

SAP Business Objects

文書バージョン: 4.0 Support Package 10 - 2014-07-24

データフェデレーション管理ツールガイド



目次

1	データフェデレーションサービスの管理またはチューニングについて.....	5
1.1	データフェデレーションサービスの管理またはチューニングについて.....	5
2	データフェデレーション管理ツールの使用.....	6
2.1	データフェデレーション管理ツールの目的.....	6
2.2	データフェデレーション管理ツールの起動.....	6
2.3	データフェデレーション管理ツールへの管理者権限を持つユーザの追加.....	7
2.4	Microsoft Active Directory 認証用にデータフェデレーション管理ツールを設定.....	7
2.5	データフェデレーション管理ツールセッションからのログアウト.....	7
2.6	データフェデレーションクエリエンジンでのクエリ実行の検証.....	7
2.7	データフェデレーションクエリサーバでの SQL クエリのテスト.....	8
2.8	データフェデレーションクエリエンジンによるクエリプラン方法の表示.....	8
2.9	データフェデレーションクエリサーバ上で実行されたクエリの履歴の参照.....	9
2.10	メタデータのクエリ.....	9
2.11	クエリのキャンセル.....	9
2.11.1	クエリのキャンセル.....	9
2.12	データフェデレーション管理ツールのクエリパネルタブ.....	10
2.13	データフェデレーション管理ツールのクエリの監視タブ.....	12
2.14	データフェデレーション管理ツールのシステムパラメータタブ.....	14
2.15	データフェデレーション管理ツールのコネクタ設定タブ.....	15
2.16	データフェデレーション管理ツールの統計情報タブ.....	17
2.17	データフェデレーション管理ツールでの [プロパティ] ビューの表示.....	19
2.18	データフェデレーション管理ツールから SSL 用に設定されたサーバに接続.....	19
3	クエリの最適化.....	20
3.1	データフェデレーションクエリ のパフォーマンス調整.....	20
3.2	システムパラメータを使用したメモリ使用の最適化.....	20
3.2.1	メモリを消費する演算子.....	22
3.3	クエリソースの最適なアルゴリズムがアプリケーションによって選択されるための統計情報の使用.....	22
3.3.1	列カーディナリティについて.....	22
3.3.2	列間の関係のファンアウト値について.....	23
3.3.3	記録された統計値をフィルタ処理し、必要なものだけを計算してレポートを最適化.....	23
3.4	クエリプランの最適化.....	24
3.4.1	データフェデレーション管理ツールの [クエリプラン] ビュー.....	24
3.4.2	[統計情報の説明] コマンド.....	25
3.4.3	フィードバックを得てクエリを調整するための、クエリの説明機能の使用.....	26
3.4.4	データフェデレーション管理ツールを使用した演算子がプッシュされているかどうかの確認.....	27

3.4.5	大規模テーブルに結合された小規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン	27
3.4.6	並べ替えることができるデータを含む大規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン	29
3.4.7	順序ベース演算子の有効化を制御するシステムパラメータの使用	31
3.4.8	データソースサブクエリの強制並列実行	31
3.4.9	準結合の実行ストラテジー	32
3.5	特定のコネクタの最適化	32
3.5.1	SAP BW へのパラレルクエリに関するコールバックの同時実行の増加	32
3.5.2	SAP BW に対するクエリからの応答パッケージのサイズ変更	33
3.6	データフェデレーションサービス用に作成した最適化設定の昇格	33
4	データのソースへのコネクタの設定	34
4.1	データフェデレーション管理ツールのコネクタの情報の表示	34
4.2	データフェデレーション管理ツールのコネクタのプロパティの変更	34
4.3	リレーショナルデータソースのコネクタの設定	34
4.3.1	リレーショナルデータソースの共通コネクタプロパティの一覧	34
4.3.2	MySQL データソースのコネクタプロパティの一覧	37
4.3.3	Teradata データソースのコネクタプロパティの一覧	37
4.3.4	Sybase ASE データソースのコネクタプロパティの一覧	38
4.3.5	SQL Server データソースのコネクタプロパティの一覧	38
4.3.6	汎用 ODBC または JDBC データソースのコネクタプロパティの一覧	39
4.3.7	Oracle データソースのコネクタプロパティの一覧	41
4.3.8	SAP HANA データソースのコネクタプロパティの一覧	41
4.3.9	MaxDB データソースのコネクタプロパティの一覧	41
4.4	SAS へのコネクタの設定	42
4.4.1	SAS データソースのコネクタプロパティの一覧	42
4.4.2	カーディナリティを基準にして FROM 句内のテーブルの順序を並べ替えることによる SAS クエリの最適化	48
4.5	SAP BW のコネクタの設定	49
4.5.1	SAP BW データソースのコネクタプロパティの一覧	49
4.5.2	SAP BW がデータフェデレーションサービスとの接続に使用するコールバック ID を手動で設定	52
4.5.3	SAP BW 接続のコールバックの ID の削除	53
4.5.4	SAP 分析権限を活用したデータの自動フィルタ	54
4.5.5	複数のソースユニバースでの SAP BW 接続のアーキテクチャ	57
4.5.6	複数のソースユニバースでの SAP BW 接続のコールバックシーケンス	58
4.6	データフェデレーション管理ツールを使用したリレーショナルコネクタ機能および SAS コネクタ機能の設定	58
4.7	リレーショナルデータソースに対応したコネクタの capabilities の全一覧	59

5	システムパラメータとセッションパラメータの管理.....	62
5.1	システムパラメータとセッションパラメータの概要.....	62
5.2	データフェデレーション管理ツールを使用したシステムパラメータの変更.....	62
5.3	データフェデレーション管理ツールを使用したセッションパラメータの変更.....	62
5.4	データフェデレーション管理ツールを使用したリレーショナルコネクタ機能および SAS コネクタ機能の設定	63
5.5	システムパラメータリスト.....	63
5.6	セッションパラメータリスト.....	75
5.7	データフェデレーションアプリケーション内での照合.....	76
5.7.1	データフェデレーションアプリケーション内でサポートされている照合.....	77
5.7.2	データフェデレーション SQL クエリの文字列並び替えおよび文字列比較動作の設定.....	78
5.7.3	バイナリ照合の使用時にデータフェデレーションアプリケーションがクエリをソースへどのようにプッシュ するのかを決定する方法.....	80
6	SQL 構文リファレンス.....	81
6.1	データフェデレーションクエリエンジンのクエリ言語.....	81
6.1.1	識別子と命名規則.....	81
6.1.2	データフェデレーションクエリエンジンで使用するデータ型.....	83
6.1.3	文.....	87
6.1.4	式.....	88
6.1.5	コメント.....	91
6.2	SELECT 句の文法.....	91
7	用語集.....	98
7.1	用語と説明.....	98
8	トラブルシューティング.....	99
8.1	データフェデレーションサービスのログギングについて.....	99
8.2	SAP BW データソースで、実行時間が長いクエリで接続が閉じられる.....	99
8.3	SAP BW コネクタのエラー NoClassDefFoundError: CpicDriver.....	99

1 データフェデレーションサービスの管理またはチューニングについて

1.1 データフェデレーションサービスの管理またはチューニングについて

データフェデレーションサービスの管理またはチューニングには、データフェデレーション管理ツールを使用します。

管理

サービスによるデータ処理方法に固有のデータフェデレーションサービスのアспектを管理する必要があるときには、データフェデレーション管理ツールを使用します。これらのアспектには、特定データソースへのコネクタのプロパティの管理や、メモリの設定、データフェデレーションクエリエンジンのクエリに影響するパラメータの設定が含まれています。

データフェデレーション管理ツールでは、コネクタの参照と管理、データソースの参照とデータソースに対するクエリの実行、統計情報の管理、および過去のクエリと実行中のクエリの一覧を表示できます。本稼動システムのレポーティングアプリケーションではクエリが生成され、人の手を介さずにクエリサーバへ送信されるため、過去のクエリまたは実行中のクエリの一覧を照会が必要なことがあります。生成されたクエリを表示することで、システムが期待どおりに実行しているかどうかを確認できます。

ユーザアカウントやログインの管理などの一般管理では、データフェデレーションサービスがインストールされているプラットフォームのツールを使用します。

チューニング

データソースのデータにコネクタまたはクエリを適応させるには、チューニング用にデータフェデレーション管理ツールを使用します。

チューニングには、できるだけ多くの処理を各データソースへパスするための各コネクタの機能設定、各データソースの適切な統計情報の設定、およびサーバへ送信される各クエリを最適化するためのパラメータの設定が含まれます。一般的に最適化とは、データソースでの処理をできるだけ多くし、ネットワークを介したデータ送信をできるだけ少なくすることです。データフェデレーションサービスには、ソースへの処理のプッシュとデータ転送の削減のためのさまざまなオプションや、システムでのクエリ処理状況を把握するためのツールがあります。

2 データフェデレーション管理ツールの使用

2.1 データフェデレーション管理ツールの目的

データフェデレーション管理ツールは、データフェデレーションサービスを管理するための使いやすい機能を提供するリッチクライアントアプリケーションです。

SAP BusinessObjects Enterprise プラットフォームと緊密に統合されているデータフェデレーションサービスは、異なるデータソースにクエリを分散することにより複数のソースユニバースの利用を可能にし、単一のデータファンデーションを通してデータを連合させることができます。

データフェデレーション管理ツールを使用してデータフェデレーションクエリを最適化し、最大限のパフォーマンスを発揮できるようデータフェデレーションクエリエンジンを微調整できます。

以下を実行するためにデータフェデレーション管理ツールを使用します。

- SQL クエリをテストする。
- 連合されたクエリの各ソースへの分散方法の詳細を規定する最適化計画を視覚化します。
- 最高のパフォーマンスを達成するよう、統計情報を計算し、データフェデレーションサービスを微調整するシステムパラメータを設定します。
- クエリがコネクタレベルの各データソースでどのように実行されるかを制御するプロパティを管理します。
- SQL クエリの実行の監視
- 実行されたクエリの履歴を参照する。

関連情報

[データフェデレーションクエリサーバでの SQL クエリのテスト](#) [8 ページ]

[データフェデレーションクエリエンジンによるクエリプラン方法の表示](#) [8 ページ]

[データフェデレーションクエリサーバ上で実行されたクエリの履歴の参照](#) [9 ページ]

[データフェデレーション管理ツールのコネクタの情報の表示](#) [34 ページ]

[クエリソースの最適なアルゴリズムがアプリケーションによって選択されるための統計情報の使用](#) [22 ページ]

2.2 データフェデレーション管理ツールの起動

1. **スタート** > **プログラム** > *BusinessObjects Data Federator XI Release 4* > **データフェデレーション管理ツール** の順にクリックします。
2. システム名、ユーザ名およびパスワードを入力し、[OK] をクリックします。

2.3 データフェデレーション管理ツールへの管理者権限を持つユーザの追加

SAP BusinessObjects Enterprise サーバでは、データフェデレーション管理者と呼ばれるユーザグループが、データフェデレーションサービスの管理権限を持っています。

グループへのユーザの追加の詳細については、*SAP BusinessObjects Enterprise* 管理者ガイドを参照してください。

2.4 Microsoft Active Directory 認証用にデータフェデレーション管理ツールを設定

Active Directory 認証用にデータフェデレーション管理ツールを設定するには、データフェデレーション管理ツールの初期化ファイルを編集する必要があります。このファイルには、ログイン設定ファイルと Kerberos 設定ファイルの 2 つの設定ファイルを指定する必要があります。

1. `<install_dir>\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\win32_x86\DFAdministrationTool.ini` ファイルを編集します。

ファイルの末尾に以下の行を追加します。

```
-Djava.security.auth.login.config=<path-to-bsclogin>\bscLogin.conf  
-Djava.security.krb5.conf=<path-to-kerberos>\krb5.ini
```

例

```
-Djava.security.auth.login.config=C:\WINNT\bscLogin.conf  
-Djava.security.krb5.conf=C:\WINNT\krb5.ini
```

2. `bscLogin.conf` および `krb5.ini` の 2 つのファイルを Kerberos を使用した Active Directory 認証用に設定します。
詳細については、*SAP BusinessObjects Enterprise* 管理者ガイドの Windows AD のための Kerberos 認証に関する章を参照してください。

2.5 データフェデレーション管理ツールセッションからのログアウト

ツールバーの左側にある [ログアウト](#) ボタンをクリックします。

2.6 データフェデレーションクエリエンジンでのクエリ実行の検証

1. データフェデレーション管理ツールを起動します。

2. [\[クエリの監視\]](#) タブをクリックします。
3. [\[最新表示\]](#) をクリックします。
[\[クエリの実行中\]](#) ペインに実行中のクエリが表示されます。

2.7 データフェデレーションクエリサーバでの SQL クエリのテスト

1. データフェデレーション管理ツールを起動します。
2. [\[クエリパネル\]](#) タブをクリックします。
3. クエリを入力します。
4. [\[実行\]](#) をクリックし、クエリを実行します。
クエリが実行され、[\[クエリ結果\]](#) パネルに結果が表示されます。

2.8 データフェデレーションクエリエンジンによるクエリプラン方法の表示

データフェデレーションクエリエンジンは、使用中の SQL クエリを分析し、複数のソースから正確なデータをできるだけ早く取得できるようにそれらのクエリを変換する方法を決定します。この分析を実行するために、クエリエンジンは複数のデータソースに対してできるだけ多くの作業を配布し、最終的な結果を作成する目的で必要とされる、ネットワークを経由して取得するデータができるだけ少なくなるようにサブクエリを作成します。

説明ツールを使用して、クエリが複数のソースの間で配布される方法を表示できます。

1. データフェデレーション管理ツールを起動します。
2. [\[クエリパネル\]](#) タブをクリックします。
3. 表示するクエリを入力します。
4. [\[実行\]](#) の横にある矢印をクリックし、[\[クエリの説明\]](#) をクリックします。

使用中のクエリは、クエリエンジンが生成したプランという形で表示されます。

関連情報

[データフェデレーション管理ツールの \[クエリプラン\] ビュー \[24 ページ\]](#)

2.9 データフェデレーションクエリサーバ上で実行されたクエリの履歴の参照

ユーザまたはユーザのアプリケーションがすでにデータフェデレーションクエリサーバにクエリを送信した場合は、データフェデレーション管理ツールを使用してこれらのクエリからなるリストを表示できます。

1. データフェデレーション管理ツールを起動します。
2. [\[クエリの監視\]](#) タブをクリックします。
[\[実行済みクエリ\]](#) 枠には、すでに実行されたクエリが表示されます。

2.10 メタデータのクエリ

特定のテーブルセットで機能するようにハードコーディングされていない動的なアプリケーションは、接続先の任意のデータベースにあるオブジェクトの構造および属性を特定するためのメカニズムを備えている必要があります。このようなアプリケーションでは、以下のような情報が必要になることがあります。

- ターゲットおよびデータソース内のテーブルの数と名前
- テーブル内の列の数、および各列の名前、データ型、スケール、精度
- テーブルで定義されているキー

データフェデレーションクエリエンジンを基盤とするアプリケーションは、以下のストアードプロシージャを使用してシステムカタログ内の情報にアクセスできます。

```
CALL getTables '<name-of-catalog>', '%', '%'  
CALL getColumns '<name-of-catalog>', '<name-of-schema>', '<name-of-table>', '%'  
CALL getKeys '<name-of-catalog>', '<name-of-schema>', '<name-of-table>'
```

2.11 クエリのキャンセル

データフェデレーションの使用時には、実行中のすべてのクエリ、または実行中の特定のクエリをコマンドでキャンセルできます。

キャンセルコマンドは非同期です。そのため、クエリをキャンセルすると、クライアントアプリケーションではそのクエリがキャンセル済であると表示されても、データフェデレーションクエリエンジンではそのキャンセルがまだ完了していないことがあります。

2.11.1 クエリのキャンセル

1. [\[クエリの監視\]](#) タブをクリックします。

2. キャンセルするクエリを右クリックします。
3. [\[キャンセル\]](#) をクリックします。

2.12 データフェデレーション管理ツールのクエリパネルタブ

パネル

パネル	説明
SQL テキスト	<p>SQL クエリを入力できます。</p> <p>要素をクエリに挿入するには、ダブルクリックするか、[カタログ]、[演算子]、[関数] パネルからドラッグアンドドロップします。</p> <p>コントロール</p> <ul style="list-style-type: none">• 最大行数: 取得する最大行数です。• 列の合計数を表示する: すべての行を取得しなかった場合でも結果に行の合計数を表示するかどうかを指定します。
カタログ	データフェデレーションサービスに存在するすべてのカタログを表示します。
演算子	使用可能な演算子の一覧を表示します。
関数	使用可能な関数のカテゴリごとの一覧を表示します。
クエリ結果	クエリ結果のコンテナです。 [実行] をクリックしてクエリを実行すると表示されます。
未処理データ	最後に実行されたクエリの未処理データ結果を表示します。 [実行] をクリックすると表示されます。
自動チャート	クエリ結果の簡単な円グラフ表示です。 [実行] または [実行] をクリックすると表示されます。
クエリプラン	<p>クエリを実行することなくクエリプランを表示します。[クエリの説明] をクリックすると表示されます。</p> <p>2 つの内部パネルが含まれます。</p> <ul style="list-style-type: none">• プラン: プラン構造をツリービューで表示します。• 詳細: [プラン] パネルで選択したノードに関する詳細情報を表示します。 <p>選択したノードについての詳細は、[プロパティ] ビューに表示されます。</p>
クエリ統計情報	現在のクエリの影響を受けた要素と統計情報が表示されます。 [統計情報の説明] をクリックすると表示されます。

ボタン

ボタンのラベル	説明
実行	<p>メニュー項目のプルダウンボタン</p> <ul style="list-style-type: none"> デフォルトアクション: [SQL テキスト] パネルに現在あるクエリを実行します。 [クエリの実行] アクション: デフォルトアクションと同じです。 [クエリの説明] アクション: クエリプランを説明します。 [統計情報の説明] アクション: 影響を受けた要素と統計情報を表示し、ユーザはカーディナリティを更新できます。 <p>結果は、[クエリ結果] パネルに表示されます。</p>
最後の変更を元に戻す	SQL テキストパネルでの最後の変更を元に戻します。
最後の変更をやり直す	[SQL テキスト] パネルでの最後の変更をやり直します。
カタログの最新表示	[カタログ] パネルを最新表示します。
カタログの表示/非表示	[カタログ] パネルを表示または非表示にします。
演算子の表示/非表示	[演算子] パネルを表示または非表示にします。
関数の表示/非表示	[関数] パネルを表示または非表示にします。
ソースクエリのみを表示	中間ノードをフィルタすることにより、ソースクエリノードのみを表示します。

コンテキストメニュー

メニュー項目	説明
計算	<p>サブメニュー項目のプルダウンメニュー</p> <ul style="list-style-type: none"> 選択項目のみ 選択したノードのみを計算します。 選択項目および子 このクエリコンテキスト内の選択されたノードとその子を計算します。 未計算のみ (子を含む) [現在のカーディナリティ] 列が不明の場合に選択項目とその子のみを計算します。
現在のカーディナリティ	<p>サブメニュー項目のプルダウンメニュー</p> <ul style="list-style-type: none"> ユーザのカーディナリティを使用 選択したオブジェクトにおいて、クエリの最適化にユーザが設定したカーディナリティを使用するように、データフェデレーションサービスに強制します。このア

メニュー項目	説明
	<p>クションは、テーブルまたは列のみを選択したときに有効です。アクション完了後、現在のカーディナリティは [ユーザのカーディナリティ] と同一になります。</p> <ul style="list-style-type: none"> ソースのカーディナリティを使用 選択したオブジェクトにおいて、クエリの最適化にデータソースから取得したカーディナリティを使用するように、データフェデレーションサービスに強制します。このアクションは、テーブルまたは列のみを選択したときに有効です。アクション完了後、現在のカーディナリティは [ソースからのカーディナリティ] と同一になります。

関連情報

[データフェデレーション管理ツールの \[クエリプラン\] ビュー \[24 ページ\]](#)

[\[統計情報の説明\] コマンド \[25 ページ\]](#)

2.13 データフェデレーション管理ツールのクエリの監視タブ

テーブル

列名	説明
クエリ	<p>クエリまたはサブクエリの ID</p> <p>異なるアイコン</p> <ul style="list-style-type: none"> 実行中アイコン: クエリが実行されています。 成功し終了アイコン: クエリが成功して終了しています。 失敗し終了アイコン: クエリが失敗して終了しています。プロパティビューを使用して例外を表示できます。
開始時刻	実行開始時刻
終了時刻	実行終了時刻
実行時間	実行の開始時刻と終了時刻との間の経過時間
行	クエリによって抽出された行数
ステータス	<p>クエリステータス</p> <ul style="list-style-type: none"> 分析中: クエリがフェデレーションエンジンによって分析されています。 実行中: クエリがフェデレーションエンジンによって実行されています。

列名	説明
	<ul style="list-style-type: none"> クローズ済み: 例外が発生した、または発生せずにクエリが閉じられました。
サーバ名	クエリを処理するサーバ名
ユーザ名	クエリを起動するサーバ名
SQL テキスト	クエリの SQL テキスト

フィルタ

フィルタのラベル	説明
フィルタ	使用できる列のテキストに対するフィルタ
ステータス	クエリステータスに対するフィルタ <ul style="list-style-type: none"> すべてのクエリ 実行中のクエリ 実行済みクエリ
型	クエリタイプに対するフィルタ <ul style="list-style-type: none"> すべてのクエリ SQL コマンド プロシージャ
接続	コネクションに対するフィルタ <ul style="list-style-type: none"> すべての接続 現在の接続: 現在のデータフェデレーション管理ツール接続のクエリのみを表示します。

ボタン

ボタンのツールヒント	説明
監視情報を XML として保存する	監視情報を XML として保存します。
最新表示	サーバから最新の監視情報を取得します。

2.14 データフェデレーション管理ツールのシステムパラメータタブ

タブ

[システムパラメータ] タブを使用して、システムパラメータ、セッションパラメータ、およびプロパティを管理できます。

タブラベル	説明
システムパラメータ	<p>システムパラメータを管理します。</p> <p>列</p> <ul style="list-style-type: none">パラメータ: パラメータの名前。現在の値: 現在のパラメータの値。ここに新しい値を入力できます。読み取り専用パラメータは、背景が灰色になっています。デフォルト値: システム起動時にパラメータに設定されていた値。現在の値を変更し、値を元に戻す場合、この値を参照として使用できます。カテゴリ: パラメータのカテゴリ。説明: パラメータの説明。『データフェデレーション管理ツールガイド』で、パラメータの完全な一覧とその説明を参照することもできます。
セッションパラメータ	<p>セッションパラメータを管理します。</p> <p>列</p> <ul style="list-style-type: none">Parameter: パラメータの名前。現在の値: 現在のパラメータの値。ここに新しい値を入力できます。説明: パラメータの説明。データフェデレーション管理ツールガイドで、パラメータの完全な一覧とその説明を参照することもできます。
システムプロパティ	<p>システムプロパティを表示します。</p> <p>列</p> <ul style="list-style-type: none">Parameter: パラメータの名前。現在の値: パラメータの現在の値。
スタートアップパラメータ	<p>スタートアップパラメータを表示します。</p> <p>列</p> <ul style="list-style-type: none">Parameter: パラメータの名前。現在の値: パラメータの現在の値。
インストールパラメータ	<p>インストールパラメータを表示します。</p> <p>列</p> <ul style="list-style-type: none">コンポーネント: パラメータのコンポーネント名。Parameter: パラメータの名前。

