EXPERIMENT NO 2

NAME: Vishal Shashikant Salvi **UID: 2019230069**

DATE:11/08/2020 **BATCH:C**

Aim:

To perform OLAP Operations

Theory:

OLAP is a widely spread technology belonging to Business Intelligence processes developed to coordinate and analyse vast amounts of data. OLAP databases are stored in the form of multidimensional cubes where each cube comprises the data supposed relevant by a cube administrator. Through certain OLAP operations, a user is able to obtain a specified view of the cube and extract requisite information from it. So this way it’s possible to get a necessary Pivot Table and Pivot Chart report.

OLAP Operations in the Multidimensional Data Model

In the multidimensional model, the records are organized into various dimensions, and each dimension includes multiple levels of abstraction described by concept hierarchies. This organization support users with the flexibility to view data from various perspectives. A number of OLAP data cube operation exist to demonstrate these different views, allowing interactive queries and search of the record at hand. Hence, OLAP supports a user-friendly environment for interactive data analysis.

Consider the OLAP operations which are to be performed on multidimensional data. The figure shows data cubes for sales of a shop. The cube contains the dimensions, location, and time and item, where the **location** is aggregated with regard to city values, **time** is aggregated with respect to quarters, and an **item** is aggregated with respect to item types.

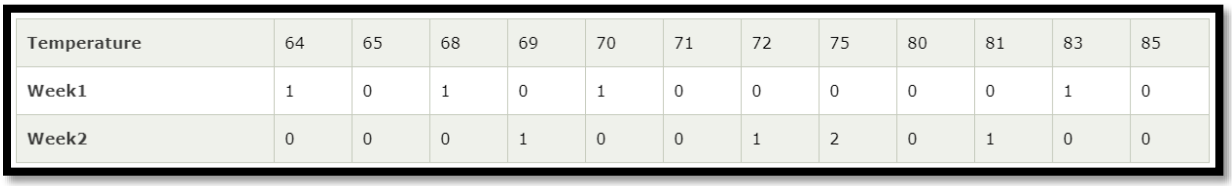
Roll-Up

The roll-up operation (also known as drill-up or aggregation operation) performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction. Roll-up is like zooming-out on the data cubes. Figure shows the result of roll-up operations performed on the dimension location. The hierarchy for the location is defined as the Order Street, city, province, or state, country. The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country.

When a roll-up is performed by dimensions reduction, one or more dimensions are removed from the cube. For example, consider a sales data cube having two dimensions, location and time. Roll-up may be performed by removing, the time dimensions, appearing in an aggregation of the total sales by location, relatively than by location and by time.

Example

Consider the following cubes illustrating temperature of certain days recorded weekly:

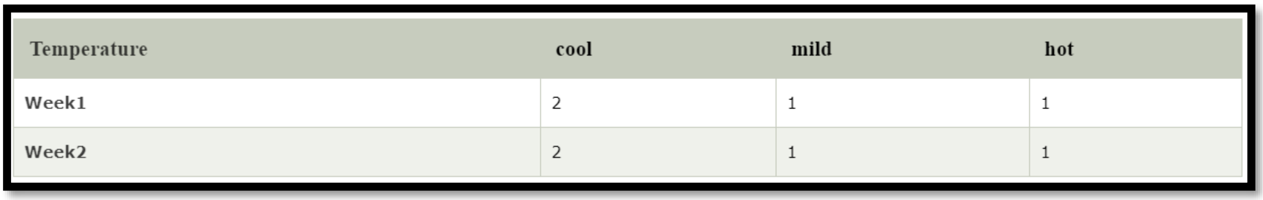


Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes.

To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a roll-up.

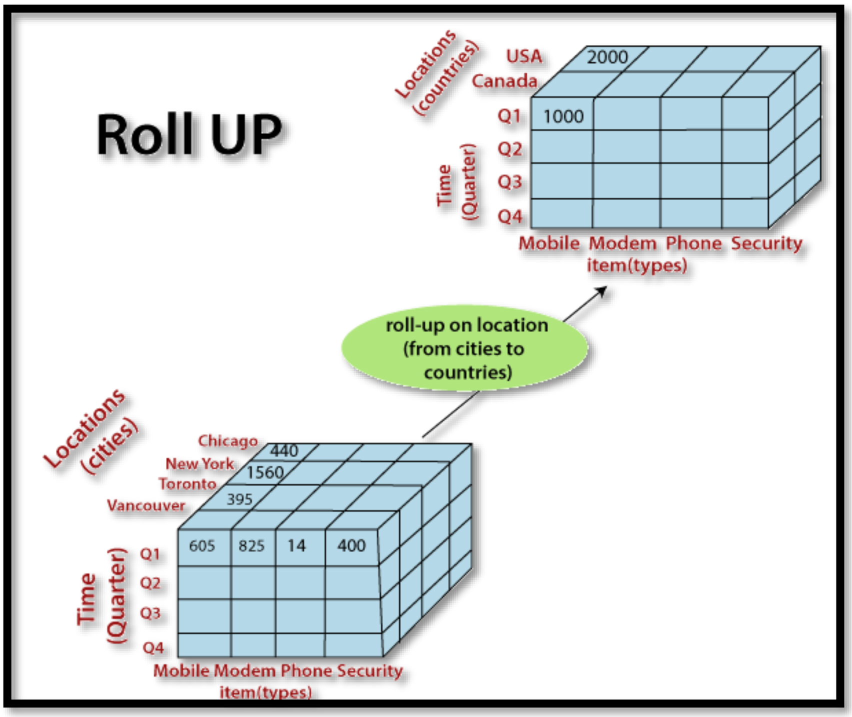
By doing this, we contain the following cube:



Temperature	cool	mild	hot
Week1	2	1	1
Week2	2	1	1

The roll-up operation groups the information by levels of temperature.

The following diagram illustrates how roll-up works.



Drill-Down

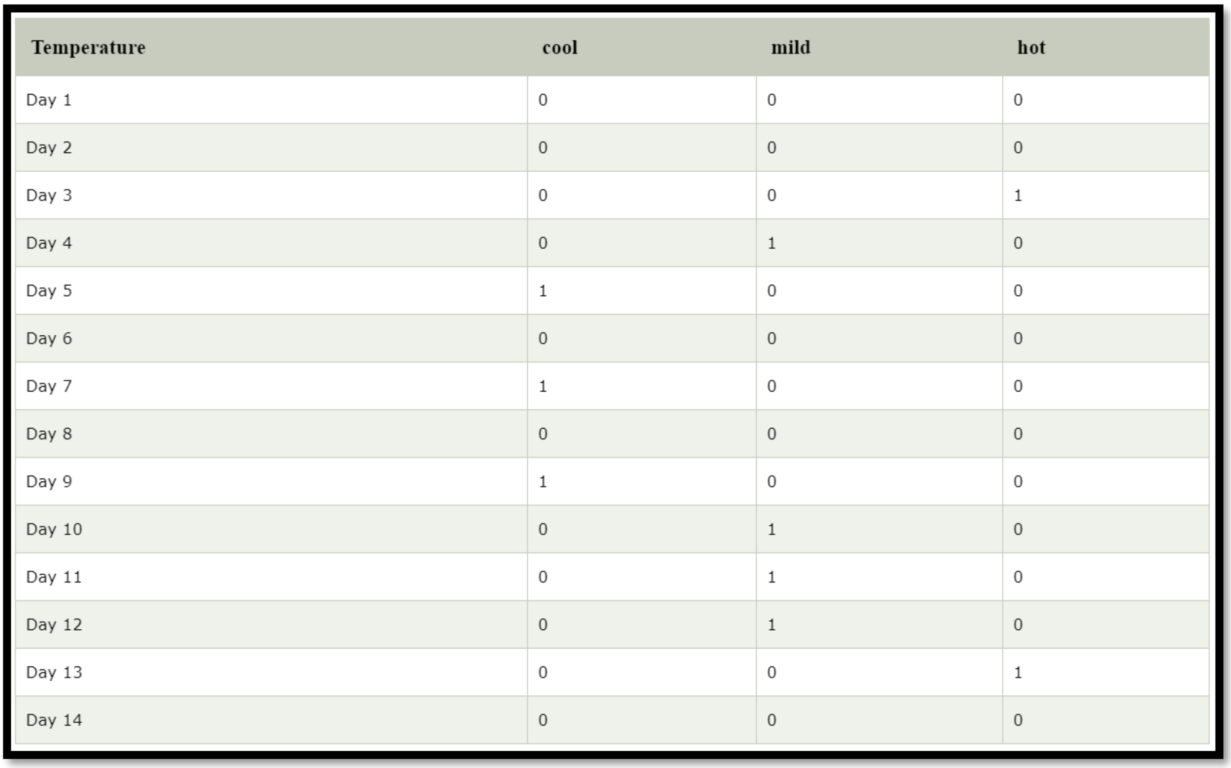
The drill-down operation **(also called roll-down)** is the reverse operation of **roll-up**. Drill-down is like **zooming-in** on the data cube. It navigates from less detailed record to more detailed data. Drill-down can be performed by either **stepping down** a concept hierarchy for a dimension or adding additional dimensions.

Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.

Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

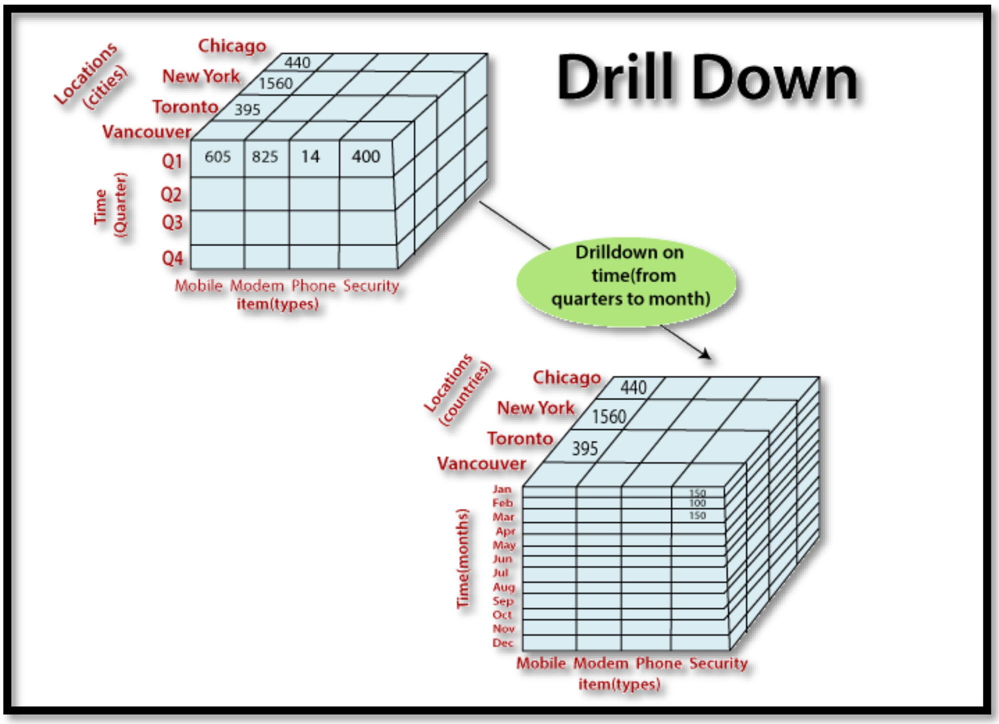
Example

Drill-down adds more details to the given data



Temperature	cool	mild	hot
Day 1	0	0	0
Day 2	0	0	0
Day 3	0	0	1
Day 4	0	1	0
Day 5	1	0	0
Day 6	0	0	0
Day 7	1	0	0
Day 8	0	0	0
Day 9	1	0	0
Day 10	0	1	0
Day 11	0	1	0
Day 12	0	1	0
Day 13	0	0	1
Day 14	0	0	0

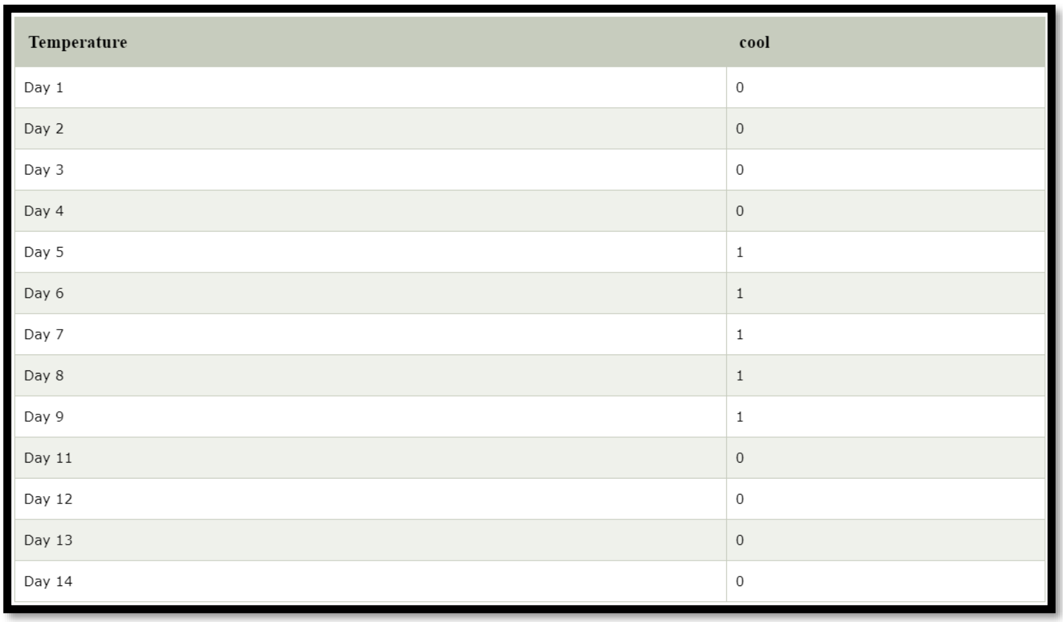
The following diagram illustrates how Drill-down works.



