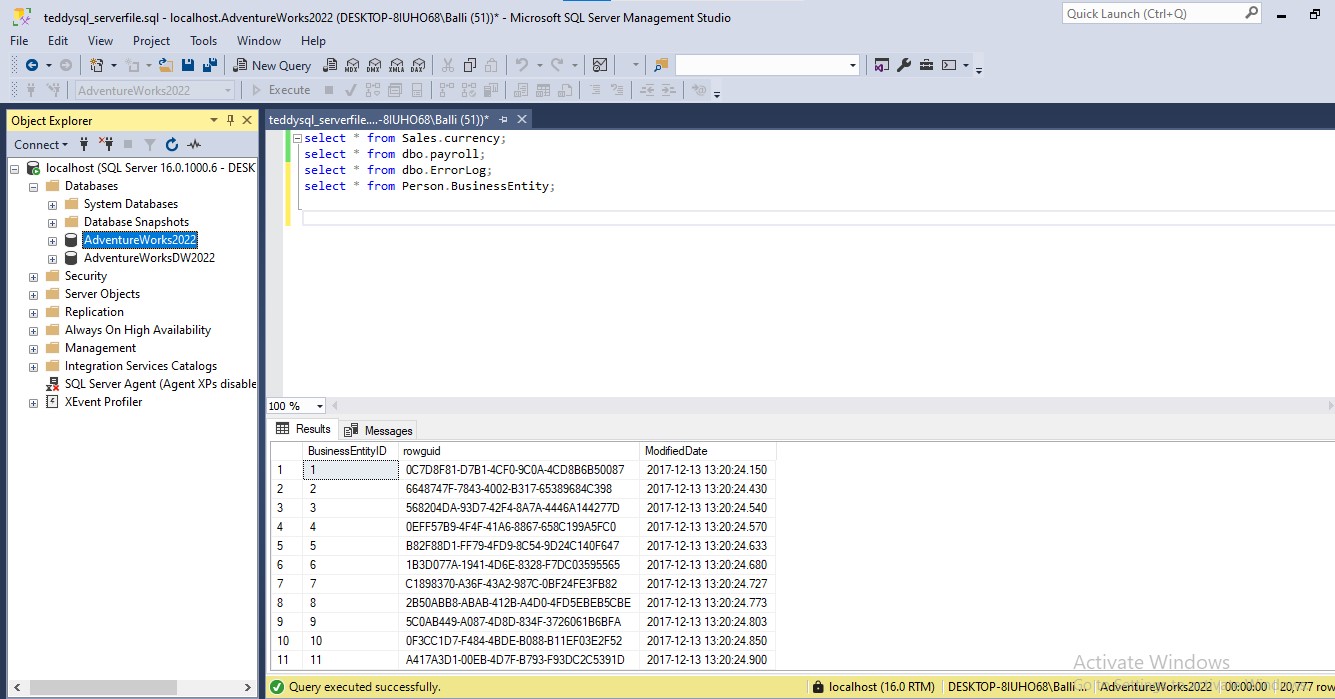
**Teddy Aigbedion**

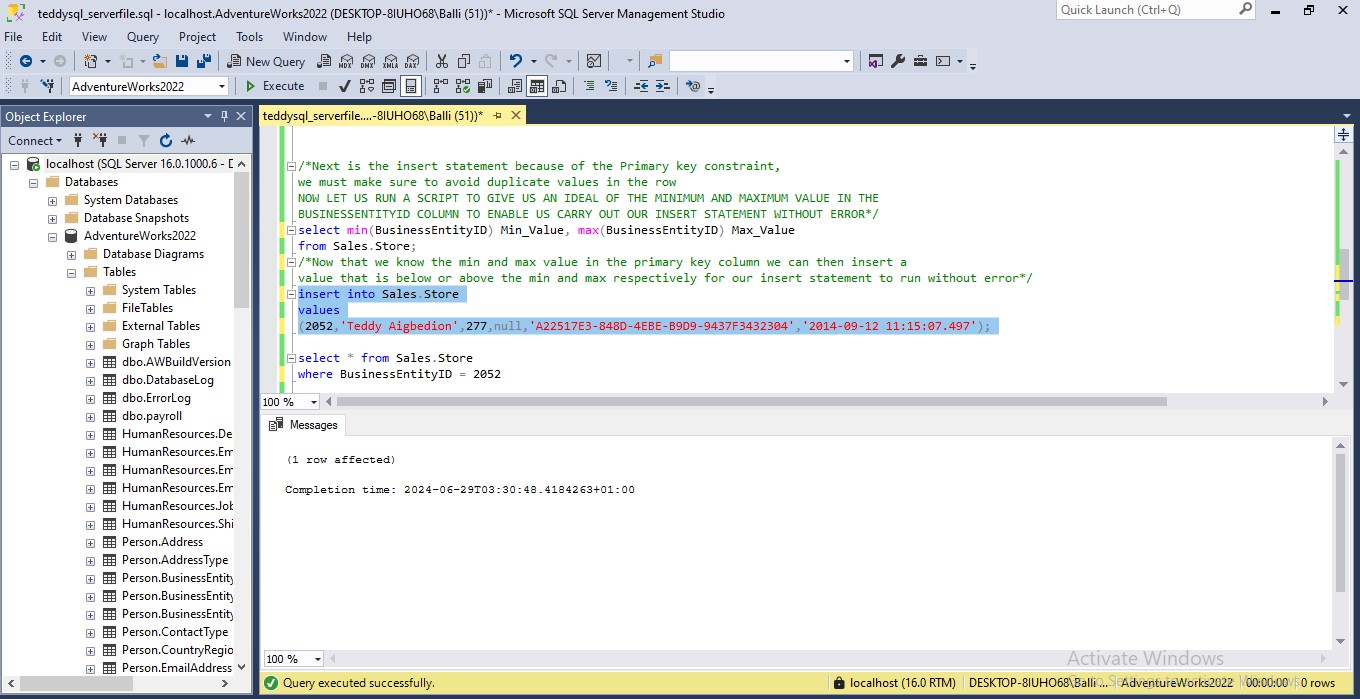
**Pivot-ED Assessment**

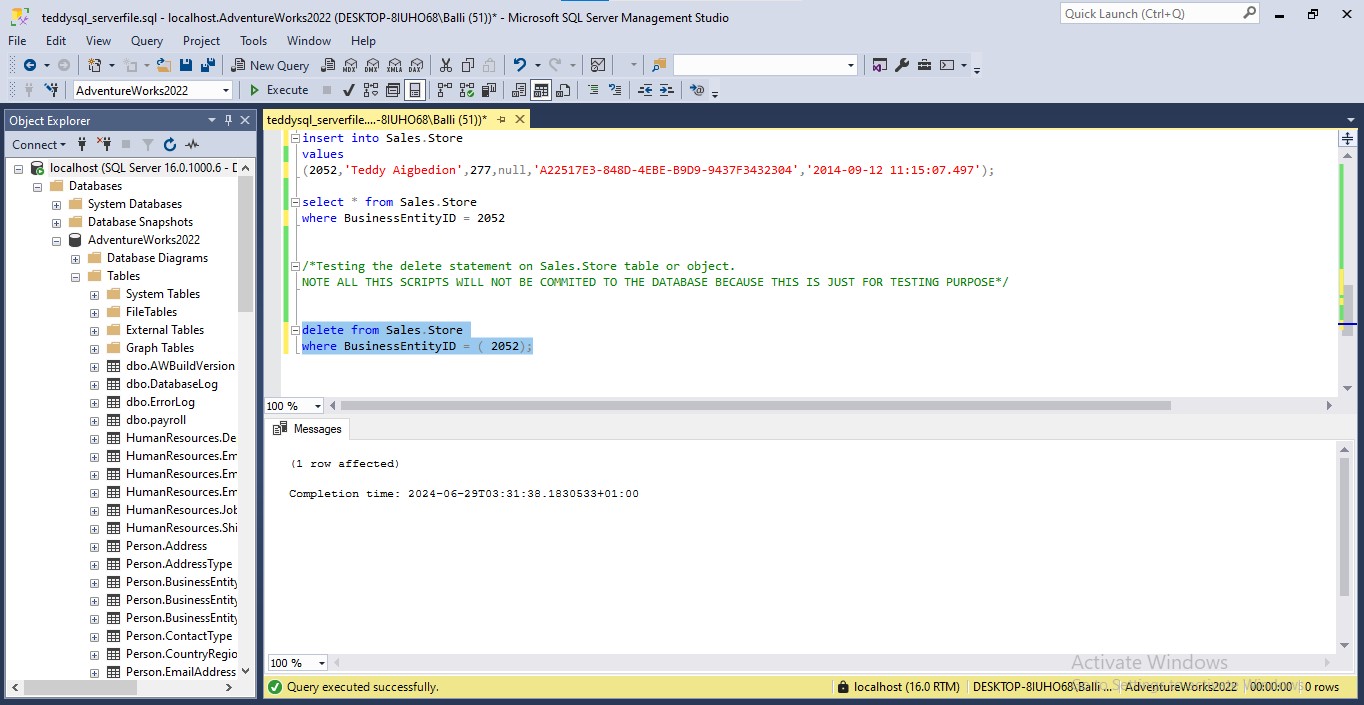
From the guide given on how to install Microsoft Sql Server in my machine, and also add two databases (AdventureWorks2022 and AdventureWorksDW2022)

