High Impact Skills Development Program for Gilgit Baltistan

Data mining Module Project

Project Title: Online Retail Segmentation

Name:	**Jahangeer Ali**
Section:	**02 DS-AI**
Roll No:	**GIL-23071**
Project:	**02**
Email:	Jahangeerali.shilwa990@gmail.com
GitHub:	https://github.com/JahangeerAli

Introduction

In the era of digital commerce, understanding customer behavior and preferences has become paramount for businesses to thrive. The "Online Retail Segmentation" project aimed to provide us with practical insights into data mining using SQL, enabling them to decipher valuable patterns and trends within a retail dataset. This report outlines the project's objectives, methodologies, findings, and the skills participants gained during the journey.

Learning Objectives:

The project had three primary learning objectives:

- To grasp the fundamentals of Data Mining.
- To learn how to apply SQL for data mining.
- To implement key mining concepts for customer segmentation and analysis.

Dataset:

The dataset provided encompassed a range of attributes including InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, and Country. This dataset simulated real-world retail transactions, forming the basis for the project's analyses and queries.

Methodologies:

The project is structured to cater to both novice and advanced learners. We embarked on a journey through SQL-driven analyses, starting from fundamental tasks such as defining metadata and understanding distribution of order values, to more complex challenges like customer segmentation by purchase frequency, customer churn analysis, and product affinity assessments.

Findings:

Participants gained insights into various facets of customer behavior and its analysis:

Beginner Queries.

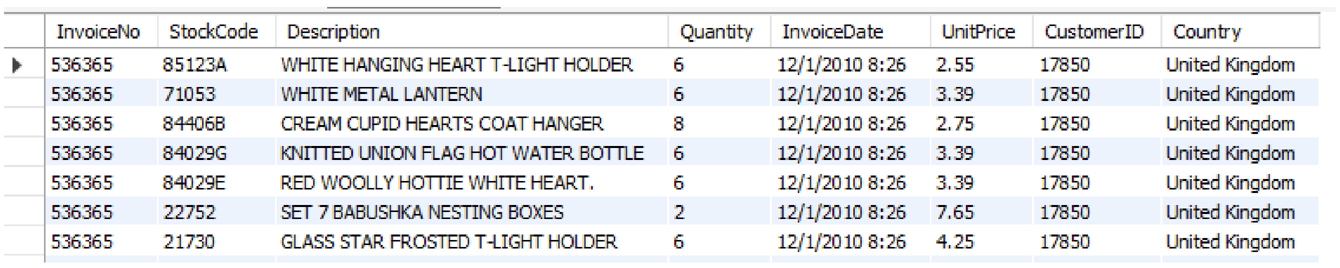
01: Define meta data in mysql workbench or any other SQL tool?

Ans: Metadata refers to the data that provides information about other data. In the context of databases and SQL tools like MySQL Workbench, metadata typically refers to the information about the structure and properties of the database objects, such as tables, columns, indexes, constraints, and more. It helps users understand the organization of the database and aids in querying and managing the data effectively.

Query:

```
show databases;
use retail;
select * from retail;
```

Output:



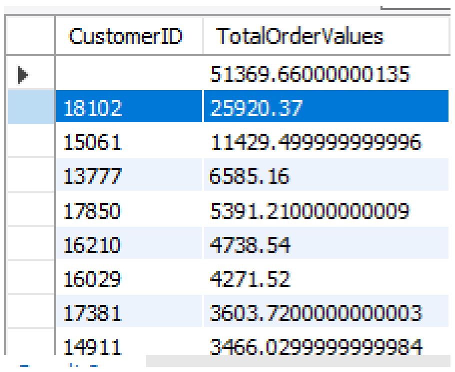
	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
▶	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850	United Kingdom
	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850	United Kingdom
	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850	United Kingdom
	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850	United Kingdom
	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7.65	17850	United Kingdom
	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4.25	17850	United Kingdom

