FINAL PROJECT

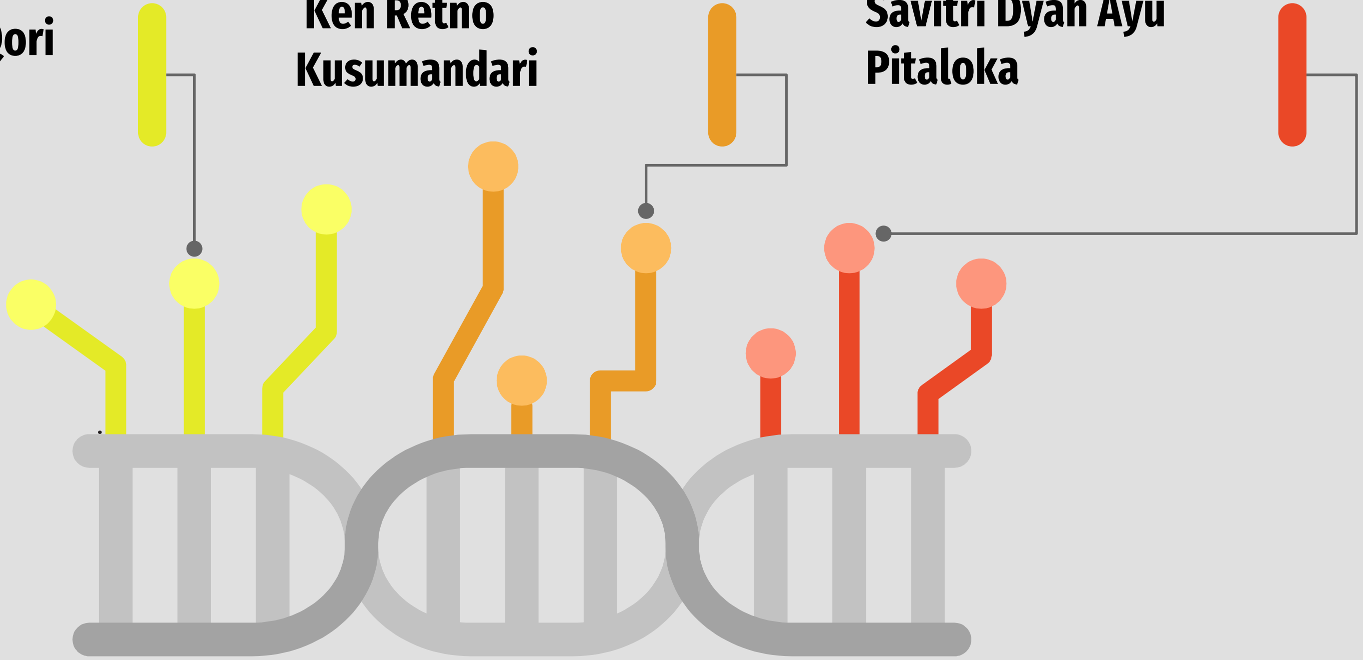
Kelompok 6
Fast Track Data Engineer
Digital Skola

Nama Anggota

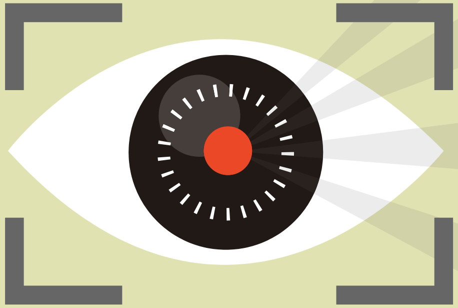
Ilham Abdi Al Qori

Ken Retno Kusumandari

Savitri Dyah Ayu Pitaloka

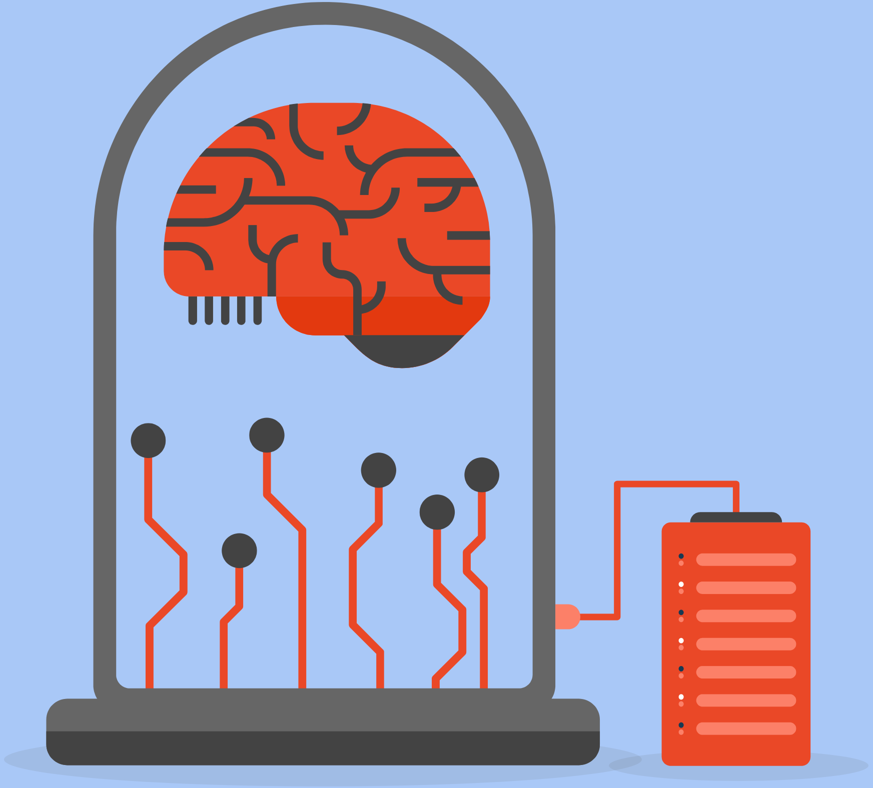


Outline



- **Project 1** — Python and Business Intelligence
- **Project 2** — Integration Batch Projection with DBT
- **Project 3** — Big Data Processing
- **Project 4** — Real-Time Processing

PROJECT 1

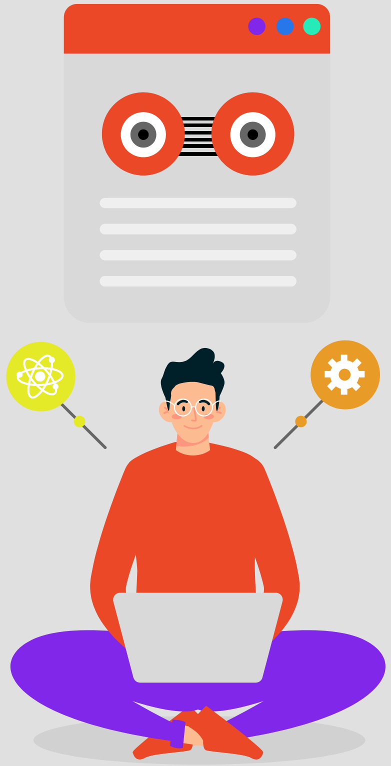


Objective of The Project

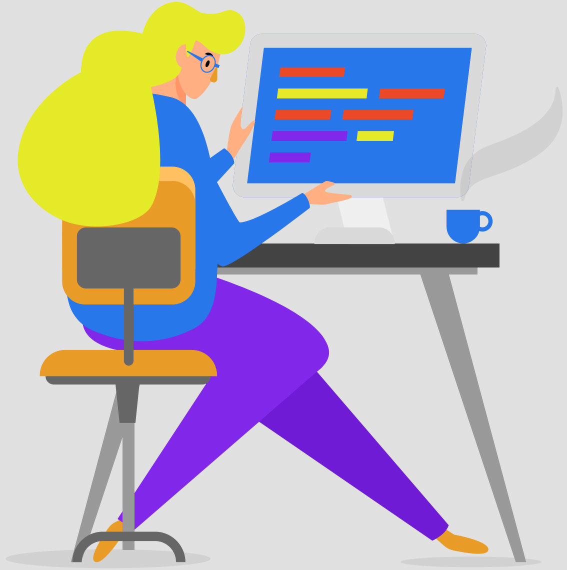
Creating a Business Intelligence Report with Python and PostgreSQL

In this project, our goal is to gain practical insights into business intelligence (BI) by creating a comprehensive report. Specifically, we aim to achieve the following: Database Connection and Exploration Datamart Creation Visualization with Google Data Studio

python™ PostgreSQL DBeaver Looker Studio



Pipeline



01 Create Connection in Dbeaver
Postgre Connection for OLTP and DWH

02 Understand Business Requirements
Carry out to determine the DWH design

03 Create Relational Data Warehouse Design
In ERD form

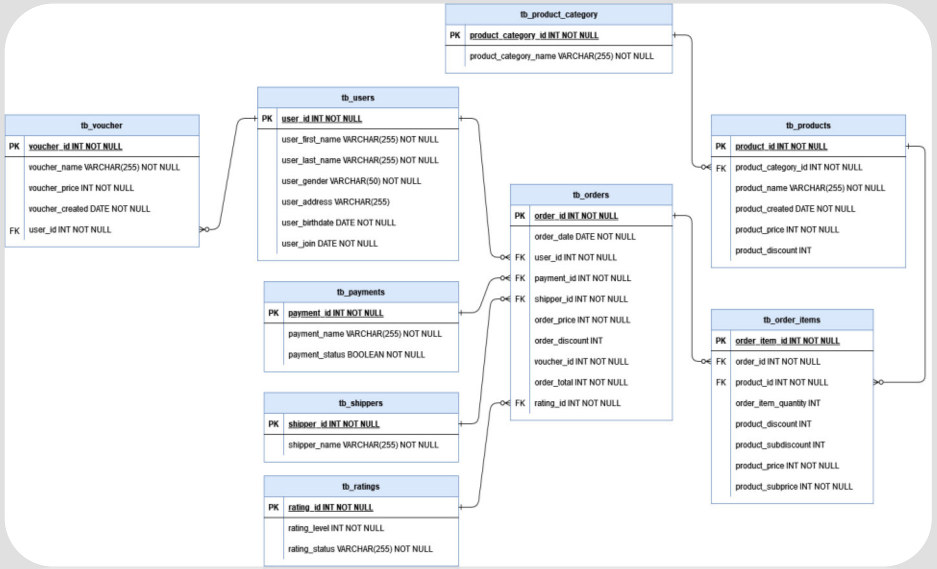
04 Create DDL Data Warehouse and Data Mart
Based on Business Requirements

