Unit 1: Introduction to Data Warehousing (08 Marks)

*Data warehousing:

A **Data Warehousing** (DW) is process for collecting and managing business data from varied sources to provide meaningful business insights.

Data warehouse is an information system that contains historical and commutative data from single or multiple sources.

A Data warehouse is typically used to connect and analyse business data from heterogeneous (different by nature) sources.

It simplifies reporting and analysis process of the organization.

It is also a single version of truth for any company for decision making and forecasting.

Characteristics of Data warehousing:

A data warehouse has following characteristics:

1. Subject-Oriented
2. Integrated
3. Time-variant
4. Non-volatile

1. Subject-Oriented:

A data warehouse is subject oriented as it offers information regarding a **theme** instead of companies ongoing operations. These subjects can be sales, marketing, distributions, etc.

A data warehouse never focuses on the ongoing operations. It concentrates on modelling and analysis of data for **decision making**.

It also provides a simple and concise (summarized) view around the specific subject by excluding data which is not helpful to support the decision process.

2. Integrated:

In Data Warehouse, integration means the establishment of a common unit of measure for all similar data from the dissimilar database.

A data warehouse is developed by integrating data from varied sources like a mainframe, relational databases, flat files, etc. and convert all data in single standard format.

This integration helps in effective analysis of data.

Example: There are three different applications labelled A, B and C.

In Application A gender field store logical values like M or F

In Application B gender field is a numerical value,

In Application C application, gender field stored in the form of a character value.

However, after transformation and cleaning process all this data is stored in common format in the Data Warehouse.

3. Time-Variant:

The data collected in a data warehouse with a particular period (time) and offers information from the historical point of view.

Another aspect of time variance is that once data is inserted in the warehouse, it can't be updated or changed.

4. Non-volatile:

Data warehouse is also non-volatile means the previous data is not erased when new data is entered in it. Data is read-only and periodically refreshed.

This also helps to analyse historical data and understand what & when happened. Activities like delete, update, and insert which are performed in an operational application environment are omitted in Data warehouse environment.

Only two types of data operations performed in the Data Warehousing are Data loading and Data access.

*Difference between Operational Database System and Data Warehouse:

Parameters	Operational Database System	Data Warehouse
Definition	Designed to support high volume transaction processing.	Designed to support high volume analytical processing.
Design	Application oriented	Subject oriented
Performance	Low for analysis process	High for analysis process
Data Used	Current data	Historical data
Updation of Data	Regularly	Rarely
Operations on data	Insert, delete, update	Read only
Data redundancy	No	Yes
Access to system	Repetitive	Ad-hoc
Function	Day-to-day operations	Decision making
Applications	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)

*Need for Data Warehousing:

1. Improving integration
2. Speeding up response times
3. Faster and more flexible reporting
4. Recording changes to build history
5. Increasing data quality
6. Unburdening the IT department (no need to track data regularly)
7. Increasing recognizability
8. Increasing findability

*Applications of Data Warehouse:

Information Processing

It deals with querying, statistical analysis, and reporting via tables, charts, or graphs.

Analytical Processing

It supports various online analytical processing such as drill-down, roll-up, and pivoting. The historical data is being processed in both summarized and detailed format.

Data Mining

It helps in the analysis of hidden design and association, constructing scientific models, operating classification and prediction, and performing the mining results using visualization tools.

1. **Airline:**

Analysis of crew assignments, flight data, flight routes, fairs.

2. **Banking:**

Analysis of customer data, transactions, loans, accounts, KYC.

3. **Healthcare:**

Generate patient's treatment reports, share data with tie-in insurance companies, medical aid services, etc.

4. **Public sector:**

In the public sector, data warehouse is used for intelligence gathering. It helps government agencies to maintain and analyse tax records, health policy records, for every individual.

5. **Investment and Insurance sector:**

In this sector, the warehouses are primarily used to analyse data patterns, customer trends, and to track market movements.

6. **Retail chain:**

In retail chains, Data warehouse is widely used for distribution and marketing. It also helps to track items, customer buying pattern, promotions and also used for determining pricing policy.

7. **Telecommunication:**

A data warehouse is used in this sector for product promotions, sales decisions and to make distribution decisions.

8. **Hospitality Industry:**

This Industry utilizes warehouse services to design as well as estimate their advertising and promotion campaigns where they want to target clients based on their feedback and travel patterns.

*Data Warehouse Architecture:

There are mainly three types of Datawarehouse Architectures:

Single-tier architecture

The objective of a single layer is to minimize the amount of data stored. This goal is to remove data redundancy. This architecture is not frequently used in practice.

Two-tier architecture

Two-layer architecture separates physically available sources and data warehouse. This architecture is not expandable and also not supporting a large number of end-users. It also has connectivity problems because of network limitations.

Three-tier architecture:

This is the most widely used architecture.

It consists of the Top, Middle and Bottom Tier.

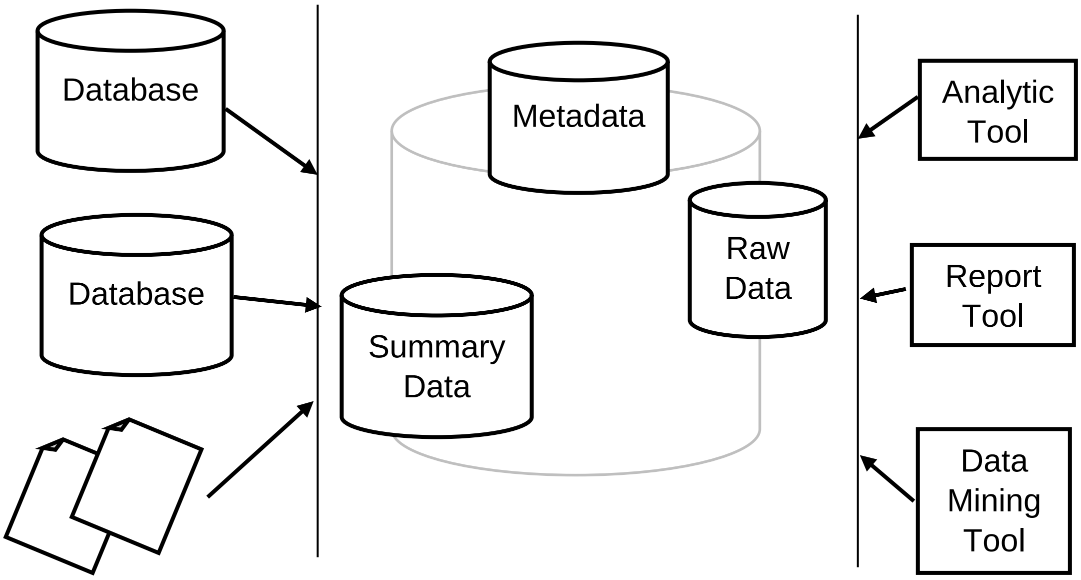


Fig: Three-Tiered Data Warehouse Architecture

1. **Bottom Tier:**

   The database or data sources of the Datawarehouse servers is the bottom tier.

   It is usually a relational database system.

2. **Middle Tier:**

   The middle tier in Data warehouse is an OLAP server which is implemented using either ROLAP or MOLAP model.

   This application tier presents an abstracted view of the database.

   This layer also acts as a mediator between the end-user and the database.

3. **Top-Tier:**

   The top tier is a front-end client layer.

   Top tier is the tools and API that user used to get data out from the data warehouse.

   The different tools are Query tools, reporting tools, managed query tools, Analysis tools and Data mining tools.

*Data Warehouse Models:

Three main types of Data Warehouses are:

1. Enterprise Data Warehouse
2. Data Marts
3. Virtual Warehouse

1. Enterprise Data Warehouse (EDW):

Enterprise Data Warehouse is a centralized warehouse, which aggregates the information automatically.

It offers a unified approach for organizing and representing data.

It also provides the ability to classify data according to the subject and give access accordingly to users.

It provides decision support service across the enterprise.

2. Data Marts:

A data mart is a subset of the data warehouse.

It is specially designed for a particular line of business, such as sales, finance, sales or finance. In an independent data mart, data can collect directly from sources.

Due to large amount of data, a single warehouse can become overburdened.

So, to prevent the warehouse from becoming impossible to navigate, subdivisions created, called as Data Marts.

These data marts divide the information saved in the warehouse into categories or specific groups of users.

In a simple word Data mart is a subsidiary of a data warehouse.

3. Virtual Warehouse:

A virtual warehouse is essentially a separate business database, which contains only required data for operation system.

The data found in a virtual warehouse is usually copied from multiple sources throughout an operation system.

Virtual warehouse is used to search the data quickly and without accessing the entire system. It speeds up the overall access process.

*ETL Process in Data Warehouse:

ETL is a process in Data Warehousing and it stands for **Extract**, **Transform** and **Load**. It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area and then finally, loads it into the Data Warehouse system.

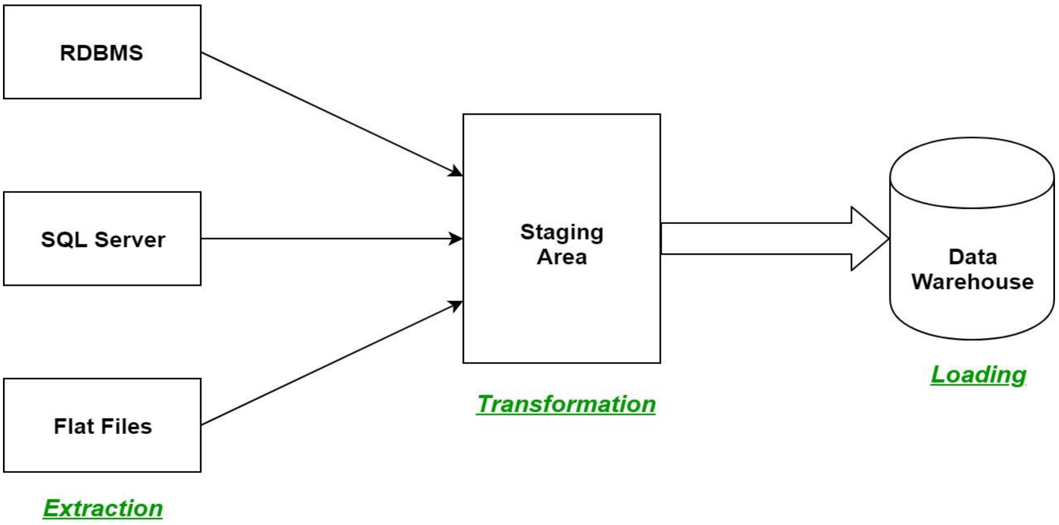


Fig: ETL process in Data Warehousing

1. **Extraction:**

   The first step of the ETL process is extraction.

   In this step, data from various source systems is extracted which can be in various formats like relational databases, No SQL, XML and flat files into the staging area.

   The data cannot be loaded in data warehouse; therefore, this is one of the most important steps of ETL process.

2. **Transformation:**

   The second step of the ETL process is transformation.

   In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format.

It may involve following processes/tasks:

- Filtering – loading only certain attributes into the data warehouse.
- Cleaning – filling up the NULL values and missing values.
- Joining – joining multiple attributes into one.
- Splitting – splitting a single attribute into multiple attributes.
- Sorting – sorting tuples on the basis of some attribute (generally key-attribute).

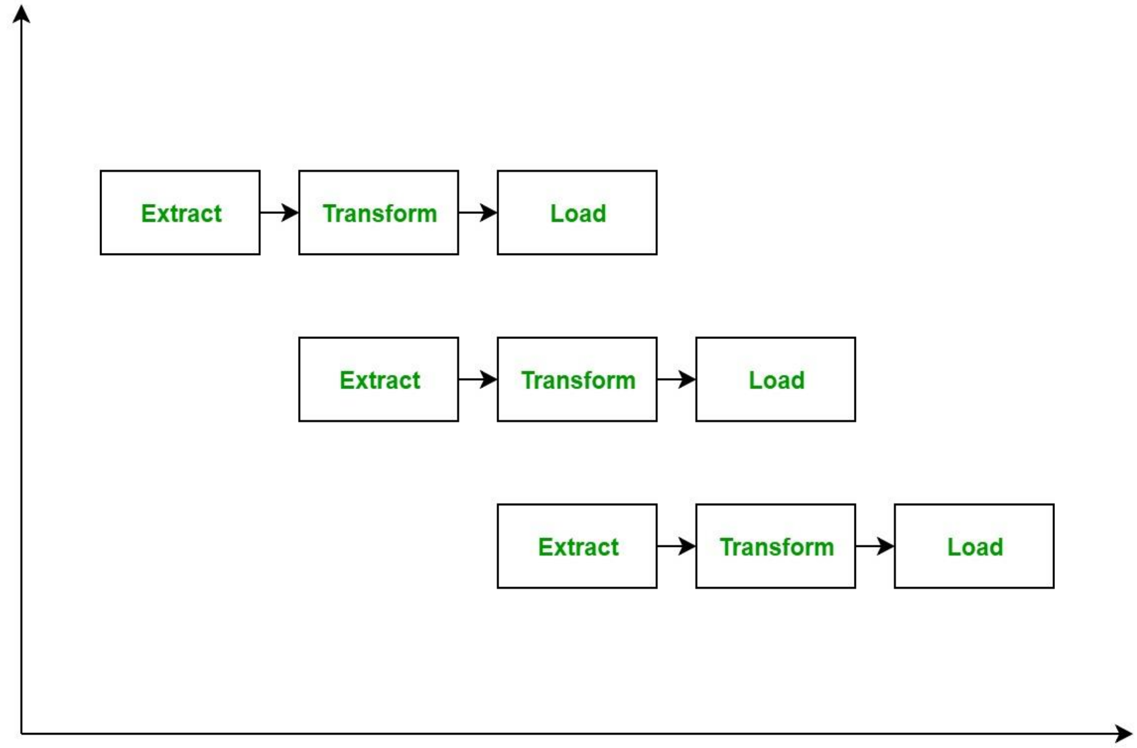
3. **Loading:**

   The third and final step of the ETL process is loading.

   In this step, the transformed data is finally loaded into the data warehouse.

ETL Pipelining:

ETL process can also use the pipelining concept i.e. as soon as some data is extracted, it can transform and during that period some new data can be extracted. And while the transformed data is being loaded into the data warehouse, the already extracted data can be transformed.



ETL Tools: *Most commonly used ETL tools are Sybase, Oracle Warehouse builder, CloverETL and MarkLogic.*

*Metadata Repository:

(Metadata: data about data, repository: big container)

Metadata is information about the structures that contain the actual data.

It is data about the structures that contain data. Metadata may describe the structure of any data, of any subject, stored in any format.

Metadata repository contains the structures of all data at one place.

This gives the plenty of data more than requirement for decision making.

With one stop information, business will have more control on the changes, and can-do impact analysis.

Metadata Repository used for building, maintain, managing Data warehouse.

For example, a line in sales database may contain: 4030 KJ732 299.90

This is a meaningless data until we consult the Meta that tell us it was.

The Meta of the data is

- Model number: 4030
- Sales Agent ID: KJ732
- Total sales amount of $299.90

Therefore, Meta Data are essential ingredients in the transformation of data into knowledge.

Benefits of Metadata Repository:

a. **Integration** of the metadata across the organization.
b. Build relationship between various **metadata types**
c. Build relationship between various **disparate** (different in nature) **systems**.
d. **Version** control of the changes at structure level.
e. link view to **master data**.
f. automatic **sync**hronization with various authorized metadata source systems.
g. More **control** to business decisions.
h. discovering **discrepancies**, **gaps**, **lineage**, **metrics** at data structure level.

Metadata can be classified into following categories:

1. **Technical Meta Data**: This kind of Metadata contains information about warehouse which is used by Data warehouse designers and administrators.
2. **Business Meta Data:** This kind of Metadata contains detail that gives end-users a way easy to understand information stored in the data warehouse.

*Benefits/Advantages of a Data Warehouse:

1. Delivers enhanced business intelligence

By having access to information from various sources in a single platform, decision makers will no longer need to rely on limited data.

2. Saves times

A data warehouse standardizes, preserves, and stores data from different sources, and integration of all the data in one place.

So, all critical data is available to all users simultaneously.

3. Enhances data quality and consistency

A data warehouse converts data from multiple sources into a consistent format.

The data from different sources can be filtered, sorted, cleaned.

This will lead to more accurate data, which will become the basis for solid decisions.

4. Generates a high Return on Investment (ROI)

Companies experience higher revenues and cost savings than those that haven't invested in a data warehouse.

5. Provides competitive advantage

Data warehouses helps to get a holistic (as a whole not parts) view of their current standing and evaluate opportunities and risks, thus providing companies with a competitive advantage.

6. Improves the decision-making process

Data warehousing provides better insights (detailed understanding) to decision makers by maintaining a related database of current and historical data.

7. Enables organizations to forecast with confidence

With advanced features of Data warehouse, organization can forecast their line of action easily.

8. Streamlines (well organized) the flow of information

Data warehousing facilitates the flow of information through all related or non-related parties.

Chapter 1: Assignment Questions

2 Marks Questions:

1. Define Data Warehouse.
2. Explain need of Data Warehousing.
3. Define extraction, transformation and loading in Data Warehouse.
4. Explain any two characteristics of Data Warehousing.

4/6 Marks Questions:

1. Differentiate between operational database system and data warehouse.
2. Explain three-tiered data warehouse architecture.
3. Explain any two data warehouse models.
4. Explain ETL process in data warehouse.
5. Explain metadata repository with its benefits.
6. Explain advantages of Data Warehousing.

Annexure 1

Metadata Repository:

Ex: Metadata of a Book Store:

1. Name of book
2. Summary of book
3. Publication of book
4. Edition of book
5. Author of book
6. Date of publication
7. Availability of book
8. Reviews of book

Above information (metadata) helps to search the book, access the book, whether to purchase or not.

