Data Mining 2078/79

Model Questions

1 a. Define data mining with its functionalities. Mention important application of data mining.

**Data Mining*

Data Mining is the process of analysing large sets of data to discover patterns, correlations, or relationships that can be useful for making decisions or predictions. It involves various techniques from statistics, machine learning, and database system to extract insights and valuable information from row data.

**Its Functionalities*

Data Mining Functionalities are can be divided into 2 categories:

#Descriptive Data Mining

It includes certain knowledge to understand what is happing within the data without a previous idea. The common data features ore highlighted in the data set. For example: count, average etc.

→ Class or Concept Description

Class or concept refers to the data that is linked or correlated with some classes or some concepts.

→ Mining of Frequent Patterns

They can defined as the patterns that takes place very aften in transactional data.

→ Mining of Association

Mining of Association are mainly used in retail sales in order to identify patterns that are very aften purchased together

→ Mining of Correlation

Mining of Correlation refers to a type of Descriptive Data Mining's Functions that are usually executed in order to revel or expose some statistical correlation between associated attributes value pairs or between two item sets.

→ Mining of Clusters

The literal meaning of the word “Cluster” is a group in some of things which are similar to one another way another.

#Predictive Data Mining

It helps developers to provides unlabelled definitions of attributes. Based on previous tests. the software estimates the characteristics that ore absent. For example: Judging from the finding of a patient's medical examinations that is he suffering from any particular disease.

→ Decision Tree

A decision tree is a like a flow chart with o tree structure, in which every junction/node is used to represent a test on an attribute value, moreover, each and branch is responsible for representing the concluding outcome of the test, and tree leaves are used to represent the classes or the distribution of classes.

→ Neural Network

Neural network. a key component of predictive data mining, are a type machine learning algorithm inspired by the structure and functioning of the human brain. They consist of interconnected nodes, or neurons, organized in layers (input, hidden, output) that process information and learn patterns from data.

**** The important application of data Mining:***

→ Marketing

Targeted advertising, customer segmentation, and personalized recommendations based on Purchasing behaviour.

→Health care

Predictive analytics for disease diagnosis. treatment optimization, and patient outcome prediction

→ Finance

Frond detection, risk assessment, and stock market analysis for investment decision

→ Retail

Inventory management, market basket analysis, demand forecasting.

→Telecommunications

Customer Churn prediction, network optimization and Improving service quality

→ Manufacturing

Quality control predictive maintenance, and supply chain optimization

→ Science and Research

Dang discovery, genomic analysis, and environmental research.

b. What is Fact constellation schemas? Explain star schema with its advantages and dis-advantages.

*** Fact constellation schemas**

fact constellation schema, also known as galaxy schema, is a multi-dimensional modelling technique used in data ware housing. It involves multiple fact tables that shore dimension tables, forming a complex network or constellation or relationships.

* Star Schema

This is the simplest and most effective schema in data warehouse. A fact table in the centre surrounded by multiple dimension tables resembles a star in the star schema model.

A star schema is a type of data warehouse schema that consists of one or more fact tables referencing multiple dimension tables. It is called star schema because the structure resembles a star, with the fact table at the centre and dimension tables surrounding it.

An example of a star schema is given below:

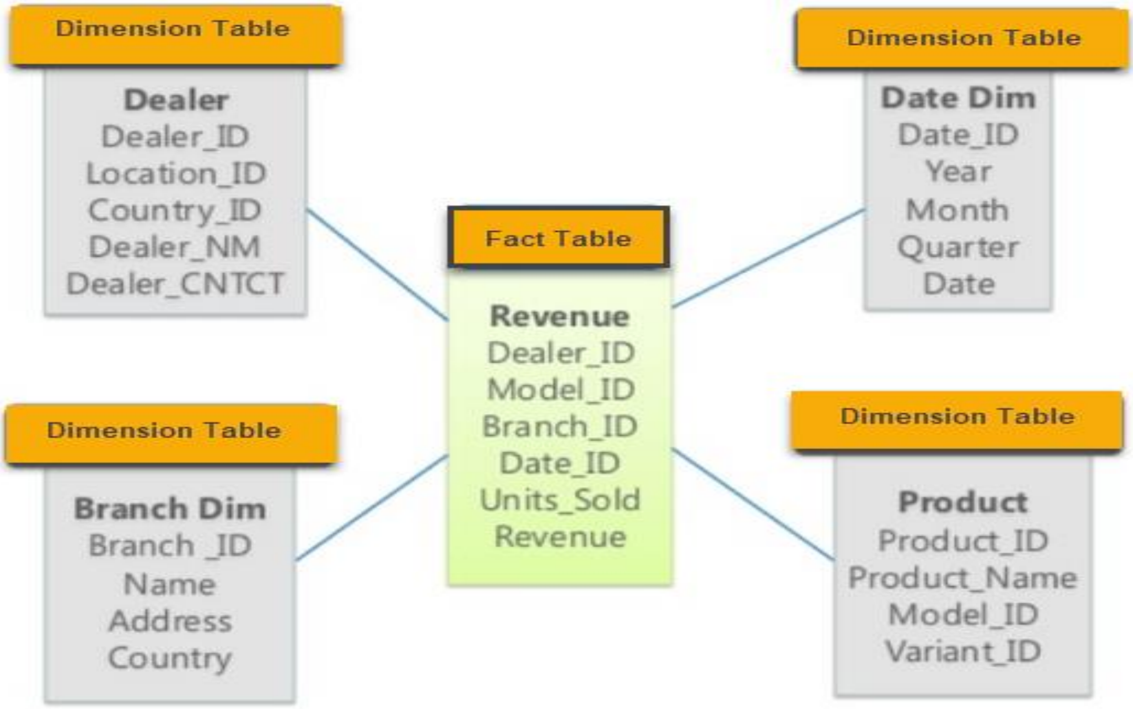


Fig:- Star Schema

*Advantages

→ Simplicity

Easy to understand and navigate, making queries simpler and more intuitive..

→ Query Performance

Generally offers faster query performance, as it involves feu L compared to other schema typer, fa analytical queries.

→ Optimized for Aggregation

Well-suited for aggregations and OLAP (Analytical Processing) tools, enabling quick efficient dato analysis.

* Disadvantages

→ Redundancy

Can potentially lead to data redundancy, especially if multiple, dimensions share similor attributes.

→ Limited Flexibility

Might not handle, complex relationships or evolving data requirements as effeciently as more complex a Schema types like snowflake scheme or fact constellaton schema.

→ Normalization Challenges

In certain cases, difficulties might arise when typing to normalize or reorganize data due to its demo denormalized nature.

2a. Describe data warehousee. Differentiate data Warehase with DBMS.

*** Data Warehouse**

A data warehouse is a centralized repository that stores large volumes of structured and often, historical data from various sources within an organization. It is designed for efficient querying, reporting, and analysis to support decision-making processes. Data warehouses provide a consolidated and integrated view of data from different departments and systems, making it easier for users to derive insights and make informed business decisions.

key characteristics of a dato warehouse:

1. Centralized Storage

It consolidates data from vasions operational Sources into a single repository, providing a unified view of the organizations informations.

2. Subject-Oriented

Organized around key subject areas of interest (eg., Sales, fironce marketing) rather than focusing on day-to-day transactions.

3. Time-Varient

Stores historical data, allowing for trend analysis, comparisons, and tracking changes over time.

*** data warehouse with DBMS**

→ Data Warehouse.

1. Purpose

Focuses on storing and analyzing large volumes of historical data multiple sources for business intelligence and decision-making.

2. Data Type

Handles structured, integrated, and aften aggregated data for reporting and analysis.

3. Design

Optimized fo querying, reporting, and analytics Supporting complex analysis across various dimensions.

4. Usage

Used by analysts, decision-makers, and business intelligence tools for strategic insights.

5. Schema

Utilizes star schema, snowflake schema, or other malti-dimensional models to organize data for analytical purposes.

6. Performance

Emphasizes fast query performance for analytical operations over a vast amount of historical data.

7. Data Quality

Requires high date quality and consistency for reliable analysis and decision making.

5. Updates

Primarily loaded with periodic or batch updates rather than frequent real-time updates.

→ DBMS (Database Management System

1. Purpose

Nonages operational data for day-to-day transactions and applications..

2. Data type

Handles structured, transactional data ensuring data integrity and ACID properties (Atomicity Consistency, Isolation, Durability).

3. Design

Organizes data for efficient transaction processing and retrieval for operational applications.

4. Usage

Used by applications and systems for routine CRUD operations (Create, Reade, Update, Delete).

S. Schema

Follow normalized or partially denormalized schemas for efficient transactional processing.

6. Performance

Optimized on for transactional performance, ensuring quick dato insertion, updates, ond retrieval.

7. Dato Quality

Prioritizes data consistency and integrity to maintain accurate transactional records.

8. Updates

Supports frequent real-time updates as it monages live operational data.

b. What are the important characteristics of OLTP. Differentiate OLAP with OLTP.

*** The important characteristics of OLTP ore:**

1. Transactional Nature

OLTP system deal with a high volume of short, otomic transactions such as record inserts. updates, and deletes.

2. Data Integrity

Ensures data consistency and integrity by enforcing ACID properties for each transaction.

3. Concurrency Control

Monages multiple transactions occurring simultaneously, ensuring that they do not interfere with each other and maintaining data consistency.

4. Normalized Database Structure

Typically follows a normalized database structure to minimize data redundancy and improve data integrity, making it suitable for frequent updates.

5. Quick Response Time

Designed for fast response times to support real-time trasaction processing, making it crucial for operational efficiency.

6. Con-current User Support

Capable of handling a Lorge number of con-current users who are performing various transactions simultaneously.

Differentiate OLAP with OLTP

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

↳ Involves historical processing of information.

↳ OLAP systems are used by knowledge workers Such as executives, managers and analysts.

↳ Useful in analyzing the business.

↳ It focuses on Information out.

↳ Based on star schema, Snowflake, Schema and fect Constellation schema.

↳ Contains historical data.

↳ Provides summarized and consolidated data.

↳ Provides summarized and multidimensional view of data.

↳ Number or users is in hundreds.

↳ Number of records accessed in millions

↳ Database size is from 100 GB to 1TB.

↳ Highly flexible.

**** OLTP (Online Transaction Processing)***

↳ Involves day-to-day processing.

↳ OLTP systems are usest by cleaks, DBAs, Or database professionals.

↳ Useful in running the business.

↳ It ficuses on Data in.

↳ Based on Entity Relationship Model.

↳ Contains corrent data

↳ Provides primitive and highly detailed dato.

↳ Provida detailed and flat relational view of doto.

↳ Namber of users is in thousands.

↳ Number of records accessed is in tens

↳ Database size is from 100 MB to 1GB

↳ Provides high performance.

3a. What is clustering. Explain linear and Non- linear regression.

*** Clustering**

Clustering is a technique in machine learning and data mining that involves grouping similar objects or data points together based on their characteristics or attributes. The goal of clustering is toidentify natural groupings or patterns within the data, which can then be used for further analysis or decision-making.

*** Linear Regression**

Linear regression is a modelling technique that assumes a linear relationship between the independent and dependent variables. It aims to fit a linear equation to the observed data to predict the values of the dependent variable.

→ ***Uses***

1. Predictive Modelling:

   Linear regression is commonly used for predicting numeric outcomes, such as sales forecasts based on various factors like advertising spending.

2. Pattern Recognition:

   It helps identify and quantify linear patterns in the data, facilitating the understanding of how changes in independent variables influence the dependent variable.

→ ***Linear Regression Example:***

↳ ***Scenario:***

Imagine you are analyzing the relationship between the number of hours students spend studying (independent variable, X) and their exam scores (dependent variable, Y).

↳ ***Linear Regression Equation:***

The linear regression equation for this scenario might look like:

Y = mx + b

Where,

m is the slope and

b is the y-intercept.

*** Non-linear Regression**

Non-linear regression involves modelling the relationship between variables using a non-linear equation. Unlike linear regression, this approach allows for more complex, non-linear patterns in the data.

→ **Uses:**

1. Complex Relationships:

Non-linear regression is utilized when the relationship between variables is not adequately captured by a linear model. It accommodates more intricate patterns, such as exponential growth or logarithmic decay.

2. Improved Model Fit:

It is applied when the data exhibits curvature or non-linearity, providing a better fit to the observed patterns.

→ **Non-linear Regression Example:**

↳ **Scenario:**

Now, consider a scenario where you're analyzing the growth of a bacterial population (dependent variable, Y) over time (independent variable, X).

↳ **Non-linear Regression Equation:**

A common non-linear model for population growth is the exponential growth equation:

$\mathbf{Y = a \times} e^{bx}$

where

a is the initial population,

b is the growth rate,

e is the base of the natural logarithm

x is time.

b. Define decision tree. Explain entropy and in formation gain in details.

* Decision Tree

A decision tree is a like a flow chart with a tree structure, in which every junction/node is used to represent a test on an attribute value, moreover, each and every branch is responsible for representing the concluding outcome of the test, and tree leaves are used to represent the classes or the distribution of classes.

*** Entropy**

Entropy is a measure of impurity or disorder in a set of data. In the context of decision trees, it quantifies the uncertainty associated with a particular group of data points. The goal in decision tree algorithms is to reduce entropy, meaning to create subsets (nodes) that are as homogeneous as possible.

Mathematical Formula:

For a binary classification problem with classes A and B, the entropy (H) is calculated as follows:

$H = -P_A . Log_2 (P_A) - P_B . Log_2(P_B)$

Where P_A and P_B are the probabilities of occurrence of classes A and B.

Interpretation:

- If a dataset is perfectly homogeneous (all samples belong to one class), the entropy is zero.
- Higher entropy indicates more disorder and uncertainty in the dataset.

*** Information Gain**

Information Gain is a metric used to evaluate the effectiveness of a feature in reducing entropy. It measures the difference in entropy before and after a dataset is split based on a particular feature. The attribute with the highest information gain is chosen as the splitting criterion.

Mathematical Formula:

For a dataset *D* and a feature *X*, the Information Gain (*IG*) is calculated as follows:

$$(D,X) = H(D) - \sum v \in \text{Values}(X)|D||Dv| \cdot H(Dv)$$

where:

- $H(D)$ is the entropy of the dataset D.
- Values(X) is the set of all possible values for feature X.
- Dv is the subset of data points in D for which feature X has the value v.

