**Sales Trend Analysis using Aggregations**

**1.Objective**

To analyze monthly revenue and order volume using SQL aggregation functions such as SUM() and COUNT() along with GROUP BY and EXTRACT in PostgreSQl

**2. Dataset Description**

Table Name: online\_sales.orders  
Columns:  
- order\_id (Primary Key)  
- order\_date (Date)  
- amount (Numeric)  
- product\_id (Integer)

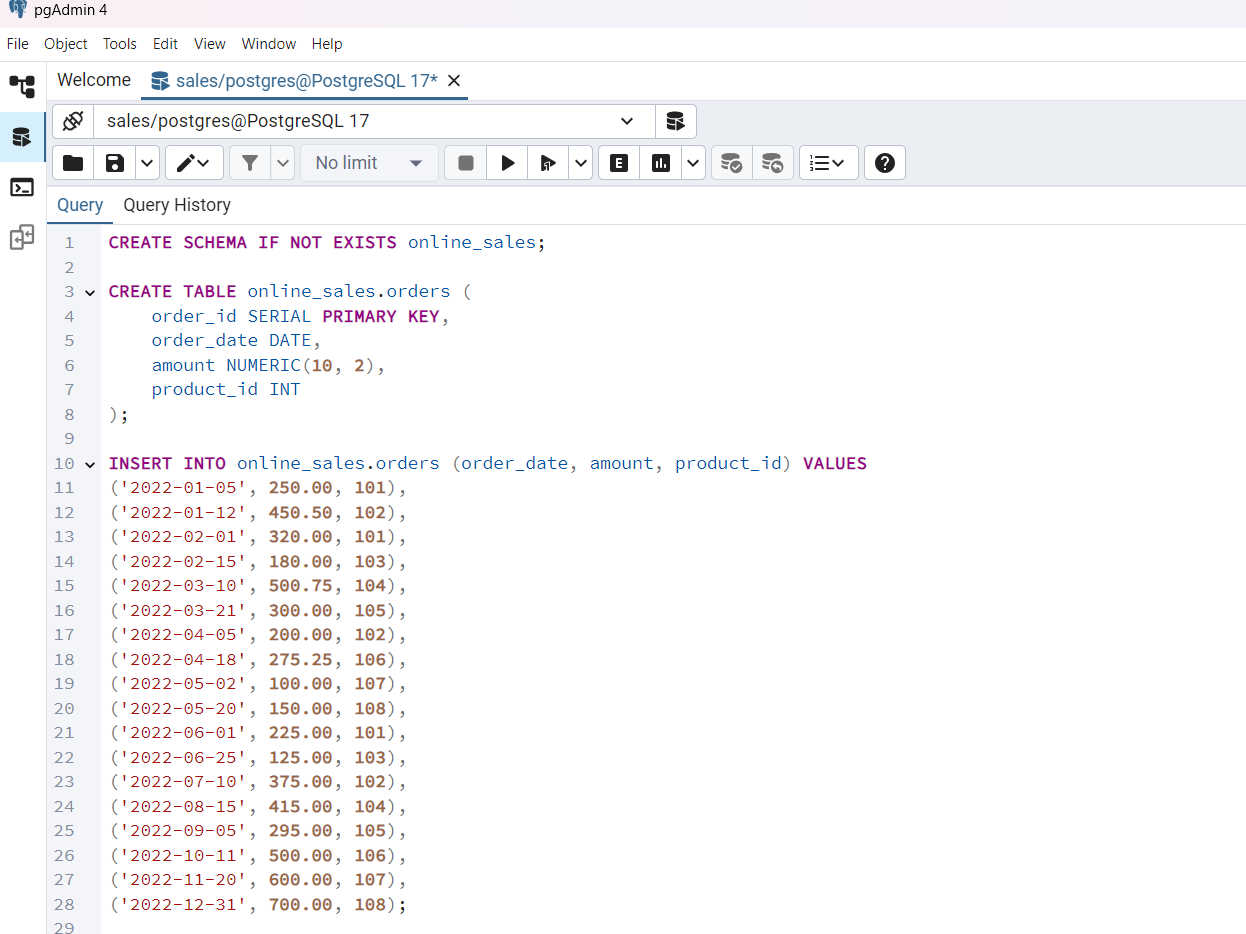
**3. Table creation SQL**

CREATE SCHEMA IF NOT EXISTS online\_sales;  
  
