NEPAL COLLEGE OF INFORMATION TECHNOLOGY

Balkumari, Lalitpur

Affiliated to Pokhara University

ASSIGNMENT FOR DATABASE MANAGEMENT SYSTEM

ASSIGNMENT 1

Submitted by:

Name: Arpan Adhikari

Roll No.: 231309

BE-CE (3rd SEM)

Submitted to:

Er. Simanta Kasaju

Arpan Adhikari (Roll 9)

Assignment 1

Q1) Differentiate between Data and Information. How database is evolved in today's wald?

→

Data	Information
1. set of random and unorganized values, quantities, and figures that do not carry any meanings.	1. Processed and organized form of quantities, values or figures that carry a meaning.
2. First level of knowledge	2. second level of knowledge.
3. Not dependent on information.	3. It is dependent on data.
4. All data may not be useful.	4. All information is useful.
5. It is input of information.	5. It is an output of data.
6. Does not provide any meanings.	6. Provides logical meaning.
7. It is vague in nature	7. It is specific in nature.

Evolution of database:

Databases have evolved to handle more data, work faster, and meet modern needs:

1. Early Days (1950s - 1970s):
Data was stored in simple files or organized in strict tree like structures, which were hard to work with.

2. Relational Databases (1970s - 1990s):
Introduced tables, SQL for easy data queries, and rules to keep data accurate and reliable. widely used in businesses.

3. Data Warehouses (1990s - 2000s):
Built for analyzing large amounts of data, helping with decision-making and reports.

4. NOSQL Databases (2000s - 2010s):
Designed for flexible, fast storage of unstructured or semi-structured data, like social media or IoT data.

5. Big data (2010s - 2020s):
~~Designed~~ Tools like Hadoop and cloud databases emerged to handle huge datasets across many computers.

6. Modern Databases (2020s)
Include AI features, support different types of data (like graphs or documents), and work in real time for applications like streaming or global systems.

Q2) Explain the importance of DBMS. How DBMS help in the field of IT? Also explain how DBMS is accessed using various DDL, DML and DCL languages.

→

Date
Page

Importance of DBMS:

(i) A DBMS management system stores, organizes and manages a large amount of information within a single s/w application. Use of this system increases efficiency of business operations and reduces overall costs.
(ii) DBMS are important to business and organization because they provide a highly efficient method for handling multiple types of data.
(iii) It manages data efficiently and allows users to perform multiple tasks with ease.
(iv) Without DBMS tasks have to be done manually and take more time. Data can be categorized and structured to suit the needs of the company or organization.
(v) Data is entered into the system and accessed on a routine basis by assigned users.

A DBMS is accessed using three main types of languages.

1. DDL (Data Definition Language): Defines or modifies the database structure

commands:
- CREATE (e.g. create a table),
- ALTER (modify table),
- DROP (delete table).

2. DML (Data Manipulation Language):
Handles data operations.

Commands:
- SELECT (retrieve data).
- INSERT (add data)
- UPDATE (modify data)
- DELETE (remove data)

3. DCL (Data Control Language):
Manages permissions and security.
Comands:
- GRANT (give access)
- REVOKE (remove access)

These work together to define, manipulate & secure database operations.

Q3) How can you apply database in Education, Bank, Airlines, Business, Entertainment? Explain in detail.

→

(i) Education
- Student Management : store details of student
- course Management : Track course info
- Library Management : Manage books
- Examination Systems : Record marks, generate reports
- Online Learning Platforms : store content, user progress

Example : Universities use databases like MySQL or Oracle to manage student records and digital learning platforms like Moodle.

(ii) Bank
- customer Management
- Transaction Processing
- Loan & Credit systems
- Fraud Detection
- ATM Operations

Example: Banks use relational databases like Oracle DB to manage millions of transactions securely.

