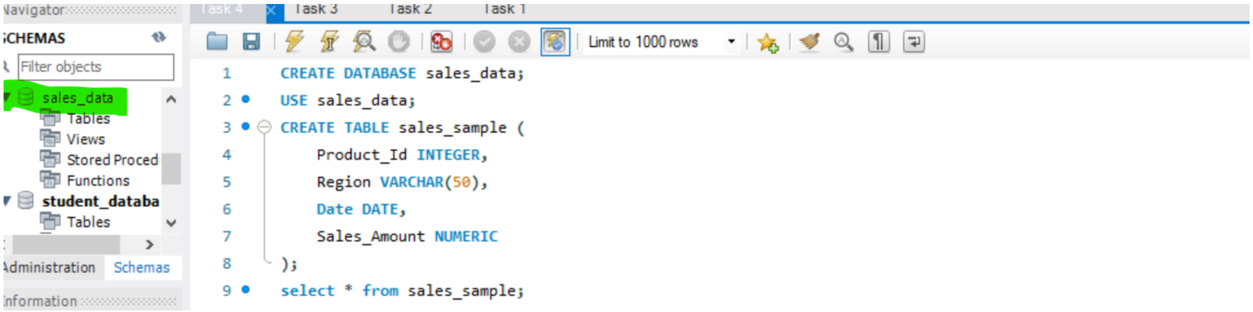
# OUTPUT SHEET

## Project: OLAP Operations using MySQL

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

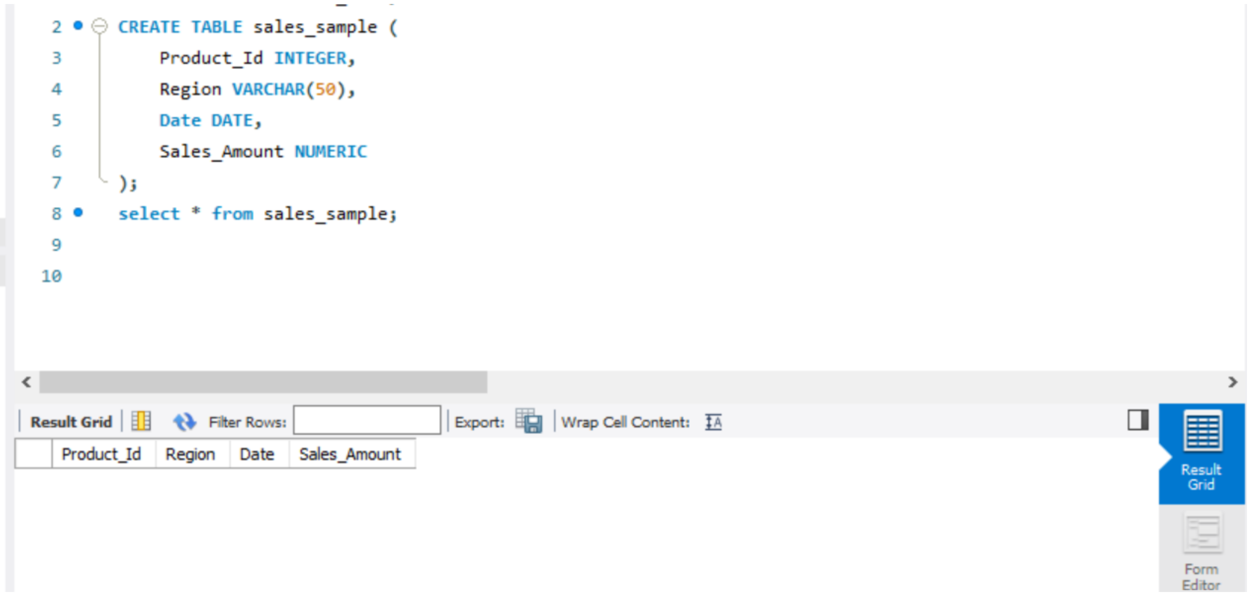
a. Create a database to store the sales data.



```
CREATE DATABASE sales_data;
USE sales_data;
CREATE TABLE sales_sample (
    Product_Id INTEGER,
    Region VARCHAR(50),
    Date DATE,
    Sales_Amount NUMERIC
);
select * from sales_sample;
```

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)

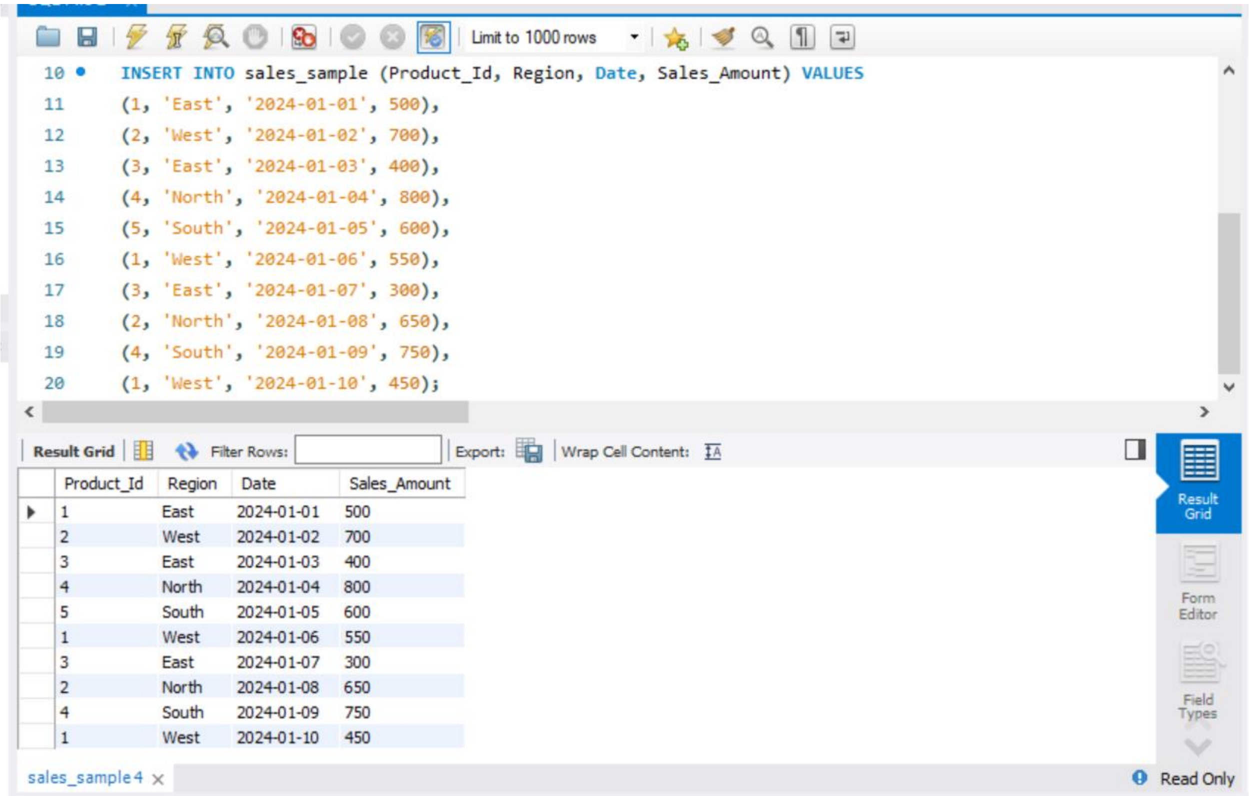


```
CREATE TABLE sales_sample (
    Product_Id INTEGER,
    Region VARCHAR(50),
    Date DATE,
    Sales_Amount NUMERIC
);
select * from sales_sample;
```

