

Database Design and Implementation Project

Academic Year: 2023/24

Batch 2023.

**Group Members:**

* Aziz Ullah
* Syed Muhammad Khizer
* Muhammad Rohaan
* Laiba Asim
* Asjad Kashif
* Bakhtawar Irfan

**Lab Instructor:**

Miss. Hafsa

**Contents**

[1](#_30j0zll) Requirement Analysis (15%) 5

[1.1](#_1fob9te) Description of the three websites chosen 5

[1.1.1](#_3znysh7) http://www.sainsbury.co.uk (EXAMPLE ONLY) 5

[1.1.2](#_2et92p0) http://www.asda.com (EXAMPLE ONLY) 5

[1.1.3](#_tyjcwt) http://www.tesco.com (EXAMPLE ONLY) 5

[1.2](#_3dy6vkm) List of data fields (Entities and their attributes) 5

[1.2.1](#_1t3h5sf) List of data fields from website 1 6

[1.2.2](#_4d34og8) List of data fields from website 2 6

[1.2.3](#_2s8eyo1) List of data fields from website 3 6

[1.3](#_17dp8vu) Finalised List 6

[1.3.1](#_19c6y18) The generic list(Optional) 6

[2](#_3rdcrjn) Database design (25%) 6

[2.1](#_26in1rg) Entity Relationship Modelling 6

[2.1.1](#_lnxbz9) Initial Entity Relationship Model 7

[2.1.2](#_35nkun2) Extended Entity Relationship Model 7

[2.3](#_1ksv4uv) Database Schema 7

[3.](#_44sinio) Database implementation (10%) 8

[4.](#_2jxsxqh) SQL Queries (50%) 8

[4.1](#_z337ya) Query 1 9

[4.1.1](#_3j2qqm3) For what purpose will this query be used in business terms? 9

[4.1.2](#_1y810tw) Query in natural language 9

[4.1.3](#_4i7ojhp) SQL Code and output 9

[4.1.4](#_2xcytpi) Explain the output of the data (was this what was predicted?) 9

[4.2](#_1ci93xb) Query 2 9

[4.2.1](#_3whwml4) For what purpose will this query be used in business terms? 9

[4.2.2](#_2bn6wsx) Query in natural language 9

[4.2.3](#_qsh70q) SQL Code and output 9

[4.2.4](#_3as4poj) Explain the output of the data (was this what was predicted?) 9

[4.3](#_1pxezwc) Query 3 9

[4.3.1](#_49x2ik5) For what purpose will this query be used in business terms? 9

[4.3.2](#_2p2csry) Query in natural language 9

[4.3.3](#_147n2zr) SQL Code and output 9

[4.3.4](#_3o7alnk) Explain the output of the data (was this what was predicted?) 9

[4.4](#_23ckvvd) Query 4 10

[4.4.1](#_ihv636) For what purpose will this query be used in business terms? 10

[4.4.2](#_32hioqz) Query in natural language 10

[4.4.3](#_1hmsyys) SQL Code and output 10

[4.4.4](#_41mghml) Explain the output of the data (was this what was predicted?) 10

[4.5](#_2grqrue) Query 5 10

[4.5.1](#_vx1227) For what purpose will this query be used in business terms? 10

[4.5.2](#_3fwokq0) Query in natural language 10

[4.5.3](#_1v1yuxt) SQL Code and output 10

[4.5.4](#_4f1mdlm) Explain the output of the data (was this what was predicted?) 10

