**OLAP**

*OLAP is a category of software technology that enables analysts, managers, and executives to gain insight into the data through fast, consistent, interactive, access in a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user.*

**Demand for OLAP:**

* Need for Multidimensional Analysis
* Fast Access & Powerful Calculations
* Limitations of other analysis methods like:
  + SQL
  + Spreadsheets
  + Report Writers
* Traditional tools of report writers, query products, spreadsheets, & language interfaces do not match the user expectations as far as performing multidimensional analysis with complex calculations is concerned.
* Tools used with OLTP and basic DW environments do not match up to the task

**Why is OLAP useful?**

* Facilitates multidimensional data analysis by pre-computing aggregates across many sets of dimensions
* Provides for:
  + Greater speed and responsiveness
  + Improved user interactivity

**A Multidimensional model**

* A data warehouse is based on a multidimensional data model which views data in the form of a data cube
* A data cube allows data to be modeled and viewed in multiple dimensions.

It is defined by dimensions and facts.

**Dimensions** are the perspectives or entities with respect to which an organization wants to keep records. Each dimension may have a table associated with it, called a **dimension table**, which further describes the dimension.

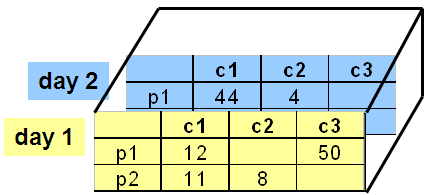
**Fact** is a value or measurement, which represents a fact about the managed entity or system. The quantities by which we want to analyze relationships between dimensions.

**CUBE**

**Fact table view:**



Multi-dimensional cube:



Dimension=3

**Aggregates**

* Aggregate- summarizing & storing data which is available in the fact table in order to improve the performance of end-user query.
* **Costs**- Storage, Processing & Monitoring
* No aggregates or Exhaustive aggregates or Selective aggregates?

**Which aggregates to create?**

**The size of the database should not become double of its original size.**

Aggregates Guidelines

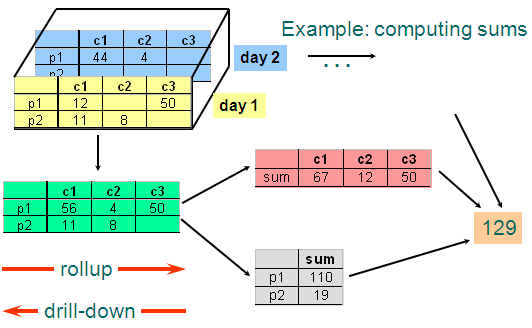
* Set an aggregate storage limit.
* Define small aggregates: 10 to 20 times smaller than the Fact table or aggregate on which it is based.

**Sparsity** – it refers to the degree to which the cells of cube are not filled with data.

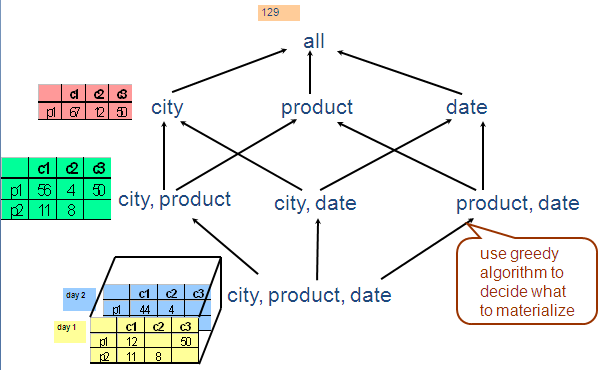
Monthly product sales aggregate: how many times smaller than daily product sales table?

