Data-Mining-Module-1-Important-Topics-PYQs

- Data-Mining-Module-1-Important-Topics-PYQs
  - 1. List out the three major features of data warehouse.
  - 2. Describe the similarities and the differences of OLTP and OLAP.
    - Similarities Between OLTP and OLAP
    - Difference Between OLTP and OLAP
  - 3. List and explain any two applications of data warehouse
    - 1. Business Intelligence and Reporting
    - 2. Customer Relationship Management (CRM)
  - 4. Describe the similarities and the differences of star schema and snowflake schema
    - 1. Star Schema
      - Characteristics:
      - Example:
      - Advantages:
      - Disadvantages:
    - 2. Snowflake Schema
      - Characteristics:
      - Example:
      - Advantages
      - Disadvantages:
  - 5. Illustrate the multi-dimensional data model with a neat figure.
    - 1. Data Cube:
    - 2. Dimensions:
    - 3. Facts:
    - 4. Example:

    - 5. Visualizing the Data Cube:
- 6. Explain different OLAP operations on multidimensional data with suitable examples.List the dififerences between ROLAP. MOLAP and HOLAP.
    - 1. Roll-up (Drill-up):
    - 2. Drill-down:
    - 3. Slice:
    - 4. Dice:
    - 5. Pivot (Rotate):
    - Example Using a Sales Data Cube:
        - Roll-up Example:
        - Drill-down Example:
        - Slice Example:
        - Dice Example:
        - Pivot Example:
    - Differences between ROLAP, MOLAP and HOLAP.
    - 1. Relational OLAP (ROLAP) Servers:
    - 2. Multidimensional OLAP (MOLAP) Servers:
    - 3. Hybrid OLAP (HOLAP) Servers:
- 7. Illustrate the various stages in Knowledge discovery process with a diagram.
    - 1. Data Cleaning:
    - 2. Data Integration:
    - 3. Data Selection:
    - 4. Data Transformation:
    - 5. Data Mining:
    - 6. Pattern Evaluation:
    - 7. Knowledge Presentation:
- 8. Suppose that a data warehouse consists of the three dimensions: time, doctor, and patient, and the two measures: count and charge, where charge is the fee that a doctor charges a patient for a visit.
    - i. Draw a schema diagram for the above data warehouse using star schema
    - ii. Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2023?

- 9. Illustrate the various stages of data mining in business intelligence .
- 10. Describe different issues in data mining
  - 1. Mining Methodology and User Interaction Issues
    - 1.1 Mining Different Kinds of Knowledge in Databases
    - 1.2 Interactive Mining at Multiple Levels of Abstraction
    - 1.3 Incorporation of Background Knowledge
    - 1.4 Data Mining Query Languages and Ad-hoc Mining
    - 1.5 Presentation and Visualization of Data Mining Results
    - 1.6 Handling Noisy or Incomplete Data
    - 1.7 Pattern Evaluation
  - 2. Performance Issues
    - 2.1 Efficiency and Scalability of Data Mining Algorithms
    - 2.2 Parallel, Distributed, and Incremental Mining Algorithms
  - 3. Issues Relating to the Diversity of Database Types
    - 3.1 Handling Relational and Complex Types of Data
    - 3.2 Mining Information from Heterogeneous Databases
- 11. Suppose that a data warehouse for a university consists of the following four dimensions: student, course, semester, and instructor, and two measures: count and avg grade.
  - (i) Draw a snowflake schema diagram for the data warehouse.
  - (ii) starting with the base cuboid, what specific OLAP operations perform in order to list the average grade of cs courses for each student.
- 12. Explain the three-tier architecture of the data warehouse with a neat figure.
  - 1. Bottom Tier (Data Warehouse Server)
  - 2. Middle Tier (OLAP Server)
  - 3. Top Tier (Front-End Tools)
- 13. List and illustrate the schemas used for the physical representation of the multi-dimensional data with examples.
  - 1. Star Schema
  - 2. Snowflake Schema
  - 3. Fact Constellation Schema (Galaxy Schema)
- 14. List and explain various data mining functionalities.
  - 1. Data Characterization:

1. List out the three major features of data warehouse.

A **Data Warehouse (DW)** is a centralized repository used to store large amounts of structured data, typically collected from various sources within an organization. It is optimized for querying, reporting, and data analysis rather than transaction processing. These are the **three major features** of a data warehouse:

1. **Subject-Oriented**:
   - A data warehouse is designed to focus on key business subjects rather than on the organization's ongoing operations.
   - Subjects can include things like **products**, **customers**, **sales**, **revenue**, **suppliers**, etc.
   - This structure helps organizations analyze data from specific business perspectives, enabling insights related to key areas of interest.
2. **Integrated**:
   - Data in a data warehouse comes from various heterogeneous sources such as **relational databases**, **flat files**, **external APIs**, etc.
   - The data is integrated to provide a **consistent view** of the information, making it easier to analyze and report across different datasets.
   - This integration helps businesses avoid silos of data and enables more comprehensive decision-making
3. **Time-Variant**:
   - The data stored in a data warehouse is typically time-stamped and organized by periods such as **year**, **quarter**, **month**, or **day**.
   - This feature enables historical analysis, as it allows users to look at data over time and compare trends, changes, and patterns from past periods.
   - Time-variant data provides valuable insights into **historical performance** and helps with trend analysis and forecasting.

2. Describe the similarities and the differences of OLTP and OLAP.

1. **OLTP (Online Transaction Processing)**: A system that handles **daily transactions** like banking, online shopping, or airline bookings. It is designed for **fast and frequent updates** to data, ensuring accuracy and quick responses.
2. **OLAP (Online Analytical Processing)**: A system that is used for **analyzing large amounts of historical data** to help businesses make decisions. It focuses on **data trends, summaries, and complex queries** rather than quick updates. Examples include sales forecasting and business intelligence reports.

Similarities Between OLTP and OLAP

- **Both are Data Processing Systems**: Both OLTP and OLAP handle large volumes of data and are part of data management systems, but they focus on different types of operations.
- **Use of Databases**: Both OLTP and OLAP systems rely on databases to store and manage data.
- **Business Applications**: Both are used to support business processes; OLTP is used in transactional applications, while OLAP is used for decision support and business intelligence.

Difference Between OLTP and OLAP

Feature	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
1. Users & Purpose	Used by **clerks, clients, IT professionals** for day-to-day transactions and queries.	Used by **managers, executives, analysts** for decision-making and data analysis.
2. Data Type	Stores **current** and frequently updated data.	Stores **historical** data for analysis, summaries, and trends.
3. Database Design	Uses **Entity-Relationship (ER) models** with an application-oriented design.	Uses **Star or Snowflake schema** with a subject-oriented design.
4. Data Scope	Focuses on **real-time data** for an enterprise or department.	Includes **historical data** from multiple versions and organizations.

Feature	OLTP (Online Transaction Processing)	OLAP (Online Analytical Processing)
5. Storage & Access	Small-sized transactions, needs fast processing, concurrency control, and recovery.	Large, complex queries that are mostly **read-only** and analyzed in a multidimensional way.

- **OLTP** is for quick transactions (e.g., banking, order processing).
- **OLAP** is for analyzing large amounts of data (e.g., sales trends, business intelligence).

3. List and explain any two applications of data warehouse

1. Business Intelligence and Reporting

- **Explanation**: Business Intelligence (BI) involves analyzing historical data to uncover patterns, trends, and insights that can drive business decisions.
- A data warehouse consolidates data from different departments and systems, which can then be queried and analyzed to generate reports, dashboards, and visualizations that support strategic decision-making.
- **Application**:
  - Companies use data warehouses to generate periodic **financial reports**, **sales reports**, or **customer performance reports**.
  - This helps in tracking business performance, identifying potential opportunities, and improving operational efficiency.
  - For example, a **retail business** can use a data warehouse to analyze sales data across different regions, products, and times to identify seasonal trends or best-selling products.
  - The insights from such reports can help the company optimize inventory management and marketing strategies.