[5.](#_2u6wntf) References 11

# Requirement Analysis (15%)

## 1.1 Description of the three websites chosen

### 1.1.1 <http://www.priceoye.pk>

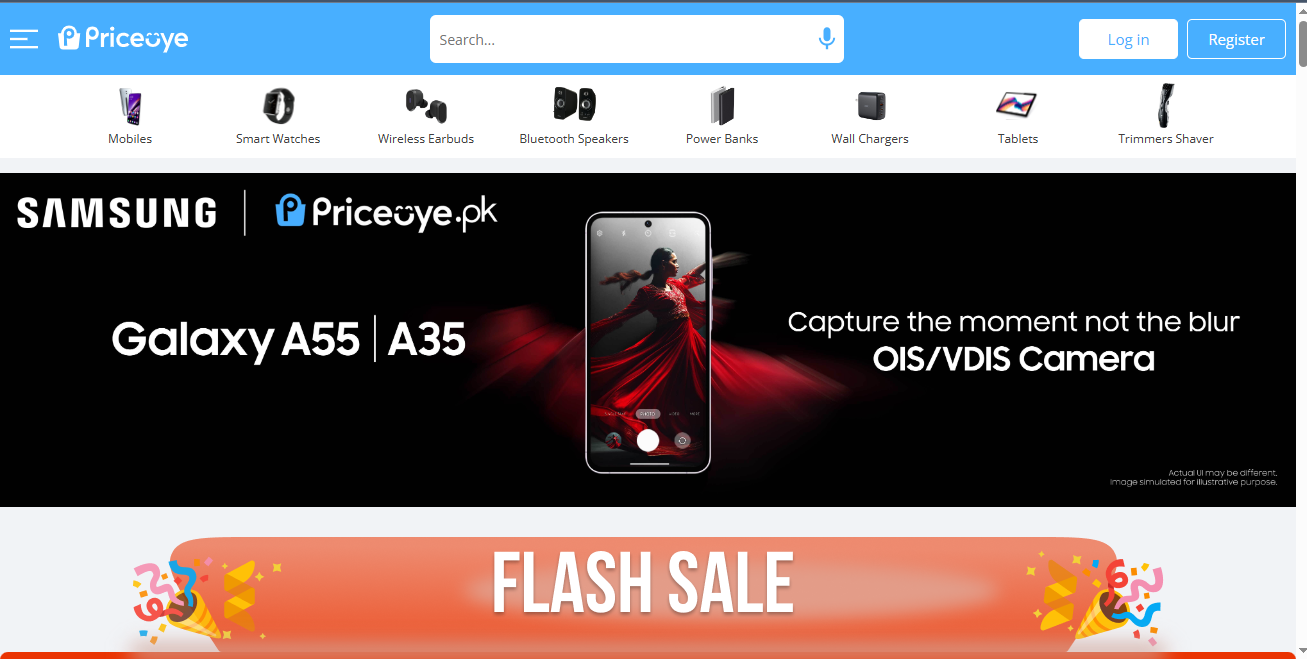
1.1.2 <https://www.caremaniac.com/>

### 1.1.3 https://shoprusset.com

## 1.2 List of data fields (Entities and their attributes)

### 1.2.1 List of data fields from website 1

Website 1 🡪 PriceOye.PK



**Entities:**

1. **Products**

* Product\_id
* Name
* Brand\_id
* Category\_ID
* Price
* Quantity
* Rating

1. **Users:**

* Users\_Id
* Username
* Email
* Password
* Address\_id
* Phone\_number

1. **Addresses**

* Addresses\_ID
* Users\_id
* Name
* Address
* City
* Country
* Postal\_code

1. **Orders:**

* Order\_id
* User\_id
* Order\_Date
* Shipping\_Address\_ID
* Status
* Traking\_ID

