Name: Asmi Jain

Div: TY-3

Roll No: 25

Batch: B

Subject: DWM

Experiment No 02

AIM : Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot for the above problem statement (experiment 1).

Problem statement :

Design a data warehouse for a regional weather bureau. The weather bureau has about 100 probs, which are scattered throughout various land and ocean locations in the region to collect basic weather data, including air pressure, temperature and precipitation at each hour. All data are sent to the central station, which has collected such data for more than 10 years. Design Star schema and Snowflake schema such that it should facilitate efficient querying and online analytical processing and derive general weather patterns in multidimensional space. Explain all aspects of the diagram. Design Star and Snowflake schema for above case.

THEORY :

Online Analytical Processing (OLAP) is a technology that enables analysts to extract and view data from different points of view. OLAP operations are designed to facilitate multidimensional analysis of business data for decision-making purposes. The core OLAP operations include:

1. Slice Operation

Slicing involves selecting a specific value for one dimension and viewing the resulting "slice" of the data cube. It's like cutting a slice from a cube, reducing the dimensionality by one while keeping all other dimensions intact.

2. Dice Operation

Dicing involves selecting a subcube by applying restrictions on two or more dimensions simultaneously. Unlike slicing which reduces dimensionality, dicing maintains the same number of dimensions but with fewer values in each.

3. Roll-up (Aggregation)

Roll-up performs aggregation on a data cube by climbing up a concept hierarchy for a dimension or by reducing dimensions. It allows for summarizing data at different levels of granularity, moving from detailed to more summarized views.

4. Drill-down

Drill-down is the opposite of roll-up. It navigates from less detailed data to more detailed data, descending the concept hierarchy for a dimension or introducing additional dimensions for more granular analysis.

5. Pivot (Rotate)

Pivoting rotates the data axes to provide an alternative presentation of data. It helps in viewing the same information from different perspectives by reorganizing the multidimensional view.

Code :

```
-- Slice: Get all temperature measurements for a specific location (LocationID = 1)
SELECT
    t.Hour,
    l.LocationName,
    mt.TypeName,
    wm.Value
FROM
    WeatherMeasurements wm
JOIN
    TimeDim t ON wm.TimeID = t.TimeID
JOIN
    LocationDim l ON wm.LocationID = l.LocationID
JOIN
    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
    l.LocationID = 1 AND mt.TypeName = 'Temperature';


-- Dice: Get all measurements for a specific location, time range, and measurement types
SELECT
    t.Hour,
    l.LocationName,
    p.ProbeModel,
    mt.TypeName,
    mt.Unit,
```

