U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, 1996.

J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.

T. Imielinski and H. Mannila. A database perspective on knowledge discovery. Communications of ACM, 39:58-64, 1996.

G. Piatetsky-Shapiro, U. Fayyad, and P. Smith. From data mining to knowledge discovery: An overview. In U.M. Fayyad, et al. (eds.), Advances in Knowledge Discovery and Data Mining, 1-35. AAAI/MIT Press, 1996.

G. Piatetsky-Shapiro and W. J. Frawley. Knowledge Discovery in Databases. AAAI/MIT Press, 1991.

Agarwal S., Agarwal R., Deshpande P.M ., and Gupta A. On the computation of multidimensional aggregates.

Virmani A. Second Generation Data Mining: Concepts and implementation;Ph.D Thesis, Rutgers University, April 1998.

Barbara D.(ed) Special Issue on mining of Large Datasets; IEEE Data Engineering Bulletin,21(1),1998.

Nestorov S., and Tsur S. Integrating data mining with relational DBMS: A tightly coupled approach, www-db.stanford.

Heckerman D. Bayesian networks for data mining. Data Mining and knowledge Discovery,1997.

Agarwal R, Gupta A., and Sarawagi S. modeling multidimensional databases ,ICDE 1997.

# Unit 2

# DATA WAREHOUSE OLAP TECHNOLOGY FOR DATA MINING

## INTRODUCTION

Data warehousing, online analytical processing(OLAP) and data mining are some of the growing trends in information technology for accessing and processing of data. In today's business scenario, a data warehouse is more of a necessity than an accessory for a progressive, competitive, and focused organization. A data warehouse can be defined as a repository of purposely selected and adapted operational data, which can be successfully answer any ad hoc, complex, statistical or analytical queries. The goal of data warehouse application in an enterprise is to increase the effectiveness of the enterprise decision making process.

Data warehouses provide on-line analytical processing(OLAP) tools for the interactive analysis of multidimensional data of varied granularities, which facilitates effective data mining. Data in the data warehouse is preprocessed and presented such that it facilitates many data mining functions, such as classification, prediction, association, and clustering.

In this chapter, the basic concepts, general architectures and major implementation techniques employed in data warehouse and OLAP technology will be discussed.

### 2.1 What is a Data Warehouse?

- A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.
- A data warehouse is an integrated and consolidated collection of data.
- A data warehouse is a read-only analytical database used for a decision support system operation.
- A data warehouse for decision support is often taking data from various platforms, databases, and files as source data.
- Data warehouse is a collection of integrated de-normalized databases for fast response performance.
- In general, a data warehousing built on the basis of a historical data model, possibly combined with other information dimensions.

Data warehouses have been defined in many ways, making it difficult to formulate a rigorous definition. The major features of a data warehouse are represented by the four keywords, subject-oriented , integrated, time-variant, and non-volatile, which distinguish data warehouses from other data repository systems, such as relational data base systems, transaction processing systems and file systems. Let's take a closer look at each of these key features.

**Subject-oriented:**

- Organized around major subjects, such as customer, product and sales.
- Focusing on the modeling and analysis of data for decision makers, not on daily operations or transaction processing.
- Provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.



**Integrated:**

- Constructed by integrating multiple, heterogeneous data sources, such as, relational databases, flat files, on-line transaction records.
- Data cleaning and data integration techniques are applied for ensure consistency in naming conventions, encoding structures, attribute measures, etc. among different data sources.
  - E.g., Hotel price: currency, tax, breakfast covered, etc.
- When data is moved to the warehouse, it is converted.

**Time-variant:**

- The time horizon for the data warehouse is significantly longer than that of operational systems.
  - Operational database: current value data.
  - Data warehouse data: provide information from a historical perspective (e.g., past 5-10 years)
- Every key structure in the data warehouse
  - Contains an element of time, explicitly or implicitly,
  - But the key of operational data may or may not contain "time element".

**Nonvolatile:**

- A physically separate store of data transformed from the operational environment.
- Operational update of data does not occur in the data warehouse environment.
- It does not require transaction processing, recovery, and concurrency control mechanisms.
- It requires only two operations in data accessing.
  - initial loading of data and access of data.

**What is data warehousing?**

Based on the above, data warehousing as the process of constructing and using data warehouses.

- The construction of a data warehouse requires data integration, data cleaning , and data consolidation.
- The utilization of data warehouse often necessitates a collection of decision support technologies. This allows "knowledge workers"(e.g., managers, analysts, and executives) to use the warehouse to quickly and conveniently obtain an overview of the data, and to make sound decisions based on information in the warehouse.

#### 2.1.1 Differences between Operational Database Systems and Data Warehouses

The major task of on-line operational database systems is to perform on-line transaction and query processing. These systems are called on-line transaction processing(OLTP) systems. They cover most of the day-to-day operations of an organization, such as purchasing, inventory

manufacturing, banking, payroll registration, and accounting. Data warehouse system, on the other hand serve users or knowledge workers in the role of data analysis and decision making. Such systems can organize and presents data in various formats in order to accommodate the diverse needs of the different users. These systems are known as on-line analytical processing(OLAP) systems. An OLAP system manages large amount of historical data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity Fig 2.1 and Fig 2.2 show the OLTP query and OLAP query. The major distinguishing features between OLTP and OLAP summarized(Table 2.1) as follows.

**Users and system orientation:** An OLTP system is customer-oriented and is used for transaction and query processing by clerk, clients and information technology professionals. An OLAP system is market-oriented and used for data analysis by knowledge workers, including managers, executives and analysts.

**Data contents:** An OLAP system manages current data that, typically, are too detailed to be easily used for decision making. An OLAP system manages large amounts of historical data provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity. These features make the data easier to use in informed decision making.

**Database Design:** An OLAP system usually adopts an entity-relationship (E-R)data model and an application-oriented database design. An OLAP system typically adopts either a star or snowflake model and a subject-oriented database design.

**View:** An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historical data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema, due to the

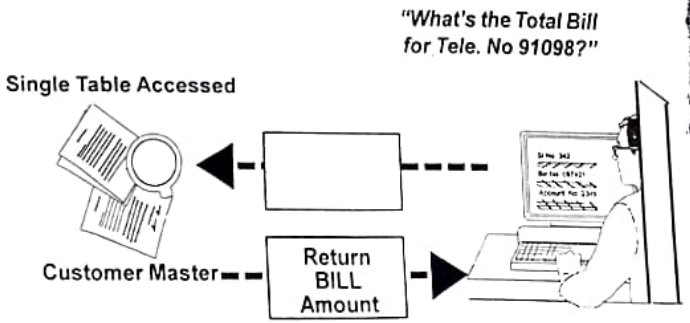



Fig 2.1: OLTP Query Example



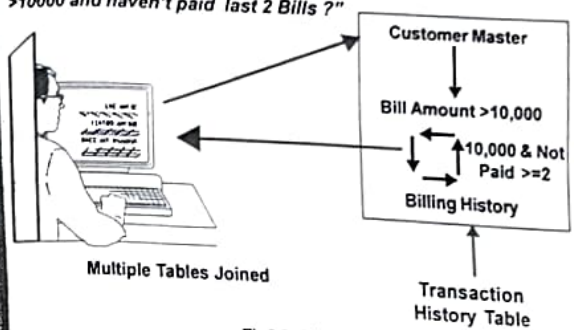



Fig 2.2: OLAP Query Example

evolutionary process of an organizations. OALP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.

**Access patterns:** The access patterns of an OLAP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanism. However, accesses to OLAP systems are mostly read only operations (since most data warehouses store historical rather than up-to-date information), although many could complex queries. These are summarized in Table 2.1

| | OLTP | OLAP |
|---|---|---|
| users | clerk, IT professional | knowledge worker |
| function | day to day operations | decision support |
| DB design | application-oriented | subject-oriented |
| data | current, up-to-datedetailed, flat relationalisolated | historical, summarized, multidimensionalintegrated, consolidated |
| usage | repetitive | ad-hoc |
| access | read/writeindex/hash on prim. key | lots of scans |
| unit of work | short, simple transaction | complex query |
| # records accessed | tens | millions |
| #users | thousands | hundreds |
| DB size | 100MB-GB | 100GB-TB |
| metric | transaction throughput | query throughput, response |

Table 2.1: OLTP vs. OLAP

### 2.2 Why Separate Data Warehouse?

Many organizations use data warehouse information to support business decisions making activities, including

- Increasing customer focus, which includes the analysis of customer buying patterns (such as buying preference, buying time, budget cycles, and appetites for spending)
- Repositioning products and managing product portfolios by comparing the performance of sales by quarters, by year and by geographic regions, in order to fine-tune the production strategies
- Managing the customer relationships, making environmental corrections and managing the cost of corporate assets.

