**JDBC Interview Questions**

Q. How many different types of JDBC drivers are present? Discuss them.

Ans. Type 1: JDBC-ODBC Bridge plus ODBC Driver:

The first type of JDBC driver is the JDBC-ODBC Bridge. It is a driver that provides JDBC access to

databases through ODBC drivers. The ODBC driver must be configured on the client for the bridge to

work. This driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DBMS.

Type 2: Native-API partly-Java Driver:

The Native to API driver converts JDBC commands to DBMS-specific native calls. This is much like the restriction of Type 1 drivers. The client must have some binary code loaded on its machine. These drivers do have an advantage over Type 1 drivers because they interface directly with the database.

Type 3: JDBC-Net Pure Java Driver:

The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a databaseindependent network protocol that is sent to a middleware server. This server then translates this DBMSindependent protocol into a DBMS-specific protocol, which is sent to a particular database. The results are then routed back through the middleware server and sent back to the client. This type of solution makes it possible to implement a pure Java client. It also makes it possible to swap databases without affecting the client.

Type 4: Native-Protocol Pur Java Driver

These are pure Java drivers that communicate directly with the vendor's database. They do this by

converting JDBC commands directly into the database engine's native protocol. This driver has no

additional translation or middleware layer, which improves performance tremendously.

Q. What is JDBC? Describe the steps needed to execute a SQL query using JDBC.

Ans. The JDBC is a pure Java API used to execute SQL statements. It provides a set of classes and interfaces that can be used by developers to write database applications.

The steps needed to execute a SQL query using JDBC:

1. Open a connection to the database.

2. Execute a SQL statement.

3. Process the results.

4. Close the connection to the database.

-----------------------------------------JDBC-------------------------------------------------------------

Q. What is JDBC?

Ans. JDBC is a Java API that is used to connect and execute query to the database. JDBC API uses jdbc drivers to connects to the database

Q. Why use JDBC?

Ans. Before JDBC, ODBC API was used to connect and execute query to the database. But ODBC API uses ODBC driver that is written in C language which is platform dependent and unsecured. That is why Sun Microsystem has defined its own API (JDBC API) that uses JDBC driver written in Java language.

Q. What are the steps to connect to database in java?

Ans. 5 Steps to connect to the database in java

There are 5 steps to connect any java application with the database in java using JDBC. They are as follows:

● Register the driver class

● Creating connection

● Creating statement

● Executing queries

● Closing connection

1) Register the driver class

The forName() method of Class class is used to register the driver class. This method is used to dynamically load the driver class.

Syntax of forName() method

1. public static void forName(String className)throws ClassNotFoundException

Example to register the OracleDriver class

1. Class.forName("oracle.jdbc.driver.OracleDriver");

2) Create the connection object

The getConnection() method of DriverManager class is used to establish connection with the database.

Syntax of getConnection() method

1) public static Connection getConnection(String url)throws SQLException

2) public static Connection getConnection(String url,String name,String password)

3)throws SQLException

Example to establish connection with the Oracle database

1. Connection con=DriverManager.getConnection(

2. "jdbc:oracle:thin:@localhost:1521:xe","system","password");

3) Create the Statement object

The createStatement() method of Connection interface is used to create statement. The object of statement is responsible to execute queries with the database.

Syntax of createStatement() method

1. public Statement createStatement()throws SQLException

Example to create the statement object

1. Statement stmt=con.createStatement();

4) Execute the query

The executeQuery() method of Statement interface is used to execute queries to the database. This method returns the object of ResultSet that can be used to get all the records of a table.

Syntax of executeQuery() method

1. public ResultSet executeQuery(String sql)throws SQLException

Example to execute query

ResultSet rs=stmt.executeQuery("select \* from emp");

while(rs.next()){

System.out.println(rs.getInt(1)+" "+rs.getString(2));

}

5) Close the connection object

By closing connection object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

Syntax of close() method

1. public void close()throws SQLException

Example to close connection.

1. con.close();

Q. What is the difference between Statement and PreparedStatement interface?

Ans. PreparedStatement:

The PreparedStatement interface is a subinterface of Statement. It is used to exeucte parameterized query.