(iii) Airlines
- Reservation systems
- Flight schedules
- Loyalty Programs
- Crew Management
- Baggage Tracking

Example: Airlines use databases like Postgres SQL for real-time booking & logistics management.

(iv) Business
- Inventory Management
- customer Relationship Management
- Sales & Marketing
- Human Resource Management
- Financial Tracking

Example: Businesses use platforms like salesforce (CRM) or SAP HANA for ERP and database needs

(v) Entertainment
- Streaming Platforms
- Ticket Booking
- User profiles
- Content Management
- Analytics

Example: Platforms like Youtube use NoSQL databases such as MongoDB to manage millions of user activities daily.

Q4) Is data redundancy always a problem? How to remove those problems using DBMS?

→

No, data redundancy is not always a problem; it can be beneficial in specific contents, such as backup systems or improving data access in distributed databases. However, in many cases, excessive redundancy can lead to issues such as increased storage costs, data inconsistency, maintenance overhead and reduced performance.

A DBMS solves data redundancy issues through:

1. Normalization: Organizes data into smaller, related tables to eliminate duplication.
2. Constraints: Enforces rules like primary keys and foreign keys to prevent duplicates.
3. Views: Uses virtual tables to avoid storing redundant data.

4. Deduplication : Identifies and removes duplicate records automatically.

5. Replication Control : Ensures consistency across distributed data copies.

These techniques improve efficiency, consistency & maintainability.

Q5) What is data integrity problem? How to remove those problems using DBMS?

→

Data integrity problems happen when the data is incorrect, inconsistent or incomplete. This can be caused by :

(i) Duplicate Data : Having same data stored in multiple places.
(ii) Human Errors : Typing mistakes or deleting important data.
(iii) Broken rules : For example, entering invalid values or missing required links between tables.
(iv) simultaneous changes : Two people editing the same data at the same time.
(v) Unfinished Transactions : If a system crashes, changes might not be saved properly.

Solve with DBMS :

A DBMS solves these problems by :

1. Rules (Constraints):
Primary Key: Makes sure every record is unique.
Foreign Key: Keeps links between tables correct.
Validation Rules: Ensures data follows certain conditions (e.g. age cannot be negative)

2. Transactions
Ensures that changes to the data happen completely or not at all, even if there's a crash.

3. Organized Data (Normalization):
Splits data into related tables to reduce duplication and errors.

4. Access control
Limits who can change or see the data, reducing accidental mistakes.

5. Automatic checks
Runs rules automatically when data is added or changed.

Q6) Differentiate betⁿ file system and DBMS. Any 6.

→

Feature	File System	DBMS
Data Organization	stores data in files, organized based on file names and directories.	stores data in tables, organized based on rows and columns.
Data Relationships	Limited ability to represent relationships between data items.	Can represent complex relationships between data items using primary & foreign keys.
Data Integrity	Provides limited data integrity features.	Offers robust data integrity features like referential integrity, data validation, and concurrency control.
Data security	Provides basic security features like access control & permissions.	Provides advanced security features like encryption, role-based access control, and auditing.
Data Independence	Limited data independence, as data is tightly coupled with application programs.	High degree of data independence, as data is separated from application programs.

Data consistency	May lead to data inconsistencies due to multiple copies of data.	Ensures data consistency through techniques like transaction processing and ACID properties.

(Q7) Differentiate between Data and database.

→

Aspect	Dataset	Database
Basic Definition	A structured collection of data related to a specific topic.	An organized collection of multiple datasets, stored electronically.
Updates	Generally static; does not update or change often.	Dynamic; frequently updated & managed to keep data accurate and accessible.
Data structure	contains specific observations or measurements.	Holds different types of data, allows operations like adding, deleting & updating data.
Data Organization	Data can be unsorted.	Data is well-organised in computerized files.

Functionality	Used for temporary or non-persistent needs, allows creating indexes on the fly.	Provides robust data management, including data protection, minimized inconsistency, and enhanced search speed.
Data Manipulation	Limited manipulation capabilities	Extensive manipulation capabilities; supports complex queries.

