SALES INSIGHT ANALYSIS

Objective

- The primary objective of this project is to analyze and manage transactional data related to customers, products, markets, and dates within a sales ecosystem.
- To perform detailed analysis on sales data, including calculating total sales, identifying top-performing products and markets, and analyzing sales trends over time.
- We need to examine dataset with SQL and help- the company will understand its Sales Insights by answering simple questions.

Question Mode

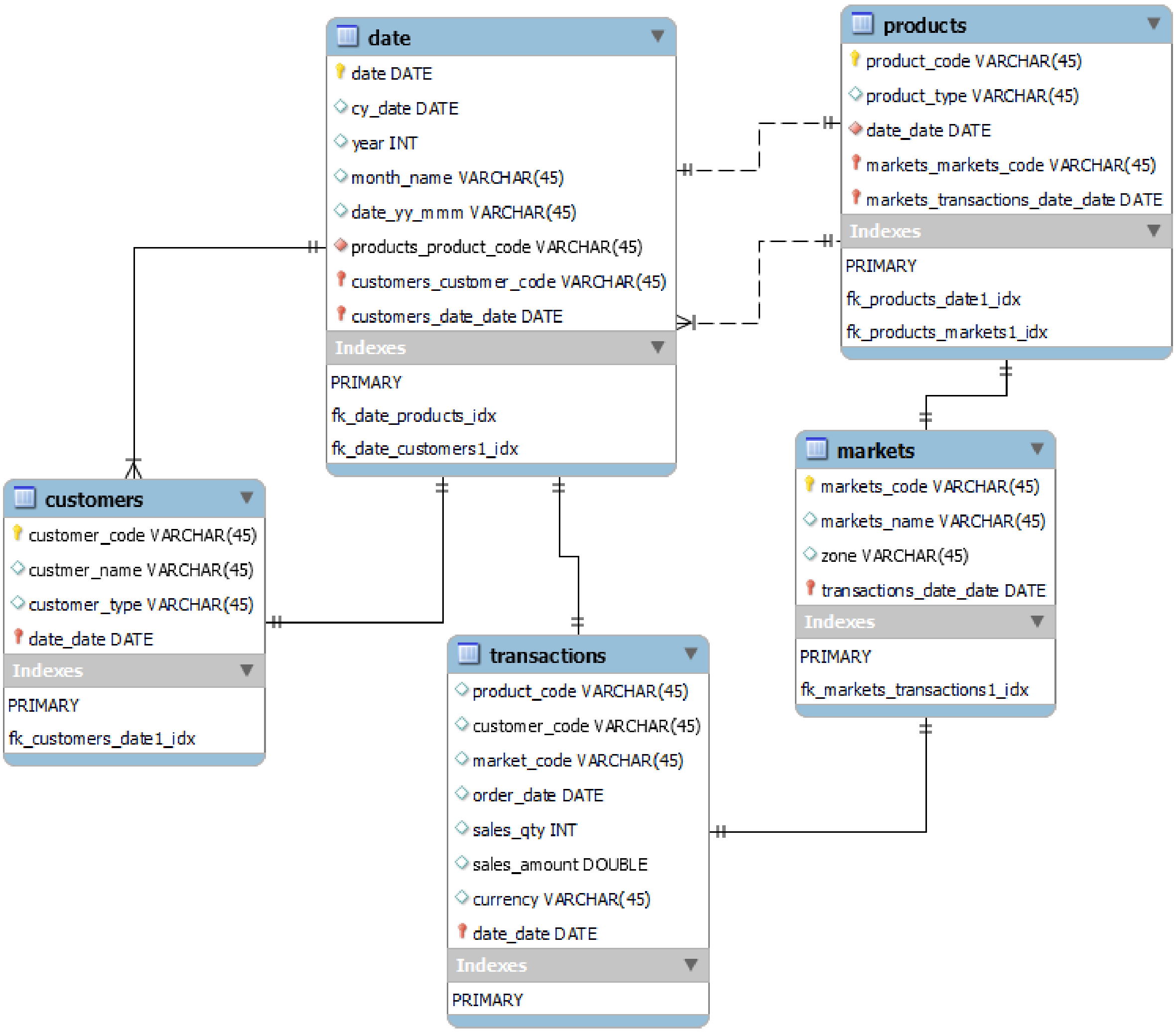
Easy - Queries include : SELECT, SUM, AVG, COUNT, LIMIT, DESC.

Moderate - Queries Include : JOINS, GROUP BY, ORDER BY, LIMIT.