タブラベル	説明
	<ul style="list-style-type: none"> ● 現在の値: パラメータの現在の値。 ● デフォルト値: システム起動時にパラメータに設定されていた値。 ● 元のパラメータ: パラメータの値の由来。以下のいずれかになります。 <i>ORIGIN_DEFAULT</i>、<i>ORIGIN_SERVER_PROPERTIES</i>、<i>ORIGIN_SYSTEM_PROPERTIES</i>

コンテキストメニューの表示

メニュー項目	説明
システムパラメータおよびセッションパラメータ	システムパラメータおよびセッションパラメータのみを表示します。
すべてのパラメータ	すべてのタブを表示します。

関連情報

[システムパラメータリスト](#) [63 ページ]

2.15 データフェデレーション管理ツールのコネクタ設定タブ

パネル

パネル	説明
[コネクタ] ツリー	<p>コネクタの一覧を表示します。</p> <p>コネクタの設定を表示するには、コネクタツリー内のコネクタをダブルクリックします。</p>
[一般情報] タブ	<p>現在のコネクタの一般情報を表示します。</p> <p>コネクタの一般情報を表示するには、コネクタツリー内のコネクタをダブルクリックします。</p>
[機能] タブ	<p>現在のコネクタの機能を表示します。</p> <p>コネクタの機能を表示するには、コネクタをダブルクリックして、機能タブを選択します。</p>
[設定プロパティ] タブ	現在のコネクタの設定プロパティを表示します。

パネル	説明
	コネクタの設定プロパティを表示するには、コネクタを展開し、 設定 ノードをダブルクリックします。

ボタン

ボタンのラベル	説明
すべて折りたたむ	コネクタリストを折りたたみます。
すべて展開	コネクタリストを展開します。
検索バーの表示/非表示	検索バーの表示/非表示を切り替えます。 これは、コネクタを名前で検索するときに使用できます。
最新表示	コネクタリストを最新表示します。
保存	設定プロパティを保存します。

コンテキストメニュー

メニュー項目	説明
設定の作成	コネクタの新しい設定を作成します。 設定プロパティに新しい値を設定できます。
設定の編集	コネクタの設定を編集できます。 コネクタをダブルクリックしても編集できます。
設定の削除	コネクタの設定を削除します。 設定を削除すると、デフォルト値が使用されます。

2.16 データフェデレーション管理ツールの統計情報タブ

テーブル

列名	説明
カタログ	次の内容が含まれます。 <ul style="list-style-type: none">• カタログ名• スキーマ名• テーブル名• 列名• エラーの説明• 待機メッセージ オブジェクトがテーブルまたは列である場合のみ、その兄弟列に値が表示されます。
最終計算日	オブジェクト上で計算アクションが行われた最新の日付。計算アクションが行われていない場合は [計算されていません] と表示されます。
リクエスト数	データフェデレーションサービスで、オブジェクトに対して実行されたクエリの数。クエリが実行されていない場合は [キャッシュされたレコードがありません] と表示されます。
現在のカーディナリティ	クエリプランを最適化するためにデータフェデレーションサービスで現在使用されているカーディナリティ。ない場合は [不明] と表示されます。
ソースからのカーディナリティ	オブジェクトに対して計算アクションが実行された後、データソースから返されるカーディナリティ。ない場合は [不明] と表示されます。
ユーザのカーディナリティ	データフェデレーションサービスでクエリプランの最適化に強制的に使用される、ユーザが指定したカーディナリティ。ない場合は [設定解除] と表示されます。 この列は編集可能です。値を編集するには、セルをクリックして整数を入力した後、Return キーを押すか、または任意の場所をクリックします。編集内容を破棄するには、Esc キーを押します。
- すべての列	ある項目においてアクション（計算や最新表示など）が実行されている場合、その項目は斜体で表示されます。

ボタン

ボタンのラベル	説明
最新表示	データフェデレーションサービスから現在表示されているすべてのデータを更新します。

ボタンのラベル	説明
	表示されるオブジェクトが多数の場合、このアクションは時間がかかることがあります。最新表示の操作に時間がかかる場合、このアクションをバックグラウンドで実行するときは、 進捗状況 ビューで進捗状況を監視することができます。
計算	<p>データフェデレーションサービスに対し、選択したオブジェクトが属するデータソースから、そのオブジェクトのカーディナリティを取得するように依頼します。アクション完了後、ユーザのカーディナリティ および 最終計算日 は更新され、現在のカーディナリティ は ユーザのカーディナリティ の値に設定されます。</p> <p>このアクションは、選択したオブジェクトにテーブルまたは列が含まれている場合のみ有効です。</p>

コンテキストメニュー

メニュー項目	説明
計算	<p>データフェデレーションサービスに対し、選択したオブジェクトが属するデータソースから、そのオブジェクトのカーディナリティを取得するように依頼します。アクション完了後、ユーザのカーディナリティ および 最終計算日 は更新され、現在のカーディナリティ は ユーザのカーディナリティ の値に設定されます。</p> <p>このアクションは、選択したオブジェクトにテーブルまたは列が含まれている場合のみ有効です。</p>
ユーザのカーディナリティを使用	選択したオブジェクトにおいて、クエリの最適化にはユーザが設定したカーディナリティを使用するように、データフェデレーションサービスに強制します。このアクションは、選択したオブジェクトにテーブルまたは列が含まれている場合のみ有効です。アクション完了後、現在のカーディナリティは ユーザのカーディナリティ と同一になります。
ソースのカーディナリティを使用	選択したオブジェクトにおいて、クエリの最適化にはデータソースから取得したカーディナリティを使用するように、データフェデレーションサービスに強制します。このアクションは、選択したオブジェクトにテーブルまたは列が含まれている場合のみ有効です。アクション完了後、現在のカーディナリティは ソースからのカーディナリティ と同一になります。

フィルタ

列	説明
カタログ	選択したすべてのカタログにフィルタを作成します。 [OK] をクリックすると、選択したカタログが検証され、他の場所をクリックすると、選択が破棄されます。
- その他のすべてのフィルタ	<p>これらのフィルタを使用すると、表示されたテーブルおよび列にフィルタを適用できます。</p> <p>テーブルがフィルタ条件を満たさない場合でも、テーブル内のいずれかの列がすべてのフィルタ条件を満たしていれば、そのテーブルは表示されます。</p>

列	説明
	これらのフィルタによって、カタログやスキーマを非表示にすることはできません。カタログおよびスキーマは、それらのオブジェクトがフィルタ条件を満たさない場合であっても表示されます。カタログをすべて非表示にするには、 [カタログ] フィルタを使用します。

関連情報

[記録された統計値をフィルタ処理し、必要なものだけを計算してレポートを最適化](#) [23 ページ]

2.17 データフェデレーション管理ツールでの **[プロパティ]** ビューの表示

データフェデレーション管理ツール内の **[プロパティ]** ビュー では、インタフェースのさまざまな要素に関する補助情報が表示されます。

► **ウィンドウ** ► **その他** ► **管理** ► **プロパティ** ► をクリックします。

2.18 データフェデレーション管理ツールから **SSL** 用に設定されたサーバに接続

多くの場合は、SSL 用のサーバ設定に対してデータフェデレーション管理ツールを接続して、セキュリティを上げる必要があります。

1. **<boe-install-dir>/win32_x86** ディレクトリの `DFAdminstrationTool.ini` ファイルを編集します。
2. 以下の JVM 引数を追加します。

```
-Dbusinessobjects.orb.ocl.protocol=ssl
-DcertDir=C:\SSLCert
-DtrustedCert=cacert.der
-DsslCert=servercert.der
-DsslKey=server.key
-Dpassphrase=passphrase.txt
```

3 クエリの最適化

3.1 データフェデレーションクエリのパフォーマンス調整

データフェデレーションクエリのパフォーマンスを調整できます。パフォーマンスの調整に使用される戦略は以下のとおりです。

1. システムパラメータを使用して、メモリの使用を最適化します。
2. クエリソースの最適なアルゴリズムがアプリケーションによって選択されるように、統計情報を使用します。
3. 準結合演算子が自動的に有効化されない場合には、準結合を有効化するためにパラメータを変更できるかどうかを確認します。
4. 準結合が適当でない場合には、*merge* 結合を有効化するためにパラメータを変更できるかどうかを確認します。
5. データがデフォルトで無効化されている機能に対応している場合には、コネクタの機能を有効化します。
たとえば、DB2 は NULL 値の予測可能な順序をサポートしていませんが、データに NULL がないことが分かっている場合には、*merge* 結合を引き続き使用できます。この場合には、*order by* を強制的に実行するようにソース機能を設定します。

i 注記

データファンデーションに対してクエリを最適化するためにシステムパラメータを変更した後、データファンデーションを別のシステムに昇格する場合は、システムパラメータも昇格する必要があります。SAP BusinessObjects Enterprise ライフサイクルマネジメントコンソールを使用して、この作業を実行できます。

関連情報

[システムパラメータを使用したメモリ使用の最適化](#) [20 ページ]

[大規模テーブルに結合された小規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン](#) [27 ページ]

[並べ替えることができるデータを含む大規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン](#) [29 ページ]

[データフェデレーションサービス用に作成した最適化設定の昇格](#) [33 ページ]

3.2 システムパラメータを使用したメモリ使用の最適化

以下の戦略を利用して、アプリケーションによるメモリ使用を最適化できます。

- アプリケーションを実行している Java Virtual Machine (JVM) で使用されるメモリ量を設定します。詳細については、*SAP BusinessObjects Enterprise 管理者ガイド*のサーバプロパティの変更に関する説明を参照してください。
アプリケーションの速度と利用できるメモリ量に応じて、デフォルト値を調整します。
- サーバパラメータ `EXECUTOR_TOTAL_MEMORY` を設定します。

このパラメータを使用すると、クエリの実行に使用されるメモリ量を設定することができます。

このパラメータを JVM で使用されるメモリのパーセント値、または単位を表す接尾辞が付いた固定値 (たとえば、512M、512m、1024K、または 1024k) で設定します。固定値を入力する場合には、JVM に指定された値よりも小さな値を入力する必要があります。

- サーバパラメータ `EXECUTOR_STATIC_MEMORY` を設定します。
このパラメータで、初期化時に演算子に対して割り当てられる最小メモリ量を設定できます。エグゼキュータで使用されるメモリのパーセント値、または固定値を設定できます。固定値を入力する場合には、クエリ実行に指定された値よりも小さな値を入力する必要があります。
- サーバパラメータ `MAX_CONCURRENT_MEMORY_CONSUMING_QUERIES` を設定します。
メモリを消費する並列実行クエリ数を定義します。その他のクエリは影響を受けません。
大きなクエリがある場合は、ここに小さな値を入力します。
小さなクエリが多くある場合は、大きな値を入力します。
- `MAX_CONCURRENT_MEMORY_CONSUMING_OPERATORS`
このパラメータで、メモリを消費する並列実行演算子の数を制限します。
クエリ内の演算子が大量のメモリを消費している場合は、この数を減らします。
アクセスされる複数のデータベース内にある大きなテーブルの数を数えることで、クエリ内の演算子の平均的なサイズと数の概算を得ることができます。たとえば、1 つのマップ規則にある複数のデータソース中に大きなテーブルが 4 つある場合、メモリを消費する結合が 3 つあります。

たとえば、JVM メモリを 1000M に設定して、JVM に 1000 メガバイトのメモリを割り当てます。

次に、`EXECUTOR_TOTAL_MEMORY` を 80% に設定して、クエリ実行に 800 メガバイトのメモリを割り当てます。

次に、`EXECUTOR_STATIC_MEMORY` を 25% に設定して、演算子ごとに 200 メガバイトのメモリを割り当てます。

次に、`MAX_CONCURRENT_MEMORY_CONSUMING_QUERIES` を 2 に設定して、並列演算子を 2 個に制限します。

上記の設定例では、2 件のクエリを並列実行できます。それぞれのクエリには最小 100 メガバイトのメモリがあり、600 メガバイトのメモリの動的プールにアクセスできます。

システムのメモリ使用を監査するには、`info buffermanager` 文を使用します。

i 注記

データファンデーションに対してクエリを最適化するためにシステムパラメータを変更した後、データファンデーションを別のシステムに昇格する場合は、システムパラメータも昇格する必要があります。SAP BusinessObjects Enterprise ライフサイクルマネジメントコンソールを使用して、この作業を実行できます。

関連情報

[データフェデレーション管理ツールを使用したシステムパラメータの変更](#) [62 ページ]

[メモリを消費する演算子](#) [22 ページ]

[データフェデレーションサービス用に作成した最適化設定の昇格](#) [33 ページ]

3.2.1 メモリを消費する演算子

以下の演算子をクエリで使用すると、データフェデレーションサービスによってメモリが消費されます。

- `join`
- `cartesian product`
- `orderby`
- `groupby`
- `groupby` グループ内に多くの異なる値がある場合 (大規模グループセット)

データフェデレーションクエリエンジンは、テーブルのスキャン、射影、フィルタ、関数の評価の実行時や、ソースに操作をプッシュする際に、それほど多くのメモリを使用しません。

3.3 クエリソースの最適なアルゴリズムがアプリケーションによって選択されるための統計情報の使用

データフェデレーションクエリエンジンでは、クエリを最適化するために統計情報が内部で使用されます。

統計情報が継続的に最新表示されることはありません。これは、システムが本稼動でデプロイされるまで待機し、サンプル時刻に統計情報を実行するということです。それから統計情報が収集され、後続のクエリプランの生成で考慮されます。

統計情報サブシステムは実際には 2 つの部分から構成されています。

- データソースレベルで既知のメジャーからカーディナリティを計算するツール
- クエリの実行時にテーブルまたは属性が要求される回数を数えるレコーダ

値を手動入力してカーディナリティを上書きして、クエリプランの最適化での使用を制御できます。

関連情報

[列カーディナリティについて](#) [22 ページ]

[記録された統計値をフィルタ処理し、必要なものだけを計算してレポートを最適化](#) [23 ページ]

3.3.1 列カーディナリティについて

カーディナリティとは、列の行数のことです。

他の要素のカーディナリティを測定することもできます。テーブル、テーブルを含むスキーマ、またはカタログ全体のカーディナリティを測定できます。いずれの場合でも、オブジェクトのカーディナリティを示すことで、そのオブジェクトに含まれているすべてのオブジェクトのカーディナリティを示すショートカットです。たとえば、スキーマのカーディナリティが 1000 の場合、そのスキーマに含まれている大部分のテーブルのほとんどの列は 1000 行あるということです。

データフェデレーションの使用時には、データソースの列のカーディナリティが正確に判明している方がクエリをよりの確に最適化できます。そのため、データフェデレーションクエリエンジンではデータソースのカーディナリティを推測でき、カーディナリティを正確に把握している場合には、カーディナリティを設定できます。

カーディナリティの推定と設定は、統計情報設定と呼ばれる最適化タスクの一部です。

関連情報

[クエリソースの最適なアルゴリズムがアプリケーションによって選択されるための統計情報の使用](#) [22 ページ]

3.3.2 列間の関係のファンアウト値について

ファンアウト値の推定と設定は、統計情報設定と呼ばれる最適化タスクの一部です。

ファンアウトは、2 つの列にあるデータ間の関係を測定します。列が 2 つある場合、最初の列にある各非重複値に対して、ファンアウトは 2 番目の列の平均列数となります。たとえば、第 1 列に国名、別の列に市町村名が一覧表示されている場合には、ファンアウトで各国の平均市町村数を測定できます。

データフェデレーションの使用時には、クエリエンジンはデータソースの列のファンアウトが正確に判明している方がクエリをよりの確に最適化できます。そのため、データフェデレーションクエリエンジンではソースの列のファンアウトを設定できます。

関連情報

[クエリソースの最適なアルゴリズムがアプリケーションによって選択されるための統計情報の使用](#) [22 ページ]

3.3.3 記録された統計値をフィルタ処理し、必要なものだけを計算してレポートを最適化

すべてのデータソースの統計情報を一度に計算できますが、この操作には時間がかかることがあります。以下の手順では、クエリに必要な統計情報のみを計算し、このプロセスを効率化する方法を示します。

この手順では、SAP BusinessObjects Interactive Analysis ドキュメントの最新表示によって生成される統計情報を取得する例を使用しますが、他のあらゆる状況にも適用できます。

カーディナリティの計算はいつでも実行でき、アクティブ化は必要ありません。

1. SAP BusinessObjects Interactive Analysis の [\[クエリの編集\]](#) パネルでレポートを開きます。
2. [\[クエリパネル\]](#) で SQL テキスト領域を開き、クエリの SQL をコピーし、テキスト領域を閉じます。
3. データフェデレーション管理ツールで、[\[クエリパネル\]](#) タブのテキスト領域に SQL を貼り付けます。
4. [\[実行\]](#) をクリックします。
5. [\[統計情報\]](#) タブで、[サーバ統計情報を最新表示ボタン](#)をクリックします。

クエリを最適化するために使用されるテーブルおよび列は、[リクエスト数] 列に記録されます。

6. [統計情報] タブで、次の操作を実行します。

- [リクエスト数] 列のフィルタの値を [記録済み] に設定してください。

7. Ctrl キーを押しながらクリックして [リクエスト数] 列に値があるすべての行を選択し、計算ボタンをクリックします。

データフェデレーション管理ツールでは、クエリに有用な統計情報のみが計算されます。

8. SAP BusinessObjects Interactive Analysis レポートの中でクエリを最新表示にすることにより、実際のクエリを実行します。

データフェデレーションクエリエンジンは、収集された統計情報を使用して、最適なプランを生成するようになります。

関連情報

[データフェデレーション管理ツールの統計情報タブ](#) [17 ページ]

3.4 クエリプランの最適化

3.4.1 データフェデレーション管理ツールの [クエリプラン] ビュー

定義

[クエリの説明] をクリックすると、クエリ最適化の結果が [クエリプラン] ビューに表示されます。[クエリプラン] ビューには次の3つのペインがあります。

- [プラン] ペイン: クエリプランをツリー構造で表示します。
- [詳細] ペイン: [プラン] ペインで強調表示されている項目の詳細を表示します。
- [プロパティ] ペイン: [プラン] ペインおよび [詳細] ペインで強調表示されている項目のプロパティを表示します。

[プラン] ペインには、クエリプランがツリー構造で表示されます。この構造内のリーフは、コネクタに送信されるコネクタクエリです。中間ノードは、プロジェクション、Order By、Group By、集計、ユニオン、Full Outer Join、計算 (フィルタ、結合) などです。

このドキュメントには、クエリおよびコネクタクエリに関する一般的な情報のみが記載されています。この情報 (中間ノードは含まれません) は、ユーザにデフォルトで表示されます。

1. クエリに関するすべての情報は次のとおりです。

a. [プロパティ] ペイン内:

- a. 使用メモリ: クエリに必要なメモリ容量の概算
- b. 演算子を消費しているメモリの同時実行数: クエリプラン内で実行される、演算子を消費しているメモリの同時実行数

b. [詳細] ペイン内:

- a. 統計情報
 - a. テーブルのカーディナリティ: このクエリによって返される推定行数
2. コネクタクエリに関する情報は次のとおりです。
- a. [プロパティ] ペイン内:
 - a. ID: コネクタクエリの ID
 - b. データフェデレーション SQL: 複数コネクタクエリエンジンで使用される SQL 構文で示されるコネクタクエリ
 - c. ネイティブコネクタクエリ: (コネクタでサポートされる) ネイティブ構文で示されるコネクタクエリ
 - d. コネクタ名: コネクタの名前
 - b. [詳細] ペイン内:
 - a. スキーマ: コネクタクエリの推定される列の一覧
 - b. キー: 派生キー (テーブルのキーから推定されたキー)
 - c. 統計情報: オプティマイザで使用された統計情報と、その統計情報それぞれの推定値
 - a. テーブルのカーディナリティ
 - b. 列のカーディナリティ
 - d. 機能: コネクタが実行可能な操作の一覧
 - e. 準結合: 準結合の一覧
 - a. フィルタが適用された列: 準結合で使用された列の一覧
 - a. 依存列: この (フィルタが適用された) 列のフィルタ適用に使用された列
 - b. 依存ソースクエリ: 準結合の値を提供するコネクタクエリの一覧
 - c. ストラテジー: 希望の順番に並べた、準結合演算子の実行方針の一覧
 - d. 減少係数: 準結合なしで返された行の数と、準結合を使用して返された行の数の比率
 - f. Data Federator SQL: クエリエンジンで使用される SQL 構文で示されるコネクタクエリ
 - g. ネイティブコネクタクエリ: (コネクタでサポートされる) ネイティブ構文で示されるコネクタクエリ

3.4.2 【統計情報の説明】コマンド

説明

[統計情報の説明] コマンドを使用すると、クエリエンジンが SQL クエリを最適化する上で必要なすべての統計情報が一覧で表示されます。このコマンドをクエリに対して実行すると、ツリー構造に似た構造が返されます。このビューでは、クエリ内では各ソースに対してどのテーブルが使用されるか、どの統計情報が必要か、およびその統計情報は更新されたかどうかを確認できます。このビューでは、次の操作が実行できます。

1. 1 回のクリックで、クエリに必要なすべての統計情報を最新表示します。
2. 特定のテーブルまたは列の統計情報を最新表示します。
3. 特定のテーブルまたは列の統計情報を設定します。
4. 最適なプランを作成するために必要なすべての統計情報を使用できるかどうかを確認します。
5. 使用中の統計情報が、ソースからの統計情報か、またはユーザが設定した統計情報かを表示します。

コマンドの結果は、次の 6 列で構成されます。

- **カタログ:** ユーザがソースおよびテーブル/列を参照できるツリービュー。
- **最終計算日:** 統計情報がソースから計算された最新の日付。
- **リクエスト数:** 列の一意の値 (テーブルカーディナリティ) が (このクエリに対してではなく) システム内でリクエストされた回数。
- **現在のカーディナリティ:** ソースからのカーディナリティまたはユーザ (管理者) からのカーディナリティの、2 種類のカーディナリティが表示されます。使用されるポリシーに応じて、適切なカーディナリティが現在のカーディナリティとして表示されます。
- **ソースからのカーディナリティ:** データソースからのカーディナリティ。
- **ユーザのカーディナリティ:** ユーザが特定のテーブルまたは列に対して異なる統計情報を設定した場合は、その統計情報が表示されます。

3.4.3 フィードバックを得てクエリを調整するための、クエリの説明機能の使用

クエリを調整するためのフィードバックとして、クエリの実験機能を使用することができます。次のクエリでは、2 つの異なるデータソースから 2 つのテーブル間の結合を実行します。〈T1〉はデータソース 〈S1〉の小サイズのテーブル、〈T2〉はデータソース 〈S2〉の大サイズのテーブルです。

1. [クエリパネル] で、「`Select * From 〈T1〉, 〈T2〉 where 〈T1〉.<C1〉 = 〈T2〉.<C2〉`」と入力します。
2. [クエリの実験] をクリックします。
3. [プラン] パネルで、ソースクエリ 〈S1〉 [〈T1〉]、および 〈S2〉 [〈T2〉] をクリックします。

[詳細] パネルに詳細情報が表示されます。詳細を見ると、〈S1〉 および 〈S2〉 のソースクエリがどちらも完全テーブルスキャンであることがわかります。しかし、〈T1〉は小サイズのテーブルということがわかっているため、〈S2〉で準結合が生成されるものと予測できます。準結合が生成されない理由を調べるには、両方のソースクエリの統計情報を確認します。すると、オプティマイザが次を使用しようとしていることがわかります。