- Many organizations typically collect diverse kinds of data and maintain large databases from multiple, heterogeneous, autonomous and distributed information sources. To integrate such data, and provide easy and efficient access to it, data warehouse is very much useful.
- The traditional database approach follows the query-driven approach, which requires complex information filtering and integration processes, and competes for resources with processing at local sources. It is inefficient and potentially expensive for frequent queries, especially for queries requiring aggregations. Whereas data warehouse employs an update-driven approach in which information from multiple, heterogeneous sources is integrated in advance and stored in a warehouse for direct querying and analysis.
- An operational database designed and tuned from known tasks and workload such as indexing and hashing using primary key, searching for particular records, and optimizing "canned" queries. On the other hand, data warehouse queries are often complex. They involve the computation of large groups of data at summarized levels, and may require the use of special data organization, access and implementation methods based on multidimensional views, processing OLAP queries in operational databases would substantially degrade the performance of operational tasks.
- An operational database supports the concurrent processing of multiple transactions. Concurrency control and recovery mechanism, such as locking and logging, are required to ensure the consistency and robustness of transactions. An OLAP query often needs read-only access of data records for summarization and aggregation.
- Decision support requires historical data which operational databases do not typically maintain.
- Decision support requires consolidation (aggregation, summarization) of data from heterogeneous sources resulting in high-quality, clean, and integrated data. In contrast, operational databases contain only detailed raw data, such as transactions, which need to be consolidated before analysis.

Since the two systems provide quite different functionalities and require different kinds of data, it is presently necessary to maintain separate databases. However, many vendors of operational relational database management systems are beginning to optimize such systems so as to support OLAP queries. As this trend continues, the separation between OLTP and OLAP systems is expected to decrease.



## 2.2 A Multidimensional Data Model

A data warehouse is based on a multidimensional data model which views data in the form of a data cube. In this section, you will learn how data cubes are modeled, concept hierarchies and how they can be used in basic OLAP operations to allow interactive mining at multiple levels of abstraction.

### 2.2.1 Data Cubes

- A data warehouse is based on a multidimensional data model which views data in the form of a data cube. The 3-D data cube is given in fig 2.3 by adding one more dimension time for the 2-D sales database shown in table 2.2.
- A data cube allows data to be modeled and viewed in multiple dimensions
  - Dimension tables.
  - Fact table contains measures and keys to each of the related dimension tables.
- In data warehousing literature(fig 2.4),
  - An n-D base cube is called a base cuboid.
  - The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid. The apex cuboid is typically denoted by **all**.

### 2.2.2 Multi-Dimensional Data

The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data-model is appropriate for on-line transaction processing. A data warehouse, however, requires a concise, subject oriented schema that facilitates on-line analysis. The most popular data model for a data warehouse is a multidimensional model. A multidimensional data model is typically organized around a central theme. This theme is represented by a fact table.

- The lattice of cuboids forms a data cube.

| Location | Product | | | |
|---|---|---|---|---|
| | Soda | Diet soda | Lime soda | Orange soda |
| California | 80 | 110 | 60 | 25 |
| Utah | 40 | 90 | 50 | 30 |
| Arizona | 70 | 55 | 60 | 35 |
| Washingtone | 75 | 85 | 45 | 45 |
| Colorado | 65 | 45 | 85 | 60 |

Table 2.2: 2-D sales database

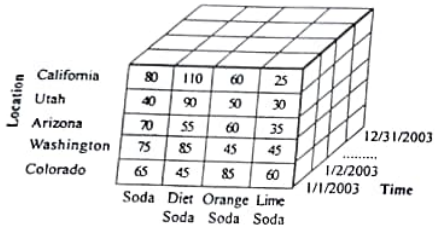



**Fig 2.3:** A 3-D data cube representation for sales data

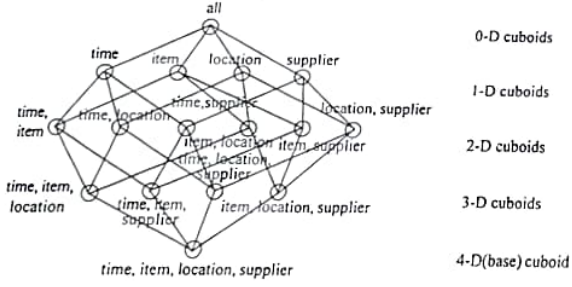



**Fig 2.4:** Lattice of cuboids for sales data

- *Facts are numerical measures. The fact table contains the names of the facts, or measures, as well as keys to each of the related dimension tables.*
- *Dimensional modeling is a technique*
  - *for structuring data around the business concepts.*
  - *for describing "measures" and "dimensions".*
  - *for describing business parameters that define a transaction.*

*For example, Analyst may want to view (fig 2.5) sales data (measure) by geography(locid), time(timeid), and by product(pid) (dimensions) given in Table 2.3.*



| pid | timeid | locid | sales |
|---|---|---|---|
| 11 | 1 | 1 | 25 |
| 11 | 2 | 1 | 8 |
| 11 | 3 | 1 | 15 |
| 12 | 1 | 1 | 30 |
| 12 | 2 | 1 | 20 |
| 12 | 3 | 1 | 50 |
| 13 | 1 | 1 | 8 |
| 13 | 2 | 1 | 10 |
| 13 | 3 | 1 | 10 |

**Table 2.3:** *sales* data by *geography*, by *time*, and by *product*

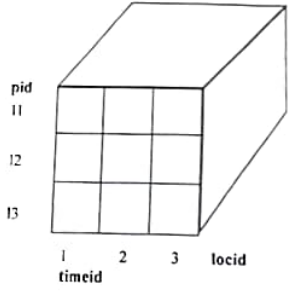



**Fig 2.5:** A 3-D data cube

## Why Multi Dimensional Databases?

- No single "best" data structure for all applications within an enterprise.
- The multidimensional database has matured into the database engine of choice for data analysis applications.
- Inherent ability to integrate and analyze large volumes of enterprise data.
- Offers a good conceptual fit with the way end-users *visualize* business data.
  - Most business people already think about their businesses in multidimensional terms.
  - Managers tend to ask questions about product sales in different markets over specific time periods.

## Contrasting Relational and Multi-Dimensional Models

**Example 2.1:** Table 2.4 and fig 2.6 give the representation of relation structure and multidimensional structure.

The Relational structure

| Model | Color | Sales Volume |
|---|---|---|
| Mini Van | Blue | 6 |
| Mini Van | Red | 5 |
| Mini Van | White | 4 |
| Sports Coupe | Blue | 3 |
| Sports Coupe | Red | 5 |
| Sports Coupe | White | 5 |
| Sedan | Blue | 4 |
| Sedan | Red | 3 |
| Sedan | Red | 2 |

**Table 2.4:** sales volume data

Multi dimensional Array Structure

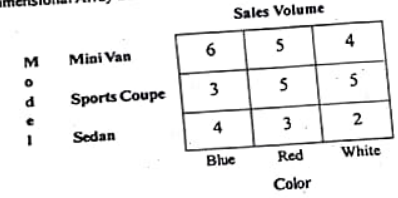
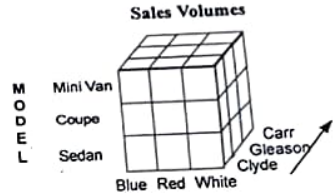



**Fig 2.6 :** A data cube of table 2.4

**Example 2.2:** In this example, another dimension is added into the sales volume data. Table 2.5 and fig 2.7 give the representation of relation structure and multidimensional structure of sales volume data for all dealership.





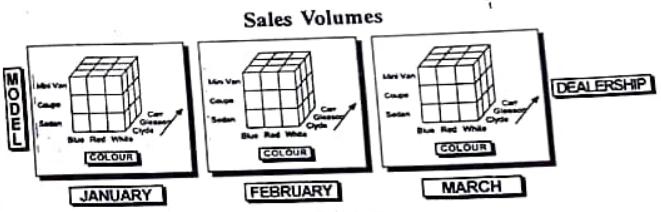




**Fig 2.8:** Sales Volumes For Dealership with different months

| Model | Color | Dealership | Sales Volume |
|---|---|---|---|
| Mini Van | Blue | Clyde | 6 |
| Mini Van | Blue | Gleason | 6 |
| Mini Van | Blue | Carr | 2 |
| Mini Van | Red | Clyde | 3 |
| Mini Van | Red | Gleason | 5 |
| Mini Van | Red | Carr | 5 |
| Mini Van | White | Clyde | 2 |
| Mini Van | White | Gleason | 4 |
| Mini Van | White | Carr | 3 |

| | | | |
|---|---|---|---|
| Sports Coupe | Blue | Clyde | 2 |
| Sports Coupe | Blue | Gleason | 3 |
| Sports Coupe | Blue | Carr | 2 |
| Sports Coupe | Red | Clyde | 7 |
| Sports Coupe | Red | Gleason | 5 |
| Sports Coupe | Red | Carr | 2 |
| Sports Coupe | White | Clyde | 4 |
| Sports Coupe | White | Gleason | 5 |
| Sports Coupe | White | Carr | 1 |
| Sedan | Blue | Clyde | 6 |
| Sedan | Blue | Gleason | 4 |
| Sedan | Blue | Carr | 2 |
| Sedan | Red | Clyde | 1 |
| Sedan | Red | Gleason | 3 |
| Sedan | Red | Carr | 4 |
| Sedan | White | Clyde | 2 |
| Sedan | White | Gleason | 2 |
| Sedan | White | Carr | 3 |

Table 2.5: The Relational structure of sales volume data all for dealership

### When is Multi Dimensional Databases appropriate?

- Our sales volume dataset has a great number of meaningful interrelationships.
- Interrelationships more meaningful than individual data elements themselves.
- The greater the number of inherent interrelationships between the elements of a dataset, the more likely it is that a study of those interrelationships will yield business information of value to the company.
- Highly interrelated dataset types be placed in a multidimensional data structure for greatest ease of access and analysis

### Benefits of Multi Dimensional Databases

- **Ease of Data Presentation and Navigation:** Obtaining the same views in a relational world requires the end user to either write complex SQL queries or use an SQL generator against the relational database to convert the table outputs into a more intuitive format.
- **Ease of Maintenance:** Because data is stored in the same way as it is viewed (i.e., according to its fundamental attributes), no additional overhead is required to translate user queries into requests for data.
- **Performance:** Multidimensional databases achieve performance levels that are difficult to match in a relational environment.

availability, scaling, complexity



### 2.2.3 Schemas for Multidimensional Databases

The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them. Such a data model is appropriate for on-line transaction processing. A data warehouse, however, requires a concise, subject oriented schema that facilitates on-line analysis. The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of

- **a star schema:** A fact table in the middle connected to a set of dimension tables.
- **a snowflake schema:** A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake.
- **a fact constellation schema:** Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation.

