# 

# High Impact Skills Development Program

# in Artificial Intelligence, Data Science, and Blockchain

**Project Title**: Online Retail Segmentation.

Devoloper: Naeem Hassan.

**Learning Objectives:**

1. **Understanding the fundamentals of Data Mining**
2. **Learn how to use SQL in data mining**
3. **Learn how to implement mining concepts in MySQL Workbench**

**Overview:**

Customer segmentation is a typical strategy used by organizations to categories clients based on their demographics, shopping patterns, or other traits.

Customer segmentation constitutes a widely adopted strategic approach within organizations, aimed at systematically categorizing clients by analyzing a spectrum of factors such as demographics, shopping patterns, and other discernible traits. By effectively segmenting the customer base, businesses gain the ability to tailor their offerings, communications, and marketing efforts with a higher degree of precision, ultimately leading to enhanced customer satisfaction, targeted engagement, and improved overall business performance.

**MetaData:**

Look for a dataset that contains information about customers such as demographic information, purchasing history, and customer interactions. The data set contains the following variables:

InvoiceNo: The invoice number for each transaction

StockCode: The unique code for each product sold

Description: The description of each product sold

Quantity: The quantity of each product sold in each transaction

InvoiceDate: The date and time of each transaction

UnitPrice: The price of each product sold

CustomerID: The unique identifier for each customer

Country: The country where each transaction occurred

SQL Project Idea: Use SQL queries to answer the following questions:

## **Beginner Queries.**

* **Define meta data in mysql workbench**

In MySQL Workbench, metadata refers to information about the structure and properties of the database objects, such as tables, columns, indexes, and constraints. It provides valuable insights into the organization of the database and helps in understanding its schema and relationships. MySQL Workbench allows you to view and manage metadata for your databases and their objects. Here's how you can interact with metadata in MySQL Workbench:

**Viewing Metadata:**

To view metadata for a specific database, you can use the "Catalogs" pane in the left sidebar. Expand the database you're interested in, and you'll see sections for tables, views, procedures, and more.

By clicking on a specific table, you can view the metadata for that table, such as columns, indexes, constraints, triggers, and foreign keys.

**Table Metadata:**

Right-click on a table and select "Alter Table" to see detailed metadata for that table.

You can view and modify properties of columns, indexes, and constraints, as well as set auto-increment settings and default values.

**Schema Inspector:**

In MySQL Workbench, you can access the "Schema Inspector" from the "Database" menu. This tool provides a comprehensive view of metadata for all the objects in the database.

You can use the "Schema Inspector" to analyze table relationships, check foreign key references, and get an overview of indexes and triggers.

**Data Dictionary**

MySQL Workbench also supports creating and managing a data dictionary, which is a document that describes the structure and properties of the database objects. It can include information such as table definitions, column data types, and indexes.

You can generate a data dictionary report for your database, which helps in documenting your schema.

Remember that MySQL Workbench is a powerful tool for database design and management, and its interface might evolve over time.

* **What is the distribution of order values across all customers in the dataset?**

Query 1

SELECT CustomerID, COUNT(\*) AS OrderCount, SUM(Quantity \* UnitPrice) AS TotalOrderValue

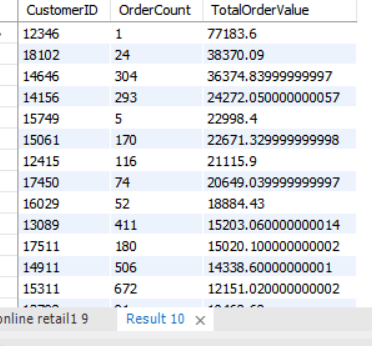
FROM dataminingproject.`online retail1`

GROUP BY CustomerID

ORDER BY TotalOrderValue DESC

LIMIT 0, 1000;

Result:



* **How many unique products has each customer purchased?**

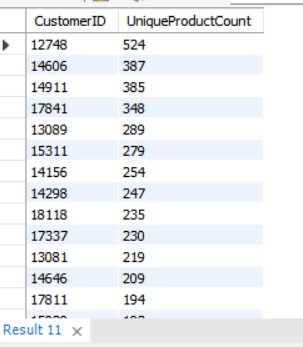
SELECT CustomerID, COUNT(DISTINCT StockCode) AS UniqueProductCount

FROM dataminingproject.`online retail1`

GROUP BY CustomerID

ORDER BY UniqueProductCount DESC;

Result:



* **Which customers have only made a single purchase from the company?**

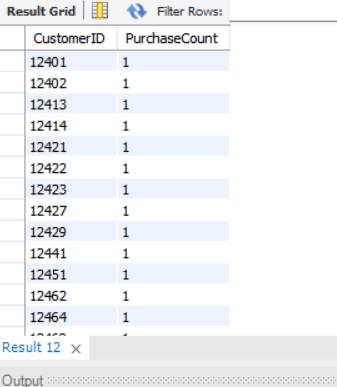
SELECT CustomerID, COUNT(DISTINCT InvoiceNo) AS PurchaseCount

FROM dataminingproject.`online retail1`

GROUP BY CustomerID

HAVING PurchaseCount = 1;

Result:



* **Which products are most commonly purchased together by customers in the dataset?**

Query 2

SELECT

a.StockCode AS ProductA,

b.StockCode AS ProductB,

COUNT(\*) AS CoPurchaseCount

FROM

dataminingproject.`online retail1` a

JOIN

dataminingproject.`online retail1` b ON a.CustomerID = b.CustomerID

AND a.InvoiceNo = b.InvoiceNo

AND a.StockCode < b.StockCode

GROUP BY

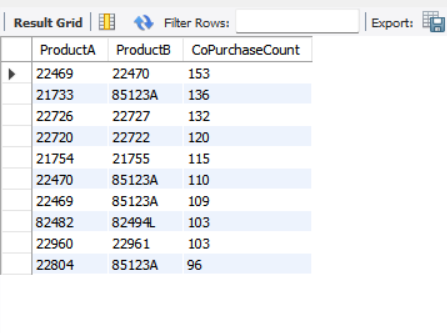
ProductA, ProductB

ORDER BY

CoPurchaseCount DESC

LIMIT 10; -- Limit the results to a certain number of most common pairs

Result:



## Advance Queries

1. **Customer Segmentation by Purchase Frequency**

Group customers into segments based on their purchase frequency, such as high, medium, and low frequency customers. This can help you identify your most loyal customers and those who need more attention.

Query 3

SELECT

CustomerID,

CASE

WHEN TotalPurchaseCount >= 10 THEN 'High Frequency'

WHEN TotalPurchaseCount >= 5 THEN 'Medium Frequency'

ELSE 'Low Frequency'

END AS PurchaseFrequencySegment

FROM (

SELECT

CustomerID,

COUNT(DISTINCT InvoiceNo) AS TotalPurchaseCount

FROM

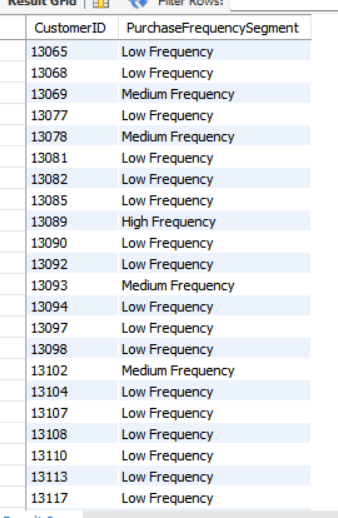
dataminingproject.`online retail1`

GROUP BY

CustomerID

) AS PurchaseCounts;

Result:



**2. Average Order Value by Country**

**Calculate the average order value for each country to identify where your most valuable customers are located.**

Query 4

SELECT

Country,

AVG(TotalOrderValue) AS AverageOrderValue

FROM (

SELECT

Country,

InvoiceNo,

SUM(Quantity \* UnitPrice) AS TotalOrderValue

FROM

dataminingproject.`online retail1`

GROUP BY

Country, InvoiceNo

) AS OrderValues

GROUP BY

Country

ORDER BY

AverageOrderValue DESC;

Result:



**3. Customer Churn Analysis**

**Identify customers who haven't made a purchase in a specific period (e.g., last 6 months) to assess churn.**

Query 5

CustomerID

FROM

dataminingproject.`online retail1`

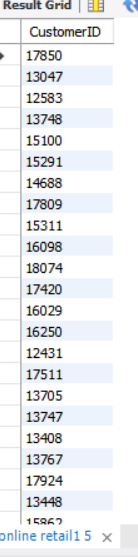
WHERE

InvoiceDate <= NOW() - INTERVAL 6 MONTH

GROUP BY

CustomerID;

Result:



**4. Product Affinity Analysis**

Determine which products are often purchased together by calculating the correlation between product purchases.

Unable to make a query.

**5. Time-based Analysis**

**Explore trends in customer behavior over time, such as monthly or quarterly sales patterns.**

SELECT

YEAR(InvoiceDate) AS Year,

MONTH(InvoiceDate) AS Month,

SUM(Quantity \* UnitPrice) AS TotalSales

FROM

dataminingproject.`online retail1`

GROUP BY

Year, Month

ORDER BY

Year, Month;

Result:

