Theory :

Stands for &quot; Online Analytical Processing.&quot; OLAP allows users to analyse database

information from multiple database systems at one time. While relational databases

are considered to be two-dimensional, OLAP data is multidimensional, meaning

the information can be compared in many different ways. For example, a company

might compare their computer sales in June with sales in July, then compare those

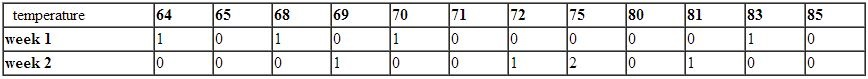
results with the sales from another location, which might be stored in a different

database.

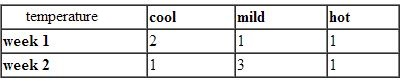
1. Roll up :

The roll-up operation (also called drill-up or aggregation operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by climbing down a concept hierarchy, i.e. dimension reduction.

Example :

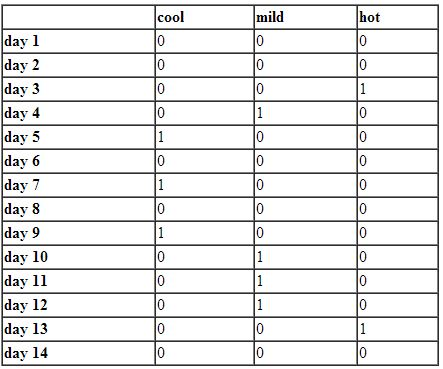


Roll up :

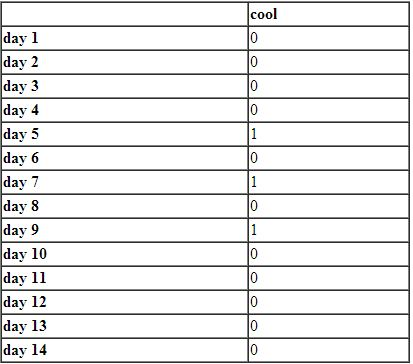


1. Roll Down :

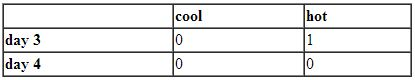
The roll down operation (also called drill down) is the reverse of roll up. It navigates from less detailed data to more detailed data. It can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.



1. Slicing : Slice performs a selection on one dimension of the given cube, thus resulting in a subcube.



1. Dicing : The dice operation defines a subcube by performing a selection on two or more dimensions.



1. Pivot : Pivot otherwise known as Rotate changes the dimensional orientation of the cube, i.e. rotates the data axes to view the data from different perspectives. Pivot groups data with different dimensions.

Program Code :

import sqlite3

conn = sqlite3.connect('sales.db')

c=conn.cursor()

c.execute("drop table if exists sales")

c.execute("""CREATE TABLE sales( item\_name text, color text, clothes\_size text, quantity int )""")

many\_types=[("skirt","dark","small",2), ("skirt","dark","large",1),("skirt","pastel","small",11),

            ("skirt","pastel","medium",9), ("skirt","pastel","large",15), ("skirt","white","small",2),

            ("skirt","dark","medium",5), ("dress","dark","medium",6), ("dress","dark","large",12),

            ("dress","pastel","small",4), ("dress","pastel","medium",3), ("dress","pastel","large",3),

            ("dress","white","small",2), ("dress","white","medium",3), ("dress","white","large",0),

            ("shirt","dark","small",2), ("shirt","dark","medium",6), ("shirt","dark","large",6),

            ("shirt","pastel","medium",1), ("shirt","pastel","small",4), ("shirt","pastel","large",2),

            ("shirt","white","small",17), ("shirt","white","medium",1), ("shirt","white","large",10),

            ("pants","dark","small",14), ("pants","dark","medium",6), ("pants","dark","large",0),

            ("pants","pastel","small",1), ("pants","pastel","medium",0), ("pants","pastel","large",1),

            ("pants","white","small",3), ("pants","white","medium",0), ("pants","white","large",2) ]

c.executemany ("INSERT INTO sales VALUES (?,?,?,?)", many\_types)

c.execute("SELECT \* FROM sales")

details=c.fetchall()

for det in details:

    print(det)

conn.commit()

c.execute("SELECT item\_name, color, clothes\_size, sum(quantity) FROM SALES ROLLUP")

#c.execute("SELECT item\_name,SUM(quantity) FROM sales GROUP BY item\_name")

details=c.fetchall()

for det in details:

    print(det)

conn.commit()

c.execute("SELECT color,clothes\_size,SUM(quantity) FROM sales WHERE item\_name='pants'GROUP BY color,clothes\_size")

details=c.fetchall()

for det in details:

    print(det)

conn.commit()

c.execute("SELECT item\_name,SUM(quantity) FROM sales WHERE item\_name='pants'AND color='dark' GROUP BY item\_name")

details=c.fetchall()

for det in details:

    print(det)

conn.commit()

conn.close()

Screenshots:

