

# SQL Queries for Sales and Customer Analysis

By Shanti Jogi

13-02-2025

## Introduction: Sales and Customer Analytics

This project focuses on analysing sales and customer data using SQL to derive key business insights. The dataset comprises two tables, TATA\_TB1 and TATA\_TB2, containing details on orders, customers, products, and sales performance.

The analysis covers various aspects, including:

- **Basic Data Exploration:** Understanding the dataset by identifying total records, unique customers, and geographical distribution across regions, states, and cities.
- **Sales & Profit Analysis:** Evaluating total sales, profit, and order quantities at different levels, such as category, subcategory, region, country, city, customer, and product.
- **Performance Metrics:** Identifying top-selling products, high-revenue cities, and the most profitable subcategories.
- **Comparative & Trend Analysis:** Extracting the oldest and latest order dates, analysing yearly trends, and calculating the percentage contribution of different categories

## Database Creation:

- `CREATE DATABASE SQL_A3;`
- `USE SQL_A3;`

## Imported Tables:

1. TATA\_TB1 - Customer table (dimension table).
2. TATA\_TB2 - Transactional data (fact table).

#Retrieve data

```
SELECT * FROM TATA_TB1 LIMIT 1;
```

#Check data types

```
DESCRIBE TATA_TB1;
```

#Updated data type (date)

```
UPDATE TATA_TB1  
SET ORDERDATE=STR_TO_DATE(ORDERDATE, '%d-%m-%Y');
```

```
ALTER TABLE TATA_TB1 MODIFY COLUMN ORDERDATE DATE;
```

#Write a query to calculate the total records in both tables

```
SELECT (  
(SELECT COUNT(*) FROM TATA_TB1) +  
(SELECT COUNT(*) FROM TATA_TB2)  
) AS TOTAL_RECORDS;
```

#Write a query to calculate the total unique count of customers

```
SELECT COUNT(DISTINCT CUSTOMERNAME) AS UNIQ_CUST_COUNT  
FROM TATA_TB1;
```

# Write a query to fetch the oldest order date and latest order date

```
SELECT MIN(ORDERDATE) OLDEST_ODATE, MAX(ORDERDATE) LATEST_ODATE  
FROM TATA_TB1;
```

#Write query to get years in table?

```
SELECT DISTINCT YEAR(ORDERDATE) YR FROM TATA_TB1;
```

#Write a query to get the no. of regions and display the region names

```
SELECT COUNT(DISTINCT REGION) AS NO_OF_REGIONS FROM TATA_TB1;  
SELECT DISTINCT REGION AS REGION_NAMES FROM TATA_TB1;
```

```
SELECT COUNT(DISTINCT REGION) AS NO_OF_REGIONS, ' ' AS NAMES FROM TATA_TB1  
UNION ALL  
SELECT ' ', REGION FROM TATA_TB1 GROUP BY REGION;
```

# Write a query to get the no. of countries and display the country names

```
SELECT COUNT(DISTINCT COUNTRY) AS NO_OF_COUNTRIES FROM TATA_TB1;  
SELECT DISTINCT COUNTRY AS COUNTRY_NAMES FROM TATA_TB1;
```

```
SELECT CAST(COUNT(DISTINCT COUNTRY) AS CHAR) AS NO_OF_COUNTRIES  
FROM TATA_TB1  
UNION ALL  
SELECT COUNTRY  
FROM TATA_TB1 GROUP BY COUNTRY;
```

# Write a query to get the no. of states and display the state names

```
SELECT COUNT(DISTINCT STATE) AS NO_OF_STATES FROM TATA_TB1;  
SELECT DISTINCT STATE AS STATE_NAMES FROM TATA_TB1;
```

```
SELECT 'TOTAL_COUNT' AS STATE_INFO  
UNION ALL  
SELECT CAST(COUNT(DISTINCT STATE) AS CHAR) FROM TATA_TB1  
UNION ALL  
SELECT 'STATE_NAMES'  
UNION ALL  
SELECT DISTINCT STATE FROM TATA_TB1;
```

