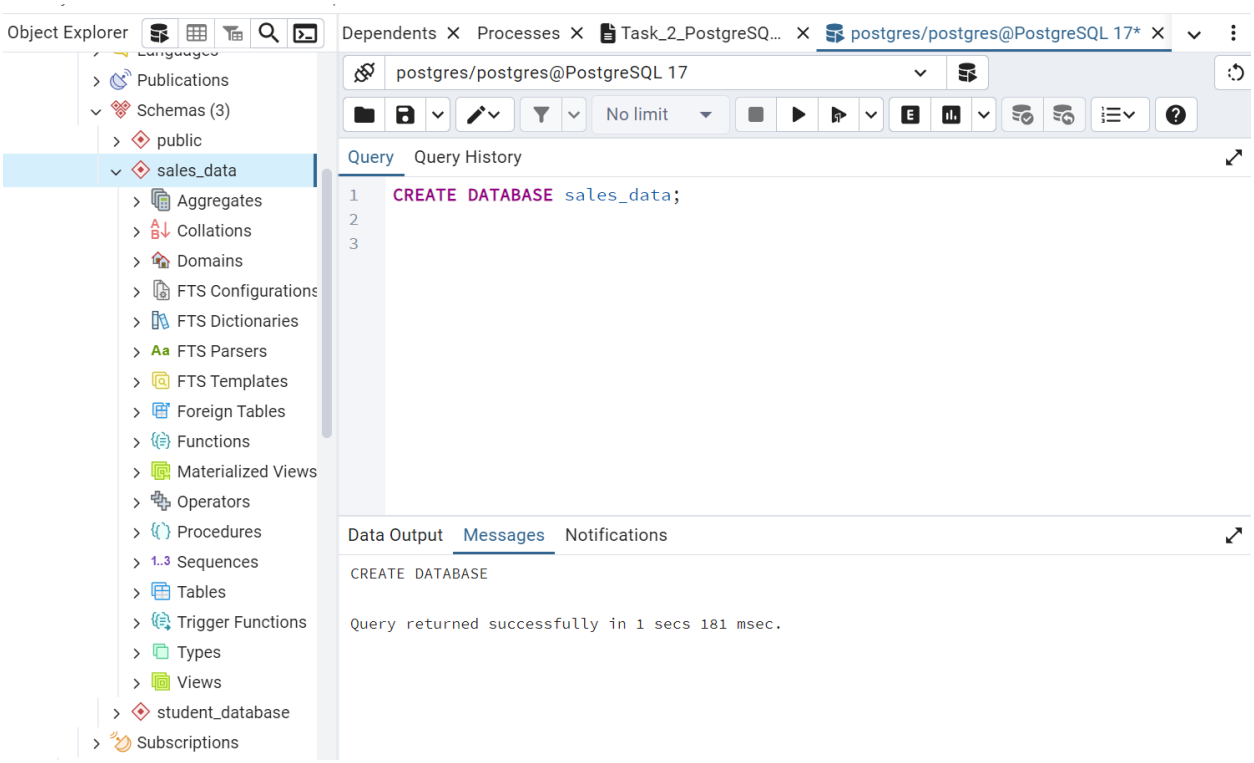
OUTPUT SHEET

Project: OLAP Operations (using PostgreSQL)

Objective: Perform OLAP operations (Drill Down, Rollup, Cube, Slice, and Dice) on the "sales_sample" table to analyze sales data. The project will include the following tasks:

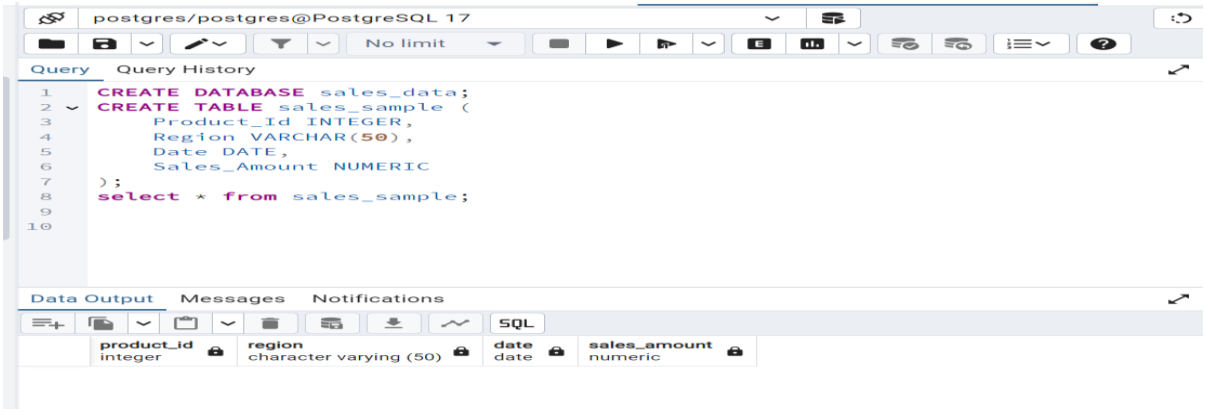
1. Database Creation

a. Create a database to store the sales data (PostgreSQL).