Slice

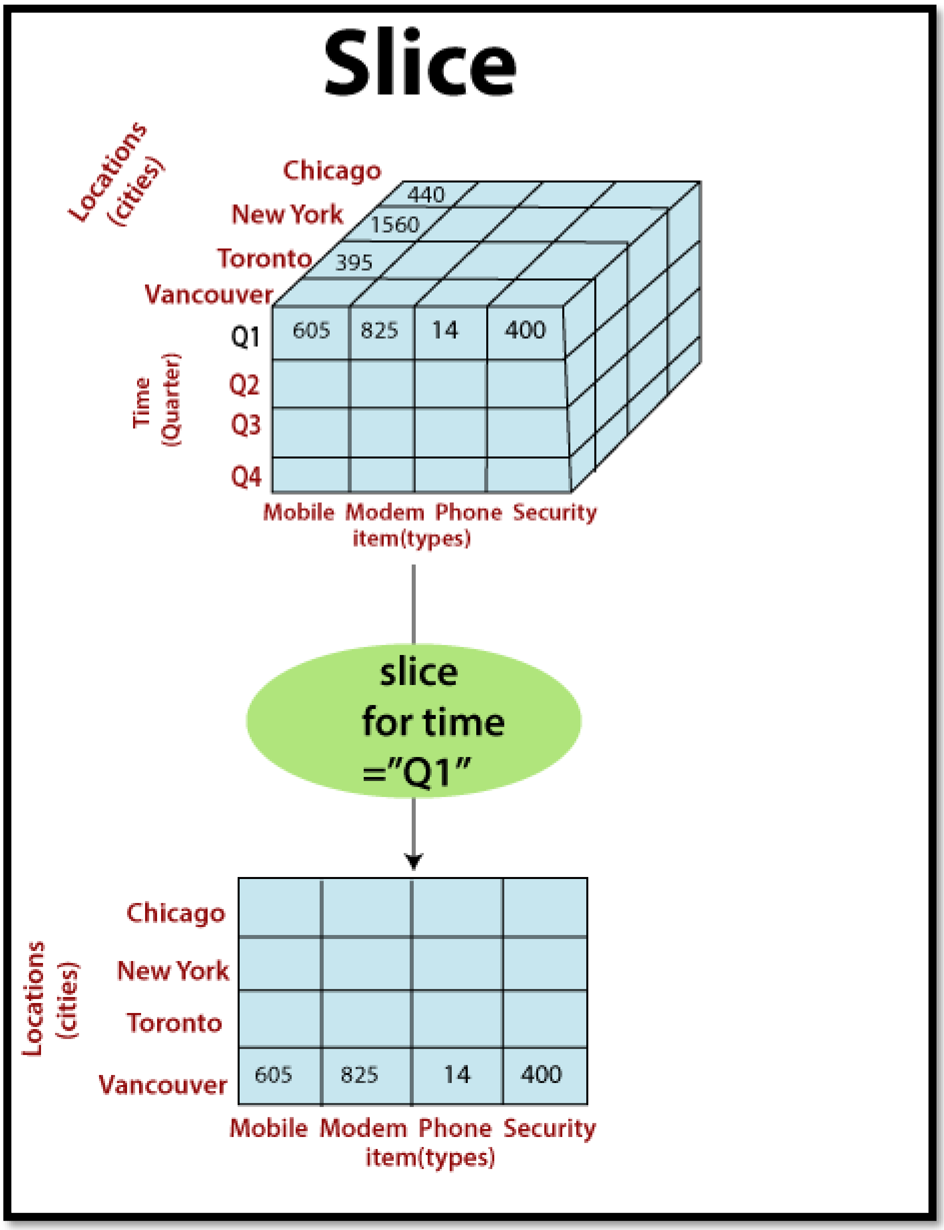
A **slice** is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the Slice operations perform a selection on one dimension of the given cube, thus resulting in a subcube.

For example, if we make the selection, temperature=cool we will obtain the following cube:



Temperature	cool
Day 1	0
Day 2	0
Day 3	0
Day 4	0
Day 5	1
Day 6	1
Day 7	1
Day 8	1
Day 9	1
Day 11	0
Day 12	0
Day 13	0
Day 14	0

The following diagram illustrates how Slice works.



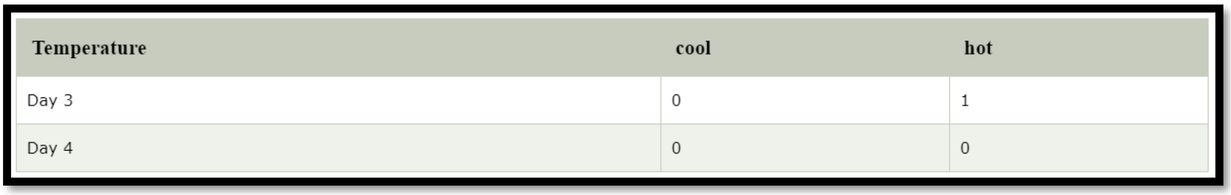
Here Slice is functioning for the dimensions "time" using the criterion time = "Q1".

It will form a new sub-cubes by selecting one or more dimensions.

Dice

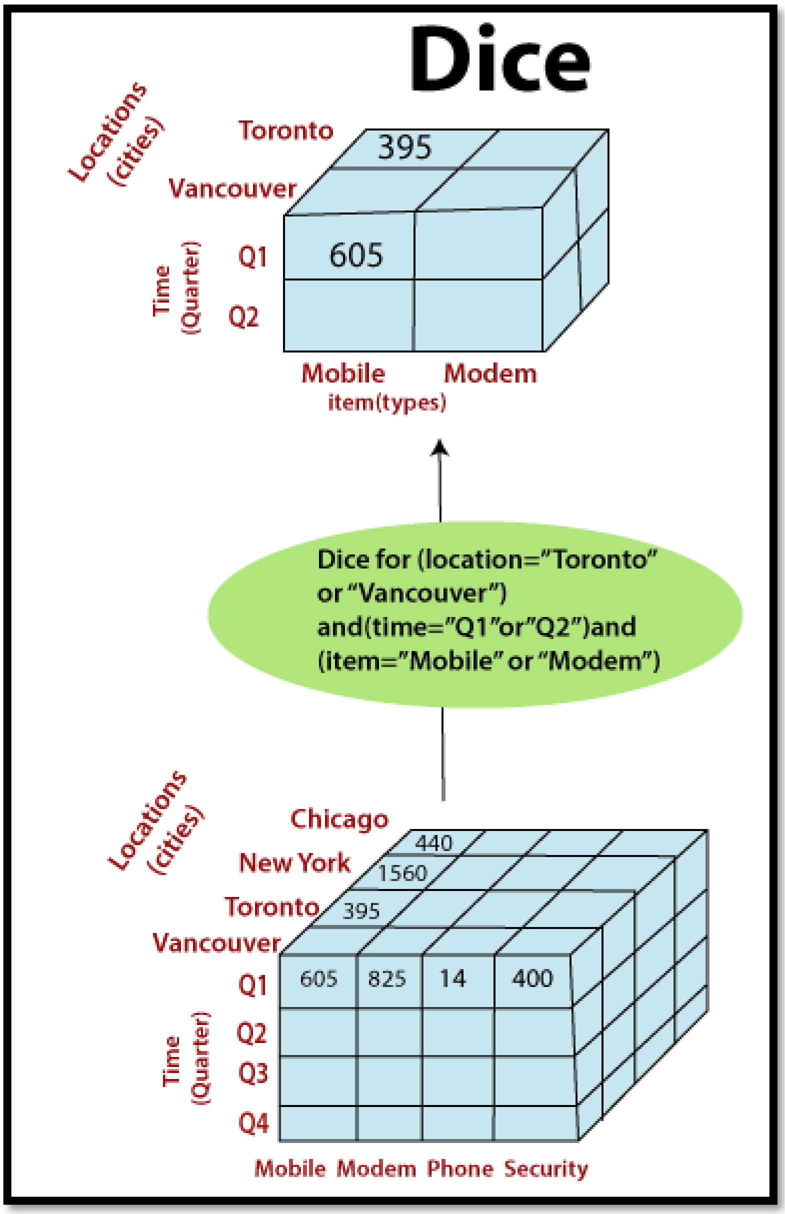
The dice operation describes a subcube by operating a selection on two or more dimension.

For example, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following subcube (still two-dimensional)



Temperature	cool	hot
Day 3	0	1
Day 4	0	0

Consider the following diagram, which shows the dice operations.

