Database Management System

Assignment - UNIT V

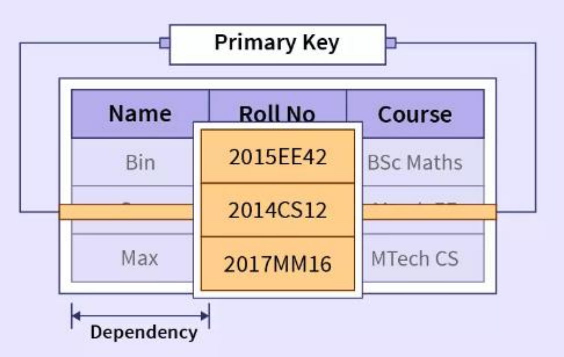
SEM – II

2 Marks Questions

1. **What is Partial Functional dependency? Give example.**

Ans – A Partial Functional Dependency is when a non-prime attribute (not part of a candidate key) depends on part of a candidate key instead of the whole key**.**

Example:



2. **What is Full Functional dependency? Give example.**

Ans - In DBMS, a Full Functional Dependency occurs when a non-prime attribute is functionally dependent on the entire candidate key, not just a part of it.

Example: <ProjectCost>

ProjectID	ProjectCost
001	1000
001	5000

- **<EmployeeProject>**

EmpID	ProjectID	Days
E099	001	320
E056	002	190

3. **What are the conditions for 1NF and 2NF?**

Ans - Conditions for 1NF (First Normal Form):

1. Atomicity: Each column must contain only atomic (indivisible) values.

2. Uniqueness: Each column must contain unique values.

3. No Repeating Groups: There should be no repeating groups or arrays in a table. Each row and column intersection must have a single value.

- **Conditions for 2NF (Second Normal Form):**

1. First Normal Form: The table must already be in 1NF.

2. No Partial Dependencies: All non-prime attributes must be fully functionally dependent on the entire candidate key, not just part of it.

4. What are the conditions for 3NF and BCNF?

Ans - Conditions for 3NF (Third Normal Form):

1. Already in 2NF: The table must be in Second Normal Form.

2. No Transitive Dependencies: Non-key columns should depend only on the primary key, not on other non-key columns.

- **Conditions for BCNF (Boyce-Codd Normal Form):**

1. Already in 3NF: The table must be in Third Normal Form.

2. Superkey Dependency: For every functional dependency X ⟶ Y, X must be a superkey. This means the left side of every dependency must be a candidate key or a superkey.

5. What is transaction in DBMS?

Ans - In DBMS, a transaction is a sequence of one or more SQL operations treated as a single unit of work. It ensures data integrity and consistency, following the ACID properties:

1. Atomicity: All operations within a transaction are completed; if not, the transaction is aborted.

2. Consistency: The database remains in a consistent state before and after the transaction.

3. Isolation: Transactions are executed in isolation from one another.

4. Durability: Once a transaction is committed, its changes are permanent.

Example: A bank transfer operation that involves:.

1. Debiting money from one account. 2. Crediting money to another account.

Both steps must succeed together, or neither should be applied.

10 Mark Questions

6. What is normalization? Explain 1NF, 2NF & 3NF.

Ans - Normalization in DBMS is organizing data to reduce redundancy and improve data integrity. It involves structuring a database in a series of normal forms to eliminate undesirable characteristics like insertion, update, and deletion anomalies.

- **1NF (First Normal Form):**

- Atomic Values: Each column must contain only indivisible values.

- No Repeating Groups: There should be no multiple values in a single column.

- Example:

OrderID	ProductName
1	Pen
2	Pencil

- **2NF (Second Normal Form):**

- Already in 1NF.
- No Partial Dependencies: Non-key columns must depend on the whole primary key, not just part of it.
- **Example:**
  Consider a table with a composite key (OrderID, ProductID):

OrderID	ProductID	ProductName	Quantity
1	101	Pen	10
1	102	Pencil	20

- To satisfy 2NF, split into:

- **Orders Table:**

OrderID	ProductID	Quantity
1	101	10
1	102	20

- **Products Table:**

ProductID	ProductName
101	Pen
102	Pencil

- **3NF (Third Normal Form):**

- Already in 2NF.
- No Transitive Dependencies: Non-key columns must depend directly on the primary key.
- **Example:**
  Consider a table with transitive dependency:

OrderID	CustomerID	CustomerName
1	501	Alice
2	502	Bob

To satisfy 3NF, split into:

- **Orders Table:**

OrderID	CustomerID
1	501
2	502

- **Customers Table:**

CustomerID	CustomerName
501	Alice
502	Bob

- **This ensures non-key columns depend only on the primary key.**

7. **Explain 2NF and 3NF with example.**

Ans - 2NF (Second Normal Form):

1. Deals with partial dependencies.

2. Ensures non-prime attributes depend on the entire primary key.

3. Helps in reducing redundancy and maintaining data integrity.

- **Example:**
  Consider a table with a composite key (OrderID, ProductID):

OrderID	ProductID	ProductName	Quantity
1	101	Pen	10
1	102	Pencil	20

- To satisfy 2NF, split into:

- **Orders Table:**

OrderID	ProductID	Quantity
1	101	10
1	102	20

- **Products Table:**

ProductID	ProductName
101	Pen
102	Pencil

- **3NF (Third Normal Form):**

1. Addresses transitive dependencies.
2. Ensures non-prime attributes depend directly on the primary key.
3. Further reduces redundancy and improves data consistency.
4. Supports flexible database design and efficient data management.