#Write a query to get the no. of cities and display the city names

```
SELECT COUNT(DISTINCT CITY) AS NO_OF_CITIES FROM TATA_TB1;  
SELECT DISTINCT CITY AS CITY_NAMES FROM TATA_TB1;
```

```
WITH CITY_INFO AS (SELECT DISTINCT CITY FROM TATA_TB1)  
SELECT 'NO_OF_CITIES' AS `RESULT`  
UNION ALL  
SELECT CAST(COUNT(*) AS CHAR) FROM CITY_INFO  
UNION ALL  
SELECT 'CITY_NAMES' AS `RESULT`  
UNION ALL  
SELECT CITY AS `RESULT` FROM CITY_INFO;
```

#Write a query to calculate the total count of customers

```
WITH CUST_COUNT AS (SELECT COUNT(DISTINCT CUSTOMERNAME) AS `TOTAL  
CUTSOMERS` FROM TATA_TB1)  
SELECT * FROM CUST_COUNT;
```

#Fetch data

```
SELECT * FROM TATA_TB2 LIMIT 1;
```

#Write a query to calculate the total count of products

```
WITH PROD_COUNT AS (SELECT COUNT(DISTINCT PRODUCTNAME) `TOTAL PRODUCTS` FROM  
TATA_TB2)  
SELECT * FROM PROD_COUNT;
```

#Write a query to calculate total sales, total profit and total order quantity

```
SELECT SUM(SALES) TOTAL_SALES, SUM(PROFIT) TOTAL_PROFIT, SUM(ORDERQUANTITY)  
TOTAL_ORDER_QTY  
FROM TATA_TB2;
```

#Write a query to calculate the total sales amount for each category.

#Display the category, total sales, and total order qty and order by total sales from highest to lowest

```
SELECT CATEGORY, SUM(SALES) TOTAL_SALES, SUM(ORDERQUANTITY) TOTAL_ORDER_QTY  
FROM TATA_TB2  
GROUP BY CATEGORY  
ORDER BY TOTAL_SALES DESC;
```

#Write a query to calculate the total profit amount for each category.

#Display the category, total profit, and total order qty and order by total profit from highest to lowest?

```
SELECT CATEGORY, SUM(PROFIT) TOTAL_PROFIT, SUM(ORDERQUANTITY) TOTAL_ORDER_QTY  
FROM TATA_TB2  
GROUP BY CATEGORY  
ORDER BY TOTAL_PROFIT DESC;
```

#Whats the percentage of sales and profit by category

```
SELECT CATEGORY, SUM(SALES) TOTAL_SALES, SUM(PROFIT) TOTAL_PROFIT,  
CONCAT(ROUND((SUM(SALES)*100)/(SELECT SUM(SALES) FROM TATA_TB2),2),'%') AS  
PERCENTAGE_SALES,  
CONCAT(ROUND((SUM(PROFIT)*100)/(SELECT SUM(PROFIT) FROM TATA_TB2),2),'%') AS  
PERCENTAGE_PROFIT  
FROM TATA_TB2  
GROUP BY CATEGORY  
ORDER BY TOTAL_SALES DESC;
```

#Write a query to fetch the subcategories where total sales are greater than 100000

```
SELECT SUBCATEGORY, SUM(SALES) AS TOTAL_SALES  
FROM TATA_TB2  
GROUP BY SUBCATEGORY  
HAVING SUM(SALES) > 100000;
```

#Write a query to fetch the products where total profit is greater than 1000

```
SELECT PRODUCTNAME, SUM(PROFIT) AS TOTAL_PROFIT
FROM TATA_TB2
GROUP BY PRODUCTNAME
HAVING SUM(PROFIT) > 1000;
```