05 Create ETL Script
Migrate data from OLTP to DWH

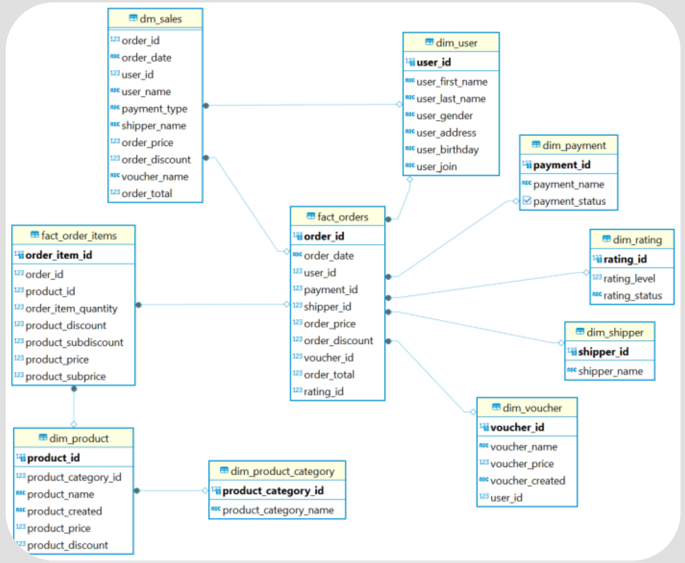
06 Design the Dashboard
Customize Business Requirements and create in Looker Studio

ERD OLTP and DWH

OLTP

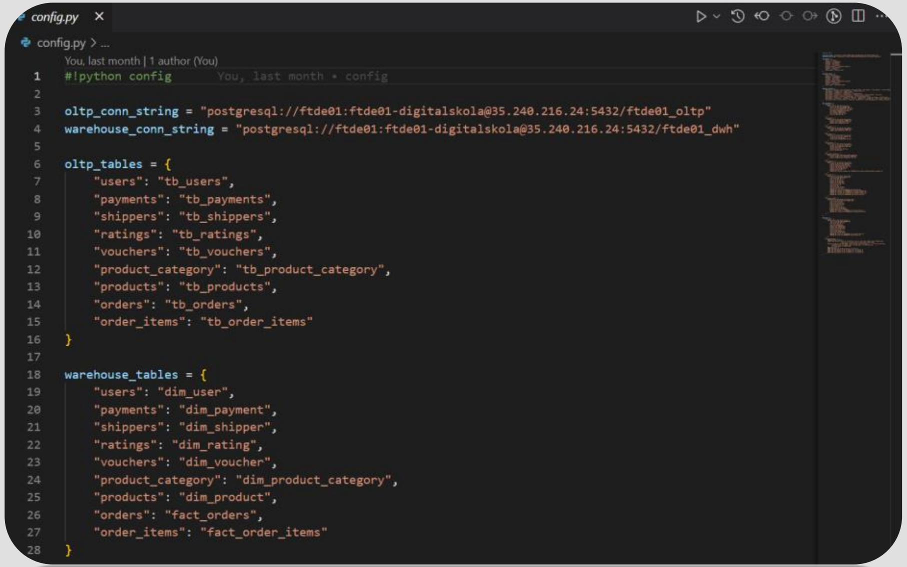

tb_product_category
PK product_category_id INT NOT NULL
product_category_name VARCHAR(255) NOT NULL
tb_users
PK user_id INT NOT NULL
user_first_name VARCHAR(255) NOT NULL
user_last_name VARCHAR(255) NOT NULL
user_gender VARCHAR(50) NOT NULL
user_address VARCHAR(255)
user_birthdate DATE NOT NULL
user_join DATE NOT NULL
tb_voucher
PK voucher_id INT NOT NULL
voucher_name VARCHAR(255) NOT NULL
voucher_price INT NOT NULL
voucher_created DATE NOT NULL
FK user_id INT NOT NULL
tb_products
PK product_id INT NOT NULL
FK product_category_id INT NOT NULL
product_name VARCHAR(255) NOT NULL
product_created DATE NOT NULL
product_price INT NOT NULL
product_discount INT
tb_orders
PK order_id INT NOT NULL
order_date DATE NOT NULL
FK user_id INT NOT NULL
FK payment_id INT NOT NULL
FK shipper_id INT NOT NULL
order_price INT NOT NULL
order_discount INT
voucher_id INT NOT NULL
order_total INT NOT NULL
FK rating_id INT NOT NULL
tb_payments
PK payment_id INT NOT NULL
payment_name VARCHAR(255) NOT NULL
payment_status BOOLEAN NOT NULL
tb_order_items
PK order_item_id INT NOT NULL
FK order_id INT NOT NULL
FK product_id INT NOT NULL
order_item_quantity INT
product_discount INT
product_subdiscount INT
product_price INT NOT NULL
product_subprice INT NOT NULL
tb_shippers
PK shipper_id INT NOT NULL
shipper_name VARCHAR(255) NOT NULL
tb_ratings
PK rating_id INT NOT NULL
rating_level INT NOT NULL
rating_status VARCHAR(255) NOT NULL

DWH and Data Mart
(Snowflake Schema)



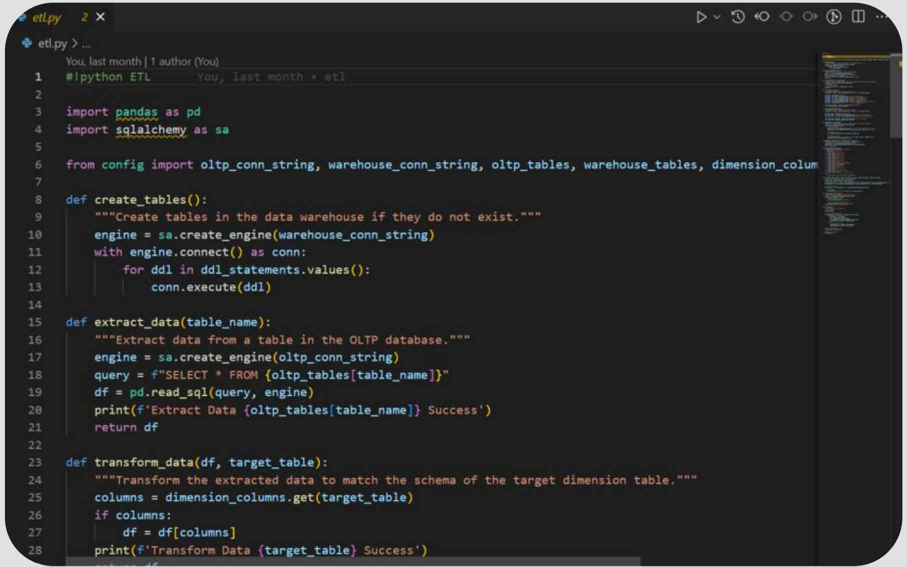
Config and ETL Scripts

Config



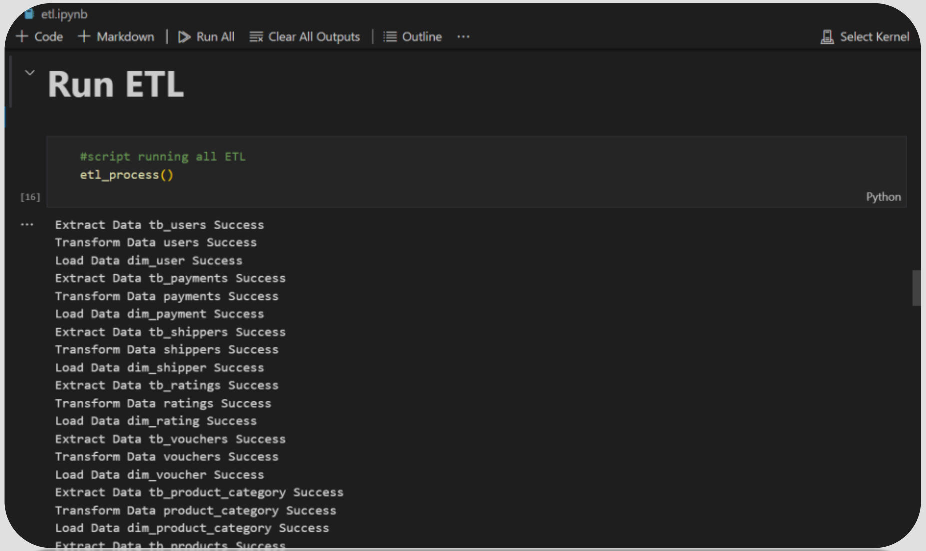
```python
#!python config

oltp_conn_string = "postgresql://ftde01:ftde01-digitalskola@35.240.216.24:5432/ftde01_oltp"
warehouse_conn_string = "postgresql://ftde01:ftde01-digitalskola@35.240.216.24:5432/ftde01_dwh"

oltp_tables = {
    "users": "tb_users",
    "payments": "tb_payments",
    "shippers": "tb_shippers",
    "ratings": "tb_ratings",
    "vouchers": "tb_vouchers",
    "product_category": "tb_product_category",
    "products": "tb_products",
    "orders": "tb_orders",
    "order_items": "tb_order_items"
}

warehouse_tables = {
    "users": "dim_user",
    "payments": "dim_payment",
    "shippers": "dim_shipper",
    "ratings": "dim_rating",
    "vouchers": "dim_voucher",
    "product_category": "dim_product_category",
    "products": "dim_product",
    "orders": "fact_orders",
    "order_items": "fact_order_items"
}
```

