



Tunis Business
School

Customer Shopping Behaviour

Presented to : Dr Manel Abdelkader



OUR TEAM



Zeineb TRABELSI



Zeineb HAJJI



Ghazi LASSOUED



Med Amine ZOUAGHI



OUTLINE

- Topic and Objectives
- Data Gathering
- Data Preparation (ETL process)
- Data Storage (Modeling and DWH creation)
- Data Analysis and Reporting

Topic and Objectives





Topic

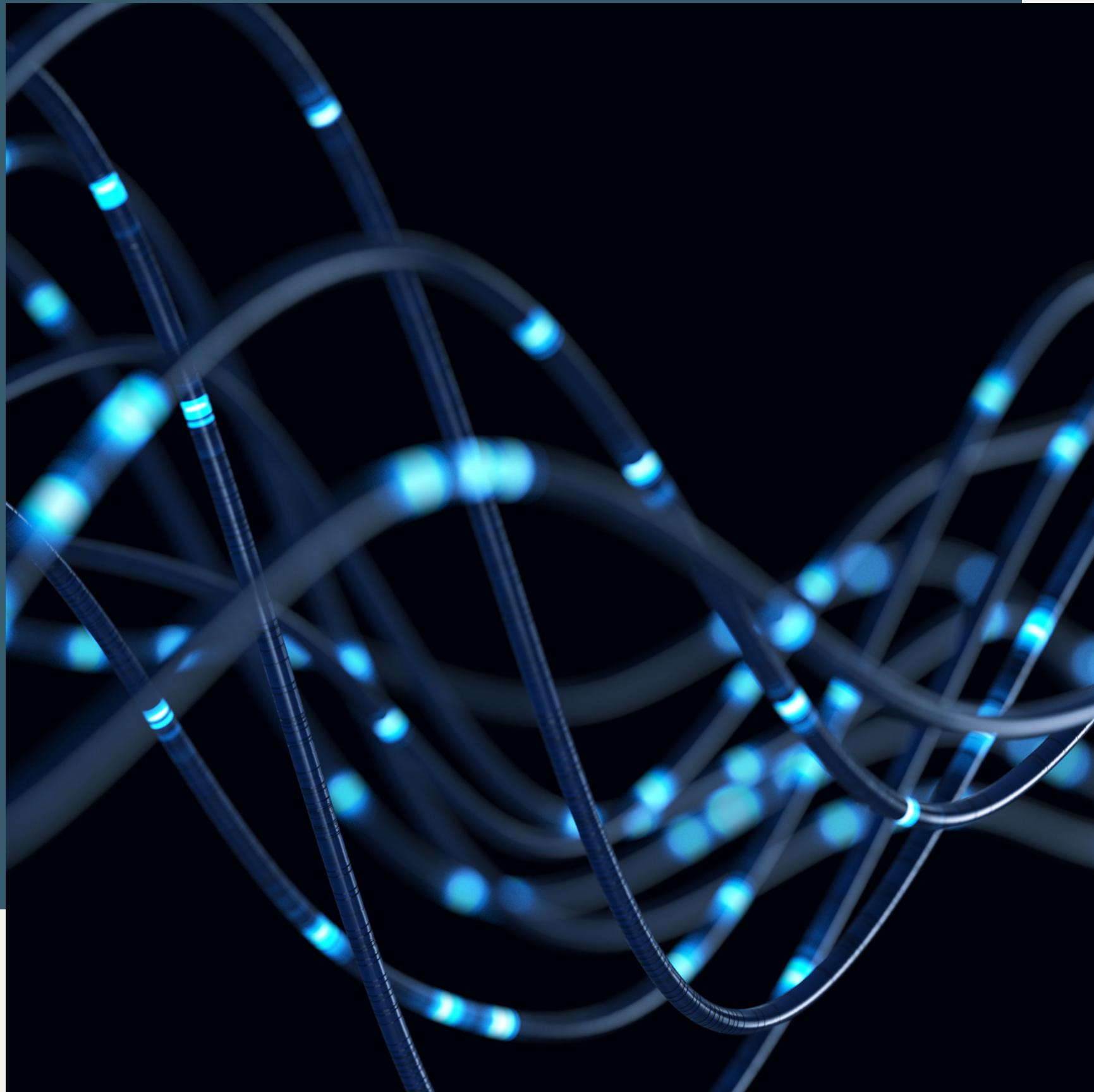
Developing a Business Intelligence (BI) solution to empower strategic decision-making based on a thorough study of last year's inventory.

Objectives

Integrate and analyze diverse data sources, utilizing a POWER BI dashboard to present key metrics. This project aims to provide actionable insights for informed decisions on store operations, territorial expansion, marketing strategies...

Data Gathering





Data Gathering

Database Search: data.gov & Kaggle.

Selection Criteria: Diversity of sources, domain, and access criteria.

Result: "Consumer Behavior and Shopping Habits Dataset" on Kaggle.