Advance - Queries Include : DISTINCT, HAVING, JOINS, NESTED QUERIES

BY PRUTHA MITRAGOTRI

Sales Insight ER-Diagram

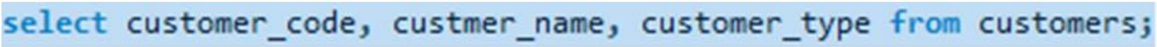


BY PRUTHA MITRAGOTRI

Questions

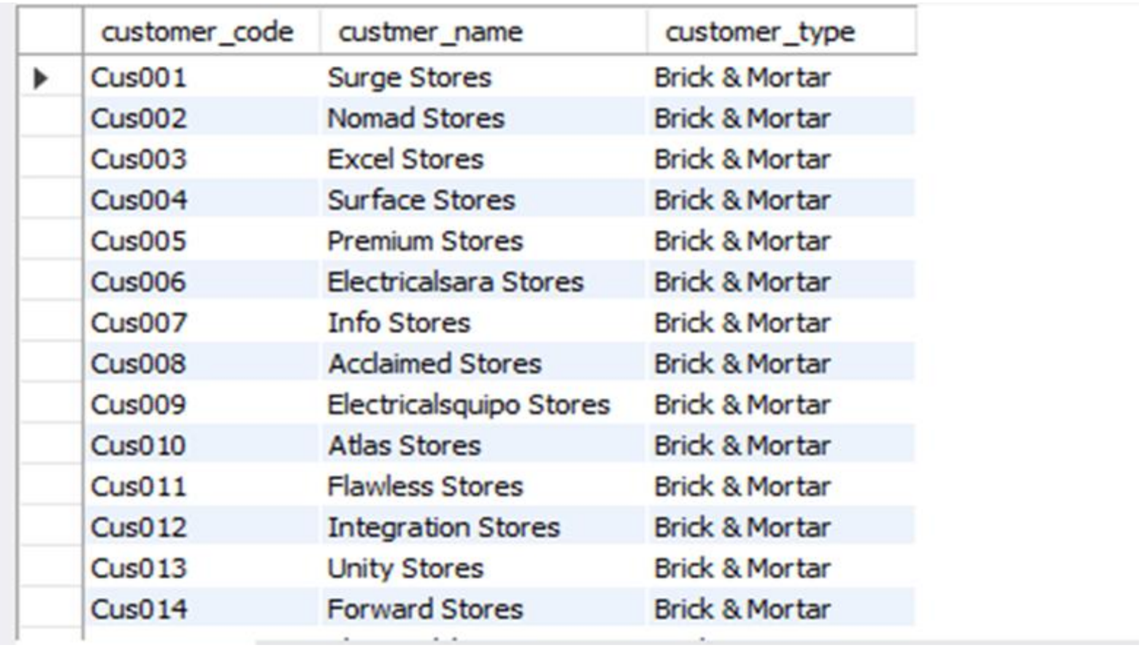
- Write an SQL query to retrieve all customers from the database. The query should display each customer's code, name, and type.

Input:



```
select customer_code, custmer_name, customer_type from customers;
```

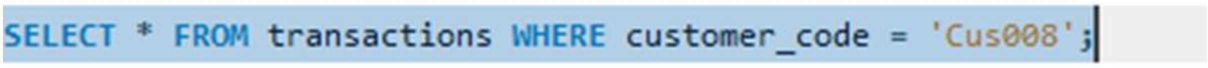
Output:



	customer_code	custmer_name	customer_type
▶	Cus001	Surge Stores	Brick & Mortar
	Cus002	Nomad Stores	Brick & Mortar
	Cus003	Excel Stores	Brick & Mortar
	Cus004	Surface Stores	Brick & Mortar
	Cus005	Premium Stores	Brick & Mortar
	Cus006	Electricalsara Stores	Brick & Mortar
	Cus007	Info Stores	Brick & Mortar
	Cus008	Acclaimed Stores	Brick & Mortar
	Cus009	Electricalsquipo Stores	Brick & Mortar
	Cus010	Atlas Stores	Brick & Mortar
	Cus011	Flawless Stores	Brick & Mortar
	Cus012	Integration Stores	Brick & Mortar
	Cus013	Unity Stores	Brick & Mortar
	Cus014	Forward Stores	Brick & Mortar

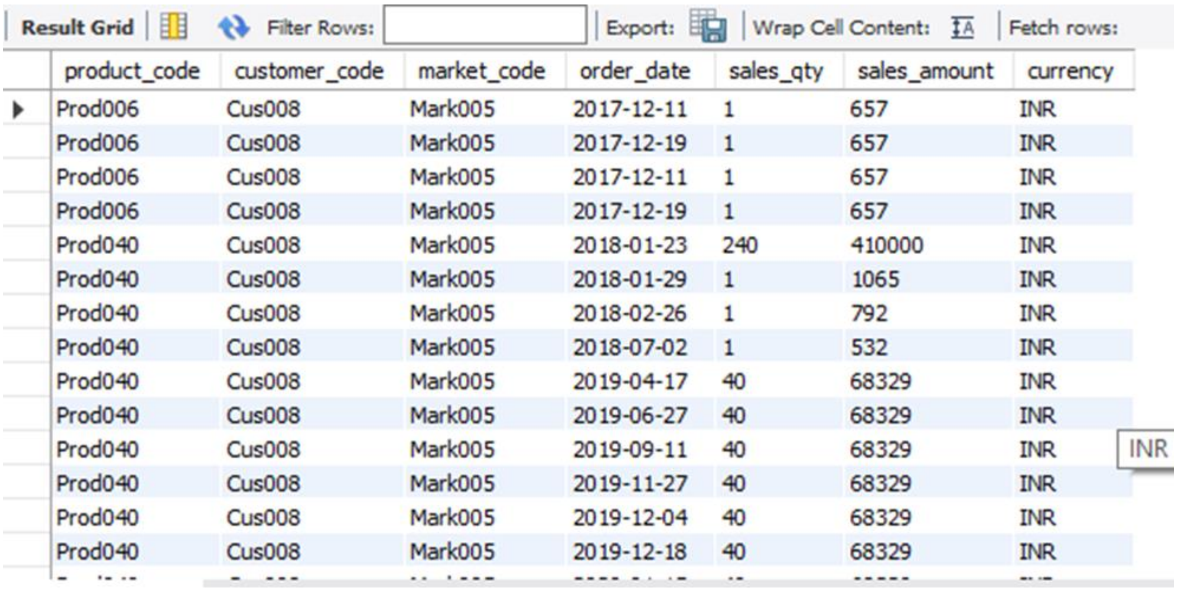
- Write an SQL query to retrieve all transactions made by a specific customer. The query should display details such as the product code, market code, order date, sales quantity, sales amount, and currency for the transactions.

