Aim:Implementation of OLAP operations: Slice, Dice, Rollup, Drill down and Pivot based on experiment 1 case study

Theory:

Online Analytical Processing (OLAP) is a category of data processing that allows users to analyze data from multiple dimensions. It supports complex analytical queries and is fundamental to decision-making processes in businesses. The main OLAP operations—Slice, Dice, Rollup, Drill-down, and Pivot—allow users to view and manipulate data in different ways to gain insights.

1. **Slice**: This operation allows users to extract a subset of data for a specific value of one dimension. For instance, if you want to analyze the health status of plants of a particular type, the Slice operation can be used to narrow down the data to just that type.
2. **Dice**: Dice is an extension of Slice and allows users to extract a subset of data based on multiple criteria across different dimensions. This operation is useful when you need to filter the data based on more than one dimension simultaneously.
3. **Rollup**: Rollup aggregates data along a dimension, moving up the hierarchy. For example, you might aggregate data to see the overall health of all plants in a specific family rather than each individual plant.
4. **Drill-down**: Drill-down is the opposite of Rollup. It allows users to move down the hierarchy and see more detailed data. For example, you can drill down from the plant family level to see data for each specific plant within that family.
5. **Pivot**: Pivot (also known as rotation) reorients the data axes, allowing users to view the data from different perspectives. It's particularly useful for comparing metrics across various dimensions.

Input:

1,Slice:Selects health status information for the plant named 'Lily'.

```
SELECT p.plant_name, h.description, h.severity
FROM Plant p
JOIN Health h ON p.plant_id = h.plant_id
WHERE p.plant_name = 'Lily';
```

2.Dice:Retrieves health status and soil type for the plants 'Potato' and 'Carrot' that grow in 'Loamy' soil.

```
SELECT p.plant_name, h.description, h.severity, s.soil_name
FROM Plant p
JOIN Health h ON p.plant_id = h.plant_id
```

```sql
JOIN Soil s ON p.soil_id = s.soil_id
WHERE p.plant_name IN ('Potato', 'Carrot')
 AND s.soil_name = 'Loamy';
```

3.Roll up:Summarize the average temperature and humidity by `Season`.

```sql
SELECT T.Season, AVG(TI.temp) AS Avg_Temperature, AVG(TI.humidity) AS Avg_Humidity
FROM TaskImpact TI
JOIN Time T ON TI.Time_id = T.Time_id
GROUP BY T.Season;
```

4.Drill down:Drill down from `Season` to `Month` to get detailed average temperature and humidity.

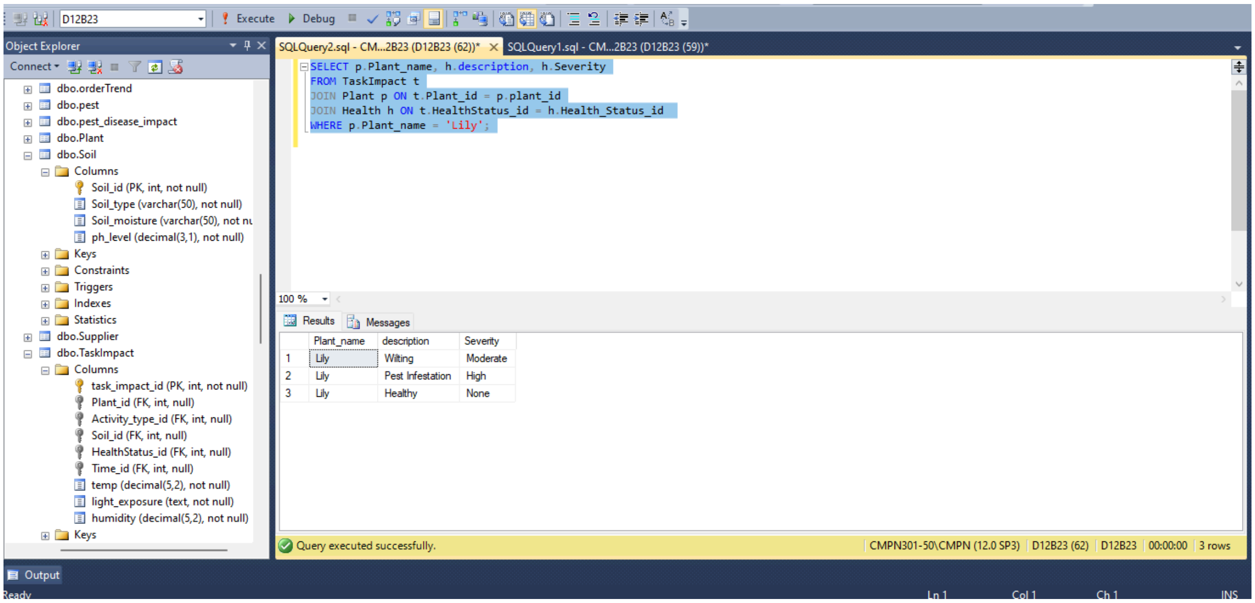
```sql
SELECT T.Season, T.Month, AVG(TI.temp) AS Avg_Temperature, AVG(TI.humidity) AS Avg_Humidity
FROM TaskImpact TI
JOIN Time T ON TI.Time_id = T.Time_id
GROUP BY T.Season, T.Month;
```

