



Revision 20200522

NOTE

This document is confidential and proprietary of **Denodo Technologies**.
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2023
Denodo Technologies Proprietary and Confidential

CONTENTS

1 WHAT ARE MICROSOFT SQL SERVER LINKED SERVERS?.....	2
2 HOW TO CONFIGURE A SQL SERVER LINKED SERVER TO CONNECT TO DENODO?.....	3
3 ACCESSING AND QUERYING DENODO THROUGH SSMS.....	7
4 HOW TO ACCESS SQL SERVER LINKED SERVER FROM DENODO?	9

Accessing Microsoft SQL Server Linked Servers

1 WHAT ARE MICROSOFT SQL SERVER LINKED SERVERS?

“Linked Server” is a feature of MS SQL Server that allows access to non-SQL Server databases through SQL Server. Only SQL Server Developer and SQL Server Enterprise versions support Linked Servers. SQL Server Express edition has several limitations that do not allow configuring or using Linked Servers.

A Linked Server enables the SQL Server Database Engine to execute commands against OLE DB data sources outside of the instance of SQL Server. Many types of OLE DB data sources can be configured as linked servers such as Microsoft Access, Excel, another instance of SQL Server, etc.

Linked servers offer the following advantages:

1. Remote server access.
2. The ability to issue distributed queries, updates, commands, and transactions on heterogeneous data sources across the enterprise.
3. The ability to address diverse data sources similarly.

A linked server definition specifies the following objects:

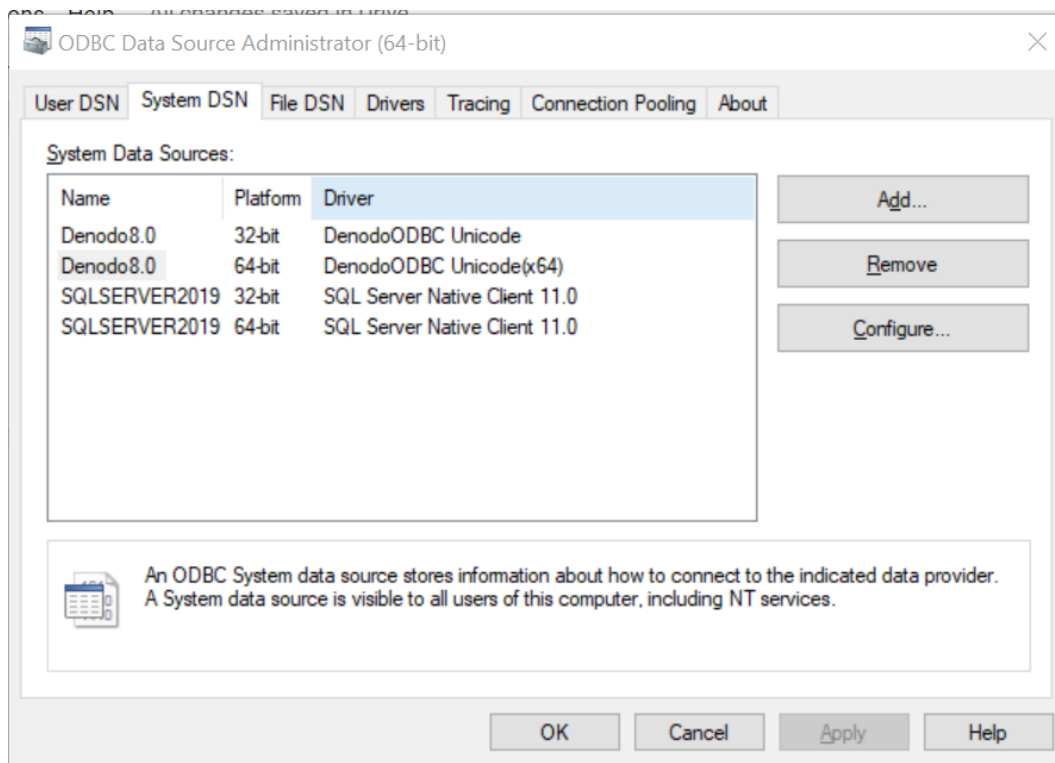
- An *OLE DB provider*: DLL that manages and interacts with a specific data source.
- An *OLE DB data source*: Identifies the specific database that can be accessed through OLE DB (although data sources queried through linked server definitions are ordinarily databases, OLE DB providers exist for a variety of files and file formats).