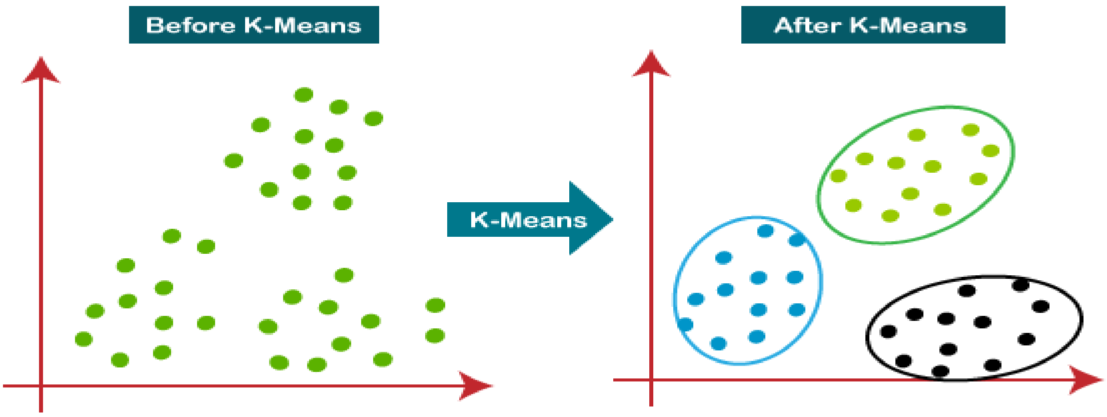
Interpretation:

- Higher Information Gain implies that the feature *X* is more informative for splitting the dataset.
- Features with higher Information Gain are preferred as they lead to subsets with lower entropy.

4a. What are the drawbacks of k-mean algorithm? Explain agglomerative clustering in brief.

→ * K-mean Algorithm

The k-means algorithm is a popular clustering algorithm used for partitioning a dataset into a specified number (k) of distinct, non-overlapping subsets or clusters. The goal is to group similar data points together and identify underlying patterns within the data. The algorithm operates iteratively and is based on the minimization of the sum of squared distances between data points and the centroid of their assigned cluster.



K-Means Algorithm Steps:

1. ***Initialization:***

    - Choose the number of clusters (*k*).
    - Randomly initialize *k* cluster centroids, one for each cluster.

2. ***Assignment:***

    - Assign each data point to the nearest cluster centroid.
    - Use a distance metric, commonly the Euclidean distance, to measure the distance between data points and centroids.

3. ***Update Centroids:***

    - Recalculate the centroid of each cluster as the mean of all data points assigned to that cluster.

4. ***Repeat:***

    - Repeat steps 2 and 3 until convergence.
    - Convergence occurs when the assignment of data points to clusters no longer changes significantly or when a specified number of iterations is reached.

Key Considerations:

1. ***Choice of k:***

    - The number of clusters, *k*, must be specified before running the algorithm. Various methods, such as the elbow method or silhouette analysis, can be used to determine an optimal *k* value.

2. ***Initialization Sensitivity:***

    - The algorithm's performance is sensitive to the initial placement of centroids. Different initializations may lead to different final clusters.

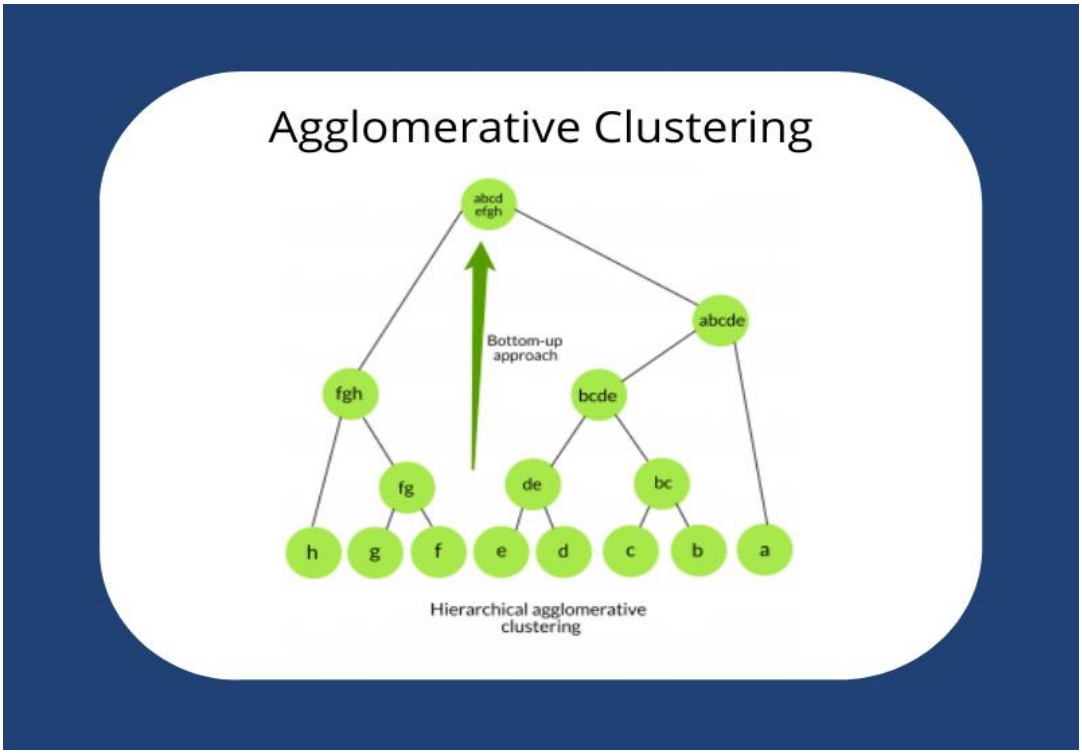
3. ***Convergence:***

    - K-means aims to minimize the sum of squared distances within clusters, but it may converge to a local minimum. Multiple runs with different initializations can help mitigate this issue.

4. ***Scalability:***
    - K-means can scale to large datasets, but for extremely large datasets, variants like Mini-Batch K-Means are often used.

* Agglomerative Clustering

Agglomerative clustering is a hierarchical clustering algorithm that builds a tree of clusters. The algorithm starts by treating each data point as a single cluster and iteratively merges the closest pairs of clusters until only a single cluster remains. This process results in a dendrogram, which is a tree-like structure that visually represents the hierarchy of clusters. Agglomerative clustering is a "bottom-up" approach, where the algorithm starts with individual data points and successively merges them into larger clusters.



Steps of Agglomerative Clustering:

1. ***Initialization:***
    - Treat each data point as a single cluster.
2. ***Calculate Pairwise Distances:***
    - Compute the pairwise distances between all clusters.

3. ***Merge Closest Clusters:***
    - Find the two clusters with the smallest distance between them and merge them into a new cluster.
    - Update the distance matrix to reflect the distances between the new cluster and the remaining clusters.
4. ***Repeat:***
    - Repeat steps 2 and 3 until only one cluster remains.

b. Explain DMQL with its syntax and example.

→ *** DMQL** (Data Mining Query Language)

DMQL (Data Mining Query Language) is a language designed for querying and extracting information from data mining models and systems. It provides a standardized way to interact with data mining models, allowing users to retrieve valuable insights, patterns, and knowledge from the underlying data.

*** Syntax of DMQL**

1. DMQL-Syntax for task-relevant data specification

Names of the relevant database or data warehouse, conditions, and relevant attributes or dimensions must be specified:

use database ‹database_name› or **use data warehouse** ‹data_warehouse_name›

from ‹relation(s)/cube(s)› [where condition] (data cubes and tables) in relevance to ‹attribute_or_dimension_list› (attributes or dimension for exploration)

order by ‹order_list› (sorting order)

group by ‹grouping_list› (specifies criteria to group)

having ‹condition› (it represent which group of data are considered relevant)

2. Syntax for specifying the kind of knowledge.

Syntax for Characterization, Discrimination, Association, Classification, and Prediction.

a. Data mining characterization

The syntax for characterization is −

mine characteristics [as pattern_name]

analyze {measure(s)}

b. Data mining discrimination

The syntax for Discrimination is −

```
mine comparison [as {pattern_name]}
For {target_class } where {t arget_condition }
{versus {contrast_class_i }
where {contrast_condition_i}}
analyze {measure(s) }
```

c. Data mining association

The syntax for Association is−

```
mine associations [ as {pattern_name} ]
{matching {metapattern} }
```

d. Data mining classification

The syntax for Classification is −

```
mine classification [as pattern_name]
analyze classifying_attribute_or_dimension
```

e. Data mining prediction

The syntax for prediction is −

```
mine prediction [as pattern_name]
analyze prediction_attribute_or_dimension
{set {attribute_or_dimension_i= value_i}}
```

3. Syntax for Concept Hierarchy Specification

To specify concept hierarchies, use the following syntax −

```
use hierarchy <hierarchy> for <attribute_or_dimension>
```

For Example −

```
with support threshold = 0.05
with confidence threshold = 0.7
```

4. Syntax for Pattern Presentation and Visualization Specification

We have a syntax, which allows users to specify the display of discovered patterns in one or more forms.

```
display as <result_form>
```

For Example −

```
display as table
```

* Examples of DMQL

Example 1: where 'numeric_field' is equal to 2

DMQL query: (numeric_field=2)

Equivalent SQL server query: WHERE numeric_field = 2

Example 2: where 'numeric_field' is greater than or equal to 3

DMQL query: (numeric_field=3+)

Equivalent SQL server query: WHERE numeric_field >= 3

Example 3: where 'numeric_field' is less than or equal to 8

DMQL query

:(numeric_field=8-)

Equivalent SQL server query: WHERE numeric_field

Example 4: where 'numeric_field' is between 5 and 9 (including 5 and 9)

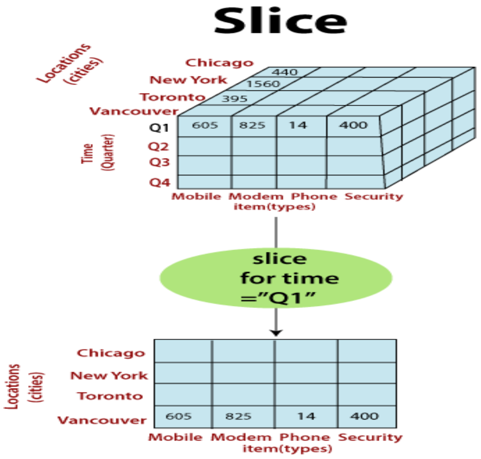
DMQL query:(numeric_field=5-9)

Equivalent SQL server query: WHERE numeric_field between 5 and 9

5a. What do you mean by slice and dice, drill up and drill down in multidimensional data?

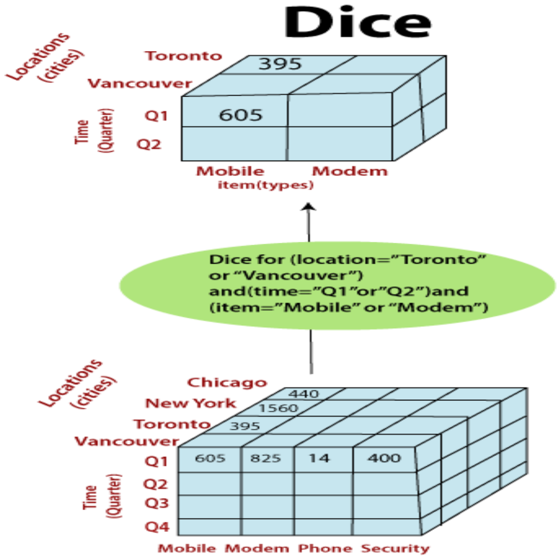
→ * Slice in Multi-dimensional Data

Slicing is a technique in multidimensional data analysis where a specific "slice" or subset of a data cube is extracted by fixing one or more dimensions at particular values while keeping others unrestricted.



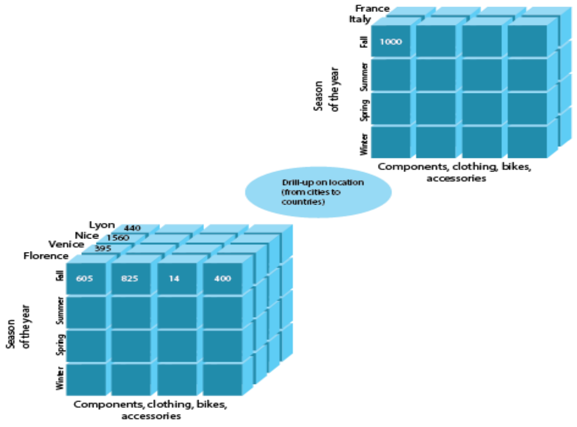
* Dice in Multi-dimensional data

Dicing is a more specific operation than slicing. It involves selecting a sub cube by fixing values for more than one dimension, narrowing down the data to a specific subset.



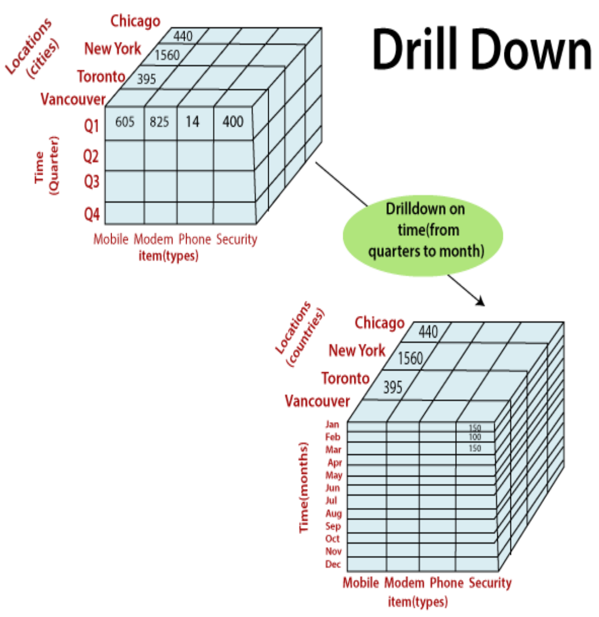
* Drill-up in Multi-dimensional Data

Drill Up is the reverse of drill down. It involves moving from a lower level of detail to a higher, more aggregated level. It summarizes detailed data to a higher level of abstraction.