ETL



```python
#!python ETL

import pandas as pd
import sqlalchemy as sa

from config import oltp_conn_string, warehouse_conn_string, oltp_tables, warehouse_tables, dimension_colum

def create_tables():
    """Create tables in the data warehouse if they do not exist."""
    engine = sa.create_engine(warehouse_conn_string)
    with engine.connect() as conn:
        for ddl in ddl_statements.values():
            conn.execute(ddl)

def extract_data(table_name):
    """Extract data from a table in the OLTP database."""
    engine = sa.create_engine(oltp_conn_string)
    query = f"SELECT * FROM {oltp_tables[table_name]}"
    df = pd.read_sql(query, engine)
    print(f'Extract Data {oltp_tables[table_name]} Success')
    return df

def transform_data(df, target_table):
    """Transform the extracted data to match the schema of the target dimension table."""
    columns = dimension_columns.get(target_table)
    if columns:
        df = df[columns]
    print(f'Transform Data {target_table} Success')
```

ETL Success

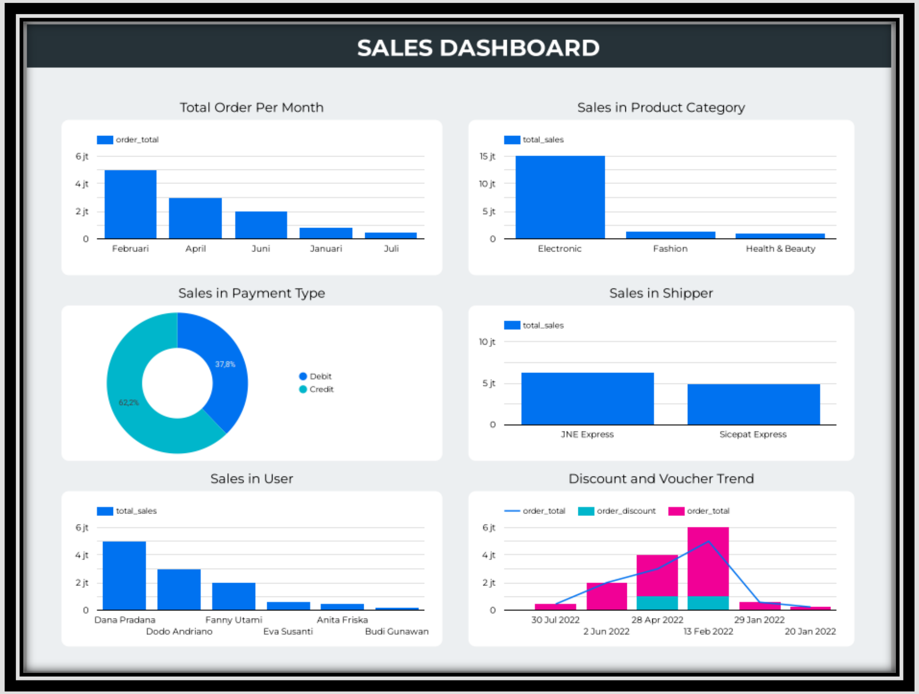


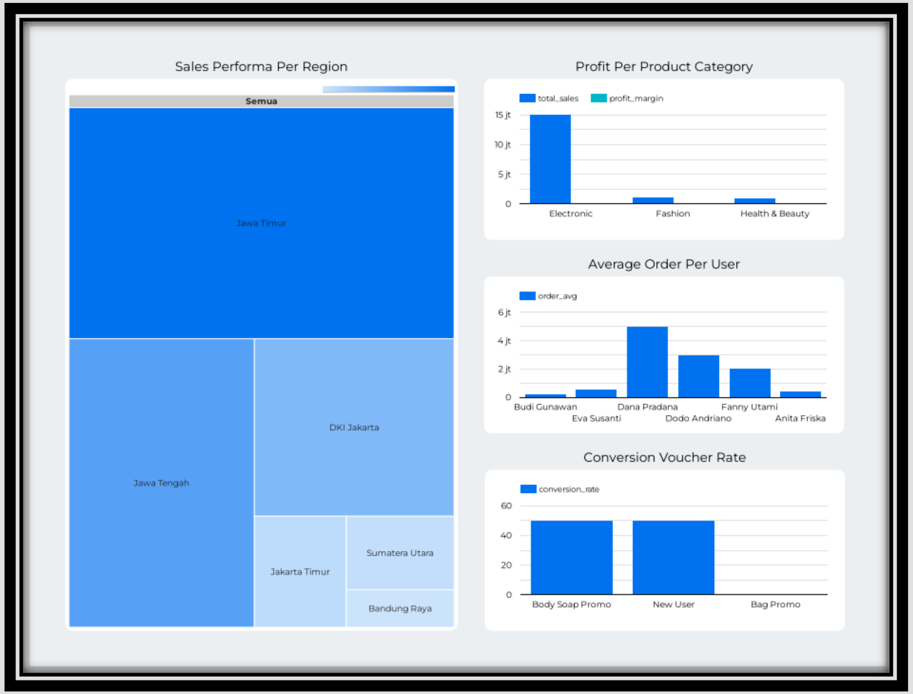
Run ETL

```python
#script running all ETL
etl_process()
```

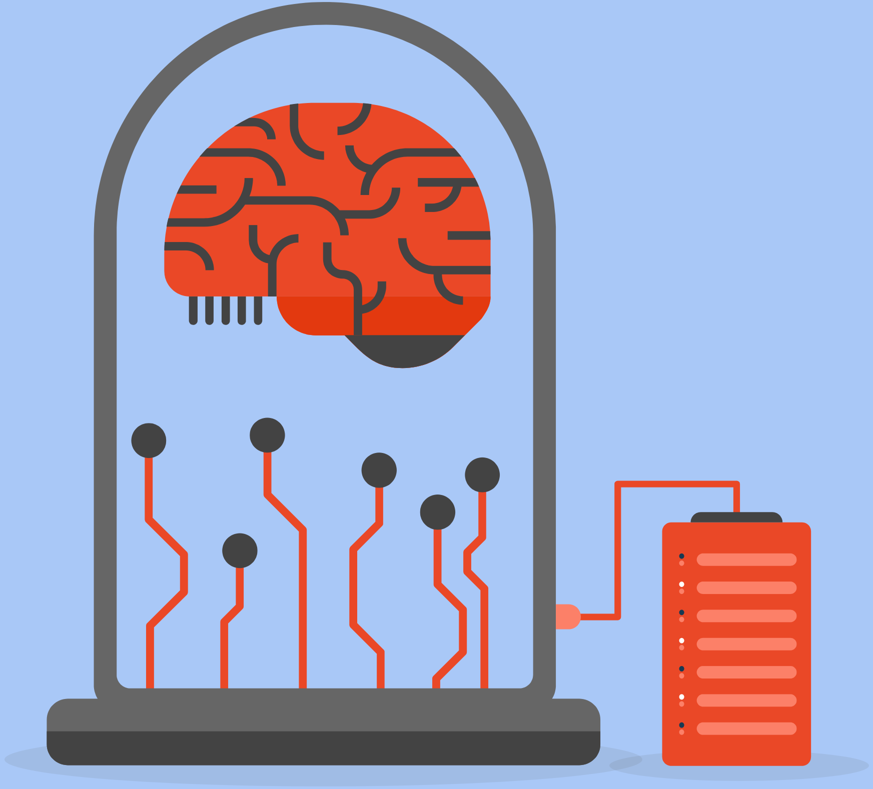
```
Extract Data tb_users Success
Transform Data users Success
Load Data dim_user Success
Extract Data tb_payments Success
Transform Data payments Success
Load Data dim_payment Success
Extract Data tb_shippers Success
Transform Data shippers Success
Load Data dim_shipper Success
Extract Data tb_ratings Success
Transform Data ratings Success
Load Data dim_rating Success
Extract Data tb_vouchers Success
Transform Data vouchers Success
Load Data dim_voucher Success
Extract Data tb_product_category Success
Transform Data product_category Success
Load Data dim_product_category Success
Extract Data tb_products Success
```

Dashboard


SALES DASHBOARD
Total Order Per Month
order_total
6 jt
4 jt
2 jt
0
Februari
April
Juni
Januari
Juli
Sales in Product Category
total_sales
15 jt
10 jt
5 jt
0
Electronic
Fashion
Health & Beauty
Sales in Payment Type
37,8%
62,2%
Debit
Credit
Sales in Shipper
total_sales
10 jt
5 jt
0
JNE Express
Sicepat Express
Sales in User
total_sales
6 jt
4 jt
2 jt
0
Dana Pradana
Dodo Andriano
Fanny Utami
Eva Susanti
Anita Friska
Budi Gunawan
Discount and Voucher Trend
order_total
order_discount
order_total
6 jt
4 jt
2 jt
0
30 Jul 2022
2 Jun 2022
28 Apr 2022
13 Feb 2022
29 Jan 2022
20 Jan 2022


Sales Performa Per Region
Semua
Jawa Timur
Jawa Tengah
DKI Jakarta
Jakarta Timur
Sumatera Utara
Bandung Raya
Profit Per Product Category
total_sales
profit_margin
15 jt
10 jt
5 jt
0
Electronic
Fashion
Health & Beauty
Average Order Per User
order_avg
6 jt
4 jt
2 jt
0
Budi Gunawan
Eva Susanti
Dana Pradana
Dodo Andriano
Fanny Utami
Anita Friska
Conversion Voucher Rate
conversion_rate
60
40
20
0
Body Soap Promo
New User
Bag Promo