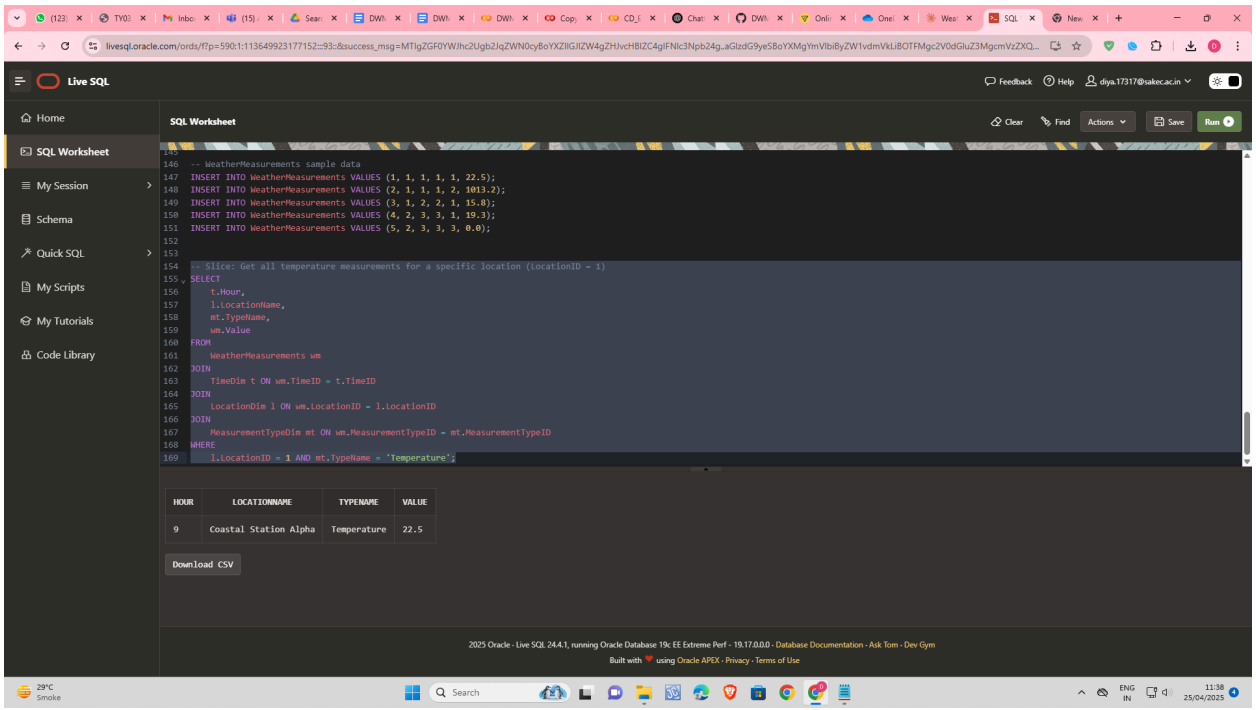
```
    wm.Value
FROM
    WeatherMeasurements wm
JOIN
    TimeDim t ON wm.TimeID = t.TimeID
JOIN
    LocationDim l ON wm.LocationID = l.LocationID
JOIN
    ProbeDim p ON wm.ProbeID = p.ProbeID
JOIN
    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
    l.LocationID = 1 AND
    t.Hour BETWEEN 9 AND 11 AND
    mt.TypeName IN ('Temperature', 'Pressure');



-- Roll-up: Aggregate temperature data by location type
SELECT
    l.LocationType,
    mt.TypeName,
    AVG(wm.Value) as AvgValue,
    MIN(wm.Value) as MinValue,
    MAX(wm.Value) as MaxValue,
    COUNT(*) as NumberOfReadings
FROM
```

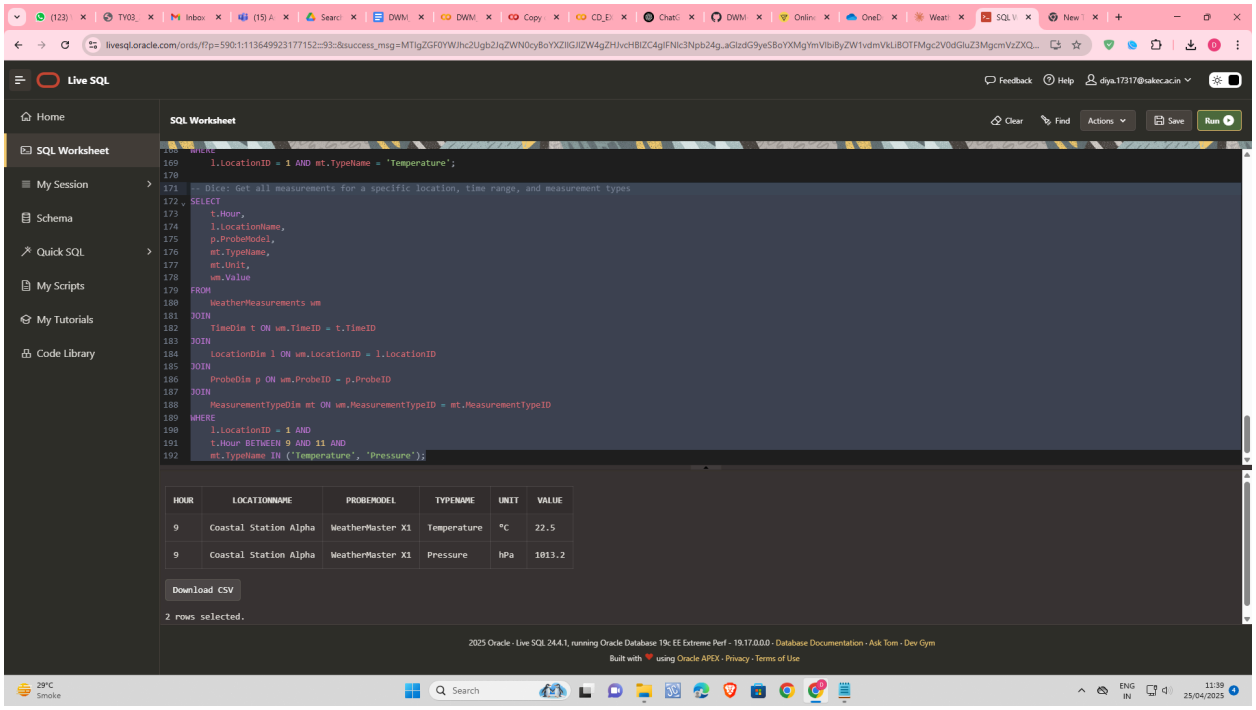
```
    WeatherMeasurements wm
JOIN
    LocationDim l ON wm.LocationID = l.LocationID
JOIN
    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
    mt.TypeName = 'Temperature'
GROUP BY
    l.LocationType, mt.TypeName;



-- Drill-down: From region level to specific locations
SELECT
    r.RegionName,
    sr.SubRegionName,
    l.LocationName,
    mt.TypeName,
    AVG(wm.Value) as AvgValue
FROM
    WeatherMeasurements wm
JOIN
    LocationDim l ON wm.LocationID = l.LocationID
JOIN
    SubRegionDim sr ON l.SubRegionID = sr.SubRegionID
JOIN
    RegionDim r ON sr.RegionID = r.RegionID
```

