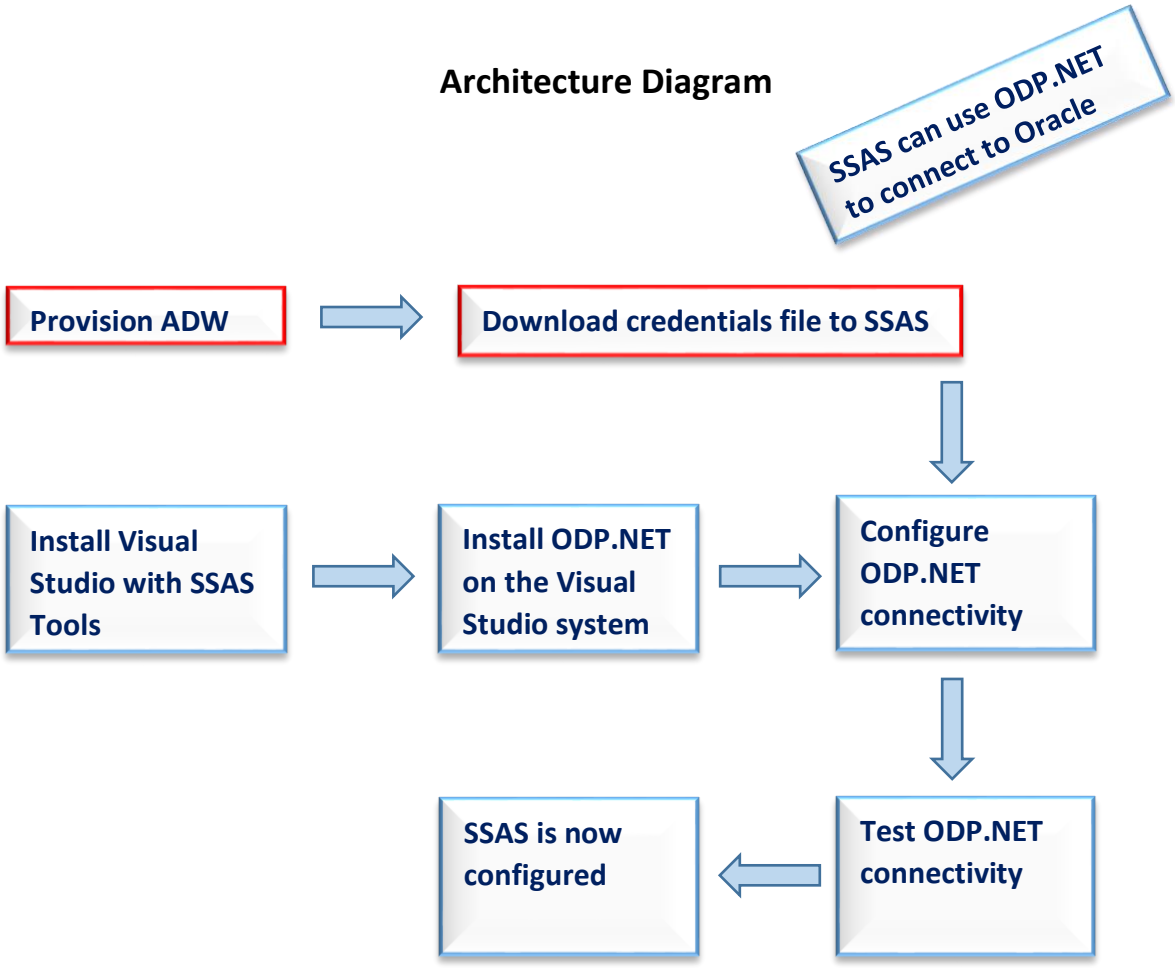
Connecting Microsoft SQL Server Analysis Services to Oracle Autonomous Database

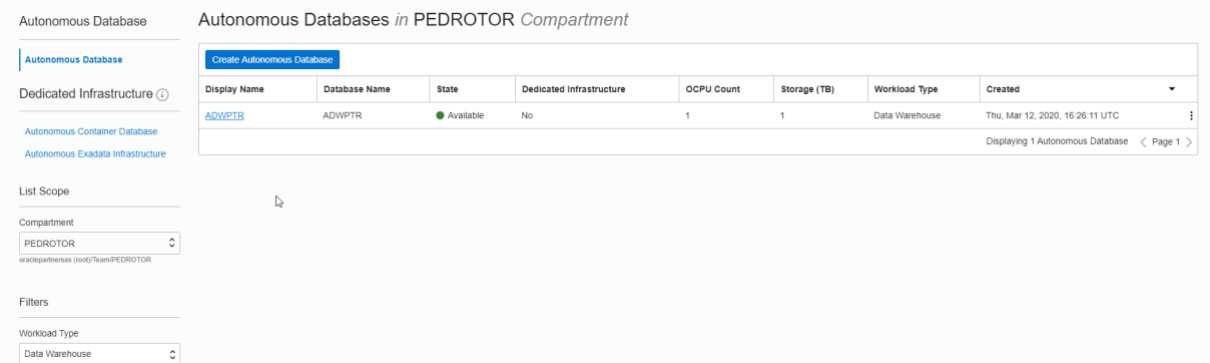
Pedro Torres, Vijay Balebail, Alex Keh

This step by step tutorial guides how to configure Microsoft SQL Server Analysis Services (SSAS) connectivity to Oracle Autonomous Database (ADB). These instructions use Oracle Data Provider for .NET (ODP.NET) for data access and work for both dedicated and shared infrastructure ADB.

Architecture Diagram

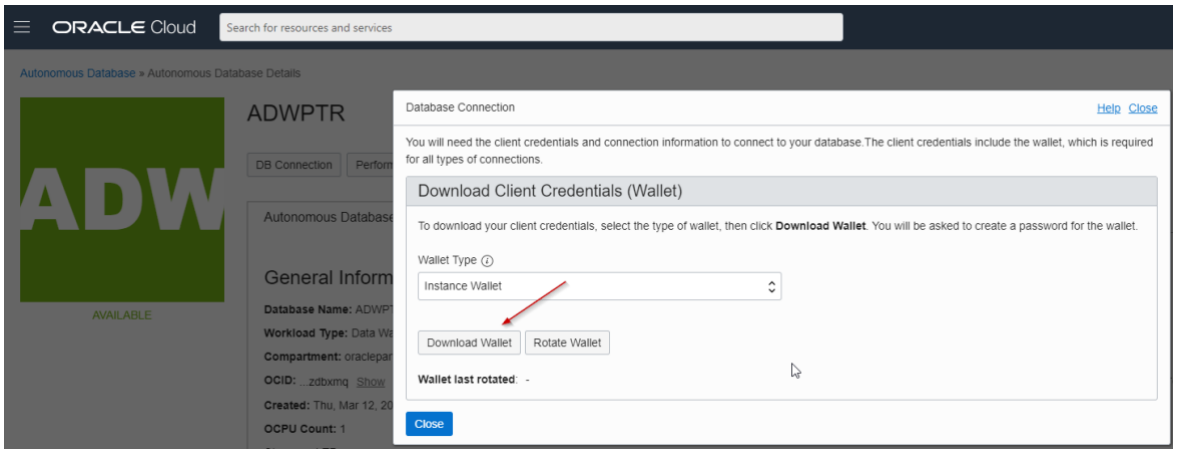


This document assumes that the Autonomous Data Warehouse (ADW) or Autonomous Transaction Processing (ATP) was provisioned.