b. Create a table named "sales_sample" with the specified columns:

- Product_Id (Integer)
- Region (varchar(50))-like East ,West etc
- Date (Date)
- Sales_Amount (int/numeric)



2. Data Creation
Insert 10 sample records into the "sales_sample" table, representing sales data.

Data Output | Messages | Notifications

	product_id integer	region character varying (50)	date date	sales_amount numeric
1	1	East	2024-01-01	500
2	2	West	2024-01-02	700
3	3	East	2024-01-03	400
4	4	North	2024-01-04	800
5	5	South	2024-01-05	600
6	1	West	2024-01-06	550
7	3	East	2024-01-07	300
8	2	North	2024-01-08	650
9	4	South	2024-01-09	750
10	1	West	2024-01-10	450

Total rows: 10 of 10 Query complete 00:00:00.174 Ln 8, Col 28

3. Perform OLAP operations

a) Drill Down-Analyze sales data at a more detailed level. Write a query to perform drill down from region to product level to understand sales performance.

```
21
22  SELECT Region, Product_Id, SUM(Sales_Amount) AS Total_Sales
23  FROM sales_sample
24  GROUP BY Region, Product_Id
25  ORDER BY Region, Product_Id;
```

Data Output | Messages | Notifications

	region character varying (50)	product_id integer	total_sales numeric
1	East	1	500
2	East	3	700
3	North	2	650
4	North	4	800
5	South	4	750
6	South	5	600
7	West	1	1000
8	West	2	700

b) Rollup- To summarize sales data at different levels of granularity. Write a query to perform roll up from product to region level to view total sales by region.

```
26
27  SELECT Region, Product_Id, SUM(Sales_Amount) AS Total_Sales
28  FROM sales_sample
29  GROUP BY ROLLUP (Region, Product_Id)
30  ORDER BY Region, Product_Id;
```

Data Output | Messages | Notifications

	region character varying (50)	product_id integer	total_sales numeric
1	East	1	500
2	East	3	700
3	East	[null]	1200
4	North	2	650
5	North	4	800
6	North	[null]	1450
7	South	4	750
8	South	5	600
9	South	[null]	1350
10	West	1	1000
11	West	2	700
12	West	[null]	1700
13	[null]	[null]	5700

Total rows: 13 of 13 Query complete 00:00:00.113 Ln 30, Col 29

c) Cube - To analyze sales data from multiple dimensions simultaneously. Write a query to Explore sales data from different perspectives, such as product, region, and date.

postgres/postgres@PostgreSQL 17

No limit

Query | Query History

```
31
32  SELECT Region, Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
33  FROM sales_sample
34  GROUP BY CUBE (Region, Product_Id, Date)
35  ORDER BY Region, Product_Id, Date;
```

Data Output | Messages | Notifications

	region character varying (50)	product_id integer	date date	total_sales numeric
1	East	1	2024-01-01	500
2	East	1	[null]	500
3	East	3	2024-01-03	400
4	East	3	2024-01-07	300
5	East	3	[null]	700
6	East	[null]	2024-01-01	500
7	East	[null]	2024-01-03	400
8	East	[null]	2024-01-07	300
9	East	[null]	[null]	1200
10	North	2	2024-01-08	650
11	North	2	[null]	650
12	North	4	2024-01-04	800
13	North	4	[null]	800
14	North	[null]	2024-01-04	800
15	North	[null]	2024-01-08	650

Data Output | Messages | Notifications

	region character varying (50)	product_id integer	date date	total_sales numeric
16	North	[null]	[null]	1450
17	South	4	2024-01-09	750
18	South	4	[null]	750
19	South	5	2024-01-05	600
20	South	5	[null]	600
21	South	[null]	2024-01-05	600
22	South	[null]	2024-01-09	750
23	South	[null]	[null]	1350
24	West	1	2024-01-06	550
25	West	1	2024-01-10	450
26	West	1	[null]	1000
27	West	2	2024-01-02	700
28	West	2	[null]	700
29	West	[null]	2024-01-02	700
30	West	[null]	2024-01-06	550