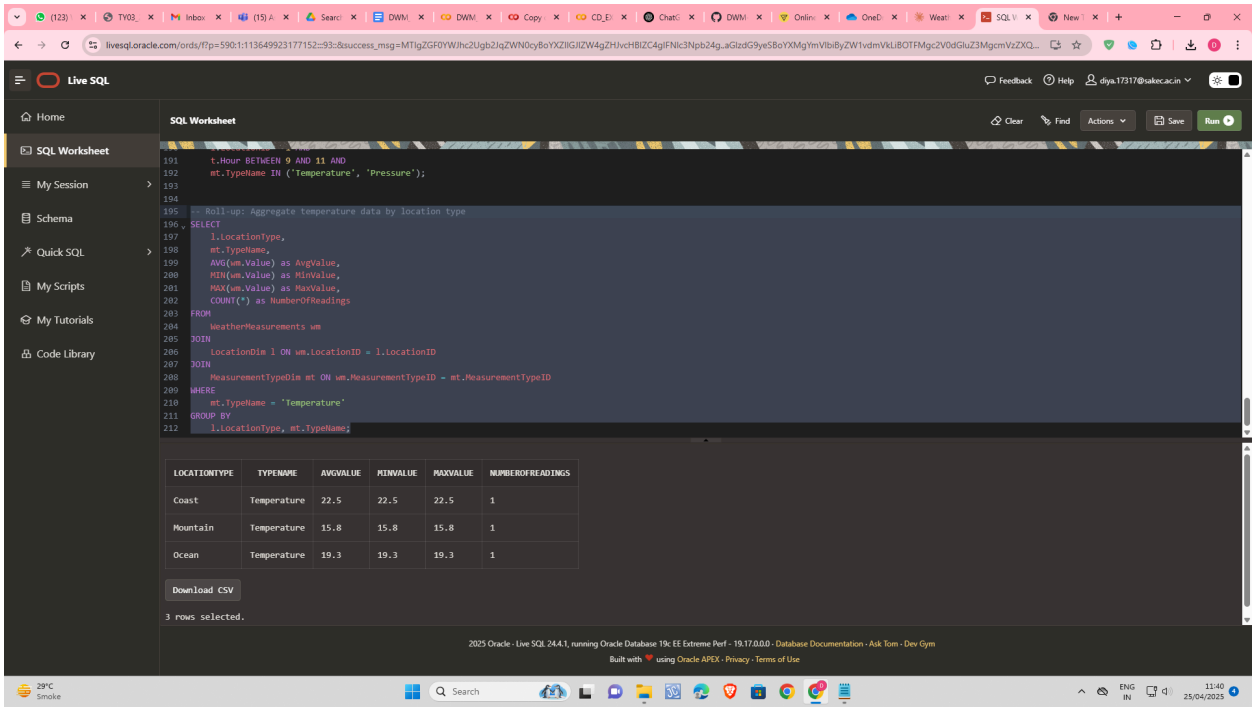
```
JOIN
    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
    mt.TypeName = 'Temperature'
GROUP BY
    r.RegionName, sr.SubRegionName, l.LocationName, mt.TypeName
ORDER BY
    r.RegionName, sr.SubRegionName, l.LocationName;


-- Pivot: Transform rows of measurement types into columns
SELECT
    t.Hour,
    l.LocationName,
    MAX(CASE WHEN mt.TypeName = 'Temperature' THEN wm.Value END) as Temperature,
    MAX(CASE WHEN mt.TypeName = 'Pressure' THEN wm.Value END) as Pressure,
    MAX(CASE WHEN mt.TypeName = 'Rainfall' THEN wm.Value END) as Rainfall
FROM
    WeatherMeasurements wm
JOIN
    TimeDim t ON wm.TimeID = t.TimeID
JOIN
    LocationDim l ON wm.LocationID = l.LocationID
JOIN
    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
GROUP BY
    t.Hour, l.LocationName
```

