Database Management (4331603) - Winter 2023 Solution

Milav Dabgar

January 18, 2024

Question 1(a) [3 marks]

Define the following terms: a). Data items b). Data dictionary c).Meta data

Solution

Table:

Table 1. Database Terms

Term	Definition
Data Items	Basic units of data that cannot be subdivided further. Individual facts or values stored in database fields
Data Dictionary	Centralized repository containing metadata about database structure, tables, columns, and relationships
Metadata	Data about data that describes structure, constraints, and properties of database elements

- **Data Items**: Smallest unit of named data
- **Data Dictionary**: "System Catalog" storing schema definitions
- **Metadata**: Structural information describing the data

Mnemonic

"DDM - Data Dictionary Manages"

Question 1(b) [4 marks]

Explain disadvantages of File oriented system.

Solution

Table:

Table 2. Disadvantages of File System

Disadvantage	Description
Data Redundancy	Same data stored in multiple files leading to storage waste
Data Inconsistency	Different versions of same data in different files
Data Isolation	Difficulty in accessing data scattered across multiple files
Security Issues	Limited access control and security mechanisms

- **Redundancy**: Duplication of data across files
- **Inconsistency**: Mismatched data due to poor synchronization
- **Isolation**: Lack of standard format hinders data sharing

- **Integrity**: Hard to enforce constraints across files

Mnemonic

"RDIS - Really Difficult Information System"

Question 1(c) [7 marks]

Describe the responsibilities of DBA in detail.

Solution

Table:

Table 3. DBA Responsibilities

Responsibility	Details
Database Design	Creating logical and physical database structures
Security Management	Implementing user access controls and data protection
Performance Monitoring	Optimizing database performance and query execution
Backup & Recovery	Ensuring data safety through regular backups
User Support	Providing technical assistance to database users
System Maintenance	Regular updates, patches, and system optimization

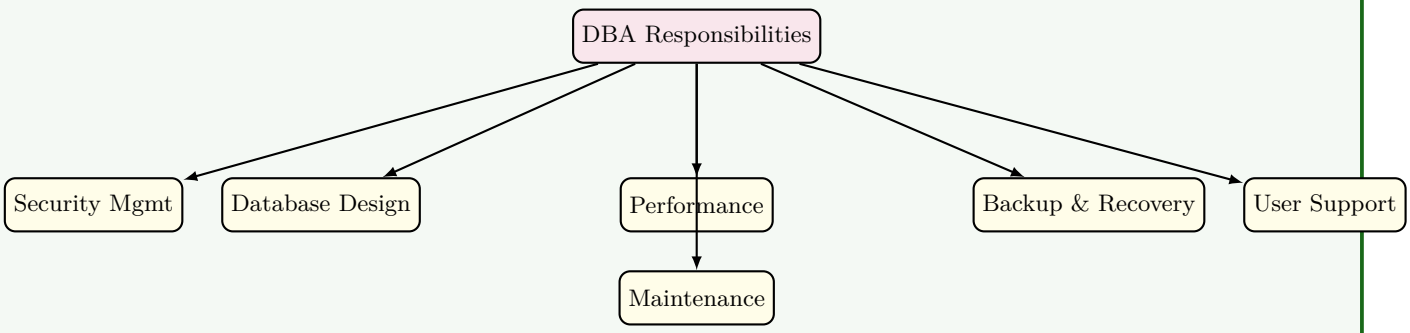

Figure 1. Key Responsibilities of Database Administrator

- **Design**: Schema definition and storage structure planning
- **Security**: Granting/revoking access and encryption
- **Recovery**: Disaster recovery planning
- **Tuning**: Indexing and query optimization

Mnemonic

"DSPBUM - Database Specialists Provide Better User Management"

Question 1(c OR) [7 marks]

Define data abstraction? Explain Three level Architecture of DBMS.

Solution

Data Abstraction: Process of hiding complex implementation details while showing only essential features to users.

