1. What is JDBC ?

JDBC is an **API** that allows to connect to the database and populate the data to java objects

1. What is the difference between a Statement and a PreparedStatement?

Statement represents the base statements interface. In terms of efficiency, it is suitable to use Statement only when we know that we will not need to execute the SQL query multiple times. Statement doesn’t offer support for the parameterized SQL queries, which is an important protection from SQL injection attacks.

Prepared Statement extends the Statement interface. In most cases it is more efficient (in the context of multiple executions) to use the PreparedStatement because the SQL statement that is sent gets pre-compiled (i.e. a query plan is prepared) in the DBMS. Furthermore, we can use PreparedStatement to safely provide values to the SQL parameters, through a range of setter methods

1. How do you call stored procedures in JDBC

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.

4.How do you manage transactions in JDBC

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

The ACID properties describes 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.

### 5. 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,* and *CallableStatement* is used to add individual statements to the 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.

6.What is connection pooling ?

Connection pooling means that connections are reused 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.

Connection pooling is performed in the background and does not affect how an application is coded; however, the application must use a DataSource object (an object implementing the DataSource interface) to obtain a connection instead of using the DriverManager class.

6.How do you create datasoruces ?

7.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 :

Primary Defenses:

* **Option #1: Use of Prepared Statements (Parameterized Queries)**
* **Option #2: Use of Stored Procedures**
* **Option #3: Escaping all User Supplied Input**

Additional Defenses:

* **Also Enforce: Least Privilege**
* **Also Perform: White List Input Validation**