| Product_Id | Region | Date | Sales_Amount |
|---|---|---|---|

## 2. Data Creation

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

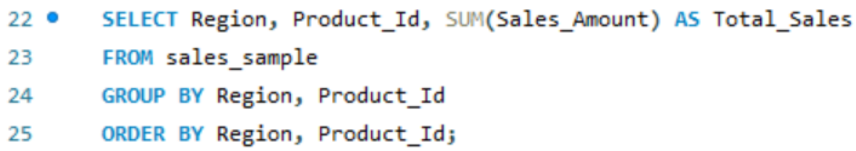


```
INSERT INTO sales_sample (Product_Id, Region, Date, Sales_Amount) VALUES
(1, 'East', '2024-01-01', 500),
(2, 'West', '2024-01-02', 700),
(3, 'East', '2024-01-03', 400),
(4, 'North', '2024-01-04', 800),
(5, 'South', '2024-01-05', 600),
(1, 'West', '2024-01-06', 550),
(3, 'East', '2024-01-07', 300),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

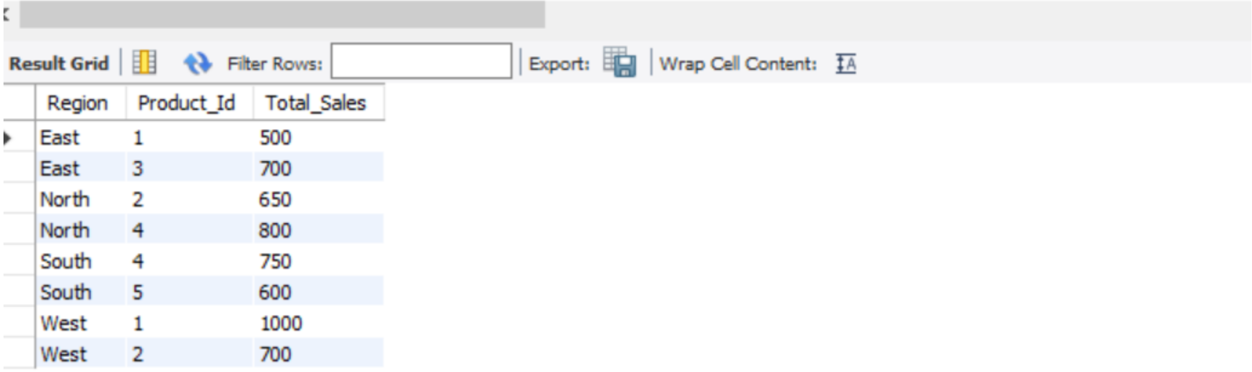
| Product_Id | Region | Date | Sales_Amount |
|---|---|---|---|
| 1 | East | 2024-01-01 | 500 |
| 2 | West | 2024-01-02 | 700 |
| 3 | East | 2024-01-03 | 400 |
| 4 | North | 2024-01-04 | 800 |
| 5 | South | 2024-01-05 | 600 |
| 1 | West | 2024-01-06 | 550 |
| 3 | East | 2024-01-07 | 300 |
| 2 | North | 2024-01-08 | 650 |
| 4 | South | 2024-01-09 | 750 |
| 1 | West | 2024-01-10 | 450 |

## 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.

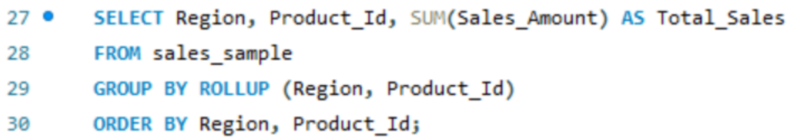


