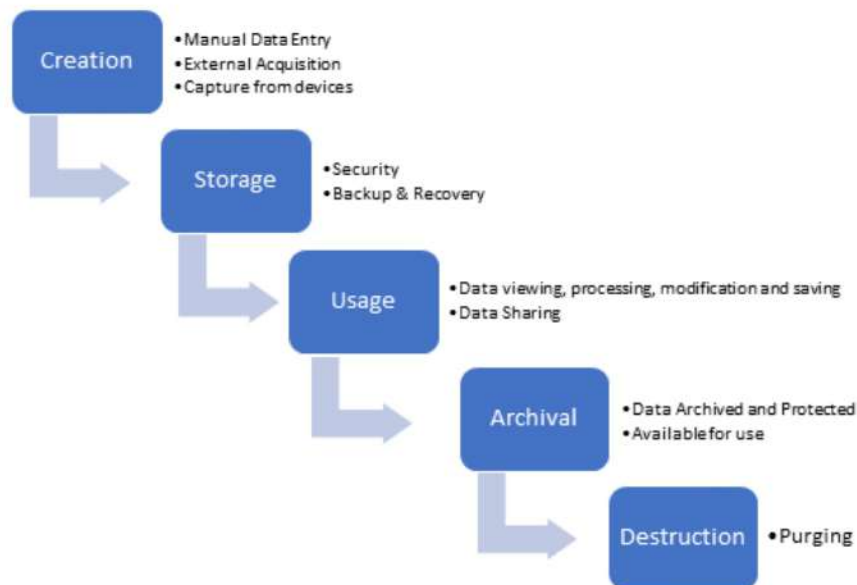


## Unit-1

### Introduction to Data Warehousing

#### Life Cycle of Data

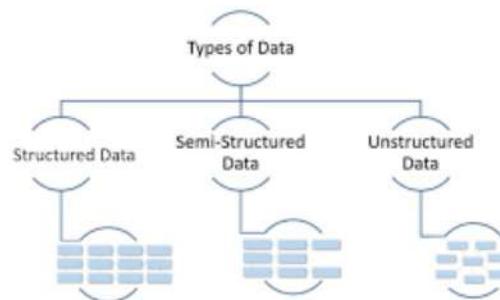
Data Life Cycle is a process that helps organizations to manage the flow of data throughout its lifecycle – from initial creation through to destruction. While there are many interpretations as to the various phases of a typical data lifecycle, they can be summarized as follows:



1. **Data Creation:** The first phase of the data lifecycle is the creation/capture of data. This data can be in many forms e.g. PDF, image, Word document, SQL database data. Data is typically created by an organisation in one of 3 ways:
  - **Data Acquisition:** acquiring already existing data which has been produced outside the organization
  - **Data Entry:** manual entry of new data by personnel within the organization
  - **Data Capture:** capture of data generated by devices used in various processes in the organisation
2. **Storage:** Once data has been created within the organisation, it needs to be stored and protected, with the appropriate level of security applied. A robust backup and recovery process should also be implemented to ensure retention of data during the lifecycle.
3. **Usage:** During the usage phase of the data lifecycle, data is used to support activities in the organisation. Data can be viewed, processed, modified and saved. An audit trail should be maintained for all critical data to ensure that all modifications to data are fully traceable. Data may also be made available to share with others outside the organisation.
4. **Archival:** Data Archival is the copying of data to an environment where it is stored in case it is needed again in an active production environment, and the removal of this data from all active production environments.
5. **Destruction:** Data destruction or purging is the removal of every copy of a data item from an organization. It is typically done from an archive storage location. If we want to save all data forever, it's not feasible. Storage cost and compliance issues create pressure to destroy data no longer need.

## Types of Data

Data is a set of facts such as descriptions, observations, and numbers used in decision making. We can classify data as structured, unstructured, or semi-structured data.



1. **Structured Data:** Structured data is generally tabular data that is represented by columns and rows in a database. Databases that hold tables in this form are called *relational databases*. In structured data, all row in a table has the same set of columns. SQL (Structured Query Language) programming language used for structured data.
2. **Semi-Structured Data:** Semi-structured data is information that doesn't consist of structured data (relational database) but still has some structure to it. Email messages are a good example. While the actual content is unstructured, it does contain structured data such as name and email address of sender and recipient, time sent, etc.
3. **Unstructured Data:** Unstructured data is information that either does not organize in a pre-defined manner or not have a pre-defined data model. Videos, audio, and binary data files might not have a specific structure. They're assigned to as unstructured data.

## Data Warehouse

A data warehouse is a repository of information collected from multiple heterogeneous sources and placed in a single site in order to facilitate management decision making.

The process of constructing and using data warehouses is known as **Data warehousing**.

Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.

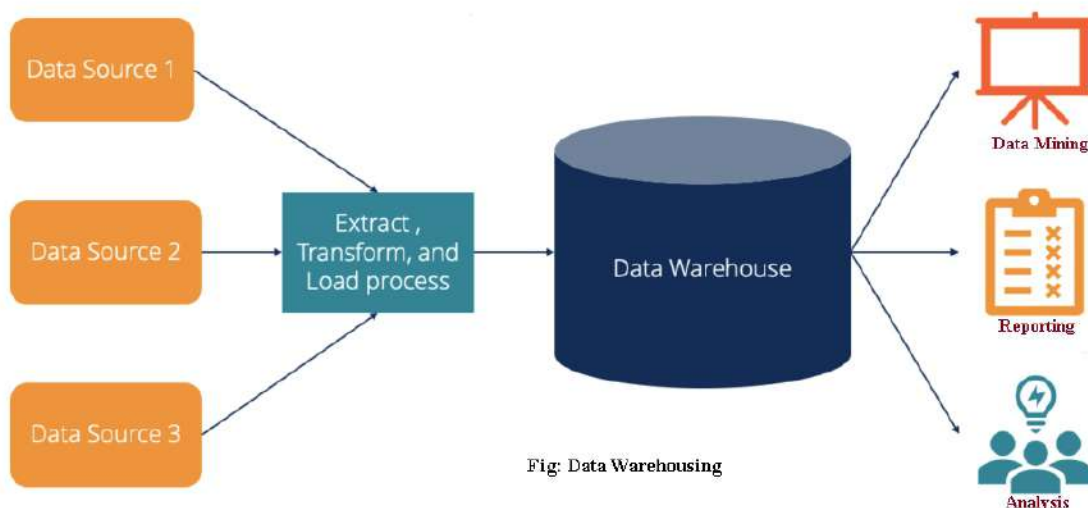


Fig: Data Warehousing

### **Data Warehouse Features / Nature of Data Warehouse**

The key features of data warehouse are:

- **Subject Oriented:** A data warehouse is subject oriented because it provides information around a subject rather than the organization's ongoing operations. These subjects can be product, customers, suppliers, sales, revenue, etc.
- **Integrated:** A data warehouse is constructed by integrating data from heterogeneous sources such as relational databases, flat files, etc. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.
- **Time Variant:** The data collected in a data warehouse is identified with a particular time period. The data in a data warehouse provides information from the historical perspective (e.g., the past 5–10 years).
- **Non-volatile:** Non-volatile means the previous data is not erased when new data is added to it. A data warehouse is kept separate from the operational database and therefore frequent changes in operational database is not reflected in the data warehouse.

### **Operational Database System vs Data Warehouse**

**Database system:** Database System is used in traditional way of storing and retrieving data. The major task of database system is to perform query processing. These systems are generally referred as online transaction processing (OLTP) system. These systems are used day to day operations of an organization.

**Data Warehouse:** Data Warehouse is the place where huge amount of data is stored. It is meant for users or knowledge workers in the role of data analysis and decision making. These systems are supposed to organize and present data in different format and different forms in order to serve the need of the specific user for specific purpose. These systems are referred as online analytical processing (OLAP).

<b>Operational Database System (OLTP System)</b>	<b>Data Warehouse (OLAP System)</b>
Operational system are generally designed to support high-volume transaction processing.	Data warehousing systems are generally designed to support high volume analytical processing. (i.e. OLAP)
It is used for day-to-day operations.	It is used for long-term informational requirements and decision support.
Operational data are the original sources of the data.	Data comes from various OLTP Databases.
In operational system data is stored with a functional or process orientation.	In data warehousing systems data is stored with a subject orientation.
It provides detailed and flat relational view of data.	It provides summarized and multidimensional view of data.
It focuses on "Data In".	It focuses on Information out.
The tables and joins are complex since they are normalized (for RDMS). This is done to reduce redundant data and to save storage space.	The tables and joins are simple since they are de-normalized. This is done to reduce the response time for analytical queries.

