**1) In data warehouse technology, a multiple dimensional view can be implemented by a relational database technique (ROLAP), or by a multidimensional database technique (MOLAP), or a hybrid database technique (HOLAP).**

**(a) Briefly describe each implementation technique.**

**Ans:**

The following three different types of OLAPs represent the middle tier of the three-tier data warehouse server architecture.

ROLAP (Relational OLAP servers): Intermediate servers that are found between relational back-end server and client front-end tools. They make use of RDBMSs or extended RDBMSs to store/manage warehouse data + OLAP middleware to fill in the gaps. ROLAP servers have the ability to optimize the all back end DBMSs, as well as deployment of “aggregation navigation logic,” and other tools/services. ROLAPs tend to be more scalable than MOLAP.

**(b) For each technique, explain how each of the following functions may be implemented:**

**Ans :**

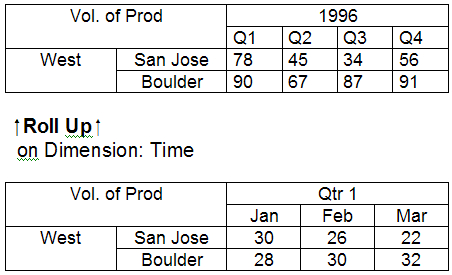
**i. The generation of a data warehouse (including aggregation).**

Initial aggregation may be accomplished in SQL via group-bys. The compute cube operator computes aggregates over all subsets of the dimensions in the specified operation; this leads to the generation of a single cube. ROLAP relies on tuples and relational tables as its basic data structures. The base fact table (a relational table) stores data at the abstraction level indicated by join keys in the schema for the given data cube. Aggregated data can also be stored in fact tables (summary fact tables). ROLAP uses value-based addressing, where dimension values are accessed via key-based addressing search strategies. To optimize ROLAP cube computation we may use the following techniques:

- sorting, hashing, grouping operations  
- grouping is performed on sub-aggregates  
- aggregates derived from previously computed aggregates

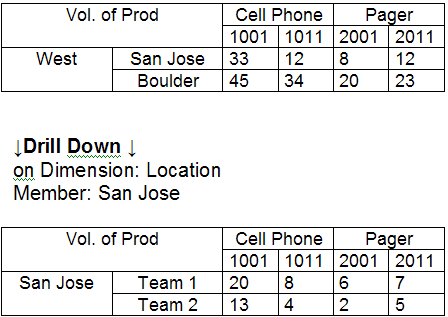
**ii. Roll-up:**

Aggregation on a data cube (aka dimension reduction). In ROLAP, this means that the relational tables are aggregated from more to less specific.

[](http://www.12345w.xyz/info-source-us/data_warehousing_mining/images/roll-up.png)

**iii. Drill-down:**

The opposite of Roll-up. We introduce additional dimensions into the relation tables and, hence, cubes.

[](http://www.12345w.xyz/info-source-us/data_warehousing_mining/images/drill-down.png)

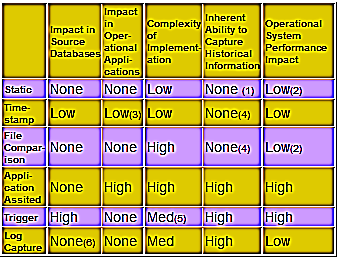
**iv. Incremental updating:**

Data warehouse implementation can be broken down into segments or increments. An increment is a defined data warehouse implementation project that has a specified beginning and end. An increment may also be referred to as a departmental data warehouse within the context of an enterprise.

A ROLAP server would take require the use of appropriate tools such as those made by Informix. Since ROLAPs are based on relation databases, the updating method would (I think) be performed in a manner similar to those in traditional RDBMS, and then grafted onto the data cube(s). Some of the techniques are summarized in this article from the above link:

Incremental data capture is a time-dependent model for capturing changes to operational systems. This technique is best applied in circumstances where changes in the data are significantly smaller than the size of the data set for a specific period of time (i.e., the time between captures). These techniques are more complex than static capture, because they are closely tied to the DBMS or the operational software which updates the DBMS. Three different techniques in this category are application-assisted capture, trigger-based capture and transaction log capture. In circumstances where DBMSs are used to store operational data, transaction log capture is the most powerful and provides the most efficient approach to incremental capture. Some of the incremental techniques used are listed in Figure 1 below.

FIGURE 1: Incremental Update Techniques

[](http://www.12345w.xyz/info-source-us/data_warehousing_mining/images/incremental_update.png)

**Which implementation do you prefer?**

Hybrid approach seems to be the best solution for most applications. It would appear to be backward compatible with older(?) ROLAPs and retains the scalability of this implementation. At the same time, it incorporates the more sophisticated features and faster computation of MOLAP.

**2) Explain difference between Rollup and cube function in postgresql.**

**Ans :**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **CUBE** | **ROLLUP** | | It’s an additional switch to GROUP BY clause. It can be applied to all aggregation functions to return cross tabular result sets. | It’s an extension to GROUP BY clause. It’s used to extract statistical and summarized information from result sets. It creates groupings and then applies aggregation functions on them. | | Produces all possible combinations of subtotals specified in GROUP BY clause and a Grand Total. | Produces only some possible subtotal combinations. | |

**Example:**  
**A table product has the following records: -**

|  |  |  |
| --- | --- | --- |
| Apparel | Brand | Quantity |
| Shirt | Gucci | 124 |
| Jeans | Lee | 223 |
| Shirt | Gucci | 101 |
| Jeans | Lee | 210 |

**CUBE** can be used to return a result set that contains the Quantity subtotal for all possible combinations of Apparel and Brand:

SELECT Apparel, Brand, SUM(Quantity) AS QtySum  
FROM product  
GROUP BY Apparel, Brand WITH CUBE

**The query above will return:**

|  |  |  |
| --- | --- | --- |
| Apparel | Brand | Quantity |
| Shirt | Gucci | 101.00 |
| Shirt | Lee | 210.00 |
| Shirt | (null) | 311.00 |
| Jeans | Gucci | 124.00 |
| Jeans | Lee | 223.00 |
| Jeans | (null) | 347.00 |
| (null) | (null) | 658.00 |
| (null) | Gucci | 225.00 |
| (null) | Lee | 433.00 |

**ROLLUP: -** Calculates multiple levels of subtotals of a group of columns.  
  
**Example:**

SELECT Apparel,Brand,sum(Quantity) FROM Product GROUP BY ROLLUP (Apparel,Brand);

The query above will return a sum of all quantities of the different brands.