

SUPERMART DATABASE

1. What are the columns in the "customer" table?
SELECT * FROM customer;
2. How many customers are there in the database?
**SELECT customer_name, customer_id
FROM customer;**

793 customers
3. List all the products in the "product" table.
SELECT * FROM product;
4. What are the distinct categories of products available?
**SELECT DISTINCT category
FROM product;**
5. How many sales records are there in the "sales" table?
**SELECT sales, product_id
FROM sales;**

9994 sales
6. Show the sales records for a specific order ID.
**SELECT *
FROM sales
WHERE order_id = 'CA-2016-152156';**
7. Which products were sold with a quantity greater than 10?
**SELECT product_id
FROM sales
WHERE quantity > 10;**
8. Find all customers who are from a specific country.
**SELECT customer_name, country
FROM customer
WHERE country = 'United States';**
9. List all products with their corresponding categories and sub-categories
**SELECT category, sub_category
FROM product;**
10. Which products have a sales value greater than \$1000?
**SELECT sales, product_id
FROM sales
WHERE sales > 1000;**
11. Find the total profit for each order.
**SELECT order_id,
SUM (profit) AS total_profit
FROM sales
GROUP BY order_id;**
12. List all customers from a specific region and state
**SELECT customer_id, region, state
FROM customer**

```
WHERE region = 'West'  
AND state = 'California';
```

13. How many products are there in each category?

```
SELECT COUNT (category), category  
FROM product  
GROUP BY category;
```

14. Find the total sales value for each category.

```
SELECT  
Category,  
SUM(Sales) AS Total_Sales  
FROM product  
INNER JOIN sales  
ON product.Product_ID = sales.Product_ID  
GROUP BY Category;
```

15. Show the customer names and their corresponding segments.

```
SELECT customer_name, segment  
FROM customer;
```

16. Find the products that were not sold yet (with zero quantity)

```
SELECT  
a.product_id,  
b.quantity,  
b.sales  
FROM product AS a  
LEFT JOIN sales AS b  
ON a.product_id = b.product_id  
WHERE b.sales IS NULL  
GROUP BY a.product_id, b.quantity, b.sales
```

17. List the products and their sales quantities ordered by sales quantity in descending order.

```
SELECT  
Product_Name,  
Quantity  
FROM product p  
INNER JOIN sales s  
ON p.Product_ID = s.Product_ID  
ORDER BY Quantity DESC;
```

18. Find the top 5 customers who made the highest purchases (based on sales value).

```
SELECT c.customer_id, Customer_Name,  
SUM(Sales) AS Total_Sales  
FROM customer c  
INNER JOIN sales s  
ON c.customer_id = s.customer_id  
GROUP BY c.customer_id, Customer_Name  
ORDER BY Total_Sales DESC LIMIT 5;
```

19. Find the products with the highest and lowest sales values.

```
SELECT Product_Name,  
MAX(Sales) AS Highest_Sales,  
MIN(Sales) AS Lowest_Sales  
FROM product p INNER JOIN sales s ON p.Product_ID = s.Product_ID  
group by product_name  
ORDER BY Highest_Sales DESC, Lowest_Sales ASC
```

20. Find the products that were sold with a discount greater than 0.2.

```
SELECT product_id  
FROM sales  
WHERE discount > 0.2;
```

TASK 2

1. Find the products with the highest and lowest sales values.
**SELECT product_id,
MAX (sales) AS "Max_sales",
MIN (sales) AS "Min_sales"
FROM sales
GROUP BY product_id
ORDER BY "Max_sales" DESC;**
2. Find the total sales value for each sub-category of products.
**SELECT a.product_id, b.sub_category,
SUM (a.sales) AS "Total_Sales"
FROM sales AS a
INNER JOIN product AS b
ON a.product_id = b.product_id
GROUP BY b.sub_category, a.product_id
ORDER BY "Total_Sales";**
3. Show the customer names and the number of orders they made.
**SELECT Customer_Name,
COUNT(DISTINCT Order_ID) AS Number_Of_Orders
FROM customer c
INNER JOIN sales s ON c.Customer_ID = s.Customer_ID
GROUP BY Customer_Name;**
4. Show the customers who made a purchase on a specific date.
**SELECT
a.customer_id,
a.customer_name,
b.order_date
FROM customer AS a
INNER JOIN sales AS b
ON a.customer_id = b.customer_id
WHERE order_date = '2016-06-12';**
5. Find the customers who made purchases more than once.
**SELECT C.Customer_ID, Customer_Name
FROM customer c
INNER JOIN sales s ON c.Customer_ID = s.Customer_ID
GROUP BY C.Customer_ID, Customer_Name
HAVING COUNT(Order_ID) > 1;**
6. Find the customers who have not made any purchases (no sales records).
**SELECT
a.customer_id,
a.customer_name,
b.sales
FROM customer AS a
LEFT JOIN sales AS b
ON a.customer_id = b.customer_id
GROUP BY a.customer_id, a.customer_name, b.sales
HAVING COUNT(b.sales) = 0;**
7. Show the customers who have a postal code starting with '123'.
**SELECT * FROM customer
WHERE CAST(Postal_Code AS TEXT) LIKE '123%';**
8. List the top 5 most profitable products along with their profit values.

```
SELECT Product_Name, Profit
FROM product p
INNER JOIN sales s ON p.Product_ID = s.Product_ID
ORDER BY Profit DESC
LIMIT 5;
```

9. List the products that were sold with a discount higher than the average discount.

```
SELECT p.Product_Name, s.Discount
FROM product p
INNER JOIN sales s ON p.Product_ID = s.Product_ID
WHERE s.Discount > (SELECT AVG(Discount) FROM sales);
```