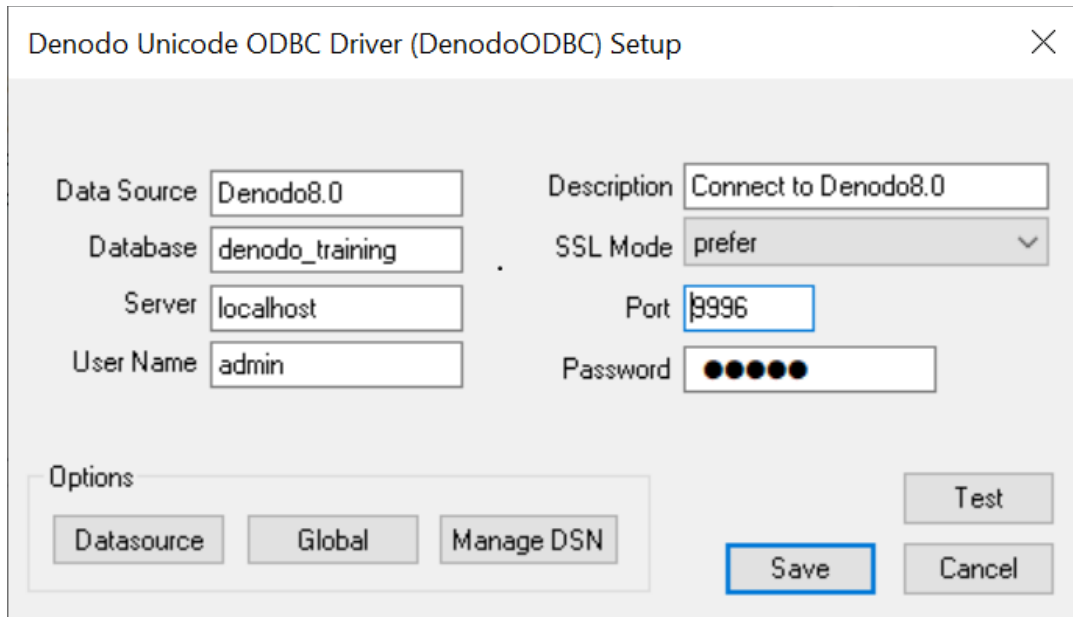
2 HOW TO CONFIGURE A SQL SERVER LINKED SERVER TO CONNECT TO DENODO?

Virtual DataPort provides access to ODBC clients through the PostgreSQL ODBC driver. To install the driver, follow the steps of the Virtual DataPort Developer Guide Section "[Access through ODBC](#)".

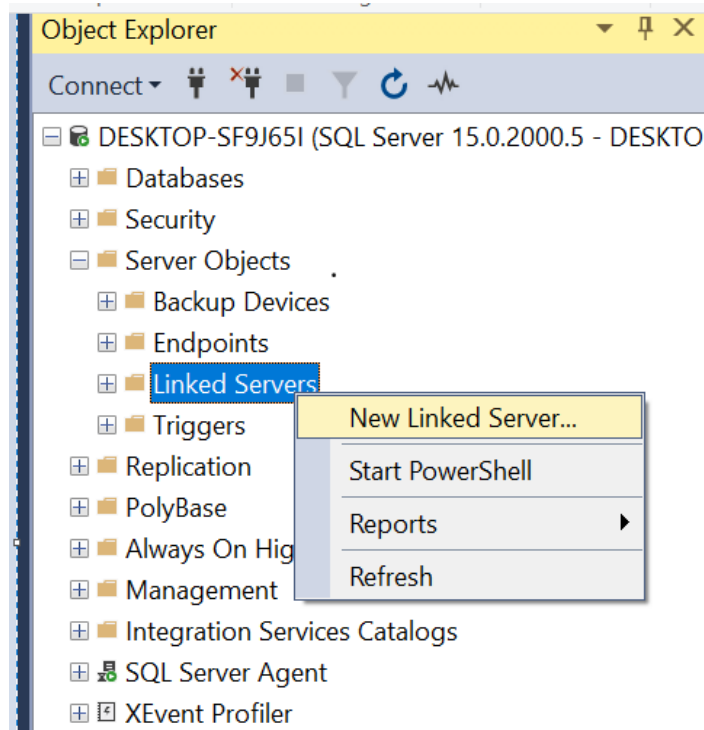
Once the DSN to a Denodo database is created on the system, a Linked Server to Denodo can be configured.

