LAB—--------->

### Integrity Constraints

#### 6. Write the query to implement the concept of Integrity constraints.
Q: How do you enforce a primary key constraint in SQL?
A: `CREATE TABLE table_name (column_name datatype PRIMARY KEY);`

Q: How do you enforce a foreign key constraint in SQL?
A: `CREATE TABLE table_name (column_name datatype, FOREIGN KEY (column_name) REFERENCES other_table(other_column));`

### Creating Views

#### 7. Write the query to create the views.
Q: How do you create a view in SQL?
A: `CREATE VIEW view_name AS SELECT column1, column2 FROM table_name WHERE condition;`

### Triggers

#### 8. Perform the queries for triggers.
Q: How do you create a trigger in SQL?
A: `CREATE TRIGGER trigger_name AFTER INSERT ON table_name FOR EACH ROW BEGIN --trigger actions END;`

### Data Manipulation

#### 9. Perform the following operations for demonstrating insertion, updating, and deletion.
Q: How do you insert a record into a table in SQL?
A: `INSERT INTO table_name (column1, column2) VALUES (value1, value2);`

Q: How do you update a record in SQL?
A: `UPDATE table_name SET column1 = value1 WHERE condition;`

Q: How do you delete a record in SQL?
A: `DELETE FROM table_name WHERE condition;`

### Referential Integrity Constraints

#### 10. Using the referential integrity constraints.
Q: What is referential integrity in SQL?
A: It ensures that a foreign key value always points to an existing, valid record in another table.

Q: How do you define a foreign key constraint in SQL?

A: `CREATE TABLE table_name (column_name datatype, FOREIGN KEY (column_name) REFERENCES other_table(other_column));`

### User and Role Management

#### 11. Write the query for creating the users and their roles.
Q: How do you create a user in SQL?
A: `CREATE USER 'username'@'host' IDENTIFIED BY 'password';`

Q: How do you assign a role to a user in SQL?
A: `GRANT role_name TO 'username'@'host';`

LECTURE --------->

### Transaction Processing

#### 1. Introduction-Transaction State
Q: What are the states of a transaction?
A: Active, Partially Committed, Committed, Failed, Aborted.

#### 2. Transaction Properties
Q: What are the ACID properties of a transaction?
A: Atomicity, Consistency, Isolation, Durability.

#### 3. Concurrent Executions
Q: Why is concurrency control necessary?
A: To ensure data consistency and isolation in a multi-user environment.

#### 4. Need of Serializability
Q: Why is serializability important in transactions?
A: It ensures that concurrent transactions result in a database state that would be obtained if the transactions were executed serially.

#### 5. Conflict vs. View Serializability
Q: What is conflict serializability?
A: It ensures that the order of conflicting operations is the same as in some serial order.

Q: What is view serializability?
A: It ensures that the outcome of transactions is the same as in some serial order.

#### 6. Testing for Serializability
Q: How can you test for conflict serializability?
A: By constructing a precedence graph and checking for cycles.

#### 7. Recoverable Schedules
Q: What is a recoverable schedule?
A: A schedule where transactions commit only after all transactions whose changes they read have committed.

#### 8. Cascadeless Schedules
Q: What is a cascadeless schedule?
A: A schedule where transactions read only the committed data to prevent cascading rollbacks.

### Concurrency Control

#### 1. Lock-based Protocols
Q: What is a two-phase locking protocol?
A: A protocol with two phases: growing (acquiring locks) and shrinking (releasing locks).

#### 2. Timestamp-based Protocols
Q: How do timestamp-based protocols ensure serializability?
A: By ordering transactions based on their timestamps.

#### 3. Validation-based Protocols
Q: What are the phases of a validation-based protocol?
A: Read phase, validation phase, and write phase.

#### 4. Deadlock Handling
Q: How can deadlocks be prevented?
A: By using methods like wait-die and wound-wait schemes.

### Database Failure and Recovery

#### 1. Database Failures
Q: What are the common types of database failures?
A: Transaction failure, system crash, and media failure.

#### 2. Recovery Schemes: Shadow Paging
Q: How does shadow paging work?
A: It maintains a shadow copy of the database pages and ensures atomicity by switching between current and shadow pages.

#### 3. Log-based Recovery
Q: What is the purpose of log-based recovery?

A: To record all transaction operations for use in rollback and crash recovery.

#### 4. Recovery with Concurrent Transactions
Q: How is recovery managed with concurrent transactions?
A: Using techniques like write-ahead logging (WAL) and checkpointing.