5.PivotPivot the data to show average temperature for each `Plant Type` by `Season`.

```sql
SELECT P.Plant_Type,
AVG(CASE WHEN T.Season = 'Winter' THEN TI.temp END) AS Winter_Temperature,
AVG(CASE WHEN T.Season = 'Summer' THEN TI.temp END) AS Summer_Temperature
FROM TaskImpact TI
JOIN Plant P ON TI.Plant_id = P.Plant_id
JOIN Time T ON TI.Time_id = T.Time_id
GROUP BY P.Plant_Type;
```

Output:

1)Slice

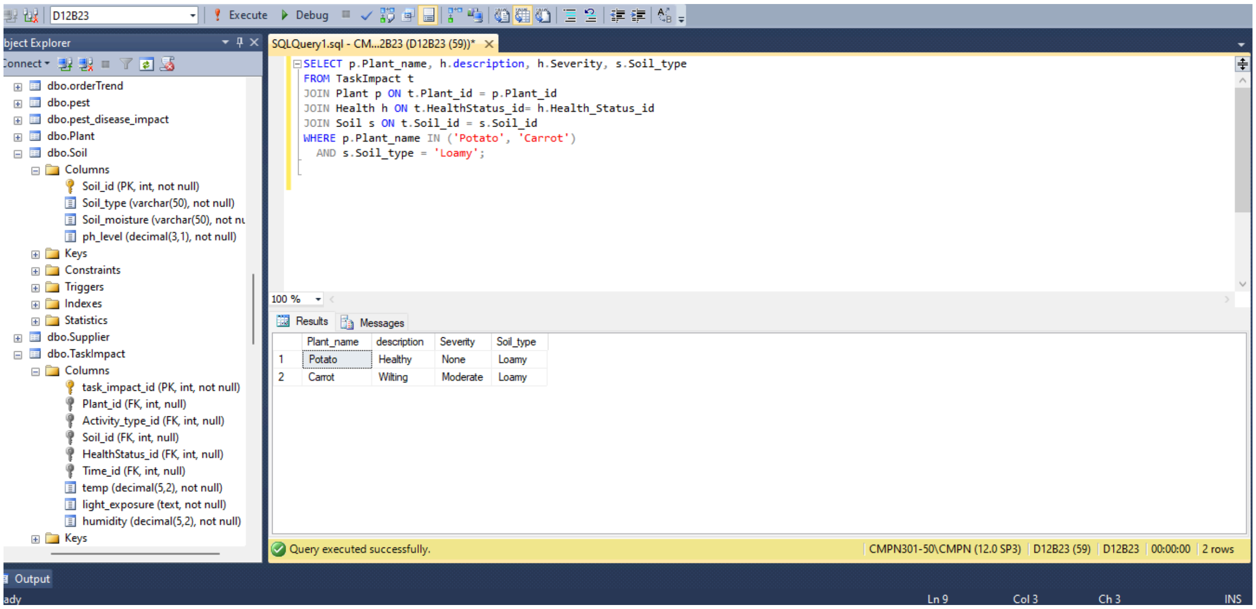

D12B23
Execute
Debug
Object Explorer
Connect
dbo.orderTrend
dbo.pest
dbo.pest_disease_impact
dbo.Plant
dbo.Soil
Columns
Soil_id (PK, int, not null)
Soil_type (varchar(50), not null)
Soil_moisture (varchar(50), not nu
ph_level (decimal(3,1), not null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Supplier
dbo.TaskImpact
Columns
task_impact_id (PK, int, not null)
Plant_id (FK, int, null)
Activity_type_id (FK, int, null)
Soil_id (FK, int, null)
HealthStatus_id (FK, int, null)
Time_id (FK, int, null)
temp (decimal(5,2), not null)
light_exposure (text, not null)
humidity (decimal(5,2), not null)
Keys
SQLQuery2.sql - CM...2B23 (D12B23 (62))*
SQLQuery1.sql - CM...2B23 (D12B23 (59))*
100 %
Results
Messages
Output
Ready
Ln 1
Col 1
Ch 1
INS

```sql
SELECT p.Plant_name, h.description, h.Severity
FROM TaskImpact t
JOIN Plant p ON t.Plant_id = p.plant_id
