MODULE I

CHAPTER 1

Data Warehouse (DWH) Fundamentals with Introduction to Data Mining

University Prescribed Syllabus w.e.f Academic Year 2021-2022

DWH characteristics, Dimensional modeling : Star, Snowflakes, OLAP operation, OLTP vs OLAP Data Mining as a step in KDD, Kind of patterns to be mined, Technologies used, Data Mining applications.

Self-learning Topics : Data Marts, Major issues in Data Mining.

Module 1

1.1 INTRODUCTION TO DATA WAREHOUSE

- Data Warehouse is a relational database management system (RDBMS) constructed to meet the requirement of transaction processing systems.
- It can be loosely described as any centralized data repository which can be queried for business benefits.
- It is a database that stores information oriented to satisfy decision-making requests. It is a group of decision support technologies that targets to enabling the knowledge worker (executive, manager, and analyst) to make superior and higher decisions.
- So, data warehousing support architectures and tool for business executives to systematically organize, understand and use their information to make strategic decisions.
- It includes historical data derived from transaction data from single and multiple sources.
- According to William H. Inmon, **"A data warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making process."**
- The four keywords subject-oriented, integrated, time-variant and non-volatile distinguish data warehouses from other data repository systems, such as relational database systems, transaction processing systems, and file systems.

1.1.1 Features of Data Warehouse

GQ. Why is data integration required in a data warehouse more so than in an operational application?

The key features of a data warehouse are discussed below :

1. Subject-Oriented

- A data warehouse target on the modeling and analysis of data for decision-makers.
- Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations.
- This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.

2. Integrated

- A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records.
- It requires performing data cleaning and integration during data warehousing to ensure consistency in naming conventions, encoding structures, attributes measures, etc., among different data sources.
- Operational back-ends do have integrations with other systems but generally speaking the integration breadth compared to data warehouse is much smaller because of the limited scope in particular with micro services.
- In contrast the scope of a data warehouse is very wide and encompasses practically all important operational systems. Thus data warehouse systems ingest data from practically all important operational systems to power analytics with a broad and complete picture of all enterprise data plus other data sources beyond the scope of the enterprise.
- Ultimately a lot of heterogeneous data needs to be integrated and combined. Data historization in the data warehouse requires surrogate keys or artificial hash keys.
- Data across separate sources needs to be aligned and harmonized, and standardized. The need for data cleansing and data quality control is significant. **This is why integration work and scope in data warehousing is higher than in operational systems.**

3. Time-variant

- Historical information is kept in a data warehouse. For example, one can retrieve files from 3 months, 6 months, 12 months, or even previous data from a data warehouse.
- These varies with other transactions systems, where often only the most current file is kept.
- Every key structure in the data warehouse contains, either implicitly or explicitly, a time element.

4. Non-volatile

- The data warehouse is a physically separate data storage, which is transformed from the source operational RDBMS.

- The operational updates of data do not occur in the data warehouse, i.e., update, insert, and delete operations are not performed.
- It usually requires only two procedures in data accessing: initial loading of data and access to data.
- Therefore, the data warehouse does not require transaction processing, recovery, and concurrency capabilities, which allows for substantial speedup of data retrieval.
- Non-Volatile defines that once entered into the warehouse, and data should not change.

1.1.2 Need of Data Warehouse

Data Warehouse is needed for the following reasons:

1. Business User
2. Store historical data
3. Make strategic decisions
4. For data consistency and quality
5. High response time

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

1.1.3 Applications of Data Warehouse

Here, are most common sectors where data warehouse is used:

1. Airline
2. Banking
3. Healthcare
4. Public sector
5. Investment and Insurance sector
6. Retain chain
7. Telecommunication
8. Hospitality Industry

- **1. Airline :** In the Airline system, it is used for operation purpose like crew assignment, analyses of route profitability, frequent flyer program promotions, etc.
- **2. Banking :** It is widely used in the banking sector to manage the resources available on desk effectively. Few banks also use for the market research, performance analysis of the product and operations.
- **3. Healthcare :** Healthcare sector also use data warehouse to strategize and predict outcomes, 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 analyze tax records, health policy records, for every individual.
- **5. Investment and Insurance sector :** In this sector, the warehouses are primarily used to analyze data patterns, customer trends, and to track market movements.
- **6. Retain 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.

1.1.4 Benefits of Data Warehouse

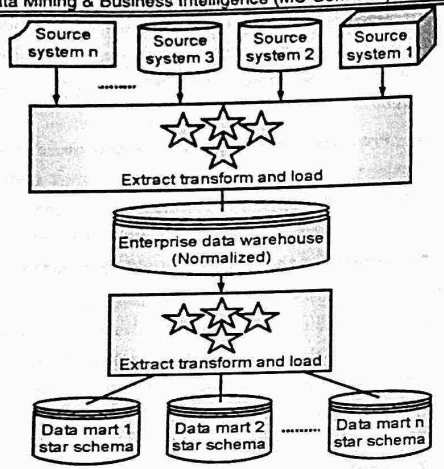
1. Delivers enhanced business intelligence
2. Saves times
3. Enhances data quality and consistency
4. Generates a high Return on Investment (ROI)
5. Provides competitive advantage
6. Improves the decision-making process
7. Enables organizations to forecast with confidence
8. Streamlines the flow of information

- **1. Delivers enhanced business intelligence :** By having access to information from various sources from a single platform, decision makers will no longer need to rely on limited data or their instinct. Additionally, data warehouses can effortlessly be applied to a business's processes, for instance, market segmentation, sales, risk, inventory, and financial management.
- **2. Saves times :** A data warehouse standardizes, preserves, and stores data from distinct sources, aiding the consolidation and integration of all the data. Since critical data is available to all users, it allows them to make informed decisions on key aspects. In addition, executives can query the data themselves with little to no IT support, saving more time and money.
- **3. Enhances data quality and consistency :** A data warehouse converts data from multiple sources into a consistent format. Since the data from across the organization is standardized, each department will produce results that are consistent. 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 help get a holistic 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 to decision makers by maintaining a cohesive database of current and historical data. By transforming data into purposeful information, decision makers can perform more functional, precise, and reliable analysis and create more useful reports with ease.
- **7. Enables organizations to forecast with confidence :** Data professionals can analyze business data to make market forecasts, identify potential KPIs, and gauge predicated results, allowing key personnel to plan accordingly.
- **8. Streamlines the flow of information :** Data warehousing facilitates the flow of information through a network connecting all related or non-related parties.

1.1.5 Approaches to Build Data Warehouse

1.1.5(A) Top-Down Approach

- This is the big-picture approach to building the overall, massive, enterprise-wide data warehouse.
- There is no collection of information sources here.
- The data warehouse is large and well-integrated.
- This approach, on the other hand, would take longer to build and has a higher failure rate.
- This approach could be dangerous if you do not have experienced professionals on your team.
- It will also be difficult to sell this approach to senior management and sponsors.
- They are unlikely to see results soon enough.



(1A1)**Fig. 1.1.1 : Top-Down Approach of Data Warehouse Design**

☞ **Advantages**

(1) Represents a data view from the perspective of the enterprise.

(2) Inherently designed—not a mash-up of disparate data marts.

(3) Data about the content is stored in a single, central location.

(4) Centralized control and rules.

☞ **Disadvantages**

(1) Even with an iterative strategy, building takes longer.

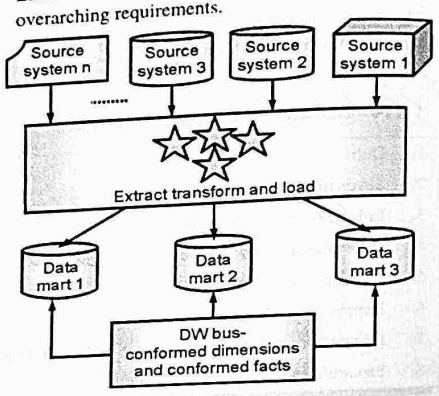
(2) High failure risk/exposure

(3) Requires a high level of cross-functional expertise

(4) Expenses are high without proof of concept.

1.1.5(B) Bottom-Up Approach

- You create departmental data marts one by one using this bottom-up method.
- To figure out which data marts to build first, you'd create a priority list.
- The most serious disadvantage of this method is data fragmentation.
- Each data mart will be blind to the organization's overarching requirements.



(1A2)**Fig. 1.1.2 : Bottom-Up Approach of Data Warehouse Design**

☞ **Advantages**

(1) Implementation of small portions is faster and easier.

(2) Favourable return on investment and proof of concept.

(3) There is a lower chance of failure.

(4) Inherently incremental; significant data marts can be scheduled first.

(5) Allows the project team to grow and develop.

☞ **Disadvantages**

(1) Each data mart has its own skewed perspective on information.

(2) Every data mart is flooded with redundant information.

(3) Perpetuates data that is inconsistent and irreconcilable.

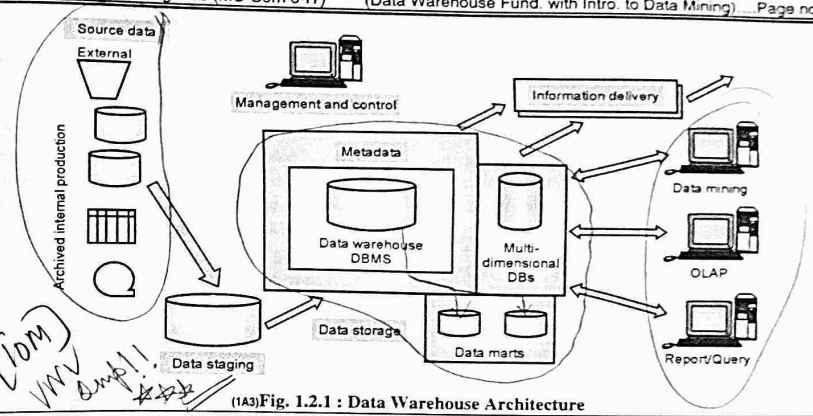
(4) Increases the number of unmanageable interfaces.

1.2 DATA WAREHOUSE ARCHITECTURE

GQ. Consider Metadata as an equivalent of Amazon book store, where each data element is book. What this metadata will contain? Explain.

We put together numerous components to make up an operational system like order entry, claims processing, or a savings account. The GUI (graphical user interface) is used to interact with users for data entry in the front-end component.

Module 1



(1A3)**Fig. 1.2.1 : Data Warehouse Architecture**

- The database management system, such as Oracle, Informix, or Microsoft SQL Server, is part of the data storage component. The display component consists of the user's screens and reports. The connectivity component consists of data interfaces and network software. We arrange these components in the most efficient manner possible, based on the information requirements and the structure of our firm.
- Architecture is the proper arrangement of the components. You put together a data warehouse using software and hardware. You organize these building blocks in a specific way to meet the needs of your company for optimal benefit.

1.2.1 Source Data Component

Source data coming into the data warehouse may be grouped into four broad categories, as discussed here.

(1) Production data	(2) Internal data
(3) Archived data	(4) External data

- **(1) Production data :** This type of information originates from the company's numerous operational systems. You select segments of data from various operational systems based on the information requirements in the data warehouse.
- **(2) Internal data :** Users preserve their "private" spreadsheets, documents, customer profiles, and sometimes even departmental databases in every company. This is internal data that could be used in a data warehouse in some form.
- **(3) Archived data :** Operational systems are largely designed to run a company's existing operations. Every operational system takes outdated data and stores it in archived files on a regular basis. After a year, certain data is archived. Data is sometimes kept in operational system databases for up to five years.
- **(4) External data :** Most executives rely on data from outside sources for a large portion of the information they use. They rely on statistics about their industry compiled by third-party organizations. They rely on competitor market share data. They use standard values of financial indicators to evaluate their company's performance.

1.2.2 Data Staging Component

- You must prepare the data for storage in the data warehouse after extracting it from various operational systems and external sources.
- The extracted data from various sources must be changed, converted, and prepared in a format suitable for storage for querying and analysis.

- To get the data ready, three major functions must be completed. You must first extract the data, then transform it and at last load it into the data warehouse storage. A staging area is where the three major functions of extraction, transformation, and loading take place. ETL
- Data staging is a place with a set of functions that clean, change, combine, convert, deduplicate, and prepare source data for storage and use in the data warehouse.

1.2.3 Data Storage Component

- The data warehouse's data storage is kept in a separate repository. Large volumes of historical data must be kept in a data warehouse's data repository for analysis.
- Furthermore, the data in the data warehouse must be kept in structures suitable for analysis rather than for quick retrieval of individual pieces of information.
- As a result, the data warehouse's data storage is kept separate from the data storage for operational systems.

1.2.4 Information Delivery Component

- The information delivery component includes various methods of information delivery in order to provide information to the large community of data warehouse users.
- Information can be delivered in the form of ad hoc reports, complex queries, multidimensional (MD) analysis, statistical analysis, enterprise information system (EIS) feeds, etc.
- This information can be delivered through online, intranet, internet or email mode.

1.2.5 Metadata Component

- Metadata is "data that describes other data". Metadata in a data warehouse is similar to the data dictionary or the data catalog in a database management system.
- In the data dictionary, you keep the information about the logical data structures, the information about the files and addresses, the information about the indexes, and so on. The data dictionary contains data about the data in the database.
- Similarly, the metadata component is the data about the data in the data warehouse.

1.2.5(A) Types of Metadata

Metadata in a data warehouse fall into **three major parts:**

(a) Operational Metadata

- As we know, data for the data warehouse comes from various operational systems of the enterprise. These source systems include different data structures. The data elements selected for the data warehouse have various fields lengths and data types.
- In selecting information from the source systems for the data warehouses, we divide records, combine factor of documents from different source files, and deal with multiple coding schemes and field lengths. When we deliver information to the end-users, we must be able to tie that back to the source data sets. Operational metadata contains all of this information about the operational data sources.

(b) Extraction and Transformation Metadata

- Extraction and transformation metadata include data about the removal of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction.
- Also, this category of metadata contains information about all the data transformation that takes place in the data staging area.

(c) End-User Metadata

- The end-user metadata is the navigational map of the data warehouses. It enables the end-users to find data from the data warehouses.
- The end-user metadata allows the end-users to use their business terminology and look for the information in those ways in which they usually think of the business.

1.2.5(B) Examples of Metadata

1. A **library catalog** may be considered metadata. The directory metadata consists of several predefined components representing specific attributes of a resource, and each item can have one or more values. These components could be the name of the author, the name of the document, the publisher's name, the publication date, and the methods to which it belongs.