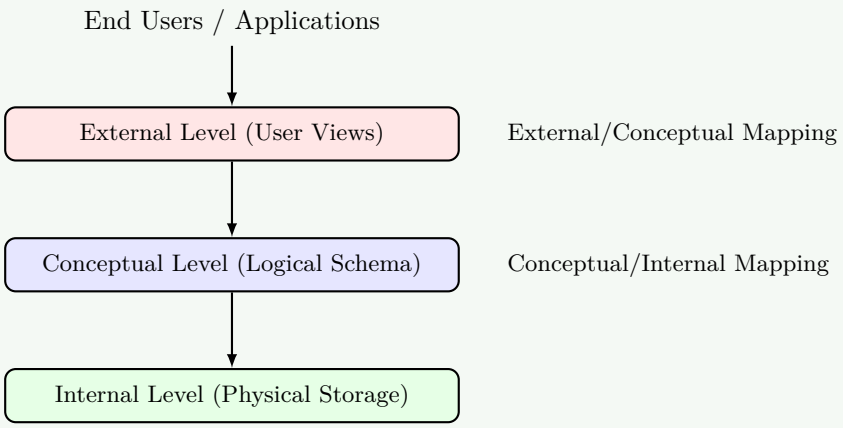

Figure 2. Three Level Architecture

Table:

Table 4. Architecture Levels

Level	Description	Purpose
External Level	User view of database	Individual user perspectives
Conceptual Level	Logical structure of entire database	Overall database organization
Internal Level	Physical storage details	How data is actually stored

- **View Level**: Describes only part of the database
- **Logical Level**: Describes what data is stored and relationships
- **Physical Level**: Describes complex low-level data structures

Mnemonic

"ECI - Every Computer Industry"

Question 2(a) [3 marks]

Define the Following Terms :a).Relationship set b).Participation c).Candidate key

Solution

Table:

Table 5. Terminology

Term	Definition
Relationship Set	Collection of relationships of same type between entity sets
Participation	Constraint specifying whether entity occurrence is mandatory in relationship
Candidate Key	Minimal set of attributes that uniquely identifies each entity in entity set

- **Relationship**: Association among several entities
- **Total Participation**: Every entity must participate (double line)
- **Candidate Key**: Super key with no proper subset being a super key

Mnemonic

"RPC - Relationship Participation Candidate"

Question 2(b) [4 marks]

Explain Generalization with example.

Solution

Generalization: Bottom-up approach where common attributes of lower-level entities are combined into higher-level entity.

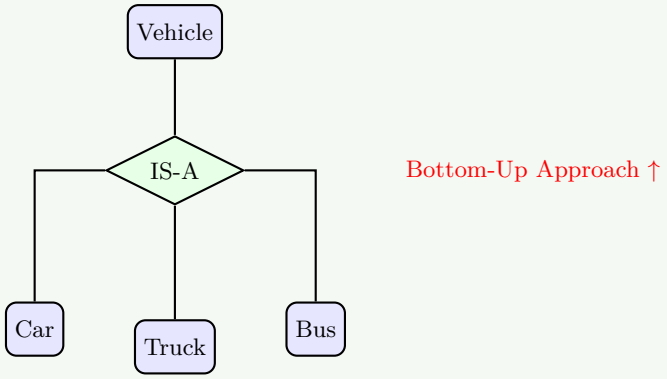

Figure 3. Generalization Example

Table:

Table 6. Generalization Concepts

Concept	Description
Purpose	Reduce redundancy by creating common superclass
Direction	Bottom-up (specific to general)
Example	Car, Truck, Bus → Vehicle

- **Superclass**: Generalized entity containing common properties
- **Subclass**: Specialized entities with unique properties
- **Attribute Inheritance**: Subclasses inherit attributes of superclass

Mnemonic

"GBU - Generalization Builds Up"

Question 2(c) [7 marks]

Define E-R diagram? Explain different symbols used in E-R diagram with example.