JOIN Health h ON t.HealthStatus_id = h.Health_Status_id
WHERE p.Plant_name = 'Lily';
```

	Plant_name	description	Severity
1	Lily	Wilting	Moderate
2	Lily	Pest Infestation	High
3	Lily	Healthy	None

```
Query executed successfully.    CMPN301-50\CMPN (12.0 SP3) | D12B23 (62) | D12B23 | 00:00:00 | 3 rows
```

2)Dice

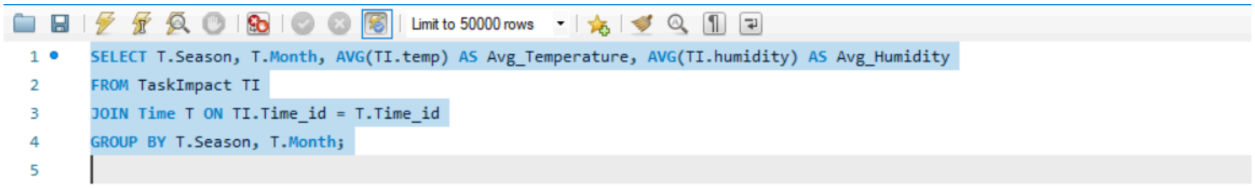

D12B23
Execute
Debug
bject Explorer
onnect
dbo.orderTrend
dbo.pest
dbo.pest_disease_impact
dbo.Plant
dbo.Soil
Columns
Soil_id (PK, int, not null)
Soil_type (varchar(50), not null)
Soil_moisture (varchar(50), not nu
ph_level (decimal(3,1), not null)
Keys
Constraints
Triggers
Indexes
Statistics
dbo.Supplier
dbo.TaskImpact
Columns
task_impact_id (PK, int, not null)
Plant_id (FK, int, null)
Activity_type_id (FK, int, null)
Soil_id (FK, int, null)
HealthStatus_id (FK, int, null)
Time_id (FK, int, null)
temp (decimal(5,2), not null)
light_exposure (text, not null)
humidity (decimal(5,2), not null)
Keys
SQLQuery1.sql - CM...2B23 (D12B23 (59))*
100 %
Results
Messages
Output
ady
Ln 9
Col 3
Ch 3
INS

```sql
SELECT p.Plant_name, h.description, h.Severity, s.Soil_type
FROM TaskImpact t
JOIN Plant p ON t.Plant_id = p.Plant_id
JOIN Health h ON t.HealthStatus_id= h.Health_Status_id
JOIN Soil s ON t.Soil_id = s.Soil_id
WHERE p.Plant_name IN ('Potato', 'Carrot')
  AND s.Soil_type = 'Loamy';
```