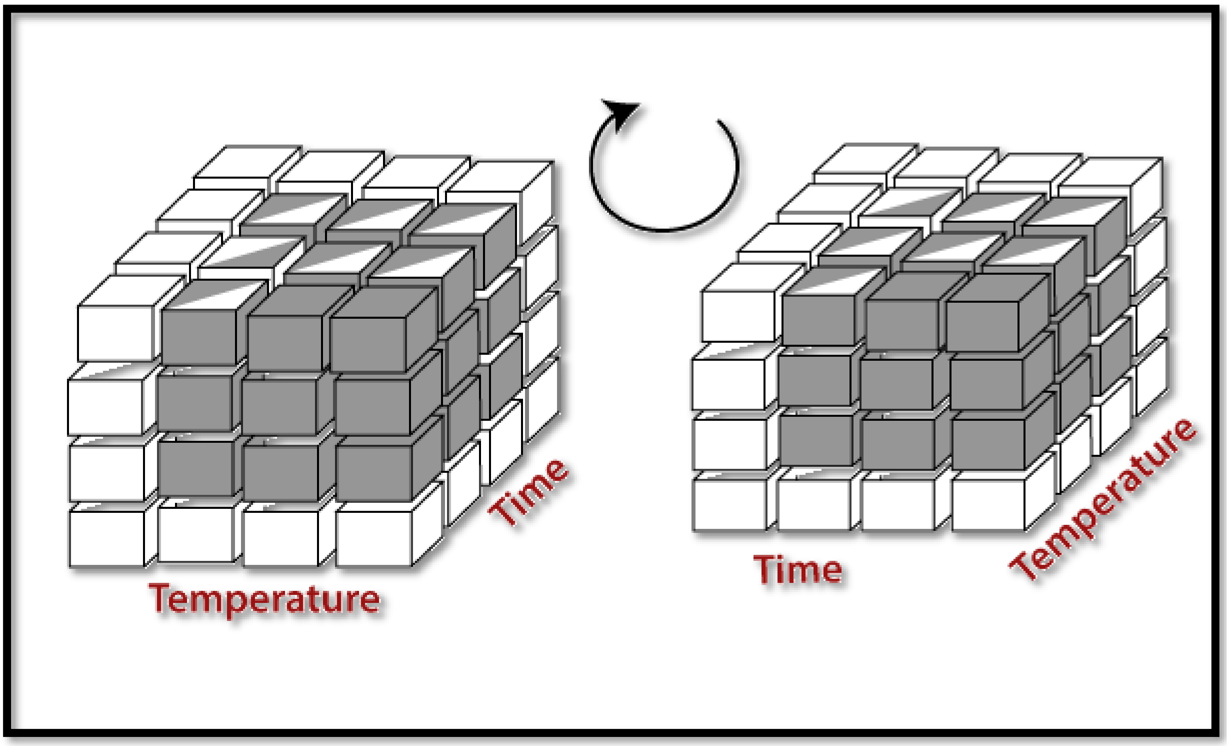


The dice operation on the cubes based on the following selection criteria involves three dimensions.

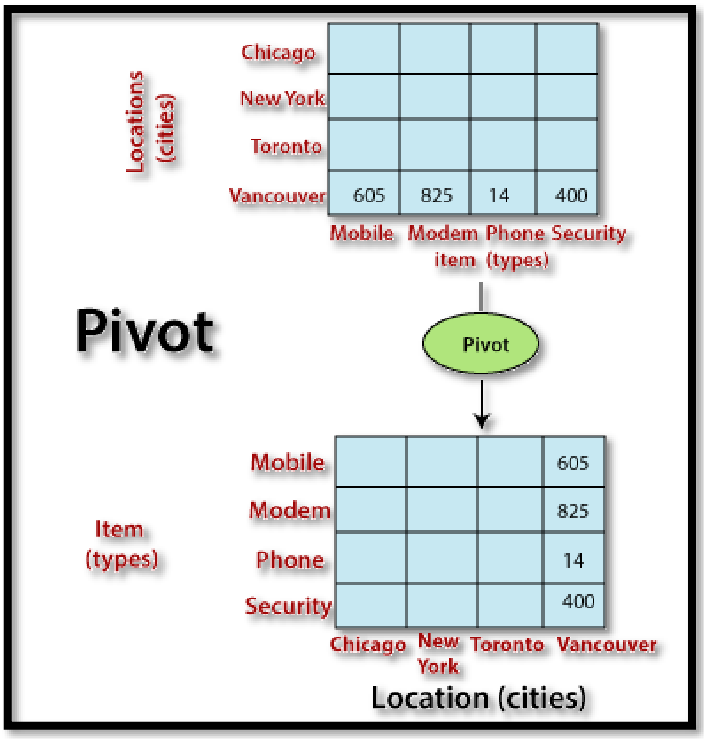
- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item =" Mobile" or "Modem")

Pivot

The pivot operation is also called a rotation. Pivot is a visualization operations which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.



Consider the following diagram, which shows the pivot operation.



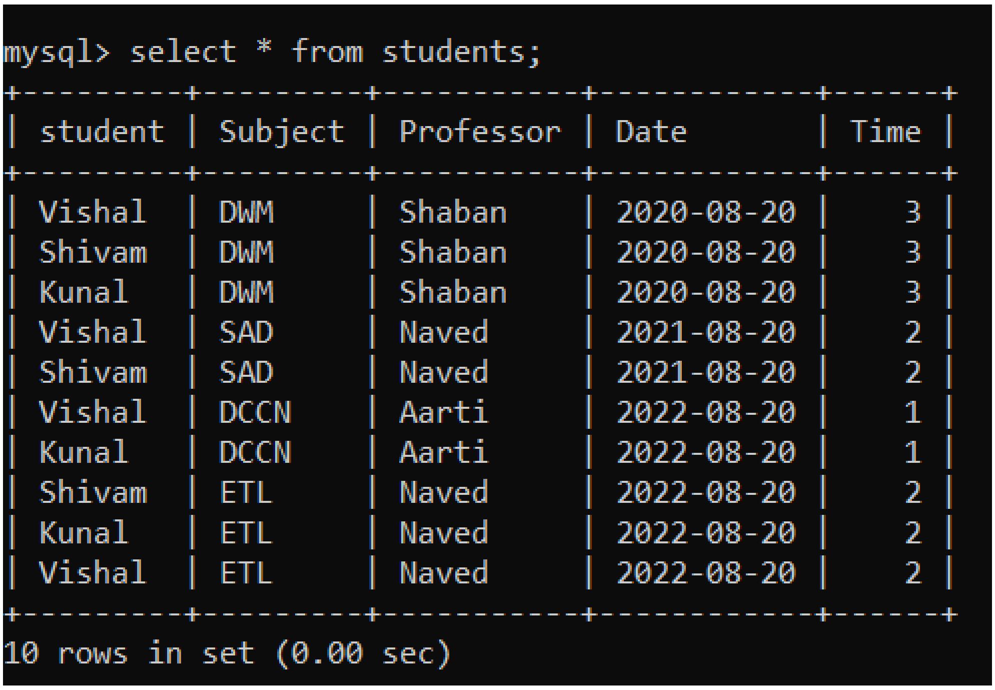
	Mobile	Modem	Phone	Security
Chicago				
New York				
Toronto				
Vancouver	605	825	14	400

	Chicago	New York	Toronto	Vancouver
Mobile				605
Modem				825
Phone				14
Security				400

Drill Up:

This operation you can meet as a part of pair drill up and drill down in OLAP. Drill-up is an operation to gather data from the cube either by ascending a concept hierarchy for a dimension or by dimension reduction in order to receive measures at a less detailed granularity. So that to see a broader perspective in compliance with the concept hierarchy a user has to group columns and unite the values. As there are fewer specifics, one or more dimensions from the data cube will be deleted, when this OLAP operation is run. In some sources drill up and roll up operations in OLAP come as synonyms, so this variant is also possible.

