

Ex No: 3 Date: 10/9/25	ETL Pipeline Implementation: Exporting Data from SQL Server to PostgreSQL
---	---

Objective:

This lab experiment provides practical experience in building a basic ETL (Extract, Transform, Load) data pipeline using Python. It guides participants through the core stages of the pipeline from extracting raw data, transforming it into a usable format, and loading it into a target system. The activity also simulates the roles and responsibilities of key stakeholders, such as data engineers, data scientists, and business analysts, to demonstrate their collaborative contributions in the data pipeline.

Outcomes:

1. Identify and describe the stages of the data engineering lifecycle.
2. Explain the roles and responsibilities of different stakeholders at each stage.
3. Perform basic data engineering tasks within a simulated environment.
4. Collaborate across simulated stakeholder roles to design and implement a data-driven solution.

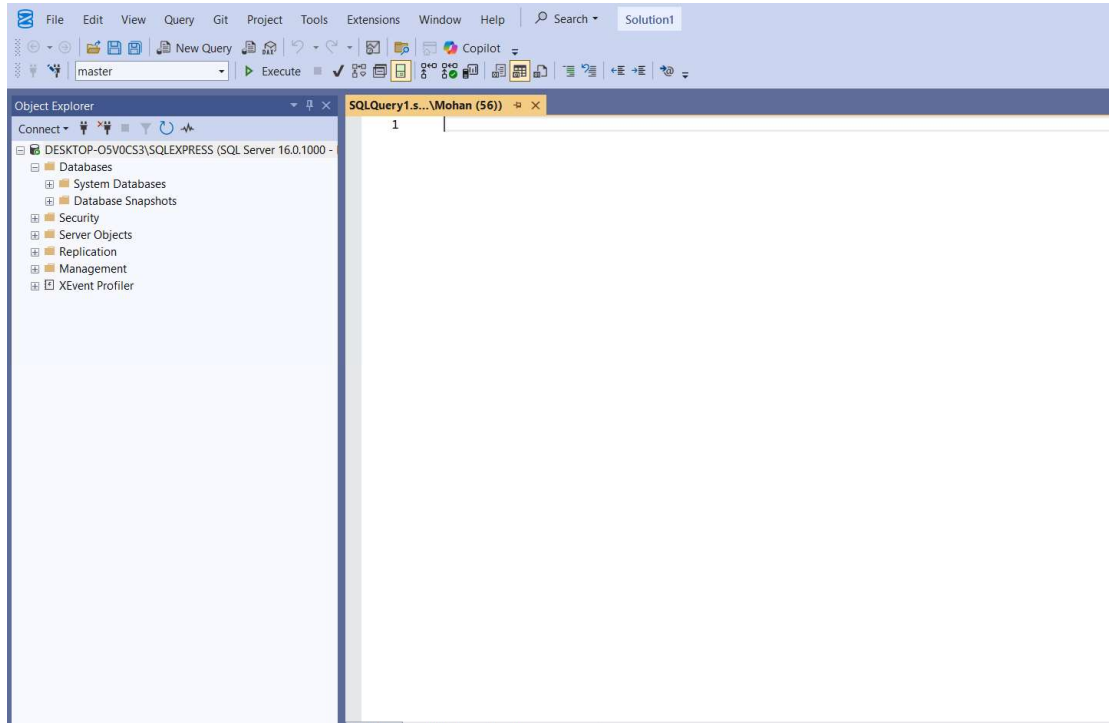
Lab Procedure**STEP 1: Install the following software :**

1. 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.
2. AdventureWorks Sample Database
 - Use the link: AdventureWorks Installation
 - Make sure it's installed in your SQL Server.

PostgreSQL

USN NUMBER: 1RVU23CSE264

NAME: Meghana G



3. 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).
Also install pgAdmin (optional but helps to visualize database).