- 〈T1〉のカーディナリティ
- 〈T1〉.<C1〉のカーディナリティ
- 〈T2〉のカーディナリティ
- 〈T2〉.<C2〉のカーディナリティ

しかし、これらすべての統計情報は不明とマークされています。

4. [統計情報の説明] をクリックします。
[クエリ統計情報] タブが表示されます。

[クエリ統計情報] タブでは、統計情報を次の値に設定できます。

- Cardinality(〈T1〉)=25
- Cardinality(〈T1〉.<C1〉)=25
- Cardinality(〈T2〉)=100000
- Cardinality(〈T2〉.<C2〉)=100000

5. [クエリの実験] を再度クリックします。

以前と異なるプランが表示され、〈S2〉の準結合が生成されています。

関連情報

[データフェデレーション管理ツールの \[クエリプラン\] ビュー \[24 ページ\]](#)

[大規模テーブルに結合された小規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン \[27 ページ\]](#)

3.4.4 データフェデレーション管理ツールを使用した演算子がプッシュされているかどうかの確認

データフェデレーションクエリエンジンではなくデータベースシステムを使用している場合は、クエリは一般的に、演算子をより効率的に評価します。

データフェデレーション管理ツールの [\[クエリの監視\]](#) タブで、演算子がプッシュされているかどうかを確認できます。

1. データフェデレーション管理ツールで、[\[クエリの監視\]](#) タブを開きます。
2. [\[最新表示\]](#) ボタンをクリックし、最新のクエリを表示します。
3. 使用中のクエリを見つけ、サブクエリを観察して、演算子がプッシュされているかどうかを調べます。
 - 演算子がサブクエリの中で表示されている場合は、その演算子がデータソースに対してプッシュされていることを意味します。
 - 演算子が上端のクエリの中にのみ表示されている場合は、その演算子がプッシュされていないことを意味します。データフェデレーションクエリエンジンが演算子をデータソースに強制的にプッシュするように設定するために、コネクタ機能が演算子を受け付けるように、データソースに対するコネクタ機能を設定して試みる必要があります。

関連情報

[データフェデレーションクエリのパフォーマンス調整 \[20 ページ\]](#)

[データフェデレーション管理ツールのクエリの監視タブ \[12 ページ\]](#)

[データフェデレーション管理ツールを使用したリレーショナルコネクタ機能および SAS コネクタ機能の設定 \[58 ページ\]](#)

3.4.5 大規模テーブルに結合された小規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン

クエリを最適化する際に、データフェデレーションオプティマイザは、データソースからクエリエンジンに転送されるデータを削減しようとします。これを達成する方法の 1 つとして、データソース内の大規模なテーブルにアクセスする際に準結合が生成されます。オプティマイザは、パフォーマンスの向上が予想される場合にのみ準結合を生成します。

準結合の生成と実行は、以下のシステムパラメータとコネクタプロパティによって管理されます。

- `ACTIVATE_SEMI_JOIN_RULE`
準結合生成ルールが有効化されたかどうか。オプティマイザは、このパラメータが `true` に設定されている場合のみ準結合を生成します。

- **MIN_SOURCE_CARDINALITY_THRESHOLD_FOR_SEMI_JOIN_RULE**
 オプティマイザが準結合を生成しようとしているソースクエリの最小カーディナリティ。オプティマイザは、大量のデータを返すソースクエリに対してのみ、準結合を生成します。ソースクエリの予想カーディナリティがこのパラメータより小さい場合は、オプティマイザは、このソースクエリに対し準結合を生成しません。
- **MIN_ACTIVATION_THRESHOLD_FOR_SEMI_JOIN_RULE**
 準結合の目的は、データソースからクエリエンジンに転送されるデータを低減することです。このパラメータは、オプティマイザが準結合を生成するために最低限必要なデータ転送減少量です。(準結合を生成しない場合の行数/準結合を生成した場合の行数)で計算された比率は、減少係数と呼ばれます。減少量がこのパラメータより大きい場合は、準結合が生成され、それ以外の場合は準結合は生成されません。

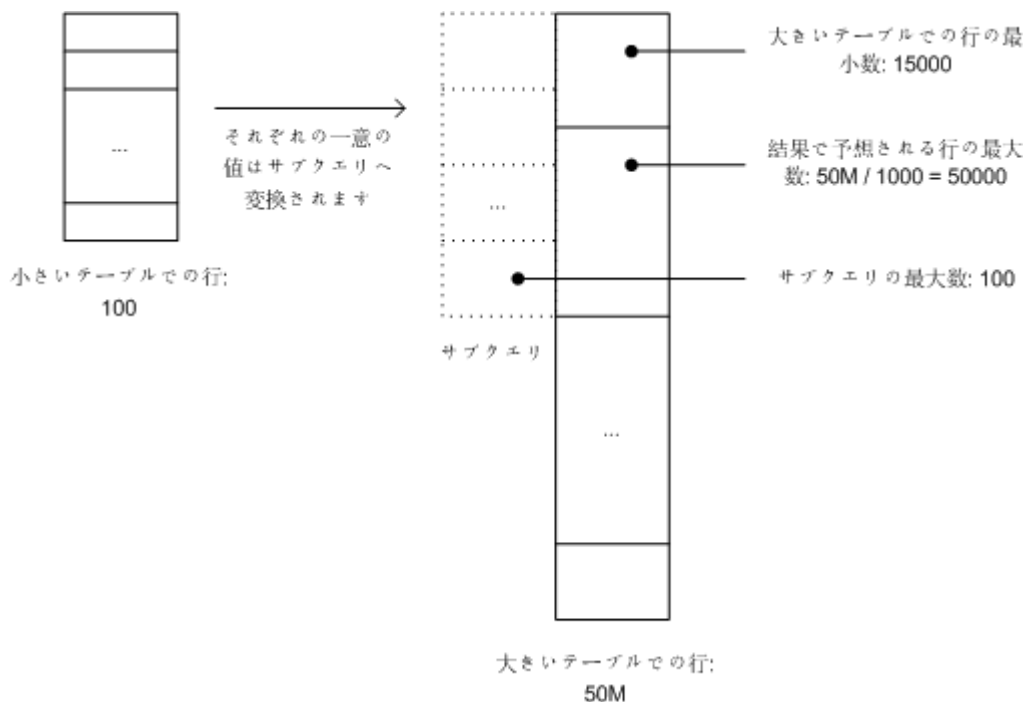


図 1: パラメータ `ACTIVATE_SEMI_JOIN_RULE=true`、`MIN_SOURCE_CARDINALITY_THRESHOLD_FOR_SEMI_JOIN_RULE=15000`、および `MIN_ACTIVATION_THRESHOLD_FOR_SEMI_JOIN_RULE=1000` を使用したデータフェデレーションクエリエンジンによる準結合を生成するかどうかの決定方法

i 注記

データファンデーションに対してクエリを最適化するためにシステムパラメータを変更した後、データファンデーションを別のシステムに昇格する場合は、システムパラメータも昇格する必要があります。SAP BusinessObjects Enterprise ライフサイクルマネジメントコンソールを使用して、この作業を実行できます。

例

大規模テーブルと小規模テーブルのクエリでの準結合の有効化

この例では、100 行の小規模テーブルと 5000 万行の大規模テーブルがある場合に、準結合を有効化するためにシステムパラメータとセッションパラメータを設定する方法を説明します。小規模テーブルの値を使って大規模テーブルの値をフィルタすると、10000 行が返されることも仮定します。

データフェデレーションプロジェクトがデプロイされたら、統計情報を最新表示します。データフェデレーション管理ツールで統計情報を最新表示できます。

`MIN_SOURCE_CARDINALITY_THRESHOLD_FOR_SEMI_JOIN_RULE` を 15000 に設定します。大規模テーブルの行数が 15000 を超えると、クエリエンジンでは準結合を使用できるようになります。

`MIN_ACTIVATION_THRESHOLD_FOR_SEMI_JOIN_RULE` を 1000 に設定します。これは適切なデフォルト値です。これは以下のように使用されます。

大規模テーブルの行数をこの数字で割り、しきい値を計算します。この例では、しきい値は 50000 ($50M/1000 = 50000$) です。次にクエリエンジンでは統計情報がチェックされます。統計情報は準結合で約 10000 行が返されることを示しています。この行数はしきい値 50000 行未満であるため、データフェデレーションアプリケーションでは準結合を使用できます。

この値を低く設定し過ぎると、効率的でない場合にもクエリエンジンで準結合が使用されます。たとえば、この値を 1 に設定すると、準結合で返される行数が 5000 万 ($5000 \text{ 万}/1 = 5000 \text{ 万}$) であったとしても、クエリエンジンで準結合が使用されます。これはテーブルのフルスキャンを実行していることと同じです。

この値を 2 に設定すると、準結合で返される行数がテーブルスキャンで返される行数の半分である場合に、クエリエンジンで準結合が使用されます。これはテーブルのフルスキャンと比べて効率が低いとは言えません。

この値を高く設定し過ぎると、効率的な場合でもクエリエンジンで準結合が使用されなくなります。たとえば、この値を 5000 万に設定すると、準結合で返される行数が 1 ($5000 \text{ 万}/5000 \text{ 万} = 1$) である場合に限り、クエリエンジンで準結合が使用されます。

この値を 1000 に設定すると、結果がテーブルスキャンの 1000 分の 1 である場合に準結合をアクティブにするよう要求するのとはほぼ同じことになります。

これらの設定をするとクエリエンジンで準結合を実行できるようになり、最適な速度とメモリ使用でクエリを実行できます。

関連情報

[データフェデレーションサービス用に作成した最適化設定の昇格 \[33 ページ\]](#)

3.4.6 並べ替えることができるデータを含む大規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン

クエリで大規模テーブルが返され、それらのテーブルに含まれているデータを並べ替えることができる場合には、アプリケーションで順序ベース演算子を使用して、処理速度を上げることができます。順序ベース演算子は *merge* 結合と順序ベース *group by* です。

merge 結合では、ソースの *order by* 演算子をプッシュしてから、順序化された結果を使用して結合をオンザフライで実行します。

この方法は、結合の必要がある結果の保存を回避するものです。そのため、順序化されていない結果に結合を適用するよりも処理速度が速くなります。

merge 結合が役立つかどうかのチェック

ただし、merge 結合が役立つのは、以下の条件がすべて満たされた場合に限りです。

- 準結合ができない
- クエリで結合対象の大規模テーブルが返される
- データソースが *order by* 演算子をサポートしている、あるいはデータに *order by* 演算子を使用できる
 - *order by* 演算子をサポートしているかどうかについて、データソースの機能をチェックできます。
たとえば、DB2 は NULL 値の予測可能な順序をサポートしていません。
 - 照合設定が予測可能ではないため、*order by* をサポートしていないソースもあります。
たとえば、DB2 は NULL 値の予測可能な順序をサポートしていませんが、データに NULL がないことが分かっている場合には、merge 結合を引き続き使用できます。この場合には、*order by* を強制的に実行するようにソース機能を設定します。

merge 結合パラメータを変更するかどうかのチェック

大規模テーブルで merge 結合 はデフォルトでアクティブです。システムパラメータを使用して merge 結合の有効化を制御できます。

以下のような特定の条件では、パラメータの設定が必要な場合があります。

- 大規模テーブルがあるが、サイズが大きな行に分散される。クエリで返されるのは必要最小限の行に限られる。
- 小規模テーブルがあるが、それでも merge 結合を使用する。

merge 結合がアクティブかどうかの確認

merge 結合が使用されていることを確認するには、データフェデレーション管理ツールでクエリ履歴を参照し、サブクエリに *order by* 演算子が含まれていることを確認します。

関連情報

[順序ベース演算子の有効化を制御するシステムパラメータの使用 \[31 ページ\]](#)

[大規模テーブルに結合された小規模テーブルのクエリの最適化を目的としたシステムパラメータの使用に関するガイドライン \[27 ページ\]](#)

[データフェデレーション管理ツールを使用したリレーショナルコネクタ機能および SAS コネクタ機能の設定 \[58 ページ\]](#)

3.4.7 順序ベース演算子の有効化を制御するシステムパラメータの使用

以下のパラメータを使用して順序ベース演算子をトリガできます。

- 順序ベース演算子の有効利用を検知するためのオプティマイザルールを有効化するために、サーバパラメータ `ACTIVATE_ORDER_BASED_OPTIMIZATION_RULE` を `true` に設定します。
- サーバパラメータ `MIN_STORE_CARDINALITY_THRESHOLD_FOR_ORDER_BASED_JOIN_RULE` および `MIN_TRANSFER_CARDINALITY_THRESHOLD_FOR_MERGE_JOIN_RULE` を設定します。これらの数字は、merge 結合演算子を選択するための入力オペランドの最小カーディナリティ (行数) を定義するものです。merge 結合を選択できるのは、一方のオペランドに `minStoreCardForMergeJoin` のカーディナリティがあり、他方のオペランドに `minTransferCardForMergeJoin` のカーディナリティがある場合に限られます。
- サーバパラメータ `MIN_CARDINALITY_THRESHOLD_FOR_GROUP_BY_TRANSFORMATION_RULE` を設定します。この数字は `orderBasedGroupBy` 演算子を選択するための入力オペランドの最小カーディナリティを定義するものです。

i 注記

データファンデーションに対してクエリを最適化するためにシステムパラメータを変更した後、データファンデーションを別のシステムに昇格する場合は、システムパラメータも昇格する必要があります。SAP BusinessObjects Enterprise ライフサイクルマネジメントコンソールを使用して、この作業を実行できます。

関連情報

[データフェデレーションサービス用に作成した最適化設定の昇格 \[33 ページ\]](#)

3.4.8 データソースサブクエリの強制並列実行

デフォルトでは、データフェデレーションアプリケーションからデータソースへサブクエリの実行が送信されるのは、データフェデレーションアプリケーションがサブクエリの結果を消費できる状態である場合に限られます。これにより、基盤データベースによるクエリ結果のキャッシュ時間が削減され、キャッシュにかかる時間が長すぎた場合のタイムアウトが回避されます。

ただし、データソースクエリを強制的に早期送信することができます。

`union` 演算子のオペランドであるデータソースサブクエリの並列送信を有効化するために、サーバパラメータ `ACTIVATE_MULTI_THREADED_UNION_OPERATOR` を `true` に設定します。

3.4.9 準結合の実行ストラテジー

概要

データフェデレーションサービスが準結合演算子を適用して小さいテーブルと大きいテーブルの間の結合を最適化する場合は、このサービスは次のストラテジーのいずれかを使用して、大きいテーブルから取得する行の数を削減できます。

これらのストラテジーのそれぞれは、比較的小さい値の一覧を作成し、大きいテーブルの中にある行をその一覧に結合します。実行ストラテジーは単純に、この一覧を作成するための技術的な方法のことです。すべてのデータソースが同じ方法をサポートしているわけではありません。

`SEMI_JOIN_EXECUTION_STRATEGIES` パラメータを使用してこれらのストラテジーを有効または無効にするか、優先順位を変更することができます。

ストラテジー	説明
<code>IN</code>	データフェデレーションクエリエンジンは、 <code>IN</code> キーワードを使用して値の一覧を作成します。
一時テーブル	データフェデレーションクエリエンジンは、データソース上に一時テーブルを作成することにより、値の一覧を作成します。
準備した文	データフェデレーションクエリエンジンは、SQL で準備した文を使用して値の一覧を作成し、一覧の中にある各値は、準備した文に対して 1 つのパラメータとして渡されます。

3.5 特定のコネクタの最適化

3.5.1 SAP BW へのパラレルクエリに関するコールバックの同時実行の増加

`jcoServerProperties` というリソースプロパティを使用してクエリサーバが SAP BW からのコールバックに対して提供するスレッドの数を増やすことができます。

1. データフェデレーション管理ツールを起動し、管理者権限のあるユーザアカウントを使用してログオンします。
2. **コネクタ設定** タブを使用して SAP BW コネクタを編集します。
3. コネクタプロパティ `jcoServerProperties` を値 `jco.server.connection_count=10` に設定します。

このプロパティのデフォルト値は 2 です。`MAX_CONCURRENT_MEMORY_CONSUMING_QUERIES` システムパラメータが 10 を上回っている場合以外は、最大推奨値は 10 です。その場合は、リソース不足を回避するために、このシステムパラメータの値よりスレッド数を多くする必要があります。

i 注記

コネクタプロパティの名前は `jcoServerProperties` です。値は、文字列 `jco.server.connection_count=10` に設定する必要があります。

関連情報

[SAP BW データソースのコネクタプロパティの一覧](#) [49 ページ]

3.5.2 SAP BW に対するクエリからの応答パッケージのサイズ変更

`packageSize` というリソースプロパティを使用し、SAP BW からクエリの応答として返されるデータパッケージのサイズを変更できます。パッケージのサイズは、パッケージごとの行数で測定されます。

パッケージのサイズを大きくすると、速度は向上しますが、より多くのメモリを使用することになります。

逆に、パッケージのサイズを小さくすると、速度は低下しますが、メモリを節約できます。

1. データフェデレーション管理ツールを起動し、管理者権限のあるユーザアカウントを使用してログオンします。
2. [コネクタ設定](#) タブを右クリックして SAP BW コネクタを編集します。
3. `packageSize` プロパティを編集します。このプロパティの値として、パッケージごとの必要な行数を入力します。

詳細については、SAP BW コネクタプロパティの一覧にある `packageSize` プロパティの説明を参照してください。

関連情報

[SAP BW データソースのコネクタプロパティの一覧](#) [49 ページ]

3.6 データフェデレーションサービス用に作成した最適化設定の昇格

データファンデーションをあるシステムから別のシステムに、たとえば、開発システムからテストシステムに移行する場合は、その作業を昇格と呼びます。

データフェデレーションサービス用にシステムパラメータに最適化の変更を加えた場合は、データファンデーションを昇格するときにそのシステムパラメータを昇格する必要があります。

SAP BusinessObjects Enterprise ライフサイクルマネジメントコンソールを使用して、この作業を実行することもできます。

1. SAP BusinessObjects Enterprise ライフサイクルマネジメントを開きます。
2. [\[フォルダとオブジェクト\]](#) フォルダを展開し、[\[データフェデレーション\]](#) をクリックします。
3. オブジェクトの [\[パラメータ\]](#) を昇格します。

詳細については、SAP BusinessObjects Enterprise ライフサイクルマネジメントコンソールのユーザガイドを参照してください。

4 データのソースへのコネクタの設定

4.1 データフェデレーション管理ツールのコネクタの情報の表示

1. データフェデレーション管理ツールを起動します。
2. [\[コネクタ設定\]](#) タブをクリックします。
3. ツリーリストにあるコネクタをダブルクリックします。
4. [\[一般情報\]](#) をクリックして設定を表示するか、[\[機能\]](#) をクリックしてコネクタの機能を表示します。

4.2 データフェデレーション管理ツールのコネクタのプロパティの変更

1. データフェデレーション管理ツールを起動します。
2. [\[コネクタ設定\]](#) タブをクリックします。
3. ツリーリスト内でコネクタを右クリックし、[\[設定の作成\]](#) をクリックします。
4. 表示される [\[設定\]](#) ノードをダブルクリックします。
5. [\[設定プロパティ\]](#) タブで、編集するプロパティをダブルクリックし、値を変更し、[データの保存](#) アイコンをクリックして変更を保存します。

4.3 リレーショナルデータソースのコネクタの設定

4.3.1 リレーショナルデータソースの共通コネクタプロパティの一覧

次の表は、リレーショナルデータソースに設定できる共通プロパティの一覧です。

プロパティ	説明
capabilities	データベースがサポートするすべての機能の一覧です。要素は文字「;」で区切ります(要素間にスペースは入れません)。 例 capabilities=fullSQL\=true;outerjoin\=false;rightouterjoin\=true
compCollationCompatible	True/Yes または False/No データソースでの比較操作の照合がデータフェデレーションサービスの現在の設定に適合できるかどうかを指定します。true に設定すると、サーバは

プロパティ	説明
	<p>比較操作の照合を無視し、述語はソースへ安全にプッシュされます。デフォルト値: <code>false</code>。</p> <p>例</p> <pre>compCollationCompatible=true</pre>
<code>sortCollationCompatible</code>	<p><code>True/Yes</code> または <code>False/No</code></p> <p>データソースでの並べ替え操作 (<code>ORDER BY</code>) の照合がデータフェデレーションサービスの現在の設定に適合できるかどうかを指定します。true に設定すると、サーバは並べ替え操作の照合を無視し、(<code>ORDER BY</code>) 式はソースへ安全にプッシュされます。デフォルト値: <code>false</code>。</p> <p>例</p> <pre>sortCollationCompatible=true</pre>
<code>longVarCharMaxSize</code>	テキストデータ型などの <code>longvarchar</code> データ型のサイズを制限します。
<code>varCharMaxSize</code>	<code>varchar</code> データ型のサイズを制限します。デフォルト値は <code>-1</code> です。これは、文字列の切り捨てが行われないことを意味します。
<code>arrayFetchBufferSize</code>	各配列フェッチに割り当てられるバッファの最大サイズ (バイト) を定義します。デフォルト値は <code>65536</code> です。
<code>enableArrayFetchSizeOptimization</code>	配列フェッチサイズの最適化を有効または無効にします。デフォルト値は <code>true</code> です。
<code>maxConnectionIdleTime</code>	<p>アイドル接続が接続プールに保持される最長時間です。単位はミリ秒です。<code>-1</code> は上限がないことを表します。このパラメータが設定されていない場合は、接続サーバのデフォルトのプールタイム <code>100000</code> (10 mn) が使用されます。</p> <p>次のいずれかの値です。</p> <ol style="list-style-type: none"> <code>-1</code>: タイムアウトしません。ラッパの存続期間中は、接続はプールに保持されます。 <code>0</code>: プールでは管理されない接続です。 <code>> 0</code>: 値は、アイドル接続が保持される最長時間 (ミリ秒) です。
<code>enableUpdateQueries</code>	<p><code>True/Yes</code> または <code>False/No</code></p> <p>更新クエリの実行が有効かどうかを指定します。</p> <p>デフォルト値: <code>True</code>。</p>
<code>enableTemporaryTableQueries</code>	<p><code>True/Yes</code> または <code>False/No</code></p> <p>一時テーブルクエリの実行が有効かどうかを指定します。</p> <p>デフォルト値: <code>True</code>。</p>

プロパティ	説明
<i>maxValuesInInClause</i>	<p>IN 句の値の最大数を指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<i>semiJoinMaxQueries</i>	<p>準結合演算子が実行できるクエリの最大数を指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<i>semiJoinTempTableMinCardinality</i>	<p>準結合演算子の一時テーブル方針に、ディメンションの最小カーディナリティを指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<i>semiJoinTempTableMaxCardinality</i>	<p>準結合演算子の一時テーブル方針に、ディメンションの最大カーディナリティを指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<i>semiJoinExecutionStrategies</i>	<p>希望の順番で、準結合演算子の実行方針の一覧を指定します。</p> <p>次のいずれかの値です。</p> <ul style="list-style-type: none"> • DEFAULT DEFAULT は、システムパラメータ SEMI_JOIN_EXECUTION_STRATEGIES の値が使用されることを表します。 • I、T、P をカンマで区切って希望の順番で組み合わせたものです。I は IN クエリの実行方針、T は一時テーブルの実行方針、P はパラメータ化されたクエリの実行方針を表します。 例 <ul style="list-style-type: none"> ○ T,P,I ○ I,T ○ P ○ 方針なし <p>I、T、または P のうちのいずれかが指定されていない場合、対応する実行方針はラッパでサポートされません。</p> <p>注: NONE は、ラッパがサポートする実行方針がないことを表します。</p> <p>戻り値には、null も空の文字列も含まれません。</p>
<i>allowPartialResults</i>	<p>True/Yes または False/No</p> <p>このパラメータは、パラメータ maxRows と組み合わせて使用されます。</p> <p>maxRows が正の数で設定され、指定上限を超える行数がクエリから返された場合、デフォルトでは例外が投げられます。この動作を変更するには、パラ</p>

