Welcome to the topic on queries in SAP Business One.
In this topic, you will learn to create SQL queries using the SAP Business One query tools – Query Wizard and Query Generator. You will also see how to restrict access to saved queries.
Queries enable you the ability to quickly display data from SAP Business One company database tables. There are several uses for queries:

- With SAP Business One customizations: alerts, approval procedures, and user-defined values
- As a simple way of producing an ad hoc report
- As the first step in creating a custom report, you can quickly generate a sample to discuss with your customer
- To validate the contents of a table during data migration

Queries enable you to quickly display and format data from an SAP Business One company database. You can use queries in several ways, for example:

- with customization tools (alerts, approval procedures, and user-defined values added to existing fields or user-defined fields)
- to produce ad hoc reports for users
- to generate a sample report as an initial step when you create a custom report,
- to validate the contents of a table or field during data migration. You can quickly display all the imported fields for a table.

You should understand that the query tools are primarily designed to produce ad hoc reports and for use with customization tools. If you need to produce complex, fully formatted reports or print layouts, you should use Crystal Reports.
SAP Business One provides two different query tools that you can access from the SAP Business One client:

- Query Wizard
- Query Generator

Both tools assist you in creating queries using the structured query language (SQL). SQL is a standardized set of commands for accessing and formatting data in relational databases.

Although both tools produce the same results in the end, you should understand their differences.
The first tool, the Query Wizard, guides you step-by-step through the process of creating a query. You should use this tool if you do not have much experience with SQL commands and syntax. The system walks you through the process of creating a SQL query and generates the SQL statements in the background so you do not require precise SQL syntax knowledge. However, because this is done by a wizard it takes several steps to produce the query.

You should also find the Query Wizard easier to use if you do not have much familiarity with SAP Business One tables and how they are related to each other. When you select a table, the Query Wizard will show you all tables that are related to your chosen table, allowing you to select data from multiple tables.
The second tool, the Query Generator, allows you to create the SQL in a single screen. If you have some basic SQL knowledge, you will find the Query Generator much faster than the Query Wizard.

The system displays the full SQL command and allows you to edit the SQL directly if necessary.

You can also enter an entire SQL statement directly into the Query Generator.

Like the Query Wizard, the Query Generator will automatically supply the inner join when you select more than one table for a query. Although it does not show you the list of tables related to your chosen table, fields that are keys to related tables are shown in **bold**. You can drag and drop the bolded field to the table selection column and the Query Generator will open the related table, allowing you to select fields from that table.
In order to run queries you need to know the names of the SAP Business One tables and fields.

In SAP Business One, many objects are represented by more than one database table. For example, the sales order document uses different tables for the header and document rows – the table ORDR stores the header data and the table RDR1 stores the document rows.

SAP Business One uses this convention for all marketing documents - the header table starts with “O” and the row table uses the last three letters of the header table name.
You can use *System Information* to display the table and field names for a window that you are viewing in SAP Business One.

1. First you toggle on the functionality from the *View* menu. The short cut key is **Control plus Shift plus I**.

2. Next, open a document or window in SAP Business One. In our example here we have opened the Business Partner Master Data window.

3. Now, when you move your mouse over a field (for example the Code), the second line of the status bar at the bottom of the screen shows the name of the table (OCRD) and the field name from the table (CardCode).

You now have the table name and field name to enter in your SQL query.

Other information about the field is displayed, such as the maximum length of the field.
The system information also shows, in addition to the table and field name, the *item number* and *column number* for the field.

For example, in a sales order, the item number for the *CardCode* field is 4. The column number for the *CardCode* field is not displayed, since this is a header field and the column number for all header fields is 0.

If you open other marketing documents, you will see that the *CardCode* field is always 4. Although the table changes for each document type, the item number for a field is consistent.
When you refer to fields in the row area of a marketing document, the column number is displayed in addition to the item number. In the example here you can see that the ItemCode field in the sales order document has item number 38 and column number 1.

You will notice that the item and column number for a field is the same across similar document types, for example, the ItemCode has item number 38 and column number 1 in all sales and purchasing documents.

Why is this useful? You can use the item and column number instead of the table and field name in a query. This makes it possible to use the same query for multiple document types. Obviously, a query that used the table and field name would not work across multiple document types.
You will find that the table and field name is not displayed for some fields, mostly totals and prices. One example is the document total field in marketing documents. In a document this field is concatenated with the currency symbol, whereas in the database the amount and currency are stored in different columns. Therefore this field does not exist in the database.

You can still run a query using these fields, you just need to look up the field name in the query tool. An easy way to do this is to create a document, then run a query on the table and select all fields.
In this demo you will see the System Information display.
The SAP Business One query tools will help you assemble the various elements of the query, but you still need to have some knowledge of SQL syntax.

A query, or the underlying SQL statement, contains one or more of the basic elements listed on the slide:

- Selection of fields
- Conditions for selection (where clause)
- Sort order (order by clause)
- Grouping and summarizing (group by clause)

SQL also provides functions to perform calculations on data. In the sample query the `getdate()` function is used to return the current date and time.

The sample query shown in the top right part of the screen will show information for open purchase orders posted in the last 7 days.
The query selects data from the purchase order table. The name of this table is OPOR.
The results of the query, a simple snapshot from the database, is shown.
Note that SQL is not case sensitive, therefore the commands do not have to be upper case.
This demo will show how to create and run a simple query using the Query Wizard.
The Select statement is the only mandatory part of a query. The Select command is normally used to fetch and display table fields. In the example, we want to display the document number, business partner code, business partner name and total amount from the Purchase Order table OPOR.