Entity-Relationship modelling techniques are used for RDMS database design.	Data-Modeling techniques are used for the Data Warehouse design.
Performance is low for analysis queries.	High performance for analytical queries.
Data within operational systems are generally updated regularly.	Data within a data warehouse is non-volatile, meaning when new data is added old data is not erased so rarely updates.
Data volumes are less and historical data is generally not maintained.	It involves large data volumes and historical data.
Simple queries are capable of fetching the data.	Complex queries are required to fetch data.
Processing speed is fast.	Processing speed is slow because of large size.
The common users are clerk, DBA, database professional.	The common users are knowledge worker ( e.g. manager, executive, analyst)

### **Functions of Data Warehousing**

- **Data Extraction:** Involves gathering data from multiple heterogeneous sources.
- **Data Cleaning:** Involves finding and correcting the errors in data.
- **Data Transformation:** Involves converting the data from legacy format to warehouse format.
- **Data Loading:** Involves sorting, summarizing, consolidating, checking integrity, and building indices and partitions.
- **Refreshing:** Involves updating from data sources to warehouse.

### **Multidimensional Data Model**

Data warehouses and OLAP tools are based on a multidimensional data model which views data in the form of a *data cube*. A data cube enables 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. *Sales* data warehouse may keep records of the store's sales with respect to the dimensions *time*, *item*, *branch*, and *location*. Each dimension may have a table associated with it, called a **dimension table**. This table further describes the dimensions. For example, a dimension table for *item* may contain the attributes *item name*, *brand*, and *type*.
- **Facts** are numerical measures which are used to analyse the relationship between dimensions. Examples of facts for a sales data warehouse include *dollars\_sold* (sales amount in dollars), *units\_sold* (number of units sold), and *amount\_budgeted*. The **fact table** contains the names of the *facts*, or measures, as well as keys to each of the related dimension tables.

#### ***From Tables and Spreadsheets to Data Cubes:***

Figure below represents dataset as 2-D table (i.e in rows and columns). It shows sales data, according to the dimension time, item, and location.

	Location="Chennai"				Location="Kolkata"				Location="Mumbai"				Location="Delhi"			
	item				item				item				item			
Time	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit
Q1	340	360	20	10	435	460	20	15	390	385	20	39	260	508	15	60
Q2	490	490	16	50	389	385	45	35	463	366	25	48	390	256	20	90
Q3	680	583	46	43	684	490	39	48	568	594	36	39	436	396	50	40
Q4	535	694	39	38	335	365	83	35	338	484	48	80	528	483	35	50

Fig: Statistical Table: Two dimensional representation

The above Table can be represented in multi-dimensional view using data cube as follows:

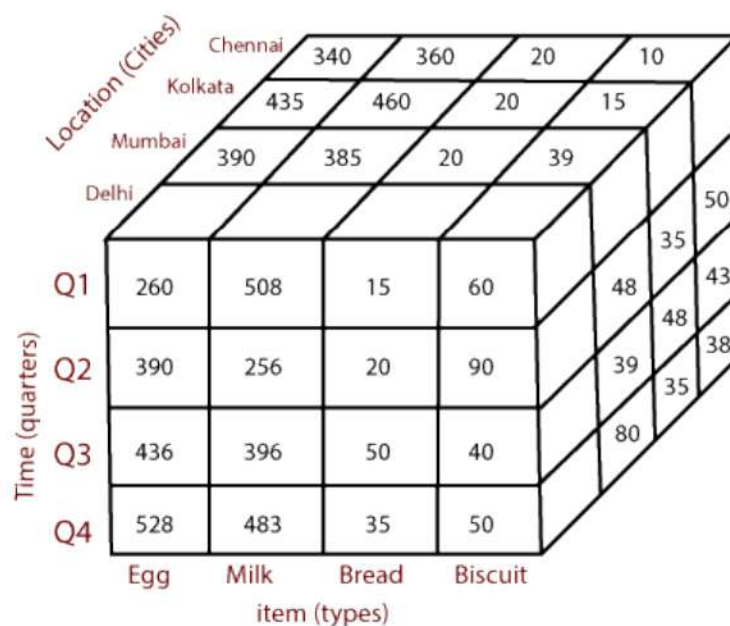


Figure: A 3-D representation of the data according to the dimensions time, item, and location.

### **Online Analytical Processing (OLAP)**

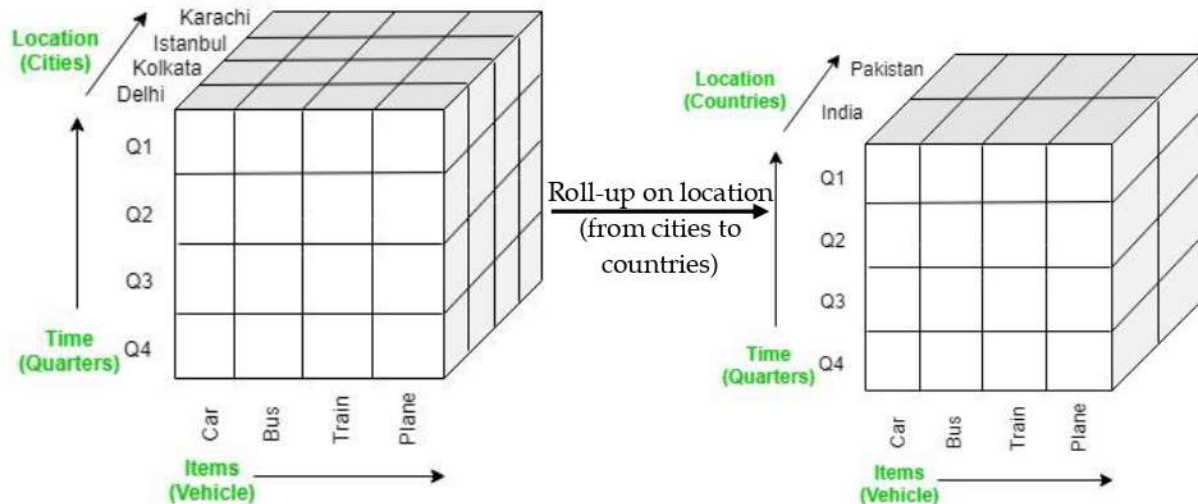
- Online Analytical Processing Server (OLAP) is based on the multidimensional data model.
- It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information.
- OLAP facilitates users to extract and present multidimensional data from different view.
- OLAP provides a user-friendly environment for interactive data analysis.

### OLAP Operations

There are five basic analytical operations that can be performed on an OLAP cube:

1. **Roll-up (Drill-up):** The roll-up operation performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction. When roll up operation is performed one or more dimension is removed from the given cube.

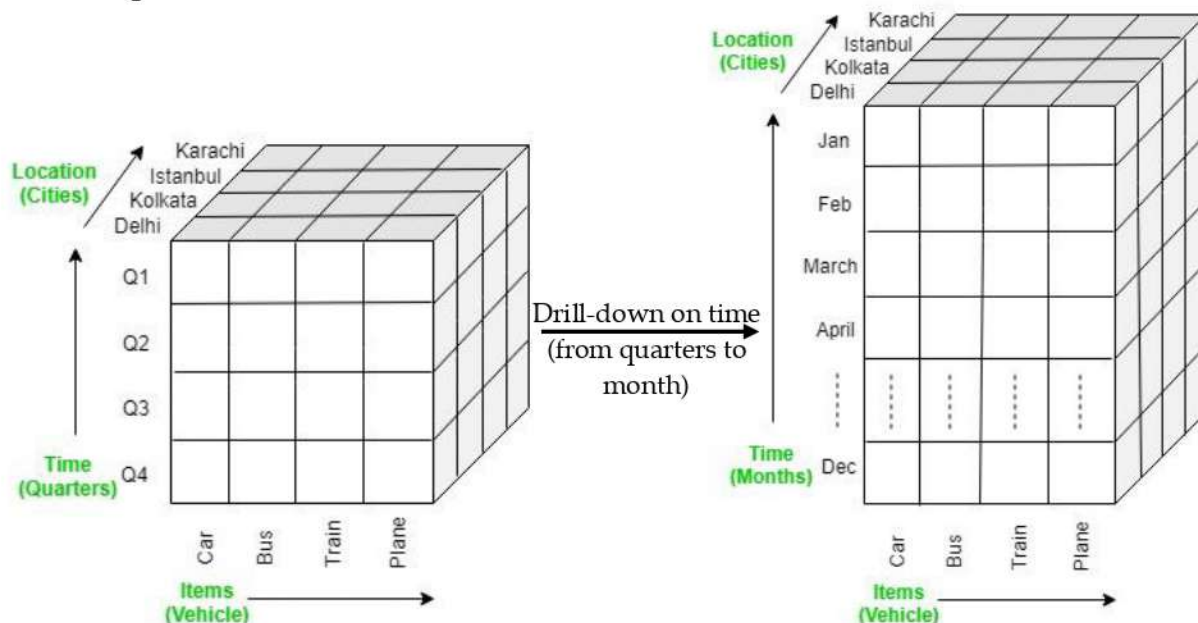
*Example:*



In this example, the roll-up operation is performed by climbing up in the concept hierarchy of *Location* dimension (City -> Country).