#Write a query to get the total sales and total profit for Office Supplies category

```
SELECT SUM(SALES) AS TOTAL_SALES_OS, SUM(PROFIT) AS TOTAL_PROFIT_OS
FROM TATA_TB2
WHERE CATEGORY = 'OFFICE SUPPLIES';
```

#Write a query to get the total sales and total profit for Furniture category and

#the Tables and Bookcases sub-categories

```
SELECT CATEGORY AS 'ITEM', SUM(SALES) AS TOTAL_SALES, SUM(PROFIT) AS
TOTAL_PROFIT
FROM TATA_TB2 WHERE CATEGORY = 'FURNITURE' GROUP BY CATEGORY
UNION ALL
SELECT SUBCATEGORY AS 'ITEM', SUM(SALES) AS TOTAL_SALES, SUM(PROFIT) AS
TOTAL_PROFIT
FROM TATA_TB2 WHERE SUBCATEGORY IN ('TABLES', 'BOOKCASES') GROUP BY
SUBCATEGORY;
```

#Write a query to get the total sales and total profit for Technology category and

#The Accessories, Copiers, Phones sub-categories

```
WITH TECH_SALE AS ( SELECT CATEGORY AS 'NAME',
                        SUM(SALES) AS TOTAL_SALES, SUM(PROFIT) AS TOTAL_PROFIT
                      FROM TATA_TB2
                      WHERE CATEGORY = 'TECHNOLOGY'
                      GROUP BY CATEGORY ),
  ACP AS ( SELECT SUBCATEGORY AS 'NAME',
                 SUM(SALES) AS TOTAL_SALES,
                 SUM(PROFIT) AS TOTAL_PROFIT
            FROM TATA_TB2
            WHERE SUBCATEGORY IN ('Accessories', 'Copiers', 'Phones')
            GROUP BY SUBCATEGORY )
SELECT * FROM TECH_SALE
UNION ALL
SELECT * FROM ACP;
```

#Write a query to get total sales and total profit by Region, Segment

```
SELECT T1.REGION, T1.SEGMENT,
       SUM(T2.SALES) AS TOTAL_SALES, SUM(T2.PROFIT) AS TOTAL_PROFIT
FROM TATA_TB1 T1
JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
GROUP BY T1.REGION, T1.SEGMENT
ORDER BY T1.REGION;
```

#Write a query to get total sales and total profit by Country, State and city

```
SELECT T1.COUNTRY, T1.STATE, T1.CITY, SUM(T2.SALES) AS TOTAL_SALES, SUM(T2.PROFIT)
AS TOTAL_PROFIT
FROM TATA_TB1 T1
JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
GROUP BY T1.COUNTRY, T1.STATE, T1.CITY
ORDER BY T1.COUNTRY;
```

#Write a query to get total sales and total orderqty by CustomerName

#sort it by totalsales from highest to lowest

```
SELECT T1.CUSTOMERNAME, SUM(T2.SALES) `TOTAL SALES`, SUM(T2.ORDERQUANTITY)
`TOTAL ORDER QTY`
FROM TATA_TB1 T1
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
GROUP BY T1.CUSTOMERNAME
ORDER BY SUM(T2.SALES) DESC;
```

#Identify the top 5 products with the highest sales (by sales amount).

#Show the product name, total sales, and total qty

```
SELECT PRODUCTNAME, SUM(SALES) `TOTAL SALES`, SUM(ORDERQUANTITY) `TOTAL QTY`
FROM TATA_TB2
GROUP BY PRODUCTNAME
ORDER BY SUM(SALES) DESC LIMIT 5;
```

#Write a query to get total sales by City having sales greater than 2000

```
SELECT T1.CITY, SUM(T2.SALES) `TOTAL SALES`
FROM TATA_TB1 T1
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
GROUP BY CITY
HAVING `TOTAL SALES` >2000;
```

#Write a query to get total sales by CustomerName having sales greater than 3000?

```
SELECT T1.CUSTOMERNAME, SUM(T2.SALES) `TOTAL SALES`
FROM TATA_TB1 T1
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
GROUP BY CUSTOMERNAME
HAVING `TOTAL SALES` >3000;
```

