**CS3492 dms 13 mark important Questions**

**Unit 1**

**13 MARKS QUESTIONS**

1.Briefly explain about Database system architecture:

2.Explain about the Purpose of Database system.

3. Briefly explain about Views of data.

4. Explain E-R Model in detail with suitable example.

5. Explain about various data models.

6. Draw an E – R Diagram for Banking, University, Company, Airlines, ATM, Hospital, Library,

Super market, Insurance Company.

**1. Briefly explain about Database System Architecture:**

**Database system architecture** refers to the design or structure of a database system that describes how data is stored, managed, and retrieved. It is mainly divided into **three levels**:

**1.1. Three-Level Architecture (ANSI/SPARC model):**

1. **Internal Level (Physical Level):**
   * Describes **how data is physically stored** in the storage medium.
   * Includes data structures, file systems, indexing, and record placement.
2. **Conceptual Level (Logical Level):**
   * Represents the **entire database content**.
   * Focuses on what data is stored and relationships among the data.
3. **External Level (View Level):**
   * Represents the **user's view** of the database.
   * Multiple users can have different views customized according to their needs.

This architecture provides **data abstraction** and **data independence**, meaning users can interact with the database without knowing internal complexities.

**1.2. Two-Tier and Three-Tier Architecture:**

* **Two-Tier:** Client directly communicates with the server.
* **Three-Tier:** Includes an intermediate **application server** between client and database server for enhanced security and scalability.

**Diagram:**

pgsql

CopyEdit

User View (External Level)

↓

Logical View (Conceptual Level)

↓

Storage View (Internal Level)

**2. Explain about the Purpose of Database System:**

The main purpose of a **Database Management System (DBMS)** is to manage data efficiently and allow users to perform multiple tasks on data easily.

**Key Purposes:**

1. **Data Redundancy Control:**
   * Prevents storing the same data at multiple locations.
2. **Data Consistency:**
   * Ensures that data is accurate and uniform across the database.
3. **Data Sharing:**
   * Allows multiple users and applications to access the data concurrently.
4. **Data Security:**
   * Restricts unauthorized access using user authentication and authorization.
5. **Data Integrity:**
   * Ensures validity and accuracy of the data using constraints.
6. **Data Independence:**
   * Allows changes in database structure without affecting application programs.
7. **Backup and Recovery:**
   * Provides mechanisms to recover data in case of hardware/software failure.
8. **Concurrency Control:**
   * Manages simultaneous data access, ensuring correct transaction execution.

**3. Briefly explain about Views of Data:**

Views of data in DBMS help in understanding the **levels of data abstraction**. These are:

**3.1. External View (User View):**

* Different users see different views of the same data.
* Example: A student can see only their marks, not others'.

**3.2. Conceptual View (Logical View):**

* Provides a community user view.
* Describes what data is stored and the relationships among them.

**3.3. Internal View (Physical View):**

* Describes how data is stored in the database using data structures and file organizations.

**Purpose of Views:**

* Enhance **security** by restricting user access.
* Simplify **complex queries**.
* Provide **data abstraction** and **data independence**.

**4. Explain E-R Model in detail with suitable example:**

The **Entity-Relationship (E-R) Model** is a high-level data model used to define the data elements and their relationships.

**Key Components:**

1. **Entity:**
   * An object that exists and is distinguishable.
   * Example: Student, Employee.
2. **Attributes:**
   * Properties of an entity.
   * Example: Student → Name, Roll\_No, Age.
3. **Entity Set:**
   * A collection of similar entities.
4. **Relationship:**
   * Association among entities.
   * Example: Student ENROLLS in Course.
5. **Types of Attributes:**
   * Simple & Composite
   * Single-valued & Multi-valued
   * Derived Attributes
6. **Types of Relationships:**
   * One-to-One
   * One-to-Many
   * Many-to-Many
7. **E-R Diagram Notations:**
   * Entity: Rectangle
   * Attribute: Ellipse
   * Relationship: Diamond
   * Line: Connects elements

**Example:**

E-R diagram for **Student-Course** Relationship

less

CopyEdit

[Student] ----(Enrolls)---- [Course]

| |

(Name, ID) (Cname, Cid)

