

JDBC Interview Questions and Answers

[Is the JDBC-ODBC Bridge multi-threaded?](#)

Answer :: No. The JDBC-ODBC Bridge does not support multi threading. The JDBC-ODBC Bridge uses synchronized methods to serialize all of the calls that it makes to ODBC. Multi-threaded Java programs may use the Bridge, but they won't get the advantages of multi-threading. [Read More](#)

[What is the fastest type of JDBC driver?](#)

Answer :: JDBC Net pure Java Driver is the fastest JDBC driver. JDBC-ODBC Bridge Driver and Network protocol Driver will be slower than Native API Partly Java Driver (the database calls are make at least three translations versus two), and JDBC Net pure Java Driver, drivers are the fastest (only one translation). [Read More](#)

[What are the different JDB drivers available?](#)

Answer :: What are the different JDB drivers available?

There are mainly four type of JDBC drivers available. They are:

Type 1 : JDBC-ODBC Bridge Driver - A JDBC-ODBC bridge provides JDBC API access via one or more ODBC drivers. Note that some ODBC native code and in many cases native database client code must be loaded on each client machine that uses this type of driver. Hence, this kind of driver is generally most appropriate when automatic installation and downloading of a Java technology application is not important. For information on the JDBC-ODBC bridge driver provided by Sun.

Type 2: Native API Partly Java Driver- A native-API partly Java technology-enabled driver converts JDBC calls into calls on the client API for Oracle, Sybase, Informix, DB2, or other DBMS. Note that, like the bridge driver, this style of driver requires that some binary code be loaded on each client machine.

Type 3: Network protocol Driver- A net-protocol fully Java technology-enabled driver translates JDBC API calls into a DBMS-independent net protocol which is then translated to a DBMS protocol by a server. This net server middleware is able to connect all of its Java technology-based clients to many different databases. The specific protocol used depends on the vendor. In general, this is the most flexible JDBC API alternative. It is likely that all vendors of this solution will provide products suitable for Intranet use. In order for these products to also support Internet access they must handle the additional requirements for security, access through firewalls, etc., that the Web imposes. Several vendors are adding JDBC technology-based drivers to their existing database middleware products.

Type 4: JDBC Net pure Java Driver - A native-protocol fully Java technology-enabled driver converts JDBC technology calls into the network protocol used by DBMSs directly. This allows a direct call from the client machine to the DBMS server and is a practical solution for Intranet access. Since many of these protocols are proprietary the database vendors themselves will be the primary source for this style of driver. Several database vendors have these in progress. [Read More](#)

[What is a ResultSet ?](#)

Answer :: A table of data representing a database result set, which is usually generated by executing a statement that queries the database.

A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set. [Read More](#)

[What is a ResultSet ?](#)

Answer :: A table of data representing a database result set, which is usually generated by executing a statement that queries the database.

A ResultSet object maintains a cursor pointing to its current row of data. Initially the cursor is positioned before the first row. The next method moves the cursor to the next row, and because it returns false when there are no more rows in the ResultSet object, it can be used in a while loop to iterate through the result set. [Read More](#)

[What are the steps required to execute a query in JDBC?](#)

Answer :: First we need to create an instance of a JDBC driver or load JDBC drivers, then we need to register this driver with DriverManager class. Then we can open a connection. By using this connection , we can create a statement object and this object will help us to execute the query. [Read More](#)

[What is JDBC Driver ?](#)

Answer :: The JDBC Driver provides vendor-specific implementations of the abstract classes provided by the JDBC API. This driver is used to connect to the database. [Read More](#)

[What is JDBC?](#)

Answer :: JDBC technology is an API (included in both J2SE and J2EE releases) that provides cross-DBMS connectivity to a wide range of SQL databases and access to other tabular data sources, such as spreadsheets or flat files. With a JDBC technology-enabled driver, you can connect all corporate data even in a heterogeneous environment [Read More](#)

[What are the steps in the JDBC connection?](#)

Answer :: While making a JDBC connection we go through the following steps :

Step 1 : Register the database driver by using :

```
Class.forName(" driver classs for that specific database" );
```

Step 2 : Now create a database connection using :

```
Connection con = DriverManager.getConnection(url,username,password);
```

Step 3: Now Create a query using :

```
Statement stmt = Connection.Statement("\select * from TABLE NAME\");
```

Step 4 : Exceute the query :

```
stmt.exceuteUpdate(); .... Read More
```