Q3) What are the advantages of DBMS over the traditional file-based system?

→ The advantages of DBMS are:

1. Reduces Data Redundancy
   Avoids duplicate data by organizing it efficiently.

2. Ensures Data Consistency
   Keeps Data accurate and synchronized.

3. Improves Security
   Provides access control and encryption.

4. Handles Concurrent Access
   Allows multiple users to work on data without conflicts.

5. Ensures Data Integrity
Enforces rules to maintain data correctness.

6. Offers Backup & Recovery
Automatically protects against data loss.

7. Efficient Querying
Allows fast and flexible data retrieval with SQL.

8. Scalable and shareable
Handles large data volumes and supports multiple users.

Q9) Explain schema vs instances. Any 5 points.

→

Parameters	Schema	Instances
Meaning	schema refers to the overall description of any given database.	Instance basically refers to a collection of data and information that the database stores at any particular moment.
Alterations	The schema remains the same for the entire database as a whole.	One can change the instances of data and information in a database using updation, deletion, and addition.

Frequency of change	It does not change very frequently.	It changes very frequently.
Uses	We use schema for defining the basic structure of any given database. It defines how the available needs to get stored.	We use instance for referring to a set of information at any given instance/time.
Purpose	The schema serves as a blueprint or framework for organizing and structuring data in the database.	The instance represents the actual snapshot of the database's data at a specific moment in time.

Q10) How does Data independence help in DBMS schema? Explain.

→

Data independence is a property of a database that allows the User or Database Administrator to change the schema at one level without affecting the data or schema at another level.

1. Physical Data Independence

We can change the way data is physically stored (like moving data to a new storage device) without affecting how the data is logically organized or how we access it.

- Benefit : This makes it easier to improve storage or change

hardware without disrupting applications or users.

2. Logical Data Independence
we can modify the database's logical structure (such as adding new fields, changing relationships, or adding tables) without changing the way we access the data.

Benefit : We don't need to adjust our queries or applications when changes are made to the database design.

Overall benefit : Data independence allows us to make changes to the database more easily, without disrupting how it is used or requiring changes to applications. This makes the system more adaptable and reduces maintenance costs.

Q11) Differentiate between 1,2,3 tier dbms architecture.
→

1-tier Architecture :
Definition : In a 1-tier architecture, all the components of the system (database, application and user interface) are located on a single machine.
components : The database, application logic, and user interface all run on the same system.
Example : A desktop database application where the database, processing logic, and user interface all exist locally on same computer.

Date
Page

2-tier Architecture :

Definition : In a 2-tier architecture, the system is divided into two layers : the client and server.

Components :

- client : The user interface and application logic are handled on the client side (user's machine).
- server : The database is hosted on a central server.

Example : A client application connects to a centralized database server for data access.

3-tier Architecture :

Definition : In a 3-tier architecture, the system is divided into three layers : the presentation layer, the application layer, and the database layer.

Components :

- Presentation Layer : The user interface that interacts with the user.
- Application Layer : The business logic that processes requests from the user and communicates with the database.
- Database Layer : The database server that stores the data.

Example : A web application where the presentation is handled by the browser, the application logic is processed on a web server, and the database is stored on a separate database server.

