



## Beginners and Intermediate Training for SAP Business One Users

June 2017

Created By: Oren Tabib

2017 PIONEER B1 LLC. ALL RIGHT RESERVED.

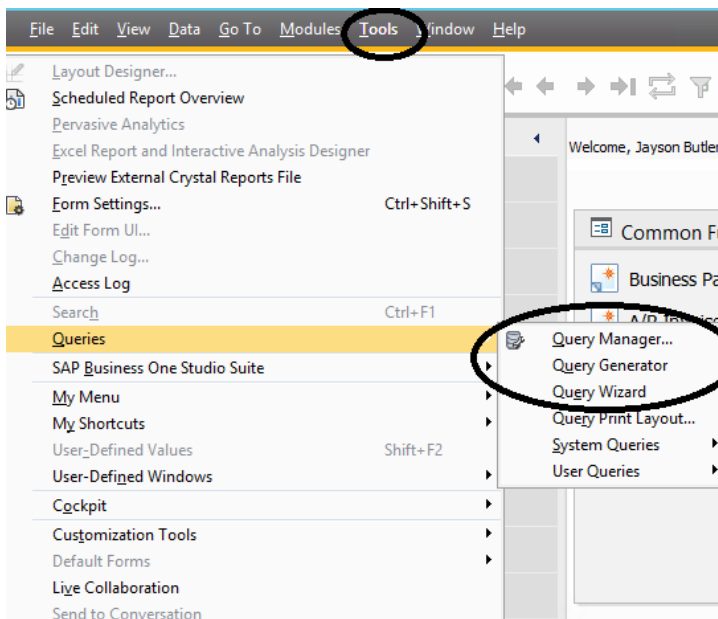
## Table of Contents

|  |    |
|--|----|
| Chapter 1: Getting started with reporting tools .....                  | 3  |
| Chapter 2: Query Manager, Query Security and Printing Options.....     | 4  |
| Chapter 3: Understanding Main SAP Tables and Tables Relationships..... | 5  |
| 1. How to find the relevant table and fields.....                      | 5  |
| 2. Main tables structure in SAP Business One.....                      | 6  |
| 3. Basic Structure of SQL Select Statement.....                        | 6  |
| 4. SQL and Tables Joins:.....  | 7  |
| Chapter 4: Filters, Variables, Sorting, Grouping .....                 | 8  |
| Chapter 5: Conditions, Nested Queries, Combine Queries .....           | 9  |
| 1. Conditional statement - Case statement.....                         | 9  |
| 2. Nested queries (Subqueries) .....                                   | 9  |
| 3. Union Statement .....   | 9  |
| Chapter 6: Formatted Search Queries.....                               | 10 |
| Chapter 7: Connect queries with alerts and approval procedures .....   | 11 |
| 1. Alert management functionality .....                                | 11 |
| 2. Approval procedure functionality .....                              | 11 |
| Chapter 8: Stored Procedures .....                                     | 13 |
| Chapter 9: Excel Connection .....                                      | 14 |

## Chapter 1: Getting started with reporting tools

SAP Business One query tools enable you to create queries using the SQL query engine. Tools are designed for data retrieval/selection only, and not for updates, insert and delete.

- Query Wizard
- Query Generator
- SQL Studio Management
- Crystal Reports
- Excel
- Additional tools
  - BI – Business Intelligent Software
  - HANA Analytics



- **Exercise:**
  - Connect to SAP Business One and access different query tools

## Chapter 2: Query Manager, Query Security and Printing Options

In this chapter, you will learn how to navigate and maintain queries in the query manager window and use the categories to manage security and authorizations as well as to organize queries in a logical way.