* 30?
* 1-way:Category by Store by Day
* 1-way:Product by Dist by Day
* 1-way:Category by Dist by Month
* 2-way:Category by Dist by Day
* 2-way:Category by Store by Month
* 2-way:Product by Dist by Month
* 3-way:Category by Dist by Month
* **Operators: sum, count, max, min, median, avg.**
* “Having” clause
* Using dimension hierarchy
  + average by region (within store)
  + maximum by month (within date)

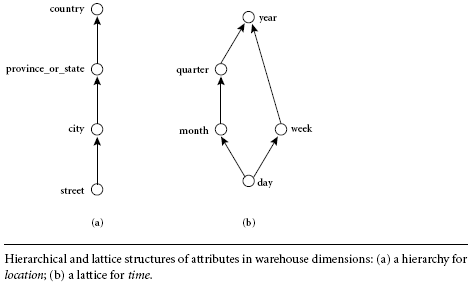
**Cube Aggregation**



**Cube Aggregates Lattice**



A **concept hierarchy** defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts.



**OLAP operations:**

* ***Roll-up:*** The roll-up operation (also called the *drill-up* operation by some vendors) performs aggregation on a data cube, either by *climbing up a concept hierarchy* for a dimension or by *dimension reduction*.

When roll-up is performed by dimension reduction, one or more dimensions are removed from the given cube. For example, consider a sales data cube containing only the two dimensions *location* and *time*. Roll-up may be performed by removing, say, the *time* dimension, resulting in an aggregation of the total sales by location, rather than by location and by time.

* ***Drill-down***: Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data. Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.
* ***Dice***: The dice operation defines a subcube by performing a selection on two or more dimensions.

When we use GROUP BY to specify part of a hierarchy, we are performing a range selection called a DICE

* ***Slice***: The slice operation performs a selection on one dimension of the given cube,

When we use WHERE to specify a particular value for an axis, we are performing a SLICE

* ***Pivot*** (rotate): Pivot (also called rotate) is a visualization operation that rotates the data axes in view in order to provide an alternative presentation of the data.

Other OLAP operations:

* **Drill-Across**: Queries involving more than one fact table
* **Drill-Through:** Makes use of SQL to drill through the bottom level of a data cube down to its back-end relational tables

Example schema:

* Fact Table

**Sales(Store\_id, Product\_id, Time\_id, Sales\_amt)**

* Dimension Tables

**Store (Store\_id, city, state, region, country)**

**Product (Product\_id, name, category)**

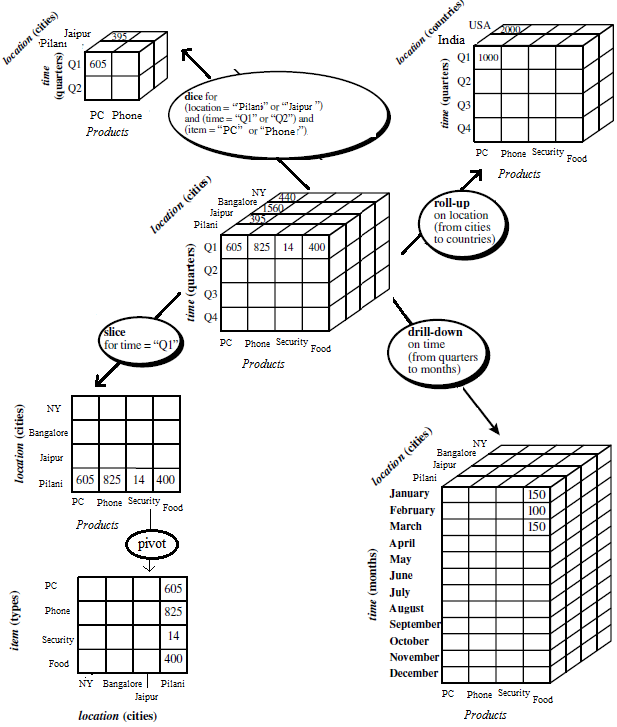
**Day (Time\_id, month, quarter, year)**