* Drill Down in Multi-dimensional Data

Drill Down is the process of navigating from a higher level of abstraction to a lower, more detailed level in a data hierarchy. It involves breaking down aggregated data into finer details by moving from higher-level categories to lower-level ones.



b. Explain advance data mining with its important features.

→ * Advance Data Mining

Advanced data mining refers to the use of sophisticated techniques and algorithms to discover patterns, trends, and insights in large datasets. It goes beyond basic data mining methods to address complex challenges and make more accurate predictions.

* Its Important Features:

↳ *Feature Engineering:*

Creation and selection of relevant features to enhance model accuracy by highlighting key information and reducing noise.

↳ *Ensemble Learning:*

Combination of multiple models to improve predictive performance, reducing overfitting and increasing model robustness.

↳ *Deep Learning:*

Use of neural networks with multiple layers to automatically learn intricate patterns and representations from complex data.

↳ *Text and Sentiment Analysis:*

Analysis of unstructured text data for sentiment analysis, topic modeling, and information extraction.

↳ *Temporal and Spatial Data Mining:*

Analysis of data over time or space to identify trends, patterns, and correlations specific to those dimensions.

↳ *Anomaly Detection:*

Identification of anomalies or outliers in data, crucial for detecting fraud, errors, or unusual patterns.

↳ *Clustering and Segmentation:*

Utilization of advanced clustering techniques to group similar data points together, providing insights into the inherent structure of the data.

↳ *Interactive Data Exploration:*

Inclusion of interactive interfaces that allow users to dynamically explore and visualize data, facilitating a deeper understanding of patterns and relationships.

↳ *Scalability:*

Design of algorithms and tools to handle large-scale datasets efficiently, often leveraging parallel and distributed computing.

↳ *Ethical and Responsible Data Mining:*

Consideration of ethical considerations and responsible AI practices, including fairness, transparency, and bias mitigation.

6.Write short notes on (Any Four)

i) K-me doid algorithm

The K-medoids algorithm is a clustering algorithm that is similar to the K-means algorithm, but it uses medoids (data points that most represent their cluster) instead of centroids to define clusters. This makes K-medoids more robust to outliers and noise in the data.

*** K-medoids Algorithm Steps:**

↳ *Initialization*:

Randomly select k data points as initial medoids.

↳ *Assignment:*

Assign each data point to the nearest medoid based on a chosen distance metric (commonly, Euclidean distance).

↳ *Update Medoids:*

For each cluster, evaluate the total cost (sum of distances) of replacing the current medoid with each non-medoid point.Choose the point with the lowest total cost as the new medoid for that cluster.

↳ *Repeat:*

Repeat steps 2 and 3 until convergence or a specified number of iterations.

*** Advantages of K-medoids:**

1. *Robust to Outliers:*
    - K-medoids is less sensitive to outliers and noise compared to K-means because it uses medoids instead of centroids.

2. *Handles Non-Euclidean Distances:*

   - K-medoids can accommodate various distance metrics beyond the Euclidean distance, making it suitable for different types of data.

*** Disadvantages of K-medoids:**

1. ***Computationally Intensive:***

   - The algorithm can be computationally expensive, especially for large datasets, as it involves pairwise distance calculations.

2. *Sensitive to Initial Medoid Selection:*

   - The choice of initial medoids can influence the final clustering result.

ii) Deep Learning

Deep learning is a subset of machine learning that involves the use of neural networks with multiple layers (deep neural networks) to automatically learn and represent complex patterns and features from data. These neural networks, inspired by the structure and function of the human brain, have the capacity to model intricate relationships and hierarchies within large datasets. Deep learning has gained immense popularity and success in various domains, including computer vision, natural language processing, speech recognition, and more.

*** Key Features of Deep Learning:**

1. *Neural Networks:*

   Deep learning relies on neural networks composed of interconnected layers of artificial neurons, mimicking the structure of biological neural networks.

2. *Deep Architectures:*

   Deep learning models often consist of multiple hidden layers, allowing them to learn complex representations of data.

3. ***Representation Learning:***

    Deep learning algorithms automatically learn hierarchical representations of data, extracting relevant features at different levels of abstraction.

4. *End-to-End Learning:*

    Deep learning models can perform end-to-end learning, directly mapping raw input data to output predictions without relying on manual feature engineering.

5. *Adaptability to Data:*

    Deep learning models can adapt to diverse and high-dimensional data types, including images, text, and sequential data.

6. *Transfer Learning:*

    Transfer learning involves using pre-trained models on one task to improve performance on a related task.

7. *Complex Task Solving:*

    Deep learning excels at solving complex tasks such as image and speech recognition, natural language understanding, and playing strategic games.

8. *Scalability:*

    Deep learning models can scale to large datasets and leverage parallel and distributed computing.

9. *Frameworks and Libraries:*

    Various deep learning frameworks and libraries, such as TensorFlow and PyTorch, facilitate the development, training, and deployment of deep learning models.

10. *Application Diversity:*

    Deep learning finds applications in diverse fields, including computer vision (image and video analysis), natural language processing (text and speech understanding), healthcare, finance, and autonomous systems.

iii) Fp-growth Algorithm

The FP-Growth (Frequent Pattern Growth) algorithm is a popular and efficient algorithm for mining frequent item sets in a transactional database. It's commonly used for association rule mining in data mining and market basket analysis. Developed by Han, Pei, and Yin in 2000, FP-Growth is known for its ability to handle large datasets and produce compact data structures for efficient mining.

*** Key Steps of FP-Growth:**

1. *Create a Frequency Table:*
    - Scan the database to count the frequency of each item (single items are called 1-itemsets).
2. *Build the FP-Tree:*
    - Create a data structure called the FP-Tree by inserting transactions into a tree structure based on the frequency of items.
3. *Generate Conditional Pattern Bases:*
    - For each frequent item, create a conditional pattern base by extracting the paths in the FP-Tree that correspond to that item.
4. *Recursive Mining:*
    - Recursively apply the FP-Growth algorithm on each conditional pattern base to generate frequent patterns.
5. *Combine Patterns:*
    - Combine the patterns obtained from the recursive mining step to produce the final set of frequent item sets.

*** Advantages of FP-Growth:**

1. *Efficiency:*
    - FP-Growth is more efficient than Apriori, another frequent itemset mining algorithm, especially for large datasets, as it requires only two passes over the data.

2. *Compact Data Structure:*

   - The FP-Tree data structure is more compact compared to the frequent itemset generation and storage in Apriori.

3. *Reduced Candidate Generation:*

   - FP-Growth eliminates the need to generate and test candidate item sets, reducing computational overhead.

4. *Handles Sparse Data:*

   - It performs well even with sparse datasets where the traditional Apriorism algorithm may struggle.

iv) MOLAP(Multidimensional Online Analytical Processing)

MOLAP is a category of OLAP (Online Analytical Processing) that stores and analyzes data in a multidimensional cube format. It is one of the three main OLAP storage modes, alongside ROLAP (Relational OLAP) and HOLAP (Hybrid OLAP). MOLAP databases are designed for efficient querying and analysis of multidimensional data structures.

*** Key Features of MOLAP:**

1. *Multidimensional Cube:*

   MOLAP databases organize data into a multidimensional cube structure, where each axis represents a dimension, and the cells contain aggregated data.

2. *Fast Query Performance:*

   MOLAP databases pre-aggregate data at different levels, facilitating fast query performance for common aggregations and calculations.

3. *Storage Efficiency:*

   MOLAP databases store aggregated data in a compressed form, optimizing storage efficiency.

4. *Support for Complex Calculations:*

   MOLAP databases support complex calculations and operations directly within the multidimensional cube, enhancing analytical capabilities.

5. *Hierarchy Navigation:*

   MOLAP supports hierarchies within dimensions, allowing users to navigate from higher-level summaries to detailed data and vice versa.

6. *Offline Data Processing:*

   MOLAP databases often support offline data processing, enabling users to perform analysis even when disconnected from the main data source.

7. *Advanced Security:*

   MOLAP systems offer advanced security features to control access to different levels of data, dimensions, and functionalities.

8. *Integrated Data Storage and Analysis:*

   MOLAP integrates data storage and analysis within a single system, minimizing the need for separate databases and analytical tools.

iv)DMQL (Data Mining Query Language)

DMQL, or Data Mining Query Language, is a specialized language designed for querying and interacting with data mining models and systems. It provides a standardized way to request and extract information from data mining models, facilitating the retrieval of valuable insights and patterns discovered during the data mining process.

*** Key Features of DMQL**

1. *Model Interaction:*

   DMQL allows users to interact with data mining models, including querying model parameters, requesting predictions, and retrieving patterns.

2. *Querying Model Metadata:*

DMQL enables queries on the metadata associated with data mining models, such as model properties, features, and statistical measures.

3. *Prediction Requests:*

   Users can submit prediction requests to data mining models using DMQL, obtaining model-generated predictions for new or existing data points.

4. *Pattern Retrieval:*

   DMQL supports the retrieval of discovered patterns, rules, or associations from data mining models.

5. *Filtering and Subset Extraction:*

   DMQL allows users to apply filters and criteria to extract specific subsets of data from the model's output.

6. *Integration with Databases:*

   DMQL is often designed to integrate seamlessly with relational databases, making it easy to combine data mining results with existing structured data.

7. *Standardization:*

   DMQL provides a standardized syntax and set of commands for interacting with data mining models, ensuring consistency across different systems.

Data Mining

2079

Model Questions

Attempt any FIVE questions.

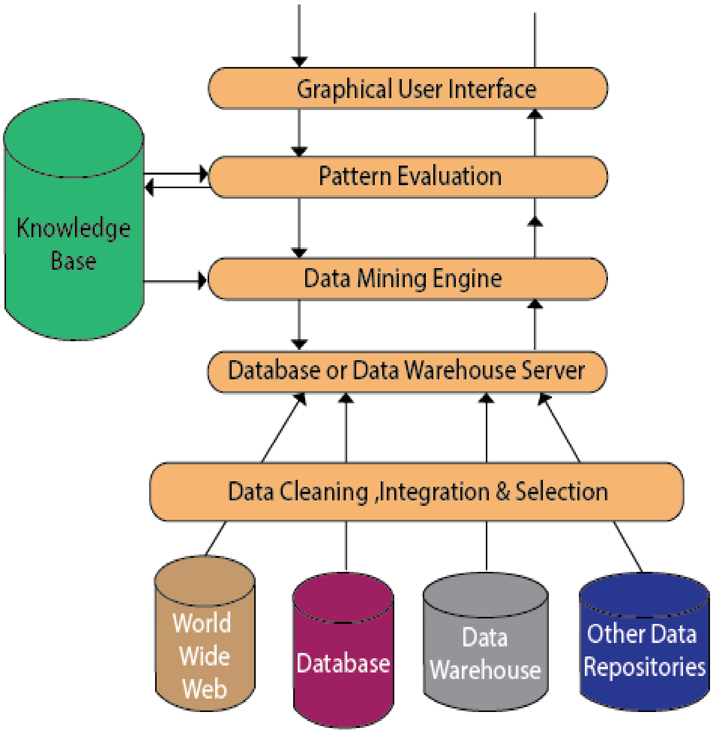
1.a. What is data warehouse? Explain data mining system architecture.[2+6]

→ * Data Warehouse

A data warehouse is a centralized repository that stores large volumes of structured and often, historical data from various sources within an organization. It is designed for efficient querying, reporting, and analysis to support decision-making processes. Data warehouses provide a consolidated and integrated view of data from different departments and systems, making it easier for users to derive insights and make informed business decisions.

* Data Mining System Architecture

Mining is the process of analysing large sets of data to discover patterns, correlations, or relationships that can be useful for making decisions or predictions. It involves various techniques from statistics, machine learning, and database system to extract insights and valuable information from row data.



↳ 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.

↳ 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.

↳ 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.

↳ 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 stores the patterns, rules, and models discovered during the data mining process. It acts as a repository for actionable knowledge.

↳ User Interface

The user interface is the front-end through which users interact with the system. It allows users to define queries, set parameters, and interpret results.

↳ Decision Support System (DSS)

The decision support system integrates data mining results into the decision-making process, providing support for strategic and operational decisions.

b. Mention data mining functionality, classification, prediction, clustering and evolution.[8]

→ * Data Mining Functionality

Its mainly two types:-

#Descriptive Data Mining

It includes certain knowledge to understand what is happing within the data without a previous idea. The common data features ore highlighted in the data set. For example: count, average etc.

→ Class or Concept Description

Class or concept refers to the data that is linked or correlated with some classes or some concepts.

→ Mining of Frequent Patterns

They can defined as the patterns that takes place very aften in transactional data.

→ Mining of Association

Mining of Association are mainly used in retail sales in order to identify patterns that are very aften purchased together

→ Mining of Correlation

Mining of Correlation refers to a type of Descriptive Data Mining's Functions that are usually executed in order to revel or expose some statistical correlation between associated attributes value pairs or between two item sets.

→ Mining of Clusters

The literal meaning of the word “Cluster” is a group in some of things which are similar to one another way another.

#Predictive Data Mining

It helps developers to provides unlabelled definitions of attributes. Based on previous tests. the software estimates the characteristics that ore absent. For example: Judging from the finding of a patient's medical examinations that is he suffering from any particular disease.

→ Decision Tree

A decision tree is a like a flow chart with o tree structure, in which every junction/node is used to represent a test on an attribute value, moreover, each and branch is responsible for representing the concluding outcome of the test, and tree leaves are used to represent the classes or the distribution of classes.

→ Neural Network

Neural network. a key component of predictive data mining, are a type machine learning algorithm inspired by the structure and functioning of the human brain. They consist of interconnected nodes, or neurons, organized in layers (input, hidden, output) that process information and learn patterns from data.

* Classification

Classification is a supervised learning task where the algorithm learns a mapping function from input features to predefined output labels based on a labelled training dataset.

↳ *Process*

The algorithm is trained on a labeled dataset, and the learned model is then used to predict the class labels of new, unseen instances.

↳ *Applications*

Common applications include spam detection, image recognition, and medical diagnosis.

* Prediction

Prediction, also known as regression, involves estimating a continuous output variable based on input features. It models the relationship between variables.