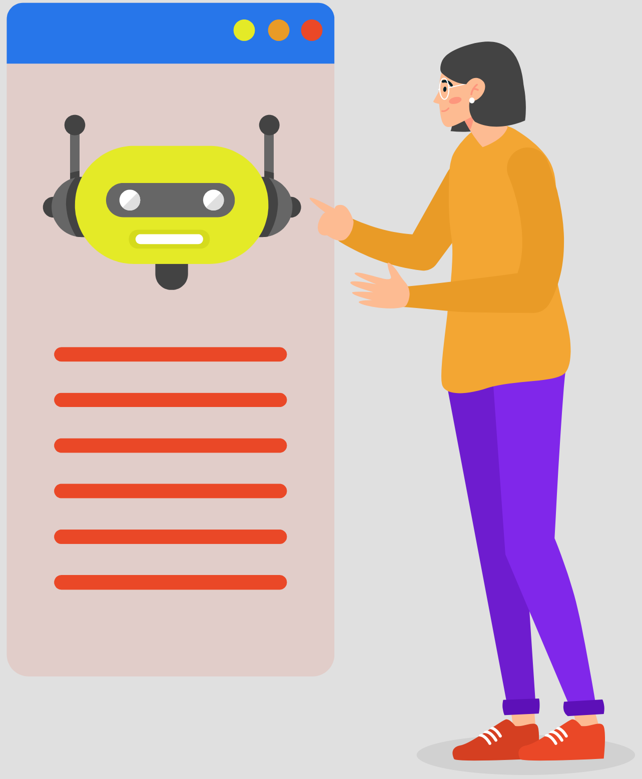
PROJECT 2



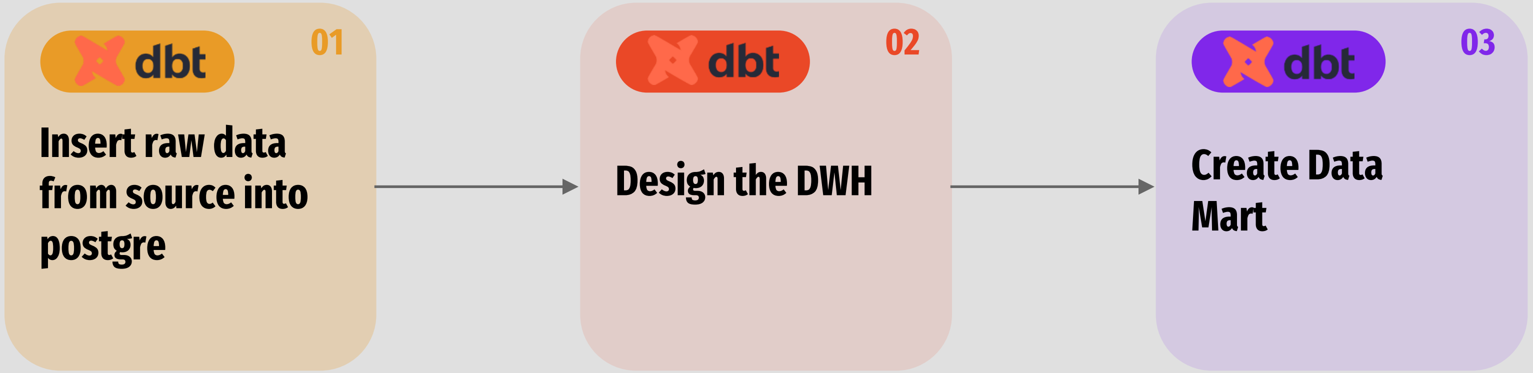
Objective of The Project

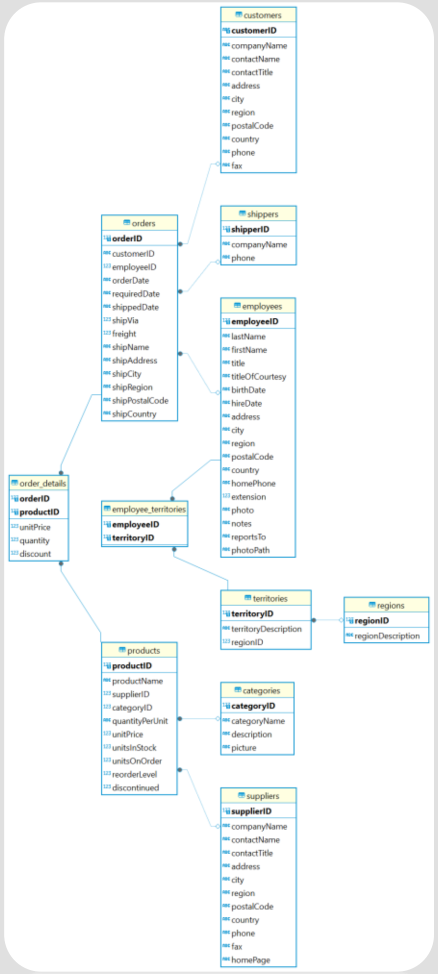
Leveraging dbt for Transforming Data in an ELT Workflow

In this project, we aim to achieve a deep understanding of dbt (Data Build Tool) as a powerful transformation tool within the Extract, Load, Transform (ELT) methodology. Specifically, our objectives include:Data ExtractionTransformation with dbtIntegration with PostgreSQL

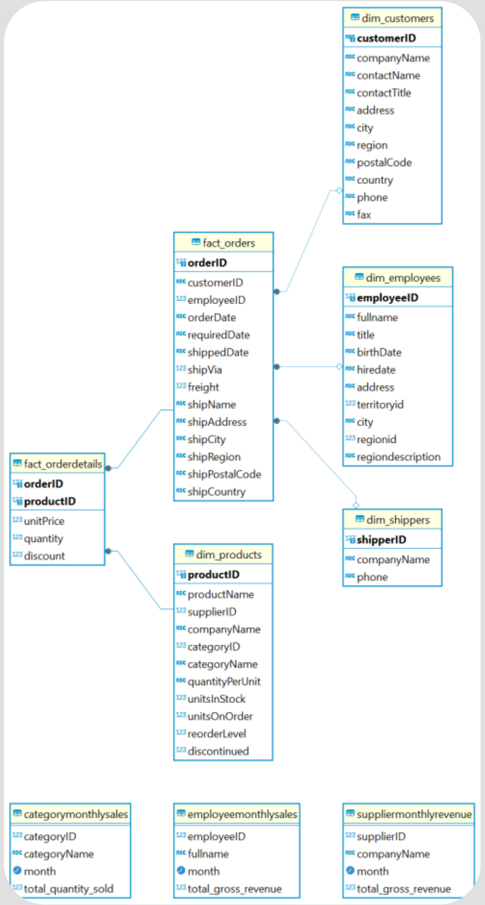


Pipeline


dbt
01
Insert raw data from source into postgre
dbt
02
Design the DWH
dbt
03
Create Data Mart

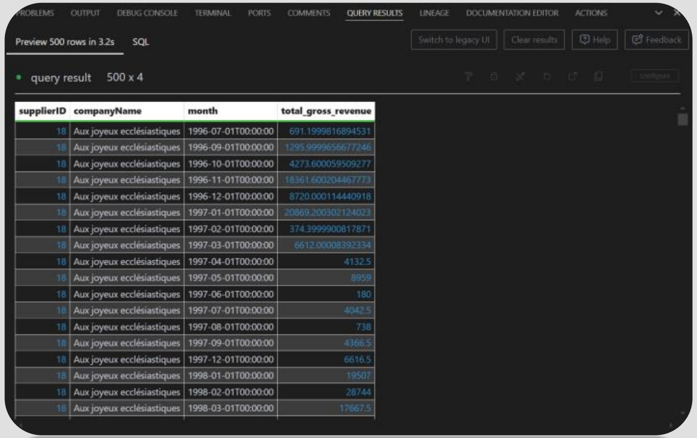

customers
customerID
companyName
contactName
contactTitle
address
city
region
postalCode
country
phone
fax
orders
orderID
customerID
employeeID
orderDate
requiredDate
shippedDate
shipVia
freight
shipName
shipAddress
shipCity
shipRegion
shipPostalCode
shipCountry
shippers
shipperID
companyName
phone
employees
employeeID
lastName
firstName
title
titleOfCourtesy
birthDate
hireDate
address
city
region
postalCode
country
homePhone
extension
photo
notes
reportsTo
photoPath
order_details
orderID
productID
unitPrice
quantity
discount
employee_territories
employeeID
territoryID
territories
territoryID
territoryDescription
regionID
regions
regionID
regionDescription
products
productID
productName
supplierID
categoryID
quantityPerUnit
unitPrice
unitsInStock
unitsOnOrder
reorderLevel
discontinued
categories
categoryID
categoryName
description
picture
suppliers
supplierID
companyName
contactName
contactTitle
address
city
region
postalCode
country
phone
fax
homePage

OLTP


dim_customers
customerID
companyName
contactName
contactTitle
address
city
region
postalCode
country
phone
fax
fact_orders
orderID
customerID
employeeID
orderDate
requiredDate
shippedDate
shipVia
freight
shipName
shipAddress
shipCity
shipRegion
shipPostalCode
shipCountry
dim_employees
employeeID
fullname
title
birthDate
hiredate
address
territoryid
city
regionid
regiondescription
fact_orderdetails
orderID
productID
unitPrice
quantity
discount
dim_shippers
shipperID
companyName
phone
dim_products
productID
productName
supplierID
companyName
categoryID
categoryName
quantityPerUnit
unitsInStock
unitsOnOrder
reorderLevel
discontinued
categorymonthlysales
categoryID
categoryName
month
total_quantity_sold
employeemonthlysales
employeeID
fullname
month
total_gross_revenue
suppliermonthlyrevenue
supplierID
companyName
month
total_gross_revenue

DWH

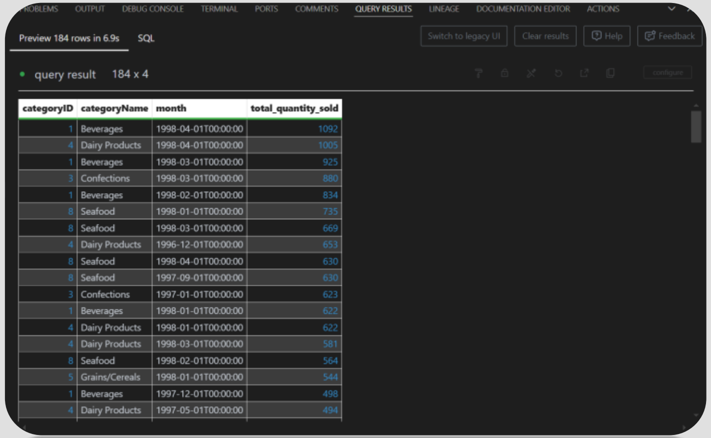
Data Mart


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS QUERY RESULTS LINEAGE DOCUMENTATION EDITOR ACTIONS
Switch to legacy UI
Clear results
Help
Feedback
configure

