















### 2.4.3 Preliminary check of notification definitions

Since the notification definitions are SQL queries, they can be checked preliminarily for correctness in SQL Server Management Studio. Here, you also get a preview of the result.

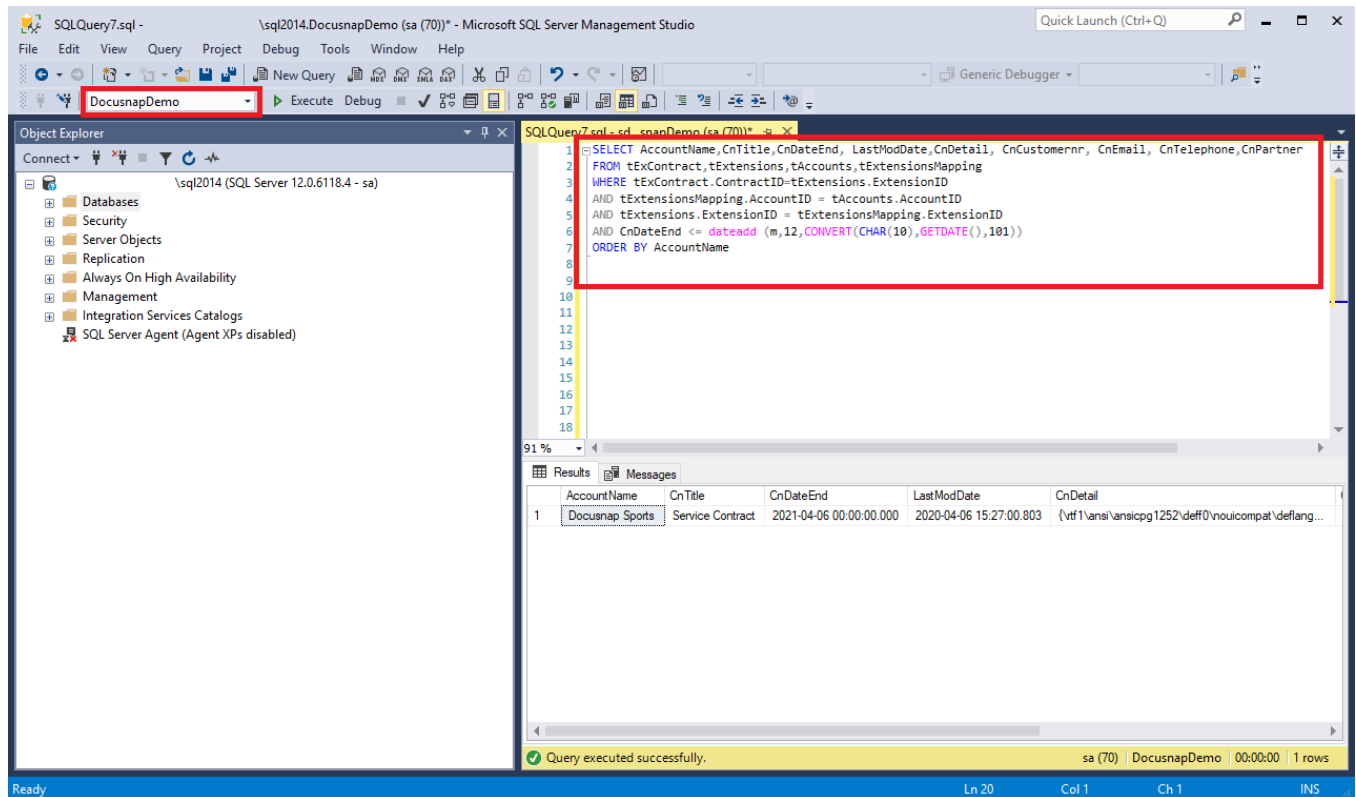


Fig. 3 - Accessing the SQL queries in SQL Server Management Studio

Important: Make sure to run the SQL query with the correct database.