2. The table of content and the index in a book may be treated as metadata for **the book**.
3. Suppose we say that a data item about **a person** is 80. This must be defined by noting that it is the person's weight and the unit is kilograms. Therefore, (weight, kilograms) is the metadata about the data is 80.
4. Another examples of metadata are data about the tables and figures in **a report** like this book. A table (which is a record) has a name (e.g., table titles), and there are column names of the tables that may be treated metadata. The figures also have titles or names.
5. Metadata for **a web page** may contain the language it is coded in, the tools used to build it, supporting browsers, etc.
6. Metadata for **a digital image** may contain the size of the picture, resolution, color intensity, image creation date, etc.
7. Metadata for **a document** may contain the document created date, last modified date, it's size, author, description, etc.

GQ. What is Metadata? Why do we need metadata when search engines like Google seem so effective?

- Metadata describes unseen HTML elements that directly communicate and clarify website information for search engines, playing a critical role in effective Search Engine Optimization for retailers. This series of micro-communications includes page titles, description tags and other protocols, and they may describe purposes, characteristics and general content.
- Metadata is a structured way to communicate information about a data set, which is used in a variety of settings with special relevance for ecommerce businesses.

Metadata for SEO and social media

- Metadata allows XML-based applications to categorize and contextualize pieces of data. For marketers, this data is usually a web pages. A search engine's job is to crawl a web page and interpret its relevancy to a given search query. While keywords matching within body content and backlinks to the page play a large role in determining ranking, metadata says more about the purpose of a page. Search engines can crawl a website and guess its general purpose based on these elements. metadata enables webmasters to tell search engines what a page's title is, which says a lot about what search queries it may be relevant for.
- Metadata is used in similar fashion by social media platforms such as Facebook. Open Graph (OG) protocol marks up web pages with information that is then displayed when a web page is shared on Facebook.

Common metadata elements

(1) **Meta tags** include basic keywords, description tags that summarize content, and robots that index pages or pass on link authority.

(2) **Title tags** are an important search engine ranking factor and should include most relevant keywords, product(s) and article name.

(3) **Image tags** identify URLs, while alternative attributes provide related text, measurements and some SEO signals.

(4) **Canonical tags** are used to consolidate similar pages and attribute the value to only one, reducing the likelihood of duplicate content penalty and providing a straightforward user experience.

(5) **Structured data** denote aspects of content that can help promote them within search results like Google answers and maps.

1.2.6 Management and Control Component

- This component of the data warehouse architecture sits on top of all the other components. The management and control component coordinates the services and activities within the data warehouse.
- This component controls the data transformation and the data transfer into the data warehouse storage.
- On the other hand, it moderates the information delivery to the users. It works with the database management systems and enables data to be properly stored in the repositories.
- It monitors the movement of data into the staging area and from there into the data warehouse storage itself.
- The management and control component interacts with the metadata component to perform the management and control functions.

- As the metadata component contains information about the data warehouse itself, the metadata is the source of information for the management module.

1.3 DATA WAREHOUSE VS DATA MARTS

- A data mart is a small, single-subject data warehouse subset that provides decision support to a small group of people.
- Data Marts can serve as a test vehicle for companies exploring the potential benefits of Data Warehouses.
- Data Marts address local or departmental problems, while a Data Warehouse involves a company-wide effort to support decision making at all levels in the organization.

Table 1.3.1 : Data Warehouse Vs Data Mart

Sr. No.	Data Warehouse	Data Mart
1.	Data warehouse is a centralized system.	Data mart is a decentralized system.
2.	In data warehouse, lightly denormalization takes place.	In Data mart, highly denormalization takes place.
3.	Data warehouse is top-down model.	Data mart is a bottom-up model.
4.	To build a warehouse is difficult.	To build a mart is easy.
5.	In data warehouse, Fact constellation schema is used.	In data mart, Star schema and snowflake schema are used.
6.	Data warehouse is flexible.	Data mart is not flexible.
7.	Data warehouse is the data-oriented in nature.	Data mart is the project-oriented in nature.
8.	Data warehouse has long life.	Data-mart has short life than warehouse.
9.	In data warehouse, data are contained in detail form.	In data mart, data are contained in summarized form.
10.	Data Warehouse is vast in size.	Data mart is smaller than warehouse.

1.4 E-R MODELLING VS DIMENSIONAL MODELLING

GQ. Why is entity-relationship modeling technique not suitable for the data warehouse? How is dimensional modeling different?

- Dimensional Modeling (DM) is a data structure technique optimized for data storage in a Data warehouse.
- The purpose of dimensional modeling is to optimize the database for faster retrieval of data. The concept of Dimensional Modelling was developed by Ralph Kimball and consists of "fact" and "dimension" tables.
- Dimensional table records information on each dimension, and fact table records all the "fact", or measures.
- A dimensional model in data warehouse is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse.
- In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.
- These dimensional and relational models have their unique way of data storage that has specific advantages.
- For instance, in the relational model, normalization and ER models reduce redundancy in data. On the contrary, dimensional model in data warehouse arranges data in such a way that it is easier to retrieve information and generate reports.
- Hence, Dimensional models are used in data warehouse systems and not a good fit for relational systems.

1.4.1 Elements of Dimensional Data Model

1. **Fact :** Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number.
2. **Dimension :** Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be :
   - Who – Customer Names
   - Where – Location

   - What – Product Name

   In other words, a dimension is a window to view information in the facts.
3. **Attributes :** The Attributes are the various characteristics of the dimension in dimensional data modeling. In the Location dimension, the attributes can be
   - State
   - Country
   - Zipcode, etc.

   Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes
4. **Fact Table :** A fact table is a primary table in dimension modelling. A Fact Table contains
   - Measurements/facts
   - Foreign key to dimension table
5. **Dimension Table**
   - A dimension table contains dimensions of a fact.
   - They are joined to fact table via a foreign key.
   - Dimension tables are de-normalized tables.
   - The dimension attributes are the various columns in a dimension table.
   - Dimensions offers descriptive characteristics of the facts with the help of their attributes.
   - No limit set for given for number of dimensions.
   - The dimension can also contain one or more hierarchical relationships.

1.4.2 Steps of Dimensional Modelling

The accuracy in creating your dimensional modeling determines the success of your data warehouse implementation. The model should describe the Why, How much, When/Where/Who and What of your business process. Here are the steps to create dimension model.

1. **Identify Business Process**
   - Identifying the actual business process, a data warehouse should cover. This could be Marketing, Sales, HR, etc. as per the data analysis needs of the organization. The selection of the business process also depends on the quality of data available for that process. It is the most important step of the data modelling process, and a failure here would have cascading and irreparable defects.
   - To describe the business process, you can use plain text or use basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML).
2. **Identify the Grain**
   - The grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse. If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.
   - During this stage, you answer questions like -
     (i) Do we need to store all the available products or just a few types of products? This decision is based on the business processes selected for data warehouse.
     (ii) Do we store the product sale information on a monthly, weekly, daily or hourly basis? This decision depends on the nature of reports requested by executives.
     (iii) How do the above two choices affect the database size?
   - Example of Grain: The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis. So, the grain is "product sale information by location by the day."
3. **Identify Dimensions and Attributes**
   - Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.
   - Example of Dimensions: The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.
     (i) **Dimensions :** Product, Location and Time
     (ii) **Attributes :** For Product: Product key (Foreign Key), Name, Type, Specifications
     (iii) **Hierarchies :** For Location: Country, State, City, Street Address, Name

4. Identify Facts

- This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse. Most of the fact table rows are numerical values like price or cost per unit, etc.
- Example of Facts : The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis. The fact here is Sum of Sales by product by location by time.

5. Build Schema

In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are two popular schemas

1. Star Schema
2. Snowflake Schema

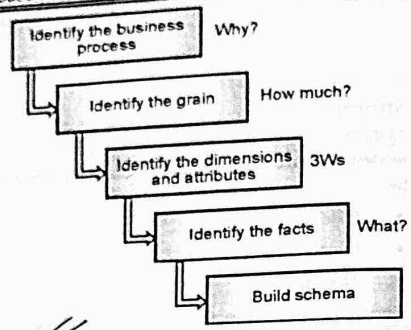


Fig. 1.4.1 : Steps in Dimensional Modeling

Table 1.4.1 : E-R Modeling Vs Dimensional Modeling

Sr. No.	E-R Modeling	Dimensional Modeling
1.	E-R modeling have **logical and physical model.**	Dimensional modeling have **physical model.**
2.	E-R modeling is used for **normalizing the OLTP design.**	Dimensional modeling is used for **de-normalizing the OLAP design.**
3.	E-R modelling revolves around the entities and their relationships to capture the overall process of the system.	Dimensional modelling revolves around dimensions(point of analysis) for decision making and not to capture the process.
4.	In E-R modeling the data is in normalized form, so more number of joins are required, which may adversely affect the system performance.	In dimensional modeling the data is de-normalized, so less number of joins are required, by which system performance will improve.
5.	In E-R modeling, a view of data is from data processing.	In dimensional modeling, a view of data is from business processing.
6.	It is not mapped for creating schemas.	It is mapped for creating schemas.
7.	It uses the current data.	It uses the historical data.
8.	Size of data varies from MBs to GBs.	Size of data varies from GBs to TBs.
9.	Data storage is volatile.	Data storage is non-volatile.
10.	High Create/Read/Update/Delete (CRUD) activity.	High Select activity.
11.	**Advantages :** Removes data redundancy. Ensures data consistency. Expresses the relationship between the entities.	**Advantages :** Captures critical measures. Views along dimensions. Useful to business users.

Module 1

1.5 INFORMATION PACKAGE DIAGRAM

- The first and most generalized level of an information model is its information package diagram. This model focuses on the data gathering activities for the users' information packaging requirements.
- An information package diagram defines the relationships between subject matter and key performance measures.
- The information package diagram has a highly targeted purpose, providing a focused scope for user requirements.
- Because information package diagrams target what the users want, they are effective in facilitating communication between the technical staff and the users, indicating any inconsistencies between the requirements and what the data warehouse will deliver.
- **Example :** Information package for analyzing sales for a certain business. It allows users to evaluate sales metrics by time, product, location, and customer demographics.

	Dimensions					
Hierarchies	**Time Periods**	**Locations**	**Products**	**Age Groups**	...	...
	Year	Country	Class	Group 1		
	Facts: Forecast Sales, Budget Sales, Actual Sales					

- The subject here is sales. The measured facts or the measurements that are of interest for analysis are shown in the bottom section of the package diagram. In this case, the measurements are actual sales, forecast sales, and budget sales.
- The business dimensions along which these measurements are to be analyzed are shown at the top of diagram as column headings. In our example, these dimensions are time, location, product, and demographic age group.
- Each of these business dimensions contains a hierarchy or levels. For example, the time dimension has the hierarchy going from year down to the level of individual day. The other intermediary levels in the time dimension could be quarter, month, and week. These levels or hierarchical components are shown in the information package diagram.
- Information Package Diagram enables you to
  a. Define the common subject areas
  b. Design key business metrics
  c. Decide how data must be presented
  d. Determine how users will aggregate or roll up
  e. Decide the data quantity for user analysis or query
  f. Decide how data will be accessed
  g. Establish data granularity
  h. Estimate data warehouse size
  i. Determine the frequency for data refreshing
  j. Ascertain how information must be packaged

1.6 DATA WAREHOUSE SCHEMAS

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema.

1.6.1 Star Schema

- A star schema is the elementary 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 or dealer credits.
- A dimension includes reference data about the fact, such as date, item, or customer.
- A star schema is a relational schema where a relational schema whose design represents a multidimensional data model. 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 center of the schema consists of a large fact table, and the points of the star are the dimension tables.

Fact Tables

- It is a table in a star schema which contains facts and connected to dimensions.
- A fact table has two types of columns: those that include fact and those that are foreign keys to the dimension table.
- The primary key of the fact tables is generally a composite key that is made up of all of its foreign keys.

Dimension Tables

- A dimension is an architecture usually composed of one or more hierarchies that categorize data.
- If a dimension has not got hierarchies and levels, it is called a **flat dimension** or **list**.
- The primary keys of each of the dimension table are part of the composite primary keys of the fact table.
- Dimensional attributes help to define the dimensional value. They are generally descriptive, textual values.
- Dimensional tables are usually small in size than fact table.

Note : Fact tables are deep whereas dimension tables are wide as fact tables will have a higher number of rows and a lesser number of columns. A primary key defined in the fact table is primarily to identify each row separately. The primary key is also called a Composite key in fact table. A dimension table contains a higher granular information so have less no of records and it needs to have all the necessary details (more columns) related to the grain of the table. On the other side, a fact table has the lowest level grain of a subject area. Lower grain causes more number of rows in the Fact table.

1.6.1(A) Characteristics of Star Schema

1. It creates a de-normalized database that can quickly provide query responses.
2. It provides a flexible design that can be changed easily or added to throughout the development cycle, and as the database grows.
3. It provides a parallel in design to how end-users typically think of and use the data.
4. It reduces the complexity of metadata for both developers and end-users.

1.6.1(B) Keys In Star Schema

1. Primary Keys
2. Surrogate Keys
3. Foreign Keys

1. Primary Keys : The primary key of the dimension table identifies each roe in a dimension table. Example: In a student dimension table, student_id is the primary key which identifies each student uniquely.

2. Surrogate Keys : System generated sequence numbers are called surrogate keys. They do not have any built in meanings.

3. Foreign Keys : Every dimension table has one-to-one relationship with the fact table. The primary key in the dimension table acts as a foreign key in the fact table.

1.6.1(C) Advantages of Star Schema

1. Query Performance
2. Load performance and administration
3. Built-in referential integrity
4. Easily Understood

1. Query Performance

- A star schema database has a limited number of table and clear join paths, the query run faster than they do against OLTP systems. Small single-table queries, frequently of a dimension table, are almost instantaneous. Large join queries that contain multiple tables takes only seconds or minutes to run.
- In a star schema database design, the dimension is connected only through the central fact table. When the two-dimension table is used in a query, only one join path, intersecting the fact tables, exist between those two tables. This design feature enforces authentic and consistent query results.

2. Load performance and administration

- Structural simplicity also decreases the time required to load large batches of record into a star schema database. By describing facts and dimensions and separating them into the various table, the impact of a load structure is reduced.

- Dimension table can be populated once and occasionally refreshed. We can add new facts regularly and selectively by appending records to a fact table.