1. **Order\_products:**

* OrderProduct\_id
* Order\_ID
* Product\_ID
* Quantity

1. **CartItems:**

* CartItemID
* CartID
* ProductID
* Quantity

1. **Cart**

* CartID
* USerID

1. **Brand**

* Brand\_ID
* Name

1. **Categories**

* Category\_ID
* Name

1. **Transactions**

* Transactions\_ID
* User\_ID FK
* Order\_ID
* Transaction\_date

1. **Totalamount:**

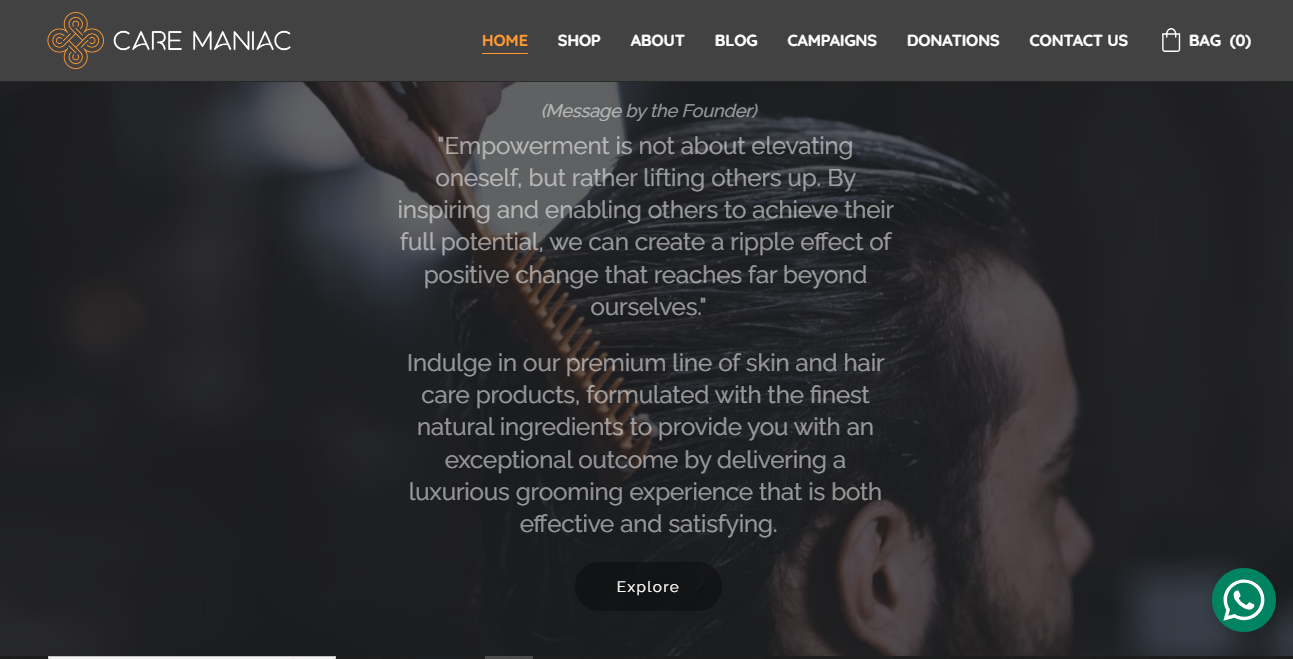
* OrderID
* Total

1. **Favourites:**

* FavouriteID
* User\_ID
* Product\_ID
* DateAdded

### 1.2.2 List of data fields from website 2

Website 2 🡪 caremaniac.com