2. Customer Relationship Management (CRM)

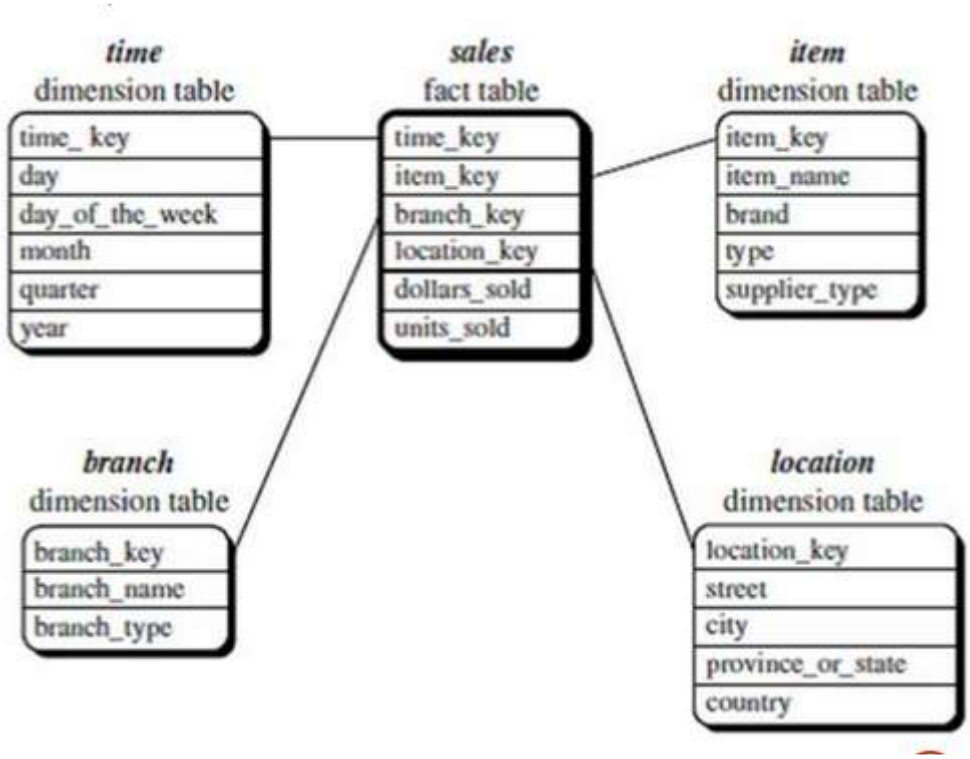
- **Explanation**: Data warehouses can be used to enhance **Customer Relationship Management (CRM)** by integrating customer data from multiple touchpoints (e.g., sales, support, marketing) into a single, unified view.

- This enables organizations to better understand customer behaviors, preferences, and buying patterns over time.
- **Application**:
  - Companies use data warehouses to track and analyze customer interactions and behavior across different channels.
  - This information helps in creating **targeted marketing campaigns**, personalized recommendations, and improving customer service.
  - For example, an **e-commerce company** can use a data warehouse to store data on customer purchases, product preferences, and browsing history.
  - This data can be used to segment customers and deliver personalized offers or loyalty rewards, thereby improving customer retention and satisfaction.

4. Describe the similarities and the differences of star schema and snowflake schema

Both the **Star Schema** and **Snowflake Schema** are used in **data warehousing** for organizing data into a multidimensional model, which allows for efficient querying and reporting. They are both popular models, but they have key differences in how they structure data.

1. Star Schema



Characteristics:

- **Simple Structure**: The Star Schema has a simple, intuitive design where a central **fact table** is connected to multiple **dimension tables**.
- **Dimension Tables**: Each dimension is represented by a single table. These tables contain descriptive information about the data (e.g., time, item, location).
- **Fact Table**: The fact table is at the center of the schema, containing foreign keys referencing each of the dimension tables. It also contains **measurable facts** (e.g., sales figures, quantities sold).

Example:

In a **sales database**, the fact table might store sales amounts and quantities, while dimension tables could include information on **time**, **items**, **branches**, and **locations**.

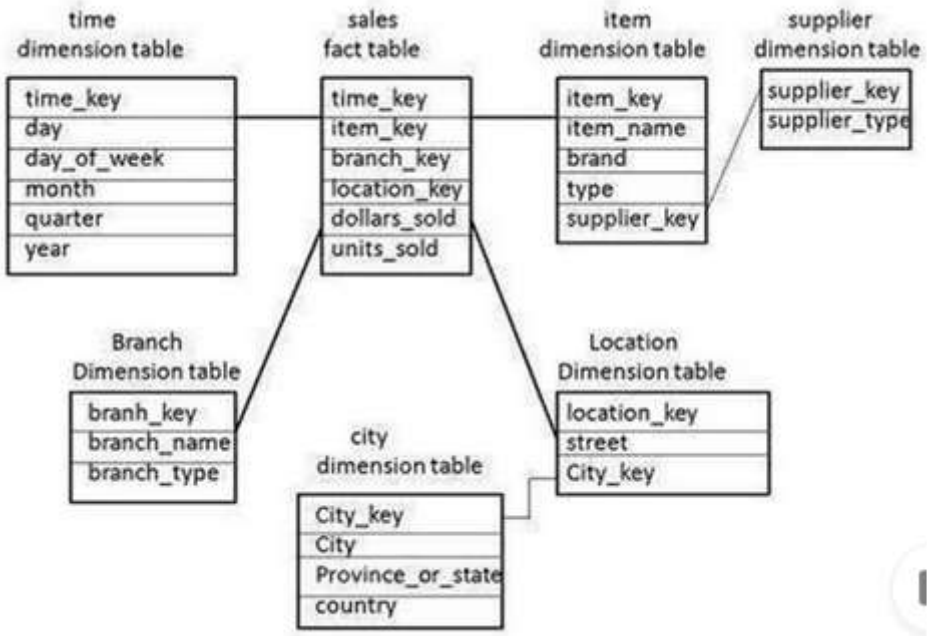
Advantages:

- **Simpler Queries**: Since the schema is less complex, queries are easier to write.
- **Faster Performance**: The denormalized nature of the tables allows for faster read performance, as fewer joins are required.

Disadvantages:

- **Data Redundancy**: Because dimension tables are not normalized, there can be some data redundancy, which can increase storage requirements.

2. Snowflake Schema



Characteristics:

- **Normalized Structure**: In the Snowflake Schema, dimension tables are **normalized** into multiple related tables. This leads to a more complex structure compared to the Star Schema.
- **Normalization**: Each dimension is broken down into multiple smaller, related tables. For example, an "item" dimension in the Star Schema may be split into "item" and "supplier" tables in the Snowflake Schema.
- **Fact Table**: Similar to the Star Schema, the fact table remains at the center and holds foreign keys that reference the dimension tables.

Example:

In the **snowflake schema**, the "item" dimension table may be divided into multiple related tables:

- **Item Table**: Contains attributes like item key, name, type, brand.
- **Supplier Table**: Contains information about suppliers, linked to the Item table via a supplier key.

Advantages

- **Data Integrity**: Because dimension tables are normalized, there is less redundancy, and data integrity is improved.

- **Efficient Storage**: Reduced storage requirements as data is split across multiple related tables.

Disadvantages:

- **Complex Queries**: Queries in a Snowflake Schema require more joins due to the normalization, making them more complex.
- **Performance Overhead**: More joins may slow down performance, especially with large datasets.

Feature	Star Schema	Snowflake Schema
Structure	Simple, with a central fact table and directly linked dimension tables.	More complex, with normalized dimension tables that are further split into smaller tables.
Normalization	Dimension tables are denormalized.	Dimension tables are normalized to reduce redundancy.
Query Complexity	Easier to query, as it requires fewer joins.	More complex queries due to additional joins between normalized tables.
Storage	Higher storage requirements due to denormalization.	Lower storage requirements due to normalization.
Performance	Faster performance due to fewer joins.	Slower performance due to more joins.
Data Redundancy	Higher redundancy in dimension tables.	Lower redundancy due to normalization.
Maintenance	Easier to maintain as data is simpler.	More difficult to maintain due to the complex structure.

5. Illustrate the multi-dimensional data model with a neat figure.

The **multidimensional data model** is a way to organize and analyze data in a structure that allows you to view it from multiple perspectives. It's like looking at a data set from different angles, and the most common way to represent this is through a **data cube**. Let's break this down step by step in simple terms:

1. Data Cube:

- Imagine you have a cube (like a Rubik's cube) where each face represents a different perspective of your data.
- In the world of data, this cube can have many dimensions. These dimensions are the different aspects of your data that you want to track or analyze.
- For example, in a store's sales data, the dimensions might be **time**, **item**, **branch**, and **location**.
- A **data cube** is used to view the data in multiple dimensions at once.

2. Dimensions:

- **Dimensions** are like categories or perspectives that describe your data.
- For example:
  - **Time**: What time period are you looking at? (Daily, monthly, quarterly, etc.)
  - **Item**: What items are being sold?
  - **Location**: Where are the items being sold (different cities, stores, etc.)?
- Each dimension has its own **dimension table**, which gives more details about that dimension.
  - For example, a **dimension table** for items might list the **item name**, **brand**, and **type**.
- Dimensions help you organize your data and let you focus on specific parts of it.

3. Facts:

- **Facts** are the numeric values you want to analyze. These are the actual measurements.
- In a sales data model, **facts** might include:
  - **Sales amount** (how much money was earned)
  - **Units sold** (how many items were sold)
  - **Budgeted amount** (how much money was expected to be earned)
- These facts are typically stored in a **fact table**, and this table also connects to the dimension tables using keys.
  - For example, a **fact table** for sales might contain sales data and include links to the time, item, and location dimensions.

4. Example:

Let's say you have a shop that sells electronics, and you want to track sales data.

