**JDBC Driver**: JDBC driver is a software component which allows java program to connect virtually with any type of database. By default the Java Development Kit (JDK) also called Java Software Development Kit (JSDK), provides JDBC-ODBC bridge driver, which allows any kind of ODBC database.

JDBC drivers are classified into four types as given below:

* + Type1: JDBC-ODBC Bridge Driver (Bridge)
  + Type2: Native-API/partly java driver (Native)
  + Type 3: All Java/Net-protocol driver (middleware)
  + Type 4:All Java/Native-protocol driver (pure)

1. Type 1: JDBC-ODBC Bridge driver:

This converts all the JDBC calls into Open database connectivity (ODBC) calls and forwards them to the ODBC driver. The Java Native Interface(JNI) is used to call the ODBC functions from the JDBC driver. Before its use we need to create a DSN connection.

Advantages:

1. Type 1 driver comes with JDK, it is not required to install externally.
2. The JDBC-ODBC bridge allows access to almost any database.

Disadvantages:

1. Since its uses JNI (Java Native interfaces) to call ODBC, Type 1 drivers are not portable.
2. Every JBDC call goes through the bridge to the ODBC driver, then to the database, and this applies even in the reverse process. They are the slowest of all driver types.
3. This is not suitable for Web applications.
4. Type 2: Native-API/partly java driver

These are written partly in java programming language and partly in native code. So these also use a native API to communicate with a database system. These are database-specific drivers that convert all JDBC calls to database-specific calls.

Advantages:

1. They are good in performance and faster in execution than Type 1.

Disadvantages:

1. Native API must be installed in client system.
2. If we change the database, we have to change the Native API as it is specific to a database.
3. Cannot be used in an applet since an applet cannot load native code.
4. Type 3: All Java/Net-protocol driver (intermediate database server)

It is written purely in java but it requires a middle tier server as it uses a network protocol between java program and the database. This middle tier server converts all JDBC calls into vendor specific databases.

Advantages:

1. Since it uses a middle tier database server, client installation is not required.
2. It satisfies “write once run anywhere” concept because it is written in Java only.
3. It is suitable for web and faster than Type1 and Type2.
4. They are the most efficient among all drivers.

Disadvantages:

1. It requires another middle-tier server application to install and maintain.
2. There is no direct connection to the database.
3. Type4: Native-protocol/all-java driver

These are all java drivers and there is no client installation or configuration. This is pure written in java. So it’s portable and mostly used by java developers. It uses java networking libraries to communicate with database server. There is a direct communication between application and database. No middleware is required.

Advantages:

1. It is installed inside Java Virtual Machine of the client.
2. It provides better performance than Type1 and Type2 drivers.
3. It is completely written in java hence it is platform independent.
4. It is most suitable for web.

Disadvantages:

1. The major disadvantage is that it is database specific driver, requires separate driver for each database.