Preview 500 rows in 3.2s   SQL

query result   500 x 4

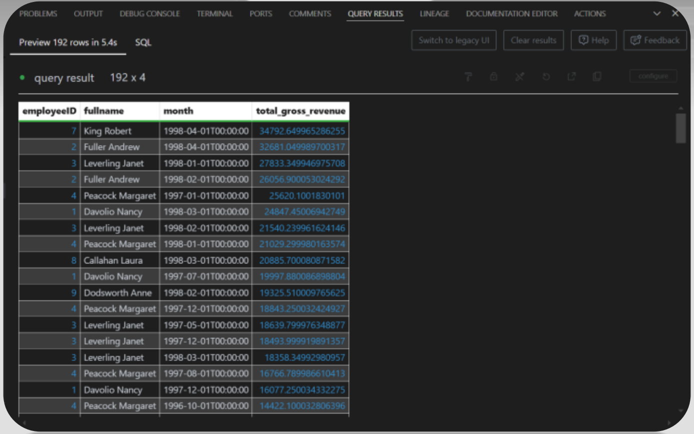
supplierID	companyName	month	total_gross_revenue
18	Aux joyeux ecclésiastiques	1996-07-01T00:00:00	691.1999816894531
18	Aux joyeux ecclésiastiques	1996-09-01T00:00:00	1295.9999656677246
18	Aux joyeux ecclésiastiques	1996-10-01T00:00:00	4273.600059509277
18	Aux joyeux ecclésiastiques	1996-11-01T00:00:00	18361.600204467773
18	Aux joyeux ecclésiastiques	1996-12-01T00:00:00	8720.000114440918
18	Aux joyeux ecclésiastiques	1997-01-01T00:00:00	20869.200302124023
18	Aux joyeux ecclésiastiques	1997-02-01T00:00:00	374.3999900817871
18	Aux joyeux ecclésiastiques	1997-03-01T00:00:00	6612.00008392334
18	Aux joyeux ecclésiastiques	1997-04-01T00:00:00	4132.5
18	Aux joyeux ecclésiastiques	1997-05-01T00:00:00	8959
18	Aux joyeux ecclésiastiques	1997-06-01T00:00:00	180
18	Aux joyeux ecclésiastiques	1997-07-01T00:00:00	4042.5
18	Aux joyeux ecclésiastiques	1997-08-01T00:00:00	738
18	Aux joyeux ecclésiastiques	1997-09-01T00:00:00	4366.5
18	Aux joyeux ecclésiastiques	1997-12-01T00:00:00	6616.5
18	Aux joyeux ecclésiastiques	1998-01-01T00:00:00	19507
18	Aux joyeux ecclésiastiques	1998-02-01T00:00:00	28744
18	Aux joyeux ecclésiastiques	1998-03-01T00:00:00	17667.5


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS QUERY RESULTS LINEAGE DOCUMENTATION EDITOR ACTIONS
Switch to legacy UI
Clear results
Help
Feedback
configure

Preview 184 rows in 6.9s   SQL

query result   184 x 4

categoryID	categoryName	month	total_quantity_sold
1	Beverages	1998-04-01T00:00:00	1092
4	Dairy Products	1998-04-01T00:00:00	1005
1	Beverages	1998-03-01T00:00:00	925
3	Confections	1998-03-01T00:00:00	880
1	Beverages	1998-02-01T00:00:00	834
8	Seafood	1998-01-01T00:00:00	735
8	Seafood	1998-03-01T00:00:00	669
4	Dairy Products	1996-12-01T00:00:00	653
8	Seafood	1998-04-01T00:00:00	630
8	Seafood	1997-09-01T00:00:00	630
3	Confections	1997-01-01T00:00:00	623
1	Beverages	1998-01-01T00:00:00	622
4	Dairy Products	1998-01-01T00:00:00	622
4	Dairy Products	1998-03-01T00:00:00	581
8	Seafood	1998-02-01T00:00:00	564
5	Grains/Cereals	1998-01-01T00:00:00	544
1	Beverages	1997-12-01T00:00:00	498
4	Dairy Products	1997-05-01T00:00:00	494

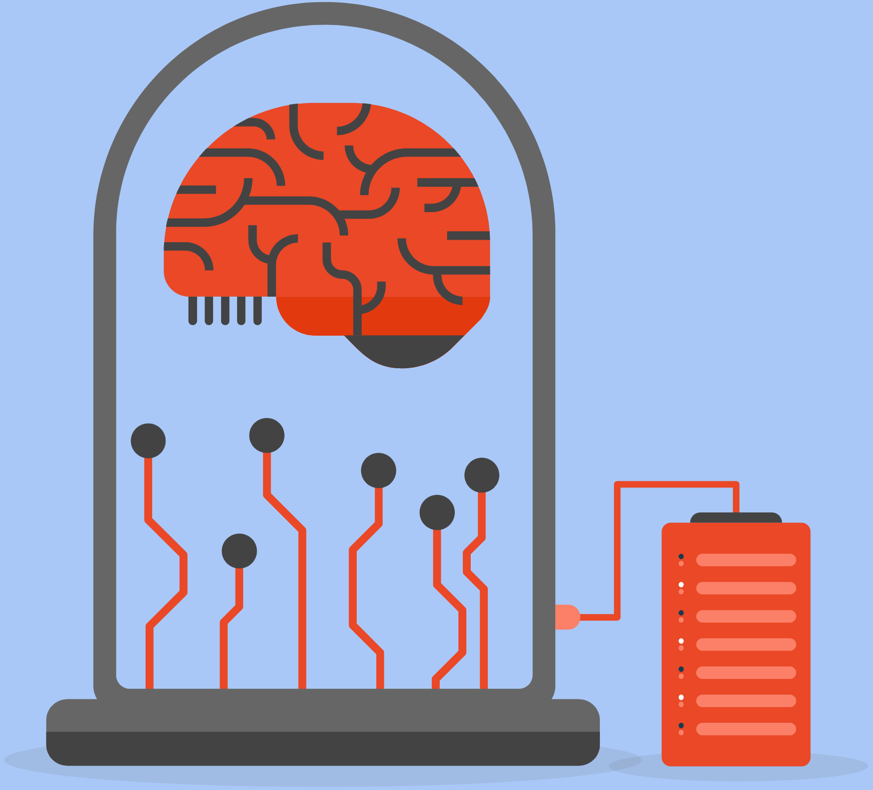

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS QUERY RESULTS LINEAGE DOCUMENTATION EDITOR ACTIONS
Switch to legacy UI
Clear results
Help
Feedback
configure

Preview 192 rows in 5.4s   SQL

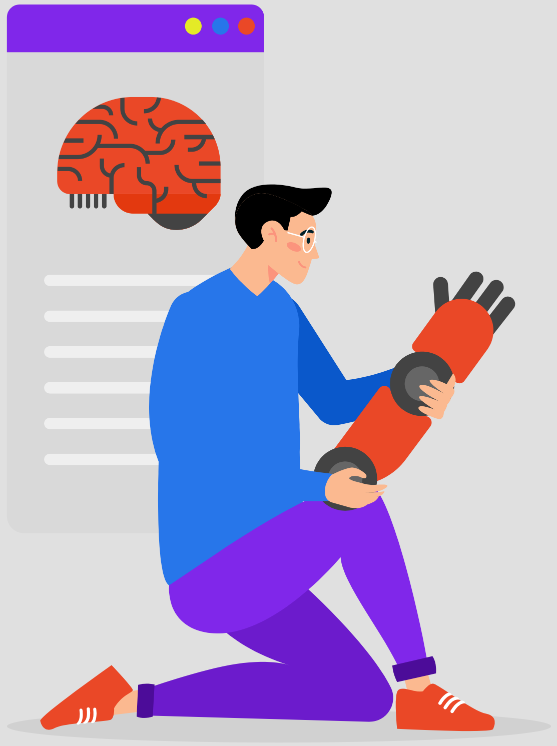
query result   192 x 4

employeeID	fullname	month	total_gross_revenue
7	King Robert	1998-04-01T00:00:00	34792.649965286255
2	Fuller Andrew	1998-04-01T00:00:00	32681.049989700317
3	Leverling Janet	1998-01-01T00:00:00	27833.349946975708
2	Fuller Andrew	1998-02-01T00:00:00	26056.900053024292
4	Peacock Margaret	1997-01-01T00:00:00	25620.1001830101
1	Davolio Nancy	1998-03-01T00:00:00	24847.45006942749
3	Leverling Janet	1998-02-01T00:00:00	21540.239961624146
4	Peacock Margaret	1998-01-01T00:00:00	21029.299980163574
8	Callahan Laura	1998-03-01T00:00:00	20885.700080871582
1	Davolio Nancy	1997-07-01T00:00:00	19997.880086898804
9	Dodsworth Anne	1998-02-01T00:00:00	19325.510009765625
4	Peacock Margaret	1997-12-01T00:00:00	18843.250032424927
3	Leverling Janet	1997-05-01T00:00:00	18639.799976348877
3	Leverling Janet	1997-12-01T00:00:00	18493.999919891357
3	Leverling Janet	1998-03-01T00:00:00	18358.34992980957
4	Peacock Margaret	1997-08-01T00:00:00	16766.789986610413
1	Davolio Nancy	1997-12-01T00:00:00	16077.250034332275
4	Peacock Margaret	1996-10-01T00:00:00	14422.100032806396