### The Design Process

- Method to design data structures based on the user and business needs.
- Must know what transaction, balance or event to model.

Before going to discuss about design process of star schema, snowflake schema, and fact constellation schema, let's look for construction of fact tables and dimension tables.

### Design Process of Fact Table

1. Model data at any possible level of detail.
2. Capturing finest level of detail.
3. Re-create aggregate summary data from detail data. Not possible re-create details from summaries.
4. Identify the measures that the user needs.
5. Record in fact table contains primary key which is made up of concatenation of foreign keys to dimension tables.
6. Facts or measures are uniquely identified by primary key.

### Design Process of Dimension Table

1. Provide meaning to each fact.
2. Identifying Dimension elements.
3. De normalization is the key feature.
4. Dimensions are organized into hierarchies
   - E.g., Time dimension: days → weeks → quarters
   - E.g., Product dimension: product → product line → brand
5. Dimensions have attributes.

### Design Process of a Star Schema

- Consists of a group of tables that describe the dimensions of the business.
- Arranged logically around a huge central table that contains all the accumulated facts and figures of the business.
- The smaller, outer tables are points of the star. The larger table the center from which the points radiate.

Fig 2.9 shows the general form of a star schema. Fig 2.10 and fig 2.11 give the Star schemas of a data warehouse for Sales Database.

It gives best performance when queries involve aggregation. The main disadvantage of Snowflake Schema is complicated in maintenance and metadata, explosion in the number of tables in the database.

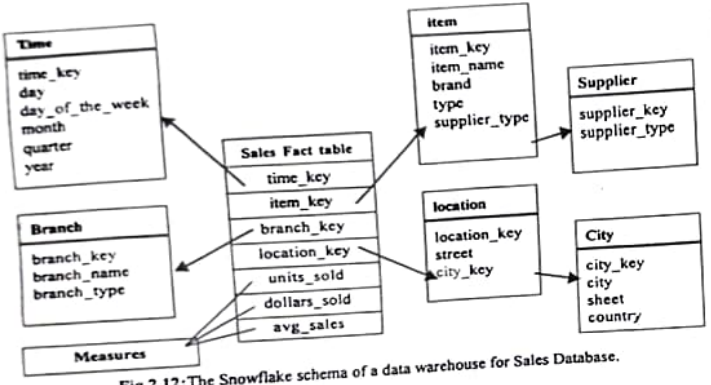



**Fig 2.12:** The Snowflake schema of a data warehouse for Sales Database.

### Fact constellation schema

Sophisticated applications may require multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation. Fig 2.13 shows the fact constellation schema. Fact Constellation is a good alternative to the Star, but when dimensions have very high cardinality, the sub-selects in the dimension tables can be a source of delay.

- The major advantage of Fact constellation schema is there is no need for the "Level" indicator in the dimension tables, since no aggregated data is stored with lower-level detail.
- The disadvantage is, dimension tables are still very large in some cases,
  - which can slow performance,
  - front-end must be able to detect existence of aggregate facts,
  - which requires more extensive metadata.



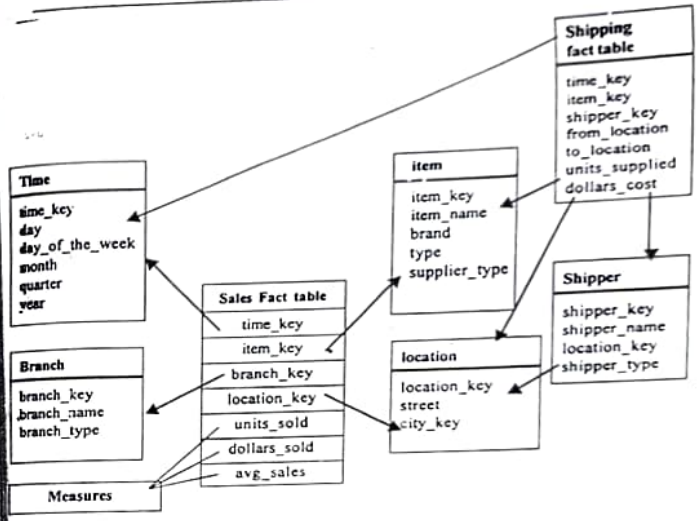



**Fig 2.13:** The Fact constellation schema of a data warehouse for Sales Database.

### Data Marts

- A data mart is defined as "A subset of a data warehouse for a single department or function". A data mart may have tens of gigabytes of data rather than hundreds of gigabytes for the entire enterprise.
- Focus on specific subject areas.
- Like a warehouse but scope is limited to specific subject area only.

For data marts, the star or snowflake schema are commonly used since both are geared towards modeling single subjects, although the star schema is more popular and efficient. Generally, data marts are two types. Those are

- Dependent Data Marts shown in fig 2.14 are derived from data warehouse, where as
- Independent Data Marts shown in fig 2.15 are not related data warehouse.

### Need for Data Marts

- End users will get much better performance querying from a data mart than from a data warehouse.
- End users will have a much easier time navigating through data marts.

### Can a data mart replace a data warehouse?

- Likely to fail over a time.
- Reasons for sourcing from a normalized warehouse are:
  - Avoids extraction repetition.
  - Ensures standard interpretation of enterprise data.
  - Provides repository that is far more flexible than de normalized structures in the high performance query or data mart layer.
- Data Warehouses Versus Data Marts are summarized in table 2.6.

| Property | Data Warehouse | Data Mart |
|---|---|---|
| Scope | Enterprise | Department |
| Subject | Multiple | Single-subject, LOB |
| Data Source | Many | Few |
| Size(typical) | 100 GB to>1 TB | <100GB |
| Implementation time | Months to years | Months |

**Table 2.6:** Data Warehouses Versus Data Marts

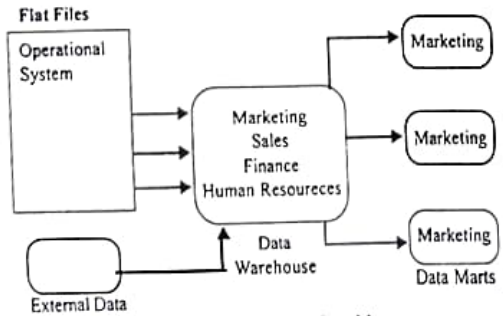



**Fig 2.14:** Dependent Data Marts



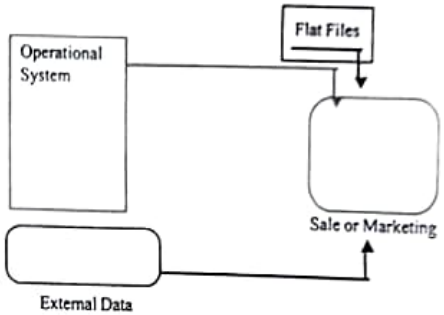



**Fig 2.15:** Independent Data Marts

### 2.2.4 Concept Hierarchies

A Concept hierarchy defines a sequence of mapping from a set of low-level concepts to higher-level, more general concepts. Consider a concept hierarchy for the dimension location given in fig 2.16. Each city, however, can be mapped to the country to which it belongs. For example, Vancouver can be mapped to Canada, and Frankfurt to Germany. The country can in turn be mapped to the region to which they belong, such as Europe or theNorth America. These mappings form a concept hierarchy for the dimension location, mapping a set of low-level concepts.(i.e. cities) to higher-level, more general concepts (i.e., countries).

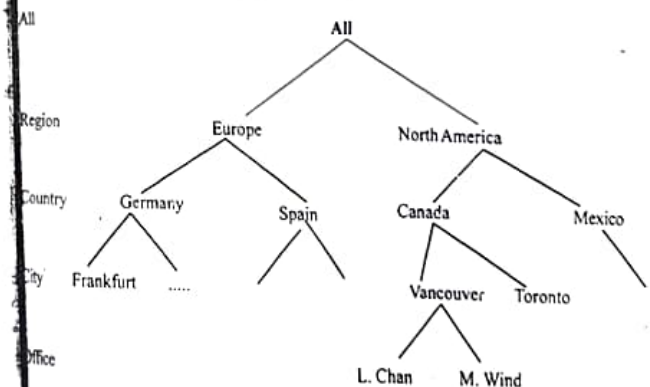



**Fig 2.16:** A Concept Hierarchy for Dimension (location)

Concept hierarchies may also be defined discretizing or grouping values for a given dimensio
or attribute, resulting in set-grouping hierarchy. A total or partial order can be defined among group
of values. An example of a set-grouping hierarchy is shown in fig 2.17 for the dimension price.