2-D View of Sales Data for *AllElectronics* According to *time* and *item*

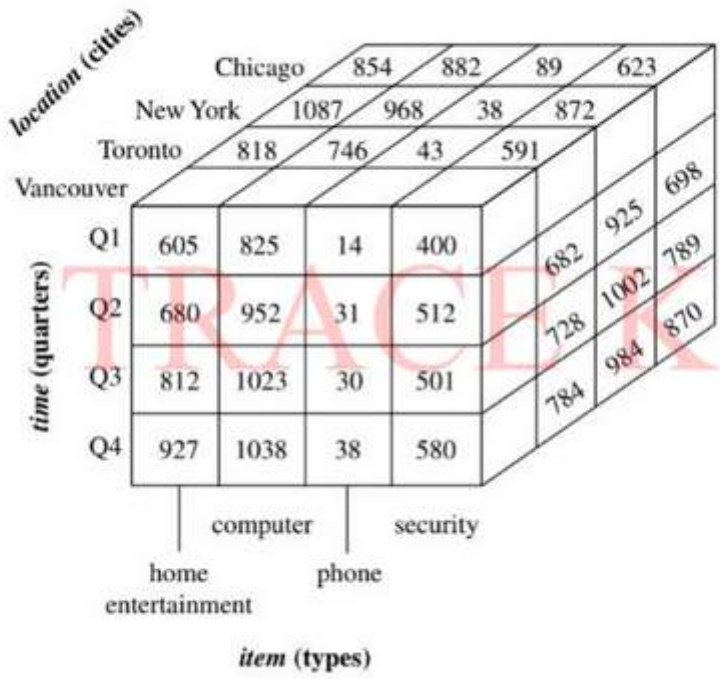
	location = "Vancouver"			
	item (type)			
time (quarter)	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

- **Dimensions** could be:
  - **Time**: Data organized by quarters (Q1, Q2, etc.)
  - **Item**: Different electronics like TVs, phones, and laptops
  - **Location**: Different cities or stores like Vancouver, Chicago, etc.
- **Facts** could be:
  - **Dollars sold**: How much money was made
  - **Units sold**: How many items were sold

Now, if you look at your sales data for a specific city, like **Vancouver**, you could organize it by **Time** (quarters) and **Item** (types of electronics). The result would look like a 2D table, where you have the **quarter** on one axis and the **item types** on another.

But, if you want to include **Location** (let's say Vancouver, Chicago, New York, etc.), you would then create a 3D table or data cube. Each additional dimension gives you more detailed insights into your data.

	location = "Chicago"				location = "New York"				location = "Toronto"				location = "Vancouver"			
	item				item				item				item			
time	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580



5. Visualizing the Data Cube:

- **2D Table Example** (for Vancouver, broken down by Time and Item):
  - The columns could represent **Items** (TVs, phones, etc.)
  - The rows could represent **Time** (Q1, Q2, etc.)
  - The data inside would show the **dollars sold**.
- **3D Data Cube Example** (with an added Location dimension):
  - Now, each city (like Vancouver, Chicago, etc.) becomes a different slice of the cube.
  - You could view sales by **Time**, **Item**, and **Location** all at once.

The **multidimensional data model** helps you view your data from multiple perspectives or dimensions. A **data cube** organizes your data into multiple dimensions (like Time, Item, and Location) and stores facts (like Sales or Units sold). This allows for easier analysis and better decision-making, as you can break down data by different angles and uncover patterns or trends.

This model is commonly used in **data warehousing** where organizations want to store, analyze, and report on large sets of data in a way that is easy to understand and explore.

6. Explain different OLAP operations on multidimensional data with suitable examples.List the dififerences between ROLAP. MOLAP and HOLAP.

OLAP (Online Analytical Processing) operations are used to manipulate and analyze data in a multidimensional data model, making it easier to explore the data from different perspectives. These operations allow users to interactively query and analyze data

These are the OLAP operations

1. Roll-up (Drill-up):

- **What it does:**
  - Roll-up aggregates data by moving from a lower level of detail to a higher level.
  - It can be done by navigating up a concept hierarchy or by reducing the number of dimensions.
- **How it works:**
  - In a **location** dimension, you might start by looking at sales by **city** (detailed level). By rolling up, you move to an aggregated view by **country** (higher level).
  - **Example:**
    - If the data cube is organized by cities (Chicago, New York, Vancouver, Toronto), rolling up by location would group the data by **country** (USA, Canada).
    - In a sales data warehouse, rolling up the **time** dimension might aggregate sales by **quarter** instead of by **month**.

2. Drill-down:

- **What it does:**
  - Drill-down is the reverse of roll-up. It navigates from more general (less detailed) data to more specific (detailed) data.
  - You can drill down by descending through a concept hierarchy or adding more dimensions to the data.
- **How it works:**
  - In the **time** dimension, you might drill down from **quarters** to **months** to get more detailed data.
  - **Example:**

- If you have data aggregated by **quarter**, drilling down would break it into sales by **month**, giving more detail.

3. Slice:

- **What it does:**
  - Slice selects data from one dimension, creating a **subcube**.
  - It is used to view data for a specific value in a particular dimension.
- **How it works:**
  - You can select a specific slice of the data based on one dimension.
  - **Example:**
    - In a sales data cube, you might use the slice operation to look at sales for **Q1** (time dimension) across all items and locations.

4. Dice:

- **What it does:**
  - Dice defines a **subcube** by selecting data from multiple dimensions at once.
  - It allows you to look at a more specific set of data by specifying conditions for two or more dimensions.
- **How it works:**
  - You can choose specific values in more than one dimension to create a subcube.
  - **Example:**
    - You might dice the sales data to view only sales for:
      - **Location = “Toronto” or “Vancouver”**
      - **Time = “Q1” or “Q2”**
      - **Item = “home entertainment” or “computer”**
    - This gives a subset of the data for the selected cities, time periods, and items.

5. Pivot (Rotate):

- **What it does:**
  - Pivot is a visualization operation that rotates or changes the data axes, offering a new perspective of the data.
  - It helps in rearranging the view of data in a different format to make analysis easier or to focus on different parts of the data.

- **How it works:**
  - Pivoting changes how dimensions are displayed in the data view, often rotating axes.
  - **Example:**
    - If the data cube shows sales data with **items** on the rows and **locations** on the columns, pivoting would rotate it to display **locations** as rows and **items** as columns.
    - In a 3D data cube, pivoting might transform it into a series of 2D slices to offer a clearer or more useful presentation.

Example Using a Sales Data Cube:

Let's say you have a sales data cube with three dimensions:

- **Location** (Chicago, New York, Toronto, Vancouver)
- **Time** (Q1, Q2, Q3, Q4)
- **Item** (Home entertainment, Computer, TV, Mobile)

Roll-up Example:

- Aggregating the data for the **location** dimension from cities to countries (e.g., from **Chicago** to **USA**).

Drill-down Example:

- If data is aggregated by **quarters**, drill down to get more detailed data by **months**.

Slice Example:

- Extract data only for **Q1** (from the **Time** dimension) and look at sales across all **locations** and **items**.

Dice Example:

- Select data for **Toronto** and **Vancouver**, **Q1** and **Q2**, and items like **Home entertainment** and **Computers**. This creates a subcube of data.

Pivot Example:

- Rotate the cube to display **Location** on the rows and **Item** on the columns, instead of the other way around.

Differences between ROLAP, MOLAP and HOLAP.

The three main types of OLAP (Online Analytical Processing) servers—**ROLAP**, **MOLAP**, and **HOLAP**—differ in how they store, manage, and process data.

1. Relational OLAP (ROLAP) Servers:

- **Storage Type**: ROLAP servers store data in relational or extended-relational databases (such as SQL databases).
- **Data Processing**: They rely on relational database management systems (RDBMS) to store and query multidimensional data. OLAP operations like roll-up, drill-down, and slice are processed dynamically by executing SQL queries

2. Multidimensional OLAP (MOLAP) Servers:

- **Storage Type**: MOLAP servers use multidimensional data storage, typically in the form of data cubes (array-based storage).
- **Data Processing**: MOLAP stores summarized data and pre-computes aggregations in multidimensional arrays (cubes), allowing fast data retrieval and analysis.

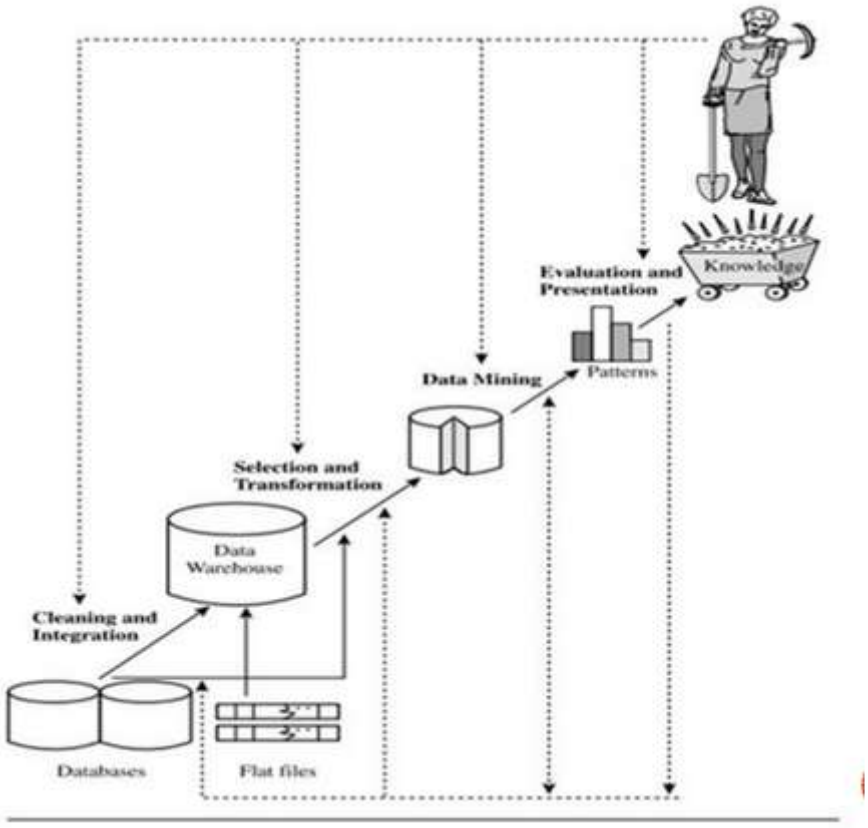
3. Hybrid OLAP (HOLAP) Servers:

- **Storage Type**: HOLAP combines both ROLAP and MOLAP techniques to benefit from the strengths of both approaches. Detailed data is stored in a relational database (ROLAP), while aggregated data is stored in multidimensional cubes (MOLAP).
- **Data Processing**: HOLAP servers allow for the storage of large amounts of detailed data in a relational database, while aggregations are kept in the MOLAP storage for faster access.