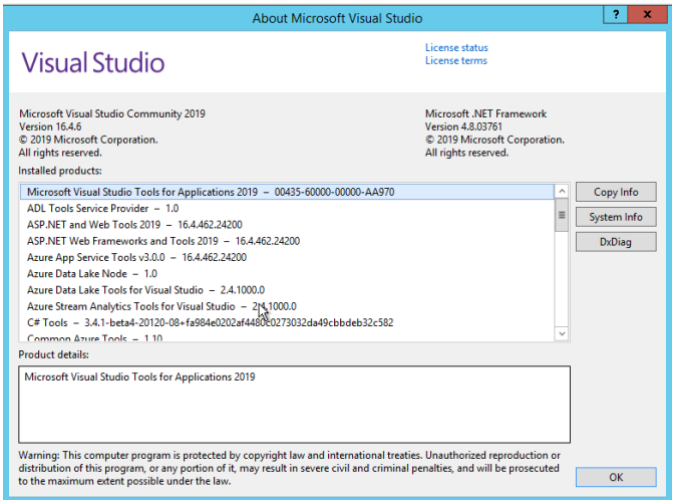
*Note: Please check here for the Oracle documentation to provision ADW.

1. Download the corresponding credentials.zip file to the system that has SSAS installed.



*Note: Also check Downloading Client Credentials (Wallets).

2. Check that SQL Services Analysis Services tools are installed on Visual Studio 2017 or Visual Studio 2019. This would be found in the Installed Products window of the About Microsoft Visual Studio dialog.

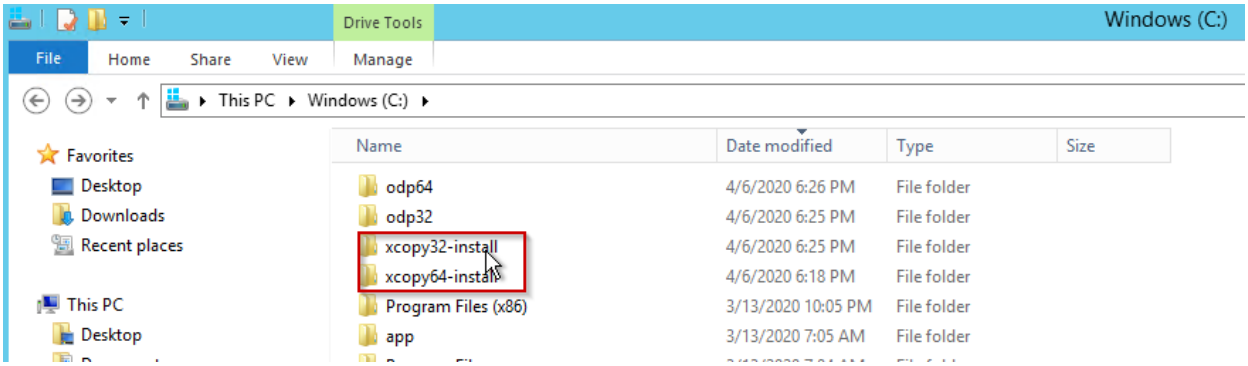


3. Download 32-bit *and* 64-bit ODAC 19.3 xcopy from Oracle.com from the ODAC Xcopy section in the middle of the page.

https://www.oracle.com/database/technologies/dotnet-odacdeploy-downloads.html

*Note: This procedure guides how to connect Visual Studio SSAS tools and retrieve data. You will need to download and enable both 32-bit and 64-bit unmanaged ODP.NET. It assumes you will be using ODP.NET 19.

4. Unzip the contents to a staging directory (e.g. c:\xcopy32-install and c:\xcopy64-install).



5. Open a Windows command prompt ***in administrator mode*.**
Navigate to the 32-bit staging directory and execute the next commands:

install.bat odp.net4 <installation directory> odp32

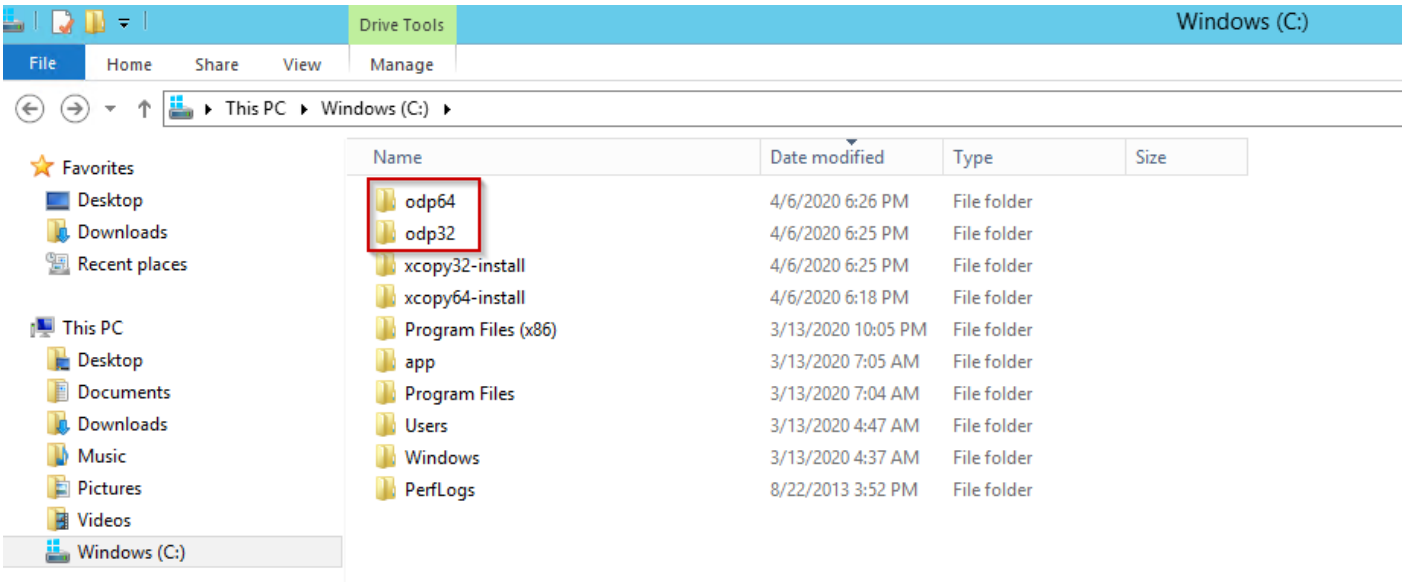
```
C:\xcopy32-install\ODAC193Xcopy_32bit>install.bat odp.net4 c:\odp32 odp32
```

Navigate to the 64-bit staging directory and run:

install.bat odp.net4 <installation directory> odp64

```
C:\xcopy64-install\ODAC193Xcopy_x64>install.bat odp.net4 c:\odp64 odp64
```

*Note: Enter the install location (e.g. c:\odp32 and c:\odp64) for the directory parameter.
These steps will install 32-bit unmanaged ODP.NET and 64-bit unmanaged ODP.NET.