プロパティ	説明
	メータ <code>allowPartialResults</code> を <code>true</code> に設定します。デフォルト値は <code>false</code> です。
<code>maxRows</code>	<p>返す最大行数を定義します。このパラメータは、パラメータ <code>allowPartialResults</code> と組み合わせて使用されます。</p> <p><code>maxRows</code> が正の数で設定され、指定上限を超える行数がクエリから返された場合、デフォルトでは例外が投げられます。</p> <p>この動作を変更するには、パラメータ <code>allowPartialResults</code> を <code>true</code> に設定します。デフォルト値は <code>0</code> です。これは、上限がないことを表します。</p>
<code>maxLevelOfFunctionNesting</code>	関数のネストでサポートされるレベルの最大数を指定します。デフォルト値: <code>0</code> 。 <code>0</code> は制限がないか、または制限値が不明であることを表します。

4.3.2 MySQL データソースのコネクタプロパティの一覧

次の表は MySQL コネクタに設定できる特定のプロパティの一覧です。

プロパティ	説明
<code>datasourceCompCollation</code>	比較に使用するソース照合 (<code>LIKE / NOT LIKE</code> 、および関数評価以外)。これは SQL Server と MySQL でクエリに <code>collate</code> 句を追加するために使用されます。設定されていない場合には、これらの操作の <code>collate</code> 句は生成されません。デフォルトでは設定されていません。
<code>datasourceSortCollation</code>	並べ替え操作 (<code>ORDER BY</code>) に使用するソース照合。これは SQL Server と MySQL でクエリに <code>collate</code> 句を追加するために使用されます。設定されていない場合には、これらの操作の <code>collate</code> 句は生成されません。デフォルトでは設定されていません。
<code>datasourceBinaryCollation</code>	バイナリ照合での評価が必要な比較に使用するソース照合 (<code>LIKE / NOT LIKE</code> および関数評価)。これは SQL Server と MySQL で、バイナリ照合のセマンティクスが必要なクエリに <code>collate</code> 句を追加するために使用されます。設定されていない場合には、これらの操作の <code>collate</code> 句は生成されません。デフォルトでは設定されていません。
<code>unicodeStrings</code>	<code>True/Yes</code> または <code>False/No</code> 。データベースにプッシュされる文字列定数で Unicode 構文を使用するかどうかを指定します。デフォルト値: <code>False</code> 。

4.3.3 Teradata データソースのコネクタプロパティの一覧

次の表は Teradata コネクタに設定できる特定のプロパティの一覧です。

プロパティ	説明
<i>sampleSize</i>	SAMPLE 演算子を使用して、返す最大行数を定義します。

関連情報

[リレーショナルデータソースに対応したコネクタの capabilities の全一覧](#) [59 ページ]

4.3.4 Sybase ASE データソースのコネクタプロパティの一覧

次の表は Sybase ASE コネクタに設定できる特定のプロパティの一覧です。

プロパティ	説明
<i>setQuotedIdentifier</i>	True/Yes または False/No setQuotedIdentifier=true を指定すると、文字列識別子を囲む引用符には " が使用されます。

関連情報

[リレーショナルデータソースに対応したコネクタの capabilities の全一覧](#) [59 ページ]

4.3.5 SQL Server データソースのコネクタプロパティの一覧

次の表は SQL Server コネクタに設定できる特定のプロパティの一覧です。

プロパティ	説明
<i>datasourceCompCollation</i>	比較に使用するソース照合 (LIKE / NOT LIKE、および関数評価以外)。これは SQL Server と MySQL でクエリに collate 句を追加するために使用されます。設定されていない場合には、これらの操作の collate 句は生成されません。デフォルトでは設定されていません。 例 datasourceCompCollation=Latin1_general_ci_ai
<i>datasourceSortCollation</i>	並べ替え操作 (ORDER BY) に使用するソース照合。これは SQL Server と MySQL でクエリに collate 句を追加するために使用されます。設定されてい

プロパティ	説明
	<p>ない場合には、これらの操作の collate 句は生成されません。デフォルトでは設定されていません。</p> <p>例</p> <pre>datasourceSortCollation=Latin1_general_ci_as</pre>
<code>datasourceBinaryCollation</code>	<p>バイナリ照合での評価が必要な比較に使用するソース照合 (LIKE / NOT LIKE および関数評価)。これは SQL Server と MySQL で、バイナリ照合のセマンティクスが必要なクエリに collate 句を追加するために使用されます。設定されていない場合には、これらの操作の collate 句は生成されません。デフォルトでは設定されていません。</p> <p>例</p> <pre>datasourceBinaryCollation=Latin1_general_bin</pre>
<code>unicodeStrings</code>	<p>True/Yes または False/No。データベースにプッシュされる文字列定数で Unicode 構文を使用するかどうかを指定します。デフォルト値: False。</p>

4.3.6 汎用 ODBC または JDBC データソースのコネクタプロパティの一覧

次の表は、汎用コネクタに設定できる特定のプロパティの一覧です。

プロパティ	説明
<code>sqlDialect</code>	<p>データベースでサポートされる SQL ダイアレクトを特定します。以下のいずれかになります。</p> <ul style="list-style-type: none"> • <code>sql92</code> • <code>sql99</code> (今後の使用のために確保) • <code>jdbc3</code> (外部結合のために JDBC 構文が使用される) • <code>odbc</code> • <code>oracle</code> • <code>sqlserver</code> • <code>ids</code> (Informix Dynamic Server) • <code>teradata</code> • <code>maxdb</code> • <code>greenplum</code> • <code>postgresql</code> <p>パラメータ <code>sourceType</code> で特定されたとおりにソースによってサポートされている SQL ダイアレクトをデフォルトに設定します。<code>sourceType</code> が定義されていない場合には、<code>sql92</code> をデフォルトに設定します。</p>

プロパティ	説明
<code>supportsCatalog</code>	コネクタがカタログの概念をサポートするかどうかを示します。デフォルト値: <code>true</code> 。
<code>supportsSchema</code>	コネクタがスキーマの概念をサポートするかどうかを示します。デフォルト値: <code>true</code> 。
<code>supportsBoolean</code>	<p><code>True/Yes</code> または <code>False/No</code></p> <p><code>False</code>: JDBC ドライバまたはデータベースが最初のクラスオブジェクトとして論理値をサポートしない場合に指定します。このパラメータのデフォルト値はデータベースごとに異なります。データベースが、サポートされているソースタイプのいずれかである場合には、このパラメータはすでに適切な値に設定されています。ただし、その値を上書きすることができます。デフォルト値: <code>false</code>。</p>
<code>useIndexInOrderBy</code>	<p>送信されたクエリの <code>ORDER BY</code> 句で、エイリアス (列名) ではなく、インデックス (列ポジション) を使用する必要があるかどうかを指定します。デフォルト値: <code>false</code> (<code>ORDER BY</code> 句でエイリアスを適切に処理しないデータベースを除く)。</p> <p>例</p> <p>列 2 および 3 に <code>order by</code> を適用するには、<code>ORDER BY C2, C3</code> ではなく <code>ORDER BY 2, 3</code> を作成します。</p>
<code>escapeIdentifierQuoteString</code>	<p>引用符が</p> <p>(<code>java.sql.DatabaseMetaData#getIdentiferQuoteString</code> が返す) ID に含まれる場合、ID の引用符文字列をエスケープするために使用される文字列を定義します。デフォルトでは、このエスケープ文字列は ID 引用符文字列自体に設定されています。空に設定されていると、エスケープは実行されません。</p>
<code>ignoreKeys</code>	<p><code>True/Yes</code> または <code>False/No</code></p> <p><code>No</code>: キー/外部キーメタデータを取得するために、ラッパが JDBC ドライバをクエリしない場合に指定します。Sun JDBC-ODBC ブリッジはそのようなコールをサポートしないので、このオプションを <code>true</code> に設定する必要があります。デフォルト値: <code>No</code>。</p>
<code>supportsTemporaryTables</code>	<p><code>True/Yes</code> または <code>False/No</code></p> <p><code>False</code>: ソースが一時テーブルをサポートしていない場合、または一時テーブルの生成を無効にする場合に指定します。</p> <p>デフォルト値: <code>false</code>。</p>
<code>supportsTableCardinality</code>	<p><code>True/Yes</code> または <code>False/No</code></p> <p><code>False</code>: ソースがテーブルのカーディナリティを計算できないことを示します。</p>

プロパティ	説明
	デフォルト値: true。
<code>supportsColumnCardinality</code>	<p>True/Yes または False/No</p> <p>False: ソースが列のカーディナリティを計算できないことを示します。</p> <p>デフォルト値: true。</p>

関連情報

[リレーショナルデータソースに対応したコネクタの capabilities の全一覧](#) [59 ページ]

4.3.7 Oracle データソースのコネクタプロパティの一覧

次の表は Oracle コネクタに設定できる特定のプロパティの一覧です。

プロパティの種類	プロパティ値	プロパティの説明
<code>unicodeStrings</code>	True/Yes または False/No	データベースにプッシュされる文字列定数で Unicode 構文を使用するかどうかを指定します。デフォルト値: False。

4.3.8 SAP HANA データソースのコネクタプロパティの一覧

次の表は SAP HANA コネクタに設定できる特定のプロパティの一覧です。

プロパティの種類	プロパティ値	プロパティの説明
<code>unicodeStrings</code>	True/Yes または False/No	データベースにプッシュされる文字列定数で Unicode 構文を使用するかどうかを指定します。デフォルト値: False。

4.3.9 MaxDB データソースのコネクタプロパティの一覧

次の表は MaxDB コネクタに設定できる特定のプロパティの一覧です。

プロパティの種類	プロパティ値	プロパティの説明
unicodeStrings	True/Yes または False/No	データベースにプッシュされる文字列定数で Unicode 構文を使用するかどうかを指定します。デフォルト値: False。

4.4 SAS へのコネクタの設定

SAS へのコネクタを設定する前に、SAS のミドルウェアおよびドライバをインストールする必要があります。

SAS のミドルウェアおよびドライバの設定に関する詳細は、データアクセスガイドを参照してください。

4.4.1 SAS データソースのコネクタプロパティの一覧

次の表は SAS コネクタに設定できる特定のプロパティの一覧です。

プロパティ	説明
<i>maxConnections</i>	基のデータベースに対する同時接続の最大数です。0 は上限がないことを表します。デフォルト値: 0。
<i>maxConnectionIdleTime</i>	アイドル接続が接続プールに保持される最長時間です。単位はミリ秒です。0 は上限がないことを表します。デフォルト値: 60000 (60 秒)。
<i>maxPoolSize</i>	プールに保持するアイドル (自由な) 接続の最大数です。0 は上限がないことを表します。デフォルト値: 32。
<i>maxIdlePools</i>	アイドルのまま保持できるプールの最大数です。この値に達すると、最も古い未使用プールが閉じられ、削除されます。0 は上限がないことを表します。デフォルト値: 24。
<i>connectionTestQuery</i>	<p>基のデータベースへの接続が有効かどうかを確認するために使用できる SQL テストクエリです。注意: このクエリの実行は低負荷である必要があります。空の文字列は、テストクエリがないことを表します。デフォルト値: 空の文字列。</p> <p>例</p> <p>テストクエリ例は <code>SELECT 1 FROM DUAL</code> です。</p>
<i>connectionFailureDetectionOnError</i>	データベースによって <code>SQLException</code> が通知された場合に、実行する必要がある接続エラー検知の種類を示すキーワードです。

プロパティ	説明
<i>connectionFailureSQLStates</i>	<p>データベースによって <code>SQLException</code> が通知された場合に、接続エラーの検知に使用できる特定の <code>SQLState</code> コードの一覧です。ここでは、接続障害を示す標準コード (2 文字クラス 08 で始まる) を指定する必要はありません。Oracle の具体的なコード例は 61000: (ORA-00028: your session has been killed) です。要素は ; 記号で区切られます (要素間にスペースは入れません)。デフォルト値: 空。</p>
<i>driverProperties</i>	<p>ドライバのプロパティ一覧です。プロパティは、; 文字を使用して区切る必要があります (プロパティの間にはスペースを入れません)。</p> <p>入力できるプロパティは、データベースへの接続に使用しているドライバに使用できるものと同じプロパティです。プロパティの一覧についてはドライバのマニュアルを参照してください。</p> <p>例</p> <pre>driverProperties=selectMethod \=cursor;connectionRetryCount\=2</pre>
<i>sessionProperties</i>	<p>データベースに設定されたセッションプロパティの一覧。プロパティは、; 文字を使用して区切る必要があります (プロパティの間にはスペースを入れません)。</p> <p>入力できるプロパティは、接続しているデータベースに使用できるものと同じプロパティです。プロパティの一覧についてはデータベースのマニュアルを参照してください。</p> <p>例</p> <pre>sessionProperties=selectMethod \=cursor;connectionRetryCount\=2</pre>
<i>capabilities</i>	<p>データベースがサポートするすべての機能の一覧です。要素は ; 記号で区切られます (要素間にスペースは入れません)。</p> <p>例</p> <pre>capabilities=fullSQL\=true;outerjoin \=false;rightouterjoin\=true</pre>
<i>useParameterInlining</i>	<p><code>true</code> に設定すると、JDBC ラッパはパラメータ化されたクエリの実行に <code>java.sql.PreparedStatement</code> オブジェクトを使用せず、代わりに <code>java.sql.Statement</code> オブジェクトを使用します。定数値のプレースホルダの代わりに、パラメータ化されたクエリが埋め込まれます。このオプションは、手の込んだ文をサポートしていない JDBC ドライバに役立ちます。デフォルト値: <code>false</code>。</p>
<i>castColumnType</i>	<p><code>databasetype=jdbctype</code> タイプマッピングの一覧。ドライバによるデフォルトのマッピングが不正確または不完全である場合に、このパラメータは便利です。</p>

プロパティ	説明
	<p>注: 正式にサポートされているデータベースの場合には、タイプマッピングは暗黙的に設定されていますが、それらを上書きできます。</p> <p>例</p> <p>Oracle JDBC ドライバの場合 <code>castColumnType=FLOAT\=FLOAT;BLOB\=BLOB</code></p>
<code>enableUpdateQueries</code>	<p>True/Yes または False/No</p> <p>更新クエリの実行が有効かどうかを指定します。</p> <p>デフォルト値: True。</p>
<code>enableTemporaryTableQueries</code>	<p>True/Yes または False/No</p> <p>一時テーブルクエリの実行が有効かどうかを指定します。</p> <p>デフォルト値: True。</p>
<code>defaultFetchSize</code>	<p><code>java.sql.Statement</code> の作成時に設定するデフォルトフェッチサイズ。0 は、フェッチサイズが設定されていないことを表します。</p> <p>さらに多くの行が必要な場合に、データベースから取得する行数のヒントをコネクタに示します。</p> <p>デフォルト値: 0 (フェッチサイズ未設定)</p>
<code>compCollationCompatible</code>	<p>True/Yes または False/No</p> <p>データソースでの比較操作の照合がデータフェデレーションサービスの現在の設定に適合できるかどうかを指定します。true に設定すると、サーバは比較操作の照合を無視し、述語はソースへ安全にプッシュされます。デフォルト値: false。</p> <p>例</p> <p><code>compCollationCompatible=true</code></p>
<code>sortCollationCompatible</code>	<p>True/Yes または False/No</p> <p>データソースでの並べ替え操作 (ORDER BY) の照合がデータフェデレーションクエリサービスの現在の設定に適合できるかどうかを指定します。true に設定すると、サーバは並べ替え操作の照合を無視し、(ORDER BY) 式はソースへ安全にプッシュされます。デフォルト値: false。</p> <p>例</p> <p><code>sortCollationCompatible=true</code></p>
<code>datasourceCompCollation</code>	<p>比較に使用するソース照合 (LIKE / NOT LIKE、および関数評価以外)。これは SQL Server と MySQL でクエリに <code>collate</code> 句を追加するために使用さ</p>

プロパティ	説明
	<p>れます。設定されていない場合には、これらの操作の <code>collate</code> 句は生成されません。デフォルトでは設定されていません。</p> <p>例</p> <pre>datasourceCompCollation=Latin1_general_ci_ai</pre>
<code>datasourceSortCollation</code>	<p>並べ替え操作 (<code>ORDER BY</code>) に使用するソース照合。これは SQL Server と MySQL でクエリに <code>collate</code> 句を追加するために使用されます。設定されていない場合には、これらの操作の <code>collate</code> 句は生成されません。デフォルトでは設定されていません。</p> <p>例</p> <pre>datasourceSortCollation=Latin1_general_ci_as</pre>
<code>datasourceBinaryCollation</code>	<p>バイナリ照合での評価が必要な比較に使用するソース照合 (<code>LIKE / NOT LIKE</code> および関数評価)。これは SQL Server と MySQL で、バイナリ照合のセマンティクスが必要なクエリに <code>collate</code> 句を追加するために使用されます。設定されていない場合には、これらの操作の <code>collate</code> 句は生成されません。デフォルトでは設定されていません。</p> <p>例</p> <pre>datasourceBinaryCollation=Latin1_general_bin</pre>
<code>sqlDialect</code>	<p>データベースでサポートされる SQL ダイアレクトを特定します。以下のいずれかになります。</p> <ul style="list-style-type: none"> • <code>sql92</code> • <code>sql99</code> (今後の使用のために確保) • <code>oracle</code> • <code>sqlserver</code> • <code>jdbc3</code> (外部結合のために JDBC 構文が使用される) • <code>sas</code> <p>パラメータ <code>sourceType</code> で特定されたとおりにソースによってサポートされている SQL ダイアレクトをデフォルトに設定します。<code>sourceType</code> が定義されていない場合には、<code>sql92</code> をデフォルトに設定します。</p>
<code>useIndexInOrderBy</code>	<p>送信されたクエリの <code>ORDER BY</code> 句で、エイリアス (列名) ではなく、インデックス (列ポジション) を使用する必要があるかどうかを指定します。デフォルト値: <code>false</code> (<code>ORDER BY</code> 句でエイリアスを適切に処理しないデータベースを除く)。</p> <p>例</p> <p>列 2 および 3 に <code>order by</code> を適用するには、<code>ORDER BY C2, C3</code> ではなく <code>ORDER BY 2, 3</code> を作成します。</p>

プロパティ	説明
<code>escapeIdentifierQuoteString</code>	引用符が (<code>java.sql.DatabaseMetaData#getIdentiferQuoteString</code> が返す) ID に含まれる場合、ID の引用符文字列をエスケープするために使用される文字列を定義します。デフォルトでは、このエスケープ文字列は ID 引用符文字列自体に設定されています。空に設定されていると、エスケープは実行されません。
<code>ignoreKeys</code>	True/Yes または False/No No: キー/外部キーメタデータを取得するために、ラッパが JDBC ドライバをクエリしない場合に指定します。Sun JDBC-ODBC ブリッジはそのようなコールをサポートしないので、このオプションを <code>true</code> に設定する必要があります。 デフォルト値: No。
<code>transactionIsolation</code>	トランザクション孤立レベル。以下のいずれかになります。 <ul style="list-style-type: none">TRANSACTION_READ_COMMITTEDTRANSACTION_READ_UNCOMMITTEDTRANSACTION_REPEATABLE_READTRANSACTION_SERIALIZABLE デフォルト値: 設定されていません。
<code>setFetchForwardDirection</code>	True/Yes または False/No True: fetch forward を明示的に設定する必要がある場合。デフォルト値: False。
<code>setReadOnly</code>	True/Yes または False/No False: 接続が読み取り専用になりません。デフォルト: False。
<code>metadataFetchMode</code>	SAS データソースのみに使用されるメタデータフェッチモード。 以下のいずれかになります。 <ul style="list-style-type: none">eager: すべてのメタデータが 1 回で取得されます。lazy: メタデータは必要に応じて取得されます。 デフォルト値: lazy。
<code>sasWeights</code>	SAS ダイアレクトでのクエリ生成時に FROM 句でのテーブル順序の決定に使用されるテーブル名とその加重との間のマッピング。FROM 句のテーブルは、加重に基づいて降順に並べられます。デフォルトでは加重はテーブルカーディナリティに設定されていますが、このパラメータを使用して上書きできます。この順序が実行されるのは内部結合のみです。 ここでのテーブル名はラッパでエクスポートされた名前です。加重は長い値です。

プロパティ	説明
	<p>このパラメータが指定されていない場合、または特定のテーブルに加重が定義されていない場合には、デフォルトで加重は (データフェデレーションサービスに設定されたとおり) テーブルのカーディナリティとなります。</p> <p>テーブル名が不明である場合には、単に無視されます。</p> <p>このパラメータが考慮されるのは、パラメータが <code>sqlDialect="sas"</code> である場合に限られます。</p> <p>例</p> <pre>sasWeights=EMPLOYEE\=16;DEPARTMENT\=4</pre> <p>この設定により、2 つのテーブルの結合で SAS へのクエリをプッシュする場合は、EMPLOYEE テーブルが表示された後に DEPARTMENT テーブルが表示されます。</p>
<code>addCompensationPredicates</code>	<p>True/Yes または False/No</p> <p>False: 特定の動作の欠如を相殺する述語の生成を無効にする場合に指定します。</p> <p>相殺する述語とは、null 値に対して SQL セマンティクスを要求するために追加される <code>IS NOT NULL</code> 条件です (SAS は SQL 形式ではありません)。</p> <p>このような余分な条件は、パフォーマンスに影響を与える場合があります。null 値に対して SQL セマンティクスを要求することに消極的な場合は、このパラメータを <code>false</code> に設定できます。</p> <p>このパラメータが考慮されるのは、パラメータが <code>sqlDialect="sas"</code> である場合に限られます。</p> <p>デフォルト値: True。</p>
<code>trimTrailingSpaces</code>	<p>True/Yes または False/No</p> <p>一部の JDBC ドライバは余白を詰めたメタデータを返します。このパラメータを <code>Yes</code> に設定すると、カタログ、スキーマ、テーブル、列、キー、および外部キーの名前から余分なスペースが削除されます。このパラメータのデフォルト値は <code>No</code> です。</p>
<code>maxValuesInInClause</code>	<p>IN 句の値の最大数を指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<code>semiJoinMaxQueries</code>	<p>準結合演算子が実行できるクエリの最大数を指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<code>semiJoinTempTableMinCardinality</code>	<p>準結合演算子の一時テーブル方針に、ディメンションの最小カーディナリティを指定します。</p>

プロパティ	説明
	デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。
<code>semiJoinTempTableMaxCardinality</code>	<p>準結合演算子の一時テーブル方針に、ディメンションの最大カーディナリティを指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<code>semiJoinExecutionStrategies</code>	<p>希望の順番で、準結合演算子の実行方針の一覧を指定します。</p> <p>次のいずれかの値です。</p> <ul style="list-style-type: none"> • DEFAULT DEFAULT は、システムパラメータ <code>SEMI_JOIN_EXECUTION_STRATEGIES</code> の値が使用されることを表します。 • I、T、P をカンマで区切って希望の順番で組み合わせたものです。I は IN クエリの実行方針、T は一時テーブルの実行方針、P はパラメータ化されたクエリの実行方針を表します。 例 <ul style="list-style-type: none"> ○ T,P,I ○ I,T ○ P ○ 方針なし <p>I、T、または P のうちのいずれかが指定されていない場合、対応する実行方針はラップでサポートされません。</p> <p>注: NONE は、ラップがサポートする実行方針がないことを表します。</p> <p>戻り値には、null も空の文字列も含まれません。</p>

