

Q1:

- a) File system based data management has several issues:
- Data Redundancy and Inconsistency: Same data stored in multiple files → inconsistencies.
 - Difficulty in Accessing Data: Need to write complex programs to extract info.
 - Lack of Data Isolation: Changes in one file structure affect all dependent programs.
 - Integrity Problems: Constraints (e.g., salary > 0) are hard to enforce.
 - Concurrent Access Anomalies: File systems can't properly handle simultaneous access.
 - Security Problems: Limited access control compared to DBMS.
- b) A Composite Primary Key is a primary key formed by combining two or more attributes to uniquely identify a record. It is used when a single attribute is not sufficient to ensure uniqueness.

Example:

LOAN(BookID, MemberID, LoanDate, ReturnDate)

- A composite primary key could be (BookID, MemberID, LoanDate) since
 - A single BookID is not unique because the same book can be borrowed by many members over time.
 - A single MemberID is not unique because the same member can borrow many books.
 - Combining BookID + MemberID + LoanDate makes each loan transaction unique.
- c) When an application uses DBMS, it performs operations like:
1. Data Definition: Creating tables, schemas (DDL).
 2. Data Manipulation: Insert, Update, Delete, Retrieve (DML).
 3. Transaction Management: Begin, Commit, Rollback(DCL).
 4. Retrieving Data: Retrieving data using SELECT and clauses like WHERE, GROUP BY, HAVING e.t.c (DRL).
 5. Concurrency Control: Ensuring multiple users access data safely.
 6. Authorization & Security: Applying access permissions.
- d) Logical Data Independence is more difficult to achieve.
- Because changes at the conceptual schema (e.g., splitting a table, merging entities, changing relationships) often require changes in user views and application programs.

Example:

- Suppose we change the schema from:
Employee(EmpID, Name, Address)
to:
Employee(EmpID, Name, City, StreetNo)
- Applications using Address directly will break, making logical independence hard to maintain.

e) A superkey is any set of attributes that can uniquely identify a record in a relation, while a key (candidate key) is a minimal superkey, meaning no attribute can be removed without losing its uniqueness. Hence, every key is a superkey, but not every superkey is a key.

Example:

STUDENT(SSN, Name, DOB)

Superkeys: {SSN}, {SSN, Name}, {SSN, DOB}, {SSN, Name, DOB}

Key (minimal): {SSN}

Here, {SSN, Name} is a superkey but not a key (extra attribute included).

Q2:**2. DATABASE CATALOG DESIGN:**

Relation Name	Number of Columns
Members	4
Books	4
IssuedBooks	5

Attribute	Data Type	Constraints	Belongs To
MemberID	INT	PRIMARY KEY, NOT NULL	Members
Name	VARCHAR(100)	NOT NULL	Members
Email	VARCHAR(100)	UNIQUE, NOT NULL	Members
JoinDate	DATE	DEFAULT CURRENT_DATE	Members
BookID	INT	PRIMARY KEY, NOT NULL	Books
Title	VARCHAR(100)	NOT NULL	Books
Author	VARCHAR(100)	NOT NULL	Books
CopiesAvailable	INT	CHECK (CopiesAvailable ≥ 0)	Books
IssueID	INT	PRIMARY KEY, NOT NULL	IssuedBooks
MemberID	INT	FOREIGN KEY → MEMBERS(MemberID)	IssuedBooks
BookID	INT	FOREIGN KEY → BOOKS(BookID)	IssuedBooks
IssueDate	DATE	DEFAULT CURRENT_DATE	IssuedBooks
ReturnDate	DATE	NULL allowed	IssuedBooks