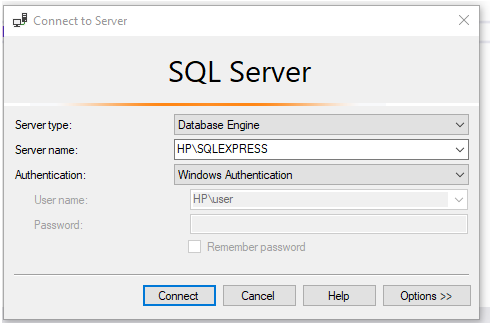
CONNECTION TO MICROSOFT SQL SERVER DATABASE

Step 1: go into your Microsoft SQL Server Studio. Copy "Server Name", "Username" as it appears from your SQL Server pop up window



Step 2: Open your JupyerNote, open a new file, Copy the code below and past it and save the file. Maybe as: DB_read_MSSM.

Step 3: install the following dependencies:

#!pip install pyodbc

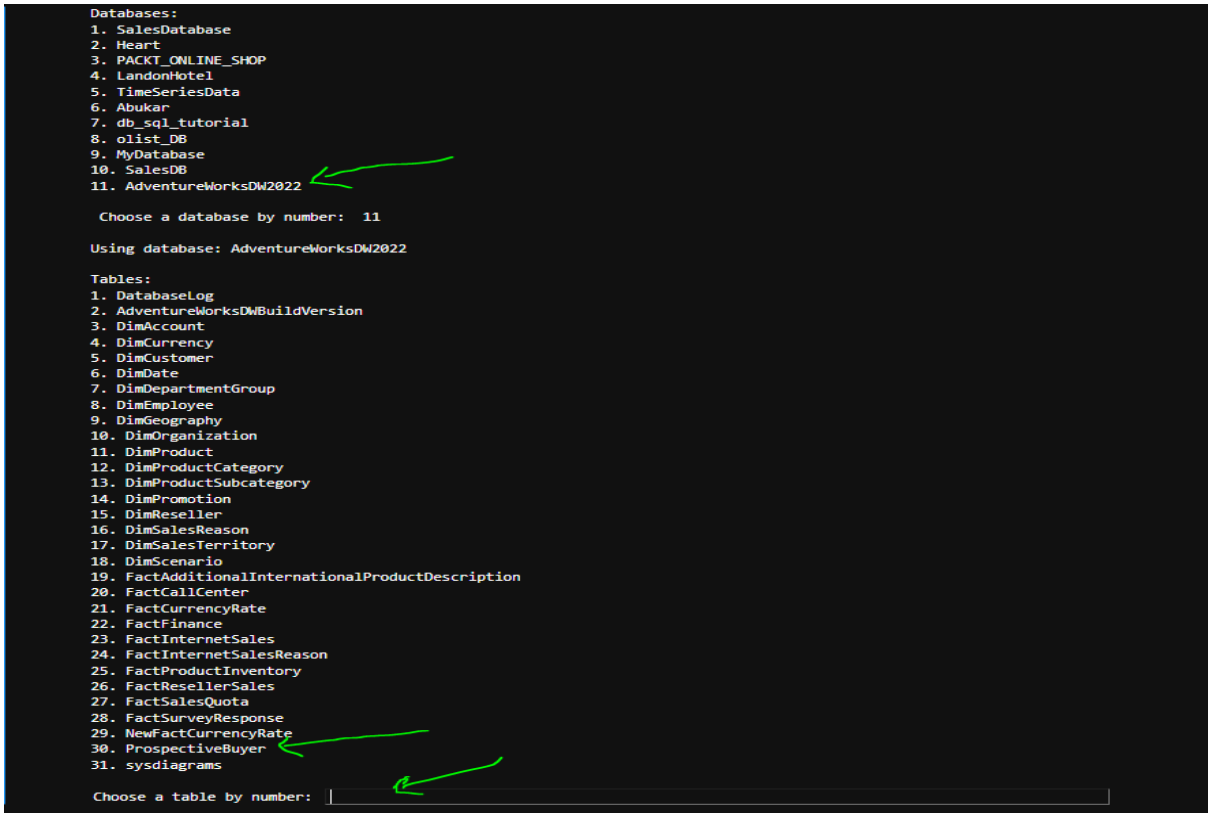
#!pip install sqlalchemy

Step 4: Change the server name**. this was part of your credentials – you should already know it!**