2. **Drill-down (Roll-down):** Drill-down is the reverse of roll-up. In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by either stepping down a concept hierarchy for a dimension or introducing additional dimensions. When drill-down is performed, one or more dimensions from the data cube are added.

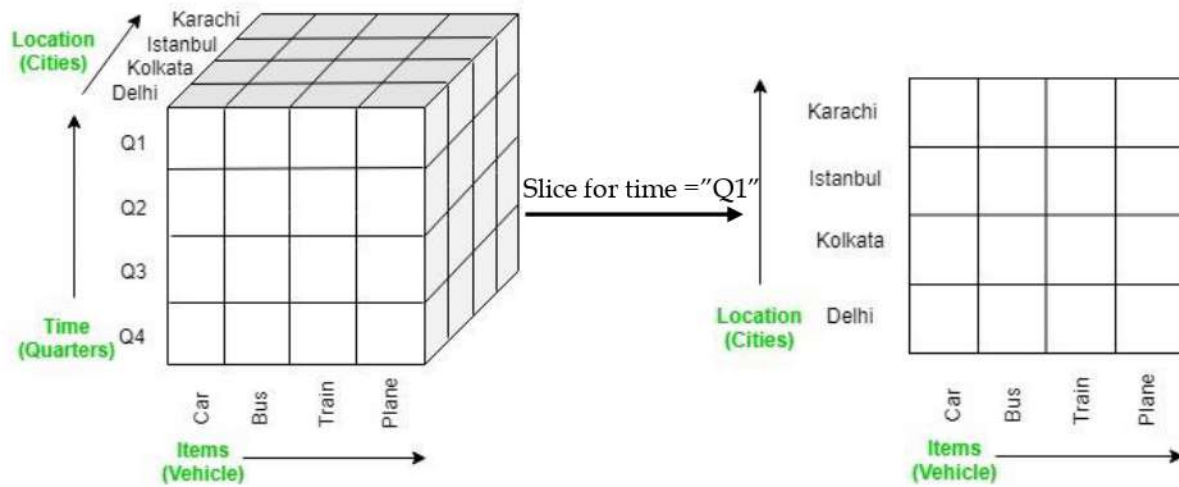
*Example:*



In this example, the drill down operation is performed by moving down in the concept hierarchy of *Time* dimension (Quarter -> Month).

3. **Slice:** The slice operation selects one particular dimension from a given cube and provides a new sub-cube. It reduces the dimensionality of the cubes.

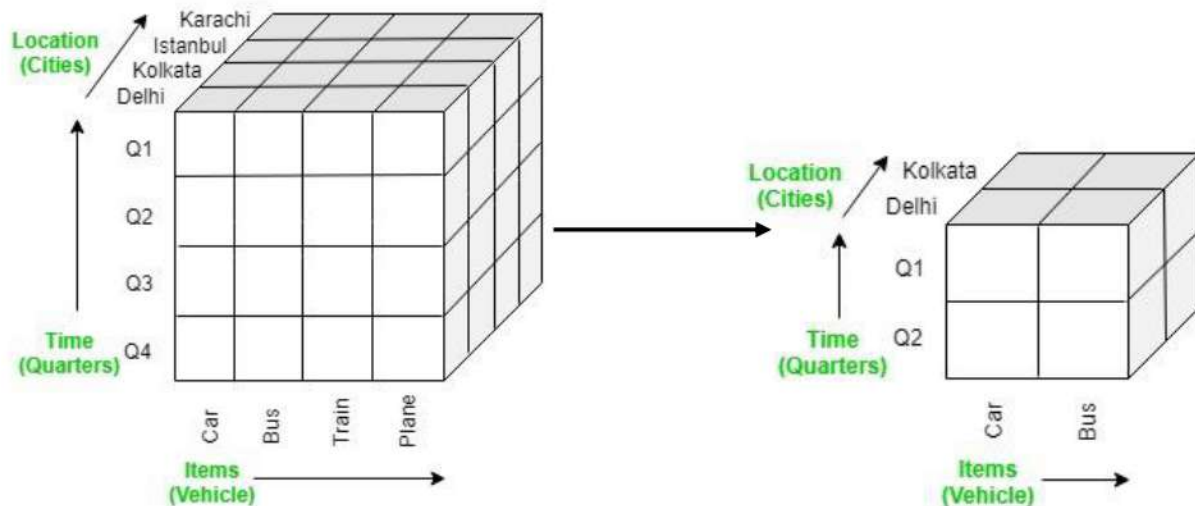
**Example:**



In this example, slice is performed for the dimension "time" using the criterion time = "Q1".

4. **Dice:** Dice selects two or more dimensions from a given cube and provides a new sub-cube.

**Example:**



In this example, a sub-cube is selected by selecting following dimensions with criteria:

Location = "Delhi" or "Kolkata"

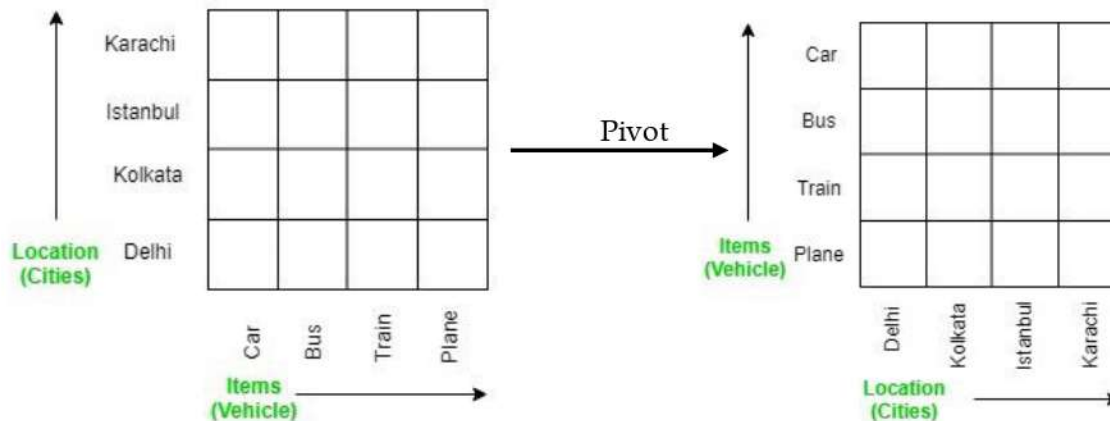
Time = "Q1" or "Q2"

Item = "Car" or "Bus"

5. **Pivot:** The pivot operation is also known as rotation. It rotates the data axis to view the data from different perspectives.

**Example:**

In the sub-cube obtained after the slice operation, performing pivot operation gives a new view of it.



### OLAP Servers

There are three main types of OLAP servers:

#### 1. Relational OLAP (ROLAP)

Relational OLAP (ROLAP) servers are placed between relational back-end server and client front-end tools. To store and manage the warehouse data, the relational OLAP uses relational or extended-relational DBMS. ROLAP servers include optimization for each DBMS back end, implementation of aggregation navigation logic, and additional tools and services.

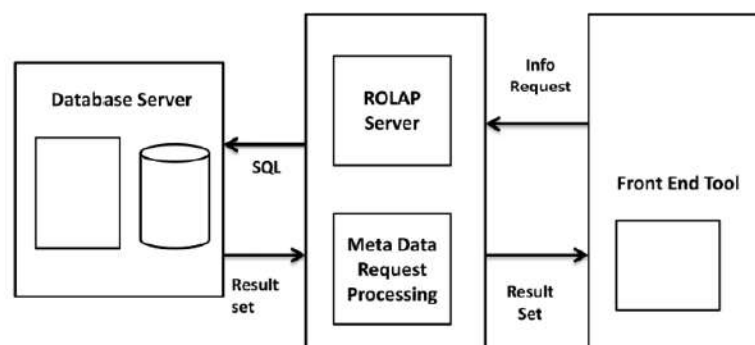


Fig: ROLAP Server

#### **Advantages:**

- ROLAP servers can be easily used with existing RDBMS.
- ROLAP tools do not use pre-calculated data cubes.
- ROLAP server offers highly scalability.
- Can handle large amounts of information.

