

## [ Task 4 ] Project: OLAP Operations

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:

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

The screenshot shows a SQL IDE with a script file named 'task4\_olap\_mysql.sql'. The script contains the following SQL statements:

```
1 -- File: task4_olap_mysql.sql
2 DROP DATABASE IF EXISTS sales_olap;
3 CREATE DATABASE sales_olap;
4 USE sales_olap;
5
6 CREATE TABLE sales_sample (
7   Product_Id INT NOT NULL,
8   Region VARCHAR(50) NOT NULL,
9   Date DATE NOT NULL,
10  Sales_Amount DECIMAL(12,2) NOT NULL CHECK (Sales_Amount >= 0)
11 ) ENGINE=InnoDB;
12
13 -- India-localized 10 rows (regions ~ India zones)
14 INSERT INTO sales_sample (Product_Id, Region, Date, Sales_Amount) VALUES
15 (101, 'North', '2025-01-05', 120000.00),
16 (101, 'West', '2025-01-06', 95000.00),
17 (102, 'South', '2025-01-07', 150000.00),
18 (102, 'West', '2025-01-08', 70000.00),
19 (103, 'East', '2025-02-10', 30000.00),
20 (103, 'West', '2025-02-11', 65000.00),
21 (101, 'North', '2025-02-12', 40000.00),
22 (102, 'South', '2025-03-01', 80000.00),
23 (103, 'East', '2025-03-02', 90000.00),
24 (101, 'West', '2025-03-03', 110000.00);
25
26 -- 3a) Drill Down: region -> product totals (WITH ROLLUP)
27 SELECT Region, Product Id, SUM(Sales Amount) AS total sales
```

The output window shows the execution results of the first four statements:

#	Time	Action	Message	Duration / Fetch
1	16:36:02	DROP DATABASE IF EXISTS sales_olap	1 row(s) affected	0.078 sec
2	16:36:02	CREATE DATABASE sales_olap	1 row(s) affected	0.016 sec
3	16:36:02	USE sales_olap	0 row(s) affected	0.000 sec
4	16:36:02	CREATE TABLE sales_sample ( Product_Id INT NOT NULL, Region VARCHAR(50) NOT NULL, Date DATE NOT N...	0 row(s) affected	0.031 sec

### 2) Data Creation:

- Insert 10 sample records into the "sales\_sample" table, representing sales data.

SQL File 6\* SQL File 5\* SQL File 4\* SQL File 3\*

```

1 -- File: task4_olap_mysql.sql
2 • DROP DATABASE IF EXISTS sales_olap;
3 • CREATE DATABASE sales_olap;
4 • USE sales_olap;
5
6 • CREATE TABLE sales_sample (
7     Product_Id INT NOT NULL,
8     Region VARCHAR(50) NOT NULL,
9     Date DATE NOT NULL,
10    Sales_Amount DECIMAL(12,2) NOT NULL CHECK (Sales_Amount >= 0)
11 ) ENGINE=InnoDB;
12
13 -- India-localized 10 rows (regions ~ India zones)
14 • INSERT INTO sales_sample (Product_Id, Region, Date, Sales_Amount) VALUES
15 (101, 'North', '2025-01-05', 120000.00),
16 (101, 'West', '2025-01-06', 95000.00),
17 (102, 'South', '2025-01-07', 150000.00),
18 (102, 'West', '2025-01-08', 70000.00),
19 (103, 'East', '2025-02-10', 30000.00),
20 (103, 'West', '2025-02-11', 65000.00),
21 (101, 'North', '2025-02-12', 40000.00),
22 (102, 'South', '2025-03-01', 80000.00),
23 (103, 'East', '2025-03-02', 90000.00),
24 (101, 'West', '2025-03-03', 110000.00);
25
26 -- 3a) Drill Down: region -> product totals (WITH ROLLUP)
27 • SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales

```

Output

#	Time	Action	Message	Duration / Fetch
1	16:36:02	DROP DATABASE IF EXISTS sales_olap	1 row(s) affected	0.078 sec
2	16:36:02	CREATE DATABASE sales_olap	1 row(s) affected	0.016 sec
3	16:36:02	USE sales_olap	0 row(s) affected	0.000 sec
4	16:36:02	CREATE TABLE sales_sample ( Product_Id INT NOT NULL, Region VARCHAR(50) NOT NULL, Date DATE NOT N...	0 row(s) affected	0.031 sec
5	16:37:12	INSERT INTO sales_sample (Product_Id, Region, Date, Sales_Amount) VALUES (101, 'North', '2025-01-05', 120000.00), (...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.016 sec

