DATASET LINK-<https://www.kaggle.com/datasets/vivek468/superstore-dataset-final>.

USE superstore;

SELECT \* FROM sales;

**-- QUERY TO RETURN DETAILS OF CUSTOMERS FROM HENDERSON CITY**

SELECT \* FROM Sales

WHERE city="Henderson"

GROUP BY CustomerName;

**-- QUERY TO RETRIEVE THE REGION WHICH GENERATES HIGHEST SALES AND PROFITS**

SELECT region, SUM(sales) total\_sales, SUM(profit) total\_profit from sales

GROUP BY region

ORDER BY total\_profit desc;

**-- QUERY TO RETRIEVE STATE THAT GENERATES HIGHEST SALES AND PROFITS WITH PROFT MARGIN**

SELECT state, SUM(sales) total\_sales, SUM(profit) total\_profit, ROUND(sum(profit)/sum(sales) \*100, 2) profit\_margin from sales

GROUP BY state

ORDER BY total\_profit desc

LIMIT 1;

**-- QUERY TO RETRIEVE TOP 10 STATES THAT GENERATES HIGHEST SALES AND PROFITS**

SELECT state, ROUND(SUM(sales),4) total\_sales, ROUND(SUM(profit),4) total\_profit from sales

GROUP BY state

ORDER BY total\_profit desc

LIMIT 10;

**-- QUERY TO RETRIEVE TOP 10 CITIES THAT GENERATES HIGHEST SALES AND PROFITS**

SELECT city, SUM(sales) total\_sales, SUM(profit) total\_profit from sales

GROUP BY city

ORDER BY total\_profit desc

LIMIT 10;

**-- QUERY TO RETRIEVE CITIES THAT GENERATES LOW SALES AND PROFITS**

SELECT city, SUM(sales) total\_sales, SUM(profit) total\_profit from sales

GROUP BY city

ORDER BY total\_profit

LIMIT 10;

**-- QUERY TO RETRIEVE TOTAL DISCOUNT PER PRODUCT CATEGORY**

SELECT category, ROUND(SUM(DISCOUNT),2) total\_discount from sales

GROUP BY category

ORDER BY total\_discount desc;

**-- QUERY TO RETRIEVE TOTAL DISCOUNT PER PRODUCT CATEGORY AND SUB-CATEGORY**

SELECT category,subcategory, ROUND(SUM(DISCOUNT),2) total\_discount from sales

GROUP BY category,subcategory

ORDER BY total\_discount desc;

**-- QUERY TO RETRIEVE CATEGORY WHICH GENERATES HIGHEST SALES,PROFIT IN EACH REGION AND STATE**

SELECT region, category, SUM(sales)total\_sales, SUM(PROFIT)total\_profit FROM sales

GROUP BY region, category

ORDER BY total\_profit desc;

SELECT state, category, SUM(sales)total\_sales, SUM(PROFIT)total\_profit FROM sales

GROUP BY state, category

ORDER BY total\_profit desc;

**-- QUERY TO RETRIEVE CATEGORY WHICH GENERATES LEAST SALES,PROFIT IN EACH REGION AND STATE**

SELECT region, category, SUM(sales)total\_sales, SUM(PROFIT)total\_profit FROM sales

GROUP BY region, category

ORDER BY total\_profit;

SELECT state, category, SUM(sales)total\_sales, SUM(PROFIT)total\_profit FROM sales

GROUP BY state, category

ORDER BY total\_profit;

-- **QUERY TO RETRIEVE SUBCATEGORY WHICH GENERATES HIGHEST SALES,PROFIT IN EACH REGION AND STATE**

SELECT region, subcategory, SUM(sales)total\_sales, SUM(PROFIT)total\_profit FROM sales

GROUP BY region, subcategory

ORDER BY total\_profit desc;

SELECT state, subcategory, SUM(sales)total\_sales, SUM(PROFIT)total\_profit FROM sales

