* Data Warehouse :-

Data Warehousing provides Architectures & tools for business Executives to Systematically organize, Understand & use their data to make Strategic decisions.

→ A data warehouse refers to a data repository that is maintained Separately from an organization operational data - base.

Data Warehouse System allows for integration of Variety of Application Systems.

→ William H. Inmon States "A data Warehouse is a Subject - oriented, Integrated, Time - Variant & Non - volatile collection of data in Support of Management's decision making process".

* Key features of Data Warehouse :- The key features of Data warehouse are as follows.

→ "Subject - Oriented" :- Data warehouse is organized around major Subjects such as Customer, Supplier, product & Sales rather than concentrating on day - to - day operations & transaction processing of organization.

Data warehouse focuses on Modelling & Analysis of Data for Decision Making.

→ "Integrated" :- A data Warehouse is usually Constructed by Integrating Multiple Heterogeneous Sources such as Relational DB

Flat files & online Transaction Records.

"Data cleaning" & "Data Integration" techniques are applied to Ensure Consistency in naming Conventions, Encoding Structure attributes, Measures & so on.

→ "Time-Variant": Data are stored to provide information from an historic perspective. for Ex 5-10 years.
Every key Structure in data Warehouse Contains, Either Implicitly or Explicitly a time Element.

→ "Non-volatile": A data Warehouse is always a physically Separate Store of data transformed from the Application data found in operational Environment.
Due to this Separation, A data Warehouse does not req -uire Transaction processing, Recovery & Concurrency Control Mechanisms.

→ The Data Warehouse allows "Knowledge Workers" Such as Managers, Analysts & Executives to use the Warehouse to Quickly & obtain an overview of the data, & make deci -sions based on Info in Warehouse.
The Construction of Data warehouse requires "Data cleaning" "Data Integration" & "Data Consolidation".

→ Data Warehousing is very useful from the point of View of "Heterogenous Database Integration".
In Traditional Database Approach to heterogenous Database Integration is to build "Wrappers & Integrators" on top of Multiple Heterogenous Databases.

→ When a Query is posed to a client site, a Metadata dictionary is used to Translate the Query into Queries Appropriate for the Individual heterogenous Sites Involved.

* "Query-driven Approach" requires complex information filtering & integration process & competes with local sites for processing resources.
"Traditional Database" makes use of query-driven approach.

* "Update-driven Approach" in which information from multiple heterogeneous sources is integrated in advance & stored in a warehouse for direct querying & analysis.
"Data Warehousing" makes use of update-Driven Approach.

* Differences between Operational Database Systems & Data Warehouses
The major task of online operational database sys is to perform online transaction & query processing. These systems are called "online Transaction processing (OLTP)".
For Example, most of day-to-day operations of an organization such as "purchasing", "Manufacturing", "Banking" etc.

→ "Online Analytical processing (OLAP)" Servers work on knowledge workers in the role of data Analysis & Decision Making. Such System can organize & present the Data in various formats in order to accomodate the diverse needs of different users.
The major distinguishing features of OLTP & OLAP are

* Users & System Orientation :- An OLTP System is "Customer-oriented" and is used by clerks, clients, & IT professionals.
→ An OLAP System is "Market-Oriented" and is used by knowledge workers such as Managers, Executives & Analysts.

* Data Contents :- OLTP System Manages Current data that, typically, are too detailed to be used.

→ OLAP System Manage large Amount historic data, provides facilities for Summarization & Aggregation & Stores & Manage Information at different levels of granularity.

* Database Design :- OLTP usually adopts an Entity - Relationship (ER) data model & Appln oriented database design.

→ OLAP System adopts either "Star" or "Snowflake" data Models.

* View :- An OLTP System Focuses Mainly on Current data within an Enterprise or department, without referring to historic data or data in different organization.

→ OLAP Systems Span Multiple Version of database Schema, due to Evolutionary process of an organization.
OLAP sys deals with info that originates from different organization, Integrating info from different Sources.

* Access :- OLTP System Consist Mainly of Short atomic transa -ction so Concurrency Control & Recovery is required.

→ OLAP Systems are Mostly read-only operations for any Complex Queries.

Feature	OLTP	OLAP
* Characteristic	* operational processing	* Informational processing
* Orientation	* Transaction	* Analysis
* Users	* DBA, Clerk, Database professional	* Knowledge Workers
* Function	* Day-to-Day operation	* long-term for Decision Support
* DB-Design	* ER-Model, Appln oriented	* Star/SnowFlake, Subject oriented
* View	* Detailed, Flat Relational	* Summarized & Multi-dimensional
* Access	* Read/Write	* Read
* Focus	* Data-in	* Information-out
* Unit of Work	* Short, Simple Transaction	* Complex query

* Operations	* Index / Hash	* lots of Scan
* No of Records Accessed	* Tens	* Millions
* No of Users	* Thousands	* Hundreds
* DB Size	* GB to High-order GB	* ≥ TB
* Priority	* High performance & Availability	* High flexibility, End-user Autonomy
* Metric	* Transaction Throughput	* Query throughput & Response time

* Data Warehousing : A Multitiered Architecture :-

Data warehouses often adopts a 3-tier Architecture such as

* Bottom tier
* Middle tier
* Top tier

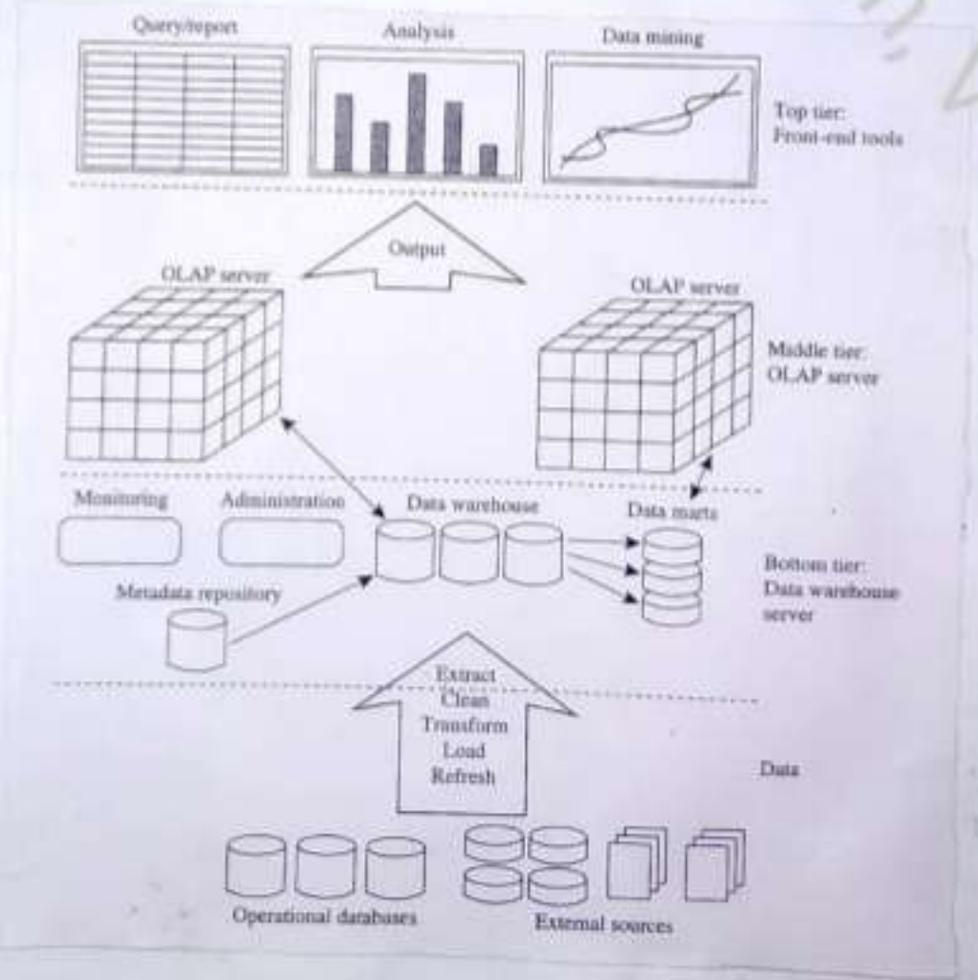


Fig:- Three-tier Datawarehouse Architecture

* Bottom tier :- Warehouse Database Server is almost a Relational - database & other External Sources.
Back-End tools & utilities are used to perform Extraction, cleaning & Transformation (Ex:- to Merge Similar data from different Sources into a unified format).
→ The data are Extracted using Appln program Interface known as "gateways"
Example of gateways include ODBC (Open Database Connection) & OLEDB (Object Linking & Embedding Database) by Microsoft & JDBC.
→ This tier also contains a Metadata Repository, which Stores Information about the Data Warehouse & Contents.

