**3. Steps for Creating JDBC Connections**

**• Theory:**

1. **Step-by-Step Process to Establish a JDBC Connection:- 1. Import the JDBC packages 2. Register the JDBC driver 3. Open a connection to the database 4. Create a statement 5. Execute SQL queries 6. Process the result set 7. Close the connection in java:-**

**1. Import the JDBC Packages:-** Before you can work with JDBC, you need to import the necessary classes from the java.sql package. This includes classes for handling connections, statements, and result sets.

**2. Register the JDBC Driver:-** JDBC drivers facilitate communication between your Java application and the database. You register the driver using Class.forName(), which loads the driver into memory. This step ensures that the Java application can utilize the appropriate database driver.

**3. Open a Connection to the Database:-** After registering the driver, you establish a connection to the database using DriverManager.getConnection(). You need to provide the database URL, username, and password. This connection is crucial for executing SQL commands and retrieving data.

**4. Create a Statement:-** Once connected, you create a Statement object using the connection. The Statement object allows you to send SQL commands to the database. There are different types of statements (e.g., Statement, PreparedStatement, and CallableStatement) based on the use case.

**5. Execute SQL Queries:-** Using the Statement object, you can execute SQL queries. This may involve executing SELECT, INSERT, UPDATE, or DELETE commands. The method used will depend on the type of SQL operation being performed.

**6. Process the Result Set:-** For queries that return results (like SELECT), you will receive a ResultSet object. This object contains the data returned from the database, and you can iterate through it to retrieve individual records. You'll typically extract data using methods like getInt(), getString(), etc.

**7. Close the Connection:-** After completing your operations, it's crucial to close the ResultSet, Statement, and Connection objects. This step releases database resources and avoids memory leaks. It’s a best practice to handle these closures in a finally block to ensure they execute even if an exception occurs.