Dataset Glossary Summary

- 
- 1 Customer ID: Unique identifier for tracking customer behavior.
 - 2 Item Purchased: Specific product selected.
 - 3 Purchase Amount (USD): Transaction monetary value.
 - 4 Age: Demographic info for segmentation and marketing.
 - 5 Category: Broad classification of the item.
 - 6 Gender: Key factor in product preferences.

7 Location: Geographical purchase origin.

8 Size: Specification for apparel, footwear, etc.

9 Payment Method: Insight into preferred payment options.

10 Frequency of Purchases: Critical metric for loyalty assessment.

11 Color: Variant influencing preferences and availability.

12 Season: Seasonal relevance impacting purchasing behavior.

13 Review Rating: Customer satisfaction assessment.

14 Subscription Status: Loyalty and potential recurring revenue insight..

15 Shipping Type: Method impacting delivery times and costs.

16 Discount Applied: Indicates promotional discount application.

17 Promo Code Used: Notes use for marketing campaign evaluation.

18 Previous Purchases: Frequency contributing to segmentation and retention.

JSON & CSV FILES

The image shows a terminal window at the top displaying JSON data and a screenshot of an Excel spreadsheet below it.

Terminal Content:

```
selection View Go Run ... ← → Search
shopping_part_2.json X
Users > msi > Downloads > {} shopping_part_2.json ...
[{"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Express;Yes;Yes;14;Venmo;Fortnightly;1"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Express;Yes;Yes;2;Cash;Fortnightly;2"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Express;Yes;Yes;23;Credit Card;Weekly;3"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Next Day Air;Yes;Yes;49;PayPal;Weekly;4"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Standard;Yes;Yes;14;Venmo;Weekly;6"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;31;PayPal;Annually;5"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Free Shipping;Yes;Yes;19;Credit Card;Weekly;8"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;8;Venmo;Annually;9"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;2-Day Shipping;Yes;Yes;4;Cash;Quarterly;10"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Bank Transfer;Bi-Weekly;11"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Store Pickup;Yes;Yes;37;Express;Yes;Yes;37;Venmo;Fortnightly;13"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Express;Yes;Yes;34;Debit Card;Weekly;15"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Store Pickup;Yes;Yes;44;Debit Card;Bi-Weekly;17"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;2-Day Shipping;Yes;Yes;17;Free Shipping;Yes;Yes;17;Cash;Weekly;19"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Standard;Yes;Yes;50;Express;Yes;Yes;50;Cash;Every 3 Months;21"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Store Pickup;Yes;Yes;32;Debit Card;Annually;23"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Express;Yes;Yes;1537;Credit Card;Annually;24"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;1660;Debit Card;Annually;25"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;3888;Credit Card;Annually;26"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;3498;Credit Card;Annually;27"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;3653;Credit Card;Annually;28"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;1537;Credit Card;Annually;29"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;2737;Credit Card;Annually;30"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;15;Credit Card;Annually;31"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;1026;Credit Card;Annually;32"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;3255;Credit Card;Annually;33"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;2182;Credit Card;Annually;34"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;433;Credit Card;Annually;35"}, {"Subscription Status": "Shipping Type:Discount Applied;Promo Code Used;Previous Purchases;Payment Method:Frequency of Purchases;priceid": "Yes;Yes;2010;Credit Card;Annually;36"}]
```

Excel Spreadsheet Content:

A	B	C	D	E
Customer ID	Age	Gender	Item Purchase Category	Purchase Price
1	840	48	Male	Clothing
2	1718	29	Male	Clothing
3	322	41	Male	Footwear
4	3188	67	Female	Clothing
5	2270	49	Male	Skirt
6	367	64	Male	Clothing
7	2645	25	Male	Pants
8	1750	43	Male	Hoodie
9	3606	68	Female	Jeans
10	1097	57	Male	Coat
11	2601	21	Male	Belt
12	1660	53	Male	Accessories
13	3888	40	Female	Shirt
14	3498	35	Female	Handbag
15	3653	27	Female	Handbag
16	1537	18	Male	Sunglasses
17	2737	44	Female	Clothing
18	15	64	Male	Coat
19	1026	60	Male	Clothing
20	3255	21	Female	Sweater
21	2182	22	Male	Handbag
22	433	47	Male	Shirt
23	2010	37	Male	Footwear