Solution

E-R Diagram: Graphical representation showing entities, attributes, and relationships in database design.
Table:

Table 7. ER Diagram Symbols

Symbol	Shape	Usage	Example
Entity	Rectangle	Represents objects	Student, Course
Attribute	Oval	Properties of entities	Name, Age, ID
Relationship	Diamond	Connections between entities	Enrolls, Teaches
Primary Key	Underlined oval	Unique identifier	Student_ID
Multivalued	Double oval	Multiple values	Phone_Numbers
Derived	Dashed oval	Calculated attributes	Age from DOB

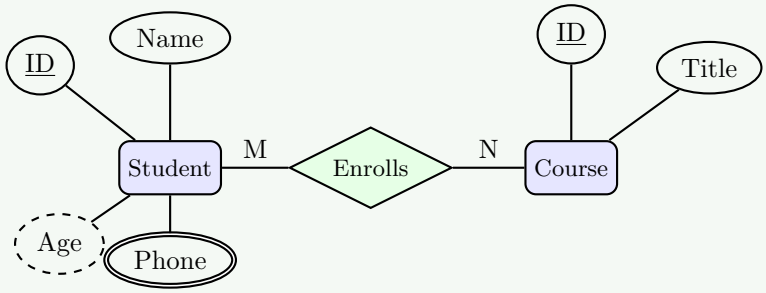

Figure 4. Sample ER Diagram

- **Entity Sets**: Things with independent existence
- **Relationship Sets**: Associations between entities
- **Attribute Types**: Key, Composite, Multivalued, Derived

Mnemonic

"EARPM - Every Attribute Represents Proper Meaning"

Question 2(a OR) [3 marks]

Define Relational Algebra? List out various operations in relational algebra?

Solution

Relational Algebra: Formal query language with operations for manipulating relational database tables.
Table:

Table 8. Relational Algebra Operations

Operation Type	Operations
Basic Operations	Select (σ), Project (π), Union ($\cup$), Set Difference ($-$), Cartesian Product ($\times$)
Additional Operations	Intersection ($\cap$), Join ($\bowtie$), Division ($\div$), Rename (ρ)

- **Procedural Language**: Specifies what to retrieve and how
- **Operators**: Takes relations as input and produces relation as output

Mnemonic

"SPUDC-IJDR - Simple People Use Database Concepts"

Question 2(b OR) [4 marks]

Explain Specialization with example.

Solution

Specialization: Top-down approach where higher-level entity is divided into specialized lower-level entities.

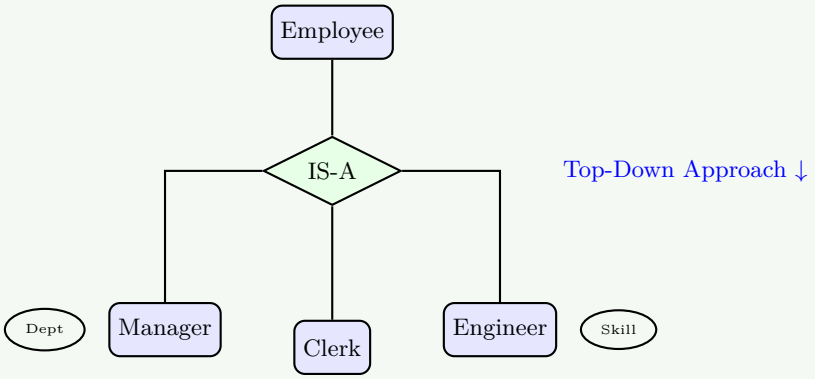

Figure 5. Specialization Example

Table:

Table 9. Specialization Concepts

Concept	Description
Purpose	Create specialized subclasses with unique attributes
Direction	Top-down (general to specific)
Example	Employee → Manager, Clerk, Engineer

- **Subgrouping**: Identifies subsets of entities with distinctive roles
- **Inheritance**: Lower-level entities inherit from higher-level

Mnemonic

"STD - Specialization Top Down"

Question 2(c OR) [7 marks]

Define attribute? Explain different types of attributes with example.