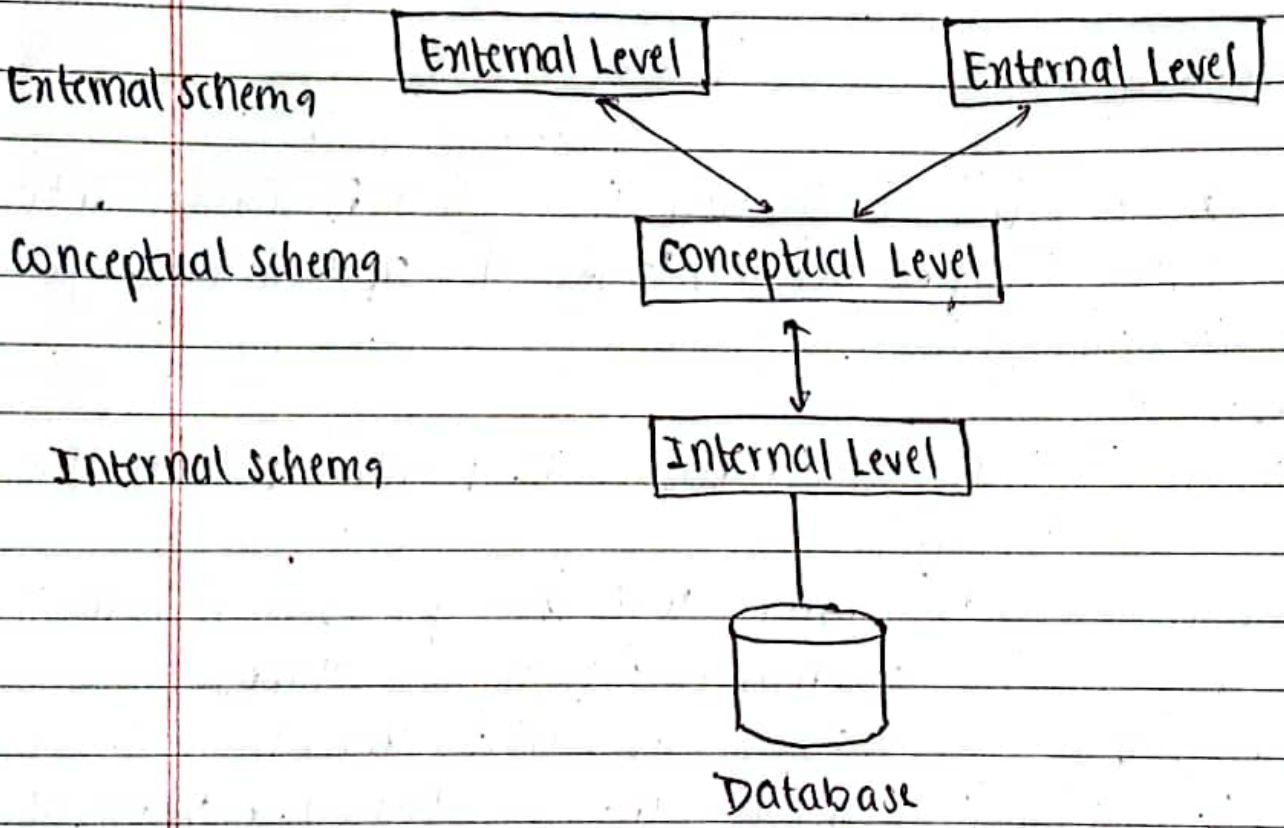
P.T.O.
→

Q12) Why there is multilayer architecture present in DBMS? Describe the 3 schema architectures.

→

Multilayer architecture in DBMS ensures better data abstraction, flexibility, and security. It divides the system into different layers, each responsible for specific functions, making the system more efficient and easier to manage.

3-schema architecture:



1. External Schema (User View)
   - Represents the user's view of database, showing only the relevant data for each user.
   - Purpose: Provides different views for different users to meet their specific needs.

- Example: A salesperson sees customer details, while an admin sees all user accounts

2. conceptual schema (Logical view)
- Describes the structure of the entire database, including data types, relationships and constraints.
- Purpose: Provides a unified view of data across all users.
- Example: Defines the relationships between tables like "Customers", "Orders", and "Products".

3. Internal schema (Physical view)
- Defines how data is physically stored in the database, including file structures and indexing.
- Purpose: Deals with the storage & retrieval of data.
- Example: How a table is stored on disk or the use of B-trees for indexing.