#### **Disadvantages:**

- ROLAP needs high utilization of manpower, software, and hardware resources.
- Query performance in this model is slow.
- SQL functionality is constrained.

#### 2. Multidimensional OLAP (MOLAP)

Multidimensional OLAP (MOLAP) uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the dataset is sparse. Therefore, many MOLAP servers use two levels of data storage representation to handle dense and sparse datasets.



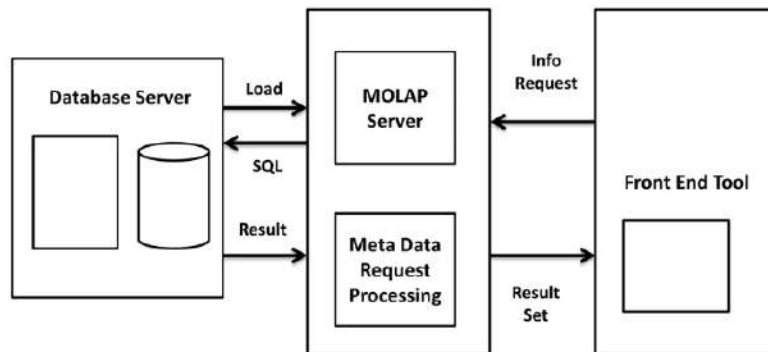


Fig: MOLAP Server

**Advantages:**

- Fast information retrieval.
- Easier to use, therefore MOLAP is suitable for inexperienced users.
- Suitable for slicing and dicing operations.
- Capable of performing complex calculations.

**Disadvantages:**

- MOLAP are not capable of containing detailed data.
- The storage utilization may be low if the data set is sparse.
- It is difficult to change the dimensions without re-aggregating.

### 3. Hybrid OLAP (HOLAP)

Hybrid OLAP is a mixture of both ROLAP and MOLAP. It offers fast computation of MOLAP and higher scalability of ROLAP. HOLAP server allows to store large data volumes of detailed information. HOLAP uses two databases.

1. Aggregated or computed data is stored in a multidimensional OLAP cube
2. Detailed information is stored in a relational database.

**Advantages:**

- HOLAP provides the benefits of both MOLAP and ROLAP.
- It provide quick access at all levels of aggregation.

**Disadvantages:**

- HOLAP architecture is very complicated because it supports both MOLAP and ROLAP servers.
- There are higher chances of overlapping especially into their functionalities.

## Conceptual Modeling of Data Warehouse

A conceptual data model recognizes the highest-level relationships between the different entities. The goal of conceptual data warehouse modeling is to develop a schema for logical representation of data stored in data warehouse.

**Schema** is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. We use **Star schema**, **Snowflake schema**, and **Fact-Constellation schema** for conceptual modeling of data warehouse.

**Star Schema**

It is the data warehouse schema that contains two types of tables: *Fact Table* and *Dimension Tables*. Fact table lies at the center point and dimension tables are connected with fact table such that star shape is formed.

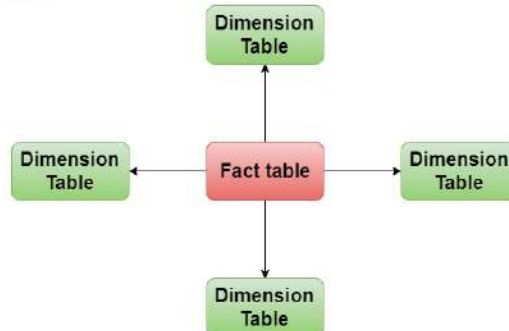


Fig: Star Schemas for Multidimensional Modal

- The **fact table** contains the detailed summary data. Its primary key has one key per dimension. Each tuple of the Fact table consists of a foreign key pointing to each of the dimension tables. It also stores numeric values.
- The **dimension** table consists of columns that correspond to the attributes of the dimensions. The primary key of a dimension table is a foreign key in fact table.

**Example:**

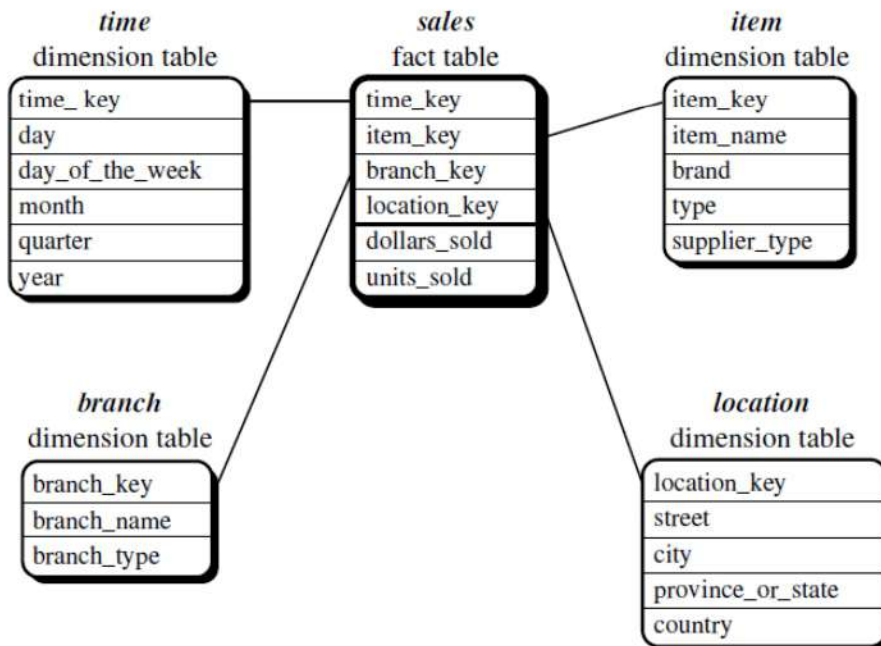


Fig: Star schema for data warehouse for sales.

**Advantages:**

- It is easy to understand and small number of tables can join.
- Since star schema contains de-normalized dimension tables, it leads to simpler queries due to lesser number of join operations and it also leads to better system performance.

**Disadvantages:**

- It is difficult to maintain integrity of data in star schema due to de-normalized tables.
- Redundancy of the data hence occupies additional space.

**Snowflake Schema**

The snowflake schema is a variant of the star schema model, where some dimension tables are *normalized* which splits data into additional tables.

The snowflake schema is represented by centralized fact table which is connected to multiple dimension table and this dimension table can be normalized into additional dimension tables.

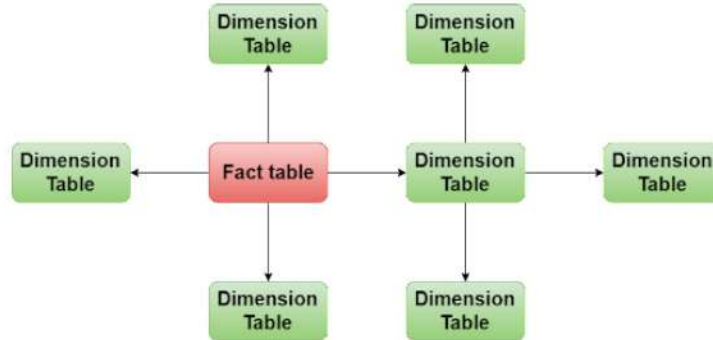


Fig: Snowflake Schemas for Multidimensional Modal

Snowflake Schema eliminates the redundancies and hence saves the storage space. It increases the number of dimension tables and requires more foreign key joins. The result is more complex queries and reduced query performance.