### 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.

SQL File 6\* SQL File 5\* SQL File 4\* SQL File 3\*

```

20 (103, 'West', '2025-02-11', 65000.00),
21 (101, 'North', '2025-02-12', 40000.00),
22 (102, 'South', '2025-03-01', 80000.00),
23 (103, 'East', '2025-03-02', 90000.00),
24 (101, 'West', '2025-03-03', 110000.00);
25
26 -- 3a) Drill Down: Single Region
27 • SELECT
28     Region,
29     Product_Id,
30     SUM(Sales_Amount) AS total_sales
31 FROM sales_sample
32 WHERE Region = 'West'
33 GROUP BY Region, Product_Id WITH ROLLUP
34 ORDER BY
35     Region IS NULL, Region,
36     Product_Id IS NULL, Product_Id;
37
38
39 -- 3b) Rollup: totals by region + grand total

```

Result Grid

Region	Product_Id	total_sales
West		850000.00
East		120000.00
East	103	120000.00
North		160000.00
North	101	160000.00

Output

#	Time	Action	Message	Duration / Fetch
1	16:43:53	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Region, Product_I...	11 row(s) returned	0.000 sec / 0.000 sec
2	16:47:25	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Region = 'West'...	5 row(s) returned	0.016 sec / 0.000 sec
3	16:48:08	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Region, Produ...	11 row(s) returned	0.000 sec / 0.000 sec

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.

The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results grid. The query is as follows:

```
-- 3a) Drill Down: Single Region
26
27 • SELECT
28     Region,
29     Product_Id,
30     SUM(Sales_Amount) AS total_sales
31 FROM sales_sample
32 WHERE Region = 'West'
33 GROUP BY Region, Product_Id WITH ROLLUP
34 ORDER BY
35     Region IS NULL, Region,
36     Product_Id IS NULL, Product_Id;
37
38
39 -- 3b) Rollup: totals by region + grand total
40 • SELECT Region, SUM(Sales_Amount) AS region_total
41 FROM sales_sample
42 GROUP BY Region WITH ROLLUP
43 ORDER BY Region IS NULL, Region;
44
45 -- 3c) Cube emulation (all grouping sets) via UNION ALL
```

The results grid shows the following data:

Region	region_total
West	850000.00
East	120000.00
North	160000.00
South	230000.00
West	340000.00

The output pane shows the following actions:

#	Time	Action	Message	Duration / Fetch
1	16:43:53	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Region, Product_Id	11 row(s) returned	0.000 sec / 0.000 sec
2	16:47:25	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Region = 'West' GROUP BY Region, Product_Id WITH ROLLUP	5 row(s) returned	0.016 sec / 0.000 sec
3	16:48:08	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Region, Product_Id	11 row(s) returned	0.000 sec / 0.000 sec
4	16:50:28	SELECT Region, SUM(Sales_Amount) AS region_total FROM sales_sample GROUP BY Region WITH ROLLUP ORDER BY Region IS NULL, Region	5 row(s) returned	0.000 sec / 0.000 sec

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.

The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results grid. The query is as follows:

```
62 SELECT Product_Id, NULL, NULL, SUM(Sales_Amount)
63 FROM sales_sample
64 GROUP BY Product_Id
65 UNION ALL
66 SELECT NULL, Region, NULL, SUM(Sales_Amount)
67 FROM sales_sample
68 GROUP BY Region
69 UNION ALL
70 SELECT NULL, NULL, Date, SUM(Sales_Amount)
71 FROM sales_sample
72 GROUP BY Date
73 UNION ALL
74 SELECT NULL, NULL, NULL, SUM(Sales_Amount)
75 FROM sales_sample;
76
77 -- 3d) Slice: specific region or date range
78 • SELECT Product_Id, Date, SUM(Sales_Amount) AS total_sales
79 FROM sales_sample
80 WHERE Region = 'West'
81 GROUP BY Product_Id, Date
```

The results grid shows the following data:

Product_Id	Region	Date	total_sales
101	North	2025-01-05	120000.00
101	West	2025-01-06	95000.00
102	South	2025-01-07	150000.00
102	West	2025-01-08	70000.00
103	East	2025-02-10	30000.00

The output pane shows the following actions:

#	Time	Action	Message	Duration / Fetch
1	16:43:53	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Region, Product_Id	11 row(s) returned	0.000 sec / 0.000 sec
2	16:47:25	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Region = 'West' GROUP BY Region, Product_Id WITH ROLLUP	5 row(s) returned	0.016 sec / 0.000 sec
3	16:48:08	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Region, Product_Id	11 row(s) returned	0.000 sec / 0.000 sec
4	16:50:28	SELECT Region, SUM(Sales_Amount) AS region_total FROM sales_sample GROUP BY Region WITH ROLLUP ORDER BY Region IS NULL, Region	5 row(s) returned	0.000 sec / 0.000 sec
5	16:50:59	SELECT Product_Id, Region, Date, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Product_Id, Region, Date	54 row(s) returned	0.016 sec / 0.000 sec

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.