Q13. Describe 3 levels of DBMS (physical, logical, and external).

→

The 3 levels of DBMS are:

1) Physical level:
- Lowest level of data abstraction.
- Describes how data is actually stored physically.
- Details storage structure, file organization.
- Manages data storage mechanisms.

P.T.O. →

2) Logical Level :
- Intermediate level of data representation.
- Describes overall logical structure of database.
- Defines relationships between data entities.
- Independent of physical details.

3) External Level :
- Highest level of data abstraction.
- ~~Describes~~ User specific view of database.
- Provides customized data ~~pre~~ representation.
- Supports multiple user perspectives.

These are three levels of DBMs.

Q14) what is a data model? Explain any 4 different kinds of data models in detail, with characteristics and merits and demerits of data model.

→

A data model is a conceptual representation of how data is organized, structured & relationships are managed within a database system.

Hierarchial Data Models :

characteristics :
a) Tree-like structure
b) Parent-child relationships
c) one-to-many relationships
d) Top-down approach

Merits:
a) simple data representation
b) Fast data retrieval
c) clear data hierarchy

Demerits:
a) Limited flexibility
b) Complex data modifications
c) Rigid structure
d) Difficult to represent many-to-many relationships.

#Network Data Model:

Characteristics:
a) Graph like structure
b) Multiple parent-child relationship
c) complex interconnections
d) supports many-to-many relationships.

Merits:
a) More flexible than hierarchial model
b) Better data representation
c) supports ~~data~~ complex relationships.

Demerits:
a) Complex implementation
b) High maintenance overhead
c) Difficult schema Modifications
d) Increased complexity in data navigation

Relational Data Model:

Characteristics:
a) Table based structure
b) Rows & columns
c) Primary & Foreign key relationships
d) Uses SQL for data manipulation

Merits:
a) Highly flexible
b) Easy data manipulation
c) strong data integrity
d) supports complen queries

Demerits:
a) Performance overhead for complen relationships.
b) scaling challenges
c) Limited support for unstructured data

Object-oriented Data Model:

Characteristics:
a) Based on object-oriented programming.
b) Supports inheritance
c) Better representation of real world entities
d) supports inheritance

Merits:
a) Supports complex data structures
b) Rich semantic modeling
c) Encapsulation of data & ~~model~~ methods.

Demerits:
a) High complexity
b) Performance overhead
c) Lack of standardization
d) steep Learning curve.

Q15) Differentiate bet^n External, logical & physical schema.
→

External schema	Logical schema	Physical schema
This is user-specific view of database.	It is conceptual database structure.	It is actual data ~~d~~ storage representation.
Highest level of abstraction.	Intermediate level of abstraction.	Lowest level of abstraction.
Its purpose is customization data presentation.	It describes the overall database structure.	It defines actual data storage mechanisms.
Its visibility is to end user level.	Database administrator level.	System-level implementation.

External schema	Logical schema	Physical schema
Its main content is user specific views & subsets.	Its content is about complete database structure.	Its content is about storage details & access methods.

Q16) Difference between physical & logical Data Independence. Any 5 points. Also, list the major steps you would take to set up a database for a particular enterprise.

→

Physical Data Independence	Logical Data Independence
Refers to the ability to change the physical storage without affecting the conceptual schema.	Refers to the ability to change the logical schema without affecting the external schema.
Changes in data storage structures (e.g. files, indices) donot affect the application.	Changes in tables, views or relationships donot affect how data is presented to users.
concerned with internal schema changes.	concerned with conceptual schema changes.
Generally easier to achieve.	Relatively harder to achieve compared to physical independence.

Modifying the format of data files or switching to a different storage device.	Adding a new attribute to a table or reorganizing relations without changing user queries.

steps to set up database:

1. Requirement Analysis
   - Identify the enterprise's data needs.
   - Gather requirements from stakeholders and determine the scope.