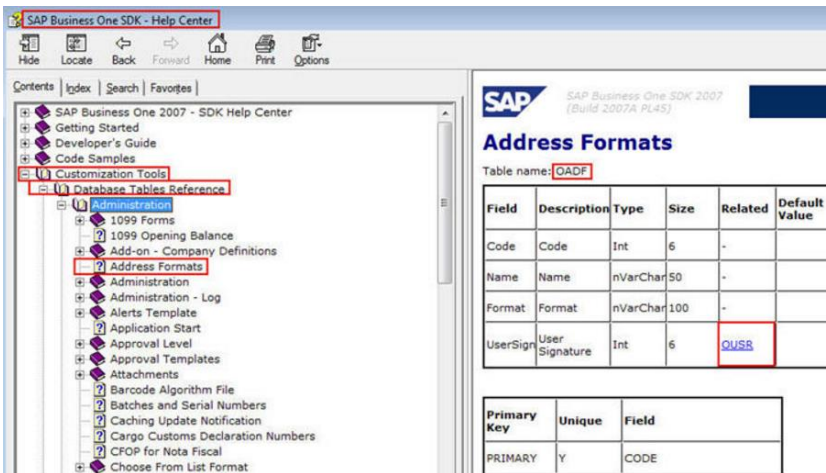
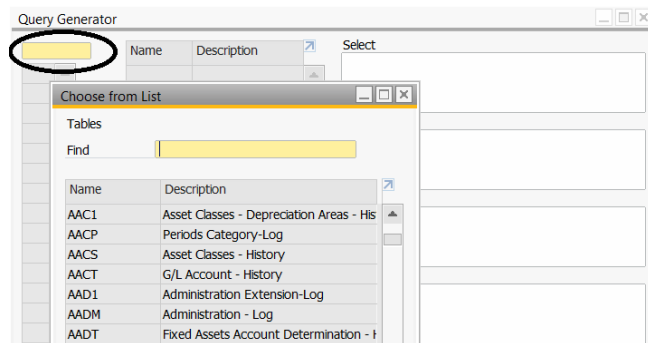
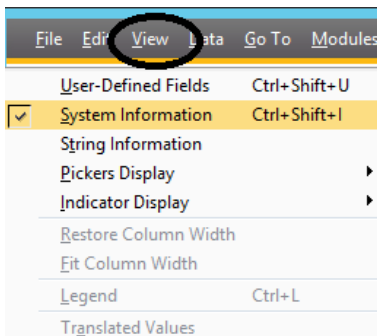
In addition, you will get familiar with different SAP tools related to queries such as printing, exporting, filtering and more.

- Query manager options
- Saved queries and categories
- Authorizations and security
- Filters and sorting within query results
- Total a column
- Export to excel
- Layout and printing a query
  
- **Exercise 1:**
  - Create a query using any of the tools and save a query under the general category, then close all windows and execute the same query from the query manager area
  
- **Exercise 2:**
  - Create a new query that present all open sales orders: Number, customer code and name, date and due date, freight and total
  - Save this query under a selected category
  - Make sure you see the grand total in the total column
  - Filter only one customer and send the results to excel
  - Create a print layout from the query

## Chapter 3: Understanding Main SAP Tables and Tables Relationships

### 1. How to find the relevant table and fields:

In most cases View>System information will show the table and field name in the message area of SAP on the bottom left. You can also see all relevant tables if you tab at the table section of the query generator. In some cases you may need to use the SDK help center.



#### • Exercise 1:

- Identify which table stores the information about customers
- Identify which table stores the information about Items
- Identify which table stores the information about sales order
- Identify which table stores the information about sales invoices
- Identify which table stores the information about purchase orders

### 2. Main tables structure in SAP Business One

- The table name will give us a hint of which table it represents, e.g: ORDR – Sales Orders, OITM=Item Master Data etc.
- Documents tables normally contain header and lines as well as other related tables. Below you will find main tables related to sales order table, though the same structure apply to all marketing documents:
  - ORDR - Sales Order Header
  - RDR1 - Sales Order Lines
  - RDR2 - Sales Order - Freight - Rows
  - RDR3 - Sales Order - Freight
  - RDR4 - Sales Order - Tax Amount per Document
  - RDR6 - Sales Order - Installments
  - RDR 20 - Sales Order - Bin Allocation Data
  - RDR 21 - Sales Order - Document Reference Information
- History Tables
  - History tables are stored under ADOC table
  - This table save information of all objects and object type needs to be filtered
- Draft Tables
  - Main tables are DRF1 and ODRF and follow the same tables structure of marketing documents
  - Those tables save information of all objects and object type needs to be filtered – E.g. Object Type='17' will show only sales orders drafts

### 3. Basic Structure of SQL Select Statement:

There is some good literature on how to structure and write SQL reports in a way that will be organized and assist other people to read the SQL statement and potentially modify it.

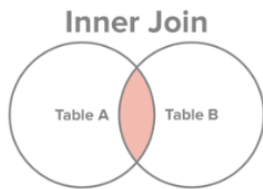
Below you will find a basic structure of any SQL query that will help you read other SQL reports as well as write queries with consistency.

