**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: N.Harsha vardhan Batch ID:nharsha569103528**

**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.
   2. Create a new table called "customers" in the Supermart\_DB database
   3. Load the data from the Customer.csv file into the customers table
   4. Create a new table called "products" in the Supermart\_DB database
   5. Load the data from the Product.csv file into the products table
   6. Create a new table called "sales" in the Supermart\_DB database
   7. Load the data from the Sales.csv file into the sales table

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

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

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

Answers:

Adding the constraints for customer and product tables:

alter table customer modify column CustomerID varchar(50);

alter table customer

add CONSTRAINT primary key (CustomerID);

alter table product modify column ProductID varchar(50);

alter table product

add CONSTRAINT primary key (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.

Answer:

select \* from customer where Region IN (Region LIKE 'north%' or Region LIKE 'east%');

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

Answer:

select \* from sales where Sales between 100 and 500;

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

Answer:

SELECT LEFT(CustomerName, 4) AS ExtractString

FROM Customer ;

SELECT RIGHT(CustomerName, 4) AS ExtractString

FROM Customer;

SELECT \* FROM Customers where customer\_name LIKE ‘%\_\_\_\_’;

**SELECTION OPERATORS:- ordering**

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

Answer:

select \* from sales where Discount>0 order by Discount desc;

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

Answer:

select \* from sales where Discount>0 order by Discount desc limit 10;

**Aggregate operators:-**

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

Answer:

select sum(sales) from sales;

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

Answer:

select count(\*) from customer where (Region IN('north')) and (age between 20 and 30);

1. Find the average age of east region customers

Answer:

select avg(age) from customer where Region in ('east');

1. Find the minimum and maximum aged customers from Philadelphia

Answer:

select min(age),max(age) from customer where City in ('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

Answer:

select count(Sales) as count\_sales ,ProductID

from sales

group by ProductID

order by sum(sales) desc

limit 0, 1000;

1. Total sales quantity

Answer:

select ProductID ,count(Quantity) as quantity\_count

from sales

group by ProductID

order by sum(Quantity) desc

limit 0, 1000;

1. The number of orders

Answer:

select ProductID ,count(orderID) as order\_count

from sales

group by ProductID

order by count(orderID) desc

limit 0, 1000;

1. Max Sales value

Answer:

select distinct ProductID ,max(Sales) as max\_sales

from sales

group by ProductID

order by max(Sales) desc

limit 0, 1000;

1. Min Sales value

Answer:

select distinct ProductID ,min(Sales) as minimum\_sales

from sales

group by ProductID

order by min(Sales) desc

limit 0, 1000;

1. Average sales value

Answer:

select distinct ProductID ,avg(Sales) as average\_sales

from sales

group by ProductID

order by avg(Sales) desc

limit 0, 1000;

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

Answer:

select ProductID ,sum(Quantity) as total\_quantity

from sales

where Quantity >10

group by ProductID

order by total\_quantity desc

limit 1000 ;