JDBC PART-1

JDBC:

JDBC stands for Java DataBase Connectivity.

JDBC is a specification for developing database applications with Java programming language.

Database Application:

An application that communicates with database is known as database application.

Application:

An application is a program in which we interact with on the desktop.

Database:

A database is a software and it is an organised collection of data.

Data organized in a database in the form of tables.

Each table contains fields and records.

Application is a front end & database is a back end.

Application uses SQL to communicate with database.

SQL:

SQL stands for Structured Query Language.

Database software contains two parts:

1) Database Application(Example: SQL*Plus) 2) Database(Example: Oracle)

List of databases:

1) Oracle

2) MySQL

3) MS-SQL Server

4) MS-Access

5) DB2

6) Derby

7) Sybase

8) DBase

9) FoxPro

10) Visual FoxPro .. etc.,

Application is called as client and database is called as server.

Client:

A client is a software that sends the request to server to get the response.

Server:

A server is a software that receives request from the client, process the request, constructs the response and sends the response back to a client.

Driver:

A driver is a software and it is used to connect application & database.

Specification:

A specification is a set of rules & guidelines that are used to develop environments & applications.

JDBC specification used by vendors to develop drivers.

JDBC specification also used by Java programmers to develop database applications.

There are four types of JDBC drivers:

1) Type-I Driver(JDBC ODBC Bridge Driver)

2) Type-II Driver(JDBC Native API Driver)

3) Type-III Driver(JDBC Network Protocol Driver)

4) Type-IV Driver(JDBC 100% Pure Java Driver)

1) Type - I Driver(JDBC ODBC Bridge Driver):

Driver Class Name:

sun.jdbc.odbc.JdbcOdbcDriver

Driver Location:

rt.jar file in JDK 1.7 & below versions.

Note: This driver removed from JDK1.8 & above versions.

rt=> Run Time jar=> Java ARchive

JDK=> Java Development Kit(It is a Java Software)

JAR file location:

C:\Program Files\Java\jdk1.7\jre\lib

Driver Vendor:

Sun Microsystems

Uniform Resource Locator(URL) to access the driver:

jdbc:odbc:dsn

jdbc=> protocol odbc=> sub protocol dsn=> data source name

Note: Type-I Driver developed in C language

Type - I Driver Functionality:

It converts Java instructions into odbc understandable format

Advantages of Type-I Driver:

1) It is very easy to connect

2) Only one driver that supports all ODBC enabled databases.

Disadvantages of Type-I Driver:

1) Performance overhead since Java calls should go through via jdbc & odbc drivers.

2) DSN creation required.

3) Database client software needs to be installed on local system

4) It is not suitable for applets because applet has the following security restrictions:

1) Applets cannot read data from local disk

2) Applets cannot write data to local disk

3) Applets cannot open new network connection otherthan the server from which it is loaded.

Type - II Driver(JDBC Native API Driver):

Type - II Driver is also called as Partial Java Driver (or) Partly Java-Partly Native Driver

Type - II Drivers are developed in Java language and native languages.

Type - II Driver Class Name for Oracle Database:

oracle.jdbc.driver.OracleDriver

Driver Vendor: Oracle Corporation

Driver Location:

ojdbc14.jar file in Oracle 10g Express Edition

ojdbc6_g.jar file in Oracle 11g Express Edition

URL to access Type-II Driver:

jdbc:oracle:oci8:@service-id

To get service-id, use the following SQL query:

SQL>select * from global_name;

Type - II Driver Functionality:

It converts Java calls into native calls.

Advantages:

1) It is little bit fast as compared to type - 1 driver

2) DSN is not required

Disadvantages:

1) Separate driver required for every database.

2) All databases are not having type-2 drivers.

3) Database software needs to be installed on system.

4) It is also not suitable for applets

Steps to develop database application:

1) Loading a specific JDBC driver.

2) Establishing a connection.

3) Performing the task.

4) Closing a connection.

JDBC API:

JDBC API is a Java API, that can access any kind of tabular data and data especially stored in RDBMS.

1) java.sql package 2) javax.sql package

java.sql package

Classes	Interfaces
1) DriverManager	1) Driver
2) SQLException	2) Connection
3) Types	3) Statement

4) Date

5) Time

4) PreparedStatement

5) CallableStatement

6) ResultSet

7) ResultSetMetaData

8) DatabaseMetaData

9) Blob

10) Clob

Program to establish the connection between Java application and Oracle Database by using Type-2 Driver:

```
import java.sql.*;
class ConnectionDemo
{
        public static void main(String args[])
        {
                try{
                Class c=Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con=DriverManager.getConnection(
"jdbc:oracle:oci8:@xe","system","manager");
                System.out.println("Connection Established Successfully");
                }catch(Exception e)
                {
                        System.err.println(e);
                }
        }
}
```