Difference: Data Warehouse Vs Data Marts:

Parameters	Data Warehouse	Data Marts
Definition	Collection of large amounts of data.	Sub division of Data warehouse
Subjects (departments)	All departments in an organization	Specific department
Design process	Complex	Simple
Implementation time required	More	Less
Data handling time required	More	Less
Storage required	More (100GB to 1TB)	Less (up to 100GB)
Flexibility	More	Less
Function	Subject independent	Subject dependent

Unit 2: Data Warehousing Modelling & OLAP I (12 Marks)

*OLAP: Online Analytical Processing:

OLAP is a software that allows users to analyse information from multiple database systems at the same time.

Using OLAP technique, analysts easily extract and view business data from different points of view.

Analysts frequently need to group, aggregate and join data. These operations in relational databases are resource intensive. With OLAP data can be pre-calculated and pre-aggregated, making analysis faster.

OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy.

Data Cube / OLAP Cube:

When data is grouped or combined in multidimensional matrices, it is called Data Cubes. Also known as "Multidimensional databases," "materialized views," and "OLAP (On-Line Analytical Processing) cube."

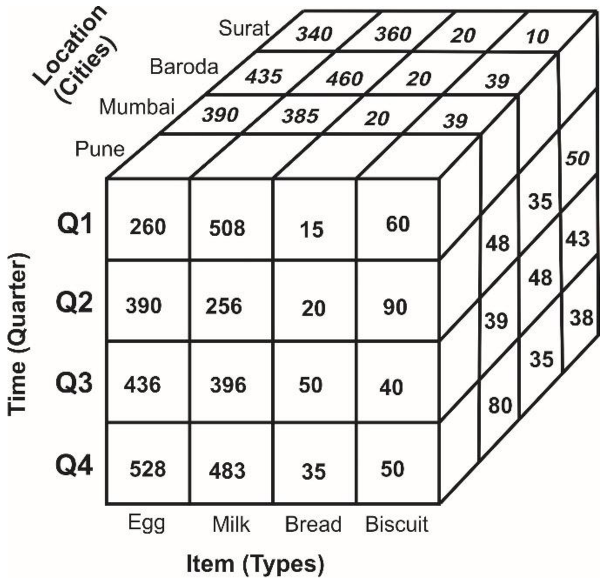


Fig: Data or OLAP Cube

The OLAP cube is a data structure optimized for very quick data analysis.

The OLAP Cube consists of numeric facts called measures *(it is property on which calculations can be made)* which are categorized by dimensions.

OLAP Cube is also called the **hypercube**.

Usually, data operations and analysis are performed using the simple spreadsheet, where data values are arranged in row and column format. This is ideal for two-dimensional data. However, OLAP contains multidimensional data, with data usually obtained from a different and unrelated source. Using a spreadsheet is not an optimal option. The cube can store and analyse multidimensional data in a logical and orderly manner.

OLAP Cube Process:

A Data warehouse would extract information from multiple data sources and formats like text files, excel sheet, multimedia files, etc.

The extracted data is cleaned and transformed. Data is loaded into an OLAP server (or OLAP cube) where information is pre-calculated in advance for further analysis.

*Multi-Dimensional Data Model:

A multidimensional model views data in the form of a data-cube.

A data cube enables data to be modelled and viewed in multiple dimensions.

Multidimensional data model consists of Fact table and dimension tables.

Fact Table:

This table contains primary key of multiple dimension tables.

It contains facts or measures like quantity sold, amount sold, etc.

Dimension Table:

This table provides descriptive information for all measures recorded in fact table, like product, item, location, time, etc.

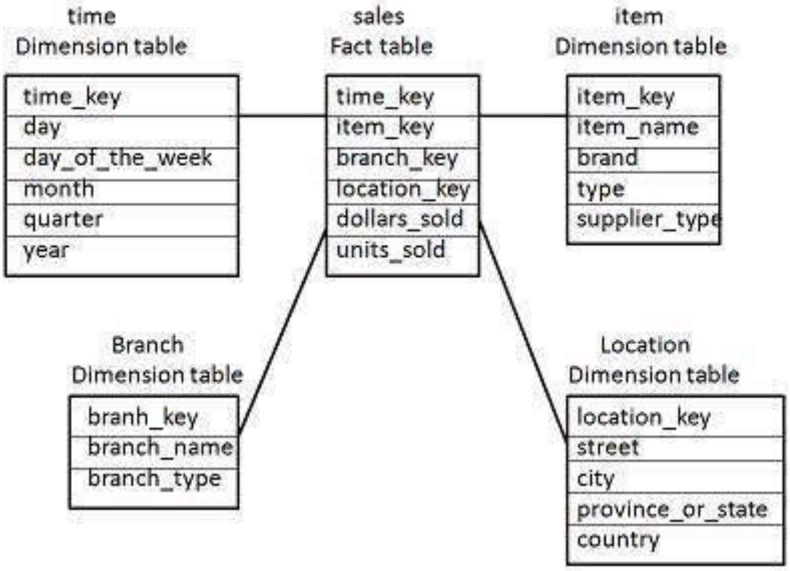
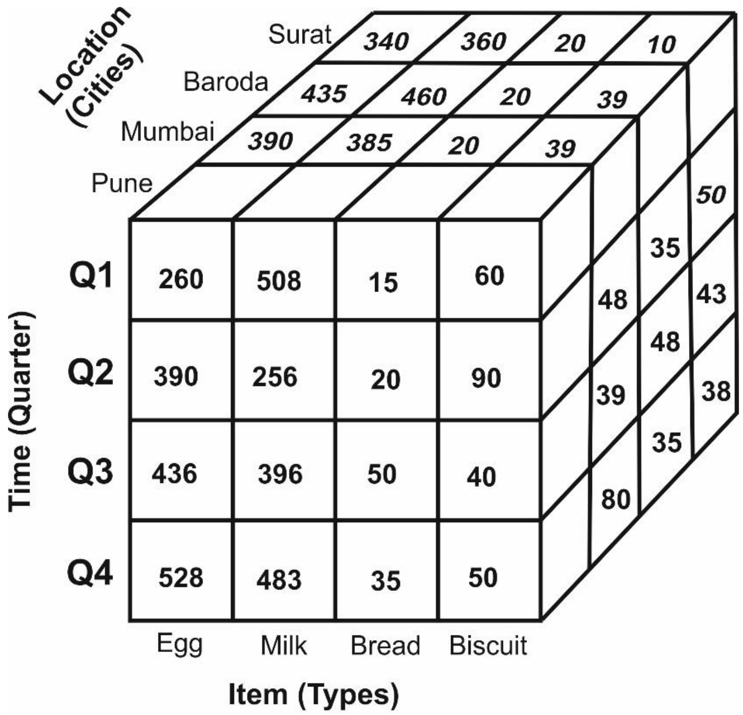


Fig: Fact & Dimension tables for Multidimensional Data Modeling

Consider the data of a shop for items sold per quarter in the city of Delhi. The data is shown in the table. In this 2D representation, the sales for Delhi are shown for the time dimension (organized in quarters) and the item dimension (classified according to the types of an item sold). The fact or measure displayed in rupee sold (in thousands).

	Location=Surat				Location=Baroda				Location=Mumbai				Location=Pune			
	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

The data from above table can be represented in the form of a 3D (3-Dimensional) data cube, as shown in fig:



Multidimensional Data (OLAP) Cube

Fig: 3-Dimensional Data Cube

*Datawarehouse Schemas:

1. Star Schema:

A star schema is the primary form of a dimensional model, in which data are organized into **facts** and **dimensions**.

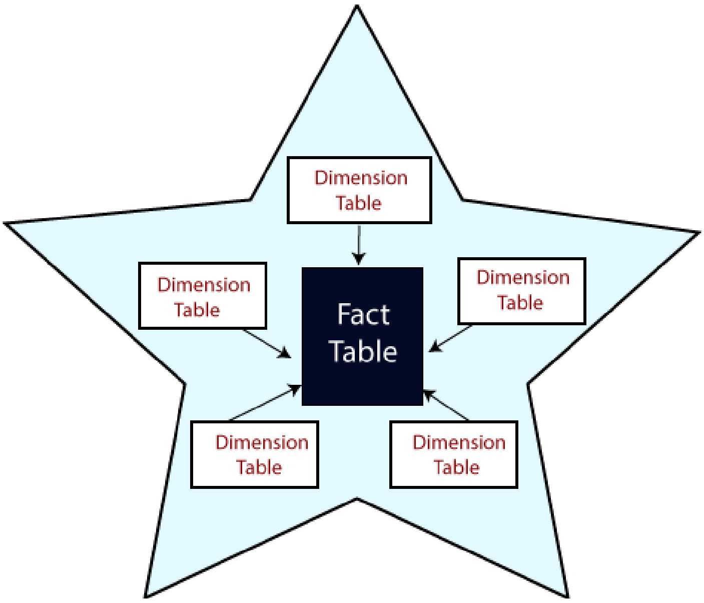
A fact is an event that is counted or measured, such as a sale.

A dimension includes all information about the fact, such as date, item, or customer.

The star schema is the explicit data warehouse schema.

It is known as **star schema** because the entity-relationship diagram of this schemas simulates a star, with points, diverge from a central table.

The centre of the schema consists of a large fact table, and the points of the star are the dimension tables.



Star Schema

Fact Table:

This table contains primary key of multiple dimension tables.

It contains facts or measures like quantity sold, amount sold, etc.

Dimension Table:

This table provides descriptive information for all measures recorded in fact table, like product, item, location, time, etc.

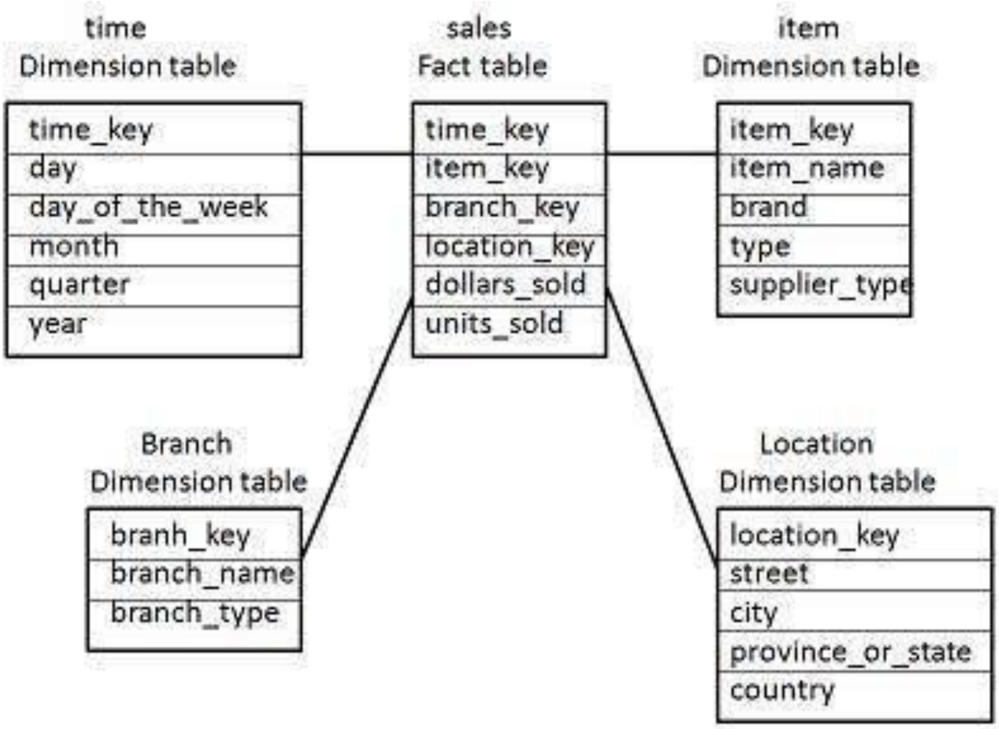


Fig: Star schema

Characteristics of Star Schema:

- It creates a DE-normalized database that can quickly provide query responses.
- It provides a flexible design that can be changed easily.
- It provides a parallel processing in design.
- It reduces the complexity of metadata for both developers and end-users.

Advantages of Star Schema

1. Star schema is easy to understand for end users.
2. Easy to navigate.
3. Provides instant analysis of large datasets.
4. Built in referential integrity.
5. Increases query performance.

2. Snowflake Schema:

A snowflake schema is refinement of the star schema.

"A schema is known as a snowflake where one or more-dimension tables do not connect directly to the fact table, but must join through other dimension tables."

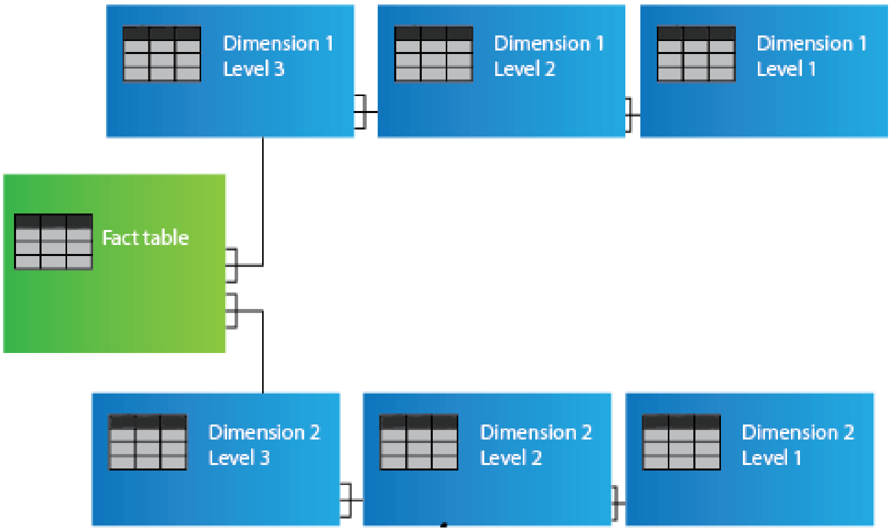
The snowflake schema is an expansion of the star schema where each point (dimension table) of the star explodes into more points (more dimension tables).

Snowflaking is a method of normalizing the dimension tables in a STAR schema.

Snowflaking is used to develop the performance of specific queries.

The snowflake schema consists of one fact table which is linked to many dimension tables, which can be linked to other dimension tables through a many-to-one relationship.

Tables in a snowflake schema are generally normalized to the third normal form.



Snowflake Schema

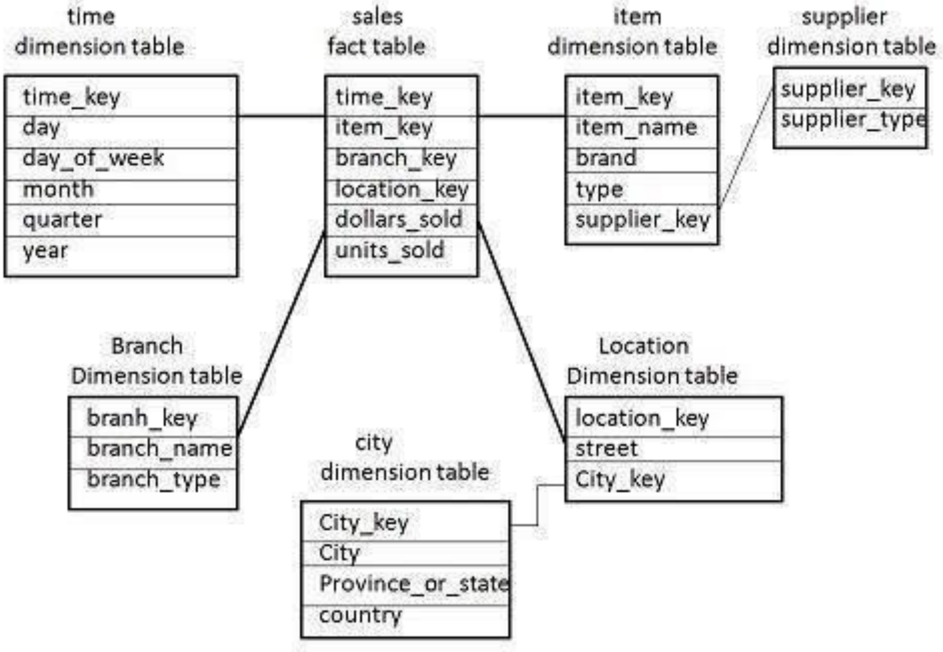


Fig: Snowflake Schema

Advantage of Snowflake Schema:

1. Improved query performance due to smaller dimension tables.
2. It provides greater scalability in the interrelationship between dimension levels and components.
3. No redundancy, so it is easier to maintain.

Disadvantage of Snowflake Schema

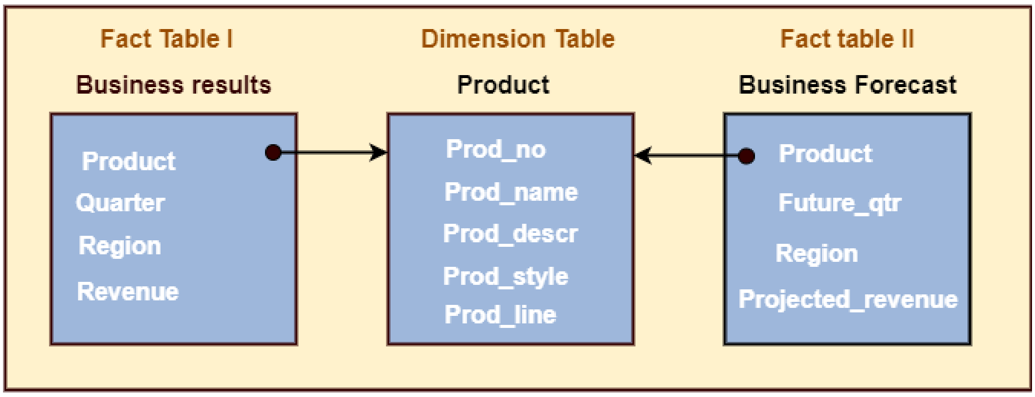
1. Additional maintenance efforts required due to the increasing number of dimension tables.
2. There are more complex queries and hence, difficult to understand.
3. More tables more join so more query execution time.