**Example:**

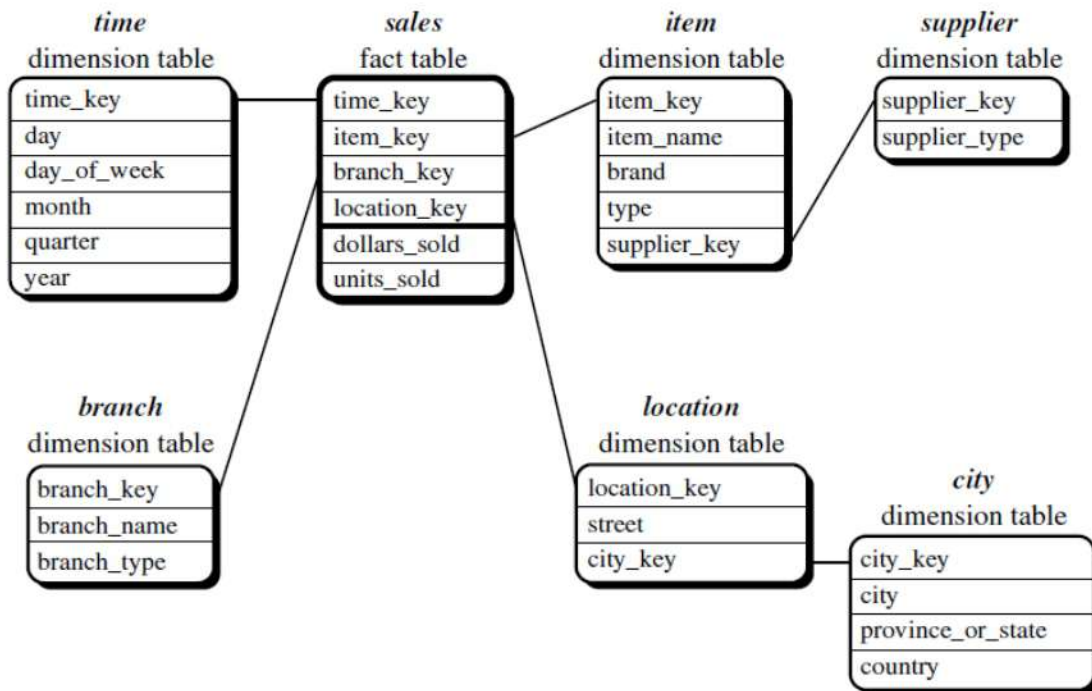


Fig: Snowflake schema of a data warehouse for sales.

The *item* dimension table in star schema is normalized and split into two dimension tables, namely *item* and *supplier* table. The *item* dimension table now contains the attributes *item key*, *item name*, *brand*, *type*, and *supplier key*, where *supplier key* is linked to the *supplier* dimension table, containing *supplier key* and *supplier type* information. Similarly, the single dimension table for *location* in the star schema can be normalized into two new tables: *location* and *city*. The *city key* in the new *location* table links to the *city* dimension.

**Fact Constellation Schema**

A Fact constellation schema is a type of schema which consists of more than one fact table that share many 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.

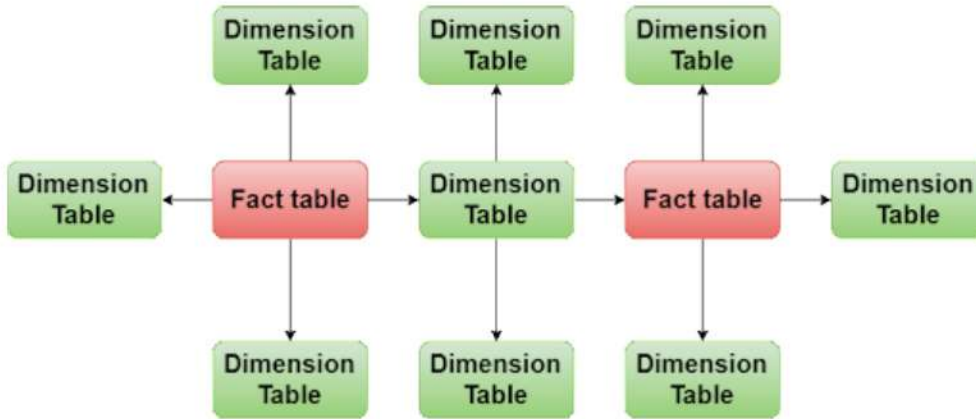


Fig: Fact constellation Schemas for Multidimensional Modal

The main disadvantage of fact constellation schemas is its more complicated design.

Example:

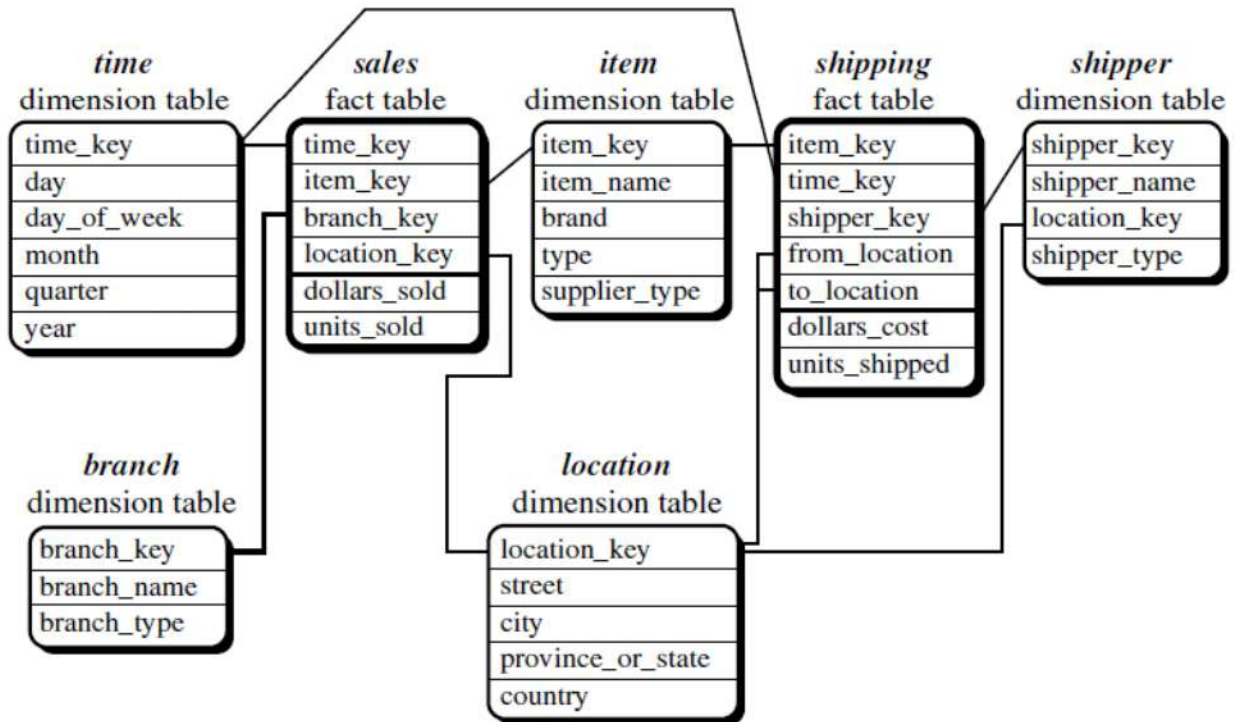


Fig: Fact constellation schema of a data warehouse for sales and shipping.

This schema specifies two fact tables, *sales* and *shipping*. The *sales* table definition is identical to that of the star schema. The *shipping* table has five dimensions, or keys: *item key*, *time key*, *shipper key*, *from location*, and *to location*, and two measures: *dollars cost* and *units shipped*. A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimensions tables for *time*, *item*, and *location* are shared between both the *sales* and *shipping* fact tables.

