1. Install the following software :

SQL Server Express + SQL Server Management Studio (SSMS)

- Links: SQL Server Downloads
- Install SQL Server Express with default settings.
- Install SSMS to manage the database.

AdventureWorks Sample Database

- Use the link: AdventureWorks Installation
- Make sure it's installed in your SQL Server.

PostgreSQL
Download & install: PostgreSQL Downloads

- Default password: 12345 (you can use demopass for your ETL role as in your code).
- Also install pgAdmin (optional but helps to visualize database).

Python environment

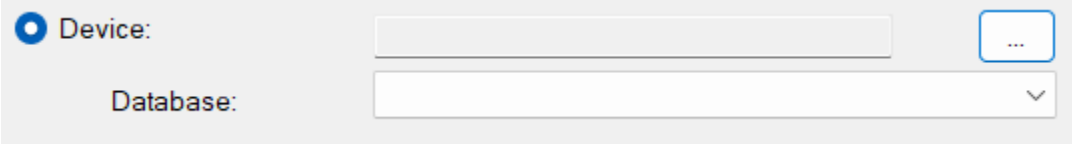
- Make sure you have Python 3.9+, pandas, sqlalchemy, and psycopg2 installed.
- Install via pip

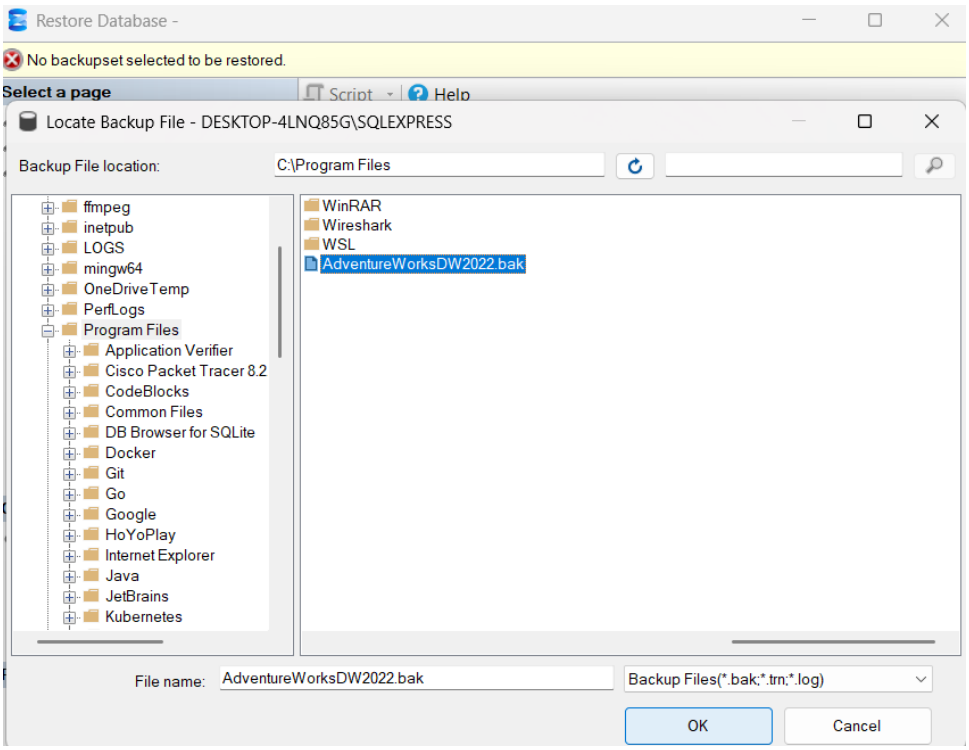
ODBC Driver for SQL Server

- Download ODBC Driver 17

2. Open SSMS(Through Visual Studio):

1. Connect to the sql server that you downloaded(usually ends with \SQLEXPRESS)
2. After connecting, Databases->restore databases->device, click on the three dots and click on add, then select your adventureworks.bak file.