Feature	ROLAP	MOLAP	HOLAP
Data Storage	Relational databases (SQL)	Multidimensional arrays (cubes)	Combination of ROLAP and MOLAP (relational for detailed data, MOLAP for aggregations)
Performance	Slower (requires on-the-fly queries)	Faster (pre-computed aggregations)	Balanced (fast for aggregations, scalable for detailed data)
Scalability	High scalability (uses RDBMS)	Limited scalability (due to cube storage)	High scalability (relies on relational storage for detail)

Feature	ROLAP	MOLAP	HOLAP
Data Aggregation	Done dynamically via SQL queries	Pre-computed in the cube	Pre-computed for aggregates, dynamic for detailed data
Best for	Large, detailed datasets requiring flexibility	Fast access to aggregated data	Large datasets with both detailed data and aggregates
Complexity	Lower (relies on relational database)	Higher (due to multidimensional cubes)	Higher (manages two storage formats)
Example	Microstrategy DSS Server	IBM Cognos OLAP, Oracle OLAP	Microsoft SQL Server 2000 (supports HOLAP)

7. Illustrate the various stages in Knowledge discovery process with a diagram.



The **Knowledge Discovery in Databases (KDD)** process is a comprehensive sequence of steps aimed at extracting useful patterns and knowledge from large datasets. Data mining is a

crucial part of this process but is just one of its stages.

1. Data Cleaning:

- **Purpose**: To remove noise and inconsistencies in the data.
- **Actions**:
  - Handling missing data (e.g., through imputation or removal).
  - Removing duplicates.
  - Correcting errors in data (e.g., fixing inconsistencies or invalid values).
- **Example**: If some sales records have missing values for the product price, those rows are cleaned up before analysis.

2. Data Integration:

- **Purpose**: To combine data from multiple sources into a unified view.
- **Actions**:
  - Merging data from different databases or sources, such as combining customer data from a CRM system with transaction data from an e-commerce platform.
  - Ensuring that data from different sources match and are consistent.
- **Example**: If a retail company has data in separate databases for customer information, inventory, and sales, the integration process would combine these into a single dataset.

3. Data Selection:

- **Purpose**: To select relevant data for analysis.
- **Actions**:
  - Filtering the data to focus only on the data points that are necessary for the task at hand.
  - Removing irrelevant data that will not contribute to the analysis.
- **Example**: If the task is to analyze customer behavior, only the customer and transaction-related data might be selected, excluding irrelevant information like employee records.

4. Data Transformation:

- **Purpose**: To transform and consolidate the data into a form suitable for mining.
- **Actions**:
  - **Normalization**: Scaling data to a common range.

    - **Aggregation**: Combining data to a higher level (e.g., summing daily sales into monthly sales).
    - **Feature extraction**: Creating new features from existing ones (e.g., deriving a "profit margin" feature from "cost" and "price").
- **Example**: If sales data is recorded daily, the transformation step could aggregate it into monthly data for easier analysis.

5. Data Mining:

- **Purpose**: To apply intelligent methods and algorithms to extract patterns and knowledge from the data.
- **Actions**:
    - Use data mining techniques such as classification, clustering, regression, and association rule mining.
    - It’s the core process where patterns are discovered through algorithms.
- **Example**: Using clustering to group customers based on purchasing behavior or applying decision trees to predict customer churn.

6. Pattern Evaluation:

- **Purpose**: To evaluate and identify the truly interesting patterns.
- **Actions**:
    - Assess the discovered patterns using interestingness measures, which could include factors like novelty, usefulness, and validity.
    - Identify patterns that offer actionable insights or represent significant knowledge.
- **Example**: Evaluating the strength of association rules like "If a customer buys product A, they are 80% likely to buy product B," and deciding if the rule is worth further exploration.

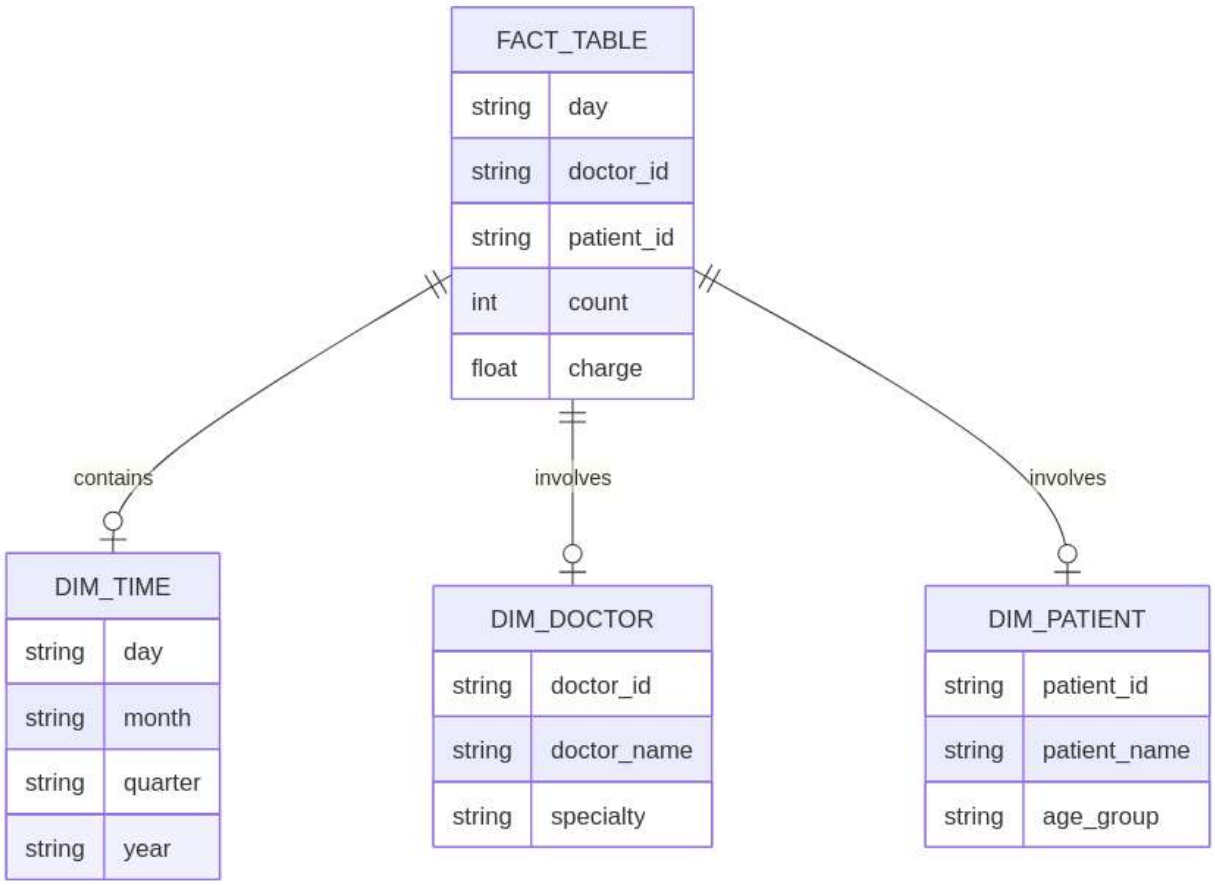
7. Knowledge Presentation:

- **Purpose**: To present the discovered patterns in a user-friendly manner.
- **Actions**:
    - Visualizing the results through charts, graphs, or dashboards.
    - Using techniques like decision trees, heat maps, and summary reports to communicate the findings to stakeholders.

- **Example**: Displaying a bar chart of customer segments based on purchasing behavior or a decision tree showing factors that influence loan approvals.