4.4.2 カーディナリティを基準にして FROM 句内のテーブルの順序を並べ替えることによる SAS クエリの最適化

SAS は、FROM 句内のテーブルの順序によって影響を受けます。SAS/Share サーバからの応答を速めるためには、カーディナリティに従って、FROM 句内のテーブル名を降順に並べ替える必要があります。

データフェデレーションアプリケーション内の統計情報を正確に維持することによって、データフェデレーションアプリケーションでテーブルがこの順序で生成されることが確認できます。これは、データフェデレーション管理ツールを使用して実行できます。

テーブルの順序は、SAS JDBC コネクタの `sasWeights` リソースプロパティを設定することで手動で制御することもできます。

4.5 SAP BW のコネクタの設定

4.5.1 SAP BW データソースのコネクタプロパティの一覧

次の表は SAP BW コネクタに設定できる特定のプロパティの一覧です。

プロパティ	説明
packageSize	<p>コールバックに関するパッケージのサイズ。</p> <p>データフェデレーションクエリエンジンに対して、SAP BW から返されるパッケージごとの行の数です。</p> <p>デフォルト値は 200 です。</p> <p>例</p> <pre>packageSize=300</pre>
programIDMapping	<p>SAP BW が Data Federator との接続に使用するコールバックのプログラム ID を定義します。ID はサーバ名とプログラム ID のマッピングリストとして提供されます。リストの形式は、' ' で区切られたキーと値を含む文字列です。キーはサーバ名を表し、値はプログラム ID を表します。各 ID は、SAP BW 上で作成された RFC 出力先の名前に一致している必要があります。</p> <p>このプロパティが定義されていない場合、Data Federator によって自動的に RFC 出力先が作成されます。RFC 出力先名と同一のプログラム ID を使用して、RFC 出力先が作成されます。</p> <p>作成される RFC 出力先の形式は <code><DF_JCO_>+<hostname>+<_>+<counter></code> です。<code><ホスト名></code> は、ローカルホストの名前であり、<code><カウンタ></code> は 0 から 9 の順に進みます。ただし、<code><ホスト名></code> の最大の長さは 23 です。ローカルホスト名が 24 文字以上の場合は、先頭の 23 文字のみが RFC 出力先名に使用されます。</p> <p>このプロパティが定義されていても、現在のサーバにマッピングリストがない場合はエラーが返されます。</p> <p>このプロパティにはデフォルト値はありません。このため、自動モードが使用されます。</p> <p>例 1</p> <pre>MySIA.AdaptiveProcessingServer=RFC1</pre> <p>例 2</p> <pre>MySIA.DFServer1=RFC1;MySIA.DFServer2=RFC2;...</pre>

プロパティ	説明
<i>useBinaryXML</i>	<p><i>true</i> に設定すると、コネクタと SAP BW サーバの間で交換されるデータは、プレーンテキスト形式ではなくバイナリ XML 形式で圧縮されます。この結果、パフォーマンスが向上します。デフォルトではその値は <i>true</i> です。</p> <p>例</p> <pre>useBinaryXML=true</pre>
<i>checkUnits</i>	<p><i>true</i> に設定すると、単位なしのメジャーを使用するクエリは拒否され、エラーが発生します。デフォルトではその値は <i>false</i> であり、単位なしでメジャーを使用できます。</p> <p>例</p> <pre>checkUnits=false</pre>
<i>forcedCapabilities</i>	<p>SAP BW の代わりにデータフェデレーションクエリエンジンが操作を実行することを希望する場合は、SAP BW コネクタの機能を意図的に制限できます。</p> <p>データフェデレーションクエリエンジンから SAP BW に依頼する機能を入力します。</p> <p>有効な値は次のとおりです。</p> <ul style="list-style-type: none"> • <i>SCAN_ONLY</i> データフェデレーションクエリエンジンは、スキャンのみを SAP BW に依頼します。 • <i>PROJECTIONS_ONLY</i> データフェデレーションクエリエンジンは、投影のみを SAP BW に依頼します。 • <i>empty</i> - データフェデレーションクエリエンジンは、すべての有効な操作を SAP BW に依頼します。 <p>例</p> <pre>forcedCapabilities=SCAN_ONLY</pre>
<i>jcoDestinationProperties</i>	<p>JCO 出力先プロパティの一覧。複数のプロパティを区切るには、文字 ; を使用します (スペースを入力しないでください)。</p> <p>使用可能なプロパティは、JCo API 用の API リファレンスのうち、<i>DestinationDataProvider</i> インタフェースの箇所に掲載されています。</p>
<i>jcoServerProperties</i>	<p>JCO サーバプロパティの一覧。複数のプロパティを区切るには、文字 ; を使用します (スペースを入力しないでください)。</p> <p>使用可能なプロパティは、JCo API 用の API リファレンスのうち、<i>ServerDataProvider</i> インタフェースの箇所に掲載されています。</p>
<i>authorityCheck</i>	<p>SAP BW の中で、権限をチェックするかどうかを示すパラメータを設定します。</p> <p>SAP BW 内のパラメータは、SAP BW が次のことを実行するかどうかを指定します。</p>

プロパティ	説明
	<p>要求したデータを表示する権限がユーザアカウントにあるかどうかをチェックする (<i>read</i>)、または権限をまったくチェックしない (<i>none</i>)。</p> <ul style="list-style-type: none"> • <i>true</i> (デフォルト): SAP BW は <i>read</i> 権限をチェックします。 • <i>false</i>: SAP BW は権限をチェックしません。
<i>pingTimeout</i>	<p>SAP サーバに対して ping を実行するときに使用する、ミリ秒単位のタイムアウト値です。デフォルト値は 10000 ミリ秒です。</p> <p>例</p> <pre>pingTimeout=60000</pre>
<i>maxValuesInInClause</i>	<p>IN 句の値の最大数を指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<i>semiJoinMaxQueries</i>	<p>準結合演算子が実行できるクエリの最大数を指定します。</p> <p>デフォルト値: 0。0 は制限がないか、または制限値が不明であることを表します。</p>
<i>semiJoinExecutionStrategies</i>	<p>希望の順番で、準結合演算子の実行方針の一覧を指定します。</p> <p>次のいずれかの値です。</p> <ul style="list-style-type: none"> • DEFAULT DEFAULT は、システムパラメータ SEMI_JOIN_EXECUTION_STRATEGIES の値が使用されることを表します。 • I、T、P をカンマで区切って希望の順番で組み合わせたものです。I は IN クエリの実行方針、T は一時テーブルの実行方針、P はパラメータ化されたクエリの実行方針を表します。 例 <ul style="list-style-type: none"> ○ T,P,I ○ I,T ○ P ○ 方針なし <p>I、T、または P のうちのいずれかが指定されていない場合、対応する実行方針はラッパでサポートされません。</p> <p>注: NONE は、ラッパがサポートする実行方針がないことを表します。</p> <p>戻り値には、null も空の文字列も含まれません。</p> <p>注: 実行方針の T および P は、SAP BW コネクタでサポートされません。</p>
<i>enableAuthorizationsFiltering</i>	<p><i>true</i> に設定すると認証フィルタリングが有効になります。現在のユーザに定義された SAP BI 権限が満たされるよう、コネクタによりフィルタが自動的に追加されます。認証フィルタリングが無効の場合、ユーザが権限のないデータを使用しようとするとエラーが発生します。デフォルト値は <i>false</i> です。</p>

プロパティ	説明
	<p>例</p> <pre>enableAuthorizationsFiltering=true</pre>
<code>debugReportPrefix</code>	<p>ABAP レポートプログラムの名前に含まれる最長 11 文字の文字列 生成されたプログラムの名前は、Z_RSDDRI_DF_TXT_\${debugReportPrefix}_ID または Z_RSDDRI_DF_DBG_\${debugReportPrefix}_ID で、ID はラッパ側で生成された 3 桁の数値です。生成されたプログラムは、SAP の専門家により Data Federator のファサードのバグを再現するために使用されます。</p> <p>設定していない場合、プログラムは生成されません。</p> <p>例</p> <pre>MY_HOSTNAME</pre>
<code>gatewayHostname</code>	<p>SAP BW ゲートウェイをホストしているマシンの名前。</p> <p>指定していない場合、RFC が実行され SAP BW により値が選択されます。</p> <p>例</p> <pre>gatewayHostname=server.wdf.sap.corp</pre>
<code>gatewayServiceName</code>	<p>SAP BW ゲートウェイサービスの名前またはポート番号。</p> <p>指定していない場合、RFC が実行され SAP BW により値が選択されます。</p> <p>例</p> <pre>gatewayServiceName=sapgw50</pre> <p>例</p> <pre>gatewayServiceName=3350</pre>

4.5.2 SAP BW がデータフェデレーションサービスとの接続に使用するコールバック ID を手動で設定

SAP BW ではデータフェデレーションサービスに接続するためにコールバック ID を使用します。コールバックは、SAP BW コネクタの最初のクエリが実行されたときに自動的に登録されますが、組織のセキュリティポリシーに従うためなどの理由で、変更する必要がある場合があります。

1. SAP Logon を開いて SAP システムにログオンします。
2. トランザクションテキストフィールドに「rz37」を入力して **実行** をクリックします。
3. 関数モジュール「**RSDDRI_DF_CONFIGURE**」を入力して、**実行** をクリックします。

パラメータパネルが開きます。

4. 以下のようにパラメータを設定します。

<code>_ONLY_CHECK</code>	" (空)
<code>_RFC_DESTINATION</code>	DF_JCO_ <code><some-hostname></code> _ <code><some-sid></code>
<code>_REMOVE_CONFIGURATION</code>	" (空)

2 番目のパラメータには、`<some-hostname>` に、データフェデレーションサービス実行用サーバをインストールしたコンピュータのホスト名が入ります。

同じ `<some-hostname>` 値を持つ複数の接続を区別するために `_<some-sid>` を一意のシステム識別子として使用します。

この場合、DF_JCO_ `<MYHOST>` は一意の識別子であり、データフェデレーション管理ツールで再利用する必要があります。

5. モジュールを実行します。

i 注記

RFC の出力先がすでに存在しますというメッセージが表示されても問題ありません。

[`_ONLY_CHECK`] チェックボックスにチェックが入っていないことを確認します。

6. [システム] をクリックし、次に [ログオフ] をクリックします。
7. データフェデレーション管理ツールを起動し、管理者権限のあるユーザアカウントを使用してログオンします。
8. **コネクタ設定** タブを使用して SAP BW コネクタを編集します。
9. `programIDMapping` プロパティに、SAP BW で使用した文字列 `_RFC_DESTINATION` (プログラム ID) とサーバとのマッピングを追加します。

この場合、`programIDMapping` プロパティの値は、`MySIA.AdaptiveProcessingServer=DF_JCO_MYHOST` です。

詳細については、SAP BW コネクタプロパティの一覧にある `programIDMapping` プロパティの説明を参照してください。

10. テーブルでクエリを実行して、データが使用できるかテストします。

関連情報

[データフェデレーション管理ツールのコネクタのプロパティの変更](#) [34 ページ]

[SAP BW データソースのコネクタプロパティの一覧](#) [49 ページ]

4.5.3 SAP BW 接続のコールバックの ID の削除

現時点では、`callbackProgramID` の最大数は 10 です。正常な実行中 (データフェデレーションサービスを実行中のサーバが突然中止されなかった場合) に、`callbackProgramID` は、SAP サーバから自動的に削除されます。

サーバ上でそれ以上 `callbackProgramID` を生成できない (すべて使用した) ときにはエラーが発生します。以下は、システムが突然中止されたためにコールバック名が自動的に削除されなかった場合の、削除手順です。

1. SAP BW サーバにログインします。
2. トランザクション `sm59` を入力します。
3. [\[TCP/IP 接続\]](#) をクリックします。
4. 各接続 (`<DF_JCO_MYHOST_0>` から `<DF_JCO_MYHOST_9>` まで) をクリックし、削除アイコンをクリックします。

4.5.4 SAP 分析権限を活用したデータの自動フィルタ

SAP BW で実行されるクエリでは常にデータベースからデータセットを選択します。このデータに権限関連の特性が含まれる場合は、選択を実行するために十分な権限を、クエリを実行するユーザに付与する必要があります。そうしないと、権限が十分でないことを示すエラーメッセージが返されます。

データフェデレーションサービスで分析権限を活用するように設定し、キューブの一部となっている各特性に認証済みの値のフィルタを自動的に追加することができます。SAP Business Intelligence プラットフォームに接続する各ユーザに基づいて簡単にデータをフィルタすることができるため、この機能は、シングルサインオン認証モードを使用して SAP BW に接続する場合に特に役立ちます。

機能の有効化


この機能は、以下の SAP BW コネクタリソースプロパティを設定することで有効にすることができます。

- `enableAuthorizationsFiltering`: true/false

データソースによって使用されるリソースを設定するには、データフェデレーション管理ツールを使用します。

前提条件

この機能は、*Data Federator* ファサードと呼ばれる SAP BW のコンポーネントを使用します。

以下の SAP ノートを読んで SAP BW 側の前提条件を確認してください: <https://service.sap.com/sap/support/notes/1500945> 

原則

SAP BW データウェアハウスワークベンチで権限関連として宣言されたすべての特性が、自動的に追加するフィルタセットの計算に考慮されます。

クエリに明示的なフィルタが含まれる場合は、それ以外の権限フィルタは追加されません。

クエリに権限関連の特定の特性 (列) のフィルタが含まれない場合は、フィルタが自動的に追加され認証済みの値が返されません。

階層権限

データフェデレーションサービスは階層を公開しません。しかし、データをフィルタリングし認証済みの値を返す際には、階層に対する分析権限が考慮されます。

コロンの権限

値に基づく権限に加え集計権限 (コロンの権限とも呼ばれます) が定義されている場合、データフェデレーションサービスでは、列が SQL クエリの SELECT 句にない場合でも、常に値に基づく権限が考慮されます。

集計権限のロジックの詳細については、SAP ノート 1140831 - Colon authorization during query execution (<https://service.sap.com/sap/support/notes/1140831>) を参照してください。

複数階層に対する権限

特性の値が複数階層を持ち、それぞれの階層に権限がある場合は、結合が実行され、すべての階層のすべての認証済みの値が返されます。

SAP Business Explorer (BEx) との比較

SAP BW のデスクトップアプリケーション BEx Query Designer で、クエリを定義して分析権限に基づいてフィルタを追加することができます。ただし、BEx Query Designer と SAP BI データフェデレーションサービスでは以下の点が異なります。

- BEx Query Designer を使用する場合は、クエリデザイナーで権限変数を使用することで権限フィルタを適用する特性を選択的に指定することができます。データフェデレーションサービスでは、SAP BW データウェアハウスワークベンチで権限関連として宣言されているすべての特性に権限フィルタが適用されます。
- データフェデレーションサービスでは、SQL クエリが特定の特性に対する明示的なフィルタを含んでいる場合、この特性には権限フィルタが適用されず、明示的なフィルタのみが使用されます。BEx Query Designer では、権限フィルタとユーザによって追加された明示的なフィルタを組み合わせることができます。

SAP BW に関する SAP ノート

SAP ノート 1578089: DBIF: Adding authorizations to filter despite aggregation (<https://service.sap.com/sap/support/notes/1578089>) を参照してください。



例

権限フィルタリング

BW のインフォキューブ ZCUBE1 には 6 行のみが含まれ、データソース BW_ZCUBE1 は SAP Business Intelligence プラットフォームで SSO を有効にしてアクセスするように設定されています。フル権限を持つユーザーがファクトテーブルを読み取る場合に、以下の SQL クエリを実行するとします。

```
select ZCHA1, ZCHA2, ZCHA3, ZKYF1
from /DF_PROJECT/sources/BW_ZCUBE1/IZCUBE1
```

結果は、次の通りです。

ZCHA1	ZCHA2	ZCHA3	ZKYF1
A	69226	2001	250.0
A	69226	2000	300.0
B	69190	2001	150.0
B	69190	2000	450.0
C	69115	2001	200.0
C	69115	2000	100.0

次に、別のユーザー JOE は、ZCHA1 および ZCHA3 に対するフル権限を持っていますが、特性 ZCHA2 に対しては値 69190 に対してのみ権限を持っているとします。JOE がこの機能を有効にせずに同じ SQL クエリを実行した場合、User JOE does not have authorization for InfoProvider ZCUBE1 というエラーメッセージが表示されます。

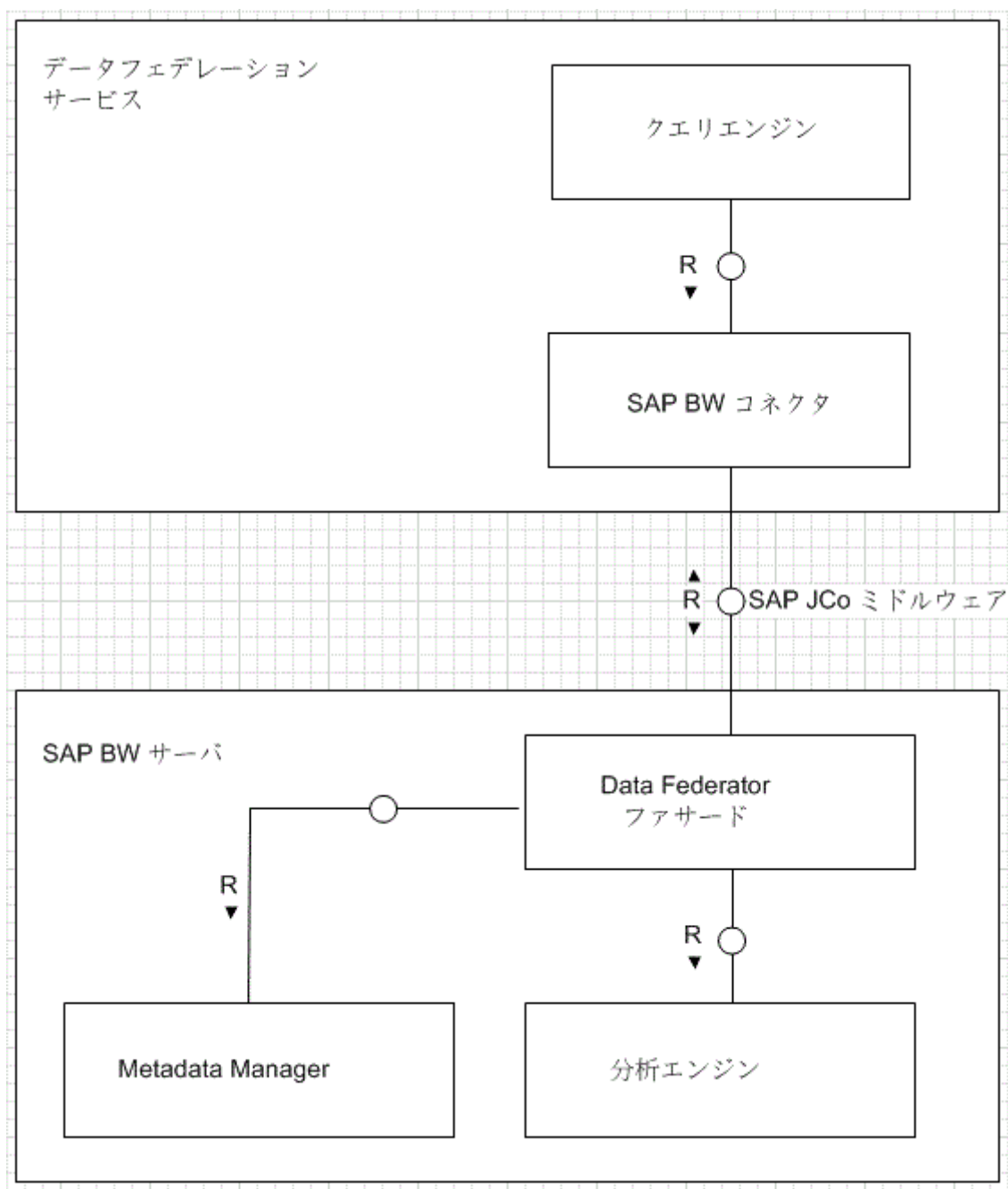
この機能を有効にすると、SAP Business Intelligence プラットフォームは認証済みのすべてのデータを返します。つまり、SQL クエリの結果は以下のようになります。

ZCHA1	ZCHA2	ZCHA3	ZKYF1
B	69190	2001	150.0
B	69190	2000	450.0

実際は、この結果は以下に示す ZCHA2 に対して明示的なフィルタを適用した SQL の結果とまったく同じです。

```
select ZCHA1, ZCHA2, ZCHA3, ZKYF1
from /DF_PROJECT/sources/BW_ZCUBE1/IZCUBE1 where ZCHA2 = 69190
```


4.5.5 複数のソースユニバースでの SAP BW 接続のアーキテクチャ



3. [設定プロパティ] タブで、[機能] 行の中にある [値] セルをクリックし、`my-capability=true`；フォームに機能を入力します。

複数の機能がセミコロン (;) で区切られていることを確認します。ほとんどの機能では、`true` または `false` という値を使用できます。

関連情報

[リレーショナルデータソースに対応したコネクタの capabilities の全一覧](#) [59 ページ]

4.7 リレーショナルデータソースに対応したコネクタの capabilities の全一覧

以下の表はコネクタの capabilities の一覧です。これらを使用して、capabilities と命名されたリソースプロパティを設定できます。

i 注記

`fullsql` は、すべての capabilities をデフォルトで `true` に設定できる特別な capabilities です。その後、必要に応じて、各 capabilities を個別に `false` に設定できます。

capabilities	説明
<code>fullsql</code>	デフォルトではすべての capabilities を <code>true</code> に設定します。その後、必要に応じて、各 capabilities を個別に <code>false</code> に設定できます。
<code>project</code>	コネクタが プロジェクション 操作をサポートするかどうかを指定します。
<code>orderby</code>	コネクタが <code>order by</code> 操作をサポートするかどうかを指定します。
<code>orderbystrings</code>	コネクタが文字列で <code>order by</code> 操作をサポートするかどうかを指定します。
<code>distinct</code>	コネクタが <code>distinct</code> 操作をサポートするかどうかを指定します。
<code>union</code>	コネクタが <code>union distinct</code> 操作をサポートするかどうかを指定します。
<code>unionall</code>	コネクタが <code>union all</code> 操作をサポートするかどうかを指定します。
<code>join</code>	コネクタが <code>join</code> 操作をサポートするかどうかを指定します。
<code>outerjoin</code>	コネクタが <code>full outer join</code> 操作をサポートするかどうかを指定します。
<code>leftouterjoin</code>	コネクタが <code>left outer join</code> 操作をサポートするかどうかを指定します。

capabilities	説明
<i>rightouterjoin</i>	コネクタが <i>right outer join</i> 操作をサポートするかどうかを指定します。
<i>aggregate</i>	コネクタが集計をサポートするかどうかを指定します。
<i>aggregatedistinct</i>	コネクタが <i>distinct</i> 句を伴う集計をサポートするかどうかを指定します。
<i>minaggregate</i>	コネクタが <i>min</i> 集計関数をサポートするかどうかを指定します。
<i>maxaggregate</i>	コネクタが <i>max</i> 集計関数をサポートするかどうかを指定します。
<i>countaggregate</i>	コネクタが <i>count</i> 集計関数をサポートするかどうかを指定します。
<i>avgaggregate</i>	コネクタが <i>average</i> 集計関数をサポートするかどうかを指定します。
<i>sumaggregate</i>	コネクタが <i>sum</i> 集計関数をサポートするかどうかを指定します。
<i>minaggregatedistinct</i>	コネクタが <i>distinct</i> 句を伴う <i>min</i> 集計関数をサポートするかどうかを指定します。
<i>maxaggregatedistinct</i>	コネクタが <i>distinct</i> 句を伴う <i>max</i> 集計関数をサポートするかどうかを指定します。
<i>countaggregatedistinct</i>	コネクタが <i>distinct</i> 句を伴う <i>count</i> 集計関数をサポートするかどうかを指定します。
<i>avgaggregatedistinct</i>	コネクタが <i>distinct</i> 句を伴う <i>average</i> 集計関数をサポートするかどうかを指定します。
<i>sumaggregatedistinct</i>	コネクタが <i>distinct</i> 句を伴う <i>sum</i> 集計関数をサポートするかどうかを指定します。
<i>equalitypredicate</i>	コネクタが等号述語をサポートするかどうかを指定します。
<i>comparisonpredicate</i>	コネクタが不等号述語をサポートするかどうかを指定します。
<i>likepredicate</i>	コネクタが <i>like</i> 述語をサポートするかどうかを指定します。
<i>nullpredicate</i>	コネクタが <i>is null</i> 述語をサポートするかどうかを指定します。
<i>inpredicate</i>	コネクタが <i>in</i> 述語をサポートするかどうかを指定します。
<i>arithmeticevaluation</i>	コネクタが算術操作をサポートするかどうかを指定します。
<i>booleanevaluation</i>	コネクタが論理型操作をサポートするかどうかを指定します。
<i>constantevaluation</i>	コネクタがリテラルをサポートするかどうかを指定します。
<i>emptystringevaluation</i>	コネクタが空の文字列のリテラルをサポートするかどうかを指定します。
キャンセル	コネクタがクエリ実行のキャンセルをサポートするかどうかを指定します。

capabilities	説明
<i>shareconcurrentstatements</i>	コネクタが単一接続で複数のクエリ (文) を共有できるかどうかを指定します。
<i>functionevaluation</i>	コネクタが関数評価をサポートするかどうかを指定します。