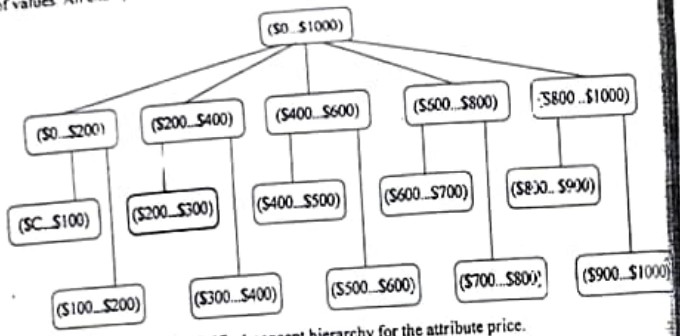



**Fig 2.17:** A concept hierarchy for the attribute price.

There may be more than one concept hierarchy for a given attribute or dimension, based o
the different user viewpoints. For instance, a user may prefer to organize price by defining ranges f
inexpensive, moderately_priced and expensive. Concept hierarchies may be provided manually
system users, domain experts, knowledge engineers, or automatically generated based on statistic
analysis of the data distribution.

### 2.2.5 OLAP Operation in the Multidimensional Data Model

As enabling as RDBMS have been for users, they were never intended to provide mo
powerful functions for data synthesis, analysis, and consolidation. In the multidimensional mode
data are organized into multi dimensions, and each dimension contains multiple levels of abstracti
defined by concept hierarchies. Their organization provides users with the flexibility to view the da
from different perspectives.

**OLAP makes**

- Data synthesis
- Analysis
- Consolidation quicker, smarter and easier

**OLAP Queries**

- Influenced by SQL and by spreadsheets.
- A common operation is to aggregate a measure over one or more dimensions. For example
  - Find total sales.
  - Find total sales for each city, or for each state.
  - Find top five products ranked by total sales.



**Why OLAP**

- Useful for rapid analysis of large quantities of data.
- Views Data from all angles.
- Basic element of the Decision support system.
- Standard reports and end-user applications inadequate.
- Structures data hierarchically the way managers think.
- Ad-hoc analysis, reliable system, information sharing, and user friendly.

**OLAP Characteristics**

- Quick response.
- User defined analysis.
- Multidimensional.

### Typical OLAP Operations

- **Roll-up:**
  - The roll-up operation(also called the drill-up operations by some vendors) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction.
  - Fig 2.18 shows the result of a roll-up operation performed on the central cube from store to area.
  - The roll-up operation shown aggregates the data by ascending the location hierarchy from the level of store.
  - 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.

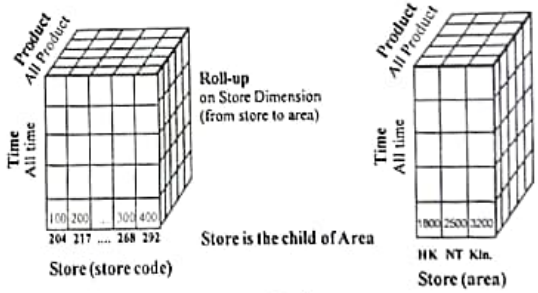



**Fig 2.18:** Graphical Description on Roll-up Example

- **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 by stepping down a concept hierarchy for a dimension or introducing additional dimensions.
  - Fig 2.19 shows the result of drill-down operation performed on the central cube by stepping down a concept hierarchy for time.
  - Drill-down occurs by descending the time hierarchy from the level of quarter to the more detailed level of month.
  - The resulting data cube details the total sales per month rather than summarized by quarter.
  - Since a drill-down adds more detail to the given data, it can also be performed by adding new dimensions to a cube.

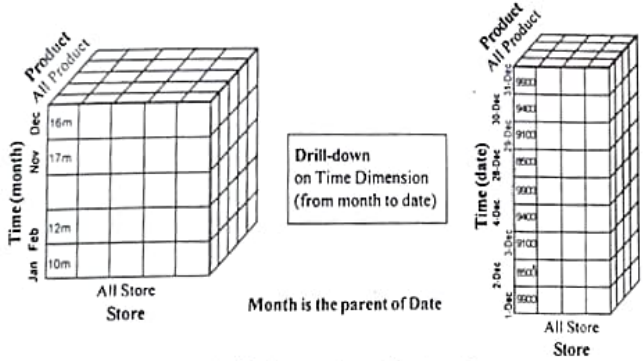



Fig 2.19: Graphical Description of Drill-down Example

- **Slice and dice:**
  - The slice operation performs a selection on one dimension of the given cube, resulting in a subcube.
  - Fig. 2.20 shows a slice operation where the sales data are selected from the central cube for the dimension time using the criterion storecode='292'.
  - The dice operation defines a subcube by performing a selection on two or more dimensions.
  - Fig. 2.21 shows a dice operation on the central cube based on the following selection criteria that involve three dimensions store, time and product.



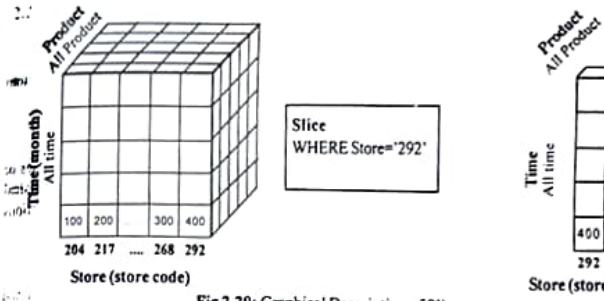



Fig 2.20: Graphical Description of Slice

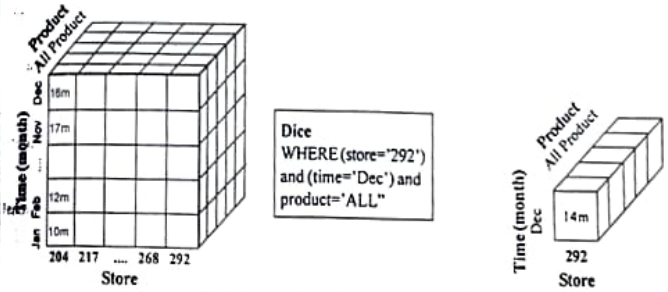



Fig 2.21: Graphical Description of Dice

- **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.
  - Fig. 2.22 shows a pivot operation where the item and location axes in a 2-D slice are rotated.

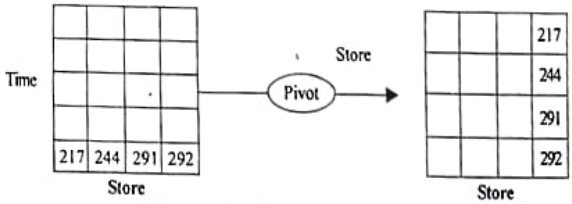



Fig 2.22: Graphical Description of pivot

- **Other OLAP operations:**
  - The **drill-across** executes queries involving (i.e., across) more than one fact table.
  - The **drill-through** operation make use of relational SQL facilities to drill through the bottom level of a data cube down to its back-end relational tables.

### 2.2.6 OLAP Server Architectures

OLAP servers present business users with multidimensional data from data warehouses or data marts, without concern regarding how or where the data are stored. However, the physical architecture and implementation of OLAP servers must consider data storage issues. Implementations of a warehouse server for OLAP processing include the following:

- **Relational OLAP (ROLAP):**
  - These are the intermediate servers that stand in between a relational back-end server and client front-end tools.
  - They use relational or extended-relational DBMS to store and manage warehouse data and OLAP middle ware to support missing pieces.
  - They include optimization of DBMS backend, implementation of aggregation navigation logic, and additional tools and services.
  - They have greater scalability than MOLAP.
  - The DSS server of Micro strategy and meta cube of Informix adopt the ROLAP approach.
- **Multidimensional OLAP (MOLAP):**
  - These servers supports multidimensional view of data through array-based multidimensional storage engines.
  - They map multidimensional views directly to data cube array structures.
  - Fast indexing to pre-computed summarized data.
  - MOLAP servers adopts a two-level storage representation to handle sparse and dense data sets.
    - The dense sub cubes are identified and stored as array structures.
    - the sparse sub cubes employ compression technology for efficient storage utilization.
- **Hybrid OLAP (HOLAP)**
  - The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster competition of MOLAP.
  - A HOLAP server may allow large volume of detailed data to be stored in a relational database, while aggregations are kept in a separate MOLAP store.
  - The Microsoft SQL server 7.0 OLAP services supports a hybrid OLAP servers.
- **Specialized SQL servers**
  - SQL servers that provide advance query language and query processing support for SQL queries over star and snowflake schemas in read-only environment.



