10	612	2	0	Global
default data cache	4	8.0	8.0	10	1636	0	0	Global
IMCache	4	20.0	20.0	0	0	1	0	Global
myCache	8	1.0	1.0	10	200	2	0	Global

8.18.3.5 Viewing the Objects Referenced By a View

Show the name, type, and owner of objects referenced by this view. .

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    procedureName = O.name,
    owner = U.name,
    type = O.type,
    number = D.number
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND O.type = 'P'
    AND O.id = D.id
    AND D.depid = <view_ID>
```

Output

procedureName	owner	type	number
sp_configure	dbo	P	1
sp_dboption	dbo	P	1

8.18.3.6 Viewing Objects that Reference a View

Show the name, type, and owner of objects that this view references.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    viewName = O.name,
    owner = U.name,
    type = O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND O.type = 'V'
    AND O.id = D.id
    AND D.depid = <view_ID>
```

Output

viewName	owner	type
syscacheinfo	dbo	V
syscachepoolinfo	dbo	V

8.18.3.7 Viewing the Tables that Reference a View

Show the name, type, and owner of objects that this view references.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    tableName = O.name,
    owner = U.name,
    type = O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND (O.type = 'S'
        OR O.type = 'U')
    AND O.id = D.depid
    AND D.id = <view_ID>
```

Output

tableName	owner	type
sysconfigures	dbo	S
syscurconfigs	dbo	S

8.18.4 Managing View Permissions

Display or modify column datatype and permissions, and on database objects that reference, and are referenced by, the view.

8.18.4.1 Viewing the Permissions of a View

Show the permissions granted on a view.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helpprotect](#).

Syntax

```
EXEC sp_helpprotect '<owner>.<view_name>'
```

Output

grantor	grantee	type	action	object	column	predicate	grantable
dbo	probe	Grant	Decrypt	dbo.myview	All	(Null)	FALSE
dbo	probe	Grant	Delete	dbo.myview	All	(Null)	FALSE
dbo	probe	Grant	Insert	dbo.myview	All	(Null)	FALSE
dbo	probe	Grant	Select	dbo.myview	All	(Null)	FALSE
dbo	probe	Grant	Update	dbo.myview	All	(Null)	FALSE

8.18.4.2 Granting and Revoking View Permissions

Grant and revoke permissions on views for users, groups, and roles.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT <permission_list> ON <view_name> TO { PUBLIC | <name_list> | <role_list>}
<permission_list> ::= { <permissions_1> | <permissions_2> }
<permissions_1> ::= SELECT, INSERT, DELETE, UPDATE with an object name to grant
permissions on their own objects.
<permissions_2> ::= IDENTITY_INSERT, IDENTITY_UPDATE when granular permissions
are enabled.
```

To revoke:

```
REVOKE <permission_list> ON <view_name> FROM { PUBLIC | <name_list> |
<role_list>}
```

8.18.5 Deleting a View

Delete views.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop view](#).

Syntax

```
DROP VIEW <view_name>
```

8.19 Manage Triggers

A trigger is a special type of procedure attached to a table column that goes into effect when a user changes the table. Triggers execute immediately after data modification statements are completed.

8.19.1 Creating a Trigger

Create a trigger on a table to enable checks whenever data is inserted, updated, or deleted.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create trigger](#).

Syntax

```
CREATE TRIGGER [<owner>.<trigger_name>
ON [<owner>.<table_name>
FOR {INSERT, UPDATE} | INSTEAD OF {INSERT, UPDATE, DELETE}
[ORDER <integer>]
[AS [IF UPDATE (<column_name>)
    [{AND | OR} UPDATE (<column_name>)...] <SQL_statements>
    [IF UPDATE (<column_name>)
    [{AND | OR} UPDATE (<column_name>)]...] <SQL_statements>]...]
```

8.19.2 Viewing Trigger Properties

Show the select trigger options.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    trigger_name = ta.name,
    creator_name = u.name,
    creation_date = ta.crdate,
    trigger_id = ta.id,
    delete_id = tr.deltrig,
    insert_id = tr.instrig,
    update_id = tr.updtrig
FROM
    <database_name>.dbo.sysobjects ta,
    <database_name>.dbo.sysobjects tr,
    <database_name>.dbo.sysusers u
WHERE
    (ta.id = tr.deltrig
    OR ta.id = tr.instrig
    OR ta.id = tr.updtrig)
    AND ta.type = 'TR'
    AND tr.type = 'U'
    AND ta.uid = u.uid
    AND tr.id = <table_id>
    AND u.name like '%'
    AND ta.name like '%'
ORDER BY
    1
```

Output

trigger_name	creator_name	creation_date	trigger_id	delete_id	insert_id	update_id
mytrigger	dbo	2020-03-16 03:26:21.21	624002223	624002223	624002223	624002223

8.19.3 Viewing Objects Referenced by a Trigger

View information on the name, type, and owner of the objects that are referenced by the specified trigger.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    objectName = O.name,
    owner = U.name,
    type = O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND (O.type = 'S'
        OR O.type = 'U')
    AND O.id = D.depid
    AND D.id = <trigger_ID>
```

Output

objectName	owner	type
Products	dbo	U

8.19.4 Modifying a Trigger Definition

Replace the SQL definition of a trigger.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create trigger](#).

Syntax

```
CREATE OR REPLACE TRIGGER[<owner>.]<trigger_name>
ON [<owner>.]<table_name>
{FOR {INSERT, UPDATE} | INSTEAD OF {INSERT, UPDATE, DELETE}}
[ORDER <integer>]
[AS [IF UPDATE (<column_name>)
    [{AND | OR} UPDATE (<column_name>)]...] <SQL_statements>
    [IF UPDATE (<column_name>)
    [{AND | OR} UPDATE (<column_name>)]...] <SQL_statements>]...
```

8.20 Manage Caches

Create, delete, and generate data definition language for caches.

8.20.1 Creating a Cache

Create new data caches.

The following syntax provides basic guidance for creating a cache. For the full syntax and required permissions, see [sp_cacheconfig](#).

Syntax

```
EXEC sp_cacheconfig <cachename>, "<cache_size>", <cache_type>
```

8.20.2 Viewing Cache Properties

Display default data cache sizes, buffer pool values, and cache bindings. Also view in-memory database and in-memory device information.

8.20.2.1 Viewing the In-memory and No In-memory Storage Cache Property

Show the storage cache for the database.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    cacheName = a.name,
    status = a.status,
    configSize = a.value,
    partition = a.value4,
    runningSize = (SELECT b.value
                   FROM master.dbo.syscurconfigs b
                   WHERE b.config = a.config AND b.comment = a.name)
FROM
    master.dbo.sysconfigures a
WHERE
    parent = 19
    AND config = parent
    AND a.name like '<cache_name>'
ORDER BY
    1
```

Result Set:

For no in-memory storage cache, the unused size is the running size.

For in-memory storage cache, the unused size is the running size minus the total size of the in-memory database device created on the cache.

cacheName	status	configSize	partition	runningSize
myCache	34	2048	1	2048

8.20.2.2 Viewing the Available Space Property

Show the amount of additional free memory.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    additional_free_memory = (MAX(b.value) - MIN(b.value)) * 2
FROM
    master.dbo.sysconfigures a,
    master.dbo.syscurconfigs b
WHERE
    a.name in ('max memory', 'total logical memory')
    AND a.config = b.config
```

Result Set:

additional_free_memory

21782

8.20.2.3 Viewing the Buffer Pool Property

Show the current buffer pool values for regular caches.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    key = substring( a.comment, 1, charindex('K', a.comment) -1),
    memorySize = b.value,
    washSize = b.memory_used,
    localLimit = a.value4
FROM
    master.dbo.sysconfigures a,
    master.dbo.syscurconfigs b
WHERE
    a.parent = 19
    AND a.config != a.parent
    AND b.config = a.config
    AND a.name = '<cache_name>'
    AND b.comment = a.name
```

Result Set:

key	memorySize	washSize	localLimit
4	1536	612	-1
8	512	200	10

8.20.2.4 Viewing the Cache Binding Property for the Database

Show object bindings for the databases for regular caches.

The following syntax provides basic guidance for viewing this information.

The following syntax provides basic guidance for using a SELECT statement to view the object bindings for databases for regular caches.

Syntax

```
SELECT
  dbName = db_name( object ),
  objectType = object_type,
  owner = (SELECT b.name
           FROM master.dbo.syslogins b,
                master.dbo.sysdatabases c
           WHERE b.suid = c.suid
                AND c.dbid = a.object)
FROM
  master.dbo.sysattributes a
WHERE
  char_value = '<cache_name>'
```

Result Set:

dbName	objectType	owner
pubs2	D	sa

8.20.2.5 Viewing the Cache Binding Property for Tables

Show object bindings for tables for regular caches.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  db_name = '<database_name>',
  id = a.object,
  name = (SELECT b.name
          FROM
            <database_name>.dbo.sysobjects b
          WHERE
            b.id = a.object ),
  objectType = object_type,
  owner = (SELECT b.name
           FROM
            <database_name>.dbo.sysusers b,
            <database_name>.dbo.sysobjects c
           WHERE
            b.uid = c.uid
```

```

        AND c.id = a.object),
    tableType = (SELECT convert(char(1), b.type)
    FROM
        <database_name>.dbo.sysobjects b
    WHERE
        b.id = a.object )
FROM
    <database_name>.dbo.sysattributes a
WHERE
    char_value = '<cache_name>'
    AND object_type = 'T'
UNION
ALL SELECT
    '<database_name>',
    a.object, (SELECT b.name
    FROM <database_name>.dbo.sysobjects b
    WHERE b.id = a.object ), (SELECT b.name
    FROM <database_name>.dbo.sysindexes b
    WHERE b.indid = a.object_infol
    AND b.id = a.object ),
    object_type, (SELECT b.name
    FROM <database_name>.dbo.sysusers b,
    <database_name>.dbo.sysobjects c
    WHERE b.uid = c.uid
    AND c.id = a.object),
    (SELECT convert(char(1), b.type)
    FROM <database_name>.dbo.sysobjects b
    WHERE b.id = a.object )
FROM
    <database_name>.dbo.sysattributes a
WHERE
    char_value = '<cache_name>'
    AND object_type = 'I'

```

Output

db_name	id	name	objectType	owner	tableType
master	1577053623	Products	T	dbo	U

8.20.2.6 Viewing the In-memory Database Property

(Available only for in-memory databases) Displays the in-memory database that is created on this cache.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
    DBName = d.name,
    Owner = l.name,
    Size = SUM(CONVERT(int, ROUND(v.low * (CONVERT(float, u.size)/1048576), 0)))
FROM

```

```

master.dbo.sysdatabases d,
master.dbo.sysusages u,
master.dbo.spt_values v,
master.dbo.sysdevices e,
master.dbo.syslogins l
WHERE
  d.suid = l.suid
  AND (d.status3 & 4194304) != 4194304
  AND d.status4 & 4096 = 4096
  AND d.dbid = u.dbid
  AND e.vdevno = u.vdevno
  AND e.status2&8=8
  AND v.type = 'E'
  AND v.number = 1
  AND e.phyname like '<cache_name>'
GROUP BY
  d.name,
  l.name

```

Result Set:

DBName	Owner	Size
IMDB	sa	6

8.20.2.7 Viewing the In-memory Devices Property

Show general information on in-memory devices.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  a.name,
  a.status,
  a.value,
  a.value4,
  (SELECT
    b.value
  FROM
    master.dbo.syscurconfigs b
  WHERE
    b.config = a.config
    AND b.comment = a.name),
  (SELECT
    SUM((1. + (d.high - d.low)) / (1048576. / @@pagesize))
  FROM
    master.dbo.sysdevices d
  WHERE
    d.status2&8=8
    AND d.phyname = a.name)
GROUP BY

```

```

        d.phyname)
FROM
    master.dbo.sysconfigures a
WHERE
    parent = 19
    AND config = parent
    AND a.name like '<cache_name>'
ORDER BY
    1

```

Result Set:

Name	Size
IMDEV	6

8.20.3 Managing Binding Options

Bind and unbind a database, table, index, text object, or image object to or from a data cache.

Bind Options

The following syntax provides basic guidance for binding to a data cache. For the full syntax and required permissions, see [sp_bindcache](#).

Syntax

```
EXEC sp_bindcache '<cache_ame>',<dbname>
```

UnBind Options

The following syntax provides basic guidance for unbinding from a data cache. For the full syntax and required permissions, see [sp_unbindcache](#).

Syntax

```
EXEC sp_unbindcache <dbname>
```

8.20.4 Managing Buffer Pools

Add and change buffer pools for your data cache.

The following syntax provides basic guidance for managing buffer pools. For the full syntax and required permissions, see [sp_poolconfig](#).

Syntax

To create a buffer pool:

```
EXEC sp_poolconfig '<cache_name>', "<mem_size> [P | K | M | G]", "<config_pool>K"
```

To add or change a pool's asynchronous prefetch percentage:

```
EXEC sp_poolconfig '<cache_name>', "<affected_pool>K", "local async prefetch limit=<percent>"
```

To add or change a pool's wash size:

```
EXEC sp_poolconfig '<cache_name>', "<affected_pool>K", "wash=<size>[P|K|M|G]"
```

To remove the memory pool from the cache:

```
EXEC sp_poolconfig '<cache_name>', '0K', '8K'
```

8.20.5 Modifying Data Cache Sizes

Modify the size of the data cache and specify the number of partitions in the data cache.