4. Python environment

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

5. ODBC Driver for SQL Server

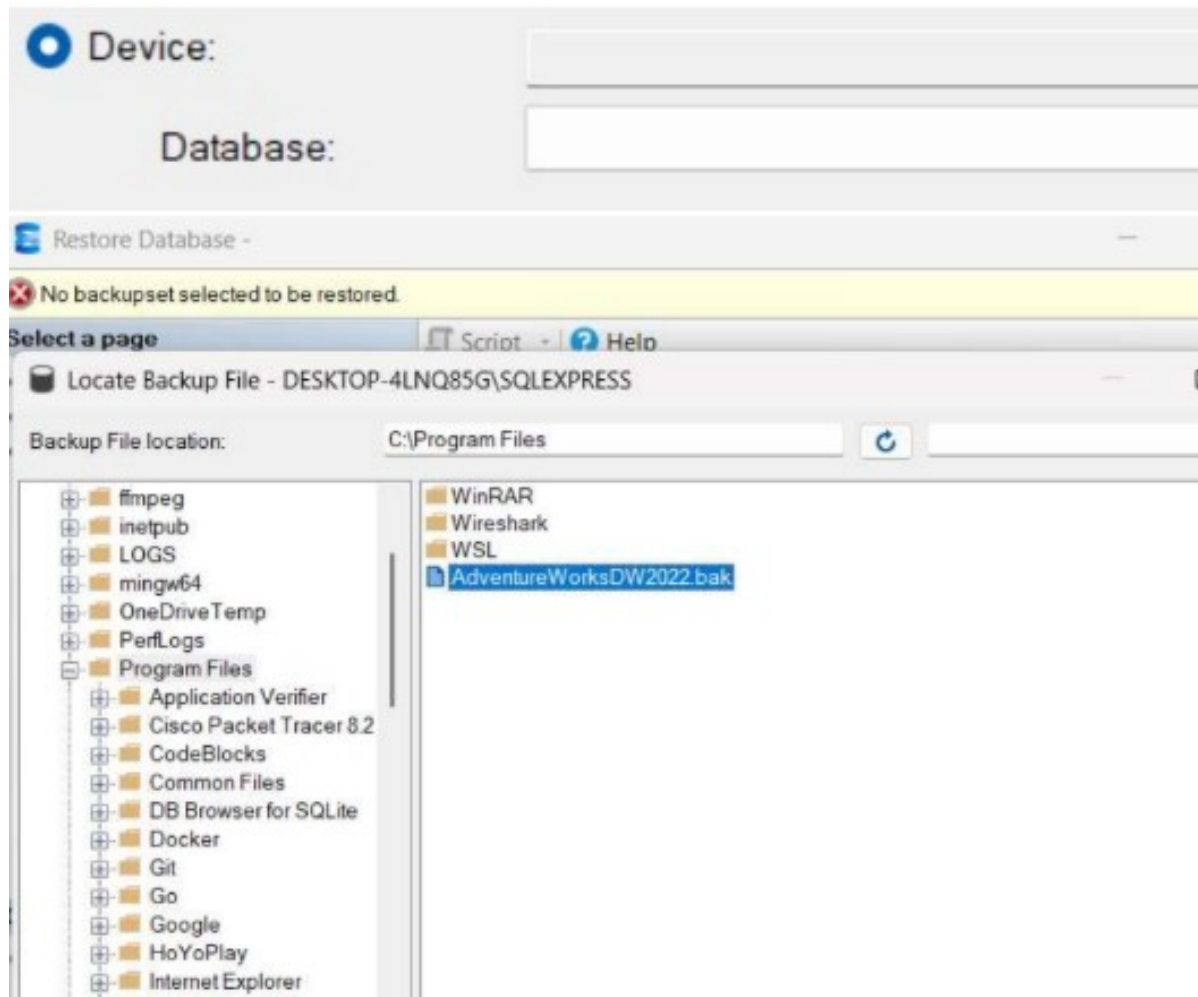
- Download [ODBC Driver 17](#)
[ODBC Driver 17](#)

Step 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.

