**Instructions:**

Please share your answers filled in line in the Word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: Vishvash C Batch ID:** 23012024

**Topic: Introduction to Database**

1. Create a Supermart\_DB with the tables created from the datasets shared (Customer.csv, Sales.csv and Product.csv files)
   1. Create a new database in your database management system, and name it Supermart\_DB.

CREATE DATABASE Supermart\_DB;

* 1. Create a new table called "customers" in the Supermart\_DB database

USE Supermart\_DB;

CREATE TABLE customers (

CustomerID VARCHAR(255) PRIMARY KEY,

CustomerName VARCHAR(255),

Segment VARCHAR(255),

Age INT,

Country VARCHAR(255),

City VARCHAR(255),

State VARCHAR(255),

PostalCode VARCHAR(255),

Region VARCHAR(255)

);

* 1. Load the data from the Customer.csv file into the customers table

load data infile 'C:/ProgramData/MySQL/MySQL Server 8.3/Uploads/Copy of Customer.csv'

into table customers

fields terminated by ','

enclosed by ''''

lines terminated by '\n'

ignore 1 rows;

* 1. Create a new table called "products" in the Supermart\_DB database

CREATE TABLE products (

ProductID VARCHAR(255) PRIMARY KEY,

Category VARCHAR(255),

SubCategory VARCHAR(255),

ProductName VARCHAR(255)

);

* 1. Load the data from the Product.csv file into the products table

load data infile 'C:/ProgramData/MySQL/MySQL Server 8.3/Uploads/Copy of Product.csv'

into table products

fields terminated by ','

enclosed by ''''

lines terminated by '\n'

ignore 1 rows;

* 1. Create a new table called "sales" in the Supermart\_DB database

CREATE TABLE sales (

OrderLine INT PRIMARY KEY,

OrderID VARCHAR(255),

OrderDate DATE,

ShipDate DATE,

ShipMode VARCHAR(255),

CustomerID varchar(255),

ProductID varchar(255),

Sales DECIMAL(10, 2),

Quantity INT,

Discount DECIMAL(5, 2),

Profit DECIMAL(10, 2)

);

* 1. Load the data from the Sales.csv file into the sales table

LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server 8.3/Uploads/Copy of Sales.csv'

INTO TABLE sales

FIELDS TERMINATED BY ','

ENCLOSED BY '\''

LINES TERMINATED BY '\n'

IGNORE 1 ROWS

(@OrderLine, @OrderID, @OrderDate, @ShipDate, @ShipMode, @CustomerID, @ProductID, @Sales, @Quantity, @Discount, @Profit)

SET OrderLine = @OrderLine,

OrderID = @OrderID,

OrderDate = STR\_TO\_DATE(@OrderDate, '%d-%m-%Y'),

ShipDate = STR\_TO\_DATE(@ShipDate, '%d-%m-%Y'),

ShipMode = @ShipMode,

CustomerID = @CustomerID,

ProductID = @ProductID,

Sales = @Sales,

Quantity = @Quantity,

Discount = @Discount,

Profit = @Profit;

**SELECTION OPERATORS:- (FILTERING):- in, like, between**

**Note:** use products, customers and sales table

1. Define the relationship between the tables using constraints/keys.

alter table sales modify OrderLine int primary key;

alter table sales add foreign key (CustomerID) references customers(CustomerID);

alter table sales add foreign key (ProductID) references products(ProductID);

1. In the database Supermart \_DB, find the following:
2. Get the list of all the cities where the region is north or east without any duplicates using the IN statement.

SELECT DISTINCT City

FROM customers

WHERE Region IN ('North', 'East');

1. Get the list of all orders where the ‘sales’ value is between 100 and 500 using the BETWEEN operator.

SELECT \*

FROM sales

WHERE Sales BETWEEN 100 AND 500;

1. Get the list of customers whose last name contains only 4 characters using LIKE.

SELECT \*

FROM customers

WHERE CustomerName LIKE '\_\_\_\_';

**SELECTION OPERATORS:- ordering**

1. Retrieve all orders where the ‘discount’ value is greater than zero ordered in descending order basis ‘discount’ value

SELECT \*

FROM sales

WHERE Discount > 0

ORDER BY Discount DESC;

1. Limit the number of results in the above query to the top 10.

SELECT \*

FROM sales

WHERE Discount > 0

ORDER BY Discount DESC

LIMIT 10;

**Aggregate operators:-**

1. Find the sum of all ‘sales’ values.

SELECT SUM(Sales) AS TotalSales

FROM sales;

1. Find count of the number of customers in the north region with ages between 20 and 30

SELECT COUNT(\*) AS CustomerCount

FROM customers

WHERE Region = 'North' AND Age BETWEEN 20 AND 30;

1. Find the average age of east region customers

SELECT AVG(Age) AS AverageAge

FROM customers

WHERE Region = 'East';

1. Find the minimum and maximum aged customers from Philadelphia

SELECT MIN(Age) AS MinAge, MAX(Age) AS MaxAge

FROM customers

WHERE City = 'Philadelphia';

**GROUP BY OPERATORS:-**

1. Create a display with the information below for each product ID.
2. Total sales (in $) order by this column in descending
3. Total sales quantity
4. The number of orders
5. Max Sales value
6. Min Sales value
7. Average sales value

SELECT

ProductID,

SUM(Sales) AS TotalSales,

SUM(Quantity) AS TotalQuantity,

COUNT(\*) AS NumberOfOrders,

MAX(Sales) AS MaxSales,

MIN(Sales) AS MinSales,

AVG(Sales) AS AverageSales

FROM sales

GROUP BY ProductID

ORDER BY TotalSales DESC;

1. Get the list of product ID’s where the quantity of product sold is greater than 10

SELECT ProductID

FROM sales

GROUP BY ProductID

HAVING SUM(Quantity) > 10;