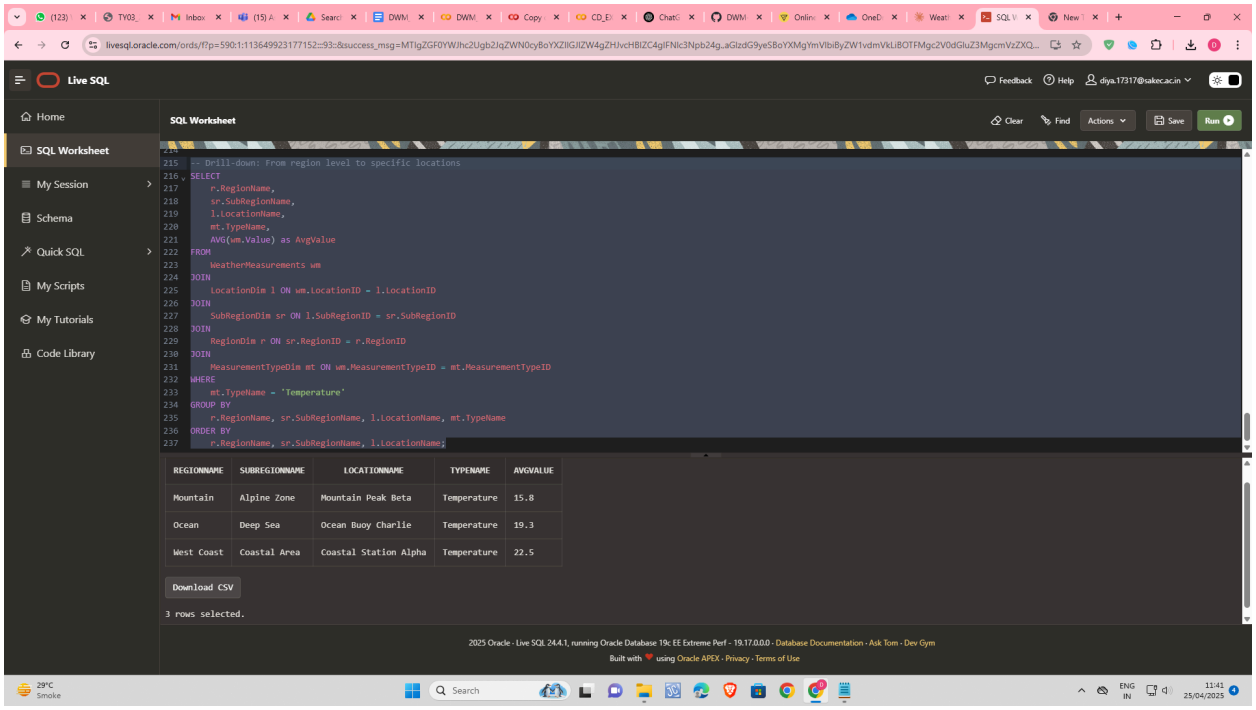
```
ORDER BY
    t.Hour, l.LocationName;
```

PROGRAM OUTPUT :




