

## Experiment No. 03

Aim:

Perform OLAP operations such as roll up : *roll up statement*

Output:

*Output of the query (to see the result)*

SQLQuery1.sql - L...HTNING\dhond (54)*				
select * from REV;				
100 %				
<b>Results</b>				<b>Messages</b>
	prod...	quar...	region	sales
1	A	Q1	EUROPE	10
2	A	Q1	AMERIKA	20
3	A	Q2	EUROPE	20
4	A	Q2	AMERIKA	50
5	A	Q3	AMERIKA	20
6	A	Q4	EUROPE	10
7	A	Q4	AMERIKA	30
8	B	Q1	EUROPE	40
9	B	Q1	AMERIKA	60
10	B	Q2	EUROPE	20
11	B	Q2	AMERIKA	10
12	B	Q3	AMERICA	20
13	B	Q4	EUROPE	10
14	B	Q4	AMERIKA	40

REV table contents

100 %		
<b>Results</b>		
	quar...	region
1	Q3	AMERICA
2	NULL	AMERICA
3	Q1	AMERIKA
4	Q2	AMERIKA
5	Q3	AMERIKA
6	Q4	AMERIKA
7	NULL	AMERIKA
8	Q1	EUROPE
9	Q2	EUROPE
10	Q4	EUROPE
11	NULL	EUROPE
12	NULL	NULL
13	Q1	NULL
14	Q2	NULL
15	Q3	NULL
16	Q4	NULL

SQLQuery1.sql - L...HTNING\dhond (54)\*

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

Creation of cube

### Experiment No. 03

Aim:

Perform OLAP operations such as Rollup, drill down, slice and dice and pivot, through SQL server.

Theory:

In computing, Online Analytical Processing or OLAP is an approach to answering multidimensional analytical queries (MDA). OLAP is part of the broader category of business intelligence, which also encompasses relational database report writing and data mining.

Here is the list of OLAP operations:

- Roll up
- Drill Down
- Slice and Dice
- Pivot (rotate)

Procedure:

1. Create a table in SQL Server.

2. Perform OLTP operations on table data

3. Observe the result.

Consider following sales table

Basically the query computes the union of  $2^2 = 4$  groupings of the SALES TABLE 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 in the table 3.1

**CREATION OF CUBE:**

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

	quar...	region	(No column na...)
1	Q1	AMERIKA	80
2	Q1	EUROPE	50
3	Q1	NULL	130
4	Q2	AMERIKA	60
5	Q2	EUROPE	40
6	Q2	NULL	100
7	Q3	AMERICA	20
8	Q3	AMERIKA	20
9	Q3	NULL	40
10	Q4	AMERIKA	70
11	Q4	EUROPE	20
12	Q4	NULL	90
13	NUL	NUL	360

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

### ROLLUP operation

	quar...	region	sales
1	Q1	AMERIKA	20
2	Q1	AMERIKA	60
3	Q1	EUROPE	10
4	Q1	EUROPE	40

`SELECT quarter, region, sales from REV WHERE quarter='Q1' order by region;`

### DRILL DOWN OPERATION

	prod...	(No column na...)
1	A	160

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

### SLICE OPERATION

- Roll up:

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

Syntax:

```
SELECT ... ROLLUP (columns) GROUP BY ROLLUP (grouping-columns)
```

Example:

```
SELECT quarter, region, sum(sales) as Sales from REV group by rollup(quarter, region);
```

- Roll down / drill down:

This is reverse of the roll up operation discussed above.

The data is aggregated from a higher level summary to a lower level summary or detailed data.

Syntax:

```
SELECT ... GROUP BY ROLLDOWN (columns)
```

Example:

```
SELECT quarter, region, sales from REV where quarter = 'Q1' order by region;
```

- Slice:

A slice in a multi dimensional 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

Example:

```
SELECT product, sum(sales) from REV where PRODUCT = 'A'  
GROUP BY Product;
```

DICE OPERATION

```
prod... (No column name specified)
1   B    70
SELECT product, SUM(sales) FROM REV WHERE PRODUCT='B' AND region='EUROPE' GROUP BY product;
```

PIVOT OPERATION

```
SQLQuery1.sql - L:\HTNING\dhond (54)*
SELECT 'sales' AS TotalsaleByProduct,
[A], [B]
FROM
(SELECT product, sales
FROM REV) AS SourceTable
PIVOT
(
SUM(sales)
FOR product IN ([A], [B])
) AS PivotTable;
```

TotalsaleByProd...	A	B
1 sales	160	200

- DICE:

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

Syntax:

SELECTION OF CONDITIONS ON SOME ATTRIBUTES using <WHERE clause>  
GROUP BY and aggregation on some attribute.

Example:

```
SELECT product, sum(sales) FROM REV where product = "B" AND
region is '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. sales>,
[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-valueN

A list of values to pivot.

source\_table:

A SELECT Statement that provides source data for pivot table.

source-table-alias:

An alias for source-table

aggregation function

An aggregation function such as SUM, COUNT, MIN, MAX or AVG.

aggregate-column:

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

pivot-column

The column that contains pivot values.

with in minutes delay with 20 jobs each time left packages in minutes

old job

old job with in minutes left with ref product name left

min - delay = sum - delay + value - delay  
sum of delay is ref

old sum of old sum which left timestamp take a

old - old - sum  
old sum ref. old na

min - delay = sum - delay  
sum of delay is ref sum which left timestamp take a

min - delay = sum - delay  
sum of delay is ref sum which left timestamp take a

### Conclusion:

Performed and Executed OLAP operations successfully.

pivot-table alias :

An alias for the pivot table.

Example :

```
Select 'sales' AS TotalSaleByProduct,  
[A], [B]  
FROM  
(SELECT product, sale FROM REV) AS sourceTable  
PIVOT (  
    SUM(sales)  
    FOR product IN ([A],[B])  
) AS PivotTable ;
```

Conclusion:

Performed and Executed OLAP operations successfully.