**Java JDBC**

Database vendors typically provide the set of API for accessing the data and manage the database server. popular database vendor such as Oracle, sybase provide proprietary API for client access

client application written in the native language such as C/c++ can use this API to get Direct Access to the data.

the JDBC API provides alternate to using the vendor specific API

A JDBC driver is a middleware layer that translates the JDBC call to the vendor specific API’s and these are the classes which are implementing JDBC interface and generally are made by the database vendor

The Java JDBC API enables Java applications to connect to relational databases via a standard API, so your Java applications become independent (almost) of the database the application uses.

|  |
| --- |
| Java application using JDBC to connect to a database. |
| **Java application using JDBC to connect to a database.** |

JDBC standardizes how to connect to a database, how to execute queries against it, how to navigate the result of such a query, and how to exeucte updates in the database. JDBC does not standardize the SQL sent to the database. This may still vary from database to database.

JDBC drivers are classified into 4 categories

**Type 1 JDBC ODBC Bridge**

open database connection (ODBC) originally created to provide an API standard for SQL on Microsoft Windows platform and was later enhanced to provide SDK for the other platform

ODBC define a set of function for Direct Access to the data without the need of embedded SQL client application.

these functions are written in C language. only the prototype of the function is given by Microsoft.

the body of functions are given by the vendor of database like Oracle.

The first category of JDBC driver provides a bridge between JDBC and ODBC API. the bridge translates the standard JDBC call to the corresponding ODBC call and send them to the ODBC data call with ODBC libraries.

A type 1 JDBC driver consists of a Java part that translates the JDBC interface calls to ODBC calls. An ODBC bridge then calls the ODBC driver of the given database. Type 1 drivers are (were) mostly intended to be used in the beginning, when there were no type 4 drivers (all Java drivers). Here is an illustration of how a type 1 JDBC driver is organized:

|  |
| --- |
| **[Type 1 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-1.png)** |
| **Type 1 JDBC driver.** |

**Type 2 Native Driver**

JDBC database call at translated into vendor specific API calls. the database will process the request and send the result back through the API which will in turn forward them back to the JDBC driver.

the JDBC driver will translate to the result to JDBC standard and return them to the JAVA application.

A type 2 JDBC driver is like a type 1 driver, except the ODBC part is replaced with a native code part instead. The native code part is targeted at a specific database product. Here is an illustration of a type 2 JDBC driver:

|  |
| --- |
| **[Type 2 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-2.png)** |
| **Type 2 JDBC driver.** |

**Type 3 Intermediate Database Access Server**

type 3 driver use an intermediate database server that has the ability to connect multiple Java client to multiple database servers.

Java client application sends a call through JDBC driver to the intermediate data access server room which completes the request to the data source using another driver

the driver uses the database independent protocol to communicate database request to a server component which then translate the request into a database with waterfall iPod driver

A type 3 JDBC driver is an all Java driver that sends the JDBC interface calls to an intermediate server. The intermediate server then connects to the database on behalf of the JDBC driver. Here is an illustration of a type 3 JDBC driver:

|  |
| --- |
| **[Type 3 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-3.png)** |
| **Type 3 JDBC driver.** |

**Type 4 Thin Driver**

is a pure Java library that translates JDBC request directly to a database specific protocol.

A type 4 JDBC driver is an all Java driver which connects directly to the database. It is implemented for a specific database product. Today, most JDBC drivers are type 4 drivers. Here is an illustration of how a type 4 JDBC driver is organized:

|  |
| --- |
| **[Type 4 JDBC driver.](http://tutorials.jenkov.com/images/java-jdbc/driver-type-4.png)** |
| **Type 4 JDBC driver.** |