### 2.2.7 A Starnet Query Model for Querying Multidimensional Databases

The querying of multidimensional databases can be based on a starnet model. A starnet model consists of radial lines emanating from a central point, where each line represents a concept hierarchy for a dimension. Each abstraction level in the hierarchy is called a footprint. These represents the granularities available for use by OLAP operations such as drill-down and roll-up. A starnet model for data warehouse is shown in fig 2.23.

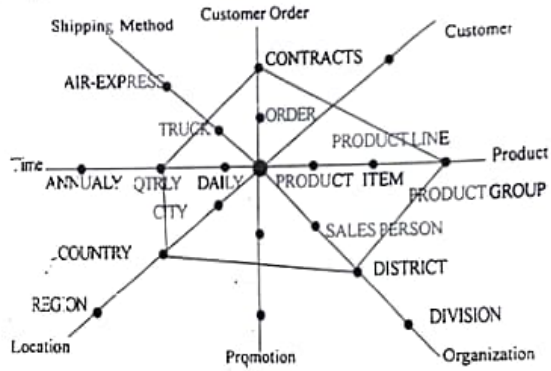



Fig 2.23: A Star-Net Query Model

## 2.3 Data Warehouse Architecture

A data warehouse is a read-only analytical database used for a decision support system operation. A data warehouse for decision support is often taking data from various platforms, databases, and files as source data. The use of advanced tools and specialized technologies may be necessary in the development of decision support systems, which affects tasks, deliverables, training, and project timelines. In this section, we discuss various phases in Design of Data Warehouses.

### 2.3.1 Design of Data Warehouses

To design an effective data warehouse one needs to understand and analyze business needs and construct a business analysis framework. The construction of a large and complex information system can be viewed as the construction of large and complex building, for which the owner, architect, and builder have different views. Four different views regarding the design of a data warehouse must be considered:

- **Top-down** view allows selection of the relevant information necessary for the data warehouse.
- **Data source** view exposes the information being captured, stored, and managed by operational systems.
- **Data warehouse** view consists of fact tables and dimension tables.

- **Business query view** sees the perspectives of data in the warehouse from the view of end-use
  Building and using a data warehouse is a complex task since it requires business skills, technolog
  skills, and program management skills.
- **Business skills** involve to understand
  - how such systems store and manage their data,
  - how to build **extractors** that transfer data from the operational systems to the data warehous
  - and how to build warehouse refresh software that keeps the data warehouse reasonably u
    to-date with the operational system's data.
- **Technology skills** are required
  - to understand how to make assessments from quantitative information and derive fact
    based on conclusions from historical information in the data warehouse
  - the ability to discover patterns and trends,
  - to present coherent managerial recommendations based on such analysis.
- **Program management skills** involve
  - the need to interface with many techniques, vendors, and end users in order to deliv
    results in a timely and cost-effective manner.

## The Process of Data Warehouse Design

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

- The **top-down** approach starts with the overall design and planning.
  - It is useful in cases where the technology is mature and well known, and where the busines
    problems that must be solved are clear and well understand.
- The **bottom-up** approach starts with experiments and prototypes.
  - This is useful in the early stage of business modeling and technology development.
  - It allows an organization to move forward at considerably less expense and to evaluate th
    benefits of the technology before making significant commitments.
- In the **combined approach**,
  - an organization can exploit the planned and strategic nature of the top-down approach
  - while retaining the rapid implementation and opportunistic application of the bottom-u
    approach.

  Phases of the Decision Support Life Cycle of data warehouse are

- **Planning** for a data warehouse is concerned with
  - Defining the project scope
  - Creating the project plan
  - Defining the necessary resources, both internal and external
  - Defining the tasks and deliverables
  - Defining timelines
  - Defining the final project deliverables
- **Gathering data requirements and Modeling** includes

**Gathering data requirements** includes

- How the user does business?
- How the user's performance is measured?
- What attributes does the user need?
- What are the business hierarchies?
- What data do users use now and what would they like to have?
- What levels of detail or summary do the users need?

**Modeling** includes

- A logical data model covering the scope of the development project including relationships, cardinality, attributes, and candidate keys.
- A Dimensional Business Model that diagrams the facts, dimensions, hierarchies, relationships and candidate keys for the scope of the development project.

- **Physical Database Design and Development** is involved in
  - Designing the database, including fact tables, relationship tables, and description (lookup) tables.
  - Denormalizing the data.
  - Identifying keys.
  - Creating indexing strategies.
  - Creating appropriate database objects.
- **Data Mapping and Transformation** is involved in
  - Defining the source systems.
  - Determining file layouts.
  - Developing written transformation specifications for sophisticated transformations.
  - Mapping source to target data.
  - Reviewing capacity plans.
- **Populating the data warehouse** phase is used for
  - Developing procedures to extract and move the data.
  - Developing procedures to load the data into the warehouse.
  - Developing programs or use data transformation tools to transform and integrate data.
  - Testing extract, transformation and load procedures

- **Automating Data Management Procedures** are used
  - Automating and scheduling the data load process.
  - Creating backup and recovery procedures.
  - Conducting a full test of all of the automated procedures.
- **Application Development phase** is involved in
  - Creating the Starter Set of Reports
  - Creating the starter set of predefined reports.
  - Developing core reports.
  - Testing reports.
  - Documenting applications.
  - Developing navigation paths.
- **Data Validation and Testing** is used for
  - Validating Data using the starter set of reports.
  - Validating Data using standard processes.
  - Iteratively changing the data.
- **Training**
  - to gain real business value from your warehouse development, users of all levels will need to be trained in:
    - The scope of the data in the warehouse.
    - The front end access tool and how it works.
    - The DSS application or starter set of reports - the capabilities and navigation paths.
    - Ongoing training/user assistance as the system evolves
- **Rollout** is the phase for
  - Installing the physical infrastructures for all users.
  - Developing the DSS application.
  - Creating procedures for adding new reports and expanding the DSS application.
  - Setting up procedures to backup the DSS application, not just the data warehouse.
  - Creating procedures for investigating and resolving data integrity related issues.

### 2.3.2 A Three-tier Data Warehouse Architecture

A general two-tier Data Warehouse Architecture is given in fig 2.24. Data warehouse often adopt a three tier architecture, as presented in the fig 2.25 and fig 2.26. A Multi tired Data Warehouse Architecture is given in fig 2.27.



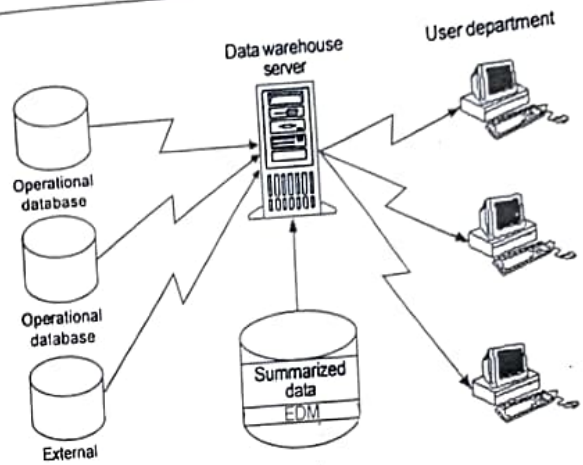



Fig 2.25: Three Tier Data Warehouse Architecture

- The bottom tier is a **warehouse database server** that is almost always a relational database systems.
  - Data from operational databases and external sources are extracted using application program interfaces known as **gateways**.
  - A gateway is supported by the underlying DBMS and allows client programs to generate SQL code to be executed at a server.
  - Examples of gateways include ODBC(Open Database Connection) and OLE-DB(Open Linking and Embedding) for databases by Microsoft and JDBC(Java Database Connection).
- The middle tier is an **OLAP server** that is typically implemented using either
  - a **relational OLAP (ROLL-UP) model**, that is, an extended relational DBMS that maps operations on multidimensional relational data to standard relational operations; or
  - a **multidimensional OLAP(MOLAP) model**, that is, a special-purpose server that directly implements multidimensional data band operations.
- The top tier is **client**, which contains query and reporting tools, analysis tools and/or data mining tool (e.g., trend analysis, prediction, and so on).

### Data Warehouse Development: A Recommended Approach

From the architecture point of view, there are three data warehouse models: the enterprise warehouse, the data mart, and the virtual warehouse.

- **An enterprise warehouse** collects of all the information about subjects spanning the entire organization.
  - It provides corporate-wide data integration, usually from one or more operational systems or external information providers, and is cross-functional in scope.
  - It typically contains detailed data as well as summarized data, and can range in the size from a few gigabyte to hundred gigabytes, terabytes, or beyond.