Input:



```
SELECT * FROM transactions WHERE customer_code = 'Cus008';
```

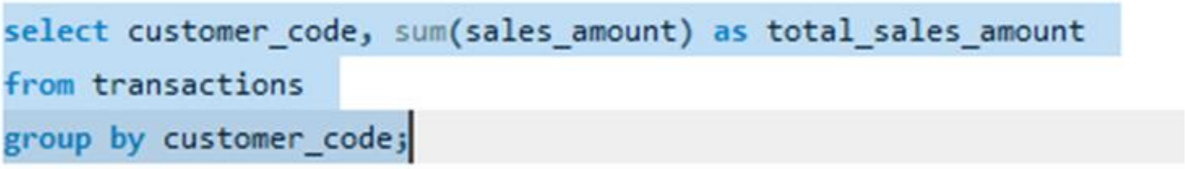
Output:



product_code	customer_code	market_code	order_date	sales_qty	sales_amount	currency
Prod006	Cus008	Mark005	2017-12-11	1	657	INR
Prod006	Cus008	Mark005	2017-12-19	1	657	INR
Prod006	Cus008	Mark005	2017-12-11	1	657	INR
Prod006	Cus008	Mark005	2017-12-19	1	657	INR
Prod040	Cus008	Mark005	2018-01-23	240	410000	INR
Prod040	Cus008	Mark005	2018-01-29	1	1065	INR
Prod040	Cus008	Mark005	2018-02-26	1	792	INR
Prod040	Cus008	Mark005	2018-07-02	1	532	INR
Prod040	Cus008	Mark005	2019-04-17	40	68329	INR
Prod040	Cus008	Mark005	2019-06-27	40	68329	INR
Prod040	Cus008	Mark005	2019-09-11	40	68329	INR
Prod040	Cus008	Mark005	2019-11-27	40	68329	INR
Prod040	Cus008	Mark005	2019-12-04	40	68329	INR
Prod040	Cus008	Mark005	2019-12-18	40	68329	INR

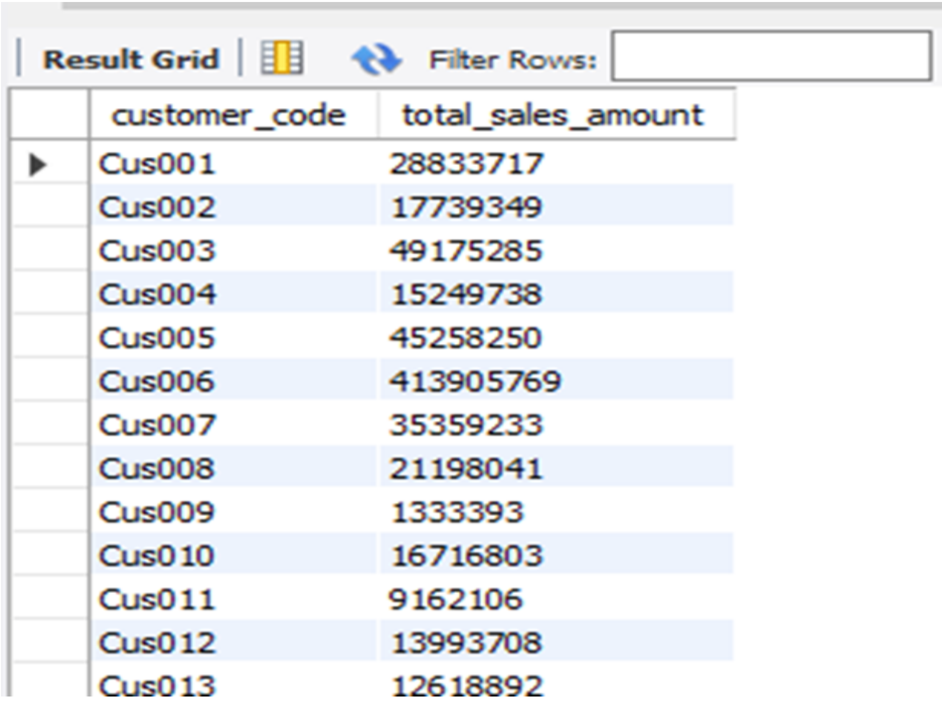
- Write an SQL query to calculate the total sales amount for each customer. The query should display the customer code along with the total sales amount generated by each customer.