**1. Product Table**

* ProductID
* ProductName
* CategoryID
* Price

**2. Category Table**

* CategoryID
* CategoryName

**3. Customer Table**

* CustomerID
* CustomerName
* Email

1. **Users:**

* Users\_Id
* Username
* Email
* Password
* Address\_id
* Phone\_number

1. **Addresses**

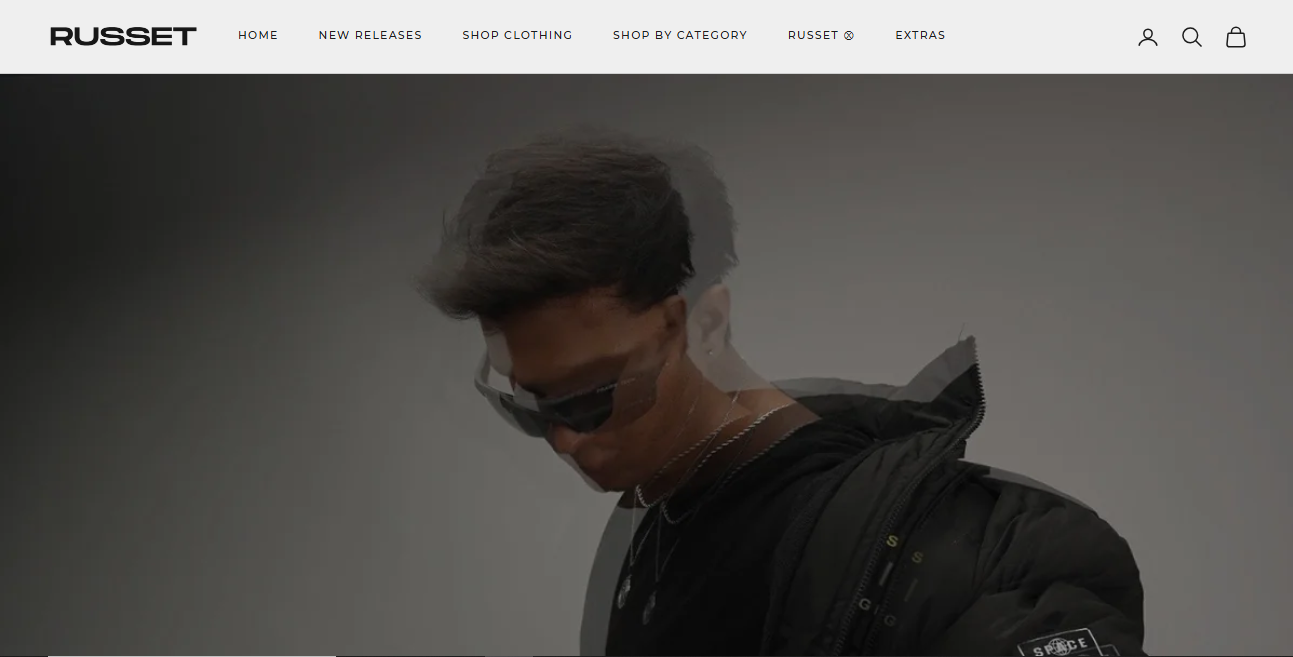
* Addresses\_ID
* Users\_id
* Name
* Address
* City
* Country
* Postal\_code

1. **Orders:**

* Order\_id
* User\_id
* Order\_Date
* Shipping\_Address\_ID
* Status
* Traking\_ID