3. Built-in referential integrity

- A star schema has referential integrity built-in when information is loaded. Referential integrity is enforced because each data in dimensional tables has a unique primary key, and all keys in the fact table are legitimate foreign keys drawn from the dimension table.
- A record in the fact table which is not related correctly to a dimension cannot be given the correct key value to be retrieved.

4. Easily Understood

- A star schema is simple to understand and navigate, with dimensions joined only through the fact table.
- These joins are more significant to the end-user because they represent the fundamental relationship between parts of the underlying business. Customer can also browse dimension table attributes before constructing a query.

1.6.1(D) Disadvantages of Star Schema

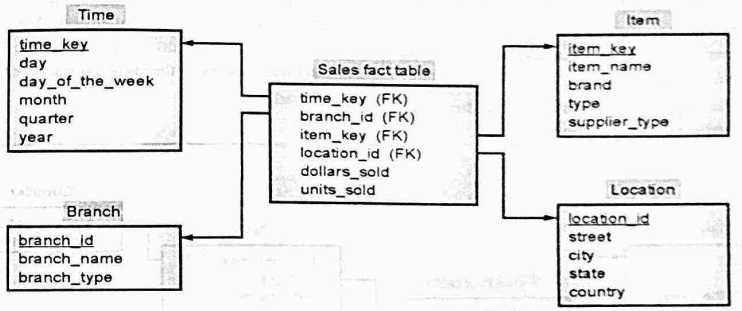
(1) Data integrity is not enforced well since in a highly de-normalized schema state.

(2) Not flexible in terms if analytical needs as a normalized data model.

(3) Star schemas don't reinforce many-to-many relationships within business entities, at least not frequently.

1.6.1(E) Example

A star schema for Digi1 Electronics sales is shown. Sales are considered along four dimensions: time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold.



(1A5)**Fig. 1.6.1 : Star Schema for Digi1 Electronics Sale**

1.6.2 Snowflake Schema

- The snowflake schema is a variant of the star schema. Here, the centralized fact table is connected to multiple dimensions. In the snowflake schema, dimensions are present in a normalized form in multiple related tables.
- The snowflake structure is materialized when the dimensions of a star schema are detailed and highly structured, having several levels of relationship, and the child tables have multiple parent table.
- The snowflake schema affects only the dimension tables and does not affect the fact tables.
- In other words, a dimension table is said to be snowflaked if the low-cardinality attribute of the dimensions has been divided into separate normalized tables. These tables are then joined to the original dimension table with referential constraints (foreign key constraint).

1.6.2(A) Characteristics of Snowflake Schema

(1) The snowflake schema uses small disk space.

(2) It is easy to implement dimension that is added to schema.

(3) There are multiple tables, so performance is reduced.

(4) The dimension table consist of two or more sets of attributes which define information at different grains.

(5) The sets of attributes of the same dimension table are being populated by different source systems.

1.6.2(B) Advantages of Snowflake Schema

(1) It provides structured data which reduces the problem of data integrity.

(2) It uses small disk space because data are highly structured.

1.6.2(C) Disadvantages of Snowflake Schema

(1) Snowflaking reduces space consumed by dimension tables, but compared with the entire data warehouse the saving is usually insignificant.

(2) Avoid snowflaking or normalization of a dimension table, unless required and appropriate.

(3) Do not snowflake hierarchies of one-dimension table into separate tables. Hierarchies should belong to the dimension table only and should never be snowflaked.

1.6.2(D) Example

- A snowflake schema for Digi1 Electronics sales is given. Here, the sales fact table is identical to that of the star schema in Fig. 1.6.1. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables.
- For example, the item dimension table now contains the attributes item_key, item_name, brand, type, and supplier_key, where supplier_key is linked to the supplier dimension table, containing supplier_key and supplier_type information.
- Similarly, the single dimension table for location in the star schema can be normalized into two new tables: location and city. The city_key in the new location table links to the city dimension. Notice that, when desirable, further normalization can be performed on state and country in the snowflake schema.

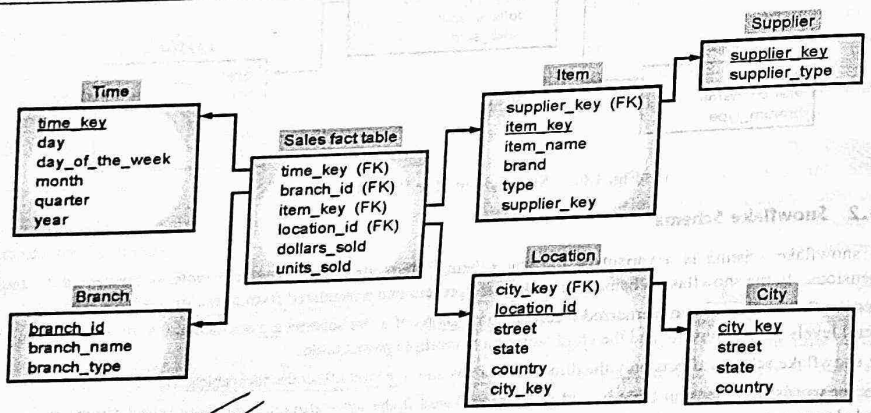


Fig. 1.6.2 : Snowflake Schema for Digi1 Electronics Sale

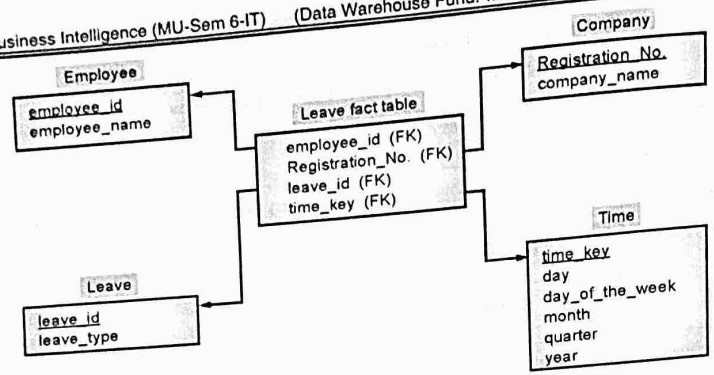
1.6.3 Star Schema Vs Snowflake Schema

Table 1.6.1 : Star Schema Vs Snowflake Schema

Sr. No.	Basis	Star Schema	Snowflake Schema
1.	Ease of Maintenance/change	It has redundant data and hence less easy to maintain/change.	No redundancy and therefore more easy to maintain and change.
2.	Ease of Use	Less complex queries and simple to understand.	More complex queries and therefore less easy to understand.
3.	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.
4.	Query Performance	Less number of foreign keys and hence lesser query execution time.	More foreign keys and thus more query execution time.
5.	Normalization	It has De-normalized tables.	It has normalized tables.
6.	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).
7.	Joins	Fewer joins	Higher number of joins.
8.	Dimension Table	It contains only a single dimension table for each dimension.	It may have more than one dimension table for each dimension.
9.	Hierarchies	Hierarchies for the dimension are stored in the dimensional table itself in a star schema.	Hierarchies are broken into separate tables in a snowflake schema. These hierarchies help to drill down the information from topmost hierarchies to the lowermost hierarchies.
10.	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.
11.	Data Warehouse system	Work best in any data warehouse/ data mart.	Better for small data warehouse/data mart.

1.6.4 Factless Fact Table

- A data warehouse factless fact table is a fact that does not have any measures stored in it. This table will only contain keys from different dimension tables. The fact-less fact is often used to resolve a many-to-many cardinality issue.
- There are two types of factless fact tables:
  1. Event capturing factless fact
  2. Coverage table – Describing condition



(1A7)Fig. 1.6.3 : Factless Fact Table for Leave

1. Event Capturing Factless Fact

- This type of fact table establishes the relationship among the various dimension members from various dimension tables without any measured value.
- For example, Student attendance (student-teacher relation table) capturing table is the factless fact. Table will have entry into it whenever student attend class.
- Following questions can be answered by the student attendance table :
  1. Which student is taught by the maximum number of teachers?
  2. Which class has maximum number of attendance?
  3. Which teacher teaches maximum number of students?
- All the above queries are based on the **COUNT ()**, **MAX ()** with **GROUP BY**.

2. Coverage Table-Describing Condition

- This is another kind of factless fact. A factless fact table can only answer 'optimistic' queries (positive query) but cannot answer a negative query.
- Coverage fact is used to support negative analysis reports. For example, an electronic store did not sell any product for given period of time.
- If you consider the student-teacher relation table, the event capturing fact table cannot answer 'which teacher did not teach any student?'
- Coverage fact attempts to answer this question by adding extra flag 0 for negative condition and 1 for positive condition.
- If the student table has 20 records and teacher table has 3 records, then coverage fact table will store 20 * 3 = 60 records for all possible combinations. If any teacher is not teaching particular student, then that record will have flag 0 in it.

1.6.5 Fact Constellation Schema

- Fact Constellation is a schema for representing multidimensional model.
- It is a collection of multiple fact tables having some common dimension tables.
- It can be viewed as a collection of several star schemas and hence, also known as *Galaxy schema*,
- It is one of the widely used schema for Data warehouse designing and it is much more complex than star and snowflake schema.
- For complex systems, we require fact constellations.

1.6.5(A) Advantages of Fact Constellation Schema

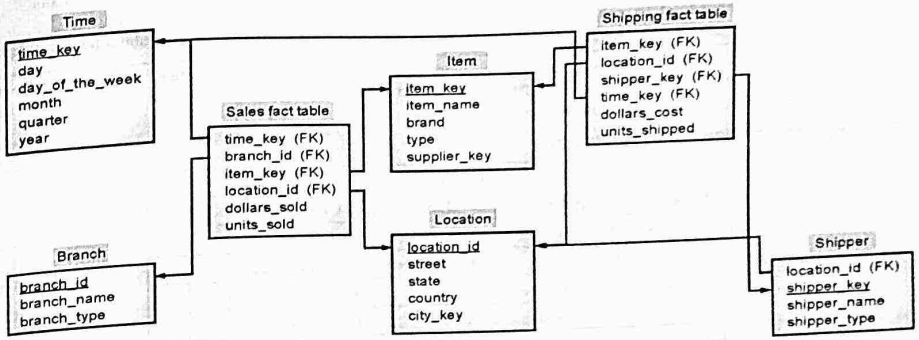
Provides a flexible schema.

1.6.5(B) Disadvantages of Fact Constellation Schema

It is much more complex and hence, hard to implement and maintain.

1.6.5(C) Example

- A fact constellation schema is shown in Fig. 1.6.4 below. This schema specifies two fact tables, sales and shipping.
- The sales table definition is identical to that of the star schema above. The shipping table has four dimensions, or keys-item_key, time_key, shipper_key, location_id and two measures dollars_cost and units_shipped.
- A fact constellation schema allows dimension tables to be shared between fact tables.
- For example, the dimension tables for time, item, and location are shared between the sales and shipping fact tables.



(1A8)Fig. 1.6.4 : Fact Constellation Schema for Digi1 Electronics Sale

1.6.6 Schema Definition

Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

Syntax for Cube Definition

```
define cube < cube_name > [ < dimension-list >]: < measure_list >
```

Syntax for Dimension Definition

```
define dimension < dimension_name > as ( < attribute_or_dimension_list > )
```

Star Schema Definition

The star schema that we have discussed can be defined using Data Mining Query Language (DMQL) as follows :

```
define cube sales star [time, item, branch, location]:
dollars_sold = sum(sales in dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week,
month, quarter, year)
define dimension item as (item_key, item_name, brand, type,
supplier_type)
define dimension branch as (branch_id, branch_name,
branch_type)
define dimension location as (location_id, street, city, state,
country)
```

Snowflake Schema Definition

Snowflake schema can be defined using DMQL as follows :

```
define cube sales snowflake [time, item, branch, location]:
dollars_sold = sum(sales in dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week,
month, quarter, year)
define dimension item as (item_key, item_name, brand, type,
supplier (supplier_key, supplier_type))
define dimension branch as (branch_id, branch_name,
branch_type)
define dimension location as (location_id, street, city
(city_key, street, state, country))
```

Fact Constellation Schema Definition

Fact constellation schema can be defined using DMQL as follows :

```
define cube sales [time, item, branch, location]:dollars_sold =
sum(sales in dollars), units sold = count(*)
define dimension time as (time_key, day, day_of_week,
month, quarter, year)
define dimension item as (item_key, item_name, brand, type,
supplier_type)
define dimension branch as (branch_id, branch_name,
branch_type)
define dimension location as (location_id, street, city,
state,country)

define cube shipping [time, item, shipper,
location_id]:dollars_cost = sum(cost in dollars), units shipped
= count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name,
location as location in cube sales, shipper_type)
define dimension location as location in cube sales
```

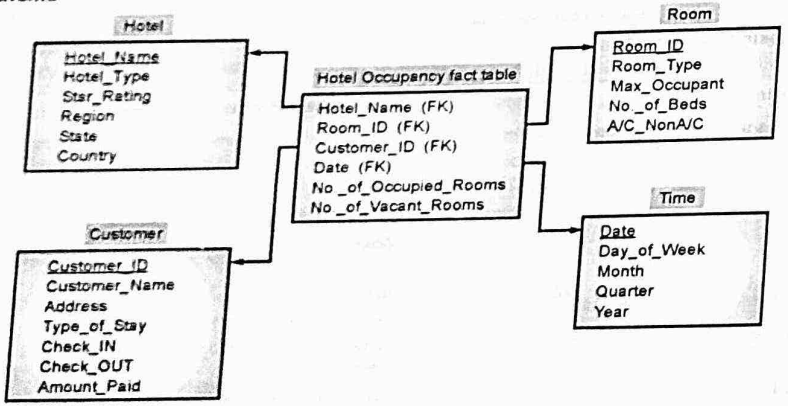
Ex. 1.6.1 : Consider a data warehouse for hotel occupancy, where there are four dimensions namely (a) Hotel (b) Room (c) Time (d) Customer and two measures (i) Occupied rooms (ii) Vacant rooms.

Draw information package diagram, star schema and snowflake schema.