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

i. Draw a schema diagram for the above data warehouse using star schema

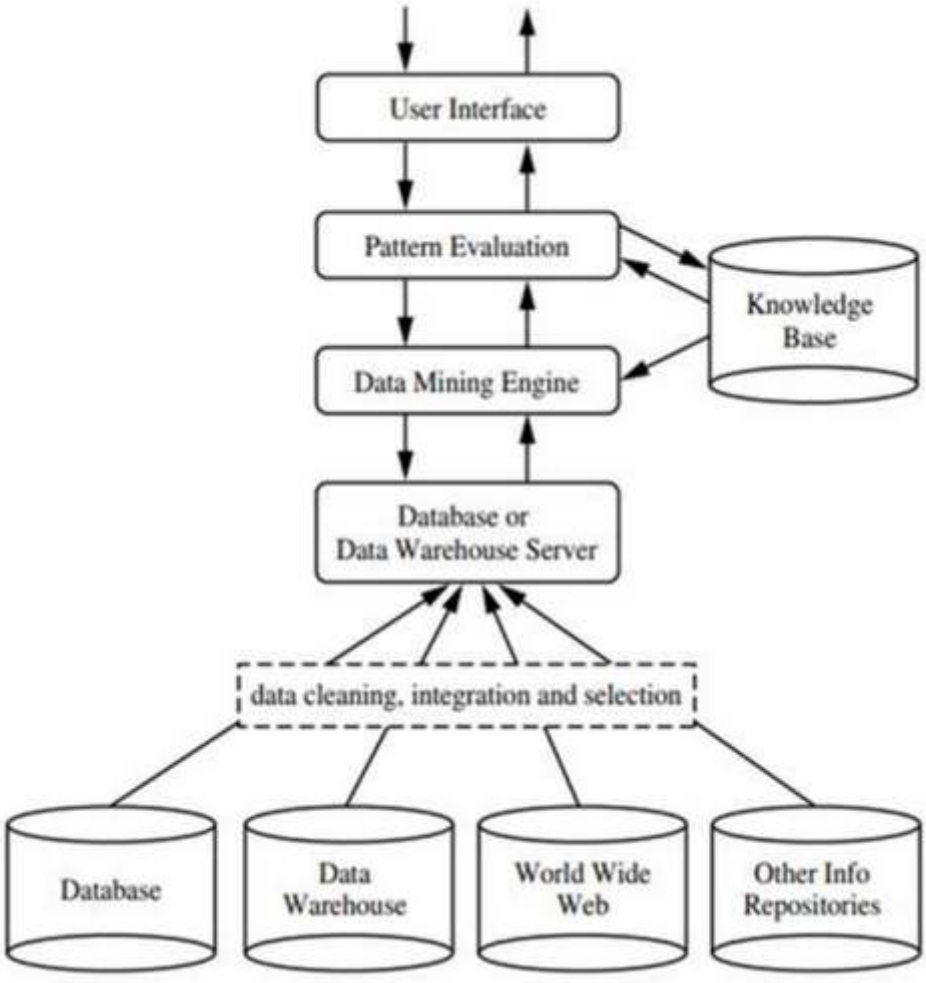


ii. Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2023?

To list the total fee collected by each doctor in 2023, starting from the base cuboid `[day, doctor, patient]`, the following OLAP operations should be performed:

1. **Selection (Filter)**: Filter the fact table to include only records for the year 2023. This will ensure that we are considering only the relevant time period.
2. **Aggregation (Roll-up)**: Aggregate the data by doctor. We need to sum the `charge` for each doctor to get the total fee collected by each doctor.
3. **Projection**: Project the doctor's details (name, specialty) from the `DIM_DOCTOR` table, and join it with the aggregated result.

9. Illustrate the various stages of data mining in business intelligence .



1. **Data Collection and Integration:**

- Collect data from various sources such as databases, data warehouses, web sources, and spreadsheets.
- Integrate this data into a consistent format suitable for analysis.

2. **Data Cleaning:**
   - Remove errors, handle missing values, and smooth out noisy data.
   - This stage ensures that the data used for mining is accurate and reliable.
3. **Data Transformation:**
   - Data is transformed into the required format (e.g., normalizing, aggregating, or discretizing data).
   - It may involve converting raw data into a more useful structure for analysis.
4. **Data Mining:**
   - **Characterization:** Summarizing the general characteristics of a dataset.
   - **Association and Correlation Analysis:** Identifying relationships between variables.
   - **Classification:** Categorizing data into predefined classes based on its features.
   - **Prediction:** Estimating unknown outcomes based on historical data.
   - **Cluster Analysis:** Grouping data points that are similar to each other.
   - **Outlier Analysis:** Detecting rare or exceptional patterns in the data.
   - **Evolution Analysis:** Identifying changes over time in the data.
5. **Pattern Evaluation:**
   - Evaluate the patterns identified in the data mining process for interestingness and relevance.
   - Use interestingness measures (e.g., statistical significance, novelty) to identify which patterns are useful for the business.
6. **Knowledge Representation:**
   - Represent the discovered patterns in a form that is easy to understand and interpret.
   - This may involve creating reports, graphs, or decision support tools for end-users.
7. **Decision-Making and Action:**
   - Use the knowledge gained from data mining to inform business decisions, optimize operations, or take action based on the insights discovered.

10. Describe different issues in data mining

Data mining, which involves extracting useful patterns from large datasets, comes with a variety of challenges. These challenges can be grouped into three main categories: **Mining Methodology and User Interaction**, **Performance Issues**, and **Diversity of Database Types**.

1. Mining Methodology and User Interaction Issues

1.1 Mining Different Kinds of Knowledge in Databases

- **Problem**: Different users want different types of knowledge. For example, a sales manager may be interested in customer buying patterns, while a marketing team might want to know about trends in advertising effectiveness.
- **Challenge**: Data mining needs to be flexible enough to handle many types of analysis. The system should allow for different techniques to be applied depending on what the user wants to discover.

1.2 Interactive Mining at Multiple Levels of Abstraction

- **Problem**: Users often need to explore data at different levels, from high-level trends to very specific details.
- **Challenge**: The system should allow users to interactively explore the data at various levels of detail. For example, a user might start by looking at sales by year, and then drill down to see sales by month, then day, and so on.

1.3 Incorporation of Background Knowledge

- **Problem**: Sometimes, users have prior knowledge about the data or the domain they are working in. For instance, a business analyst may know that certain products sell better in certain seasons.
- **Challenge**: It’s important to use this background knowledge to guide the data mining process and to help express the discovered patterns in a meaningful way.

1.4 Data Mining Query Languages and Ad-hoc Mining

- **Problem**: Users often need to define specific, one-time queries to mine data, which may not fit into predefined templates.
- **Challenge**: A flexible query language should allow users to describe their specific data mining tasks. This would help users ask more customized questions without needing technical expertise.

1.5 Presentation and Visualization of Data Mining Results

- **Problem**: After patterns are discovered, they need to be presented in a way that is easy to understand, especially for non-technical users.
- **Challenge**: The system must provide clear visual representations (charts, graphs, etc.) to communicate the findings effectively. These representations should be simple and intuitive.

1.6 Handling Noisy or Incomplete Data

- **Problem**: Real-world data is often messy—there could be errors or missing values in the dataset.
- **Challenge**: Effective data cleaning techniques must be applied to handle noisy (incorrect or inconsistent) or incomplete data. If this is not done well, the patterns discovered may be inaccurate or misleading.

1.7 Pattern Evaluation

- **Problem**: Not all discovered patterns are useful. Some may be obvious, while others might be too complex or irrelevant.
- **Challenge**: It is important to evaluate the patterns based on their usefulness and novelty. A pattern that's already well-known may not be interesting, while new, actionable patterns are valuable.

2. Performance Issues

2.1 Efficiency and Scalability of Data Mining Algorithms

- **Problem**: Databases can be very large, and running data mining algorithms on huge datasets can take a long time.
- **Challenge**: Algorithms need to be efficient so that they can handle large volumes of data quickly. The speed of the algorithms should be predictable, ensuring that users don't have to wait too long for results.

2.2 Parallel, Distributed, and Incremental Mining Algorithms

- **Problem**: Large datasets are often distributed across multiple machines, and the mining process can become slower if done sequentially.
- **Challenge**: Parallel and distributed algorithms divide the data into parts and process them at the same time across different machines. This helps speed up the mining process. Additionally, incremental algorithms can update results as new data arrives, improving efficiency.