**5. Explain about various Data Models:**

**Data models** define how data is connected, stored, and manipulated in a database.

**Types of Data Models:**

1. **Hierarchical Data Model:**
   * Data is organized in a **tree-like structure**.
   * One-to-many relationships.
   * Example: Parent-Child.
2. **Network Data Model:**
   * More flexible than hierarchical.
   * Allows **many-to-many** relationships.
   * Data represented using **graph structure**.
3. **Relational Data Model:**
   * Uses **tables (relations)** to represent data.
   * Based on **first-order predicate logic**.
   * Most widely used model.
   * Example: MySQL, PostgreSQL.
4. **Entity-Relationship Model (E-R):**
   * High-level conceptual model.
   * Uses entities and relationships.
5. **Object-Oriented Data Model:**
   * Combines object-oriented programming concepts with database technology.
   * Supports complex data types, inheritance, encapsulation.
6. **Document Model (used in NoSQL):**
   * Data stored in documents (e.g., JSON, XML).
   * Example: MongoDB.

**6. Draw an E–R Diagram for:**

Below are simplified E-R diagrams for different real-world systems. Use basic ER notations.

**a) Banking System:**

* Entities: Customer, Account, Branch
* Relationships: Owns, Located\_In

css

CopyEdit

[Customer] --- Owns --- [Account] --- Located\_In --- [Branch]

**b) University:**

* Entities: Student, Professor, Course, Department
* Relationships: Enrolls, Teaches, Offers

csharp

CopyEdit

[Student] -- Enrolls --> [Course] <-- Teaches -- [Professor]

|

Offered\_By

|

[Department]

**c) Company:**

* Entities: Employee, Department, Project
* Relationships: Works\_In, Assigned

less

CopyEdit

[Employee] -- Works\_In --> [Department]

|

Assigned

|

[Project]

**d) Airlines:**

* Entities: Passenger, Flight, Airport
* Relationships: Books, Departs\_From, Arrives\_At

less

CopyEdit

[Passenger] -- Books --> [Flight]

/ \

Departs\_From Arrives\_At

| |

[Airport] [Airport]

**e) ATM System:**

* Entities: Customer, Account, ATM
* Relationships: Uses, Has

css

CopyEdit

[Customer] -- Has --> [Account] -- Accessed\_By --> [ATM]

**f) Hospital:**

* Entities: Patient, Doctor, Appointment, Department
* Relationships: Visits, Works\_In

css

CopyEdit

[Patient] -- Visits --> [Doctor] -- Works\_In --> [Department]

**g) Library:**

* Entities: Student, Book, Author
* Relationships: Borrows, Written\_By

css

CopyEdit

[Student] -- Borrows --> [Book] -- Written\_By --> [Author]

**h) Super Market:**

* Entities: Customer, Product, Employee, Bill
* Relationships: Buys, Processes

csharp

CopyEdit

[Customer] -- Buys --> [Product]

|

Pays

|

[Bill] <-- Processes -- [Employee]

**i) Insurance Company:**

* Entities: PolicyHolder, Policy, Agent
* Relationships: Holds, Issued\_By

css

CopyEdit

[PolicyHolder] -- Holds --> [Policy] -- Issued\_By --> [Agent]

**Unit 2**

1.Consider the following relational schema

Employee (empno,name,office,age)

Books(isbn,title,authors,publisher)

Loan(empno, isbn,date)

Write the following queries in relational algebra.

a. Find the names of employees who have borrowed a book Published by McGraw-Hill.

b. Find the names of employees who have borrowed all books Published by McGraw-Hill.

c. Find the names of employees who have borrowed more than five different books published by

McGraw-Hill.

d. For each publisher, find the names of employees who have borrowed more than five books of

that publisher.

2. Explain Embedded and Dynamic SQL.

3.Explain briefly about the steps required in query processing.

4..Explain the three kinds of database tunning.

5.Write about the following

i.Nested loop join

ii.Block Nested loop join

iii. Merge join

iv Hash join

**ANSWERS:**

## **2. Relational Algebra Queries**

Given schema:

* **Employee** (empno, name, office, age)
* **Books** (isbn, title, authors, publisher)
* **Loan** (empno, isbn, date)

### ****a. Find the names of employees who have borrowed a book published by McGraw-Hill.****

**Relational Algebra:**

rust

CopyEdit

R1 ← σ publisher='McGraw-Hill' (Books)

R2 ← Loan ⨝ R1 (Loan.isbn = R1.isbn)

R3 ← R2 ⨝ Employee (R2.empno = Employee.empno)

Result ← π name (R3)

### ****b. Find the names of employees who have borrowed all books published by McGraw-Hill.****

This is a case of **relational division**.