☑ **Soln. : (a) Information Package Diagram**

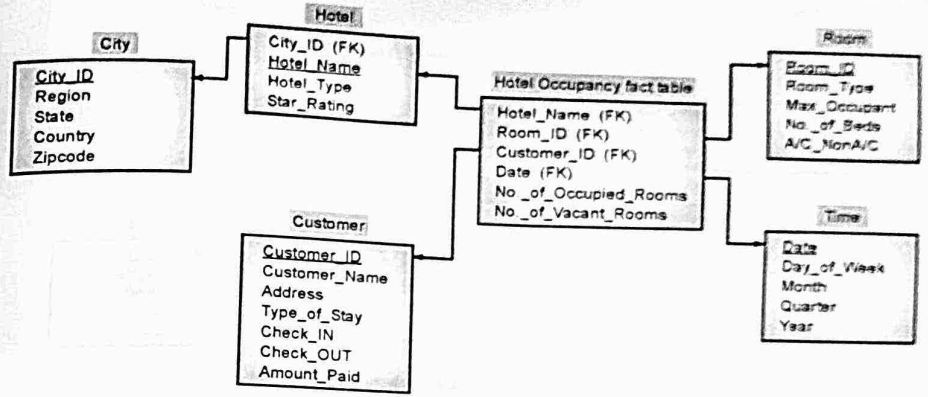
Hierarchies/Categories	Dimensions					
	Hotel	**Room**	**Time**	**Customer**	...	...
	Hotel Name	Room ID	Date	Customer ID		
	Hotel Type	Room Type	Day of Week	Customer Name		
	Star Rating	Max. Occupant	Month	Address		
	Region	No. of Beds	Quarter	Type of Stay		
	State	A/C Non A/C	Year	Check IN		
	Country			Check OUT		
				Amount Paid		
	Facts : Occupied Rooms, Vacant Rooms					

(b) Star Schema



(1A9)**Fig. P. 1.6.1(a) : Star Schema for Hotel Occupancy**

(c) Snowflake Schema



(1A10)**Fig. P. 1.6.1(b) : Snowflake Schema for Hotel Occupancy**

Ex. 1.6.2 : Consider a data warehouse for a hospital where there are three dimensions namely (a) Doctor (b) Patient (c) Time and two measures (i) count (ii) charge where charge is the fee that the doctor charges a patient for a visit.

(i) Draw star and snowflake schema.

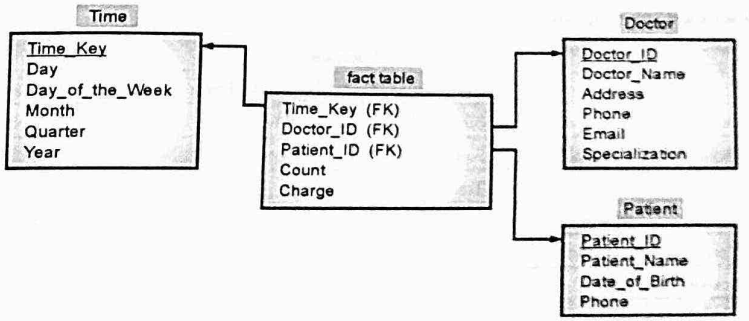
(ii) Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2010?

(iii) To obtain the same list, write an SQL query assuming the data are stored in a relational database with the schema fee (day, month, year, doctor, hospital, patient, count, charge).

☑ **Soln. :**

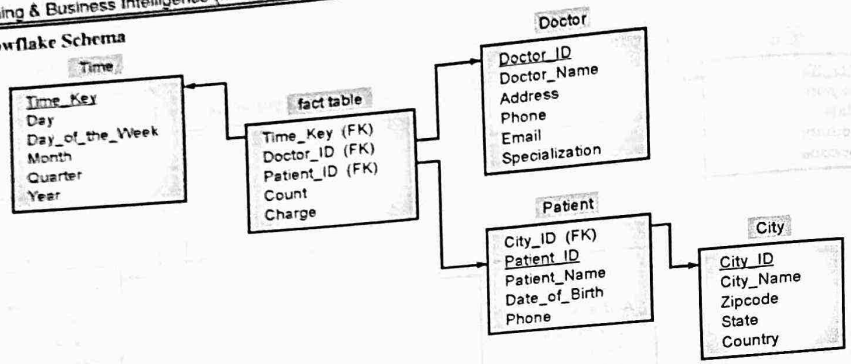
(I) Star Schema and Snowflake Schema

(a) Star Schema



(1A11)**Fig. P. 1.6.2(a) : Star Schema for Hospital**

(b) Snowflake Schema



(1A12)**Fig. P. 1.6.2(b) : Snowflake Schema for Hospital**

(ii) First, we should use roll-up operation to get the year 2010 (rolling-up from day then month to year). After getting that, we need to use slice operation to select (2010). Second, we should use roll-up operation again to get all patients. Then, we need to use slice operation to select (all). Finally, we get list the total fee collected by each doctor in 2010. So,

1. roll up from day to month to year
2. slice for year = "2010"
3. roll up on patient from individual patient to all
4. slice for patient = "all"
5. get the list of total fee collected by each doctor in 2010.

(iii) SELECT doctor, sum(charge) FROM fee WHERE year = 2004 GROUP BY doctor;

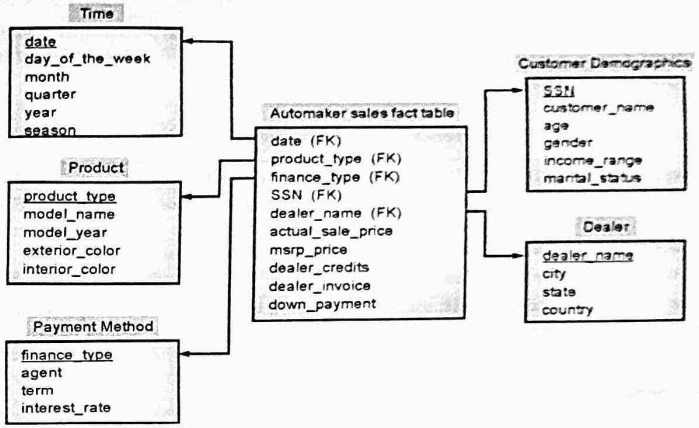
Ex. 1.6.3 : Consider a data warehouse for automaker sales where there are five dimensions, namely (a) Time (b) Product (c) Payment Method (d) Customer Demographics (e) Dealer and facts/measures like Actual Sale Price, MSRP Sale Price, Dealer Credits, Dealer Invoice, Down Payment. Draw the information package diagram and star schema.

☑ **Soln. :**

(a) Information Package Diagram

Hierarchies/Categories	Dimensions					
	Time	**Product**	**Payment Method**	**Customer Demographics**	**Dealer**	...
	Date	Product Type	Finance Type	SSN Number	Dealer Name	
	Day of week	Model Name	Agent	Customer Name	City	
	Month	Model Year	Term	Age	State	
	Quarter	Exterior Color	Interest Rate	Gender	Country	
	Year	Interior Color		Income Range		
	Season			Marital Status		
	Facts : Actual Sale Price, MSRP Sale Price, Dealer Credits, Dealer Invoice, Down Payment					

(b) Star Schema



(1A13)**Fig. P. 1.6.3 : Star Schema for Automaker Sales**

Ex. 1.6.4 : For a supermarket chain, consider the following dimensions namely product, store, time and promotion. The schema contains a central fact table for sales with three measures unit_sales, dollars_sales and dollar_cost.

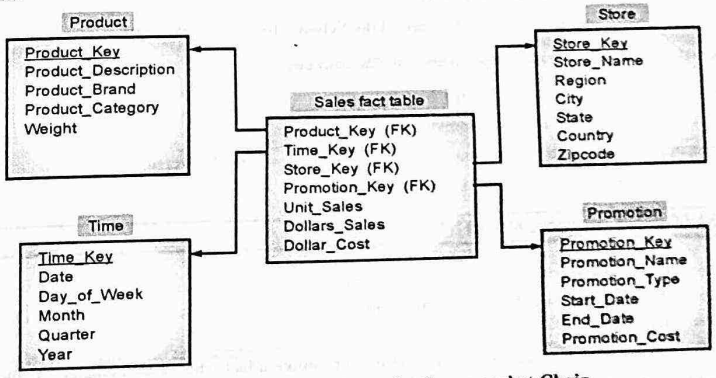
i. Draw star schema.

ii. Calculate the maximum number of base fact table records for warehouse with the following values given below:

- Time period-5 years
- Store - 300 stores reporting daily sales
- Product - 40,000 products in each store (about 4000 sell in each store daily).
- Promotion: a sold item may be in only one promotion in a store on a given day.

☑ **Soln. :**

i. Star Schema



(1A14)**Fig. P. 1.6.4 : Star Schema for Supermarket Chain**

ii. Fact table records

- Time period = 5 years = 5 × 365 days = 1825 days
- Number of Stores = 300
- Product sell in each store daily = 4000
- Promotion = 1

Therefore, maximum number of fact table records = 1825 × 300 × 4000 × 1 = 2190000000 records

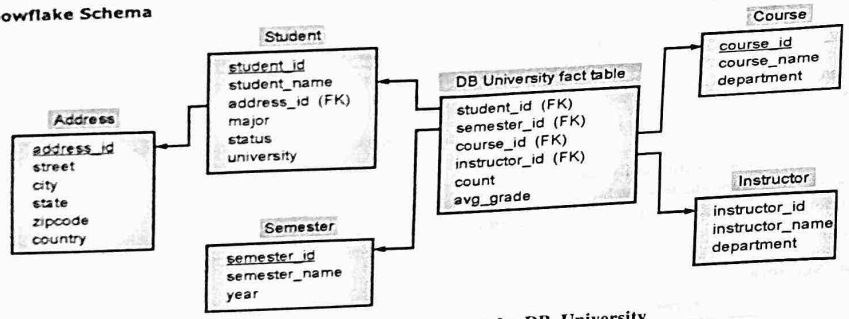
Ex. 1.6.5 : Suppose that a data warehouse for DB-University consists of the following four dimensions: student, course, semester, and instructor, and two measures count and avg_grade. When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg_grade measure stores the actual course grade of the student. At higher conceptual levels, avg_grade stores the average grade for the given combination.

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

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

☑ **Soln. :**

(a) Snowflake Schema



(1A15)**Fig. P. 1.6.5 : Snowflake Schema for DB_University**

(b) OLAP Operations to list the average grade of CS courses

i. Roll-up on course from course_id to department.

ii. Roll-up on student from student_id to university.

iii. Dice on course, student with department ="CS" and university = "DB_University".

iv. Drill-down on student from university to student_name.

Read!

1.7 UPDATE TO THE DIMENSION TABLES

- The number of rows in the fact table continues to increase over time. Rows in a fact table are rarely updated with changes. Even if there are adjustments to the prior numbers, these are processed as additional adjustment rows and added to the fact table.
- Dimension tables are more stable and less volatile. However, unlike a fact table, which changes as the number of rows increases, a dimension table changes as the attributes themselves change.

1.7.1 Slowly Changing Dimensions

- Slowly changing dimensions are the dimensions that change slowly over time, rather than changing on regular schedule, time-base. In data warehouse there is a need to track changes in dimension attributes in order to report historical data. In other words, implementing one of the slowly changing dimension types should enable users assigning proper dimension's attribute value for given date. Example of such dimensions could be: customer, geography, employee.
- There are many approaches how to deal with slowly changing dimension. The most popular are:
  - **Type 0** - The passive method
  - **Type 1** - Overwriting the old value
  - **Type 2** - Creating a new additional record
  - **Type 3** - Adding a new column
  - **Type 4** - Using historical table
  - **Type 6** - Combine approaches of types 1,2,3 (1+2+3=6)
- **Type 0 - The passive method**. In this method no special action is performed upon dimensional changes. Some dimension data can remain the same as it was first time inserted, others may be overwritten.
- **Type 1 - Overwriting the old value**. In this method no history of dimension changes is kept in the database. The old dimension value is simply overwritten be the new one. This type is easy to maintain and is often use for data in which changes are caused by processing corrections(e.g. removal of special characters, correcting spelling errors).

Before the change

Customer_ID	Customer_Name	Customer_Type
1	Cust_1	Corporate

After the change

Customer_ID	Customer_Name	Customer_Type
1	Cust_1	Retail

- **Type 2** - Creating a new additional record. In this methodology all history of dimension changes is kept in the database. You capture attribute change by adding a new row with a new surrogate key to the dimension table. Both the prior and new rows contain as attributes the natural key(or other durable identifier). Also 'effective date' and 'current indicator' columns are used in this method. There could be only one record with current indicator set to 'Y'. For 'effective date' columns, i.e. start_date and end_date, the end_date for current record usually is set to value 31-12-9999. Introducing changes to the dimensional model in type 2 could be very expensive database operation so it is not recommended to use it in dimensions where a new attribute could be added in the future.

Before the change

Customer_ID	Customer_Name	Customer_Type	Start_Date	End_Date	Current_Flag
1	Cust_1	Corporate	22-07-2010	31-12-9999	Y

After the change:

Customer_ID	Customer_Name	Customer_Type	Start_Date	End_Date	Current_Flag
1	Cust_1	Corporate	22-07-2010	17-05-2012	N
2	Cust_1	Retail	18-05-2012	31-12-9999	Y

- **Type 3 - Adding a new column**. In this type usually only the current and previous value of dimension is kept in the database. The new value is loaded into 'current/new' column and the old one into 'old/previous' column. Generally speaking, the history is limited to the number of column created for storing historical data. This is the least commonly needed technique.

Before the change

Customer_ID	Customer_Name	Current_Type	Previous_Type
1	Cust_1	Corporate	Corporate

After the change

Customer_ID	Customer_Name	Current_Type	Previous_Type
1	Cust_1	Retail	Corporate

- **Type 4 - Using historical table.** In this method a separate historical table is used to track all dimension's attribute historical changes for each of the dimension. The 'main' dimension table keeps only the current data e.g. customer and customer_history tables.

Current table

Customer_ID	Customer_Name	Customer_Type
1	Cust_1	Corporate

Historical table

Customer_ID	Customer_Name	Customer_Type	Start_Date	End_Date
1	Cust_1	Retail	01-01-2010	21-07-2010
1	Cust_1	Other	22-07-2010	17-05-2012
1	Cust_1	Corporate	18-05-2012	31-12-9999

- **Type 6 - Combine approaches of types 1,2,3 (1+2+3=6).** In this type we have in dimension table such additional columns as :
  - current_type - for keeping current value of the attribute. All history records for given item of attribute have the same current value.
  - historical_type - for keeping historical value of the attribute. All history records for given item of attribute could have different values.
  - start_date - for keeping start date of 'effective date' of attribute's history.
  - end_date - for keeping end date of 'effective date' of attribute's history.
  - current_flag - for keeping information about the most recent record.
- In this method to capture attribute change we add a new record as in type 2. The current_type information is overwritten with the new one as in type 1. We store the history in a historical_column as in type 3.