The following syntax provides basic guidance for setting data cache sizes. For the full syntax and required permissions, see [sp_cacheconfig](#).

Syntax

To set cache type:

```
EXEC sp_cacheconfig '<cache_name>' [, logonly | mixed | inmemory_storage | lockless data cache | row_storage]
```

To set cache partition size:

```
EXEC sp_cacheconfig '<cache_name>', cache_partition=[1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256]
```

To set cache configuration size:

```
exec sp_cacheconfig '<cache_name>', "<cache_size> [P | K | M | G]"
```

8.20.6 Generating DDL for a Cache

Generate a DDL script to create a cache.

The following syntax provides basic guidance for generating a DDL script. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server> -TC -N<cache_name>
```

Output

DDL for Cache 'myCache'

```
print '<<<<< CREATING Cache - "myCache" >>>>>'
go
use master
go
IF EXISTS (SELECT 1 FROM master.dbo.sysconfigures WHERE name = 'myCache' AND parent = 19)
exec sp_cacheconfig 'myCache', '0'
IF (@@error != 0)
BEGIN
PRINT 'Error CREATING Cache myCache'
SELECT syb_quit()
END
go
exec sp_cacheconfig 'myCache', '2048K', mixed
go
exec sp_poolconfig 'myCache', '512K', '8K'
```

go

8.20.7 Deleting a Cache

Delete a cache in SAP ASE.

The following syntax provides basic guidance for deleting a cache. For the full syntax and required permissions, see [sp_cacheconfig](#).

Syntax

```
EXEC sp_cacheconfig '<cache_name>', '0'
```

8.20.8 Setting the Statement Cache Size

Set the statement cache size.

The following syntax provides basic guidance for setting the statement cache size. For the full syntax and required permissions, see [sp_configure](#).

Syntax

```
EXEC sp_configure <configname>, <configvalue>
```

8.21 Manage Constraints

Use constraints to specify rules for table data. Constraint prevent violations between the specified rule and the data action.

8.21.1 Creating a Foreign Key

Create a foreign key to constrain a column based on values in a reference table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <database_name>.<owner>.<table_name>
  ADD CONSTRAINT <constraint_name> FOREIGN KEY (<column_name>[, <column_name>
  [,...]])
  REFERENCES [[<database_name>>.] [<owner>].]<ref_table>
  [( <ref_column>[,<ref_column> [,...]])]
```

8.21.2 Foreign Key Properties

The Foreign Key Properties window shows current foreign key options and the matching primary keys.

8.21.2.1 Viewing General Foreign Key Properties

Show the selected foreign key options.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  fk_name = O.name,
  creator_name = U.name,
  ref_db_name = isnull(R.pmydbname, '<database_name>'),
  ref_table_id = R.reftabid,
  ref_table_name = object_name(R.reftabid, M.dbid),
```

```

creation_date = O.crdate,
object_id = O.id
FROM
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysreferences R,
  <database_name>.dbo.sysusers U,
  <database_name>.dbo.sysdatabases M
WHERE
  O.type = 'RI'
  AND O.uid = U.uid
  AND O.id = R.constrid
  AND R.tableid = <table_ID>
  AND M.name = isnull(R.pmrydbname, '<database_name>')

```

Output

fk_name	creator_name	ref_db_name	ref_table_id	ref_table_name	creation_date	object_id
js_jobs_FK	dbo	sybmgrmtdb	624002223	js_jobs	2020-03-16 19:08:37.5	149572540

8.21.2.2 Viewing Column Foreign Key Properties

Show the defined foreign keys and the primary keys to which the foreign keys applies.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  fk_col1 = col_name(tableid, fokey1, frgnbid),
  fk_col2 = col_name(tableid, fokey2, frgnbid),
  fk_col3 = col_name(tableid, fokey3, frgnbid),
  fk_col4 = col_name(tableid, fokey4, frgnbid),
  fk_col5 = col_name(tableid, fokey5, frgnbid),
  fk_col6 = col_name(tableid, fokey6, frgnbid),
  fk_col7 = col_name(tableid, fokey7, frgnbid),
  fk_col8 = col_name(tableid, fokey8, frgnbid),
  fk_col9 = col_name(tableid, fokey9, frgnbid),
  fk_col10 = col_name(tableid, fokey10, frgnbid),
  fk_col11 = col_name(tableid, fokey11, frgnbid),
  fk_col12 = col_name(tableid, fokey12, frgnbid),
  fk_col13 = col_name(tableid, fokey13, frgnbid),
  fk_col14 = col_name(tableid, fokey14, frgnbid),
  fk_col15 = col_name(tableid, fokey15, frgnbid),
  fk_col16 = col_name(tableid, fokey16, frgnbid),
  fk_col17 = col_name(reftabid, refkey1, pmrybid),
  fk_col18 = col_name(reftabid, refkey2, pmrybid),
  fk_col19 = col_name(reftabid, refkey3, pmrybid),
  fk_col20 = col_name(reftabid, refkey4, pmrybid),
  fk_col21 = col_name(reftabid, refkey5, pmrybid),
  fk_col22 = col_name(reftabid, refkey6, pmrybid),

```

```

fk_col23 = col_name(reftabid, refkey7, pmrydbid),
fk_col24 = col_name(reftabid, refkey8, pmrydbid),
fk_col25 = col_name(reftabid, refkey9, pmrydbid),
fk_col26 = col_name(reftabid, refkey10, pmrydbid),
fk_col27 = col_name(reftabid, refkey11, pmrydbid),
fk_col28 = col_name(reftabid, refkey12, pmrydbid),
fk_col29 = col_name(reftabid, refkey13, pmrydbid),
fk_col30 = col_name(reftabid, refkey14, pmrydbid),
fk_col31 = col_name(reftabid, refkey15, pmrydbid),
fk_col32 = col_name(reftabid, refkey16, pmrydbid),
pmrydbname = pmrydbname,
object_name = object_name(reftabid),
FK_Name = object_name(constrid, frgnbid)
FROM
<database_name>.dbo.sysreferences
WHERE
tableid=<table_ID>

```

Output

fk_col1	fk_col2	fk_col3	fk_col4	fk_col5	fk_col6	fk_col7
title_id	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)
fk_col8	fk_col9	fk_col10	fk_col11	fk_col12	fk_col13	fk_col14
(Null)	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)
fk_col15	fk_col16	fk_col17	fk_col18	fk_col19	fk_col20	fk_col21
(Null)	(Null)	title_id	(Null)	(Null)	(Null)	(Null)
fk_col22	fk_col23	fk_col24	fk_col25	fk_col26	fk_col27	fk_col28
(Null)	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)
fk_col29	fk_col30	fk_col31	fk_col32	pmrydbname	object_name	FK_Name
(Null)	(Null)	(Null)	(Null)	(Null)	myprs	myprs_FK

8.21.3 Creating a Check Constraint

Creating a check constraint specifies a condition that any value must pass before it is inserted into the table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <database_name>.<owner>.<table_name>
```

```
ADD CONSTRAINT <constraint_name> CHECK (<search_condition>)
```

8.21.4 Viewing General Check Constraint Properties

Show the name, owner, AND creation date of the check constraint properties AND show the check constraint expression or condition that values must pass before being inserted into the table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    CHECKCONSTRAINT_NAME=O.name,
    CHECKCONSTRAINT_OWNER=U.name,
    CHECKCONSTRAINT_ID=O.id,
    CHECKCONSTRAINT_CREATION_DATE=O.crdate
FROM
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysprocedures P,
    <database_name>.dbo.sysusers U,
    <database_name>.dbo.sysconstraints CON
WHERE
    O.uid = U.uid
    AND O.id = P.id
    AND O.id = CON.constrid
    AND CON.tableid = <table_ID>
    AND CON.colid=0
    AND P.sequence = 0
    AND P.status & 4096 = 4096
    AND O.type = 'R'
```

Output

CHECKCONSTRAINT_NAME	CHECKCONSTRAINT_OWNER	CHECKCONSTRAINT_ID	CHECKCONSTRAINT_CREATION_DATE
testcheck	dbo	165572597	2020-03-16 19:31:42.603

8.21.5 Creating a Unique Constraint or Primary Key

Creating a unique constraint or primary key constraint to ensure that no two rows in a table have the same values in the specified columns.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <database_name>.<owner>.<table_name>  
  ADD CONSTRAINT <constraint_name> UNIQUE NONCLUSTERED  
  (<column_name> [ASC | DESC][, <column_name> [ASC | DESC] [,...]])
```

8.21.6 Binding Defaults and Rules to a Column

Specify constraints on column data by binding defaults or rules to a column.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_bindrule](#) and [sp_binddefault](#).

Syntax

To set a rule:

```
EXEC sp_bindrule '<owner>.<rulename>', '<objname>'
```

To set a default:

```
EXEC sp_bindefault '<owner>.<defname>', '<objname>'
```

8.22 Manage Rules

Create or delete rules, or replace rule definitions.

8.22.1 Creating a Rule

Create a rule that specifies the domain of acceptable values for a particular column.

Only a database owner, or a user or group with create rule permission can create a rule.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create rule](#).

Syntax

```
CREATE RULE [<owner>.<rule_name> AS <condition_expression>
```

8.22.2 Replacing a Rule Definition

You can replace the rule expression of an existing rule.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create rule](#).

Syntax

```
CREATE OR REPLACE RULE [<owner>.<rule_name> AS <condition_expression>
```

8.22.3 Manage Rule properties

View and modify rule properties.

8.22.3.1 Viewing General Rule Properties

View the name, type, database, owner, creation date, AND rule expression.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = A.name,
    id = A.id,
    owner = B.name,
    createDate = convert(datetime, A.crdate)
FROM
    <database_name>.dbo.sysobjects A,
    <database_name>.dbo.sysusers B,
    <database_name>.dbo.sysprocedures P
WHERE
    A.uid = B.uid
    AND A.id = P.id
```

```

AND P.sequence = 0
AND P.status & 4096 != 4096
AND A.type = 'R'
AND B.name like '%'
AND A.name like '<rule_name>'
ORDER BY
1

```

Output

name	id	owner	createDate
myrule	576002052	dbo	2020-03-16 02:04:57.896

8.22.3.2 Viewing the SQL for Rules

View the SQL statements for creating the rule.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
    text
FROM
    <database_name>.dbo.syscomments
WHERE
    id = <rule_ID>
ORDER BY
    colid2,
    colid

```

Output

text

```
create rule dbo.myrule as @advance < $1000
```

8.22.3.3 Viewing Objects Referenced By Rules

View the name, type, owner, AND properties of objects that referenced by this rule.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = distinct O.name,
    owner = U.name ,
    type = O.type
FROM
    <database_name>.dbo.syscolumns C,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    (O.type = 'U'
    OR O.type = 'S')
    AND O.uid = U.uid
    AND C.id = O.id
    AND C.domain = <rule_ID>
```

Output

name	owner	type
Products	dbo	U

8.22.4 Deleting a Rule

Delete rules.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop rule](#).

Syntax

```
DROP RULE [<owner>.]<rule_name>
```

8.22.5 Generating DDL for a Rule

Generate a DDL script for rules.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server> -TR -N<object_name>
```

Output

DDL for Rule 'IMDB.dbo.myrule'

```
print '<<<<< CREATING Rule - "IMDB.dbo.myrule" >>>>>'  
  
go  
  
use IMDB  
  
go  
  
set quoted_identifier on  
  
go  
  
IF EXISTS (SELECT 1 FROM sysobjects o, sysusers u WHERE o.uid=u.uid AND o.name = 'myrule' AND u.name = 'dbo' AND  
o.type = 'R')  
  
BEGIN  
  
setuser 'dbo'  
  
drop rule "myrule"  
  
END  
  
go  
  
set quoted_identifier off  
  
go  
  
IF (@@error != 0)  
  
BEGIN  
  
PRINT 'Error CREATING Rule IMDB.dbo.myrule'  
  
SELECT syb_quit()  
  
END
```

DDL for Rule 'IMDB.dbo.myrule'

```
go
setuser 'dbo'
go
set quoted_identifier on
go
create rule dbo.myrule as @advance < $1000
go
set quoted_identifier off
go
setuser
go
```

8.23 Manage Precomputed Result Sets

Precomputed result sets (PRS) are views for which the result is computed, stored, and available for future use. SAP ASE precomputes queries and attempts to use the precomputed result during subsequent iterations.

8.23.1 Creating a Precomputed Result Set

Create precomputed result sets.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create precomputed result set](#).

Syntax

```
CREATE RECOMPUTED RESULT SET [<owner_name>].<prs_name>
    UNIQUE (<column_name>) AS <query_expression>
```

<prs_name> Specifies the name of the precomputed result set. A fully qualified **<prs_name>** cannot include the server or database name.

<column_name> Constrains the values in the indicated column or columns so that no two rows have the same value.

8.23.2 Deleting a Precomputed Result Set

Deleting a precomputed result set deletes its data, removes any system table entries, and deletes the precomputed result set.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop precomputed result set](#).

Syntax

```
DROP PRECOMPUTED RESULT SET <owner_name>.<prs_name>
```

8.23.3 Altering a Precomputed Result Set

Altering a precomputed result set changes its policies or properties.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter precomputed result set](#).

Syntax

```
ALTER PRECOMPUTED RESULT SET <owner_name>.<prs_name> <alter_clauses>
```

8.23.4 Refreshing Precomputed Result Sets

Refresh precomputed result sets to prevent their data from getting stale.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [refresh precomputed result set](#).

Syntax

```
REFRESH PRECOMPUTED RESULT SET <owner_name>.<prs_name>
```

8.23.5 Truncating a Precomputed Result Set

Truncating a precomputed result set removes the data but retains the definition of the precomputed result set in the system table.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [truncate precomputed result set](#).

Syntax

```
TRUNCATE PRECOMPUTED RESULT SET <owner_name>.<prs_name>
```

8.23.6 Managing Permissions on Precomputed Results Sets

View and modify permissions on precomputed results sets.

8.23.6.1 Viewing Permissions on a Precomputed Result Set

Show permissions on a precomputed result set.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helprotect](#).

Syntax

```
USE <database_name>  
EXEC sp_helprotect '<owner>.<prs_name>'
```

Output

grantor	grantee	type	action	object	column	predicate	grantable
dbo	guest	Grant	Select	dbo.mypr s	All	(Null)	FALSE
dbo	jstask	Grant	Select	dbo.mypr s	All	(Null)	FALSE

grantor	grantee	type	action	object	column	predicate	grantable
dbo	probe	Grant	Select	dbo.mypr s	All	(Null)	FALSE

8.23.6.2 Granting and Revoking Permissions on Precomputed Results Sets

Grant or revoke SELECT permission on precomputed result sets for users, groups, and roles.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT SELECT ON <owner_name>.<prs_name> TO { PUBLIC | <name_list> |
<role_list> } [ WITH GRANT OPTION ]
```

To revoke:

```
REVOKE SELECT ON <owner_name>.<prs_name> FROM { PUBLIC | <name_list> |
<role_list> }
```

8.23.7 Precomputed Result Sets Properties

Display information (such as column IDs, user names, partition names, and so on) associated with precomputed result sets.