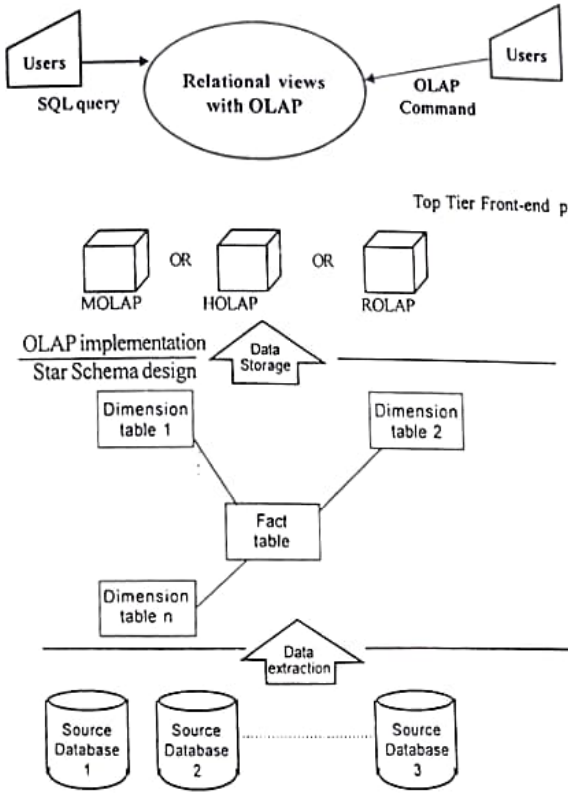



Fig 2.26: A detailed Three Tier Data Warehouse Architecture



  - It may be implemented on traditional mainframes, UNIX super servers, or parallel architecture platforms.
  - It requires extensive business modeling and may take years to design and build.
- A **data mart** contains a subset of corporate-wide data that is of value to a specific group of users. These are discussed in section 2.2.3.
- A **virtual warehouse** is a set of views over operational databases.
  - For efficient query processing, only sum of the possible summary view may be materialized.
  - A virtual warehouse is easy to build but requires excess capacity on operational database servers.

A recommended method for the development of data warehouse systems is to implement the warehouse in an incremental and evolutionary manner, as shown in fig 2.27.

- First, a high-level corporate data model is defined within a reasonably short period of time(such as one or two months) that provides a corporate wide consistent, integrated view of data among different subjects and potential usages.
- Second, independent data marts can be implemented in parallel enterprise warehouse based on the same corporate data model set as above.
- Third distributed data marts can be constructed to integrate different data marts via hub server.

Finally, a **multi-tier data warehouse**(fig 2.28) is constructed where the enterprise warehouse is the sole custodian of all warehouse data, which is then distributed to the various dependent data marts.

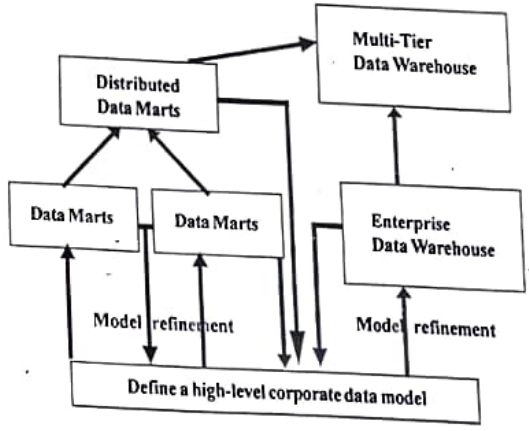



Fig 2.27: Data Warehouse Development

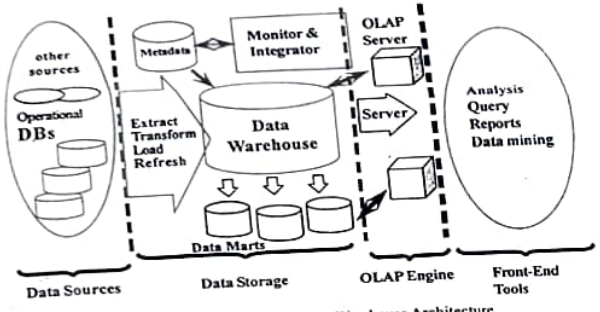



Fig 2.28: A Multi tired Data Warehouse Architecture

## 2.4 Data Warehouse Implementation

Data warehouse contain huge volumes of data. OLAP servers demand that decision support queries be answered in the order of seconds. Therefore, it is critical for data warehouse systems to support highly efficient cube computation techniques, access methods, and query processing techniques. In this section, the methods for the efficient implementation of data warehousing systems are discussed.

### 2.4.1 Efficient Computation of Data Cubes

At the core of multidimensional data analysis is the efficient computation of aggregation across many sets of dimensions. In SQL terms, these aggregation are referred to as **group-by's.**

#### The CUBE Operator

One approach to cube computation extends SQL so as to include a cube operator.

- It is Used to generalize all possible sub totals for all combinations of specified dimensions.
- It Generates also grand total like rollup operator.
- If there are k dimensions, we have 2^k possible SQL GROUP BY queries that can be generated through pivoting on a subset of dimensions.
- Example for CUBE operator is

```
select year, region, dept, sum(profit)
  from sales
  group by CUBE ( year, region, dept )
```



#### Precomputation

On-line analytical processing may need to access different cuboids for different queries. Therefore, it does seem like a good idea to compute all or at least sum of the cuboids in a data cube in advance.

- Precomputation leads to fast response time and avoids sum redundant computation.
- A major challenge related to this precomputation, however, is that the required storage space may explored if all of the cuboids in a data cube are precomputed, especially when the cube has many dimensions associated with multiple level hierarchies.

#### Materialized Views

Materialized Views are summarized as follows:

- Stored view
- Periodically refreshed with source data
- Usually contain summary data
- Fast query response for summary data
- Appropriate in query dominant environments

If there are cuboids, and these cuboids are large in size, a more reasonable option is partial materialization that is, to materialize only some of the possible cuboids that can be generated.

#### Partial Materialization: Selected Computation of Cuboids

There are three choices for data cube materialization given a base cuboid:

- Don't precompute any of the "non-base" cuboids **(no materialization)**. It leads to computing expensive multidimensional aggregates on the fly, which could be slow.
- Precompute all of the cuboids**(full materialization)**. It may require huge amounts of memory space in order to store all of the precomputed cuboids.
- Selectively compute a proper subset of the whole set of possibly cuboids**(partial materialization)**. It presents an interesting trade-of between storage space and response time.

The partial materialization of cuboids should consider three factors:

- identify the subset of cuboids to materialize,
- exploit the materialize cuboid during query processing, and
- efficiently update the materialize cuboids during load and refresh.

#### Cube Computation

In order to ensure fast on-line analytical processing, however, we may need to precompute all of the cuboids for a given data cube. Cuboids may be stored on secondary storage on accessed when necessary. Hence, it is necessary to explore efficient methods for computing all of the cuboids making up a data cube, that is, for materialization. These methods must take into consideration the limited amount of main memory available computation, as well as the time required for such computation. To simplify matters, we may exclude the cuboids generated by climbing up the existing hierarchies along each dimension. Efficient cube computation methods are

- ROLAP-based cubing algorithms.
- Array-based cubing algorithm.
- Bottom-up computation method.

### ROLAP-based cubing algorithms

- Sorting, hashing, and grouping operations are applied to the dimension attributes in order to reorder and cluster related tuples
- Grouping is performed on some subaggregates as a "partial grouping step"
- Aggregates may be computed from previously computed aggregates, rather than from the base fact table

### Array-based cubing algorithm

- It partitions arrays into chunks (a small subcube which fits in memory).
- It compressed sparse array addressing: (chunk_id, offset).
- It computes aggregates in "multiway" by visiting cube cells in the order which minimizes the # of times to visit each cell, and reduces memory access and storage cost. Fig 2.29 shows a 3-D data array containing a three dimensions A, B and C organized into 64 chunks.
- Limitation of the method is computing well only for a small number of dimensions.
- If there are a large number of dimensions, **"bottom-up computation"** and **iceberg cube computation methods** can be explored

## 2.4.2 Indexing OLAP Data

To facilitate efficient data accessing, most data warehouse systems support index structures and materialized views (using cuboids). In this section, we examine how to index OLAP data by bitmap indexing and join indexing.

- The **bitmap indexing** method is popular in OLAP products.
  - It allows quick searching in data cubes.
  - It is especially useful low-cardinality domains because comparison, join, and aggregation operations are then reduced to bit arithmetic, which substantially reduces the processing time.
  - It leads to significant reductions in space and I/O since a string of characters can be represented by single bit for higher-cardinality domains, the method can be adopted using compression.
  - The bitmap index is an alternative representation of the record-ID(RID) list.
  - In the bitmap index for a given attribute, there is a distinct bit vector, Bv, for each value v in the domain of the attribute.
  - If the domain of a given attribute consists of n values, then n bits are needed for each entry in the bitmap index(i. e., there are n bit vectors).
  - If the attribute has the value v for a given row in the data table, then the bit representing that value is set to 1 in the corresponding row of the bitmap index. All other bits for that row are set to 0. Fig 2.30 gives overview of bitmap indexing.