## Data Warehouse Architecture

Data warehouses often adopt a three-tier architecture, as presented in Figure below:

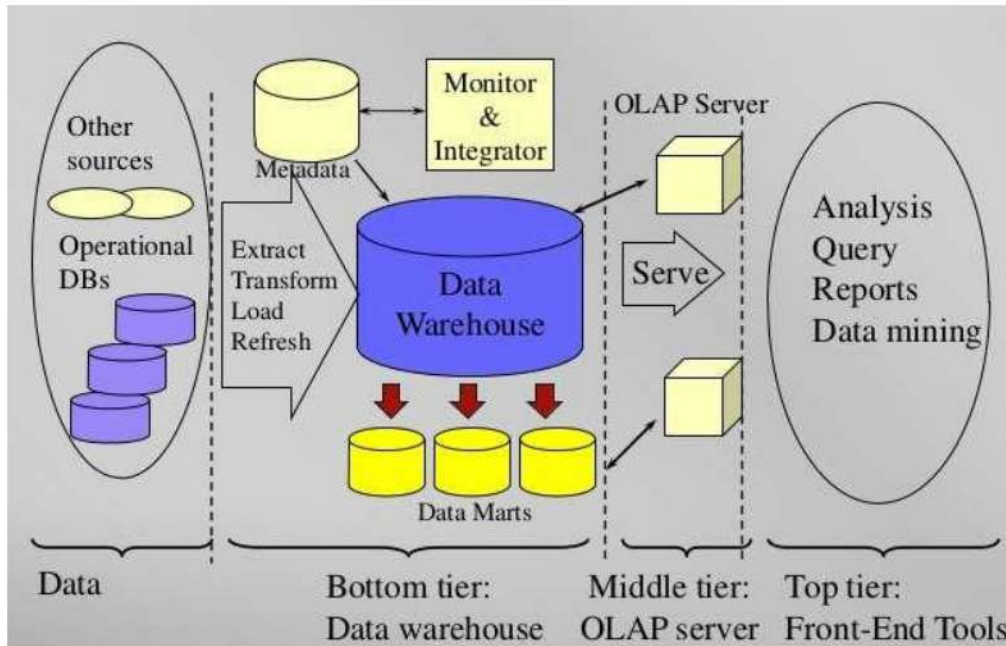


Fig: A three-tier data warehousing architecture.

### 1. Data Sources

A data warehouse system uses heterogeneous sources of data either from operational databases or from some external sources.

### 2. Bottom Tier

The bottom tier of the architecture is the data warehouse database server. It is the relational database system. Data is feed into bottom tier by some back-end tools and utilities. The back end tools and utilities perform the following functions:

- **Data extraction:** gathers data from multiple, heterogeneous and external sources.
- **Data cleaning:** Detect errors in data and correct them when possible.
- **Data transformation:** converts data from legacy or host format to warehouse format.
- **Load:** which sorts, summarizes, checks integrity, and builds indices and partitions.
- **Refresh:** which involves updating from data sources to the warehouse.

### 3. Middle Tier

Middle tier is an OLAP server that can be implemented using either relational OLAP (ROLAP) model or multidimensional OLAP (MOLAP) model.

- **ROLAP** is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
- **MOLAP** directly implements multidimensional data and operations.

### 4. Top Tier

The top tier is a front-end client layer. The top tier layer holds following tools:

- **Query and Reporting tools:** Production reporting tool.
- **Analysis tools:** Prepare charts based on analysis.
- **Data mining tools:** Discover hidden knowledge, pattern.

**Data Warehouse Implementation**

Data warehouse is represented by data cube. The following things we must consider to implement data warehouse:

1. Efficient cube computation techniques
2. Access methods
3. Query processing techniques

**1. Efficient Cube Computation Techniques**

- Data cube computation is an essential task in data warehouse implementation. The pre computation of all or part of a data cube can greatly reduce the response time and enhance the performance of on-line analytical processing.
- It computes aggregates of overall subsets of the dimensions specified in the operation.
- A major challenge related to precomputation would be time and storage space if all the cuboids in the data cube are computed, especially when the cube has many dimensions.
- The storage requirements are more expensive when many of the dimensions have associated concept hierarchies, this problem is referred as *curse of dimensionality*.
- Total number of cuboids for an *n*-dimensional data cube is  $2^n$ . If the dimensions have hierarchy, then the total number of cuboid is calculated by,

$$Total\ number\ of\ cuboids = \prod_{i=1}^n (L_i + 1)$$

Where  $L_i$  is the number of levels associated with dimension  $i$ . One is added to  $L_i$  to include the *virtual* top level, all.

**Example:** If the cube has 10 dimensions and each dimension has 4 levels, the total number of cuboids that can be generated is  $5^{10} \approx 9.8 \times 10^6$ .

**2. Access Methods**

There are two access methods: *Bitmap Index* and *Join Index*.

- **Bitmap Indexing:** This is an alternative representation of base table. It allows quick searching in data cube. In the bitmap index for a given attribute, there is a distinct bit vector,  $B_v$ , for each value  $v$  in the attribute's domain. If a given attribute's domain 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. Another bits for that row are set to 0. For example:

Base table			Index on Region				Index on Type		
Cust	Region	Type	RecID	Asia	Europe	America	RecID	Retail	Dealer
C1	Asia	Retail	1	1	0	0	1	1	0
C2	Europe	Dealer	2	0	1	0	2	0	1
C3	Asia	Dealer	3	1	0	0	3	0	1
C4	America	Retail	4	0	0	1	4	1	0
C5	Europe	Dealer	5	0	1	0	5	0	1

- **Join Indexing:** Join indexing registers the joinable rows of two relations from a relational database. Hence, the join index records can identify joinable tuples without performing costly join operations. Join indexing is especially useful for maintaining the relationship between a foreign key and its matching primary keys, from the joinable relation. Join indices may span multiple dimensions to form composite join indices. We can use join indices to identify sub cubes that are of interest.

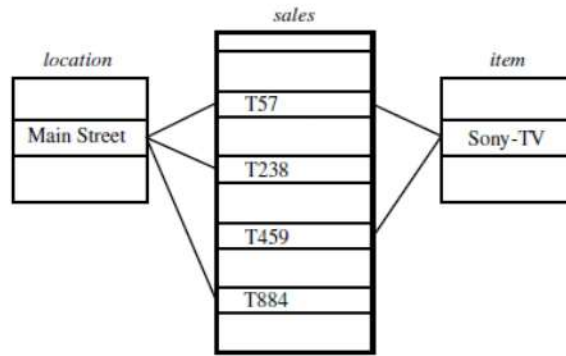


Fig: Linkage between a sales fact table and location and item dimension table

location	sales_key
...	...
Main Street	T57
Main Street	T238
Main Street	T884
...	...

item	sales_key
...	...
Sony-TV	T57
Sony-TV	T459
...	...

location	item	sales_key
...	...	...
Main Street	Sony-TV	T57
...	...	...

Fig: Join index table based on the linkage between the sales fact table and location and item dimension tables shown in above figure

### 3. Query Processing Techniques

Query processing should proceed as follows:

1. Determine which operations should be performed on the available cuboids.
2. Determine which materialized cuboid(s) should be selected for OLAP operation.

Suppose that we define a data cube for *AllElectronics* of the form

“sales\_cube [time, item, location]: sum(sales in dollars)”.

The dimension hierarchies used are

- “day < month < quarter < year” for time,
- “item name < brand < type” for item, and
- “street < city < province or state < country” for location

Suppose that the query to be processed is on

{brand, province or state} with the selection constant “year = 2004”.

Also, suppose that there are four materialized cuboids available, as follows:

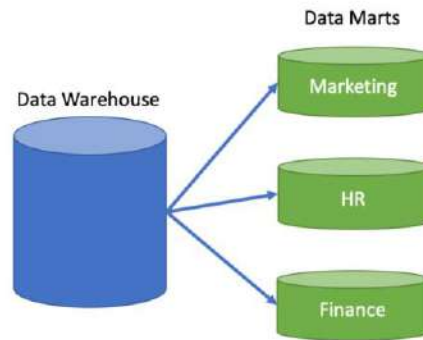
- cuboid 1: {year, item name, city}
- cuboid 2: {year, brand, country}
- cuboid 3: {year, brand, province or state}
- cuboid 4: {item name, province or state} where year = 2004

Which of the above four cuboids should be selected to process the query?

**Data Marts**

Data mart is a subset of data warehouse built specifically for a particular group. For example, the marketing data mart may contain data related to items, customers, and sales. Data marts are confined to specific selected subjects.

The primary purpose of a data mart is to isolate - or partition - a smaller set of data from a whole to provide easier data access for the end consumers.



**Types of Data Marts**

Depending on the source of data, data marts can be categorized as *dependent*, *independent*, and *hybrid*.

- **Dependent data marts** draw data from a central data warehouse that has already been created.
- **Independent data marts** are standalone systems built by drawing data directly from operational or external sources of data or both.
- **Hybrid data marts** can draw data from operational systems or data warehouses.