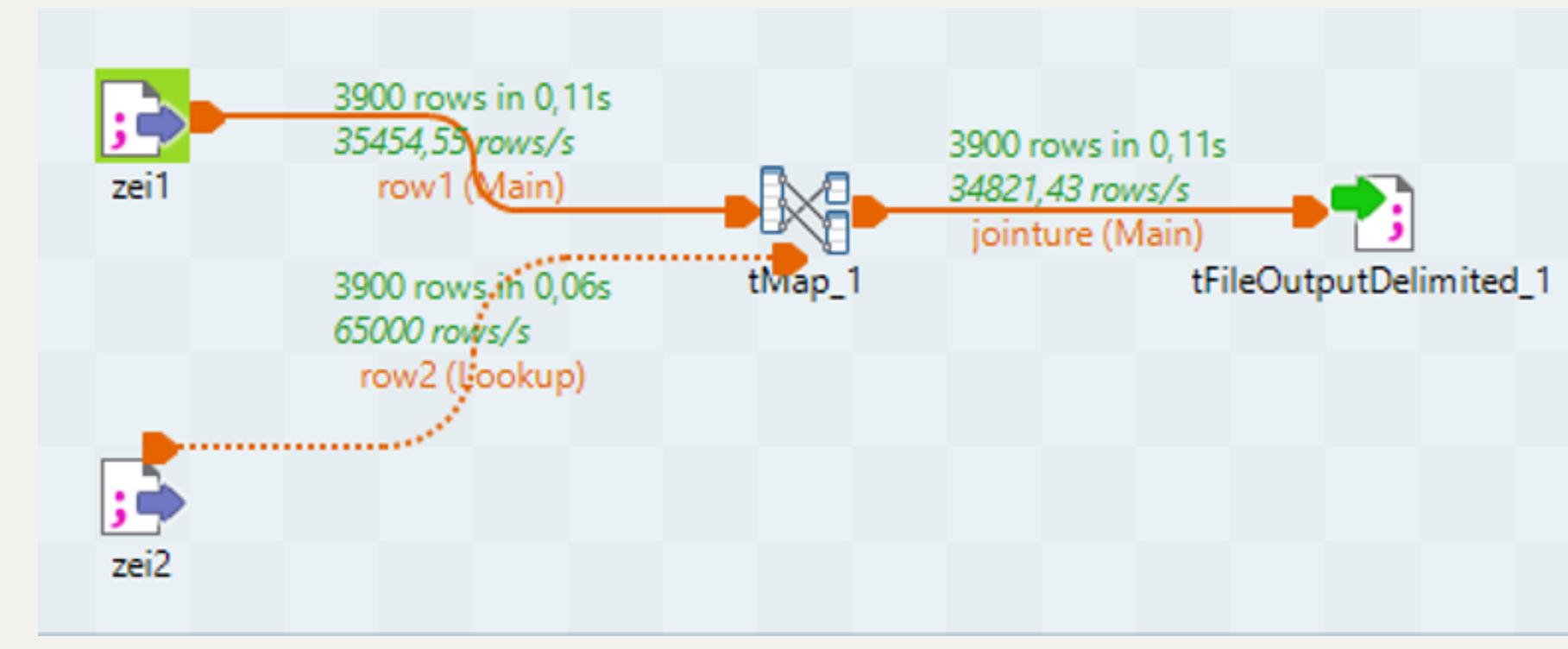
Data
Preparation
(ETL process)

Python was used to eliminate duplicates and refine a dataset, reducing the number of rows from 7,711 to 3,900 unique entries. Its flexibility and effectiveness make it a suitable tool for data manipulation tasks.

```
[ 7342 1339;56;Male;Handbag;Accessories  
 7402 1909;18;Male;Scarf;Accessories  
 7421 1509;35;Male;Blouse;Clothing  
 7428 142;56;Male;Gloves;Accessories  
 7711 1414;51;Male;Blouse;Clothing  
 [3900 rows x 1 columns]
```

```
import pandas as pd  
df = pd.read_csv('shopping_part_1.csv')  
df_no_duplicates = df.drop_duplicates()  
print(df_no_duplicates)  
df_no_duplicates.to_csv('shopping_part_1_no_duplicates.csv', index=False)
```

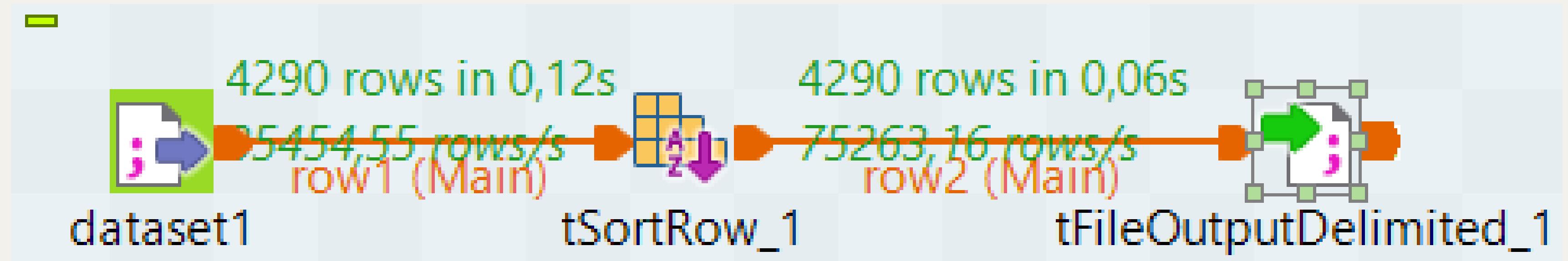
Next, we employ Talend's TMap to merge columns from two CSV files.