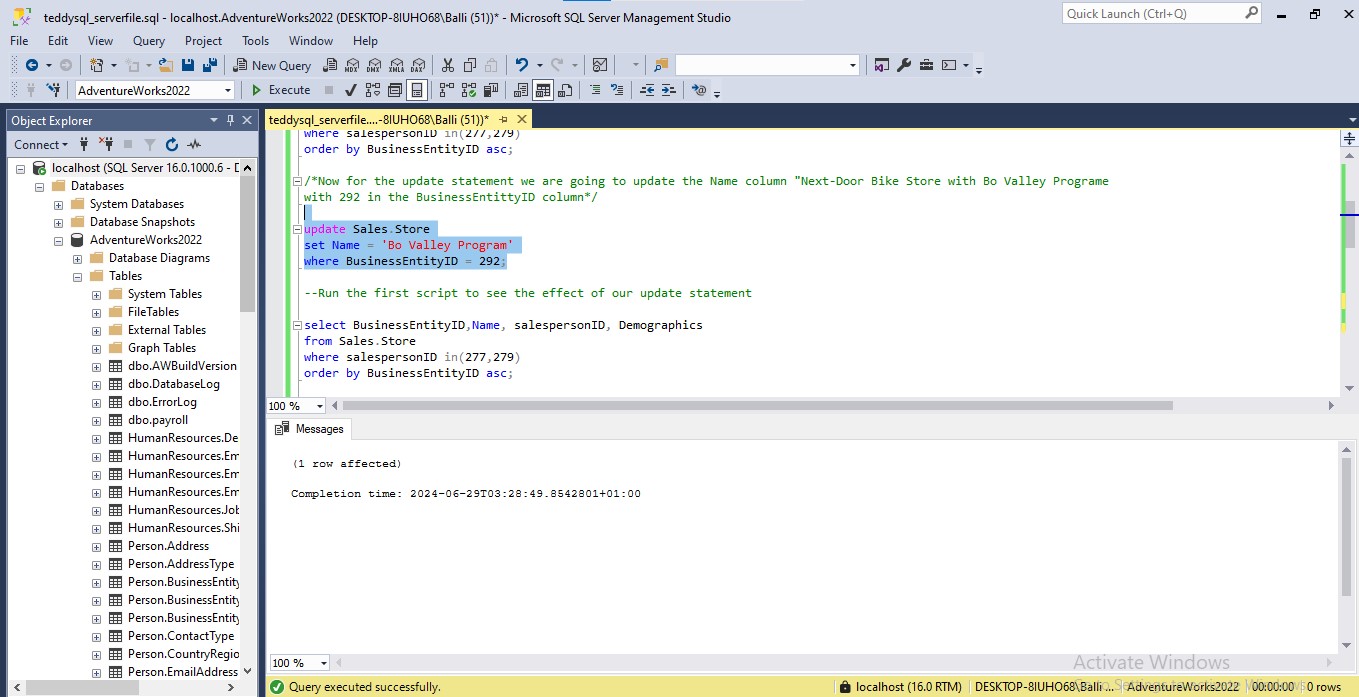
Keeping to the guide above enumerated on a step by step process, I successfully installed Microsoft Sql Server(MSS) with its accompanying tool Sql Server Management Studio(**SSMS):** SSMS is a tool used to write scripts to Microsoft Sql Server giving it instructions to effect specific programing goal or objective.