8.23.7.1 Viewing Precomputed Result Set SQL

Show the SQL used to create the precomputed result set.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
```

```

text
FROM
  <database_name>.dbo.syscomments
WHERE
  id = <prs_id>
ORDER BY
  colid2,
  colid;

```

Output

text

(Null)

CREATE PRECOMPUTED RESULT SET dbo.myprs

UNIQUE(ProductID)

IMMEDIATE REFRESH

POPULATE

ENABLE

DISABLE USE IN OPTIMIZATION

LOCK allpages

AS

select ProductID,ProductName,Discontinued from Products

8.23.7.2 Viewing Columns in the Precomputed Result Set

Show column names and character sets.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  column_name = C.name,
  column_type = isnull(get_xtypename(C.xtype, C.xdbid), T.name)
FROM
  <database_name>..syscolumns C,
  <database_name>..systypes T
WHERE
  C.id = <prs_id>
  and C.usertype *= T.usertype
ORDER BY

```

```
colid;
```

Output

column_name	column_type
ProductID	int
ProductName	varchar
Discontinued	varchar

8.23.7.3 Viewing Data in the Precomputed Result Set

Show the data contained in the precomputed result set.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT * FROM <database_name>.<owner>.<prs_name>
```

Output

ProductID	ProductName
1	cookies

8.23.7.4 Viewing Objects the Precomputed Result Set References

Show information of all of the objects the precomputed result set references.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    O.name,
    U.name,
    O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND (O.type = 'S'
        OR O.type = 'U')
    AND O.id = D.depid
    AND D.id = <prs_id>
```

Output

name	name	type
Products	dbo	U

8.23.7.5 Viewing Partitions in the Precomputed Result Set

Show the name of the partitions, the segment on which they reside, AND the date they were created.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    partition_name = case
        WHEN p.indid = 1 THEN p.cdaptname
        ELSE p.name
    END,
    segmentName = s.name,
    createDate = p.crdate
FROM
    <database_name>.dbo.syspartitions p,
    <database_name>.dbo.sysindexes i,
    <database_name>.dbo.syssegments s
WHERE
    i.id = p.id
    AND i.id = <prs_id>
    AND p.indid < 2
    AND i.indid = p.indid
    AND s.segment = p.segment
    AND p.name like '<prs_name>'
ORDER BY
```

Output

partitionName	segmentName	createDate
DBtest0_1369052882	default	2020-03-01 18:51:55.853

8.23.7.6 Viewing Indexes in the Precomputed Result Set

Show information on indexes.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  NAME = I.name,
  OWNER = U.name,
  INDEX_ID = I.indid,
  COLUMN_COUNT = I.keycnt,
  STATUS = I.status,
  STATUS2 = I.status2,
  MAX_ROW_PER_PAGE = I.maxrowsperpage,
  FILL_FACTOR = I.fill_factor,
  RES_PAGE_GAP = I.res_page_gap,
  SEGMENT_NAME = S.name,
  TABLE_NAME = O.name
FROM
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysindexes I,
  <database_name>.dbo.sysusers U,
  <database_name>.dbo.syssegments S
WHERE
  I.id = O.id
  and U.uid = O.uid
  and I.segment = S.segment
  and I.indid between 1
  and 254
  and (I.status2 & 2) = 2
  and I.name like '%'

```

Output

NAME	OWNER	INDEX_ID	COLUMN_COUNT	STATUS	STATUS2
myprs_212105556 12	dbo	2	2	2	10

MAX_ROW_PER_PAGE	FILL_FACTOR	RES_PAGE_GAP	SEGMENT_NAME	TABLE_NAME
0	0	0	default	myprs

8.23.7.7 Viewing Index Unique Constraints in the Precomputed Result Set

Show information on unique constraints on indexes.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
  NAME = I.name,
  OWNER = U.name,
  INDEX_ID = I.indid,
  COLUMN_COUNT = I.keycnt,
  STATUS = I.status,
  STATUS2 = I.status2,
  MAX_ROW_PER_PAGE = I.maxrowsperpage,
  FILL_FACTOR = I.fill_factor,
  RES_PAGE_GAP = I.res_page_gap,
  SEGMENT_NAME = S.name,
  TABLE_NAME = O.name
FROM
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysindexes I,
  <database_name>.dbo.sysusers U,
  <database_name>.dbo.syssegments S
WHERE
  I.id = O.id
  AND U.uid = O.uid
  AND O.id = < prs_id >
  AND I.segment = S.segment
  AND I.indid between 1
  AND 254
  AND (I.status2 & 2) = 2
  AND I.name like '%'
```

Output

NAME	OWNER	INDEX_ID	COLUMN_COUNT	STATUS	STATUS2
myprs_181572654 2	dbo	2	2	2	10

MAX_ROW_PER_PAGE	FILL_FACTOR	RES_PAGE_GAP	SEGMENT_NAME	TABLE_NAME
0	0	0	default	myprs

8.23.8 Generating DDL for a Precomputed Result Set

Generate object definitions for existing precomputed result sets.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen - U<login> - P<password> - S<server> - TPRS - N<prs_name>
```

Output

DDL for Pre Computed Result Sets 'master.dbo.myprs'

```
print '<<<<< CREATING Pre Computed Result Sets - "master.dbo.myprs" >>>>>'
go
use master
go
set quoted_identifier on
go
IF EXISTS (SELECT 1 FROM sysobjects o, sysusers u WHERE o.uid=u.uid AND o.name = 'myprs' AND u.name = 'dbo' AND
o.type = 'RS')
BEGIN
setuser 'dbo'
drop PRS "myprs"
```

DDL for Pre Computed Result Sets 'master.dbo.myprs'

```
END
go
set quoted_identifier off
go
IF (@@error != 0)
BEGIN
PRINT 'Error CREATING Pre Computed Result Sets master.dbo.myprs'
SELECT syb_quit()
END
go
setuser 'dbo'
go
set ansinull on
go
set arithabort on
go
set arithignore off
go
set string_rtruncation on
go
set quoted_identifier on
go
CREATE PRECOMPUTED RESULT SET dbo.myprs
UNIQUE(ProductID)
IMMEDIATE REFRESH
POPULATE
ENABLE
DISABLE USE IN OPTIMIZATION
LOCK allpages
AS
select ProductID,ProductName,Discontinued from Products
go
```

DDL for Pre Computed Result Sets 'master.dbo.myprs'

Grant Select on dbo.myprs to guest Granted by dbo

go

Grant Select on dbo.myprs to probe Granted by dbo

go

Grant Select on dbo.myprs to jstask Granted by dbo

go

set quoted_identifier off

go

setuser

go

8.24 Manage Defaults

Specify a default value that can be referenced by an object if no value is explicitly supplied.

8.24.1 Creating a Default

Specify a value to insert in a column if no value is explicitly supplied at insert time.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create default](#).

Syntax

```
CREATE DEFAULT [<owner>.<default_name> AS <constant_expression>
```

8.24.2 Manage Defaults Properties

Display or modify defaults.

8.24.2.1 Viewing General Defaults Properties

View name, type, database, owner, creation date AND expression for defaults.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = A.name,
    id = A.id,
    owner = B.name,
    createDate = A.crdate
FROM
    <database_name>.dbo.sysobjects A,
    <database_name>.dbo.sysusers B,
    <database_name>.dbo.sysprocedures P
WHERE
    A.uid = B.uid
    AND A.id = P.id
    AND P.sequence = 0
    AND P.status & 4096 != 4096
    AND A.type = 'D'
    AND B.name like '%'
    AND A.name like '<default_name>'
ORDER BY
    1
```

Output

name	id	owner	createDate
mydefault	656002337	dbo	2020-03-16 20:34:07.6

8.24.2.2 Viewing the SQL for Defaults Properties

View the SQL statements for creating the default.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    text
FROM
    <database_name>.dbo.syscomments
WHERE
    id = <default_ID>
ORDER BY
    colid2,
    colid
```

Output

text

create default dbo.mydefault

as

@@spid

8.24.2.3 Viewing Objects Referenced By Defaults Properties

Viewing properties of objects referenced by this default.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = t.name,
    owner = b.name
FROM
    <database_name>.dbo.systypes t,
    <database_name>.dbo.sysusers b,
    <database_name>.dbo.systypes p
WHERE
    t.uid = b.uid
    AND t.usertype >= 100
    AND t.type = p.type
    AND p.usertype = (SELECT min(usertype)
                      FROM <database_name>.dbo.systypes
                      WHERE type = t.type)
    AND t.tdefault = <default_ID>
```

Output

name	owner
mydety	dbo

8.24.3 Replacing a Default Definition

You can replace an existing default expression with a new definition.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create default](#).

Syntax

```
CREATE OR REPLACE DEFAULT [<owner>.<default_name> AS <constant_expression>
```

8.24.4 Deleting a Default

Delete a default.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop default](#).

Syntax

```
DROP DEFAULT [<owner>.<default_name>
```

8.25 Manage Stored Procedures

Manage stored and SQLJ procedures.

8.25.1 Manage Stored Procedures

Create, delete, or modify stored procedures.

Stored procedures are named collections of SQL statements and flow control statements. A stored procedure that performs a select, execute, or data modification command must have the same owner as the object acted upon.

A system administrator, a database owner, or a user or group with create procedure permission can create a stored procedure.

8.25.1.1 Creating a Stored Procedure

A stored procedure is a collection of SQL statements and optional control-of-flow statements stored under a name.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create procedure](#).

Syntax

```
CREATE PROCEDURE <sql_procedure_name>
  ( [ [ {in | out | inout} ] <sql_parameter_name> <sql_datatype>
    [ {( <length> | <precision>[, <scale>)}]]
  [, ( [ [ {in | out | inout} ] <sql_parameter_name> <sql_datatype>
    [ {( <length> | <precision>[, <scale>)}]]
    [= <default>] ] ) [, ...] ]
  LANGUAGE JAVA
  PARAMETER STYLE JAVA
  EXTERNAL NAME '<java_method_name> [ ( <java_datatype>[, <java_datatype>
    [, ...] ] ) ]'
```

<sql_procedure_name>

Specifies the Transact-SQL name of the procedure, which must conform to the rules for identifiers and cannot be a variable. Specify the owner's name to create another procedure of the same name owned by a different user in the current database. The default value for <owner> is the current user.

Use *OR REPLACE* to change the name and number of parameters.

IN | OUT | INOUT

Specifies the mode of the listed parameter. IN indicates an input parameter; OUT indicates an output parameter; and INOUT indicates a parameter that is both an input and an output parameter. The default mode is IN.

Use *OR REPLACE* to change the mode of the listed parameter.

<sql_parameter_name>

Specifies the name of an argument to the procedure. The value of each input parameter is supplied when the procedure is executed. Parameters are optional. A SQLJ stored procedure need not take arguments.

Parameter names must conform to the rules for identifiers. If the value of a parameter contains nonalphanumeric characters, it must be enclosed in quotes. This includes object names qualified by a database name or owner name, since they include a period. If the value of the parameter begins with a numeric character, it also must be enclosed in quotes.

<sql_datatype> [{ (<length> | <precision>[, <scale>]) }]

Specifies the Transact-SQL datatype of the parameter.

<sql_datatype> is the SQL procedure signature.

Use *OR REPLACE* to change the type, length, precision, and scale of the parameters.

<default>

Defines a default value for the procedure's parameter. If a default is defined, you can execute the procedure without a parameter value. The default must be a constant. It can include wildcard characters (% , _ , [, and ^) if the procedure uses the parameter name with the keyword like.

The default can be NULL. The procedure definition can specify some action to be taken if the parameter value is NULL.

Use *OR REPLACE* to change the default to NULL for the parameters, or set a different value when the procedure is changed.

8.25.1.2 Replacing a Stored Procedure Definition

You can replace the SQL definition of a stored procedure and change whether the procedure is recompiled each time it is executed.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create procedure](#).

Syntax