* Middle tier :- Consists of OLAP Servers that is typically Implemented using
→ "Relational OLAP (ROLAP)" : Extended Relational DBMS that Maps Multidimensional data to Standard Relational - operations.
→ "Multidimensional OLAP (MOLAP)": Direct Implementation of Multidimensional data & operation.

* Top tier :- Consists of Front-End Client layer, which contains Query & reporting tools, Analysis tools & Data mining tools. Such as "Trend Analysis", "prediction" & so on.

* Data Warehouse Models :-
There are three Data Warehouse Models
* Enterprise Warehouse
* Data Mart
* Virtual Warehouse

* Enterprise Warehouse :- Collects all the Information about Subjects Spanning Entire Organization.

It provide Corporate-Wide data with Corporate-Wide data -Integration.

→ Enterprise Warehouse Contains detailed data as well as Summarized data & can range in Size from of few Gigabytes to hundred GB, Terabyte or beyond.

→ An Enterprise data Warehouse may be Implemented on traditional Mainframes, Computer SuperServers or parallel Architecture platforms.

It Requires Extensive Business Modeling & May take years to Design.

* Data Mart :- Contains a Subset of Corporate-Wide data that is of Value to Specific Group of users.

For Ex: A Marketing Data Mart May confine its Subject to Customer, Item & Sales.

→ Data Marts are Usually Implemented on "low-cost Department Servers" that are Unix/Linux or Window based.

The Implementation Cycle of a Data mart is More likely to be Measured in Weeks rather than Months or Years.

→ Data Marts Can be Categorized as

* Independent → Sourced from operational Systems or External Info providers.

* Dependent → Sourced Directly from Enterprise Data -Warehouse

* Virtual Warehouse :- is a set of Views over operational database

for Efficient Query processing.

A Virtual Warehouse is Easy to build but requires Excess capacity on operational database Servers.

→ For Ex: Airline ticketing System, collects & displays business data relating to a Specific Moment in time, Creating a Snapshot

of the condition of business at that Moment.

* Recommended Approach for Warehouse development :-

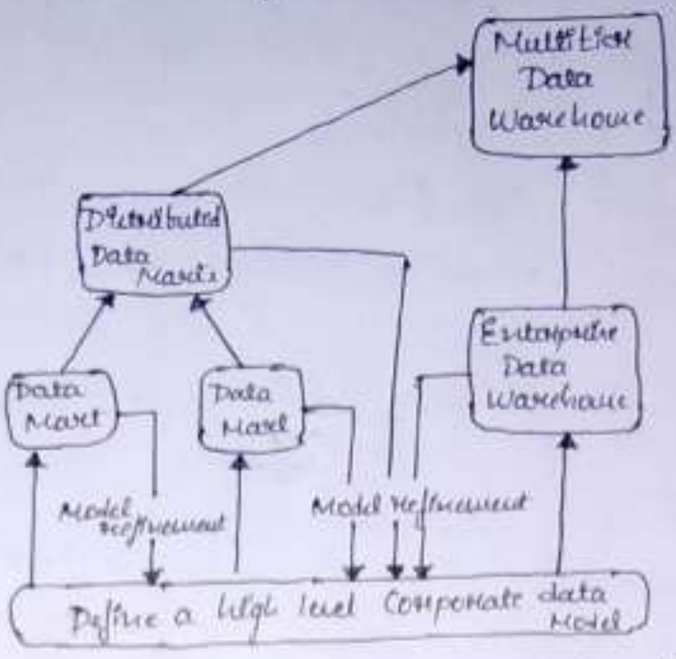


Fig:- A Recommended Approach for Warehouse Development

A Recommended Method for the development of data Warehouse Systems is to implement the Warehouse in an Incremental & Evolutionary Manner as shown in above figure.

→ Firstly, a high-level corporate data Model is defined within a short period of time, that provides a corporate-wide view of data among different subject & potential usages.

→ Second, Independent data Marts can be implemented in parallel with the Enterprise Warehouse based on the Corporate data Model.

→ Third, Distributed data Marts can be constructed to integrate different data Marts via hub Servers.

Finally a "Multitier Data Warehouse" is constructed, where the Enterprise Warehouse is the sole custodian of all warehouse data, which is then distributed to various dependent Data Marts.

* Data Cube : A Multidimensional Data Model

A Data Cube allows data to be Modeled & Viewed in Multiple Dimensions. It is defined by "Facts" & "Dimensions".

* Dimensions → are the perspectives or Entities with respect to which an organization wants to keep Records.
For Ex: "All Electronics" may Create a "Sales" data Warehouse in order to keep records of Store Sales with respect to Dimensions "time", "Item", "branch" & "location".

* Facts → are numeric Measures. Quantities by which we want to analyze relationship between Dimensions.
For Ex: Facts for a Sales data warehouse include "dollars_sold", "Units_Sold", & "Amount_budgeted".

→ Consider a 2D Simple data Cube Example from "All Electronics" Sales data for Items Sold per Quarter in City Vancouver.

Location = "Vancouver"

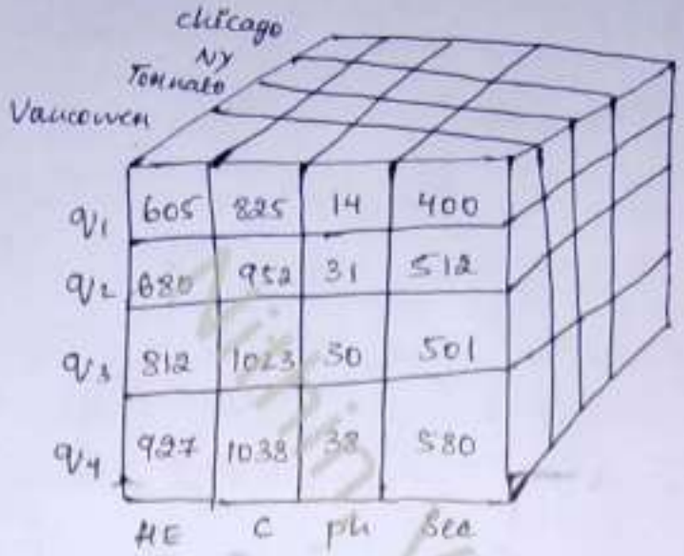
Time (Quarter)	Item (Type)			
	Home Ent	Computer	Phone	Security
q_1	605	825	14	400
q_2	680	952	31	512
q_3	812	1023	30	501
q_4	927	1038	38	580

In this 2-D Repsn, the Sales for Vancouver are shown w.r.t time dimension & Item.