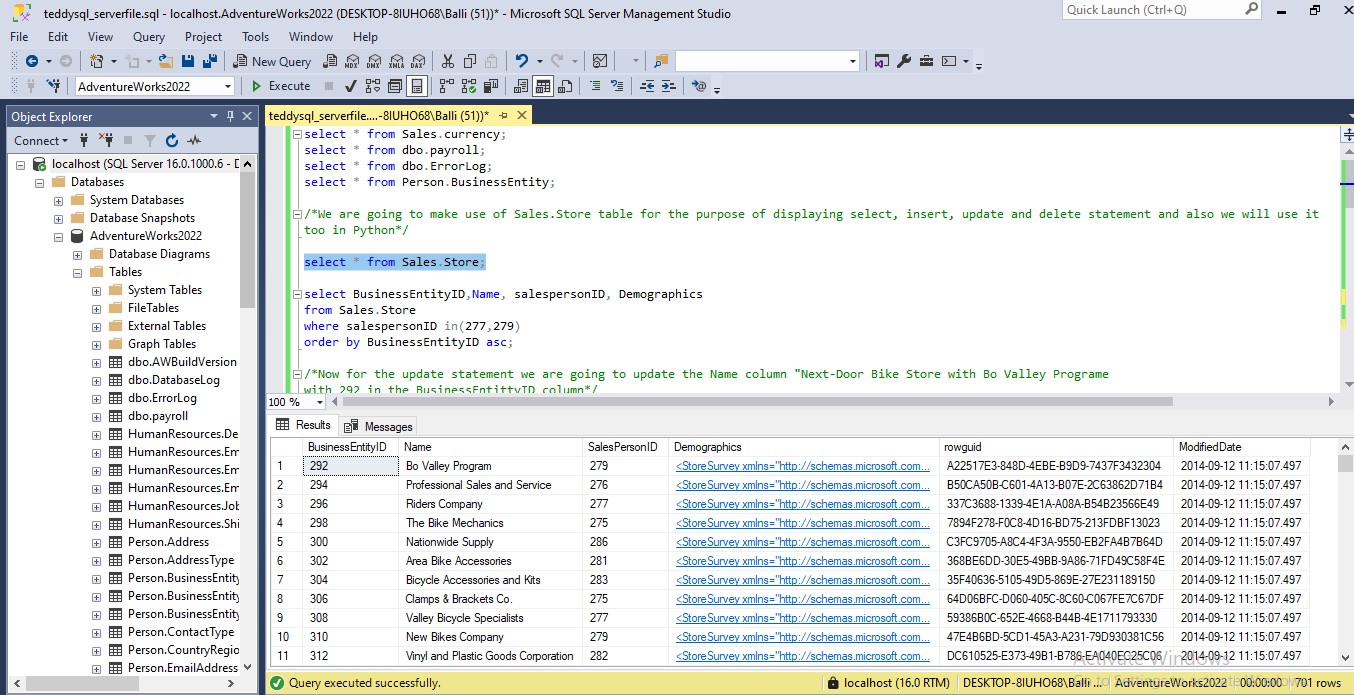
Below is a screenshot of Microsoft Sql Server and Sql Server Management Studio and also SQL scripts or command making data manipulation and data definition languages e.g. select, insert, update and delete statements.











Having installed the Microsoft Server and Microsoft server management studio and tested various SQL statements in my machine, I then proceed to diversify and utilize other tools such as Python.

making use of the python on relational databases requires important Python module;

Sqlalchemy This is a tool in Python used to access databases and perform SQL statement operations. It is obvious from the aforementioned definition of SQLALCHEMY that it total operation involve two steps which are Access(Connection) and SQL statement operations.

**ACCESS/CONNECTION**

To further expand our knowledge about Sqlalchemy and its connection to different databases structure or relational database management system(RDBMS). It does this through connection strings (a string statement/expression that Sqlalchemy uses to connect to database system from Python). Every database has its own Python connection client or module that SQLALCHEMY uses in its connection string expression to connect, for the sake of this study I will only focus on identifying for Oracle and Microsoft server.