10. Retrieve all customers and their associated sales, if any

```
SELECT
  a.customer_name,
  a.customer_id,
  b.sales
FROM customer AS a
LEFT JOIN sales AS b
ON a.customer_id = b.customer_id;
```

11. Show all products and their corresponding sales, if any.

```
SELECT
  a.product_id,
  b.sales
FROM product AS a
LEFT JOIN sales AS b
ON a.product_id = b.product_id;
```

12. List all customers and the number of sales they have made.

```
SELECT c.Customer_ID, c.Customer_Name,
COUNT(s.Order_ID) AS Num_Sales
FROM customer c
LEFT JOIN sales s ON c.Customer_ID = s.Customer_ID
GROUP BY c.Customer_ID, c.Customer_Name;
```

13. Retrieve all sales and their corresponding customer information, if available.

```
SELECT
  a.sales,
  b.customer_id,
  b.customer_name,
  b.city,
  b.state,
  b.postal_code
FROM sales AS a
LEFT JOIN customer AS b
ON a.customer_id = b.customer_id
ORDER BY a.sales DESC;
```

14. Show all sales and their corresponding product information, if available.

```
SELECT
  a.sales,
  b.product_id,
  b.product_name,
  b.category,
  b.sub_category
FROM sales AS a
LEFT JOIN product AS b
```

ON a.product_id = b.product_id;

15. List all customers who have not made any sales

```
SELECT  
  a.customer_id,  
  a.customer_name,  
  b.sales  
FROM customer AS a  
LEFT JOIN sales AS b  
ON a.customer_id = b.customer_id  
WHERE b.customer_id IS NULL;
```

16. Show all products that have never been sold.

```
SELECT  
  a.product_id,  
  a.product_name,  
  b.sales  
FROM product AS a  
LEFT JOIN sales AS b  
ON a.product_id = b.product_id  
WHERE b.product_id IS NULL;
```

17. Retrieve all sales and their corresponding customer information, including sales without customers.

```
SELECT  
  a.sales,  
  b.customer_id,  
  b.customer_name,  
  b.city,  
  b.state,  
  b.postal_code  
FROM sales AS a  
LEFT JOIN customer AS b  
ON a.customer_id = b.customer_id  
WHERE a.customer_id IS NULL;
```

18. Show all products and their corresponding sales, including products that have not been sold

```
SELECT  
  a.product_id,  
  a.product_name,  
  b.sales  
FROM product AS a  
LEFT JOIN sales AS b  
ON a.product_id = b.product_id  
ORDER BY a.product_id
```

CASE STUDY

PROBLEM STATEMENT

You are a data analyst working for an ECart company selling products online and your job is to provide relevant insights from data based on which the company can take appropriate business decisions.

You received an email from the marketing manager

Hi Ludan,

For my new marketing campaign, I'm trying to finalize the marketing channel for the campaign. I have several options, if I choose social media the focus would be on younger age people whereas if I choose newspaper, it would be on the aged category people. I would need to figure out the profile of the customers (age) in different regions.

Please help me find the number of customers who belong to the below three categories in all the four regions.

1. Less than 36 years
2. Between 36 to 54 years
3. Above 54 years

```
SELECT
    region,
    SUM(CASE WHEN age < 36 THEN 1 ELSE 0 END) AS YOUNG_ADULT,
    SUM(CASE WHEN age BETWEEN 36 AND 54 THEN 1 ELSE 0 END) AS MIDDLE_AGE,
    SUM(CASE WHEN age > 54 THEN 1 ELSE 0 END) AS OLDER_ADULT
FROM customer
GROUP BY region
ORDER BY region;
```

Hi manager there are 793 people that fit your description.

You received an email from the supply chain manager

Hi Ludan,

I am facing issues in managing the inventory in my South and East warehouse. The warehouse in the South region is full, even over utilized, whereas the warehouse of East region has a lot of space that is underutilized. Please help me with the following information

1. Top 5 selling products in East region

```
SELECT
    a.product_id,
    a.product_name,
    c.region,
    SUM(b.quantity) AS Total_Quantity
FROM product AS a
LEFT JOIN sales AS b
ON a.product_id = b.product_id
LEFT JOIN customer AS c
ON b.customer_id = c.customer_id
WHERE c.region = 'East'
GROUP BY a.product_id, a.product_name, c.region
ORDER BY Total_Quantity DESC
LIMIT 5;
```

2. 5 least selling product in South region

```
SELECT
    a.product_id,
```

```
a.product_name,  
c.region,  
SUM(b.quantity) AS Total_Quantity  
FROM product AS a  
LEFT JOIN sales AS b  
ON a.product_id = b.product_id  
LEFT JOIN customer AS c  
ON b.customer_id = c.customer_id  
WHERE c.region = 'South'  
GROUP BY a.product_id, a.product_name, c.region  
ORDER BY Total_Quantity ASC  
LIMIT 5;
```

You received an email from the finance manager.

Hi Ludan,

I am assessing the effectiveness of our discounting policies. Please help in finding out the below information

1. Total revenue loss due to discounts

```
SELECT  
  
SUM(sales * discount) AS Total_Revenue_Loss  
  
FROM sales;
```

2. Total Revenue and Discount for each product

```
SELECT  
a.product_name,  
a.product_id,  
SUM(b.discount) AS Total_Discount,  
SUM(b.sales * b.quantity) AS Total_Revenue  
FROM product AS a  
INNER JOIN sales AS b  
ON a.product_id = b.product_id  
GROUP BY a.product_name, a.product_id;
```