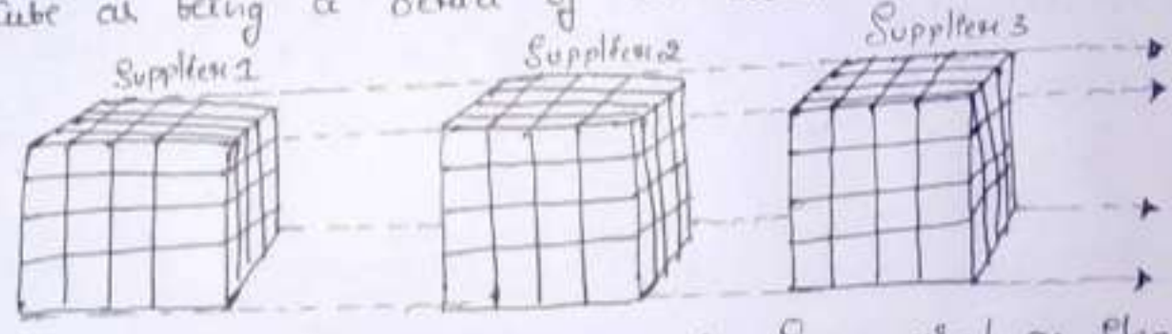
→ Suppose that we would like to view the Sales-data with 3-D Repsn the data according to time, Item & Loc as shown below

	Loc = "Chicago"				Loc = "New York"				Loc = "Toronto"				Loc = "Vancouver"			
	Item				Item				Item				Item			
Time	HE	C	Ph	See	HE	C	Ph	See	HE	C	Ph	See	HE	C	Ph	See
q_1	854	882	89	623	1087	720	64	620	1023	760	65	620	1000	620	65	60
q_2	943	890	64	698	1130	841	72	619	1014	820	66	619	1001	619	60	50
q_3	1052	924	59	789	1034	900	63	618	1025	823	67	720	1002	700	55	40
q_4	1129	993	63	870	1142	821	79	617	1026	824	68	723	1003	710	50	30

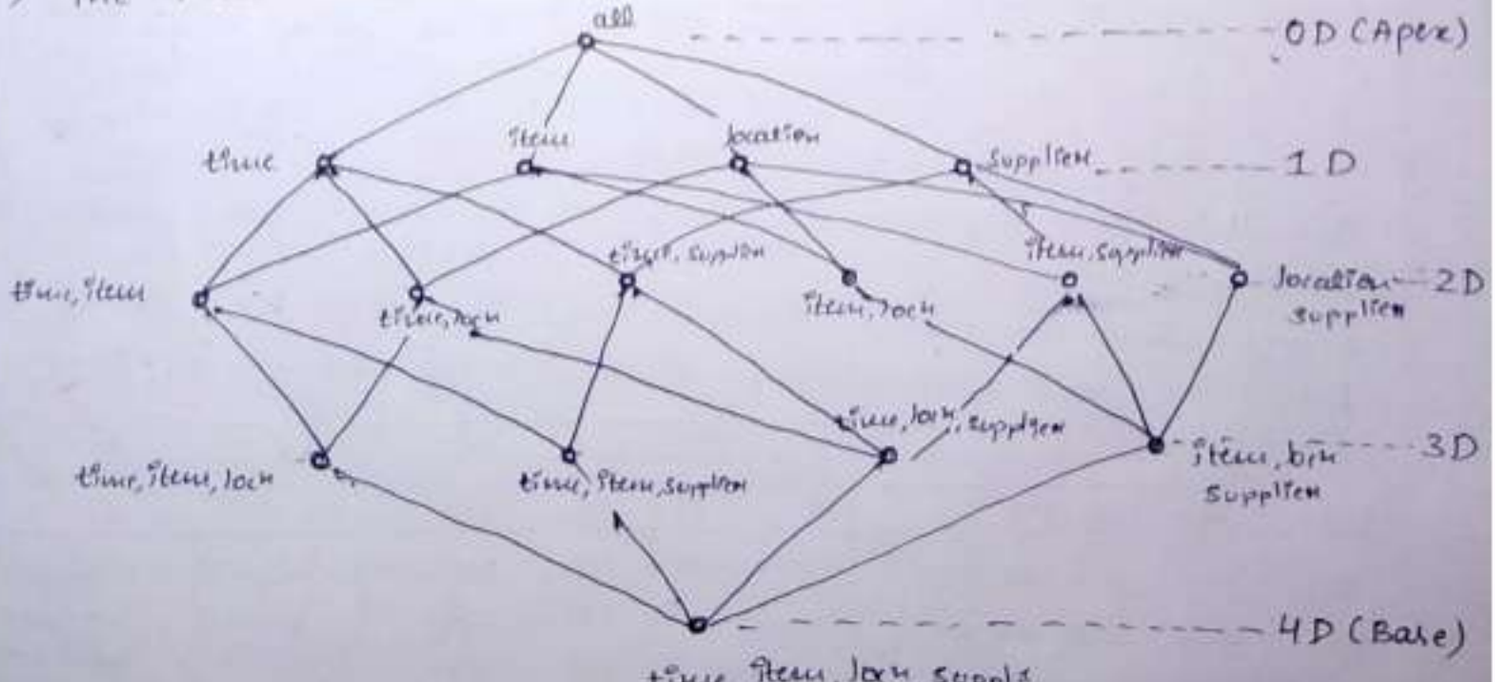
→ We may also represent the same data in the form of 3D data cube.



→ Suppose that we would like to view our sales data with addition of 4th Dimension such that Supplier. Viewing things on 4-D cube becomes complex. However, we can think of a 4-D cube as being a series of 3D-Cubes



→ The Multidimensional Data Model can be summarized as shown



The 0D cuboid, which holds the highest level of Summarizat -ion is called "Apex Cuboid"
In our Example, the "total_sales" or "dollars_sold", Summariz -ed over all four dimensions.

* Star, Snowflake & Fact Constellations : Schema for Multidimen -sional Data Model

The Entity - Relationship data model is Commonly used to design the relational database, where database Schema Consists of a Set of Entities & the Relation -ships between them. Such data Model is Appropriate for online Transaction processing.

→ The Most popular data Model for data Warehouse is a Multidimensional Model which can Exist in the forms of

* Star Schema
* Snowflake Schema
* Fact Constellation Schema

* Star Schema :- It is the Most Commonly used Multidimens -ional data Model, in which the data Warehouse Contains

* A large Central table (Fact table) containing the bulk data with no Redundancy.
* A Set of Smaller Attendant - tables (Dimension tables) One for Each dimension.

The Schema Graph resembles a Star burst, with the Dimension tables displayed in Radial patterns around the Central Fact table.

→ A Star Schema for "all Electronics" is shown in the foll figure.

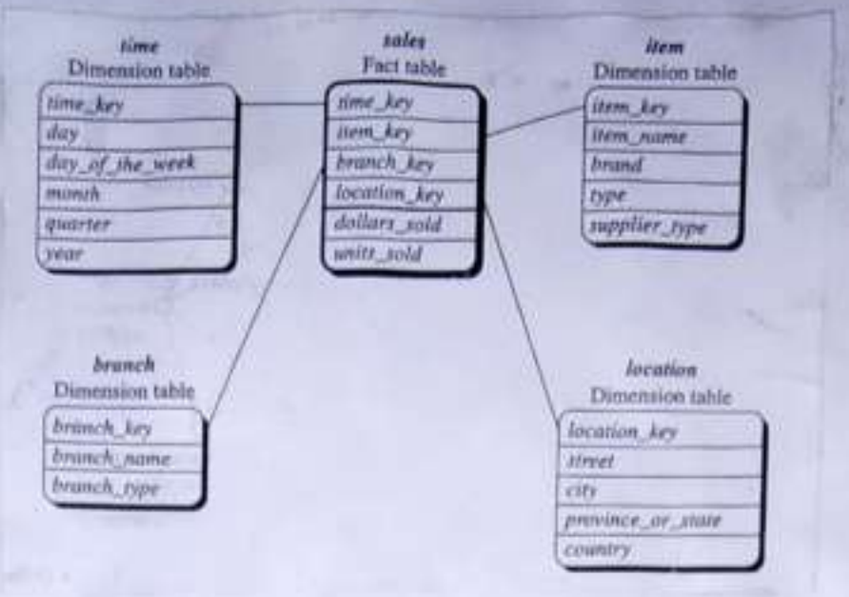


Fig:- Star Schema of Sales Data Warehouse

In the above Schema, Sales is Considered along four Dimensions "Time, Item, branch & location". The Schema contains a Central Fact table for Sales that Contains keys to Each of the four Dimensions with two Measures "dollars_sold" & "units_sold". To Minimize the Size of fact table, dimension_Identifiers (time_key, item_key Etc) are System generated Identifiers.