This screenshot provides a detailed look at the 'jointure' tab of the tMap configuration. It lists the columns from both 'row1' and 'row2' datasets, mapping them to their corresponding columns in the resulting output. The 'row1' dataset includes columns like Customer_ID, Age, Gender, Item_Purchased, Category, Purchase_Amount_USD, Location, Size, Color, Season, Review_Rating, and priceid. The 'row2' dataset includes columns like Subscription_Status, Shipping_Type, Discount_Applied, Promo_Code_Used, Previous_Purchases, Payment_Method, and Frequency_of_Purchases. The 'jointure' section maps these columns together.

Expression	Column
row1.Customer_ID	Customer_ID
row1.Age	Age
row1.Gender	Gender
row1.Item_Purchased	Item_Purchased
row1.Category	Category
row1.Purchase_Amount_USD	Purchase_Amount_USD
row1.Location	Location
row1.Size	Size
row1.Color	Color
row1.Season	Season
row1.Review_Rating	Review_Rating
row1.priceid	priceid
row2.Subscription_Status	Subscription_Status
row2.Shipping_Type	Shipping_Type
row2.Discount_Applied	Discount_Applied
row2.Promo_Code_Used	Promo_Code_Used
row2.Previous_Purchases	Previous_Purchases
row2.Payment_Method	Payment_Method
row2.Frequency_of_Purchases	Frequency_of_Purchases

jointure	
Expression	Column
row1.Customer_ID	Customer_ID
row1.Age	Age
row1.Gender	Gender
row1.Item_Purchased	Item_Purchased
row1.Category	Category
row1.Purchase_Amount_USD	Purchase_Amount_USD
row1.Location	Location
row1.Size	Size
row1.Color	Color
row1.Season	Season
row1.Review_Rating	Review_Rating
row1.priceid	priceid
row2.Subscription_Status	Subscription_Status
row2.Shipping_Type	Shipping_Type
row2.Discount_Applied	Discount_Applied
row2.Promo_Code_Used	Promo_Code_Used
row2.Previous_Purchases	Previous_Purchases
row2.Payment_Method	Payment_Method
row2.Frequency_of_Purchases	Frequency_of_Purchases



Colonne du schéma	tri num ou alpha ?	Ordre asc ou desc ?
Customer_ID	num	asc

Démarrage du Job projet à 22:32 18/01/2024.
[statistics] connecting to socket on port 3629
[statistics] connected
[statistics] disconnected

Job projet terminé à 22:32 18/01/2024. [Code de sortie = 0]

Loading sorted data into PostgreSQL

Python code :

```
1 import pandas as pd
2 from sqlalchemy import create_engine
3
4 db_user = 'postgres'
5 db_password = '1210zeineb'
6 db_host = 'localhost'
7 db_port = '5432'
8 db_name = 'projetBI'
9 csv_file_path = 'C:/Users/user/Desktop/mini projet/python/elezem f lezem/sortt.csv'
10 df = pd.read_csv(csv_file_path)
11 engine = create_engine(f'postgresql://{{db_user}}:{{db_password}}@{{db_host}}:{{db_port}}/{{db_name}}')
12 df.to_sql('final_table', engine, index=False, if_exists='replace')
13 print(f'Data from CSV file has been inserted into the PostgreSQL database.')
```

Result in POSTGRESQL:

```
1 SELECT * FROM public.final_table
2
```

	Customer_ID;Age;Gender;Item_Purchased;Category;Purchase_Amount_text
1	1;55;Male;Blouse;Clothing;53;Kentucky;L;Gray;Winter;3.1;774;Yes;Express;Yes;Yes;14;Debit Card;Bi-Weekly
2	2;19;Male;Sweater;Clothing;64;Maine;L;Maroon;Winter;3.1;3636;No;Free Shipping;No;No;17;Credit Card;Fortnightly
3	3;50;Male;Jeans;Clothing;73;Massachusetts;S;Maroon;Spring;3.1;2277;No;2-Day Shipping;No;No;25;Debit Card;Weekly
4	4;21;Male;Sandals;Footwear;90;Rhode Island;M;Maroon;Spring;3.5;2793;No;Express;No;No;11;Credit Card;Weekly
5	5;45;Male;Blouse;Clothing;49;Oregon;M;Turquoise;Spring;2.7;3373;No;Free Shipping;No;No;14;Credit Card;Every 3 Months
6	6;46;Male;Sneakers;Footwear;20;Wyoming;M;White;Summer;2.9;2797;No;Store Pickup;No;No;21;Cash;Annually
7	7;63;Male;Shirt;Clothing;85;Montana;M;Gray;Fall;3.2;503;Yes;Express;Yes;Yes;8;PayPal;Fortnightly

