What Are JDBC Drivers?

**JDBC API cannot** not directly communicate with the database. It uses **JDBC driver** of the database to interact with the database. **JDBC driver** is a software component provided along with the database which is required by the JDBC API to interact with the database. Each database will have its own JDBC driver.

In simple terms, **JDBC drivers** are nothing but the implementations of interfaces provided in the JDBC API **(java.sql and javax.sql packages)** with respect to a particular database. These implementations are bundled in a JAR file and supplied along with the database. These implementations are used by the JDBC API to interact with that database.

Types Of JDBC Drivers :

There are 4 types of JDBC drivers. They are,

1) Type 1 JDBC Driver / JDBC-ODBC Bridge Driver

2) Type 2 JDBC Driver / Native API Driver

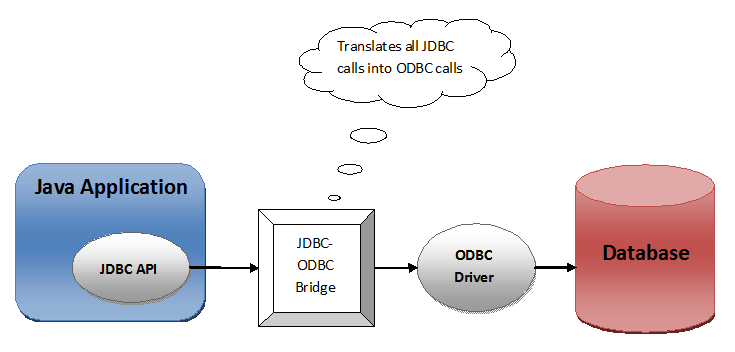
3) Type 3 JDBC Driver / Network Protocol Driver

4) Type 4 JDBC Driver / Native Protocol Driver

1) Type 1 JDBC Driver / JDBC-ODBC Bridge Driver

**Type 1 JDBC Drivers** provide the bridge between JDBC and ODBC API and hence the name **‘JDBC-ODBC Bridge Drivers’.**This type of drivers translate all JDBC calls into ODBC calls and sends them to ODBC driver which interacts with the database. These types of drivers are slowest of all types. Because, all JDBC calls will go to the ODBC driver through the bridge and then to database. So it is time consuming and raises the performance issues. This type of drivers are not recommended for high transaction java applications. And also this driver is not entirely written in java language. It causes the portability issues.

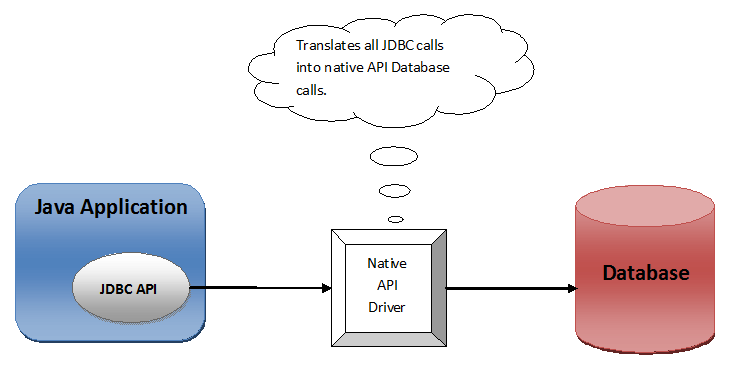
Below diagram shows how JDBC-ODBC bridge driver is used to interact with the database.



2) Type 2 JDBC Driver / Native API Driver

**Type 2 JDBC Driver** translates all JDBC method calls into database specific calls using native API of the database. Its performance is slightly better than the Type 1 driver as communication layer is reduced in this driver. But, like Type 1 Driver, it is also not entirely written in java language. This causes the portability issues. And also this driver is database specific. So once you switch from one database to another, you have to change the driver. That is also one of the disadvantage of this driver.

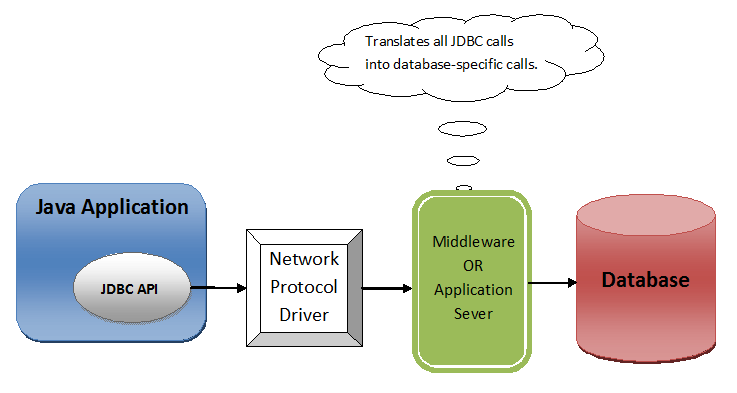
Below diagram shows how Native API Driver works.



3) Type 3 JDBC Drivers / Network Protocol Driver

**Type 3 JDBC Drivers** make use of **middle ware** or **application server** that translates all JDBC calls into database specific calls. One of the main advantage of this driver is that it is entirely written in java language. So no portability issues. But it is costly as extra application server or middle ware component has to be maintained.

Below diagram shows how Network Protocol Driver works.



4) Type 4 JDBC Drivers / Native Protocol Drivers

**Type 4 JDBC Driver** is also called **Thin Driver** as it directly converts JDBC calls into database specific calls. This driver is most popular among all 4 type of JDBC drivers. This driver is preferred over Type 3 Driver as it removes extra layer of communication (Application Server / Middle ware) and this makes it faster than the Type 3 JDBC Driver. And also, like Type 3 JDBC Driver, It is also entirely written in java language and hence portable.