Input:



```
select customer_code, sum(sales_amount) as total_sales_amount
from transactions
group by customer_code;
```

Output:

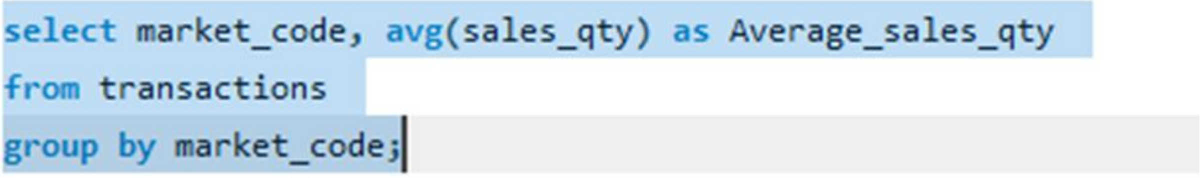


customer_code	total_sales_amount
Cus001	28833717
Cus002	17739349
Cus003	49175285
Cus004	15249738
Cus005	45258250
Cus006	413905769
Cus007	35359233
Cus008	21198041
Cus009	1333393
Cus010	16716803
Cus011	9162106
Cus012	13993708
Cus013	12618892

BY PRUTHA MITRAGOTRI

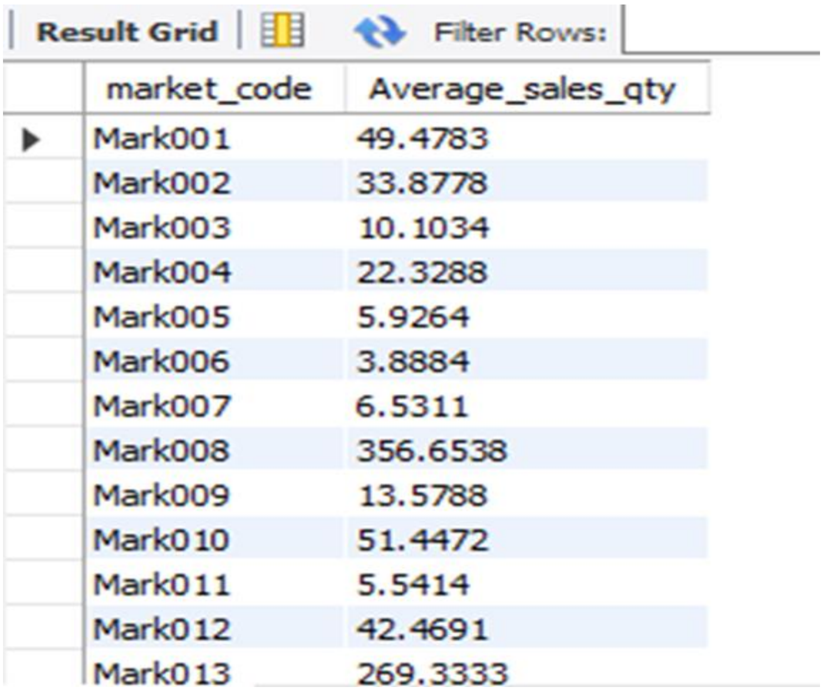
- Write an SQL query to calculate the average sales quantity for each market. The query should display the market code along with the average quantity of products sold in that market.

Input:



```
select market_code, avg(sales_qty) as Average_sales_qty
from transactions
group by market_code;
```

Output:

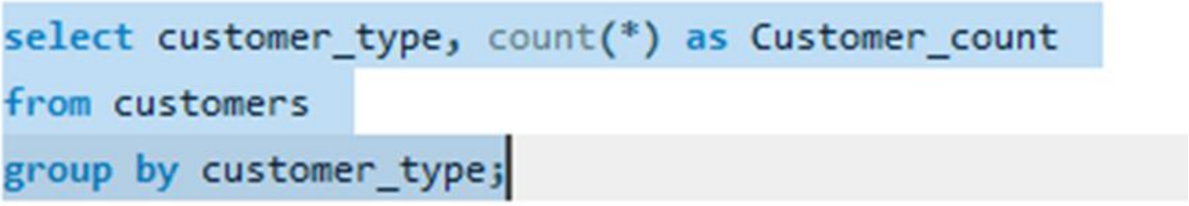


market_code	Average_sales_qty
Mark001	49.4783
Mark002	33.8778
Mark003	10.1034
Mark004	22.3288
Mark005	5.9264
Mark006	3.8884
Mark007	6.5311
Mark008	356.6538
Mark009	13.5788
Mark010	51.4472
Mark011	5.5414
Mark012	42.4691
Mark013	269.3333

BY PRUTHA MITRAGOTRI

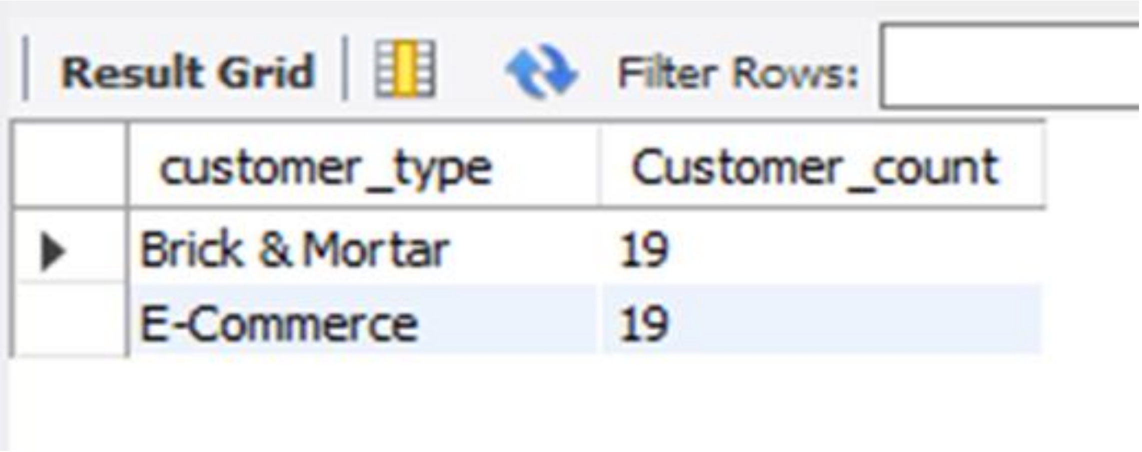
- Write an SQL query to count the number of customers for each customer type. The query should display the customer type and the corresponding count of customers within that type.