- **Example:**
  Consider a table with transitive dependency:

OrderID	CustomerID	CustomerName
1	501	Alice
2	502	Bob

To satisfy 3NF, split into:

- **Orders Table:**

OrderID	CustomerID
1	501
2	502

- **Customers Table:**

CustomerID	CustomerName
501	Alice
502	Bob

- **This ensures non-key columns depend only on the primary key.**

8. Explain 3NF and BCNF with example.

Ans - 3NF (Third Normal Form):

1. Addresses transitive dependencies.

2. Ensures non-prime attributes depend only on the primary key, not on other non-prime attributes.

3. Supports data integrity by eliminating indirect dependencies.

4. Improves data consistency by reducing redundancy.

Example:

- **Consider a table named "Employee" with the following attributes:**

EmployeeID	DepartmentID	DepartmentName	ManagerID	ManagerName
101	501	Sales	201	John
102	502	Marketing	202	Alice
103	501	Sales	203	Bob

Here, "ManagerName" depends on "ManagerID", which is not the primary key. This creates a transitive dependency since "ManagerName" indirectly depends on "DepartmentID", violating 3NF.

To convert it to 3NF: Employee Table

EmployeeID	DepartmentID	ManagerID
101	501	201
102	502	202
103	501	203

Manager Table:

ManagerID	ManagerName
201	John
202	Alice
203	Bob

- **BCNF (Boyce-Codd Normal Form):**

1. Stricter version of 3NF.

2. Ensures all functional dependencies are directly related to a candidate key.

3. Prevents anomalies by eliminating indirect dependencies.

4. Enhances database design by promoting cleaner and more efficient structures.

Example - Consider a table named "Book" with the following attributes:

ISBN	Title	Author	Genre
123456789	Java Basics	John Smith	Programming
987654321	Python Intro	Alice Jones	Programming
456789123	Art History	Bob Johnson	Art

Here, both "ISBN" and "Title" could serve as candidate keys. "Genre" depends only on "ISBN", while "Author" depends on both "ISBN" and "Title". This creates a dependency that is not directly on a candidate key, violating BCNF.

To convert it to BCNF:

Book Table

ISBN	Title	Genre
123456789	Java Basics	Programming
987654321	Python Intro	Programming
456789123	Art History	Art

Author Table

ISBN	Title	Author
123456789	Java Basics	John Smith
987654321	Python Intro	Alice Jones
456789123	Art History	Bob Johnson

9. What is functional dependency in DBMS? Explain with example.

Ans - In Database Management Systems (DBMS), a functional dependency is a relationship between attributes in a relation (table), where the value of one attribute uniquely determines the value of another attribute.

Example: Consider a table Employees with attributes:

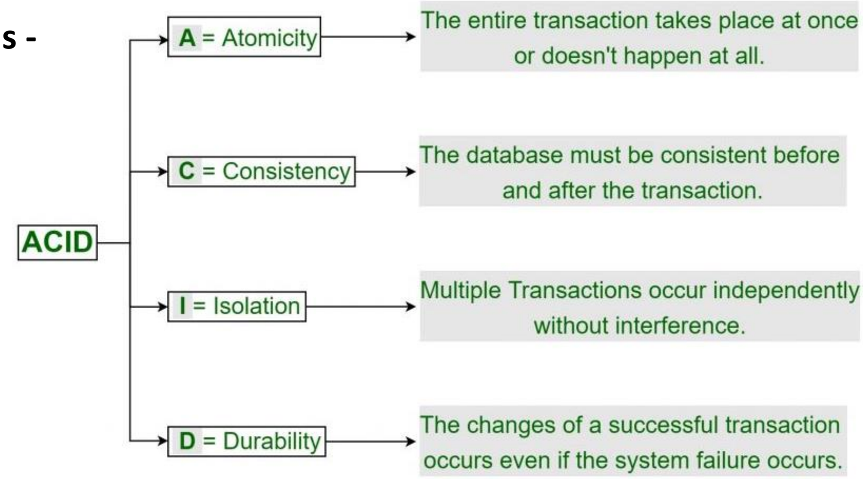
EmployeeID	FirstName	LastName	Department
101	John	Smith	Sales
102	Alice	Jones	Marketing
103	Bob	Johnson	Sales

Here, let's say **EmployeeID** uniquely determines **FirstName** and **LastName**. We can represent this functional dependency as:

- **EmployeeID → FirstName, LastName**

10.Explain ACID properties in DBMS. What are the transaction states?

Ans -



ACID

A = Atomicity: The entire transaction takes place at once or doesn't happen at all.

C = Consistency: The database must be consistent before and after the transaction.

I = Isolation: Multiple Transactions occur independently without interference.

D = Durability: The changes of a successful transaction occurs even if the system failure occurs.

- **Transaction States:**

1. Active: Transaction is currently being executed.

2. Partially Committed: Final operation executed; changes not permanent yet.

3. Committed: Transaction successfully completed; changes permanent.

4. Failed: Transaction encountered an error; cannot be completed.

5. Aborted: Failed transaction rolled back; changes undone.

6. Terminated: Transaction completed or aborted; no longer active.

Disclaimer: **Answers are based on available data and calculations. We strive for accuracy but cannot guarantee it. Users should verify information independently. We are not responsible for any errors or outcomes.**

THANK YOU, CREATED BY SAURABH

All THE BEST