3. Issues Relating to the Diversity of Database Types

3.1 Handling Relational and Complex Types of Data

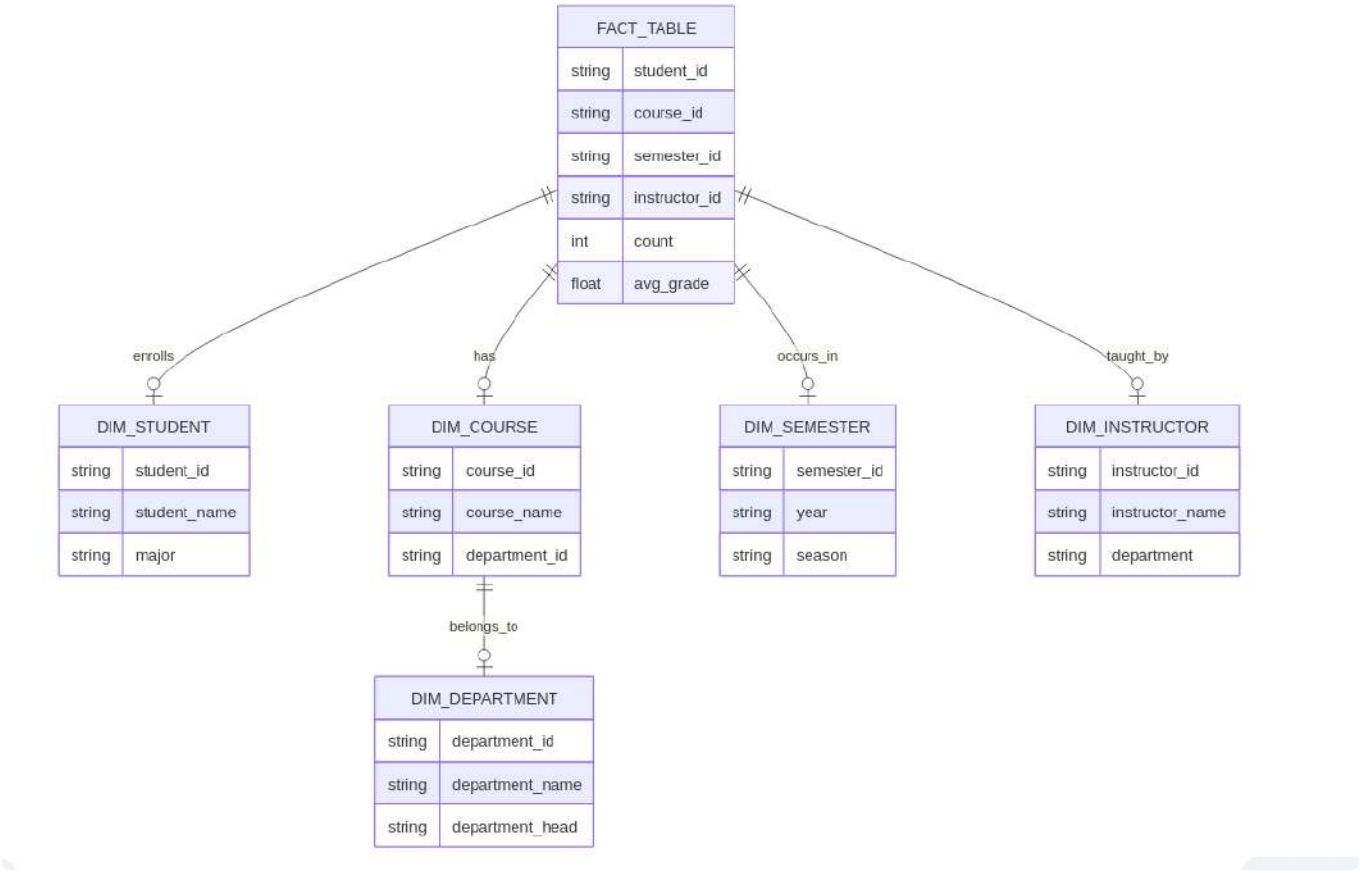
- **Problem**: Data comes in many forms: some data is simple (like numbers in tables), while other data types are complex, like images, videos, or even geographical information.
- **Challenge**: Different types of data require different mining methods. A single mining system may not be able to handle all types of data, so specialized systems are needed for each type (e.g., image mining, spatial data mining).

3.2 Mining Information from Heterogeneous Databases

- **Problem**: Data is often spread across different sources, and these sources may not all have the same structure. For example, some data might be in a relational database, while other data might be in a text file or even in a non-traditional format like social media data.
- **Challenge**: Mining data from heterogeneous sources (databases, web, etc.) can be difficult because the data may be structured differently (structured, semi-structured, unstructured). A unified approach is needed to handle and integrate this data effectively.

11. Suppose that a data warehouse for a university consists of the following four dimensions: student, course, semester, and instructor, and two measures: count and avg grade.

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



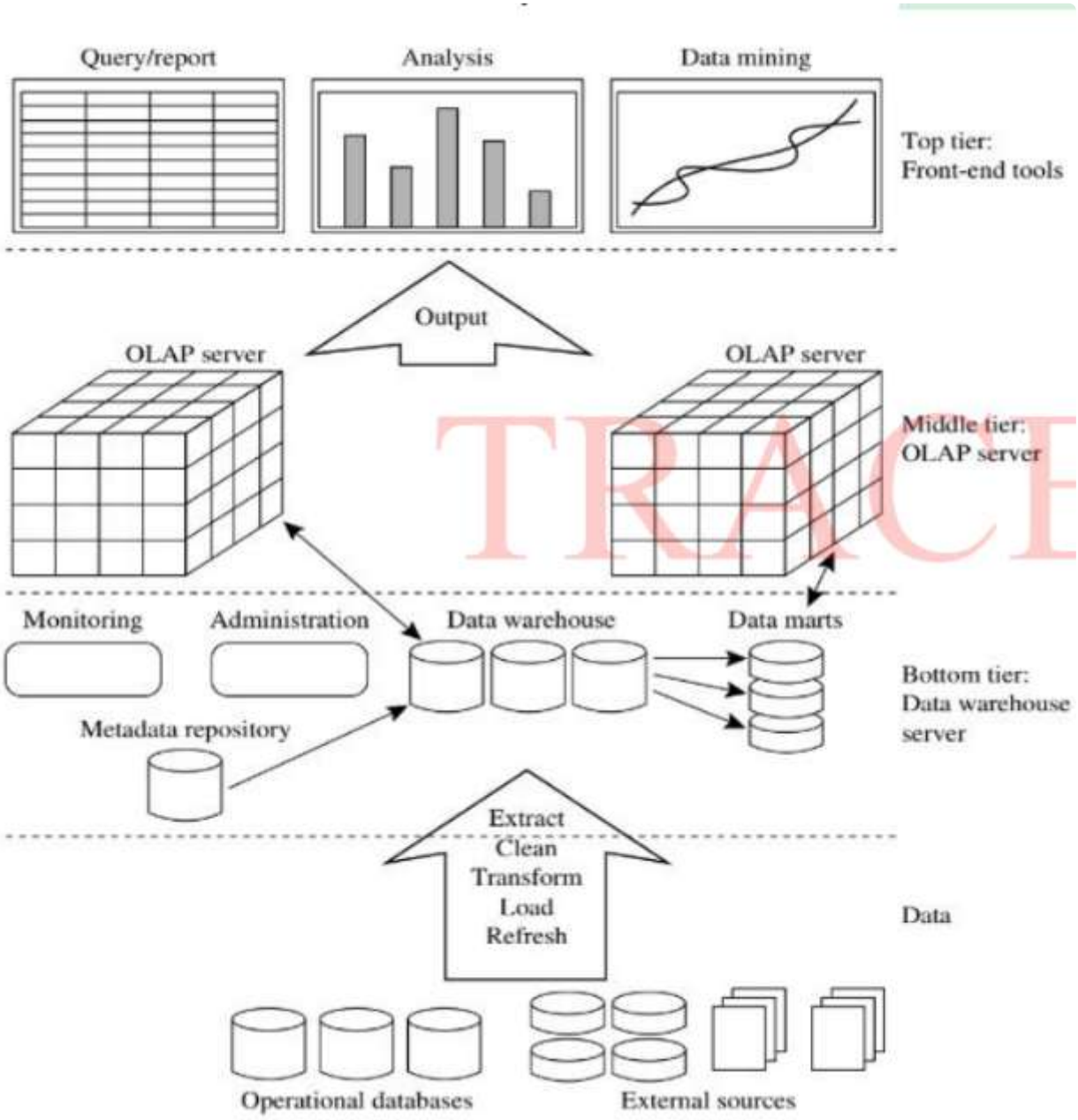
(ii) starting with the base cuboid, what specific OLAP operations perform in order to list the average grade of cs courses for each student.

Starting from the base cuboid `[student, course, semester, instructor]`, the following OLAP operations should be performed to list the average grade of CS courses for each student:

1. **Selection (Filter)**: Filter the fact table to include only records related to CS courses. This can be done by selecting courses from the `DIM_COURSE` table where the department is CS.
2. **Aggregation (Roll-up)**: Aggregate the data by student, specifically calculating the average grade for each student in CS courses. The aggregation will sum up the `avg_grade` and count the number of courses.
3. **Projection**: Optionally, project the student's details (name, major) from the `DIM_STUDENT` table, and join it with the aggregated result.

12. Explain the three-tier architecture of the data warehouse with a neat figure.

The three-tier architecture of a data warehouse divides the system into three layers: the bottom tier, middle tier, and top tier. Each tier plays a specific role in managing data, facilitating querying, and presenting insights.



1. Bottom Tier (Data Warehouse Server)

- **Role**: This tier is responsible for storing the raw data that will be analyzed.
- **Components**:
  - **RDBMS (Relational Database Management System)**: The core system in the bottom tier is the database server, usually an RDBMS, which stores and manages data.
  - **Data Marts and Metadata Repository**: This layer may include specialized data marts (smaller subsets of the warehouse data for specific purposes) and a metadata

repository (stores information about the data, such as its structure and sources).
    - **Data Extraction**: Data from operational databases and external sources is extracted using application program interfaces (APIs) called **gateways**.

