**What is JDBC?**

JDBC (Java DataBase Connectivity) is a java API that is used to interact with the database. JDBC has Interfaces and Classes that can be used to connect to database. It uses JDBC drivers to connect to database.

**JDBC Drivers?**

JDBC Drivers are software components that enables java application to interact with the database. There are 4 types of drivers

1. JDBC-ODBC Bridge Driver.

2. Native-API Driver.

3. Network Protocol Driver.

4. Thin Driver.

**JDBC Statements?**

There are 3 types Statements in JDBC

1. Statement

2. Prepared Statement

3. Callable Statement

**Statement:** Statement is used to execute static SQL queries. In this case queries are compiled each time so performance is low. Statement doesn’t offer support for the parameterized SQL queries, which is an important protection from SQL injection attacks.

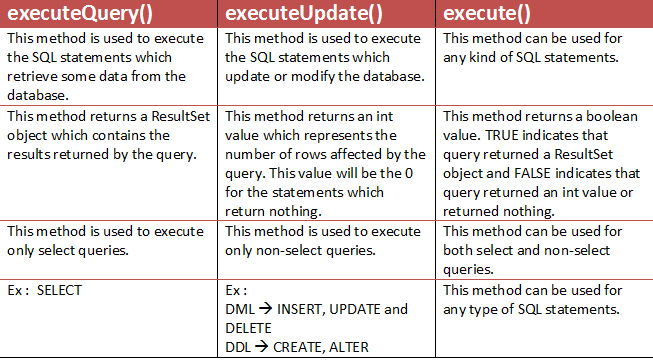
**Prepared Statement:** It is a sub interface of Statement. It is used for the parameterized SQL queries and automatically escapes special characters, which is an important protection from SQL injection attacks. The query is compiled only once so performance is fast.

**Callable Statement:** It is used to call the stored procedures and functions. Performance is fast as these are precompiled.

**Difference between execute, executeUpdate and executeQuery?**

**executeQuery()**, **executeUpdate()** and **execute()** are the methods of java.sql.Statement interface of JDBC API which are used to execute SQL statements.

executeQuery() Vs executeUpdate() Vs execute() In JDBC :



**How do you call stored procedures in JDBC?**

Callable Statement interface is used to call the **stored procedures and functions**.

Creating and using a stored procedure in Java DB involves the following steps:

1. Create the stored procedure with an SQL script or JDBC API

2. Call the stored procedure with the CALL SQL statement.

CallableStatement stmt=con.prepareCall("{call myprocedure(?,?)}");

**How do you manage transactions in JDBC?**

Transaction represents **a single unit of work**.

The ACID properties describe the transaction management well. ACID stands for Atomicity, Consistency, isolation and durability.

**Atomicity** means either all successful or none.

**Consistency** ensures bringing the database from one consistent state to another consistent state.

**Isolation** ensures that transaction is isolated from other transaction.

**Durability** means once a transaction has been committed, it will remain so, even in the event of errors, power loss etc.

### **What is batch processing and how to perform batch processing in JDBC?**

By using batch processing technique in JDBC, we can execute multiple queries. It makes the performance fast. The java.sql.Statement and java.sql.PreparedStatement interfaces provide methods for batch processing.

* JDBC drivers are not required to support this feature. You should use the*DatabaseMetaData.supportsBatchUpdates()* method to determine if the target database supports batch update processing. The method returns true if your JDBC driver supports this feature.
* The **addBatch()** method of *Statement, PreparedStatement,* is used to add queries to batch. The **executeBatch()** is used to start the execution of all the statements grouped together.
* The **executeBatch()** returns an array of integers, and each element of the array represents the update count for the respective update statement.
* Just as you can add statements to a batch for processing, you can remove them with the **clearBatch()**method. This method removes all the statements you added with the addBatch() method. However, you cannot selectively choose which statement to remove.

**What is connection pooling?**

Connection pooling is a technique used for reuse of connections rather than created each time a connection is requested. To facilitate connection reuse, a memory cache of database connections, called a connection pool, is maintained by a connection pooling module as a layer on top of any standard JDBC driver product. So it minimizes expensive operations in creation and closing sessions.

**What is SQL Injection and how do you prevent it?**

SQL injection is a technique where malicious users can inject SQL commands into an SQL statement, via web page input. Injected SQL commands can alter SQL statement and compromise the security of a web application.

Prevention:

1. Use of Prepared Statements (Parameterized Queries)

2. Use of Stored Procedures

3. Form validations.