* Hierarchies

**Store 🡺 City 🡺 State 🡺 Region 🡺 Country**

**Product 🡺 Category**

**Day 🡺 Month 🡺 Quarter 🡺 Year**

****

**Fig. OLAP operations**

**Conceptual vs. Actual**

* The “cube” is a logical way of visualizing the data in an OLAP setting
* Not how the data is actually represented on disk
* Two ways of storing data:
  + ROLAP: Relational OLAP
  + MOLAP: Multidimensional OLAP

It is all about which DBMS you choose to store your data warehouse data

* RDBMS – ROLAP
* MDDB – MOLAP
* BOTH – HOLAP

Three possibilities for OLAP servers

(1) **Relational OLAP (ROLAP)**

* + Relational and specialized relational DBMS to store and manage warehouse data
  + OLAP middleware to support missing pieces

(2) **Multidimensional OLAP (MOLAP)**

* + Array-based storage structures
  + Direct access to array data structures

(3) **Hybrid OLAP (HOLAP)**

* + Storing detailed data in RDBMS
  + Storing aggregated data in MDBMS
  + User access via MOLAP tools

**ROLAP -** These are the intermediate servers that stand in between a relational back-end server and client front-end tools. They use a *relational or extended-relational DBMS* to store and manage warehouse data, and OLAP middleware to support missing pieces.

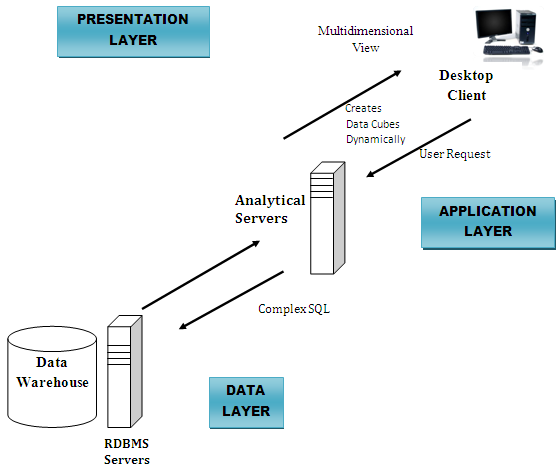
* In ROLAP model, data is stored as rows and columns in relational form. To present data multidimensionally, a semantic layer of metadata is created. The metadata layer supports the mapping of dimensions to the relational tables.
* An analytical server creates multidimensional views on the fly. When the user issues complex queries based on this multidimensional view, the queries are transformed into complex SQL directed to the relational database.

.

* Special schema design: *star, snowflake*
* Special indexes: bitmap, multi-table join
* Proven technology (relational model, DBMS), tend to outperform specialized MDDB especially on large data sets
* Products
  + IBM DB2, Oracle, Sybase IQ, RedBrick, Informix
* Defines complex, multi-dimensional data with simple model
* Reduces the number of joins a query has to process
* Allows the data warehouse to evolve with relatively low maintenance
* Can contain both detailed and summarized data.
* ROLAP is based on familiar, proven, and already selected technologies.

**BUT!!!**

* **SQL for multi-dimensional manipulation of calculations.**

****

**Fig: The ROLAP model**

**Multidimensional OLAP (MOLAP)** servers: These servers support multidimensional views of data through *array-based multidimensional storage engines*. They map multidimensional views directly to data cube array structures. The advantage of using a data cube is that it allows fast indexing to precomputed summarized data.