### 1.2.3 List of data fields from website 3

Website 3 🡪 <https://shoprusset.com>



**1. Product Table**

* ProductID
* ProductName
* CategoryID
* Size
* Description

**2. Category Table**

* CategoryID,
* Category Name

**3. Customer Table**

* CustomerID
* CustomerName
* Email
* Address
* Phone\_no

**4.Order**

* OrderID
* CustomerID
* OrderDate

**5. OrderDetails :**

* OrderDetailID
* OrderID
* ProductID
* Quantity

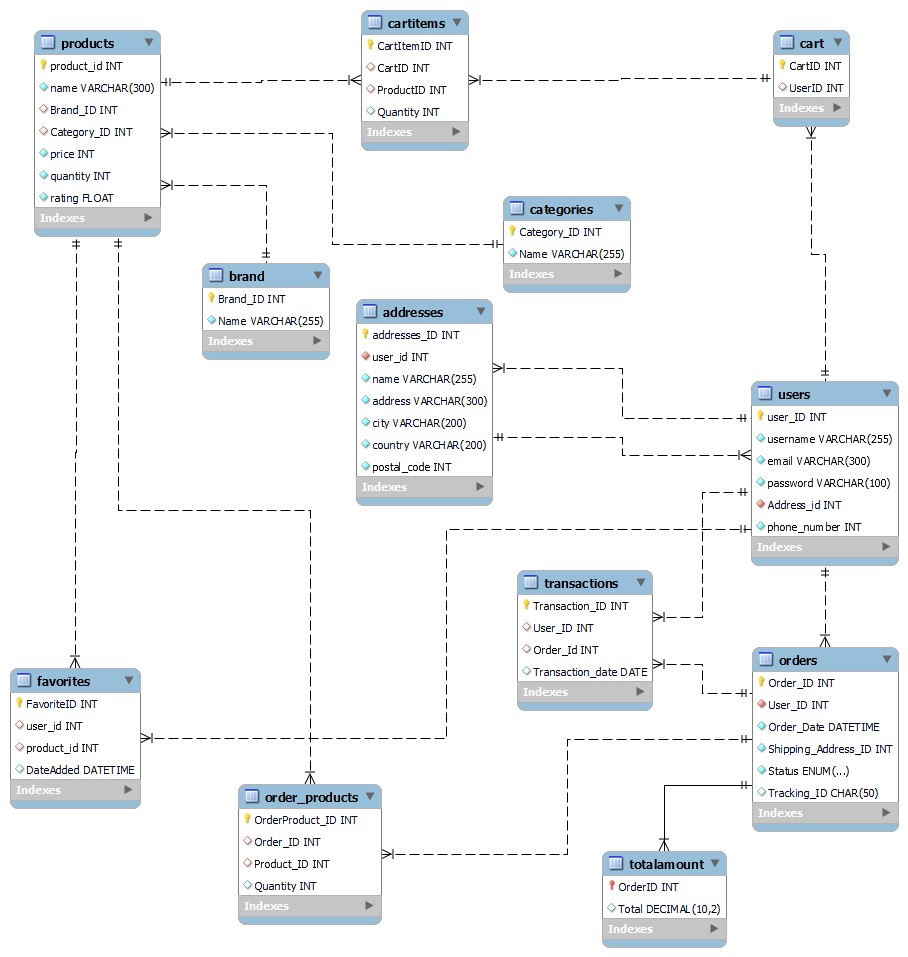
## 1.3 Finalised List

[Delete this text: include one list of the attributes you have chosen to take forward to your diagram. For better marks, consider adding self-suggested or identified hidden entities and attributes.]

# Database design (25%)

## 2.1 Entity Relationship Modelling

### 2.1.2 Extended Entity Relationship Model



## Database Schema

**Website 1:**

**1.Products**

* Product\_id INT / PRIMARY KEY
* Name VARCHAR / NOT NULL
* Brand\_id INT / FOREIGN KEY
* Category\_ID INT / FOREIGN KEY
* Price INT
* Quantity INT
* Rating FLOAT

**2.Users:**

* Users\_Id INT / PRIMARY KEY
* Username VARCHAR / NOT NULL
* Email VARCHAR / NOT NULL / UNIQUE
* Password VARCHAR / NOT NULL
* Address\_id INT / FOREIGN KEY
* Phone\_number INT

**3.Addresses**

* Addresses\_ID PK INT / PRIMARY KEY
* Users\_id FK INT / FOREIGN KEY
* Name VARCHAR / NOT NULL
* Address VARCHAR / NOT NULL
* City VARCHAR / NOT NULL
* Country VARCHAR / NOT NULL
* Postal\_code INT

**4.Orders:**

* Order\_id INT / PRIMARY KEY
* User\_id INT / FOREIGN KEY
* Order\_Date DATETIME
* Shipping\_Address\_ID INT
* Status ENUM(…)
* Traking\_ID CHAR(50)

**5.Order\_products:**

* OrderProduct\_id INT / PRIMARY KEY
* Order\_ID INT / FOREIGN KEY
* Product\_ID FK INT / FOREIGN KEY
* Quantity INT

**6.CartItems:**

* CartItemID INT / PRIMARY KEY
* CartID INT / FOREIGN KEY
* ProductID INT / FOREIGN KEY
* Quantity INT