```
CREATE OR REPLACE PROCEDURE <sql_procedure_name>
  ( [ [ {in | out | inout} ] <sql_parameter_name> <sql_datatype>
    [ { ( <length> | <precision>[, <scale> ] ) } ] [ = <default> ]
```

```
[, ( [ [ {in | out | inout} ] <sql_parameter_name> <sql_datatype>
[ { ( <length> | <precision>[, <scale> ] ) } ]
[= <default>] ] ) [, ... ] ]
LANGUAGE JAVA
PARAMETER STYLE JAVA
EXTERNAL NAME '<java_method_name> [ ( <java_datatype>[, <java_datatype>
[, ...] ] ) ]'
```

<sql_procedure_name>

Specifies the Transact-SQL name of the procedure, which must conform to the rules for identifiers and cannot be a variable. Specify the owner's name to create another procedure of the same name owned by a different user in the current database. The default value for <owner> is the current user.

Use *OR REPLACE* to change the name and number of parameters.

IN | OUT | INOUT

Specifies the mode of the listed parameter. IN indicates an input parameter; OUT indicates an output parameter; and INOUT indicates a parameter that is both an input and an output parameter. The default mode is IN.

Use *OR REPLACE* to change the mode of the listed parameter.

<sql_parameter_name>

Specifies the name of an argument to the procedure. The value of each input parameter is supplied when the procedure is executed. Parameters are optional. A SQLJ stored procedure need not take arguments.

Parameter names must conform to the rules for identifiers. If the value of a parameter contains nonalphanumeric characters, it must be enclosed in quotes. This includes object names qualified by a database name or owner name, since they include a period. If the value of the parameter begins with a numeric character, it also must be enclosed in quotes.

<sql_datatype> [{ (<length> | <precision>[, <scale>]) }]

Specifies the Transact-SQL datatype of the parameter.

<sql_datatype> is the SQL procedure signature.

Use *OR REPLACE* to change the type, length, precision, and scale of the parameters.

<default>

Defines a default value for the procedure's parameter. If a default is defined, you can execute the procedure without a parameter value. The default must be a constant. It can include wildcard characters (% , _ , [, and ^) if the procedure uses the parameter name with the keyword like.

The default can be NULL. The procedure definition can specify some action to be taken if the parameter value is NULL.

Use *OR REPLACE* to change the default to NULL for the parameters, or set a different value when the procedure is changed.

8.25.1.3 Managing Stored Procedure Properties

Display or modify stored procedures.

8.25.1.3.1 Viewing General Stored Procedure Properties

View the name, type, database, owner, creation date, AND group number of the procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    proc_name=o.name,
    group_number=p.number,
    owner_name=u.name,
    creation_date=convert(datetime, o.crdate),
    object_id=o.id
FROM
    <database_name>.dbo.sysobjects o,
    <database_name>.dbo.sysprocedures p,
    <database_name>.dbo.sysusers u
WHERE
    o.type='P'
    AND o.sysstat2&33554432 != 33554432
    AND p.sequence=0
    AND o.id=p.id
    AND u.uid=o.uid
    AND u.name like '%'
    AND o.name like '<procedure_name>'
    AND convert(char(12), p.number) like '%'
order by
    1
```

Output

proc_name	group_number	owner_name	creation_date	object_id
sp_createjob	1	dbo	2019-12-18 22:53:56.756	92524332

8.25.1.3.2 Viewing SQL for Stored Procedure Properties

View the SQL statements for creating the procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    text = B.text
FROM
    <database_name>.dbo.sysobjects A,
    <database_name>.dbo.syscomments B
WHERE
    A.id = <procedure_ID>
    AND A.id = B.id
    AND number = <group_number>
ORDER BY
    B.colid2,
    B.colid
```

Output

text

```
-- Internal JobScheduler stored procedure (ASE Plugin)
--
-- Warning this procedure is not a published interface, do not use it.
--
create or replace proc sp_createjob ( @jname univarchar(64),
@jdesc univarchar(128) =
NULL,
@properties varchar(128) = NULL,
@default_timeout int = 60,
@tmpl_id int = NULL ) as
begin
declare @err int
, @rows int
, @msgno int
, @job_id int
, @len int
, @curpos int
, @int_jproperties int
, @seqno int
, @jproperties varchar(128)
```

text

```
, @owner char(30)
, @tjname univarchar(64)
, @tsnam
e univarchar(64)
, @created datetime
, @vcname varchar(64)
, @chunk univarchar(901)
, @sp_name varchar(32)
if @@trancount > 0
begin
/*
** 17260, "Can't run %! from within a transaction."
*/
rai
serror 17260, "sp_createjob"
return -@@error
end
else
begin
set chained off
end
set transaction isolation level 1
select @sp_name='sp_createjob'
-- Check job name is not NULL, empty, a number or job_<number>
select @tjname = ltrim(rtrim(@j
name))
exec @job_id=sp_js_check_id @name_id=@tjname, @cmd=0, @table='job_'
if @job_id < 0
begin
select @msgno = abs(@job_id)
raiserror @msgno, @sp_name, '@jname'
return -@@error
```

text

```
end

-- Check job name does not already exist

if exists

(select job_name

from sybmgmtdb.js_jobs

where job_name = @tjname and job_owner = suser_name())

begin

-- 19059 'Procedure %1!, a %2! already exists with the name %3!',

select @vcname=@jname

raiserror

19059, @sp_name, 'job', @vcname

return -@@error

end

--

-- build the new properties for the job

-- We ignore -2 errors as we pass in all the properties each time.

-- err -1 -> 19088 'Procedure %1!, @properties="%2!",

-- invalid prope

rty name or value.'

-- err -2 -> 19089 'Procedure %1!, @properties="%2!",

-- a property cannot be used with the scheduledjob, job or schedule.

--

exec @err=sp_js_properties 0, 'job2', @properties,

@int_jproperties out

put, @jproperties output

if @err = -1

begin

raiserror 19088, @sp_name, @properties

return -@@error

end

-- get the current date and user name for future use

select @created=getdate(), @owner=suser_name()
```

text

```
--  
-- Generate job key (id  
)  
--  
exec @job_id=sp_js_getkey 'job_id'  
if @job_id < 0  
begin  
select @msgno=abs(@job_id)  
raiserror @msgno, @sp_name, 'job_id'  
return -@@error  
end  
begin tran sp_createjob1  
--  
-- create the job  
--  
insert sybmgmtdb.js_jobs  
(job_id  
, job_type, job_properties, job_owner, job_created,  
job_default_timeout, job_tmpl_id, job_name, job_description,  
job_uproperties)  
values (@job_id, 0, @int_jproperties, @owner, @created,  
@default_timeou  
t, @tmpl_id, @tjname, @jdesc,  
@jproperties)  
select @err=@@error, @rows=@@rowcount  
-- Rollback tran if insert error  
if @err != 0  
begin  
rollback tran  
exec sp_js_putkey 'job_id', @job_id  
-- 19075 'Procedure %1!, f  
ailed to insert an entry in the %2!
```

text

```
-- table, insert error %3!.'
raiserror 19075, @sp_name, 'js_jobs', @err
return -@@error
end
commit tran
return @job_id
end
```

8.25.1.3.3 Viewing Parameters for Stored Procedure Properties

View the name, type, mode, AND order of all the procedure parameters.

The mode value indicates whether it is an input or an output parameter. The order is a numeric value that indicates the place of the parameter in the list of parameters.

To change the parameters, change the definition of the stored procedure by dropping AND re-creating the procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```
USE <database_name>
SELECT
    Parameter_name = convert(varchar(15) NULL, isnull(c.name, 'Return Type')),
    Type = isnull(convert(char(30), x.xtname), isnull(convert(char(30),
get_xtypename(c.xtype, c.xdbid), t.name)),
    Length = c.length,
    Prec = c.prec,
    Scale = c.scale,
    Param_order = c.colid,
    rtype = t.type,
    utype = t.usertype,
    @@ncharsize,
    @@unicharsize,
    Mode = case c.status2
        WHEN NULL THEN 'unknown'
        ELSE (SELECT convert(char(11),mi.description)
            FROM
                master.dbo.spt_values vi,
                master.dbo.sysmessages mi
            WHERE
                vi.number = c.status2
                AND vi.msgnum = mi.error
                AND mi.error BETWEEN 17583 AND 17586)
    END
```

```

FROM
    syscolumns c,
    systypes t,
    sysxtypes x
WHERE
    c.id = <procedure_ID>
    AND c.usertype *= t.usertype
    AND c.xtype *= x.xtid
order by
    c.colid

```

Output

Parameter_ name	Type	Length	Prec	Scale	Param_orde r	rtype	utype	Mode
@jname	univarchar	128	(Nul (Null 1	155	35	1	2	in
@jdesc	univarchar	256	(Nul (Null 2	155	35	1	2	in
@propertie s	varchar	128	(Nul (Null 3	39	2	1	2	in
@default_ti meou	int	4	(Nul (Null 4	56	7	1	2	in
@tmpl_id	int	4	(Nul (Null 5	56	7	1	2	in

8.25.1.3.4 Viewing Objects Referenced by the Stored Procedure

View the name, type, owner, and properties of objects that this procedure references.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
    name = O.name,
    owner = U.name,
    type = O.type,
    number = D.number
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND O.type = 'P'
    AND O.id = D.id
    AND D.depid = <procedure_ID>

```

Output

name	owner	type	number
sp_createjob	dbo	P	1
sp_createschedule	dbo	P	1
sp_createscheduledjob	dbo	P	1
sp_bindscheduledjob	dbo	P	1
sp_sjobcreate	dbo	P	1
sp_sjobcopy	dbo	P	1

8.25.1.3.5 Viewing Objects that Reference the Stored Procedure

View the name, type, owner, and properties of objects that reference the procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = O.name,
    owner = U.name,
    type = O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND O.type = 'P'
    AND O.id = D.depid
    AND D.id = <procedure_ID>
```

Output

name	owner	type
sp_js_getkey	dbo	P
sp_js_putkey	dbo	P
sp_js_properties	dbo	P

name	owner	type
sp_js_check_id	dbo	P

8.25.1.3.6 Viewing Tables that Reference Stored Procedure

Show the name, type, owner of objects that this procedure references.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = O.name,
    owner = U.name,
    type = O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
where
    O.uid = U.uid
    AND (O.type = 'S'
        OR O.type = 'U')
    AND O.id = D.depid
    AND D.id = <procedure_ID>
```

Output

name	owner	type
sp_createjob	dbo	P
sp_createschedule	dbo	P
sp_createscheduledjob	dbo	P
sp_bindscheduledjob	dbo	P
sp_sjobcreate	dbo	P
sp_sjobcopy	dbo	P

8.25.1.4 Managing Stored Procedure Permissions

Show and modify permissions on a stored procedure.

8.25.1.4.1 Viewing Stored Procedure Permissions

View permissions on a procedure.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helprotect](#).

Syntax

```
EXEC sp_helprotect '<owner>.<procedure_name>'
```

Output

grantor	grantee	type	action	object	column	predicate	grantable
dbo	js_admin_role	Grant	Execute	dbo.sp_createjob	All	(Null)	FALSE
dbo	js_user_role	Grant	Execute	dbo.sp_createjob	All	(Null)	FALSE

8.25.1.4.2 Granting and Revoking Stored Procedure Permissions

Grant and revoke EXECUTE permission on a stored procedure to users, groups, or roles.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT EXECUTE ON <stored_procedure_name> TO {PUBLIC | <name_list> | <role_list>}  
[WITH GRANT OPTION]
```

To revoke:

```
REVOKE EXECUTE ON <stored_procedure_name> FROM {PUBLIC | <name_list> | <role_list>}
```

8.25.1.5 Deleting a Stored Procedure

Delete stored procedures.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop procedure](#).