Customer_ID	Customer_Name	Current_Type	Historical_Type	Start_Date	End_Date	Current_Flag
1	Cust_1	Corporate	Retail	01-01-2010	21-07-2010	N
2	Cust_1	Corporate	Other	22-07-2010	17-05-2012	N
3	Cust_1	Corporate	Corporate	18-05-2012	31-12-9999	Y

1.7.2 Rapidly Changing Dimension (RCD)

- A dimension is a fast changing or rapidly changing dimension if one or more of its attributes in the table changes very fast and in many rows. Handling rapidly changing dimension in data warehouse is very difficult because of many performance implications.
- As you know slowly changing dimension type 2 is used to preserve the history for the changes. But the problem with type 2 is, with each and every change in the dimension attribute, it adds new row to the table. If in case there are dimensions that are changing a lot, table become larger and may cause serious performance issues. Hence, use of the type 2 may not be the wise decision to implement the rapidly changing dimensions.
- **For example :** Consider patient dimension where there are 1000 rows in it. On average basis, each patient changes the 10 of attributes in a year. If you use the type 2 to manage this scenario, there will be 1000 * 10 = 10000 rows. Imagine if the table has 1 million rows, it will become very hard to handle the situation with type 2. Hence we use rapidly changing dimension approach.

1.7.3 Conformed Dimension

- A conformed dimension is the dimension that is shared across multiple data mart or subject area. Company may use the same dimension table across different projects without making any changes to the dimension tables.
- Conformed dimension example would be Customer dimension, i.e. both marketing and sales department can use Customer dimension for their reporting purpose.

1.7.4 Junk Dimension

- A junk dimension is a grouping of typically low cardinality attributes, so you can remove them from main dimension.
- You can use Junk dimensions to implement the rapidly changing dimension where you can use it to stores the attribute that changes rapidly. For example, attributes such as flags, weights, BMI (body mass index), etc.

1.7.5 Degenerated Dimension

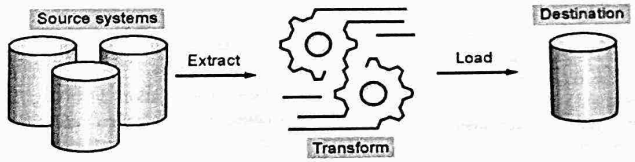
- A degenerated dimension is a dimension that is derived from fact table and does not have its own dimension table. For example, receipt number does not have dimension table associated with it. Such details are just for information purpose.

1.7.6 Role Playing Dimension

Dimensions which are often used for multiple purposes within the same database are called role-playing dimensions. For example, you can use a date dimension for "date of sale", as well as "date of delivery", or "date of hire".

1.8 MAJOR STEPS IN ETL PROCESS

- The process of extracting data from source systems and bringing it into the data warehouse is commonly called **ETL**, which stands for extraction, transformation, and loading. Note that ETL refers to a broad process, and not three well-defined steps.
- 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.
- The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives and is technically challenging.
- To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes.
- ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.
- **ETL Tools :** Most commonly used ETL tools are Sybase, Oracle Warehouse builder, Clover ETL and MarkLogic.



(1A16)**Fig. 1.8.1 : ETL Process**

Let us understand each step of the ETL process in depth:

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, NoSQL, XML and flat files into the staging area.
- It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also.
- Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process.

☞ Data Extraction Techniques

There are two types of data warehouse extraction methods: **Logical and Physical extraction methods.**

(A) Logical Extraction

Logical Extraction method in-turn has two methods:

(I) Full Extraction

- In this method, data is completely extracted from the source system. The source data will be provided as-is and no additional logical information is necessary on the source system. Since it is complete extraction, so no need to track source system for changes.
- For example, exporting complete table in the form of flat file.

(II) Incremental Extraction

- In incremental extraction, the changes in source data need to be tracked since the last successful extraction. Only these changes in data will be extracted and then loaded. Identifying the last changed data itself is the complex process and involve many logics.
- You can detect the changes in the source system from the specific column in the source system that has the last changed timestamp. You can also create a change table in the source system, which keeps track of the changes in the source data.

(B) Physical Extraction

Physical extraction has two methods: Online and Offline extraction.

(i) Online Extraction

In this process, extraction process directly connects to the source system and extract the source data.

(ii) Offline Extraction

- The data is not extracted directly from the source system but is staged explicitly outside the original source system.
- You can consider the following common structure in offline extraction:
  1. **Flat file :** Generic format
  2. **Dump file :** Database specific file

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 :
  1. **Filtering –** loading only certain attributes into the data warehouse.
  2. **Cleaning –** filling up the NULL values with some default values, mapping U.S.A, United States and America into USA, etc.
  3. **Joining –** joining multiple attributes into one.
  4. **Splitting –** splitting a single attribute into multiple attributes.
  5. **Sorting –** sorting tuples on the basis of some attribute (generally key-attribute).

☞ Data Transformation Techniques

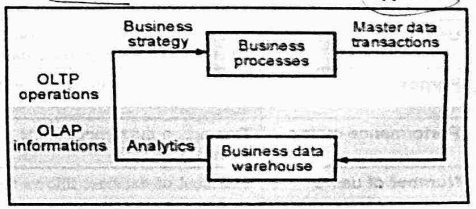
- **Data Smoothing :** This method is used for removing the noise from a dataset. Noise is referred to as the distorted and meaningless data within a dataset. Smoothing uses algorithms to highlight the special features in the data. After removing noise, the process can detect any small changes to the data to detect special patterns.
- **Data Aggregation :** Aggregation is the process of collecting data from a variety of sources and storing it in a single format. Here, data is collected, stored, analyzed and presented in a report or summary format. It helps in gathering more information about a particular data cluster. The method helps in collecting vast amounts of data.
- **Discretization :** This is a process of converting continuous data into a set of data intervals. Continuous attribute values are substituted by small interval labels. This makes the data easier to study and analyze.
- **Generalization :** In this process, low-level data attributes are transformed into high-level data attributes using concept hierarchies. For example, age data can be in the form of (20, 30) in a dataset. It is transformed into a higher conceptual level into a categorical value (young, old).
- **Attribute construction :** In the attribute construction method, new attributes are created from an existing set of attributes. For example, in a dataset of employee information, the attributes can be employee name, employee ID and address. These attributes can be used to construct another dataset that contains information about the employees who have joined in the year 2019 only. This method of reconstruction makes mining more efficient and helps in creating new datasets quickly.
- **Normalization :** Also called data pre-processing, this is one of the crucial techniques for **data transformation in data mining.** Here, the data is transformed so that it falls under a given range. When attributes are on different ranges or scales, data modelling and mining can be difficult. Normalization helps in applying data mining algorithms and extracting data faster.

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.
- Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals.
- The rate and period of loading solely depends on the requirements and varies from system to system.
- Loading can be carried in two ways:
  - **(A) Refresh :** Data Warehouse data is completely rewritten. This means that older file is replaced. Refresh is usually used in combination with static extraction to populate a data warehouse initially.
  - **(B) Update :** Only those changes applied to source information are added to the Data Warehouse. An update is typically carried out without deleting or modifying pre-existing data. This method is used in combination with incremental extraction to update data warehouses regularly.

1.9 OLTP VS OLAP

- We can divide IT systems as transactional (OLTP) and analytical (OLAP). In general, we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.
- **OLTP (On-line Transaction Processing)** is characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3NF).
- **OLAP (On-line Analytical Processing)** is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema). For example, a bank storing years of historical records of check deposits could use an OLAP database to provide reporting to business users. OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy. At the core of the OLAP concept, is an OLAP Cube. The OLAP cube is a data structure optimized for very quick data analysis. The OLAP Cube consists of numeric facts called measures which are categorized by dimensions. OLAP Cube is also called the **hypercube**.



(IA17)**Fig. 1.9.1 : OLTP Vs OLAP Operations**

- The following table summarizes the major differences between OLTP and OLAP system design.

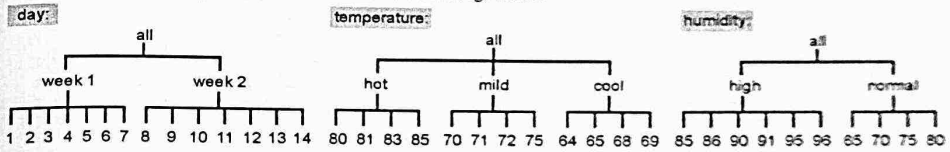
Table 1.9.1 : OLTP Vs OLAP

Parameters	OLTP	OLAP
Process	It is an online transactional system. It manages database modification.	OLAP is an online analysis and data retrieving process.
Characteristic	It is characterized by large numbers of short online transactions.	It is characterized by a large volume of data.
Functionality	OLTP is an online database modifying system.	OLAP is an online database query management system.
Method	OLTP uses traditional DBMS.	OLAP uses the data warehouse.
Query	Insert, Update, and Delete information from the database.	Mostly Select operations
Table	Tables in OLTP database are normalized.	Tables in OLAP database are not normalized.
Source	OLTP and its transactions are the sources of data.	Different OLTP databases become the source of data for OLAP.
Storage	The size of the data is relatively small as the historical data is archived. For e.g. MB, GB.	Large amount of data is stored typically in TB, PB.
Data Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not an issue.
Response time	It's response time is in millisecond.	Response time in seconds to minutes.
Data quality	The data in the OLTP database is always detailed and organized.	The data in OLAP process might not be organized.
Usefulness	It helps to control and run fundamental business tasks.	It helps with planning, problem-solving, and decision support.
Operation	Allow read/write operations.	Only read and rarely write.
Audience	It is a market orientated process.	It is a customer orientated process.
Query Type	Queries in this process are standardized and simple.	Complex queries involving aggregations.
Back-up	Complete backup of the data combined with incremental backups.	OLAP only need a backup from time to time. Backup is not important compared to OLTP
Design	DB design is application oriented. Example: Database design changes with industry like Retail, Airline, Banking, etc.	DB design is subject oriented. Example: Database design changes with subjects like sales, marketing, purchasing, etc.
User type	It is used by Data critical users like clerk, DBA & Data Base professionals.	Used by Data knowledge users like workers, managers, and CEO.
Purpose	Designed for real time business operations.	Designed for analysis of business measures by category and attributes.
Performance metric	Transaction throughput is the performance metric.	Query throughput is the performance metric.
Number of users	This kind of database allows thousands of users.	This kind of database allows only hundreds of users.
Productivity	It helps to Increase user's self-service and productivity.	Help to Increase productivity of the business analysts.

Parameters	OLTP	OLAP
Challenge	Data Warehouses historically have been a development project which may prove costly to build.	An OLAP cube is not an open SQL server data warehouse. Therefore, technical knowledge and experience is essential to manage the OLAP server.
Process	It provides fast result for daily used data.	It ensures that response to the query is quicker consistently.
Characteristic	It is easy to create and maintain.	It lets the user create a view with the help of a spreadsheet.
Style	OLTP is designed to have fast response time, low data redundancy and is normalized.	A data warehouse is created uniquely so that it can integrate different data sources for building a consolidated database

1.10 OLAP OPERATIONS

- In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives.
- For example, we have attributes as day, temperature and humidity, we can group values in subsets and name these subsets, thus obtaining a set of hierarchies as shown in Fig. 1.10.1.



(1A18)**Fig. 1.10.1**

- OLAP provides a user-friendly environment for interactive data analysis. A number of OLAP data cube operations exist to materialize different views of data, allowing interactive querying and analysis of the data.
- The most popular end user operations on dimensional data are:

1. Roll-Up

- The roll-up operation (also called drill-up or aggregation operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by climbing down a concept hierarchy, i.e. dimension reduction. Let us explain roll up with an example:
- Consider the following cubes illustrating temperature of certain days recorded weekly:

Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

- Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes.
- To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a **roll-up**.
- By doing this, we get the following cube :

Temperature	Cool	Mild	Hot
Week1	2	1	1
Week2	2	1	1

- The concept hierarchy can be defined as hot→day→week. The roll-up operation groups the data by levels of temperature.

2. Drill-Down

- The drill-down operation (also called roll-down) is the reverse operation of roll-up. Drill-down is like zooming-in on the data cube. It navigates from less

detailed record to more detailed data. Drill-down can be performed by either stepping down a concept hierarchy for a dimension or adding additional dimensions.

- Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.
- Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group. Drill-down adds more details to the given data.

Temperature	Cool	mild	Hot
Day 1	0	0	0
Day 2	0	0	0
Day 3	0	0	1
Day 4	0	1	0
Day 5	1	0	0
Day 6	0	0	0
Day 7	1	0	0
Day 8	0	0	0
Day 9	1	0	0
Day 10	0	1	0
Day 11	0	1	0
Day 12	0	1	0
Day 13	0	0	1
Day 14	0	0	0

3. Slice

- A slice is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the slice operations perform a selection on one dimension of the given cube, thus resulting in a sub-cube. It will form a new sub-cubes by selecting one or more dimensions.
- For example, if we make the selection, temperature = cool we will obtain the following cube:

Temperature	Cool
Day 1	0
Day 2	0
Day 3	0
Day 4	0
Day 5	1
Day 6	1
Day 7	1
Day 8	1
Day 9	1
Day 11	0
Day 12	0
Day 13	0
Day 14	0

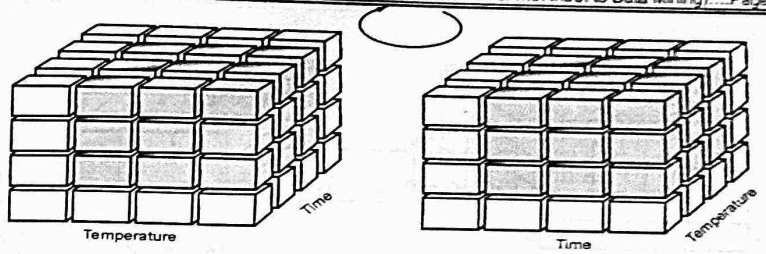
4. Dice

- The dice operation describes a sub-cube by operating a selection on two or more dimension.
- **For example**, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following sub-cube (still two-dimensional)

Temperature	Cool	hot
Day 3	0	1
Day 4	0	0

5. Pivot

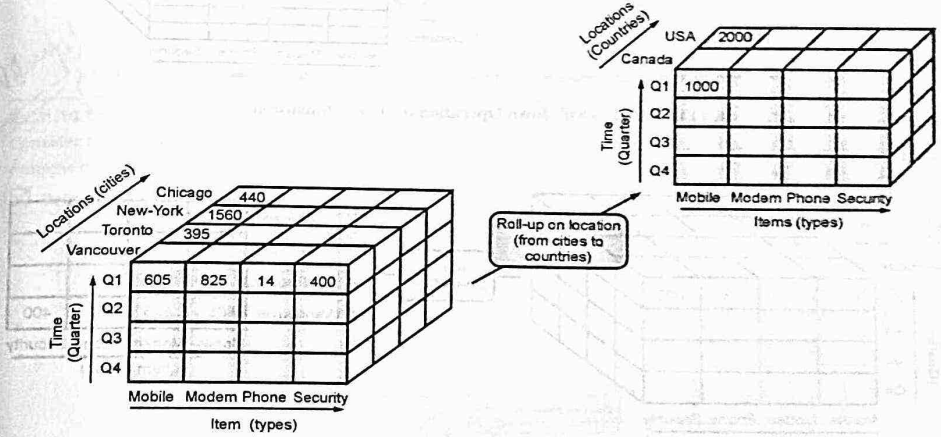
- The pivot operation is also called a rotation. Pivot is a visualization operation which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.