USN NUMBER: 1RVU23CSE264

NAME: Meghana G

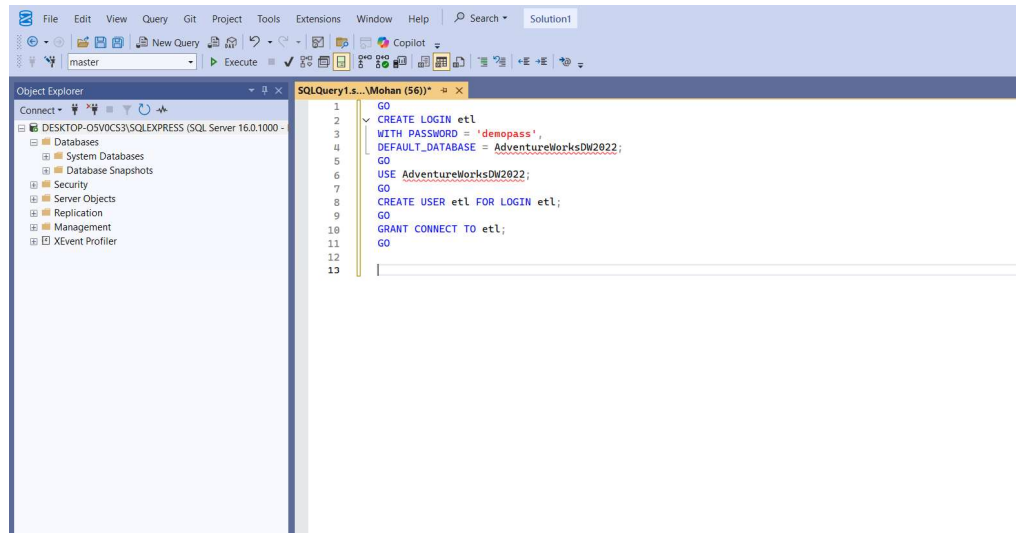


3. Now open 'New Query' and run the following code:

```
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
```

USN NUMBER: 1RVU23CSE264

NAME: Meghana G



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

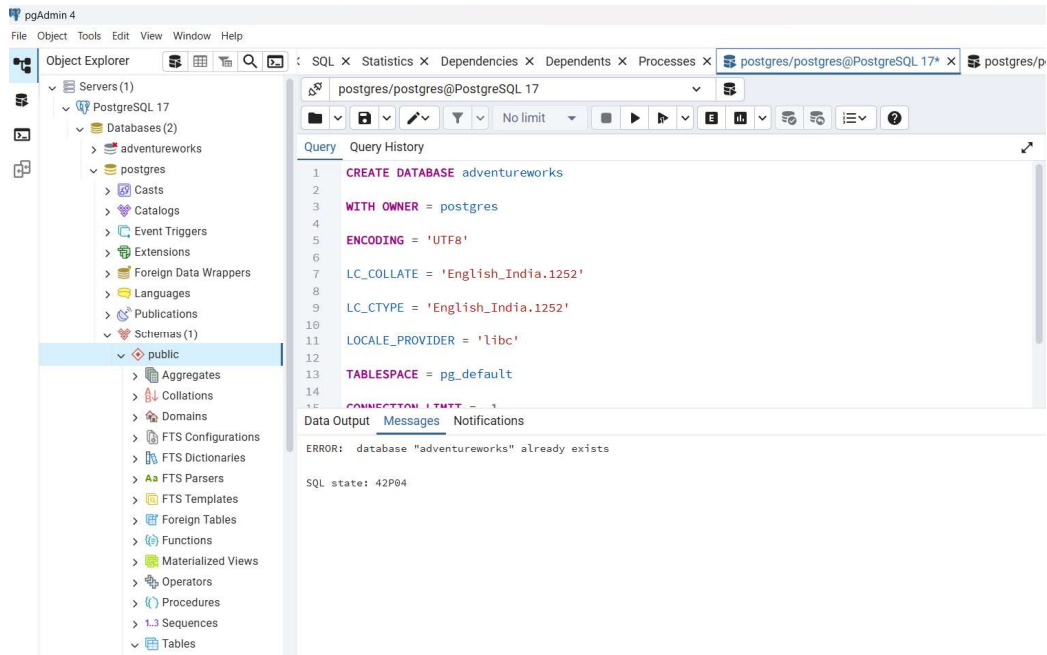
GRANT ALL PRIVILEGES ON TABLES TO etl;

ALTER DEFAULT PRIVILEGES IN SCHEMA public

GRANT ALL PRIVILEGES ON SEQUENCES TO etl;

USN NUMBER: 1RVU23CSE264

NAME: Meghana G



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.