Let:

* **B** = Books published by McGraw-Hill → B ← π isbn (σ publisher='McGraw-Hill' (Books))
* **L** = Employee-ISBN pairs of borrowed books → L ← π empno, isbn (Loan)
* Then perform division:

sql

CopyEdit

Result ← π empno (L ÷ B)

Final ← Result ⨝ Employee

Answer ← π name (Final)

### ****c. Find the names of employees who have borrowed more than five different books published by McGraw-Hill.****

We perform a **grouping and counting operation**.

Steps:

sql

CopyEdit

R1 ← σ publisher='McGraw-Hill' (Books)

R2 ← R1 ⨝ Loan (Books.isbn = Loan.isbn)

R3 ← γ empno; COUNT(isbn)→book\_count (R2)

R4 ← σ book\_count > 5 (R3)

R5 ← R4 ⨝ Employee

Result ← π name (R5)

### ****d. For each publisher, find the names of employees who have borrowed more than five books of that publisher.****

This involves **grouping by empno and publisher**.

scss

CopyEdit

R1 ← Books ⨝ Loan (Books.isbn = Loan.isbn)

R2 ← γ empno, publisher; COUNT(isbn)→book\_count (R1)

R3 ← σ book\_count > 5 (R2)

R4 ← R3 ⨝ Employee

Result ← π publisher, name (R4)

## **3. Explain Embedded and Dynamic SQL**

### ****Embedded SQL:****

* Embedded SQL is used to **integrate SQL queries inside a host language** (like C, Java, etc.).
* SQL statements are precompiled and then used within the code.
* Host variables are declared and used to pass values to/from SQL.

**Example (in C):**

c

CopyEdit

EXEC SQL SELECT name INTO :emp\_name FROM Employee WHERE empno = :emp\_id;

#### ****Advantages:****

* More secure and faster as SQL is precompiled.
* Integration with host language control structures.

### ****Dynamic SQL:****

* Dynamic SQL allows SQL statements to be **constructed and executed at runtime**.
* Useful when queries are not known at compile time.

**Example (in PL/SQL):**

sql

CopyEdit

EXECUTE IMMEDIATE 'SELECT COUNT(\*) FROM Employee WHERE age > ' || var\_age;

#### ****Advantages:****

* Flexibility for building queries dynamically.
* Useful for applications where query structure varies.

## **4. Steps Required in Query Processing**

Query processing refers to the steps taken by a DBMS to execute SQL queries efficiently.

### ****Steps:****

1. **Parsing and Translation:**
   * SQL query is parsed for syntax and semantic correctness.
   * Translated into an **internal representation** (e.g., relational algebra).
2. **Query Optimization:**
   * Multiple equivalent relational algebra expressions are generated.
   * DBMS selects the most **efficient query plan** using statistics and cost estimations.
3. **Evaluation / Execution:**
   * Optimized plan is executed.
   * Operations such as join, selection, projection are performed.
4. **Result Output:**
   * Final results are returned to the user/application.

## **5. Explain the Three Kinds of Database Tuning**

Database tuning improves the performance and efficiency of a database system.

### ****1. Query Tuning:****

* Optimize SQL queries for better performance.
* Techniques:
  + Using proper **indexes**.
  + Avoiding **nested queries**.
  + Selecting only **needed columns**.

### ****2. Index Tuning:****

* Indexing frequently used columns improves access speed.
* Includes:
  + **Creating/deleting indexes** based on query patterns.
  + Using **composite or covering indexes**.

### ****3. Database Design Tuning:****

* Restructuring schema for better performance.
* Includes:
  + **Normalization/Denormalization**.
  + **Partitioning** tables (horizontal/vertical).
  + Reducing **redundancy** while maintaining performance.

## **6. Write about the following Joins**

### ****i. Nested Loop Join:****

* For every tuple in relation R, compare with all tuples in relation S.

**Pseudo-code:**

yaml

CopyEdit

for each tuple r in R:

for each tuple s in S:

if r.A = s.B:

output (r, s)

**Time Complexity:** O(m × n)

### ****ii. Block Nested Loop Join:****

* Improves over nested loop by reading a block of R into memory.