(1A19)**Fig. 1.10.2 : Pivot Operation on Multidimensional Cube**

- **Example :** Let's look at some typical OLAP operations for multidimensional data. Each of the following operations described is illustrated below. In the figure is a data cube for Digi1 Electronics sales. The cube contains the dimension location, time, and item, where location is aggregated with respect to city values, time is aggregated with respect to quarters, and item is aggregated with respect to item types. The measure displayed is dollars sold (in thousands). (For improved readability, only some of the cubes' cell values are shown.) The data examined are for the cities Chicago, NewYork, Toronto, and Vancouver.

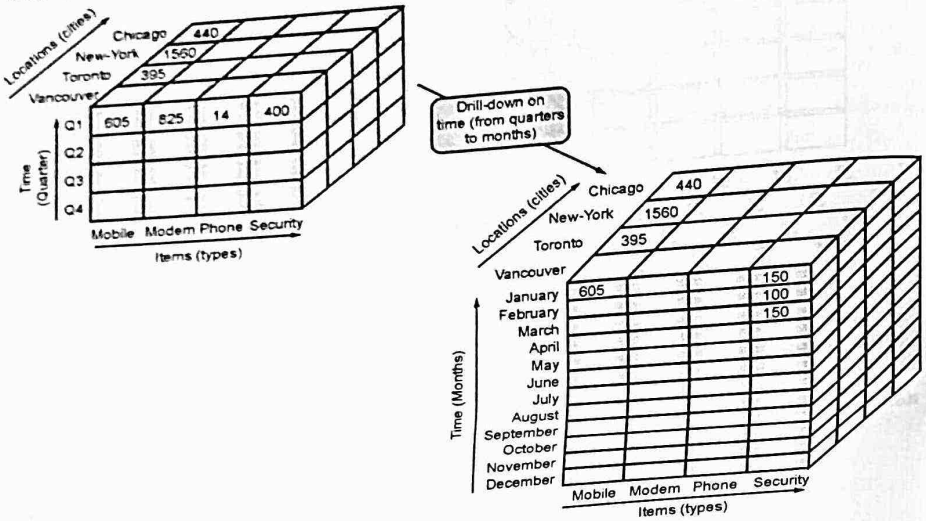
1. Roll-up Operation



(1A20) **Fig. 1.10.3 : Roll-up Operation on Location Dimension**

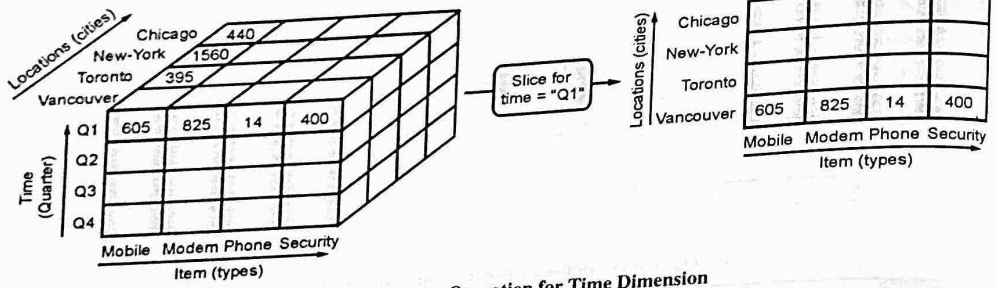
NOTES

2. Drill-down Operation



(1A21) **Fig. 1.10.4 : Drill-down Operation on Time Dimension**

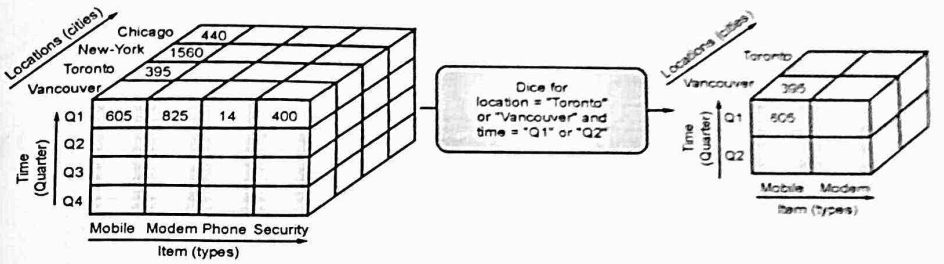
3. Slice Operation



(1A22) **Fig. 1.10.5 : Slice Operation for Time Dimension**

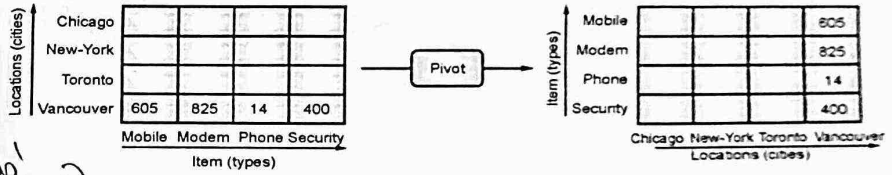
NOTES

4. Dice Operation



(1A23) **Fig. 1.10.6 : Dice Operation for Location and Time Dimensions**

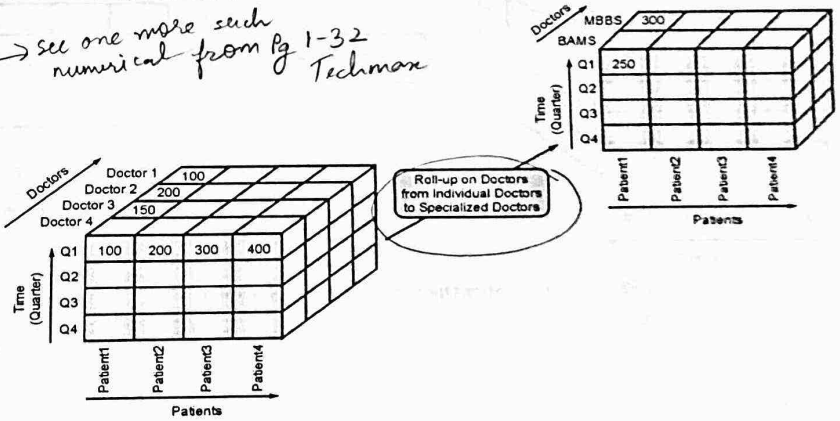
5. Pivot Operation



(1A24) **Fig. 1.10.7 : Pivot Operation for Location and Item Dimensions**

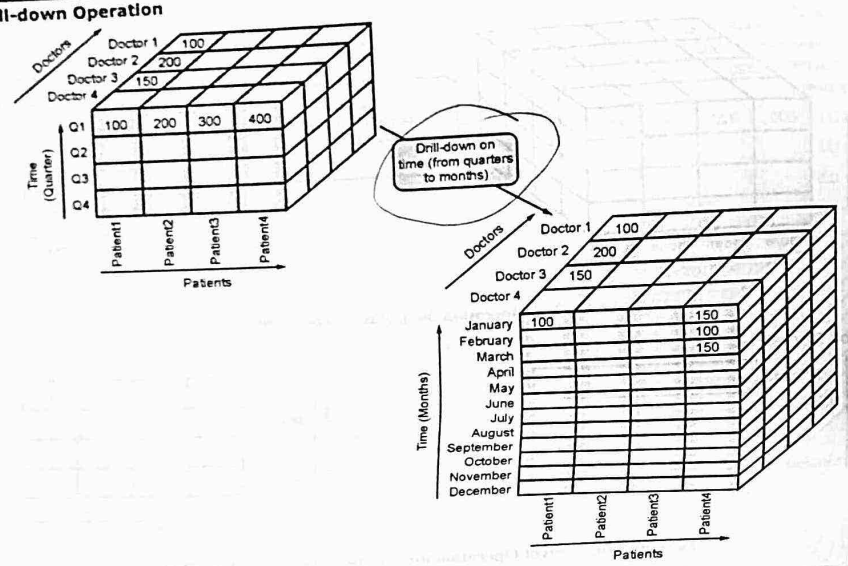
Ex. 1.10.1: Consider a data warehouse for a hospital where there are three dimensions: a) Doctor b) Patient c) Time. Consider two measuresi) Countii) Charge where charge is the fee that the doctor charges a patient for a visit. For the above example create a cube and illustrate the following OLAP operations.1) Rollup 2) Drill down 3) Slice 4) Dice 5) Pivot.

Soln. : 1. Roll-up Operation



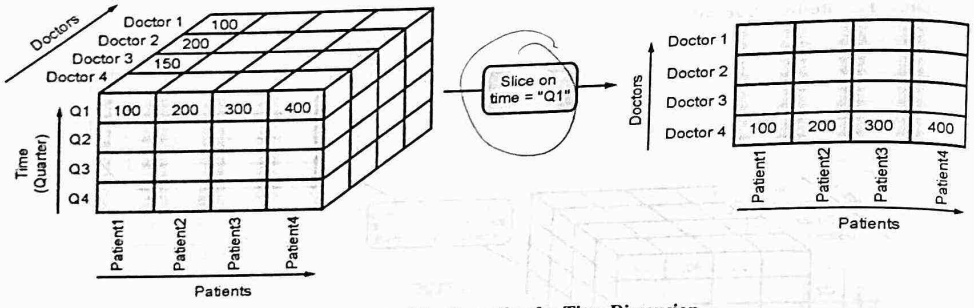
(1A25) **Fig. P. 1.10.1(a) : Roll-up Operation on Doctors Dimension**

2. Drill-down Operation



(1A26) **Fig. 1.10.1(b) : Drill-down Operation on Time Dimension**

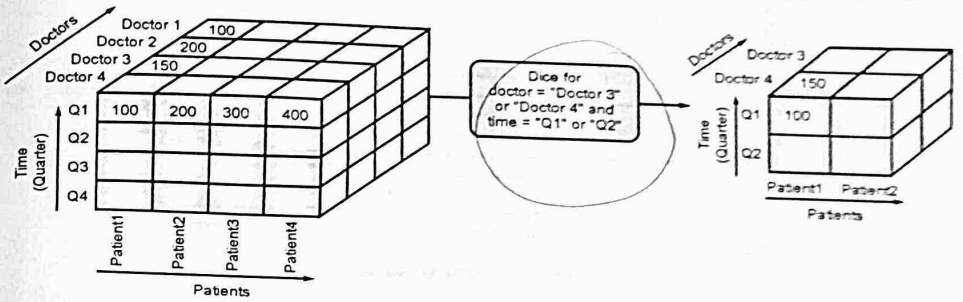
3. Slice Operation



(1A27) **Fig. 1.10.1(c) : Slice Operation for Time Dimension**

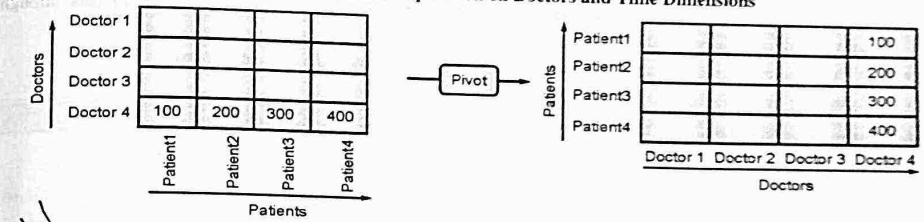
Module 1

4. Dice Operation



(1A28) **Fig. 1.10.1(d) : Dice Operation on Doctors and Time Dimensions**

5. Pivot



(1A29) **Fig. 1.10.1(e) : Pivot Operation on Doctors and Patients Dimensions**

1.11 OLAP SERVERS

There are three types of OLAP servers, namely, Relational OLAP (ROLAP), Multidimensional OLAP (MOLAP) and Hybrid OLAP (HOLAP).