Syntax

```
DROP PROCEDURE [<owner>.<procedure_name>[, <owner>.<procedure_name> [,...]]
```

8.25.1.6 Generating DDL for a Stored Procedure

Generate a DDL script for stored procedures.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server> -TP -N<object_name>
```

8.25.2 Manage SQLJ Procedures

Create, delete, modify, and administer SQLJ procedures.

SQLJ procedures are named collections of SQLJ statements and flow control statements. A stored procedure that performs a select, execute, or data modification command must have the same owner as the object acted upon.

A system administrator, a database owner, or a user or group with create procedure permission can create a stored procedure.

8.25.2.1 Creating an SQLJ Procedure

Create a SQLJ procedure by adding a SQL wrapper to a Java static method. Create a SQLJ procedure by adding a SQL wrapper to a Java static method.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create procedure \(SQLJ\)](#).

Syntax

```
CREATE PROCEDURE <sql_procedure_name>
  ( [ [ in | out | inout ] <sql_parameter_name> <sql_datatype> [ ( <length> )
    | ( <precision> [, <scale> ] ) ] [=<default> ]... ] [, [ in | out | inout ]
    <sql_parameter_name> <sql_datatype> [ ( <length> )
    | ( <precision> [, <scale> ] ) ] [=<default> ]... ]
  language java parameter style java
  external name '<java_method_name>'
  [ ( [ <java_datatype> [, <java_datatype> [,... ] ] ) ] ]'
```

8.25.2.2 Replacing a SQLJ Procedure Definition

You can replace the SQL definition of a SQLJ procedure.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create procedure \(SQLJ\)](#).

Syntax

```
CREATE OR REPLACE PROCEDURE <sql_procedure_name>
  ( [ [ in | out | inout ] <sql_parameter_name> <sql_datatype> [ ( <length> )
    | ( <precision> [, <scale> ] ) ] [=<default> ]... ]
    [, [ in | out | inout ] <sql_parameter_name> <sql_datatype>
    [ ( <length> )
    | ( <precision> [, <scale> ] ) ] [=<default> ]... ]
  language java parameter style java
  external name '<java_method_name>'
  [ ( [ <java_datatype> [, <java_datatype> ... ] ] ) ]'
```

8.25.2.3 Managing SQLJ Procedure Properties

Display or modify SQLJ procedures.

8.25.2.3.1 Viewing General SQLJ Procedure Properties

View general information about SQLJ procedures.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    proc_name = o.name,
    owner_name = u.name,
    creation_date = convert(datetime,o.crdate),
    object_id = o.id
FROM
    <database_name>.dbo.sysobjects o,
    <database_name>.dbo.sysusers u
WHERE
    o.uid = u.uid
    AND o.type = 'P'
    AND u.name like '%'
    AND o.name like '<procedure_name>'
    AND sysstat2 & 33554432 = 33554432
ORDER BY
    1
```

Output

proc_name	owner_name	creation_date	object_id
my_max	dbo	2020-03-20 02:16:38.033	1737054193

8.25.2.3.2 Viewing the SQL for SQLJ Procedure

SQL statements for creating the procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    text = B.text
FROM
    <database_name>.dbo.sysobjects A,
    <database_name>.dbo.syscomments B
WHERE
    A.id = <procedure_ID>
    AND A.id = B.id
    AND number = <group_number>
ORDER BY
    B.colid2,
    B.colid
```

Output

text

create procedure my_max (a int = 10, b int = 10)

language java parameter style java

external name 'java.lang.Math.max'

(1 row affected)

8.25.2.3.3 Viewing the Parameters for SQLJ Procedures

View name, type, mode and order of procedure parameters.

Mode indicates whether it is an input or an output parameter. Order indicates the order of all the procedure parameters.

The following syntax provides basic guidance for viewing this information.

Syntax

```
USE <database_name>
SELECT
    Parameter_name = convert(varchar(15) NULL, isnull(c.name, 'Return Type')),
    Type = isnull(convert(char(30), x.xtname), isnull(convert(char(30),
get_xtypename(c.xtype, c.xdbid), t.name)),
    Length = c.length,
    Prec = c.prec,
    Scale = c.scale,
    Param_order = c.colid,
    rtype = t.type,
    utype = t.usertype,
```

```

@@ncharsize,
@@unicharsize,
Mode = case c.status2
  WHEN NULL THEN 'unknown'
  ELSE (SELECT convert(char(11), mi.description)
        FROM master.dbo.spt_values vi,
        master.dbo.sysmessages mi
        WHERE vi.number = c.status2
              AND vi.msgnum = mi.error
              AND mi.error BETWEEN 17583 AND 17586)
  END
FROM
  syscolumns c,
  systypes t,
  sysxtypes x
WHERE
  c.id = <procedure_ID>
  AND c.usertype *= t.usertype
  AND c.xtype *= x.xtid
ORDER BY
  c.colid

```

Output

Parameter_ name	Type	Length	Prec	Scale	Param_order	rtype	utype	Mode
@a	int	4	(Nul (Null 1	56	7	1	2	in
@b	int	4	(Nul (Null 2	56	7	1	2	in

8.25.2.3.4 Viewing Objects Referenced By the SQLJ Procedure

View the name, type, AND owner of objects referenced by the procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  name = O.name,
  owner = U.name,
  type = O.type,
  number = D.number
FROM
  <database_name>.dbo.sysdepends D,
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysusers U
WHERE
  O.uid = U.uid
  AND O.type = 'P'

```

```
AND O.id = D.id
AND D.depid = <procedure_ID>
```

Output

Name	owner	type	number
sp_testcreatejob	dbo	P	1

8.25.2.3.5 Viewing Objects that Reference The SQLJ Procedure

View the name, type, AND owner of objects that reference the procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = O.name,
    owner = U.name,
    type = O.type
FROM
    <database_name>.dbo.sysdepends D,
    <database_name>.dbo.sysobjects O,
    <database_name>.dbo.sysusers U
WHERE
    O.uid = U.uid
    AND O.type = 'P'
    AND O.id = D.depid
    AND D.id = <procedure_ID>
```

Output

name	owner	type
sp_js_getkey	dbo	P

8.25.2.4 Managing Permissions for SQLJ Procedures

Show and modify permissions on an SQLJ procedure.

8.25.2.4.1 Viewing SQLJ Procedure Permissions

View permissions on an SQLJ procedure.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helpprotect](#).

Syntax

```
EXEC sp_helpprotect '<owner>.<procedure_name>'
```

Output

grantor	grantee	type	action	object	column	predicate	grantable
dbo	guest	Grant	Execute	dbo.my_max	All	(Null)	FALSE
dbo	jstask	Grant	Execute	dbo.my_max	All	(Null)	FALSE
dbo	probe	Grant	Execute	dbo.my_max	All	(Null)	FALSE

8.25.2.4.2 Granting and Revoking SQLJ Procedure Permissions

Grant or revoke EXECUTE permission on SQLJ procedures for users, groups, and roles.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [.revoke](#)

Syntax

To grant:

```
GRANT SELECT ON <owner_name>.<prs_name> TO { PUBLIC | <name_list> | <role_list> } [ WITH GRANT OPTION ]
```

To revoke:

```
REVOKE SELECT ON <owner_name>.<prs_name> FROM { PUBLIC | <name_list> |  
<role_list> }
```

8.25.2.5 Deleting an SQLJ Procedure

Delete SQLJ procedures.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop procedure](#).

Syntax

```
DROP PROCEDURE [<owner>.<procedure_name> [, <owner>.]<procedure_name> [,... ]
```

8.25.2.6 Generating DDL for an SQLJ Procedure

Generate a DDL script for SQLJ procedures.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server> -TP -N<object_name>
```

DDL for Stored Procedure 'master.dbo.my_max'

```
print '<<<<< CREATING Stored Procedure - "master.dbo.my_max" >>>>>'
```

```
go
```

```
use master
```

```
go
```

```
IF EXISTS (SELECT 1 FROM sysobjects o, sysusers u WHERE o.uid=u.uid AND o.name = 'my_max' AND u.name = 'dbo'  
AND o.type = 'P')
```

```
BEGIN
```

```
setuser 'dbo'
```

DDL for Stored Procedure 'master.dbo.my_max'

```
drop procedure my_max
END
go
IF (@@error != 0)
BEGIN
PRINT 'Error CREATING Stored Procedure master.dbo.my_max'
SELECT syb_quit()
END
go
setuser 'dbo'
go
create procedure my_max (a int = 10, b int = 10)
language java parameter style java
external name 'java.lang.Math.max'
go
sp_procxmode 'my_max', unchained
go
setuser
go
```

8.26 Manage Extended Stored Procedures

Create, delete, modify, and administer extended stored procedures.

8.26.1 Creating an Extended Stored Procedure

Create an extended stored procedure.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [Creating an Extended Stored Procedure](#).

Syntax

```
CREATE PROCEDURE <owner>.<procedure_name> AS EXTERNAL NAME '<dll_name>'
```

<procedure_name> Specifies the name of the procedure. It must conform to the rules for identifiers and cannot be a variable.

<dll_name> Specifies the name of the dynamic link library (DLL) or shared library containing the functions that implement the extended stored procedure.

8.26.2 Replacing an Extended Stored Procedure Definition

You can replace the SQL definition of a extended procedure or change whether the procedure is recompiled each time it is executed.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create procedure](#).

Syntax

```
CREATE [OR REPLACE] PROCEDURE <owner>.<procedure_name > AS EXTERNAL NAME '<dll_name>'
```

8.26.3 Extended Stored Procedure Properties

Display or modify extended stored procedures.

8.26.3.1 Viewing Extended Stored Procedure Properties

View the name, type, database, owner, creation date, AND dynamic link library (DLL) path of the stored procedure. The DLL need not exist when you create the extended stored procedure, but it must exist when you execute the extended stored procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
```

```

O.name name,
C.text dll_name,
U.name owner,
convert(datetime, O.crdate) creationdate,
O.id oid
FROM
  <database_name>.dbo.syscomments C,
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysusers U
WHERE
  O.type = 'XP'
  AND C.id = O.id
  AND C.colid = 1
  AND O.uid = U.uid
  AND U.name like '%'
  AND O.name like '<procedure_name>'
ORDER BY
  1

```

Output

name	dll_name	owner	creationdate	oid
xp_freelld	sybsyesp	dbo	2019-12-18 22:51:02.096	284525016

8.26.3.2 Viewing Objects Referenced by Extended Stored Procedures

View the name, type, owner, AND properties of objects that are referenced by this extended stored procedure.

The following syntax provides basic guidance for viewing this information.

Syntax

```

SELECT
  O.name,
  U.name,
  O.type,
  D.number
FROM
  <database_name>.dbo.sysdepends D,
  <database_name>.dbo.sysobjects O,
  <database_name>.dbo.sysusers U
WHERE
  O.uid = U.uid
  AND O.type = 'P'
  AND O.id = D.id
  AND D.depid = <procedure_id>

```

Output

Name	name	type	number
sp_modifylogin	dbo	P	1
sp_addobjectdef	dbo	P	1
sp_defaultloc	dbo	P	1
sp_helpobjectdef	dbo	P	1
sp_showoptstats	dbo	P	1
sp_autoformat	dbo	P	1
sp_replication_path	dbo	P	1
sp_imrs_check_and_namecrack	dbo	P	1
sp_help	dbo	P	1
sp_showtext	dbo	P	1

8.26.4 Managing Extended Stored Procedure Permissions

View and modify permissions on extended stored procedures.

8.26.4.1 Viewing Extended Stored Procedure Permissions

Show the permissions on an extended stored procedure.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helpprotect](#).

Syntax

```
USE < database_name >  
EXEC sp_helpprotect '<owner>.<procedure_name>'
```

Output

grantor	grantee	type	action	object	column	predicate	grantable
dbo	guest	Grant	Execute	dbo.xp_freedl 	All	(Null)	FALSE)

8.26.4.2 Granting and Revoking Extended Stored Procedure Permissions

Grant and revoke execute permission on extended stored procedures.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

```
GRANT EXECUTE ON <owner>.<procedure_name> TO {PUBLIC | <name_list> |  
<role_list>} [ WITH GRANT OPTION ]
```

To revoke:

```
REVOKE EXECUTE ON <owner>.<procedure_name> FROM {PUBLIC | <name_list> |  
<role_list>}
```

8.26.5 Deleting an Extended Stored Procedure

Delete extended stored procedures.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop procedure](#).

Syntax

```
DROP PROCEDURE <owner>.<procedure_name>
```

8.26.6 Generating DDL for an Extended Stored Procedure

Generate a DDL script for extended stored procedures.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen - U<login> - P<password> - S<server> - TXP - N<procedure_name>
```

Output

DDL for Extended Stored Procedure 'IMDB.dbo.myesp'

```
print '<<<<< CREATING Extended Stored Procedure - "IMDB.dbo.myesp" >>>>>'
go
use IMDB
go
set quoted_identifier on
go
IF EXISTS (SELECT 1 FROM sysobjects o, sysusers u WHERE o.uid=u.uid AND o.name = 'myesp' AND u.name = 'dbo' AND
o.type = 'XP')
BEGIN
setuser 'dbo'
drop procedure "myesp"
END
go
set quoted_identifier off
go
IF (@@error != 0)
BEGIN
PRINT 'Error CREATING Extended Stored Procedure IMDB.dbo.myesp'
SELECT syb_quit()
END
```

DDL for Extended Stored Procedure 'IMDB.dbo.myesp'

```
go
setuser 'dbo'
go
set quoted_identifier on
go
CREATE PROCEDURE dbo.myesp
AS EXTERNAL NAME 'esplib'
go
Grant Execute on dbo.myesp to public Granted by dbo
go
sp_procxmode 'myesp', unchained
go
set quoted_identifier off
go
setuser
go
```

8.27 Manage Functions

Manage scalar and SQLJ functions.

8.27.1 Scalar Functions

Create, delete, modify, and administer scalar functions.

8.27.1.1 Creating a Scalar Function

A scalar function takes a list of scalar arguments and return a single scalar value.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create function](#).

Syntax

```
CREATE FUNCTION [<owner>.]<function_name>
  ( [ [<@parameter_name> <parameter_datatype>
    [, <@parameter_name> <parameter_datatype> [,...n] ] ] )
  RETURNS <return_datatype> AS
  [BEGIN]
    <function_body>
    RETURN <scalar_expression>
  [END]
```

<function_name>

Specifies the name of the user-defined function. Function names must conform to the rules for identifiers and must be unique within the database and to its owner. Function names cannot be the same as other SAP ASE built-in functions.

<@parameter_name>

Specifies the parameter in the user-defined function. You can declare one or more parameters in a create function statement.

A function can have up to 2,047 parameters. The user supplies the value of each declared parameter when the function is executed, unless you define a default for the parameter.

A table user-defined function can have up to 1,023 parameters.

Specify a parameter name using an "@" sign (@) as the first character. The parameter name must conform to the rules for identifiers. Parameters are local to the function. You can use the same parameter names in other functions.

If a parameter has a default value, the user must specify the keyword "default" when they call the function to get the default value. This behavior differs from parameters with default values in stored procedures in which omitting the parameter also implies the default value.

<parameter_datatype>

Specifies the datatype of the parameter. You can use all scalar datatypes, Java abstract datatypes (ADTs), and table types as a parameter for user-defined and table user-defined functions. However, user-defined functions do not support the timestamp, text, image and unitext.