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

Query:

```
Select CustomerID, sum(Quantity* UnitPrice) as TotalOrderValues
From retail
group by CustomerID
order by TotalOrderValues desc;
```

Output:



	CustomerID	TotalOrderValues
▶		51369.66000000135
	18102	25920.37
	15061	11429.499999999996
	13777	6585.16
	17850	5391.210000000009
	16210	4738.54
	16029	4271.52
	17381	3603.7200000000003
	14911	3466.0299999999984

Description:
We learned how to visualize and analyze the spread of order values across the entire dataset, providing an understanding of purchasing trends.
This query calculates the total amount spent by each customer on their purchases. It adds up the cost of each product for each customer, groups the results by customer, and then arranges the list in descending order based on the total spent amount. This helps identify the customers who have spent the most.

03: How many unique products has each customer purchased?

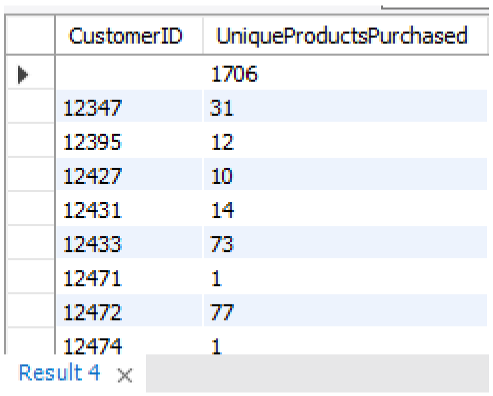
The project helped us identify customers who made purchases and the number of unique products each customer bought, unveiling the diversity of customer preferences.

Query:

```
select CustomerID , count(distinct StockCode) as UniqueProductsPurchased
from retail

group by CustomerID;
```

Output:



CustomerID	UniqueProductsPurchased
	1706
12347	31
12395	12
12427	10
12431	14
12433	73
12471	1
12472	77
12474	1

Description:
This query is used to show each customer's shopping history. It calculates how many different items each customer bought, and it shows the number of unique items each customer purchased. We used grouped by customer to see this information for unique products has each customer purchased in ascending order.

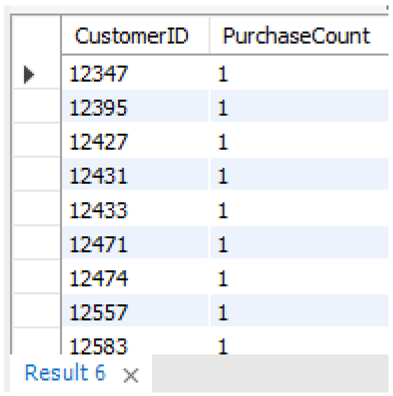
04: Which customers have only made a single purchase from the company?

Through SQL, we were able to identify customers who made only a single purchase, which has implications for customer retention strategies.

Query:

```
select CustomerID, count(distinct InvoiceNo) as PurchaseCount
from retail
group by CustomerID
having PurchaseCount=1;
```

Output:



CustomerID	PurchaseCount
12347	1
12395	1
12427	1
12431	1
12433	1
12471	1
12474	1
12557	1
12583	1

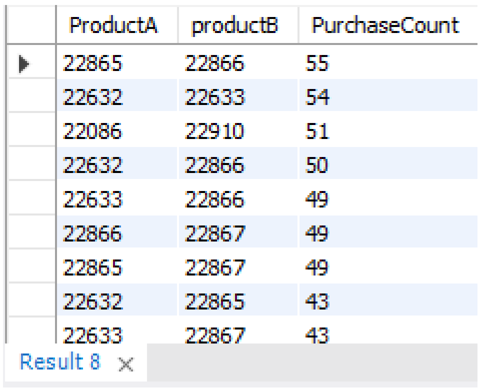
Description:
We used this query to check the customer who has purchased only one item form store and It examines the shopping behavior of each customer. It counts the number of times each customer made a purchase from the store. Then, it only shows the customers who made just one purchase. This helps us identify the customers who have bought only once.