5 システムパラメータとセッションパラメータの管理

5.1 システムパラメータとセッションパラメータの概要

Data Federator には、システムとセッションの 2 つのレベルのパラメータがあります。

システムパラメータは、データフェデレーションクエリエンジンの実行中のインスタンスが共有します。

セッションパラメータは、1 つの接続に対して定義されます。これらのパラメータの値は、接続ごとに異なってもかまいません。

各セッションパラメータは、同じ名前のシステムパラメータからデフォルト値を取得します。セッションパラメータに対応するシステムパラメータの値を変更すると、新しい値は新しいセッションでのみ認識されます。

システムパラメータとセッションパラメータを使用して、以下のようなデータフェデレーションクエリエンジンのさまざまな面を設定することができます。

- メモリの使用
- ネットワークの使用
- クエリの実行順序
- 最適化

5.2 データフェデレーション管理ツールを使用したシステムパラメータの変更

1. パラメータを管理するためにデータフェデレーション管理ツールのインタフェースにアクセスするには、データフェデレーション管理ツールにログインし、[システムパラメータ] タブをクリックします。
2. パラメータのある行で、[現在の値] ボックスに新しい値を入力して、`Enter` キーを押します。

5.3 データフェデレーション管理ツールを使用したセッションパラメータの変更

1. パラメータを管理するためにデータフェデレーション管理ツールのインタフェースにアクセスするには、データフェデレーション管理ツールにログインし、[システムパラメータ] タブをクリックし、[セッションパラメータ] タブをクリックします。
2. パラメータのある行で、[現在の値] ボックスに新しい値を入力して、`Enter` キーを押します。

5.4 データフェデレーション管理ツールを使用したリレーショナルコネクタ機能および SAS コネクタ機能の設定


コネクタ機能の中には、データソースがどのような種類の演算子をサポートしているか、などの事項が含まれます。

データフェデレーションクエリエンジンが操作を自ら実行するか、またはデータソースに依頼するかを選択するように、コネクタ機能を設定できます。

一般的に、操作をデータベースシステムに依頼する方がより効率的ですが、すべてのデータベースシステムが同じ演算子をサポートしているわけではありません。機能のリストは、各演算子をどのデータソースに依頼できるかをデータフェデレーションクエリエンジンに伝えます。この演算子の依頼を通常はプッシュと呼びます。

i 注記

リレーショナルコネクタ機能または SAS コネクタ機能のいずれかのみを設定できます。

1. データフェデレーション管理ツールで、[\[コネクタの設定\]](#) タブをクリックします。
2. ツリーリスト内で使用中のコネクタを右クリックし、次に  [設定の作成](#) をクリックします。
3. [\[設定プロパティ\]](#) タブで、[\[機能\]](#) 行の中にある [\[値\]](#) セルをクリックし、`my-capability=true`; フォームに機能を入力します。

複数の機能がセミコロン (;) で区切られていることを確認します。ほとんどの機能では、`true` または `false` という値を使用できます。

関連情報

[リレーショナルデータソースに対応したコネクタの capabilities の全一覧](#) [59 ページ]

5.5 システムパラメータリスト

システムパラメータ	説明
NUM_PARTITIONS_FOR_DISTINCT_OPERATOR	<p>非重複演算子について作成する第 1 レベルパーティションの最適な数。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 300</p>

システムパラメータ	説明
MAX_CONCURRENT_MEMORY_CONSUMING_QUERIES	<p>パラレルクエリの最大数。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 2</p>
MAX_CONCURRENT_MEMORY_CONSUMING_OPERATORS	<p>メモリを消費する並行演算子の最大数。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。現時点では、サーバを再起動する必要があります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 5</p>
EXECUTOR_STATIC_MEMORY	<p>このパラメータは、初期化時に演算子に割り当てられる最小メモリスペースを表します。これは、厳密な値 (たとえば EXECUTOR_STATIC_MEMORY=50M。エグゼキュータに割り当てられたメモリスペースよりも小さな値である必要があります。EXECUTOR_TOTAL_MEMORY パラメータを参照してください) であるか、あるいはエグゼキュータのメモリサイズに占めるパーセンテージ (たとえば EXECUTOR_STATIC_MEMORY=25%) のいずれかになります。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 25%</p>
EXECUTOR_TOTAL_MEMORY	<p>このパラメータは、エグゼキュータに割り当てられるメモリスペースを示します。これは、メモリサイズの値 (たとえば EXECUTOR_TOTAL_MEMORY=256M)、あるいは JVM によってエグゼキュータに割り当てられるメモリスペースがメモリサイズに占めるパーセンテージ (たとえば EXECUTOR_TOTAL_MEMORY=80%) のいずれかになります。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 80%</p>

システムパラメータ	説明
EXECUTOR_BUFFER_OVERHEAD	<p>このパラメータは、クエリ実行中に発生するメモリオーバーヘッドを表します。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 10%</p>
ACCEPT_MORE_CONCURRENT_QUERIES	<p>最大数の並列クエリが存在する場合、例外を通知する代わりに、待機中のクエリのリストを記録する (true の場合) ことを示す論理値。このパラメータの新規値は、BufferManager に登録されたクエリがない場合に有効化されます。</p> <p>型: 論理型</p> <p>再起動の必要性: なし</p> <p>デフォルト値: true</p>
MIN_BUFFER_PAGES_PER_OPERATOR	<p>演算子に返される最小ページ数。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 8</p>
EXECUTOR_BUFFER_SIZE	<p>ページパラメータのサイズ。行数で示します。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 128</p>
MAX_BUFFER_SHARE_PER_OPERATOR	<p>最大動的バッファの最大割合。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 80</p>
SEMI_JOIN_DIMENSION_CACHE_MEMORY_SIZE	<p>準結合において 1 つのディメンションキャッシュに割り当てられるメモリ容量 (キロバイト単位)。</p>

システムパラメータ	説明
	<p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 1024</p>
QUERY_HISTORY_SIZE	<p>実行されたクエリのリポジトリの最大履歴サイズ</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 10</p>
MAX_SUBQUERIES_IN_HISTORY	<p>履歴における各クエリセットのサブクエリの最大数。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 100</p>
MAX_EXECUTIONS_PER_SUBQUERY_IN_HISTORY	<p>履歴に保持されるサブクエリの最大実行数。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 10</p>
MAX_ESTIMATED_SIZE_FOR_STRINGS_OR_DECIMALS	<p>文字列または小数值のバイト単位の最大推定サイズ。このパラメータの新しい値は次のクエリで有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 512</p>
MAX_PARTITIONS_FOR_HASH_OPERATORS	<p>ハッシュアルゴリズムについて作成する第 1 レベルパーティションの最大数。このパラメータの新しい値は次のクエリで有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 1987</p>
MIN_SIZE_FOR_BUFFER_HASH_TABLE	<p>ハッシュ結合/非重複で使用するバッファハッシュテーブル (各エントリの二次ハッシュ) の最大サイズ。このパラメータの新しい値は次のクエリで有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 200</p>

システムパラメータ	説明
MAX_TEMPORARY_TABLES	<p>1つのコネクタで生成される一意の一時テーブルの最大数を定義します。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 1000</p>
SEMI_JOIN_EXECUTION_STRATEGIES	<p>希望の順番で、準結合演算子の実行方針の一覧を指定します。I、T、および P をカンマで区切って希望の順番で組み合わせる値を指定できます。I は IN クエリの実行方針、T は一時テーブルの実行方針、P はパラメータ化されたクエリの実行方針を表します。例 T,P,I、I,T P NONE (方針がない場合)。I、T、または P のうちのいずれかが指定されていない場合、対応する実行方針はラッパでサポートされません。注: NONE は、ラッパがサポートする実行方針がないことを表します。空の文字列および null 値はサポートされていません。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: I,T,P</p>
MAX_CONJUNCTIONS	<p>述語内の接続詞の最大数。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 512</p>
ACTIVATE_MULTI_THREADED_UNION_OPERATOR	<p>ユニオン演算子のマルチスレッド実装を使用するかどうかを指定します。</p> <p>型: 論理型</p> <p>再起動の必要性: なし</p> <p>デフォルト値: false</p>
ACTIVATE_ORDER_BASED_OPTIMIZATION_RULE	<p>true に設定されると、順序ベース最適化のすべてのルールが有効化されます。</p> <p>型: 論理型</p> <p>再起動の必要性: なし</p> <p>デフォルト値: true</p>
ACTIVATE_PROFITABILITY_BASED_JOIN_ORDERING_RULE	<p>true に設定されると、有益性に基づいてノードが多いツリーを構築しようとする順序結合ルールが有効化されます。</p> <p>型: 論理型</p>

システムパラメータ	説明
	再起動の必要性: なし デフォルト値: <i>true</i>
ACTIVATE_JOIN_DISTRIBUTION_RULE	true に設定されると、結合配分ルールが有効化されます。 型: 論理型 再起動の必要性: なし デフォルト値: <i>true</i>
ACTIVATE_JOIN_ELIMINATION_RULE	true に設定されると、不要結合排除ルールが有効化されます。 型: 論理型 再起動の必要性: なし デフォルト値: <i>false</i>
ACTIVATE_SEMI_JOIN_RULE	true に設定されると、準結合を生成するルールが有効化されます。 型: 論理型 再起動の必要性: なし デフォルト値: <i>true</i>
ACTIVATE_SEMI_JOIN_DIMENSION_RUNTIME_CARDINALITY_LIMIT	準結合の実行時に計算されたカーディナリティの制限を有効化するかどうかを指定します。このオプションを有効化した場合、準結合ディメンションのランタイムカーディナリティは予測値と比較されます。ランタイム値が予測値を上回る場合、ディメンションは破棄されます。 型: 論理型 再起動の必要性: なし デフォルト値: <i>true</i>
AVG_SIZE_OF_BUFFER_ROW	1 行の平均サイズの予測値。このパラメータの新しい値は、BufferManager 内にクエリが登録されていない場合に有効となります。 型: 整数 再起動の必要性: なし デフォルト値: 256
MIN_CARDINALITY_THRESHOLD_FOR_GROUP_BY_TRANSFORMATION_RULE	ソース <p> の順番を使用して GroupBy ノードの削除を決定する非重複値の最小カーディナリティ。0 は Group by 削除が常に行われる必要があることを示します。

システムパラメータ	説明
	<p>型: long</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 300</p>
NUMBER_OF_PARTITIONS_FOR_MERGE_AGGREGATE_RULE	<p>MergeBasedGroupByAggregate アルゴリズムで使用するパーティションの数。このパラメータの新しい値は次のクエリで有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 300</p>
MIN_CARDINALITY_FOR_ASYNC_PREFETCH	<p>非同期プリフェッチを決定するための最小カーディナリティのパラメータ。-1 は、非同期プリフェッチが許可されないことを意味します。</p> <p>型: long</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 50000</p>
MAX_ORDERING_LIMIT_FOR_ORDER_JOINS_RULE	<p>推定ルール OrderJoinsRule のパラメータ。これは、作成する結合順序の最大数です。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 1</p>
MAX_QUEUE_SIZE_LIMIT_FOR_ORDER_JOINS_RULE	<p>推定ルール OrderJoinsRule のパラメータ。これは、優先順位キューの最大サイズです。これにより、検索スペースの最大サイズが定義されます。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 1024</p>
DEFAULT_DECIMAL_PRECISION	<p>コネクタが列の値を返さない場合に、Data Federator クエリサーバによってレポートされる列の小数精度の値。通常の状態では、コネクタが常にこの値を指定します。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 27</p>

システムパラメータ	説明
DEFAULT_DECIMAL_SCALE	<p>コネクタが列の値を返さない場合に、Data Federator クエリサーバによってレポートされる列の小数スケール値。通常の場合では、コネクタが常にこの値を指定します。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 6</p>
MAX_DECIMAL_PRECISION	<p>Data Federator クエリサーバによってレポートされる列の小数精度の最大値。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 40</p>
SCALE_FOR_MAX_DECIMAL_PRECISION	<p>データフェデレーションクエリエンジンによってレポートされる、精度が最大の列の 10 進法の値。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 6</p>
MIN_DECIMAL_SCALE	<p>データフェデレーションクエリエンジンによってレポートされる列の 10 進法の最小値。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 6</p>
DEFAULT_STRING_SIZE	<p>コネクタが列の値を返さない場合に、データフェデレーションクエリエンジンによってレポートされる列の文字列サイズの値。通常の場合では、コネクタが常にこの値を指定します。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 40</p>
MAX_STRING_SIZE	<p>データフェデレーションクエリエンジンによってレポートされる列の文字列サイズの最大値。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 255</p>

システムパラメータ	説明
MAX_NUMBER_OF_FRACTIONAL_DIGITS_FOR_TOSTRING_DOUBLE_IN_LOCALE	<p>ロケール依存関数 toStringL(double, varchar) を使用する場合の double の文字列表示における最大小数点以下桁数</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 20</p>
NUMBER_OF_PARTITIONS_FOR_HASH_JOIN_OPERATOR	<p>HashJoin/HashOuterJoin アルゴリズムの第 1 レベルパーティションの推定最適数 (このパラメータの新しい値は次のクエリで有効となります)</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 300</p>
MAX_THREADS_IN_UNION_OPERATOR	<p>演算子 UNION によって使用されるアクティブスレッドの最大数。このパラメータの新しい値は次のクエリで有効となります。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 2</p>
ACTIVATE_OPTIMIZED_PREPARED_STATEMENTS	<p>すべての実行、または準備した文のそれぞれの実行に対して、クエリが最適化されるかどうかを示す論理型。true に設定すると、準備した文のすべての実行に対して同じクエリプランが使用されます。false に設定すると、準備した文のそれぞれの実行に対してクエリが再度最適化されます。</p> <p>型: 論理型</p> <p>再起動の必要性: なし</p> <p>デフォルト値: false</p>
NUMBER_OF_PARTITIONS_FOR_ORDER_AGGREGATE_RULE	<p>OrderBasedGroupByAggregate アルゴリズムで使用するパーティションの数 (このパラメータの新しい値は次のクエリで有効となります)</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 1987</p>
MAX_CARDINALITY_FOR_HOP_STORE_IN_HASH_JOIN_OPERATOR	<p>HOP アルゴリズムのストアの最大カーディナリティのパラメータ。</p>

システムパラメータ	説明
	<p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 5000</p>
MIN_STORE_CARDINALITY_THRESHOLD_FOR_ORDER_BASED_JOIN_RULE	<p>指示された MergeJoin の使用を正当化する保存サイズの最小カーディナリティを示すパラメータ。</p> <p>型: long</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 10000</p>
MIN_TRANSFER_CARDINALITY_THRESHOLD_FOR_MERGE_JOIN_RULE	<p>指示された Merge Join の使用を正当化する転送の最小カーディナリティを示すパラメータ。</p> <p>型: long</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 30000</p>
THREADPOOL_ACTION_ON_OUT_OF_MEMORY	<p>緊急メモリ不足状況が発生した場合に、実行されるアクションを設定するための特殊パラメータが、メモリトラッカーで見つかりました。可能なアクションは、freeze、kill&freeze、cancel running、cancel all、none です。freeze を選択すると、管理クエリを実行するスレッドをすべて停止します。これにより、特殊な外部ツールを使用して仮想マシンの状態を調査できます。kill&freeze を選択すると、現在実行中の管理クエリを強制停止します（管理クエリの現在のスレッドは強制停止され、クエリはキャンセルされて閉じられます）。このソリューションでは、プロファイラが動作できるようにメモリをさらにいくらか解放します。その代わりに、サーバが不整合な状態になり、それ以上クエリを実行できなくなります。クエリがキャンセルされた後、サーバは仮想的に凍結されます（それ以上クエリは実行できません）。cancel running を選択すると、現在管理されているすべてのクエリと、それに対しサーバで現在実行されるアクションがキャンセルされます。これにより、メモリを回復させて、サーバを実行し続けることができます。cancel all を選択すると、すべてのクエリをキャンセルします。キャンセルは、現在実行中のクエリが問題の原因であり、内部サーバエラーではない場合のみ、メモリを解放します。注: 現在は、ThinDriver またはリモートサーバ接続から送信されたクエリをすべて管理クエリとして表示します。管理コンソールまたはシンプルテキストコンソールは、管理クエリを使用しません。そのため管理クエリは明示的に凍結されません。</p>

システムパラメータ	説明
	<p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: <i>cancel all</i></p>
ACTIVATE_FREEZE_WHEN_OUT_OF_MEMORY	<p>OutOfMemory が発生した場合に実行されるアクションを設定するための特殊パラメータ。設定されると、すべての管理スレッドが停止されます。一部のアクションは非管理スレッドを介して機能する場合がありますが、システムの状態は不安定になります。設定されなかった場合、システムが単純に終了し、Java プロセスが停止します。</p> <p>型: 論理型</p> <p>再起動の必要性: なし</p> <p>デフォルト値: <i>false</i></p>
DEFAULT_LOCALE	<p>ロケールの ISO ロケールコードを定義します。このシステムパラメータは、セッションパラメータ LOCALE のデフォルト値です。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: <i>en_US</i></p>
DEFAULT_SORT	<p>sort 照合を定義します。このシステムパラメータは、セッションパラメータ SORT のデフォルト値です。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: <i>binary</i></p>
DEFAULT_COMP	<p>comp 照合を定義します。このシステムパラメータは、セッションパラメータ COMP のデフォルト値です。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: <i>binary</i></p>
OPTIMIZER_COMPUTE_BINDINGS_PARAMETER	<p>ResolveBindings ルールプロパティのパラメータを定義します。0 はルールを無効化、1 はバインド結合で解決、2 はキャッシュノードで解決です。</p> <p>型: 整数</p> <p>再起動の必要性: なし</p> <p>デフォルト値: <i>2</i></p>

システムパラメータ	説明
MIN_SOURCE_CARDINALITY_THRESHOLD_FOR_SEMI_JOIN_RULE	<p>準結合演算子を有効化するのに必要な大規模テーブルのカーディナリティのしきい値を指定します。</p> <p>型: long</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 15000</p>
MIN_ACTIVATION_THRESHOLD_FOR_SEMI_JOIN_RULE	<p>これは、Data Federator で準結合が有効であると見なされるために、テーブルのフルスキャンと比較して準結合で返されるタブルの比率です。取得する必要がある値が多過ぎると、準結合の有効性が薄れ、Data Federator では代わりにテーブルスキャンが実行されます。たとえば、テーブルに 1000 万行ある場合、最小有効化しきい値を 1000 に設定し、その結果 $1000 \text{ 万} / 1000 = 10 \text{ 000}$ になります。準結合を実行するために 10000 行未満を取得すると計算する場合、Data Federator は準結合演算子を使用します。頻度を低めに準結合を使用する場合、この値を大きくします。頻繁に準結合を使用する場合、この値を小さくします。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 100</p>
MAX_ACTIVATION_LIMIT_FOR_PUSH_AGGREGATE_RULE	<p>これは、"Group-By" 演算子がソースでプッシュされた場合に返される行数と "Group-By" 演算子がソースでプッシュされない初期カーディナリティとの比率であり、Data Federator が "Group-By" 演算子のプッシュを有用であるとみなすための基準となります。取得する必要がある値の数が多過ぎると、"Group-By" のプッシュの有用性は薄れます。たとえば、パラメータが 80% (0.80) に設定されており、"Group-By" のプッシュが原因の新規カーディナリティが初期カーディナリティの 80% を超過する場合、"Group-By" はソースにプッシュされません。頻繁に "Group-By" 演算子をプッシュする場合、この値を大きくします。頻度を低めに "Group-By" 演算子をプッシュする場合、この値を小さくします。</p> <p>型: 文字列</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 1.00</p>
CLUSTER_SYNCHRONIZE_DELAY	<p>2 つの同期イベントの間隔の時間を秒単位で定義します。フォールトトレランスモジュールは、同期イベントを使用して、リポジットで可能性のある変更を識別します。これら</p>

システムパラメータ	説明
	<p>は、リソース変更アクションの途中で発生したサーバエラーのため、他のクラスタメンバーに通知されなかった変更です。</p> <p>型: long</p> <p>再起動の必要性: なし</p> <p>デフォルト値: 3600</p>

関連情報

[準結合の実行ストラテジー](#) [32 ページ]

5.6 セッションパラメータリスト

セッションパラメータ	説明
CATALOG	現在のカタログを定義し、クエリにカタログがない場合に使用されます。
SCHEMA	現在のスキーマを定義し、クエリにスキーマがない場合に使用されます。
COMP	文字列の比較に使用する照合を定義します。SQL クエリでの文字列の比較方法を定義するために使用されます。このパラメータの値は、サポートされているいずれかの照合値、またはキーワード <i>LINGUISTIC</i> です。この場合、使用される照合は <i>SORT</i> パラメータで定義された照合です。デフォルトは <i>BINARY</i> です。デフォルト値は、システムパラメータ <i>DEFAULT_COMP</i> で変更できます。
SORT	文字列の並べ替えに使用する照合を定義します。SQL クエリでの文字列の並べ替え方法を定義するために使用されます。このパラメータの値は、サポートされているいずれかの照合値です。デフォルトは <i>BINARY</i> です。デフォルト値は、システムパラメータ <i>DEFAULT_SORT</i> で変更できます。
LOCALE	ロケールの ISO コードを定義します。デフォルトは <i>en_US</i> です。デフォルト値は、システムパラメータ <i>DEFAULT_LOCALE</i> で変更できます。
DATA_LOCALE	データで使用するロケールを定義します。このパラメータは、ローカライズされたデータを返すことができるコネクタ (現在は SAP BW コネクタ) で使用されます。