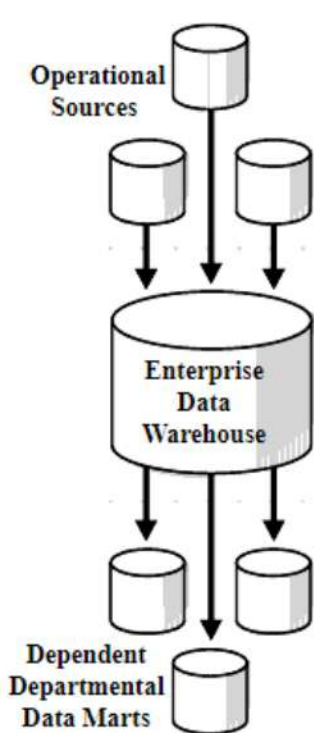


Fig: Dependent Data Mart

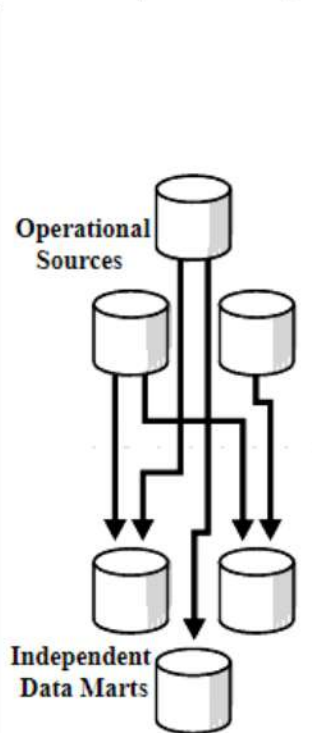


Fig: Independent Data Mart

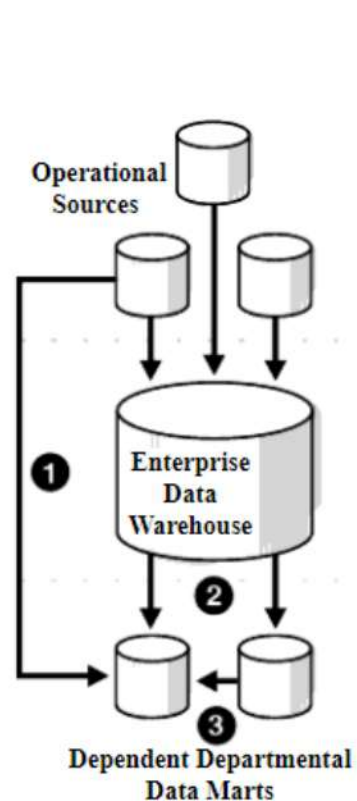


Fig: Hybrid Data Mart



### **Difference Between Data Warehouse and Data Mart**

<i>Data Warehouse</i>	<i>Data Mart</i>
A Data Warehouse is a large repository of data collected from different organizations or departments within a corporation.	A data mart is an only subtype of a Data Warehouse. It is designed to meet the need of a certain user group.
It may hold multiple subject areas.	It holds only one subject area. For example, marketing, finance etc.
It is a centralized system.	It is a decentralized system.
Data warehouse is top-down model.	It is a bottom-up model.
In data warehouse, Fact constellation schema is used.	In data mart, Star schema and snowflake schema are used.
Data Warehouse is the data-oriented in nature.	Data mart is the project-oriented in nature.
Data Ware house has long life.	Data mart has short life than warehouse.
In data warehouse, data are contained in detail form.	In data mart, data are contained in summarized form.
The designing process of Data Warehouse is quite difficult.	The designing process of Data Mart is easy.
Long time for processing the data because of large data.	Less time for processing the data because of handling only a small amount of data.

### **Components of Data Warehouse**

1. **Data Warehouse Database:** The central component of a data warehouse architecture is a database that stocks all enterprise data and makes it manageable for reporting.
2. **Load Manager:** This component performs the operations associated with the extraction and load of data into the warehouse. These tasks include the simple transformation of data to prepare data for entry into the warehouse.
3. **Warehouse Manager:** A warehouse manager is responsible for the warehouse management process. The operations performed by the warehouse manager are the analysis, aggregation, backup and collection of data, de-normalization of the data.
4. **Query Manager:** It performs all the operations related to the management of user queries. The query manager is responsible for directing the queries to suitable tables. By directing the queries to appropriate tables, it speeds up the query request and response process. In addition, the query manager is responsible for scheduling the execution of the queries posted by the user.
5. **Meta Data:** Metadata is defined as data about data that describes the data warehouse. The data that is used to represent other data is known as metadata. For example, the index of a book serves as a metadata for the contents in the book. In other words, we can say that metadata is the summarized data that leads us to detailed data. There are two types of metadata in data warehousing:
  - **Technical Metadata** comprises of information that can be used by developers and managers when executing warehouse development and administration tasks.
  - **Business Metadata** includes information that offers an easily understandable standpoint of the data stored in the warehouse.

6. **Data Warehouse Access Tools:** Access tools allow users to interact with the data in data warehouse. These tools fall into four different categories: query and reporting tools, application development tools, data mining tools, and OLAP tools.

### **Need for Data Warehousing**

Data Warehouse is needed for the following reasons:

1. **Business User:** Business users require a data warehouse to view summarized data from the past. Since these people are non-technical, the data may be presented to them in an elementary form.
2. **Store historical data:** Data Warehouse is required to store the time variable data from the past. This input is made to be used for various purposes.
3. **Make strategic decisions:** Some strategies may be depending upon the data in the data warehouse. So, data warehouse contributes to making strategic decisions.
4. **For data consistency and quality:** Bringing the data from different sources at a commonplace, the user can effectively undertake to bring the uniformity and consistency in data.
5. **High response time:** Data warehouse has to be ready for somewhat unexpected loads and types of queries, which demands a significant degree of flexibility and quick response time.

### **Trends in Data Warehousing**

- **Multiple Data Types:** Different types of data needs to be integrated into data warehouse systems. These data types include structured numeric data, structured text, images, video, audio, spatial data, etc. Data warehouses need to provide facilities of searching all types of data stored.
- **Visualization Types:** Data warehouses must support various types of charts and interactive visualization system. Beside, modern applications needs 3D representations of visualization and mechanisms of summary at different levels.
- **Parallel Processing:** Analysts need to analyze large volume of data stored in data warehouse and need to produce results fast. Uniprocessor systems may not be sufficient in many cases therefore data warehouse systems need to support parallel processing. It can be achieved either by using parallel processor or by using parallel query processing technique.
- **Query Tools:** Data warehouses systems need to provide query tools to users so that users can specify task, provide feedback, and seek more explanation from the system. Such tools must be use friendly so that non-technical users can also use these tools easily.
- **Data Fusion:** It is the technology dealing with merging of data from disparate sources. It has wider scope and includes real-time merging of data from instruments and monitoring systems.
- **Software Agents:** Software agent is program that is executed in certain environment autonomously and is capable of making decisions based on data obtained from the environment and from other agents. Such agents needs to be integrated into data warehouse systems to provide alerts about predefined business conditions to users.

## Questions and Answers

### **Q. Why a Data Warehouse is separated from Operational Databases?**

**Solution:**

A data warehouse is kept separate from operational databases due to the following reasons:

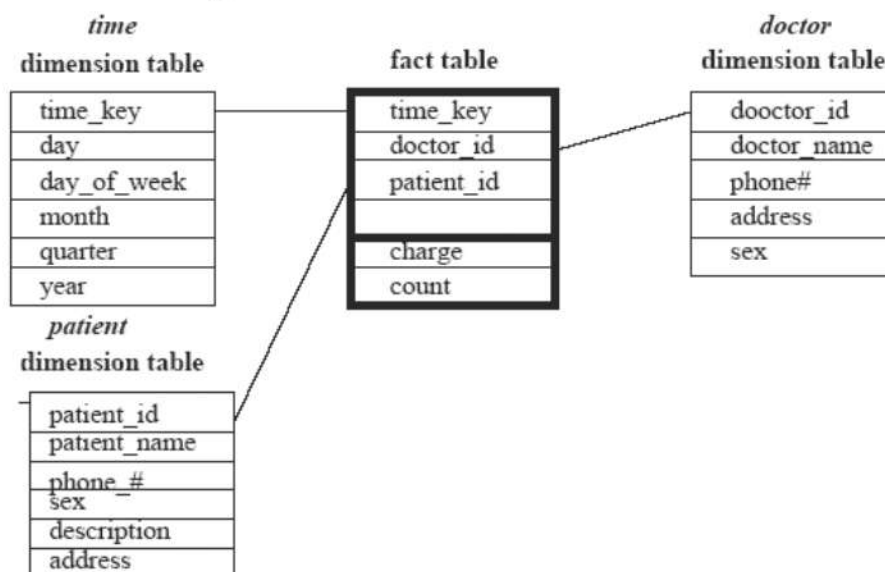
- Data Warehouse queries are complex because they involve the computation of large groups of data at summarized levels.
- It may require the use of distinctive data organization, access, and implementation method based on multidimensional views.
- Performing OLAP queries in operational database degrade the performance of functional tasks.
- Data Warehouse is used for analysis and decision making in which extensive database is required, including historical data, which operational database does not typically maintain.
- The separation of an operational database from data warehouses is based on the different structures and uses of data in these systems.
- Because the two systems provide different functionalities and require different kinds of data, it is necessary to maintain separate databases.

