**B1. What is JDBC?**

Java Database Connectivity (JDBC) is an **application program interface (API)** specification for connecting programs written in Java to the data in popular databases.

**B2. Differences Between JDBC and ODBC.**

JDBC and ODBC, both are the API (Application Programming Interface) that help the applications on the client side to access the database on the server side. ... The point that fundamentally differentiates JDBC and ODBC is that JDBC is language dependent and it is Java specific whereas, the ODBC is a language independent.

**B3. What is JDBC driver?**

**Java Database Connectivity (JDBC)** is an application programming interface (API) for the programming language Java, which defines how a client may access any kind of tabular data, especially relational database. It is part of Java Standard Edition platform, from Oracle Corporation. It acts as a middle layer interface between java applications and database.

The JDBC classes are contained in the Java Package **java.sql** and **javax.sql**.  
JDBC helps you to write Java applications that manage these three programming activities:

**B4. Which is the pure Java driver among the 4 types of JDBC Driver?**

There are 4 types of JDBC drivers: JDBC-ODBC bridge driver. Native-API driver (partially java driver) Network Protocol driver (fully java driver)

**B5. What are the steps involved in creating a JDBC connection?**

The fundamental steps involved in the process of connecting to a database and executing a query consist of the following:

1. Import JDBC packages.
2. Load and register the JDBC driver.
3. Open a connection to the database.
4. Create a statement object to perform a query.
5. Execute the statement object and return a query resultset.

**B6. What is a Driver in JDBC?**

JDBC Driver is a software component that enables java application to interact with the database. There are 4 types of JDBC drivers:

1. JDBC-ODBC bridge driver
2. Native-API driver (partially java driver)
3. Network Protocol driver (fully java driver)
4. Thin driver (fully java driver)

**B7. What is the role of Driver Manager in JDBC?**

The DriverManager provides a basic service for managing a set of JDBC drivers. As part of its initialization, the DriverManager class will attempt to load the driver classes referenced in the "jdbc.drivers" system property. This allows a user to customize the JDBC Drivers used by their applications.

**B8. Explain JDBC statement.**

The Java JDBC Statement, java.sql.Statement , interface is used to execute SQL statements against a relational database. You obtain a JDBC Statement from a JDBC Connection. Once you have a Java Statement instance you can execute either a database query or an database update with it.

**B9. Difference between execute, executeQuery and executeUpdate in JDBC.**

Difference between execute, executeQuery and executeUpdate in JDBC. Execute method can be used with any type of SQL statements and it returns a oolean. … It executes only select statements. executeUpdate method execute sql statements that insert/update/delete data at the database.

**B10. Explain JDBC PreparedStatement.**

PreparedStatement is a class in java.sql package and allows Java programmer to execute SQL queries by using JDBC package. You can get PreparedStatement object by calling connection.prepareStatement() method.SQL queries passed to this method goes to Database for pre-compilation if JDBC driver supports it.

**B11. Why JDBC PreparedStatement is preferred?**

Advantages of PreparedStatement over Statement in JDBC. ... PreparedStatement provides clear separation between the query code and the parameter values that improves readability. PreparedStatement provides convenient way to transform Java object types to SQL Data types to pass input parameters.

**B12. How do I set NULL as an input value for JDBC PreparedStatement?**

Use the setNull method to bind null to the parameter. The setNull method accepts two parameter, index and the sql type as arguments.

PreparedStatement ps = conn.prepareStatement(sqlQuery);

ps.setNull(5, java.sql.Types.INTEGER);

**B13. What are the steps to connect to the database in JDBC?**

Fundamental Steps in JDBC

* Import JDBC packages.
* Load and register the JDBC driver.
* Open a connection to the database.
* Create a statement object to perform a query.
* Execute the statement object and return a query resultset.
* Process the resultset.
* Close the resultset and statement objects.
* Close the connection.

**B14. What are the types of JDBC statements?**

There are 3 types of Statements, as given below:

* Statement: It can be used for general-purpose access to the database. ...
* PreparedStatement: It can be used when you plan to use the same SQL statement many times. ...
* CallableStatement: CallableStatement can be used when you want to access database stored procedures

**B15. How do you execute stored procedures and functions using JDBC?**

How to call stored procedure using JDBC?

JDBC MySQL stored procedure example

Connection conn = DriverManager.getConnection(); Then, prepare a stored procedure call and create a CallableStatement object by calling prepareCall() method of the Connection object. Next, pass all the parameters to the stored procedure.

**B16. What is JDBC ResultSet?**

A ResultSet is a Java object that contains the results of executing an SQL query. In other words, it contains the rows that satisfy the conditions of the query. The data stored in a ResultSet object is retrieved through a set of get methods that allows access to the various columns of the current row.

**B17. What is the return type of Class.forName method in Java?**

Java.lang.Class.forName() Method

The java.lang.Class.forName(String name, boolean initialize, ClassLoader loader) method returns the Class object associated with the class or interface with the given string name, using the given class loader. The specified class loader is used to load the class or interface.