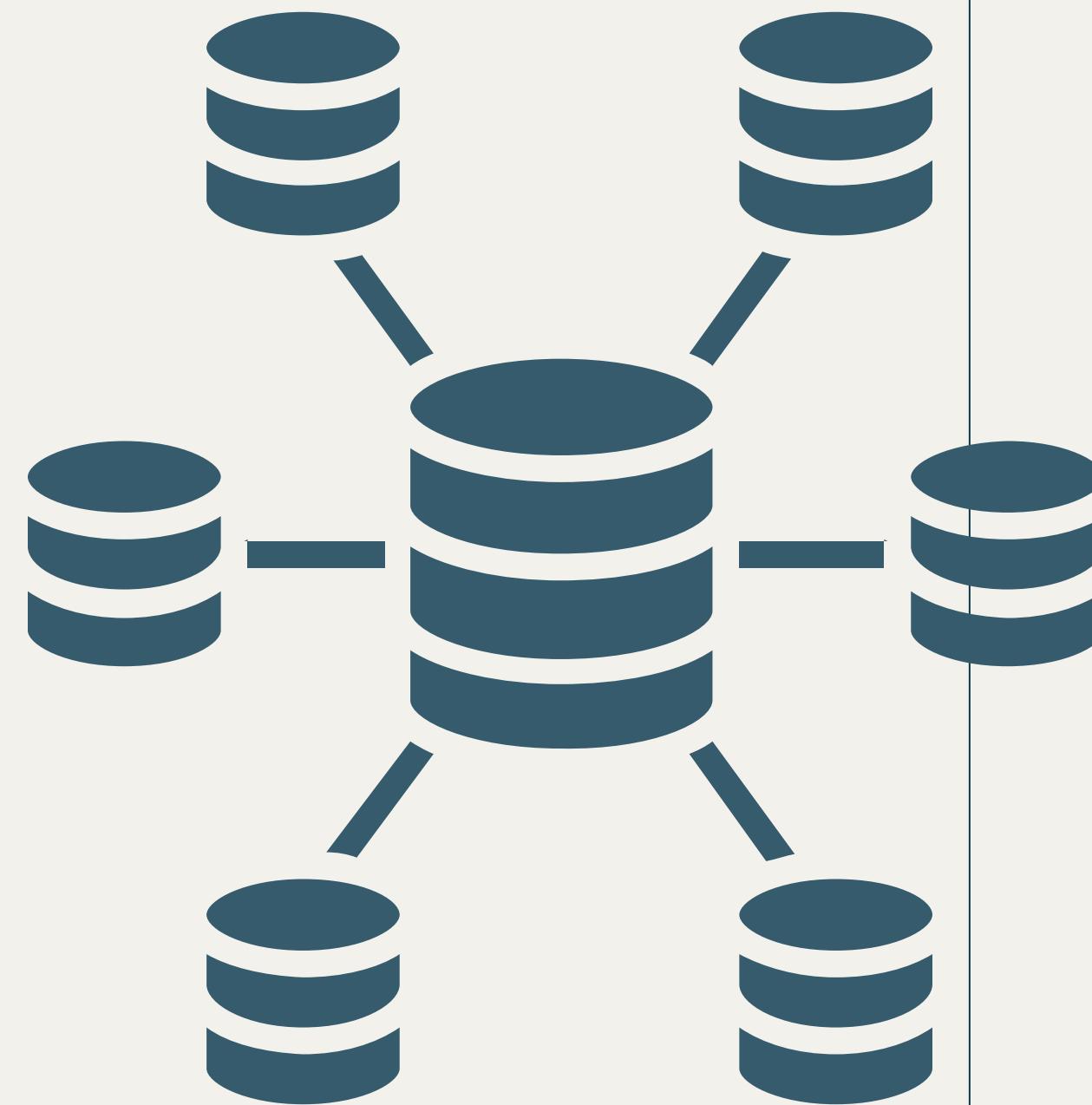
Total rows: 1000 of 3900 | Query complete 00:00:00.273