**Pseudo-code:**

yaml

CopyEdit

for each block Br in R:

for each block Bs in S:

for each tuple r in Br:

for each tuple s in Bs:

if r.A = s.B:

output (r, s)

**Time Complexity:** Reduced due to fewer disk I/Os.

### ****iii. Merge Join:****

* Both relations must be **sorted** on join attribute.

**Steps:**

1. Sort R and S.
2. Scan both in parallel and match tuples with equal join values.

**Efficient when:** Inputs are already sorted.

### ****iv. Hash Join:****

* A hash table is built on the smaller relation (usually memory-resident).

**Steps:**

1. Build hash table on smaller relation R using join attribute.
2. Scan larger relation S and probe the hash table.

**Phases:**

* **Build phase:** Hash R into buckets.
* **Probe phase:** Check matching tuples from S.

**Best when:** Data fits in memory.

**UNIT 3**

1. Explain in detail about Lock based protocols and Timestamp based protocols.

2. Write briefly about serializability with example.

3. Explain Two phase locking protocol in detail.

**ANSWER:**

**1. Explain in detail about Lock-based Protocols and Timestamp-based Protocols**

**A. Lock-Based Protocols**

Lock-based protocols ensure **concurrent transaction execution** while maintaining database consistency using **locks**.

**Types of Locks:**

1. **Shared Lock (S-lock):**
   * Acquired when a transaction wants to **read** a data item.
   * Multiple transactions can hold S-locks on the same item simultaneously.
2. **Exclusive Lock (X-lock):**
   * Acquired for **write (update)** operations.
   * Only one transaction can hold an X-lock on a data item.

**Lock Compatibility Matrix:**

|  | **Shared (S)** | **Exclusive (X)** |
| --- | --- | --- |
| **Shared** | Yes | No |
| **Exclusive** | No | No |

**Protocol Rules:**

* A transaction must **acquire a lock** on a data item before accessing it.
* After use, the lock must be **released**.
* Prevents **conflicts** during concurrent access.

**Issues in Locking:**

* **Deadlock**: Two or more transactions waiting for each other’s locks.
* **Starvation**: A transaction never gets the required lock.

**B. Timestamp-Based Protocols**

Timestamp-based protocols order transactions **based on timestamps** assigned at their start.

**Timestamps:**

* Each transaction **Ti** is assigned a unique timestamp **TS(Ti)**.
* Ensures **serializability** by executing transactions in timestamp order.

**Rules:**

For a data item **Q**:

* Maintain:
  + **W-timestamp(Q):** Latest write.
  + **R-timestamp(Q):** Latest read.

**Read(Q) by Ti:**

* If TS(Ti) < W-timestamp(Q) → **Reject & Abort**
* Else → Read allowed, update R-timestamp(Q)

**Write(Q) by Ti:**

* If TS(Ti) < R-timestamp(Q) or TS(Ti) < W-timestamp(Q) → **Abort**
* Else → Write allowed, update W-timestamp(Q)

**Advantages:**

* No deadlocks (no waiting).
* Guarantees serializability.

**Disadvantages:**

* High abort rate.
* Overhead of maintaining timestamps.

**2. Write briefly about Serializability with Example**

**Serializability** is the main criterion for correctness in concurrent transaction execution. A schedule is **serializable** if it results in the same state as some **serial schedule**.

**Types of Serializability:**

1. **Conflict Serializability:**
   * A schedule is conflict serializable if it can be transformed into a serial schedule by **swapping non-conflicting operations**.

**Example:**

Transactions:

css

CopyEdit

T1: R(A), W(A)

T2: R(A), W(A)

Schedule:

css

CopyEdit

S1: R1(A), R2(A), W1(A), W2(A)

* R1(A) and R2(A) → OK
* W1(A) before W2(A) → Conflict
* Cannot swap W1 and W2 without changing result

This schedule is **not conflict serializable**.

1. **View Serializability:**
   * More general form. Based on views (what data is read and written).
   * If final values and reads match a serial schedule, it is view serializable.

**3. Explain Two-Phase Locking (2PL) Protocol in Detail**

The **Two-Phase Locking Protocol (2PL)** ensures conflict serializability by structuring a transaction into two phases:

**Phases:**

1. **Growing Phase:**
   * Transaction may **acquire locks**.
   * No lock is released.
2. **Shrinking Phase:**
   * Transaction may **release locks**.
   * No new locks can be acquired.

**Important Rule:**

* Once a transaction **releases any lock**, it cannot obtain new ones.

