**Experiment No 3**

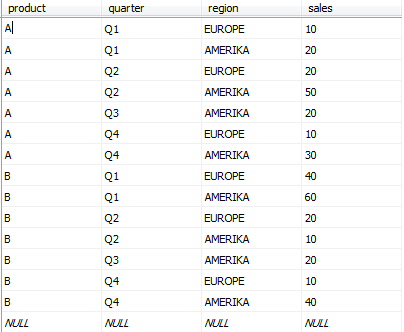
**Aim:** Perform OLAP operations such as Roll Up, Drill Down, Slice, Dice through SQL- Server.

**Theory:**

**Consider following sales table**

Basically, this query computes the union of 2² = 4 groupings of the SALESTABLE being: {(quarter,region), (quarter), (region), ()},

In other words, since quarter has 4 values and region 2 values, the resulting multiset will have 4\*2+4\*1+1\*2+1 or 15 tuples as you can see illustrated in Table below.

****

* **Creation of Cube:**

SELECT quarter,region,SUM(sales)from REV group by cube(quarter,region);

* **Roll-Up:** A Query could involve a ROLLUP of year>month>day or country>state>city. When a roll-up is performed, one or more dimensions from the data cube are removed because the output would display blank for certain rows.

**Syntax:**

**SELECT …GROUP BY ROLLUP ( GROUPING\_COLUMN\_REFERENCE\_LIST);**

SELECT quarter,region,SUM(sales)from REV group by rollup (quarter,region);

* **Roll-Down:** This is a reverse of the ROLL UP operation discussed above. The data is aggregated from a higher level summary to a lower level summary/detailed data.

**Syntax:**

##### SELECT … GROUP BY ROLLDOWN(COLUMNS);

##### Slice: A slice in a multidimensional array is a column of data corresponding to a single value for one or more members of the dimension.

**Syntax: SELECTION CONDITIONS ON SOME ATTRIBUTES USING <WHERE CLAUSE> <GROUP BY> AND   AGGREGATION ON SOME ATTRIBUTE**

SELECT product,SUM(sales) FROM REV WHERE PRODUCT= 'A' GROUP BY product ;

* **Dice:** Dicing is similar to slicing, but it works a little bit differently. When one thinks of slicing, filtering is done to focus on a particular attribute.

##### Syntax: SELECTION CONDITIONS ON SOME ATTRIBUTES USING <WHERE CLAUSE> GROUP BY AND AGGREGATION ON SOME ATTRIBUTE

SELECT product, SUM(sales) FROM REV WHERE PRODUCT='B' AND region='EUROPE' GROUP BY product;

* **Pivot:** The SQL Server (Transact-SQL) PIVOT clause allows you to write a cross-tabulation. This means that you can aggregate your results and rotate rows into columns.

**Syntax:**

SELECT first\_column AS <first\_column\_alias>,

[pivot\_value1], [pivot\_value2], ... [pivot\_value\_n]

FROM

(<source\_table>) AS <source\_table\_alias>

PIVOT

(

aggregate\_function(<aggregate\_column>)

FOR <pivot\_column>

IN ([pivot\_value1], [pivot\_value2], ... [pivot\_value\_n])

) AS <pivot\_table\_alias>;

### Parameters or Arguments

**first\_column**

A column or expression that will display as the first column in the pivot table.

**first\_column\_alias**

The column heading for the first column in the pivot table.

**pivot\_value1, pivot\_value2, ... pivot\_value\_n**

A list of values to pivot.

**source\_table**

A [SELECT statement](https://www.techonthenet.com/sql_server/select.php) that provides the source data for the pivot table.

**source\_table\_alias**

An alias for source\_table.

**aggregate\_function**

An aggregate function such as[SUM](https://www.techonthenet.com/sql_server/functions/sum.php),[COUNT](https://www.techonthenet.com/sql_server/functions/count.php),[MIN](https://www.techonthenet.com/sql_server/functions/min.php),[MAX](https://www.techonthenet.com/sql_server/functions/max.php), or [AVG](https://www.techonthenet.com/sql_server/functions/avg.php).

**aggregate\_column**

The column or expression that will be used with the aggregate\_function.

**pivot\_column**

The column that contains the pivot values.

**pivot\_table\_alias**

An alias for the pivot table.

**Example:**

SELECT 'sale' AS TotalsaleByProduct,

[A], [B]

FROM

(SELECT product, sale

FROM REV) AS SourceTable

PIVOT

(

SUM(sale)

FOR product IN ([A], [B])

) AS PivotTable;

**Conclusion: Performed and Executed OLAP Operation Sucessfully**