2. Middle Tier (OLAP Server)

- **Role**: This tier handles the querying and processing of data, enabling efficient retrieval and analysis.
- **Components**:
    - **OLAP Server (Online Analytical Processing)**: This server allows fast querying of the data warehouse for analytical purposes.
    - **OLAP Models**:
        - **ROLAP (Relational OLAP)**: Uses a relational DBMS and maps multidimensional operations to relational database operations.
        - **MOLAP (Multidimensional OLAP)**: A specialized server that directly supports multidimensional data and operations, making it faster for some types of queries.

3. Top Tier (Front-End Tools)

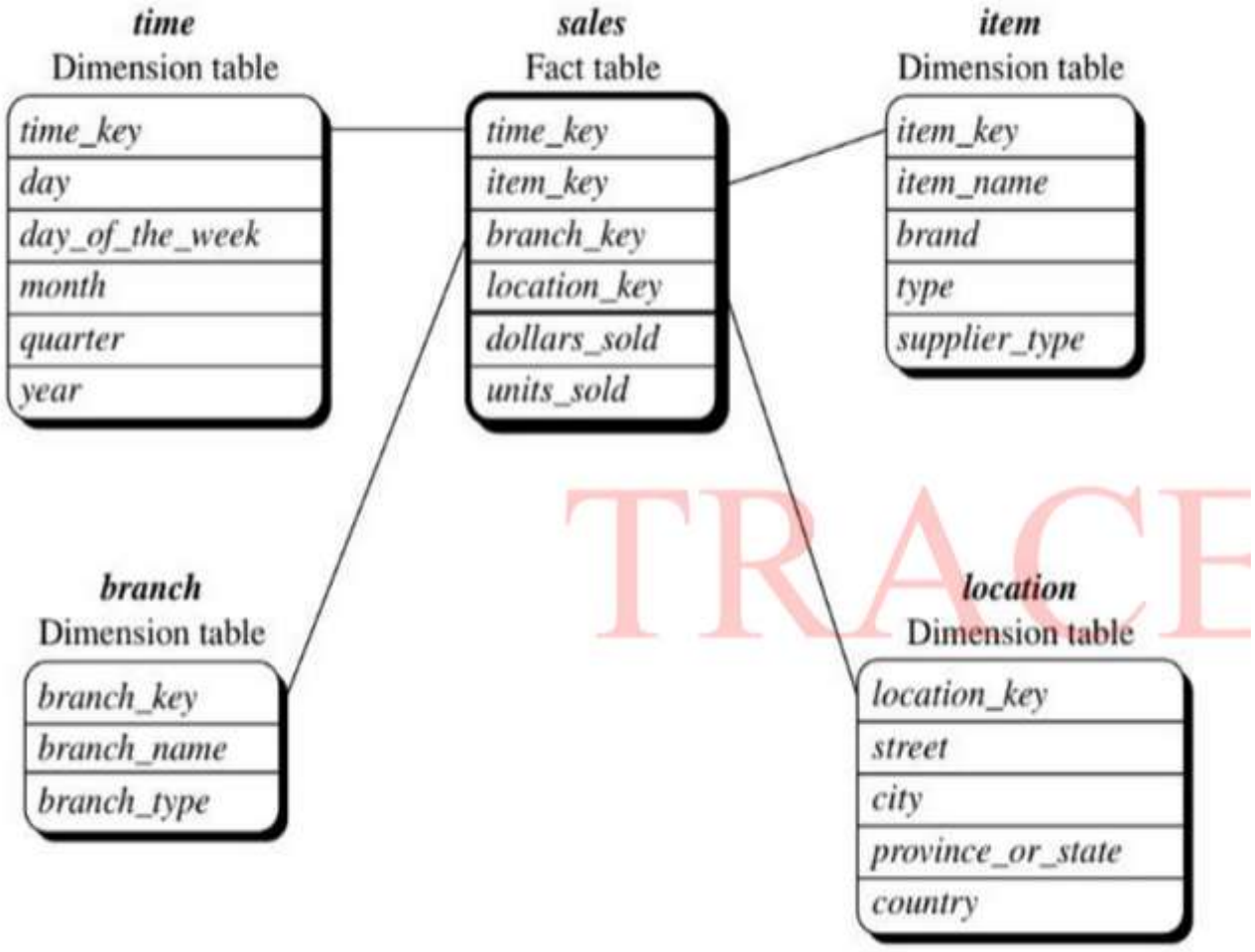
- **Role**: This tier focuses on presenting the data and insights in a way that end users can understand and utilize.
- **Components**:
    - **Query and Reporting Tools**: These tools allow users to run queries and generate reports based on the data stored in the warehouse.
    - **Analysis Tools**: These tools enable deeper analysis of data, including generating visualizations, trends, and other insights.
    - **Data Mining Tools**: These tools apply machine learning or statistical methods to find patterns and insights from the data generated in the OLAP server.

13. List and illustrate the schemas used for the physical representation of the multi-dimensional data with examples.

In data warehousing, data is organized in schemas that help represent and manage multi-dimensional data for efficient querying and analysis. The three primary schemas used for

representing multi-dimensional data are **Star Schema**, **Snowflake Schema**, and **Fact Constellation Schema**. Let's understand these schemas with examples.

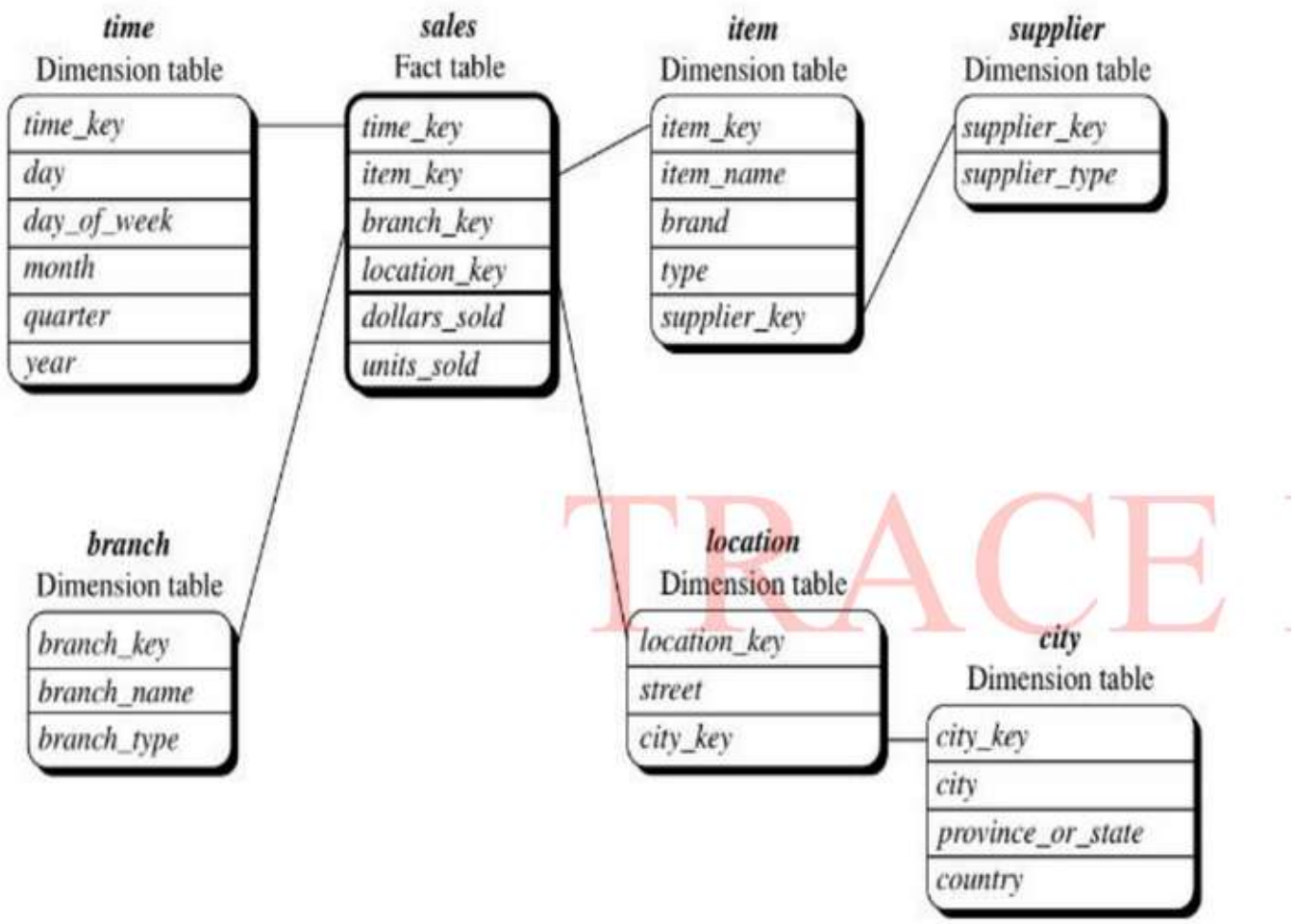
1. Star Schema



- **Structure**: In the **Star Schema**, there is a central fact table surrounded by dimension tables. The schema looks like a star, with the fact table in the middle and dimension tables as "arms" of the star.
- **Fact Table**: The fact table holds the numeric data or metrics that we want to analyze (e.g., sales amounts, units sold). It also contains foreign keys linking to the dimension tables.
- **Dimension Tables**: These are smaller tables surrounding the fact table, each containing descriptive attributes for a specific dimension (e.g., time, item, location).
- **Example**:
  - **Fact Table**: `Sales_Fact`

    - Columns: `time_key`, `item_key`, `branch_key`, `location_key`, `dollars_sold`, `units_sold`
- **Dimension Tables**:
    - `Time`: Columns: `time_key`, `day`, `month`, `year`
    - `Item`: Columns: `item_key`, `item_name`, `brand`, `category`
    - `Branch`: Columns: `branch_key`, `branch_name`, `region`
    - `Location`: Columns: `location_key`, `city`, `state`, `country`
- **Drawback**: Redundancy can occur in dimension tables (e.g., multiple entries for cities with the same state and country), leading to larger storage requirements.