```
SELECT Region, Product_Id, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Region, Product_Id
ORDER BY Region, Product_Id;
```

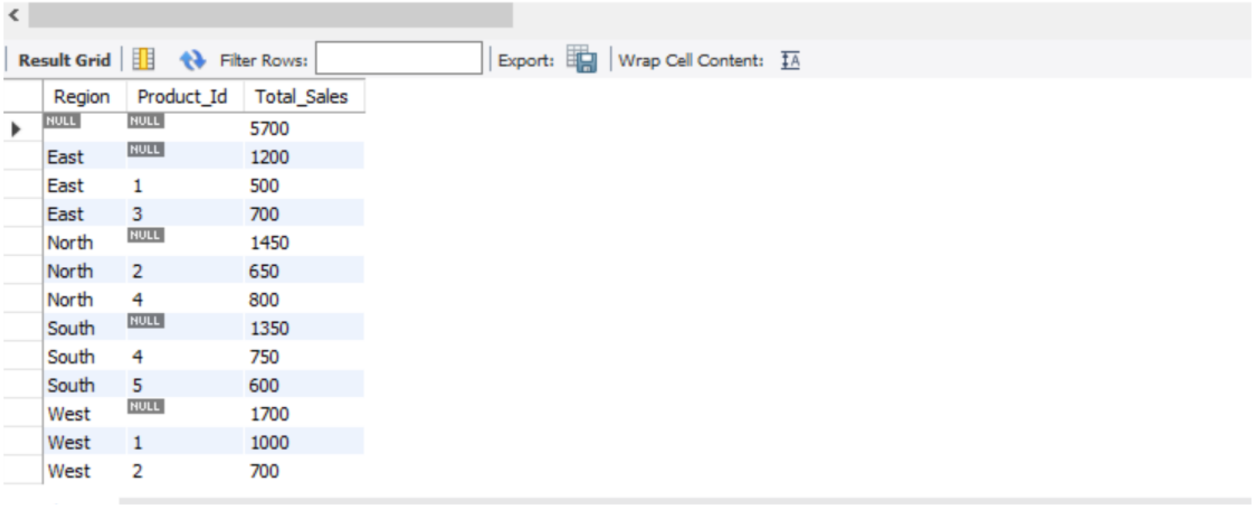


| Region | Product_Id | Total_Sales |
|---|---|---|
| East | 1 | 500 |
| East | 3 | 700 |
| North | 2 | 650 |
| North | 4 | 800 |
| South | 4 | 750 |
| South | 5 | 600 |
| West | 1 | 1000 |
| West | 2 | 700 |

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.

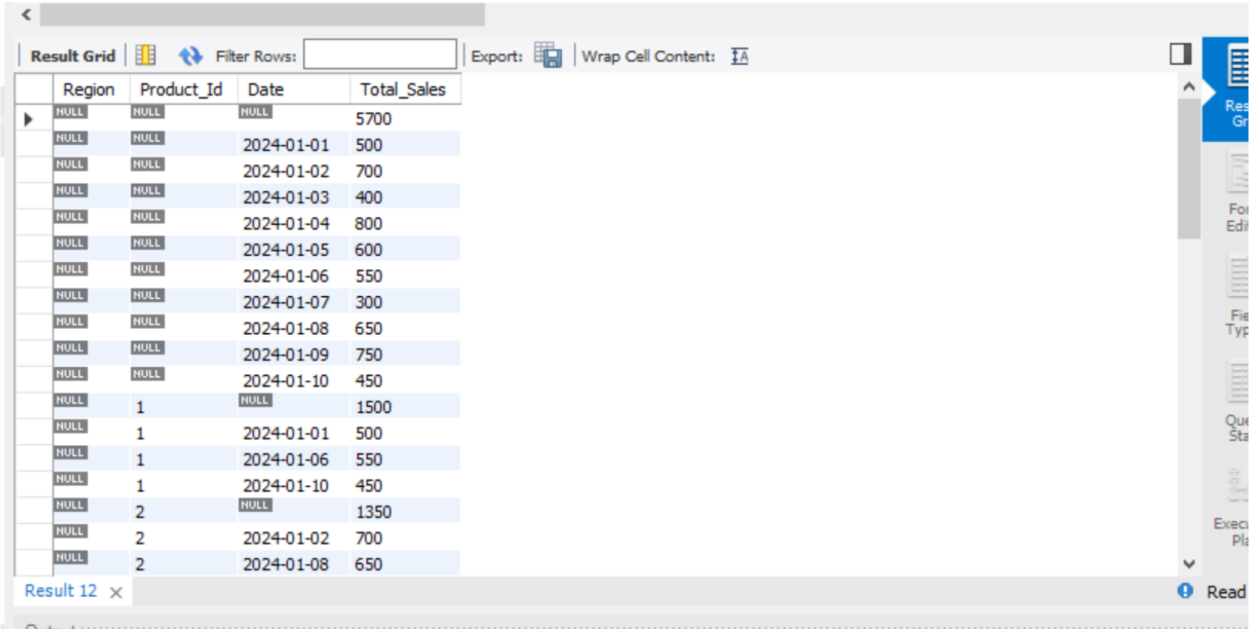


