

Basic Information of JDBC Drivers

Ankush Parise^[1], Aniket Katakpure^[2], Rupali Rajurkar^[3], Harshal Kukade^[4]
 Department of Computer Science & Engineering
 Prof. Ram Meghe College of Engineering & Management, Badnera

Abstract:- A JDBC is a based-on Java technology. The stand for JDBC is a Java Database Connectivity. It is used to executing a SQL statements. In with include some classes and interface to help established connection between database and any Java application such as servlet, applet. JDBC gives a standard API to device/database developer and makes it possible to compose database applications utilizing a pure Java API. Using JDBC drivers it is possible to send virtually statement to database.in simple words we can't need to write one program to access system database, another program to access oracle database some others databases. One can write a single program is able to executed all SQL operations on databases. We know that Java is a platform independent language that's why we don't need to worry about different application run on different platforms. JDBC is a Java-based technology get to innovation (Java Standard Edition stage) from Oracle.in this technology describes how an end-user may access a database. It will support a facility such as select, delete, update, insert on a database. Java Database Connectivity API (Application Program Interface) is the business standard for database autonomous network between the Java programming dialect and a wide scope of databases – SQL databases and other forbidden information sources, for example, spreadsheets or level documents. The JDBC API gives a call-level API to SQL-based database get to.

Keywords:- JDBC, ODBC, JAVA, API, MYSQL.

I. INTRODUCTION

Java Database Connectivity is a mechanism to connect a database any java application. This is a technology provided from Oracle Corporation. A JDBC is a standard API for accessing a relational database. This interface provides an easy to access data from database. JDBC also provide an abstraction layer to hide background details for security purpose.it allow to access multiple relational database simultaneously. JDBC is utilized to cooperate with different kind of Database, for example, Oracle, MS Access, My SQL and SQL Server [1]. JDBC can likewise be characterized as the stage autonomous interface between a social database and Java programming.

For example, JDBC drivers allow to open database connection and it allow to interact with database or used SQL commands to receiving result a java application. A JDBC

mechanism allow perform various type of operations on a database such as inserting a data, delete a data, update a data as per requirement of a user's. It can also provide a facility to retrieve from a database for a data analytic.

The JDBC API provides the various interfaces and classes using this we can have established and manage connection between application and database. Are followed;

- DriverManager: This class manages to all databases drivers.
- Driver: This interface is managing to the communication between application and server.
- Connection: In this interface all method includes for a contracting a database.
- Statement: This interface is used to the send the SQL statements to the database.
- ResultSet: It is used to the retrieve the information from database.
- SQLException: This class is used to handles all error are occurred in database application.

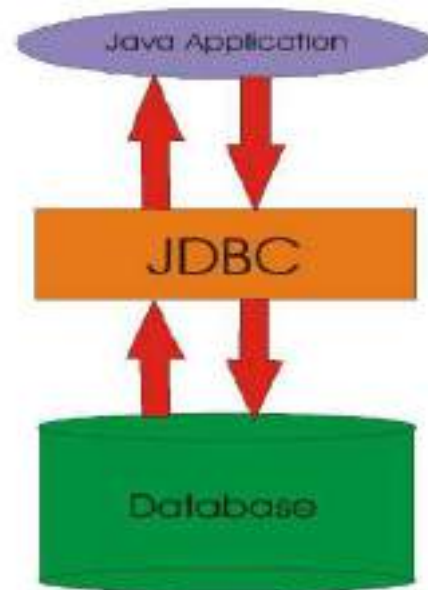


Fig 1:- Design of JDBC

In Fig.1 showing the architecture of a JDBC. JDBC is a mediator between database and Java application. We can have stored a data as well as retrieve the data from any database with the help of JDBC mechanism.

The JDBC API support both two-tier and three-tier architecture models for data accessing from a database. In consist JDBC two important layer are follows.

- JDBC API: To provide the JDBC connection between Application-to JDBC Manager.
- JDBC Driver API: It handle the connection between JDBC Manager and Driver.

The JDBC Driver Manager ensure that the correct Driver is used to access the data from database. The Driver Manager is capable of supporting a multiple concurrent driver to connected a database[2].

Facility Provided by JDBC: -

- Association Management.
- Automatic loading of Driver Interface.
- Exchanging among Global and Local Transactions.
- Save point in exchange the executives.
- Programmed Loading of Driver class.
- Annotation in SQL Query.
- Doesn't required special installation it will automatically download.

II. LITERATURE

Java is a high-level object-situated programming language created by the Sun Microsystems. It was created remembering the shopper gadgets and correspondence equipment's. It appeared as a piece of web application, web administrations and a stage independent programming dialect during the 1990s. A group of Sun Microsystems including Patrick Naught on, Mike Sheridan in the direction of James Goslings chose to build up a propelled programming dialect for the improvement of customer electronic gadgets. Java Language is initially called as "Oak" Then rename in "Java". They needed to make it new programming dependent on the intensity of systems that can keep running on various application zones, for example, PCs and electronic gadgets[3].

