**Q -01. What is JDBC?**

**ANS =** Java Database Connectivity is unofficially known as JDBC. It is used to perform DB operations in Database from Java application. It supports interaction with any kind of DB like Oracle, MySQL, MS Access, etc.

**Q-02. What are the different types of drivers in JDBC?**

**ANS =** There are 4 different JDBC drivers out there in the market.

Type I: JDBC – ODBC Bridge

Type II: Native API – Half Java Driver

Type III: Network Protocol– Totally Java Driver

Type IV: Thin Driver- Totally Java Driver.

**Type I: JDBC-ODBC Bridge**

JDBC-ODBC bridge is going to behave as an interface between the client and the DB server. The client should put the JDBC-ODBC driver in it. The database ought to support the ODBC driver. If we are not concerned about the driver installation within the client system, we will use this driver.

**Type II: Native API: Half Java Driver**

It is almost like a JDBC-ODBC driver. Rather than an ODBC driver, we are using native API here. Libraries of the client-side database are used.

**Type III: Network Protocol**

It works like a 3-tier approach to access the database. An intermediate server will be used to connect to DB. JDBC method calls send data to an intermediate server then the server will communicate with DB.

**Type IV: Thin Driver**

It is absolutely written in Java. It explicitly converts JDBC method calls into the vendor-specific database protocol. Nowadays, Database merchant itself is providing this type of driver for their customers. So, programmers don’t rely on other sources. It gives higher performance than the other drivers.

**Q-03. Which type of JDBC driver is used by most people?**

**ANS =** Type IV Thin driver is used in most of the applications. It is developed by the database vendor itself so the developers can use it directly without depending on any other sources. It allows for simple and easy development. It gives higher performance than the other drivers.

**Q-04. What are the types of JDBC Architecture?**

**ANS =** JDBC supports 2 kinds of processing models to access the DB

Two-tier Architecture: Here Java programs explicitly connect with DB. We don’t need any mediator like applications server to connect with DB except the JDBC driver. It is also known as a client-server architecture.

Three-tier Architecture: It is totally inverse of two-tier architecture. There will be no explicit communication between the JDBC driver or Java program and Database. An application server is used as a mediator between them. Java program will send the request to an application server, and the server will send it and receive the response to/ from DB.

**Q-05. What are the steps to connect with JDBC?**

**ANS =** There are 6 basic steps to connect with DB in Java.

1.Import package

2.Load driver

3.Establish connection

4.Creation and execution of the statement

5.Retrieve results

6.Close connection