Solution

Attribute: Property or characteristic that describes an entity.
Table:

Table 10. Attribute Types

Attribute Type	Description	Example
Simple	Cannot be divided further	Age, Name
Composite	Can be subdivided	Address (Street, City)
Single-valued	Has one value	SSN, Employee_ID
Multi-valued	Can have multiple values	Phone_Numbers
Derived	Calculated from other attributes	Age from Birth_Date
Key	Uniquely identifies entity	Student_ID

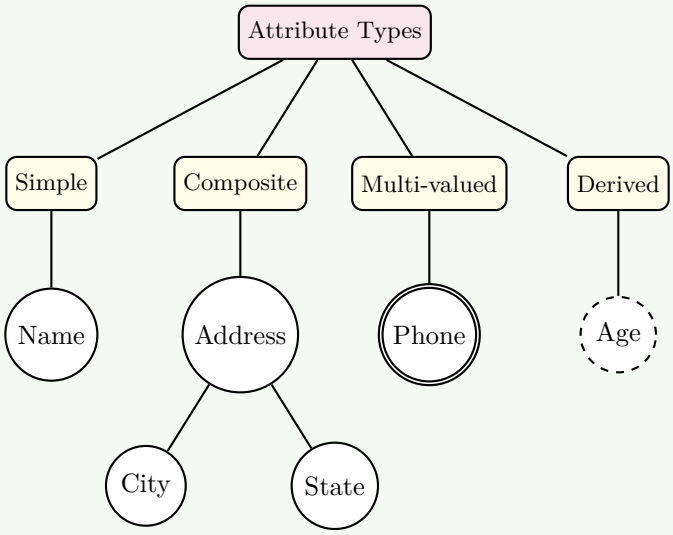

Figure 6. Classification of Attributes

- **Domain**: Set of permitted values for each attribute
- **Null**: Value used when attribute is unknown or not applicable

Mnemonic

"SCSMDK - Simple Composite Single Multi Derived Key"

Question 3(a) [3 marks]

Explain the GRANT and REVOKE statement in SQL.

Solution

Table:

Table 11. DCL Commands

Statement	Purpose	Syntax Example
GRANT	Provides privileges to users	`GRANT SELECT ON table TO user`
REVOKE	Removes privileges from users	`REVOKE SELECT ON table FROM user`

- **Privileges**: Permissions like SELECT, INSERT, UPDATE, DELETE
- **Control**: Manage which users can access or modify data
- **Security**: Fundamental for database security implementation

Mnemonic

"GR - Grant Removes (via REVOKE)"

Question 3(b) [4 marks]

Explain following Character functions. 1) INSTR 2) LENGTH

Solution

Table:

Table 12. SQL Character Functions

Function	Purpose	Syntax	Example
INSTR	Finds position of substring	`INSTR(str, substr)`	`INSTR('Hello', 'e')` → 2
LENGTH	Returns string length	`LENGTH(str)`	`LENGTH('Hello')` → 5

```
SELECT INSTR('Database', 'a') FROM dual; -- Returns 2
SELECT LENGTH('Database') FROM dual;     -- Returns 8
```

- **INSTR**: Case sensitive search for substring position
- **LENGTH**: Counts total characters including spaces

Mnemonic

"IL - INSTR Locates, LENGTH measures"

Question 3(c) [7 marks]

Write SQL statements for following table: Student(Enno,name,branch,sem,clgname,bdate)

Solution

```
-- 1. Create a table Student
CREATE TABLE Student (
    Enno VARCHAR(10) PRIMARY KEY,
    name VARCHAR(50),
    branch VARCHAR(20),
    sem INT,
    clgname VARCHAR(100),
    bdate DATE
);

-- 2. Add a column mobno in Student table
ALTER TABLE Student ADD mobno VARCHAR(15);

-- 3. Insert one record in student table
INSERT INTO Student VALUES
('E001', 'Raj Patel', 'IT', 3, 'GTU College', '2003-05-15', '9876543210');

-- 4. Find out list of students who have enrolled in "IT" branch
SELECT * FROM Student WHERE branch = 'IT';

-- 5. Retrieve all information about student where name begin with 'a'
SELECT * FROM Student WHERE name LIKE 'a%';

-- 6. Count the number of rows in student table
SELECT COUNT(*) FROM Student;

-- 7. Delete all record of student table
DELETE FROM Student;
```