→ In Star Schema, Each dimension is represented by only one table & Each table Contains a Set of attributes. For Ex:- the "Location" Dimension table Contains the attribute Set [location_key, Street, City, State, Country]. This Constraint May introduce Some Redundancy.

* Snowflake Schema :- It is a Variant of Star-Schema Model, where Some dimension tables are Normalized, thereby Splitting the data into additional tables.

→ The Major difference between the Star & Snowflake Schema Model is that the Dimension tables of the Snowflake Model May be kept in Normalized form to reduce redundancies. Such a table is Easy to Maintain & Saves Storage Space.

→ The Snowflake Structure Can reduce the Effectiveness of

Browsing, since more joins will be needed to execute a query. The system performance may be adversely impacted. Hence, although the Snowflake Schema reduces redundancy, it is not as popular as Star Schema in Data Warehouse Design.

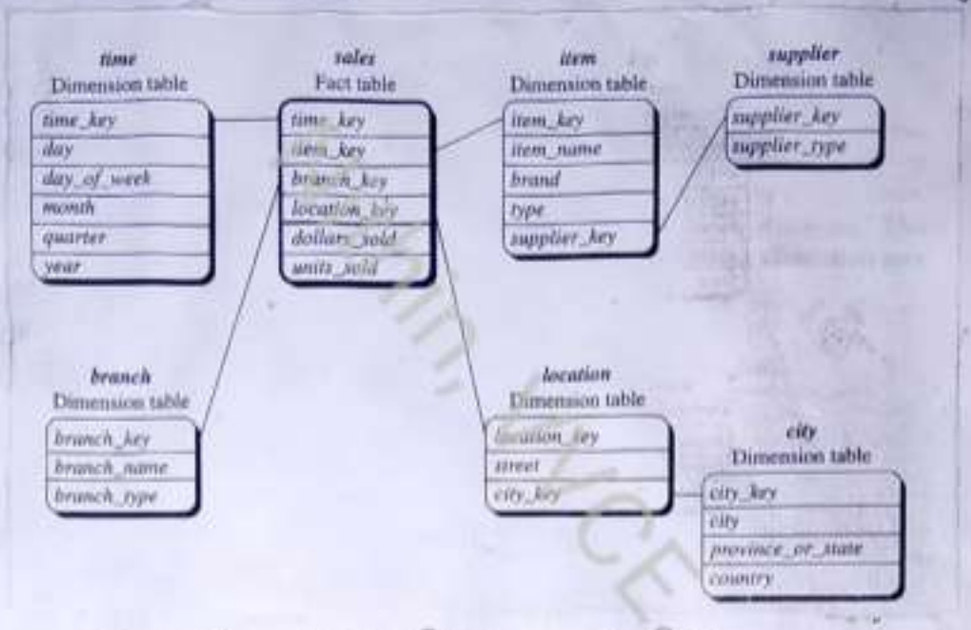


Fig:- Snowflake Schema of Sales

As shown in above Snowflake Schema for "All Electronics", the Sales fact table is identical to the Star Schema.

→ The main difference between Star & Snowflake Schema is Definition of Dimension table.

* Dimension table of "Item" in Star Schema is Normalized in Snowflake Schema, resulting table forms "Item" and "Supplier" Dimension tables.
* Similarly Dimension table of "Location" in Star Schema is Normalized in Snowflake Schema, resulting table forms "Location" & "City" Dimension tables.

* Fact Constellation :-

For Each Star Schema or Snowflake Schema, it is possible to construct "Fact Constellation Schema".

→ The fact Constellation Architecture contains Multiple Fact tables that share many Dimensions.

→ Consider the foll Example of Fact Constellation

Fig. Fact Constellation Schema of Sales

The above Schema Specifies two Fact tables "Sale" & "Shipping"

→ The Shipping table Consists of 5 Dimensions or keys & Two Measures

* Dimensions → Item_key, Time_key, Shipper_key, to_location From_Location
* Measures → Dollars_Cost, Units_Shipped.

A Fact Constellation Schema allows Dimension tables to be Shared between Fact tables.

→ A "Data Warehouse" Collects Information about Subject, that Span whole organization Such as Customers, Sales, Shipping Etc.

For data Warehouses, the fact Constellation Schema is Commonly used to Model Multiple Interrelated Subjects.

→ A "Data Mart" is a Subset of Data Warehouse that Focuses on Selected Subjects.

For Data Mart, the Star or Snowflake Schema is Commonly Used Since both are used to Model Single Subjects.

* Role of Concept Hierarchies:-

A "concept Hierarchy" defines a Sequence of Mappings from a Set of low-level concepts to High-level More general concepts.

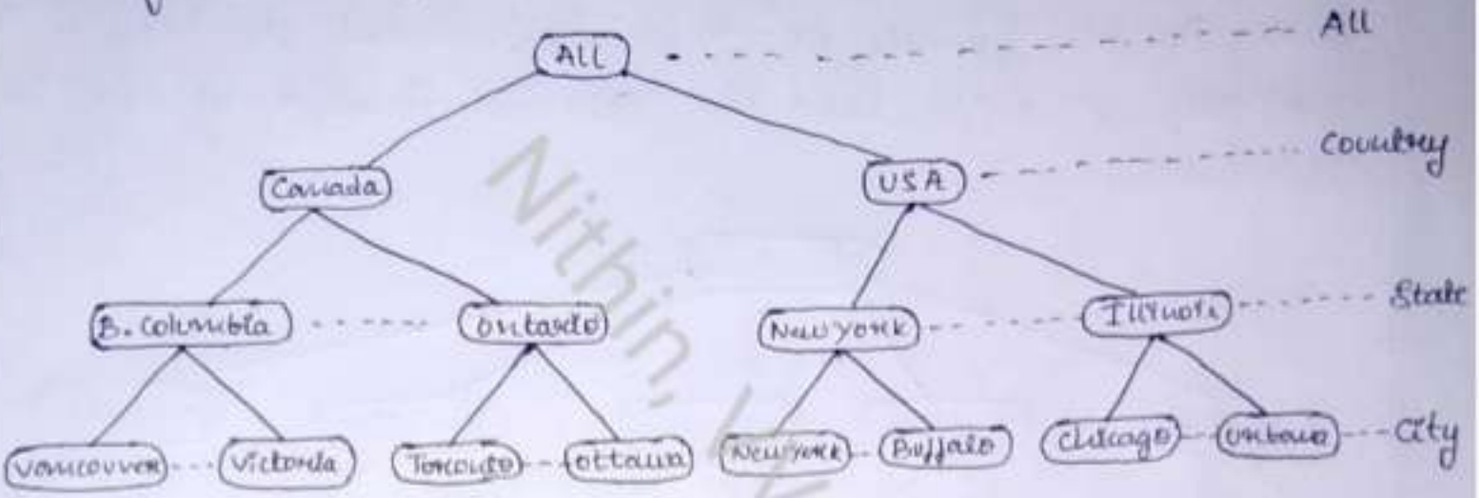


Fig:- Concept Hierarchy of location

Mapping a Set of low-level concepts (Cities) to High-level More general Concepts (Countries).

→ Many concept hierarchies are Implicit within the database Schema

→ Due to Space Limitations, not all of the hierarchy nodes are Shown we can use lattice Structures of attributes in Warehouse dimension.

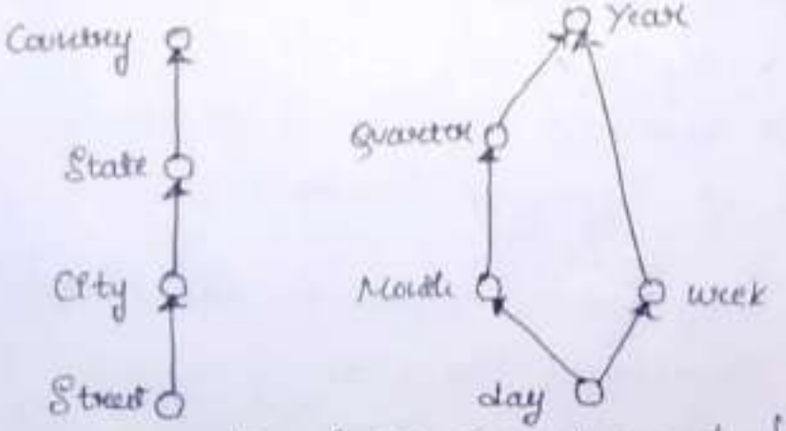


Fig:- lattice Structures of location & time

→ Concept Hierarchies May also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a "Set-Grouping Hierarchy".