However, the placeholders used in Docusnap must be replaced for a query in SQL Server Management Studio:

Meaning	Placeholder in Docusnap	Replacement in SQL Server Mgmt Studio	Remark
Today	{NOW}	'04/20/15'	Make sure to indicate the date in the correct format as used in the database. For more information, see the SQL date / time format section.
Domain	{DOMAIN}	DomainID	The Docusnap domain IDs can be displayed via the following SQL query:  <code>SELECT TOP 100 [DomainID],[DomainName] FROM [tDomains]</code>
Company	{ACCOUNT}	AccountID	The Docusnap account IDs can be displayed via the following SQL query:  <code>SELECT TOP 100 [AccountID],[AccountName] FROM [tAccounts]</code>

## 2.4.4 Contracts

The following SQL query lists all expired contracts assigned to an object:

```
SELECT AccountName, CnTitle, CnDateEnd, CnDetail
FROM tExContract, tExtensions, tAccounts, tExtensionsMapping
WHERE tExContract.ContractID=tExtensions.ExtensionID
AND tExtensionsMapping.AccountID = tAccounts.AccountID
AND tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
AND CnDateEnd <= {now}
```

A modified version supplies additional information on the expired contract (e.g. associated e-mail address or phone number):

```
SELECT AccountName, CnTitle, CnDateEnd, LastModDate, CnDetail, CnCustomernr, CnEmail, CnTelephone,
CnPartner
FROM tExContract, tExtensions, tAccounts, tExtensionsMapping
WHERE tExContract.ContractID=tExtensions.ExtensionID
AND tExtensionsMapping.AccountID = tAccounts.AccountID
AND tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
AND CnDateEnd <= {now}
```

Please note that only contracts assigned to an object are considered in the above statements. The reason for this is the use of INNER JOINS with the included table "tExtensionsMapping".

If Contracts are also to be considered which are not assigned to any object, then OUTER JOINS must be used for this purpose, which is described by means of an example in the next subitem.

## 2.4.5 Contracts with no object assignment

The following SQL query lists all expired contracts:

```
SELECT tEC.CnTitle, tEC.CnDateEnd, tEC.CnDetail, tA.AccountName FROM tExContract tEC
LEFT OUTER JOIN tExtensions tE on tE.ExtensionID = tEC.ContractID
LEFT OUTER JOIN tExtensionsMapping tEM ON tEM.ExtensionID = tE.ExtensionID
LEFT OUTER JOIN tAccounts tA ON tEM.AccountID = tA.AccountID
AND CnDateEnd <= {now}
```

A modified version lists contracts which expire within one month or earlier. This data is output with additional information on the contracts:

```
SELECT AccountName, CnTitle, CnDateEnd, LastModDate, CnDetail, CnCustomernr, CnEmail, CnTelephone,
CnPartner
FROM tExContract, tExtensions, tAccounts, tExtensionsMapping
WHERE tExContract.ContractID=tExtensions.ExtensionID
AND tExtensionsMapping.AccountID = tAccounts.AccountID
AND tExtensions.ExtensionID = tExtensionsMapping.ExtensionID
AND CnDateEnd <= dateadd (m,1, CONVERT (CHAR (10), GETDATE (), 101))
ORDER BY AccountName
```

















## LIST OF FIGURES

FIG. 1 - ACCESSING THE NOTIFICATION DEFINITION.....	5
FIG. 2 - CREATING A NOTIFICATION DEFINITION.....	6
FIG. 3 - ACCESSING THE SQL QUERIES IN SQL SERVER MANAGEMENT STUDIO.....	8
FIG. 4 – TABLE FOR THE DIFFERENT HOST TYPES.....	12
FIG. 5 – NOTIFICATION WIZARD.....	13
FIG. 6 - CREATE NOTIFICATION .....	14

LIST OF TABLES TABLE 1: FORMAT CODES.....	17
--	----

## VERSION HISTORY

---

<b>Date</b>	<b>Description</b>
1/3/2016	Statement modified (2.4.6)
3/3/2017	List of figures modified
24/10/2018	Screenshots updated; Statement modified (2.4.5, 2.4.7 und 2.4.8); Insert Version history
29/04/2020	Version 2.0 - Revision of the HowTo for Docusnap 11

---