Mnemonic

"CAIRSCD - Create Add Insert Retrieve Search Count Delete"

Question 3(a OR) [3 marks]

Explain equi join with example in SQL.

Solution

Equi Join: Join operation using equality condition to combine tables.
Table:

Table 13. Equi Join

Join Type	Condition	Result
Equi Join	Column1 = Column2	Matching rows from both tables

```sql
-- Example: List student names and their course names
SELECT s.name, c.course_name
FROM Student s, Course c
WHERE s.course_id = c.course_id;
```

- **Operator**: Uses equality operator (=)
- **Columns**: Typically compares primary key and foreign key

Mnemonic

"EE - Equi Equals"

Question 3(b OR) [4 marks]

Explain following Aggregate functions. 1) MAX 2) SUM

Solution

Table:

Table 14. SQL Aggregate Functions

Function	Purpose	Syntax	Example
MAX	Returns maximum value	`MAX(column)`	`MAX(salary)`
SUM	Returns total sum	`SUM(column)`	`SUM(marks)`

```sql
SELECT MAX(salary) FROM Employee; -- e.g., 50000
SELECT SUM(marks) FROM Student;   -- e.g., 450
```

- **Aggregation**: Performs calculation on a set of values to return single value
- **Usage**: Often used with GROUP BY clause

Mnemonic

"MS - MAX Sum"

Question 3(c OR) [7 marks]

Write SQL statements for the following table: Employee(EmpID,Ename,DOB,Dept,Salary)

Solution

```sql
-- 1. Create a table Employee
CREATE TABLE Employee (
    EmpID VARCHAR(10) PRIMARY KEY,
    Ename VARCHAR(50),
    DOB DATE,
    Dept VARCHAR(30),
    Salary DECIMAL(10,2)
);

-- 2. Find sum of salaries of all employee
SELECT SUM(Salary) FROM Employee;

-- 3. Insert one record in Employee table
INSERT INTO Employee VALUES
('E001', 'John Doe', '1990-05-15', 'IT', 35000);

-- 4. Find names of employees who salary between 25000/- and 48000/-
SELECT Ename FROM Employee WHERE Salary BETWEEN 25000 AND 48000;

-- 5. Display detail of all employees in descending order of their DOB
SELECT * FROM Employee ORDER BY DOB DESC;

-- 6. List name of all employees whose name ends with 'a'
SELECT Ename FROM Employee WHERE Ename LIKE '%a';

-- 7. Find highest and least salaries of all employees
SELECT MAX(Salary) AS Highest, MIN(Salary) AS Lowest FROM Employee;
```

Mnemonic

"CSIDDHL - Create Sum Insert Display Display List HighLow"

Question 4(a) [3 marks]

Consider a following relational schema & give Relational Algebra Expressions for the following queries. Student (Enrollment_No,Name,DOB,SPI)

Solution

1. List all students whose SPI is greater than 7.0

$$\sigma_{SPI>7.0}(Student)$$

2. Find Name of student whose Enrollment_No is 007

$$\pi_{Name}(\sigma_{Enrollment\_No=007}(Student))$$

Table:

Table 15. Relational Algebra Symbols

Symbol	Operation	Purpose
σ	Selection	Filter rows based on condition
π	Projection	Select specific columns

- **Selection**: Returns subset of tuples
- **Projection**: Returns subset of attributes

Mnemonic

"SP - Select Project"

Question 4(b) [4 marks]

Write a short note on partial functional dependency.

