**PROJECT REPORT**

***-***Saurabh Kumar(CAP\_DS\_05)

[DataSets](https://drive.google.com/file/d/1OR2enZd3OIpOX76QBj226lfrD9JuQJ_E/view?usp=sharing)

SOFTWARE USED : -

* Database : My SQL Workbench
* Visualization: MS Power-BI
* Preprocessing : Jupyter Notebook

DATA DETAILS

This dataset containing customer purchase information for an online retail company. The dataset includes the following details:

* Transaction ID
* Customer ID
* Customer Name
* Product ID
* Product Name
* Product Category
* Purchase Quantity
* Purchase Price
* Purchase Date
* Country

**1. Data Extraction and Transformation (SQL):**

* **Setup a Database**: Create a database to store the customer purchase data.
* create database Project ;
* **Data Ingestion**: Write SQL scripts to import the provided purchase data into the database.

**Steps :**

* + 1. Format of data is csv
    2. Data is directly imported into the table using Table data import wizard
* **Data Transformation:**

**Normalization of the data.**

Data is normalized into 3 Tables :-

* 1. Customers
  2. Products
  3. Purchase

NORMALIZATIONS :

create table customers as

select row\_number() over(order by purchasedate)+5000 as customerid ,

customername , country from customer\_purchase\_data;

CREATE TABLE Products AS

(

with cte as

(

select distinct productname , productcategory ,

dense\_rank() over(partition by productcategory order by productName)+200 as pid

from customer\_purchase\_data

)

select row\_number() over(order by productname)+200 as productid ,productname,productcategory

from cte

);

CREATE TABLE Purchase AS

with cte1 as

(

SELECT row\_number()over(order by purchasedate asc)+5000 as PurchaseID,PurchaseQuantity,PurchasePrice,PurchaseDate,

cs.customerid,p.ProductID,TransactionID

FROM customer\_purchase\_data c

join products p

on c.productname = p.productname

join customers cs

on c.customername = cs.customername AND c.Country=cs.Country

)

select PurchaseID, PurchaseQuantity, PurchasePrice, PurchaseDate,customerid, ProductID, TransactionID

from cte1 ;

UNDERSTANDING SCHEMA DATATYPES, CONSTRAINTS

* describe customers ;
* describe products ;
* describe purchase ;

**Handling missing values.**

CHECKING NULL VALUES

select \* from customer\_purchase\_data

where TransactionID is null or

CustomerId is null or

CustomerName is null or

ProductID is null or

ProductName is null or

ProductCategory is null or

PurchaseQuantity is null or

PurchasePrice is null or

PurchaseDate is null or

country is null ;

Checking Duplicate data

Select

TransactionID ,CustomerId ,CustomerName ,ProductID ,ProductName ,ProductCategory,PurchaseQuantity ,PurchasePrice ,PurchaseDate , country, count(\*)

from customer\_purchase\_data

group by TransactionID ,CustomerId ,CustomerName ,ProductID ,ProductName ,ProductCategory,PurchaseQuantity ,PurchasePrice ,PurchaseDate , country

having count(\*) > 1 ;

**Creating relationships between tables.**

**BUILD RELATIONSHIP IN TABLES**

* ALTER TABLE customers

ADD PRIMARY KEY (CustomerID);

* ALTER TABLE purchase

ADD PRIMARY KEY (purchaseID);

* ALTER TABLE products

ADD PRIMARY KEY (productID);

* ALTER TABLE purchase

ADD CONSTRAINT fk FOREIGN KEY (CustomerID)

REFERENCES customers(CustomerID);

* ALTER TABLE purchase

ADD CONSTRAINT fk2 FOREIGN KEY (productID)

REFERENCES products(productID);

**Advanced queries to aggregate data**

* *Finding total purchase by each customer*

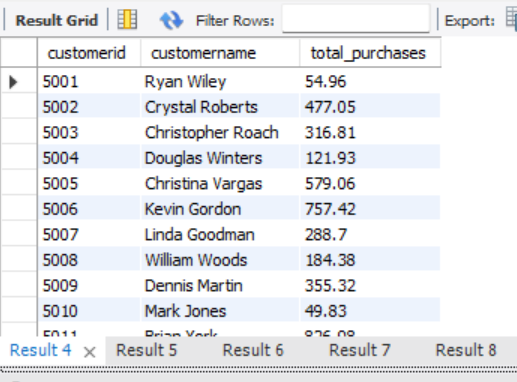
SELECT c.customerid, c.customername,SUM(p.PurchasePrice) AS total\_purchase