Difference between Star and Snowflake Schema:

Parameters	Star Schema	Snowflake Schema
Ease of Maintenance	It has redundant data and hence **less easy** to maintain	No redundancy and therefore **easier** to maintain
Ease of change	It has redundant data and hence **less easy** to change	No redundancy and therefore **easier** to change
Ease of Use	Less complex queries and **simple** to understand	More complex queries and therefore less easy to understand **(complex)**
Parent table	In a star schema, a dimension table will **not have** any parent table	In a snowflake schema, a dimension table will have **one or more** parent tables
Execution Time	Less number of foreign keys and hence **lesser** query execution time	More foreign keys and thus **more** query execution time
Normalization	It has **De-normalized tables**	It has **normalized tables**
Type of Data Warehouse	Good for **data marts** with simple relationships (one to one or one to many)	Good to use for **data warehouse** core to simplify complex relationships (**many to many**)
Joins	**Fewer** joins Higher	**Higher** number of joins
Dimension Table	It contains only a **single dimension table for each dimension**	It may have **more than one-dimension table for each dimension**
Foreign keys used	**Less** (due to single dimension table)	**More** (due to one or more dimension tables)
When to use	When the dimensional table **contains less number of rows**, we can go for Star schema.	When dimensional table store a **huge number of rows** with redundancy information and space is such an issue, we can choose snowflake schema to store space.

3. Fact Constellation Schema:

A Fact constellation means two or more fact tables sharing one or more dimensions. It is also called **Galaxy schema**.



FACT Constellation Schema

Fact Constellation Schema is a sophisticated (advanced but difficult to understand) database design that is difficult to summarize information. Fact Constellation Schema can implement between aggregate Fact tables.

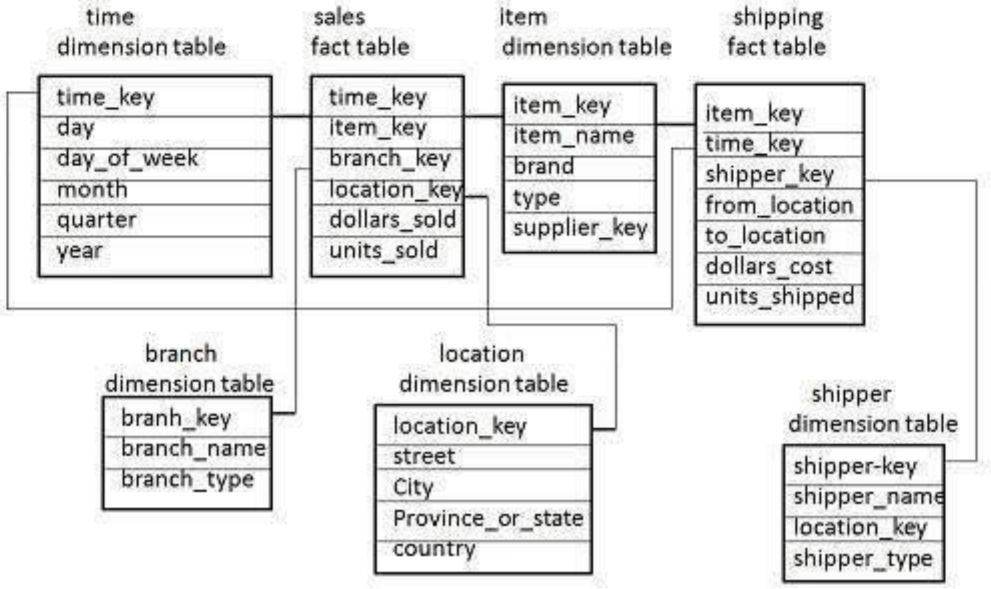


Fig: Fact Constellation Schema

Advantages of Fact Constellation Schema:

1. Provides flexible schema design.
2. One or more fact tables shares one or more-dimension tables.

Disadvantages of Fact Constellation Schema:

1. More complex as aggregation of fact and dimension tables.
2. Hard to implement and maintain.

*Need of OLAP and OLAP Guidelines:

OLAP supports multidimensional view of data.

Provides fast and steady access to various views of information.

Processes complex queries.

Easy to analyse the information.

Advantages of OLAP:

1. Pre-calculate and pre-aggregate the data.
2. OLAP is a platform for all type of business includes planning, budgeting, reporting, and analysis.
3. Information and calculations are consistent in an OLAP cube.
4. Quickly create and analyze "What if" scenarios
5. Easily search OLAP database for broad or specific terms.
6. OLAP provides the building blocks for business modeling tools, Data mining tools, performance reporting tools.
7. Allows users to do slice and dice cube data all by various dimensions, measures, and filters.
8. It is good for analysing time series.
9. Finding some clusters and outliers is easy with OLAP.
10. It is a powerful visualization online analytical process system which provides faster response times

Disadvantages of OLAP:

1. OLAP requires organizing data into a star or snowflake schema. These schemas are complicated to implement.
2. Cannot have large number of dimensions in a single OLAP cube.
3. Transactional data cannot be accessed with OLAP system.
4. Any modification in an OLAP cube needs a full update of the cube. This is a time-consuming process.

*OLAP operations:

Four types of analytical operations in OLAP are:

1. Roll-up
2. Drill-down
3. Slice and dice
4. Pivot (rotate)

1. Roll-up:

Roll-up is also known as "consolidation" or "aggregation." The Roll-up operation can be performed in 2 ways

a. Reducing dimensions
b. Climbing up concept hierarchy. Concept hierarchy is a system of grouping things based on their order or level.

Consider the following diagram:

In this overview section, roll-up operation performed by climbing up (merging) in concept hierarchy of **Location dimension (City to State)**

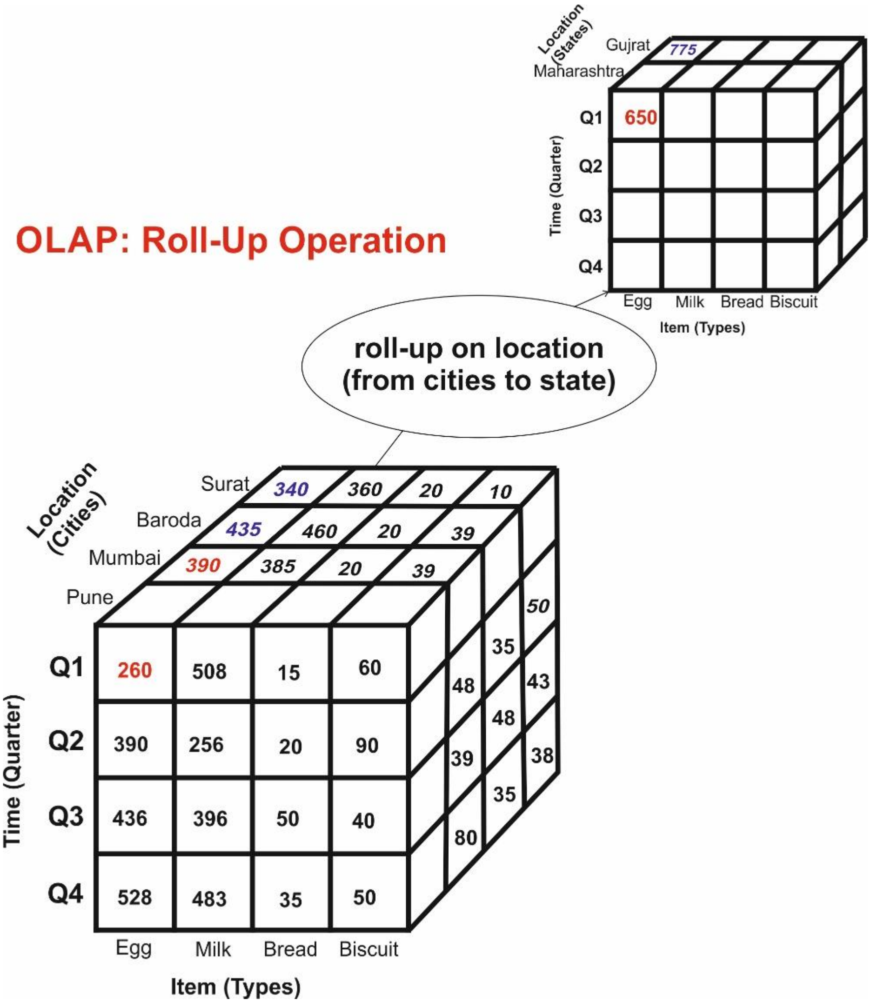


Fig: OLAP Roll-up operation

- In this example, cities Pune and Mumbai are rolled up into State Maharashtra.
- The sales figure of Pune and Mumbai are 260 and 390 respectively. They become 650 after roll-up.
- In this aggregation process, data is location hierarchy moves up from city to the state.

2. Drill-down:

In drill-down data is fragmented (divided) into smaller parts. It is the opposite of the rollup process. It can be done via

a. Moving down in the concept hierarchy.

b. Increasing a dimension.

Consider the following diagram:

In this overview section, drill-down operation is performed by moving down in concept hierarchy of **Time dimension (Quarter to Months).**

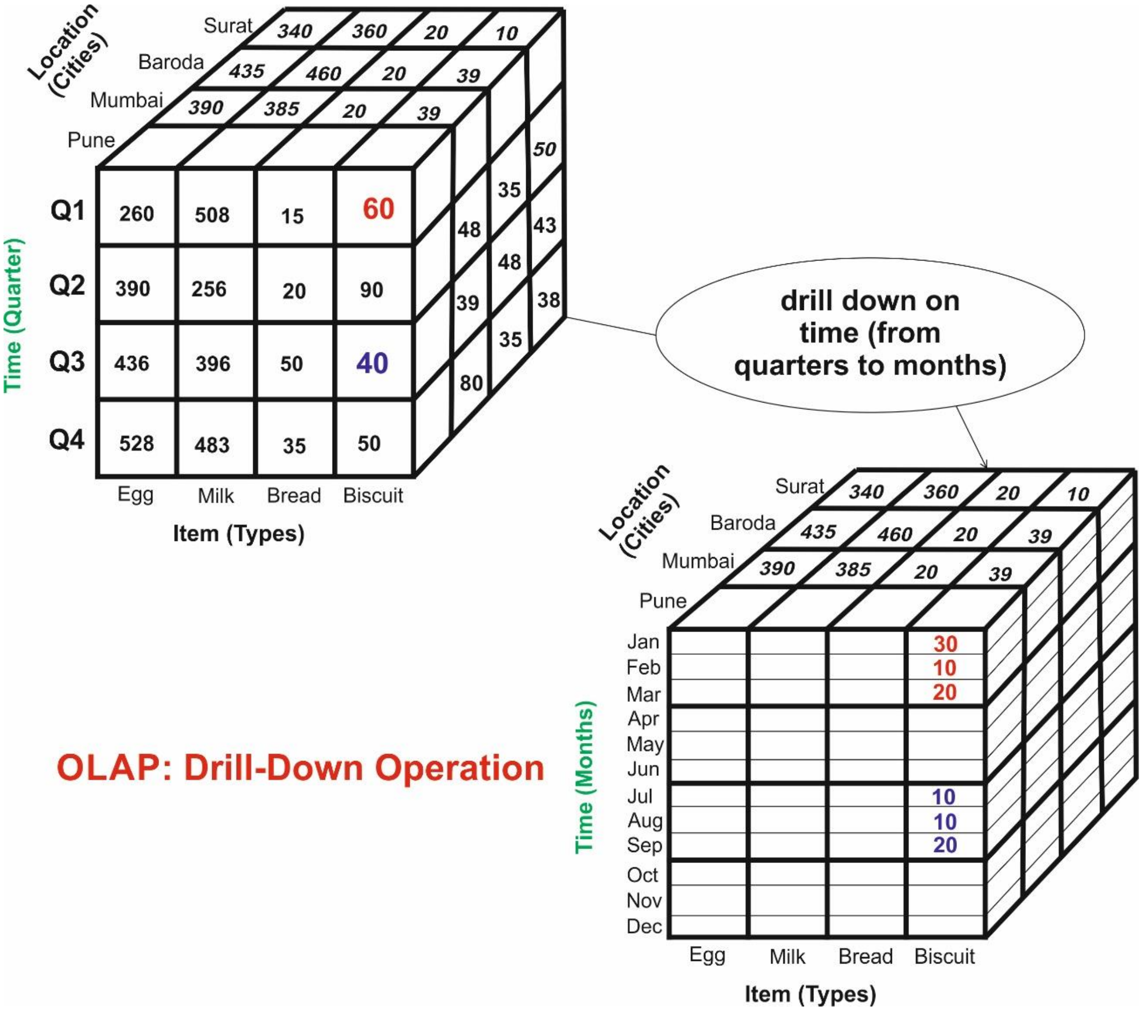


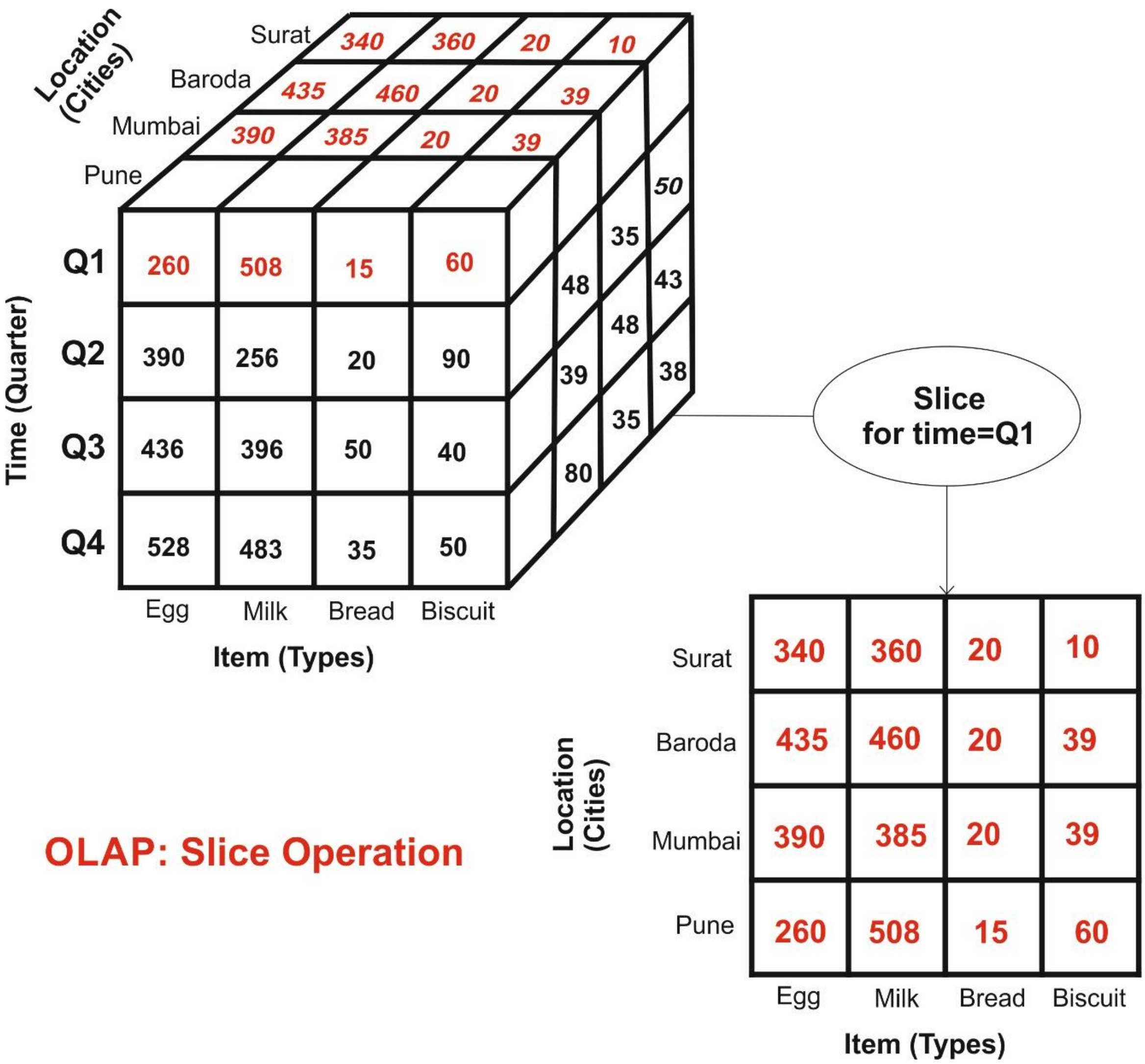
Fig: OLAP Drill-down operation

In this example, Quarter Q1 is drilled down to months January, February, and March. Corresponding sales are also registered. i.e. dimension months are added.

3. Slice:

In this operation, one dimension is selected, and a new sub-cube is created.

In the overview section, slice is performed on the dimension **Time (Q1).**



	Egg	Milk	Bread	Biscuit
Q1	260	508	15	60
Q2	390	256	20	90
Q3	436	396	50	40
Q4	528	483	35	50

	Egg	Milk	Bread	Biscuit
Surat	340	360	20	10
Baroda	435	460	20	39
Mumbai	390	385	20	39
Pune	260	508	15	60

Fig: OLAP Slice operation

In this example, dimension Time is Sliced with quarter Q1 as the filter.

A new cube is created altogether.

4. Dice:

This operation is similar to a slice.

The difference in dice is that, you can select 2 or more dimensions that result in the creation of a sub-cube.

In the overview section, a sub-cube is selected by selecting **Location Pune Or Mumbai and Time Q1 or Q2.**

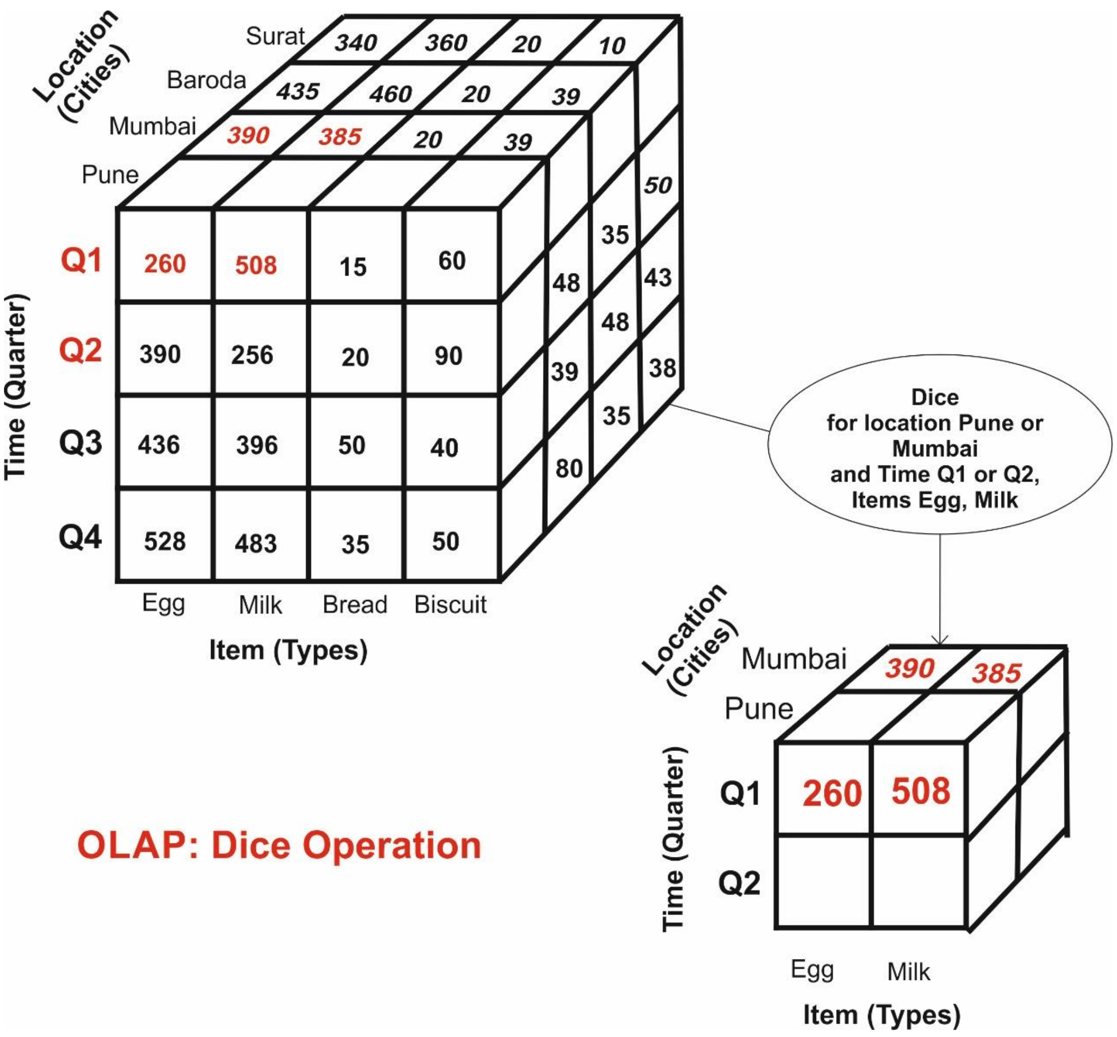
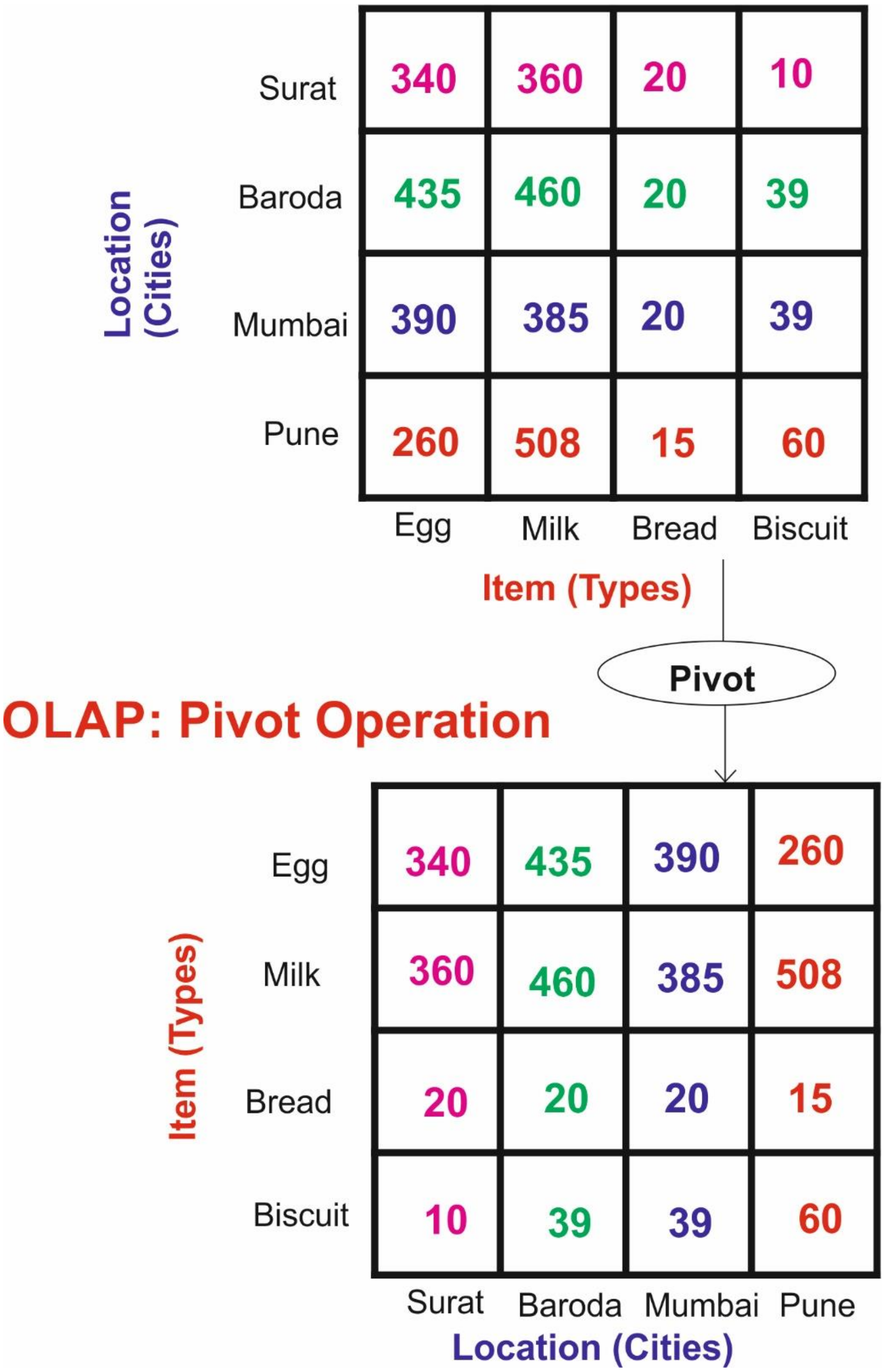


Fig: OLAP Dice operation

5. Pivot

In Pivot operation, you rotate the data axes to provide a substitute presentation of data.

In this overview section, a sub-cube obtained after Slice operation performing Pivot operation gives a new view of that slice.



Location (Cities)	Egg	Milk	Bread	Biscuit
Surat	340	360	20	10
Baroda	435	460	20	39
Mumbai	390	385	20	39
Pune	260	508	15	60

Item (Types)	Surat	Baroda	Mumbai	Pune
Egg	340	435	390	260
Milk	360	460	385	508
Bread	20	20	20	15
Biscuit	10	39	39	60

Fig: OLAP Pivot operation

Chapter 2: Assignment Questions

2 Marks Questions:

1. Define OLAP cube.
2. What is the need of OLAP?
3. Enlist different OLAP operations.
4. Explain any two characteristics of Star schema.

4/6 Marks Questions:

1. Differentiate between Star schema and Snowflake schema.
2. Define OLAP. Explain OLAP cube with diagram.
3. Explain multidimensional data modelling with example.
4. Explain star and snowflake schema with diagram.
5. Explain different OLAP operations with neat sketch.
6. Explain advantages of OLAP.
7. Consider Hospital Datawarehouse consisting of
   Three dimensions: Doctor, Patient, Time and
   Two measure: Count, fees.
   Using above data describe different OLAP operations.

Unit 3: Data Warehousing Designing & OLAP II (18 Marks)

*Data Warehouse Design:

A data warehouse is a single data repository where a record from multiple (heterogeneous) data sources is integrated for online business analytical processing (OLAP).

Thus, data warehouse design is a hugely complex, lengthy, and hence error-prone process. Furthermore, business analytical functions change over time, which results in changes in the requirements for the systems. Therefore, data warehouse and OLAP systems are dynamic, and the design process is continuous.

The target of the design is how the record from multiple data sources should be extracted, transformed, and loaded (ETL) to be organized in a database as the data warehouse.

*Business Analysis Framework for Data Warehouse Design:

The business analyst gets the information from the data warehouses to measure the performance and make critical adjustments in order to win over other business holders in the market.

Having a data warehouse offers the following advantages −

- Since a data warehouse can gather information quickly and efficiently, it can enhance business productivity.
- A data warehouse provides us a consistent view of customers and items; hence, it helps us manage customer relationship.
- A data warehouse also helps in bringing down the costs by tracking trends, patterns over a long period in a consistent and reliable manner.

To design an effective and efficient data warehouse, we need to understand and analyze the business needs and construct a **business analysis framework**. Each person has different views regarding the design of a data warehouse. These views are as follows −

- **The top-down view** − This view allows the selection of relevant information needed for a data warehouse.
- **The data source view** − This view presents the information being captured, stored, and managed by the operational system.
- **The data warehouse view** − This view includes the fact tables and dimension tables. It represents the information stored inside the data warehouse.
- **The business query view** − It is the view of the data from the viewpoint of the end-user.

*Data Warehouse Design Process:

Three methods:

1. Top-down and Bottom-up Approach
2. Software Engineering Model
3. Typical Design Process

1. Top-down and Bottom-up Approach:

a. Top-down Approach:

An approach is a data-driven approach as the information is gathered and integrated first and then business requirements by subjects for building data marts are formulated.

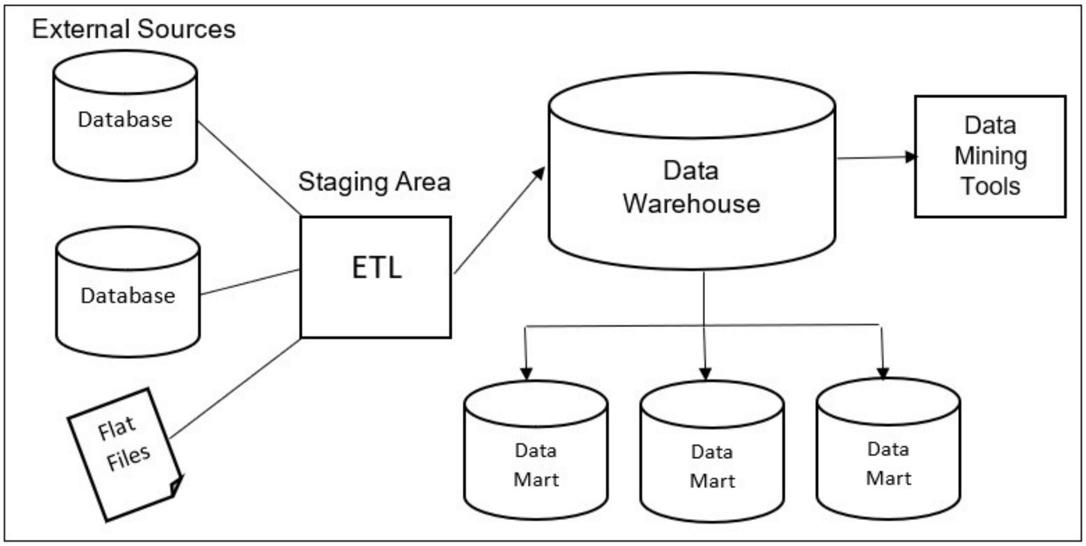


Fig: DW Design: Top-Down Approach

i. **External Sources:**
External source is a source from where data is collected irrespective of the type of data. Data can be structured, semi structured and unstructured as well.

ii. **Stage Area:**
Since the data, extracted from the external sources does not follow a particular format, so there is a need to validate this data to load into datawarehouse.
For this purpose, it is recommended to use **ETL** tool.
- **E(Extracted):** Data is extracted from External data source.
- **T(Transform):** Data is transformed into the standard format.
- **L(Load):** Data is loaded into datawarehouse after transforming it into the standard format.

iii. **Data-warehouse:**

After cleansing of data, it is stored in the datawarehouse as central repository.

It actually stores the meta data and the actual data gets stored in the data marts.

iv. **Data Marts:**

Data mart is also a part of storage component (subset of Data Warehouse).

It stores the information of a particular function of an organisation which is handled by single authority. There can be as many numbers of data marts in an organisation depending upon the functions.

v. **Data Mining:**

It is used to find the hidden patterns that are present in the database or in datawarehouse with the help of algorithm of data mining.

Advantages of Top-Down Approach:

1. Since the data marts are created from the datawarehouse, provides consistent dimensional view of data marts.
2. This model is considered as the strongest model for business changes.
3. Creating data mart from datawarehouse is easy.

Disadvantages of Top-Down Approach:

1. The cost, time taken in designing and its maintenance is very high.
2. Consumes more time for analysis of business data.

b. Bottom-Up approach:

In this approach, a data mart is created first for particular business processes (or subjects).

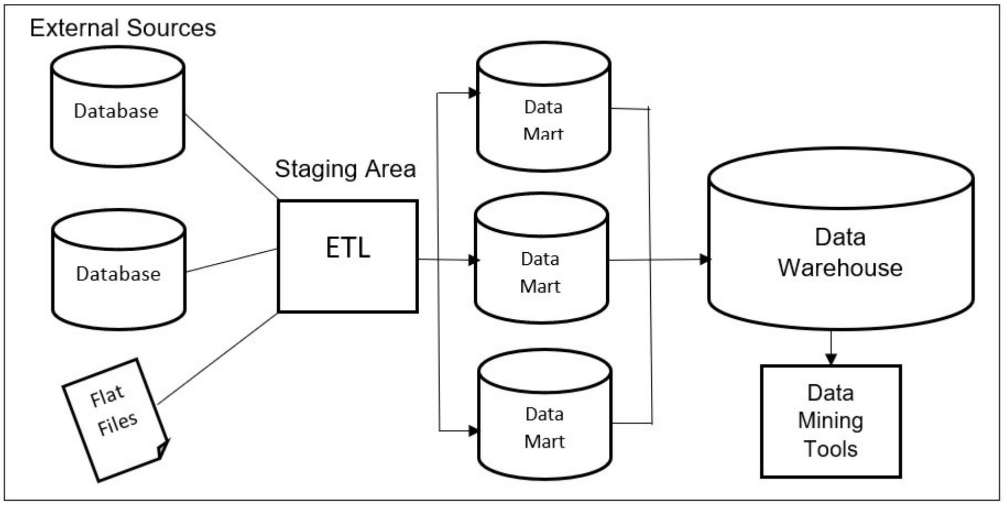


Fig: DW Design: Bottom-Up Approach

1. First, the data is extracted from external sources.
2. Then, the data go through the staging area and loaded into data marts instead of datawarehouse.
3. The data marts are created first and provide reporting capability. It addresses a single business area.
4. These data marts are then integrated into datawarehouse.

Advantages:

1. As the data marts are created first, so the reports are quickly generated.
2. We can accommodate a greater number of data marts here and in this way datawarehouse can be extended.
3. Also, the cost and time taken in designing this model is low comparatively.

Disadvantages:

1. This model is not strong as top-down approach as dimensional view of data marts is not consistent as it is in above approach.
2. Creating data marts prior to data warehouse is complex.

Difference between Top-Down and Bottom-Up Approach:

Parameters	Top-Down Approach	Bottom-Up Approach
Definition	Firstly, data warehouse is designed and then data marts.	Firstly, data marts are created and integrating data marts data warehouse is created.
Function	Breaks big problem into smaller one	Solve sub problems and integrate them into higher one
Type of Data Stored	Centralized	Department wise
Rules applied	Centralized	Departmental
Redundancy	Yes	No
Quick data analysis	No	Yes
Risk of Failure	More	Less
Maintenance	High	Low
Popularity	More	Less

2. Software Engineering Model for DW Design Process:

8 Steps to Designing a Data Warehouse

a. Defining Business Requirements (or Requirements Gathering):

Gathering requirements is first step of the data warehouse design process.

The goal of the requirements gathering phase is to determine the criteria for a successful implementation of the data warehouse. It is a blueprint of data warehouse.

This **Requirements Gathering stage** should focus on the following objectives.

Aligning department goals with the overall project

Determining the scope of the project in relation to business objectives

Creating a disaster recovery plan in the case of system failure

Thinking about each layer of security.

b. Physical Environment Setup:

Once the business requirements are set, the next step is to determine the physical environment for the data warehouse.

There should be separate physical application and database servers as well as separate ETL/ELT, OLAP, cube, and reporting processes set up for development, testing, and production environments.

Building separate physical environments ensure that all changes can be tested before moving them to production, development, and testing can occur without halting the production environment.

c. Data Modeling (Design):

Once requirement gathering and physical environments have been defined, the next step is to define how data structures will be accessed, connected, processed, and stored in the data warehouse. This process is known as data modeling.

Data modeling is the process of visualizing data distribution in your warehouse i.e blueprint.