Solution

Table:

Table 16. Partial Dependency

Concept	Description
Definition	Non-prime attribute depends on part of composite primary key
Occurs in	Tables with composite primary keys
Problem	Causes redundancy and update anomalies
Solution	Decompose into 2NF

Example: In table StudentCourse(StudentID, CourseID, StudentName, CourseName):
- Key: (StudentID, CourseID)
- Dependency: StudentID → StudentName (Partial Key)
- **Composite Key**: Primary key made of multiple attributes
- **Violation**: Breaks 2nd Normal Form rules

Mnemonic

"PDPR - Partial Dependency Problems Resolved"

Question 4(c) [7 marks]

Explain need of Normalization? Discuss about 2NF with example.

Solution

Need for Normalization:
- Eliminates **Data Redundancy** (duplicate data)
- Prevents **Update Anomalies** (inconsistent updates)
- Prevents **Insert Anomalies** (inability to add data)
- Prevents **Delete Anomalies** (accidental data loss)

Second Normal Form (2NF):
1. Must be in 1NF
2. No partial functional dependencies allowed

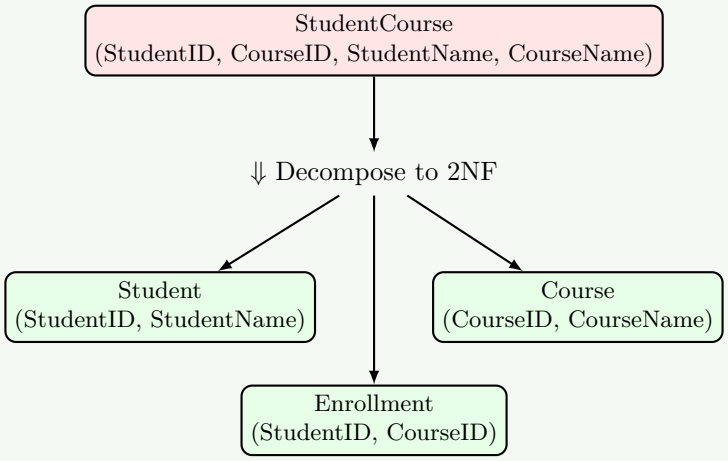

Figure 7. 2NF Decomposition

- **Decomposition**: Splitting table to separate themes
- **Full Dependency**: Non-keys must depend on whole primary key

Mnemonic

"NUID2 - Normalization Unifies Important Data to 2NF"

Question 4(a OR) [3 marks]

Consider a following relational schema & give Relational Algebra Expressions for the following queries. Student(Enno,name,age,address)

Solution

1. Find names of students living in Surat

$$\pi_{name}(\sigma_{address='Surat'}(Student))$$

2. Find names of students older than 30

$$\pi_{name}(\sigma_{age>30}(Student))$$

Table:

Table 17. Query Operations

Query	Algebra Expression
Filter Rows	$\sigma_{condition}(Table)$
Select Columns	$\pi_{columns}(Result)$

- **Nesting**: Operations are nested inside each other
- **Order**: Selection usually performed before projection

Question 4(b OR) [4 marks]

Define 1 NF? Explain 1NF with suitable example.

Solution

First Normal Form (1NF): Database schema where each column contains atomic (indivisible) values, and there are no repeating groups.
Table:

Table 18. 1NF Rules

Rule	Description
Atomic Values	No multiple values in single cell
No Repeating Groups	No duplicate columns for same data
Unique Rows	Each row must be unique

Example:

Before 1NF	After 1NF
ID: 1, Subjects: Math, Science	ID: 1, Subject: Math ID: 1, Subject: Science

- **Atomicity**: Fundamental requirement for relational model
- **Scalability**: Makes querying and indexing efficient

Mnemonic

"ANU - Atomic No-repeat Unique"

Question 4(c OR) [7 marks]

Define Transitive Dependency? Explain 3NF with suitable example.