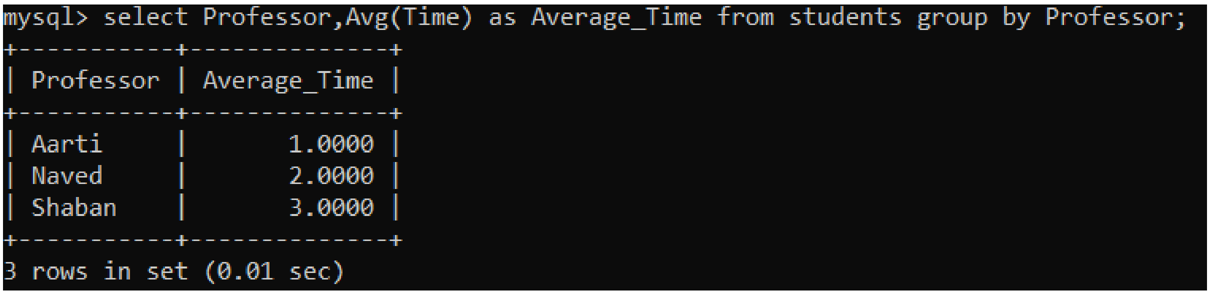
Fact Table:



```
mysql> select * from students;
+---------+---------+-----------+------------+------+
| student | Subject | Professor | Date       | Time |
+---------+---------+-----------+------------+------+
| Vishal  | DWM     | Shaban    | 2020-08-20 |    3 |
| Shivam  | DWM     | Shaban    | 2020-08-20 |    3 |
| Kunal   | DWM     | Shaban    | 2020-08-20 |    3 |
| Vishal  | SAD     | Naved     | 2021-08-20 |    2 |
| Shivam  | SAD     | Naved     | 2021-08-20 |    2 |
| Vishal  | DCCN    | Aarti     | 2022-08-20 |    1 |
| Kunal   | DCCN    | Aarti     | 2022-08-20 |    1 |
| Shivam  | ETL     | Naved     | 2022-08-20 |    2 |
| Kunal   | ETL     | Naved     | 2022-08-20 |    2 |
| Vishal  | ETL     | Naved     | 2022-08-20 |    2 |
+---------+---------+-----------+------------+------+
10 rows in set (0.00 sec)
```

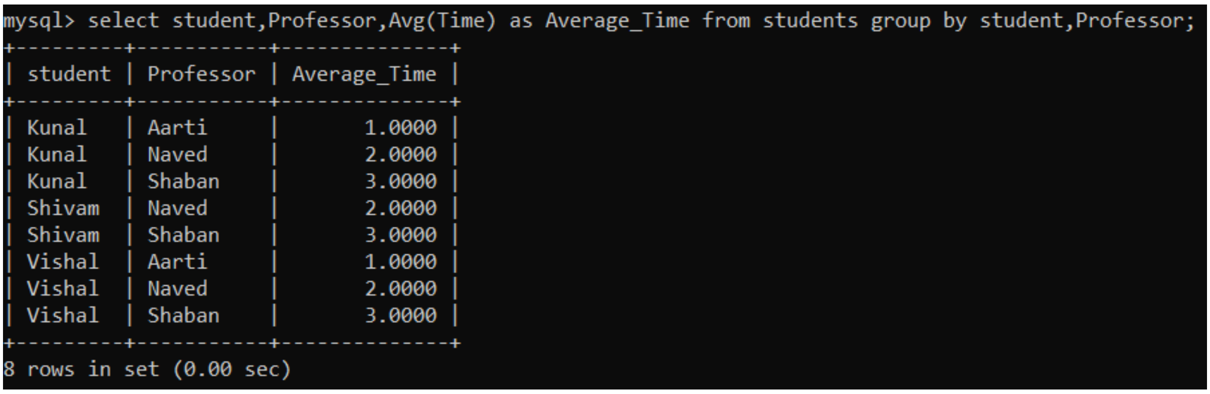
Operation:

Group By 1 D



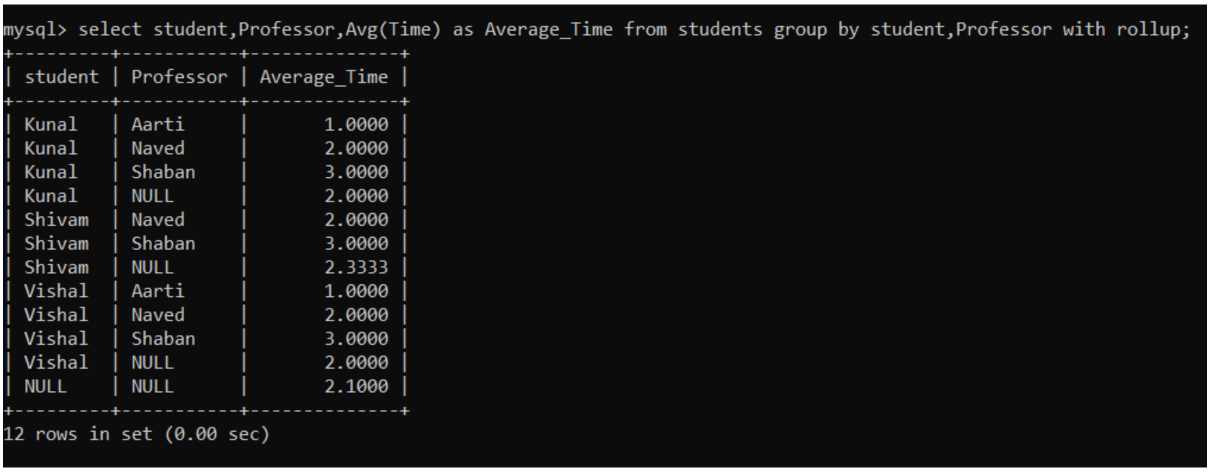
```
mysql> select Professor,Avg(Time) as Average_Time from students group by Professor;
+-----------+--------------+
| Professor | Average_Time |
+-----------+--------------+
| Aarti     |       1.0000 |
| Naved     |       2.0000 |
| Shaban    |       3.0000 |
+-----------+--------------+
3 rows in set (0.01 sec)
```

Group By 2D



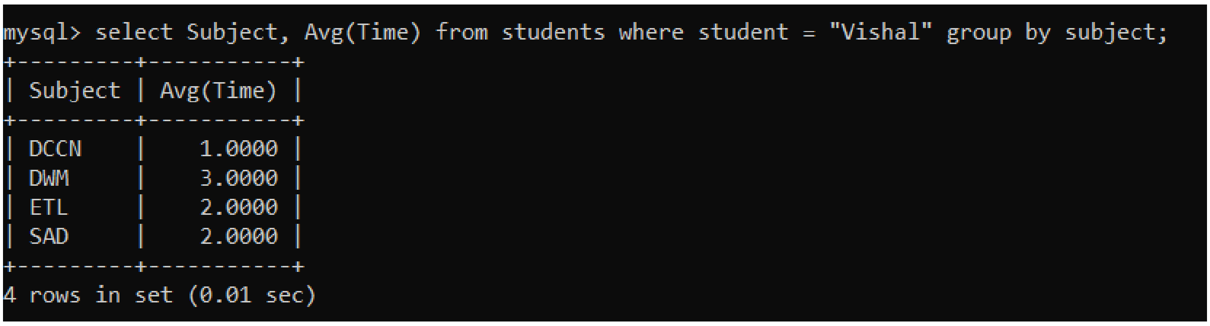
```
mysql> select student,Professor,Avg(Time) as Average_Time from students group by student,Professor;
+---------+-----------+--------------+
| student | Professor | Average_Time |
+---------+-----------+--------------+
| Kunal   | Aarti     |       1.0000 |
| Kunal   | Naved     |       2.0000 |
| Kunal   | Shaban    |       3.0000 |
| Shivam  | Naved     |       2.0000 |
| Shivam  | Shaban    |       3.0000 |
| Vishal  | Aarti     |       1.0000 |
| Vishal  | Naved     |       2.0000 |
| Vishal  | Shaban    |       3.0000 |
+---------+-----------+--------------+
8 rows in set (0.00 sec)
```

