|  |  |
| --- | --- |
| |  | | --- | | **JDBC Driver Types** | |

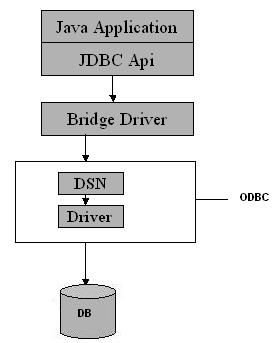
**JDBC drivers** are divided into four types or levels. The **different types of jdbc drivers** are:

**Type 1:** JDBC-ODBC Bridge driver (Bridge)  
**Type 2:** Native-API/partly Java driver (Native)  
**Type 3:** AllJava/Net-protocol driver (Middleware)  
**Type 4:** All Java/Native-protocol driver (Pure)  
  
**4 types of jdbc drivers** are elaborated in detail as shown below:

**Type 1 JDBC Driver**

**JDBC-ODBC Bridge driver**

The Type 1 driver translates all JDBC calls into ODBC calls and sends them to the ODBC driver. ODBC is a generic API. The JDBC-ODBC Bridge driver is recommended only for experimental use or when no other alternative is available.



Type 1: JDBC-ODBC Bridge

Advantage

The JDBC-ODBC Bridge allows access to almost any database, since the database's ODBC drivers are already available.

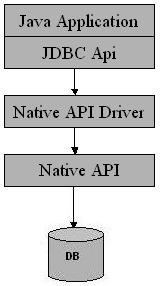
Disadvantages

1. Since the Bridge driver is not written fully in Java, Type 1 drivers are not portable.  
2. A performance issue is seen as a JDBC 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. The client system requires the ODBC Installation to use the driver.  
4. Not good for the Web.

**Type 2 JDBC Driver**

**Native-API/partly Java driver**

The distinctive characteristic of type 2 jdbc drivers are that Type 2 drivers convert JDBC calls into database-specific calls i.e. this driver is specific to a particular database. Some distinctive characteristic of type 2 jdbc drivers are shown below. Example: Oracle will have oracle native api.



Type 2: Native api/ Partly Java Driver

Advantage

The distinctive characteristic of type 2 jdbc drivers are that they are typically offer better performance than the JDBC-ODBC Bridge as the layers of communication (tiers) are less than that of Type  
1 and also it uses Native api which is Database specific.

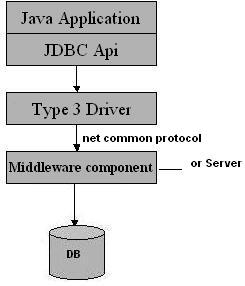
Disadvantage

1. Native API must be installed in the Client System and hence type 2 drivers cannot be used for the Internet.   
2. Like Type 1 drivers, it’s not written in Java Language which forms a portability issue.   
3. If we change the Database we have to change the native api as it is specific to a database  
4. Mostly obsolete now  
5. Usually not thread safe.

**Type 3 JDBC Driver**

**All Java/Net-protocol driver**

Type 3 database requests are passed through the network to the middle-tier server. The middle-tier then translates the request to the database. If the middle-tier server can in turn use Type1, Type 2 or Type 4 drivers.



Type 3: All Java/ Net-Protocol Driver

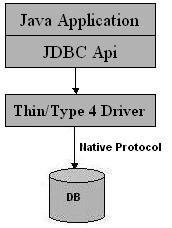
Advantage

1. This driver is server-based, so there is no need for any vendor database library to be present on client machines.  
2. This driver is fully written in Java and hence Portable. It is suitable for the web.  
3. There are many opportunities to optimize portability, performance, and scalability.   
4. The net protocol can be designed to make the client JDBC driver very small and fast to load.   
5. The type 3 driver typically provides support for features such as caching (connections, query results, and so on), load balancing, and advanced   
system administration such as logging and auditing.  
6. This driver is very flexible allows access to multiple databases using one driver.  
7. They are the most efficient amongst all driver types.  
  
Disadvantage

It requires another server application to install and maintain. Traversing the recordset may take longer, since the data comes through the backend server.

**Type 4 JDBC Driver**

**Native-protocol/all-Java driver**

The Type 4 uses java networking libraries to communicate directly with the database server.  
  


Type 4: Native-protocol/all-Java driver

Advantage

1. The major benefit of using a type 4 jdbc drivers are that they are completely written in Java to achieve platform independence and eliminate deployment administration issues. It is most suitable for the web.   
2. Number of translation layers is very less i.e. type 4 JDBC drivers don't have to translate database requests to ODBC or a native connectivity interface or to pass the request on to another server, performance is typically quite good.   
3. You don’t need to install special software on the client or server. Further, these drivers can be downloaded dynamically.

Disadvantage  
  
With type 4 drivers, the user needs a different driver for each database.