PROJECT 3



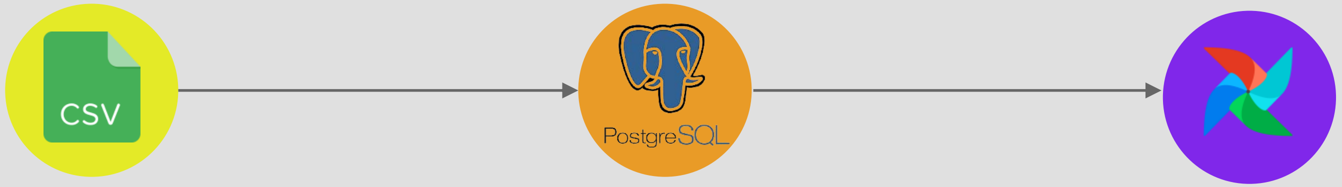
Objective of The Project



Understanding and Implementing an End-to-End Data Pipeline with Apache Airflow, Docker, and PostgreSQL

DBeaver PostgreSQL docker Apache Airflow Google Big Query

Pipeline

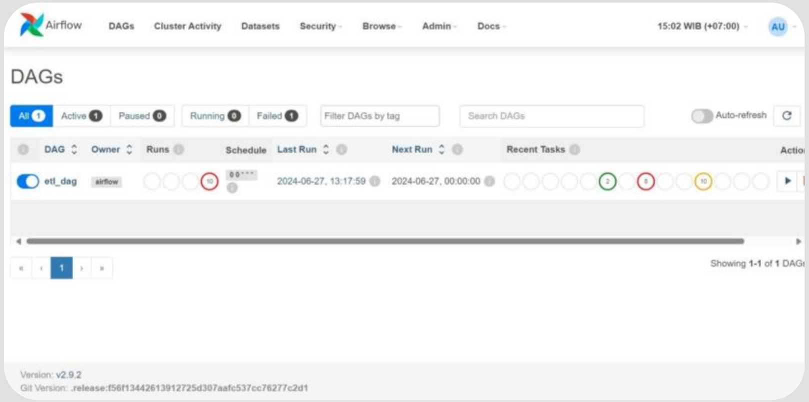


Source data from kaggle

Insert raw data from source into postgre and create dwh

Create dag airflow with docker

DAG AIRFLOW



Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs 15:02 WIB (+07:00) AU

DAGs

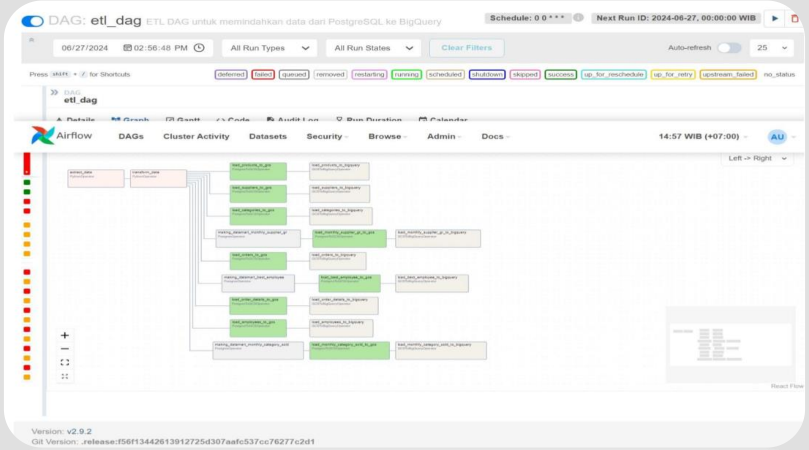
All 1 Active 1 Paused 0 Running 0 Failed 1 Filter DAGs by tag Search DAGs Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actio
etl_dag	airflow	10	0 0 * * *	2024-06-27, 13:17:59	2024-06-27, 00:00:00	2 8 10	

« ‹ 1 › »

Showing 1-1 of 1 DAGs

Version: v2.9.2
Git Version: .release:f56f13442613912725d307aafc537cc76277c2d1



DAG: etl_dag ETL DAG untuk memindahkan data dari PostgreSQL ke BigQuery Schedule: 0 0 * * * Next Run ID: 2024-06-27, 00:00:00 WIB

06/27/2024 02:56:48 PM All Run Types All Run States Clear Filters Auto-refresh 25

Press shift + / for Shortcuts deferred failed queued removed restarting running scheduled shutdown skipped success up_for_reschedule up_for_retry upstream_failed no_status

DAG
etl_dag

Details Graph Gantt Code Audit Log Run Duration Calendar

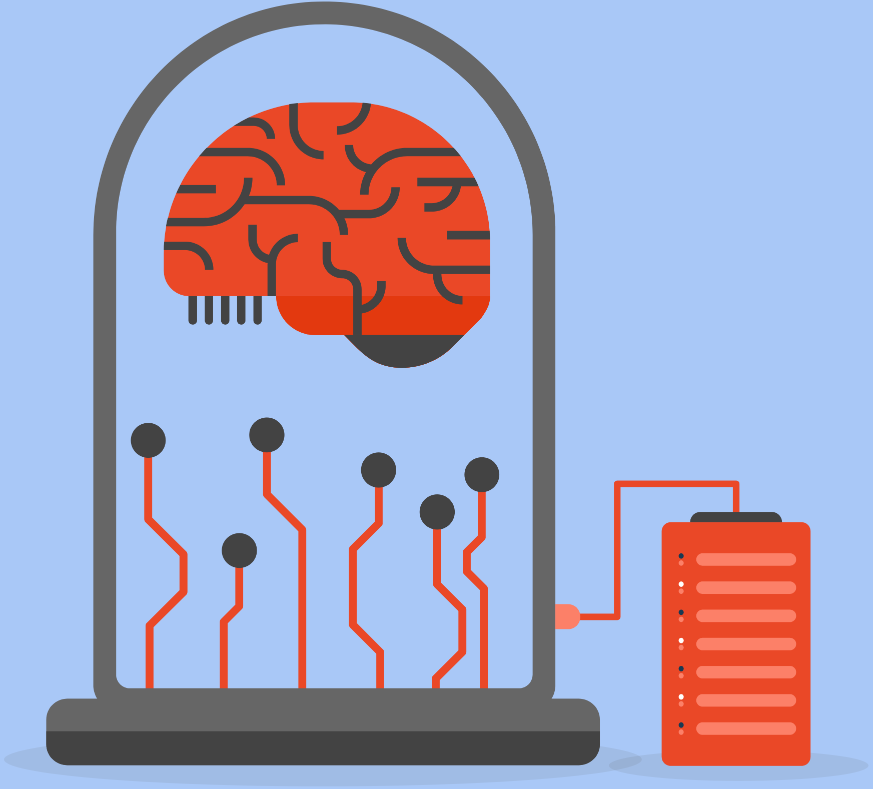
Airflow DAGs Cluster Activity Datasets Security Browse Admin Docs 14:57 WIB (+07:00) AU

Left -> Right

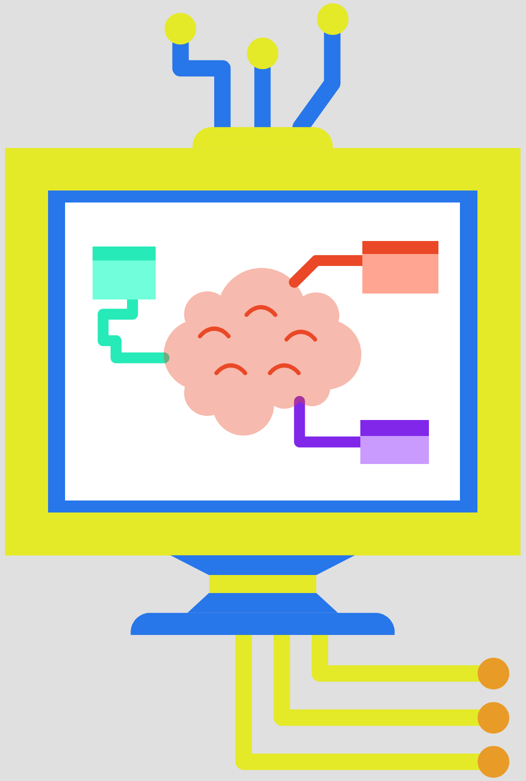
extract_data
transform_data
load_products_to_gcs
load_products_to_bigquery
load_suppliers_to_gcs
load_suppliers_to_bigquery
load_categories_to_gcs
load_categories_to_bigquery
making_datamart_monthly_supplier_gr
load_monthly_supplier_gr_to_gcs
load_monthly_supplier_gr_to_bigquery
load_orders_to_gcs
load_orders_to_bigquery
making_datamart_best_employee
load_best_employee_to_gcs
load_best_employee_to_bigquery
load_order_details_to_gcs
load_order_details_to_bigquery
load_employees_to_gcs
load_employees_to_bigquery
making_datamart_monthly_category_sold
load_monthly_category_sold_to_gcs
load_monthly_category_sold_to_bigquery
React Flow