The three most popular data models (schemas) for warehouses are:

Snowflake Schema, Star Schema, Galaxy Schema (fact constellation)

d. Extract, Transfer, Load (ETL) Solution:

ETL or Extract, Transfer, Load is the process used to pull data out of existing data sources and put it into warehouse.

Identifying data sources during the data modeling phase may help to reduce ETL development time.

The goal of ETL is to provide quick response time without sacrificing quality.

Failure at this stage of the process can lead to poor performance of entire data warehouse system.

e. Online Analytic Processing (OLAP) Cube:

On-Line Analytical Processing (OLAP) is the answer engine that provides multi-dimensional analysis to business users.

The three critical elements of OLAP design include:

i. Grouping measures: numerical values to analyze (charges and count).

ii. Dimension: describes all information of measures.

iii. Granularity: the lowest level of detail that can include in the OLAP dataset.

During development, OLAP cube gets little time to update after data warehouse Updation. So, not updating OLAP cube in a timely manner could lead to reduced system performance.

f. Creating the Front End:

At this point, business requirements have been captured, physical environment complete, data model decided, and ETL process has been documented.

The next step is to work on how users will access the data warehouse.

Using front end, users will access the data for analysis and run reports.

g. Optimizing Queries:

Optimizing queries is a complex process.

There are some general strategies for optimizing queries:

i. Ensure that production, testing, and development environment have mirrored resources. This will prevent the server from hanging when user push the projects from one environment to the next.

ii. Try to minimize data retrieval.

Don't run SELECT on the whole database if you only need a column of results. Instead, run your SELECT query on specific columns.

iii. Understand the limitations of OLAP provider.

h. Establishing a Rollout: (Deployment)

Deciding to make the system available to everyone, will depend on the number of end users and how they will access the data warehouse system.

Before deploying, end users training should be required.

If the actual end user finds the tool difficult to use, or do not understand the benefits of using the data warehouse for reporting and analysis, it results in poor performance.

3. Typical process of Data Warehouse Design:

a. Choose a **business process** to model.

If business process is an organizational choose the data warehouse.

If process is a departmental choose DataMart.

b. Choose the **grain** of business process model.

Fundamental details of data to be represented in fact table.

c. Choose the **dimensions** that will apply to each fact table record.

The typical dimensions like time, location, item, etc.

d. Choose the **measure** that will populate each fact table.

Typical measures (numeric values) are charges and count.

*DW for Information Processing:

Data warehouses and data marts are used in a wide range of applications. Business executives use the data in data warehouses and data marts to perform data analysis and make strategic decisions.

Data warehouses are used extensively in banking and financial services, consumer goods and retail distribution sectors, and controlled manufacturing such as demand-based production.

Initially, the data warehouse is mainly used for generating reports and answering predefined queries. Progressively, it is used to analyze summarized and detailed data, where the results are presented in the form of reports and charts. Later, the data warehouse is used for strategic purposes, performing multidimensional analysis and sophisticated slice-and-dice operations. Finally, the data warehouse may be employed for knowledge discovery and strategic decision-making using data mining tools.

The tools for data warehousing can be categorized into *access and retrieval tools*, *database reporting tools*, *data analysis tools*, and *data mining tools*.

There are three kinds of data warehouse applications: *information processing, analytical processing*, and *data mining*.

- **Information Processing** – A data warehouse allows to process the data stored in it. The data can be processed by means of querying, basic statistical analysis, reporting using crosstabs, tables, charts, or graphs.
- **Analytical Processing** – A data warehouse supports analytical processing of the information stored in it. The data can be analysed by means of basic OLAP operations, including slice-and-dice, drill down, drill up, and pivoting.
- **Data Mining** – Data mining supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction. These mining results can be presented using the visualization tools.

*From Online Analytical Processing to Multidimensional Data Mining:

The data mining field has conducted massive research regarding mining on various data types, including relational data, data from data warehouses, transaction data, time-series data, spatial data, text data, and flat files.

Multidimensional data mining integrates OLAP with data mining to uncover knowledge in multidimensional databases.

Multidimensional data mining is particularly important for the following reasons:

1. High quality of data in data warehouses:

Most data mining tools need to work on integrated, consistent, and cleaned data.

A data warehouse constructed by such preprocessing serves as a valuable source of high-quality data for OLAP as well as for data mining.

2. Available information processing infrastructure surrounding data warehouses:

Comprehensive information processing and data analysis infrastructures have been constructed surrounding data warehouses, which includes accessing, integration, consolidation, and transformation of multiple heterogeneous databases, reporting and OLAP analysis tools.

It is sensible to make the best use of the available infrastructures rather than constructing everything from scratch.

3. OLAP-based exploration of multidimensional data:

Effective data mining needs exploratory data analysis.

Multidimensional data mining provides facilities for mining on different subsets of data and at varying levels of abstraction by drilling, pivoting, filtering, dicing, and slicing on a data cube.

4. Online selection of data mining functions:

By integrating OLAP with various data mining functions, multidimensional data mining provides users with the flexibility to select desired data mining functions and swap data mining tasks dynamically.

*Data Warehouse Implementation: Efficient Data Cube Computation:

Need of Data Cube Computation:

- To retrieve the info from the data cube in most efficient way possible.
- Queries run on the cube will be fast.

Multidimensional data analysis mainly depends on the efficient computation of aggregations across many sets of dimensions.

In SQL terms, these aggregations are referred to as group-by's. Each group-by can be represented by a *cuboid*, where the set of group-by's forms a lattice of cuboids defining a data cube.

A data cube is a lattice of cuboids.

Suppose that you want to create a data cube for *AllElectronics* sales that contains the following: *city, item, year*, and *sales in dollars*. You want to be able to analyze the data, with queries such as the following:

"*Compute the sum of sales, grouping by city and item.*"

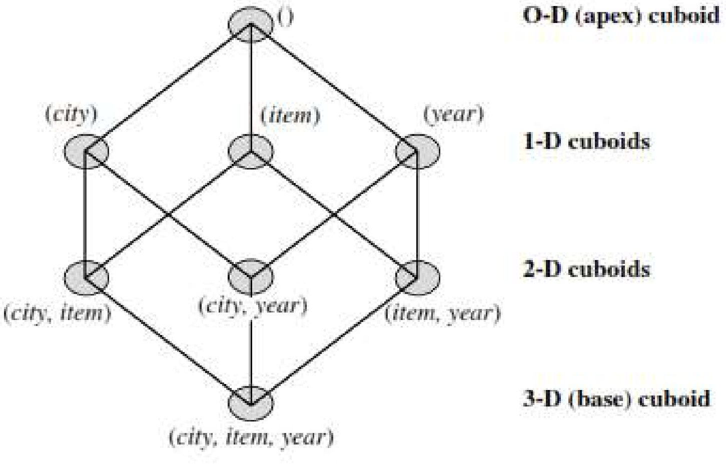
"*Compute the sum of sales, grouping by city*."

"*Compute the sum of sales, grouping by item.*"

Taking the three attributes, *city, item*, and *year*, as the dimensions for the data cube, and *sales in dollars* as the measure, the total number of cuboids, or groupby's, that can be computed for this data cube is $2^{3=}8$.

The possible group-by's are the following: {(*city, item, year*), (*city, item*), (*city, year*), (*item, year*), (*city*), (*item*), (*year*), ()} where () means that the group-by is empty (i.e., the dimensions are not grouped).

These group-by's form a lattice of cuboids for the data cube, as shown in Figure



The **base cuboid** contains all three dimensions, *city, item*, and *year*. It can return the total sales for any combination of the three dimensions.

The **apex cuboid**, or 0-D cuboid, refers to the case where the group-by is empty. It contains the total sum of all sales.

The base cuboid is the least generalized (most specific) of the cuboids.

The apex cuboid is the most generalized (least specific) of the cuboids, and is often denoted as all.

Materialization (Precomputation of Data Cube):

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

1. No materialization:

Do not precompute cuboids. This leads to computing expensive multidimensional aggregates, which can be extremely slow.

2. Full materialization:

Precompute all of the cuboids.

The resulting lattice of computed cuboids is referred to as the *full cube*.

This choice typically requires huge amounts of memory space in order to store all of the precomputed cuboids.

Full Cube Computation:

Multi way aggregation method used to compute full data cube.

Aggregation done by:

- Partition array into chunks.

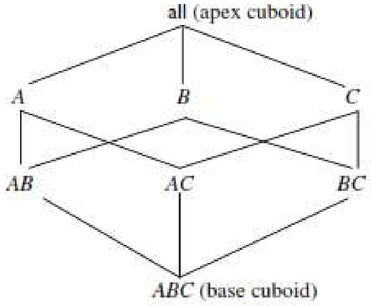
  Partition the array into chunks. A chunk is a subcube that is small enough to fit into the memory available for cube computation.

- Compute aggregate by visiting cube cells.

  The order in which cells are visited can be optimized so as to minimize the number of times that each cell must be revisited, thereby reducing memory access and storage costs.

Consider a 3-D data array containing the three dimensions *A*, *B*, and *C*. The 3-D array is partitioned into small, memory-based chunks. In this example, the array is partitioned into 64 chunks. Dimension *A* is organized into four equal-sized partitions: *a*0, *a*1, *a*2, and *a3*.

Dimensions *B* and *C* are similarly organized into four partitions each. Chunks 1, 2, . . . , 64 correspond to the subcubes *a*0*b*0*c*0, *a*1*b*0*c*0, . . . , *a*3*b*3*c*3, respectively.



Full materialization of the corresponding data cube involves the computation of all the cuboids defining this cube. The resulting full cube consists of the following cuboids:

a. The base cuboid, denoted by *ABC* (from which all the other cuboids are directly or indirectly computed). This cube is already computed and corresponds to the given 3-D array.
b. The 2-D cuboids, *AB*, *AC*, and *BC*, which respectively correspond to the group-by's *AB*, *AC*, and *BC*. These cuboids must be computed.
c. The 1-D cuboids, *A*, *B*, and *C*, which respectively correspond to the group-by's *A*, *B*, and *C*. These cuboids must be computed.
d. The 0-D (apex) cuboid, denoted by all, which corresponds to the group-by (); that is, there is no group-by here. This cuboid must be computed. It consists of only one value. If, say, the data cube measure is count, then the value to be computed is simply the total count of all the tuples in *ABC*.

Advantage: queries run on cube will be very fast.
Disadvantage: Precomputed cube requires a lot of memory.

3. Partial materialization:
Selectively compute a proper subset of the whole set of possible cuboids.
Compute a subset of the cube, which contains only those cells that satisfy some user-specified criterion.
It uses subcube where only some of the cells may be precomputed for various cuboids.
Iceberg Cube Computation:
Contains only those cells of data cube that meet an aggregate.
It is called iceberg, as it contains only some cells of full cube (tip of an iceberg)
Purpose of this cube is to identify and compute those values which are required for decision support queries.

Iceberg cube can be specified with an SQL query, as shown

compute cube *sales iceberg* as

select *month*, *city*, *customer_group*, count(*)

from *salesInfo*
cube by *month*, *city*, *customer_group*
having count(*) >= *min_sup*

Advantages: precompute only those cells in cube which are required for decision support queries.

Strategies for Data cube computation:

a. Sorting hashing and grouping.
b. Simultaneous aggregation and caching intermediate results.
c. Aggregation from smallest child where there exist multiple child cuboid.
d. The Apriori pruning method can be explored to compute iceberg cube efficiently.

a. Sorting, hashing and grouping:

In cube computation, aggregation is performed on the tuples (or cells) that share the same set of dimension values.

Thus, it is important to explore sorting, hashing, and grouping operations to access and group such data together to facilitate computation of such aggregates.

Ex: To compute total sales by *branch*, *day*, and *item*, for example, it can be more efficient to sort tuples or cells by ***branch***, and then by ***day***, and then group them according to the ***item*** name.

b. Simultaneous aggregation and caching intermediate results:

In cube computation, it is efficient to compute higher-level aggregates from previously computed lower-level aggregates, rather than from the base fact table.

Simultaneous aggregation from cached intermediate computation results may lead to the reduction of expensive disk input/output (I/O) operations.

Ex: To compute sales by *branch*, for example, we can use the intermediate results derived from the computation of a lower-level cuboid such as sales by *branch* and *day*.

c. Aggregation from the smallest child:

If a parent cuboid has more than one child, it is efficient to compute it from the smallest previously computed child cuboid.

Ex: To compute a sales cuboid, Cbranch, when there exist two previously computed cuboids, C{branch,year} and c{branch,item}, it is obviously more efficient to compute Cbranch from the former than from the latter if there are many more distinct items than distinct years.

d. The Apriori pruning (trimming unwanted) method:

Apriori requires a priori knowledge to generate the frequent itemsets and involves two time-consuming pruning steps to exclude the infrequent candidates and hold frequents.
It is used to reduce the computation of iceberg cubes.

Ex: The original database

TID	A	B	C	D	E
1	a1	b1	c1	d1	e1
2	a1	b2	c1	d2	e1
3	a1	b2	c1	d1	e2
4	a2	b1	c1	d1	e2
5	a2	b1	c1	d1	e3

Inverted Index:

Attribute Value	Tuple ID List	List Size
a1	{1,2,3}	3
a2	{4,5}	2
b1	{1,4,5}	3
b2	{2,3}	2
c1	{1,2,3,4,5}	5
d1	{1,3,4,5}	4
d2	{2}	1
e1	{1,2}	2

Cuboid of AB (pruning method: only considers the frequent items)

Cell	Intersection	Tuple ID List	List Size
(a1,b1)	{1,2,3} ∩ {1,4,5}	{1}	1
(a1,b2)	{1,2,3} ∩ {2,3}	{2,3}	2
(a2,b1)	{4,5} ∩ {1,4,5}	{4,5}	2
(a2,b2)	{4,5} ∩ {2,3}	{}	0

*Indexing OLAP Data:

1. Bitmap Index:

The bitmap index is an alternative representation of the record ID (RID) list.

Each attribute is represented by distinct bit value.

If attribute's domain consists of n values, then n bits are needed for each entry in the bitmap index.

If the attribute value is present in the row then it is represented by 1 in the corresponding row of the bitmap index and rest are 0 (zero).

Example:

Base Table

Cust_ID	Region	Type
C1	Asia	Retail
C2	Europe	Dealer
C3	Asia	Retail
C4	America	Dealer
C5	Europe	Dealer

Base table mapping to bitmap index tables for dimensions Region and Type are:

Index on Region

RecID	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	1	0
4	0	1
5	0	1

Advantages of Bitmap Index:

- Reduced response time for large classes.
- Reduced storage requirements compared to other indexing techniques.
- Performance gains even on minimum hardware and small amount of memory.
- Efficient maintenance during parallel DML and loads.

Example: Bitmap Index

Company's customers table.

SELECT cust_id, cust_gender, cust_income FROM customers;

Cust_id	Cust_gender	Cust_income
101	M	10000
102	F	20000
103	M	15000
104	F	21000
105	F	11000

Because cust_gender and cust_income are all low-cardinality columns (there are two possible values for gender and 12 for income), bitmap indexes are ideal for these columns. Do not create a bitmap index on cust_id because this is a unique column.

The following table illustrates the bitmap index for the cust_gender column in this example. It consists of two separate bitmaps, each one for gender.

Sample Bitmap Index

RID	Gender F	Gender M
1	0	1
2	1	0
3	0	1
4	1	0
5	1	0

2. Bitmap Join Indexes:

In addition to a bitmap index on a single table, we can create a bitmap join index, which is a bitmap index for the join of two or more tables.

In a bitmap join index, the bitmap for the table to be indexed is built for values coming from the joined tables.

Example: Bitmap Join Index:

A bitmap join index on the fact table "sales" for the joined column customers (cust_gender).

Table sales must contain cust_id values.

SELECT time_id, cust_id, amount_sold FROM sales;

Time_id	Cust_id	Amount_sold
Jan	101	2000
Feb	103	3000
Mar	106	5000
Apr	104	6000
May	107	7000

The following query illustrates the join result that is used to create the bitmaps that are stored in the bitmap join index:

Customer Table

Cust_id	Cust_gender	Cust_income
101	M	10000
102	F	20000
103	M	15000
104	F	21000
105	F	11000

```
SELECT sales.time_id, customers.cust_gender, sales.amount_sold
FROM sales, customers
WHERE sales.cust_id = customers.cust_id;
```

Time_id	Cust_gender	Amount_sold
Jan	M	2000
Feb	M	3000
Apr	F	6000

Sample Bitmap Join Index

RID	Cust_gender M	Cust_gender F
1	1	0
2	1	0
3	0	1

*Efficient Processing of OLAP Queries:

The purpose of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes.

By using materialized views, query processing should proceed as follows:

1. Determine which operations should be performed on the available cuboids:

Operations like transforming any selection, projection, roll-up (group-by), and drill-down operations specified in the query into corresponding SQL and/or OLAP operations.

2. Determine the materialized cuboid(s) and its relevant operations:

This involves identifying all of the materialized cuboids that may potentially be used to answer the query.

Example OLAP query processing:

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*

"*itemname* < *brand* < *type*" for *item*

"*street* < *city* < *state* < *country*" for *location*.

Suppose that the query to be processed is on *brand, state*, with the selection constant "*year = 2010*."

There are four materialized cuboids available for processing, as follows:

cuboid 1: {*year, itemname, city*}

cuboid 2: {*year, brand, country}*

cuboid 3: {*year, brand, state*}

cuboid 4: {*itemname, state, year}*

Task is to identify the cuboids from above available, which provides finer granularity data.

Therefore, cuboid 2 cannot be used because *country* is a more general concept than *state*.

Cuboids 1, 3, and 4 can be used to process the query.

*OLAP Server Architectures:

1. Relational OLAP (ROLAP)
2. Multidimensional OLAP (MOLAP)
3. Hybrid OLAP (HOLAP)

1. Relational OLAP (ROLAP):

ROLAP servers are placed between relational back-end server and client front-end tools.
To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.
ROLAP works directly with relational databases and does not require pre-computation.
ROLAP is relational OLAP where the data is arranged in traditional methods like rows and columns in the data warehouse.
ROLAP tools do not use pre-calculated data cubes but instead fire the query to the standard relational database.
ROLAP includes the following components:

- Database server
- ROLAP server
- Front-end tool.