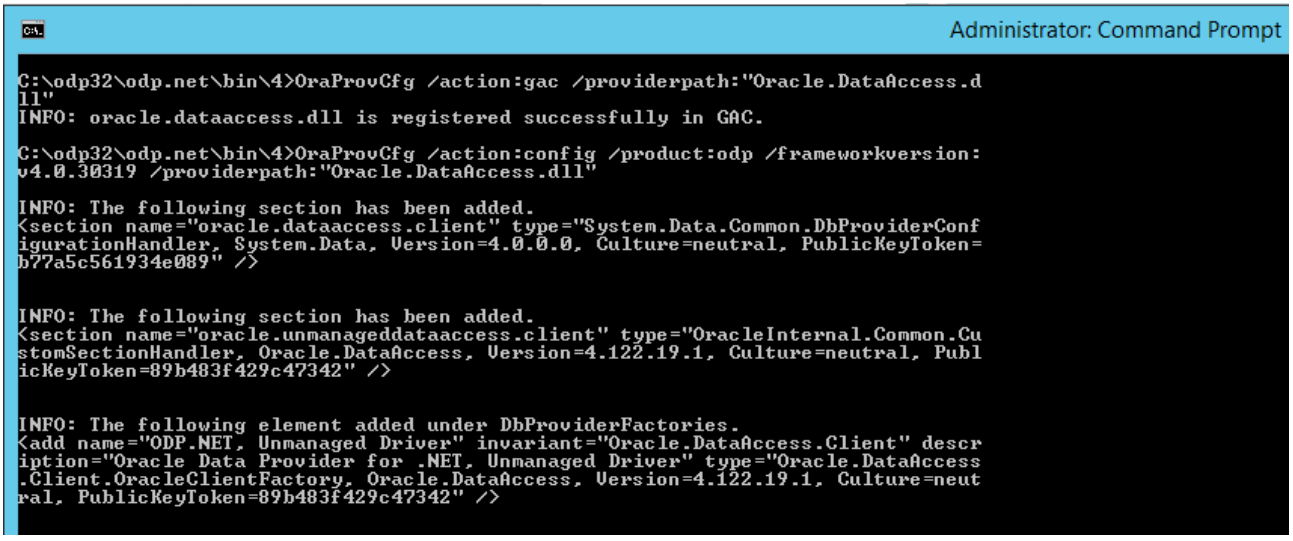


6. In the same command prompt **with administrator privileges,** navigate to the 32-bit install subdirectory <installation directory>\odp.net\bin\4

   Execute the following two commands:

   OraProvCfg /action:gac /providerpath:"Oracle.DataAccess.dll"

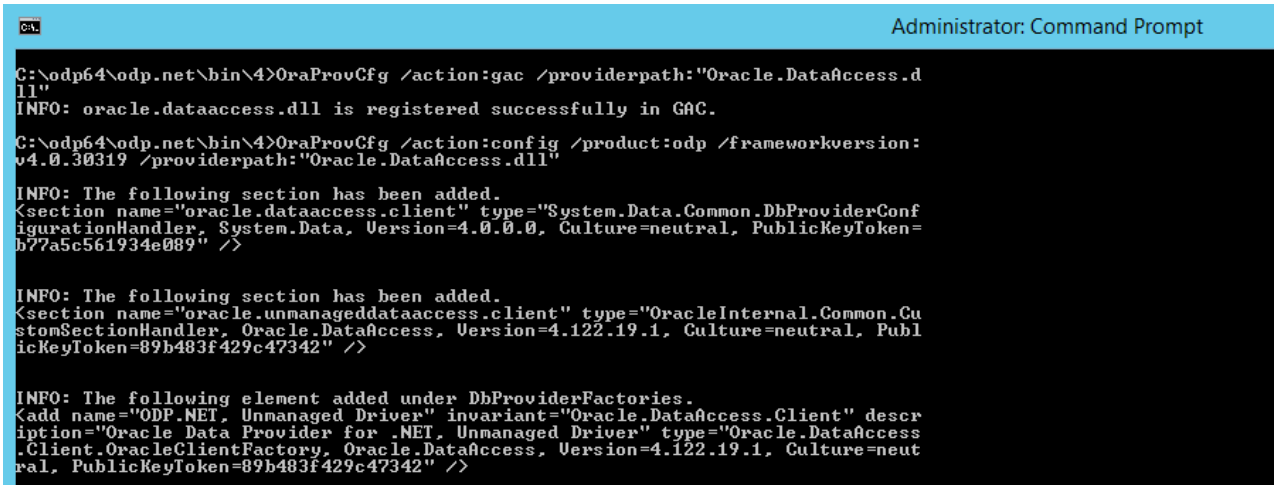
   OraProvCfg /action:config /product:odp /frameworkversion:v4.0.30319 /providerpath:"Oracle.DataAccess.dll"



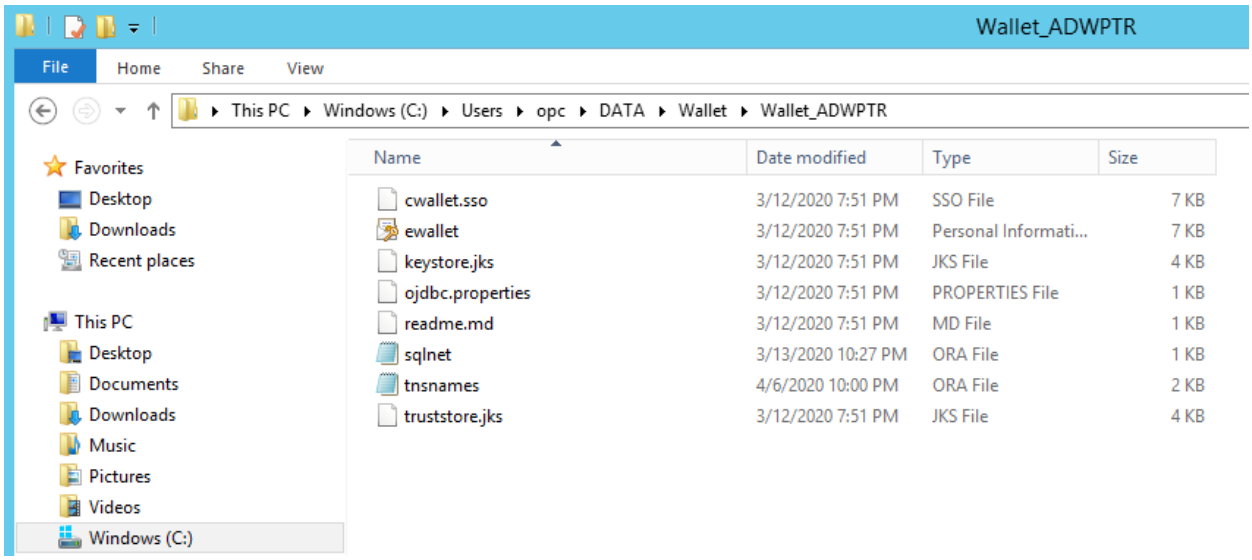
Navigate to the 64-bit install subdirectory <installation directory>\odp.net\bin\4 and execute these two commands too:

OraProvCfg /action:gac /providerpath:"Oracle.DataAccess.dll"

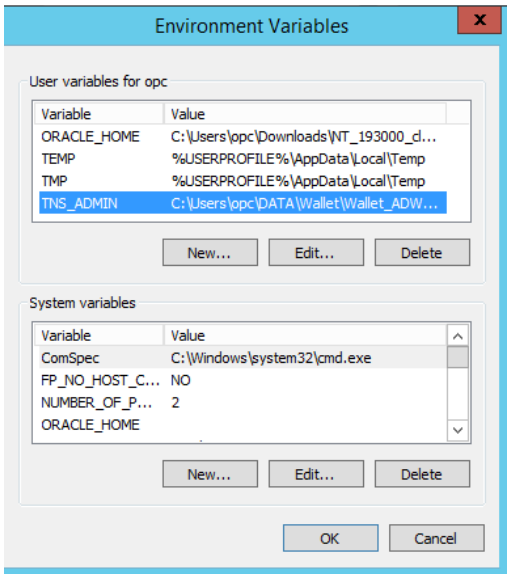
OraProvCfg /action:config /product:odp /frameworkversion:v4.0.30319 /providerpath:"Oracle.DataAccess.dll"



7. Download your ADB credentials from the Oracle Cloud and unzip the contents into a directory.



8. In the Windows environment variables dialog, create the TNS_ADMIN variable and set it to the directory location of where you unzipped the ADB credentials.



*Note: The tnsnames.ora net service names will be used to connect to ADB.

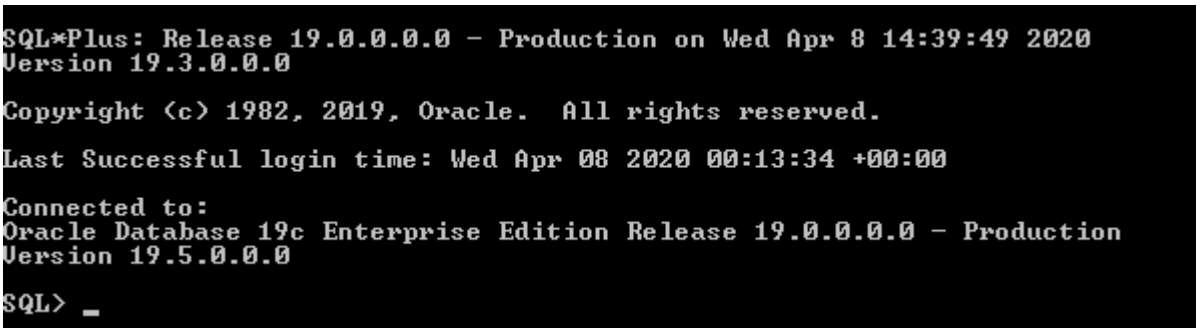
10. Edit the Windows Path environment variable by adding two new entries.

    1) 32-bit Oracle Client install directory (i.e. c:\odp32)
    2) 64-bit Oracle Client install directory (i.e. c:\odp64)

    *Note: To ensure these two entries have precedence over existing Oracle Homes, move them both to the top of the Path environment variable priority order with the 32-bit client path first.

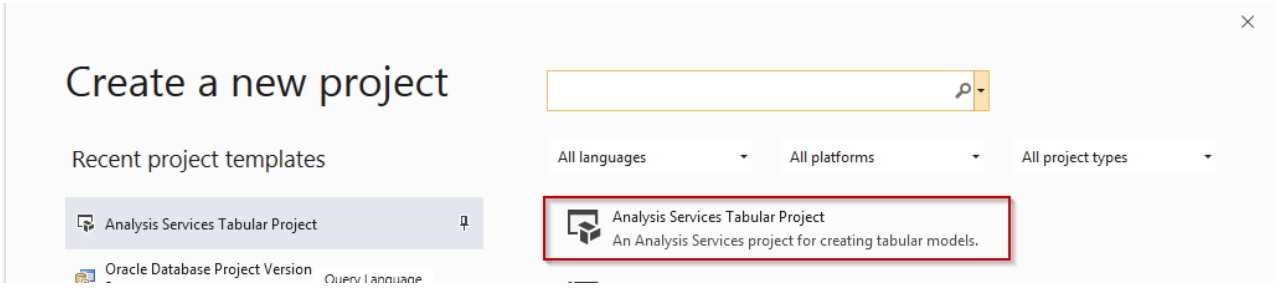
11. Modify the sqlnet.ora file to change the directory location where the wallet (cwallet.sso) has been unzipped. It is recommended you remove the quotes around the directory location as well.

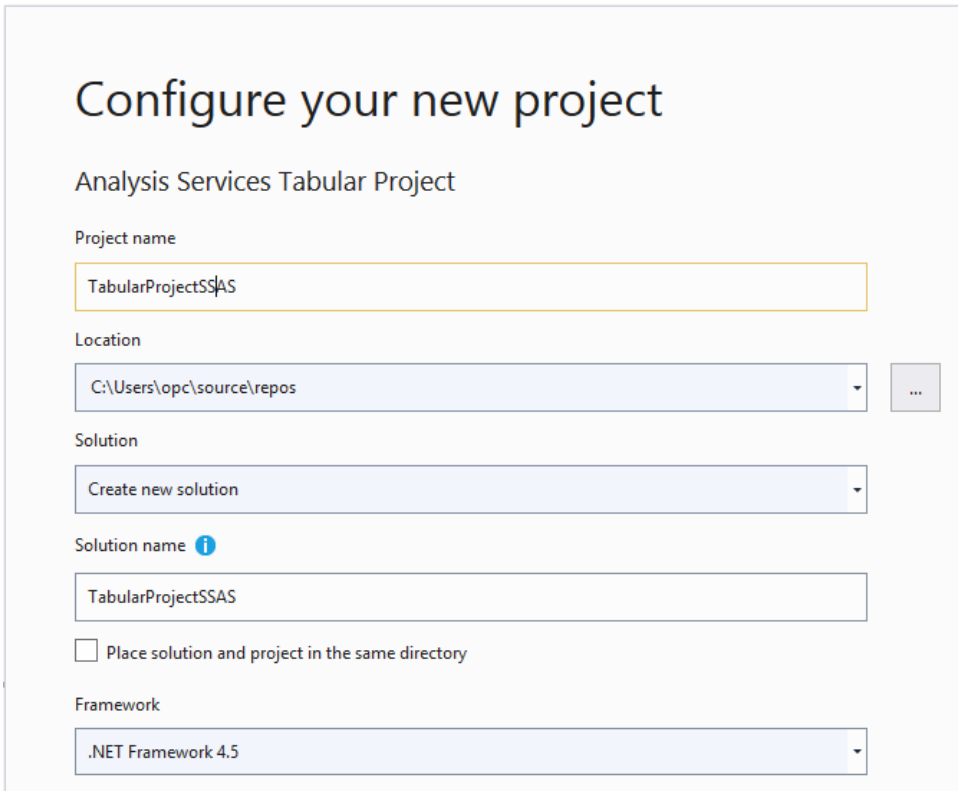
12. (Optional) Test connection from the Client using SQL* Plus