The screenshot shows a SQL Server Enterprise Manager interface. The query window contains the following SQL code:

```
73 UNION ALL
74 SELECT NULL, NULL, NULL, SUM(Sales_Amount)
75 FROM sales_sample;
76
77 -- 3d) Slice: specific region or date range
78 • SELECT Product_Id, Date, SUM(Sales_Amount) AS total_sales
79 FROM sales_sample
80 WHERE Region = 'West'
81 GROUP BY Product_Id, Date
82 ORDER BY Product_Id, Date;
83
84 • SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales
85 FROM sales_sample
86 WHERE Date BETWEEN '2025-02-01' AND '2025-02-28'
87 GROUP BY Region, Product_Id
88 ORDER BY Region, Product_Id;
89
90 -- 3e) Dice: multiple filters
91 • SELECT Product_Id, Region, Date, SUM(Sales_Amount) AS total_sales
92 FROM sales_sample
```

The results grid shows the output of the queries:

Region	Product_Id	total_sales
East	103	30000.00
North	101	40000.00
West	103	65000.00

The output pane shows the execution of the queries:

#	Time	Action	Message	Duration / Fetch
3	16:48:08	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Region, Product_Id	11 row(s) returned	0.000 sec / 0.000 sec
4	16:50:28	SELECT Region, SUM(Sales_Amount) AS region_total FROM sales_sample GROUP BY Region WITH ROLLUP OR...	5 row(s) returned	0.000 sec / 0.000 sec
5	16:50:59	SELECT Product_Id, Region, Date, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Product_Id, Date	54 row(s) returned	0.016 sec / 0.000 sec
6	16:52:42	SELECT Product_Id, Date, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Region = 'West' GROUP BY Product_Id, Date	4 row(s) returned	0.000 sec / 0.000 sec
7	16:52:42	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Date BETWEEN '2025-02-01' AND '2025-02-28' GROUP BY Region, Product_Id	3 row(s) returned	0.000 sec / 0.000 sec

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

The screenshot shows a SQL Server Enterprise Manager interface. The query window contains the following SQL code:

```
78 • SELECT Product_Id, Date, SUM(Sales_Amount) AS total_sales
79 FROM sales_sample
80 WHERE Region = 'West'
81 GROUP BY Product_Id, Date
82 ORDER BY Product_Id, Date;
83
84 • SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales
85 FROM sales_sample
86 WHERE Date BETWEEN '2025-02-01' AND '2025-02-28'
87 GROUP BY Region, Product_Id
88 ORDER BY Region, Product_Id;
89
90 -- 3e) Dice: multiple filters
91 • SELECT Product_Id, Region, Date, SUM(Sales_Amount) AS total_sales
92 FROM sales_sample
93 WHERE Product_Id IN (101,102)
94 AND Region IN ('North','West')
95 AND Date BETWEEN '2025-01-01' AND '2025-03-31'
96 GROUP BY Product_Id, Region, Date
97 ORDER BY Product_Id, Region, Date;
```

The results grid shows the output of the queries:

Product_Id	Region	Date	total_sales
101	North	2025-01-05	120000.00
101	North	2025-02-12	40000.00
101	West	2025-01-06	95000.00
101	West	2025-03-03	110000.00
102	West	2025-01-08	70000.00

The output pane shows the execution of the queries:

#	Time	Action	Message	Duration / Fetch
4	16:50:28	SELECT Region, SUM(Sales_Amount) AS region_total FROM sales_sample GROUP BY Region WITH ROLLUP OR...	5 row(s) returned	0.000 sec / 0.000 sec
5	16:50:59	SELECT Product_Id, Region, Date, SUM(Sales_Amount) AS total_sales FROM sales_sample GROUP BY Product_Id, Date	54 row(s) returned	0.016 sec / 0.000 sec
6	16:52:42	SELECT Product_Id, Date, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Region = 'West' GROUP BY Product_Id, Date	4 row(s) returned	0.000 sec / 0.000 sec
7	16:52:42	SELECT Region, Product_Id, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Date BETWEEN '2025-02-01' AND '2025-02-28' GROUP BY Region, Product_Id	3 row(s) returned	0.000 sec / 0.000 sec
8	16:53:17	SELECT Product_Id, Region, Date, SUM(Sales_Amount) AS total_sales FROM sales_sample WHERE Product_Id IN (101,102) AND Region IN ('North','West') AND Date BETWEEN '2025-01-01' AND '2025-03-31' GROUP BY Product_Id, Region, Date	5 row(s) returned	0.000 sec / 0.000 sec