**Example:**

Transaction T1:

scss

CopyEdit

Lock(A)

Read(A)

Lock(B)

Write(B)

Unlock(A)

Unlock(B)

This follows 2PL:

* All locks acquired before any are released.

**Types of 2PL:**

1. **Strict 2PL:**
   * All **exclusive (write) locks** are held until transaction **commits or aborts**.
   * Ensures recoverability.
2. **Rigorous 2PL:**
   * **All locks** (shared and exclusive) are held until commit/abort.
   * Provides serializability and recoverability.

**Advantages:**

* Guarantees **conflict serializability**.
* With strict/rugged versions, also ensures **recoverability**.

**Disadvantages:**

* **Deadlocks** can occur.
* Potential for **reduced concurrency** due to conservative locking.

UNIT 4

1.Explain detail in distributed databases and client/server databases.

2.Explain in detail about Dataware housing and data mining

3.Explain in detail about mobile and web databases

**ANSWES:**

**Explain in detail about Distributed Databases and Client/Server Databases**

**A. Distributed Databases**

A **Distributed Database System (DDBS)** is a collection of logically interrelated databases distributed across multiple locations and connected by a network.

**Key Characteristics:**

* Data is **physically stored** at multiple sites.
* Each site is **autonomous** and may run its own DBMS.
* Appears as a **single logical database** to the user.

**Types:**

1. **Homogeneous DDBS**:
   * All sites use the **same DBMS**.
   * Easy to manage.
2. **Heterogeneous DDBS**:
   * Sites may use **different DBMSs**.
   * Requires translation layers.

**Advantages:**

* **Improved reliability** (if one site fails, others work).
* **Local autonomy** for faster processing.
* **Scalability** and **modularity**.

**Challenges:**

* **Complex query processing**
* **Data consistency and synchronization**
* **Increased communication overhead**

**B. Client/Server Databases**

A **Client/Server Database System** consists of two main components:

* **Client**: User interface, sends requests (queries) to server.
* **Server**: Stores database, processes and responds to queries.

**How it works:**

* Clients interact with the database through applications.
* Servers handle database access and operations centrally.

**Advantages:**

* **Centralized control** over the database.
* **Efficient processing** using powerful servers.
* **Reduced data redundancy**

**Drawbacks:**

* **Single point of failure** (server crash).
* **Network dependency**

**5. Explain in detail about Data Warehousing and Data Mining**

**A. Data Warehousing**

A **Data Warehouse** is a **central repository** that stores large volumes of historical data from multiple sources for **analytical and decision-making** purposes.

**Characteristics:**

* **Subject-oriented**: Focuses on specific business areas (sales, marketing, etc.)
* **Integrated**: Combines data from various sources.
* **Time-variant**: Stores historical data.
* **Non-volatile**: Data is read-only; not updated in real-time.

**Architecture:**

* **ETL Process**: Extract, Transform, Load data from source systems.
* **Data Staging Area**
* **Data Warehouse**
* **Data Marts**: Subsets for specific departments.
* **OLAP Tools**: Used for analysis and reporting.

**Advantages:**

* Better decision-making
* Fast query performance
* Trend analysis

**B. Data Mining**

**Data Mining** is the process of **discovering patterns, trends, correlations, and useful information** from large datasets using machine learning, statistics, and database systems.

**Key Techniques:**

* **Classification** (e.g., Spam vs. Non-Spam)
* **Clustering** (e.g., Customer segmentation)
* **Association Rules** (e.g., Market basket analysis)
* **Regression** (e.g., Sales prediction)

**Applications:**

* Fraud detection
* Customer behavior analysis
* Recommendation systems

**Difference Between Data Warehousing and Data Mining:**

| **Data Warehousing** | **Data Mining** |
| --- | --- |
| Stores large volumes of data | Analyzes stored data |
| Mainly for storage and reporting | For pattern discovery |
| Uses ETL and OLAP | Uses AI/ML and statistical models |

**6. Explain in detail about Mobile and Web Databases**

**A. Mobile Databases**

A **Mobile Database** is a database that is accessible from **mobile devices**, either stored locally or accessed through a network.

**Types:**

1. **Local Databases**: Stored on the mobile device (e.g., SQLite).
2. **Remote Databases**: Accessed via internet (cloud or server-based).
3. **Synchronizable Databases**: Allow syncing between local and remote.