**7.Cart**

* CartID INT / PRIMARY KEY
* USerID INT / FOREIGN KEY

**8.Brand**

* Brand\_ID INT / PRIMARY KEY
* Name VARCHAR

**9.Categories**

* Category\_ID INT / PRIMARY KEY
* Name VARCHAR

**10.Transactions**

* Transactions\_ID INT / PRIMARY KEY
* User\_ID INT / FOREIGN KEY
* Order\_ID INT / FOREIGN KEY
* Transaction\_date DATE

**11.Totalamount:**

* OrderID INT / PRIMARY KEY
* Total DECIMAL(10,2)

1. **Favourites:**

* FavouriteID INT / PRIMARY KEY
* User\_ID INT / FOREIGN KEY
* Product\_ID INT / FOREIGN KEY
* DateAdded DATETIME

**Website 2:**

**1. Product Table**

* ProductID INT / PRIMARY KEY
* ProductName VARCAHR
* CategoryID INT / FOREIGN KEY
* Price FLOAT

**2. Category Table**

* CategoryID INT / PRIMARY KEY
* CategoryName VARCAHR

**3. Customer Table**

* CustomerID INT / PRIMARY KEY
* CustomerName VARCHAR
* Email VARCHAR

1. **Users:**

* Users\_Id INT / PRIMARY KEY
* Username VARCHAR
* Email VARCHAR
* Password VARCHAR
* Address\_id INT / FOREIGN KEY
* Phone\_number INT

1. **Addresses**

* Addresses\_ID INT / PRIMARY KEY
* Users\_id INT / FOREIGN KEY
* Name VARCHAR
* Address VARCHAR
* City VARCHAR
* Country VARCHAR
* Postal\_code INT

1. **Orders:**

* Order\_id INT / PRIMARY KEY
* User\_id INT / FOREIGN KEY
* Order\_Date DATE
* Shipping\_Address\_ID INT
* Status ENUM(..)
* Traking\_ID CHAR(50)

**Website 3:**

**1. Product Table**

* ProductID INT PRIMARY KEY
* ProductName VARCHAR (255) NOT NULL
* CategoryID INT
* Size VARCHAR (50),
* Description TEXT

**2. Category Table**

* CategoryID INT PRIMARY KEY
* Category\_Name VARCHAR (255) NOT NULL

**3. Customer Table**

* CustomerID INT PRIMARY KEY
* CustomerName VARCHAR (255) NOT NULL
* Email VARCHAR (255) UNIQUE NOT NULL
* Address VARCHAR (255)
* Phone\_no VARCHAR

**4.Order**

* OrderID INT PRIMARY KEY
* CustomerID INT
* OrderDate DATE

**5. OrderDetails :**

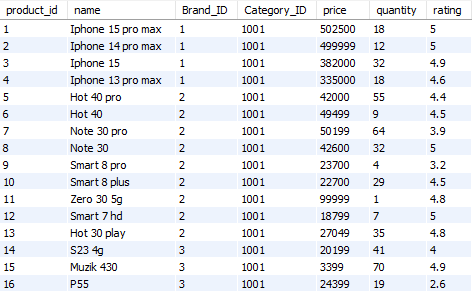
* OrderDetailID INT PRIMARY KEY
* OrderID INT
* ProductID INT
* Quantity INT