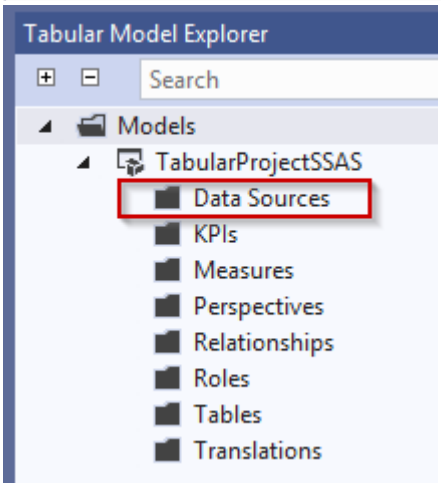


*Note: SSAS uses unmanaged ODP.NET to connect to ADB.

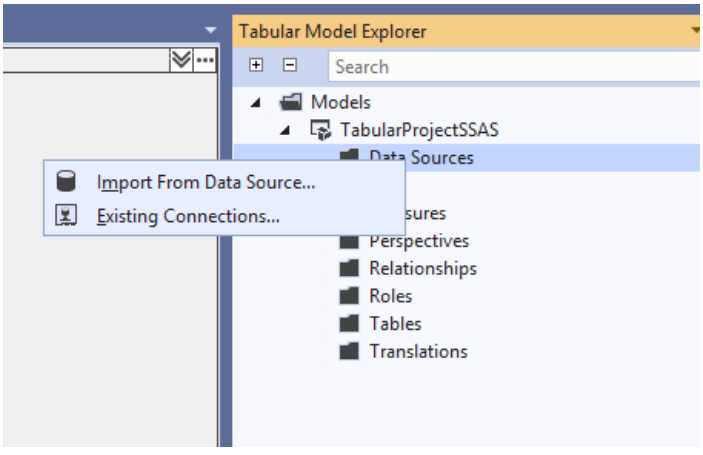
13. Open Visual Studio and create a SSAS project.

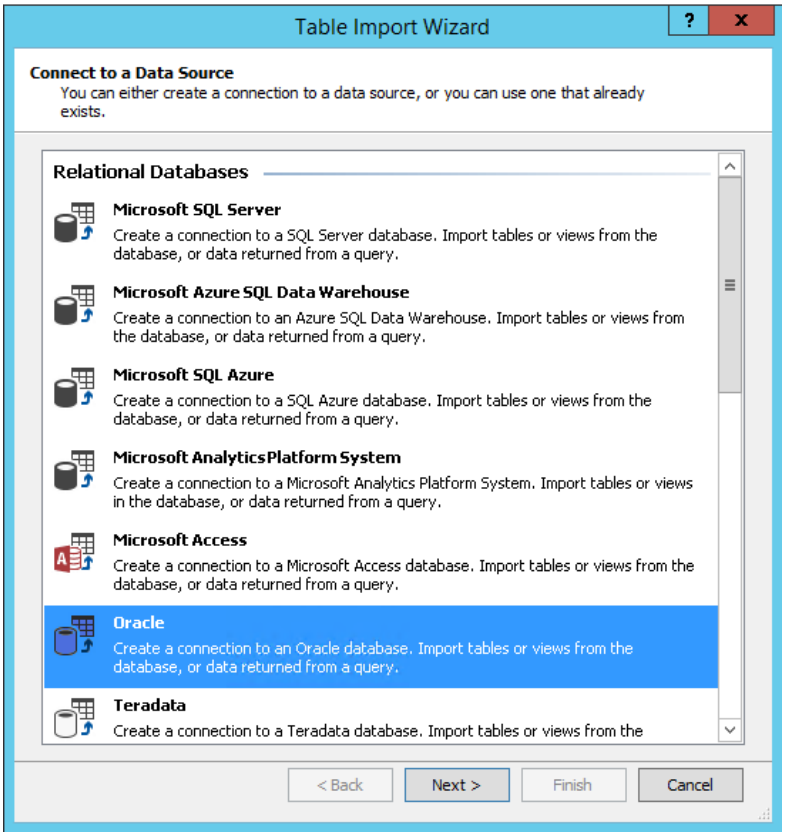




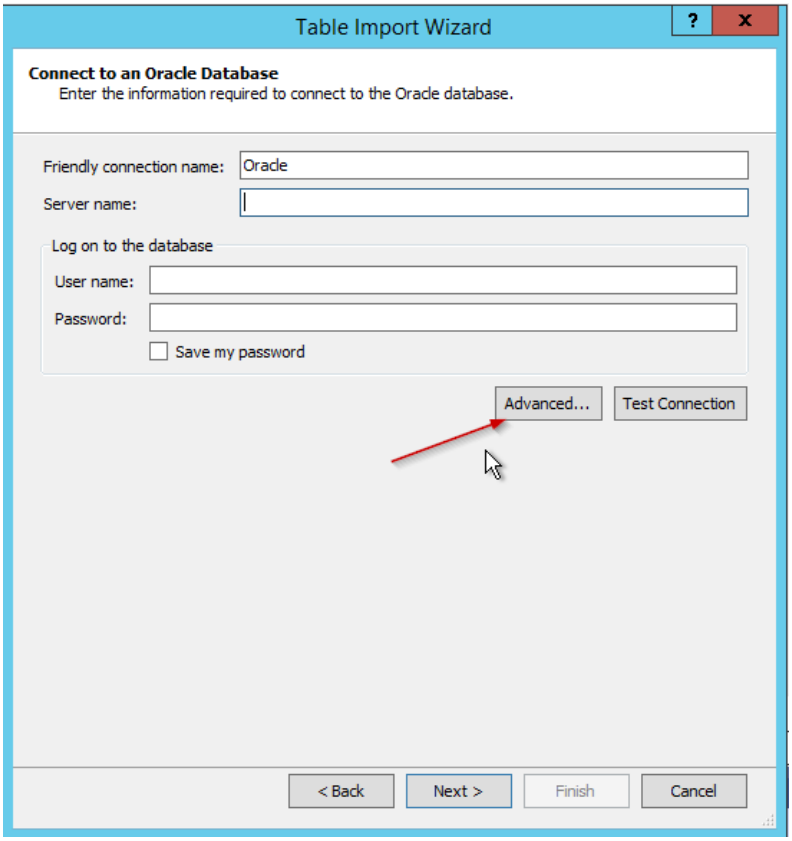


14. Right-click "Data Sources", select “Import from Data Source…”, and then select "Oracle" in the Table Import Wizard. Click on “Next”.