↳ *Process*

The algorithm learns from historical data to make predictions about numerical values for new instances.

↳ *Applications*

Used in financial forecasting, stock price prediction, and demand forecasting.

* Clustering

Clustering is an unsupervised learning task that groups similar data points together based on their inherent similarities.

↳ *Process*

Algorithms identify clusters without predefined class labels, aiming to discover natural structures within the data.

↳ *Applications*

Commonly used in customer segmentation, image segmentation, and anomaly detection.

* Evolution (Time Series Analysis)

Evolution or time series analysis involves examining data over time to identify patterns, trends, and changes in behaviour.

↳ *Process:*

Time-dependent data is analysed to understand how variables evolve and predict future states.

↳ *Applications*

Used in financial market analysis, weather forecasting, and monitoring system performance over time.

2a. How does a snowflake schema differ from a star schema? List any two advantages and disadvantages of snowflake schema. [5+3]

S.N	Star Schema	Snowflake Schema
1	In star schema, The fact tables and the dimension tables are contained.	While in snowflake schema, The fact tables, dimension tables as well as sub dimension tables are contained.
2	Star schema is a top-down model.	While it is a bottom-up model.
3	Star schema uses more space.	While it uses less space.
4	It takes less time for the execution of queries.	While it takes more time than star schema for the execution of queries.
5	In star schema, Normalization is not used.	While in this, Both normalization and denormalization are used.
6	It's design is very simple.	While it's design is complex.
7	The query complexity of star schema is low.	While the query complexity of snowflake schema is higher than star schema.
8	It's understanding is very simple.	While it's understanding is difficult.
9	It has less number of foreign keys.	While it has more number of foreign keys.
10	It has high data redundancy.	While it has low data redundancy.

* Advantages and Disadvantages of Snowflake Schema

↳ Advantage

Normalized Structure: Enhances data integrity by reducing redundancy.

Improved Storage Efficiency: Promotes compact data representation.

↳ Disadvantages

Query Performance Impact: May experience slower query performance.

Increased Complexity: More complex to design and maintain.

b. Explain concept of time series data and analysis.[8]

→ * Concept of Time Series Data and Analysis

A time series is a sequence of data points that occur in successive order over some period of time. This can be contrasted with cross-sectional data, which captures a point-in-time.

A time series can be taken on any variable that changes over time. In investing, it is common to use a time series to track the price of a security over time. This can be tracked over the short term, such as the price of a security on the hour over the course of a business day, or the long term, such as the price of a security at close on the last day of every month over the course of five years.

Time series analysis can be useful to see how a given asset, security, or economic variable changes over time. It can also be used to examine how the changes associated with the chosen data point compare to shifts in other variables over the same time period.

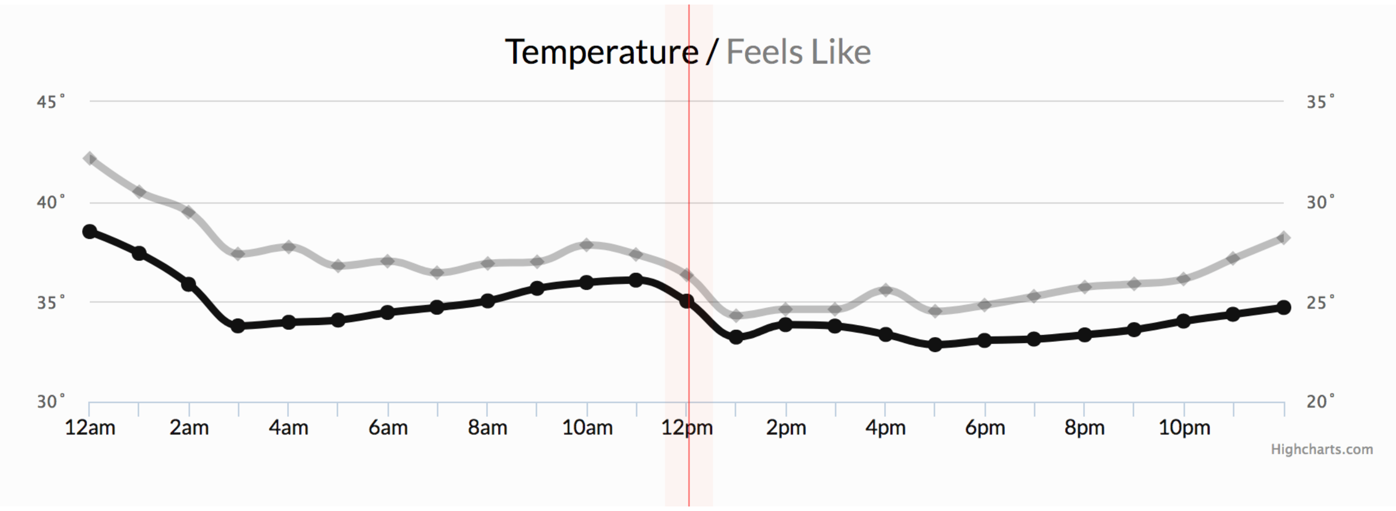
Time series forecasting uses information regarding historical values and associated patterns to predict future activity. Most often, this relates to trend analysis, cyclical fluctuation analysis, and issues of seasonality.

A time series can be constructed by any data that is measured over time at evenly-spaced intervals. Historical stock prices, earnings, GDP, or other sequences of financial or economic data can be analysed as a time series.

Examples of time series analysis in action include:

- Weather data
- Rainfall measurements
- Temperature readings
- Heart rate monitoring (EKG)
- Brain monitoring (EEG)
- Quarterly sales

- Stock prices
- Automated stock trading
- Industry forecasts
- Interest rates



Another familiar example of time series data is patient health monitoring, such as in an electrocardiogram (ECG), which monitors the heart's activity to show whether it is working normally.

* Models of time series analysis include

- ✓ Classification: Identifies and assigns categories to the data.
- ✓ Curve fitting: Plots the data along a curve to study the relationships of variables within the data.
- ✓ Descriptive analysis: Identifies patterns in time series data, like trends, cycles, or seasonal variation.
- ✓ Explanative analysis: Attempts to understand the data and the relationships within it, as well as cause and effect.
- ✓ Exploratory analysis: Highlights the main characteristics of the time series data, usually in a visual format.
- ✓ Forecasting: Predicts future data. This type is based on historical trends. It uses the historical data as a model for future data, predicting scenarios that could happen along future plot points.
- ✓ Intervention analysis: Studies how an event can change the data.
- ✓ Segmentation: Splits the data into segments to show the underlying properties of the source information.

3a. Explain term data cleaning, data integration and data transformation.[8]

→ * Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabelled. There is no one absolute way to prescribe the exact steps in the data cleaning process because the processes will vary from dataset to dataset.

Advantage

- Removal of errors when multiple sources of data are at play.
- Fewer errors make for happier clients and less-frustrated employees.
- Ability to map the different functions and what your data is intended to do.
- Monitoring errors and better reporting to see where errors are coming from, making it easier to fix incorrect or corrupt data for future applications.
- Using tools for data cleaning will make for more efficient business practices and quicker decision-making.

* Data Integration

Data integration is the process of combining data from multiple sources to provide a unified, coherent, and comprehensive view. It involves consolidating data from different systems, formats, or databases into a single, centralized repository.

Advantage

1. *Unified View:*
    - Data integration creates a unified view of information, allowing users to access and analyze data from various sources seamlessly.
2. *Improved Decision-Making:*
    - Integration enables more informed decision-making by providing a holistic understanding of the organization's data landscape.
3. *Enhanced Data Quality:*
    - Centralized integration can improve data quality through standardization, reducing errors and inconsistencies across disparate datasets.

4. *Efficiency in Analysis:*

    - Integrated data facilitates more efficient analysis and reporting, eliminating the need to switch between multiple systems or formats.

5. *Business Process Optimization:*

    - Integration supports business process optimization by ensuring that relevant data is available when and where it's needed.

* Data Transformation

Data transformation involves converting and reshaping data to meet the requirements of the target system or analysis. It includes processes like normalization, aggregation, encoding, and filtering.

Advantage

1. *Improved Data Suitability:*

    - *Transformation ensures that raw data is prepared and formatted to align with the specific requirements of the analytical methods or models, enhancing its suitability for analysis.*

2. *Enhanced Data Consistency:*

    - Through processes like normalization and encoding, data transformation promotes consistency by standardizing formats, units, and representations.

3. *Facilitates Effective Analysis:*

    - Transformed data is more amenable to analysis, allowing for better insights, pattern recognition, and trend identification.

4. *Supports Integration:*

    - Data transformation plays a key role in supporting the integration process by aligning diverse datasets with varying structures and formats.

5. *Optimizes Storage:*

    - *Transformation can optimize storage by reducing redundancy, aggregating data, and eliminating unnecessary details, making data more efficient to store and retrieve.*

b. What do you mean by DMQL? Explain DMQL with syntax and proper example.[2+6]

→ * **DMQL** (Data Mining Query Language)

DMQL (Data Mining Query Language) is a language designed for querying and extracting information from data mining models and systems. It provides a standardized way to interact with data mining models, allowing users to retrieve valuable insights, patterns, and knowledge from the underlying data.

* Syntax of DMQL

1. DMQL-Syntax for task-relevant data specification

Names of the relevant database or data warehouse, conditions, and relevant attributes or dimensions must be specified:

use database ‹database_name› or **use data warehouse** ‹data_warehouse_name›

from ‹relation(s)/cube(s)› [where condition] (data cubes and tables) in relevance to ‹attribute_or_dimension_list› (attributes or dimension for exploration)

order by ‹order_list› (sorting order)

group by ‹grouping_list› (specifies criteria to group)

having ‹condition› (it represent which group of data are considered relevant)

2. Syntax for specifying the kind of knowledge.

Syntax for Characterization, Discrimination, Association, Classification, and Prediction.

a. Data mining characterization

The syntax for characterization is −

mine characteristics [as pattern_name]

analyze {measure(s)}

b. Data mining discrimination

The syntax for Discrimination is −

mine comparison [as {pattern_name]}

For {target_class } where {t arget_condition }

{versus {contrast_class_i }

where {contrast_condition_i}}

analyze {measure(s) }

c. Data mining association

The syntax for Association is−

```
mine associations [ as {pattern_name} ]
{matching {metapattern} }
```

d. Data mining classification

The syntax for Classification is −

```
mine classification [as pattern_name]
analyze classifying_attribute_or_dimension
```

e. Data mining prediction

The syntax for prediction is −

```
mine prediction [as pattern_name]
analyze prediction_attribute_or_dimension
{set {attribute_or_dimension_i= value_i}}
```

3. Syntax for Concept Hierarchy Specification

To specify concept hierarchies, use the following syntax −

```
use hierarchy <hierarchy> for <attribute_or_dimension>
```

For Example −

```
with support threshold = 0.05
with confidence threshold = 0.7
```

4. Syntax for Pattern Presentation and Visualization Specification

We have a syntax, which allows users to specify the display of discovered patterns in one or more forms.

```
display as <result_form>
```

For Example −

```
display as table
```

* Examples of DMQL

Example 1: where 'numeric_field' is equal to 2

DMQL query: (numeric_field=2)

Equivalent SQL server query: WHERE numeric_field = 2

Example 2: where 'numeric_field' is greater than or equal to 3

DMQL query: (numeric_field=3+)

Equivalent SQL server query: WHERE numeric_field >= 3

Example 3: where 'numeric_field' is less than or equal to 8

DMQL query:(numeric_field=8-)

Equivalent SQL server query: WHERE numeric_field

Example 4: where 'numeric_field' is between 5 and 9 (including 5 and 9)

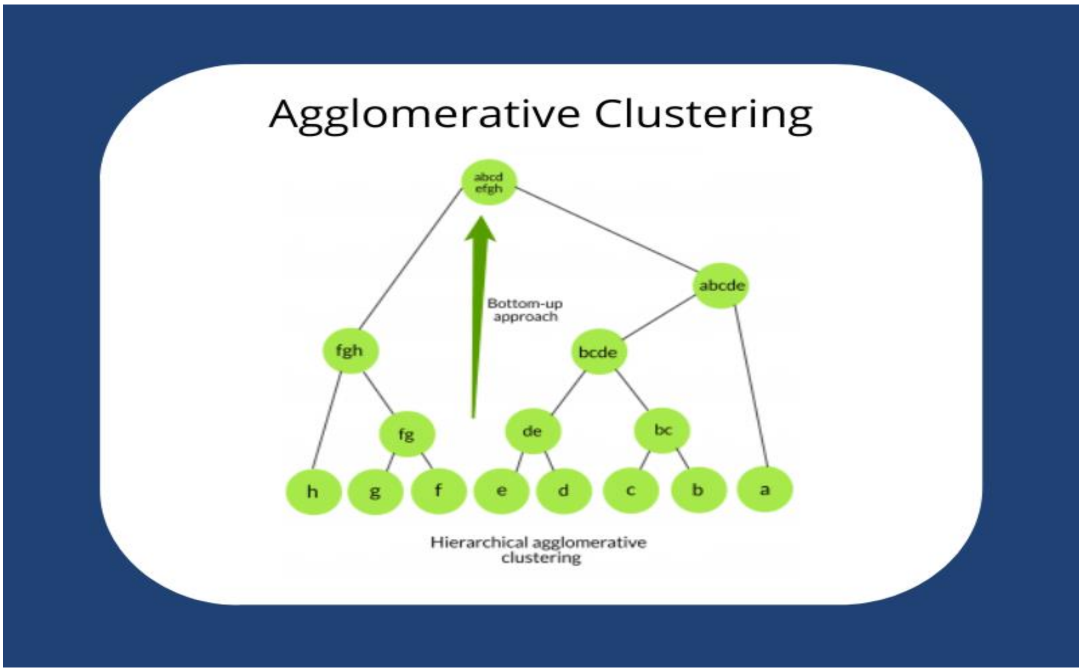
DMQL query:(numeric_field=5-9)

Equivalent SQL server query: WHERE numeric_field between 5 and 9

4a. What is agglomerative clustering? Explain concept of divisive clustering.[4+4]

→ * Agglomerative Clustering

Agglomerative clustering is a hierarchical clustering algorithm that builds a tree of clusters. The algorithm starts by treating each data point as a single cluster and iteratively merges the closest pairs of clusters until only a single cluster remains. This process results in a dendrogram, which is a tree-like structure that visually represents the hierarchy of clusters. Agglomerative clustering is a "bottom-up" approach, where the algorithm starts with individual data points and successively merges them into larger clusters.