```
SELECT Region, Product_Id, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY ROLLUP (Region, Product_Id)
ORDER BY Region, Product_Id;
```

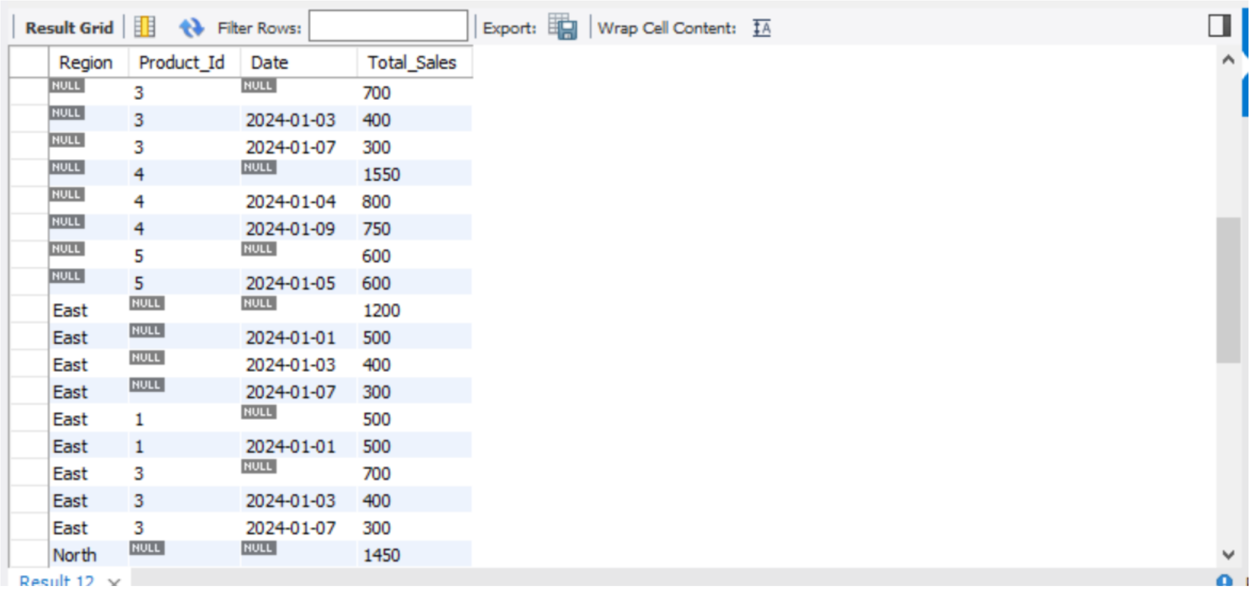


| Region | Product_Id | Total_Sales |
|---|---|---|
| NULL | NULL | 5700 |
| East | NULL | 1200 |
| East | 1 | 500 |
| East | 3 | 700 |
| North | NULL | 1450 |
| North | 2 | 650 |
| North | 4 | 800 |
| South | NULL | 1350 |
| South | 4 | 750 |
| South | 5 | 600 |
| West | NULL | 1700 |
| West | 1 | 1000 |
| West | 2 | 700 |

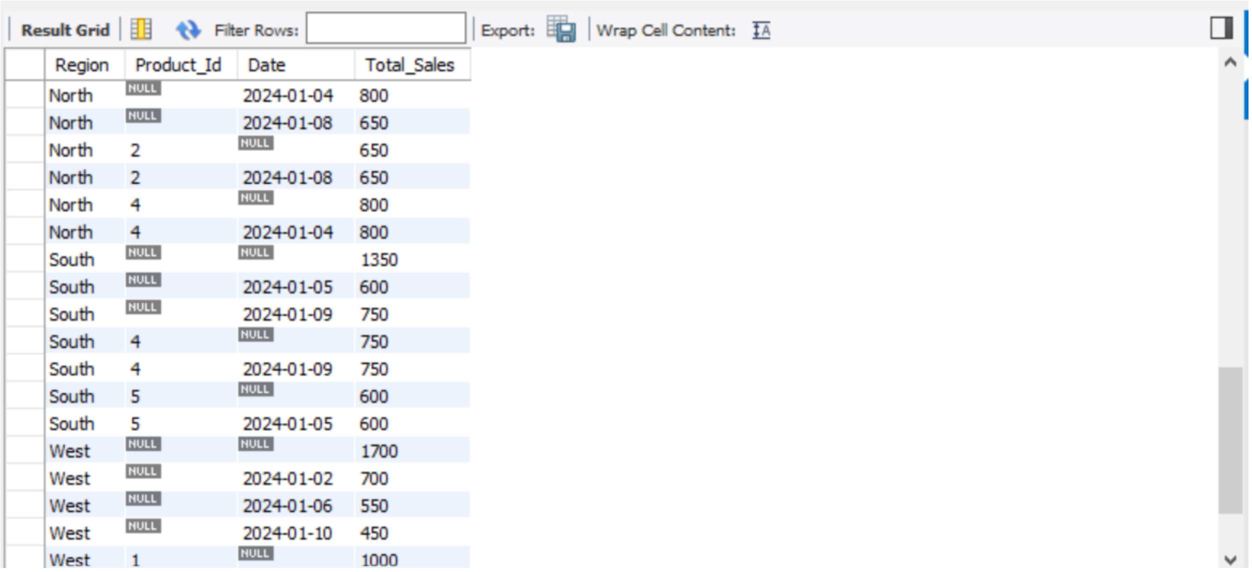
c) Cube - To analyze sales data from multiple dimensions simultaneously. Write a query to Explore sales data from different perspectives, such as product, region, and date.



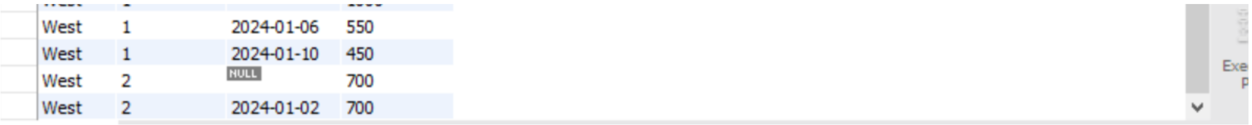
| Region | Product_Id | Date | Total_Sales |
|---|---|---|---|
| NULL | NULL | NULL | 5700 |
| NULL | NULL | 2024-01-01 | 500 |
| NULL | NULL | 2024-01-02 | 700 |
| NULL | NULL | 2024-01-03 | 400 |
| NULL | NULL | 2024-01-04 | 800 |
| NULL | NULL | 2024-01-05 | 600 |
| NULL | NULL | 2024-01-06 | 550 |
| NULL | NULL | 2024-01-07 | 300 |
| NULL | NULL | 2024-01-08 | 650 |
| NULL | NULL | 2024-01-09 | 750 |
| NULL | NULL | 2024-01-10 | 450 |
| NULL | 1 | NULL | 1500 |
| NULL | 1 | 2024-01-01 | 500 |
| NULL | 1 | 2024-01-06 | 550 |
| NULL | 1 | 2024-01-10 | 450 |
| NULL | 2 | NULL | 1350 |
| NULL | 2 | 2024-01-02 | 700 |
| NULL | 2 | 2024-01-08 | 650 |



| Region | Product_Id | Date | Total_Sales |
|---|---|---|---|
| NULL | 3 | NULL | 700 |
| NULL | 3 | 2024-01-03 | 400 |
| NULL | 3 | 2024-01-07 | 300 |
| NULL | 4 | NULL | 1550 |
| NULL | 4 | 2024-01-04 | 800 |
| NULL | 4 | 2024-01-09 | 750 |
| NULL | 5 | NULL | 600 |
| NULL | 5 | 2024-01-05 | 600 |