FROM customers c

JOIN purchase p ON c.customerid = p.customerid

GROUP BY c.customerid , c.customername

ORDER BY c.customerid;



* *Total sales per product*

SELECT

pd.ProductId,

pd.productname,

SUM(pc.PurchasePrice) AS total\_sales\_per\_product

FROM

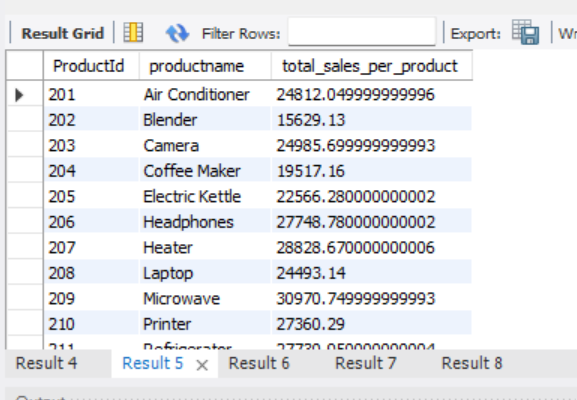
products AS pd

JOIN

Purchase AS pc ON pd.productid = pc.productid

GROUP BY pd.ProductId , pd.productname

ORDER BY pd.ProductId;



* + *5 top selling product*

SELECT

pd.ProductId,

pd.productname,

round(SUM(pc.PurchasePrice),2) AS total\_sales\_per\_product

FROM

products AS pd

JOIN

Purchase AS pc ON pd.productid = pc.productid

GROUP BY pd.ProductId , pd.productname

ORDER BY total\_sales\_per\_product desc

limit 5 ;

A screenshot of a computer

Description automatically generated

* + *Top selling product in each category*

WITH ranked\_products AS (

SELECT

p.productid,

p.productname,

p.productcategory,

SUM(pc.PurchaseQuantity) AS total\_quantity\_sold,

ROW\_NUMBER() OVER (PARTITION BY p.productcategory ORDER BY SUM(pc.PurchaseQuantity) desc) AS rank\_in\_category

FROM

products p

JOIN

Purchase pc ON p.productid = pc.productid

GROUP BY

p.productid,p.productname,p.productcategory

)

SELECT

p.productcategory,

rp.productname,

rp.total\_quantity\_sold

FROM

ranked\_products as rp

JOIN products as p ON rp.productid = p.productid

WHERE

rp.rank\_in\_category = 1;

A screenshot of a computer

Description automatically generated

* + *Total revenue by year*

SELECT

YEAR(PurchaseDate), Round(SUM(PurchasePrice),2) AS total\_revenue

FROM

purchase

GROUP BY YEAR(PurchaseDate)

ORDER BY YEAR(PurchaseDate);

A screenshot of a computer

Description automatically generated

1. **Data Analysis (Python):**

Libraries used :-

* + - import sqlalchemy
    - import pymysql
    - import pandas as pd
    - import matplotlib.pyplot as plt
    - import seaborn as sns
* **Data Extraction**: Write a Python script to extract data from the SQL database.

#connecting sql and python

python\_engine = sqlalchemy.create\_engine('mysql+pymysql://root:123456789@localhost:3306/project')

#loading data from MySQL database

customers = pd.read\_sql\_table("customers",python\_engine)

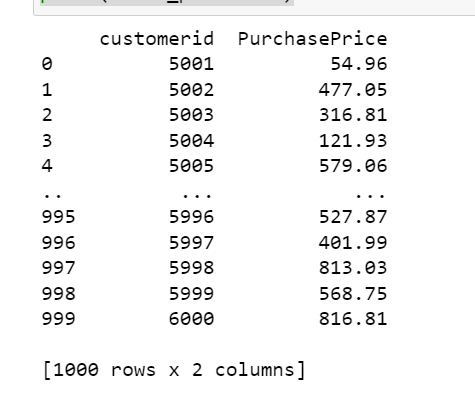
purchase = pd.read\_sql\_table("purchase",python\_engine)

products=pd.read\_sql\_table("products",python\_engine)