When the query runs, headings for the report columns are taken from the database column names, but you can change these in SQL using the “as” keyword and entering the heading name in quotes. For example, to change the column heading for the DocTotal field, use Select DocTotal as ‘Total Amount’.

You can also specify calculation fields that display the result of an addition, subtraction, multiplication, or division of two fields.

If you need more than one table for a query, you normally need to create relationships between them. The Query Wizard and Query Generator tools make this easy for you by automatically creating joins. For example, if you select the purchase orders table (OPOR) and also select the business partners master data table (OCRD), the query tools will link these tables using the business partner code which is common to both tables.
The optional **Where** clause lets you select only records that meet specified criteria. For example, in our sample query, only purchase orders that are open (**DocStatus 'O'**) will be displayed. To find out the possible values for a field such as **DocStatus**, the easiest way is to run a query on the table and select the field name. You will see the possible values that are stored in the database.

In the **Where** clause you can include:

- Fixed conditions as comparisons
- Calculations and functions
- **AND** and **OR** operators. In the example query, we only want to include purchase orders that were posted in the last 7 days; therefore we use the **AND** operator to add a second condition to match the posting date value to the current date – 7.
- Variables. Variables are specified as [%0], [%1], [%2], etc. When you include a variable, the user will be prompted to enter a value as a parameter when the query runs.

```sql
SELECT DocNum, CardCode, CardName, DocTotal
FROM OPOR
WHERE DocStatus = 'O' AND (DocDate > getdate() -7)
ORDER BY DocDate
```
You can optionally sort the results by adding the **Order By** clause to the query. The results will be sorted by default in ascending sequence using the specified field. In this example the rows of results (purchase orders) will be sorted by posting date (**DocDate**).

You can sort by descending sequence by adding the keyword **DESC**

You can sort by multiple fields, and these fields can be either part of the select clause or other fields in any of the tables of the query.
The optional **Group By** clause allows you to display the query results grouped or summarized by a specified field, for example, by business partner. In our example, the query has been rewritten to use the Group By element and is shown underneath the original query on the slide.

The grouped results are collected into sets using the Group By field or fields as the common value.

*Group by* is usually used in conjunction with a mathematical (aggregate) function, such as *Count or Sum*. 
When the query runs it will count the number of open Purchase Orders, then sum the total amount of each PO. The selected fields are displayed according to the Group by clause, therefore the results are consolidated by vendor, and the query shows one consolidated row for each vendor, with the count and the total amount of all open purchase orders for the vendor.

Notice that we need to supply the column headings for the fields that are counted and summed, since these fields are not in the database. We supply these headings in the Select statement, for example, SUM(DocTotal) as ‘Total Amount’. The heading name must be encased in quotation marks.

You can group by more than one field. One word of caution though; the fields that you use in the Select statement must appear in either the Group By clause or the aggregate function.
In this demo, you will see how to create a query using the Query Generator.
You can save a query for later use, or for other users to run. When you save a query you must assign it to a category. Categories are used to organize related queries.

The General category is provided in the system, but you can add your own categories too. In the example here we have categories for sales, marketing, and purchasing queries.

Queries that you save are located as User Queries. To run a saved user query, choose the Tools menu in SAP Business One, and then select the category and the query name.
When a new user is added to the system, the user does not have any authorization to run saved queries. Each user must be granted authorization to run a saved query. There are two parts to the authorization process:

1. When you create a new category, you select one or more report authorization groups for the category.
2. To authorize a user to run the reports in the category, you assign the user to the report authorization group. You can find this authorization in the general authorization hierarchy under Reports > Query Generator > Saved Queries – Group No.

There are 15 different report authorization groups, and you must select at least one authorization group, or you will not be able to save the category.

<table>
<thead>
<tr>
<th>Query Categories</th>
<th>Authorization Groups</th>
<th>General Authorization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td></td>
<td>Reports &gt; Query Generator &gt; Saved Queries – Group No.</td>
</tr>
<tr>
<td>Orders with Discount &gt; 10% Orders by Sales Employees</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Marketing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sales Analysis by Product Customer Warranty Expiry</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Purchasing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stock Near Low Threshold Service POs over 5000</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2013 SAP AG. All rights reserved.
Once a user is authorized to a report authorization group, they can then run all queries saved in any category with that authorization group.

Let us look at the example provided here.... Here you can see that authorization groups 1 and 2 are assigned to the category called Sales. Authorization groups 2 and 3 are assigned to the category called Marketing, and authorization group 4 to the category Purchasing.
Users Bill and Donna are assigned to authorization group 1. Users Sophie and Tim are assigned to authorization group 2.

So, Bill and Donna can run all queries saved in the Sales category through its association with authorization group 1. Sophie and Tim can run all queries saved in both the Sales and the Marketing categories through the association with authorization group 2.

If you want to make saved queries available to end users, you need to carefully plan the categories and the authorization groups to ensure that users get authorization only to the correct reports.

**Note:** You can assign full or read-only authorization to a user, as desired. You can also use general authorizations to define which users can create new queries, and which users can modify the existing SQL syntax in a saved query.
This demo will show how to save and manage queries.
Here are some key points to take away from this session. Please take a minute to review these key points:

- SQL queries can be used in conjunction with SAP Business One customization tools, to produce ad hoc reports, as the first step to design a custom report, and to validate migrated data in tables.
- View → System Information can help you to identify table and field names, or item and column numbers, for use in queries.
- There are two tools to help you create SQL queries – the Query Wizard and the Query Generator.
- You can save queries as user queries and organize them by category.
- Users need authorization to run saved user queries. First select an authorization group for the category, then assign to the user the general authorization for the category’s authorization group.
You have completed the topic for queries in SAP Business One. Thank you for your time!