CREATE TABLE online\_sales.orders (  
 order\_id SERIAL PRIMARY KEY,  
 order\_date DATE,  
 amount NUMERIC(10, 2),  
 product\_id INT  
);

**4 showing data insert SQL**

INSERT INTO online\_sales.orders (order\_date, amount, product\_id) VALUES  
('2022-01-05', 250.00, 101),  
('2022-01-12', 450.50, 102),  
('2022-02-01', 320.00, 101),  
('2022-02-15', 180.00, 103),  
('2022-03-10', 500.75, 104),  
('2022-03-21', 300.00, 105),  
('2022-04-05', 200.00, 102),  
('2022-04-18', 275.25, 106),  
('2022-05-02', 100.00, 107),  
('2022-05-20', 150.00, 108),  
('2022-06-01', 225.00, 101),  
('2022-06-25', 125.00, 103),  
('2022-07-10', 375.00, 102),  
('2022-08-15', 415.00, 104),  
('2022-09-05', 295.00, 105),  
('2022-10-11', 500.00, 106),  
('2022-11-20', 600.00, 107),  
('2022-12-31', 700.00, 108);

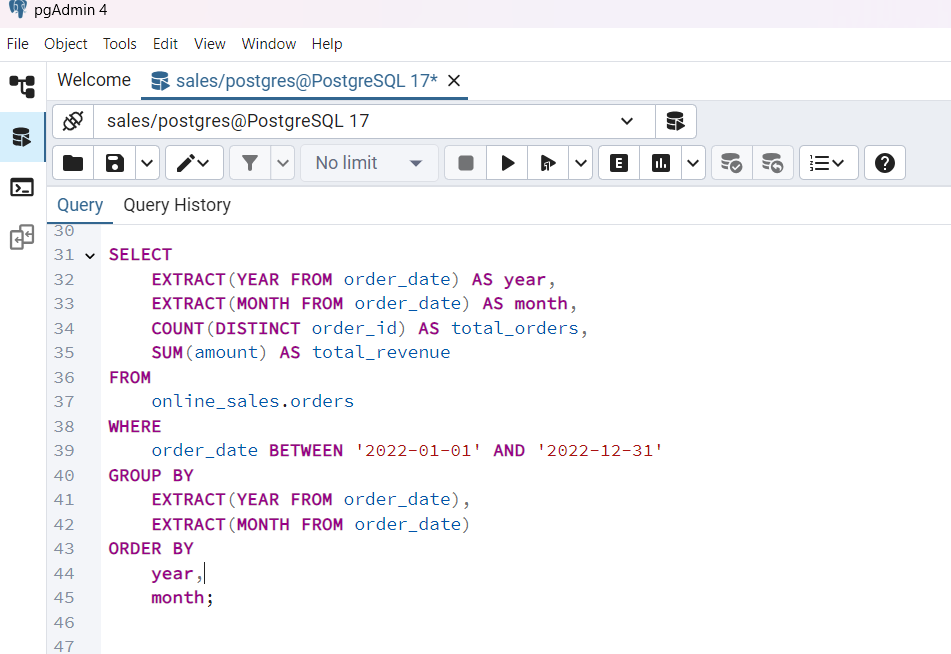
**SQL script for table created and inserted values**