**Features:**

* **Portability** and **mobility**
* Support for **offline access**
* Require **sync mechanisms** for consistency

**Challenges:**

* Limited device resources (battery, storage)
* Network latency
* Data synchronization issues

**Use Cases:**

* Mobile apps (e.g., WhatsApp message storage)
* Offline map or note-taking apps

**B. Web Databases**

A **Web Database** is accessible via the internet through **web interfaces or APIs**.

**Characteristics:**

* Stored on **web servers or cloud platforms**
* Accessed using **HTTP**, **APIs**, or **web applications**
* Integrated with **backend** and **frontend** via web technologies

**Components:**

* **Frontend**: HTML/CSS/JS interface
* **Backend**: Server-side logic (PHP, Node.js, Django)
* **Database**: MySQL, MongoDB, PostgreSQL, etc.

**Advantages:**

* Centralized access and updates
* Cross-platform compatibility
* Scalable and secure

**Examples:**

* Online shopping systems
* Social media platforms
* Cloud document storage (e.g., Google Drive)

UNIT -5

1.What is XML? Explain Breifly

2. Explain the concepts of data mining and data warehousing in detail.

**ANSWERS:**

**1. What is XML? Explain Briefly**

**XML (eXtensible Markup Language)**

**XML** is a markup language designed to store and transport data. Unlike HTML, which focuses on **data presentation**, XML focuses on **data representation and structure**.

**Key Features:**

* **Extensible**: You can define your own tags.
* **Structured**: Stores data in a hierarchical (tree-like) format.
* **Human-readable**: Easy to understand and edit.
* **Platform-independent**: Can be used across various systems and applications.

**Example of XML:**

xml

CopyEdit

<Student>

<Name>John Doe</Name>

<RollNo>1023</RollNo>

<Department>CSE</Department>

<Year>3</Year>

</Student>

This represents a student's data in a structured format.

**Uses of XML:**

* **Data exchange** between applications (e.g., Web services).
* **Configuration files** in software.
* **Storage format** for documents (e.g., MS Word, Excel internally use XML).
* **Database representation** in semi-structured databases.

**Advantages:**

* Self-descriptive
* Platform and language independent
* Widely supported by tools and technologies

**Disadvantages:**

* Verbose (more data than needed)
* Slower processing compared to JSON
* Requires proper parsing

**2. Explain the Concepts of Data Mining and Data Warehousing in Detail**

**A. Data Warehousing**

A **Data Warehouse** is a centralized system used to store **large volumes of historical data** from various sources, optimized for **querying and analysis**, not real-time transaction processing.

**Characteristics:**

* **Subject-Oriented**: Organized around key subjects like sales, finance, etc.
* **Integrated**: Combines data from multiple sources (databases, flat files, etc.)
* **Time-Variant**: Maintains historical data.
* **Non-Volatile**: Data is stable; not frequently updated or deleted.

**Architecture:**

1. **Data Source**: Operational databases, flat files, etc.
2. **ETL Process**: Extract, Transform, Load
3. **Data Warehouse Storage**
4. **Data Marts**: Department-specific subsets
5. **OLAP Tools**: For analysis, reporting

**Benefits:**

* Faster and better decision-making
* Unified view of enterprise data
* Trend and pattern analysis

**B. Data Mining**

**Data Mining** is the process of discovering **hidden patterns, relationships, or trends** from large datasets using **statistical, AI, and machine learning** techniques.

**Key Techniques:**

1. **Classification**: Assigns data into predefined categories (e.g., pass/fail).
2. **Clustering**: Groups similar data together (e.g., customer segmentation).
3. **Association Rules**: Finds relationships between variables (e.g., if A → B).
4. **Prediction/Regression**: Forecasts future trends (e.g., sales forecasting).

**Steps in Data Mining:**

1. Data cleaning
2. Data integration
3. Data selection
4. Data transformation
5. Mining
6. Evaluation and presentation

**Applications:**

* Fraud detection in banks
* Customer churn prediction
* Recommendation systems (e.g., Amazon, Netflix)
* Market basket analysis

**Differences between Data Warehouse and Data Mining:**

| **Data Warehousing** | **Data Mining** |
| --- | --- |
| Stores and manages historical data | Analyzes stored data to extract patterns |
| Involves ETL process | Involves pattern recognition, ML |
| Used for reporting, OLAP | Used for prediction, insights |