Input:

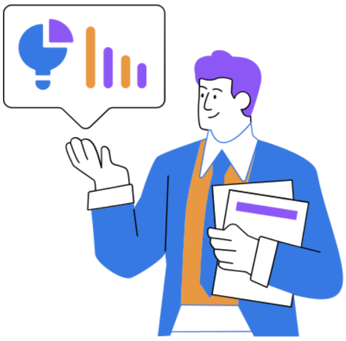


```
select customer_type, count(*) as Customer_count
from customers
group by customer_type;
```

Output:

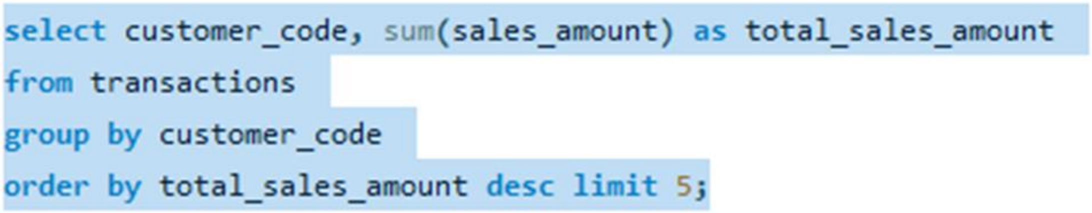


customer_type	Customer_count
Brick & Mortar	19
E-Commerce	19



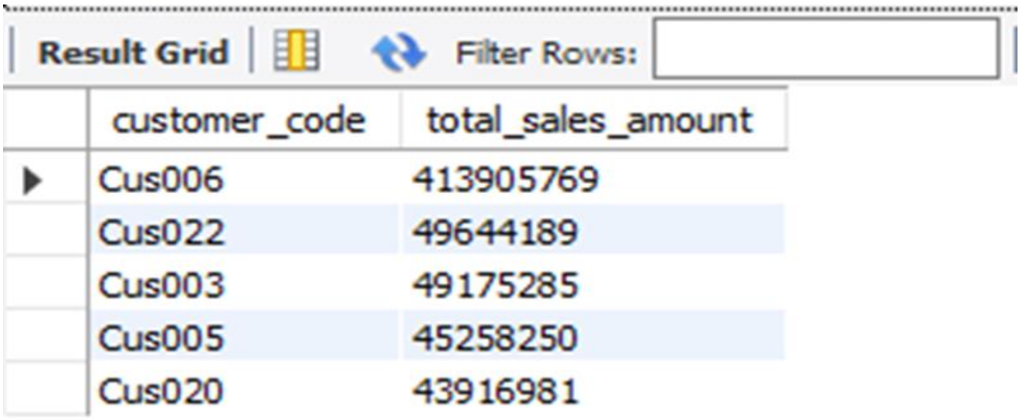
- **Write an SQL query to identify the top 5 customers based on their total sales amount. The query should display the customer code and the total sales amount for each of the top 5 customers, sorted in descending order of sales.**

Input:



```
select customer_code, sum(sales_amount) as total_sales_amount
from transactions
group by customer_code
order by total_sales_amount desc limit 5;
```

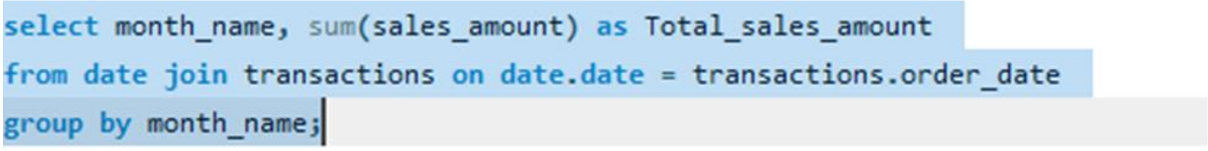
Output:



customer_code	total_sales_amount
Cus006	413905769
Cus022	49644189
Cus003	49175285
Cus005	45258250
Cus020	43916981