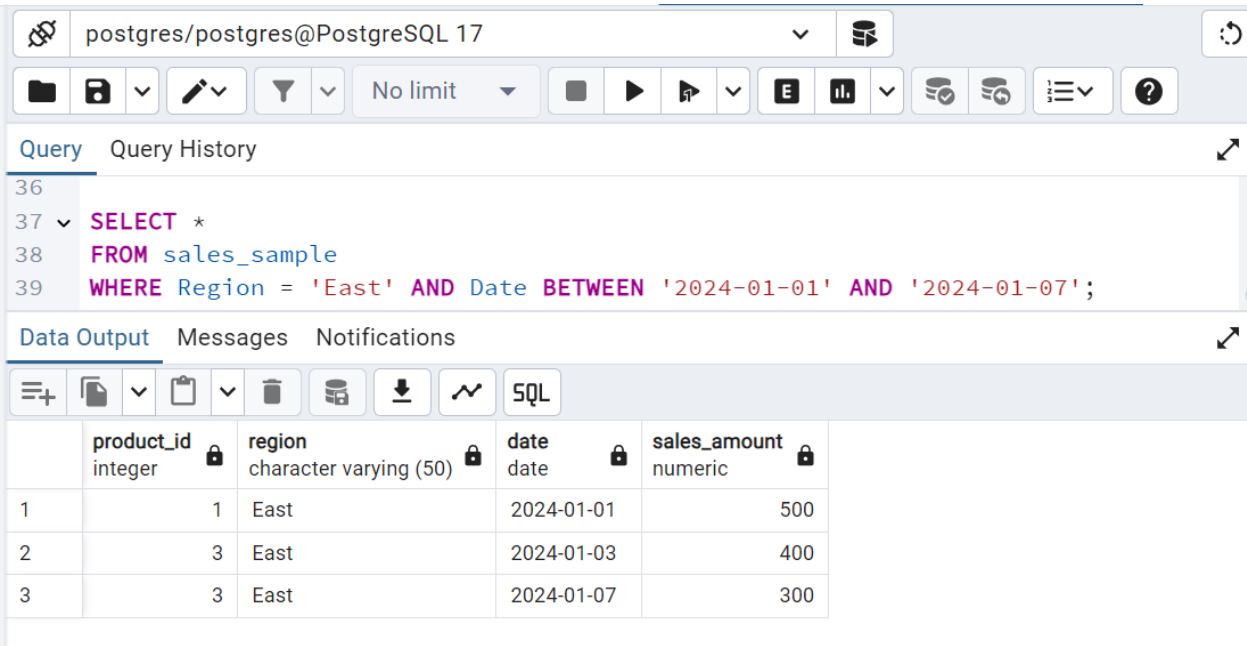
Data Output | Messages | Notifications

	region character varying (50)	product_id integer	date date	total_sales numeric
30	West	[null]	2024-01-06	550
31	West	[null]	2024-01-10	450
32	West	[null]	[null]	1700
33	[null]	1	2024-01-01	500
34	[null]	1	2024-01-06	550
35	[null]	1	2024-01-10	450
36	[null]	1	[null]	1500
37	[null]	2	2024-01-02	700
38	[null]	2	2024-01-08	650
39	[null]	2	[null]	1350
40	[null]	3	2024-01-03	400
41	[null]	3	2024-01-07	300
42	[null]	3	[null]	700
43	[null]	4	2024-01-04	800
44	[null]	4	2024-01-09	750

Data Output | Messages | Notifications

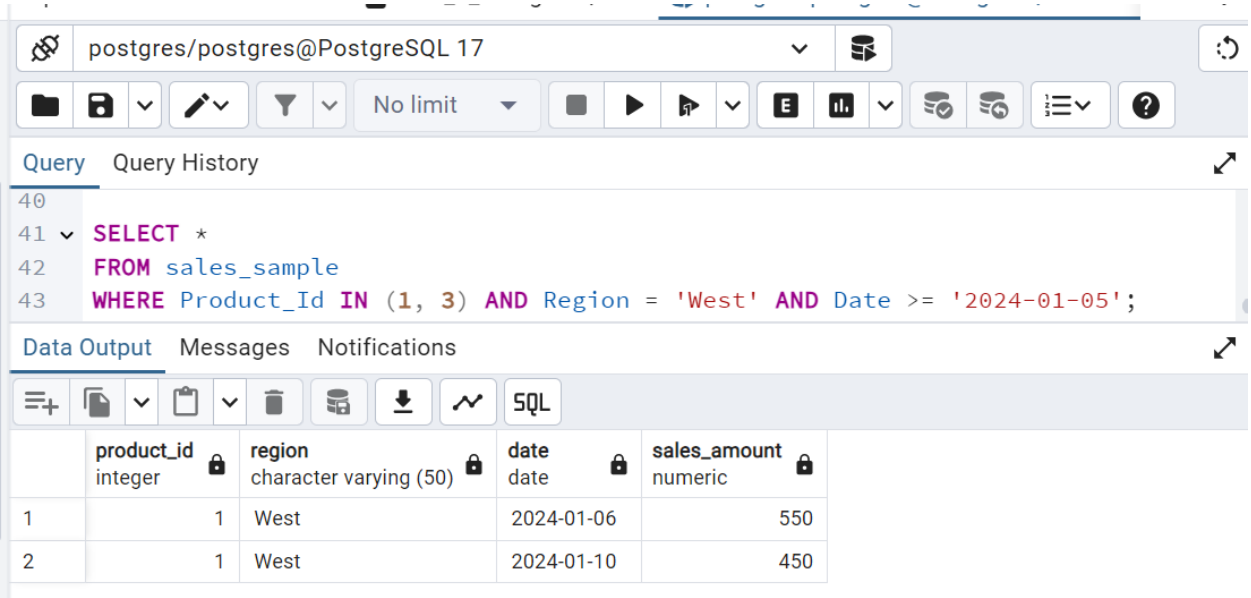
	region character varying (50)	product_id integer	date date	total_sales numeric
43	[null]	4	2024-01-04	800
44	[null]	4	2024-01-09	750
45	[null]	4	[null]	1550
46	[null]	5	2024-01-05	600
47	[null]	5	[null]	600
48	[null]	[null]	2024-01-01	500
49	[null]	[null]	2024-01-02	700
50	[null]	[null]	2024-01-03	400
51	[null]	[null]	2024-01-04	800
52	[null]	[null]	2024-01-05	600
53	[null]	[null]	2024-01-06	550
54	[null]	[null]	2024-01-07	300
55	[null]	[null]	2024-01-08	650
56	[null]	[null]	2024-01-09	750
57	[null]	[null]	2024-01-10	450
58	[null]	[null]	[null]	5700