Steps of Agglomerative Clustering:

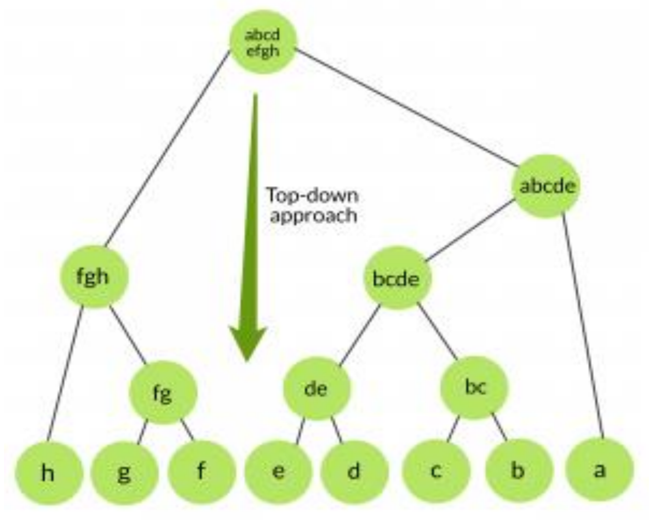
1. ***Initialization:***
   - Treat each data point as a single cluster.

2. ***Calculate Pairwise Distances:***
   - Compute the pairwise distances between all clusters.

3. ***Merge Closest Clusters:***
   - Find the two clusters with the smallest distance between them and merge them into a new cluster.
   - Update the distance matrix to reflect the distances between the new cluster and the remaining clusters.
4. ***Repeat:***
   - Repeat steps 2 and 3 until only one cluster remains.

* Divisive Clustering

Divisive clustering, also known as "top-down" clustering, is a hierarchical clustering technique where the process starts with a single cluster containing all data points and recursively divides it into smaller clusters. The goal is to create a hierarchy of clusters, ultimately resulting in a tree-like structure known as a dendrogram.



Hierarchical divisive clustering

Steps of Divisive Clustering:

1. *Initialization:*
   - *Begin with a single cluster that includes all data points.*
2. *Dissimilarity Measurement:*
   - Evaluate the dissimilarity (distance) between data points within the cluster.
3. Divisive Step:
   - Identify the pair of data points or subclusters with the highest dissimilarity.
   - Split this pair into two distinct clusters.
4. Recursion:
   - Repeat the dissimilarity measurement and divisive steps recursively for each new cluster until a stopping criterion is met. This criterion could be a predetermined number of clusters or a threshold dissimilarity level.
5. *Dendrogram Construction:*
   - As the divisive process proceeds, a dendrogram is constructed, illustrating the hierarchical relationships among clusters. The vertical lines in the dendrogram represent the division points, and the horizontal lines represent the merging of clusters.
6. *Cluster Identification:*
   - Determine the final clusters based on the desired number of clusters or the chosen dissimilarity threshold.

5a. Cluster the following instances of given data with the help of K-mean algorithm (K=2).

Instances	X	Y
1	1.0	2.5
2	1.0	4.5
3	2.5	3.0
4	2.0	1.5
5	4.5	1.5
6	4.0	5.0

→ * To cluster the given instances using the K-means algorithm with �=2*K*=2, we'll follow these steps:

1. **Initialize centroids: Randomly choose two initial centroids.**
2. **Assign instances to clusters: Assign each instance to the cluster whose centroid is the closest.**
3. **Update centroids: Recalculate the centroids based on the instances in each cluster.**
4. **Repeat steps 2 and 3: Repeat the assignment and update steps until convergence (centroids no longer change significantly or a set number of iterations is reached).**

Let's perform these steps:

Given Instances:

Copy code

Instances X Y 1 1.0 2.5 2 1.0 4.5 3 2.5 3.0 4 2.0 1.5 5 4.5 1.5 6 4.0 5.0

Step 1: Initialize Centroids (Randomly): Let's assume initial centroids:

- **Centroid 1: (1.0, 2.5)**
- **Centroid 2: (4.5, 1.5)**

Step 2 and 3: Assign Instances to Clusters and Update Centroids: Now, we'll iterate between assigning instances to clusters and updating centroids until convergence.

Iteration 1:

- **Assign instances to clusters based on the nearest centroid.**
- **Update centroids based on instances in each cluster.**

Instance	X	Y	Distance to Centroid 1	Distance to Centroid 2	Assigned Cluster
1	1.0	2.5	0.0	4.924	1

Instance	X	Y	Distance to Centroid 1	Distance to Centroid 2	Assigned Cluster
2	1.0	4.5	2.0	3.606	1
3	2.5	3.0	1.802	3.354	1
4	2.0	1.5	1.802	3.162	1
5	4.5	1.5	4.924	0.0	2
6	4.0	5.0	4.123	4.301	1

- **Updated Centroid 1: (1.5, 3.0)**
- **Updated Centroid 2: (4.25, 2.25)**

Iteration 2:

- **Repeat the process with the updated centroids.**

Instance	X	Y	Distance to Centroid 1	Distance to Centroid 2	Assigned Cluster
1	1.0	2.5	0.707	4.745	1
2	1.0	4.5	2.121	3.535	1
3	2.5	3.0	0.354	3.536	1
4	2.0	1.5	1.5	3.354	1
5	4.5	1.5	4.95	0.354	2
6	4.0	5.0	3.162	3.201	2

- **Updated Centroid 1: (1.625, 3.125)**
- **Updated Centroid 2: (4.25, 2.25)**

b. Explain FP-growth algorithm with the properties.

→ * FP-Growth Algorithm

The FP-Growth (Frequent Pattern Growth) algorithm is a popular and efficient algorithm for mining frequent item sets in a transactional database. It's commonly used for association rule mining in data mining and market basket analysis. Developed by Han, Pei, and Yin in 2000, FP-Growth is known for its ability to handle large datasets and produce compact data structures for efficient mining.

*** Key Steps of FP-Growth:**

1. *Create a Frequency Table:*
   Scan the database to count the frequency of each item (single items are called 1-itemsets).

2. *Build the FP-Tree:*
   Create a data structure called the FP-Tree by inserting transactions into a tree structure based on the frequency of items.

3. *Generate Conditional Pattern Bases:*
   For each frequent item, create a conditional pattern base by extracting the paths in the FP-Tree that correspond to that item.

4. *Recursive Mining:*
   Recursively apply the FP-Growth algorithm on each conditional pattern base to generate frequent patterns.

5. *Combine Patterns:*
   Combine the patterns obtained from the recursive mining step to produce the final set of frequent item sets.

* Advantages of FP-Growth:

1. *Efficiency:*
   FP-Growth is more efficient than Apriori, another frequent itemset mining algorithm, especially for large datasets, as it requires only two passes over the data.

2. *Compact Data Structure:*
   The FP-Tree data structure is more compact compared to the frequent itemset generation and storage in Apriori.

3. *Reduced Candidate Generation:*
   FP-Growth eliminates the need to generate and test candidate item sets, reducing computational overhead.

4. *Handles Sparse Data:*
   It performs well even with sparse datasets where the traditional Apriorism algorithm may struggle.

* Properties of FP-Growth Algorithm:

1. Compact Data Representation:
   The FP-tree compactly represents the dataset, reducing the memory requirements compared to traditional candidate generation approaches.

2. Efficient Pattern Mining:
   FP-growth is more efficient than Apriori, especially when dealing with large datasets, as it eliminates the need to generate and test candidate itemsets.

3. No Candidate Generation:
   Unlike the Apriori algorithm, FP-growth does not generate candidate itemsets explicitly. It uses a divide-and-conquer strategy to mine frequent itemsets directly from the FP-tree.

4. Conditional Pattern Bases:
   FP-growth uses conditional pattern bases to efficiently mine frequent itemsets. These are sub-datasets associated with each frequent item, facilitating the recursive mining process.

5. FP-Tree Structure:
   The FP-tree is a tree structure that represents the relationships between frequent items in the dataset. It helps streamline the mining process by efficiently capturing the frequency information.

6. Header Table:
   The algorithm uses a header table associated with the FP-tree to link occurrences of the same item and facilitate the extraction of frequent itemsets.

7. Recursive Mining:
   FP-growth employs a recursive approach to mine frequent itemsets. The process involves constructing conditional FP-trees for each frequent item and extracting patterns from these conditional trees.

6. Write shorts notes on:

a. Advanced Data Mining

Advanced data mining refers to the use of sophisticated techniques and algorithms to discover patterns, trends, and insights in large datasets. It goes beyond basic data mining methods to address complex challenges and make more accurate predictions.

*** Its Important Features:**

↳ *Feature Engineering:*

Creation and selection of relevant features to enhance model accuracy by highlighting key information and reducing noise.

↳ *Ensemble Learning:*

Combination of multiple models to improve predictive performance, reducing overfitting and increasing model robustness.

↳ *Deep Learning:*

Use of neural networks with multiple layers to automatically learn intricate patterns and representations from complex data.

↳ *Text and Sentiment Analysis:*

Analysis of unstructured text data for sentiment analysis, topic modeling, and information extraction.

↳ *Temporal and Spatial Data Mining:*

Analysis of data over time or space to identify trends, patterns, and correlations specific to those dimensions.

↳ *Anomaly Detection:*

Identification of anomalies or outliers in data, crucial for detecting fraud, errors, or unusual patterns.

↳ *Clustering and Segmentation:*

Utilization of advanced clustering techniques to group similar data points together, providing insights into the inherent structure of the data.

↳ *Interactive Data Exploration:*

Inclusion of interactive interfaces that allow users to dynamically explore and visualize data, facilitating a deeper understanding of patterns and relationships.

↳ *Scalability:*

Design of algorithms and tools to handle large-scale datasets efficiently, often leveraging parallel and distributed computing.

↳ *Ethical and Responsible Data Mining:*

Consideration of ethical considerations and responsible AI practices, including fairness, transparency, and bias mitigation.

b. DBMS Vs Data Warehouse

* DBMS

A database is a collection of related data which represents some elements of the real world. It is designed to be built and populated with data for a specific task. It is also a building block of your data solution.

* Data Warehouse

A data warehouse is an information system which stores historical and commutative data from single or multiple sources. It is designed to analyze, report, integrate transaction data from different sources.

DBMS	Data Warehouse
It supports operational processes.	It supports analysis and performance reporting.
Capture and maintain the data.	Explore the data.
Current data.	Multiple years of history.
Data is balanced within the scope of this one system.	Data must be integrated and balanced from multiple system.
Data is updated when transaction occurs.	Data is updated on scheduled processes.
Data verification occurs when entry is done.	Data verification occurs after the fact.
100 MB to GB.	100 GB to TB.
ER based.	Star/Snowflake.
Application oriented.	Subject oriented.
Primitive and highly detailed.	Summarized and consolidated.
Flat relational.	Multidimensional.
Follows the ACID properties (Atomicity, Consistency, Isolation, Durability).	Emphasizes read performance over write performance.
Handles moderate to high volumes of current transactional data.	Manages large volumes of historical data for analysis.
Prioritizes fast write operations.	Aggregations and optimizations are tailored for analytical queries.
Primarily focuses on OLTP (Online Transaction Processing) operations.	Primarily focuses on OLAP (Online Analytical Processing) operations.
Supports low-latency operations for real-time data needs.	Tolerates higher latency for batch processing and reporting.
Accessed by applications for real-time data needs.	Supports reporting and analysis for strategic decision support.

c. Application of Data Mining

**** The important application of data Mining:***

→ *Marketing*

Targeted advertising, customer segmentation, and personalized recommendations based on Purchasing behaviour.

→*Health care*

Predictive analytics for disease diagnosis. treatment optimization, and patient outcome prediction

→ *Finance*

Frond detection, risk assessment, and stock market analysis for investment decision

→ *Retail*

Inventory management, market basket analysis, demand forecasting.

→*Telecommunications*

Customer Churn prediction, network optimization and Improving service quality

→ *Manufacturing*

Quality control predictive maintenance, and supply chain optimization

→ *Science and Research*

Dang discovery, genomic analysis, and environmental research.

→ *Education*

Analyze academic data to identify factors influencing student performance and implement interventions.

→ *Transportation and Logistics*

Optimize delivery routes and schedules based on historical traffic and demand patterns.

d. KDD

* KDD (Knowledge Discovery in Databases)

Knowledge Discovery in Databases (KDD) is a process of extracting useful knowledge or patterns from large volumes of data. KDD involves various steps, and it is often considered synonymous with the broader process of data mining. Here are the key steps involved in the KDD process:

1. *Data Selection:*

    - Identify and select relevant data from various sources, which may include databases, data warehouses, and external data repositories.

2. *Data Preprocessing:*

    - Cleanse and preprocess the selected data to handle missing values, outliers, and inconsistencies. This step may also involve data transformation and normalization.

3. *Data Transformation:*

    - Convert the preprocessed data into a suitable format for analysis. This may include aggregating data, creating new features, or encoding categorical variables.

4. *Data Mining:*

    - Apply data mining techniques to extract patterns, trends, and knowledge from the prepared dataset. Common data mining methods include classification, regression, clustering, and association rule mining.

5. *Pattern Evaluation:*

    - Evaluate the patterns or models generated by the data mining algorithms. Assess the significance and quality of the discovered knowledge against predefined criteria.

6. *Knowledge Representation:*

    - Represent the discovered knowledge in a form that is understandable and usable for decision-making. This may involve visualization, summary statistics, or other representation techniques.

7. *Knowledge Interpretation:*

    - Interpret the knowledge in the context of the problem domain. Understand the implications of the discovered patterns and assess their practical value.

8. *Use of Knowledge:*

    - Apply the discovered knowledge to address specific business problems or make informed decisions. The ultimate goal is to add value to the organization based on the insights gained from the data.

Data Mining

2077

Model Questions

Attempt Any Eight Questions.

1. Explain data mining? Describe the importance of the data mining for social network.[4+6]

→ * Data Mining

Data Mining is the process of analysing large sets of data to discover patterns, correlations, or relationships that can be useful for making decisions or predictions. It involves various techniques from statistics, machine learning, and database system to extract insights and valuable information from row data.