Version: v2.9.2
Git Version: .release:f56f13442613912725d307aafc537cc76277c2d1

PROJECT 4



Objective of The Project

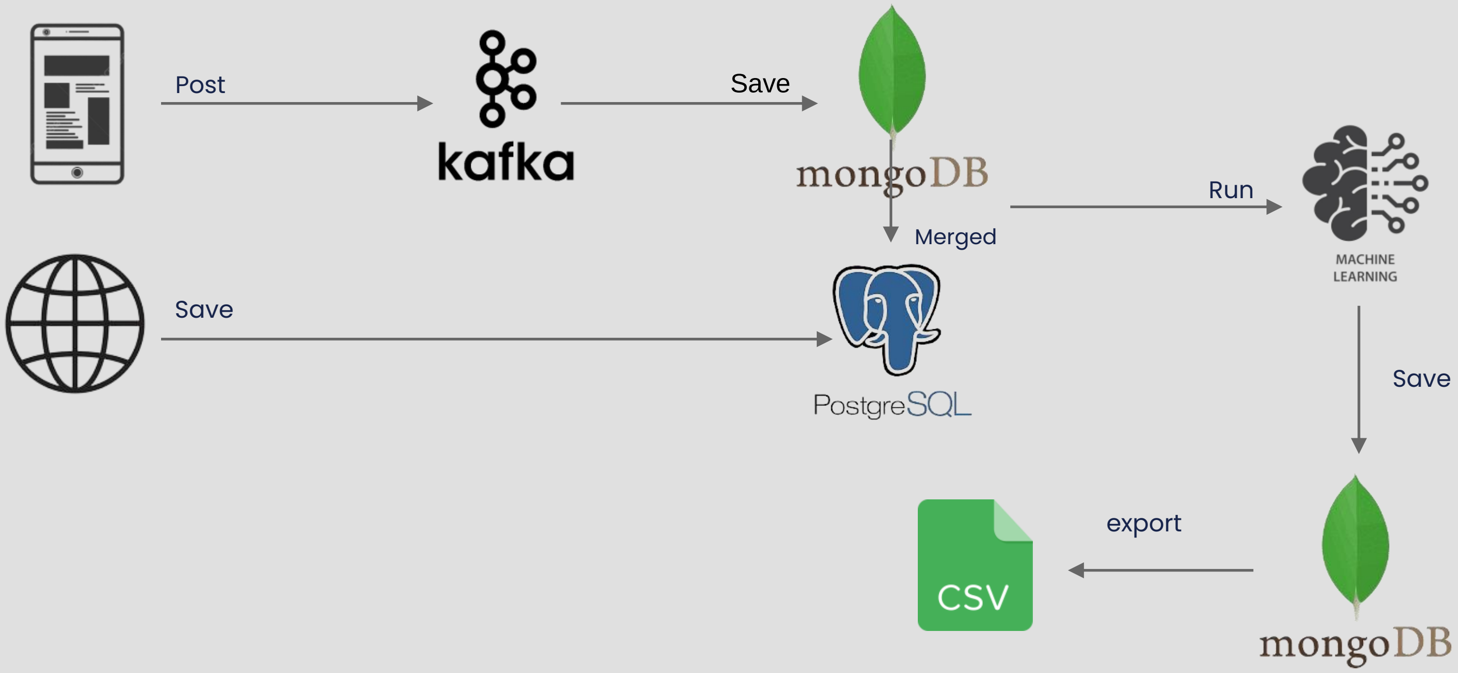


Integrating Apache Kafka, PostgreSQL, and Machine Learning.

In this project, we aim to achieve the following: Understand the workings of Apache Kafka for real-time streaming. Combine Kafka data with information available in PostgreSQL. Utilize this integrated data in a machine learning model.

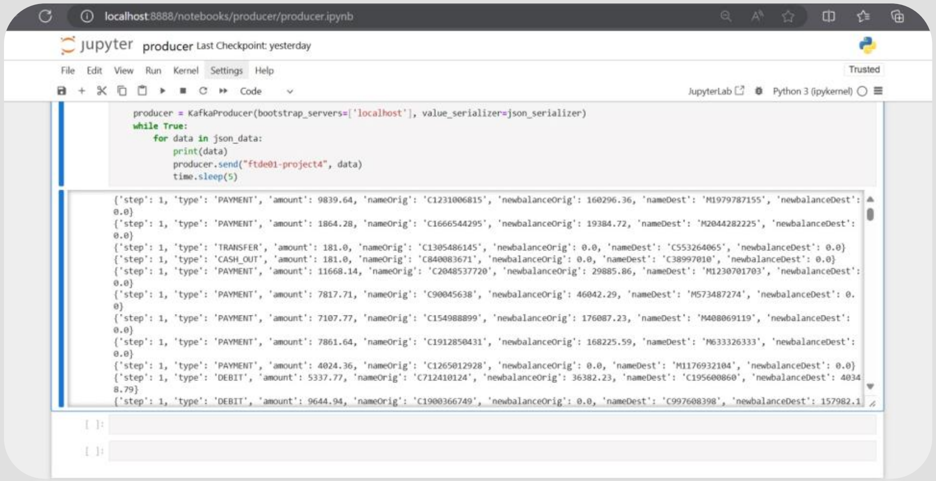
DBeaver PostgreSQL docker Apache Airflow kafka mongoDB MACHINE LEARNING

Pipeline



Post
kafka
Save
mongoDB
Merged
Run
MACHINE LEARNING
Save
PostgreSQL
Save
export
CSV
mongoDB

Producer



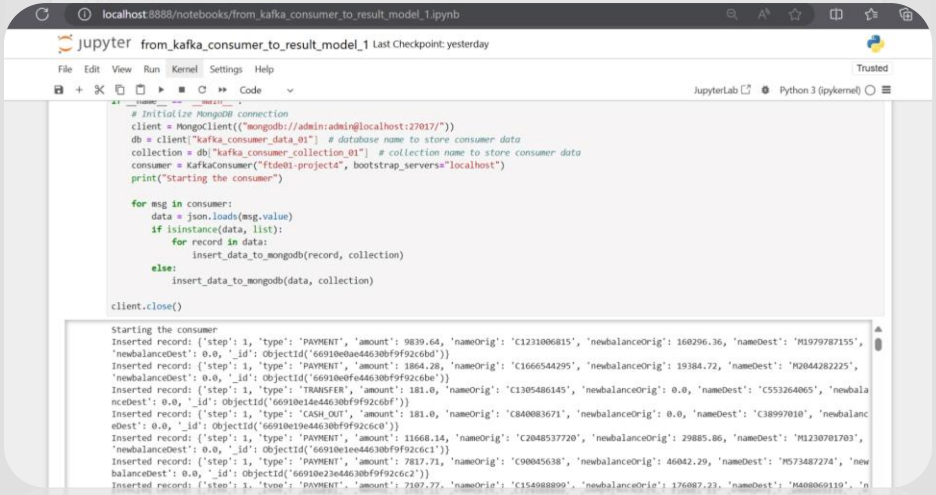
localhost:8888/notebooks/producer/producer.ipynb

jupyter producer Last Checkpoint: yesterday

```
producer = KafkaProducer(bootstrap_servers=['localhost'], value_serializer=json_serializer)
while True:
    for data in json_data:
        print(data)
        producer.send("ftde01-project4", data)
        time.sleep(5)
```

```
{'step': 1, 'type': 'PAYMENT', 'amount': 9839.64, 'nameOrig': 'C1231006815', 'newbalanceOrig': 160296.36, 'nameDest': 'M1979787155', 'newbalanceDest':
0.0}
{'step': 1, 'type': 'PAYMENT', 'amount': 1864.28, 'nameOrig': 'C1666544295', 'newbalanceOrig': 19384.72, 'nameDest': 'M2044282225', 'newbalanceDest':
0.0}
{'step': 1, 'type': 'TRANSFER', 'amount': 181.0, 'nameOrig': 'C1305486145', 'newbalanceOrig': 0.0, 'nameDest': 'C553264065', 'newbalanceDest': 0.0}
{'step': 1, 'type': 'CASH_OUT', 'amount': 181.0, 'nameOrig': 'C840083671', 'newbalanceOrig': 0.0, 'nameDest': 'C38997010', 'newbalanceDest': 0.0}
{'step': 1, 'type': 'PAYMENT', 'amount': 11668.14, 'nameOrig': 'C2048537720', 'newbalanceOrig': 29885.86, 'nameDest': 'M1230701703', 'newbalanceDest':
0.0}
{'step': 1, 'type': 'PAYMENT', 'amount': 7817.71, 'nameOrig': 'C90045638', 'newbalanceOrig': 46042.29, 'nameDest': 'M573487274', 'newbalanceDest': 0.
0}
{'step': 1, 'type': 'PAYMENT', 'amount': 7107.77, 'nameOrig': 'C154988899', 'newbalanceOrig': 176087.23, 'nameDest': 'M408069119', 'newbalanceDest':
0.0}
{'step': 1, 'type': 'PAYMENT', 'amount': 7861.64, 'nameOrig': 'C1912850431', 'newbalanceOrig': 168225.59, 'nameDest': 'M633326333', 'newbalanceDest':
0.0}
{'step': 1, 'type': 'PAYMENT', 'amount': 4024.36, 'nameOrig': 'C1265012928', 'newbalanceOrig': 0.0, 'nameDest': 'M1176932104', 'newbalanceDest': 0.0}
{'step': 1, 'type': 'DEBIT', 'amount': 5337.77, 'nameOrig': 'C712410124', 'newbalanceOrig': 36382.23, 'nameDest': 'C195600860', 'newbalanceDest': 4034
8.79}
{'step': 1, 'type': 'DEBIT', 'amount': 9644.94, 'nameOrig': 'C1900366749', 'newbalanceOrig': 0.0, 'nameDest': 'C997608398', 'newbalanceDest': 157982.1
```