3. Now open ‘New Query’ and run the following code:

```
USE master;
GO
CREATE LOGIN etl
WITH PASSWORD = 'demopass',
DEFAULT_DATABASE = AdventureWorksDW2022;
GO
USE AdventureWorksDW2022;
GO
CREATE USER etl FOR LOGIN etl;
GO
GRANT CONNECT TO etl;
GO
```

3.Open PGADMIN4:

1. Click on ‘Databases’->Create-> Database-> name it ‘adventureworks’->save
2. Open Query Tool and run the following code :

```
-- Creating the Role and Granting the privileges
CREATE ROLE etl WITH
LOGIN
PASSWORD 'demopass';
GRANT ALL PRIVILEGES ON DATABASE adventureworks TO etl;
```

```
-- Run this while connected to adventureworks
GRANT USAGE, CREATE ON SCHEMA public TO etl;
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO etl;
GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO
etl;
ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL PRIVILEGES ON TABLES TO etl;
ALTER DEFAULT PRIVILEGES IN SCHEMA public
GRANT ALL PRIVILEGES ON SEQUENCES TO etl;
```

4.Open cmd:

1. Run the following code to install dependencies:
   pip install psycopg2-binary pandas sqlalchemy pyodbc

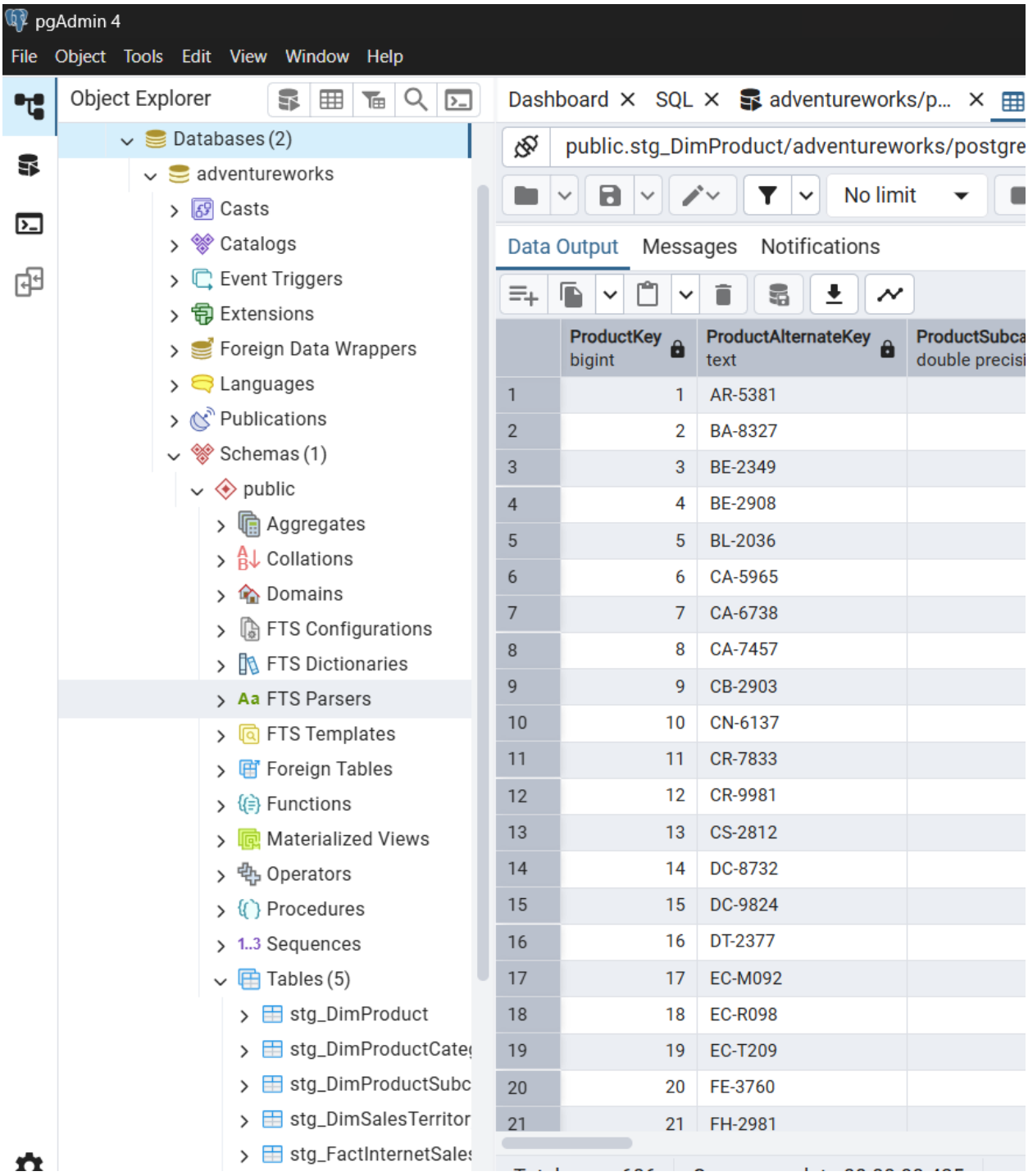
5.Running Python Script:

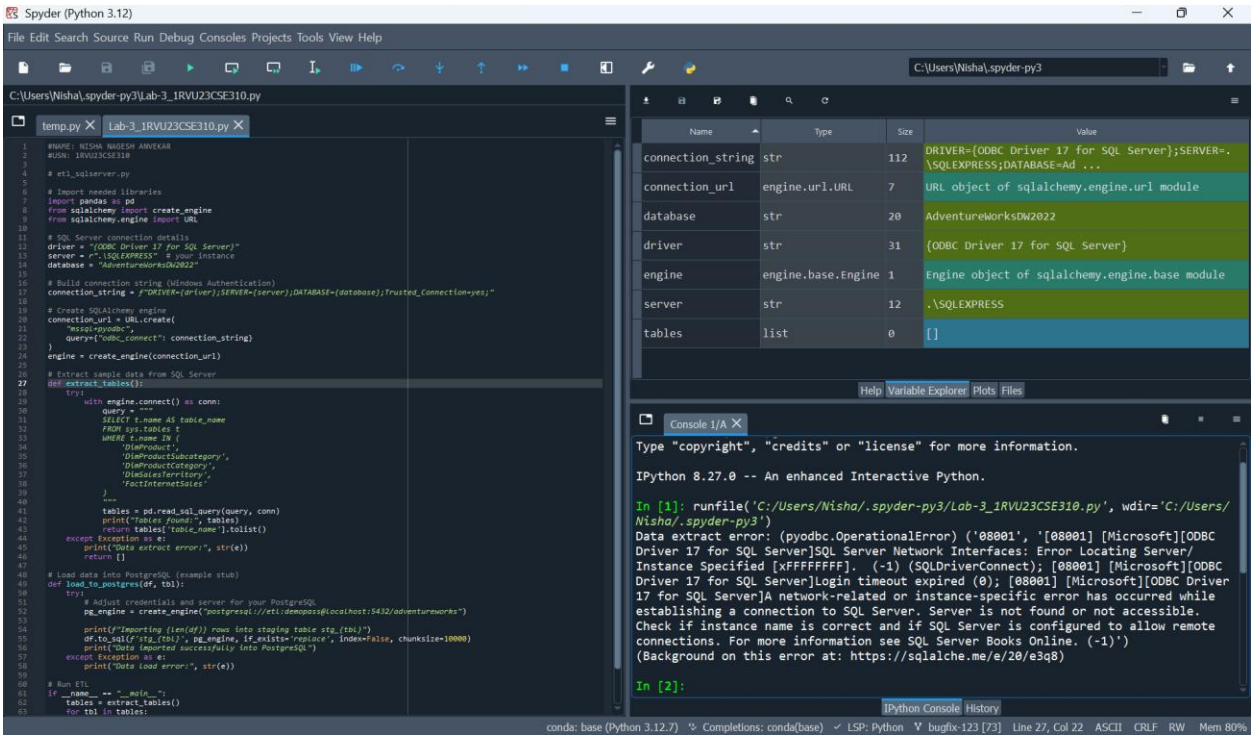
In the python script, on line 10, **'server = r".\SQLEXPRESS" # your instance'** , you need to replace it with your SQL server name. For eg: if you sql server name is DESKTOP-4LNQ85G\SQLEXPRESS then your line of code will be: **server = r"DESKTOP-4LNQ85G\SQLEXPRESS". THIS IS THE SAME SERVER THAT YOU CONNECTED TO SSMS.**