**SELECT** [Fields names separate by commas]  
**FROM** [Table names separate by Joins]  
**WHERE** [Conditions]

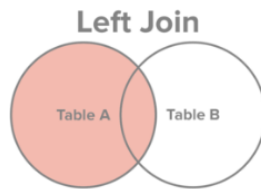
**GROUP BY** [Group the report – fields that are not grouped needs to be summaries]  
**HAVING** [Condition on grouped fields]  
**ORDER BY** [Sort the report – Descending/ Ascending]

#### 4. SQL and Tables Joins:

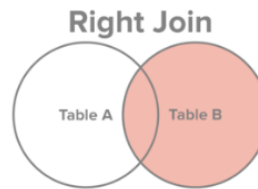
There are four basic types of SQL joins: inner, left join, right join, and full join. The easiest and most intuitive way to explain the difference between these four types is by using a Venn diagram, which shows all possible logical relations between data sets.



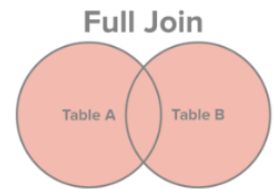
Select all records from Table A and Table B, where the join condition is met.



Select all records from Table A, along with records from Table B for which the join condition is met (if at all).



Select all records from Table B, along with records from Table A for which the join condition is met (if at all).



Select all records from Table A and Table B, regardless of whether the join condition is met or not.

Example of join statement structure:

```
SELECT T1.[DocNum], T1.[CardName], T0.[ItemCode], T0.[ItemCode], T0.[Dscription]
FROM RDR1 T0
INNER JOIN ORDR T1 ON T0.[DocEntry] = T1.[DocEntry]
```

- **Exercise:**
  - Items availability Report - Create a new query on Items table.
  - Show item code, items name, on hand Quantity, committed and On Order
  - Execute the report to make sure it's working and save it
  - Add to the report one field 'main vendor' – execute the report and make sure you get the same number of results

## Chapter 4: Filters, Variables, Sorting, Grouping

In this chapter you will learn how to use filters and variables using SQL as well as specific syntax relevant for SAP Business One (% variables).

- Where options
- Using SAP variables - %1
- Order by statement
- Group by statement
- Advanced - Variables declaration

```
/* SELECT FROM [dbo].[OINV] TO */  
DECLARE @CurrFrom AS Date  
/* WHERE */  
SET @CurrFrom = /*T0.DocDate */ ' [%0]'
```

Documentation:

How to Customize Printing Layouts

[http://www.pioneerb1.com/wp-content/uploads/2012/04/HowTo\\_PrintLayouts\\_PLD\\_88.pdf?x42527](http://www.pioneerb1.com/wp-content/uploads/2012/04/HowTo_PrintLayouts_PLD_88.pdf?x42527)

- **Exercise 1:**
  - Write a query to present all sales invoices between dates
  - Sort the query above by document date
  - Add a condition to show only customers that have the letter 'a' in the customer name field
- **Exercise 2:**
  - Write a second query to show a total of sales invoices by customer (one line per customer)
- **Exercise 3 (if time permits):**
  - Write a query to show a total of sales invoices by sales rep. The report should have a parameter to choose the sales rep from a list of reps.



## Chapter 5: Conditions, Nested Queries, Combine Queries

In this chapter you will get yourself familiar with advanced SQL statements that will enhanced your SQL knowledge and will enable you to create more complicated reports.

### 1. Conditional statement - Case statement:

The CASE statement is SQL way of handling if/then logic. Below is the syntax:

```
SELECT
    CASE WHEN [Condition] THEN [Result]
    ELSE [Result] END
FROM [Table Name]
```

#### • Exercise:

- Build a query of all vendors' POs from 1/1/2017 – select your favorit 6 columns to present
- In addition to the above, present a column called size:
  - If total is smaller than \$100 it will show 'Small'
  - If total between \$100 and \$1,000 it will show 'Medium'
  - If total is above \$1,000 it will show 'Large'

### 2. Nested queries (Subqueries)

Nested query is a query within another SQL query and embedded within the WHERE clause. A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

#### • Exercise:

- Create a query to present basic client information of clients that bought more than twice (have at least 3 invoices or more).

### 3. Union Statement

The SQL UNION operator is used to combine the result sets of 2 or more SELECT statements. It removes duplicate rows between the various SELECT statements. Each SELECT statement within the UNION must have the same number of fields in the result sets with similar data types.

#### • Exercise:

- Write a query to show all customers sales (invoices minus credit memos)

## Chapter 6: Formatted Search Queries

The Formatted Search function enables you to enter values, originated by a pre-defined search process, into any field in the system (including user-defined fields).

Formatted search function can be used to enter values automatically into fields as well as create dependencies between fields in the system. For example, the value in field X influences the value in field Y.