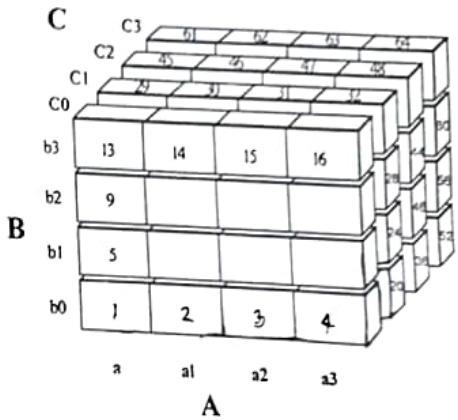



Fig 2.29: a 3-D data array containing a three dimensions A, B and C

**Base table**

| Cust | Region | Type |
|---|---|---|
| C1 | Asia | Retail |
| C2 | Europe | Dealer |
| C3 | Asia | Dealer |
| C4 | America | Retail |
| C5 | Europe | Dealer |

**Index on Region**

| RealD | Asia | Europe | America |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 2 | 0 | 1 | 0 |
| 3 | 1 | 0 | 0 |
| 4 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 |

**Index on Type**

| RecID | Retail | Dealer |
|---|---|---|
| 1 | 1 | 0 |
| 2 | 0 | 1 |
| 3 | 0 | 1 |
| 4 | 1 | 0 |
| 5 | 0 | 1 |

Fig 2.30: bitmap indexing for sales database

- The **join indexing** method gained popularity from its use in relational database query processing.
  - Traditional indexing maps the value in a given column to a list of rows having that value.
  - In contrast, join indexing registers the joinable rows of two relations from a relational database.
  - Join indexing is especially useful for maintaining the relationship between a foreign key and its matching primary keys, from the joinable relation. Fig 2.31 shows join indexing sales database.

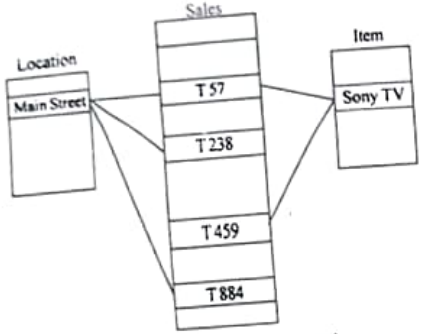



Fig 2.31: join indexing for sales database

To further speed up the query processing, the join indexing and bitmap indexing methods can be integrated to from bitmapped **join indices**.. Microsoft SQL server and Sybase IQ support bitmap indices . Oracle 8 uses bitmap and join indices.

### 2.4.3 Efficient Processing of OLAP Queries

The purpose of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes. Given materialize views, query processing should proceed as follows:

- **Determine which operations should be perfumed on the available cuboids:** This involves transforming any selection, projection, roll-up(group by), and drill-down operations in the specified in the corresponding SQL and/or OLAP operations. For example slicing and dicing a data cube may correspond to selection and/or projection operations on a materialize cuboid.
- **Determine to which materialize cuboid(s) the relevant operations should be applied:** This involves identifying all of the materialized cuboids that may potentially be used to answer the query, pruning the above set using knowledge "domains" relationship among the cuboids, estimating the costs of using the remaining materialized cuboids, and selecting the cuboid with the least cost.

### 2.4.4 Metadata Repository

- Metadata are data about data.
- When used in a data warehouse, metadata are the data that define warehouse objects.
- Metadata are created for the data names and definitions of the given warehouse.

A metadata repository should contain the following:

- A description of the **structure of the data warehouse**, which includes the warehouse schema, view, dimensions,hierarchies, and derived data definitions, as well as data mart locations and contents.



- **Operational metadata**, which include data lineage(history of migrated data and the sequence of transformations applied to it),currency of data (active, archived, or purged),and monitoring information (warehouse usage statistics , errors reports, and audit trails).
- **The algorithms used for summarization**, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports.
- **The mapping from the operational environment** to the data warehouse, which includes source databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control ).
- **Data related to system performance**, which include indices and profiles that improve data access and retrieval performance, in addition to rules for the timing and scheduling of refresh,update , and replication cycles.
- **Business metadata**,which include business terms and definitions, data ownership information, and charging policies

A data warehouse contains different levels of summarization, of which metadata is one type. Other types include current detailed data (which are almost always on disk), older detailed data(which are usually on tertiary storage), lightly summarized data and highly summarized data(which may or may not be physically housed). Metadata should be stored and managed persistently (i.e., on disk).

### 2.4.5 Data Warehouse Back-End Tools and Utilities

Data warehouse systems use back-end tools and utilities to populate and refresh their date. These tools and facilities include the following functions:

- **Data extraction**, which typically gathers data from multiple, heterogeneous, and external sources.
- **Data cleaning**, which detects errors in the data and rectifies them when possible.
- **Data transformation**, which converts data from legacy or host format to warehouse format.
- **Load**, which sorts, summarizes, consolidates, computers views,checks integrity,and builds indices and partitions.
- **Refresh**, which propagates the updates from the data sources to the warehouse.

## 2.5 Further Development of Data Cube Technology

In this section, we will study further developments of data cube technology.

### 2.5.1 Discovery-Driven Exploration of Data Cubes

As we have seen in this chapter, data can be summarized and stored in a multidimensional data cube of an OLAP system. A user or analyst can search for interesting patterns in the cube by specifying a number of OLAP operations, such as drill-down, roll-up, slice, and dice. While these tools are available to help the user explore the data ,the discovery process is not automated .

- **Hypothesis-driven exploration** is the user who, following his/her own intuition or hypothes tries to recognize exceptions or anomalies in the data. This hypothesis-driven exploration ha number of disadvantages.
  - The search space can be very large, making manual inspection of the data a daunting overwhelming task.
  - High-level aggregations may give no indication of anomalies at lower levels, making it e to overlook interesting patterns.
  - Even when looking at a subset of the cube, such as a slice, the user is typically faced w many data values to examine.
  - The sheer volume of data values alone makes it easy for users to miss exceptions in the d if using hypothesis-driven exploration.
- **Discovery-driven exploration** is an alternative approach in which precomputed measu indicating data exceptions are used to guide the user in the data analysis process, at all level aggregation.
  - an exception is a data cube cell value that is significantly different from the value anticipa based on statistical model.
  - The model considers variations and patterns in the measure value across all of the dimensio to which a cell belongs.
  - For example, if the analysis of the item-sales data reveals an increase in sales in Decembe comparision to all other months, this may seem like an exception in the time dimensi However, it is not an exception if the item dimension is considered, since there is sim increase in sales for other items during December.
  - The model considers exceptions hidden at all aggregated group-by's of a data cube.
  - Visual cues such as background color are used to reflect the degree of exception of e cell, based on the precomputed exception indicators.
  - The computation of exception indicators can be overlapped with cube construction, so t the overall construction of data cubes for discovery-driven exploration is efficient.
  - Three measures are used as exception indicators to help identify data anomalies.
  - These measures indicate the degree of surprise that the quantity in a cell holds, with resp to its expected value.
  - The measure are computed and associated with every cell, for all levels of aggregati They are
    - SelfExp: This indicates the degree of surprise of the cell value, relative to other cell the same level of aggregation.
    - InExp: This indicates the degree of surprise somewhere beneath the cell, if we were drill-down from it.
    - PathExp: : This indicates the degree of surprise for each drill-down path from the ce



### 2.5.2 Complex Aggregation at Multiple Granularities: Multifeature Cubes

Data cubes facilitate the answering of data mining queries as they allow the computation of aggregate data at multiple levels of granularity.

- Multi-feature cubes compute complex queries involving multiple dependent aggregates at multiple granularities.
- Grouping by all subsets of {item, region, month}, find the maximum price in 1997 for each group, and the total sales among all maximum price tuples is queried as

```
select item, region, month, max(price), sum(R.sales)
from purchases
where year = 1997
cube by item, region, month: R
such that R.price = max(price)
```

The tuples representing purchases in 1997 are first selected. The **cube by** clause computes aggregates (or group-by's) for all possible combinations of the attributes item, region, and month. It is an n-dimensional generalization of the **group by** clause are the **grouping attributes**. Tuples with the same value son all grouping attributes for one group. Let the groups be $g_1, \ldots, g_r$. For each group of the tuples $g_i$ the maximum price max $g_i$ among the tuples forming the group is computed. The variable R is a **grouping variable**, ranging over all tuples in group $g_i$ ( as specified in the **such that** case). The sum of sales of the tuples in $g_i$ that R ranges over is computed and returned with the values of the grouping attributes of $g_i$. The resulting cube is **mutifeature cube** in that it supports complex data mining queries for which multiple dependent aggregates are computed at a variety of granularities.

## 2.6 From Data Warehousing to Data Mining

In this section, we study the usage of data warehousing for information processing, analytical process, and data mining. We also introduce on-line analytical mining (OLAM), a powerful paradigm that integrates OLAP with data mining technology.

### 2.6.1 Data Warehouse Usage

Data warehouses and data marts are used in a wide range of applications. Business executives in almost every industry use the data collected, integrated, preprocessed and stored in data warehouses and data marts to perform data analysis and make strategic decisions. In many firms, data warehouses are used as an integral part of a plane-execute-asses "closed-loop" feedback system for enterprise management. Data warehouses are used extensively in banking and financial services, consumer goods and retails distribution sectors and controlled manufacturing, such as demand-based production. There are three kinds of data warehouse applications: information processing, analytical processing and data mining:

Information processing

- supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs.