1. Relational OLAP (ROLAP)

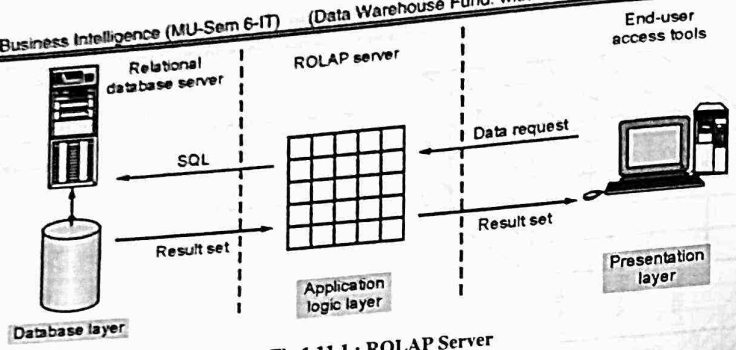
- Relational On-Line Analytical Processing (ROLAP) work mainly for the data that resides in a relational database, where the base data and dimension tables are stored as relational tables.
- ROLAP servers are placed between the relational back-end server and client front-end tools.
- ROLAP servers use RDBMS to store and manage warehouse data, and OLAP middleware to support missing pieces.
- **Example :** DSS Server of Mierostrategy

☞ **Advantages of ROLAP**

1. ROLAP can handle large amounts of data.
2. Can be used with data warehouse and OLTP systems.

☞ **Disadvantages of ROLAP**

1. Limited by SQL functionalities.
2. Hard to maintain aggregate tables.



(1A30)Fig.1.11.1 : ROLAP Server

2. Multidimensional OLAP (MOLAP)

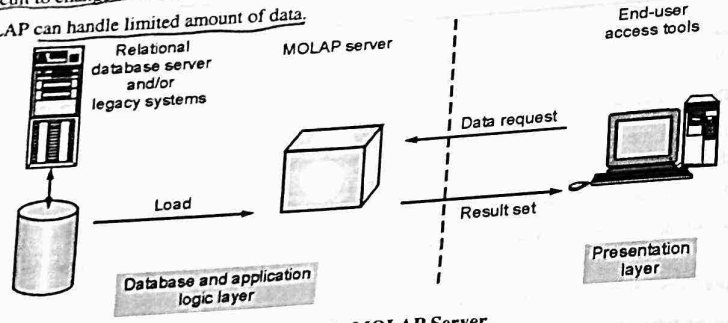
- Multidimensional On-Line Analytical Processing (MOLAP) support multidimensional views of data through array-based multidimensional storage engines.
- With multidimensional data stores, the storage utilization may be low if the data set is sparse.
- Example : Oracle Essbase

☞ **Advantages of MOLAP**

1. Optimal for slice and dice operations.
2. Performs better than ROLAP when data is dense.
3. Can perform complex calculations.

☞ **Disadvantages of MOLAP**

1. Difficult to change dimension without re-aggregation.
2. MOLAP can handle limited amount of data.



(1A31)Fig.1.11.2 : MOLAP Server

3. Hybrid OLAP (HOLAP)

- Hybrid On-Line Analytical Processing (HOLAP) is a combination of ROLAP and MOLAP.
- HOLAP provide greater scalability of ROLAP and the faster computation of MOLAP.
- Example: Microsoft SQL Server 2000

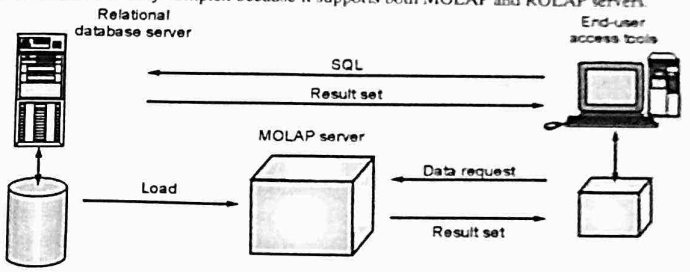
Module 1

☞ **Advantages of HOLAP**

1. HOLAP provide advantages of both MOLAP and ROLAP.
2. Provide fast access at all levels of aggregation.

☞ **Disadvantage of HOLAP**

1. HOLAP architecture is very complex because it supports both MOLAP and ROLAP servers.



(1A32)Fig. 1.11.3 : HOLAP Server

1.12 APPLICATIONS OF OLAP

OLAP system is to analyze the business which helps in decision-making, forecasting, planning, problem solving. Some of the applications of OLAP include:

1. Financial Applications

(i) Resource (man-power, raw material) allocation

(ii) Budgeting.

2. Sales Applications

(i) Research on market analysis

(ii) Forecasting sales

(iii) Analyzing sales promotions

(iv) Analyzing customer requirements

(v) Dividing market based on customer.

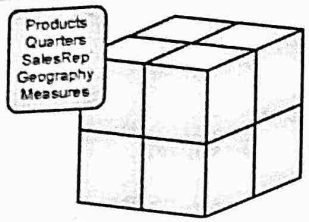
3. Business Modelling

(i) Understanding and simulating the market trend and business behavior

(ii) Decision support system for managers, executives, CEO, data scientists.

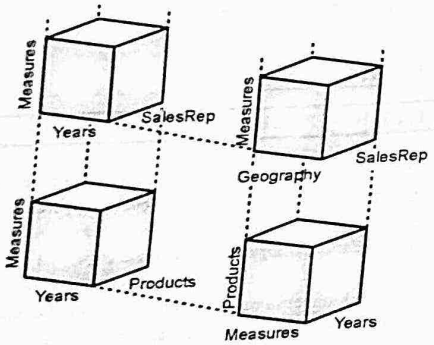
1.13 HYPERCUBE

- Multidimensional databases can present their data for an application using two types of cubes: hypercube and multi-cubes. In a hypercube, as shown in Fig. 1.13.1, all data appears logically as a single cube. All parts of the manifold represented by this hypercube have identical dimensionality. Each dimension belongs to one cube only. A dimension is owned by the hypercube. This simplicity makes easy for users to understand.
- Designing a hypercube model is a top-down process with three major steps.
  1. You decide which process of the business you want to capture in the model, such as sales activity.
  2. You identify the values that you want to capture, such as sales amounts. This information is always numeric.
  3. You identify the granularity of the data, meaning the lowest level of detail at which you want to capture. These elements are the dimensions. And time, geography, product, and customer are some common dimensions. For example, a single cell in a cube could refer to the sales amount of Sony TVs in the first quarter of the year, in PA, USA.



(1A33)**Fig. 1.13.1 : Hypercube**

- In the multi-cube model, data is segmented into a set of smaller cubes, each of which is composed of a subset of the available dimensions, as shown in Fig. 1.13.2. They are used to handle multiple fact tables, each with different dimensionality.



(1A34)**Fig. 1.13.2 : Multi-cube**

- *A dimension can be part of multiple cubes. Dimensions are not owned by any one cube, like under the hypercube model. Rather, they are available to all cubes, or there can be some dimensions that do not belong to any cube. This makes it much more efficient and versatile. It is also a more efficient way of storing very sparse data, and it can reduce the pre-calculation database explosion effect, which will be covered in a later section.*
- *The drawback is that this is less straightforward than hypercube and can carry steeper learning curves. Some systems use the combined approach of hypercube and multi-cubes, by separating the storage, processing, and presentation layers. It stores data as multi-cubes but presents as a hypercube.*

1.14 AGGREGATE FACT TABLES

- Aggregate fact tables are special fact tables in a data warehouse that contain new metrics derived from one or more aggregate functions (AVERAGE, COUNT, MIN, MAX, etc.) or from other specialized functions that output totals derived from a grouping of the base data.
- These new metrics, called "aggregate facts" or "summary statistics" are stored and maintained in the data warehouse database in special fact tables at the grain of the aggregation.
- Likewise, the corresponding dimensions are rolled up, and condensed to match the new grain of the fact.
- These specialized tables are used as substitutes whenever possible for returning user queries. The reason is Speed. Querying a tidy aggregate table is much faster and uses much less disk I/O than the base, atomic fact table, especially if the dimensions are large as well.
- If you want to wow your users, start adding aggregates. You can even use this "trick" in your operational systems to serve as a foundation for operational reports.
- For example, take the "Orders" business process from an online catalog company where you might have customer orders in a fact table called FactOrders with dimensions Customer, Product, and OrderDate.
- With possibly millions of orders in the transaction fact, it makes sense to start thinking about aggregates.
- To further the above example, assume that the business is interested in a report: "Monthly orders by state and product type".
- While you could generate this easily enough using the FactOrders fact table, you could likely speed up the data retrieval for the report by at least half (but likely much, much more) using an aggregate.

1.15 INTRODUCTION TO DATA MINING

- There is a huge amount of data available in the Information Industry. This data is of no use until it is converted into useful information. It is necessary to analyze this huge amount of data and extract useful information from it.

- William J Frawley, Gregory Piatetsky-Shapiro and Christopher J Matheus define **data mining as "The non-trivial extraction of implicit, previously unknown, and potentially useful information from data."**
- The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.
- The computer is responsible for finding the patterns by identifying the underlying rules and features in the data.
- In other words, we can say that Data Mining is the process of investigating hidden patterns of information to various perspectives for categorization into useful data, which is collected and assembled in particular areas such as data warehouses, efficient analysis, data mining algorithm, helping decision making and other data requirement to eventually cost-cutting and generating revenue.

1.15.1 Sources of Data that can be Mined

The data from multiple sources are integrated into a common source known as Data Warehouse. Let's discuss what type of data can be mined:

1. Flat Files	2. Relational Databases
3. Data Warehouse	4. Transactional Databases
5. Multimedia Databases	6. Spatial Database
7. Time-series Databases	8. WWW

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. E.g.: CSV file.
- **Application :** Used in Data Warehousing 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 data warehouse is defined as the collection of data integrated from multiple sources that will queries and decision making.
- There are three types of data warehouse: Enterprise data warehouse, Data Mart and Virtual Warehouse.
- Two approaches can be used to update data in Data Warehouse: Query-driven Approach and Update-driven Approach.
- **Application**: Business decision making, Data mining, etc.

4. Transactional Databases

- Transactional database is 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 (Atomicity, Consistency, Isolation and Durability) 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 formats.
- **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.
- Handles array of numbers indexed by time, date, etc.
- 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 network.
- 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.

1.15.2 Kind of Patterns to be mined (Data Mining Techniques)

Data mining is highly effective, so long as it draws upon one or more of these techniques :

1. Tracking patterns	2. Classification
3. Association	4. Outlier detection
5. Clustering	6. Regression
7. Prediction	

1. Tracking patterns : One of the most basic techniques in data mining is learning to recognize patterns in your data sets. This is usually a recognition of some aberration in your data happening at regular intervals, or an ebb and flow of a certain variable over time. For example, you might see that your sales of a certain product seem to spike just before the holidays, or notice that warmer weather drives more people to your website.

2. Classification : Classification is a more complex data mining technique that forces you to collect various attributes together into discernible categories, which you can then use to draw further conclusions, or serve some function. For example, if you're evaluating data on individual customers' financial backgrounds and purchase histories, you might be able to classify them as "low," "medium," or "high" credit risks. You could then use these classifications to learn even more about those customers.

3. Association : Association is related to tracking patterns, but is more specific to dependently linked variables. In this case, you'll look for specific events or attributes that are highly correlated with another event or attribute; for example, you might notice that when your customers buy a specific item, they also often buy a second, related item. This is usually what's used to populate "people also bought" sections of online stores.

4. Outlier detection : In many cases, simply recognizing the overarching pattern can't give you a clear understanding of your data set. You also need to be able to identify anomalies, or outliers in your data. For example, if your purchasers are almost exclusively male, but during one strange week in July, there's a huge spike in female purchasers, you'll want to investigate the spike and see what drove it, so you can either replicate it or better understand your audience in the process.

5. Clustering : Clustering is very similar to classification, but involves grouping chunks of data together based on their similarities. For example, you might choose to cluster different demographics of your audience into different packets based on how much disposable income they have, or how often they tend to shop at your store.

6. Regression : Regression, used primarily as a form of planning and modeling, is used to identify the likelihood of a certain variable, given the presence of other variables. For example, you could use it to project a certain price, based on other factors like availability, consumer demand, and competition. More specifically, regression's main focus is to help you uncover the exact relationship between two (or more) variables in a given data set.

7. Prediction : Prediction is one of the most valuable data mining techniques, since it's used to project the types of data you'll see in the future. In many cases, just recognizing and understanding historical trends is enough to chart a somewhat accurate prediction of what will happen in the future. For example, you might review consumers' credit histories and past purchases to predict whether they will be a credit risk in the future.

1.15.3 Data Mining Technologies

Several techniques are used in the development of data mining methods. Some of them are mentioned below :

1. Statistics
2. Machine learning
3. Information retrieval
4. Database systems and data warehouse
5. Decision support system

1. Statistics

- It uses the mathematical analysis to express representations, model and summarize empirical data or real world observations.
- Statistical analysis involves the collection of methods, applicable to large amount of data to conclude and report the trend.

2. Machine learning

- Arthur Samuel defined machine learning as "a field of study that gives computers the ability to learn without being programmed".
- When the new data is entered in the computer, algorithms help the data to grow or change due to machine learning.
- In machine learning, an algorithm is constructed to predict the data from the available database (Predictive analysis).
- It is related to computational statistics.
- The four types of machine learning are:
  - (i) **Supervised learning :** It is based on the classification. It is also called as inductive learning. In this method, the desired outputs are included in the training dataset.
  - (ii) **Unsupervised learning :** Unsupervised learning is based on clustering. Clusters are formed on the basis of similarity measures and desired outputs are not included in the training dataset.
  - (iii) Semi-supervised learning: Semi-supervised learning includes some desired outputs to the training dataset to generate the appropriate functions. This method generally avoids the large number of labeled examples (i.e. desired outputs).
  - (iv) **Semi-supervised learning :** Active learning is a powerful approach in analyzing the data efficiently. The algorithm is designed in such a way that; the desired output should be decided by the algorithm itself (the user plays important role in this type).

3. Information retrieval

Information deals with uncertain representations of the semantics of objects (text, images). For example: Finding relevant information from a large document.

4. Database systems and data warehouse

- Databases are used for the purpose of recording the data as well as data warehousing.
- Online Transactional Processing (OLTP) uses databases for day to day transaction purpose.
- To remove the redundant data and save the storage space, data is normalized and stored in the form of tables.
- Entity-Relational modeling techniques are used for relational database management system design.
- Data warehouses are used to store historical data which helps to take strategical decision for business.
- It is used for online analytical processing (OLAP), which helps to analyze the data.

5. Decision support system

- Decision support system is a category of information system. It is very useful in decision making for organizations.
- It is an interactive software based system which helps decision makers to extract useful information from the data, documents to make the decision.