<return_datatype>

Specifies the return value of a scalar, user-defined function. <return_datatype> can be any of the scalar datatypes and Java abstract datatypes except text, image, unitext, and timestamp.

<function_body>

Specifies a series of Transact-SQL statements that together generates a set of rows that define the value of the function, but which together do not produce a side effect. Use <function_body> only in scalar functions and multistatement table-valued functions. In scalar functions, <function_body> is a series of Transact-SQL statements that evaluate to a scalar value.

<scalar_expression>

Specifies the scalar value the scalar function returns.

You can invoke scalar-valued functions where scalar expressions are used, including computed columns and check constraint definitions.

8.27.1.2 Replacing a Scalar User Defined Function Definition

You can replace the SQL definition, the return type, and change whether the function is recompiled each time it is executed.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create function](#).

Syntax

```
CREATE OR REPLACE FUNCTION [<owner>.<function_name>
  ( [ <@parameter_name> <parameter_datatype>
    [, <@parameter_name> <parameter_datatype> [,...n] ] )
  RETURNS <return_datatype> AS
  [BEGIN]
    <function_body>
    RETURN <scalar_expression>
  [END]
```

<function_name>

Specifies the name of the user-defined function. Function names must conform to the rules for identifiers and must be unique within the database and to its owner. Function names cannot be the same as other SAP ASE built-in functions.

When used with *OR REPLACE*, the name of the function remains the same, although its definition is changed.

<@parameter_name>

Specifies the parameter in the user-defined function. You can declare one or more parameters in a create function statement.

A function can have up to 2,047 parameters. The user supplies the value of each declared parameter when the function is executed, unless you define a default for the parameter.

A table user-defined function can have up to 1,023 parameters.

Specify a parameter name using an "at" sign (@) as the first character. The parameter name must conform to the rules for identifiers. Parameters are local to the function. You can use the same parameter names in other functions.

If a parameter has a default value, the user must specify the keyword "default" when they call the function to get the default value. This behavior differs from parameters with default values in stored procedures in which omitting the parameter also implies the default value.

When used with *OR REPLACE*, you can alter the names and number of parameters.

<parameter_datatype>

Specifies the datatype of the parameter. You can use all scalar datatypes, Java abstract datatypes (ADTs), and table types as a parameter for user-defined and table user-defined functions. However, user-defined functions do not support the timestamp, text, image and unitext.

When used with *OR REPLACE*, you can change the datatype of the parameter to the function.

<return_datatype>

Specifies the return value of a scalar, user-defined function. **<return_datatype>** can be any of the scalar datatypes and Java abstract datatypes except text, image, unitext, and timestamp.

When used with *OR REPLACE*, you can alter the return datatype of the function.

<function_body>

Specifies a series of Transact-SQL statements that together generates a set of rows that define the value of the function, but which together do not produce a side effect. Use **<function_body>** only in scalar functions and multistatement table-valued functions. In scalar functions, **<function_body>** is a series of Transact-SQL statements that evaluate to a scalar value.

When used with *OR REPLACE*, you can change the SQL statements that define the value of the function

<scalar_expression>

Specifies the scalar value the scalar function returns.

You can invoke scalar-valued functions where scalar expressions are used, including computed columns and check constraint definitions.

When used with *OR REPLACE*, you can change the value returned by the function.

8.27.1.3 Scalar Function Properties

Display or modify scalar function properties.

8.27.1.3.1 Viewing General Scalar Function Properties

Show the name, type, database, owner, creation date, group number for a scalar function.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = o.name,
    owner_name = u.name,
    creation_date = convert(datetime,
    o.crdate),
    object_id = o.id,
    object_type = o.type
FROM
    <database_name>.dbo.sysobjects o,
    <database_name>.dbo.sysusers u
WHERE
    o.uid = u.uid
    AND o.id = <function_ID>
    AND o.type in ('SF')
    AND lower(o.name)
LIKE '%'
    AND u.name like '%'
ORDER BY
    1
```

Output

name	owner_name	creation_date	object_id	object_type
nameFunction	dbo	2020-03-09 04:17:41.453	1641053851	SF

8.271.3.2 Viewing SQL for Scalar Functions

Show the SQL statements for creating the function.

To change the parameters, change the definition of the function by dropping and re-creating the function.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    TEXT
FROM
    <database_name>.dbo.syscomments
WHERE
    id = <function_ID>
ORDER BY
    colid2, colid
```

Output

text

```
create or replace function sp_f_size2count(
@size_spec varchar(30)
,@to_what char(2) = 'AU'
) returns bigint
as
begin
declare @accum numeric(38,12)
,@digit numeric(38,12)
,@ret_ct bigint -- output
,@10 bigint
,@KB bigint
,@GB bigint
,@max
_db_pages bigint
,@max_db_size bigint
,@MB bigint
,@mpier bigint
,@result bigint
,@shift bigint
,@TB bigint
,@b_ix int
,@blen int
,@c char(1)
,@me varchar(30)
-- Error codes for different input / parsing errors
,@err_illegal_output_un
its int
,@err_input_too_wide int
,@err_syntax_in_input int
,@err_illegal_input int
```

text

```
-- Sanitize our input variables
set @size_spec = ltrim( rtrim( isnull(@size_spec, '0') ) )
,@to_what = isnull(@to_what, 'AU')
-- Initialize
set @10 = 10
,@accu
m = 0.0
,@b_ix = 1
,@blen = datalength(@size_spec)
,@me = 'sp_f_size2count'
,@mpier = @@maxpagesize
,@ret_ct = 0
,@shift = 0
,@err_illegal_output_units = -1
,@err_input_too_wide = -2
,@err_syntax_in_input = -3
,@err_illegal_input = -4
set @KB = 1024
set @MB = @KB * @KB
set @GB = @MB * @KB
set @TB = @GB * @KB
-- Develop max database size as pages, then as bytes. Max possible page
-- count is 2**32 - 256. Max size is max pages * @@maxpagesize.
set @max_db_pages = 2
set @max_db_p
ages = power(@max_db_pages, 32) - 256
set @max_db_size = @max_db_pages * @@maxpagesize
-- Check what output is desired: AU (alloc units), EX (extents), PG (pages).
if upper(@to_what) not in ('AU', 'EX', 'PG')
begin
-- Unknown conversion request. Return
an error.
```

text

```
raiserror 19078, @me, @to_what
return @err_illegal_output_units
end
/*
** Begin by converting the number into 2 parts:
** - a number portion in @accum
** - a size spec in @mpier, indicated by a letter at the end of the string
** Our syntax
is "\d*.?\d*[BPKMGT]?". Any string that doesn't fit that syntax
** causes us to exit with a syntax error. If the integer portion overflows
** our work buffer, we exit with an overflow error.
*/
while @b_ix <= @blen
begin
set @c = substring(@size_spec, @
b_ix, 1)
set @b_ix = @b_ix + 1
if @c like '[0-9]'
begin
-- Collect another integer digit in our work buffer.
-- We collect max 12 decimal places worth of fraction.
-- If we encounter an overflow, stop.
if @shift < 12
begin
select @digit =
convert(int, @c)
if @shift > 0
begin
set @accum = @accum
+ (@digit / power(@10, @shift))
set @shift = @shift + 1
end
end
end
end
```

text

```
else
if @accum >= @max_db_size
begin
raiserror 18291, @size_spec
return @err_input_too_wide
end
else
begin
-- Add this digit to our accumulator
set @accum = (@accum * 10.0) + @digit
end
end
end
-- "." is a fraction separator
else if @c in (',')
begin
-- This number will be a fraction
if @shift > 0
begin
-- Syntax err
or: too many decimal points in number
raiserror 19913, @size_spec
return @err_syntax_in_input
end
set @shift = 1
end
else if @b_ix <= @blen
begin
-- Syntax error: non-digit, non-decimal embedded in number
raiserror 19913, @size_spec
ret
urn @err_syntax_in_input
```

text

```
end
else
begin
-- The input seems to contain a units specifier
set @c = upper(@c)
set @mpier = case when @c = 'B' then 1
when @c = 'P' then @@maxpagesize
when @c = 'K' then @KB
when @c = 'M' then @MB
when @c = 'G' then @GB
when @c = 'T' then @TB
else -1
end
if @mpier < 0
begin
-- Syntax error: unknown units specifier
raiserror 19078, @me, @c
return @err_illegal_input
end
end
end
/*
** We have a size. Convert th
at to # bytes. We know that our byte limit will
** fit into a bigint: a database can be no larger than (16384 * (2^32 - 256))
** = 70,368,739,983,360 bytes. Check early for overflow: '99999999999999T'
** can cause overflow errors on conversion, but that s
ize is ludicrous; we
** are limited to 64T.
*/
if (@c = 'T') and (@accum > 64.0)
begin
```

text

```
-- This request can't fit our size limit.

raiserror 18291, @size_spec

return @err_input_too_wide

end

set @accum = @accum * @mpier

if @accum > @max_db_size

begin

--

This request can't fit our size limit.

raiserror 18291, @size_spec

return @err_input_too_wide

end

/*

** Now @accum has a number of bytes. Convert that to a page count.

** If that page count is beyond our limit, we can't work with it.

*/

set @result = (

convert(bigint, @accum) + @@maxpagesize - 1) / @@maxpagesize

if @result > @max_db_pages

begin

-- The size overflows our limit.

raiserror 18291, @size_spec

return @err_input_too_wide

end

/*

** Convert # of pages into the user's requested units, and we'

re done.

*/

set @mpier = case when upper(@to_what) = 'AU' then 256

when upper(@to_what) = 'EX' then 8

else 1

end
```

text

```
set @ret_ct = (@result + @mpier - 1) / @mpier

return @ret_ct

end -- }
```

8.271.3.3 Viewing Scalar Function Parameters Properties

View scalar function parameter properties.

The following syntax provides basic guidance for viewing this information.

Syntax

```
USE <database_name>
SELECT
    Parameter_name = convert(varchar(15) NULL,
        isnull(c.name,
            'Return Type')),
    Type = isnull(convert(char(30),
        x.xtname),
        isnull(convert(char(30),
            get_xtypename(c.xtype,
                c.xdbid),
                t.name)),
            Length = c.length,
            Prec = c.prec,
            Scale = c.scale,
            Param_order = c.colid,
            rtype = t.type,
            utype = t.usertype,
            @@ncharsize,
            @@unicharsize,
            Mode = case c.status2
                WHEN NULL then 'unknown'
                ELSE (SELECT convert(char(11),
                    mi.description)
                FROM
                    master.dbo.spt_values vi,
                    master.dbo.sysmessages mi
                WHERE
                    vi.number = c.status2
                    AND vi.msgnum = mi.error
                    AND mi.error BETWEEN 17583 AND 17586)
            END
FROM
    syscolumns c,
    systypes t,
    sysxtypes x
WHERE
    c.id = <function_ID>
    AND c.usertype != t.usertype
    AND c.xtype != x.xtid
ORDER BY
```

Output

Parameter_ name	Type	Length	Prec	Scale	Param_order	rtype	utype	Mode	
@firstname	char	30	(Null	(Null 1	47	1	1	2	in
@lastname	char	30	(Null	(Null 2	47	1	1	2	in
Return Type	char	61	(Null	(Null 3	47	1	1	2	out

8.27.1.4 Managing Scalar Function Permissions

Create, delete, modify, and administer scalar functions.

8.27.1.4.1 Viewing Scalar Function Permissions

View the permissions on a scalar function.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_helpprotect](#).

Syntax

```
exec sp_helpprotect '<owner>.<function_name>'
```

Output

grantor	grantee	type	action	object	column	predicate	grantable
dbo	guest	Grant	Execute	dbo.nameFunction	All	(Null)	FALSE
dbo	jstask	Grant	Execute	dbo.nameFunction	All	(Null)	FALSE

grantor	grantee	type	action	object	column	predicate	grantable
dbo	probe	Grant	Execute	dbo.nameFunction	All	(Null)	FALSE

8.271.4.2 Granting and Revoking Scalar Function Permissions

Grant and remove execute permission on scalar functions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [grant](#) and [revoke](#).

Syntax

To grant:

To grant:

```
GRANT EXECUTE on <owner>.<SQL_function_name> TO { PUBLIC | <name_list> | <role_list> } [WITH GRANT OPTION]
```

To revoke:

```
REVOKE EXECUTE on <owner>.<SQL_function_name> FROM { PUBLIC | <name_list> | <role_list> }
```

8.271.5 Deleting a Scalar Function

Delete scalar functions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop function](#).