Group by Roll up (Drill Up)

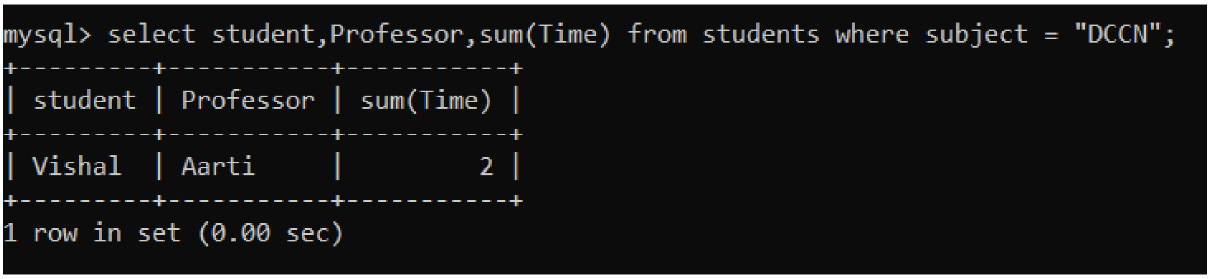


```
mysql> select student,Professor,Avg(Time) as Average_Time from students group by student,Professor with rollup;
+---------+-----------+--------------+
| student | Professor | Average_Time |
+---------+-----------+--------------+
| Kunal   | Aarti     |       1.0000 |
| Kunal   | Naved     |       2.0000 |
| Kunal   | Shaban    |       3.0000 |
| Kunal   | NULL      |       2.0000 |
| Shivam  | Naved     |       2.0000 |
| Shivam  | Shaban    |       3.0000 |
| Shivam  | NULL      |       2.3333 |
| Vishal  | Aarti     |       1.0000 |
| Vishal  | Naved     |       2.0000 |
| Vishal  | Shaban    |       3.0000 |
| Vishal  | NULL      |       2.0000 |
| NULL    | NULL      |       2.1000 |
+---------+-----------+--------------+
12 rows in set (0.00 sec)
```

Slice:

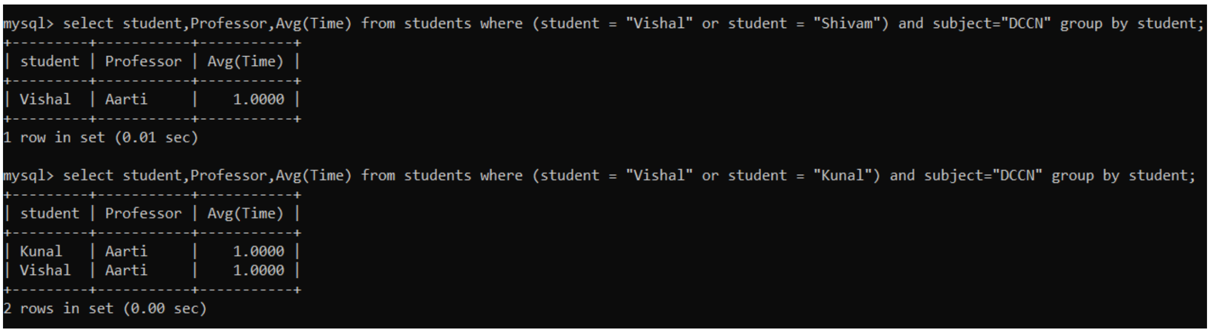


```
mysql> select Subject, Avg(Time) from students where student = "Vishal" group by subject;
+---------+-----------+
| Subject | Avg(Time) |
+---------+-----------+
| DCCN    |    1.0000 |
| DWM     |    3.0000 |
| ETL     |    2.0000 |
| SAD     |    2.0000 |
+---------+-----------+
4 rows in set (0.01 sec)
```

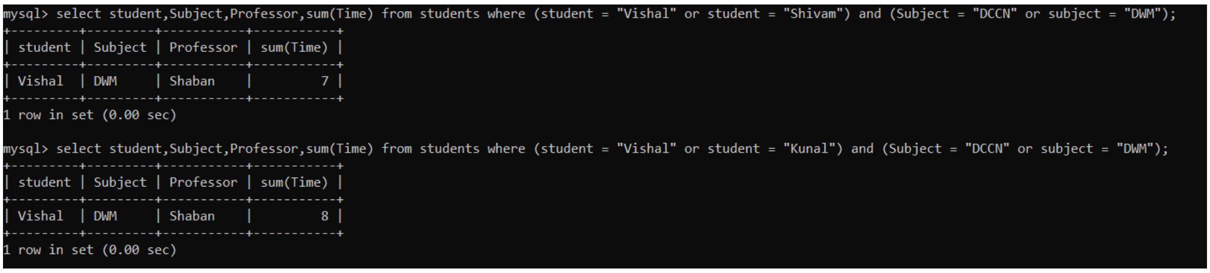


```
mysql> select student,Professor,sum(Time) from students where subject = "DCCN";
+---------+-----------+-----------+
| student | Professor | sum(Time) |
+---------+-----------+-----------+
| Vishal  | Aarti     |         2 |
+---------+-----------+-----------+
1 row in set (0.00 sec)
```

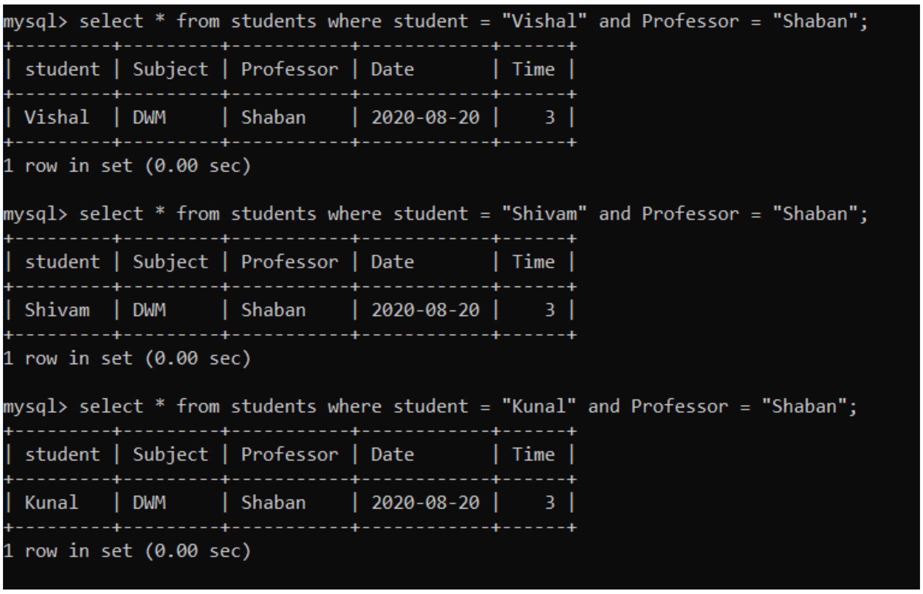
Dice:



```
mysql> select student,Professor,Avg(Time) from students where (student = "Vishal" or student = "Shivam") and subject="DCCN" group by student;
+---------+-----------+-----------+
| student | Professor | Avg(Time) |
+---------+-----------+-----------+
| Vishal  | Aarti     |    1.0000 |
+---------+-----------+-----------+
1 row in set (0.01 sec)

mysql> select student,Professor,Avg(Time) from students where (student = "Vishal" or student = "Kunal") and subject="DCCN" group by student;
+---------+-----------+-----------+
| student | Professor | Avg(Time) |
+---------+-----------+-----------+
| Kunal   | Aarti     |    1.0000 |
| Vishal  | Aarti     |    1.0000 |
+---------+-----------+-----------+
2 rows in set (0.00 sec)
```