05: Which products are most commonly purchased together by customers in the dataset?
By calculating correlations between product purchases, we discovered which products were commonly bought together, informing potential cross-selling strategies.

Query:

```
select  a.StockCode as ProductA, b.StockCode as productB, count(*) as PurchaseCount
from retail a
join retail b on a.InvoiceNo= b.InvoiceNo and a.StockCode < b.StockCode
group by ProductA, ProductB
order by PurchaseCount desc
limit 10;
```

Output:



ProductA	productB	PurchaseCount
22865	22866	55
22632	22633	54
22086	22910	51
22632	22866	50
22633	22866	49
22866	22867	49
22865	22867	49
22632	22865	43
22633	22867	43

Result 8

Description :

This query is used to check that items which are bought together by customers.. It finds pairs of products and counts how many times they were bought together. The results show the pairs with the most purchases first. The query only prints the top 10 pairs

Advanced Analyses/Advance Queries

We delved into more intricate analyses, including customer segmentation based on purchase frequency, identifying loyal customers, calculating average order values by country, performing churn analysis, and exploring time-based trends.

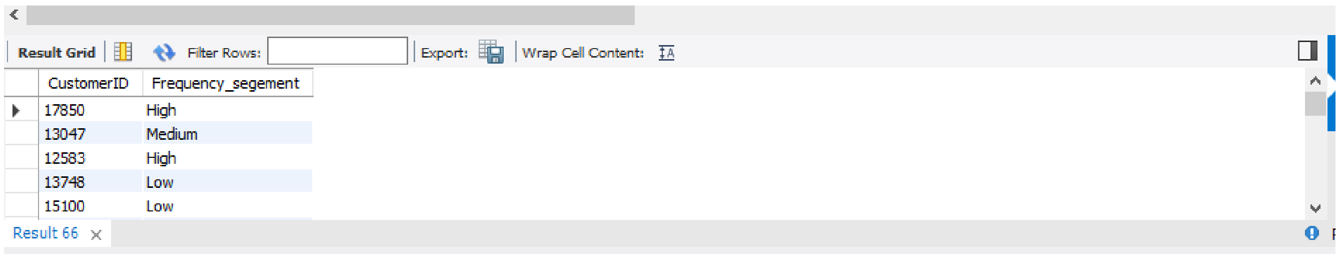
1. Customer Segmentation by Purchase Frequency

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:

```
select CustomerID,
            case
                  when Purchase_Frequency >= 20 then "High"
      when Purchase_Frequency >= 5 then "Medium"
      else "Low"
            end as Frequency_segement
from (
            select CustomerID, count(InvoiceNo) as Purchase_Frequency
    from retail
    group by CustomerID
      )   as sPurchase_Frequency;
```

Output:



CustomerID	Frequency_segement
17850	High
13047	Medium
12583	High
13748	Low
15100	Low

Description:
This query is used to check the shopping frequency of customer ,who is the loyal customer. It categorizes customers based on their shopping frequency. Customers who purchased more than 19 items a lot are called "High" frequency, those who purchased more than 4 items are "Medium," and those who purchased less then 4 tems are "Low." The commands together help understand and group customers based on their shopping habits.

2. Average Order Value by Country

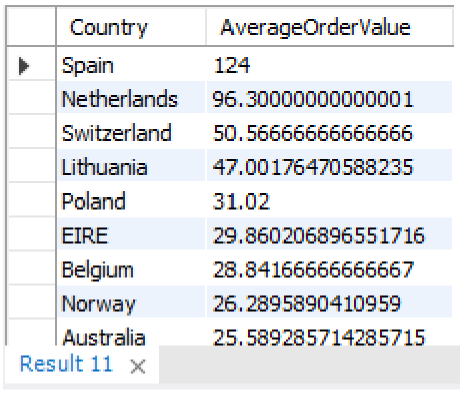
Calculate the average order value for each country to identify where your most valuable customers are located.