1. Create a DSN to a Denodo database.

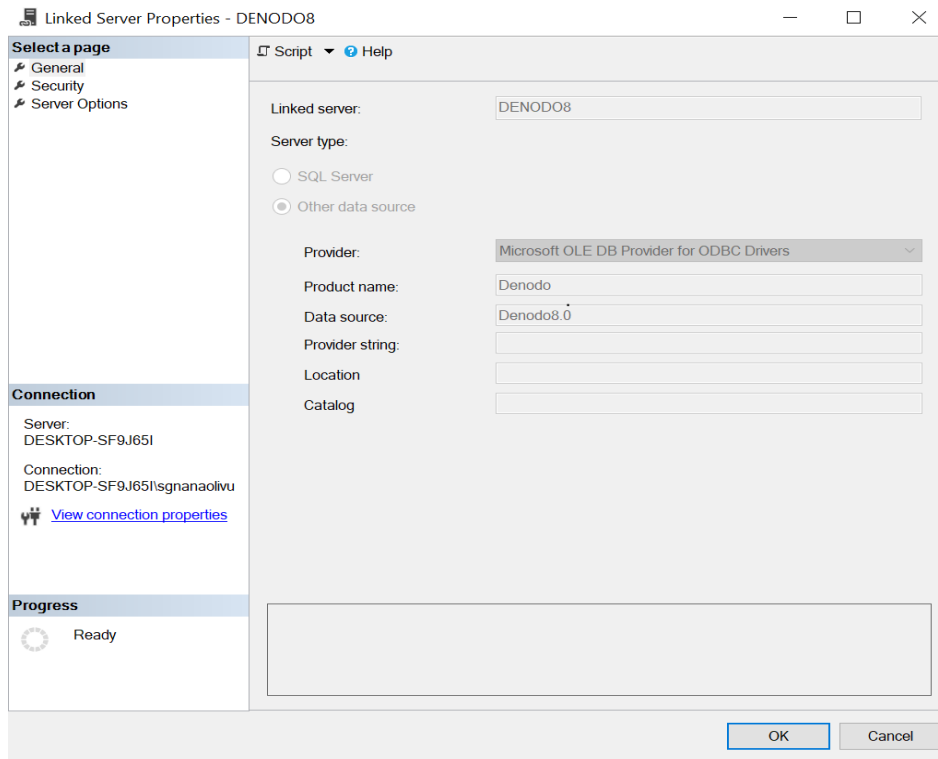




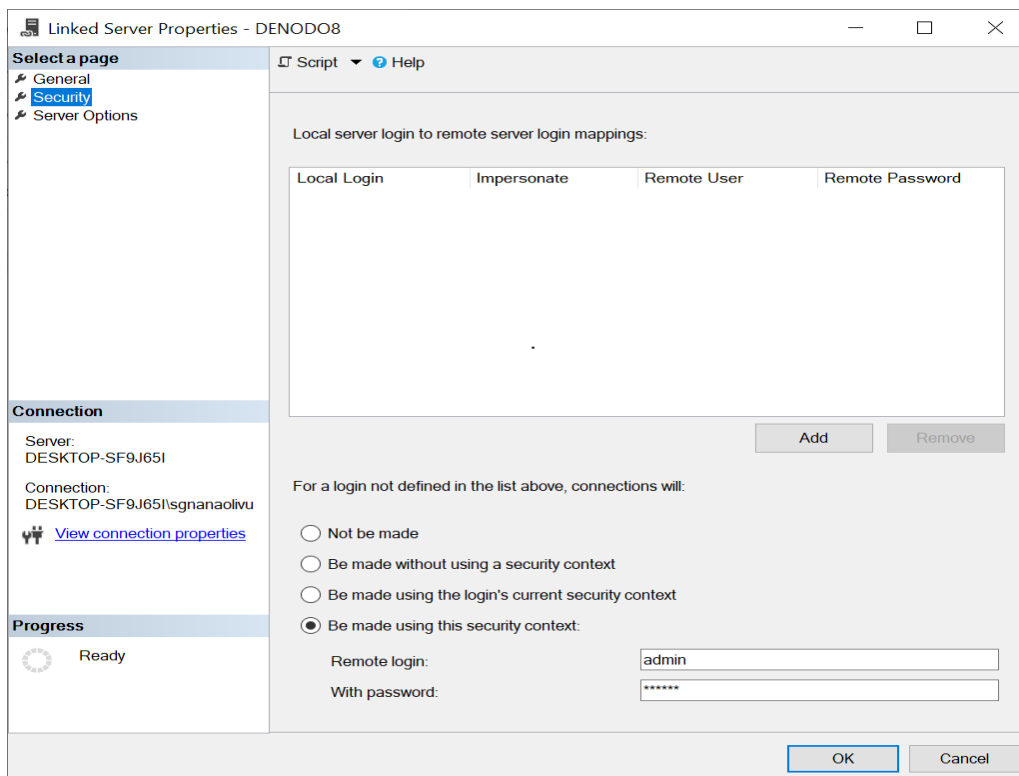
2. Start the SQL Server Management Studio and under **Object Browser** expand **Server Objects** and right click **Linked Servers** as shown below:



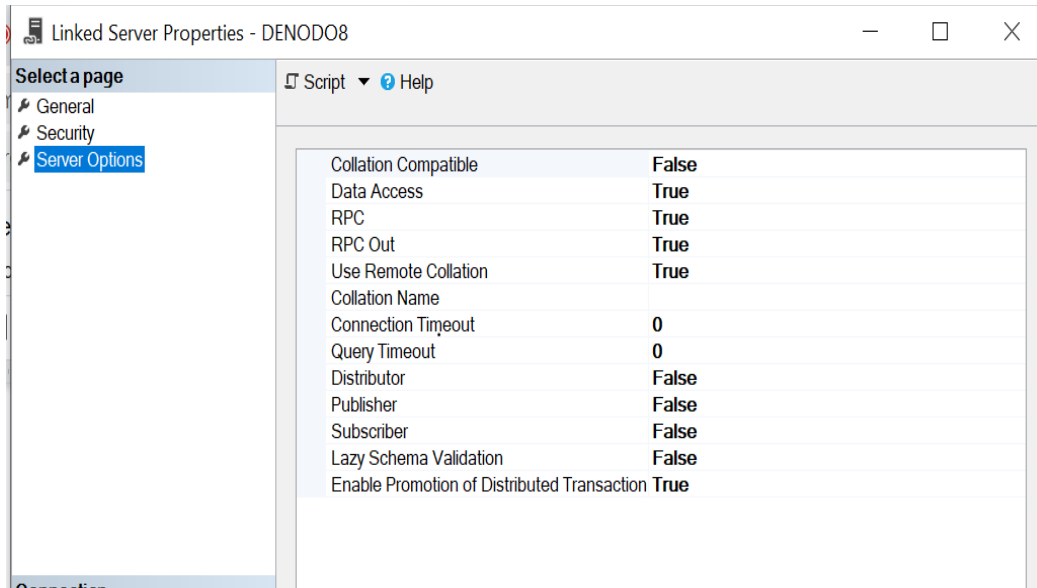
3. Click on **New Linked Server**. On the new window fill in the information as shown to create a linked server under the **General** tab.



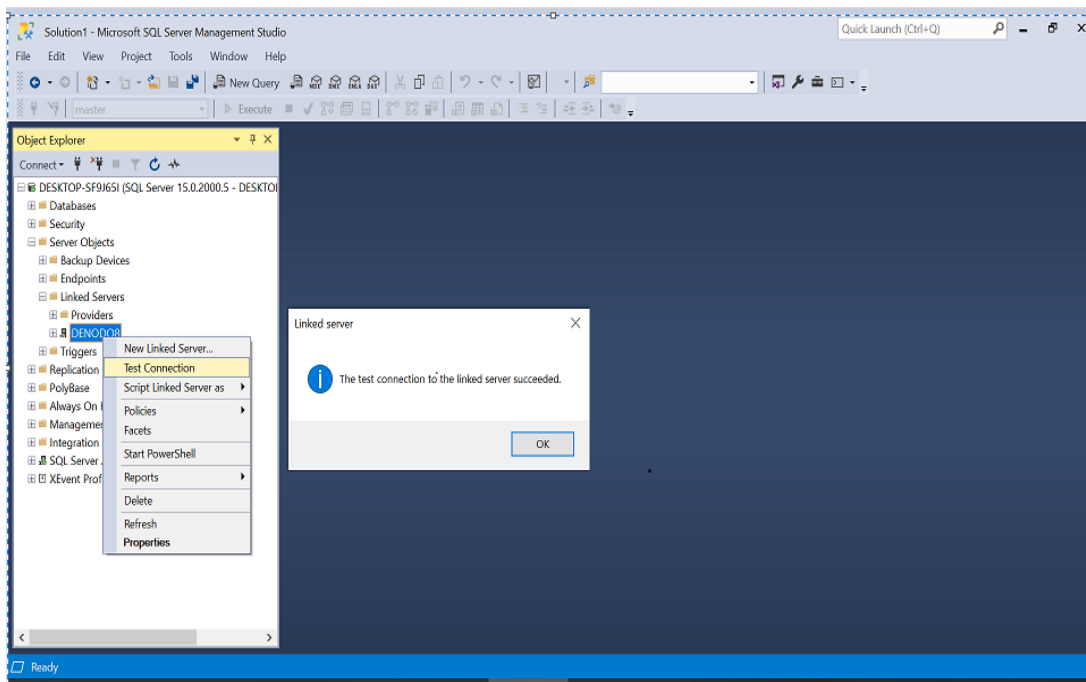
4. On the **Security** tab, select the 'Be made using this security context' option and enter the credentials to the Denodo Server.