関連情報

[データフェデレーションアプリケーション内での照合](#) [76 ページ]

[システムパラメータリスト](#) [63 ページ]

5.7 データフェデレーションアプリケーション内での照合

照合とは、データの並び替え方法と比較方法を定めたルールセットのことです。

データフェデレーションアプリケーションと、データフェデレーションアプリケーションがアクセスするデータベースシステムでは、文字の正確な順序を定義したルールを使用して、文字データが並び替えおよび比較されます。大部分のデータベースシステムの場合、データベースシステムで大文字と小文字、アクセント記号の有無、半角と全角、またはひらがなとカタカナを区別する必要があるかどうかをオプションで設定できます。

大文字と小文字の区別

文字 *M* が *m* と同様に扱われている場合には、大文字と小文字が区別されていません。ASCII コードを使用して入力が区別されるため、コンピュータでは *M* と *m* が異なる文字として扱われます。*M* の ASCII 値は 77 ですが、*m* の ASCII 値は 109 です。

アクセントの区別

文字 *a* が *á* と同様に扱われている場合には、アクセントが区別されていません。ASCII コードを使用して入力が区別されるため、コンピュータでは *a* と *á* が異なる文字として扱われます。*a* の ASCII 値は 97 ですが、*á* の ASCII 値は 225 です。

かなの区別

日本語のひらがなとカタカナの扱いが異なることをかなの区別と呼びます。

半角と全角の区別

1 バイト文字 (半角) と 2 バイト文字 (全角) の扱いが異なる場合には、半角と全角が区別されています。

関連情報

[データフェデレーションアプリケーション内でサポートされている照合 \[77 ページ\]](#)

[バイナリ照合の使用時にデータフェデレーションアプリケーションがクエリをソースへどのようにプッシュするのかを決定する方法 \[80 ページ\]](#)

5.7.1 データフェデレーションアプリケーション内でサポートされている照合

Data Federator では以下の照合がサポートされています。

binary	ユニコードバイナリソート (またはユニコードバイナリソートと互換性があるもの。たとえば ASCII 文字セットのソートはユニコード文字セットのソートと互換性があります)。
locale_AI_CI	ロケール、アクセント記号の有無の区別なし、大文字と小文字の区別なし
locale_AS_CI	ロケール、アクセント記号の有無の区別あり、大文字と小文字の区別なし
locale_AS_CI	ロケール、アクセント記号の有無の区別あり、大文字と小文字の区別なし
locale_AI_CS	ロケール、アクセント記号の有無の区別なし、大文字と小文字の区別あり
locale_AS_CS	ロケール、アクセント記号の有無の区別あり、大文字と小文字の区別あり

locale は LN_CY として定義されています。

- LN: ISO 言語コード (たとえば **<en>**)
- CY: ISO 国コード (たとえば **<us>**)

注記

すべての Data Federator 照合ではひらがなとカタカナは区別されず、半角と全角も区別されません。

例

<en_US_AS_CI>: 英語、米国、アクセント記号の有無の区別あり、大文字と小文字の区別なし

関連情報

[データフェデレーションアプリケーション内での照合 \[76 ページ\]](#)

5.7.2 データフェデレーション SQL クエリの文字列並び替えおよび文字列比較動作の設定

sort パラメータと comp パラメータを使用して、データフェデレーションクエリエンジンによる文字列の並び替え方法と比較方法を設定できます。

この sort パラメータは、データフェデレーションクエリエンジンでの文字列の並び替え方法を定義するために使用されます。sort パラメータの値は、サポートされているいずれかの照合値です。デフォルトはバイナリです。

comp パラメータは、SQL クエリでの文字列の比較方法を定義するために使用されます。comp パラメータの値は以下のいずれかです。

- サポートされているいずれかの照合値。
- キーワード言語: この場合、使用された照合は sort パラメータで定義された照合です。

sort パラメータと comp パラメータをセッションパラメータ、システムパラメータ、またはユーザアカウントのプロパティとして定義できます。

- sort パラメータまたは comp パラメータがセッションパラメータで定義されている場合には、この値が現在の接続に使用されます。
- セッションパラメータで定義されていない場合には、ユーザアカウントの sort または comp プロパティが現在の接続に使用されます。
- 現在のユーザアカウントのプロパティとして定義されていない場合には、sort または comp システムパラメータが現在の接続に使用されます。

sort パラメータと comp パラメータの値は、文字列値に適用されている SQL オペレーションの結果に影響します。オペレーションは関数、GROUP BY や ORDER BY などの SQL 演算子、または `T.A < e` などのフィルタ式です。以下の表は comp パラメータと sort パラメータの影響を受ける SQL 演算子をまとめたものです。

SQL 式	影響
<code>=, !=, >, <=, >=</code>	comp の影響を受ける
<code>BETWEEN, NOT BETWEEN</code>	comp の影響を受ける
<code>CASE</code>	comp の影響を受ける
<code>DISTINCT</code>	comp の影響を受ける
<code>GROUP BY</code>	comp の影響を受ける
<code>HAVING</code>	comp の影響を受ける
<code>IN, NOT IN</code>	comp の影響を受ける
<code>LIKE, NOT LIKE</code>	影響を受けない: バイナリのみ
<code>ORDER BY</code>	sort の影響を受ける
<code>UNION ALL</code>	影響を受けない

SQL 関数	影響
MAX、MIN	comp の影響を受ける
データフェデレーション文字列関数	影響を受けない: バイナリのみ

例

```
SELECT LASTNAME, count(*)
FROM EMPLOYEE E
WHERE SALARY < 5000 AND DEPARTMENT_NAME =
  <Sales>
GROUP BY LASTNAME
```

表 1: Employee テーブル

LASTNAME	FIRSTNAME	SALARY	DEPARTMENT_NAME
Smith	John	6000	Sales
Smith	John	4000	Sales
Smith	John	2000	Sales
Smith	Albert	7000	Sales

comp パラメータが **<en_US_AS_CS>** であると、結果は以下のようになります。

Smith	3
Smith	1

comp パラメータが **<en_US_AI_CI>** であると、結果は以下のようになります。

Smith	4
-------	---

関連情報

[データフェデレーションアプリケーション内での照合 \[76 ページ\]](#)

[データフェデレーションアプリケーション内でサポートされている照合 \[77 ページ\]](#)

5.7.3 バイナリ照合の使用時にデータフェデレーションアプリケーションがクエリをソースへどのようにプッシュするのかを決定する方法

データフェデレーションクエリエンジンのオプティマイザでは、SQL オペレーションをデータソースへプッシュダウンできるかどうかを判断するために、プッシュダウン分析が実行されます。

照合がバイナリの場合には、データソースの SQL 機能のみを検証することにより、サブクエリを特定のデータソースへプッシュするかどうかはクエリエンジンで決定されます。

そのため、一般的な場合には、データフェデレーションアプリケーションでのバイナリ照合に準拠したデフォルト照合が基盤データソースで使用されるとクエリエンジンでは仮定されます。

SQL Server、MySQL、および Oracle に限っては、ソースのデフォルト照合がバイナリ照合に準拠していない場合でも、データフェデレーションクエリエンジンでのバイナリ照合の使用を強制できます。(バイナリ照合のリソースパラメータ設定方法の詳細については、MySQL、SQL Server、および Oracle を参照してください)

関連情報

[データフェデレーションアプリケーション内での照合](#) [76 ページ]

[データフェデレーション SQL クエリの文字列並び替えおよび文字列比較動作の設定](#) [78 ページ]

[データフェデレーションアプリケーション内でサポートされている照合](#) [77 ページ]

6 SQL 構文リファレンス

6.1 データフェデレーションクエリエンジンのクエリ言語

可能な場合、データフェデレーションアプリケーションは標準 SQL-92 構文に合わせます。ただし、一部の要素がデータフェデレーションクエリエンジンの文でどのように使用されるか（影響するか）を理解することが重要です。この節では、オブジェクト管理、データ型、選択、式など、データフェデレーションアプリケーションで実装されている SQL-92 の要素について説明します。

6.1.1 識別子と命名規則

テーブルを参照するには、そのテーブルが含まれるカタログおよびスキーマを指定します。カタログ、スキーマ、およびテーブルはドット (.) で区切る必要があります。

例

テーブル名の定義

テーブルの参照には修飾名を使用する必要があります。修飾名はカタログ名、スキーマ名、およびテーブル名で構成されています。

- `c.s.t`
- `"c"."s"."t"`

デフォルトで定義されているカタログまたはスキーマがある場合は、テーブルへの参照でカタログまたはスキーマの名前を省略できます。

関連情報

[二重引用符区切り文字の使用](#) [83 ページ]

[デフォルトのカタログおよびスキーマ](#) [83 ページ]

6.1.1.1 カタログ

カタログは、スキーマの名前付きグループです。カタログの名前は、そのカタログに属するスキーマの名前を修飾します。クエリ内でカタログ名を明示的に示すか、またはデフォルトのカタログを設定します。

関連情報

[デフォルトのカatalogおよびスキーマ \[83 ページ\]](#)

6.1.1.2 スキーマ

SQL スキーマは、テーブルまたはビューの名前付きグループです。スキーマはカatalogに属します。スキーマ名は、それが属しているカatalog内で一意である必要があります。

スキーマ識別子は、完全パス(デフォルトのカatalogが設定されていない場合)またはデフォルトのカatalogディレクトリからの相対パスです。

デフォルトのスキーマは、データフェデレーション管理ツールで、セッションパラメータを使用して設定できます。

関連情報

[デフォルトのカatalogおよびスキーマ \[83 ページ\]](#)

6.1.1.3 テーブル

テーブルは 1 つのスキーマに属します。テーブル名は、それが属しているスキーマ内で一意である必要があります。

テーブルは、カatalog名、スキーマ名、およびテーブル名によって識別される必要があります。標準 SQL 構文では、テーブル識別子はカatalog名、スキーマ名、およびテーブル名を . (ピリオド) で区切って連結することで作成されます。

デフォルトのカatalogまたはデフォルトのスキーマ、あるいはその両方が設定されている場合は、テーブル識別子でカatalog名とスキーマ名を省略できます。

関連情報

[デフォルトのカatalogおよびスキーマ \[83 ページ\]](#)

6.1.1.4 列

テーブルは、一連の列で記述されます。列名は、それが属しているテーブル内で一意である必要があります。標準 SQL 構文では、列識別子はテーブル識別子をピリオド (.) で区切った列名と連結することで作成されます。

6.1.1.5 デフォルトのカatalogおよびスキーマ

デフォルトのカatalogまたはスキーマは、データフェデレーション管理ツールで、セッションパラメータを使用して指定できます。デフォルトのカatalogを指定すると、完全修飾テーブル名を使用せずにクエリを送信できます。

参照するテーブル	デフォルトのカatalog	デフォルトのスキーマ	使用する修飾名
c.s.t	c		s.t
"c1".s.t	"c1"		s.t
c.s.t	c	s	t

関連情報

[データフェデレーション管理ツールを使用したシステムパラメータの変更](#) [62 ページ]

6.1.1.6 二重引用符区切り文字の使用

パーサが ID を誤って解釈することを防ぐため、カatalog、スキーマ、テーブル、および列の名前に英数字以外の文字が含まれる場合には、これらの名前に二重引用符区切り文字を使用する必要があります。

修正	"c1/c2"."sche+ma"."Tab-le1".col1
誤	/c1/c2.sche+ma.Tab-le1.col1

詳細については、[オブジェクト識別子と数値定数](#) [90 ページ]を参照してください。

6.1.2 データフェデレーションクエリエンジンで使用されるデータ型

データフェデレーションクエリエンジンでは、列、ローカル変数、式、パラメータにそれぞれ関連するデータ型があります。データ型は、整数データ、文字データ、日付と時刻のデータ、小数データといった、オブジェクトが保持できるデータのサイズおよび構造の定義です。

1つのオブジェクトに関するデータ型は、次の3つのオブジェクト属性を定義します。

- データ型: オブジェクトに含まれるデータの種類
- 長さおよびサイズ: 値の長さまたはサイズ
- スケールおよび精度: 数のスケールおよび精度 (数値のデータ型のみ)

従来のデータベースでは、長さ、精度、スケールは保存された値のプロパティを定義するため、設定は列を作成するときに行われていました。データフェデレーションクエリエンジンは仮想データベースであり、値を保存しません。そのため、長さ、精度、スケールはスキーマの定義時には定義されていません。それぞれの値はソーステーブルから動的に推量されます。

関連情報

[既知のデータ型](#) [84 ページ]

[式内の型の推定](#) [86 ページ]

[スケールと精度](#) [86 ページ]

6.1.2.1 既知のデータ型

データフェデレーションクエリエンジンは、`java.sql.Types` で定義される標準 SQL 型をサポートします。サポートされるデータ型の一覧は以下のとおりです。

- `BIT`
- `DATE`
- `TIMESTAMP`
- `TIME`
- `INTEGER`
- `DOUBLE`
- `DECIMAL`
- `VARCHAR`
- `NULL`

すべてのデータベースにおいて使用するデータ型やデータ型の解釈方法が同じというわけではないので、クエリエンジンは共通データベース型からクエリエンジンまでのマッピングを標準化しています。

関連情報

<http://java.sun.com/j2se/1.3/docs/api/java/sql/Types.html> 

6.1.2.2 データフェデレーションクエリエンジンデータ型から JDBC データ型へのマッピング

次の表に、データフェデレーションクエリエンジンで使用される内部データ型とデータフェデレーション JDBC ドライバによって返される JDBC データ型の対応を示します。

データフェデレーションのデータ型	JDBC データ型
<code>BIT</code>	<code>BIT</code>
<code>DATE</code>	<code>DATE</code>

データフェデレーションのデータ型	JDBC データ型
<i>TIMESTAMP</i>	<i>TIMESTAMP</i>
<i>TIME</i>	<i>TIME</i>
<i>INTEGER</i>	<i>INTEGER</i>
<i>DOUBLE</i>	<i>DOUBLE</i>
<i>DECIMAL</i>	<i>DECIMAL</i>
<i>VARCHAR</i>	<i>VARCHAR</i>
<i>NULL</i>	<i>NULL</i>

6.1.2.3 JDBC データ型からデータフェデレーションデータ型へのマッピング

JDBC データソースにアクセスする際、データフェデレーションクエリエンジンは、JDBC ドライバによって返される JDBC 型から内部データフェデレーションアプリケーションデータ型へのマッピングを行います。次の表に、マッピングで使用する JDBC データ型とデータフェデレーション型の対応を示します。

JDBC データ型	データフェデレーションのデータ型
精度が 10 以下でスケールが 0 となる <i>TINYINT</i> 、 <i>SMALLINT</i> 、 <i>INTEGER</i> 、 <i>DECIMAL</i>	<i>INTEGER</i>
<i>BIT</i>	<i>BIT</i>
<i>REAL</i> 、 <i>FLOAT</i> 、 <i>DOUBLE</i>	<i>DOUBLE</i>
<i>BIGINT</i> 、 <i>DECIMAL</i> 、 <i>NUMERIC</i>	<i>DECIMAL</i>
<i>VARCHAR</i> 、 <i>LONGVARCHAR</i> 、 <i>CHAR</i>	<i>VARCHAR</i>
<i>DATE</i>	<i>DATE</i>
<i>TIME</i>	<i>TIME</i>
<i>TIMESTAMP</i>	<i>TIMESTAMP</i>
<i>NULL</i> および他の JDBC 型	<i>NULL</i>

6.1.2.4 日付と時刻の変換

データフェデレーションクエリエンジンは、日付を 1970-01-01 に設定することにより、*TIME* データを *TIMESTAMP* データに変換します。



例

時刻からタイムスタンプへの変換

TIME 12:01:01 は、タイムスタンプ 1970-01-01 12:01:01.0 に変換されます。

データフェデレーションクエリエンジンは、時刻 00:00:00.000000000 を加えることにより、*DATE* データを *TIMESTAMP* データに変換します。



例

日付からタイムスタンプへの変換

DATE '1999-01-01' は、*TIMESTAMP* '1999-01-01 00:00:00.000000000' に変換されます。

6.1.2.5 式内の型の推定

2つの式が異なるデータ型を持っている場合、算術演算子で2つの式を組み合わせてできる1つの式の結果のデータ型は、データ型優先度を適用して決定します。

データフェデレーションクエリエンジンは、次のようなデータ型優先度を使用します。

NULL
VARCHAR
INTEGER
DOUBLE
DECIMAL

6.1.2.6 スケールと精度

式の結果の長さ、スケール、精度は、その結果の型から推定されます。結果の型が *VARCHAR* または *DECIMAL* の場合、長さ、スケール、精度は、入力式のスケールと精度、およびそれらを組み合わせた関数と演算子から推定されます。

次のテーブルに、データフェデレーションの式に関するベクトル (長さ、精度、スケール) を示します。

列の種類	制限 (長さ、精度、スケール)
<i>BIT</i>	(1, 1, 0)
<i>INTEGER</i>	(11, 10, 0)

列の種類	制限 (長さ、精度、スケール)
DOUBLE	(22, 15, 0)
DATE	(10, 0, 0)
TIMESTAMP	(29, 9, 0)
TIME	(8, 0, 0)
NULL	(0, 0, 0)
DECIMAL	推定
VARCHAR	精度およびスケールは常に (0, 0) 長さは推測される

6.1.3 文

データフェデレーションクエリエンジンに保存されたデータを取得または使用するための SQL クエリを作成できます。クエリは以下のフォームで作成することができます。

- データフェデレーション管理ツール: データフェデレーションクエリエンジン上のグラフィカルユーザインターフェース (GUI)。
- コマンドライン SQL アプリケーション
- *SELECT* 文を作成できる、互換性のある別のユーティリティ
- Microsoft Visual Basic のようなクライアントまたは中間層ベースのアプリケーションは、グリッドのような SQL サーバテーブルから連結コントロールヘデータをマッピングすることができます。

6.1.3.1 *SELECT* 文

クエリとユーザの相互作用にはさまざまな方法がありますが、どの方法でも、*SELECT* 文の結果セットをユーザに提示するという同一のタスクを遂行します。

SELECT 文は、データフェデレーションクエリエンジンからデータを取得し、1 つまたは複数の結果セットとしてユーザに返します。結果セットは、*SELECT* 文からのデータをテーブル状に配置したものとなります。SQL テーブルのように、結果セットは列と行で構成されます。

SELECT 文の完全な構文は複雑ですが、ほとんどの *SELECT* 文は以下に示す結果セットの 4 つの主要プロパティを説明しています。

- 結果セットにおける列の数と属性
- データを提供するテーブルの名前
- ソーステーブル内の行が *SELECT* の基準を満たすために必要となる条件。条件を満たさない行は無視されます。
- 結果セットの行を順序付けるシーケンス



例

SELECT 文

次の SELECT 文は、単価が \$40 を超える製品の製品 ID、名前、リスト価格を検索します。

```
SELECT <ProductID>, <Name>, <ListPrice>
FROM <Production.Product>
WHERE <ListPrice> > <$40>
ORDER BY <ListPrice> ASC
```

- **SELECT 句**
SELECT キーワードの後に列挙されている列の名前 (<ProductID>、<Name>、<ListPrice>) が選択リストを形成しています。このリストは、結果セットが 3 つの列を持ち、それぞれの列に、FROM 句で指定されているテーブル (<Product> テーブル) 内の関連付けられた列の名前、データ型、サイズが定められていることを示しています。FROM 句は 1 つのテーブルだけしか指定しないので、SELECT 文の中のすべての列の名前はそのテーブルの列を参照します。
- **FROM 句**
FROM 句は、データを取得するテーブルとして、<Product> テーブルを示しています。
- **WHERE 句**
WHERE 句は、<Product> テーブル内の行で SELECT 文の基準を満たすのは <ListPrice> 列の値が <\$40> を超えている行のみであるという条件を示しています。
- **ORDER BY 句**
ORDER BY 句は、結果セットが <ListPrice> 列の値に基づいて昇順 (ASC) に並べ替えられることを示しています。

6.1.3.2 データフェデレーションクエリエンジンがサポートする SQL-92 文

データフェデレーションクエリエンジンは、データ操作言語 (DML) およびプロシージャとコマンドの一覧をサポートします。SELECT 文の特定セットがサポートされており、また記述がない限り、標準の SQL-92 構文全体がサポートされています。とくに、外部結合に関する SQL-92 文法と外部結合に関する JDBC 構文の両方がサポートされています。

関連情報

[SELECT 句の文法](#) [91 ページ]

6.1.4 式

この節では、データフェデレーション SQL 構文の式について詳細に説明します。

6.1.4.1 式の関数

最新の関数一覧を確認するには、インフォメーションデザインツールユーザガイドの複数のソースが有効化されたユニバースでの SAP BusinessObjects SQL 関数を参照してください。

6.1.4.2 式の演算子

式内の演算子は、1 つ以上の単純な式を組み合わせてより複雑な式を形成します。

演算子名	説明
+ (加算)	数値型の加算と VARCHAR 型の連結を意味する算術演算子。
- (減算)	減算を意味する算術演算子。
* (乗算)	乗算を意味する算術演算子。
/ (除算)	除算を意味する算術演算子。
% (剰余)	算術演算子除算の整数の余りを返します。たとえば、12 を 5 で割ると余りは 2 であるため、 $12 \% 5 = 2$ のようになります。
** (累乗)	算術演算子指定された式を指定された値で累乗した値を返します。
= (等しい)	"等しい" を意味する比較演算子。
> (大なり)	"大なり" を意味する比較演算子。
< (小なり)	"小なり" を意味する比較演算子。
>= (以上)	"以上" を意味する比較演算子。
<= (以下)	"以下" を意味する比較演算子。
<> (等しくない)	"等しくない" を意味する比較演算子。
ALL	比較のセットがすべて TRUE の場合に TRUE となる論理演算子。
AND	両方の BOOLEAN 式が TRUE の場合に TRUE となる論理演算子。
ANY	比較のセットのいずれかが TRUE の場合に TRUE となる論理演算子。

演算子名	説明
<i>BETWEEN</i>	オペランドが範囲内の場合に <i>TRUE</i> となる論理演算子。
<i>EXISTS</i>	サブクエリに行が含まれる場合に <i>TRUE</i> となる論理演算子。
<i>IN</i>	オペランドが式のリストのいずれかと等しい場合に <i>TRUE</i> となる論理演算子。
<i>LIKE</i>	オペランドがパターンと一致する場合に <i>TRUE</i> となる論理演算子。
<i>NOT</i>	他の <i>BOOLEAN</i> 演算子の値を反転する論理演算子。
<i>OR</i>	どちらかの <i>BOOLEAN</i> 式が <i>TRUE</i> の場合に <i>TRUE</i> となる論理演算子。
<i>SOME</i>	比較のセットの一部が <i>TRUE</i> の場合に <i>TRUE</i> となる論理演算子。
+(正数)	数値が正の値である単項演算子
-(負数)	数値が負の値である単項演算子

6.1.4.3 演算子の優先レベル

複数の演算子を含む複雑な式の場合は、演算子の優先レベルによって演算子を実行する順番が決まります。実行順序は、結果値に大きく影響します。

演算子には、以下の優先レベルがあります。高いレベルの演算子は、低いレベルの演算子の前に評価されます。

- +(正数)、-(負数)
- *(乗算)、/(除算)、%(剰余)、**(累乗)
- +(加算)、+(連結)、-(減算)
- =、>、<、>=、<=、<> (比較演算子)
- *NOT*
- *AND*
- *OR*
- *ALL*、*ANY*、*BETWEEN*、*IN*、*LIKE*、*SOME*