**Q. Suppose that a data warehouse consists of the three dimensions time, doctor, and patient, and the two measures count and charge, where charge is the fee that a doctor charges a patient for a visit.**

- a) Draw a schema diagram for the above data warehouse using one of the schemas. [star, snowflake, fact constellation]
- b) Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2004?
- c) To obtain the same list, write an SQL query assuming the data are stored in a relational database with the schema fee (day, month, year, doctor, hospital, patient, count, charge)

**Solution:**

a) Star Schema is shown in figure below:



**b)** First, we should use roll-up operation to get the year 2004(rolling-up from day then month to year). After getting that, we need to use slice operation to select (2004). Second, we should use roll-up operation again to get all patients. Then, we need to use slice operation to select (all). Finally, we get list the total fee collected by each doctor in 2004. So,

1. roll up from day to month to year
2. slice for year = "2004"
3. roll up on patient from individual patient to all
4. slice for patient = "all"
5. get the list of total fee collected by each doctor in 2004

**c)**

```
Select doctor, Sum(charge)
From fee
Where year = 2004
Group by doctor
```

**Q.** Suppose that a data warehouse for Big-University consists of the following four dimensions: student, course, semester, and instructor, and two measures count and avg\_grade. When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg\_grade measure stores the actual course grade of the student. At higher conceptual levels, avg\_grade stores the average grade for the given combination.

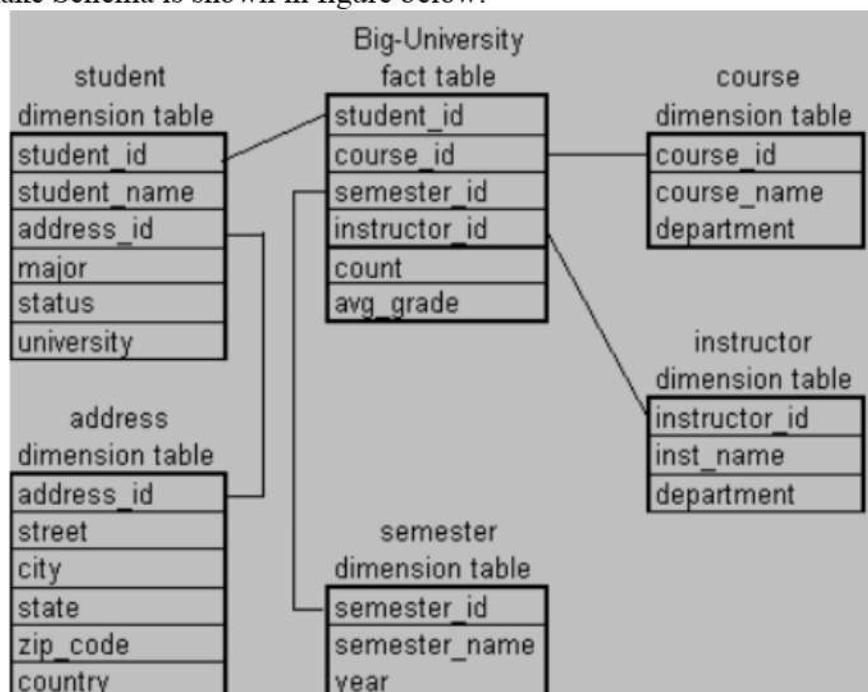
**a)** Draw a snowflake schema diagram for the data warehouse.

**b)** Starting with the base cuboid [student, course, semester, instructor], what specific OLAP operations (e.g., roll-up from semester to year) should one perform in order to list the average grade of CS courses for each BigUniversity student.

**c)** If each dimension has five levels (including all), such as "student < major < status < university < all", how many cuboids will this cube contain (including the base and apex cuboids)?

**Solution:**

**a)** A Snowflake Schema is shown in figure below:



- b) The specific OLAP operations to be performed:
1. Roll-up on course from course\_id to department.
  2. Roll-up on student from student\_id to university.
  3. Dice on course, student with department = "CS" and university = "biguniversity"
  4. Drill-down on student from university to student\_name.
- c) N = 4 dimensions  
The cube will contain  $5^4 = 625$  cuboids.

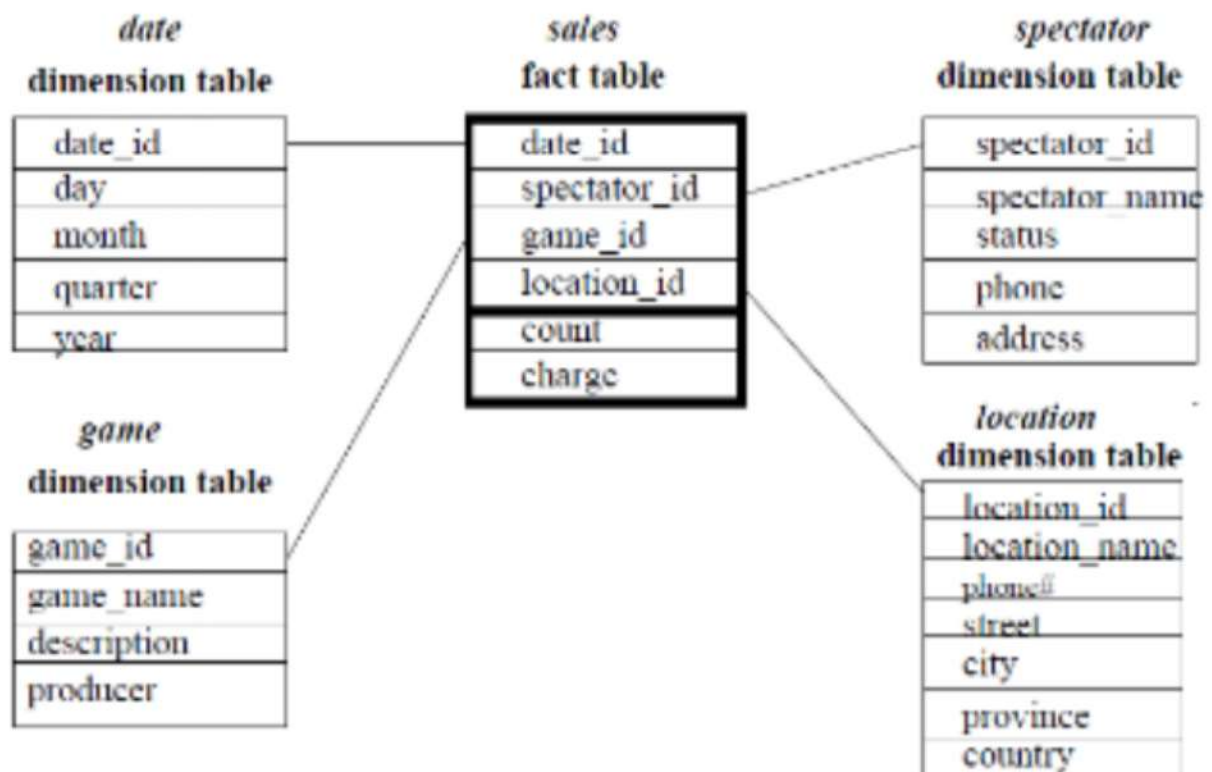
**Q. Suppose that a data warehouse consists of the four dimensions; date, spectator, location, and game, and the two measures, count and charge, where charge is the fee that a spectator pays when watching a game on a given date.**

**Spectators may be students, adults, or seniors, with each category having its own charge rate.**


- a) Draw a star schema diagram for the data
- b) Starting with the base cuboid [date; spectator; location; game], what specific OLAP operations should perform in order to list the total charge paid by student spectators at GM Place in 2004?

**Solution:**

a)



- b) The specific OLAP operations to be performed are:
1. Roll-up on date from date id to year.
  2. Roll-up on spectator from spectator id to status.
  3. Roll-up on location from location id to location name.
  4. Roll-up on game from game id to all.
  5. Dice with status= "students", location name= "GM Place", and year=2004



Please let me know if I missed anything or  
anything is incorrect.  
[poudeljayanta99@gmail.com](mailto:poudeljayanta99@gmail.com)