USN NUMBER: 1RVU23CSE264

NAME: Meghana G

The top screenshot shows a Jupyter Notebook environment with a file named 'etl.py'. The code in the file is as follows:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Sep 3 11:40:59 2025
4
5 @author: Mohan
6 """
7
8 # etl_sqlserver.py
9
10 # Import needed libraries
11 import pandas as pd
12 from sqlalchemy import create_engine
13 from sqlalchemy.engine import URL
14
15 # SQL Server connection details
16 driver = "(ODBC Driver 17 for SQL Server)"
17 server = r".\SQLEXPRESS" # your instance
18 database = "AdventureworksDW2022"
19
20 # Build connection string (Windows Authentication)
21 connection_string = f"DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Co
22
23 # Create SQLAlchemy engine
24 connection_url = URL.create(
25     "mssql+pyodbc",
26     query={"odbc_connect": connection_string}
27 )
28 engine = create_engine(connection_url)
29
30 # Extract sample data from SQL Server
31 def extract_tables():
32     try:
33         with engine.connect() as conn:
34             query = """
35             SELECT t.name AS table_name
36             FROM sys.tables t
37             WHERE t.name IN (
38                 'DimProduct',
39                 'DimProductSubcategory',
40             """
```

The console output shows the following error:

```
Python 3.13.5 | packaged by Anaconda, Inc. | (main, Jun 12 2025, 16:37:03) [MSC v.1929 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.30.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: %runfile C:/Users/Mohan/Downloads/etl.py --wdir
Data extract error: (pyodbc.InterfaceError) ('28000', '[28000] [Microsoft][ODBC Driver 17 for SQL Server][SQL
Server]Login failed for user '\DESKTOP-05VKC3\Mohan\''. (18456) (SQLDriverConnect); [28000] [Microsoft][ODBC
Driver 17 for SQL Server][SQL Server]Cannot open database 'AdventureworksDW2022' requested by the login. The
login failed. (4800); [28000] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Login failed for user
'\DESKTOP-05VKC3\Mohan\''. (18456); [28000] [Microsoft][ODBC Driver 17 for SQL Server][SQL Server]Cannot open
database 'AdventureworksDW2022' requested by the login. The login failed. (4800)')
(Background on this error at: https://sqlalche.me/e/20/rvf5)
```

The bottom screenshot shows the same Jupyter Notebook environment with the 'etl.py' file. The code in the file is as follows:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Sep 10 21:48:12 2025
4
5 @author: ADMIN
6 """
7
8 # etl_sqlserver.py
9
10 # Import needed libraries
11 import pandas as pd
12 from sqlalchemy import create_engine
13 from sqlalchemy.engine import URL
14
15 # SQL Server connection details
16 driver = "(ODBC Driver 17 for SQL Server)"
17 server = r".\SQLEXPRESS" # your instance
18 database = "AdventureworksDW2022"
19
20 # Build connection string (Windows Authentication)
21 connection_string = f"DRIVER={driver};SERVER={server};DATABASE={database};Trusted_Co
22
23 # Create SQLAlchemy engine
24 connection_url = URL.create(
25     "mssql+pyodbc",
26     query={"odbc_connect": connection_string}
27 )
28 engine = create_engine(connection_url)
29
30 # Extract sample data from SQL Server
31 def extract_tables():
32     try:
33         with engine.connect() as conn:
34             query = """
```

The console output shows the following successful data import:

```
In [2]:
1 DimProductCategory
2 DimProductSubcategory
3 DimSalesTerritory
4 FactInternetSales
Importing 606 rows into staging table stg_DimProduct
Data imported successfully into PostgreSQL
Importing 4 rows into staging table stg_DimProductCategory
Data imported successfully into PostgreSQL
Importing 37 rows into staging table stg_DimProductSubcategory
Data imported successfully into PostgreSQL
Importing 11 rows into staging table stg_DimSalesTerritory
Data imported successfully into PostgreSQL
Importing 60398 rows into staging table stg_FactInternetSales
Data imported successfully into PostgreSQL
```

6. OUTPUT

Open PgAdmin->Databases->adventureworks ->schemas->public->tables-> right click on any

one(stg_dimprodu

ct stg_dimproduct

category

stg_dimproductsubcategory

stg_dimsalesterritory

stg_factinternetsales) ->View and Edit-> Take an ss and submit

USN NUMBER: 1RVU23CSE264
NAME: Meghana G

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- 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

public.stg_DimProduct/adventureworks/postgres@PostgreSQL ...