For the dimension "price" with Interval ($x ... $y) denote the range from $x (exclusive) to $y (inclusive)

* Role of Concept Hierarchies:-

A "concept Hierarchy" defines a Sequence of Mappings from a Set of low-level concepts to High-level More general concepts.

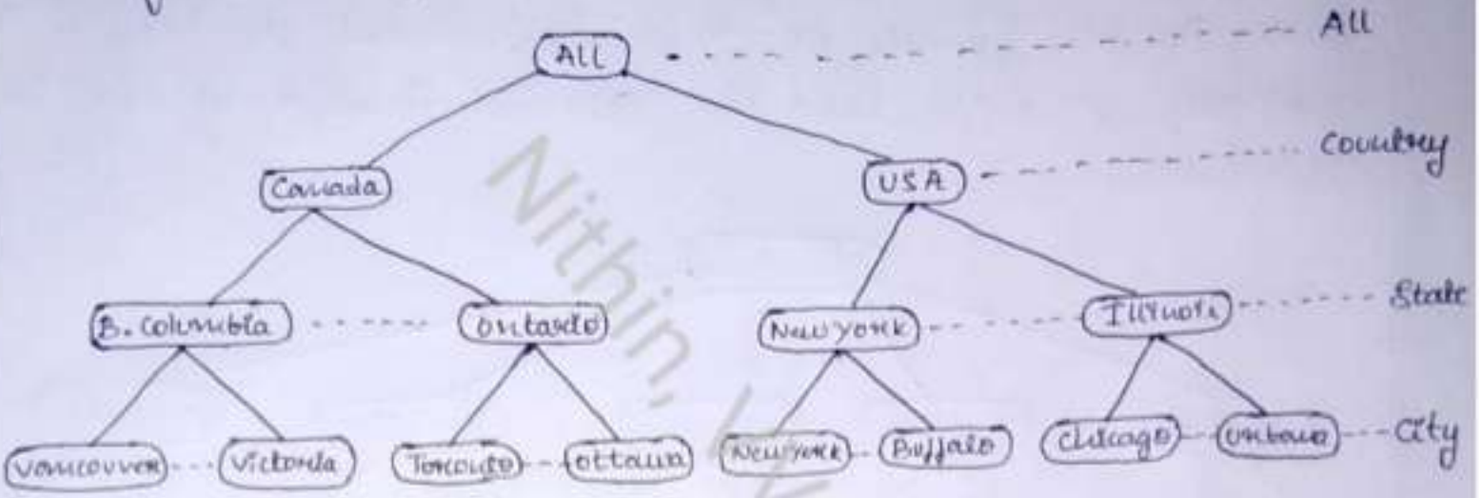


Fig:- Concept Hierarchy of location

Mapping a Set of low-level concepts (Cities) to High-level More general Concepts (Countries).

→ Many concept hierarchies are Implicit within the database Schema

→ Due to Space Limitations, not all of the hierarchy nodes are Shown we can use lattice Structures of attributes in Warehouse dimension.

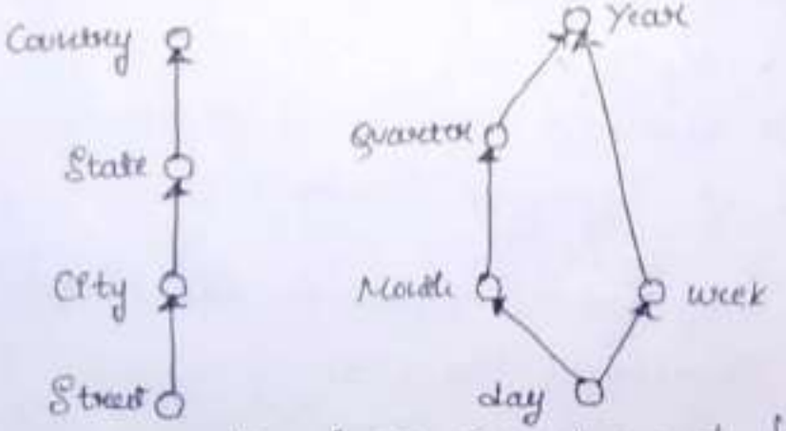


Fig:- lattice Structures of location & time

→ Concept Hierarchies May also be defined by discretizing or grouping values for a given dimension or attribute, resulting in a "Set-Grouping Hierarchy".

For the dimension "price" with Interval ($x ... $y) denote the range from $x (exclusive) to $y (inclusive)

→ There May be More than one Concept Hierarchy for a given attribute or Dimension based on different user View-points.

Concept Hierarchies May be provided Manually by System-Users, Domain Experts or Knowledge Engineers or May be Auto-matically generated based on Statistical Analysis of the Data Distribution.

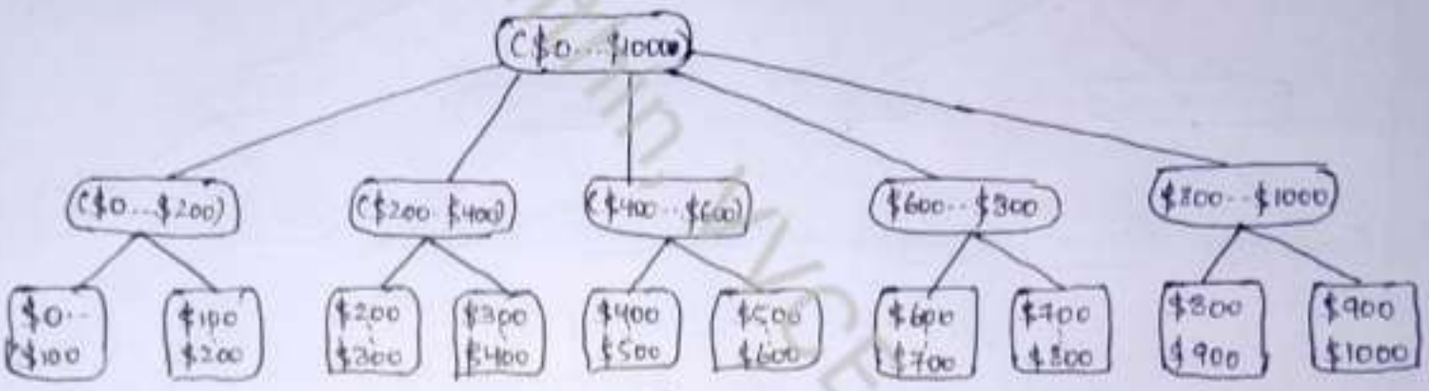


Fig:- Set Grouping Hierarchy of price

* Measures: Their Categorization & Computation:-

Measures can be organized into three categories such as

* Distributive
* Algebraic
* Holistic

Based on the kind of Aggregate Functions used.

* Distributive:- An Aggregate Function is distributed, if it can be computed in a Distributed Manner.
Suppose the data are partitioned into 'n' sets, we apply the Function which results in n Aggregate Values.
→ If the result derived by Applying the Function to the 'n' Aggregate values is the same as that derived by applying the Function to entire Data Set in Distributed Manner.
→ Some of good Examples of Distributive Aggregate Functions are

"count()", "Min()", & "Max()".

* Algebric :- An Aggregate Function is Algebric, if it can be computed by an Algebric function with M Arguments, each of which is obtained by applying a distributive Aggregate Functions.

→ A Measure is "Algebric" if it is obtained by Applying an Algebric Aggregate Function.

For Ex:- "Avg()" can be computed by Sum()/count(), where both Sum() & count() are distributive Aggregate Functions.