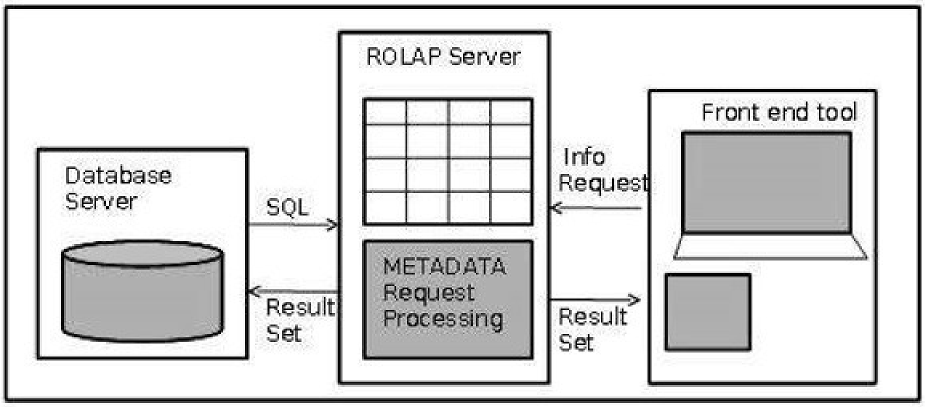


Fig: ROLAP Architecture

Advantages:

I. ROLAP servers are highly scalable.
II. ROLAP tools analyze large volumes of data across multiple dimensions.
III. ROLAP tools store and analyze highly volatile and changeable data.
IV. ROLAP servers can be easily used with existing RDBMS.
V. Data can be stored efficiently, since no zero facts can be stored.
VI. ROLAP tools do not use pre-calculated data cubes.
VII. The data are stored in a standard relational database and can be accessed by any SQL reporting tool.

2. Multidimensional OLAP (MOLAP):

MOLAP uses multidimensional storage units for multidimensional views of data.

MOLAP sometimes referred to as just OLAP (Data Cube).

The data cube contains all the possible answers to a given range of questions. As a result, they have a very fast response to queries.

MOLAP includes the following components:

- Database server.
- MOLAP server.
- Front-end tool.

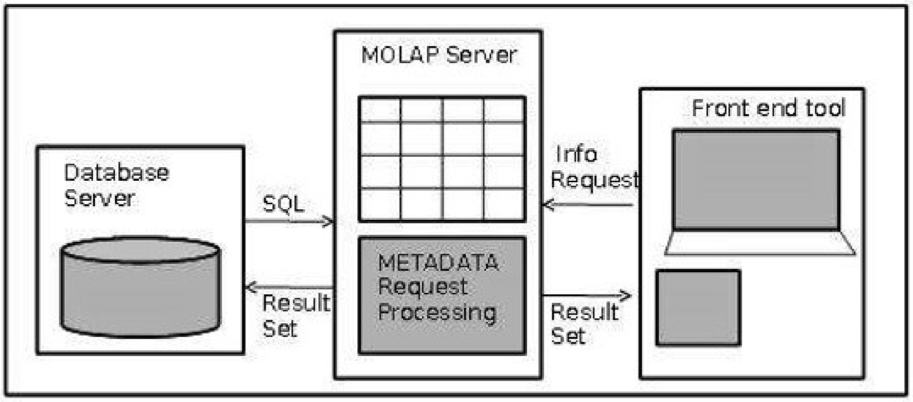


Fig: MOLAP Architecture

Advantages:

I. MOLAP allows fastest indexing to the pre-computed summarized data.
II. Helps the users connected to a network who need to analyze larger, less-defined data.
III. Easier to use, therefore MOLAP is suitable for inexperienced users.
IV. Fast query performance.
V. Automated computation of higher-level aggregates of the data.

3. Hybrid OLAP (HOLAP)

Hybrid OLAP is a combination of both ROLAP and MOLAP.

It offers higher scalability of ROLAP and faster computation of MOLAP.

In this mode HOLAP stores aggregations in MOLAP for fast query performance, and detailed data in ROLAP to optimize time of cube processing.

HOLAP tools can utilize both pre-calculated cubes and relational data sources.

ROLAP Vs MOLAP:

Parameters	ROLAP	MOLAP
Acronym	Relational Online Analytical Processing	Multidimensional Online Analytical Processing
Information retrieval	Slow	Fast
Storage Method	relational table	sparse array to store data-sets
Easy to use	Yes	NO (Data cubes)
When to use	when data warehouse contains relational data	when data warehouse contains relational as well as non-relational data
Implementation	Easy	Complex
Response Time Required	More	Less
Storage Space	Less	More

ROLAP vs MOLAP vs HOLAP:

Basics for comparison	ROLAP	MOLAP	HOLAP
Acronym	Relational online analytical processing	Multi-dimensional online analytical processing	Hybrid online analytical processing
Storage methods	Data is stored on the registered database MDDB	Data warehouse	Data is stored on the relational databases
Fetching methods	Data is fetched from the Proprietary database	Data is fetched from the main repository	Data is fetched from the relational databases
Data Arrangement	Data is arranged and saved in the form of tables with rows and columns	Data is arranged and stored in the form of data cubes	Data is arranged in multi-dimensional form
Amount of data processed	Enormous data is processed	Limited data which is kept in proprietary is processed	Large data can be processed
Technique	SQL	Sparse Matrix technology	both Sparse matrix technology and SQL
Designed view	dynamic	static	dynamic
Response time	It has Maximum response time	It has Minimum response time	It takes Minimum response time

Chapter 3: Assignment Questions

2 Marks Questions:

1. Define ROLAP and MOLAP.
2. Enlist the types of data cube materialization.
3. Give the example of efficient processing of OLAP queries.
4. Define Bitmap index.

4/6 Marks Questions:

5. Describe multidimensional data mining.
6. Explain data warehouse design process.
7. Explain bitmap index and join index.
8. Explain OLAP server architecture.
9. Differentiate between ROLAP and MOLAP.
10. Explain efficient data cube computation with materialization.

Unit 4: Introduction to Data Mining (18 Marks)

*Data Mining:

Data mining means searching for knowledge (interesting patterns) in data.

Data mining refers to extraction of information from large amount of data.

The data sources can include databases, data warehouses, the Web, other information repositories, or data that are streamed into the system dynamically.

Data mining is the *process* of discovering interesting patterns and knowledge from *large* amounts of data.

Data mining is used by companies in order to get customer preferences, determine price of their product and services and to analyse market.

Data mining is also known as knowledge discovery in Database (KDD).

Steps in the process of KDD:

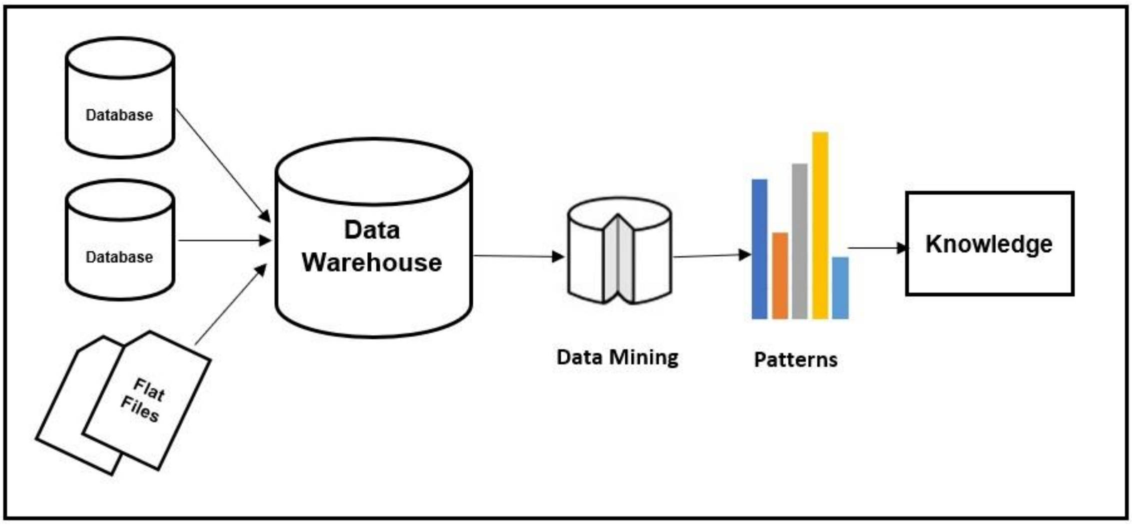


Fig: Steps in KDD Process

1. Data cleaning:

In data cleaning it removes the noise and inconsistent data.

2. Data integration:

Multiple data sources may be combined.

3. Data selection:

The data relevant to the analysis task are retrieved from the database.

4. Data transformation:

The data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations.

i.e. the data from different data sources which is of varied types can be converted into a single standard format.

5. Data mining:

Data mining is the process in which intelligent methods or algorithms are applied on data to extract useful data patterns.

6. Pattern evaluation:

This process identifies the truly interesting patterns representing actual knowledge based on user requirements for analysis.

7. Knowledge presentation:

In this process, visualization and knowledge representation techniques are used to present mined knowledge to users for analysis.

*What kind of data can be mined?

1. **Flat Files:**
    - Flat files are defined as data files in text form or binary form with a structure that can be easily extracted by data mining algorithms.
    - Data stored in flat files have no relationship or path among themselves, like if a relational database is stored on flat file, then there will be no relations between the tables.
    - Flat files are represented by data dictionary. Eg: CSV file.
    - **Application**: Used in DataWarehousing to store data, used in carrying data to and from server, etc.

2. **Relational Databases:**
    - A Relational database is defined as the collection of data organized in tables with rows and columns.
    - Physical schema in Relational databases is a schema which defines the structure of tables.
    - Logical schema in Relational databases is a schema which defines the relationship among tables.
    - Standard API of relational database is SQL.
    - **Application**: Data Mining, ROLAP model, etc.

3. **Data Warehouse**:
    - A datawarehouse is defined as the collection of data integrated from multiple sources that will queries and decision making.
    - There are three types of datawarehouse: **Enterprise** datawarehouse, **Data Mart** and **Virtual** Warehouse.
    - **Application**: Business decision making, Data mining, etc.
4. **Transactional Databases**:
    - Transactional databases are a collection of data organized by time stamps, date, etc to represent transaction in databases.
    - This type of database has the capability to roll back or undo its operation when a transaction is not completed or committed.
    - Highly flexible system where users can modify information without changing any sensitive information.
    - Follows ACID property of DBMS.
    - **Application**: Banking, Distributed systems, Object databases, etc.
5. **Multimedia Databases:**
    - Multimedia databases consists audio, video, images and text media.
    - They can be stored on Object-Oriented Databases.
    - They are used to store complex information in a pre-specified format.
    - **Application**: Digital libraries, video-on demand, news-on demand, musical database, etc.
6. **Spatial Database:**
    - Store geographical information.
    - Stores data in the form of coordinates, topology, lines, polygons, etc.
    - **Application**: Maps, Global positioning, etc.
7. **Time-series Databases:**
    - Time series databases contains stock exchange data and user logged activities.
    - It requires real-time analysis.
    - **Application**: eXtremeDB, Graphite, InfluxDB, etc.
8. **WWW:**
    - WWW refers to **World wide web** is a collection of documents and resources like audio, video, text, etc which are identified by Uniform Resource Locators (URLs) through web browsers, linked by HTML pages, and accessible via the Internet.
    - It is the most heterogeneous repository as it collects data from multiple resources.
    - It is dynamic in nature as Volume of data is continuously increasing and changing.
    - **Application**: Online shopping, Job search, Research, studying, etc.

*Major Issues in Data Mining:

Data mining systems face a lot of challenges and issues like:

A. Mining methodology and user interaction issues
B. Performance issues
C. Issues relating to the diversity of database types

A. Mining methodology and user interaction issues:

i. Mining different kinds of knowledge in databases:

Different user - different knowledge - different way.

That means different client want a different kind of information so it becomes difficult to cover vast range of data that can meet the client requirement.

ii. Incorporation of background knowledge:

Background knowledge is used to guide discovery process and to express the discovered patterns. So, in mining process to know the background of data is must for easy process.

iii. Query languages and ad hoc mining:

Relational query languages allow users to use ad-hoc queries for data retrieval.

The language of data mining query language and the query language of data warehouse should be matched.

iv. Handling noisy or incomplete data:

In a large database, many of the attribute values will be incorrect.

This may be due to human error or because of any instruments fail.

B. Performance issues:

i. Efficiency and scalability of data mining algorithms:

To effectively extract information from a huge amount of data in databases, data mining algorithms must be efficient and scalable.

ii. Parallel, distributed, and incremental mining algorithms:

There are huge size of databases, the wide distribution of data, and complexity of some data mining methods.

These factors should be considered during development of parallel and distributed data mining algorithms.

C. Issues relating to the diversity of database types:

i. Handling of relational and complex types of data:

There are many kinds of data stored in databases and data warehouses.

It is not possible for one system to mine all these kinds of data. So, different data mining system should be constructed for different kinds data.

ii. Mining information from heterogeneous databases and global information systems:

Since data is fetched from different data sources on Local Area Network (LAN) and Wide Area Network (WAN), the discovery of knowledge from different sources of structured is a great challenge to data mining.

*Data Objects and Attribute Types:

Data Objects:

Data sets are made up of data objects.

A **data object** represents an entity.

Example: in a sales database, the objects may be customers, store items, and sales; in a medical database, the objects may be patients.

Data objects are typically described by attributes.

If the data objects are stored in a database, they are *data tuples*. That is, the rows of a database correspond to the data objects, and the columns correspond to the attributes.

Attribute:

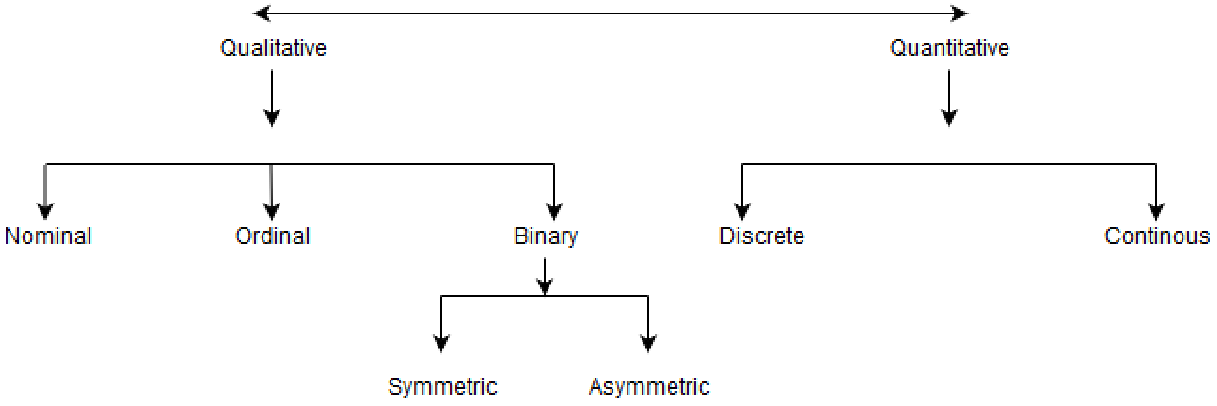
Attribute is a data field that represents characteristics or features of a data object.

For a customer object, attributes can be customer Id, address etc.

Set of attributes used to describe an object.

Types of attributes:

1. Qualitative Attributes
2. Quantitative Attributes



1. Qualitative Attributes:

a. Nominal Attributes (N):

These attributes are related to names.

The values of a Nominal attribute are name of things, some kind of symbols.

Values of Nominal attributes represents some category or state and that's why nominal attribute also referred as **categorical attributes** and there is no order (rank, position) among values of nominal attribute.

Example:

Attribute	**Values**
Colors	Black, Red, Green
Categorical Data	Lecturer, Professor

b. Binary Attributes (B):

Binary data has only 2 values/states.

Example: yes or no, affected or unaffected, true or false.

i.Symmetric: Both values are equally important (Gender).

ii.**Asymmetric:** Both values are not equally important (Result).

Attribute	**Values**
Gender	Male, Female

Attribute	**Values**
Result	Pass, Fail

c. Ordinal Attributes (O):

The Ordinal Attributes contains values that have a meaningful sequence or ranking(order) between them.

Attribute	**Values**
Grade	A, B, C, D, E
Income	low, medium, high
Age	Teenage, young, old

2. Quantitative Attributes:

a. Numeric:

A numeric attribute is quantitative because, it is a measurable quantity, represented in integer or real values.

Attribute	**Values**
Salary	2000, 3000
Units sold	10, 20
Age	5,10,20..

b. Discrete:

Discrete data have finite values, it can be numerical and can also be in categorical form. These attributes have finite or countably infinite set of values.

Example:

Attribute	**Values**
Profession	Teacher, Businessman, Peon
Zip Code	413736, 413713

c. Continuous:

Continuous data have infinite no. of states. Continuous data is of float type. There can be many values between 2 and 3.

Example:

Attribute	**Values**
Height	2.3, 3, 6.3……
Weight	40, 45.33,…….

*Data Preprocessing:

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format.

Real-world data is incomplete, inconsistent and contain many errors.

Data preprocessing is a proven method of resolving such issues.

Data preprocessing prepares raw data for further processing.

Data preprocessing is used database-driven applications such as customer relationship management and rule-based applications (like neural networks).

Why preprocess the data?

Data Preprocessing is required because real world data are generally:

Incomplete:

Missing attribute values, missing certain attributes of importance, or having only aggregate data.

Noisy:

Containing errors or outliers.

Inconsistent:

Containing discrepancies in codes or names.