* **Data Analysis**: Perform the following analysis using basic Python:
  + ***Calculate total purchases, total revenue, and average purchase value.***

total\_purchases = purchase.groupby('customerid')['PurchasePrice'].sum().reset\_index()

print(total\_purchases)



total\_revenue = round(purchase['PurchasePrice'].sum(),2)

print(total\_revenue)

A screenshot of a computer

Description automatically generated

Avg\_purchase = round(purchase['PurchasePrice'].mean(),2)

print(Avg\_purchase)

A close-up of a computer screen

Description automatically generated

* ***Identify top customers and their purchasing behavior.***

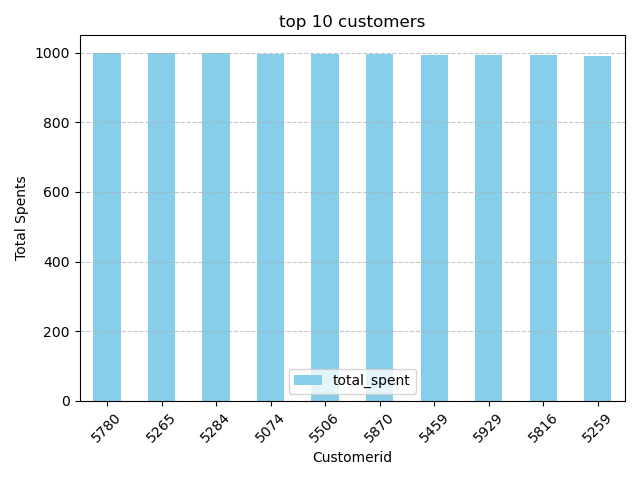
customer\_purchase = purchase.groupby('customerid')['PurchasePrice'].sum().reset\_index()

top\_customers = customer\_purchase.sort\_values(by='PurchasePrice', ascending=False)

print(top\_customers.head(10))

A screenshot of a computer screen

Description automatically generated



* ***Analyze purchase trends over time (monthly, quarterly, yearly).***

MONTHLY TRENDS

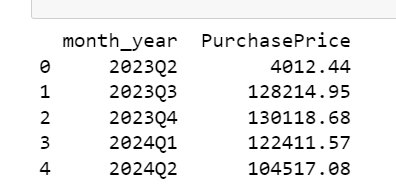
A screenshot of a computer screen

Description automatically generated

A graph with a line going up

Description automatically generated

QUATERLY TREND



A green bar graph with numbers and a white background

Description automatically generated

YEARLY TREND

A number of years and a few years

Description automatically generated with medium confidence

A graph of a sales chart

Description automatically generated with medium confidence

* ***Identify the top-performing product categories.***

A screenshot of a computer

Description automatically generated

A graph of sales

Description automatically generated

**3. Data Visualization and Reporting (PowerBI):**

* **Dashboard Creation**: Use PowerBI to create an interactive dashboard that includes:
  + ***Purchase trends over time.***

A screen shot of a graph

Description automatically generated

A graph with blue lines and white text

Description automatically generated

A graph on a screen

Description automatically generated

* + ***Top customers and their purchase behavior.***

A screenshot of a graph

Description automatically generated

* + ***Performance by product categories.***

A graph of blue bars

Description automatically generated

A graph of blue bars

Description automatically generated

* + ***Any other insights you find relevant.***

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**E-R DIAGRAM**

**A screenshot of a computer

Description automatically generated**

USEFUL LINKS

* + - 1. [Power-BI Dashboard](https://drive.google.com/file/d/1Z9HoJg5sa3gi58IFncXoA_PCOscV8GGJ/view?usp=sharing)
      2. [My-SQL Query](https://drive.google.com/file/d/1Dw8OuFmV335Xaja_-HZui0DhTvqHeWMG/view?usp=sharing)
      3. [Python](https://drive.google.com/file/d/16gkmZEh4amlCkc0ATXIkV_Jq2-FnZkaA/view?usp=sharing)

**SUBMITTED BY**

Saurabh Kumar

Cap\_ds\_05

+91 6201147454

Saurabhkumar843327@gmail.com