Query:

select Country, avg(Quantity * UnitPrice) as AverageOrderValue
from retail
group by Country

order by AverageOrderValue desc;

Output:



Country	AverageOrderValue
Spain	124
Netherlands	96.30000000000001
Switzerland	50.5666666666666
Lithuania	47.00176470588235
Poland	31.02
EIRE	29.860206896551716
Belgium	28.8416666666667
Norway	26.2895890410959
Australia	25.589285714285715

Result 11

Description:

This query is used to gathers information about the countries where customers come from. It calculates the average amount of money people spend on their orders in each country. Then, it arranges the countries based on how much money people spend on average, from the highest to the lowest. This helps identify which countries have customers with higher average spending.

3. Customer Churn Analysis

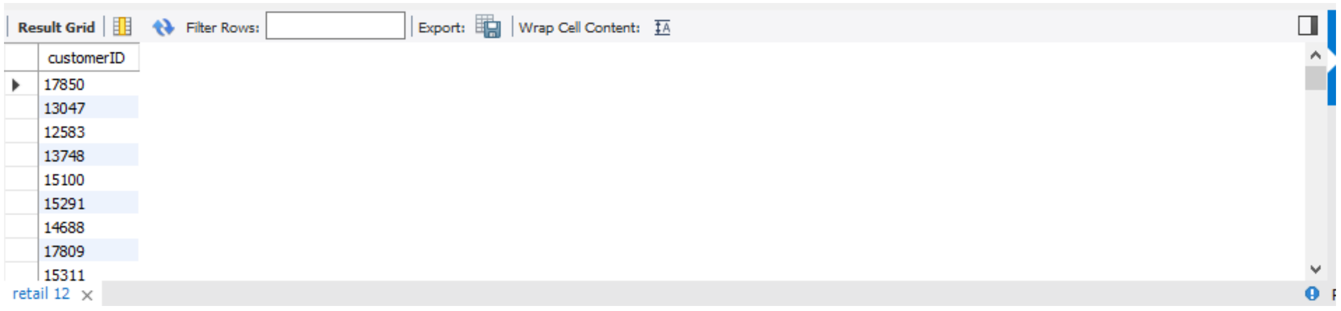
Identify customers who haven't made a purchase in a specific period (e.g., last 6 months) to assess churn.

Query:

select customerID

from retail
where InvoiceNo <= date_sub(now(), interval 6 month)
group by CustomerID
having max(InvoiceDate) < date_sub(now(),interval 6 month);

Output:



customerID
17850
13047
12583
13748
15100
15291
14688
17809
15311

retail 12

Description:

This query is used to finding customers who might not have shopped in the last 6 months. It looks at the records of customers' purchases. If a customer's last purchase was more than 6 months ago, the query includes their customer ID in the results. This helps identify customers who might not have shopped recently.

4. Product Affinity Analysis

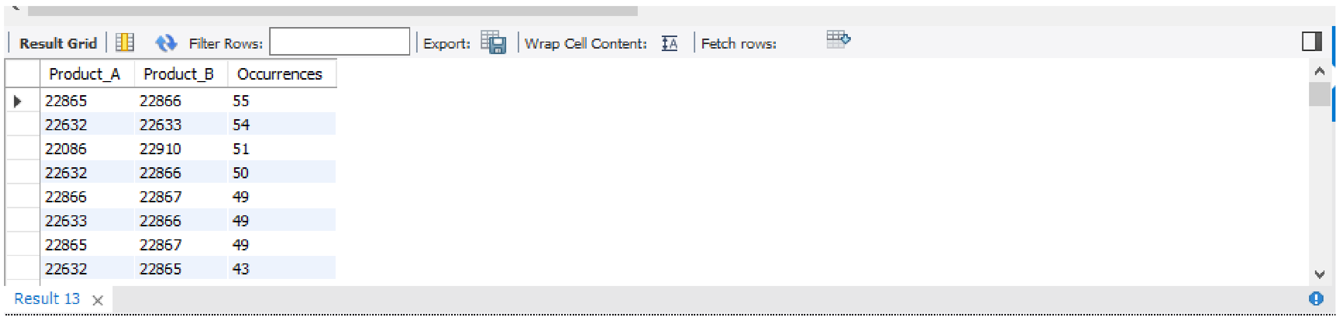
Determine which products are often purchased together by calculating the correlation between product purchases.