Python output :

```
>>> %Run clean.py
```

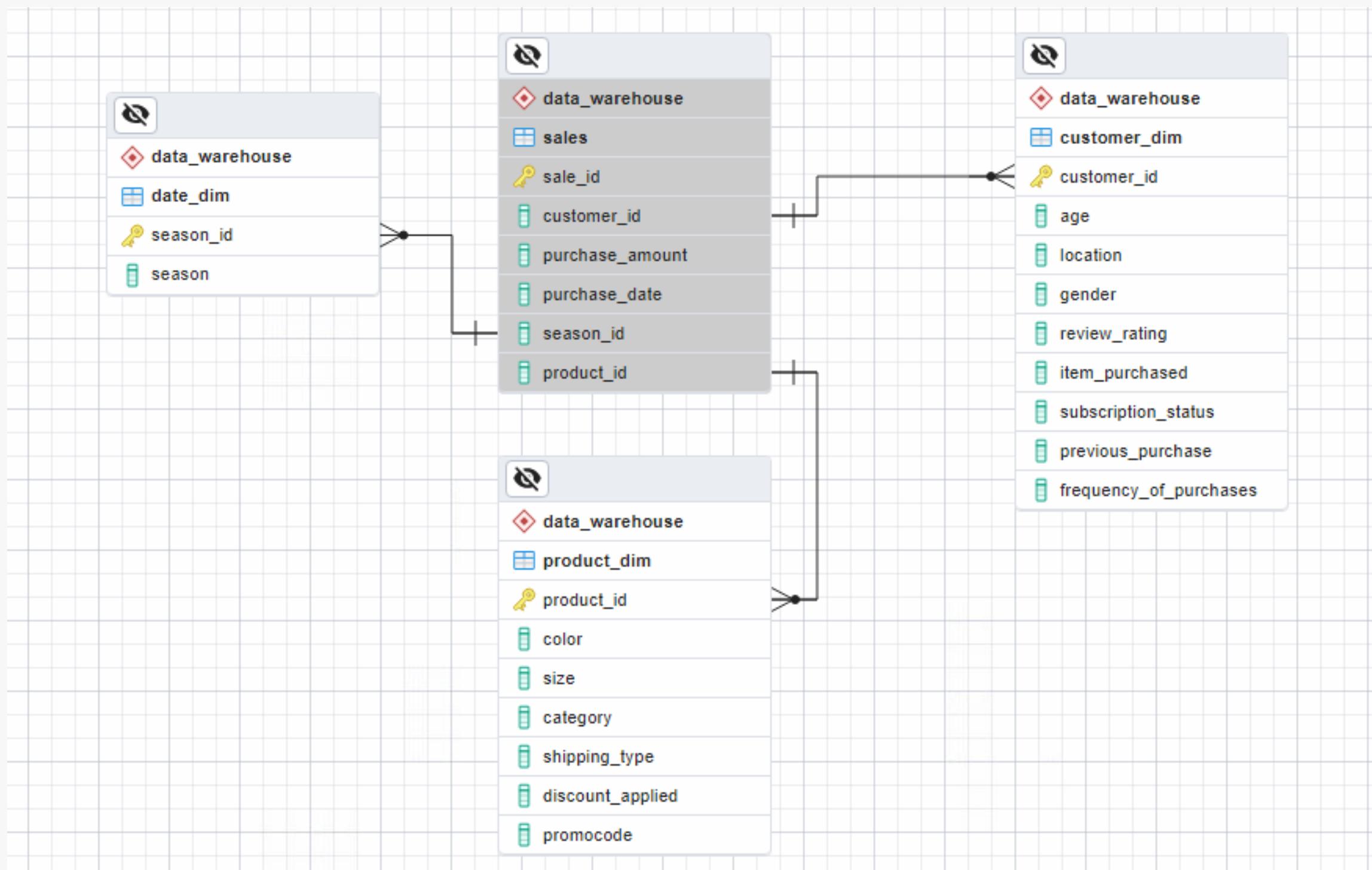
```
Data from CSV file has been inserted into the PostgreSQL database.
```

```
>>>
```



Data Storage (Modeling
and DWH creation)

STAR SCHEMA :