* MDDB: a special-purpose data model
* Facts stored in multi-dimensional arrays
* Dimensions used to index array
* Sometimes on top of relational DB
* Products
  + Pilot, Arbor Essbase, Gentia

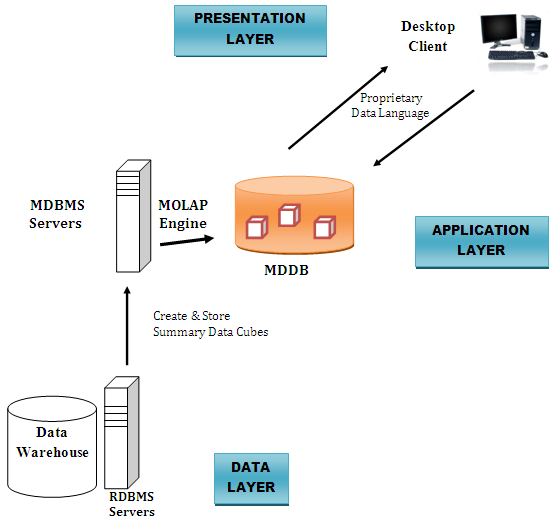
Pre-calculating or pre-consolidating transactional data improves speed.

BUT

* Fully pre-consolidating incoming data, MDDs require an enormous amount of overhead both in processing time and in storage. An input file of 200MB can easily expand to 5GB

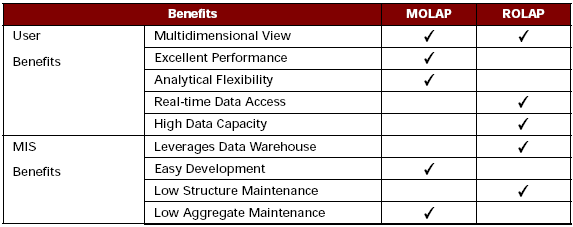
MDDBs are great candidates for the **<** 100GB department data marts.

* With MDDs, application design is essentially the definition of dimensions and calculation rules, while the RDBMS requires that the database schema be a star or snowflake.



**Fig: The MOLAP model**

**ROLAP vs MOLAP**



* Performance:
  + - How fast will the system appear to the end-user?
    - MDD server vendors believe this is a key point in their favor.
* Data volume and scalability:
  + - While MDD servers can handle up to 100GB of storage, RDBMS servers can handle hundreds of gigabytes and terabytes.

**Hybrid OLAP (HOLAP)** servers: The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP.

* + Best of both worlds
  + Storing detailed data in RDBMS
  + Storing aggregated data in MDBMS
  + User access via MOLAP tools

**ROLAP, MOPAL, or HOLAP**

**IF**

1. You require write access
2. Your data is under 50 GB
3. Your timetable to implement is 60-90 days
4. Lowest level already aggregated
5. Data access on aggregated level
6. You’re developing a general-purpose application for inventory movement or assets management

**THEN**

**Consider an MDD /MOLAP solution for your data mart**

**IF**

1. Your data is over 100 GB
2. You have a "read-only" requirement
3. Historical data at the lowest level of granularity
4. Detailed access, long-running queries
5. Data assigned to lowest level elements

**THEN**

**Consider an RDBMS/ROLAP solution for your data mart.**

**IF**

1. OLAP on aggregated and detailed data
2. Different user groups
3. Ease of use and detailed data

**THEN**

**Consider an HOLAP for your data mart**

**The Process of Data Warehouse Design**

A data warehouse can be built using a *top-down approach*, a *bottom-up approach*, or a *combination of both*.

In general, the warehouse design process consists of the following steps:

1. Choose a ***business process***to model, for example, orders, invoices, shipments, inventory, account administration, sales, or the general ledger. If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed. However, if the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen.
2. Choose the **grain** of the business process. The grain is the fundamental, atomic level of data to be represented in the fact table for this process, for example, individual transactions, individual daily snapshots, and so on.
3. Choose the **dimensions** that will apply to each fact table record. Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status.
4. Choose the **measures** that will populate each fact table record. Typical measures are numeric additive quantities like dollars sold and units sold.

**Three-Tier Data Warehouse Architecture**

1. **The bottom tier** is a warehouse database server that is almost always a relational database system. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources (such as customer profile information provided by external consultants).

This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

1. **The middle tier** is an OLAP server that is typically implemented using either

(1) a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or

(2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

1. **The top tier** is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on).