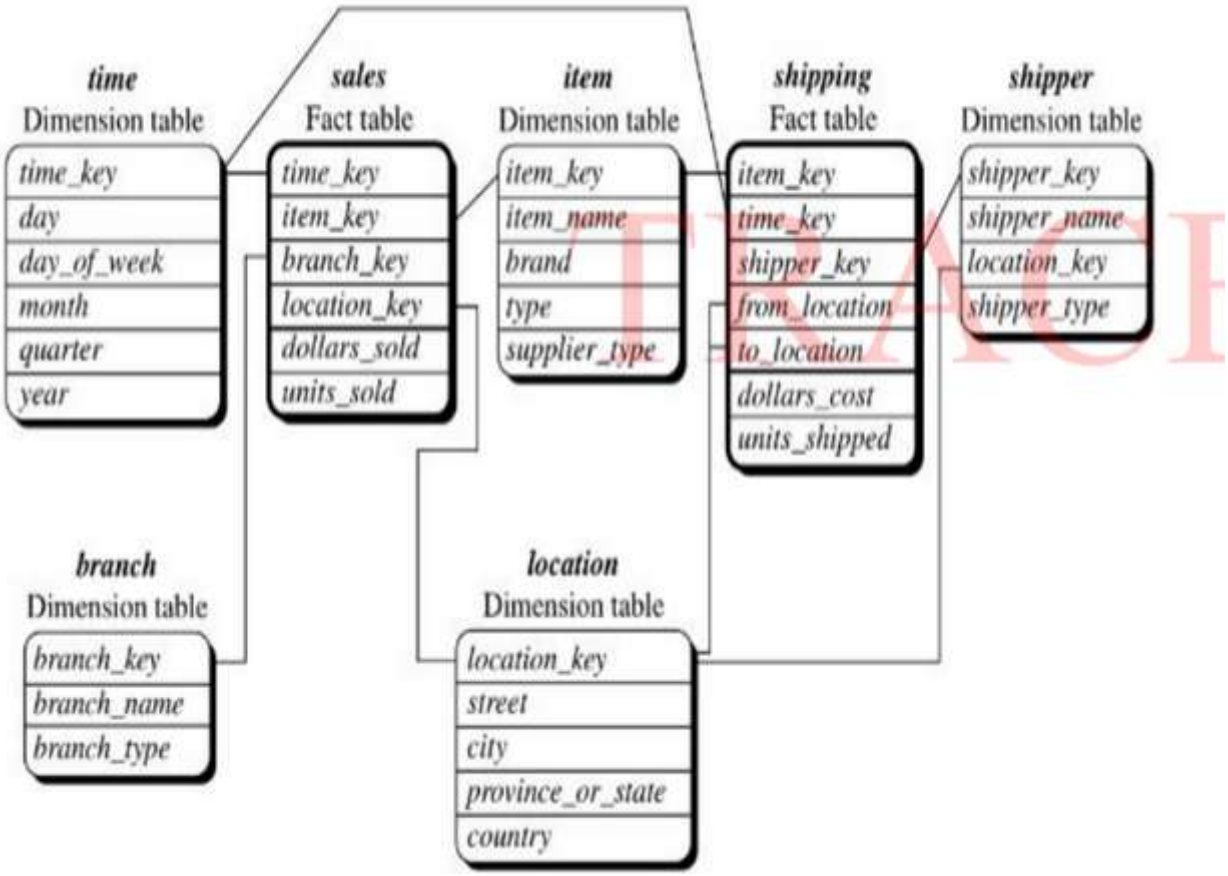
2. Snowflake Schema



- **Structure**: The **Snowflake Schema** is a more normalized version of the star schema. It still has a central fact table, but some dimension tables are broken down into multiple related tables to reduce redundancy.

- **Normalization**: Dimension tables are split into multiple levels of tables to avoid redundancy and save space.
- **Example**:
  - **Fact Table**: `Sales_Fact`
    - Same as in the star schema, containing `time_key`, `item_key`, `branch_key`, `location_key`, `dollars_sold`, `units_sold`.
  - **Normalized Dimension Tables**:
    - **Time Table**: `time_key`, `day`, `month`, `year`
    - **Item Table**: `item_key`, `item_name`, `category`
    - **Supplier Table**: `supplier_key`, `supplier_name`
    - **Location Table**: `location_key`, `city_key`
    - **City Table**: `city_key`, `city_name`, `state_key`
    - **State Table**: `state_key`, `state_name`, `country_key`
    - **Country Table**: `country_key`, `country_name`
  - **Advantage**: It saves storage space due to the normalization of dimension tables.
  - **Drawback**: Since dimension tables are broken down into more levels, it may require more joins when querying the data, leading to performance issues.

3. Fact Constellation Schema (Galaxy Schema)



- **Structure**: The **Fact Constellation Schema** consists of multiple fact tables that share common dimension tables. It can be seen as a collection of star schemas, hence it is also called a **Galaxy Schema**.
- **Fact Tables**: This schema is used for complex applications that require multiple types of analysis, where more than one fact table is necessary (e.g., sales and shipping).
- **Shared Dimension Tables**: Common dimension tables are shared by multiple fact tables.
- **Example**:
  - **Fact Tables**:
    - **Sales_Fact**: Columns: `time_key`, `item_key`, `branch_key`, `location_key`, `dollars_sold`, `units_sold`
    - **Shipping_Fact**: Columns: `time_key`, `item_key`, `shipper_key`, `from_location_key`, `to_location_key`, `dollars_cost`, `units_shipped`
  - **Shared Dimension Tables**:
    - `Time`: Columns: `time_key`, `day`, `month`, `year`
    - `Item`: Columns: `item_key`, `item_name`, `brand`
    - `Branch`: Columns: `branch_key`, `branch_name`

- `Location`: Columns: `location_key`, `city`, `state`, `country`
- `Shipper`: Columns: `shipper_key`, `shipper_name`
- **Advantage**: This schema allows for complex analysis where multiple measures can be compared side by side (e.g., sales and shipping).
- **Drawback**: It can be complex to manage and query due to the larger number of fact tables and shared dimensions.

14. List and explain various data mining functionalities.

1. Data Characterization:

- **What it is**: This involves summarizing the main features or characteristics of a specific group of data.
- **Example**: If you're studying students' grades, data characterization might involve summarizing the average grade, highest grade, and common trends of all students who passed a particular exam.

2. Data Discrimination:

- **What it is**: This involves comparing the features of a target class (group) of data with those of another group (contrasting class).
- **Example**: If you want to compare students who passed an exam with those who failed, you would look at the characteristics (like study hours, attendance) to highlight the differences between these two groups.

3. Classification:

- **What it is**: Classification is the process of sorting data into predefined categories or classes based on their characteristics.
- **Example**: Imagine classifying emails as "Spam" or "Not Spam" based on features like subject line, sender, and content. Classification can be done using algorithms like decision trees or Naive Bayes.
- **Usage**: It's useful for organizing information and predicting unknown data.

4. Prediction:

- **What it is**: Prediction involves forecasting the value of something (like sales or weather) based on existing data.
- **Example**: Predicting next month's sales based on historical sales data or forecasting the weather using temperature patterns from the past.
- **Usage**: It's used for continuous values, like predicting stock prices or house prices.

5. Clustering:

- **What it is**: Clustering groups similar data together, but unlike classification, there are no predefined labels for these groups.
- **Example**: Grouping customers based on their buying behavior, without knowing beforehand how many groups (clusters) there will be. Customers who shop similarly will be grouped together.
- **Usage**: Common in image processing, pattern recognition, and market segmentation.

6. Association Rule Mining:

- **What it is**: This involves finding relationships or patterns between items in a dataset.
- **Example**: A well-known example is **market basket analysis**: "If a customer buys bread, they often buy butter."
- **Usage**: Used to understand the connection between different items and behaviors, helpful in recommendations and marketing strategies.

7. Outlier Analysis:

- **What it is**: This focuses on identifying data points that don’t fit well with the general trends or patterns in the dataset.
- **Example**: A student who scored much higher or lower than all others in an exam could be an outlier. These unusual data points can provide valuable insights or indicate errors in data.
- **Usage**: Important in fraud detection, anomaly detection, and quality control.

8. Evolution Analysis:

- **What it is**: This analyzes how data changes over time to find trends and patterns.
- **Example**: Tracking how the popularity of certain products changes over different seasons or years.

- **Usage**: Helpful in trend analysis, like predicting market demand or understanding long-term customer behavior.