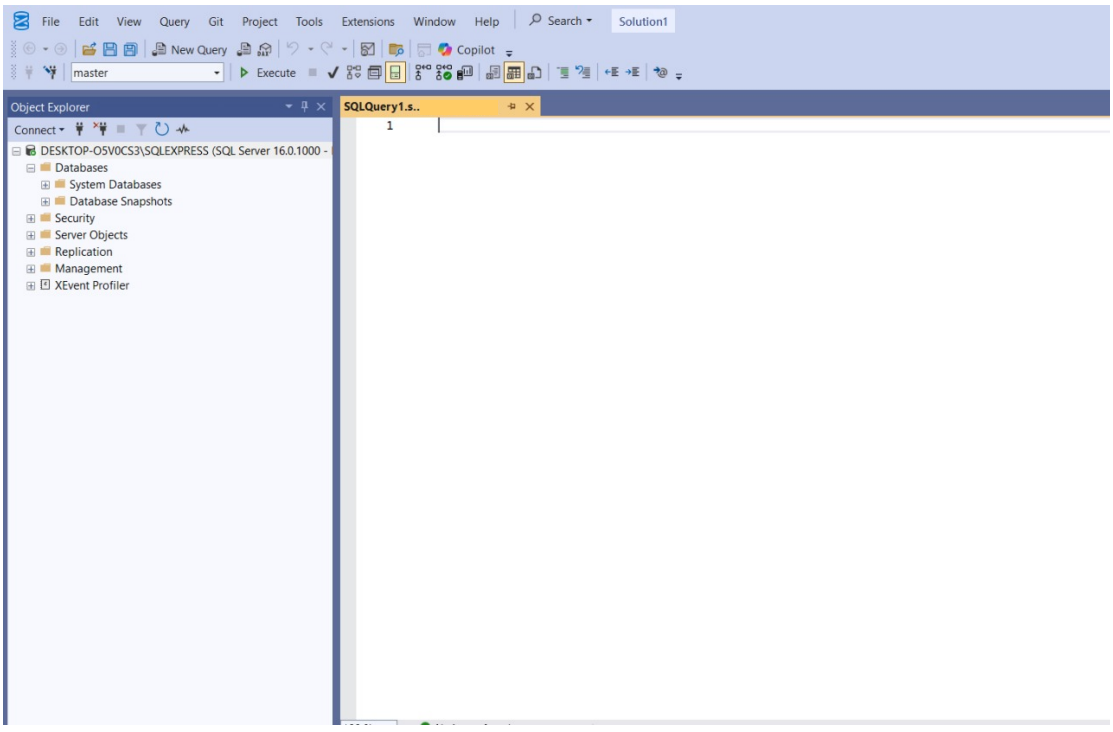
Now run the script and you should see the output in the terminal.

6. OUTPUT

Open PgAdmin->Databases->adventureworks->schemas->public->tables-> right click on any one(stg_dimproduct
stg_dimproductcategory
stg_dimproductsubcategory
stg_dimsalesterritory
stg_factinternetsales) ->View and Edit-> Take an ss and submit


pgAdmin 4
File Object Tools Edit View Window Help
Object Explorer
Databases (2)
adventureworks
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas (1)
public
Aggregates
Collations
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (5)
stg_DimProduct
stg_DimProductCate
stg_DimProductSubc
stg_DimSalesTerritor
stg_FactInternetSales
Dashboard
SQL
adventureworks/p...
public.stg_DimProduct/adventureworks/postgre
No limit
Data Output
Messages
Notifications
ProductKey bigint
ProductAlternateKey text
ProductSubca double precisi
1 1 AR-5381
2 2 BA-8327
3 3 BE-2349
4 4 BE-2908
5 5 BL-2036
6 6 CA-5965
7 7 CA-6738
8 8 CA-7457
9 9 CB-2903
10 10 CN-6137
11 11 CR-7833
12 12 CR-9981
13 13 CS-2812
14 14 DC-8732
15 15 DC-9824
16 16 DT-2377
17 17 EC-M092
18 18 EC-R098
19 19 EC-T209
20 20 FE-3760
21 21 FH-2981