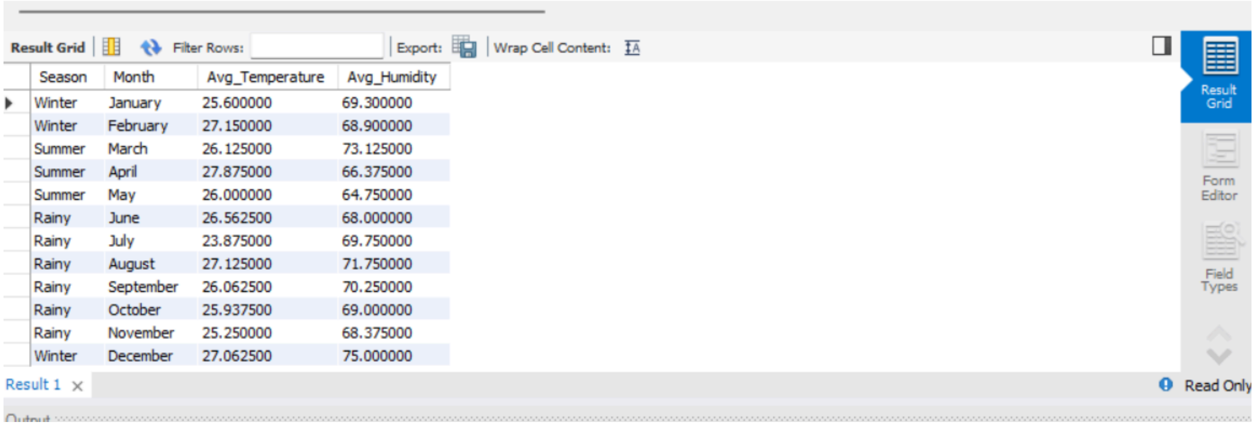
	Plant_name	description	Severity	Soil_type
1	Potato	Healthy	None	Loamy
2	Carrot	Wilting	Moderate	Loamy

```
Query executed successfully.    CMPN301-50\CMPN (12.0 SP3) | D12B23 (59) | D12B23 | 00:00:00 | 2 rows
```

3)Drill Down

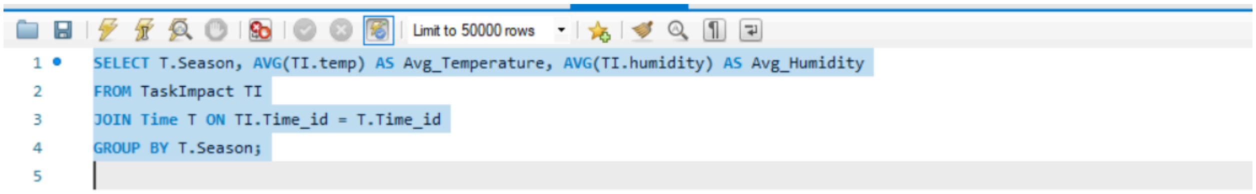


```sql
SELECT T.Season, T.Month, AVG(TI.temp) AS Avg_Temperature, AVG(TI.humidity) AS Avg_Humidity
FROM TaskImpact TI
JOIN Time T ON TI.Time_id = T.Time_id
GROUP BY T.Season, T.Month;
```

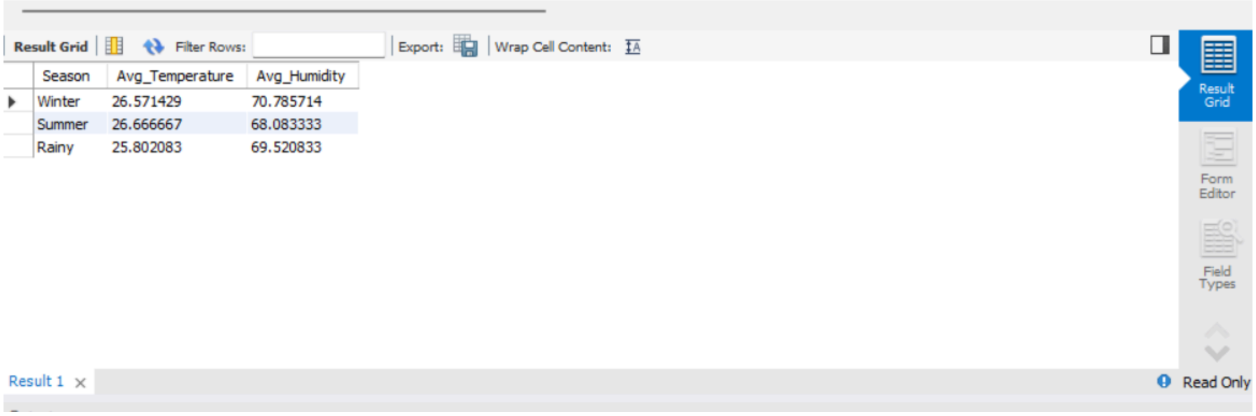


Season	Month	Avg_Temperature	Avg_Humidity
Winter	January	25.600000	69.300000
Winter	February	27.150000	68.900000
Summer	March	26.125000	73.125000
Summer	April	27.875000	66.375000
Summer	May	26.000000	64.750000
Rainy	June	26.562500	68.000000
Rainy	July	23.875000	69.750000
Rainy	August	27.125000	71.750000
Rainy	September	26.062500	70.250000
Rainy	October	25.937500	69.000000
Rainy	November	25.250000	68.375000
Winter	December	27.062500	75.000000