**ORACLE:**

As a norm there are important parameters that should be configured or made available before any connection to a database or relational database management system. Oracle is no different and these parameters include.

**Oracledb:** A python connection client or module for Oracle database

**Username:** this is the user who wishes to access Oracle database, this user must have been created at the oracle database system

**Password:** the password used by the username or user to gain access

**Service Name:** consisting of the database name and the domain the user wishes to connect to. That is the instance the user will be able to write Sql commands to.

**Port:** oracle uses a port in its connection, usually the default is 1521

**Host name:** this is usually the machine in which the user is located

It is important to talk about oracle listener and Tnsname when discussing connections because this are the two file that must be configured before any connection can be possible. And for this assessment it is not in the scope to discuss further. So I will suggest reading it up. There is another important thing to note in connections to database using python. And it is the python module that is unique to every database system, this module must be present to enable python recognize the database we are trying to connect. For Oracle this python module is Oracledb which is an upgraded version with python 3 and above, different from the conventional cx\_oracle.

Now that we have fully listed the parameters we need to connect to oracle database using python Sqlalchemy, we cannot proceed to creating the connection strings.

Note we will not be discussing further, after the connection strings is created because it is not the database we are going to use for this assessment.

Python code

**Oracle connection strings=**

**f” oracle+oracledb://{username}: {password}@{hostname}: {port}/ {service name}”**

**Connection becomes**

**Engine =create engine (Oracle connection strings)**

Note: **The create engine statement is from SQLALCHEMY**

**MySQL:**

Mysql is another relational database management system that manages databases for its client. To get to MYSQL using python Sqlalchemy we should define the necessary connection parameter as we did with Oracle.

As we discussed above the unique python SQLALCHEMY connection module for MYSQL SERVER is mysqlclient. This is what we will be making use of in defining our connection strings. First thing first let us define our other parameters.

**Username:** this is the user who wishes to access Oracle database

**Password:** the password used by the username or user to gain access

**Database name:** this is the database we are connecting to inside the Mysql server

**Port:** Mysql server port for connection, usually the default is 3306

**Host name:** this is usually the machine in which the user is located

Python code

**MySQL server connection strings=**

**f”mysql+mysqlclient://{username}:{password}@{host name}:{port}/{database name}”**

**Connection becomes**

**Engine =create engine (MySQL server connection strings)**

**Microsoft SQL SERVER**

It is no longer news that every server of a relational database management system needs a connection parameter to be configured before initiating any connection command using python Sqlalchemy. Pyodbc is the unique python SQLALCHEMY module for connection to Microsoft Sql server just like others above.

**Username:** this is the user who wishes to access Oracle database

**Password:** the password used by the username or user to gain access

**Database name:** this is the database we are connecting to inside the Mysql server

**Driver:** this is a file that aids communication between python and Microsoft Sql server

**Host name:** this is usually the machine in which the server is located

**Microsoft SQL Server connection string =**

**f'mssql+pyodbc://{user}@{server}/{database}? Trusted Connection=Yes&driver={Driver}'**

**Connection becomes:**

**Engine = create engine (Microsoft SQL Server connection string)**

The assessment is to focus on Microsoft SQL server, you can find attached the Python file with Python methods that creates a connection to Microsoft SQL server automatically once the code runs and also another method that acts like the Microsoft server management studio to write and read SQL statement to the database dynamically interacting and prompting the user to write his or her SQL statement.

Why is the Project a game changer?

It is a game changer, because the Python programs written for this project is dynamic and interactive, it communicates with user and also allow the opportunity to write SQL statement as if it were a database tool like the SSMS/SQL developer.

Another reason This program was designed to access too database which are MSSQL Sever and Oracle. How does it do this? It does it by first asking the user to choose which of the database he or she wants to access, and there after it proceed to creating a connection for the respective database and open up an SQL statement prompt for the user to work with. This is important because it saves the user resources and time of having to install the database tool for both MSSQL Sever and Oracle (SSMS, SQL developer), with this program the user can utilize both functionalities seamlessly.

**Please find attached the Python file.**