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Nisha\.spyder-py3
C:\Users\Nisha\.spyder-py3\Lab-3_1RVU23CSE310.py
temp.py
Lab-3_1RVU23CSE310.py
#NAME: NISHA NAGESH ANVEKAR
#USN: 1RVU23CSE310
# etl_sqlserver.py
# Import needed libraries
import pandas as pd
from sqlalchemy import create_engine
from sqlalchemy.engine import URL
# SQL Server connection details
driver = "{ODBC Driver 17 for SQL Server}"
server = r".\SQLEXPRESS" # your instance
database = "AdventureWorksDW2022"
# Build connection string (Windows Authentication)
connection_string = f"DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Connection=yes;"
# Create SQLAlchemy engine
connection_url = URL.create(
"mssql+pyodbc",
query={"odbc_connect": connection_string}
)
engine = create_engine(connection_url)
# Extract sample data from SQL Server
def extract_tables():
try:
with engine.connect() as conn:
query = """
SELECT t.name AS table_name
FROM sys.tables t
WHERE t.name IN (
'DimProduct',
'DimProductSubcategory',
'DimProductCategory',
'DimSalesTerritory',
'FactInternetSales'
)
"""
tables = pd.read_sql_query(query, conn)
print("Tables found:", tables)
return tables['table_name'].tolist()
except Exception as e:
print("Data extract error:", str(e))
return []
# Load data into PostgreSQL (example stub)
def load_to_postgres(df, tbl):
try:
# Adjust credentials and server for your PostgreSQL
pg_engine = create_engine("postgresql://etl:demopass@localhost:5432/adventureworks")
print(f"Importing {len(df)} rows into staging table stg_{tbl}")
df.to_sql(f'stg_{tbl}', pg_engine, if_exists='replace', index=False, chunksize=10000)
print("Data imported successfully into PostgreSQL")
except Exception as e:
print("Data load error:", str(e))
# Run ETL
if __name__ == "__main__":
tables = extract_tables()
for tbl in tables:
Name
Type
Size
Value
connection_string
str
112
DRIVER={ODBC Driver 17 for SQL Server};SERVER=.\SQLEXPRESS;DATABASE=Ad ...
connection_url
engine.url.URL
7
URL object of sqlalchemy.engine.url module
database
str
20
AdventureWorksDW2022
driver
str
31
{ODBC Driver 17 for SQL Server}
engine
engine.base.Engine
1
Engine object of sqlalchemy.engine.base module
server
str
12
.\SQLEXPRESS
tables
list
0
[]
Help
Variable Explorer
Plots
Files
Console 1/A
Type "copyright", "credits" or "license" for more information.
IPython 8.27.0 -- An enhanced Interactive Python.
In [1]: runfile('C:/Users/Nisha/.spyder-py3/Lab-3_1RVU23CSE310.py', wdir='C:/Users/Nisha/.spyder-py3')
Data extract error: (pyodbc.OperationalError) ('08001', '[08001] [Microsoft][ODBC Driver 17 for SQL Server]SQL Server Network Interfaces: Error Locating Server/ Instance Specified [xFFFFFFFF]. (-1) (SQLDriverConnect); [08001] [Microsoft][ODBC Driver 17 for SQL Server]Login timeout expired (0); [08001] [Microsoft][ODBC Driver 17 for SQL Server]A network-related or instance-specific error has occurred while establishing a connection to SQL Server. Server is not found or not accessible. Check if instance name is correct and if SQL Server is configured to allow remote connections. For more information see SQL Server Books Online. (-1)')
(Background on this error at: https://sqlalche.me/e/20/e3q8)
In [2]:
IPython Console
History
conda: base (Python 3.12.7)
Completions: conda(base)
LSP: Python
bugfix-123 [73]
Line 27, Col 22
ASCII
CRLF
RW
Mem 80%


File Edit View Query Git Project Tools Extensions Window Help
Search
Solution1
New Query
Copilot
master
Execute
Object Explorer
Connect
DESKTOP-O5V0CS3\SQLEXPRESS (SQL Server 16.0.1000 -
Databases
System Databases
Database Snapshots
Security
Server Objects
Replication
Management
XEvent Profiler
SQLQuery1.s..
1