Its Functionalities

Data Mining Functionalities are can be divided into 2 categories:

↳ ***Descriptive Data Mining***

It includes certain knowledge to understand what is happing within the data without a previous idea. The common data features ore highlighted in the data set. For example: count, average etc.

↳ ***Predictive Data Mining***

It helps developers to provides unlabelled definitions of attributes. Based on previous tests. the software estimates the characteristics that ore absent. For example: Judging from the finding of a patient's medical examinations that is he suffering from any particular disease.

* The Importance of the data mining for social network

1. *User Behavior Analysis:*

    - Data mining enables the analysis of user behavior on social networks. Patterns such as posting frequency, content preferences, and interaction trends can be extracted, providing valuable insights for platform optimization.

2. *Personalized Content Recommendations:*

    - By analyzing user interactions and preferences, data mining allows social networks to deliver personalized content recommendations. This enhances user engagement and satisfaction.

3. *Targeted Advertising:*

    - Social networks leverage data mining to analyze user demographics, interests, and online behavior. This information is used to deliver targeted advertisements, improving the effectiveness of advertising campaigns.

4. *Community Detection:*

    - Data mining helps identify communities and groups within a social network based on user interactions and common interests. This information can be valuable for fostering community engagement and enhancing user experience.

5. *Anomaly Detection:*

    - Detecting unusual or anomalous behavior on social networks is crucial for maintaining security and preventing fraudulent activities. Data mining techniques can identify patterns deviating from the norm.

6. *Sentiment Analysis:*

    - By analyzing user-generated content, such as posts and comments, data mining facilitates sentiment analysis. Understanding the sentiment of users helps in gauging public opinion and managing brand reputation.

7. *Predictive Modeling:*

    - Data mining enables social networks to build predictive models for user behavior, allowing them to anticipate trends, forecast engagement, and make data-driven decisions for platform improvements.

8. *Fraud Detection:*

    - Social networks utilize data mining to detect and prevent fraudulent activities, including fake accounts, spam, and phishing attempts. This contributes to a safer and more trustworthy online environment.

9. *Enhanced User Experience:*

    - By applying data mining insights, social networks can continuously enhance the overall user experience. Understanding user preferences and adapting features accordingly contributes to user satisfaction and retention.

10. *Research and Academic Insights:*

- Social networks provide rich datasets for research purposes. Data mining enables researchers to gain insights into social dynamics, information diffusion, and the impact of online interactions on society.

2. Write short notes on data warehousing? Describe the Architecture of data warehouse.[4+6]

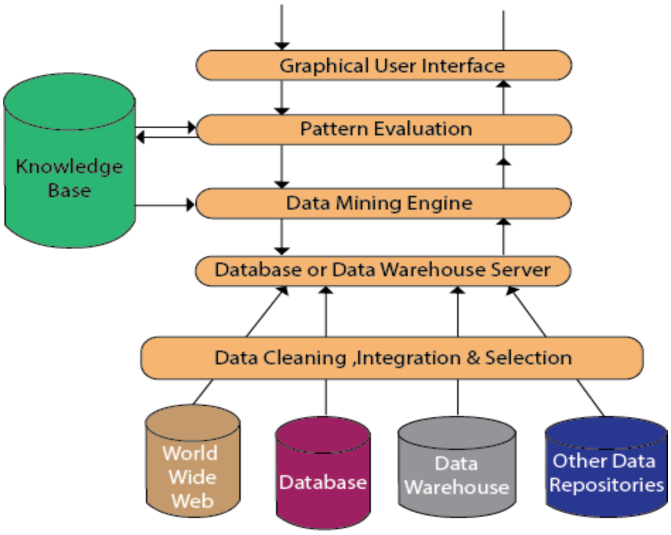
→ * Data Warehouse

A data warehouse is a centralized repository that stores large volumes of structured and often, historical data from various sources within an organization. It is designed for efficient querying, reporting, and analysis to support decision-making processes. Data warehouses provide a consolidated and integrated view of data from different departments and systems, making it easier for users to derive insights and make informed business decisions.

Data warehousing provides a consolidated view of an organization's data, allowing users to perform complex queries, generate reports, and gain insights into trends and patterns. Key components include the Extract, Transform, Load (ETL) process, a data warehouse schema, and OLAP (Online Analytical Processing) tools for multidimensional analysis.

* The Architecture of Data Warehouse

Mining is the process of analysing large sets of data to discover patterns, correlations, or relationships that can be useful for making decisions or predictions. It involves various techniques from statistics, machine learning, and database system to extract insights and valuable information from row data.



↳ 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.

↳ 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.

↳ 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.

↳ 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 stores the patterns, rules, and models discovered during the data mining process. It acts as a repository for actionable knowledge.

↳ User Interface

The user interface is the front-end through which users interact with the system. It allows users to define queries, set parameters, and interpret results.

↳ Decision Support System (DSS)

The decision support system integrates data mining results into the decision-making process, providing support for strategic and operational decisions.

3. Explain the star schema, fact tables, dimension tables and dimension hierarchies of the data warehouse with example.[10]

→ * Star Schema

A star schema is a type of data warehouse schema that consists of one or more fact tables referencing multiple dimension tables. It is called star schema because the structure resembles a star, with the fact table at the centre and dimension tables surrounding it.

An example of a star schema is given below:

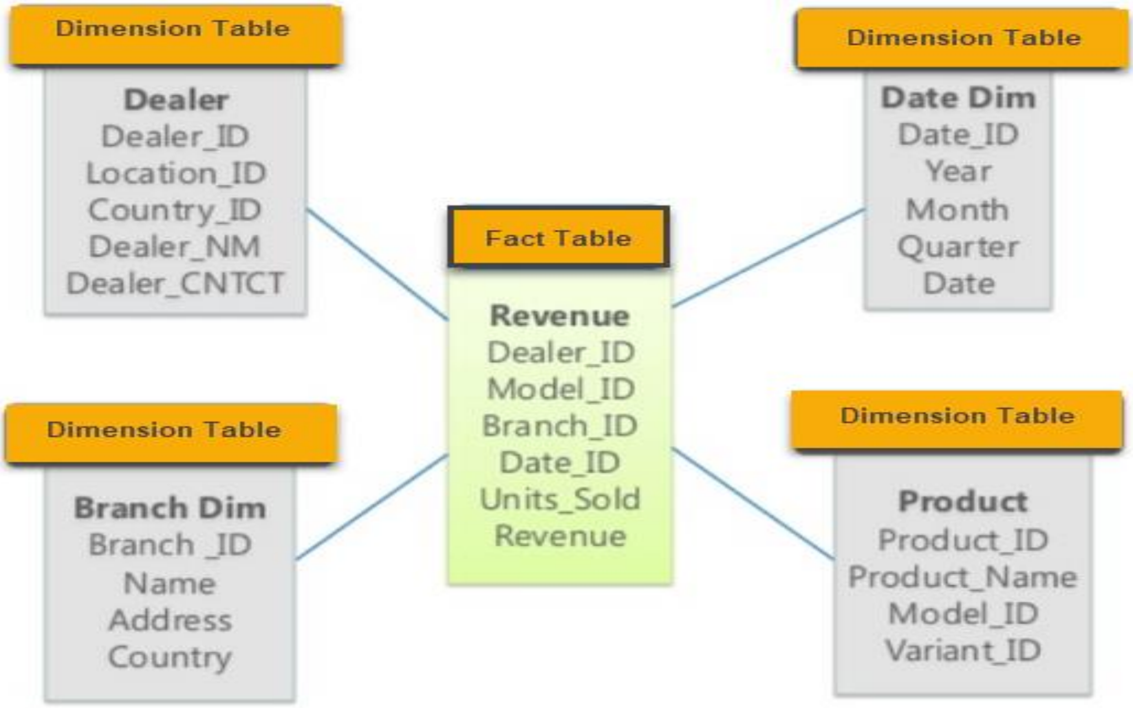


Fig:- Star Schema

* Fact Tables

A fact table is the central table in a star schema that stores quantitative information (facts) about a business process. It typically contains numeric or additive data, such as sales revenue, quantity sold, or profit.

For Example:

Date_key	Product_key	Customer_key	Sales_key
2022-10-12	201	101	4000
2023-02-13	202	102	5000
2023-12-14	203	103	3500

*** Dimension Tables**

Dimension tables are auxiliary tables in a star schema that provide descriptive information about the business entities related to the facts in the fact table. They contain attributes or descriptive information that is used for filtering, grouping, and labelling the data.

For example:

Product_key	Product_Name	Catogory
101	Product_A	Electronics
102	Product_B	Apparel
103	Product_C	Vegitables

*** Dimension Hierarchies**

Dimension hierarchies represent the organized structure of attributes in a dimension table. Each level in the hierarchy provides a different level of granularity for analysis.

Example Dimension Hierarchy (Date):

- Date Hierarchy: Year > Quarter > Month > Day
- For example, the date '2022-01-01' is part of the hierarchy as follows: 2022 > Q1 > January > 01.

Example Dimension Hierarchy (Product):

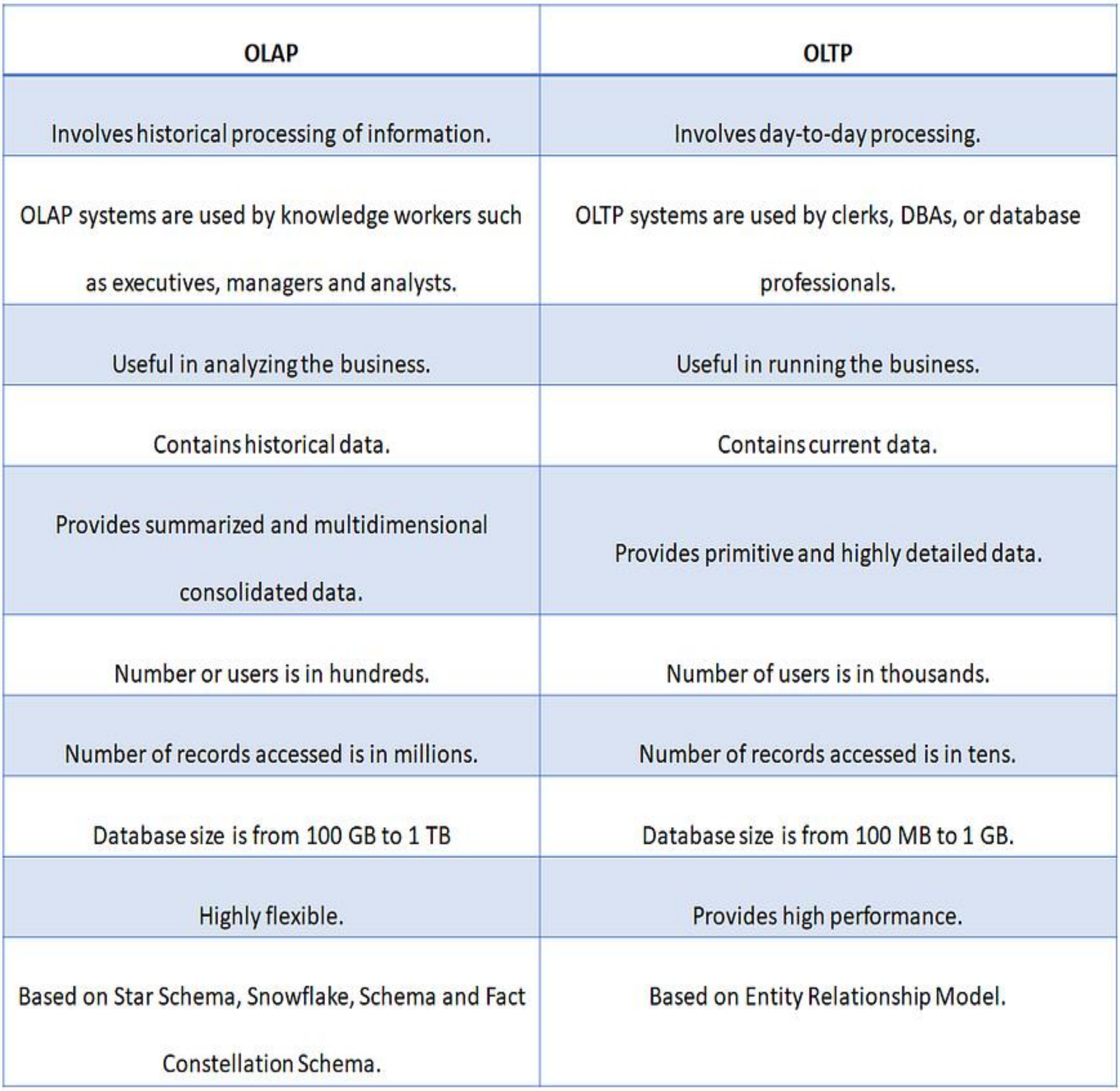
- Product Hierarchy: Category > Subcategory > Product
- For example, the product 'Product_A' in the 'Electronics' category is part of the hierarchy as follows: Electronics > [Subcategory] > Product_A.

Example Dimension Hierarchy (Customer):

- Customer Hierarchy: Region > City > Customer
- For example, the customer 'Customer_X' in the 'North' region is part of the hierarchy as follows: North > [City] > Customer_X.