Live SQL
Feedback
Help
diya.17317@sakec.ac.in
Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
WHERE
l.LocationID = 1 AND mt.TypeName = 'Temperature';
-- Dice: Get all measurements for a specific location, time range, and measurement types
SELECT
t.Hour,
l.LocationName,
p.ProbeModel,
mt.TypeName,
mt.Unit,
wm.Value
FROM
WeatherMeasurements wm
JOIN
TimeDim t ON wm.TimeID = t.TimeID
JOIN
LocationDim l ON wm.LocationID = l.LocationID
JOIN
ProbeDim p ON wm.ProbeID = p.ProbeID
JOIN
MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
l.LocationID = 1 AND
t.Hour BETWEEN 9 AND 11 AND
mt.TypeName IN ('Temperature', 'Pressure');
HOUR
LOCATIONNAME
PROBEMODEL
TYPENAME
UNIT
VALUE
9
Coastal Station Alpha
WeatherMaster X1
Temperature
°C
22.5
9
Coastal Station Alpha
WeatherMaster X1
Pressure
hPa
1013.2
Download CSV
2 rows selected.
2025 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with using Oracle APEX - Privacy - Terms of Use


Live SQL
Feedback
Help
diya.17317@sakec.ac.in
Home
SQL Worksheet
My Session
Schema
Quick SQL
My Scripts
My Tutorials
Code Library
SQL Worksheet
Clear
Find
Actions
Save
Run
t.Hour BETWEEN 9 AND 11 AND
mt.TypeName IN ('Temperature', 'Pressure');
-- Roll-up: Aggregate temperature data by location type
SELECT
l.LocationType,
mt.TypeName,
AVG(wm.Value) as AvgValue,
MIN(wm.Value) as MinValue,
MAX(wm.Value) as MaxValue,
COUNT(*) as NumberOfReadings
FROM
WeatherMeasurements wm
JOIN
LocationDim l ON wm.LocationID = l.LocationID
JOIN
MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
mt.TypeName = 'Temperature'
GROUP BY
l.LocationType, mt.TypeName;
LOCATIONTYPE
TYPENAME
AVGVALUE
MINVALUE
MAXVALUE
NUMBEROFREADINGS
Coast
Temperature
22.5
22.5
22.5
1
Mountain
Temperature
15.8
15.8
15.8
1
Ocean
Temperature
19.3
19.3
19.3
1
Download CSV
3 rows selected.
2025 Oracle - Live SQL 24.4.1, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with using Oracle APEX - Privacy - Terms of Use

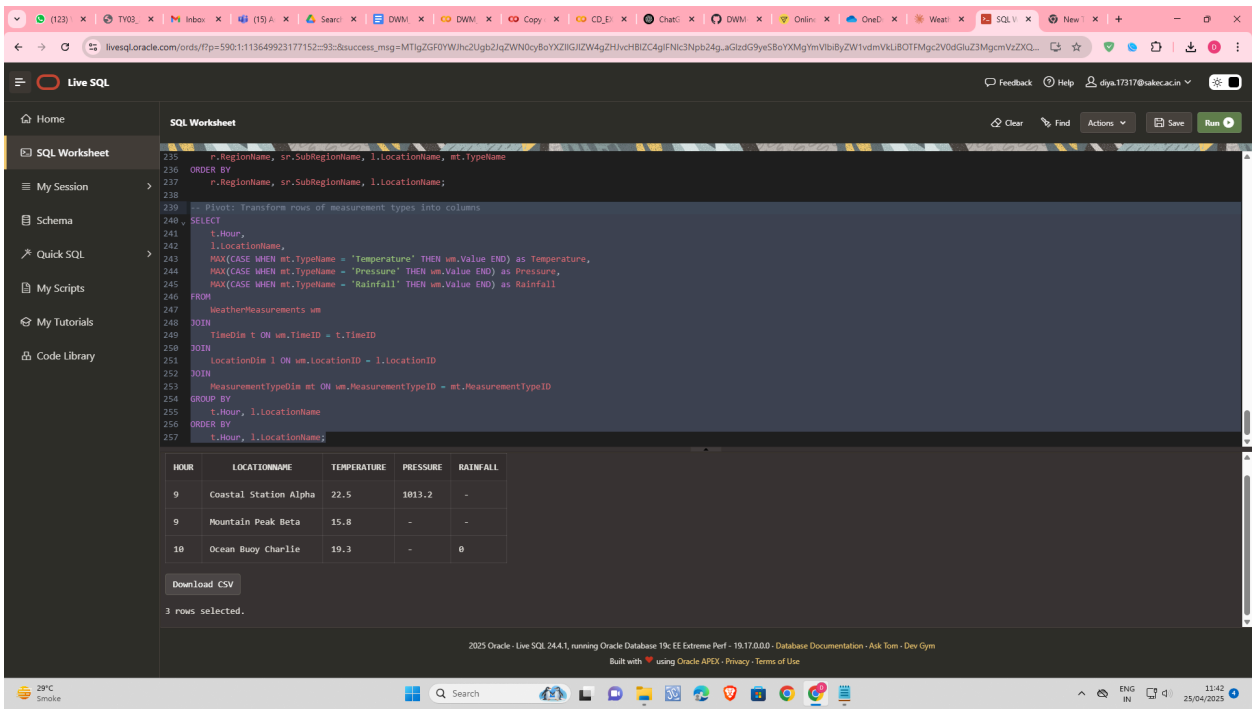


```
-- Drill-down: From region level to specific locations
SELECT
    r.RegionName,
    sr.SubRegionName,
    l.LocationName,
    mt.TypeName,
    AVG(wm.Value) as AvgValue
FROM
    WeatherMeasurements wm
JOIN
    LocationDim l ON wm.LocationID = l.LocationID
JOIN
    SubRegionDim sr ON l.SubRegionID = sr.SubRegionID
JOIN
    RegionDim r ON sr.RegionID = r.RegionID
JOIN
    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
WHERE
    mt.TypeName = 'Temperature'
GROUP BY
    r.RegionName, sr.SubRegionName, l.LocationName, mt.TypeName
ORDER BY
    r.RegionName, sr.SubRegionName, l.LocationName;
```