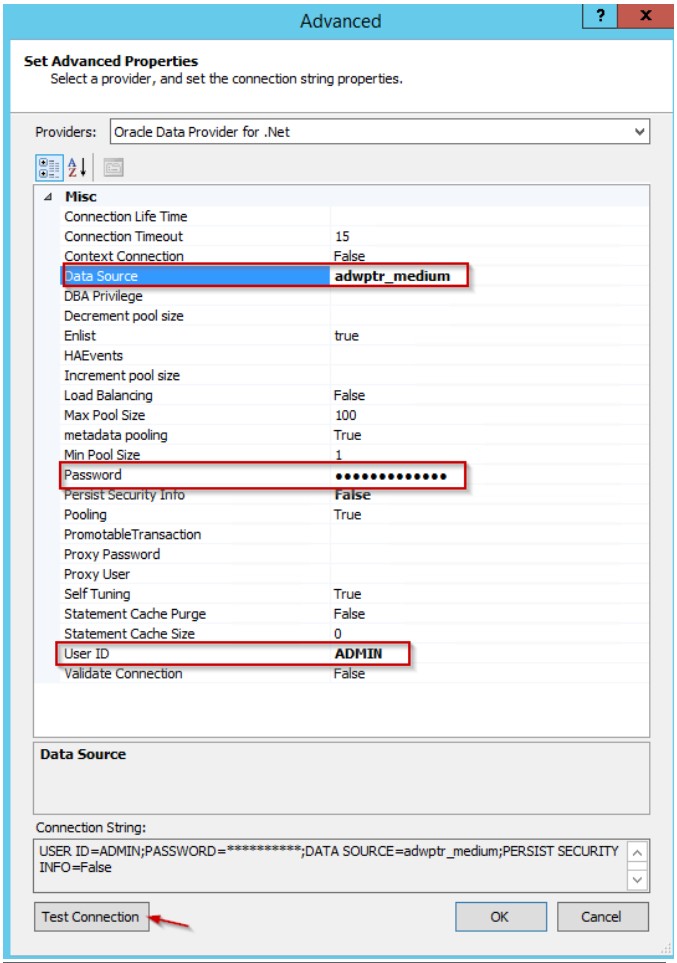


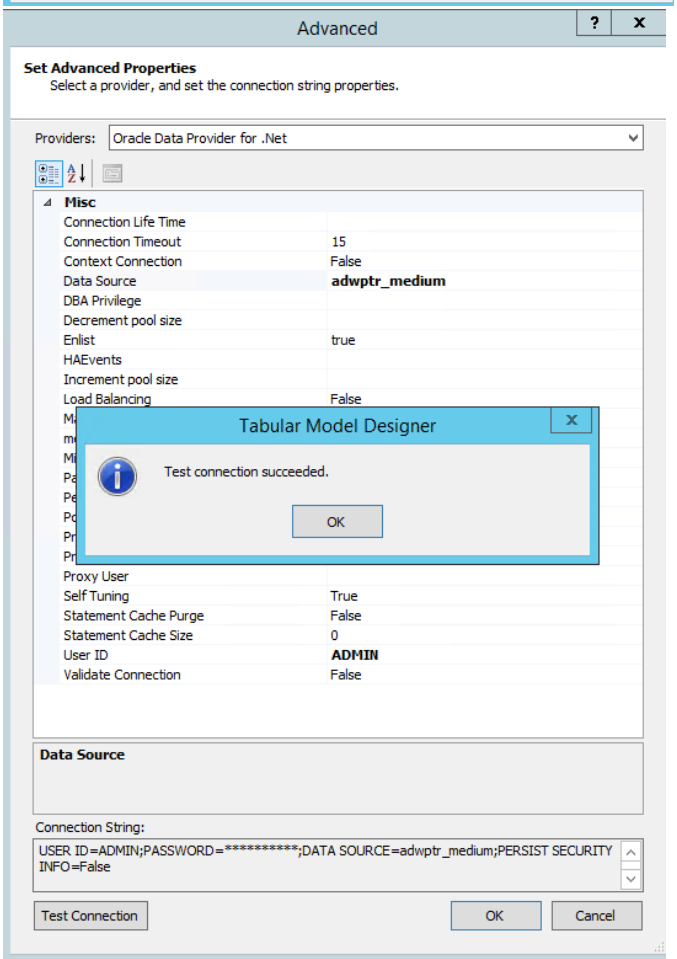


15. Click the "Advanced" button.

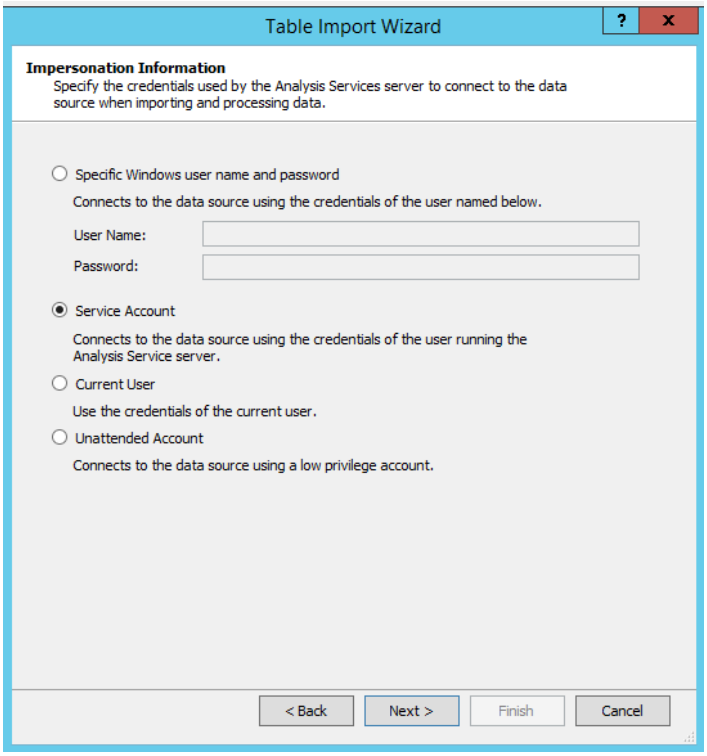


16. In the drop down, select "Oracle Data Provider for .NET" in the choice of providers. Enter the user id, password, and data source (i.e. net service name) entries. Test the connection. Click “OK”.