4. Differentiate between OLAP and OLTP. Explain the drill up and slice operation in the data warehouse.[5+5]

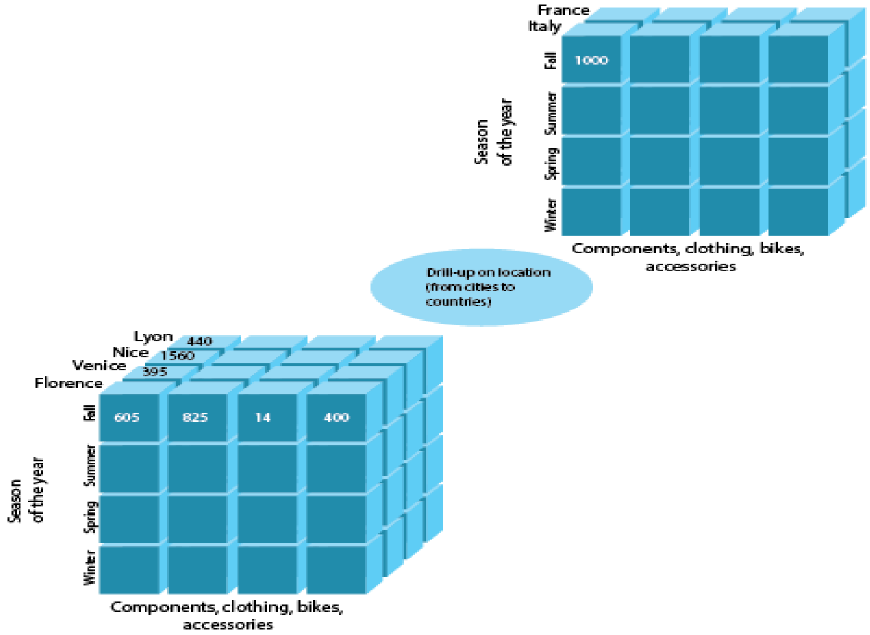
→ * Differentiate between OLAP and OLTP



OLAP	OLTP
Involves historical processing of information.	Involves day-to-day processing.
OLAP systems are used by knowledge workers such as executives, managers and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
Useful in analyzing the business.	Useful in running the business.
Contains historical data.	Contains current data.
Provides summarized and multidimensional consolidated data.	Provides primitive and highly detailed data.
Number or users is in hundreds.	Number of users is in thousands.
Number of records accessed is in millions.	Number of records accessed is in tens.
Database size is from 100 GB to 1 TB	Database size is from 100 MB to 1 GB.
Highly flexible.	Provides high performance.
Based on Star Schema, Snowflake, Schema and Fact Constellation Schema.	Based on Entity Relationship Model.

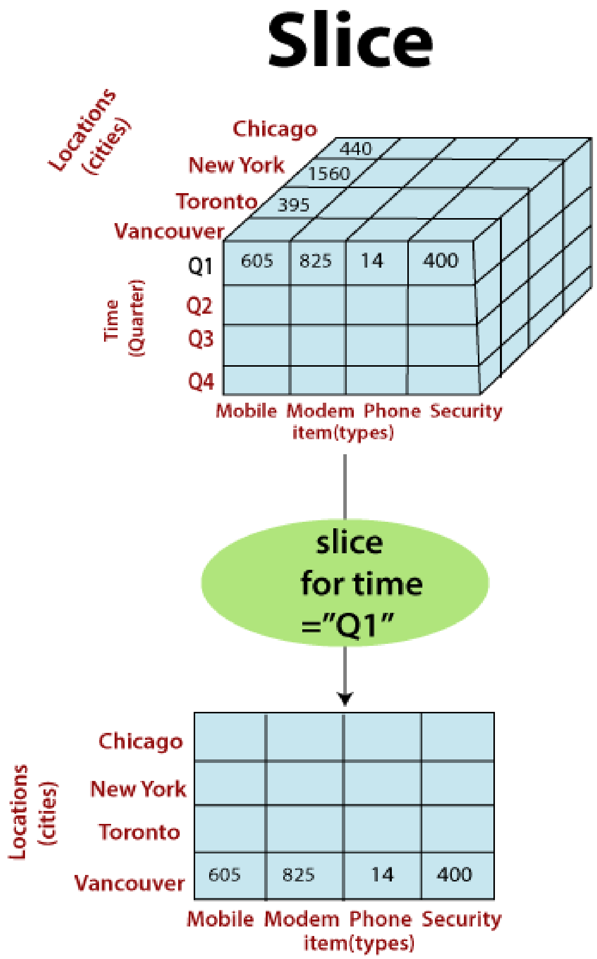
* Drill Up

Drill Up is the reverse of drill down. It involves moving from a lower level of detail to a higher, more aggregated level. It summarizes detailed data to a higher level of abstraction.



* Slice Operation

Slicing is a technique in multidimensional data analysis where a specific "slice" or subset of a data cube is extracted by fixing one or more dimensions at particular values while keeping others unrestricted.



5. Explain the Partitioning, parallelism and compression in data warehouse.[10]

→ * Partitioning

Partitioning is a database design technique used in data warehousing to divide large tables into smaller, more manageable pieces called partitions. Each partition contains a subset of the data based on a defined criteria, such as a range of values in a specific column.

Purpose:

1. **Query Performance:** Partitioning enhances query performance by allowing the database engine to perform operations only on the relevant partitions, reducing the amount of data that needs to be scanned.

2. **Parallelism:** Partitions can be processed in parallel, distributing the workload across multiple processors or nodes, further improving query performance.

3. **Data Management:** Partitioning facilitates efficient data loading and maintenance tasks. For example, loading data into a specific partition or removing data from a partition is more efficient than manipulating the entire table.

Example:

```
CREATE TABLE Fact_Sales
(
  Sales_Date DATE,
  Product_ID INT,
  Sales_Amount DECIMAL(10,2),
)
PARTITION BY RANGE (Sales_Date)(
  PARTITION Jan2022 VALUES LESS THAN (TO_DATE('2022-02-01', 'YYYY-MM-DD')),
  PARTITION Feb2022 VALUES LESS THAN (TO_DATE('2022-03-01', 'YYYY-MM-DD')),
);
```

* Parallelism

Parallelism in data warehousing refers to the ability to perform multiple operations simultaneously by distributing the workload across multiple processors, nodes, or servers.

Types of Parallelism:

1. **Data Parallelism:** Involves splitting a large task into smaller subtasks that can be executed concurrently on different processors, each working on a subset of the data.
2. **Task Parallelism:** Involves dividing a complex task into subtasks that can be executed concurrently. This is more about parallelizing operations rather than data.

Purpose:

1. **Improved Performance:** Parallel processing accelerates query performance by leveraging the capabilities of multiple processors to handle different parts of a task simultaneously.
2. **Scalability:** Parallelism enables systems to scale horizontally by adding more processors or nodes, allowing data warehouses to handle increasing volumes of data and user queries.
3. **Efficient Resource Utilization:** Parallel processing efficiently utilizes the available hardware resources, reducing query response times and optimizing overall system performance.

Examples:

```
SELECT /*+ PARALLEL(Fact_Sales, 4) */

   Product_ID, SUM(Sales_Amount)

FROM

   Fact_Sales

WHERE

   Sales_Date BETWEEN TO_DATE('2022-01-01', 'YYYY-MM-DD') AND TO_DATE('2022-01-31', 'YYYY-MM-DD')

GROUP BY

   Product_ID;
```

* Compression

Compression is the process of reducing the size of data stored in a data warehouse. It involves using algorithms and techniques to represent data in a more compact form without losing information.

Purpose:

1. **Storage Space Optimization:** Compressed data requires less storage space, which is crucial for large data warehouses dealing with massive volumes of data.
2. **Improved Query Performance:** Smaller data volumes mean that more data can fit into memory, leading to improved query performance as less data needs to be read from storage.
3. **Reduced I/O Overhead:** Compressed data reduces the I/O (input/output) overhead, resulting in faster data retrieval times and more efficient data processing.

Examples:

```
CREATE TABLE Compressed_Fact_Sales

AS

SELECT

  Product_ID,

  TO_COMPRESS(SUM(Sales_Amount)) AS Compressed_Sales_Amount,

FROM

  Fact_Sales

GROUP BY

  Product_ID;
```

6. Define data warehouse construction process. Explain the EPL process in data warehouse.[5+5]

→ * Data Warehouse Construction Process:

The construction of a data warehouse involves several key steps to design, build, and populate the warehouse with relevant data. The process can be outlined as follows:

1. ***Business Requirement Analysis:***

    - Understand and analyze the business requirements and objectives to determine the scope and goals of the data warehouse project.

2. ***Data Source Identification:***

    - Identify and select relevant data sources, which may include operational databases, external data feeds, spreadsheets, and other sources.

3. ***Data Modeling:***

    - Design the data model, including the definition of fact tables, dimension tables, and relationships between them. Choose an appropriate schema (e.g., star schema or snowflake schema).

4. ***ETL (Extract, Transform, Load):***

    - Develop ETL processes to extract data from source systems, transform it into the desired format, and load it into the data warehouse. This step may involve data cleansing, integration, and aggregation.

5. ***Data Storage Design:***

    - Determine the physical storage structures, indexing, and partitioning strategies to optimize data retrieval and query performance.

6. ***Metadata Management:***

    - Establish a metadata repository to store information about the data warehouse, including data definitions, transformations, and relationships.

7. ***Implementation of Security Measures:***

    - Implement security measures to ensure that data is accessed only by authorized users. This includes user authentication, role-based access control, and encryption.

8. ***Testing and Validation:***

    - Conduct testing to ensure the accuracy and reliability of data in the data warehouse. Perform validation against business rules and user requirements.

9. ***User Training:***

    - Provide training to end-users and stakeholders to familiarize them with the data warehouse environment, tools, and capabilities.

10. Deployment:

- Deploy the data warehouse for production use. Monitor and fine-tune performance as needed.

11. Maintenance and Evolution:

- Implement a maintenance plan to ensure ongoing data quality, refresh data regularly, and evolve the data warehouse based on changing business needs.

* ETL process

ETL stands for Extract, Transform, Load, and it is a crucial process in data warehousing that involves the movement and transformation of data from source systems to the data warehouse.

1. Extract:

Retrieve data from various source systems, which can include databases, applications, flat files, and external sources.

Methods:

- Full Extraction: Extract the entire dataset from source systems.
- Incremental Extraction: Extract only the new or changed data since the last extraction.
- Change Data Capture (CDC): Identify and extract only the data that has changed since the last extraction.

2. Transform:

Clean, modify, and structure the extracted data to make it suitable for analysis in the data warehouse.

Transformations:

- Cleaning and Validation: Remove errors, handle missing data, and validate data integrity.
- Aggregation: Summarize and aggregate data for higher-level analysis.
- Data Formatting: Convert data types, apply formatting rules, and standardize values.

- Enrichment: Add additional information or derived fields to enhance the dataset.

3. Load:

Load the transformed data into the data warehouse for storage and analysis.

Methods:

- Bulk Loading: Load data in large batches for efficiency.
- Incremental Loading: Add new or modified data to the data warehouse incrementally.
- Trickle Loading: Load data continuously in smaller increments to maintain near real-time updates.

Importance of ETL:

- **Data Integration:** ETL ensures the integration of data from diverse sources into a unified format in the data warehouse.
- **Consistency:** ETL processes enforce consistency in data structure and format across the data warehouse.
- **Data Quality:** ETL processes include data cleansing and validation, contributing to improved data quality in the data warehouse.
- **Performance:** ETL optimizations, such as indexing and partitioning, enhance query performance in the data warehouse.

7. Explain the Neural Network in Data Mining.[10]

→ * Neural Network

A neural network is a computational model inspired by the structure and functioning of the human brain. It is used in data mining and machine learning to identify patterns, make predictions, and perform various tasks based on input data. Neural networks consist of interconnected nodes (neurons) organized into layers, each layer having a specific function in the learning process.

*** Types of Neural Networks:**

1. *Feedforward Neural Networks (FNN):*
    - The simplest type, where information flows in one direction, from input to output.
2. *Recurrent Neural Networks (RNN):*
    - Allow information to be stored and reused, making them suitable for sequential data (e.g., time series).
3. *Convolutional Neural Networks (CNN):*
    - Designed for image processing and feature extraction, using convolutional layers to identify patterns in spatial data.
4. *Radial Basis Function Networks (RBFN):*
    - Use radial basis functions as activation functions, often applied to pattern recognition and classification tasks.

*** Key Components of a Neural Network:**

1. *Input Layer:*
   Receives the input data or features. Each node in the input layer represents a feature of the input data.

2. *Hidden Layers:*
   Intermediate layers between the input and output layers. These layers perform complex computations to learn patterns and relationships in the data.

3. *Output Layer:*
   Produces the final output or prediction. The number of nodes in the output layer depends on the type of task (e.g., binary classification, multiclass classification, regression).

4. *Neurons (Nodes):*
   Nodes in a neural network perform computations on input data using activation functions. The strength of connections (weights) between nodes is adjusted during training.

5. *Weights and Biases:*
   Neural networks learn from data by adjusting weights and biases associated with connections between nodes. These adjustments occur during a training process to minimize errors.

6. Activation Functions:
   Determine the output of a node based on its input. Common activation functions include sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU).

7. Learning Algorithm:
   The algorithm used to adjust weights and biases during training. Backpropagation is a widely used learning algorithm that minimizes the difference between predicted and actual outputs.

* Working of Neural Networks in Data Mining:

1. *Training:*

   - Neural networks are trained using labeled data, where the input features are associated with known output values. During training, the network adjusts weights and biases to minimize the difference between predicted and actual outputs.

2. *Forward Propagation:*

   - Input data is fed forward through the network layer by layer. Each node's output is calculated based on the input and weights, and activation functions determine the node's final output.

3. *Backpropagation:*

   - After forward propagation, the network compares the predicted output with the actual output. The error is then propagated backward through the network, and weights are adjusted using the gradient descent optimization algorithm.

4. *Iterative Learning:*

   - The training process is iterative, with multiple passes through the data. Each iteration refines the network's weights, improving its ability to make accurate predictions.

*** Applications of Neural Networks in Data Mining:**

1. *Classification:*
   - Neural networks are used for classification tasks, such as image recognition, spam detection, and disease diagnosis.
2. *Regression:*
   - Neural networks can predict numerical values, making them suitable for regression tasks like stock price prediction or demand forecasting.
3. *Pattern Recognition:*
   - Neural networks excel in recognizing complex patterns in data, making them valuable for tasks like handwriting recognition and speech analysis.
4. *Anomaly Detection:*
   - Neural networks can identify anomalies or outliers in data, aiding in fraud detection and network security.
5. *Natural Language Processing (NLP):*
   - Neural networks are applied in NLP tasks, including sentiment analysis, language translation, and chatbot development.
6. *Image and Speech Recognition:*
   - CNNs are particularly effective in image and speech recognition tasks, where they can learn hierarchical features from raw data.
7. *Recommender Systems:*
   - Neural networks contribute to building personalized recommender systems by learning user preferences and suggesting relevant items.