Creation of Fact and Dimensions tables

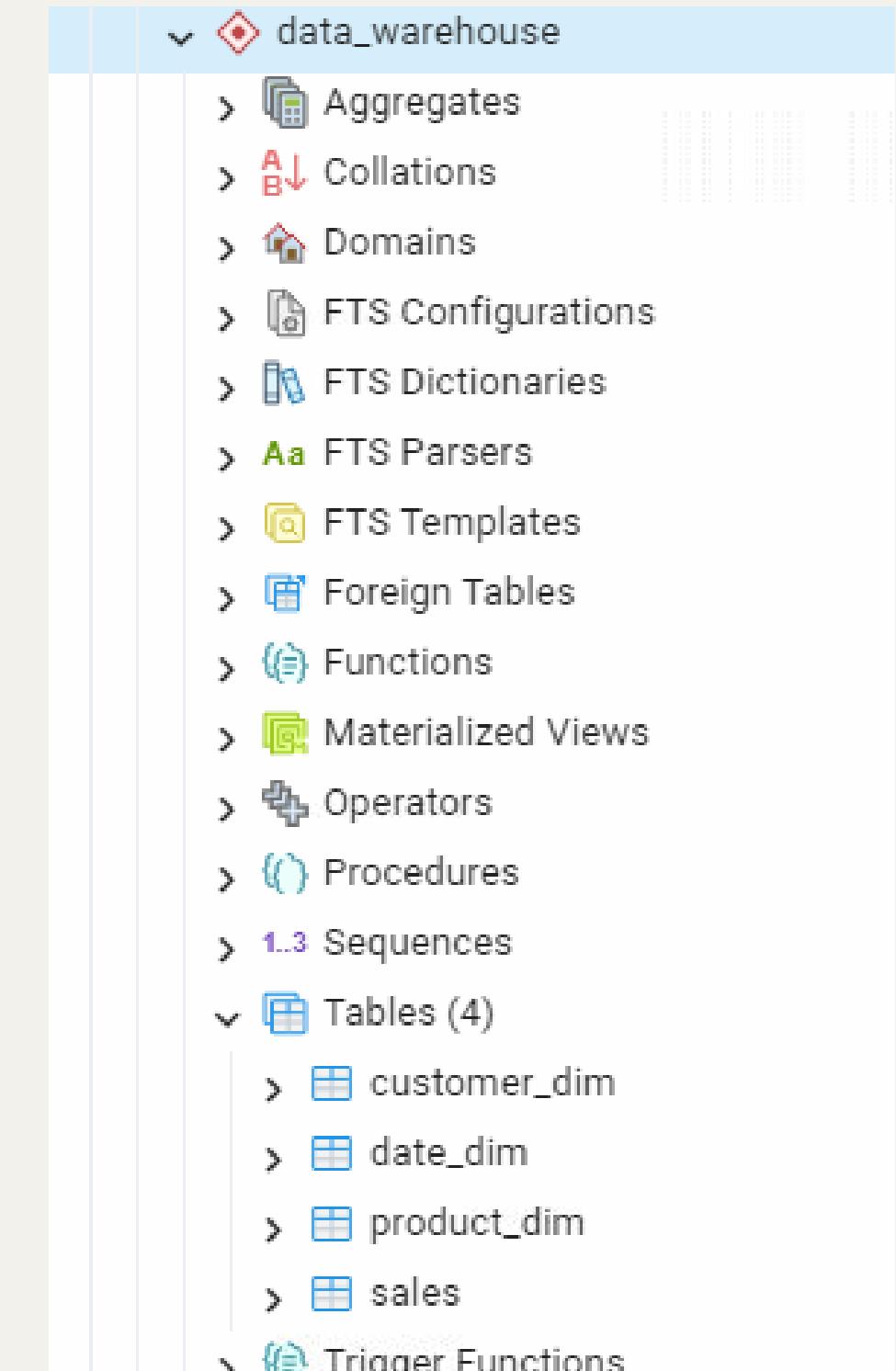
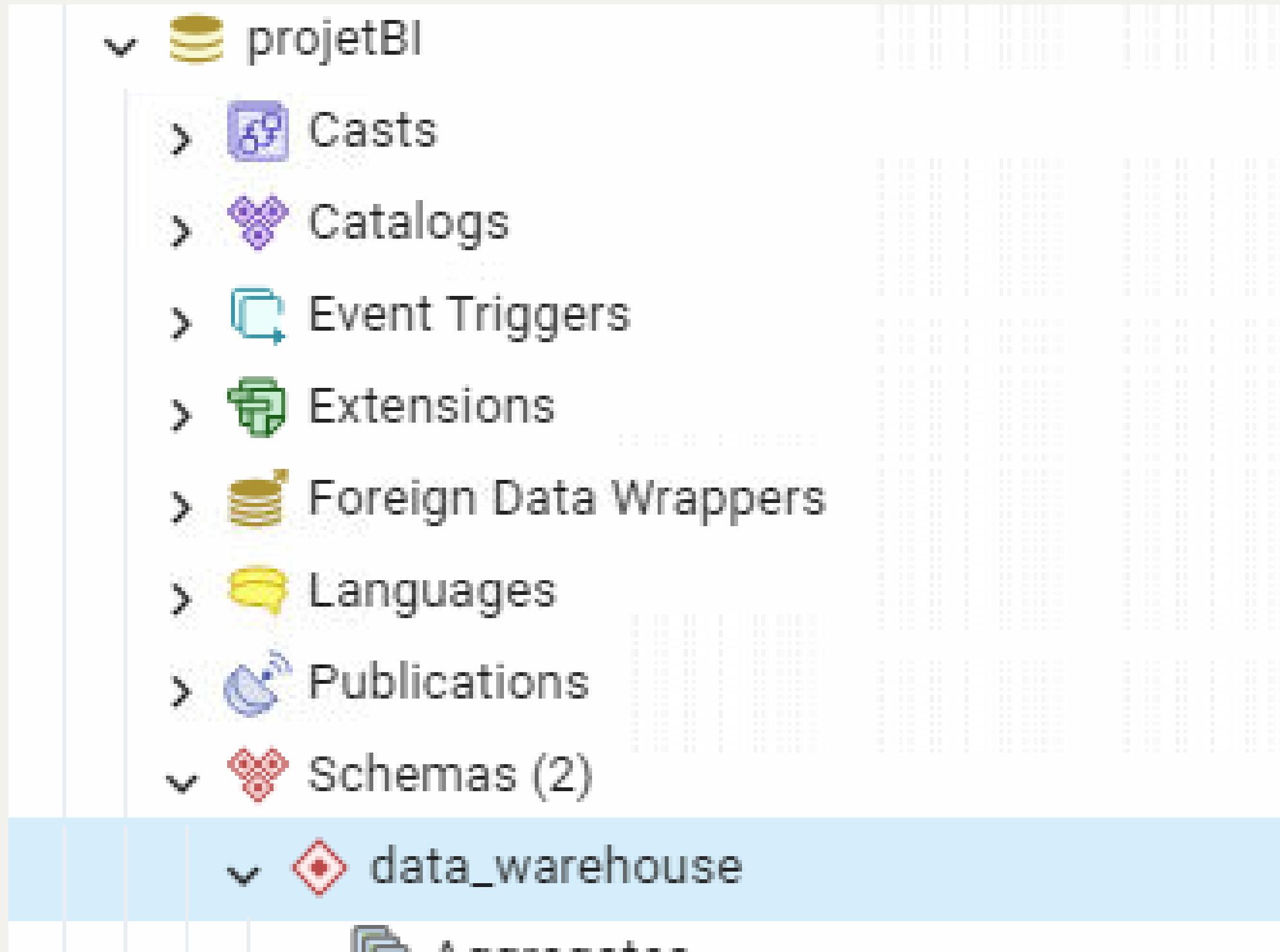
```
import psycopg2
from sqlalchemy import create_engine
db_user = 'postgres'
db_password = '1210zeineb'
db_host = 'localhost'
db_port = '5432'
db_name = 'projetBI'
engine = create_engine(f'postgresql://{{db_user}}:{{db_password}}@{{db_host}}:{{db_port}}/{{db_name}}')
conn = psycopg2.connect(
    user=db_user,
    password=db_password,
    host=db_host,
    port=db_port,
    database=db_name
)
cursor = conn.cursor()
create_schema_query = "CREATE SCHEMA IF NOT EXISTS data_warehouse;"
cursor.execute(create_schema_query)
create_customer_dim_query = """
CREATE TABLE IF NOT EXISTS data_warehouse.customer_dim (
    customer_id SERIAL PRIMARY KEY,
    Age INTEGER,
    Location VARCHAR(255),
    Gender VARCHAR(20),
    Review_Rating INTEGER,
    Item_Purchased VARCHAR(50),
    Subscription_Status VARCHAR(50),
    Previous_purchase VARCHAR(50),
    Frequency_of_purchases VARCHAR(50)
);
"""
cursor.execute(create_customer_dim_query)
create_product_dim_query = """
```

```
34 CREATE TABLE IF NOT EXISTS data_warehouse.product_dim (
35     product_id SERIAL PRIMARY KEY,
36     Color VARCHAR(255),
37     Size VARCHAR(50),
38     Category VARCHAR(60),
39     Shipping_type VARCHAR(50),
40     Discount_applied VARCHAR(50),
41     Promocode VARCHAR(50)
42 );
43 """
44 cursor.execute(create_product_dim_query)
45 create_date_dim_query = """
46 CREATE TABLE IF NOT EXISTS data_warehouse.date_dim (
47     Season_id SERIAL PRIMARY KEY,
48     Season VARCHAR(30)
49 );
45 """
50 cursor.execute(create_date_dim_query)
51 create_table_query = """
52 CREATE TABLE IF NOT EXISTS data_warehouse.sales (
53     sale_id SERIAL PRIMARY KEY,
54     customer_id INTEGER REFERENCES data_warehouse.customer_dim(customer_id),
55     product_id INTEGER REFERENCES data_warehouse.product_dim(product_id),
56     season_id INTEGER REFERENCES data_warehouse.date_dim(season_id),
57     purchase_amount DECIMAL(10, 2)
58 );
59 """
60 cursor.execute(create_table_query)
61 conn.commit()
62 cursor.close()
63 conn.close()
```

```
>>> %Run clean.py
Data warehouse schema and tables created.

>>>
```