Sun Microsystems released JDBC as a component of JDK1.1 on February 19, 1997.It has since shaped piece of the Java Standard Edition. The JDBC classes are contained in the Java bundle or packages java.sql and javax.sql.

Beginning with variant 3.1, JDBC has been created under the Java Community Process. JSR 54 indicates JDBC 3.0 (incorporated into J2SE 1.4), JSR 114 determines the JDBC Row set increases, and JSR 221 is the detail of JDBC 4.0 (incorporated into Java SE 6). The most recent variant, JDBC 4.1, is determined by an upkeep arrival of JSR 221] and is incorporated into Java SE 7.

We can have stored the data using file structure but some disadvantage of file structure Are followed.

- First we know the IO packages of a file system.
- Can't support query language.
- Limited data stored.
- Data operation such as insert update delete select are very time consuming.
- can't support normalization
- Can't apply any constrained on data.
- Security are very low.

To remove all of those problem we can used a database software. This is a best way to stored data permanently[4]. It support query language and take very less time to performing insert update delete and select operations. List of database software are follows.

- Mysql
- Oracle
- Sqlserver
- Db2

Using this JDBC technology we can connect java code with an any above database. Java language support WORA ("Write Once, Run Anywhere") concept. The JDBC API also provide a call-level API for SQL database. Table 1 signifies the versions of JDBC.

Year	JDBC Version	JDK Implementation
2014	JDBC 4.2	JAVA SE 8
2011	JDBC 4.1	JAVA SE 7
2006	JDBC 4.0	JAVA SE 6
2001	JDBC 3.0	JDK 1.4
1999	JDBC 2.1	JDK 1.2
1997	JDBC 1.2	JDK 1.1

Table 1:- Versions of JDBC Drivers.

III. JDBC DRIVER

JDBC is an Application Programming Interface used in Java programming language. This API defines how a client access any tabular data. It is part of Java Standard Edition platform, provided from Oracle Corporation. It can work as a middle layer interface between database and Java application [5].

There are four types of JDBC Drivers:

- A. Type-1 Driver or JDBC-ODBC Bridge Driver
- B. Type-2 Driver or Native-API Driver
- C. Type-3 Driver or Network Protocol Driver
- D. Type-4 Driver or Thin Driver

A. Type-1 Driver

Fig 2 states, Type-1 driver or JDBC-ODBC bridge driver uses ODBC driver to connect to the database. The conversion of JDBC method into the ODBC function is done by the JDBC-ODBC bridge driver. Type-1 driver is also known as Universal driver.

This type of Driver goes about as an extension among JDBC and other database availability mechanism(ODBC). This driver changes over JDBC calls into ODBC calls and diverts the demand to the ODBC driver. The sort 1 drivers depend on ODBC which lessens their movability and allure. Additionally, questioning through the ODBC Bridge backs off exchanges.

- As a typical driver is utilized so as to cooperate with various databases, the information exchanged through this driver isn't so secured.
- The ODBC connect driver is should have been properly installed on individual client machines.
- Type-1 driver is developed in Java that's why this driver is not portable.

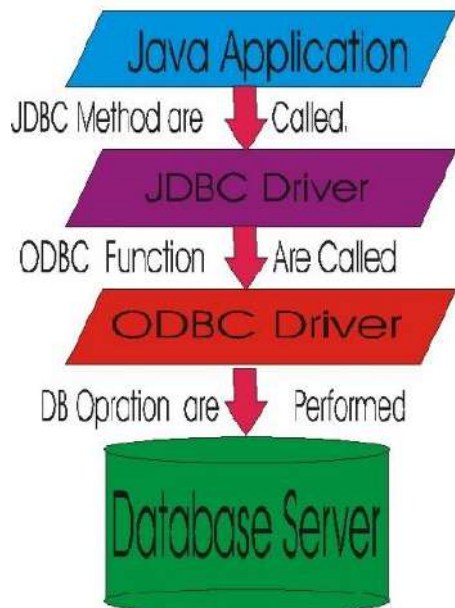


Fig 2:- JDBC-ODBC Design

B. Type-2 Driver

As shown in fig 3, the Native API driver utilizes the client-side libraries of the database. This driver changes over JDBC strategy calls into local calls of the database API [6]. So as to connect with various database, this driver needs their nearby API, that is the reason information exchange is significantly more secure when contrasted with sort 1 driver. This kind of driver make utilization of Java Native Interface(JNI) approach database explicit local customer API. These type of driver are developed in a C and C++.

- Driver should be installed independently in individual client machines.
- The Vendor client library should be in client on customer machine.
- Type-2 driver isn't written in Java, that is the reason is not a portable.