# 3. Database implementation (10%)

**Product Table:**

A screen shot of a computer code

Description automatically generated



**OrderProduct Table:**

**A screen shot of a computer code

Description automatically generated**

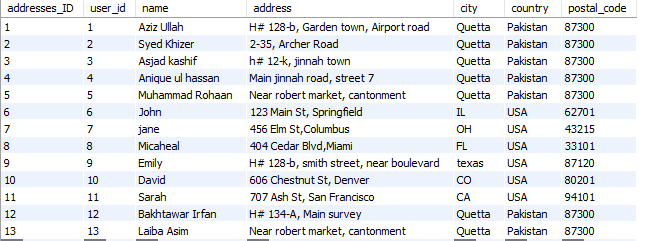
**A screenshot of a table

Description automatically generated**

**Addresses Table:**

**A screenshot of a computer code

Description automatically generated**

****

**Brand Table:**

**A screenshot of a computer

Description automatically generated**

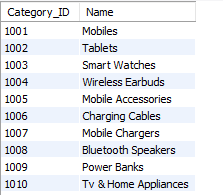
**A screenshot of a computer

Description automatically generated**

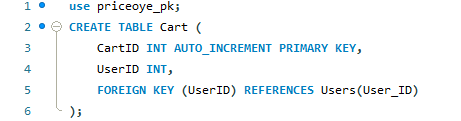
**Categories Table:**

**A close-up of a computer code

Description automatically generated**

****

**Cart Table:**

****

****

**CartItems Table:**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

**Favourites Table:**

**A screen shot of a computer code

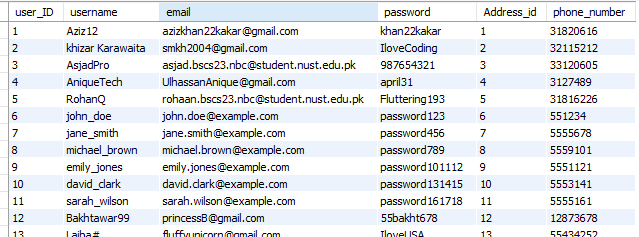
Description automatically generated**

**A screenshot of a table

Description automatically generated**

**Users Table:**

****

****

**Order Table:**

**A screen shot of a computer code

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Transaction Table:**

A screen shot of a computer code

Description automatically generated

A screenshot of a data

Description automatically generated

**TotalAmount Table:**

A close-up of a computer code

Description automatically generated

A screenshot of a calculator

Description automatically generated

# 4. SQL Queries (50%)

## 4.1 Query 1

### 4.1.1 For what purpose will this query be used in business terms?

This SQL query is used to calculate the total sales for each user. Specifically, it retrieves the username of each user and the sum of the total amounts from their orders. The results show the total sales associated with each user.

### 4.1.2 Query in natural language

### 4.1.3 SQL Code and output

SELECT

u.username AS Username,

SUM(ta.Total) AS TotalSales

FROM

Users u

JOIN

Orders o ON u.user\_ID = o.User\_ID

JOIN

TotalAmount ta ON o.Order\_ID = ta.OrderID

GROUP BY

u.username;

A screenshot of a computer

Description automatically generated

### 4.1.4 Explain the output of the data (was this what was predicted?)

The output of the data shows a list of usernames and their corresponding total sales. Each row represents a user, and the TotalSales column shows the sum of all the sales (or total amounts) associated with that user's orders.

## 4.2 Query 2

### 4.2.1 For what purpose will this query be used in business terms?

This query is used to calculate the average product rating for each brand. Businesses can use this information to assess and compare the performance of different brands based on customer feedback. It can help in identifying brands with high or low customer satisfaction and making decisions about product stocking, marketing strategies, or potential improvements.

### 4.2.2 Query in natural language

This SQL query retrieves the average rating of products for each brand. It lists the name of each brand along with the average rating of its products.

### 4.2.3 SQL Code and output

SELECT

b.Name AS BrandName,

AVG(p.rating) AS AverageRating

FROM

Brand b

JOIN

Products p ON b.Brand\_ID = p.Brand\_ID

GROUP BY

b.Name;

A screenshot of a computer

Description automatically generated

### 4.2.4 Explain the output of the data (was this what was predicted?)

The output of the data displays the average product rating for each brand. Each row represents a brand, and the AverageRating column shows the average of all ratings for products associated with that brand.

this is likely what was predicted:

## 4.3 Query 3

### 4.3.1 For what purpose will this query be used in business terms?

This query is used to analyze the distribution and financial impact of orders based on their status. Businesses can use this information to understand how many orders are in different statuses (such as 'Pending', 'Shipped', 'Cancelled') and the total amount of money involved in each status category. This can help in identifying bottlenecks, understanding revenue at different stages, and improving order processing efficiency.