| East | NULL | NULL | 1200 |
| East | NULL | 2024-01-01 | 500 |
| East | NULL | 2024-01-03 | 400 |
| East | NULL | 2024-01-07 | 300 |
| East | 1 | NULL | 500 |
| East | 1 | 2024-01-01 | 500 |
| East | 3 | NULL | 700 |
| East | 3 | 2024-01-03 | 400 |
| East | 3 | 2024-01-07 | 300 |
| North | NULL | NULL | 1450 |

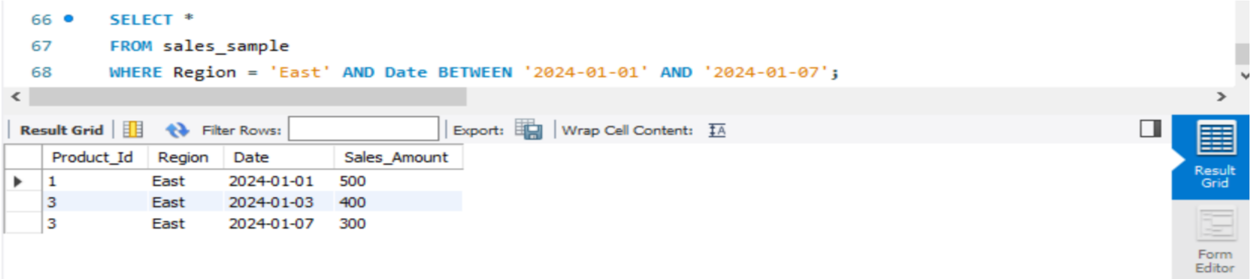


| Region | Product_Id | Date | Total_Sales |
|---|---|---|---|
| North | NULL | 2024-01-04 | 800 |
| North | NULL | 2024-01-08 | 650 |
| North | 2 | NULL | 650 |
| North | 2 | 2024-01-08 | 650 |
| North | 4 | NULL | 800 |
| North | 4 | 2024-01-04 | 800 |
| South | NULL | NULL | 1350 |
| South | NULL | 2024-01-05 | 600 |
| South | NULL | 2024-01-09 | 750 |
| South | 4 | NULL | 750 |
| South | 4 | 2024-01-09 | 750 |
| South | 5 | NULL | 600 |
| South | 5 | 2024-01-05 | 600 |
| West | NULL | NULL | 1700 |
| West | NULL | 2024-01-02 | 700 |
| West | NULL | 2024-01-06 | 550 |
| West | NULL | 2024-01-10 | 450 |
| West | 1 | NULL | 1000 |



| | | | |
|---|---|---|---|
| West | 1 | 2024-01-06 | 550 |
| West | 1 | 2024-01-10 | 450 |
| West | 2 | NULL | 700 |
| West | 2 | 2024-01-02 | 700 |

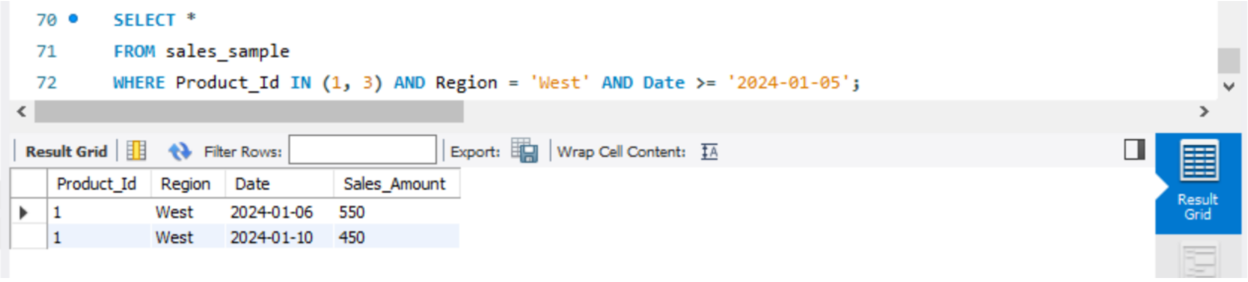
d) Slice- To extract a subset of data based on specific criteria. Write a query to slice the data to view sales for a particular region or date range.



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

| Product_Id | Region | Date | Sales_Amount |
|---|---|---|---|
| 1 | East | 2024-01-01 | 500 |
| 3 | East | 2024-01-03 | 400 |
| 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



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

| Product_Id | Region | Date | Sales_Amount |
|---|---|---|---|
| 1 | West | 2024-01-06 | 550 |