Syntax

```
DROP FUNCTION [ <owner_name>.] <function_name> [, ...n]
```

8.27.1.6 Generating DDL for a Scalar Function

Generate a DDL script for scalar functions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server> -TF -N<object_name>
```

Output

-- DDL for Function 'master.dbo.nameFunction'

```
print '<<<<< CREATING Function - "master.dbo.nameFunction" >>>>>'
```

```
go
```

```
use master
```

```
go
```

```
set quoted_identifier on
```

```
go
```

```
IF EXISTS (SELECT 1 FROM sysobjects o, sysusers u WHERE o.uid=u.uid AND o.name = 'nameFunction' AND u.name = 'dbo' AND (o.type = 'F' or o.type = 'SF'))
```

```
BEGIN
```

```
setuser 'dbo'
```

```
drop function "nameFunction"
```

```
END
```

```
go
```

```
set quoted_identifier off
```

```
go
```

```
IF (@@error != 0)
```

```
BEGIN
```

```
PRINT 'Error CREATING Function master.dbo.nameFunction'
```

```
SELECT syb_quit()
```

```
END
```

```
-- DDL for Function 'master.dbo.nameFunction'
```

```
go
setuser 'dbo'
go
set quoted_identifier on
go
create function dbo.nameFunction(
@firstname char(30),
@lastname char(30))
returns char(61)
as
begin
declare @name char(61)
set @name = @firstname||' '||@lastname
return @name
end
go
Grant Execute on dbo.nameFunction to guest Granted by dbo
go
Grant Execute on dbo.nameFunction to probe Granted by dbo
go
Grant Execute on dbo.nameFunction to jstask Granted by dbo
go
set quoted_identifier off
go
setuser
go
```

8.27.2 SQLJ Functions

Create, delete, modify , and administer SQLJ functions.

8.27.2.1 Creating a SQLJ Function

Create a user-defined function by adding a SQL wrapper to a Java static method.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create function \(SQLJ\)](#).

Syntax

```
CREATE FUNCITON [<owner>.<sql_function_name>
  ( [ <sql_parameter_name> <sql_datatype>
    [ { (<length>) | (<precision> [ , <scale> ] ) } ]
  [ , [ <sql_parameter_name> <sql_datatype>
    [ { (<length>) | (<precision> [ , <scale> ] ) } ]
    [ ,... ] ] )
  RETURNS <sql_datatype>
  LANGUAGE JAVA
  PARAMETER STYLE JAVA
  EXTERNAL NAME '<java_method_name>'
    [ ( [ <java_datatype> [ , <java_datatype> [ ,... ] ] ) ] )'
```

<sql_function_name>

Specifies the Transact-SQL name of the function, must conform to the rules for identifiers, and cannot be a variable. The name remains the same even when you use or replace.

<sql_parameter_name>

Specifies the name of an argument to the function. The value of each input parameter is supplied when the function is executed. Parameters are optional; a SQLJ function need not take arguments.

Parameter names must conform to the rules for identifiers. If the value of a parameter contains nonalphanumeric characters, it must be enclosed in quotes. This includes object names qualified by a database name or owner name, since they include a period. If the value of the parameter begins with a numeric character, it also must be enclosed in quotes.

<sql_datatype> [{ (<length>) | (<precision> [, <scale>]) }

Specifies the Transact-SQL datatype of the parameter. See create procedure for more information about these parameters.

<sql_datatype> is the SQL procedure signature.

<java_method_name>

Specifies the name of the external Java method.

<java_datatype>

Specifies a Java datatype that is mappable or result-set mappable. This is the Java method signature.

8.27.2.2 Replacing a SQLJ Function Definition

You can replace the SQL definition, the return type, and change whether the function is recompiled each time it is executed.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [create function \(SQLJ\)](#).

Syntax

```
CREATE OR REPLACE FUNCITON [<owner>.<sql_function_name>
  ( [ <sql_parameter_name> <sql_datatype>
    [ { (<length>) | (<precision> [, <scale> ] ) } ]
  [, [ <sql_parameter_name> <sql_datatype>
    [ { (<length>) | (<precision>[, <scale> ] ) } ]
  [,... ] ] )
  RETURNS <sql_datatype>
  LANGUAGE JAVA
  PARAMETER STYLE JAVA
  EXTERNAL NAME '<java_method_name>'
  [ ( [ <java_datatype> [, <java_datatype> [,... ] ] ) ] ]'
```

<sql_function_name>

Specifies the Transact-SQL name of the function, must conform to the rules for identifiers, and cannot be a variable. The name remains the same even when you use or replace.

<sql_parameter_name>

Specifies the name of an argument to the function. The value of each input parameter is supplied when the function is executed. Parameters are optional; a SQLJ function need not take arguments.

Parameter names must conform to the rules for identifiers. If the value of a parameter contains nonalphanumeric characters, it must be enclosed in quotes. This includes object names qualified by a database name or owner name, since they include a period. If the value of the parameter begins with a numeric character, it also must be enclosed in quotes.

Use *OR REPLACE* to change the names and the number of parameters.

<sql_datatype> [{ (<length>) | (<precision> [, <scale>]) }

Specifies the Transact-SQL datatype of the parameter. See create procedure for more information about these parameters.

<sql_datatype> is the SQL procedure signature.

Use *OR REPLACE* to change the Transact-SQL datatype of the parameter to the function.

<java_method_name>

Specifies the name of the external Java method.

Use *OR REPLACE* to change the Java method.

<java_datatype>

Specifies a Java datatype that is mappable or result-set mappable. This is the Java method signature.

Use *OR REPLACE* to change the Java datatype.

8.27.2.3 SQLJ Function Properties

Display or modify SQLJ functions.

8.27.2.3.1 Viewing General SQLJ Function Properties

View general information on the properties of an SQLJ function.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    name = o.name,
    owner_name = u.name,
    creation_date = convert(datetime,
        o.crdate),
    object_id = o.id,
    object_type = o.type
FROM
    <database_name>.dbo.sysobjects o,
    <database_name>.dbo.sysusers u
WHERE
    o.uid = u.uid
    AND o.id = <function_ID>
    AND o.type in (
        'F'
    )
    AND lower(o.name)
    LIKE '%'
    AND u.name like '%'
ORDER BY
    1
```

Output

name	owner_name	creation_date	object_id	object_type
region_of1	dbo	2020-03-20 02:09:21.993	1721054136	F

8.272.3.2 Viewing SQL for SQLJ Functions

Show SQL statements for creating the function.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT TEXT FROM <database_name>.dbo.syscomments
WHERE
    id = <function_ID>
ORDER BY
    colid2, colid
```

Output

text

```
create function region_of1(state char(20))
returns integer
language java parameter style java
external name 'SQLJExamples.region(java.lang.String)'
```

8.272.3.3 Viewing SQLJ Function Parameters Properties

View SQLJ function parameter properties.

To change the parameters, change the definition of the function by dropping AND re-creating the function.

The following syntax provides basic guidance for viewing this information.

Syntax

```

USE <database_name>
SELECT
    Parameter_name = convert(varchar(15) NULL,
        isnull(c.name,
            'Return Type')),
    Type = isnull(convert(char(30),
        x.xtname),
        isnull(convert(char(30),
            get_xtynename(c xtype,
                c.xdbid),
                t.name))),
        Length = c.length,
        Prec = c.prec,
        Scale = c.scale,
        Param_order = c.colid,
        rtype = t.type,
        utype = t.usertype,
        @@ncharsize,
        @@unicharsize,
        Mode = case c.status2
            WHEN NULL THEN 'unknown'
            ELSE (SELECT
                convert(char(11),
                    mi.description)
            FROM
                master.dbo.spt_values vi,
                master.dbo.sysmessages mi
            WHERE
                vi.number = c.status2
                AND vi.msgnum = mi.error
                AND mi.error between 17583 AND 17586)
        END
FROM
    syscolumns c,
    systypes t,
    sysxtypes x
WHERE
    c.id = <function_ID>
    AND c.usertype *= t.usertype
    AND c.xtype *= x.xtid
ORDER BY
    c.colid

```

Output

Parameter _name	Type	Length	Prec	Scale	Parameter _name	Type	Length	Prec	Scale
Return Type	int	4	(Nul (Null 0	56	Return Type	int	4	(Nul (Null 0	56
@state	char	20	(Nul (Null 1	47	@state	char	20	(Nul (Null 1	47

8.27.2.4 Deleting a SQLJ Function

Delete SQLJ functions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [drop function \(SQLJ\)](#).

Syntax

```
DROP FUNCTION [<owner>.<function_name>
              [, [<owner>.<function_name>]
```

8.27.2.5 Generating DDL for a SQLJ Function

Generate a DDL script for SQLJ functions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [ddlgen](#).

Syntax

```
ddlgen -U<login> -P<password> -S<server> -TF -N<object_name>
```

Output

```
-- DDL for Function 'master.dbo.region_of1'
```

```
print '<<<<< CREATING Function - "master.dbo.region_of1" >>>>>'
```

```
go
```

```
use master
```

```
go
```

```
IF EXISTS (SELECT 1 FROM sysobjects o, sysusers u WHERE o.uid=u.uid AND o.name = 'region_of1' AND u.name = 'dbo'
AND (o.type = 'F' or o.type = 'SF'))
```

```
BEGIN
```

```
setuser 'dbo'
```

```
-- DDL for Function 'master.dbo.region_of1'
```

```
drop function region_of1  
  
END  
  
go  
  
IF (@@error != 0)  
  
BEGIN  
  
PRINT 'Error CREATING Function master.dbo.region_of1'  
  
SELECT syb_quit()  
  
END  
  
go  
  
setuser 'dbo'  
  
go  
  
create function region_of1(state char(20))  
  
returns integer  
  
language java parameter style java  
  
external name 'SQLJExamples.region(java.lang.String)'  
  
go  
  
setuser  
  
go
```

8.28 Manage Data Store Access Management (DSAM)

Data store access management (DSAM) tracks access patterns for devices, segments, and data partitions, providing detailed insights on how data is accessed in an SAP ASE database. DSAM allows you to define data storage policies that compress and move data to appropriate storage based on the activity level of data.

Benefits for using DSAM include:

- Being able to monitor data usage.
- Moving data to faster devices or to archive devices based on data access activity.
- Enhancing performance by keeping active data on high performance storage.
- Enabling a more cost-effective use of storage assets.

8.28.1 DSAM Configuration

Use the DSAM Guidance Wizard the first time you configure DSAM. Changes can be made after this initial setup using DSAM Configure.

8.28.1.1 Configuring DSAM for the First Time

Use the DSAM Guidance Wizard to configure DSAM for the first time.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [sp_sjobcreate](#) and [sp_sjobmodify](#).

Syntax

```
EXEC sybmgmtdb.sp_sjobcreate @name = 'sname=<sname>',
    @option = 'sdesc = '<sdesc>', startdate = '<startdate>',
    starttime = '<starttime>', endtime = '<endtime>', repeats = <repeats>'
GO
EXEC sybmgmtdb.sp_sjobcreate @name = 'jname=<jname>', @option = 'jdesc =
    '<jdesc>',
    default_timeout = <default_timeout>'
GO
EXEC sybmgmtdb.sp_sjobmodify @name = 'name=<sjobname>', @option = '@jcmd =
    '<jcmd>''
GO
EXEC sybmgmtdb.sp_sjobcreate @name = '<name>', @option = 'sname = '<sname>',
    server = '<server>', timeout = <timeout>'
GO
```

<sname> The name of the schedule. When you create a name for a schedule, the name must begin with a letter. If you create a name beginning with a digit, an error occurs.

<repeats> Specifies the interval at which the schedule repeats. 0 (zero) or NULL means it does not repeat. The value is a number followed by:

- day or d
- days or dd
- hour or h
- hours or hh
- minute or m
- minutes or mm

<jname> Specifies the name of the job, which must be unique.

<default_timeout> Specifies the maximum amount of time allowed for the execution of a scheduled job.

<name> Specifies the name of the schedule job.

<jcmd> Specifies the SQL text, used for simple jobs when it is easy to provide the text directly.

<timeout> Specifies the maximum amount of time permitted for the execution of a job.

8.28.1.2 Defining the DSAM Collection Frequency

After you perform initial DSAM setup using the DSAM Guidance Wizard, you can use DSAM Configure to change your collection frequency settings.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_set_collection_frequency](#).

Syntax

```
EXEC dsam_set_collection_frequency <@time>
```

<@time> Specifies the time specifier to be decoded. It holds a string-encoded number and an optional units specifier.

8.28.1.3 Enabling Database Monitoring

When you enable database monitoring, all of its active partition activities (physical I/O) are recorded.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [enable monitoring](#) and [sp_configure](#).

Syntax

```
EXEC sp_configure 'enable monitoring', 1
```

Output

Parameter Name	Default	Memory Used	Config Value	Run Value	Unit	Type
enable monitoring	0	0	1	1	switch	dynamic

Resulting configuration value and memory use have not changed from previous values: new configuration value 1, previous value 1.

8.28.1.4 Device Classes

Add, edit, or delete device classes.

8.28.1.4.1 Adding a Device Class

Add a device class.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_add_device_class](#).

Syntax

```
EXEC dsam_add_device_class <@tag>, <@name>, <@comment>
```

- <@tag>** Specifies a two-character code for the device class to be added.
- <@name>** Specifies the name of the device class.
- <@comment>** Specifies a comment that describes this device class.

8.28.1.4.2 Deleting a Device Class

Delete a device class.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsm_delete_device_class](#).

Syntax

```
EXEC dsam_delete_device_class <@name>
```

- <@name>** Specifies the name or tag of the device class to be removed.

8.28.1.4.3 Editing a Device Class

Edit a device class.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_update_device_class](#).

Syntax

```
EXEC dsam_update_device_class <@tag>, <@name>, <@comment>
```

- <@tag>** Specifies a two-character code for the device class being modified.
- <@name>** Specifies the name of the device class.
- <@comment>** Specifies a comment that describes this device class.

8.28.1.5 Policies

A policy controls what to do with data that fits that access pattern.

8.28.1.5.1 Creating a Policy

Create a policy.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_add_policy](#).

Syntax