* Holistic :- An Aggregate Function is Holistic, if there is no constant bound.

There does not Exist any Algebric function that characterize the computation Such Measures are called "Holistic".

→ Some of good Examples of Holistic Functions Include "Median()", "Mode()" & "Rank()".

More large data Cube Application requires Efficient Computation of distributive & Algebric Measures. Many efficient techniques for this Exist.

In contrast, it is difficult to Compute Holistic Measures Efficiently.

* Typical OLAP operations :-

Number of operation May be applied to Data-Cubes are

* Roll-up
* Drill-Down
* Slice & Dice
* Pivot or Rotate

Consider the foll Example of "All Electronics" Sales Data Cube which

Contains the Dimension [location, Time & Item].

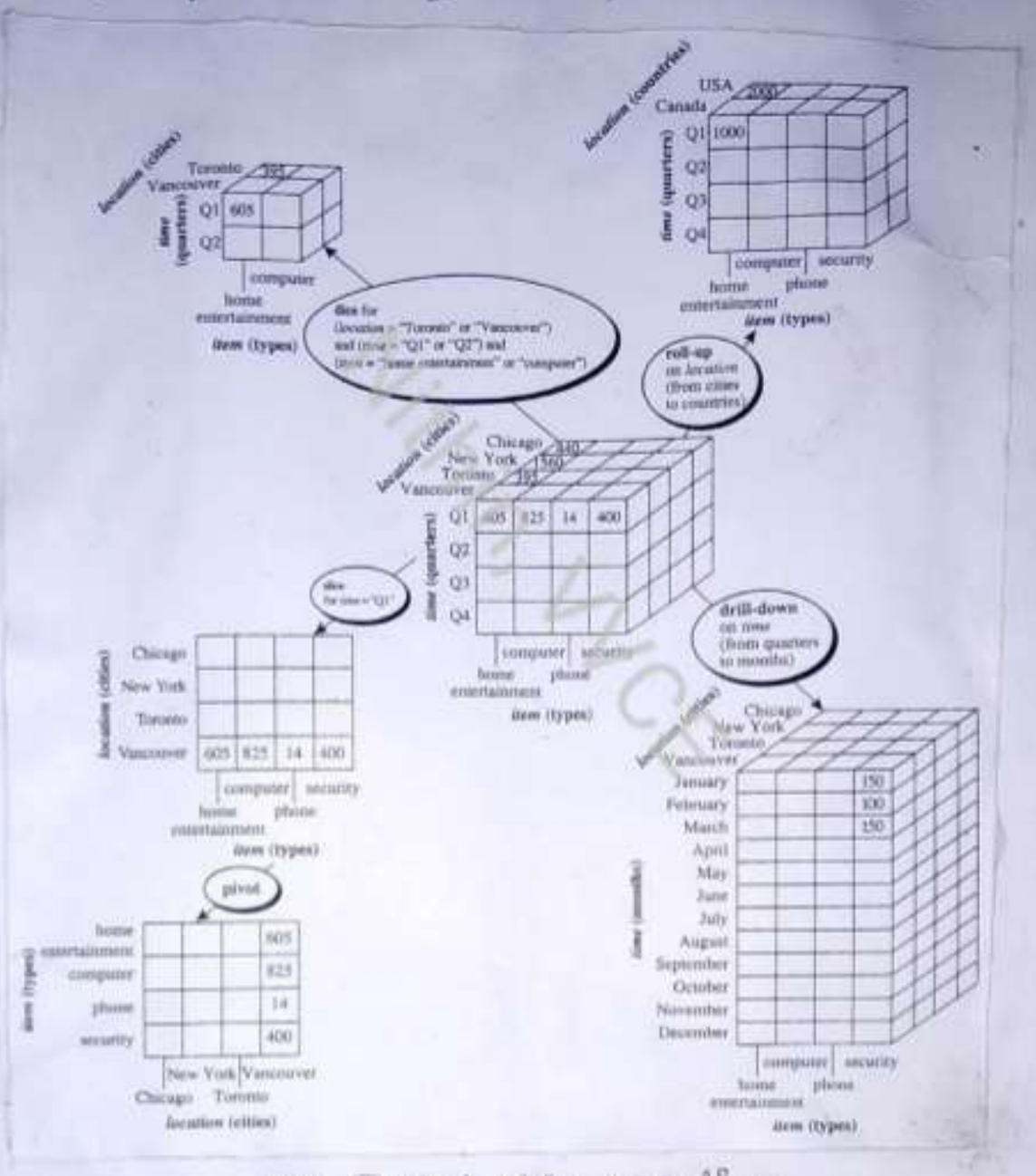


Fig: Typical OLAP operations

* Roll-up :- It is like "Zooming-out" on the Data Cube, It is required when the user needs further Abstraction or Less Detail.
→ This operation performs Further Aggregation on the Data.
For Ex. Roll-up on location from "cities" to "Countries"

* Drill-Down :- It is like "Zooming-In" on the Data Cube, i.e Reverse of roll-up.
→ It is an Appropriate operation when the user needs further details or when user wants to partition More Finely or

Wants to focus on some particular values of certain Dimensions

→ For Ex, if the time in data cube is represented in quarters then it can be drilled down or zoomed-in the time to Month wise.

* Slice & Dice :- Slice & Dice are operations for Browsing the data in the Cube. the terms refers to the ability to look at information from different Viewpoints.
  * A "Slice" is a Subset of Cube Corresponding to a Single Value for one or More Members of Dimension.
  * A "Dice" operation is Similar to Slice but dicing does not involve reducing the no of dimension. A dice is obtained by performing a Selection on two or More Dimensions.

* Pivot or Rotate :- Pivot operation is used when the user wishes to re-orient the View of Data Cube. It may involve Swapping the rows & Columns. or Moving one of row dimension into Column dimension.

* Data mining :-

Data Mining is a technology that include traditional Data analysis Method with Sophisticated Algorithms for processing large volume of data.

→ It has also opened up Exciting opportunities for Exploring & Analyzing new types of data & for analyzing diff types of data in new ways.

"Data Mining is a process of automatically discovering useful Information in Large Data Repositories".

* Data Mining & Knowledge Discovery :- Data Mining is an integral part of knowledge Discovery in Database (KDD). which is a overall process of converting raw data into useful info.

→ This process consists of a series of Transition Steps from "Data preprocessing" to "post processing of Data Mining Results".

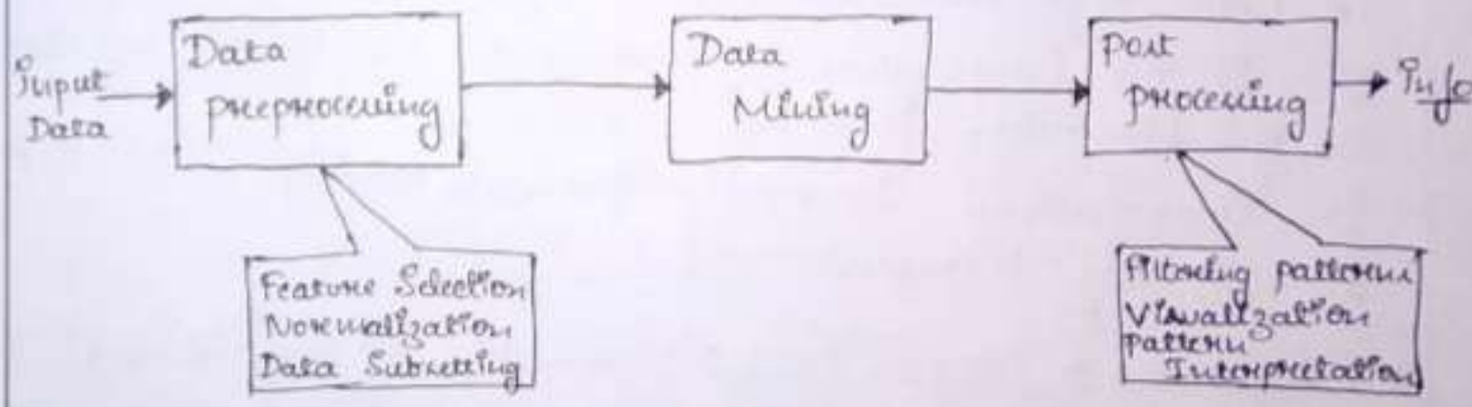


Fig :-

The Input data can be Stored in a Variety of files formats Such as Flat files, Spread Sheets or Relational tables.