6.1.4.4 オブジェクト識別子と数値定数

識別子と定数の名前は、英字で始まる必要があり、その後は英数字とアンダースコアしか使用できません。ただし、識別子/定数の名前を二重引用符 (") で囲む場合は、名前に任意の文字を使用できます。

たとえば、ABC_12 や "!%任意の名前\$#\$%" のような識別子名を使用できます。

次の表では、識別子および数値定数のデータフェデレーション構文について説明します。

対象の型	使用する定義	例
Integer	<i>INTEGER</i> : nnn (数字のみ: 1 つ以上)	12 14 15
Double または Decimal	<i>DOUBLE/DECIMAL</i> : nn.nn (1 つ以上の数字の後に小数点と 1 つ以上の数字が続く)	12.3 13.222 11.3
Date	<i>DATE</i> : {d 'yyyy-mm-dd' }	{d '2005-03-28' }
Time	<i>TIME</i> : {t 'hh:mm:ss' }	{t '01:10:12' }
Timestamp	<i>TIMESTAMP</i> : {ts 'yyyy-mm-dd hh:mm:ss.ffff' }	{ts '2005-03-28 01:11:34.23222' }
String または Varchar	いずれか一重引用符で囲まれた文字列	'asdasdas'
単純識別子	英字で始まり、英数字とアンダースコアの任意の組み合わせが続く文字列	ABC_12
特殊文字を含む識別子	二重引用符で囲まれた任意の文字列	"!%any name you like\$#\$%"

6.1.5 コメント

SQL 文にコメントを追加するには、テキストの先頭に二重のハイフン (--) またはポンド記号 (#) を追加します。コメントは行の末尾で終了します。

6.2 SELECT 句の文法

次の節では、データフェデレーションクエリエンジンで使用される SQL Select 句の完全な文法を詳細に説明します。

```
start    := ( query ) ( ";" )? <EOF>

query    := ( <WITH> withListElement ( "," withListElement )* )?
           SQLSelectFromWhere (
             ( <UNION> | <INTERSECT> | <EXCEPT> ) ( <DISTINCT> | <ALL> )?
             SQLSelectFromWhere QueryExpression )?
           ( <ORDER> <BY> orderByTerms ( "," orderByTerms )* )?
```

```

QueryExpression :=
  ( ( <UNION> | <INTERSECT> | <EXCEPT> ) ( <DISTINCT> | <ALL> )?
    SQLSelectFromWhere ) *

withListElement := anyIdentifier <AS> ( WITHView | nativeQuery )

WITHView      := "(" query ")"

nativeQuery := <NATIVE> "("
  dataSourceIdentifier ","
  nativeQueryStatement ","
  columnSpecificationList
  ( "," paramSpecificationList )? ")"

dataSourceIdentifier := anyIdentifier

nativeQueryStatement := quotedString

columnSpecificationList := columnSpecification ( "," ( columnSpecification ) ) *
paramSpecificationList := paramSpecification ( "," ( paramSpecification ) ) *
columnSpecification := anyIdentifier columnDataType

paramSpecification := ( ( ( ( <DATE_LITERAL> | <TIME_LITERAL> |
  <TIMESTAMP_LITERAL> ) )
  | quotedString ) columnDataType )
  | <NULL_LITERAL>

columnDataType := identifier ( "(" integerLiteral ( "," integerLiteral )? ")" )?

integerLiteral := <INT_LITERAL>

SQLSelectFromWhere :=
  <SELECT> ( <DISTINCT> )? ( selectExpression ( "," selectExpression ) * |
  ( <MULT> ) )
  ( fromClause
  ( <WHERE> disjunction )?
  ( <GROUP> <BY> ( additiveTerm ) ( "," additiveTerm ) * )?
  ( <HAVING> disjunction )? )

fromClause := ( <FROM> tableReferenceList )

tableReferenceList := ( tableReference ( "," tableReference ) * )

tableReference := tableReferenceAtomicTerm ( qualifiedJoinPart ) *

tableReferenceAtomicTerm := ( tablePrimary )
  | jdbcOuterJoin
  | "(" query ")" ( ( <AS> )? ( identifier | delimitedIdentifier ) )?
  | "(" tableReference ")" ( ( <AS> )?
  identifier ( "(" projectAlias ( "," projectAlias ) * ")" )? )?

tablePrimary := ( table ( ( <AS> )? ( tableAlias ) )? )

table := ( anyIdentifier ( "." anyIdentifier ( "." anyIdentifier )? )? )

qualifiedJoinPart := ( ( <NATURAL> )? ( joinType )?
  <JOIN> tableReferenceAtomicTerm ( joinSpecification )? )

jdbcOuterJoin := "{" <OUTER_JOIN_JDBC> jdbcOuterJoinPart "}"

jdbcOuterJoinPart := tableReferenceAtomicTerm
  ( outerJoinType <OUTER> <JOIN> ( jdbcOuterJoinPart ) joinSpecification )?

joinType := ( ( <INNER> ) | ( <CROSS> ) | ( outerJoinType ( <OUTER> )? ) )

```

```

outerJoinType      := ( <LEFT> | <RIGHT> | <FULL> )

joinSpecification  := ( joinCondition | namedColumnsJoin )

joinCondition      := ( <ON> disjunction )

namedColumnsJoin   := ( <USING> "(" addUsing ( "," addUsing )* ")" )

addUsing           := columnName

projectAlias       := ( anyIdentifier )

selectExpression   := ( ( tableStar )
    | ( disjunction ( ( <AS> )? anyIdentifier )? ) )

tableStar          := table "." <MULT>

functionTermJdbc    := ( "{" <FUNCTION_JDBC> (
    ( identifier )
    | ( <LEFT> )
    | ( <RIGHT> ) ) "(" ( disjunction ( "," disjunction )* )? ")" "}" )

functionTerm        := ( (
    ( identifier ) |
    ( <LEFT> )
    | ( <RIGHT> ) )
    "(" ( ( <DISTINCT> | <ALL> )?
    ( disjunction ( "," disjunction )* | <MULT> ) )? ")" )

analyticFunctionPart := ( <OVER> "("
    ( <PARTITION> <BY> ( variable ) ( "," variable )* )?
    <ORDER> <BY> ( ( variable ( <ASC> | <DESC> )? ) )
    ( "," ( variable ( <ASC> | <DESC> )? ) )* ")" )

disjunction := ( conjunction ( <OR> conjunction )* )

conjunction := ( negationTerm ( <AND> negationTerm )* )

escapeChar     := quotedString

quotedString    := <QUOTED_STRING_LITERAL>

anyIdentifier   := <IDENTIFIER>
    | <DELIMITED_IDENTIFIER>

delimitedIdentifier := <DELIMITED_IDENTIFIER>

identifier      := <IDENTIFIER>

columnName      := anyIdentifier

negationTerm     := ( <NOT> )? ( ( comparisonTerm ) | ( <EXISTS> "(" query ")" ) )

comparisonTerm  := additiveTerm ( <COMPARISON_OPERATOR> (
    ( additiveTerm )
    | ( ( ( <ANY> ) | ( <SOME> ) | ( <ALL> ) ) "(" query ")" ) )
    | ( <BETWEEN> additiveTerm <AND> additiveTerm )
    | ( inValuesOrQuery )
    | <LIKE> additiveTerm ( <ESCAPE> escapeChar )?
    | <IS> ( <NULL_LITERAL> | <NOT> <NULL_LITERAL> )
    | <NOT> (
        <BETWEEN> additiveTerm <AND> additiveTerm
        | <LIKE> additiveTerm ( <ESCAPE> escapeChar )? ) ) )?

nativeExpression := <NATIVE> <EXPRESSION> "("
    dataSourceIdentifier ","
    columnName ","
    quotedString bindingArgumentList ")"

```

```

bindingArgumentList := ( "," additiveTerm )*

inValuesOrQuery := ( ( <NOT> )? <IN> "(" ( ( inValues ) | ( query ) ) ")" )

inValues      := ( signedConstant ( "," signedConstant )* )

additiveTerm   := ( factor ( ( <PLUS> | <MINUS> ) factor )* )

factor := unaryTerm ( (
    <MULT>
  | <DIVIDE>
  | <POWER>
  | <INT_DIVIDE>
  | <MOD> ) unaryTerm )*

unaryTerm := atomicTerm
           | <PLUS> atomicTerm
           | <MINUS> atomicTerm

variable   := ( anyIdentifier
  ( "." anyIdentifier
    ( "." anyIdentifier
      ( "." anyIdentifier )? )? )? )

variableFullName := anyIdentifier (
  "." anyIdentifier
  ( "." anyIdentifier
    ( "." anyIdentifier )? )? )?

constant := <BOOL_LITERAL>
           | <INT_LITERAL>
           | <FLOAT_LITERAL>
           | <SCIENTIFIC_NOTATION_LITERAL>
           | <DATE_LITERAL>
           | <TIMESTAMP_LITERAL>
           | <TIME_LITERAL>
           | <NULL_LITERAL>
           | quotedString
           | <PARAMETER>

signedConstant := <BOOL_LITERAL>
                 | ( <PLUS> | <MINUS> )? ( <INT_LITERAL> | <FLOAT_LITERAL> )
                 | <SCIENTIFIC_NOTATION_LITERAL>
                 | <DATE_LITERAL>
                 | <TIMESTAMP_LITERAL>
                 | <TIME_LITERAL>
                 | <NULL_LITERAL>
                 | quotedString
                 | <PARAMETER>

atomicTerm := functionTerm ( analyticFunctionPart )?
            | functionTermJdbc
            | variable
            | constant
            | "(" disjunction ")"
            | caseExpression
            | coalesceExpression
            | castExpression
            | convertFunction
            | nativeExpression

caseExpression := ( <CASE> ( ( additiveTerm ( (
  <WHEN> additiveTerm <THEN> additiveTerm ) )+ )
  | ( ( ( <WHEN> disjunction <THEN> additiveTerm ) )+ ) )
  ( <ELSE> additiveTerm )? <END> ) )

```

```

coalesceExpression := ( <COALESCE> "(" additiveTerm ( ( "," additiveTerm ) )+
                        ")" )

castExpression := ( <CAST> "(" disjunction <AS> identifier ")" )

convertFunction := ( <CONVERT> "(" disjunction "," identifier ")" )

tableAlias := ( delimitedIdentifier | identifier )

orderByTerms := ( variableFullName | integerLiteral ) ( <ASC> | <DESC> )?

bindingFunction := ( variable <COMPARISON_OPERATOR> additiveTerm )

startStoredProcedure := ( procedureCall ) ( ";" )? <EOF>

procedureCall := <CALL> anyIdentifier ( ( "(" procedureArguments ")" )
    | ( procedureArguments ) )

procedureArguments := ( procedureArgument ( "," procedureArgument )* )?

procedureArgument := ( procedureConstant )
    | ( <CAST> "(" procedureConstant <AS> identifier ")" )

procedureConstant := (
    <BOOL_LITERAL>
    | <INT_LITERAL>
    | <FLOAT_LITERAL>
    | <SCIENTIFIC_NOTATION_LITERAL>
    | <DATE_LITERAL>
    | <TIMESTAMP_LITERAL>
    | <TIME_LITERAL>
    | <NULL_LITERAL>
    | quotedString
    | <PARAMETER> )

<DEFAULT> TOKEN [IGNORE_CASE] : {
<FROM: "from">
| <SELECT: "select">
| <DISTINCT: "distinct">
| <WHERE: "where">
| <GROUP: "group">
| <ORDER: "order">
| <BY: "by">
| <HAVING: "having">
| <DESC: "desc">
| <ASC: "asc">
| <AS: "as">
| <UNION: "union">
| <INTERSECT: "intersect">
| <EXCEPT: "except">
| <WITH: "with">
| <USING: "using">
| <ON: "on">
| <MERGE: "merge">
| <MERGING: "merging">
| <NATIVE: "native">
| <EXPRESSION: "expression">
| <NATURAL: "natural">
| <JOIN: "join">
| <CROSS: "cross">
| <INNER: "inner">
| <OUTER: "outer">
| <LEFT: "left">
| <RIGHT: "right">
| <FULL: "full">
| <ESCAPE: "escape">
| <OUTER JOIN JDBC: "oj">
| <FUNCTION JDBC: "fn">

```

```

| <OVER: "over">
| <PARTITION: "partition">
| <CASE: "case">
| <WHEN: "when">
| <THEN: "then">
| <ELSE: "else">
| <END: "end">
| <COALESCE: "coalesce">
| <CALL: "call">
| <CAST: "cast">
| <CONVERT: "convert">
}

<DEFAULT> TOKEN [IGNORE_CASE] : {
    <NULL_LITERAL: "null">
}

<DEFAULT> TOKEN [IGNORE_CASE] : {
    <BOOL_LITERAL: "true" | "false">
}

<DEFAULT> TOKEN [IGNORE_CASE] : {
<AND: "and">
| <OR: "or">
| <IN: "in">
| <ANY: "any">
| <SOME: "some">
| <ALL: "all">
| <EXISTS: "exists">
| <BETWEEN: "between">
| <COMPARISON_OPERATOR: ">" | ">=" | "<" | "<=" | "=" | "<>">
| <LIKE: "like">
| <NOT: "not">
| <MULT: "*">
| <PLUS: "+">
| <MINUS: "-">
| <DIVIDE: "/">
| <INT_DIVIDE: "//">
| <POWER: "**">
| <MOD: "%">
| <IS: "is">
| <PARAMETER: "?">
}

<DEFAULT> SPECIAL : {
    <SINGLE_LINE_COMMENT: ("#" | "--") (~["\n", "\r"])* ("\n" | "\r" | "\r\n")*>
}

<DEFAULT> TOKEN : {
<INT_LITERAL: ([ "0"-"9" ])+>

| <FLOAT_LITERAL: ([ "0"-"9" ])+ "." ([ "0"-"9" ])+>

| <SCIENTIFIC_NOTATION_LITERAL: ("-" | "+")? (([ "0"-"9" ])+ ("." ([ "0"-"9" ])+)?
| ("." ([ "0"-"9" ])+)) ("e"|"E") ("-"|"+")? ([ "0"-"9" ])+>

| <DATE_LITERAL: "{" (" ")* "d" (" ")* "\" <DIGIT> <DIGIT> <DIGIT> <DIGIT>
    "-" <DIGIT> <DIGIT> "-" <DIGIT> <DIGIT> "\" (" ")* ">

| <TIME_LITERAL: "{" (" ")* "t" (" ")* "\" <DIGIT> <DIGIT>
    ":" <DIGIT> <DIGIT> ":" <DIGIT> <DIGIT> "\" (" ")* ">

| <TIMESTAMP_LITERAL: "{" (" ")* "ts" (" ")* "\"
    <DIGIT> <DIGIT> <DIGIT> <DIGIT> "-" <DIGIT> <DIGIT> "-" <DIGIT> <DIGIT> " "

```



```

    <DIGIT> <DIGIT> ":" <DIGIT> <DIGIT> ":" <DIGIT> <DIGIT>
    ("." (<DIGIT>)*)? "\" (' (' ')* ')">

| <DELIMITED_IDENTIFIER: "\" (~["\"", "\n", "\r"] | "\"\"")* "\"">

| <QUOTED_STRING_LITERAL: "\" (~["\""] | "\"'\"")* "\"'">

| <IDENTIFIER: <LETTER> (<LETTER> | <DIGIT>)*>

| <#URLCHAR: [":", "?", ".", "/", "@", "_", "-", "+", "%", "!"]>

| <#LETTER: ["$", "A"-"Z", "_", "a"-"z",
    "\u00c0"-" \u00d6", "\u00d8"-" \u00f6", "\u00f8"-" \u00ff", "\u0100"-" \u024f",
    "\u0370"-" \u052f", "\u0530"-" \u05ff", "\u0600"-" \u06ff", "\u0900"-" \u10ff",
    "\u1100"-" \u11f9", "\u1e00"-" \u1ef9", "\u0100"-" \u1fff", "\u3040"-" \u319f",
    "\u3200"-" \u32fe", "\u3300"-" \u33fe", "\u3400"-" \u3d2d", "\u4e00"-" \u9fff",
    "\uac00"-" \ud7a3", "\uf900"-" \ufa2d", "\ufb00"-" \ufb4f", "\ufb50"-" \ufdfb",
    "\ufe70"-" \ufefc", "\uff00"-" \uffff"]>

| <#DIGIT: ["0"-"9", "\u0660"-" \u0669", "\u06f0"-" \u06f9", "\u0966"-" \u096f",
    "\u09e6"-" \u09ef", "\u0a66"-" \u0a6f", "\u0ae6"-" \u0aef", "\u0b66"-" \u0b6f",
    "\u0be7"-" \u0bef", "\u0c66"-" \u0c6f", "\u0ce6"-" \u0cef", "\u0d66"-" \u0d6f",
    "\u0e50"-" \u0e59", "\u0ed0"-" \u0ed9", "\u1040"-" \u1049"]>
}

```

7 用語集

7.1 用語と説明

この節は、データフェデレーションアプリケーションおよび説明文書で使用される用語のリストを示します。

用語または表現	定義
コネクタ	データフェデレーションクエリエンジンがデータのソースに接続できるようにするドライバ。
ファンアウト	列間の関係で、最初の列の各エントリに関連する 2 番目の列のエントリの平均数。
merge 結合	テーブルの結合にかかる時間を短縮するために、2 つの大規模なデータテーブルを結合前にソートする、データフェデレーションで使用される操作。
プッシュ (動詞)	ある操作の実行のためにソースデータベースをリクエストすることです (データフェデレーションエンジン内で操作を実行するのではなく、ソースデータベースで操作を実行すると、一般的に、データフェデレーションエンジンで実行するより効率的です)。
準結合	2 つのテーブル間の操作で、2 番目のテーブルの少なくとも 1 行と一致する 1 番目のテーブルの行を返します。つまり、1 番目のテーブルが、2 番目のテーブルの行でフィルタ処理されます。
統計情報	データフェデレーションに使用されるソースに保存されたデータに関する数値情報で、テーブル内のエントリ予想数、列内の固有の値の予想数、ある列と他の列の各値間の関係の平均数などがあります。

8 トラブルシューティング

8.1 データフェデレーションサービスのロギングについて

データフェデレーションサービスは、SAP BusinessObjects Enterprise プラットフォーム上の Adaptive Processing Server によってホストされます。

データフェデレーションサービスをホストしている Adaptive Processing Server で、このサービスに関するログが見つかります。

SAP BusinessObjects Enterprise サーバのロギングの詳細については、*SAP BusinessObjects Enterprise* 管理者ガイドを参照してください。

8.2 SAP BW データソースで、実行時間が長いクエリで接続が閉じられる

SAP BW データソースでクエリの実行時間が 10 分を超えると、メッセージなしで接続が閉じられます。この現象は、SAP BW のデフォルトタイムアウト値がこのクエリを実行するには短すぎることが原因で発生します。

タイムアウト値を長くするには、次のステップを実行します。

1. 以下の手順にしたがってタイムアウト値を上げます。
2. SAP システムにログオンします。
3. トランザクションテキストフィールドに「rz11」を入力して実行します。
4. パラメータ `rdisp/max_wprun_time` を表示します。
5. [値の変更] をクリックし、パラメータの値を 600 以上に設定してレポートを実行できるようにします。
この値は秒単位で設定されます。

8.3 SAP BW コネクタのエラー NoClassDefFoundError: CpicDriver

例外 `NoClassDefFoundError: com.sap.conn.rfc.driver.CpicDriver` が表示されることがあります。

この例外は、SAP Java コネクタ (JCo) の依存がホストにインストールされていないために発生します。JCo は、SAP BW に接続するためにデータフェデレーションクエリエンジンで使用されるミドルウェアです。不足している依存は Microsoft Visual Studio 2005 C/C++ ランタイムライブラリのセットです。

Microsoft Visual Studio 2005 C/C++ ランタイムライブラリをインストールするには、SAPNote 684106 (<https://service.sap.com/sap/support/notes/684106>) を参照してください。

法的側面に関する重要免責事項

この文書は、情報提供のみを目的としています。その内容は予告なしに変更される場合があります。又、SAP はその内容に間違いがないことの保証を行いません。SAP は、商品性又は特定目的との適合性に関する明示的又は暗示的保証も一切行いません。

コードサンプル

この文書に含まれるソフトウェアコード及び / 又はコードライン / 文字列 (「コード」) はすべてサンプルとしてのみ提供されるものであり、本稼動システム環境で使用することが目的ではありません。「コード」は、特定のコードの構文及び表現規則を分かりやすく説明及び視覚化することのみを目的としています。SAP は、この文書に記載される「コード」の正確性及び完全性の保証を行いません。更に、SAP は、「コード」の使用により発生したエラー又は損害が SAP の故意又は重大な過失が原因で発生させたものでない限り、そのエラー又は損害に対して一切責任を負いません。

アクセシビリティ

この SAP 文書に含まれる情報は、公開日現在のアクセシビリティ基準に関する SAP の最新の見解を表明するものであり、ソフトウェア製品のアクセシビリティ機能の確実な提供方法に関する拘束力のあるガイドラインとして意図されるものではありません。SAP は、この文書に関する一切の責任を明確に放棄するものであり、この文書により直接又は間接的に契約上の義務又は誓約が発生することは一切ありません。

ジェンダーニュートラルな表現

SAP 文書では、可能な限りジェンダーニュートラルな表現を使用しています。文脈により、文書の読者は「あなた」と直接的な呼ばれ方をされたり、ジェンダーニュートラルな名詞 (例: 「販売員」又は「勤務日数」) で表現されます。ただし、男女両方を指すとき、三人称単数形の使用が避けられない又はジェンダーニュートラルな名詞が存在しない場合、SAP はその名詞又は代名詞の男性形を使用する権利を有します。これは、文書を分かりやすくするためです。

インターネットハイパーリンク

SAP 文書にはインターネットへのハイパーリンクが含まれる場合があります。これらのハイパーリンクは、関連情報を見いだすヒントを提供することが目的です。SAP は、この関連情報の可用性や正確性又はこの情報が特定の目的に役立つことの保証を行いません。SAP は、関連情報の使用により発生した損害が、SAP の重大な過失又は意図的な違法行為が原因で発生したものでない限り、その損害に対して一切責任を負いません。リンクの分類に関しては、<http://help.sap.com/disclaimer> を参照してください。



www.sap.com/contactsap

© 2014 SAP AG or an SAP affiliate company. All rights reserved.

本書のいかなる部分も SAP AG の明示的許可なしに、いかなる形式、目的を問わず、複写、または送信することを禁じます。本書に記載された情報は、予告なしに変更されることがあります。

SAP AG がライセンス、またはその頒布業者が頒布するソフトウェア製品には、他のソフトウェア会社の専有ソフトウェアコンポーネントが含まれています。製品仕様は、国ごとに変わる場合があります。

これらの文書は SAP AG およびその関連会社（「SAP グループ」）が情報提供のためにのみ提供するもので、いかなる種類の表明および保証を伴うものではなく、SAP グループは文書に関する錯誤または脱漏等に対する責任を負うものではありません。SAP グループの製品およびサービスに対する唯一の保証は、当該製品およびサービスに伴う明示的保証がある場合に、これに規定されたものに限られます。本書のいかなる記述も、追加の保証となるものではありません。

SAP、および本書で言及されるその他 SAP の製品およびサービス、ならびにそれらのロゴは、ドイツおよびその他諸国における SAP AG の商標または登録商標です。

商標に関する情報および表示の詳細については、<http://www.sap.com/corporate-en/legal/copyright/index.epx> をご覧ください。