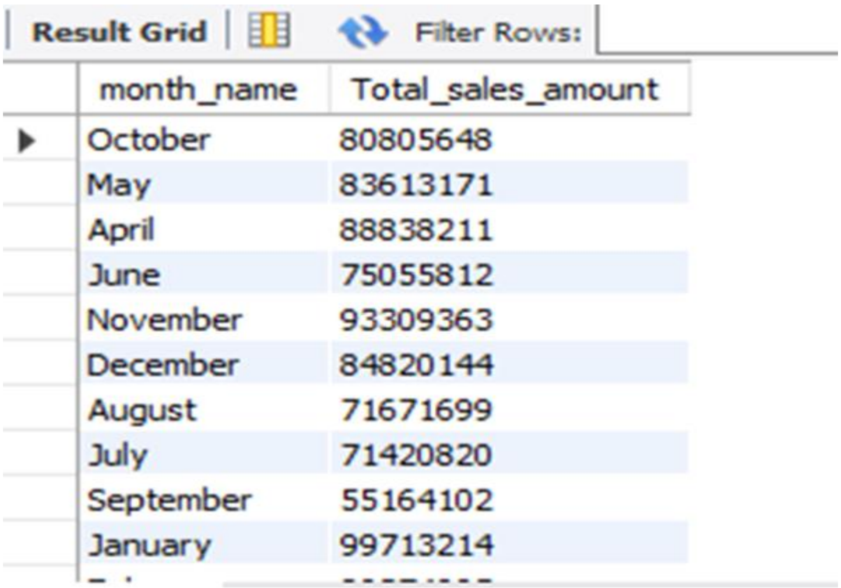
- Write an SQL query to generate a report of total sales for each month. The query should display the month name and the total sales amount for that month, sorted in descending order of total sales.

Input:



```
select month_name, sum(sales_amount) as Total_sales_amount
from date join transactions on date.date = transactions.order_date
group by month_name;
```

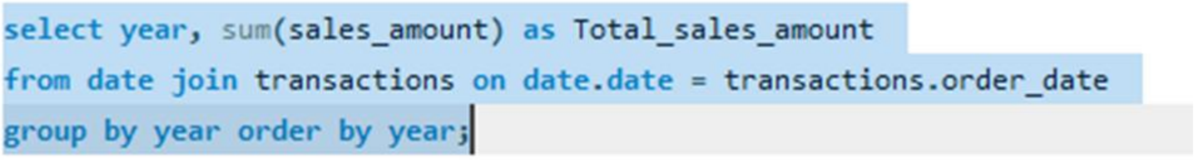
Output:



month_name	Total_sales_amount
October	80805648
May	83613171
April	88838211
June	75055812
November	93309363
December	84820144
August	71671699
July	71420820
September	55164102
January	99713214

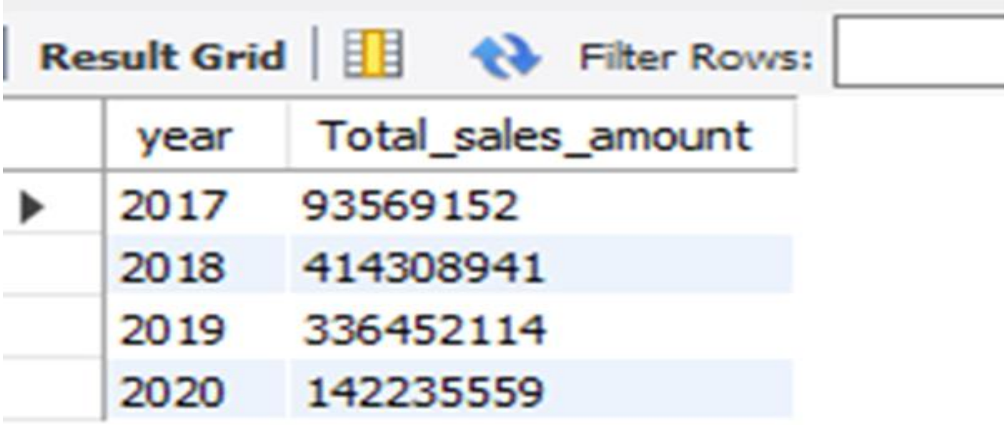
- **Write an SQL query to analyze the sales trend over multiple years. The query should display the year and the total sales amount for each year, sorted chronologically.**

Input:



```
select year, sum(sales_amount) as Total_sales_amount
from date join transactions on date.date = transactions.order_date
group by year order by year;
```

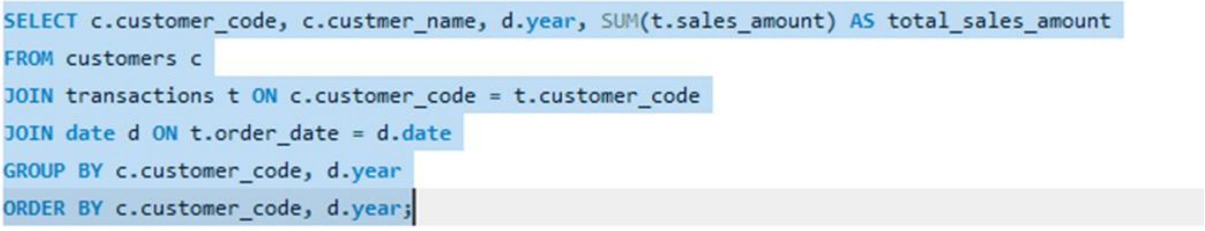
Output:



year	Total_sales_amount
2017	93569152
2018	414308941
2019	336452114
2020	142235559