REGIONNAME	SUBREGIONNAME	LOCATIONNAME	TYPENAME	AVGVALUE
Mountain	Alpine Zone	Mountain Peak Beta	Temperature	15.8
Ocean	Deep Sea	Ocean Buoy Charlie	Temperature	19.3
West Coast	Coastal Area	Coastal Station Alpha	Temperature	22.5

Download CSV

3 rows selected.



```
    r.RegionName, sr.SubRegionName, l.LocationName, mt.TypeName
ORDER BY
    r.RegionName, sr.SubRegionName, l.LocationName;

-- Pivot: Transform rows of measurement types into columns
SELECT
    t.Hour,
    l.LocationName,
    MAX(CASE WHEN mt.TypeName = 'Temperature' THEN wm.Value END) as Temperature,
    MAX(CASE WHEN mt.TypeName = 'Pressure' THEN wm.Value END) as Pressure,
    MAX(CASE WHEN mt.TypeName = 'Rainfall' THEN wm.Value END) as Rainfall
FROM
    WeatherMeasurements wm
JOIN
    TimeDim t ON wm.TimeID = t.TimeID
JOIN
    LocationDim l ON wm.LocationID = l.LocationID
JOIN
    MeasurementTypeDim mt ON wm.MeasurementTypeID = mt.MeasurementTypeID
GROUP BY
    t.Hour, l.LocationName
ORDER BY
    t.Hour, l.LocationName;
```

HOUR	LOCATIONNAME	TEMPERATURE	PRESSURE	RAINFALL
9	Coastal Station Alpha	22.5	1013.2	-
9	Mountain Peak Beta	15.8	-	-
10	Ocean Buoy Charlie	19.3	-	0

Download CSV

3 rows selected.

Review Question :

1. What is the difference between Slice and Dice operations in OLAP?

Ans : Slice selects one specific value for one dimension, reducing dimensionality by one (like cutting a single slice from a cube). Dice selects multiple values across multiple dimensions, creating a smaller subcube while maintaining all dimensions.

2. How does Roll-up operation help in summarizing large volumes of data?

Ans : Roll-up summarizes large volumes of data by aggregating along concept hierarchies (e.g., daily→monthly→quarterly), reducing dimensionality, improving query performance, revealing high-level patterns, and supporting strategic decision-making by eliminating excessive detail.

3. Example of Pivoting providing clearer insight than traditional tabular view:

Ans : In retail sales analysis, pivoting transforms data from a linear list (month, region, product, sales) into a cross-tabulation with regions as rows and months as columns for each product category. This format instantly reveals regional performance patterns, seasonal trends, and product category comparisons that would be difficult to identify in traditional tabular format.

GITHUB LINK : https://github.com/asmi-04/DWM-ASMI-25.git