- Analytical processing
  - multidimensional analysis of data warehouse data.
  - supports basic OLAP operations, slice-dice, drilling, pivoting.
- Data mining
  - knowledge discovery from hidden patterns.
  - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

### 2.6.2 From On-Line Analytical Processing to On-Line Analytical Mining

In the field of data mining, substantial research has been performed for data mining at various platforms, including transaction databases, relational databases, spatial databases, text databases, time-series databases, flat files, data warehouses, and so on.

Among many different paradigms and architectures of data mining systems, on-line analytical mining (OLAM) (also called OLAP mining), which integrates on-line analytical processing (OLAP) with data mining and mining knowledge in multidimensional databases, is particularly important for the following reasons:

- **High quality of data in data warehouses:**
  - Most data mining tools need to work on integrated, consistent, and cleaned data, which requires costly data cleaning, data transformation, and data integration as preprocessing steps.
  - A data warehouse constructed by such preprocessing serves as a valuable source of high quality data for OLAP as well as for data mining.
- **Available information processing infrastructure surrounding data warehouses:**
  - Comprehensive information processing and data analysis infrastructures have been constructed surrounding data warehouses, which include accessing, integration consideration, and transformation of multiple heterogeneous databases, ODBC/OLE DB connections, Web-accessing and service facilities, and reporting and OLAP analysis tools.
- **OLAP-based exploratory data analysis:**
  - Effective data mining needs exploratory data analysis.
  - On-line analytical mining provides facilities for data mining on different subsets of data and at different levels of abstraction, by drilling, pivoting, filtering, dicing and slicing on a data cube and on some intermediate data mining results.
- **On-line selection of data mining functions:**
  - By integrating OLAP with multiple data mining functions, on-line analytical mining provides users with the flexibility to select desired data mining functions and swap data mining tasks dynamically.



### Architecture for On-Line Analytical Mining

- An OLAM server performs analytical mining in data cubes in a similar manner as an OLAP server performs on-line analytical processing.
- An integrated OLAM and OLAP architecture is shown in fig. 2.32, where
  - the OLAM and OLAP servers both accept user on-line queries via an graphical user interface API and
  - work with the data cube in the data analysis via a cube API.
- A metadata directory is used to guide the access of the data cube.
- The data cube can be constructed by accessing and/or integrating multiple databases via an MDDB API and/or by filtering a data warehouse via a database API that may support OLE DB or ODBC connections.
- Since an OLAM server may perform multiple data mining tasks, such as concept description, association, classification, prediction, clustering, time-series analysis, and so on, it usually consists of multiple integrated data mining modules and is more sophisticated than an OLAP server.

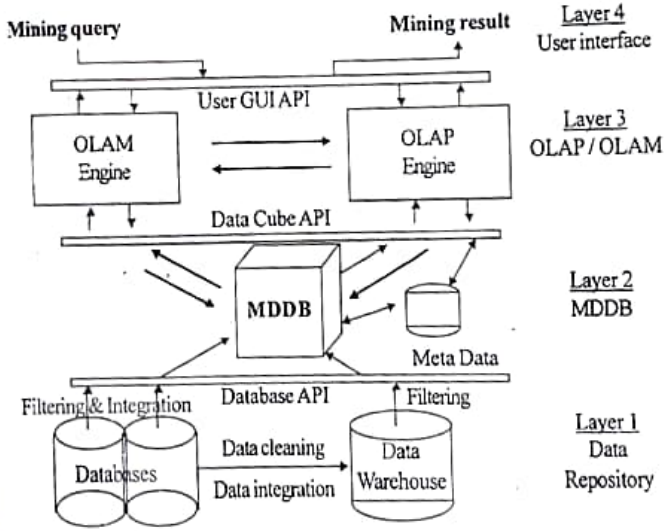



Fig 2.32: An integrated OLAM and OLAP architecture.

## Summary

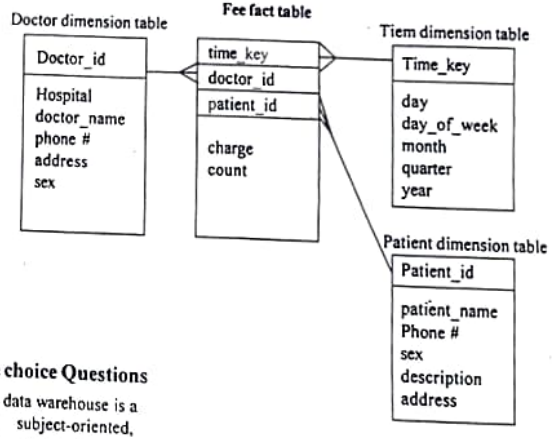
- Data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management's decision-making process.
- A multi-dimensional model of a data warehouse adopts Star schema, snowflake schema, and fact constellations. The core of multi-dimensional model is a data cube consists of dimensions & measures.
- OLAP operations are drilling, rolling, slicing, dicing and pivoting.
- OLAP servers are ROLAP, MOLAP, HOLAP.
- Efficient computation of data cubes perform partial vs. full vs. no materialization, multiway array aggregation, and Bitmap index and join index implementations.
- Further development of data cube technology gives discovery-drive, multi-feature cubes, and from OLAP to OLAM (on-line analytical mining).

## Excercise

1. What is a Data Warehouse?
2. Define Subject-oriented.
3. Define Integration.
4. Define Time-variant
5. Define Nonvolatile.
6. What is data warehousing?
7. Distinguish between OLTP vs. OLAP.
8. Why Separate Data Warehouse?
9. Describe A Multidimensional Data Model.
10. Compare and Contrasting Relational and Multi-Dimensional Models.
11. What are the Benefits of Multi Dimensional Databases?
12. When is Multi Dimensional Databases appropriate?
13. Discuss Design Process of Fact Table
14. Discuss Design Process of Dimension Table.
15. What is the Need for Data Marts?
16. Can a data mart replace a data warehouse?
17. Explain OLAP Operation in the Multidimensional Data Model.
18. Describe OLAP Server Architectures.



19. Explain the Process of Data Warehouse Design.
20. Describe a Three-tier Data Warehouse Architecture.
21. Explain the CUBE Operator
22. Discuss three methods of implementing an online analytical processing command. Give an example of using one of them with a given Star schema.
23. Suppose that a data warehouse consists of the three dimensions *time*, *doctor*, and *patent*, and the two measures *count* and *charge*, where charge is the fee that a doctor charges a patient for a visit. Starting with the base *cuboid* [*day, doctor, patient*], provide a MDX (Multidimensional Expression) query to list the total fee collected by each doctor in 2000?




## Multiple choice Questions

1. A data warehouse is a
   a) subject-oriented,
   b) nonvolatile
   c) both
2. data warehouse requires
   a. data integration
   b. association rules
   c. none of the above
3. OLTP system is
   a. customer-oriented
   b. market-oriented
   c. both

4. OLAP system is
   a. customer-oriented
   b. market-oriented
   c. both

5. Measures are in
   a. fact tables
   b. dimension tables
   c. both

6. Dimensional modeling is a technique
   a. for structuring data around the business concepts
   b. for describing "measures" and "dimensions"
   c. both

7. A subset of a data warehouse for a single department or function is defined as
   a. a data cube
   b. a data marf
   c. None of the above

8. Mapping from a set of low-level concepts to higher-level is called as
   a. a concept hierarchy
   b. a data cube
   c. a data mart

9. OLAP makes
   a. data synthesis
   b. analysis
   c. both

10. Roll-up is
    a. OLAP operation
    b. OLTP operation
    c. none of the above

11. Slice operation performs
    a. selection on one dimension of the given cube
    b. selection on two or more dimensions of the given cube
    c. rotation

12. A logical data model includes
    a. facts
    b. dimensions
    c. none of the above



13. Warehouse database server
    a. bottom tier
    b. middle tier
    c. top tier

14. Cube computation methods are
    a. ROLAP-based cubing algorithms
    b. Array-based cubing algorithm
    c. both

15. Data extraction means
    a. gathers data
    b. detects errors
    c. converts data

**Answers**

1.c 2.a 3.a 4.b 5.b 6.b 7.b 8.a 9.b 10.a 11.a
12.a 13.a 14.a 15.a

**References**


- S.. Agarwal, R. Agrawal, P. M. Deshpande, A. Gupta, J. F. Naughton, R. Ramakrishnan, and S. Sarawagi. On the computation of multidimensional aggregates. In Proc. 1996 Int. Conf. Very Large Data Bases, 506-521, Bombay, India, Sept. 1996.
- D. Agrawal, A. E. Abbadi, A. Singh, and T. Yurek. Efficient view maintenance in data warehouses. In Proc. 1997 ACM-SIGMOD Int. Conf. Management of Data, 417-427, Tucson, Arizona, May 1997.
- R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proc. 1998 ACM-SIGMOD Int. Conf. Management of Data, 94-105, Seattle, Washington, June 1998.
- R. Agrawal, A. Gupta, and S. Sarawagi. Modeling multidimensional databases. In Proc. 1997 Int. Conf. Data Engineering, 232-243, Birmingham, England, April 1997.
- K. Beyer and R. Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg CUBEs. In Proc. 1999 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'99), 359-370, Philadelphia, PA, June 1999.
- S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. ACM SIGMOD Record, 26:65-74, 1997.
- OLAP council. MDAPI specification version 2.0. In http://www.olapcouncil.org/research/apily.htm, 1998.