#Write a query to get total sales and total profit by shipmode

```
SELECT T1.SHIPMODE, SUM(T2.SALES) `TOTAL SALES`, SUM(T2.PROFIT) `TOTAL PROFIT`
FROM TATA_TB1 T1
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
GROUP BY T1.SHIPMODE;
```

#Write a query to get total sales for North and central region?

```
SELECT T1.REGION, SUM(T2.SALES) `TOTAL SALES` FROM TATA_TB1 T1
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
WHERE T1.REGION IN ('NORTH', 'CENTRAL')
GROUP BY T1.REGION;
```

#Write a query to get total sales and total profit for Italy and Spain countries

```
SELECT T1.COUNTRY, SUM(T2.SALES) `TOTAL SALES`, SUM(T2.PROFIT) `TOTAL PROFIT`  
FROM TATA_TB1 T1  
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID  
WHERE T1.COUNTRY IN ('ITALY', 'SPAIN')  
GROUP BY T1.COUNTRY;
```

#Write a query to get the total sales and total profit for each year

```
SELECT YEAR(T1.ORDERDATE), SUM(T2.SALES) `TOTAL SALES`, SUM(T2.PROFIT) `TOTAL PROFIT`  
FROM TATA_TB1 T1  
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID  
GROUP BY YEAR(T1.ORDERDATE);
```

#Yearly sales Growth

```
WITH yearly_sales AS (  
    SELECT  
        YEAR(T1.ORDERDATE) AS order_year,  
        SUM(T2.SALES) AS total_sales,  
        SUM(T2.PROFIT) AS total_profit  
    FROM TATA_TB1 T1  
    LEFT JOIN TATA_TB2 T2 ON T1.ORDERID = T2.ORDERID  
    GROUP BY YEAR(T1.ORDERDATE)  
)  
SELECT  
    order_year, total_sales, total_profit,  
    ROUND((total_sales - LAG(total_sales) OVER (ORDER BY order_year)) /  
        LAG(total_sales) OVER (ORDER BY order_year) * 100, 2) AS sales_growth_percent,  
    ROUND((total_profit - LAG(total_profit) OVER (ORDER BY order_year)) /  
        LAG(total_profit) OVER (ORDER BY order_year) * 100, 2) AS profit_growth_percent  
FROM yearly_sales;
```

#Find the top 10 customers who spent the most across all transactions.

#Display the customer name, total amount spent, and number of orders placed

```
SELECT T1.CUSTOMERNAME, SUM(T2.SALES) `TOTAL AMOUNT`, COUNT(T2.ORDERID) `NO.  
OF ORDERS`  
FROM TATA_TB1 T1  
LEFT JOIN TATA_TB2 T2 ON T1.ORDERID = T2.ORDERID  
GROUP BY T1.CUSTOMERNAME  
ORDER BY `TOTAL AMOUNT` DESC LIMIT 10;
```

#Write query to get total sales and total profit between year 2011 and 2013

```
SELECT YEAR(T1.ORDERDATE) `YEAR`, SUM(T2.SALES) `TOTAL SALES`, SUM(T2.PROFIT)  
`TOTAL PROFIT`  
FROM TATA_TB1 T1  
JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID  
WHERE YEAR(T1.ORDERDATE) BETWEEN 2011 AND 2013  
GROUP BY YEAR(T1.ORDERDATE);
```

#Write a query to find which products are most preferred by customers based on the total sales.