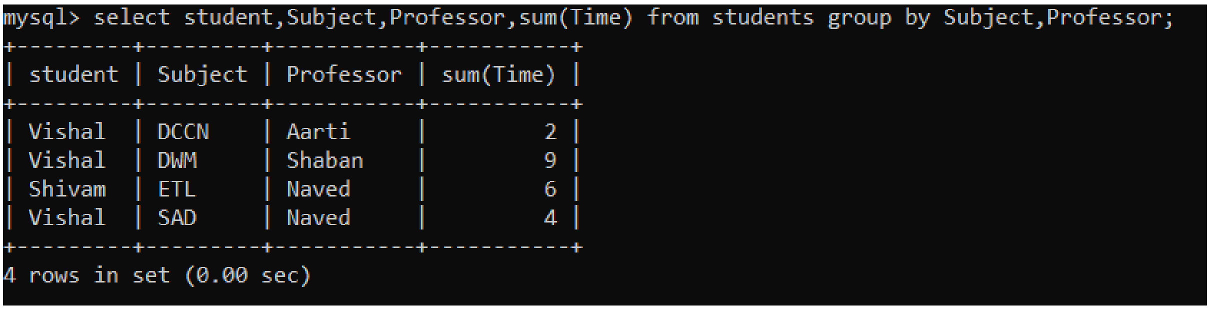


```
mysql> select student,Subject,Professor,sum(Time) from students where (student = "Vishal" or student = "Shivam") and (Subject = "DCCN" or subject = "DWM");
+---------+---------+-----------+-----------+
| student | Subject | Professor | sum(Time) |
+---------+---------+-----------+-----------+
| Vishal  | DWM     | Shaban    |         7 |
+---------+---------+-----------+-----------+
1 row in set (0.00 sec)

mysql> select student,Subject,Professor,sum(Time) from students where (student = "Vishal" or student = "Kunal") and (Subject = "DCCN" or subject = "DWM");
+---------+---------+-----------+-----------+
| student | Subject | Professor | sum(Time) |
+---------+---------+-----------+-----------+
| Vishal  | DWM     | Shaban    |         8 |
+---------+---------+-----------+-----------+
1 row in set (0.00 sec)
```

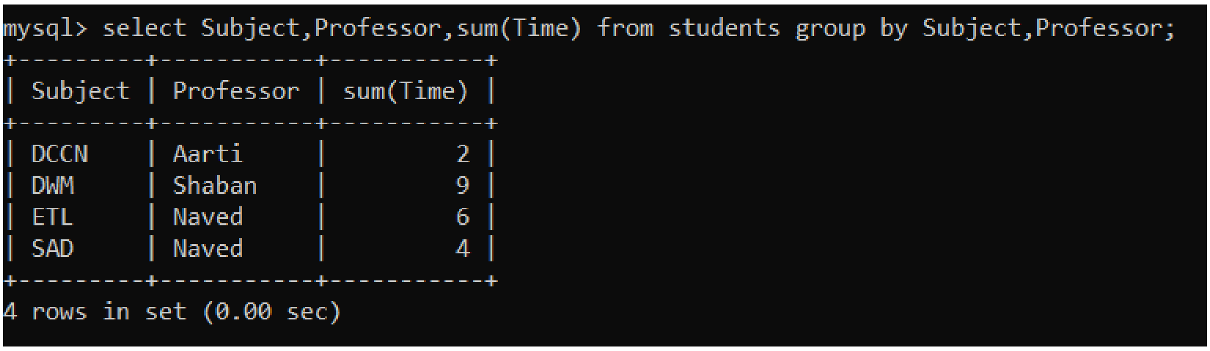


```
mysql> select * from students where student = "Vishal" and Professor = "Shaban";
+---------+---------+-----------+------------+------+
| student | Subject | Professor | Date       | Time |
+---------+---------+-----------+------------+------+
| Vishal  | DWM     | Shaban    | 2020-08-20 |    3 |
+---------+---------+-----------+------------+------+
1 row in set (0.00 sec)

mysql> select * from students where student = "Shivam" and Professor = "Shaban";
+---------+---------+-----------+------------+------+
| student | Subject | Professor | Date       | Time |
+---------+---------+-----------+------------+------+
| Shivam  | DWM     | Shaban    | 2020-08-20 |    3 |
+---------+---------+-----------+------------+------+
1 row in set (0.00 sec)

mysql> select * from students where student = "Kunal" and Professor = "Shaban";
+---------+---------+-----------+------------+------+
| student | Subject | Professor | Date       | Time |
+---------+---------+-----------+------------+------+
| Kunal   | DWM     | Shaban    | 2020-08-20 |    3 |
+---------+---------+-----------+------------+------+
1 row in set (0.00 sec)
```

Simple Group by

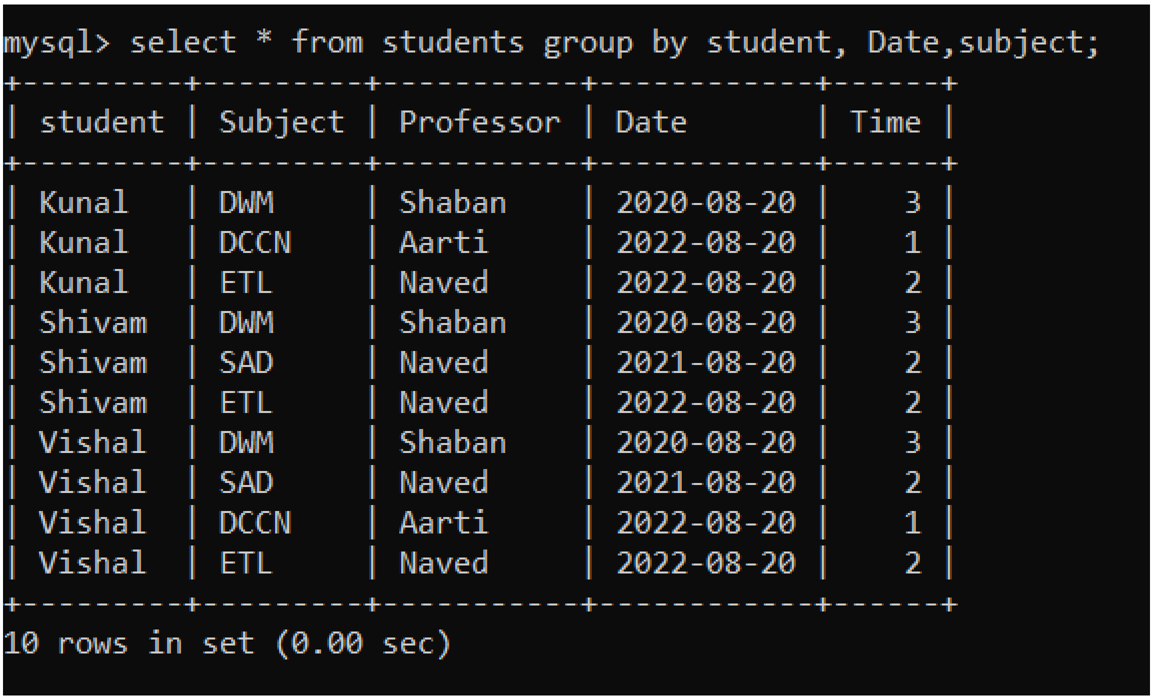


```
mysql> select student,Subject,Professor,sum(Time) from students group by Subject,Professor;
+---------+---------+-----------+-----------+
| student | Subject | Professor | sum(Time) |
+---------+---------+-----------+-----------+
| Vishal  | DCCN    | Aarti     |         2 |
| Vishal  | DWM     | Shaban    |         9 |
| Shivam  | ETL     | Naved     |         6 |
| Vishal  | SAD     | Naved     |         4 |
+---------+---------+-----------+-----------+
4 rows in set (0.00 sec)
```