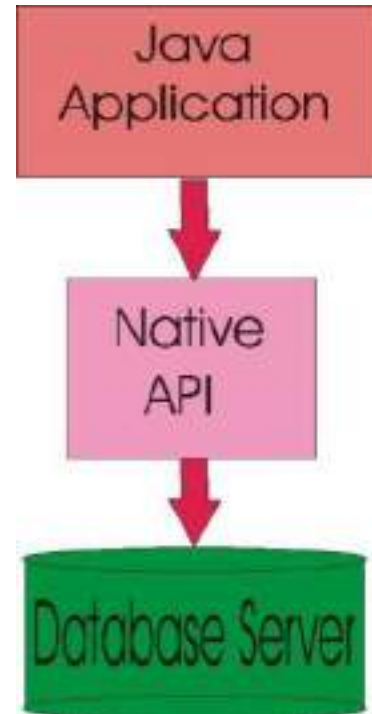


Fig 3:- Native-API driver Design

C. Type-3 Driver

Fig 4 gives the Network Protocol driver utilizes middleware (application server) that changes over JDBC calls directly or indirectly into a vender-specific database protocol. Here all the database network drivers are available in a single server, consequently no need of individual customer side establishment. Type-3 driver is used to translate JDBC calls into a JDBC independently. Further, they reduce the complexity as requests are passed through the network to the database server, which then translates the requests into the database specific native-connectivity interface for the database server. This type also provides the facility to switch over one database to another database[7].

- Type-3 drivers are completely developed in Java thus they are portable drivers.
- Doesn't required client side library because of application server. It can be performing many operations such as auditing, load balancing, logging etc.
- Network support on client machine
- Single driver provide access to multiple database.

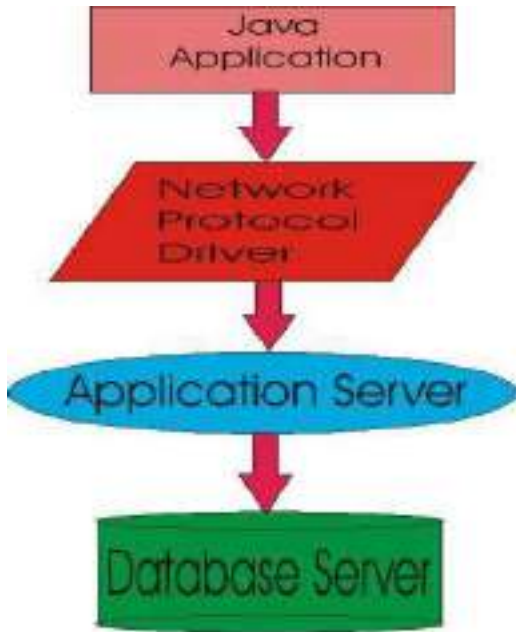


Fig 4:- Network and Protocol Driver Design

D. Type-4 Driver

Fig 5 shows the Type-4 driver is also called as native protocol driver. This driver interface directly with database. It doesn't require any native database library, that is the reason it is called Thin Driver. This driver associate specifically with database. They installed inside in a Java virtual machine of the client. This driver provide a better facilities and performance as compared to Type-1 and Type-2 drivers. All combination of results is performed on the customer machine.

MySQL's Connector/J driver is a Type 4 driver. In light of the exclusive idea of their system conventions, database sellers ordinarily supply type 4 drivers.

- These drivers can be downloaded dynamically.
- 100% pure java driver.
- It is completely written in Java, so it can be portable drivers.
- This type of driver is flexible. Doesn't required to any special software to installation.



Fig 5:- Thin driver Design

IV. WHICH SHOULD BE USED?

- When we want access only one type of database such as Oracle, Sybase then preferred Type-4 Driver.
- If we want multiple type of database access simultaneously in your java application, then preferred Type-3 Driver.
- If Type-3 and Type-4 are not available yet for your database, then preferred Type-2 Driver.
- The type 1 driver is used for development as well as testing purpose but it is not considered as a deployment-level driver.

V. CONCLUSION

Java is a platform independent, secure, robust language it also easy to understand. And it a best language to any database applications. Some java application is need to communicated a database for storing data and fetch or retrieve data from a database. JDBC is a mechanism to do this.

The JDBC driver manager is the backbone of the JDBC design. It really is very little and straightforward; its essential capacity is to associate Java applications to the right JDBC driver and afterward escape the way.

REFERENCES

- [1]. Sun Microsystems Inc., JDBC drivers, <http://www.oracle.com/technetwork/java/javase/>
- [2]. Sun Microsystems Inc., JDBC drivers, <http://www.oracle.com/technetwork/java/javase/jdbc/index.html>
- [3]. <https://www.javatpoint.com/jdbc-driver>
- [4]. <https://www.geeksforgeeks.org/jdbc-drivers>
- [5]. <http://www.tutorialspoints.com/jdbc/jdbc-introduction.html>
- [6]. http://en.wikipedia.org/wiki/jdbd_drivers
- [7]. <http://searchoracle.techtarget.com/definition/Java-Database-connectivity>