POSTGRESQL :



DATA WAREHOUSE IMPLEMENTATION:

Step 1 :

The screenshot shows a PostgreSQL database management interface. On the left, a sidebar menu is open under the 'Tables' section, which contains four items: 'custom', 'date_dim', 'product', and 'sales'. The 'product' item is currently selected. A context menu is displayed over the 'product' item, listing options like 'Count Rows', 'Create', 'Delete', 'Delete (Cascade)', 'Refresh...', 'Restore...', 'Backup...', 'size', 'color', 'shipping_type', 'discount_applied', 'promo_code_used', 'Import/Export Data...', 'Reset Statistics', 'ERD For Table', 'Maintenance...', 'Scripts', 'Truncate', 'View/Edit Data', 'Rules', 'Search Objects...', and 'Query Tool'. The 'Import/Export Data...' option is highlighted with a mouse cursor.

Step 2 :

The screenshot shows the 'Import/Export data - table 'product_dim'' dialog. The 'General' tab is selected. The 'Import/Export' section has a checked 'Import' button. The 'Filename' field is set to 'D:\Hajji\productid.csv', the 'Format' field is set to 'csv', and the 'Encoding' field is set to 'Select an item...'. Below the dialog, a message indicates 'Total rows: 1000 of 3900' and 'Query complete 00:00:00.224'.

Result :

The screenshot shows a PostgreSQL query tool window. The query is: 'SELECT * FROM data_warehouse.product_dim ORDER BY product_id ASC'. The results are displayed in a table with the following data:

product_id	size	color	shipping_type	discount_applied	promo_code_used
1	L	Gray	Express	Yes	Yes
2	L	Maroon	Free Shipping	No	No
3	S	Maroon	2-Day Shipping	No	No
4	M	Maroon	Express	No	No
5	M	Turquoise	Free Shipping	No	No
6	M	White	Store Pickup	No	No
7	M	Gray	Express	Yes	Yes
8	L	Charcoal	Express	Yes	Yes
9	L	Silver	Standard	Yes	Yes
10	M	Pink	Express	No	No

Data Analysis

Dashboard created using POWER BI



Power BI

**Thank
You**