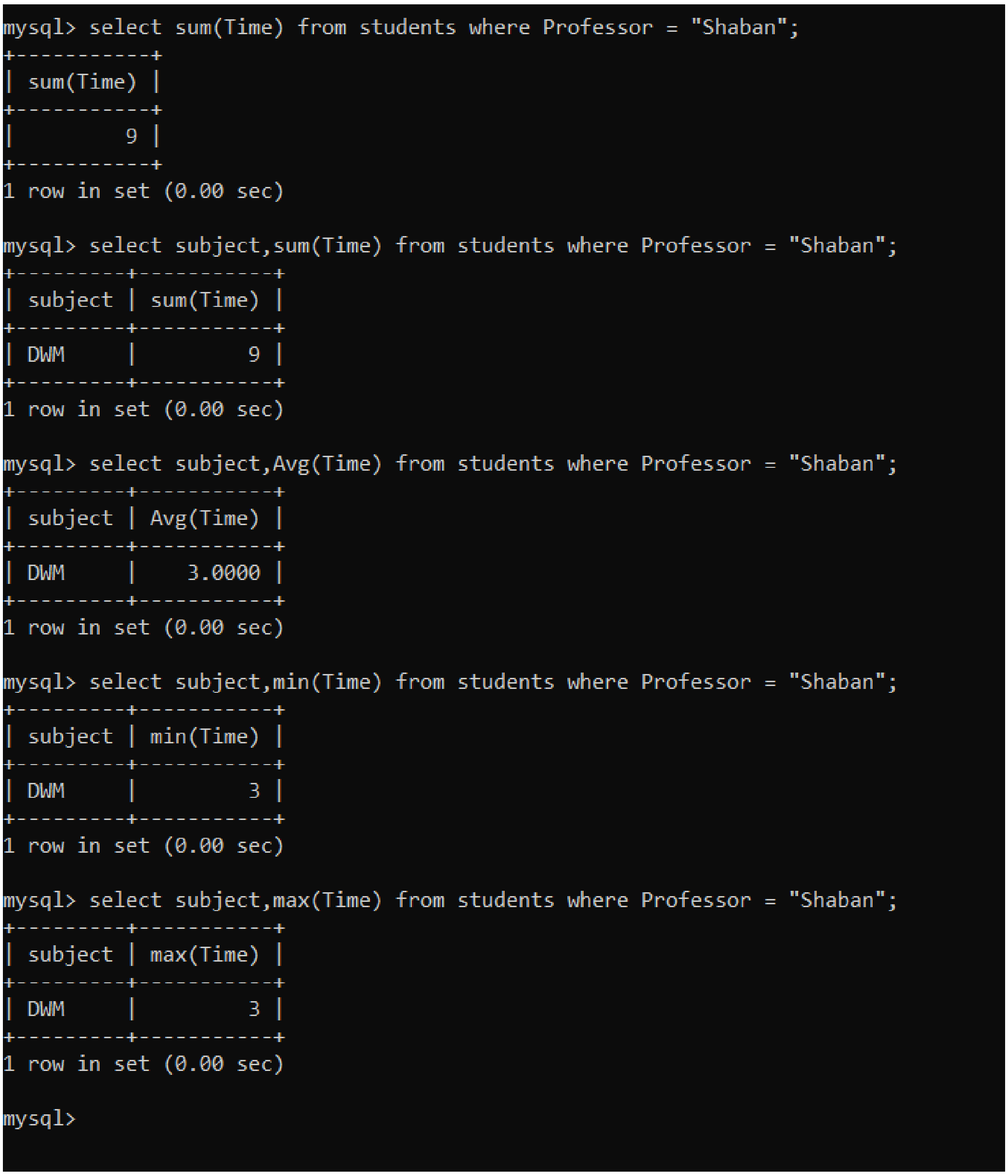
```
mysql> select Subject,Professor,sum(Time) from students group by Subject,Professor;
+---------+-----------+-----------+
| Subject | Professor | sum(Time) |
+---------+-----------+-----------+
| DCCN    | Aarti     |         2 |
| DWM     | Shaban    |         9 |
| ETL     | Naved     |         6 |
| SAD     | Naved     |         4 |
+---------+-----------+-----------+
4 rows in set (0.00 sec)
```

3 Group By



```
mysql> select * from students group by student, Date,subject;
+---------+---------+-----------+------------+------+
| student | Subject | Professor | Date       | Time |
+---------+---------+-----------+------------+------+
| Kunal   | DWM     | Shaban    | 2020-08-20 |    3 |
| Kunal   | DCCN    | Aarti     | 2022-08-20 |    1 |
| Kunal   | ETL     | Naved     | 2022-08-20 |    2 |
| Shivam  | DWM     | Shaban    | 2020-08-20 |    3 |
| Shivam  | SAD     | Naved     | 2021-08-20 |    2 |
| Shivam  | ETL     | Naved     | 2022-08-20 |    2 |
| Vishal  | DWM     | Shaban    | 2020-08-20 |    3 |
| Vishal  | SAD     | Naved     | 2021-08-20 |    2 |
| Vishal  | DCCN    | Aarti     | 2022-08-20 |    1 |
| Vishal  | ETL     | Naved     | 2022-08-20 |    2 |
+---------+---------+-----------+------------+------+
10 rows in set (0.00 sec)
```

Aggregate Functions:



```
mysql> select sum(Time) from students where Professor = "Shaban";
+-----------+
| sum(Time) |
+-----------+
|         9 |
+-----------+
1 row in set (0.00 sec)

mysql> select subject,sum(Time) from students where Professor = "Shaban";
+---------+-----------+
| subject | sum(Time) |
+---------+-----------+
| DWM     |         9 |
+---------+-----------+
1 row in set (0.00 sec)

mysql> select subject,Avg(Time) from students where Professor = "Shaban";
+---------+-----------+
| subject | Avg(Time) |
+---------+-----------+
| DWM     |    3.0000 |
+---------+-----------+
1 row in set (0.00 sec)

mysql> select subject,min(Time) from students where Professor = "Shaban";
+---------+-----------+
| subject | min(Time) |
+---------+-----------+
| DWM     |         3 |
+---------+-----------+
1 row in set (0.00 sec)

mysql> select subject,max(Time) from students where Professor = "Shaban";
+---------+-----------+
| subject | max(Time) |
+---------+-----------+
| DWM     |         3 |
+---------+-----------+
1 row in set (0.00 sec)

mysql>
```

Conclusion:

Through this experiment we understood the various OLAP operations which are to be performed on multidimensional data and how to write OLAP queries for our case study problem.

OLAP offers analytical modeling capabilities, containing a calculation engine for determining ratios, variance, etc. and for computing measures across various dimensions. It can generate summarization, aggregation, and hierarchies at each granularity level and at every dimension’s intersection. OLAP also provide functional models for forecasting, trend analysis, and statistical analysis.