8. Explain the cluster analyses. Write down the steps to cluster the given data using k-means algorithms.[5+5]

→ * Cluster Analyses

Cluster analysis, also known as clustering, is a data mining technique that groups similar data points into clusters or segments based on certain criteria. The goal is to discover inherent patterns and structures in the data, where items within the same cluster are more similar to each other than to those in other clusters.

*** Key Concepts:**

1. *Clusters:*

   - Groups of data points that share similarities or patterns.

2. *Centroid:*

   - A representative point for a cluster, often the mean or median of the data points in the cluster.

3. *Similarity or Dissimilarity Measures:*

   - Methods to quantify the similarity or dissimilarity between data points, such as Euclidean distance or cosine similarity.

4. Hierarchical vs. Partitional:

   - Hierarchical clustering builds a hierarchy of clusters, while partitional clustering directly divides data into non-overlapping groups.

5. *K-Means vs. DBSCAN vs. Hierarchical Clustering:*

   - Different algorithms exist for cluster analysis, each with its strengths and weaknesses. K-means is a popular partitional method, while DBSCAN is a density-based method.

*** The steps to cluster the given data using k-means algorithms**

5. *Initialization:*

   - Choose the number of clusters (*k*).
   - Randomly initialize *k* cluster centroids, one for each cluster.

6. *Assignment:*

   - Assign each data point to the nearest cluster centroid.
   - Use a distance metric, commonly the Euclidean distance, to measure the distance between data points and centroids.

7. *Update Centroids:*

   - Recalculate the centroid of each cluster as the mean of all data points assigned to that cluster.

8. *Repeat:*

    - Repeat steps 2 and 3 until convergence.
    - Convergence occurs when the assignment of data points to clusters no longer changes significantly or when a specified number of iterations is reached.

Key Considerations:

5. *Choice of k:*

    - The number of clusters, *k*, must be specified before running the algorithm. Various methods, such as the elbow method or silhouette analysis, can be used to determine an optimal *k* value.

6. *Initialization Sensitivity:*

    - The algorithm's performance is sensitive to the initial placement of centroids. Different initializations may lead to different final clusters.

7. *Convergence:*

    - K-means aims to minimize the sum of squared distances within clusters, but it may converge to a local minimum. Multiple runs with different initializations can help mitigate this issue.

8. *Scalability:*

    - K-means can scale to large datasets, but for extremely large datasets, variants like Mini-Batch K-Means are often used.

9. Explain text mining and visual data mining.[5+5]

→ * Text Mining

Text mining, also known as text analytics, is a data mining technique that involves extracting meaningful patterns, information, and knowledge from unstructured text data. Unstructured text data can include documents, emails, social media posts, articles, and more.

Key Steps in Text Mining:

1. *Text Preprocessing:*

    - Clean and preprocess the raw text data by removing stop words, punctuation, and irrelevant characters.

2. *Tokenization:*

   - Break the text into smaller units, such as words or phrases (tokens), to facilitate analysis. This step helps convert the text into a structured format for further processing.

3. *Text Representation:*

   - Convert the text into numerical or vector representations. Common methods include bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), and word embeddings (e.g., Word2Vec, GloVe).

4. *Feature Extraction:*

   - Extract relevant features from the text data, such as keywords, entities, or sentiment scores. This step helps represent the content in a form suitable for analysis.

5. *Text Mining Algorithms:*

   - Apply various text mining algorithms for tasks like sentiment analysis, topic modeling, named entity recognition, and text categorization. Machine learning techniques, such as Naive Bayes, Support Vector Machines, and deep learning, are commonly used.

6. *Pattern Recognition:*

   - Identify patterns, trends, and insights within the text data. This may involve clustering similar documents, discovering topics, or predicting sentiment.

7. *Evaluation and Interpretation:*

   - Evaluate the performance of text mining models and interpret the results. Assess the accuracy of sentiment predictions, topic distributions, or any other task-specific metrics.

Applications of Text Mining:

1. *Sentiment Analysis:*

   - Determine the sentiment expressed in text data, such as positive, negative, or neutral sentiments. Useful for brand monitoring, customer feedback analysis, and social media monitoring.

2. *Topic Modeling:*

   - Discover latent topics within a collection of documents. This is valuable for organizing and summarizing large sets of documents.

3. *Named Entity Recognition (NER):*

    - Identify and classify entities, such as people, organizations, locations, and dates, mentioned in text data.

4. *Text Categorization:*

    - Classify documents into predefined categories or topics. Common in document classification, news categorization, and content tagging.

5. *Information Extraction:*

    - Extract specific information or facts from text data, such as extracting key events, relationships, or trends.

*** Visual Data Mining**

Visual data mining involves the use of graphical representations, visualization techniques, and interactive interfaces to explore and analyze large and complex datasets. It leverages the human ability to recognize patterns visually and enhances the understanding of data through visual exploration.

Key Concepts in Visual Data Mining:

1. *Visualization Techniques:*

    - Use various visualization methods, including charts, graphs, heatmaps, scatter plots, and network diagrams, to represent data patterns and relationships.

2. *Interactive Interfaces:*

    - Provide users with interactive interfaces that allow them to explore and manipulate visualizations. Interactivity enhances the user's ability to uncover insights and patterns.

3. *Dimensionality Reduction:*

    - Apply dimensionality reduction techniques, such as PCA (Principal Component Analysis) or t-SNE (t-Distributed Stochastic Neighbor Embedding), to represent high-dimensional data in lower-dimensional space for visualization.

4. *Clustering and Classification Visualization:*

    - Visualize clusters and classifications to reveal the grouping and distribution of data points. This is particularly useful in understanding the structure of complex datasets.

5. *Temporal and Spatial Visualization:*

    - Represent temporal and spatial patterns in data through time-series charts, maps, and other visualizations. This is beneficial for understanding trends and geographical relationships.

Applications of Visual Data Mining:

1. *Exploratory Data Analysis (EDA):*

    - Explore datasets visually to identify patterns, outliers, and trends. Visualizations aid in the initial understanding of the data.

2. *Pattern Recognition:*

    - Identify and recognize patterns, correlations, and anomalies in the data through visual exploration.

3. *Cluster Analysis:*

    - Visualize clusters and relationships between data points to understand the grouping and structure of the data.

4. *Decision Support:*

    - Provide visual tools for decision-makers to interact with data, helping them make informed decisions based on a clear understanding of the information.

5. *Anomaly Detection:*

    - Detect anomalies or outliers in the data by visualizing patterns that deviate from the norm.

10. Write short notes on: (Any Two) [2x5=10]

a) Data Mart

A data mart is a subset of a data warehouse that is designed to serve the data and analytical needs of a specific business unit, department, or functional area within an organization. It is a smaller, more focused version of a data warehouse and is created to address the unique requirements of a particular group of users.

Key Characteristics of Data Marts:

1. *Subject-Specific:*
    - Data marts are subject-specific and cater to the needs of a particular business domain, such as finance, marketing, sales, or human resources.
2. *Subset of Data Warehouse:*
    - While a data warehouse is an integrated repository of enterprise-wide data, a data mart is a subset of this data warehouse.
3. *User-Friendly:*
    - Data marts are designed to be user-friendly and accessible to non-technical business users.
4. *Faster Implementation:*
    - Due to their smaller scope, data marts can be implemented more quickly compared to the entire data warehouse.
5. *Increased Performance:*
    - Since data marts only contain a subset of the overall data, queries and reports generated from a data mart often have faster response times.

Types of Data Marts:

1. *Dependent Data Mart:*
    - A dependent data mart is built directly from the data warehouse. It relies on the centralized data warehouse for its source data and is often created to address the specific needs of a department or business unit.
2. *Independent Data Mart:*
    - An independent data mart is built separately from the data warehouse.

Benefits of Data Marts:

1. *Improved Performance:*
    - Data marts provide faster query performance since they contain a focused subset of data relevant to specific business needs.

2. *Enhanced User Satisfaction:*

    - Users in a particular business unit find data marts more user-friendly and aligned with their specific requirements, leading to increased satisfaction.

3. *Quick Deployment:*

    - Data marts can be implemented relatively quickly, allowing organizations to address immediate business needs without the complexity of a full-scale data warehouse.

4. *Cost-Effectiveness:*

    - By focusing on specific business units, data marts can be a cost-effective solution compared to building and maintaining a comprehensive enterprise-wide data warehouse.

b). Privacy and security issues in data mining

Privacy Issues in Data Mining

Privacy concerns in data mining arise from the potential disclosure of sensitive or personally identifiable information during the process of extracting knowledge from large datasets. As data mining techniques involve analyzing and uncovering patterns in data, there is a risk of revealing information that individuals may want to keep private.

Key Privacy Challenges:

↳ *Data Re-identification:*

Even after anonymization or de-identification, there is a risk of re-identifying individuals by combining seemingly non-sensitive attributes. This poses a significant threat to privacy.

↳ *Inference Attacks:*

Attackers may use auxiliary information to infer sensitive details about individuals, exploiting patterns and correlations in the data.

↳ *Group Privacy:*

Protecting the privacy of groups, such as minorities or communities, is challenging, as patterns in the data may inadvertently disclose information about these groups.

- *Contextual Privacy:*

  The context in which data is used matters. Information that might be non-sensitive in one context can become sensitive when combined with other data or used for specific purposes.

Privacy Preservation Strategies:

- *Anonymization and De-identification:*

  Removing or disguising personally identifiable information (PII) in the dataset to prevent direct identification of individuals.

- *Differential Privacy:*

  Introducing noise or perturbation to the data before analysis to provide statistical guarantees of privacy, even when dealing with aggregated results.

- *K-Anonymity and L-Diversity:*

  Ensuring that each individual in the dataset is indistinguishable from at least k-1 other individuals with respect to certain attributes, and ensuring diverse values within sensitive attribute groups.

Security Issues in Data Mining

Security issues in data mining pertain to the protection of data, algorithms, and results from unauthorized access, manipulation, or disclosure. Ensuring the security of the data mining process is essential to maintain the integrity and confidentiality of the information.

Key Security Challenges:

- *Data Theft and Unauthorized Access:*

  Unauthorized access to sensitive data or algorithms can lead to data theft, potentially compromising the confidentiality of information.

- *Model Inversion Attacks:*

  Attackers may attempt to reverse-engineer the model by probing it with inputs to understand its structure and gain insights into the training data.

- *Adversarial Attacks:*

  Deliberate manipulation of input data to mislead or compromise the accuracy of machine learning models.

- *Secure Multi-Party Computation:*

  Collaborative data mining where multiple parties contribute to a model without revealing their raw data introduces challenges in maintaining security.

Security Preservation Strategies:

- *Access Control and Authentication:*

  Implement strict access controls to limit who can access the data and ensure that users are authenticated before interacting with the data mining system.

- *Encryption:*

  Encrypt sensitive data during storage and transmission to prevent unauthorized access. Homomorphic encryption allows computations on encrypted data without decrypting it.

- *Secure Enclaves:*

  Use hardware-based secure enclaves, such as Intel SGX, to protect sensitive computations and algorithms from being accessed or tampered with.

- *Model Robustness:*

  Enhance the robustness of models against adversarial attacks by incorporating techniques like adversarial training and model diversification.

c) Data manipulation language

Data Manipulation Language (DML) is a subset of SQL (Structured Query Language) used for managing and manipulating data stored in a relational database management system (RDBMS). DML enables users to interact with the database by performing operations such as inserting, updating, and deleting data. Unlike Data Definition Language (DDL), which focuses on defining and managing the structure of the database, DML deals with the actual data within the database.

Key Features and Operations of DML:

1. *SELECT Statement:*

   - The SELECT statement is a fundamental component of DML and is used to retrieve data from one or more tables in the database. It allows users to specify the columns they want to retrieve, apply filtering conditions, and sort the results.

Example

SELECT column1, column2 FROM table_name WHERE condition;

2. *INSERT Statement:*

- The INSERT statement is used to add new records to a table. Users can explicitly specify values for each column or use a subquery to insert data from another table.

  Example

  INSERT INTO table_name (column1, column2) VALUES (value1, value2);

3. *UPDATE Statement:*

- The UPDATE statement is used to modify existing records in a table. Users can set new values for specified columns based on certain conditions.

  Example

  UPDATE table_name SET column1 = value1 WHERE condition;

4. *DELETE Statement:*

- The DELETE statement is used to remove records from a table based on specified conditions. If no conditions are provided, all records in the table will be deleted.

  Example

  DELETE FROM table_name WHERE condition;

5. *MERGE Statement:*

- The MERGE statement is a powerful DML operation that performs an "upsert," meaning it can either insert new records or update existing records based on specified conditions.

  Example

  MERGE INTO target_table USING source_table ON (condition) WHEN MATCHED THEN UPDATE SET column1 = value1 WHEN NOT MATCHED THEN INSERT (column1, column2) VALUES (value1, value2);

6. *Transaction Control Statements:*

- DML operations often occur within transactions. Transaction control statements, such as COMMIT and ROLLBACK, are used to manage the integrity and consistency of data by either making changes permanent or undoing them.

Example

COMMIT;

ROLLBACK;

Importance of DML:

1. *Data Modification:*

    - DML provides the necessary tools to modify the data in a database, allowing users to add, update, or delete records as needed.

2. *Querying Data:*

    - Through the SELECT statement, DML enables users to retrieve specific information from tables, facilitating data analysis and reporting.

3. *Maintaining Data Integrity:*

    - DML operations are essential for maintaining data integrity by ensuring that changes to the database adhere to predefined constraints and rules.

4. *Support for Transactions:*

    - DML statements are often part of transactions, which are units of work that ensure data consistency and reliability. Transactions can be committed to make changes permanent or rolled back to undo them.

5. *Data Security:*

    - DML operations can be controlled and restricted based on user roles and permissions, enhancing data security and preventing unauthorized modifications.

d) Mean square deviation

Mean Squared Deviation is a measure of the average squared difference between each data point and the mean of the dataset. It quantifies the dispersion or variability of a set of values.

Key Points:

- Squaring the differences ensures that both positive and negative deviations contribute to the measure without canceling each other out.

- Mean Squared Deviation is commonly used in statistics and regression analysis.
- It is a part of the calculation for variance and standard deviation, where variance is the average of squared deviations.

Purpose:

- MSD provides a measure of the spread of data points around the mean, helping to understand the variability within a dataset.
- It is useful for assessing the goodness of fit of a model or the accuracy of predictions.