Solution

Transitive Dependency: Non-prime attribute depends on another non-prime attribute rather than directly on primary key.
Third Normal Form (3NF):
1. Must be in 2NF
2. No transitive dependencies exist

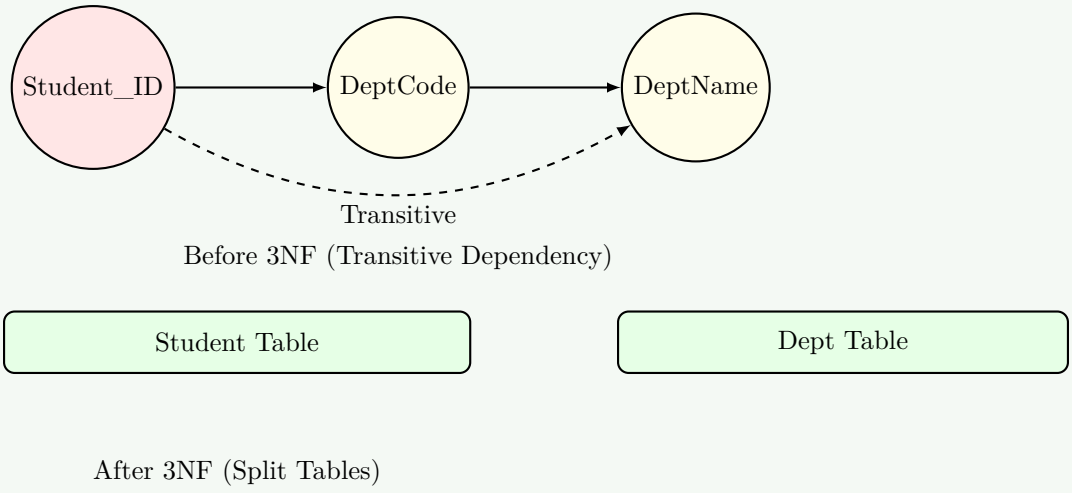

Figure 8. Transitive Dependency & 3NF

- **Indirect Dependency**: A → B → C implies A → C
- **Solution**: Move transitive attributes to new table

Mnemonic

"T3ND - Transitive Third Normal Form No Dependencies"

Question 5(a) [3 marks]

Define Serializability? Explain rules of serializability?

Solution

Serializability: Property ensuring concurrent transaction execution produces same result as serial execution.
Table:

Table 19. Serializability Rules

Rule	Description
Conflict Serializability	Schedule implies same order of conflicting operations as serial schedule
View Serializability	Schedule has same read-write patterns as serial schedule

Example 2: Consider a schedule with two transactions T1 and T2. If T1 reads a value, then T2 writes it, and then T1 writes it, this might not be serializable. A serial schedule means transactions are executed one after another.

- **Consistency**: Ensures database integrity during concurrency

Mnemonic

"SCV - Serial Conflict View"

Question 5(b) [4 marks]

Explain Attributes of Implicit Cursors.

Solution

Table:

Table 20. Implicit Cursor Attributes

Attribute	Description
%FOUND	TRUE if last SQL affected at least one row
%NOTFOUND	TRUE if last SQL affected no rows
%ROWCOUNT	Number of rows affected by last SQL
%ISOPEN	Always FALSE for implicit cursors (automatically closed)

- **SQL%Attribute**: Accessed using SQL prefix
- **Implicit**: Created automatically for DML statements

Mnemonic

"FNRI - Found NotFound RowCount IsOpen"

Question 5(c) [7 marks]

Explain two phase locking protocol with suitable example.

