

14 Lecture - CS506

Important Subjective

Certainly, here are 10 subjective short questions along with their answers related to Java Database Connectivity (JDBC):

****Question 1: What is Java Database Connectivity (JDBC)?****

****Answer:** Java Database Connectivity (JDBC) is a Java API that provides a standard interface for connecting Java applications to relational databases. It allows developers to execute SQL queries, interact with databases, and retrieve results programmatically.**

****Question 2: Explain the steps involved in establishing a database connection using JDBC.****

****Answer:** The steps include loading the appropriate JDBC driver, creating a connection URL with database information, using `DriverManager` to establish a connection, and handling exceptions that might arise during connection establishment.**

****Question 3: What is the role of JDBC drivers in database connectivity?****

****Answer:** JDBC drivers are software components that facilitate communication between Java applications and databases. They translate Java calls into database-specific calls and handle the details of database communication.**

****Question 4: How does JDBC handle SQL queries and updates?****

****Answer:** JDBC allows developers to create `Statement` or `PreparedStatement` objects for executing SQL queries and updates. `Statement` is used for static SQL queries, while `PreparedStatement` is used for parameterized queries, enhancing performance and security.**

****Question 5: What is the significance of the `ResultSet` interface in JDBC?****

****Answer:**** The `ResultSet` interface provides methods to retrieve and process query results obtained from database queries. It allows developers to iterate through rows of data and extract values from columns.

****Question 6:** What are the benefits of using a `PreparedStatement` over a `Statement` in JDBC?

****Answer:**** `PreparedStatement` offers better performance and security by allowing parameterized queries. It pre-compiles the SQL statement, reducing the risk of SQL injection attacks and improving query execution efficiency.

****Question 7:** How does JDBC handle exceptions in database operations?

****Answer:**** JDBC methods may throw exceptions related to database connectivity, SQL syntax, or data retrieval issues. Developers use try-catch blocks to handle these exceptions and ensure graceful error handling.

****Question 8:** What is the role of the `Connection` interface in JDBC?

****Answer:**** The `Connection` interface represents a connection to a database. It provides methods to create `Statement` and `PreparedStatement` objects, manage transactions, and interact with the database.

****Question 9:** Explain the concept of connection pooling in JDBC.

****Answer:**** Connection pooling involves creating and managing a pool of database connections that can be reused by different clients. It improves efficiency and reduces the overhead of establishing a new connection for each database interaction.

****Question 10:** How does JDBC contribute to the separation of concerns in application architecture?

****Answer:**** JDBC separates database interaction logic from the rest of the application, adhering to the principle of separation of concerns. This modular approach enhances code organization, maintainability, and scalability in software development.