Data Output Messages Notifications

Showing rows: 1 to 606 Page No:

	ProductKey bigint	ProductAlternateKey text	ProductSubcategoryKey double precision	WeightUnitMeasureCode text	SizeUnitMeasureCode text
1	1	AR-5381	[null]	[null]	[null]
2	2	BA-8327	[null]	[null]	[null]
3	3	BE-2349	[null]	[null]	[null]
4	4	BE-2908	[null]	[null]	[null]
5	5	BL-2036	[null]	[null]	[null]
6	6	CA-5965	[null]	[null]	[null]
7	7	CA-6738	[null]	[null]	[null]
8	8	CA-7457	[null]	[null]	[null]
9	9	CB-2903	[null]	[null]	[null]
10	10	CN-6137	[null]	[null]	[null]
11	11	CR-7833	[null]	[null]	[null]

pgAdmin 4

File Object Tools Edit View Window Help

Object Explorer

- 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

public.stg_DimProductCategory/adventureworks/postgres@Po...

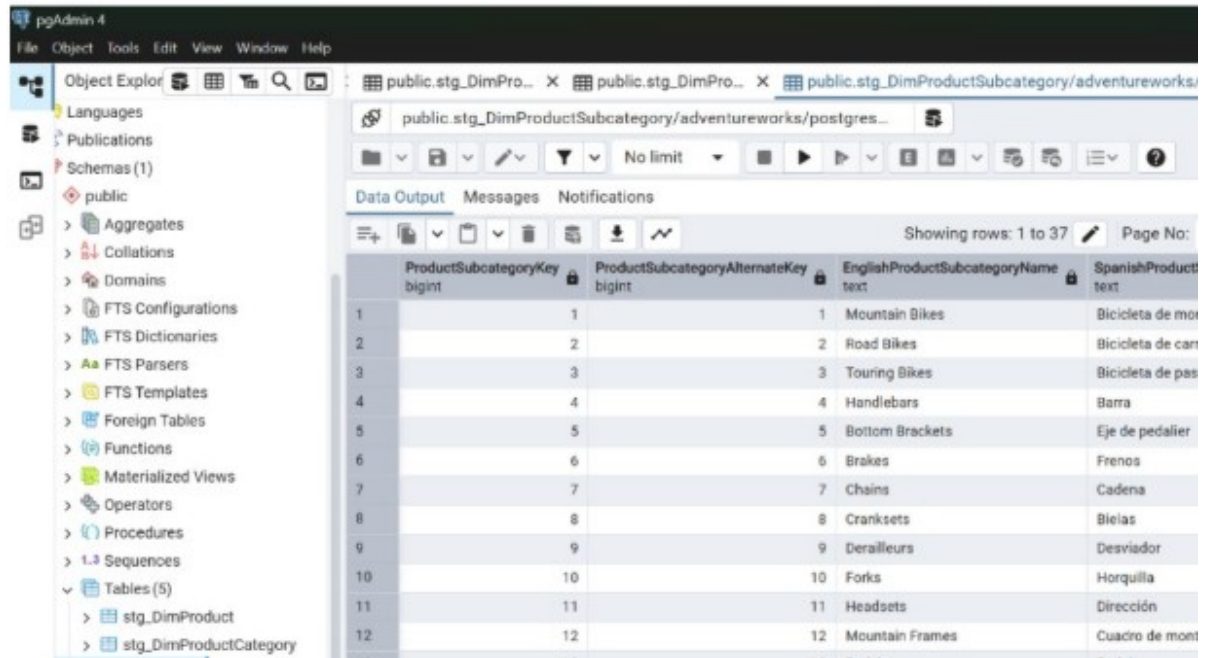
Data Output Messages Notifications

Showing rows: 1 to 4 Page No:

	ProductCategoryKey bigint	ProductCategoryAlternateKey bigint	EnglishProductCategoryName text	SpanishProductCategoryName text
1	1	1	Bikes	Bicicleta
2	2	2	Components	Componente
3	3	3	Clothing	Prenda
4	4	4	Accessories	Accesorio