4)Roll Up

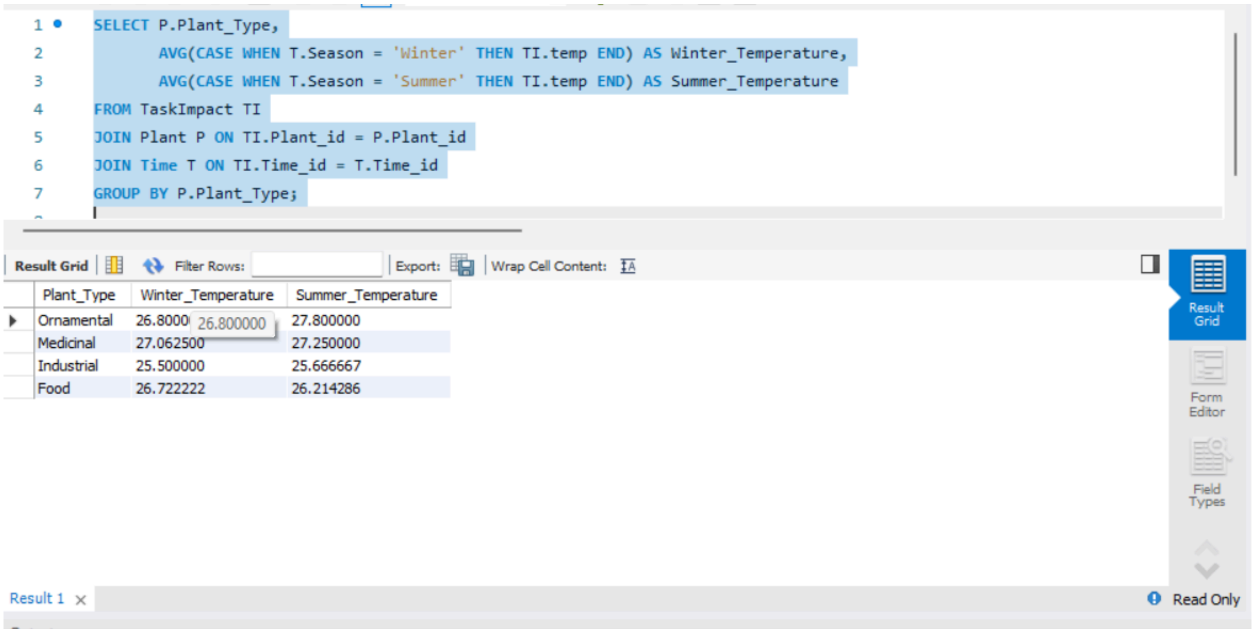


```sql
SELECT T.Season, AVG(TI.temp) AS Avg_Temperature, AVG(TI.humidity) AS Avg_Humidity
FROM TaskImpact TI
JOIN Time T ON TI.Time_id = T.Time_id
GROUP BY T.Season;
```



Season	Avg_Temperature	Avg_Humidity
Winter	26.571429	70.785714
Summer	26.666667	68.083333
Rainy	25.802083	69.520833

5)Pivot



```sql
SELECT P.Plant_Type,
       AVG(CASE WHEN T.Season = 'Winter' THEN TI.temp END) AS Winter_Temperature,
       AVG(CASE WHEN T.Season = 'Summer' THEN TI.temp END) AS Summer_Temperature
FROM TaskImpact TI
JOIN Plant P ON TI.Plant_id = P.Plant_id
JOIN Time T ON TI.Time_id = T.Time_id
GROUP BY P.Plant_Type;
```

Plant_Type	Winter_Temperature	Summer_Temperature
Ornamental	26.800000	27.800000
Medicinal	27.062500	27.250000
Industrial	25.500000	25.666667
Food	26.722222	26.214286

Conclusion:

The implementation of OLAP operations on the given dataset provided valuable insights into the relationships between different dimensions such as plant type, health status, soil type, and activities. By performing Slice and Dice operations, we were able to filter the data and focus on specific aspects of the dataset. Rollup and Drill-down operations allowed us to analyze data at varying levels of granularity, revealing patterns and trends at both high and detailed levels. The Pivot operation enabled us to rearrange the data, offering different perspectives and helping us to identify correlations across dimensions. Overall, the experiment demonstrated the power of OLAP operations in enhancing data analysis and decision-making processes.