| 1 | West | 2024-01-10 | 450 |

# Query Explanation

## 1. Database Creation

1.Create the database:

CREATE DATABASE sales_data;

USE 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.

OR

While using MySQL we can replace cube function using UNION as MySQL doesn’t supports cube function. Below is the Query we can use while using MySQL platform.

1.Aggregation by Region, Product_Id, and Date

```
SELECT Region, Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Region, Product_Id, Date
UNION ALL
```

2. Aggregation by Region and Product_Id (ignoring Date)

```
SELECT Region, Product_Id, NULL AS Date, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Region, Product_Id
UNION ALL
```

3. Aggregation by Region and Date (ignoring Product_Id)

```
SELECT Region, NULL AS Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Region, Date
UNION ALL
```

4. Aggregation by Product_Id and Date (ignoring Region)

```
SELECT NULL AS Region, Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Product_Id, Date
UNION ALL
```

5. Aggregation by Region only (ignoring Product_Id and Date)

```
SELECT Region, NULL AS Product_Id, NULL AS Date, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Region
UNION ALL
```

6. Aggregation by Product_Id only (ignoring Region and Date)

```
SELECT NULL AS Region, Product_Id, NULL AS Date, SUM(Sales_Amount) AS Total_Sales
FROM sales_sample
GROUP BY Product_Id
UNION ALL
```

7. Aggregation by Date only (ignoring Region and Product_Id)

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

```
GROUP BY Date
UNION ALL
```

8. Grand Total (ignoring Region, Product_Id, and Date)

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

### 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.