d) Slice- To extract a subset of data based on specific criteria. Write a query to slice the data to view sales for a particular region or date range.



	product_id integer	region character varying (50)	date date	sales_amount numeric
1	1	East	2024-01-01	500
2	3	East	2024-01-03	400
3	3	East	2024-01-07	300

e) Dice - To extract data based on multiple criteria. Write a query to view sales for specific combinations of product, region, and date



	product_id integer	region character varying (50)	date date	sales_amount numeric
1	1	West	2024-01-06	550
2	1	West	2024-01-10	450

Query Explanation

1. Database Creation

1.Create the database:

```
CREATE DATABASE sales_data;
```

2.Create the sales_sample table:

```
CREATE TABLE sales_sample (
Product_Id INTEGER,
Region VARCHAR(50),
Date DATE, Sales_Amount NUMERIC );
```

2. Data Creation

```
INSERT INTO sales_sample (Product_Id, Region, Date, Sales_Amount) VALUES
(1, 'East', '2024-01-01', 500),
(2, 'West', '2024-01-02', 700),
(3, 'East', '2024-01-03', 400),
(4, 'North', '2024-01-04', 800),
(5, 'South', '2024-01-05', 600),
(1, 'West', '2024-01-06', 550),
(3, 'East', '2024-01-07', 300),
(2, 'North', '2024-01-08', 650),
(4, 'South', '2024-01-09', 750),
(1, 'West', '2024-01-10', 450);
```

Explanation: 10 sample records were inserted into the sales_sample table

3. OLAP Operations

a) Drill Down

Analyze sales data at a more detailed level (from region to product):

```
SELECT Region, Product_Id, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Region, Product_Id
ORDER BY Region, Product_Id;
```

Explanation: Groups the sales data by both Region and Product_Id to show the sales performance at a detailed level.

b) Rollup

Summarize sales data at different levels of granularity (from product to region):

```
SELECT Region, Product_Id, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
```

```
GROUP BY ROLLUP (Region, Product_Id)
ORDER BY Region, Product_Id;
```

Explanation:
The ROLLUP operator provides subtotals at each level.
Outputs total sales for each region, each product within the region, and a grand total for all regions.

c) Cube

Analyze sales data across multiple dimensions (product, region, and date):

```
SELECT Region, Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY CUBE (Region, Product_Id, Date)
ORDER BY Region, Product_Id, Date;
```

Explanation:
The CUBE operator generates all possible combinations of dimensions for aggregation.
Allows analysis of total sales by region, product, and date from various perspectives.

d) Slice

To extract a subset of data based on specific criteria. Write a query to slice the data to view sales for a particular region or date range.

```
SELECT *
FROM sales_sample
WHERE Region = 'East' AND Date BETWEEN '2024-01-01' AND '2024-01-07';
```

Explanation: Filters the sales data to show records for the East region within a specified date range.

e) Dice

Extract data based on multiple criteria (e.g., specific product, region, and date):

```
SELECT *
FROM sales_sample
WHERE Product_Id IN (1, 3) AND Region = 'West' AND Date >= '2024-01-05';
```

Explanation: Combines filters for product IDs, region, and date to extract the desired subset of sales data.