```
EXEC dsam_add_policy <@name>, <@class>, <@compress>, <@comment>
```

- <@name>** Specifies the policy's name.
- <@class>** Specifies the device class to which objects matching this policy should belong.
 - <@compress>** Is the compression level to apply to objects matching this policy('None', 'Default', 'Page', 'Row').
 - <@comment>** Describes this policy.

<@compress> Specifies the compression level to apply to objects matching this policy. NULL indicates that objects matching this policy should keep the compression level they currently have. Valid compression levels are:

- 'None' – does not compress data.
- 'Default' – compression data that is currently set to remain unchanged.
- 'Page' – compress data at the page level.
- 'Row' – in addition to page-level compression, compress individual rows.

<@comment> Specifies a comment that describes this policy.

8.28.1.5.2 Deleting a Policy

Deleting a policy.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_delete_policy](#).

Syntax

```
EXEC dsam_delete_policy <@name>
```

8.28.1.5.3 Editing a Policy

Modifying a policy.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_update_policy](#).

Syntax

```
EXEC dsam_update_policy <@policy_id>, <@name>, <@class>, <@compress>, <@comment>
```

<@policy_id> Specifies the ID of the policy to be updated.

<@name> Specifies the unique name for the policy.

<@class> Specifies the device class that the policy should use. This parameter may have the special value ' ' (a blank) to indicate that the policy does not specify a device class.

<@compress> Specifies the compression level that the policy should use. Valid values are:

- 'None' – do not compress data.
- 'Default' – compression data which is currently set to remain unchanged.
- 'Page' – compress data at the page level.
- 'Row' – in addition to page level compression, compress individual rows.

`<@comment>` Specifies the comment describing this policy.

8.28.1.6 Access Groups

An access group manages a partition by reporting and managing the storage for that partition.

8.28.1.6.1 Creating an Access Group

Create an access group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_add_access_group](#).

Syntax

```
EXEC dsam_add_access_group <@grpname>, <@comment>
```

8.28.1.6.2 Deleting an Access Group

Delete an access group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_delete_access_group](#).

Syntax

```
EXEC dsam_delete_access_group <@grpname>
```

8.28.1.6.3 Editing an Access Group

Edit an access group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_update_access_group](#).

Syntax

```
EXEC dsam_update_access_group <@grp_id>, <@grpname>
```

8.28.1.6.4 Adding Access Tiers to an Access Group

Add an access tier to an access group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_add_access_tier](#).

Syntax

```
EXEC dsam_add_access_tier <@grpname>, <@mbrname>, <@time>, <@ct_lo>, <@ct_hi>,  
<@color>, <@comment>
```

- <@grpname>** Specifies the name of the access group that will contain the new tier.
- <@mbrname>** Specifies the name of the tier to be added.
- <@time>** Specifies the time interval to be used by this tier, given as a number of seconds.
- <@ct_lo>** Specifies the lower bound of the count range to be used by this tier. If specified as NULL, it indicates a lower limit of 0.
- <@ct_hi>** Specifies the upper bound of the count range to be used by this tier. If specified as NULL, it indicates infinity (i.e., no upper limit).
- <@color>** Specifies the color used when displaying data accesses matching this tier. It is an integer RGB value. If not specified, the default is 0 (black).
- <@comment>** Specifies a comment describing this access tier.

8.28.1.6.5 Deleting Access Tiers

Remove an access tier from an access group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_delete_access_tier](#).

Syntax

```
EXEC dsam_delete_access_tier <@grpname>, <@mbrname>
```

8.28.1.6.6 Editing Access Tiers

Edit an access tier.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_update_access_tier](#).

Syntax

```
EXEC dsam_update_access_tier <@mbr_id>, <@name>
```

8.28.1.6.7 Binding or Unbinding an Access Group Object

Bind or unbind an object (database or table) to an access group.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_track_object](#).

Syntax

```
EXEC dsam_track_object <@grpname>, <@set>, <database_name>
```

8.28.1.6.8 Viewing an Access Group from a Database

View or modify a database's access group from its properties page.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    DISTINCT c.name as 'AccessGroup'
FROM
    sybdsamdb.dbo.DSAMCustomizations a,
    sybdsamdb.dbo.DSAMCustomizations b,
    sybdsamdb.dbo.DSAMAccessGroups c,
    <database_name>.dbo.sysobjects o,
    <database_name>.dbo.sysusers u
WHERE
    a.tag = 'OB'
    AND b.tag = 'DB'
    AND a.infol = b.id
    AND a.id = c.group_id
    AND o.id = a.int_value
    AND o.uid = u.uid
    AND b.name = '<database_name>'
ORDER BY
    1
```

Output

AccessGroup

Default access_group

8.28.1.6.9 Viewing an Access Group from a Table

View or modify a table's access group from its properties page.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    group_id,
    name,
    comment
```

```
FROM
    sybdsamdb.dbo.DSAMAccessGroups
```

Output

group_id	name	comment
1	Default	The default DSAM access group
2	access_group	comment

8.28.1.7 Selecting the Table for DSAM to Display

Display DSAM data for a selected table.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    Name = t.name
FROM
    sybdsamdb.dbo.DSAMAccessTiers t,
    sybdsamdb.dbo.DSAMAccessGroups g
WHERE
    g.group_id = t.group_id
    AND g.name = '<grpname>'
```

Output

Name
Active
Less Active
Historical

8.28.1.8 Managing Partitions Based on Access Tiers

View a list of partitions based on access tiers, and manage their storage and compression.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [alter table](#).

Syntax

```
ALTER TABLE <table_name> MOVE PARTITION <table_name> TO
<destination_segment_name>
GO
ALTER TALE <table_name> MODIFY PARTITION <table_name> SET compression = page
GO
```

8.28.1.9 Editing the Policy Evaluation Time

You can view and edit the policy evaluation time.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [p_sjobcreate](#).

Syntax

```
EXEC sybmgmtdb.sp_sjobcreate @name = '<name>', @option = 'jname = '<jname>',
sname = '<sname>', server = '<server>', timeout = <timeout>'
```

8.28.1.10 Viewing Partitions That Do Not Meet Policy Conditions

View partitions that do not comply with the policy of their qualified access tier.

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT
    distinct id,
```

```

name,
device_class,
device_class_name,
compression,
compression_name,
comment
FROM
  sybdsamdb.dbo.DSAMPolicies
WHERE
  name = '<policy_name>'

```

Output

id	name	device_class	device_class_name	compression	compression_name	comment
2	GP, compression as-is	GP	General Purpose	1	Default	Normal storage, no change to compression

8.28.1.11 Manually Execute Policy Evaluation

You can immediately execute a policy for a partition or schedule a time to do so with Job Scheduler.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [dsam_ptn_eval](#).

Syntax

```
EXEC sybdsamdb.dsam_ptn_eval
```

8.28.1.12 Execute Policy Movement with Job Scheduler

You can use Job Scheduler to set a date and time to evaluate partitions.

The following syntax provides basic guidance for this task. For the full syntax and required permissions, see [p_sjobcreate](#).

Syntax

```
EXEC sp_sjobcreate @name = '<sjname>', @option = 'jname = '<jname>',  
sname = '<sname>', server = '<server>', timeout = <timeout>'
```

8.28.113 Batch Execute Policy Movement

The following syntax provides basic guidance for viewing this information.

Syntax

```
SELECT  
    partition_name = case  
        WHEN p.indid = 1  
        THEN p.cdaptptnname  
        ELSE p.name END,  
    partiton_id = p.id,  
    segment_name = s.name,  
    partition_type = (SELECT v.name  
        FROM  
            master.dbo.spt_values v  
        WHERE  
            v.type = 'PN' AND v.number = isnull(i.partitiontype,  
            (SELECT number  
                FROM  
                    master.dbo.spt_values  
                WHERE  
                    name = 'roundrobin'))),  
    partition_id = isnull(p.partitionid, 0),  
    i.conditionid,  
    p.crdate,  
    space_used = data_pages(db_id, table_id, 0, p.partitionid),  
    compression_status = p.status  
FROM  
    <database_name>.dbo.syspartitions p,  
    <database_name>.dbo.sysindexes i,  
    <database_name>.dbo.syssegments s  
WHERE  
    i.id = p.id  
    AND i.id = table_id  
    AND p.indid < 2  
    AND i.indid = p.indid  
    AND s.segment = p.segment  
    AND p.partitionid = partition_id  
ORDER BY  
    1
```

Output

partition_name	partiton_id	segment_name	partition_type	partition_id
ASE_ALERT_HISTORY _640002280	640002280	system	roundrobin	640002280

conditionid	crdate	space_used	compression_status
(Null)	2020-03-16 02:37:32.003	0	2

9 Capturing, Analyzing, and Replaying Workloads

The SAP ASE workload analyzer option allows you to capture, analyze, and replay a production workload nondisruptively. You can then utilize the captured workload to diagnose problems, and understand and manage configuration changes proactively.

SAP ASE workload analyzer also allows you to run replays of captured workloads to measure and analyze application performance under different conditions.

Use the SAP ASE workload analyzer option to:

- Identify problematic queries, such as queries with a long response time due to missing indexes.
- Identify client activity patterns, such as the number of requests per IP address.
- Measure the performance of captured workloads in different server configurations.
- Compare query and overall workload performance between different server configurations.
- Evaluate database upgrades and understand benefits from new options.
- Diagnose product problems by replaying functionality in a controlled environment.

i Note

Whenever you use this option, make sure to start both SAP ASE and SAP ASE Cockpit server using the same operating system login. SAP ASE Cockpit server must be able to read the PCAP files that SAP ASE generates. SAP ASE Cockpit server creates the output directories for the PCAP files and SAP ASE must be able to write to those directories. Because permission to access to the directories and files are controlled by the operating system login that creates them, SAP ASE Cockpit server cannot perform a workload analysis unless it is running under the same operating system login as the SAP ASE server.

For more information on capturing, analyzing, and replaying workloads, see [Workload Analyzer Command Line Utility](#).

10 Collect Diagnostic Data

Collect comprehensive SAP ASE configuration and environment data for diagnostic analysis. SAP Technical Support uses this information to diagnose server issues, thus expediting customer cases.

Use [sybdiag](#) to collect the data.

11 Troubleshoot SAP ASE Cockpit Server

Solve problems that occur in SAP ASE Cockpit server.

11.1 Authentication and Access Problems

Troubleshoot problems that involve log in, starting, stopping, and authentication.

11.1.1 Cockpit Server Fails to Start

Problem: SAP ASE Cockpit server does not start.

Solution 1: Port conflict

Solution: SAP ASE Cockpit server might use a port that is used by another server or application. To check for port conflicts:

1. Execute this command:

```
cockpit --info ports
```

The command lists all the ports on which SAP ASE Cockpit server and its services listen, indicates whether each port is in use, and shows the service running on each port. If the SAP ASE Cockpit server is not running, any port shown to be in use represents a conflict.

2. If you discover a conflict, use `cockpit --port` to change the port used by the SAP ASE Cockpit service.

Solution 2: Insufficient memory

Increase the maximum memory setting if you see this error when you try to start: `Could not create the Java Virtual machine.`

11.1.2 SAP Adaptive Server and Replication Server Fails to Start

(Linux platform only) If the environment variable `LANG` is set to `POSIX`, the start up script for both SAP Adaptive Server and the Replication Server, which is invoked by the cockpit, will fail to run.

To resolve this issue, set the `LANG` environment explicitly in the start up script file to `en_US.UTF-8`.

For example, add the following lines to the `RUN_<myServer>.sh` start up script:

```
LANG=en_US.UTF-8
export LANG
```

For more information, see SAP Note [2022985](#) - SYB: Start of SAP ASE fails because variable `LANG` is not set correctly.

11.1.3 Authentication Removed After Host Name Change

Changing the SAP ASE host name in the interfaces file to a different name removes the authentication and connection information for the SAP ASE cockpit.

When the SAP ASE cockpit is started, the first query entry of the interfaces is used to establish the host name and port number. If you change the host name in the interfaces file, the SAP ASE server is treated as a new server when the SAP ASE cockpit is restarted.

For example, the following information in the interfaces file is used to establish the host name and port number:

- `master=NLWNSCK,host1,5000`
- `query=NLWNSCK,host1,5000`

The new entry's host name is `localhost`.

- `master=NLWNSCK,localhost,5000`
- `query=NLWNSCK,localhost,5000`
- `master=NLWNSCK,host1,5000`
- `query=NLWNSCK,host1,5000`

Once the host name is changed, both the ASE Agent and Replication Management Agent (RMA), must be manually registered. SAP also recommends that you change the host name in ASE Agent Plug-in configuration file (`agent-plugin.xml`).

11.2 Performance Problems

Troubleshoot problems that involve performance and memory errors.

11.2.1 Out of Memory Errors

Problem: SAP ASE Cockpit server generates `OutOfMemory` errors and might fail to start.

Solution:

- If the `OutOfMemory` error says that SAP ASE Cockpit server is out of heap space, increase the maximum memory setting (`<COCKPIT_MEM_MAX>` or `jvmopt=-Xmx`).
- If the `OutOfMemory` error says that SAP ASE Cockpit server is out of permanent generation space, increase the permanent memory setting (`<COCKPIT_MEM_PERM>` or `jvmopt=-XX:MaxPermSize`).
- Repeated `OutOfMemory` errors may indicate a memory leak. `OutOfMemory` errors generate heap dumps. Heap dump files have a file extension of `.hprof`, in the `COCKPIT-4\log` directory.

Send the heap dump files to Technical Support for analysis.

11.2.2 Memory Warnings at Startup

Problem: When SAP ASE Cockpit server starts, you see warnings about system memory or heap memory allocation.



Solution: Increase the maximum memory setting (`<SCC_MEM_MAX>` or `jvmopt=-Xmx`).

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2022 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.