→ The purpose of preprocessing is to Transform the raw Input data into an appropriate Format for Subsequent Analysis.

The Steps Involved in Data preprocessing are

* Fusing Data from Multiple Sources.

* Cleaning Data to Remove noise.
* Selecting Records & features to data Mining, this is the most time Consuming Step in knowledge discovery process.

→ "post processing" Steps that Ensures that only valid & useful Results are Incorporated into the Decision Support System.

* Challenges :-
The Motivating challenges of Data Mining are.

* Scalability :- Advances in Data generation & collection of data sets with Sizes of Giga bytes, Terabytes or Even Petabytes are become Common.

→ If the data Mining Algorithms are to handle these Massive data Sets, then they Must be Scalable.

* High Dimensionality :- It is now Common to Encounter data Sets with hundreds or thousands of Attributes

→ Data Sets with temporal or Spatial Components also tend to have High Dimensionality.
Ex:- If the temperature Measurements are taken Repeatedly w.r.t of Dimensions Increases.

→ The Computational Complexity Increases Rapidly with the Dimensionality Increase.

* Heterogenous & Complex Data :- Traditional Data Analysis Methods often Deal with data Sets Containing Attributes of the Same type, Either Continuous or Categorical.

→ The role of Data Mining in Business, Science, Medicine & other fields have grown, so has the need for Techniques that handle heterogenous Attributes.

* Data Ownership & Distribution :- Sometimes the Data Needed for Analysis is not Stored in one location or owned by one org Instead data is Geographically Distributed.

→ The key challenges faced by distributed data Mining Algorithms include

* How to reduce the Amount of communication required to perform Distributed computation.
* How to Efficiently Consolidate the Data Mining results obtained from Multiple Sources.
* How to Address Data Security Issues.

* Non-Traditional Analysis :- Traditional Statistical Approach is based on a "hypothesize-and-Test" paradigm.

→ Current Data Analysis tasks often requires the Generation & Evolution of thousands of hypothese & Consequently the development of some data Mining techniques has been Motivated. It is a non-traditional Approach.

* Data Mining Tasks :- The Data Mining tasks can be classified generally into two types based on what a Specific task tries to achieve.

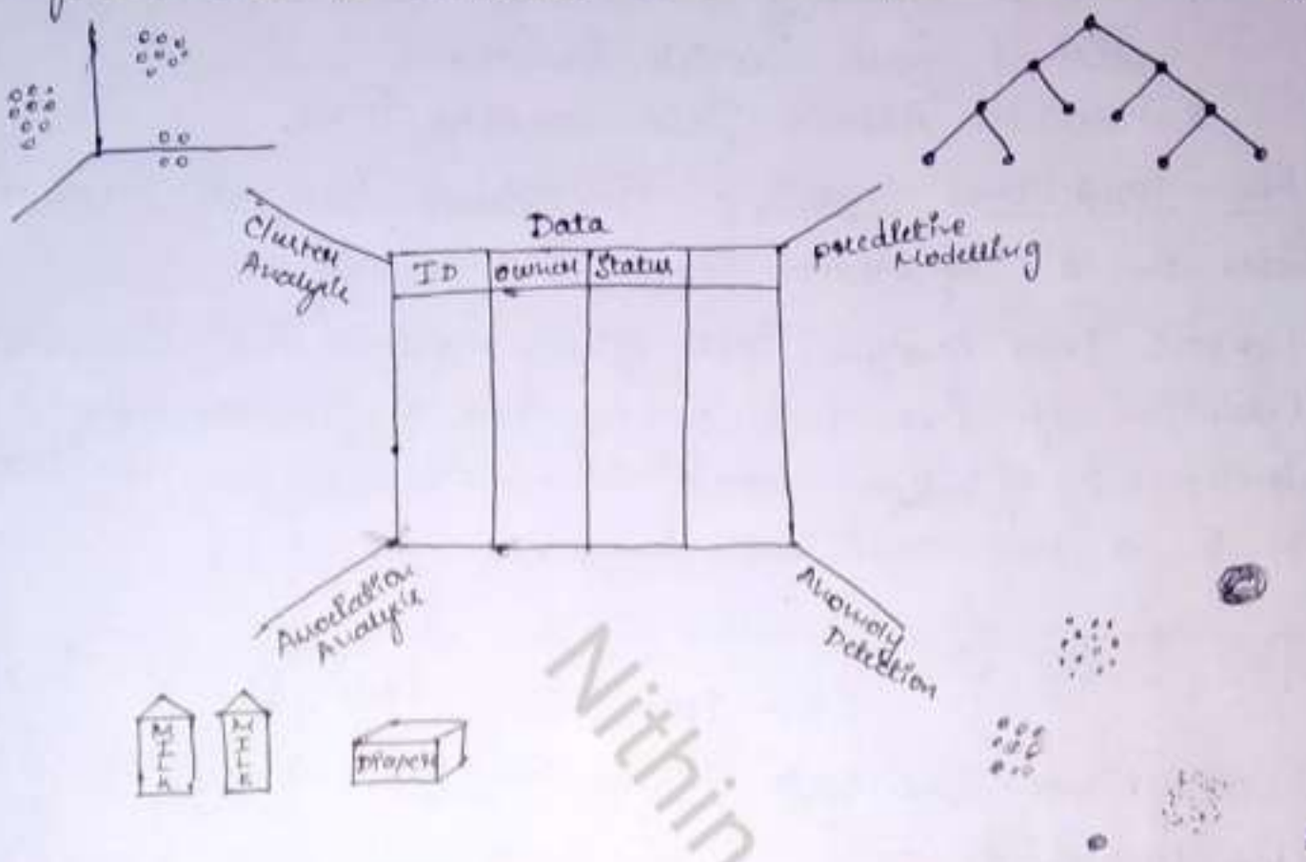
* predictive Tasks
* Descriptive Tasks.

There are a number of Data Mining tasks such as Classification, prediction, Time-Series Analysis, Association, Clustering, Summarization etc. All these tasks are Either predictive tasks or Descriptive tasks.

* predictive Tasks → Usually from the available data set that is helpful in predicting unknown or future values of another data set of interest.
A Medical practitioner trying to diagnose a disease based on the Medical test results of a patient can be considered as a predictive Data Mining task.

* Descriptive Tasks → Usually finds data describing patterns & comes up with new, significant information from the Avail -able Data Set.

A retailer trying to identify products that are purchased together can be considered as a descriptive data Mining Task.



* predictive Modeling :- It a process that uses Data Mining & probability to Forecast outcomes.

→ There are two types of predictive Modeling

* Classification → used for Discrete Variables
* Regression → used for continuous Variables.

The Goal of both tasks is to learn a Model, in which Error between the predicted & True values is low.

→ For Example, a Model can predict the Income of an Employe based on Education, Experience & other demographic factors like place of Stay, gender Etc.

* Anomaly Detection :- It is a task of identifying observations whose characteristics are significantly different from the rest of Data.
It is also called as "outlier Detection".

→ A good Anomaly detector must have a High detection rate & a Low fake alarm rate.
The good anomaly detection Algo is to discover Real Anomalies & Avoid Fake Labelling.

→ Typically the anomalous items will translate to some kind of problem such as bank fraud, structural defect, Medical problems Etc

* Association Analysis :- Association discovers the association or connection among a set of items. Association identifies the relationships among objects.

→ Association Analysis is used for Commodity Management, Advertising, direct Marketing Etc.

→ For Example, A retailer can identify the products that normally customers purchase together or Even find the Customers who respond to the promotion of same kind of products.

* Clustering Analysis :- It is used to identify data objects that are similar to one another.