- Finally, under the **Server Options** tab, change **RPC** and **RPC Out** options to **True** (by default they are set to **False**).



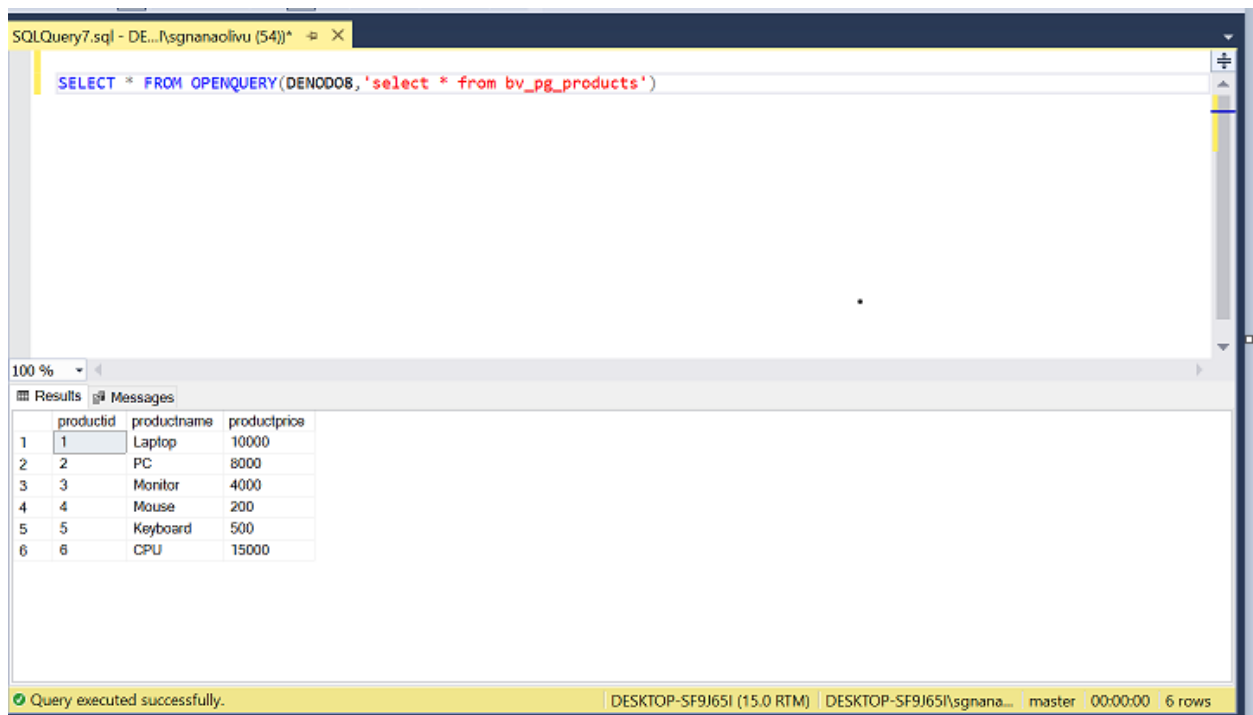
- Click **Ok**, after making all the mentioned changes. This will create the linked server and it will be listed under **SSMS Object Browser**. Right Click on the new linked server and click on **Test Connection**



3 ACCESSING AND QUERYING DENODO THROUGH SSMS

IMPORTANT NOTE: For performance reasons, due to some limitations on the OLEDB driver, to query Virtual DataPort views, the use of the **openquery** function is recommended. When using this function any WHERE condition or operation included in the query will be delegated to Virtual DataPort. However, if a regular SELECT statement using the full path for the view within the Linked Server is issued, any condition or operation in the query will not be sent to Virtual DataPort and the query will return all the records coming from the Virtual DataPort view that will have to be post-filtered.