Solution

Two Phase Locking (2PL): Concurrency control protocol that ensures serializability by defining how transactions acquire and release locks.
Table:

Table 21. 2PL Phases

Phase	Description	Rule
Growing Phase	Transaction acquires locks	Cannot release any lock
Lock Point	Point where all locks are acquired	Maximum locks held
Shrinking Phase	Transaction releases locks	Cannot acquire new locks

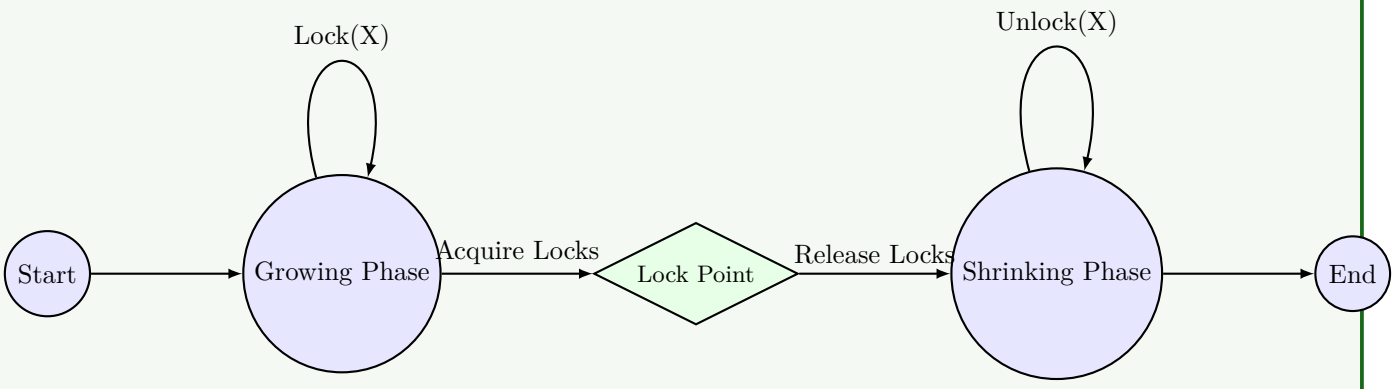

Figure 9. Two Phase Locking Protocol

- **Serializability**: Guaranteed by 2PL
- **Deadlocks**: Possible in 2PL (unlike conservative 2PL)

Mnemonic

"2PGS - Two Phase Growing Shrinking"

Question 5(a OR) [3 marks]

Explain ACID properties of transaction.

Solution

Table:

Table 22. ACID Properties

Property	Description
Atomicity	Transaction is all-or-nothing unit of work
Consistency	Database transitions from one valid state to another
Isolation	Concurrent transactions do not interfere with each other
Durability	Committed changes are permanent despite failures

Mnemonic

"ACID - All Changes In Database"

Question 5(b OR) [4 marks]

Define Triggers? Explain advantages of triggers.

Solution

Triggers: Special stored procedures that automatically execute (fire) in response to specific events on a table or view.
Table:

Table 23. Advantages of Triggers

Advantage	Description
Automatic Execution	Runs automatically without manual invocation
Data Integrity	Enforces complex business rules and constraints
Auditing	Tracks history of data changes (who, when, what)
Security	Controls access and validates data modification

Mnemonic

"ADAS - Automatic Data Auditing Security"

Question 5(c OR) [7 marks]

List down problems of concurrency control. Explain any two with suitable example.

Solution

Problems of Concurrency Control:
1. **Lost Update**: Overwriting uncommitted data
2. **Dirty Read**: Reading uncommitted data
3. **Non-repeatable Read**: Re-reading changed data
4. **Phantom Read**: New rows appearing in range query

Table:

Table 24. Concurrency Problems

Problem	Example Scenario
Lost Update	T1 reads X, T2 reads X. T1 updates X, T2 updates X. T1's update is lost.
Dirty Read	T1 updates X. T2 reads X. T1 fails/rollbacks. T2 has invalid data.

Example 3: Another example for Lost Update: T1: Read A(100) → A=A+50 T2: Read A(100) → A=A+30 T1: Write A(150) T2: Write A(130) **[Overwrites T1's 150!]** Final should be 180, but is 130.

Example 4 (Dirty Read): T1: Update A=200 **[Not Committed]** T2: Read A(200) **[Dirty]** T1: Rollback to 100 T2 uses 200, which never officially existed.

Mnemonic

"LDNP - Lost Dirty Non-repeatable Phantom"