To retrieve data from the active window, you need to create a special query. SAP Business One provides two different syntaxes for this type of query:

- `$(Tablename.Fieldname)`
  - Tablename – Indicates the table to which the active window relates.
  - Fieldname – The field from the active window.
- `$$[Field Index.Field Column.Number/Currency/Date/0]`
  - Field Index – You can relate to each field in the active window using a unique index.
  - Field Column - If the required field is a table field in marketing documents. If the required field is a header field, set this element to 0.
  - Number / Currency / Date / 0 – Use the relevant data field or use 0 for AlphaNumeric

Documentation:

How to Define and Use Formatted Search

<http://www.pioneerb1.com/wp-content/uploads/2012/09/How-to-Define-and-Use-Formatted-Search.pdf?x42527>

- **Exercise:**
  - Go to AR invoice and add FMS in the remarks field to show the business partner account balance

## Chapter 7: Connect queries with alerts and approval procedures

### 1. Alert management functionality

User-defined alerts can be used to notify users of complex or unique business situations. You can also use user-defined alerts to create routine task lists for users.

Documentation: Alert Management

[https://help.sap.com/doc/saphelpiis\\_hc\\_b1\\_image\\_repository\\_consultant\\_training\\_basic\\_b1\\_90\\_tB1200\\_03\\_04\\_pdf/9.0/en-US/B1\\_90\\_TB1200\\_03\\_04.pdf](https://help.sap.com/doc/saphelpiis_hc_b1_image_repository_consultant_training_basic_b1_90_tB1200_03_04_pdf/9.0/en-US/B1_90_TB1200_03_04.pdf)

- Exercise:
  - Display weekly alert for financial users of all AP invoices that are over \$1,000

### 2. Approval procedure functionality

Select Distinct 'True' is advised to execute any approval procedure query.

Example 1:

Approval for any quote for customer 'AAA'

```
SELECT distinct 'True'  
From OQUT T0  
WHERE T0.CardCode = 'AAA'
```

Example 2:

Approval for any purchase order with UDF BaseRef set to 'N'

```
SELECT distinct 'True'  
FROM OPOR T0  
WHERE $[OPOR.U_BaseREF] = 'N'
```

Example 3:

Approval for any delivery document for a customer with open A/R Invoices that have overdue **DocDueDates**

```
SELECT DISTINCT 'True'  
FROM OINV T0  
WHERE DateDiff(DD,GetDate(),T0.DocDueDate) > 0  
AND T0.CardCode = $('[4.0.0]'  
AND T0.DocStatus='O'
```

Documentation: Example of complicated approval procedure

<https://wiki.scn.sap.com/wiki/pages/viewpage.action?pageId=96403575>

- **Exercise:**
  - Create an approval procedure for every Goods Issue document created without the word 'Reason' in remarks field.

### Chapter 8: Stored Procedures

A stored procedure is a prepared SQL code that you save so you can reuse. So if you think about a query that you write repeatedly, instead of having to write that query each time you would save it as a stored procedure and then just call the stored procedure. In addition, you can also pass parameters to the stored procedure, so depending on what the need is the stored procedure can act accordingly based on the parameter values that were passed.

SAP Business One database has many stored procedures out of the box that enable different features and system reporting. You are not allowed to modify existing stored procedures other than the 2 listed below. You can create new stored procedures as you wish.

The 2 stored procedures that you can modify are:

- Transaction Notification
  - This stored procedure is checking logic before an object being added or updated
  - You can use this SP in order to apply logic, mandatory fields, restrictions etc.
  - Syntax Example – Block PO if Vendor Ref Number is null

```
if @object_type = '22' and @transaction_type IN (N'A', N'U')
begin
```

```
Declare @NumAtCard as nvarchar(100)
Select @NumAtCard=NumAtCard from OPOR where DocEntry =
@list_of_cols_val_tab_del
```

```
If @NumAtCard Is Null or @NumAtCard=""
begin
set @error = 1
set @error_message =N'SP_PB1 Vendor Ref Number is Mandatory'
end
end
```

- Post Transaction Notice
  - This stored procedure is being executed after every object is added or updated
  - You can use this SP to update User Defined Tables and Fields after a transaction is executed.
- Exercise:
  - Block adding a sales order if remarks field is null

## Chapter 9: Excel Connection

Microsoft Excel provides capabilities to integrate with SQL databases that can turn your excel into a very powerful analytics tool. Utilizing the format options in Excel and pulling real time data from SAP database can provide comprehensive reports that can be shared throughout the organization.

Connection to a database:

- If you never connected to SQL, you will need to establish a new connection following the screenshot below
- Once a connection to the database have been established, you can have options like saving the credentials and refreshing the data every so often
- Once connecting to a data source, you are able to retrieve information from any table and view you have access to
  - Tip: It's easier to first create a view in the system gathering the information you need from different tables and then connect Excel directly to the view