#Display customer name, favourite product9 (top 3 products per each customer) and total sales on that product

```
WITH CUST_PROD_SALE AS (SELECT T1.CUSTOMERNAME, T2.PRODUCTNAME,
                           SUM(T2.SALES) `TOTAL AMOUNT`,
                           DENSE_RANK() OVER (PARTITION BY T1.CUSTOMERNAME ORDER BY SUM(T2.SALES)
                           DESC) AS SALES_RANK
                        FROM TATA_TB1 T1
                        JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
                        GROUP BY T1.CUSTOMERNAME,T2.PRODUCTNAME )
SELECT CUSTOMERNAME, PRODUCTNAME, `TOTAL AMOUNT`
FROM CUST_PROD_SALE
WHERE SALES_RANK <= 3;
```

#Write a query to get 7th rank customer name based on total sales? Display customer name, sales amount and rank

```
WITH CUST_SALES AS ( SELECT T1.CUSTOMERNAME,
                           SUM(T2.SALES) `TOTAL AMOUNT`,
                           DENSE_RANK() OVER(ORDER BY SUM(SALES) DESC) `RANK`
                        FROM TATA_TB1 T1 JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
                        GROUP BY T1.CUSTOMERNAME )
SELECT CUSTOMERNAME, `TOTAL AMOUNT`, `RANK`
FROM CUST_SALES
WHERE `RANK` = 7;
```

#Write a query to get total sales, total profit and total order qty by country, state, category and sub-category

```
SELECT T1.COUNTRY, T1.STATE, T2.CATEGORY, T2.SUBCATEGORY,
       SUM(T2.SALES) TOTAL_SALES, SUM(T2.PROFIT) TOTAL_PROFIT,
       SUM(ORDERQUANTITY) TOTAL_ORDER_QTY
FROM TATA_TB1 T1
JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
GROUP BY T1.COUNTRY, T1.STATE, T2.CATEGORY, T2.SUBCATEGORY;
```

#Write stored procedure to get top 10 customers based on total sales?

```
DELIMITER §§
CREATE PROCEDURE TOP10_CUSTOMERS_BY_SALES()
BEGIN
WITH CUST_SALES AS ( SELECT T1.CUSTOMERNAME, SUM(T2.SALES) TOTAL_SALES
                     FROM TATA_TB1 T1
                     JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID
                     GROUP BY CUSTOMERNAME ),
    CUST_RANK AS ( SELECT CUSTOMERNAME, TOTAL_SALES,
                        DENSE_RANK() OVER(ORDER BY TOTAL_SALES DESC ) AS RNK
                   FROM CUST_SALES )
SELECT CUSTOMERNAME, TOTAL_SALES FROM CUST_RANK
WHERE RNK <=10;
END
§§
```



```
CALL TOP10_CUSTOMERS_BY_SALES;
```

#Create a virtual table (view) by using OrderID, OrderDate, CustomerName, Region,  
#country, sales, profit and order qty

```
CREATE VIEW CUST_INFO AS (  
SELECT T1.ORDERID,T1.ORDERDATE,T1.CUSTOMERNAME,T1.REGION,T1.COUNTRY,  
      T2.SALES,T2.PROFIT,T2.ORDERQUANTITY  
FROM TATA_TB1 T1 JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID);
```

```
SELECT * FROM CUST_INFO;
```

#Create a stored procedure to get region sales

```
DELIMITER $$  
CREATE PROCEDURE REGIONALSALES()  
BEGIN  
SELECT REGION, SUM(SALES) TOTAL_SALES FROM TATA_TB1 T1  
JOIN TATA_TB2 T2 ON T1.ORDERID=T2.ORDERID  
GROUP BY REGION;  
END  
$$
```

```
CALL REGIONALSALES;
```

### Conclusion

This project successfully analyses sales and customer data using SQL, providing valuable insights into business performance. By exploring key metrics such as total sales, profit trends, top-performing products, and regional distribution, we gain a deeper understanding of customer behaviour and sales patterns.

Key takeaways from the analysis include:

- Identification of high-revenue regions, cities, and product categories.
- Insights into yearly sales trends and customer purchasing behaviour.
- Recognition of top-selling products and their contribution to overall revenue.
- Data-driven recommendations for optimizing sales strategies and improving business growth.

The findings from this project can help businesses make informed decisions, improve sales performance, and enhance customer engagement through targeted strategies.