Consumer



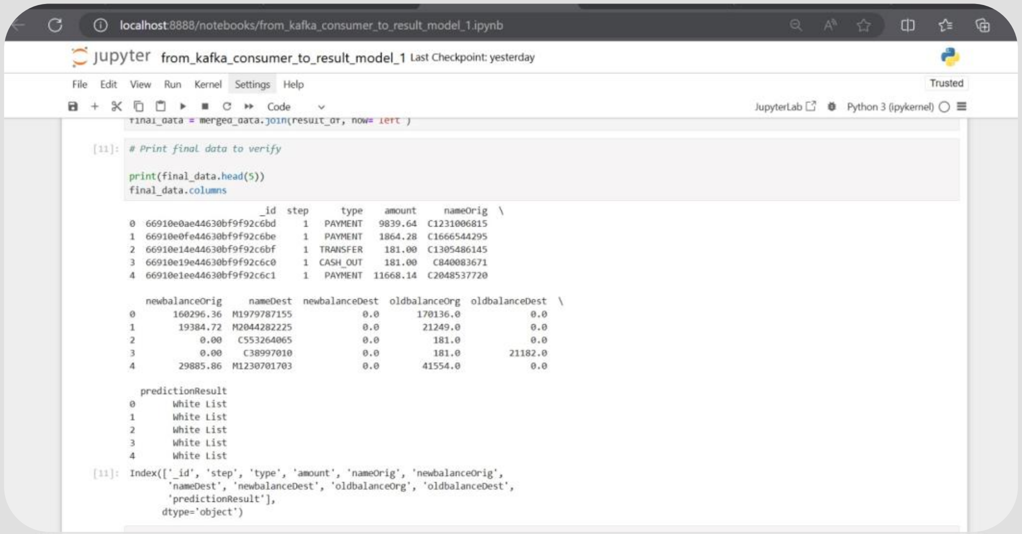
localhost:8888/notebooks/from_kafka_consumer_to_result_model_1.ipynb

jupyter from_kafka_consumer_to_result_model_1 Last Checkpoint: yesterday

```
    # Initialize MongoDB connection
    client = MongoClient(("mongodb://admin:admin@localhost:27017/"))
    db = client["kafka_consumer_data_01"]  # database name to store consumer data
    collection = db["kafka_consumer_collection_01"]  # collection name to store consumer data
    consumer = KafkaConsumer("ftde01-project4", bootstrap_servers="localhost")
    print("Starting the consumer")

    for msg in consumer:
        data = json.loads(msg.value)
        if isinstance(data, list):
            for record in data:
                insert_data_to_mongodb(record, collection)
        else:
            insert_data_to_mongodb(data, collection)

client.close()
```

```
Starting the consumer
Inserted record: {'step': 1, 'type': 'PAYMENT', 'amount': 9839.64, 'nameOrig': 'C1231006815', 'newbalanceOrig': 160296.36, 'nameDest': 'M1979787155',
'newbalanceDest': 0.0, '_id': ObjectId('66910e0ae44630bf9f92c6bd')}
Inserted record: {'step': 1, 'type': 'PAYMENT', 'amount': 1864.28, 'nameOrig': 'C1666544295', 'newbalanceOrig': 19384.72, 'nameDest': 'M2044282225',
'newbalanceDest': 0.0, '_id': ObjectId('66910e0fe44630bf9f92c6be')}
Inserted record: {'step': 1, 'type': 'TRANSFER', 'amount': 181.0, 'nameOrig': 'C1305486145', 'newbalanceOrig': 0.0, 'nameDest': 'C553264065', 'newbala
nceDest': 0.0, '_id': ObjectId('66910e14e44630bf9f92c6bf')}
Inserted record: {'step': 1, 'type': 'CASH_OUT', 'amount': 181.0, 'nameOrig': 'C840083671', 'newbalanceOrig': 0.0, 'nameDest': 'C38997010', 'newbalanc
eDest': 0.0, '_id': ObjectId('66910e19e44630bf9f92c6c0')}
Inserted record: {'step': 1, 'type': 'PAYMENT', 'amount': 11668.14, 'nameOrig': 'C2048537720', 'newbalanceOrig': 29885.86, 'nameDest': 'M1230701703',
'newbalanceDest': 0.0, '_id': ObjectId('66910e1ee44630bf9f92c6c1')}
Inserted record: {'step': 1, 'type': 'PAYMENT', 'amount': 7817.71, 'nameOrig': 'C90045638', 'newbalanceOrig': 46042.29, 'nameDest': 'M573487274', 'new
balanceDest': 0.0, '_id': ObjectId('66910e23e44630bf9f92c6c2')}
Inserted record: {'step': 1, 'type': 'PAYMENT', 'amount': 7107.77, 'nameOrig': 'C154988899', 'newbalanceOrig': 176087.23, 'nameDest': 'M408069119', 'n
```

Model Result


localhost:8888/notebooks/from_kafka_consumer_to_result_model_1.ipynb
jupyter from_kafka_consumer_to_result_model_1 Last Checkpoint: yesterday
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

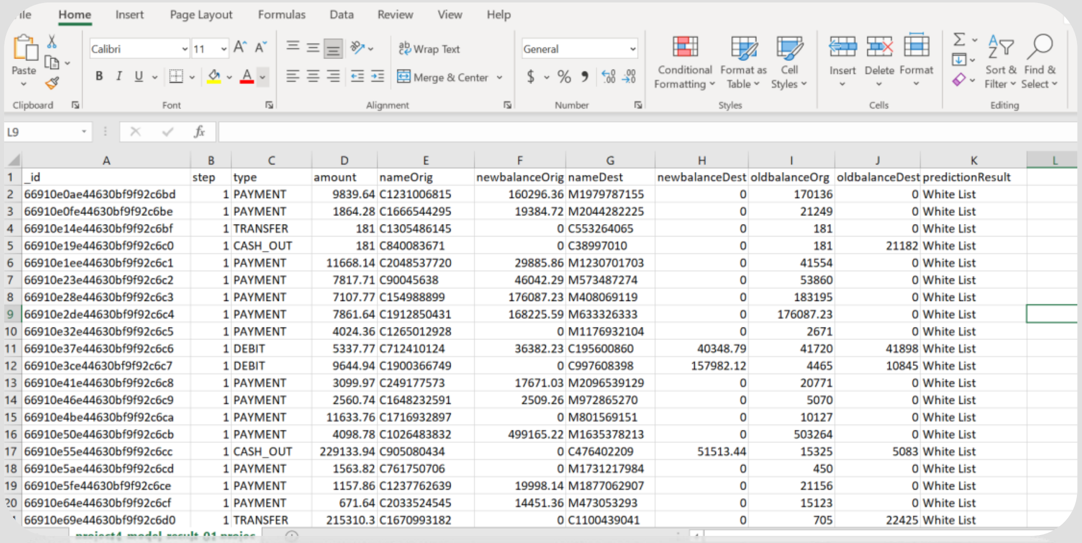
```
final_data = merged_data.join(result_df, how='left')
```

```
[11]: # Print final data to verify

      print(final_data.head(5))
      final_data.columns
```

```
                        _id  step      type    amount     nameOrig  \
0  66910e0ae44630bf9f92c6bd     1   PAYMENT   9839.64  C1231006815
1  66910e0fe44630bf9f92c6be     1   PAYMENT   1864.28  C1666544295
2  66910e14e44630bf9f92c6bf     1  TRANSFER    181.00  C1305486145
3  66910e19e44630bf9f92c6c0     1  CASH_OUT    181.00   C840083671
4  66910e1ee44630bf9f92c6c1     1   PAYMENT  11668.14  C2048537720

   newbalanceOrig     nameDest  newbalanceDest  oldbalanceOrg  oldbalanceDest  \
0       160296.36  M1979787155             0.0       170136.0             0.0
1        19384.72  M2044282225             0.0        21249.0             0.0
2            0.00   C553264065             0.0          181.0             0.0
3            0.00    C38997010             0.0          181.0         21182.0
4        29885.86  M1230701703             0.0        41554.0             0.0

  predictionResult
0       White List
1       White List
2       White List
3       White List
4       White List
```

```
[11]: Index(['_id', 'step', 'type', 'amount', 'nameOrig', 'newbalanceOrig',
             'nameDest', 'newbalanceDest', 'oldbalanceOrg', 'oldbalanceDest',
             'predictionResult'],
            dtype='object')
```

Model Result in CSV



_id	step	type	amount	nameOrig	newbalanceOrig	nameDest	newbalanceDest	oldbalanceOrg	oldbalanceDest	predictionResult
66910e0ae44630bf9f92c6bd	1	PAYMENT	9839.64	C1231006815	160296.36	M1979787155	0	170136	0	White List
66910e0fe44630bf9f92c6be	1	PAYMENT	1864.28	C1666544295	19384.72	M2044282225	0	21249	0	White List
66910e14e44630bf9f92c6bf	1	TRANSFER	181	C1305486145	0	C553264065	0	181	0	White List
66910e19e44630bf9f92c6c0	1	CASH_OUT	181	C840083671	0	C38997010	0	181	21182	White List
66910e1ee44630bf9f92c6c1	1	PAYMENT	11668.14	C2048537720	29885.86	M1230701703	0	41554	0	White List
66910e23e44630bf9f92c6c2	1	PAYMENT	7817.71	C90045638	46042.29	M573487274	0	53860	0	White List
66910e28e44630bf9f92c6c3	1	PAYMENT	7107.77	C154988899	176087.23	M408069119	0	183195	0	White List
66910e2de44630bf9f92c6c4	1	PAYMENT	7861.64	C1912850431	168225.59	M633326333	0	176087.23	0	White List
66910e32e44630bf9f92c6c5	1	PAYMENT	4024.36	C1265012928	0	M1176932104	0	2671	0	White List
66910e37e44630bf9f92c6c6	1	DEBIT	5337.77	C712410124	36382.23	C195600860	40348.79	41720	41898	White List
66910e3ce44630bf9f92c6c7	1	DEBIT	9644.94	C1900366749	0	C997608398	157982.12	4465	10845	White List
66910e41e44630bf9f92c6c8	1	PAYMENT	3099.97	C249177573	17671.03	M2096539129	0	20771	0	White List
66910e46e44630bf9f92c6c9	1	PAYMENT	2560.74	C1648232591	2509.26	M972865270	0	5070	0	White List
66910e4be44630bf9f92c6ca	1	PAYMENT	11633.76	C1716932897	0	M801569151	0	10127	0	White List
66910e50e44630bf9f92c6cb	1	PAYMENT	4098.78	C1026483832	499165.22	M1635378213	0	503264	0	White List
66910e55e44630bf9f92c6cc	1	CASH_OUT	229133.94	C905080434	0	C476402209	51513.44	15325	5083	White List
66910e5ae44630bf9f92c6cd	1	PAYMENT	1563.82	C761750706	0	M1731217984	0	450	0	White List
66910e5fe44630bf9f92c6ce	1	PAYMENT	1157.86	C1237762639	19998.14	M1877062907	0	21156	0	White List
66910e64e44630bf9f92c6cf	1	PAYMENT	671.64	C2033524545	14451.36	M473053293	0	15123	0	White List
66910e69e44630bf9f92c6d0	1	TRANSFER	215310.3	C1670993182	0	C1100439041	0	705	22425	White List

Thank You