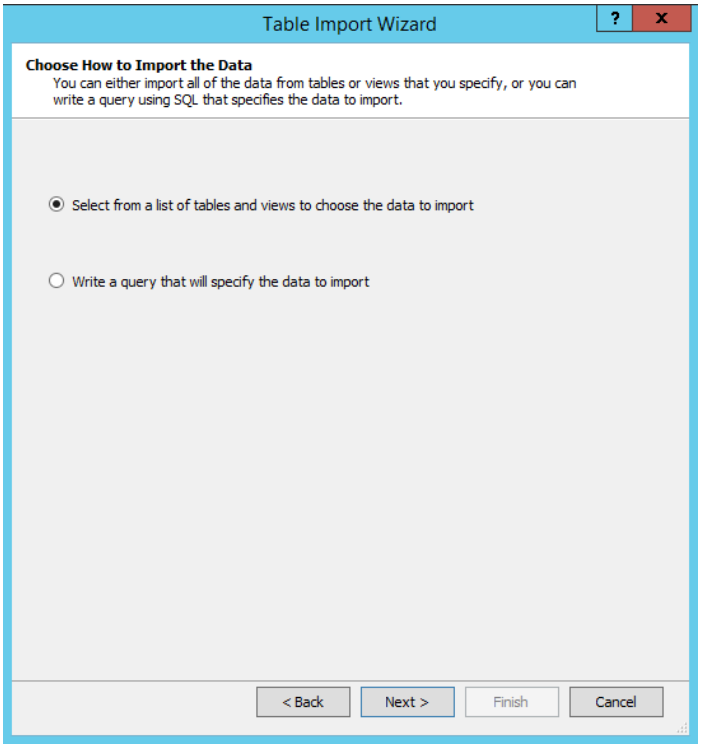




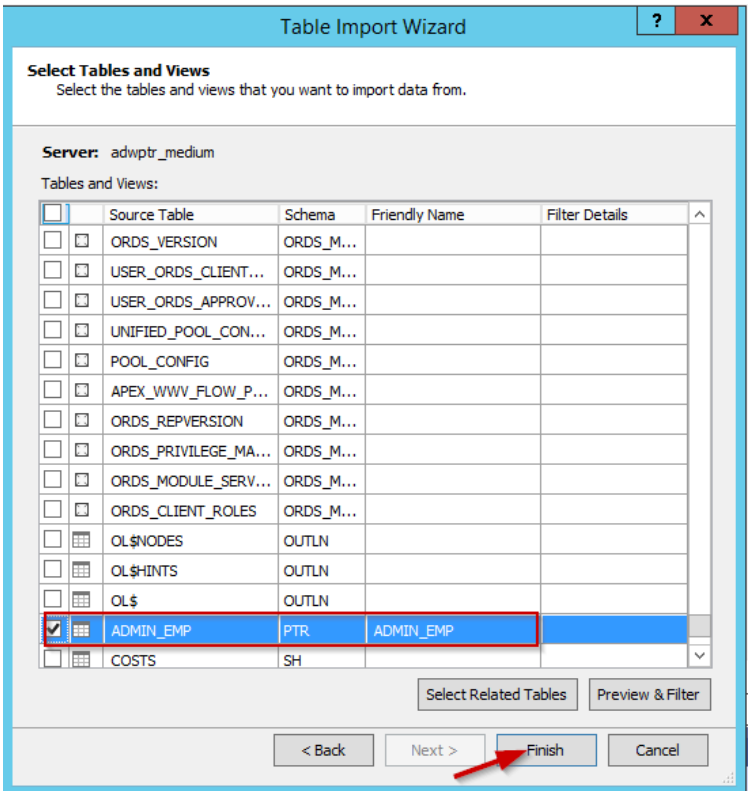
17. Choose "Service Account" for Impersonation information and then “Next”.

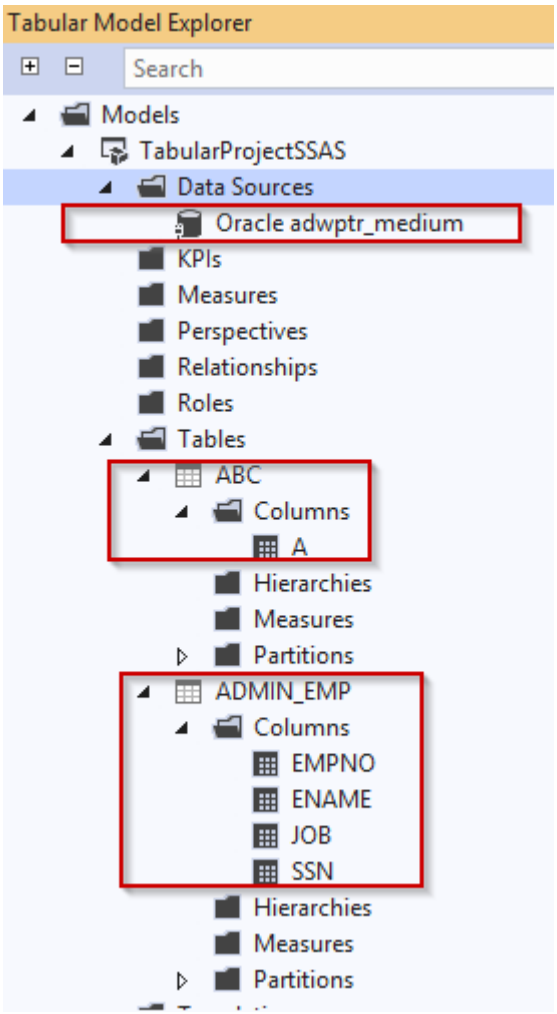


18. Choose "Select from a list …" for check the tables and then “Next”.



19. You can now connect and view ADB schema metadata. In this sample database, we select “ADMIN.ABC” table and “PTR.ADMIN_EMP” table.





20. SSAS will now use 64-bit ODP.NET to perform queries. Enter in SQL or choose tables to retrieve. Execute and your results should appear in Visual Studio

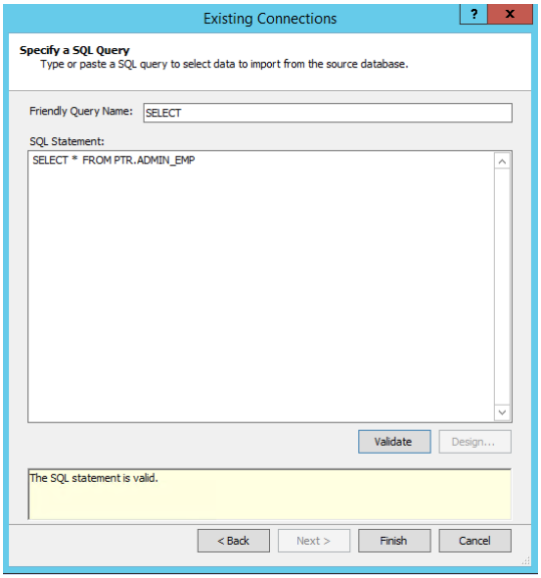
Using SSAS Tools with ADB

Let's insert records into ADB from SSAS and retrieve the records into SSAS.

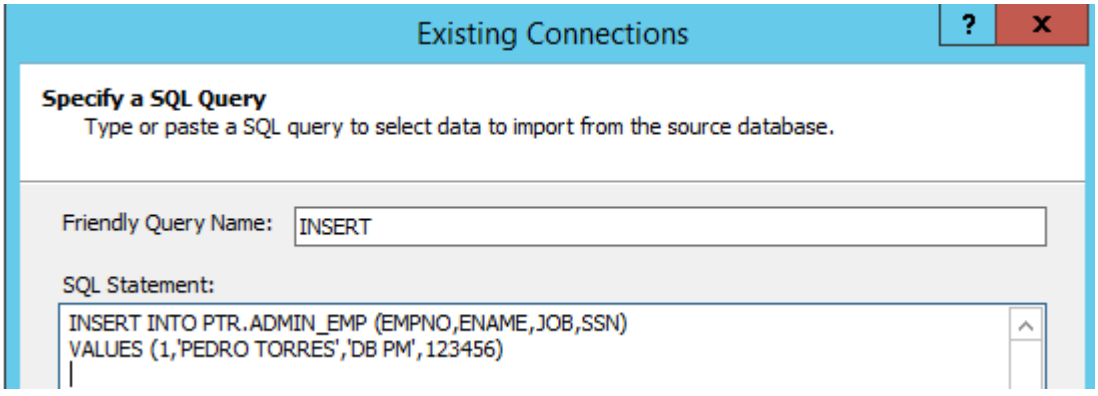
In SQL*Plus we see there are no rows in the ADMIN_EMP table currently.

```
SQL> select * from PTR.ADMIN_EMP;
no rows selected
SQL> _
```

In Visual Studio, you can retrieve data and validate the Oracle SQL's validity.