### 4.3.2 Query in natural language

This SQL query retrieves the count and total amount of orders for each order status. It lists the status of orders, the number of orders in each status, and the sum of total amounts for those orders.

### 4.3.3 SQL Code and output

SELECT

o.Status AS OrderStatus,

COUNT(o.Order\_ID) AS OrderCount,

SUM(ta.Total) AS TotalAmount

FROM

Orders o

JOIN

TotalAmount ta ON o.Order\_ID = ta.OrderID

GROUP BY

o.Status;

A screenshot of a data

Description automatically generated

### 4.3.4 Explain the output of the data (was this what was predicted?)

he output of the data shows the count and total amount of orders for each order status. Each row represents an order status, and the OrderCount column shows the number of orders in that status, while the TotalAmount column shows the sum of total amounts for those orders.

This output likely matches the predictions:

## 4.4 Query 4

### 4.4.1 For what purpose will this query be used in business terms?

This query is used to generate a detailed report of orders placed by each user, including the products in those orders and their quantities, along with the total amount for each order. Businesses can use this information to track customer purchasing behavior, analyze order details, and gain insights into product popularity and sales trends. It can also help in identifying high-value customers and managing customer relationships.

### 4.4.2 Query in natural language

This SQL query retrieves detailed information about orders placed by users. It lists each order along with the username of the user who placed the order, the order ID, order date, product names included in the order, the quantity of each product, and the total amount of the order. The results are ordered by username and order date.

### 4.4.3 SQL Code and output

SELECT

u.username AS Username,

o.Order\_ID AS OrderID,

o.Order\_Date AS OrderDate,

p.name AS ProductName,

op.Quantity AS Quantity,

ta.Total AS OrderTotal

FROM

Users u

JOIN

Orders o ON u.user\_ID = o.User\_ID

JOIN

Order\_Products op ON o.Order\_ID = op.Order\_ID

JOIN

Products p ON op.Product\_ID = p.product\_id

JOIN

TotalAmount ta ON o.Order\_ID = ta.OrderID

ORDER BY

u.username, o.Order\_Date;

A screenshot of a table

Description automatically generated

### 4.4.4 Explain the output of the data (was this what was predicted?)

The output of the data provides a detailed breakdown of orders placed by users. Each row represents a product in an order, showing the username of the user who placed the order, the order ID, order date, product name, quantity of the product in the order, and the total amount of the order.

This output is likely what was predicted

## 4.5 Query 5

### 4.5.1 For what purpose will this query be used in business terms?

This query is used to determine how many favorite items each user has saved in their favorites list. Businesses can use this information to understand user engagement and preferences. High counts of favorites can indicate active and engaged users, while low counts might suggest users who are less involved. This data can help in personalizing marketing efforts, improving user retention strategies, and enhancing the overall user experience.

### 4.5.2 Query in natural language

This SQL query retrieves the number of favorite items for each user. It lists each user and counts how many items they have marked as favorites.

### 4.5.3 SQL Code and output

USE priceoye\_pk;

SELECT

USER\_ID,

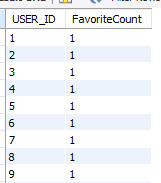
COUNT(USER\_ID) AS FavoriteCount

FROM

favorites

GROUP BY

USER\_ID;



### 4.5.4 Explain the output of the data (was this what was predicted?)

The output of the data provides a summary of how many items each user has saved to their favorites list. Each row represents a user, with the USER\_ID column showing the unique identifier of the user and the FavoriteCount column showing the total number of items that user has marked as a favorite.

This output likely matches what was predicted

# 5. References

https://www.geeksforgeeks.org/dbms/