2. Conceptual Design
   - Design an ER model to define entities, attributes and relationships.

3. Logical Design
   Convert the ER model into logical schema (e.g. relational schema) using normalization principles.

4. Physical Design
   Decide on storage structures, indening, and data partioning based on performance needs.

5. Implementation
   - Choose a DBMS and create the schema.
   - Populate the database with initial data and setup necessary constraints.

6. Testing and validation.
- Test the database for accuracy, consistency & performance
- Validate that it meets the enterprise requirements.

7. Deployment & Maintenance
- Deploy the database and integrate it with enterprise systems
- Regularly monitor and optimize database performance.

Q17) Importance of Data Independence? Also mention their pros & cons.

→

Importance of data independence are:
1) Structural Flexibility
2) Abstraction Levels
3) Performance Optimization
4) Security Enhancement
5) Cost-Effectiveness
6) Types of independences

Pros:
1) Flexibility in database design
2) Abstraction & separation of concerns
3) Enhanced security

Cons:
1) complexity
2) Performance Overhead
3) Learning curve

4) system Limitation
5) Initial Development cost

Q19) Explain the structure of DBMS with the appropriate figure.
→

The structure of DBMS is:

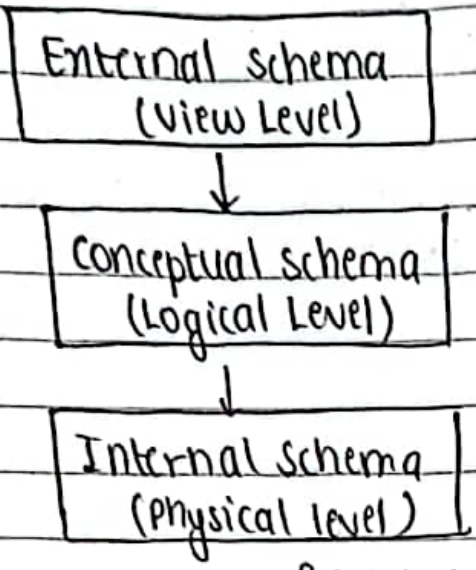


Fig. structure of DBMS

1) External schema (View level)
- User specific view of database
- Customized data representation
- Provides data privacy & security

2) conceptual schema (Logical level)
- Unfield organizational database views
- Describes data relationships & constraints
- Independent of physical storage details

3) Internal schema (Physical level)
- Actual physical storage representation
- Details of data storage & access methods
- Defines storage structures & indexing.

Q18) Data abstraction v/s Data independence

Data Abstraction	Data independence
1) It is the process of hiding complen implementation details.	1) It is ability to modify database structure without impacting applications.
2) It focuses on simplifying data representation.	2) It focuses on ~~mini~~ ensuring flexibility & adaptability to changes in database system.
3) It scopes to represents data at different abstraction levels.	3) It scopes to support modification across database layers
4) Its level are external, conceptual, internal schemas.	4) Its level are physical, & logical independence.
5) It involves schema definitions.	5) It involves architectural design strategies.

Q20)

what are the major responsibilities of DBMS? For each responsibility explain the problem that would arise if the responsibility were not discharged?

→

Major Responsibility with associated problems are:

1) Data storage
Responsibility: Efficiently store & manage data.
if Problem: Data loss, inconsistent storage, inability to retrieve information

2) Data Retrieval
Responsibility: Quick & accurate data access
if Problem: slow query processing, incomplete or incorrect data extraction

3) Data security:
Responsibility: Protect data from unauthorized access
Problem if not discharged: Data breaches, privacy violations, potential information theft.

4) concurrency control
Responsibility: Manage simultaneous user access
if Problem: Data conflict, race conditions, transaction failures.

5) Transaction Management
Responsibility: Ensure reliable transaction processing.
if Problem: Incomplete transactions, data inconsistency, system failures.