→ The similarity can be decided based on a no of factors like purchase behaviour, responsiveness, geographic location & so on.

→ For Example, An Insurance Company can cluster its Customers based on age, residence, Income Etc.

* Types of Data :-
Data Set can be often be viewed as a Collection of data objects. other names for a data object are "record", "point", "vector", "pattern", "Event", "case", "Sample" or "observation".

Data can be classified into two broad types such as

* Qualitative
* Quantitative (numeric)

* Qualitative Data :- arise when the observations fall into separate distinct categories.

→ for Ex:- Colour of Eyes – blue, green brown
Exam result – Fail, pass

Such data are inherently "discrete", in that there are finite number of possible categories into which each observation may fall.

* Quantitative Data :- or numerical data arise when the observations are "Counts" or "Measurements".

→ The data are said to be "discrete" if the Measurements are integers (Ex No of people in a Home) &
"Continuous" if the Measurements can take on any value, usually within some range (Ex weight).

* Attribute & Measurement of Data :- An "Attribute" is a property or characteristic of an object that may vary, either from one object to another or from one time to another.

→ For Ex: Eye colour varies from person to person, while the temperature of an object varies over time.

→ "Measurement Scale" is a rule (Function) that associates a numerical or symbolic value with an attribute of an object.

For Ex:- we count the no of chairs in a room to see if there will be enough to seat all the people coming to a meeting.

* Different Types of Attributes :- A useful way to specify the type of an attribute is to identify the properties of numbers that correspond to underlying properties of the attribute.

→ Distinctness = & ≠ (Nominal) : the values of a nominal attribute are just different names
Nominal values provide only enough info to distinguish one object from another (=, ≠)
For Ex:- Zip code, Employee ID

→ order < ≤ > & ≥ (ordinal) : The values of an ordinal attribute provide enough info to order objects.
For Ex:- Grades, Street Numbers

→ Interval + & - (Addition) : For interval attributes, the differences between values are meaningful, i.e a unit of measurement exists (+, -)
For Ex:- Calender dates.

→ Multiplication * & / (Ratio) : For ratio variables both differences & ratios are meaningful (*, /)
For Ex:- Counts, Age, Mass Etc.

* General Characteristics of Data Sets :- There are 3 characteristics that apply to many data sets & have a significant impact on the data Mining techniques.

* Dimensionality
* Sparsity
* Resolution

→ Dimensionality : refers to how many attributes a dataset has.
For Ex, Healthcare data is notorious for having vast amounts of variables (Blood pressure, Weight, Cholesterol level)
This data could be represented in Spreadsheet, with one column representing each dimension.

→ Sparsity :- It is one in which a relatively high percentage of the Variables cells do not contain actual data.
For Ex:- a new variable determined by MONTH for which you do not have data for part Months.

→ Resolution :- It is frequently possible to obtain data at different levels of resolution, & often the properties of the data are different at different resolutions.
For Ex:- the Surface of Earth seems very uneven at a resolution of few Meters, but is relatively smooth at a resolution of tens of kilometers.

* Types of Data Sets :- There are Many types of data sets & as the field of Data Mining develops & Matures, a greater Variety of Data Sets become available for Analysis.

→ Record Data :- It is a collection of records (data objects), Each of which consists of a fixed set of data fields (Attributes).
There is no Explicit relationship among records or data fields, & Every record has the Same set of attributes.
→ Record data is usually Stored Either in flat files or in rel-tional databases.
→ Different types of record data are
  * Transaction Data → where Each record (Transaction) involves a set of items.
  * Data Matrix → Collection of Data with Same fixed set of Numeric attributes.
  * Document - Term Matrix → where Each term is a Component (Attribute) of the Vector.
→ Graph-Based Data :- A graph can Sometimes be a Convenient & powerful representation of Data.

→ There are two diff types of Graph-Based Data

* Data with relationship among objects → relationship among objects frequently convey important information.
  Ex: Web page in World Wide Web

* Data with objects that are Graphs → if objects contain subobjects that have relationships, then such objects frequently represented as Graphs.

→ Ordered Data: In one in which the attributes have relationship that involve order in time or space

→ Different types of ordered Data are

* Sequential Data → an extension of record data, where each record has a time associated with it.
  Ex: retail transaction data with time of transaction.

* Time Series Data → is a special type of Sequential Data in which each record is a time series.
  a series of measurements taken over time.

* Spatial Data → some objects have spatial attributes, such as position or areas, as well as other types of attributes.
  Ex: Weather Data.

* Data Quality :-
Today real-world databases are highly susceptible to noisy, Missing & inconsistent Data due to their typically huge size & their likely origin from multiple heterogeneous sources.

→ The Data Mining Applications focus on

* the detection & correction of Data quality problems
* Use of Algorithms that can tolerate poor Data quality

"Detection" & "correction" is often called "Data cleaning".

* Measurement & Data Collection Errors :- refers to any problem resulting from the Measurement process.
A common problem is that the value recorded differs from the true value to some extent.

→ The term "data collection Error" refers to Errors such as omitting data objects or attribute values or inappropriately including data object.

* Noise and Artifacts :- Noise is the random component of a Measurement Error. It may involve the distortion of a value or addition of spurious objects.

Fig: Time Series

Fig: Time Series with Noise

→ Data Errors may be the result of a more deterministic phenomenon. Such deterministic distortion of the data are often referred to as 'Artifacts'
Ex: A Streak in same place on a set of photographs.

* Precision, Bias & Accuracy :- "precision" is a closeness of repeated Measurements to one another.
"Bias" is a systematic variation of Measurements from the quantity being Measured.

→ precision is often Measured by the Standard deviation of a set of values, while bias is Measured by taking the difference between the Mean of the set of values & quantity being Measured.

→ It is common to use General term, "Accuracy" to refer the degree of Measurement Error in data.
Accuracy depends on precision & Bias.

* Outliers :- An outlier is an observation point that is distant from other observations (Data).
An outlier may be due to variability in the measurement or it may indicate Experimental Error.
→ The latter are sometimes, outliers will be excluded from the data set.
An outlier can cause serious problems in Analysis.

* Missing Values :- Imagine you need to analyze an Enterprise data, you note that many tuples have no recorded value for several attributes in such case, fill the Missing values by fall Methods
1. Ignore the tuple → Usually done when class label is Missing, this Method is very suggestive.
2. fill the Missing value Manually → Its very time Consuming & not feasible.
3. Use a global Constant to fill Missing value → Replace all Missing values by some Constant such as "Unknown" or "∞" this Method is simple, Its not fool proof.
4. Use a Measure of Central tendancy for the attribute (Mean or Median) to fill Missing values.
5. Use the Most probable Values to fill the Missing value → this may be determined with regression, Interface based tools using Bayesian or Decision tree.

* Inconsistent Values :- Data can contain inconsistent values.
Some types of inconsistent values are Easy to detect.
For Ex :- A person's Height should not be Negative
Once An inconsistency has been detected, it is sometimes possible

to Connect the data. The Connection of an Inconsistency requires additional or redundant Information.

* Duplicate Data :- A data Set May Include data objects that are duplicates, or almost duplicates of one another.

→ To detect & Eliminate Such duplicates, two Main Issues Must be addressed

* If there are two objects that actually represent a Single object with different values, & these Inconsistent values Must be resolved
* Care to be taken to avoid accidentally Combining data - objects that are Similar, but not duplicate

The "Deduplication" is often used to deal with duplicate data.

* Data preprocessing :-

is a Data Mining technique that involves transforming raw data into an Understandable format.

Real-world data is often Incomplete, Inconsistent, and/or lacking in certain behaviors or trends, & is likely to Contain Many Errors.

Data preprocessing is a proven Method of resolving Such Issues.

→ Some of the Most Important Approaches for Data preprocessing are

* Aggregation
* Sampling
* Dimensionality reduction
* Feature Subset Selection
* Feature Creation
* Discretization & Binarization
* Variable Transformation

These Approaches fall into two categories, Selecting data objects & Attributes for Analysis or Creating/changing the attributes.