*Major Tasks in data preprocessing:

Data goes through a series of tasks during preprocessing:

1. **Data Cleaning:** Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.
2. **Data Integration:** Data with different representations are put together and conflicts within the data are resolved.
3. **Data Transformation:** Data is normalized, aggregated and generalized.
4. **Data Reduction:** This step aims to present a reduced representation of the data in a data warehouse.
5. **Data Discretization:** Involves the reduction of a number of values of a continuous attribute by dividing the range of attribute intervals.

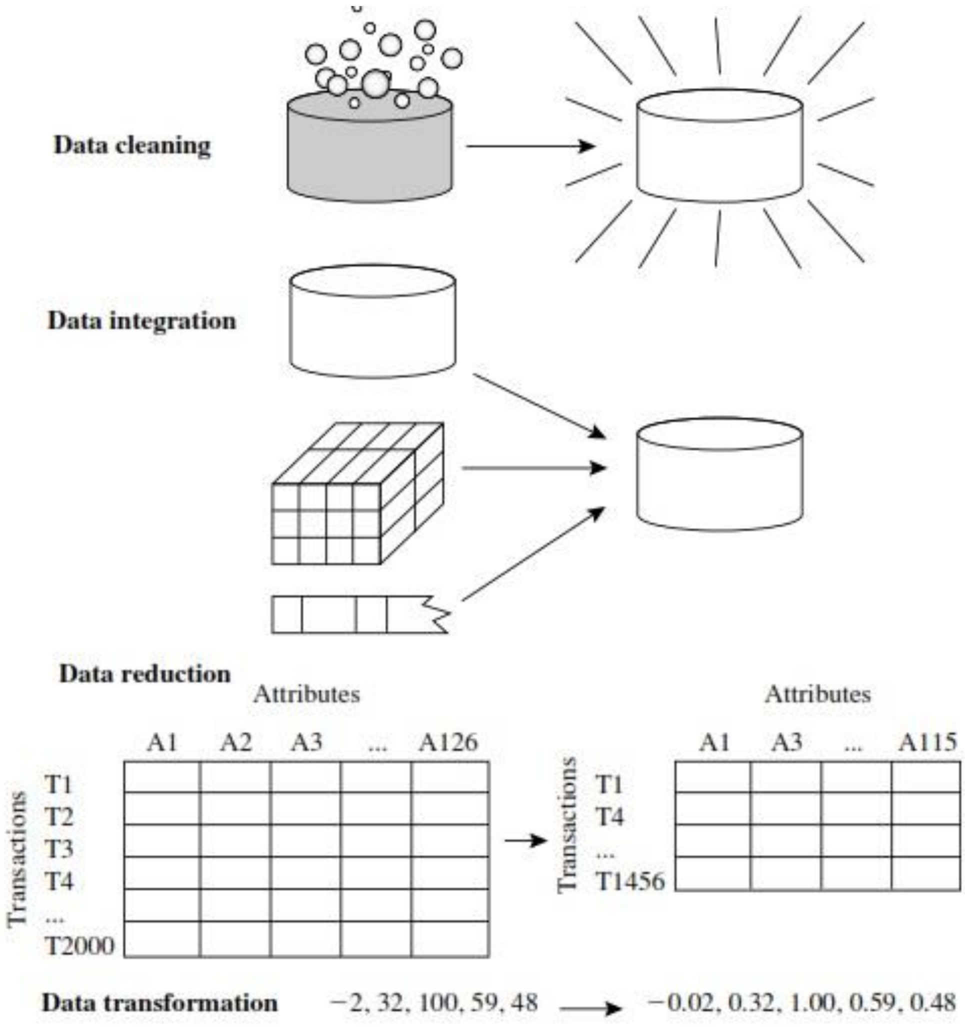


Fig: Tasks in Data Preprocessing

1. Data Cleaning in Data Mining:

Quality of your data is critical in getting to final analysis. Any data which is incomplete, noisy and inconsistent can affect your result.

Data cleaning in data mining is the process of detecting and removing corrupt or inaccurate records from a record set, table or database.

Some data cleaning methods:

A. Missing Values:

a. Ignore the tuple:

This is done when class label is missing.

Ignore the tuple only if maximum attributes have the missing values.

b. Fill in the missing value manually:

This approach is effective on small data set with some missing values.

c. Replace all missing attribute values with global constant:

Global constants like “Unknown”.

d. Use the attribute mean to fill in the missing value:

For example, customer average income is 25000 then you can use this value to replace missing value for income.

e. Use the most probable value to fill in the missing value.

B. Cleaning the Noisy Data:

Noise is a random error or variance in a measured variable.

Noisy Data may be due to faulty data collection instruments, data entry problems and technology limitation.

Binning Method to clean the noisy data:

Binning methods sorted data value by consulting its “neighbour- hood,” that is, the values around it.

Example:

Price: 8, 15, 21, 24, 21, 4, 28, 25, 34

Sort the values

Price = 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal-frequency) bins:

Bin a: 4, 8, 15

Bin b: 21, 21, 24

Bin c: 25, 28, 34

In this example, the data for price are first sorted and then partitioned into equal-frequency bins of size 3.

Smoothing by bin means:

Bin a: 9, 9, 9

Bin b: 22, 22, 22

Bin c: 29, 29, 29

In smoothing by bin means, each value in a bin is replaced by the mean value of the bin.

Smoothing by bin boundaries:

Bin a: 4, 4, 15

Bin b: 21, 21, 24

Bin c: 25, 25, 34

In smoothing by bin boundaries, each bin value is replaced by the closest boundary value. (find small and largest values)

C. Regression:

Data can be smoothed by fitting the data into a regression function.

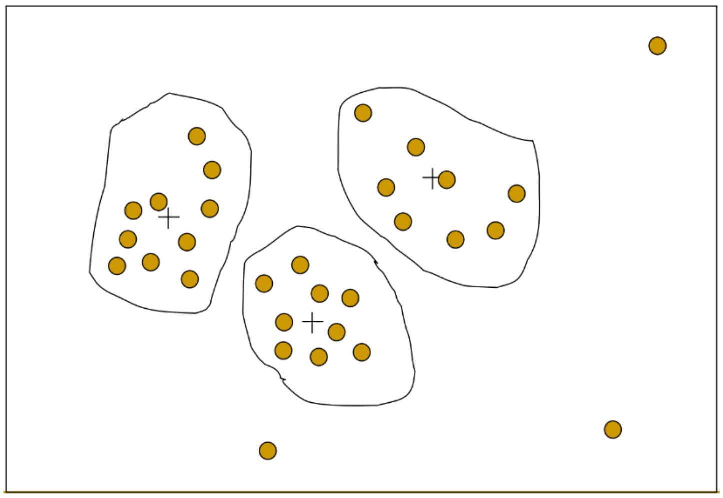
Example:

If we measured the height of child per year, if child grows 3 inches approximately, then the regression function may be: growing 3 inches per year

D. Clustering:

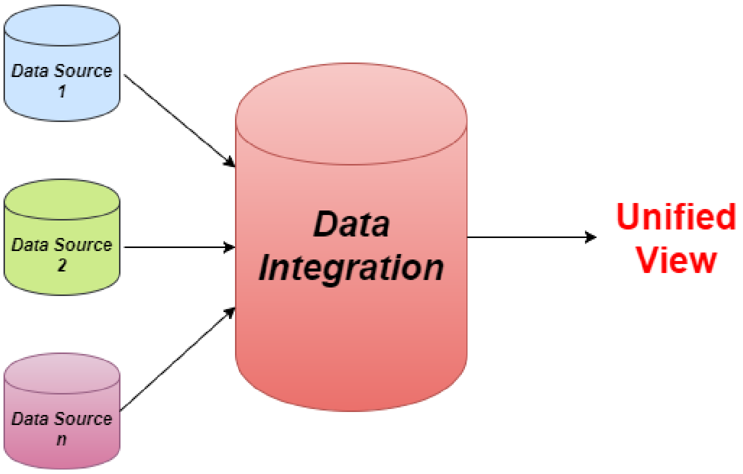
Outliers may be detected by clustering, where similar values are organized into groups, or “clusters.

Values that fall outside of the set of clusters may be considered outliers.



2. Data Integration in Data Mining:

Data Integration is a data preprocessing technique that combines data from multiple sources and provides users a unified view of these data.



These sources may include multiple databases, data cubes, or flat files. One of the most well-known implementation of data integration is building an enterprise's data warehouse.

The benefit of a data warehouse enables a business to perform analysis based on the data in the data warehouse.

There are mainly 2 major approaches for data integration:

Tight Coupling

In tight coupling data is combined from different sources into a single physical location through the process of ETL - Extraction, Transformation and Loading.

Loose Coupling

In loose coupling data only remains in the actual source databases. In this approach, an interface is provided that takes query from user and then sends the query directly to the source databases to obtain the result.

3. Data Transformation in Data Mining:

In data transformation process data are transformed from one format to another format, that is more appropriate for data mining.

Ex: Original data: 1.2, 3.2, 4.6, 123

Transformed data: 120, 320, 460, 123

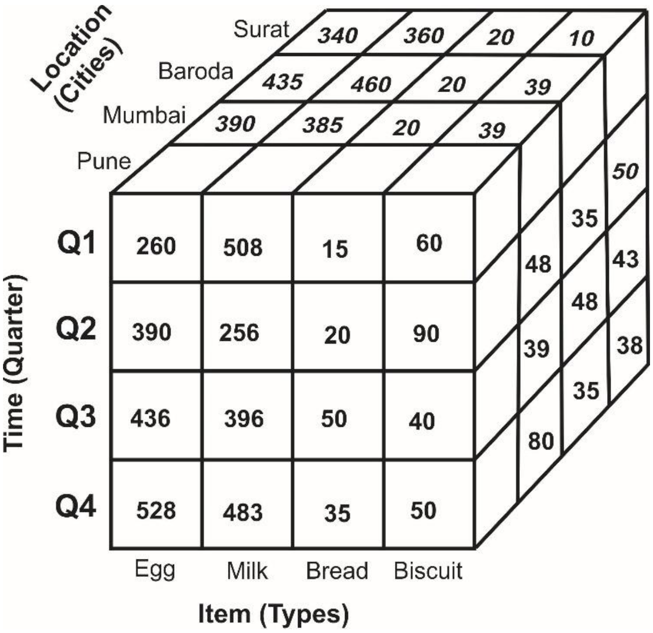
Some Data Transformation Strategies:

a. Smoothing:

Smoothing is a process of removing noise from the data. ***(For example, refer page no. 10)***

b. Aggregation:

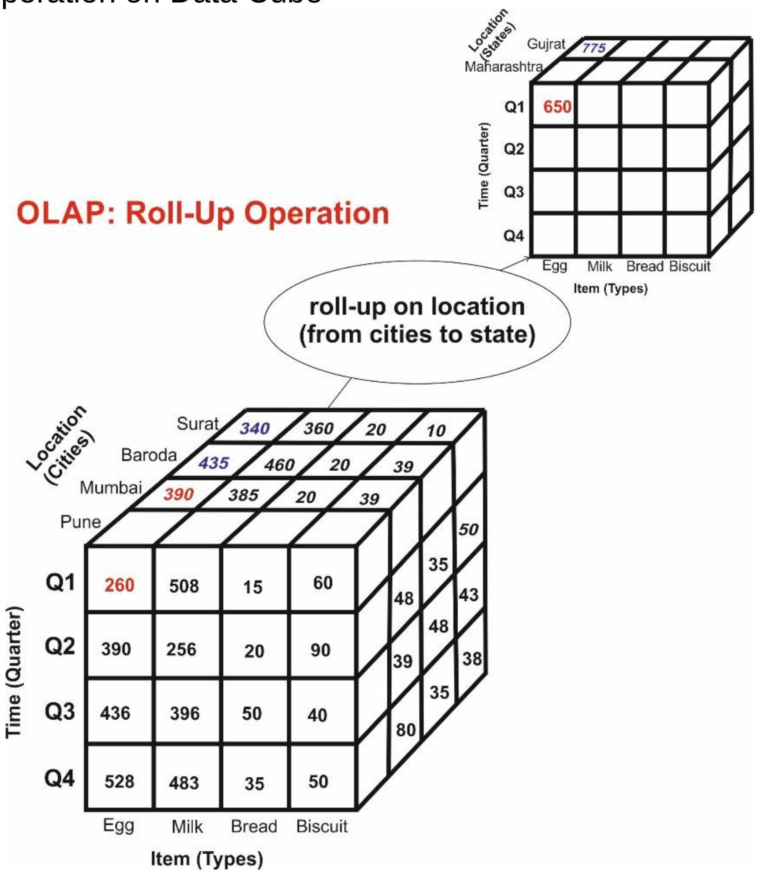
Aggregation is a process where summary or aggregation operations are applied to the data.



c. Generalization:

In generalization low-level data are replaced with high-level data by using concept hierarchies climbing.

Example: Roll-up operation on Data Cube



4. Data Reduction in Data Mining:

A database or date warehouse may store terabytes of data. So, it may take very long to perform data analysis and mining on such huge amounts of data.

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume but still contain critical information.

Data Reduction Strategies:

a. Data Cube Aggregation:

Aggregation operations are applied to the data in the construction of a data cube.

(For example, refer page no. 13)

b. Dimensionality Reduction:

In dimensionality reduction redundant attributes are detected and removed which reduce the data set size.

Example:

Before reduction

A1	A2	**A1**	A3

After reduction

A1	A2	A3

c. Discretization process:

(for concept refer the below section)

5. Data Discretization:

Data Discretization techniques can be used to divide the range of continuous attribute into intervals.

i.e. it divides the large dataset into smaller parts.

Numerous continuous attribute values are replaced by small interval labels.

This leads to a brief, easy-to-use, knowledge-level representation of mining results.

Data mining on a reduced data set means fewer input/output operations and is more efficient than mining on a larger data set.

Because of these benefits, discretization techniques and concept hierarchies are typically applied before data mining, rather than during mining.

Typical methods for Discretization and Concept Hierarchy Generation for Numerical Data

a. Binning Method:

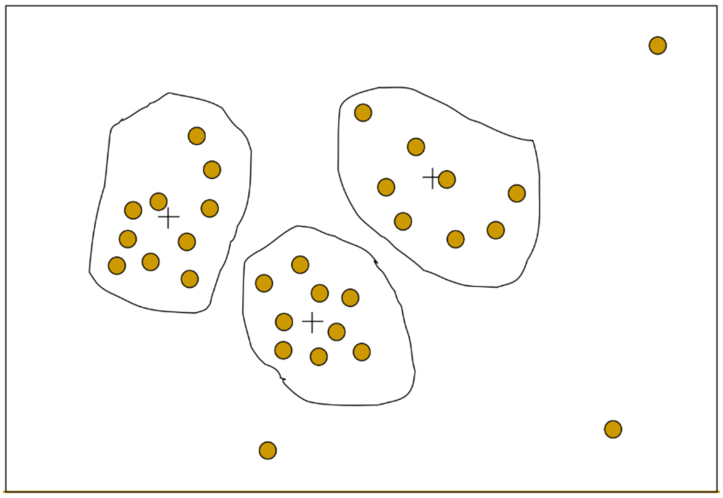
(Refer page no. 10)

b. Cluster Analysis:

Cluster analysis is a popular data discretization method.

A clustering algorithm can be applied to discrete a numerical attribute of A by partitioning the values of A into clusters or groups.

Each initial cluster or partition may be further decomposed into several subcultures, forming a lower level of the hierarchy.



Assignment 4

1. Define the term Data cleaning with example. (2)
2. Define the term Data mining. (2)
3. List methods of data preprocessing. (2)
4. Describe any four Challenges of Data mining. (4)
5. Explain Data Cleaning Process. (4)
6. Describe the need of data preprocessing. (4)
7. Explain Data preprocessing technique in data mining. (6)
8. Explain steps involved in KDD process with diagram. (6)

Unit 5: Mining Frequent Patterns and Cluster Analysis (14 Marks)

Frequent Patterns:

Frequent patterns are patterns (e.g., itemsets, subsequences, or substructures) that appear frequently in a data set.

For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set is a *frequent itemset*.

A subsequence, such as buying first a PC, then a digital camera, and then a memory card, if it occurs frequently in a shopping history database, is a (*frequent*) *sequential pattern*.

Market Basket Analysis:

Market Basket Analysis is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items.

Ex: (Computer→Antivirus)

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items.

It works by looking for combinations of items that occur together frequently in transactions. i.e it allows retailers to identify relationships between the items that people buy.

Market basket analysis can be used in deciding the location and promotion of goods inside a store.

Market Basket Analysis creates If-Then scenario rules, for example, if item A is purchased then item B is likely to be purchased.

How is it used?

As a first step, market basket analysis can be used in deciding the location and promotion of goods inside a store.

If, it has been observed, purchasers of Barbie dolls have been more likely to buy candy, then high-margin candy can be placed near to the Barbie doll display.

Customers who would have bought candy with their Barbie dolls had they thought of it will now be suitably tempted.

But this is only the first level of analysis. Differential market basket analysis can find interesting results and can also eliminate the problem of a potentially high volume of trivial results.

In differential analysis, compare results between different stores, between customers in different demographic groups, between different days of the week, different seasons of the year, etc.

If we observe that a rule holds in one store, but not in any other (or does not hold in one store, but holds in all others), then we know that there is something interesting about that store.

Investigating such differences may yield useful insights which will improve company sales.