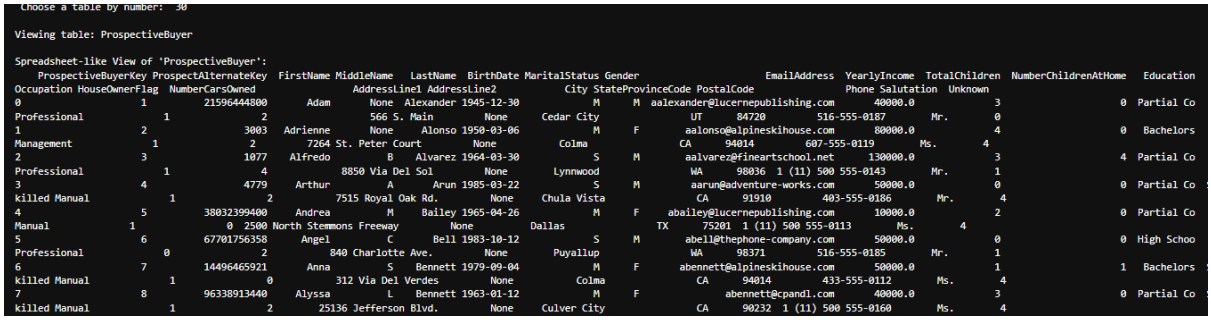
```
import pyodbc
import pandas as pd
from sqlalchemy import create_engine

# Step 1: Connect using Windows Authentication
server = r'HP\SQLEXPRESS'
connection_string = f'''
DRIVER={{ODBC Driver 17 for SQL Server}};
SERVER={server};
Trusted_Connection=yes;
'''

try:
    # Initial connection
    connection = pyodbc.connect(connection_string, autocommit=True)
    cursor = connection.cursor()

    # Step 2: Show all non-system databases
    cursor.execute("SELECT name FROM sys.databases WHERE database_id > 4")
    databases = [row[0] for row in cursor.fetchall()]
    print("Databases:")
    for i, db in enumerate(databases, 1):
        print(f"{i}. {db}")

    db_choice = int(input("\n Choose a database by number: ")) - 1
    selected_db = databases[db_choice]
    print(f"\nUsing database: {selected_db}")

    # Reconnect to selected DB
    connection = pyodbc.connect(f'''
        DRIVER={{ODBC Driver 17 for SQL Server}};
        SERVER={server};
        DATABASE={selected_db};
        Trusted_Connection=yes;
    ''')
    cursor = connection.cursor()

    # Step 3: List tables
    cursor.execute("""
        SELECT TABLE_NAME
        FROM INFORMATION_SCHEMA.TABLES
        WHERE TABLE_TYPE='BASE TABLE'
    """)
    tables = [row[0] for row in cursor.fetchall()]
    print("\nTables:")
    for i, tbl in enumerate(tables, 1):
        print(f"{i}. {tbl}")

    tbl_choice = int(input("\n Choose a table by number: ")) - 1
    selected_table = tables[tbl_choice]
    print(f"\nViewing table: {selected_table}")

    # Step 4: Load table data into a Pandas DataFrame
    conn_str = (
        f"mssql+pyodbc://{server}/{selected_db}"
        "?trusted_connection=yes&driver=ODBC+Driver+17+for+SQL+Server"
    )
    engine = create_engine(conn_str)

    query = f"SELECT * FROM [{selected_table}]"
    df = pd.read_sql(query, engine)

    print(f"\nSpreadsheet-like View of '{selected_table}':")
    print(df.head(20))  # Show top 20 rows like a preview

except Exception as e:
    print("\nSOMETHING WENT WRONG:")
    print(e)
```

Step 5: run it. you should connect to your MS SQL server and if it has worked, it will show you the list of all the Databases you have access to. You will be prompted to select any of the database. Enter the number for the preferred database.

It will then show you a list of all the tables from the selected database. Awesome!



Select a table, and you will see the complete details of that table.