1.15.4 Difference between Data Mining and Data Warehouse

Table 1.15.1 : Data Mining Vs Data Warehouse

Sr. No.	Data Mining	Data Warehouse
1.	Data mining is the process of analyzing unknown patterns of data.	A data warehouse is database system which is designed for analytical instead of transactional work.
2.	Data mining is a method of comparing large amounts of data to finding right patterns.	Data warehousing is a method of centralizing data from different sources into one common repository.
3.	Data mining is usually done by business users with the assistance of engineers.	Data warehousing is a process which needs to occur before any data mining can take place.
4.	Data mining is the considered as a process of extracting data from large data sets.	On the other hand, Data warehousing is the process of pooling all relevant data together.
5.	One of the most important benefits of data mining techniques is the detection and identification of errors in the system.	One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features.
6.	Data mining helps to create suggestive patterns of important factors. Like the buying habits of customers, products, sales. So that, companies can make the necessary adjustments in operation and production.	Data Warehouse adds an extra value to operational business systems like CRM systems when the warehouse is integrated.
7.	The Data mining techniques are never 100% accurate and may cause serious consequences in certain conditions.	In the data warehouse, there is great chance that the data which was required for analysis by the organization may not be integrated into the warehouse. It can easily lead to loss of information.
8.	The information gathered based on Data Mining by organizations can be misused against a group of people.	Data warehouses are created for a huge IT project. Therefore, it involves high maintenance system which can impact the revenue of medium to small-scale organizations.
9.	After successful initial queries, users may ask more complicated queries which would increase the workload.	Data Warehouse is complicated to implement and maintain.
10	Organisations can benefit from this analytical tool by equipping pertinent and usable knowledge-based information.	Data warehouse stores a large amount of historical data which helps users to analyze different time periods and trends for making future predictions.
11.	Organisations need to spend lots of their resources for training and Implementation purpose. Moreover, data mining tools work in different manners due to different algorithms employed in their design.	In Data warehouse, data is pooled from multiple sources. The data needs to be cleaned and transformed. This could be a challenge.
12.	The data mining methods are cost-effective and efficient compares to other statistical data applications.	Data warehouse's responsibility is to simplify every type of business data. Most of the work that will be done on user's part is inputting the raw data.
13.	Another critical benefit of data mining techniques is the identification of errors which can lead to losses. Generated data could be used to detect a drop-in sale.	Data warehouse allows users to access critical data from the number of sources in a single place. Therefore, it saves user's time of retrieving data from multiple sources.
14.	Data mining helps to generate actionable strategies built on data insights.	Once you input any information into Data warehouse system, you will unlikely to lose track of this data again. You need to conduct a quick search, helps you to find the right statistic information.

1.16 DATA MINING TASK PRIMITIVES

- Each user will have a data mining task in mind, that is, some form of data analysis that he or she would like to have performed.
- A data mining task can be specified in the form of a data mining query, which is input to the data mining system.
- A data mining query is defined in terms of data mining task primitives.
- These primitives allow the user to interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths.
- Here is the list of Data Mining Task Primitives.

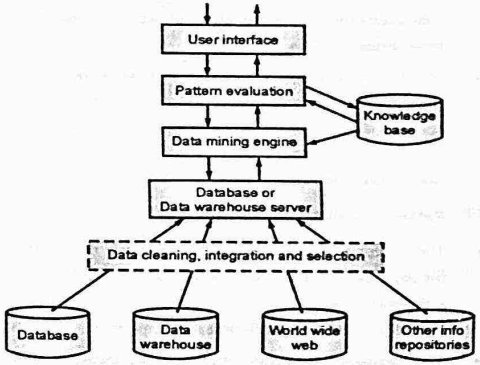
1. **The set of task-relevant data to be mined :** This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions).
2. **The kind of knowledge to be mined :** This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.
3. **The background knowledge to be used in the discovery process :** This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction.
4. **The interestingness measures and thresholds for pattern evaluation :** They may be used to guide the mining process or, after discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.
5. **The expected representation for visualizing the discovered patterns :** This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

1.17 DATA MINING ARCHITECTURE

The significant components of data mining systems are a data source, data mining engine, data warehouse server, the pattern evaluation module, graphical user interface, and knowledge base.

Data Source

- The actual source of data is the Database, data warehouse, World Wide Web (WWW), text files, and other documents. You need a huge amount of historical data for data mining to be successful.
- Organizations typically store data in databases or data warehouses. Data warehouses may comprise one or more databases, text files spreadsheets, or other repositories of data.
- Sometimes, even plain text files or spreadsheets may contain information. Another primary source of data is the World Wide Web or the internet.



(1B1)**Fig. 1.17.1: Data Mining Architecture**

☞ Data Cleaning, Integration and Selection

- Before passing the data to the database or data warehouse server, the data must be cleaned, integrated and selected.
- As the information comes from various sources and in different formats, it can't be used directly for the data mining procedure because the data may not be complete and accurate. So, the first data requires to be cleaned and unified.
- More information than needed will be collected from various data sources, and only the data of interest will have to be selected and passed to the server. These procedures are not as easy as we think. Several methods may be performed on the data as part of selection, integration, and cleaning.

☞ Database or Data Warehouse Server

The database or data warehouse server consists of the original data that is ready to be processed. Hence, the server is cause for retrieving the relevant data that is based on data mining as per user request.

☞ Data Mining Engine

- The data mining engine is a major component of any data mining system. It contains several modules for operating data mining tasks, including association, characterization, classification, clustering, prediction, time-series analysis, etc.
- In other words, we can say data mining is the root of our data mining architecture. It comprises instruments and software used to obtain insights and knowledge from data collected from various data sources and stored within the data warehouse.

☞ Pattern Evaluation Module

- The Pattern evaluation module is primarily responsible for the measure of investigation of the pattern by using a threshold value. It collaborates with the data mining engine to focus the search on exciting patterns.
- This segment commonly employs stake measures that cooperate with the data mining modules to focus the search towards fascinating patterns. It might utilize a stake threshold to filter out discovered patterns.
- On the other hand, the pattern evaluation module might be coordinated with the mining module, depending on the implementation of the data mining techniques used. For efficient data mining, it is abnormally suggested to push the evaluation of pattern stake as much as possible into the mining procedure to confine the search to only fascinating patterns.

☞ Graphical User Interface

- The graphical user interface (GUI) module communicates between the data mining system and the user. This module helps the user to easily and efficiently use the system without knowing the complexity of the process.
- This module cooperates with the data mining system when the user specifies a query or a task and displays the results.

☞ Knowledge Base

- The knowledge base is helpful in the entire process of data mining. It might be helpful to guide the search or evaluate the stake of the result patterns.
- The knowledge base may even contain user views and data from user experiences that might be helpful in the data mining process.
- The data mining engine may receive inputs from the knowledge base to make the result more accurate and reliable.
- The pattern assessment module regularly interacts with the knowledge base to get inputs, and also update it.

GQ. Suppose your task as a software engineer at DB-University is to design a data mining system to examine their university course database, which contains the following information: the name, address, and status (e.g., undergraduate or graduate) of each student, the courses taken, and their cumulative grade point average (GPA). Describe the architecture you would choose. What is the purpose of each component of this architecture?

A data mining architecture that can be used for this application would consist of the following major components :

- A database, data warehouse, or other information repository, which consists of the set of databases, data warehouses, spreadsheets, or other kinds of information repositories containing the student and course information.
- A database or data warehouse server which fetches the relevant data based on users' data mining requests.
- A knowledge base that contains the domain knowledge used to guide the search or to evaluate the interestingness of resulting patterns. For example, the knowledge base may contain metadata which describes data from multiple heterogeneous sources.
- A data mining engine, which consists of a set of functional modules for tasks such as classification, association, classification, cluster analysis, and evolution and deviation analysis.
- A pattern evaluation module that works in tandem with the data mining modules by employing interestingness measures to help focus the search towards interestingness patterns.
- A graphical user interface that allows the user an interactive approach to the data mining system.

1.18 KDD PROCESS

UQ. Explain KDD process with diagram. **MU – Dec. 2019**

UQ. What is data mining? Explain KDD process with diagram. **MU – May 2019**

- Knowledge discovery in the database (KDD) is the process of searching for hidden knowledge in the massive amounts of data that we are technically capable of generating and storing.
- The basic task of KDD is to extract knowledge (or information) from a lower level data (databases).
- It is the non-trivial (significant) process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.
- The goal is to distinguish between unprocessed data something that may not be obvious but is valuable or enlightening in its discovery.
- The overall process of finding and interpreting patterns from data involves the repeated application of the following steps :

1.	Data Cleaning	2.	Data Integration
3.	Data Selection	4.	Data Transformation
5.	Data Mining	6.	Pattern Evaluation
7.	Knowledge Presentation		

1. Data Cleaning

- Removal of noise, inconsistent data, and outliers
- Strategies to handle missing data fields.

2. Data Integration

- Data from various sources such as databases, data warehouse, and transactional data are integrated.
- Multiple data sources may be combined into a single data format.

3. Data Selection

- Data relevant to the analysis task is retrieved from the database.
- Collecting only necessary information to the model.
- Finding useful features to represent data depending on the goal of the task.

4. Data Transformation

- Data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations.
- By using transformation methods invariant representations for the data is found.

5. Data Mining

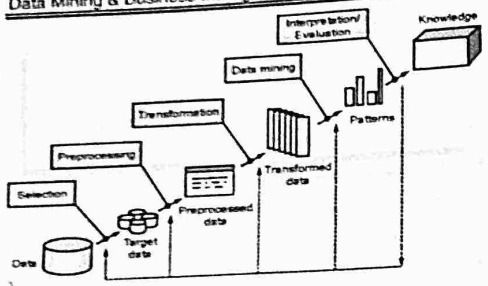
- An essential process where intelligent methods are applied to extract data patterns.
- Deciding which model and parameter may be appropriate.

6. Pattern Evaluation

To identify the truly interesting patterns representing knowledge based on interesting measures.

7. Knowledge Presentation

- Visualization and knowledge representation techniques are used to present mined knowledge to users.
- Visualizations can be in form of graphs, charts or table.



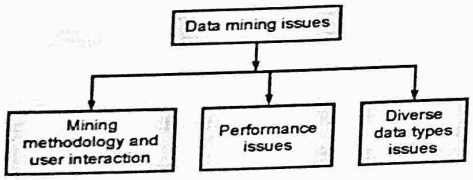
(182)Fig. 1.18.1: KDD Process

1.19 ISSUES IN DATA MINING

UQ. What are the major issues in data mining? **MU- May 2019**

Data mining systems face a lot of challenges and issues in today's world. Some of them are:

1. Mining methodology and user interaction issues
2. Performance issues
3. Issues relating to the diversity of database types



(183)Fig. 1.19.1 : Data Mining Issues

1. Mining Methodology and User Interaction Issues

The issues in this category are as follows:

- **Mining different kinds of knowledge in databases :** This issue is responsible for addressing the problems of covering a big range of data in order to meet the needs of the client or the customer. Due to the different information or a different way, it becomes difficult for a user to cover a big range of knowledge discovery task.
- **Interactive mining of knowledge at multiple levels of abstraction :** Interactive mining is very crucial because it permits the user to focus the search for patterns, providing and refining data mining requests based on the results that were returned. In simpler words, it allows user to focus the search on patterns from various different angles.
- **Incorporation of background of knowledge :** The main work of background knowledge is to continue the process of discovery and indicate the patterns or trends that were seen in the process. Background knowledge can also be used to express the patterns or trends observed in brief and precise terms. It can also be represented at different levels of abstraction.
- **Data mining query languages and ad hoc data mining :** Data Mining Query language is responsible for giving access to the user such that it describes ad hoc mining tasks as well and it needs to be integrated with a data warehouse query language.
- **Presentation and visualization of data mining results :** In this issue, the patterns or trends that are discovered are to be rendered in high level languages and visual representations. The representation has to be written so that it is simply understood by everyone.
- **Handling noisy or incomplete data :** For this process, the data cleaning methods are used. It is a convenient way of handling the noise and the incomplete objects in data mining. Without data cleaning methods, there will be no accuracy in the discovered patterns. And then these patterns will be poor in quality.

2. Performance Issues

It has been noticed several times there are performance related issues in data mining as well. These issues are as follows :

- **Efficiency and Scalability of data mining algorithm :** Efficiency and Scalability is very important when it comes to data mining process. It is also very necessary because with the help of using this, the user can withdraw the information from the data in a more effective and productive manner. On top of that, the user can withdraw that information effectively from the large amount of data in various databases.

- **Parallel, distributed and incremental mining algorithm :** There are a lot factors which can be responsible for the development of parallel and distributed algorithms in data mining. These factors are large in size of database, huge distribution of data, and data mining method that are complex. In this process, in the first and foremost step, the algorithm divides the data from database into various partition. In the next step, that data is processed such that it is situated in parallel manner. Then in the last step, the result from the partition is merged.

3. Diverse Data Types Issues

The issues in this category are given below:

- **Handling of relational and complex types of data :** The database may contain the various data objects. For example, complex, multimedia, temporal data, or spatial data objects. It is very difficult to mine all these data with the help of a single system.
- **Mining information from heterogeneous databases and global information systems :** The problem in this kind of issue is to mine the knowledge from various data sources. These data are not available as a single source instead these data are available at the different data sources on LAN or WAN. The structures of these data are different as well.

1.20 APPLICATIONS OF DATA MINING

The importance of data mining and analysis is growing day by day in our real life. Today most organizations use data mining for analysis of Big Data. Let us see how this technology benefit different users.

1. Mobile Service Providers	2. Retail Sector
3. Artificial Intelligence	4. E-commerce
5. Science and Engineering	6. Crime Prevention
7. Research	8. Farming
9. Automation	10. Dynamic Pricing
11. Transportation	12. Insurance

1. Mobile Service Providers

- Mobile service providers use data mining to design their marketing campaigns and to retain customers from moving to other vendors.
- From a large amount of data such as billing information, email, text messages, web data transmissions, and customer service, the data mining tools can predict "churn" that tells the customers who are looking to change the vendors.
- With these results, a probability score is given. The mobile service providers are then able to provide incentives, offers to customers who are at higher risk of churning. This kind of mining is often used by major service providers such as broadband, phone, gas providers, etc.

2. Retail Sector

- Data Mining helps the supermarket and retail sector owners to know the choices of the customers. Looking at the purchase history of the customers, the data mining tools show the buying preferences of the customers.
- With the help of these results, the supermarkets design the placements of products on shelves and bring out offers on items such as coupons on matching products, and special discounts on some products.
- These campaigns are based on RFM grouping. RFM stands for recency, frequency, and monetary grouping. The promotions and marketing campaigns are customized for these segments. The customer who spends a lot but very less frequently will be treated differently from the customer who buys every 2-3 days but of less amount.
- Data Mining can be used for product recommendation and cross-referencing of items.

3. Artificial Intelligence

- A system is made artificially intelligent by feeding it with relevant patterns. These patterns come from data mining outputs. The outputs of the artificially intelligent systems are also analysed for their relevance using the data mining techniques.
- The recommender systems use data mining techniques to make personalized recommendations when the customer is interacting with the machines. The artificial intelligence is used on mined data such as giving product recommendations based on the past purchasing history of the customer in Amazon.