

QUERY 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.

```
1 CREATE DATABASE sales_data;
```

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 CREATE TABLE sales_sample (  
3     Product_Id INTEGER,  
4     Region VARCHAR(50),  
5     Date DATE,  
6     Sales_Amount NUMERIC  
7 );
```

2. Data Creation

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

```
10 INSERT INTO sales_sample (Product_Id, Region, Date, Sales_Amount) VALUES  
11 (1, 'East', '2024-01-01', 500),  
12 (2, 'West', '2024-01-02', 700),  
13 (3, 'East', '2024-01-03', 400),  
14 (4, 'North', '2024-01-04', 800),  
15 (5, 'South', '2024-01-05', 600),  
16 (1, 'West', '2024-01-06', 550),  
17 (3, 'East', '2024-01-07', 300),  
18 (2, 'North', '2024-01-08', 650),  
19 (4, 'South', '2024-01-09', 750),  
20 (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.

```
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;
```

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.

```
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;
```

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.

```
27 • SELECT Region, Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
28     FROM sales_sample
29     GROUP BY Region, Product_Id, Date
30     UNION ALL
31     -- Aggregation by Region and Product_Id (ignoring Date)
32     SELECT Region, Product_Id, NULL AS Date, SUM(Sales_Amount) AS Total_Sales
33     FROM sales_sample
34     GROUP BY Region, Product_Id
35     UNION ALL
36     -- Aggregation by Region and Date (ignoring Product_Id)
37     SELECT Region, NULL AS Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
38     FROM sales_sample
39     GROUP BY Region, Date
40     UNION ALL
41     -- Aggregation by Product_Id and Date (ignoring Region)
42     SELECT NULL AS Region, Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
43     FROM sales_sample
44     GROUP BY Product_Id, Date
45     UNION ALL
46     -- Aggregation by Region only (ignoring Product_Id and Date)
47     SELECT Region, NULL AS Product_Id, NULL AS Date, SUM(Sales_Amount) AS Total_Sales
48     FROM sales_sample
49     GROUP BY Region
50     UNION ALL
51     -- Aggregation by Product_Id only (ignoring Region and Date)
52     SELECT NULL AS Region, Product_Id, NULL AS Date, SUM(Sales_Amount) AS Total_Sales
53     FROM sales_sample
54     GROUP BY Product_Id
55     UNION ALL
```

```

56  -- Aggregation by Date only (ignoring Region and Product_Id)
57  SELECT NULL AS Region, NULL AS Product_Id, Date, SUM(Sales_Amount) AS Total_Sales
58  FROM sales_sample
59  GROUP BY Date
60  UNION ALL
61  -- Grand Total (ignoring Region, Product_Id, and Date)
62  SELECT NULL AS Region, NULL AS Product_Id, NULL AS Date, SUM(Sales_Amount) AS Total_Sales
63  FROM sales_sample
64  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.

```

66 •  SELECT *
67      FROM sales_sample
68      WHERE Region = 'East' AND Date BETWEEN '2024-01-01' AND '2024-01-07';

```

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

```

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

```

Full Query

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 );
```

```
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);
```

3. OLAP Operations

a) Drill Down

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

b) Rollup

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

c) Cube

```
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;
```

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.

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

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

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

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

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

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

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

```
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

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

e) Dice

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