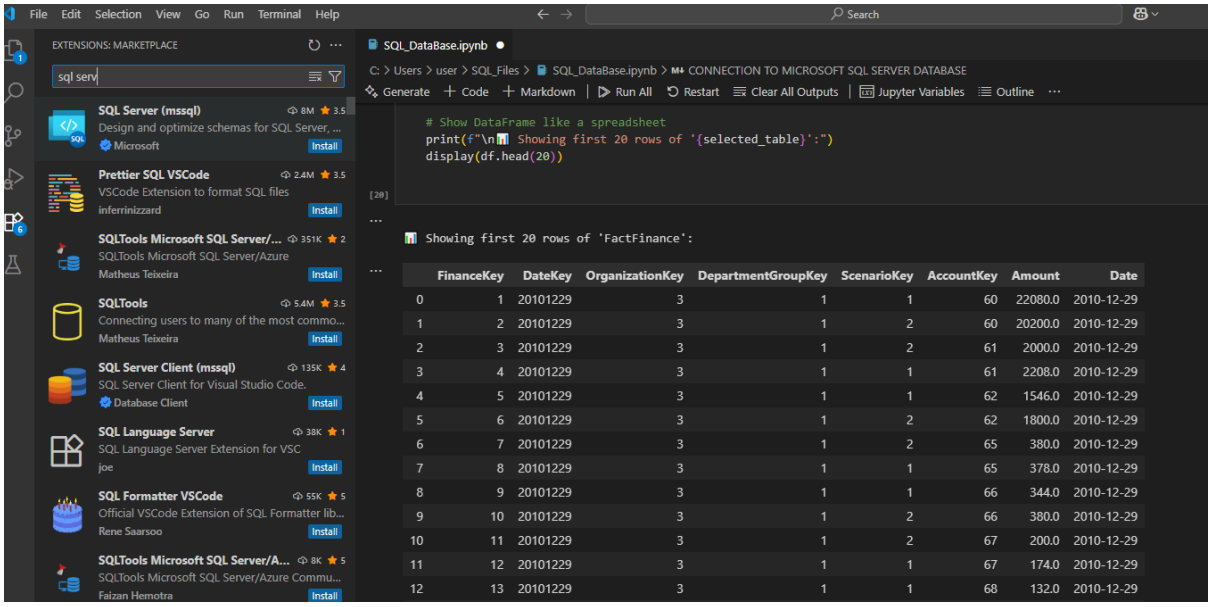
Step 5: now you can do your analysis. View the data in panda data frame, see the size of the data, very quickly. Run the following code in the next cell to see the following output.

```python
from IPython.display import display
import pandas as pd
from sqlalchemy import create_engine

# Setup (reusing previous variables)
conn_str = (
    f"mssql+pyodbc://{server}/{selected_db}"
    "?trusted_connection=yes&driver=ODBC+Driver+17+for+SQL+Server"
)
engine = create_engine(conn_str)

# Load table
query = f"SELECT * FROM [{selected_table}]"
df = pd.read_sql(query, engine)

# Optional display settings
pd.set_option('display.max_columns', None)
pd.set_option('display.width', 1000)

# Show DataFrame like a spreadsheet
print(f"\n📊 Showing first 20 rows of '{selected_table}':")
display(df.head(20))
```

```
📊 Showing first 20 rows of 'ProspectiveBuyer':
```

	ProspectiveBuyerKey	ProspectAlternateKey	FirstName	MiddleName	LastName	BirthDate	MaritalStatus	Gender	EmailAddress	YearlyIncome	TotalChildren	NumberChildrenAtHome	Education	Occupatic
0	1	21596444800	Adam	None	Alexander	1945-12-30	M	M	aalexander@lucernepublishing.com	40000.0	3	0	Partial Co	Profession
1	2	3003	Adrienne	None	Alonso	1950-03-06	M	F	aalonso@alpineskihouse.com	80000.0	4	0	Bachelors	Manageme
2	3	1077	Alfredo	B	Alvarez	1964-03-30	S	M	aalvarez@fineartschool.net	130000.0	3	4	Partial Co	Profession