6) Backup & Recovery
Responsibility: Protect against data loss
If Problem: Permanent data loss, inability to restore system state

7) Data Abstraction
Responsibility: Provide simplified data representation
If problem: complex data understanding, increased user complexity.

Q21) compare hierarchial, Network & Relational models with an example.

→

Feature	Hierarchial	Network	Relational
structure	Tree-like with parent-child links	Graph with multiple pointers	Tables with rows and columns
Relationships	one-to-many (strict hierarchy)	many-to-many (flexible links)	Any type (via keys)
Flexibility	Low, rigid hierarchy	Medium, requires pointers	High, supports dynamic queries
Ease of use	Moderate, needs predefined paths	Difficult, complex pointers	Easy, SQL-based operations
Data Redundancy	High due to hierarchy duplication	Medium, pointer dependency	Low, normalized structure

Feature	Hierarchical	Network	Relational
Example use case	File systems, org charts	Transport systems, networks	E-commerce, CRM systems

Q22) Explain Database System Architecture and Database Application Architecture with examples and differences.

→

Database System Architecture:

Defines how data is stored, managed, and accessed at three levels:

1. Internal Level: Physical data storage.
2. Conceptual Level: Logical structure of the entire database.
3. External Level: Specific user views of the data.

Example: A university database where
- Physical storage includes tables and indexes.
- Logical structure organizes students, courses, and faculty.
- External views let students see grades or courses.

Database Application Architecture:

Describes how applications interact with the database, focusing on layers:

1. 1-Tier: Database and application on the same system.
2. 2-Tier: Direct client-server interaction.
3. 3-Tier: Separate presentation, logic, and database layers for scalability.

Date
Page

4. Example: An e-commerce system where:
- The browser is the presentation layer.
- The web server handles business logic.
- The database stores product and order data.

Aspect	Database system Architecture	Database Application Architecture
Focus	structure and storage of data	Interaction between apps and DB
Layers	Internal, conceptual, external	Presentation, application, DB
Scope	Database design and management.	Manages application logic and user interaction.
User Interaction	Provides data views based on user roles.	Enables interaction via software or web interfaces.
Example Use case	Designing a database schema	Developing an e-commerce website
Technologies Involved	Database engines, indexing, query processing.	Web servers, middleware, APIs, database drivers

Q23) Explain the object-based data model and semi-structured data model.

→

Object-based data model:

Key characteristics:

a) combines Object oriented-programming concepts with database capabilities.
b) Supports encapsulation, inheritance, & polymorphism.
c) Object have both attributes (data) & methods (behaviour).
d) Maintains complex relationships between objects.

Components:

a) Object
b) classes
c) Relationships

Semi-structured data model:

Key characteristics:

a) Flexible schema
b) self-describing data
c) Irregular or incomplete data support.
d) Hierarchial structure with varying depth
e) Dynamic schema evolution

Components:

a) Data Element
b) Document Structure

Q24) Write short notes on:

a) Distributed and centralised database model

→

Distributed Database Model:

Definition: Data is stored across multiple locations but appears unified.
Advantages: Reliable, faster access for distributed users.
Disadvantages: Complex management, potential network delays.
Example: Banking systems with regional databases.

Centralised Database Model:

Definition: A database system where all data is stored in a single location, typically on one server.
Advantages: Easy to manage, consistent data
Disadvantages: single point of failure, slower for remote servers.
Example: University database on a central server.

b) Data Dictionary

Definition: A centralized repository that stores metadata i.e. information about the structure, relationships, and attributes of data in a database

Key contents:
- Table names, column types, constraints
- Relationships, indexes, and views.
- User permissions and triggers

Functions:
- Ensures consistency and standardization of data.
- Helps database administrators manage and maintain the database.
- supports developers in understanding database structure.

Types:
- Active : Automatically updated by the DBMS.
- Passive : Requires manual updates.

Example Use:
storing details about tables like students, courses, and their attributes in a university database.