GROUP BY state, subcategory

ORDER BY total\_profit desc;

**-- QUERY TO RETRIEVE TOP 10 PRODUCTNAMES WHICH ARE MOST AND LEAST PROFITABLE**

SELECT productname, SUM(profit)total\_profit, SUM(sales)total\_sales from sales

GROUP BY productname

ORDER BY total\_profit desc

LIMIT 10; -- MOST PROFITABLE

SELECT productname, SUM(profit)total\_profit, SUM(sales)total\_sales from sales

GROUP BY productname

ORDER BY total\_profit

LIMIT 10; -- LEAST PROFITABLE

-- **QUERY TO RETRIEVE SEGMENT THAT MAKES THE MOST OF OUR PROFITS AND SALES**

SELECT segment,SUM(profit)total\_profit, SUM(sales)total\_sales from sales

GROUP BY segment

ORDER BY total\_profit desc

LIMIT 1;

-- **QUERY TO RETRIEVE TOTAL CUSTOMERS**

SELECT COUNT(DISTINCT customerid) total\_customers from sales;

-- **QUERY TO RETRIEVE TOTAL CUSTOMERS IN EACH REGION AND STATE**

SELECT region, COUNT(DISTINCT customerid) total\_customers from sales

GROUP BY region

ORDER BY total\_customers desc; -- IN EACH REGION

SELECT state, COUNT(DISTINCT customerid) total\_customers from sales

GROUP BY state

ORDER BY total\_customers desc; -- IN EACH STATE

**/\*We surely had customers moving around regions**

**which explains why they all do not add up to 793(Distinct count of customers).**

**SO, there could be double counting\*/**

**-- QUERY TO RETRIEVE BOTTOM 10 REGIONS AND STATES WITH LEAST CUSTOMERS**

SELECT state, COUNT(DISTINCT customerid) total\_customers from sales

GROUP BY state

ORDER BY total\_customers

LIMIT 10;

**-- QUERY TO RETURN TOP 15 CUSTOMERS WHO GENERATED THE HIGHEST SALES**

SELECT customerid, customername, ROUND(SUM(sales),2)total\_sales, ROUND(SUM(PROFIT),2) total\_profit FROM sales

GROUP BY customerid

ORDER BY total\_sales desc

LIMIT 15;

**-- QUERY TO RETRIEVE TOTAL SALES AND PROFITS OF EACH YEAR**

SELECT date\_format(orderdate, "%Y") year,

ROUND(SUM(sales),2) total\_sales,

ROUND(SUM(profit),2) total\_profit FROM sales

GROUP BY year

ORDER BY year;

**-- QUERY TO RETRIEVE TOTAL SALES AND PROFITS PER QUARTER**

SELECT date\_format(orderdate, "%Y") year,

CASE

WHEN date\_format(orderdate, "%m") IN (01,02,03) THEN "Q1"

WHEN date\_format(orderdate, "%m") IN(04,05,06) THEN "Q2"

WHEN date\_format(orderdate, "%m") IN(07,08,09) THEN "Q3"

ELSE "Q4"

END AS quarter,

ROUND(SUM(sales),2) total\_sales,

ROUND(SUM(profit),2) total\_profit

FROM sales

GROUP BY year,quarter

ORDER BY year,quarter;

**-- QUERY TO RETURN AVERAGE SHIPPING TIME PER SHIPMODE**

SELECT shipmode, AVG(DATEDIFF(shipdate,orderdate))average\_shipping FROM sales

GROUP BY shipmode

ORDER BY average\_shipping;

**-- QUERY TO RETRIEVE UNIQUE CUSTOMER DETAILS USING JOIN**

SELECT \* FROM sales s INNER JOIN

(SELECT rowid FROM sales

GROUP BY customername) a

ON s.rowid=a.rowid;

**-- QUERY TO REMOVE DUPLICATE CUSTOMERNAMES**

SELECT \* FROM sales

GROUP BY customername;