**Module Project Data Mining**

**Name :- Muhammad Ali Khan** **Roll no :-** GIL-DSAI-090

**Github acc :-** **https://github.com/muhammada1y/data-mining-project.git**

**Simple quires (Solutions)**

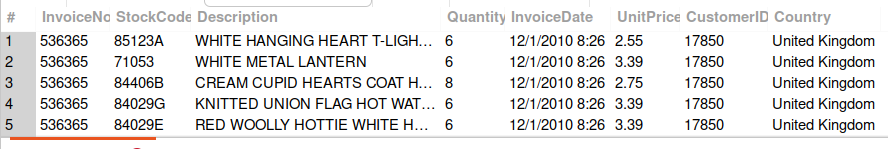
* Define meta data in mysql workbench or any other SQL tool

**Query used :-**

CREATE TABLE orders ( InvoiceNo VARCHAR(10), StockCode VARCHAR(10), Description VARCHAR(255), Quantity INT, InvoiceDate DATETIME, UnitPrice DECIMAL(10, 2), CustomerID INT, Country VARCHAR(50) );

Create table is used to create the required table with the attributes orders,StockCode,Description,Quantity,InvoiceDate,UnitePrice,CustomerID and Country which can be validated using select key word.

**Related Screenshot :-**



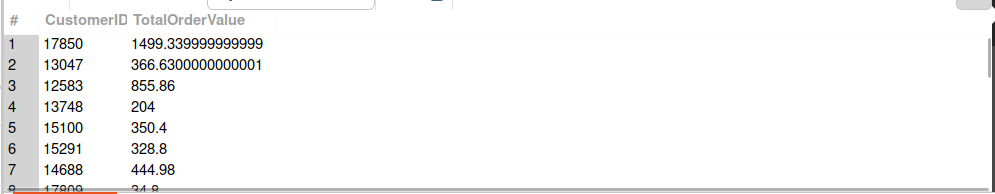
* What is the distribution of order values across all customers in the dataset?

**Query used :-**

SELECT CustomerID,SUM(Quantity \* UnitPrice) AS TotalOrderValue FROM online\_retail GROUP BY CustomerID;

The total value of the orders for each customers are calculated using Quantity \* UnitPrice formula then the group by is used to view total value for each customer.

**Related Screenshot**



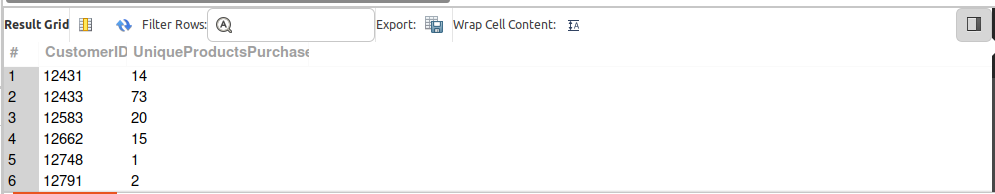
* How many unique products has each customer purchased?

**Query used :-**

SELECT CustomerID,COUNT(DISTINCT StockCode) AS UniqueProductsPurchased FROM online\_retail GROUP BY CustomerID;

This query uses distinct keyword to avoid the repetition while counting the StockCode using key word Count and group by is used to group the same CustomerID, After that prints the number of unique purchases.

**Related Screenshot :-**



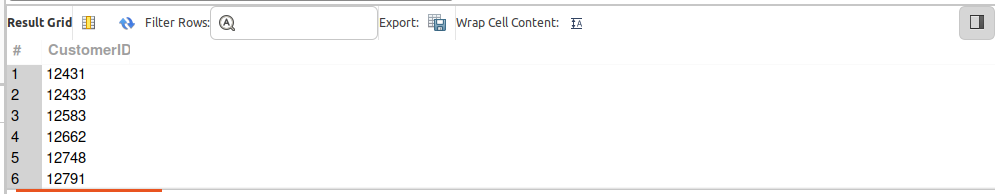
* Which customers have only made a single purchase from the company?

**Query used :-**

SELECT CustomerID FROM online\_retail GROUP BY CustomerID HAVING COUNT(DISTINCT InvoiceNo) = 1;

In this query the Id of those customers is being printed who has been allotted only one Invoice number to count the invoices count is used and to avoid repeatation distinct keyword is used.

**Related Screenshot :-**

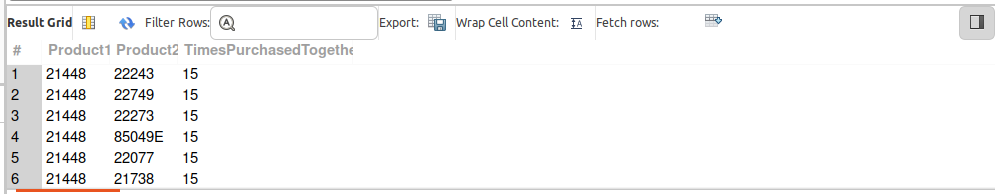
 **.** Which products are most commonly purchased together by customers in the dataset?