Once the Linked server is configured, open a query tab from SQL Server Management Studio and run a select query like the following, where 'DENODO8' is the given name for the Linked Server:



```
SQLQuery7.sql - DE... \sgnanaoliv (54)* - X
SELECT * FROM OPENQUERY(DENODO8, 'select * from bv_pg_products')
```

	productid	productname	productprice
1	1	Laptop	10000
2	2	PC	8000
3	3	Monitor	4000
4	4	Mouse	200
5	5	Keyboard	500
6	6	CPU	15000

Query executed successfully. DESKTOP-SF9J65I (15.0 RTM) DESKTOP-SF9J65I\sgnana... master 00:00:00 6 rows

Note that to run the query we have to use the **openquery** function that takes two input parameters: first, the name of the linked server, second, the query to run.

When we use this type of queries, and **Lazy Schema Validation** is not set to True, SQL Server will issue two queries against Denodo, the first one will be a metadata discovery query (only the first time we issue the query).

To create a view in a SQL Server schema that points directly to a linked table, this sentence can be executed in SQL Server:

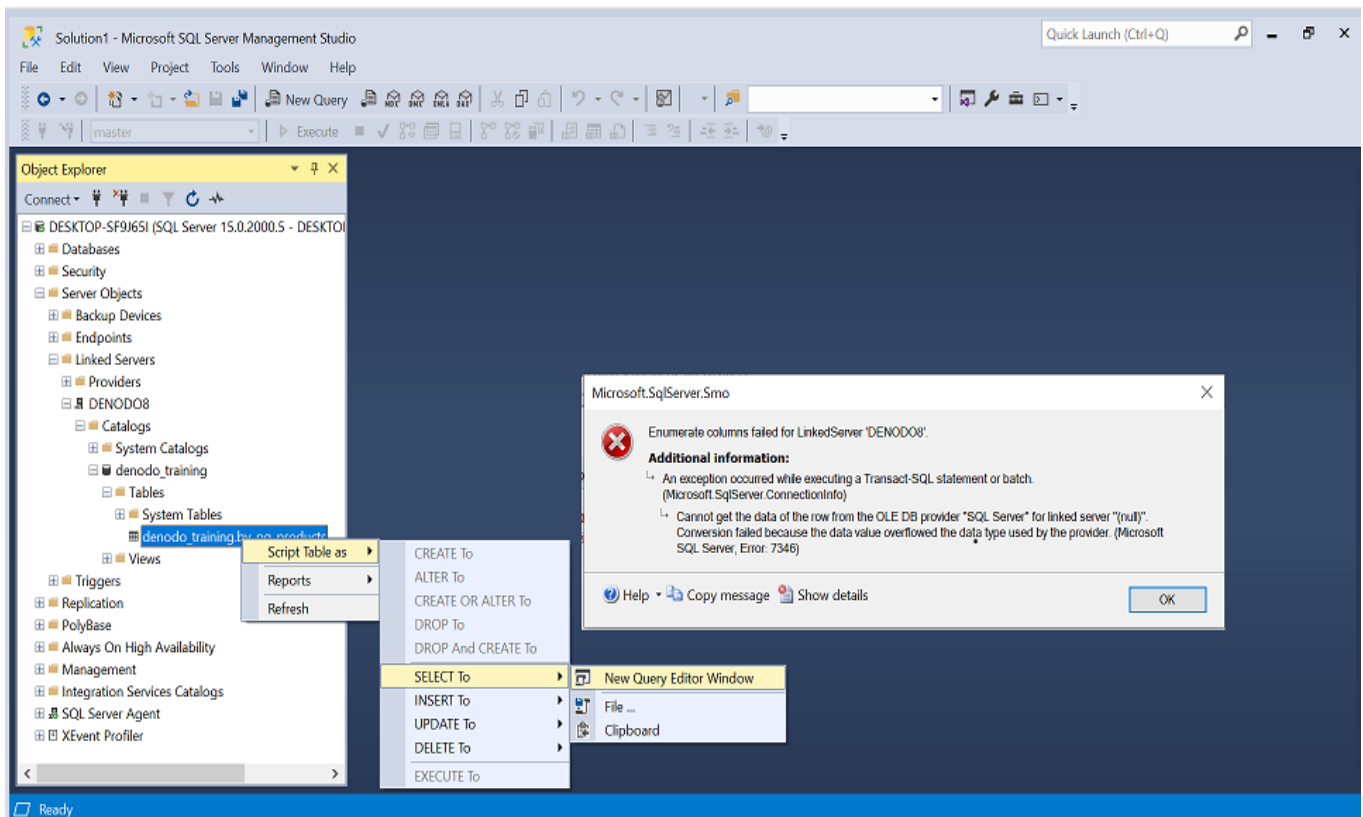
```
CREATE VIEW [DBO].[CLIENT]
AS
SELECT * FROM OPENQUERY(DENODO8, 'SELECT * FROM bv_pg_products')
```