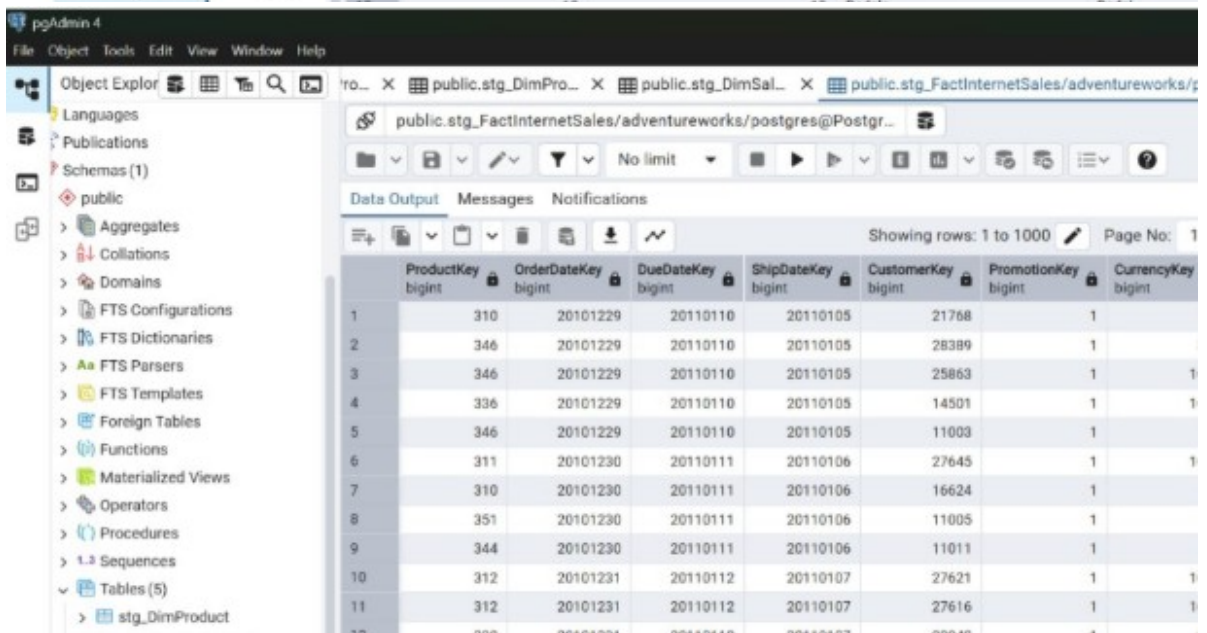
USN NUMBER: 1RVU23CSE264

NAME: Meghana G



The screenshot shows the pgAdmin 4 interface with a query window open. The query is executed against the database 'public.stg_DimProductSubcategory/ventureworks/postgres...'. The result set displays 12 rows of product subcategory data.

	ProductSubcategoryKey bigint	ProductSubcategoryAlternateKey bigint	EnglishProductSubcategoryName text	SpanishProductSubcategoryName text
1	1	1	Mountain Bikes	Bicicleta de montaña
2	2	2	Road Bikes	Bicicleta de carretera
3	3	3	Touring Bikes	Bicicleta de paseo
4	4	4	Handlebars	Barra
5	5	5	Bottom Brackets	Eje de pedalier
6	6	6	Brakes	Frenos
7	7	7	Chains	Cadena
8	8	8	Cranksets	Bielas
9	9	9	Derailleurs	Desviador
10	10	10	Forks	Horquilla
11	11	11	Headsets	Dirección
12	12	12	Mountain Frames	Cuadro de montaña



The screenshot shows the pgAdmin 4 interface with a query window open. The query is executed against the database 'public.stg_FactInternetSales/ventureworks/postgres...'. The result set displays 11 rows of fact data.

	ProductKey bigint	OrderDateKey bigint	DueDateKey bigint	ShipDateKey bigint	CustomerKey bigint	PromotionKey bigint	CurrencyKey bigint
1	310	20101229	20110110	20110105	21768	1	
2	346	20101229	20110110	20110105	28389	1	
3	346	20101229	20110110	20110105	25863	1	1
4	336	20101229	20110110	20110105	14501	1	1
5	346	20101229	20110110	20110105	11003	1	
6	311	20101230	20110111	20110106	27645	1	1
7	310	20101230	20110111	20110106	16624	1	
8	351	20101230	20110111	20110106	11005	1	
9	344	20101230	20110111	20110106	11011	1	
10	312	20101231	20110112	20110107	27621	1	1
11	312	20101231	20110112	20110107	27616	1	1

Github link:

<https://github.com/meghana1653/Data-Engineering/blob/main/LAB-3.zip>