Other Application Areas

Market Basket Analysis used for:

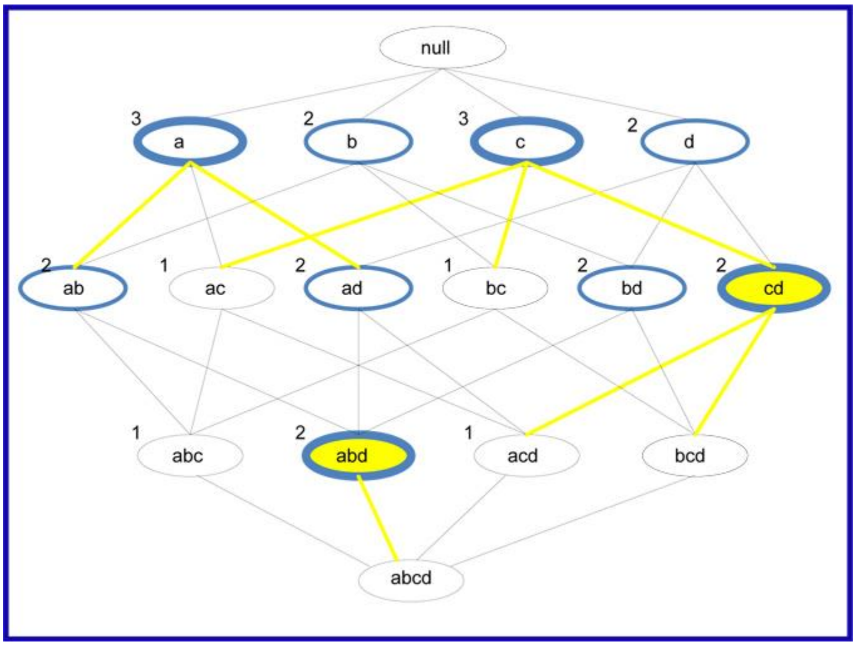
Analysis of credit card purchases.

Analysis of telephone calling patterns.

Identification of fraudulent medical insurance claims.

Analysis of telecom service purchases.

Frequent Itemsets and Closed Itemsets:



Frequent Itemsets:

Frequent itemsets are patterns that appear frequently in a data set.

For example, a set of items, such as milk and bread, that appear frequently together in a transaction data set is a *frequent itemset*.

An itemset whose support is greater than or equal to a minimum support threshold.

From diagram frequent itemsets are: a, b, c, d, ab, ad, bd, cd, abd

Closed Itemsets:

An itemset is closed if none of its immediate supersets has the same support as that of the itemset.

Consider two itemsets X and Y, if every item of X is in Y but there is at least one item of Y, which is not in X, then Y is not a proper super itemset of X, here itemset X is closed itemset.

If X is both closed and frequent, called as closed frequent itemset.

From diagram closed frequent itemsets are: a, c, cd, abd
cd is closed itemset as its supersets acd and bcd have support less than 2

Association Rules:

They are widely used to analyze retail basket or transaction data, and are intended to identify strong rules discovered in transaction data using measures of interestingness, based on the concept of strong rules.

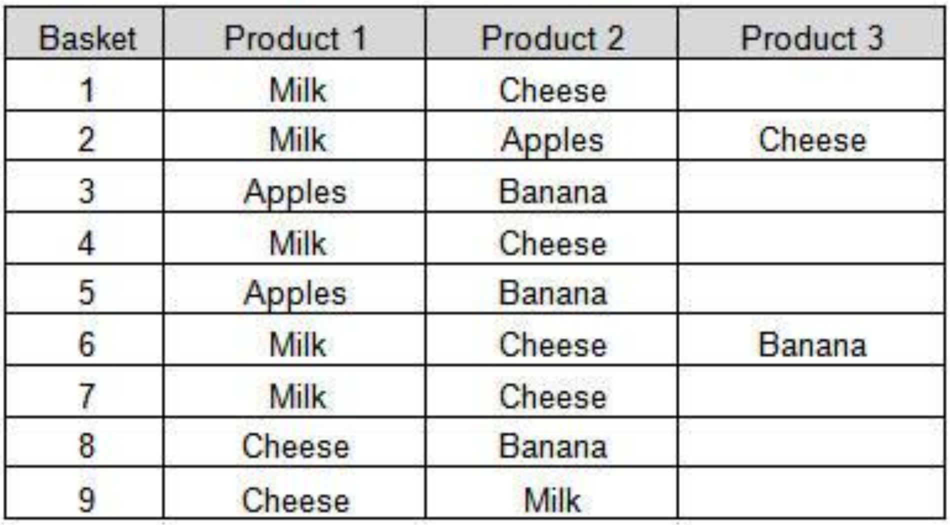
Association Rule – An implication expression of the form X → Y, where X and Y are itemsets.

Example: {Milk, Cheese} → {Banana}

Example:

You are in a supermarket to buy milk. Based on the analysis, are you more likely to buy apples or cheese in the same transaction than somebody who did not buy milk?

In the following table, there are nine baskets containing varying combinations of milk, cheese, apples, and bananas.



Basket	Product 1	Product 2	Product 3
1	Milk	Cheese	
2	Milk	Apples	Cheese
3	Apples	Banana	
4	Milk	Cheese	
5	Apples	Banana	
6	Milk	Cheese	Banana
7	Milk	Cheese	
8	Cheese	Banana	
9	Cheese	Milk	

Determine the relationships and the rules.

The first measure called the **support** is the number of transactions that include items in the {A} and {B} parts of the rule as a percentage of the total number of transactions. It is a measure of how frequently the collection of items occur together as a percentage of all transactions.

Support= X+Y / N (N: total transactions or baskets)

Fraction of transactions that contain both X and Y.

The second measure called the **confidence** of the rule is the ratio of the number of transactions that include all items in {Y} as well as the number of transactions that include all items in {X} to the number of transactions that include all items in {X}.

Confidence= X+Y / X

How often items in Y appear in transactions that contain X only.

The third measure called the **lift or lift ratio** is the ratio of confidence to expected confidence. Expected confidence is the confidence divided by the frequency of Y. The Lift tells us how much better a rule is at predicting the result than just assuming the result in the first place. Greater lift values indicate stronger associations.

Lift= Confidence / (Y/N)

How much our confidence has increased that Y will be purchased given that X was purchased.

Rules	Support (X+Y/N)	Confidence (X+Y/X)	Lift (Confidence/(Y/N)
Milk→Cheese	6/9=0.66	6/6=1	1/ (7/9) =9/7=1.28
Apple, Milk→Cheese	1/9=0.11	1/1=1	1/ (7/9) =9/7=1.28
Apple, Cheese→Milk	1/9=0.11	1/1=1	1/ (6/9) =9/6=1.5

Apriori Algorithm – Frequent Pattern Algorithms

A set of items together is called an itemset. If any itemset has k-items it is called a k-itemset. An itemset consists of two or more items. An itemset that occurs frequently is called a frequent itemset.

Thus, frequent itemset mining is a data mining technique to identify the items that often occur together.

For Example, Bread and butter, Laptop and Antivirus software, etc.

Name of the algorithm is Apriori because it uses prior knowledge of frequent itemset properties. We apply an iterative approach or level-wise search where k-frequent itemsets are used to find k+1 itemsets.

This algorithm uses two steps “join” and “prune” to reduce the search space.

It is an iterative approach to discover the most frequent itemsets.

Apriori Property:

All non-empty subset of frequent itemset must be frequent.

Apriori says:

The probability that item x is not frequent is if:

- P(x) is less than minimum support threshold, then x is not frequent.

The steps followed in the Apriori Algorithm of data mining are:

1. **Join Step**: This step generates (K+1) itemset from K-itemsets by joining each item with itself.
2. **Prune Step**: This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

Apriori Algorithm:

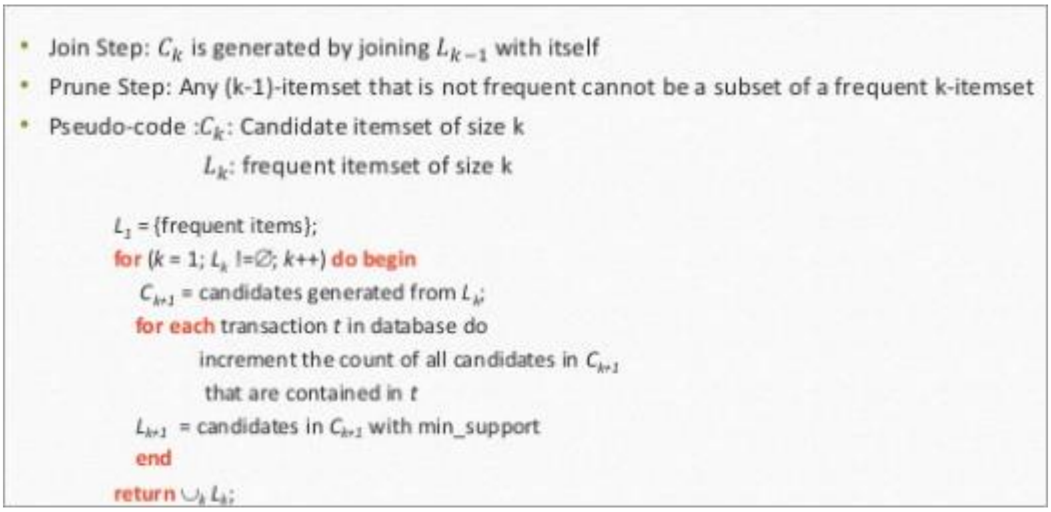
D: Database

Min_sup: minimum support count

K: items in itemset

C: candidate list

L: frequent itemsets in D



- Join Step: C_k is generated by joining L_{k-1} with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- Pseudo-code :C_k: Candidate itemset of size k
  L_k: frequent itemset of size k

```
$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=∅; $k$++) do begin
   $C_{k+1}$ = candidates generated from $L_k$;
   for each transaction $t$ in database do
        increment the count of all candidates in $C_{k+1}$
         that are contained in $t$
   $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
   end
return $\cup_k L_k$;
```

Example Apriori Method:

Consider the given database D and minimum support 50%. Apply the Apriori algorithm and find frequent itemsets with confidence greater than 70%

TID	Items
1	1 3 4
2	2 3 5
3	1 2 3 5
4	2 5

Solution:

Calculate min_supp=0.5*4=2

(0.5: given minimum support, 4: total transactions in database D)

Step 1: Generate candidate list C1 from D

C1=

Itemsets
1
2
3
4
5

Step 2: Scan D for count of each candidate and find the support.

C1=

Itemsets	Support count
1	2
2	3
3	3
4	1
5	3

Step 3: Compare candidate support count with min_supp (i.e. 2)

(prune or remove the itemset which have support count less than min_supp i.e. 2)

L1=

Itemsets	Support count
1	2
2	3
3	3
5	3

Step 4: Generate candidate list C1 from L1

(k-itemsets converted to k+1 itemsets)

C2=

Itemsets (k+1)
1,2
1,3
1,5
2,3
2,5
3,5

Step 5: Scan D for count of each candidate and find the support.

C2=

Itemsets	Support count
1,2	1
1,3	2
1,5	1
2,3	2
2,5	3
3,5	2

Step 6: Compare candidate support count with min_supp (i.e. 2)

(prune or remove the itemset which have support count less than min_supp i.e. 2)

L2=

Itemsets	Support count
1,3	2
2,3	2
2,5	3
3,5	2

Step 7: Generate candidate list C3 from L2

(k-itemsets converted to k+1 itemsets)

C3=

Itemsets (k+1)
1,2,3
1,2,5
1,3,5
2,3,5

Step 8: Scan D for count of each candidate and find the support.

C3=

Itemsets	Support count
1,2,3	1
1,2,5	1
1,3,5	1
2,3,5	2

Step 9: Compare candidate support count with min_supp (i.e. 2)

(prune or remove the itemset which have support count less than min_supp i.e. 2)

L3=

Itemsets	Support count
2,3,5	2

Step 10: Frequent itemset is {2,3,5}

Apply Association rules:

Rule	Support	Confidence	Confidence %
2 3→5	2	2/2=1	100
3 5→2	2	2/2=1	100
2 5→3	2	2/3=0.66	66
2→3 5	2	2/3=0.66	66
3→2 5	2	2/3=0.66	66
5→2 3	2	2/3=0.66	66

As minimum confidence threshold is 70%, the first two rules are the output.

i.e. 2 3→5, 3 5→2

Cluster Analysis:

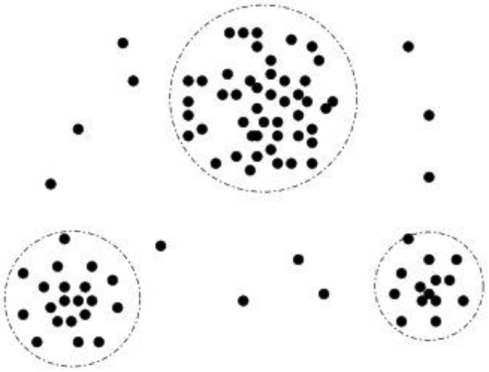
Clustering is a data mining technique used to place data elements into related groups without advance knowledge.

Clustering is the process of grouping a set of data objects into multiple groups or *clusters* so that objects within a cluster have high similarity, but are very dissimilar to objects in other clusters.

Dissimilarities and similarities are assessed based on the attribute values describing the objects and often involve distance measures.

Cluster analysis or simply clustering is the process of partitioning a set of data objects (or observations) into subsets.

Each subset is a **cluster**, such that objects in a cluster are similar to one another, yet dissimilar to objects in other clusters. The set of clusters resulting from a cluster analysis can be referred to as a **clustering**.



Requirements of Cluster Analysis:

- **Scalability**: Need highly scalable clustering algorithms to deal with large databases.
- **Ability to deal with different kinds of attributes**: Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.
- **Discovery of clusters with attribute shape:** The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.
- **High dimensionality**: the clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data:** Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability:** The clustering results should be interpretable, comprehensible, and usable.

***Basic Clustering Methods:**

Clustering methods can be classified into the following categories −

- Partitioning Method
- Hierarchical Method
- Density-based Method
- Grid-Based Method
- Model-Based Method
- Constraint-based Method

1. Partitioning Method:

Suppose we are given a database of ‘n’ objects and the partitioning method constructs ‘k’ partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contains at least one object.
- Each object must belong to exactly one group.

Points to remember:

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.

Algorithm: *k*-means.

The *k*-means algorithm for partitioning, where each cluster’s center is represented by the mean value of the objects in the cluster.

Input:

k: the number of clusters,

D: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

```
(1) arbitrarily choose k objects from D as the initial cluster centers;
(2) repeat
(3) (re)assign each object to the cluster to which the object is the most similar,
based on the mean value of the objects in the cluster;
(4) update the cluster means, that is, calculate the mean value of the objects for
each cluster;
(5) until no change;
```

Example: K-means

Q: Use k-means algorithm to create 3 clusters for given set of values:

{2,3,6,8,9,12,15,18,22}

Answer:

Set of values: 2,3,6,8,9,12,15,18,22

1. Break given set of values randomly in to 3 clusters and calculate the mean value.

K1: 2,8,15	mean=8.3
K2: 3,9,18	mean=10
K3: 6,12,22	mean=13.3

2. Reassign the values to clusters as per the mean calculated and calculate the mean again.

K1: 2,8,3,9,6	mean=5.6
K2:	mean=0
K3: 15,18,12,22	mean=16.75

3. Reassign the values to clusters as per the mean calculated and calculate the mean again.

K1: 8,3,9,6	mean=6.5
K2: 2	mean=2
K3: 15,18,12,22	mean=16.75

4. Reassign the values to clusters as per the mean calculated and calculate the mean again.

K1: 6,8,9	mean=7.6
K2: 2,3	mean=2.5
K3: 12,15,18,22	mean=16.75

5. Reassign the values to clusters as per the mean calculated and calculate the mean again.

K1: 6,8,9	mean=7.6
K2: 2,3	mean=2.5
K3: 12,15,18,22	mean=16.75

6. Mean of all three clusters remain same.

Final 3 clusters are {6,8,9}, {2,3}, {12,15,18,22}

2. Hierarchical Methods

This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here:

- Agglomerative Approach
- Divisive Approach

Agglomerative Approach

This approach is also known as the bottom-up approach. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keeps on doing so until all of the groups are merged into one or until the termination condition holds.

Divisive Approach

This approach is also known as the top-down approach. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.

3. Density-based Method

This method is based on the notion of density. The basic idea is to continue growing the given cluster as long as the density in the neighbourhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.

4. Grid-based Method

In this, the objects together form a grid. The object space is quantized into finite number of cells that form a grid structure.

Advantages

- The major advantage of this method is fast processing time.
- It is dependent only on the number of cells in each dimension in the quantized space.

5. Model-based methods

In this method, a model is hypothesized for each cluster to find the best fit of data for a given model. This method locates the clusters by clustering the density function. It reflects spatial distribution of the data points.

This method also provides a way to automatically determine the number of clusters based on standard statistics, taking outlier or noise into account. It therefore yields robust clustering methods.

6. Constraint-based Method

In this method, the clustering is performed by the incorporation of user or application-oriented constraints. A constraint refers to the user expectation or the properties of desired clustering results. Constraints provide us with an interactive way of communication with the clustering process. Constraints can be specified by the user or the application requirement.

Applications of Clustering:

Clustering algorithms can be applied in many fields, for instance:

- **Marketing**: finding groups of customers with similar behaviour given a large database of customer data containing their properties and past buying records;
- **Biology**: classification of plants and animals given their features;
- **Libraries**: book ordering;
- **Insurance**: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
- **City-planning**: identifying groups of houses according to their house type, value and geographical location;
- **Earthquake studies**: clustering observed earthquake epicenters to identify dangerous zones;
- **WWW**: document classification; clustering weblog data to discover groups of similar access patterns.

Assignment 5

1. State Application of cluster analysis. (2)
2. Define cluster Analysis. (2)
3. Define frequent itemset and closed itemset. (2)
4. Describe Association rule of data mining. (2)
5. Explain Market basket analysis. (4)
6. Describe the requirement of clustering in data mining. (4)
7. Consider the database (D) with min_supp=60% and min_confidence=80%

TID	Items
1	K, A, D, B
2	D, A, C, E, B
3	C, A, B, E
4	B, A, D

Find all frequent itemsets using apriori method. List strong association rules. (6)

8. List clustering Methods explain any two. (6)
9. Explain Apriori algorithms for frequent itemset using candidate generation. (6)
10. Consider the data set given and create 3 clusters using k-means method. (6)

Data set: {10,4,2,12,3,20,30,11,25,31}