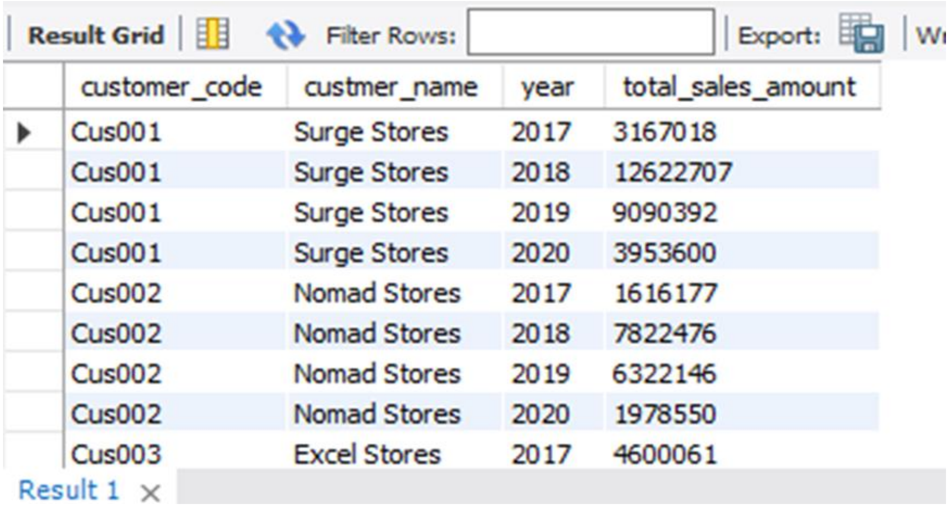
- Write an SQL query to calculate the total sales amount for each customer over multiple years. The query should display the customer code, customer name, year, and the corresponding total sales amount for each year, sorted by customer code and year.

Input:



```
SELECT c.customer_code, c.custmer_name, d.year, SUM(t.sales_amount) AS total_sales_amount
FROM customers c
JOIN transactions t ON c.customer_code = t.customer_code
JOIN date d ON t.order_date = d.date
GROUP BY c.customer_code, d.year
ORDER BY c.customer_code, d.year;
```

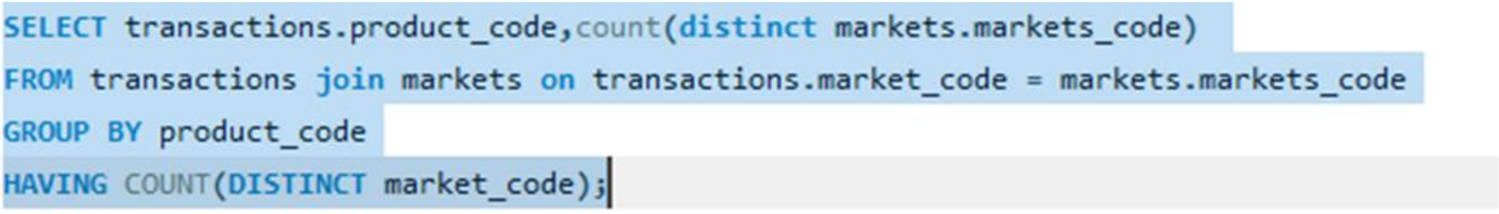
Output:



customer_code	custmer_name	year	total_sales_amount
Cus001	Surge Stores	2017	3167018
Cus001	Surge Stores	2018	12622707
Cus001	Surge Stores	2019	9090392
Cus001	Surge Stores	2020	3953600
Cus002	Nomad Stores	2017	1616177
Cus002	Nomad Stores	2018	7822476
Cus002	Nomad Stores	2019	6322146
Cus002	Nomad Stores	2020	1978550
Cus003	Excel Stores	2017	4600061

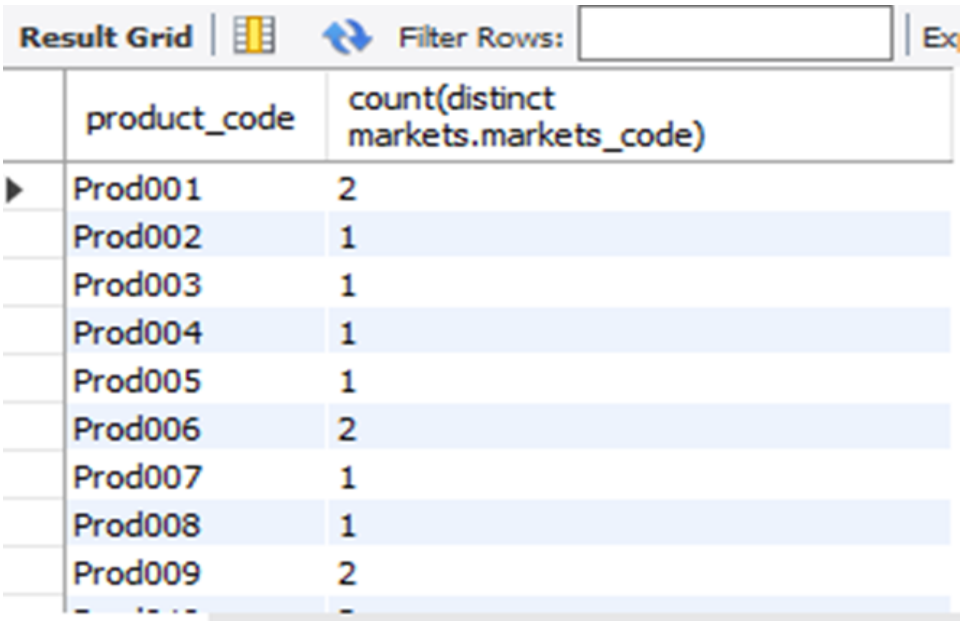
- Write a query to identify products that have been sold in every market.