In this case, a 'client' view will be created on the local SQL Server schema.

Even without creating a local view it is possible to access the linked table through the Linked Server full path, for instance:

```
SELECT * FROM [DENOD08].denodo_training.denodo_training.bv_pg_products
```

However, the option "Script Table as...SELECT To" does not work



4 HOW TO ACCESS SQL SERVER LINKED SERVER FROM DENODO?

To access Linked Server an ODBC data source from Virtual DataPort must be created using the SQL Server configuration:

1. Create a DSN

Create a DSN on the system that points to the database to access. Choose SQL Native Client driver and do not select any database to be able to access all the configured Linked Servers.

2. Create a new ODBC data source

Create a new ODBC data source, choose the DSN previously created and give a name to the data source.

3. Create a base view

Introspection will not be available for the configured Linked Servers. Introspection will work for the elements in the SQL Server schemas.

An option to use the introspection is to create views on the local schemas on top of the linked tables as we saw in the previous section. Other options are:

Create the base views to Linked Servers using VQL, for instance:

```
CREATE OR REPLACE WRAPPER ODBC bv_odbc_readdata_linkedserver
  FOLDER = '/02-baseview'
  DATASOURCENAME=linkedserver
  SQLSENTENCE='SELECT * FROM [DENOD08].[denodo_training].[denodo_training].bv_pg_products'
  OUTPUTSCHEMA (
    productid = 'productid' : 'java.lang.Integer' (sourcetypedecimals='0', sourcetypesize='10',
sourcetypeid='4', sourcetypename='int') SORTABLE NOT UPDATEABLE,
    productname = 'productname' : 'java.lang.String' (sourcetypedecimals='0',
sourcetypesize='10', sourcetypeid='-9', sourcetypename='nvarchar') SORTABLE NOT UPDATEABLE,
    productprice = 'productprice' : 'java.lang.Integer' (sourcetypedecimals='0',
sourcetypesize='10', sourcetypeid='4', sourcetypename='int') SORTABLE NOT UPDATEABLE
  );

CREATE OR REPLACE TABLE bv_odbc_readdata_linkedserver I18N us_pst (
  productid:int (sourcetypeid = '4', sourcetypedecimals = '0', sourcetypesize = '10'),
  productname:text (sourcetypeid = '-9', sourcetypedecimals = '0', sourcetypesize =
'10'),
  productprice:int (sourcetypeid = '4', sourcetypedecimals = '0', sourcetypesize = '10')
)
  FOLDER = '/02-baseview'
  CACHE OFF
  TIMETOLIVEINCACHE DEFAULT
  ADD SEARCHMETHOD bv_odbc_readdata_linkedserver(
    I18N us_pst
    CONSTRAINTS (
      ADD productid NOS ZERO ()
      ADD productname NOS ZERO ()
      ADD productprice NOS ZERO ()
    )
  )
  OUTPUTLIST (productid, productname, productprice
)
```

```
WRAPPER (odbc bv_odbc_readdata_linkedserver)
);
```

Create the base view using the “Create Base View from query” option using as SQL query:

```
SELECT * FROM [DENOD08].[denodo_training].
[denodo_training].bv_pg_products;
```

where DENOD08 is the name of the Linked Server.