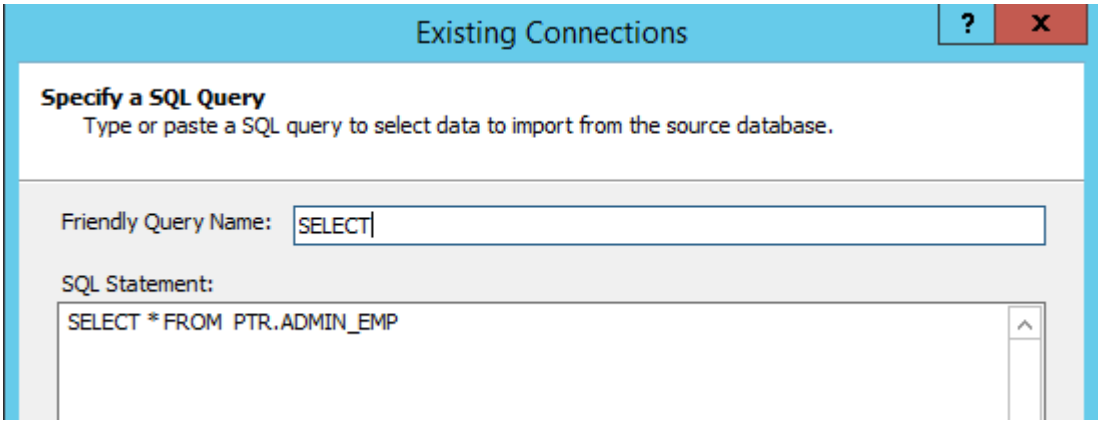
You can also insert records as needed by executing INSERT statements.

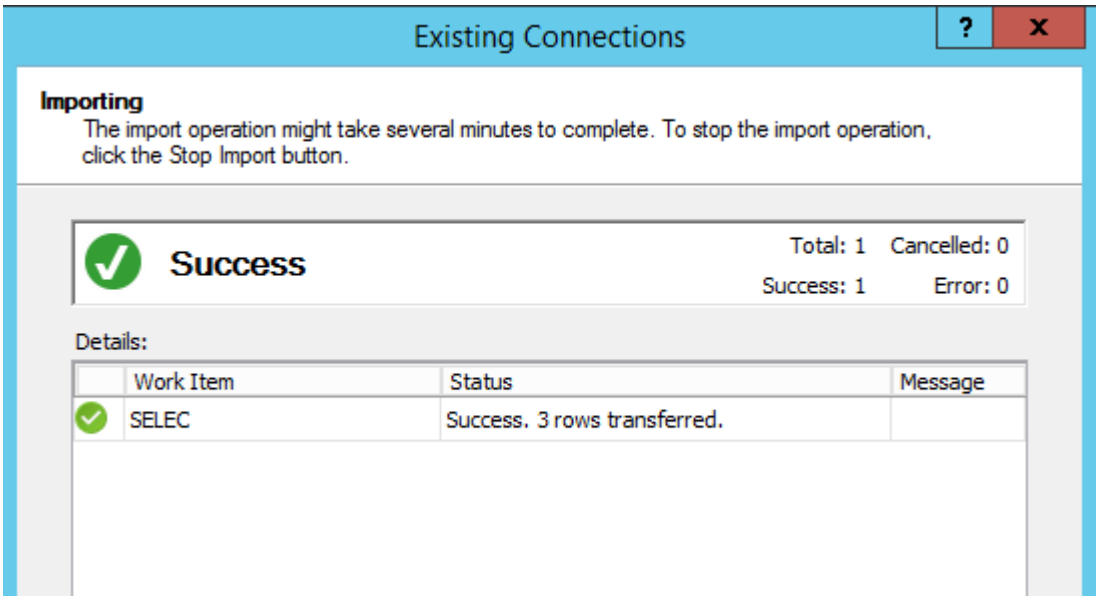


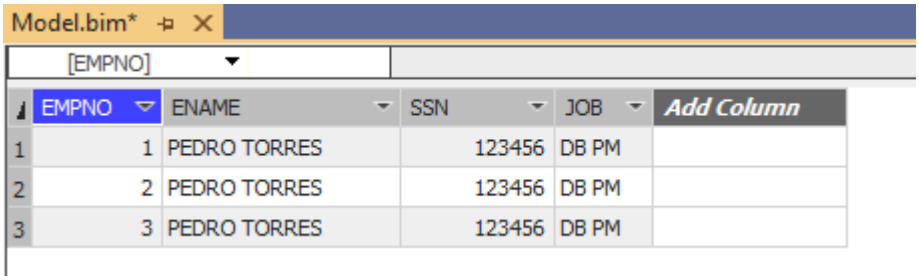
Verify the records were inserted in SQL*Plus.

```
SQL> select * from PTR.ADMIN_EMP;

     EMPNO ENAME                SSN JOB
---------- --------------- ---------- ----------
         1 PEDRO TORRES        123456 DB PM

SQL> /

     EMPNO ENAME                SSN JOB
---------- --------------- ---------- ----------
         1 PEDRO TORRES        123456 DB PM
         2 PEDRO TORRES        123456 DB PM

SQL> /

     EMPNO ENAME                SSN JOB
---------- --------------- ---------- ----------
         1 PEDRO TORRES        123456 DB PM
         2 PEDRO TORRES        123456 DB PM
         3 PEDRO TORRES        123456 DB PM

SQL>
```

Retrieve the results from Visual Studio into SSAS.







ODP.NET Performance Tuning for Large Data Retrievals

Typically, BI and ETL applications retrieve large amounts of data from a source database for further processing. To speed up Oracle data retrieval via SSIS or SSDT, the ODP.NET “FetchSize” can be increased from its default 128K value (131,072 bytes) to as large as int.MaxValue. The FetchSize determines the amount of data ODP.NET fetches into its internal cache upon each database round trip. It’s possible to improve performance by an order of magnitude by significantly increasing FetchSize when retrieving large result sets.

To increase the FetchSize, launch the Windows Registry editor (regedit.exe) and go to the following Registry key: **HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\ODP.NET\4.122.19.1**

Add the String Value "FetchSize" and set it to a value larger than the default (131072), such as 4194304 (4 MB).

Restart Visual Studio and run the SSAS query.