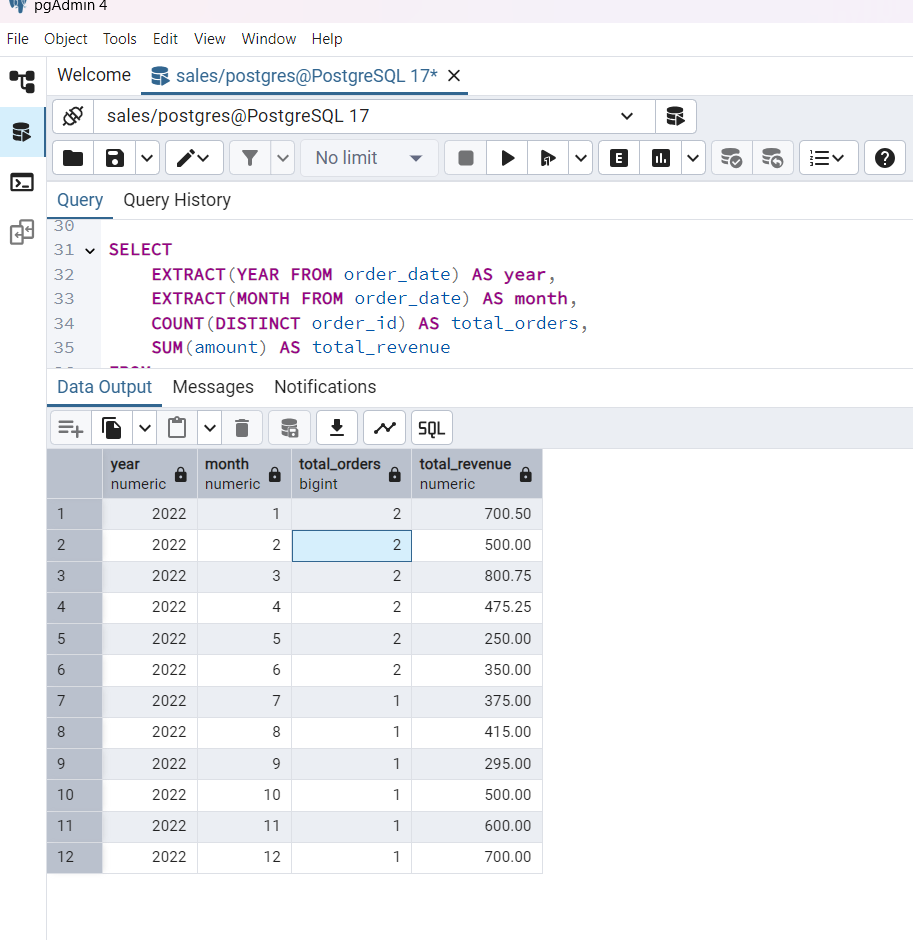
**Analysis Query**

SELECT  
 EXTRACT(YEAR FROM order\_date) AS year,  
 EXTRACT(MONTH FROM order\_date) AS month,  
 COUNT(DISTINCT order\_id) AS total\_orders,  
 SUM(amount) AS total\_revenue  
FROM  
 online\_sales.orders  
WHERE  
 order\_date BETWEEN '2022-01-01' AND '2022-12-31'  
GROUP BY  
 EXTRACT(YEAR FROM order\_date),  
 EXTRACT(MONTH FROM order\_date)  
ORDER BY  
 year,  
 month;

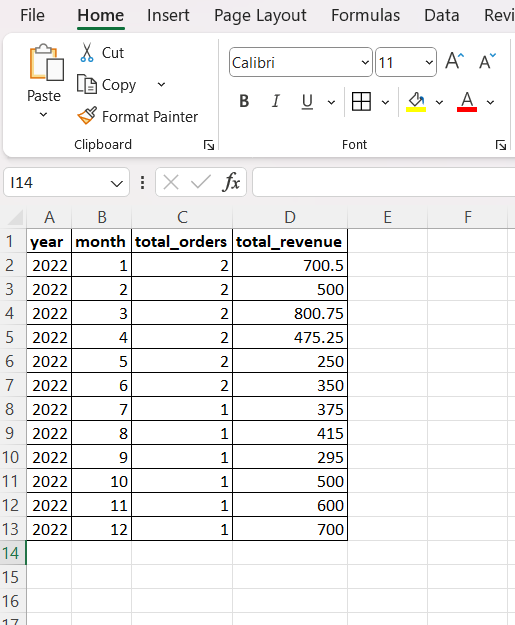
**SQL script for Analysis query**



**Table OUTPUT in SQL**



**Extract output file in sql**

z

**Outcome:**

I learned how to use aggregate functions such as SUM() and COUNT to calculate monthly revenue and order volume from a sales dataset.  
I used the EXTRACT (YEAR FROM order\_date) and EXTRACT (MONTH FROM order\_date) functions to group sales data by year and month, which is essential for time-series trend analysis.  
I successfully generated a monthly summary report showing 1) Total number of orders each month 2) Total revenue per month , This helped me identify patterns in sales over time.  
I learned how to sort (ORDER BY) and filter (WHERE ... BETWEEN) data efficiently for generating business reports.  
I practiced creating tables, inserting data, and running queries in pgAdmin, which improved my hands-on experience with PostgreSQL.  
I learned how to organize the task into well-documented formats like SQL scripts, result tables, Word, Excel, and PDF reports — useful for real-world reporting and stakeholder presentations.