Query:

```
create temporary table ProductPairs as
select A.StockCode as Product_A, B.StockCode as Product_B
from retail A
join retail B on A.InvoiceNo = B.InvoiceNo and A.StockCode < B.StockCode;

select Product_A, Product_B, count(*) as Occurrences
from  ProductPairs
group by Product_A, Product_B
order by Occurrences desc
limit 10;
```

Output:



Product_A	Product_B	Occurrences
22865	22866	55
22632	22633	54
22086	22910	51
22632	22866	50
22866	22867	49
22633	22866	49
22865	22867	49
22632	22865	43

Description:
This query is used to find which products are often bought together. First we create a temporary table "ProductPairs". It collects pairs of products from customers' purchases where one product's code is less than the other product's code. Then, the second query, checks these pairs and counts how often each pair of products was bought together. The results show the pairs with the most occurrences first, it shows only first 10 rows. This helps understand which products are commonly purchased together by customers.

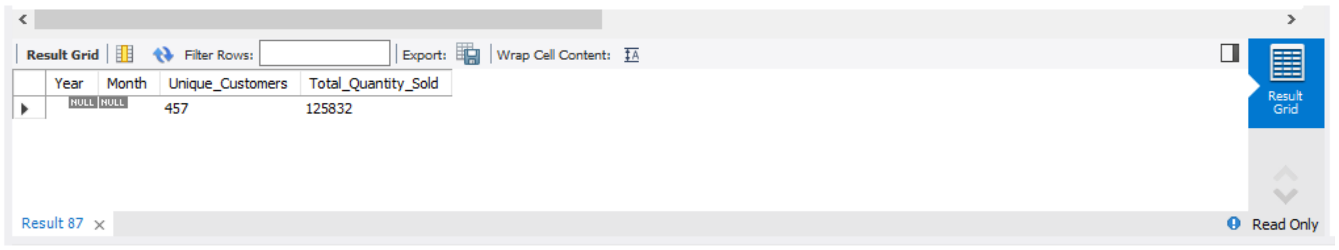
5. Time-based Analysis

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

Query:

```
select
    year(InvoiceDate) as Year,
    month(InvoiceDate) as Month,
    count(distinct CustomerID) as Unique_Customers,
    sum(Quantity) as Total_Quantity_Sold
from retail
group by Year, Month
order by Year, Month;
```

Output:



Year	Month	Unique_Customers	Total_Quantity_Sold
NULL	NULL	457	125832

Description:

This query is used to check the purchasing pattern of customer over time . It groups sales data by year and month, calculating the number of unique customers and the total quantity of items sold in each period. The results show how many customers made purchases and how much was sold in each month, sorted by year and month. This helps analyze sales trends over time and understand customer behavior in different months and years.

Skills Acquired:

Throughout the project, participants honed several critical skills:

SQL Proficiency: We gained hands-on experience in writing complex SQL queries for data mining and analysis.

Data Interpretation: By working with real-world data, participants learned to interpret results and extract meaningful insights.

Segmentation Strategies: Participants grasped the importance of customer segmentation in tailoring marketing and business strategies for enhanced outcomes.

Problem-Solving: The diverse challenges presented in the project encouraged us to think critically and devise effective solutions using MY SQL.

Conclusion:

The "Online Retail Segmentation" project successfully achieved its objectives by providing us with practical exposure to data mining techniques using My SQL. Through a series of analyses and queries, we not only improved our technical skills but also gained a deep understanding of customer behavior analysis, segmentation strategies, and their relevance in the realm of online retail. The project underscores the importance of leveraging data mining tools to unlock valuable insights, empowering businesses to make informed decisions that drive customer satisfaction and growth.

***** The End*****