Input:



```
SELECT transactions.product_code,count(distinct markets.markets_code)
FROM transactions join markets on transactions.market_code = markets.markets_code
GROUP BY product_code
HAVING COUNT(DISTINCT market_code);
```

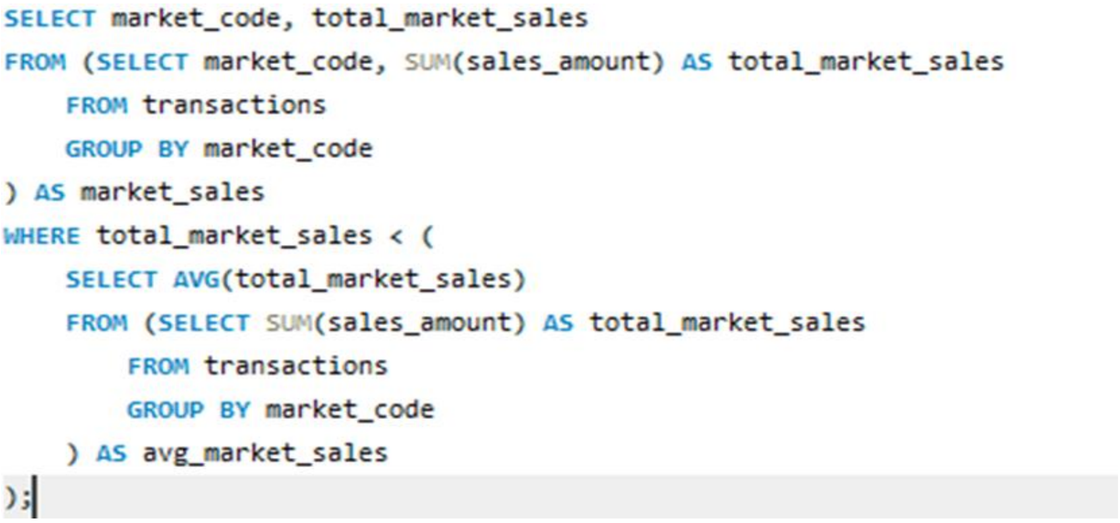
Output:



product_code	count(distinct markets.markets_code)
Prod001	2
Prod002	1
Prod003	1
Prod004	1
Prod005	1
Prod006	2
Prod007	1
Prod008	1
Prod009	2

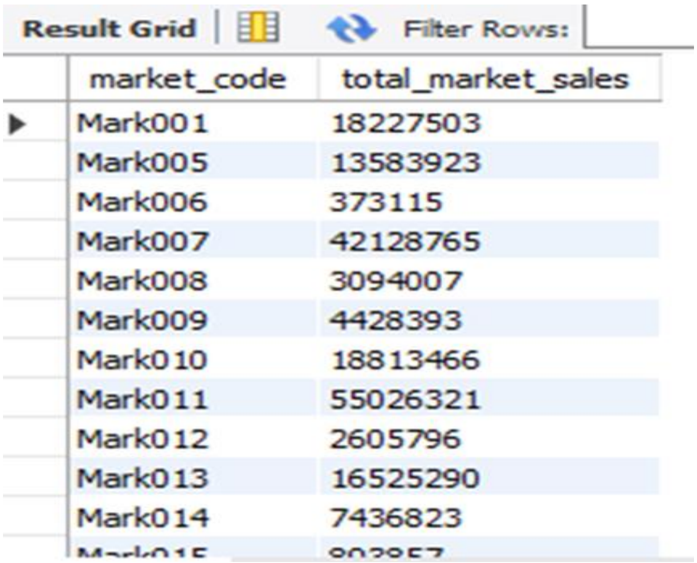
- Write a query to find markets whose total sales are below the average total sales of all markets.

Input:



```
SELECT market_code, total_market_sales
FROM (SELECT market_code, SUM(sales_amount) AS total_market_sales
    FROM transactions
    GROUP BY market_code
) AS market_sales
WHERE total_market_sales < (
    SELECT AVG(total_market_sales)
    FROM (SELECT SUM(sales_amount) AS total_market_sales
        FROM transactions
        GROUP BY market_code
    ) AS avg_market_sales
);
```

Output:



market_code	total_market_sales
Mark001	18227503
Mark005	13583923
Mark006	373115
Mark007	42128765
Mark008	3094007
Mark009	4428393
Mark010	18813466
Mark011	55026321
Mark012	2605796
Mark013	16525290
Mark014	7436823
Mark015	903857

Conclusion

In conclusion, this project aims to provide a comprehensive analysis and management of transactional data within a sales ecosystem. By utilizing SQL to examine the dataset, the project seeks to deliver valuable insights into sales performance, including total sales calculations, identification of top-performing products and markets, and an analysis of sales trends over time.

These insights will enable the company to make informed decisions, optimize sales strategies, and ultimately improve overall business performance.

BY PRUTHA MITRAGOTRI

Project Resource

SQL Database Link : https://rb.gy/2zspua

CSV Files Dataset Link : https://rb.gy/5timiu

GitHub Project Link : GitHub - Prutha-mitragotri/SQL-Sales-Insight-Project

Profile Links

LinkedIn Profile : www.linkedin.com/in/prutha-mitragotri

GitHub Profile : Prutha-mitragotri - Overview

Mail Id : mitragotriprutha@gmail.com

Contact : +918208430603

Thank you

BY PRUTHA MITRAGOTRI

BY PRUTHA MITRAGOTRI