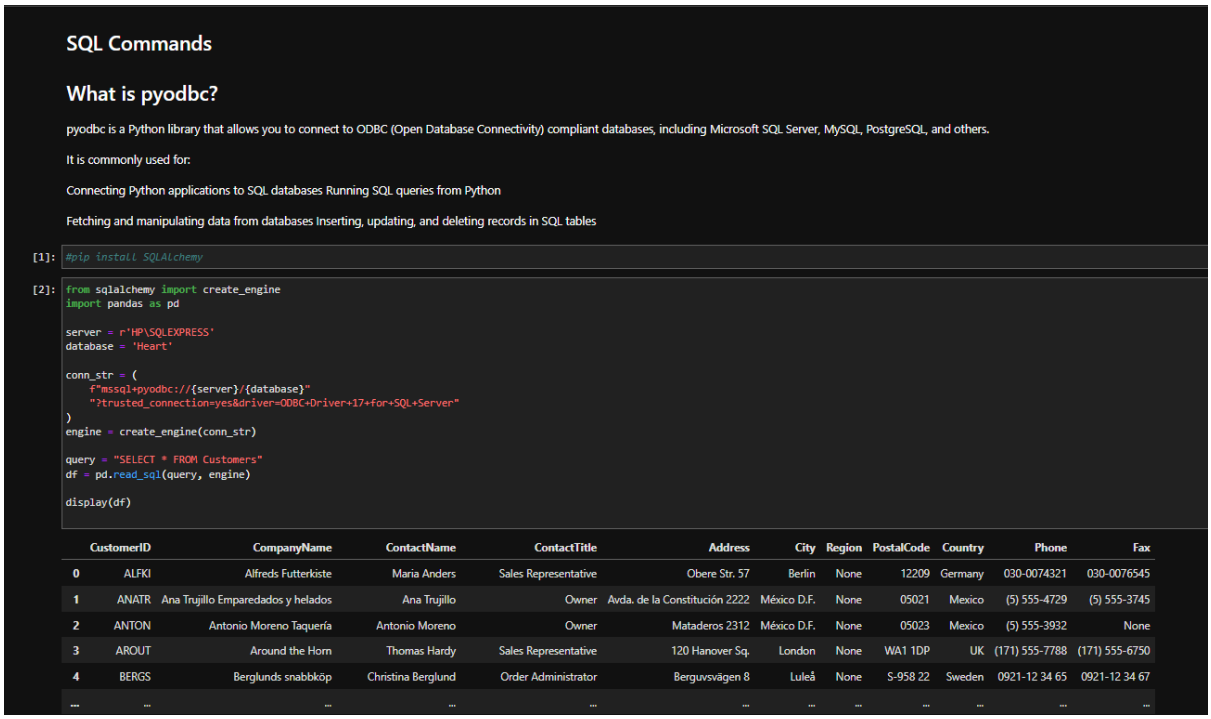
Step 5: if you prefer, you can open the same file in VSCODE and run it there as well. Check if you may need to install MS SQL Server or simply add. It did work for me, and it is possible I may have already installed this sometime ago. 😊.



	FinanceKey	DateKey	OrganizationKey	DepartmentGroupKey	ScenarioKey	AccountKey	Amount	Date
0	1	20101229	3	1	1	60	22080.0	2010-12-29
1	2	20101229	3	1	2	60	20200.0	2010-12-29
2	3	20101229	3	1	2	61	2000.0	2010-12-29
3	4	20101229	3	1	1	61	2208.0	2010-12-29
4	5	20101229	3	1	1	62	1546.0	2010-12-29
5	6	20101229	3	1	2	62	1800.0	2010-12-29
6	7	20101229	3	1	2	65	380.0	2010-12-29
7	8	20101229	3	1	1	65	378.0	2010-12-29
8	9	20101229	3	1	1	66	344.0	2010-12-29
9	10	20101229	3	1	2	66	380.0	2010-12-29
10	11	20101229	3	1	2	67	200.0	2010-12-29
11	12	20101229	3	1	1	67	174.0	2010-12-29
12	13	20101229	3	1	1	68	132.0	2010-12-29

So, there you go have it. I really like running this from JupyeterNote book either on it is own or via VSCODE.

If you want to interact with the database quickly and write your SQL queries right away, you can also follow this simple code.



SQL Commands

What is pyodbc?

pyodbc is a Python library that allows you to connect to ODBC (Open Database Connectivity) compliant databases, including Microsoft SQL Server, MySQL, PostgreSQL, and others.

It is commonly used for:

Connecting Python applications to SQL databases Running SQL queries from Python

Fetching and manipulating data from databases Inserting, updating, and deleting records in SQL tables

```
[1]: #pip install SQLAlchemy
```

```
[2]: from sqlalchemy import create_engine
     import pandas as pd

     server = r'HP\SQLEXPRESS'
     database = 'Heart'

     conn_str = (
         f"mssql+pyodbc://{server}/{database}"
         "?trusted_connection=yes&driver=ODBC+Driver+17+for+SQL+Server"
     )
     engine = create_engine(conn_str)

     query = "SELECT * FROM Customers"
     df = pd.read_sql(query, engine)

     display(df)
```

	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone	Fax
0	ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57	Berlin	None	12209	Germany	030-0074321	030-0076545
1	ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2222	México D.F.	None	05021	Mexico	(5) 555-4729	(5) 555-3745
2	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	None	05023	Mexico	(5) 555-3932	None
3	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	None	WA1 1DP	UK	(171) 555-7788	(171) 555-6750
4	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berguvsvägen 8	Luleå	None	S-958 22	Sweden	0921-12 34 65	0921-12 34 67
...	...	...	...	...	...	...	...	...	...	...	...

I think this is a quick and easy way to interact with a database and run SQL queries. it does not necessarily have to be MS SQL server, but other databases can be connected to via JupyetNote.

Simply download the code from the following link:

https://github.com/abukar10/MSSQLSERVER_JUPYETNOTE

Any comments, Feedback:

Abukar Ali

Abukar10@icloud.com