**Query used :-**

SELECT o1.StockCode AS Product1,o2.StockCode AS Product2,COUNT(\*) AS TimesPurchasedTogether FROM online\_retail o1 JOIN online\_retail o2 ON o1.InvoiceNo = o2.InvoiceNo AND o1.StockCode < o2.StockCode GROUP BY o1.StockCode, o2.StockCode

ORDER BY TimesPurchasedTogether DESC

**Related Screenshot :-**

 **Complex Queries (Solutions):-**

* Time-based Analysis

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

**Query used :-**

SELECT DATE\_FORMAT(InvoiceDate, '%Y-%m') AS Month,SUM(Quantity \* UnitPrice) AS TotalSales FROM orders GROUP BY DATE\_FORMAT(InvoiceDate, '%Y-%m') ORDER BY Month;

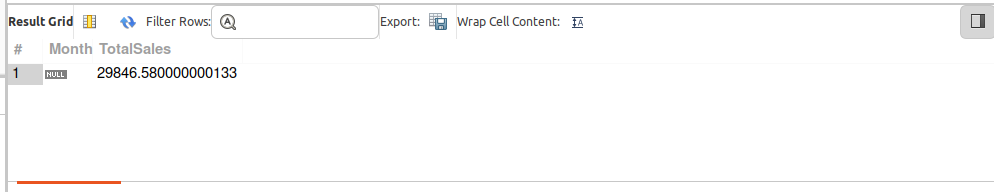
SELECT CONCAT(YEAR(InvoiceDate), ' Q', QUARTER(InvoiceDate)) AS Quarter,

SUM(Quantity \* UnitPrice) AS TotalSales FROM online\_retails GROUP BY

YEAR(InvoiceDate), QUARTER(InvoiceDate) ORDER BY Quarter;

These queries group sales data by months or quarters to show trends in customer purchases over time.

**Related Screenshot :-**



* Product Affinity Analysis

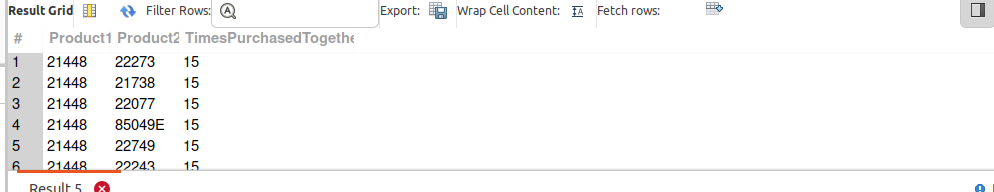
Determine which products are often purchased together by calculating the correlation between product purchases.

**Query used :-**

SELECT p1.StockCode AS Product1,p2.StockCode AS Product2,COUNT(\*) AS TimesPurchasedTogether FROM online\_retail p1 JOIN online\_retail p2 ON p1.InvoiceNo = p2.InvoiceNo AND p1.StockCode < p2.StockCode GROUP BY p1.StockCode, p2.StockCode HAVING COUNT(\*) > 1 ORDER BY TimesPurchasedTogether DESC;

This query counts how many times each pair of products has been purchased together, helping identify popular product combinations.

**Related Screenshot :-**

* Customer Churn Analysis

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

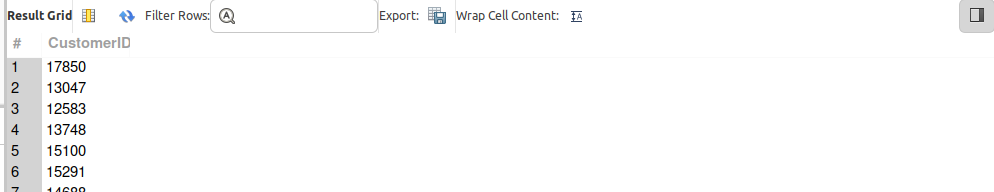
**Query used :-**

SELECT CustomerID FROM onine\_retail GROUP BY CustomerID HAVING

MAX(InvoiceDate) < DATE\_SUB(CURDATE(), INTERVAL 6 MONTH);

This query checks the maximum purchase date for each customer and compares it to the current date, identifying those who haven't made a purchase in the last 6 months.

**Related Screenshot :-**

* Average Order Value by Country

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

**Query used :-**

SELECT Country,AVG(TotalOrderValue) AS AverageOrderValue

FROM (

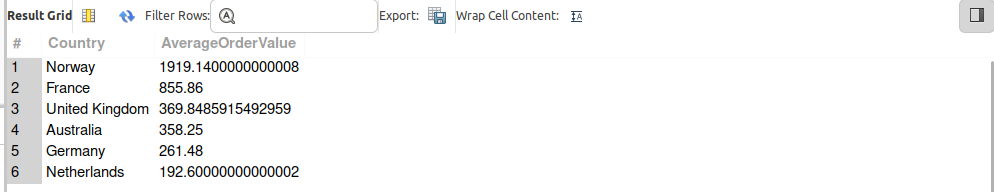
SELECT Country,InvoiceNo,SUM(Quantity \* UnitPrice) AS TotalOrderValue FROM online\_retail GROUP BY Country, InvoiceNo

)

AS OrderValues GROUP BY Country ORDER BY AverageOrderValue DESC;

The inner query computes the total order value per invoice for each country. The outer query averages these values by country, revealing the countries with the highest average spending.

**Related Screenshot :-**

* 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 used :-**

WITH CustomerFrequency AS (

SELECT CustomerID,COUNT(DISTINCT InvoiceNo) AS PurchaseFrequency

FROM online\_retail GROUP BY CustomerID

)

SELECT CustomerID,PurchaseFrequency,

CASE

WHEN PurchaseFrequency > 10 THEN 'High Frequency'

WHEN PurchaseFrequency BETWEEN 5 AND 10 THEN 'Medium Frequency'

ELSE 'Low Frequency'

END AS CustomerSegment

FROM CustomerFrequency;

The WITH clause is used to calculate the number of distinct purchases per customer. Then, customers are segmented into high, medium, and low frequency groups based on their purchase counts.

**Related Screenshot :-**

