

Day 36

Java Database Connectivity with Oracle

To connect java application with the oracle database, we need to follow 5 following steps. In this example, we are using Oracle 10g as the database. So we need to know following information for the oracle database:

Driver class: The driver class for the oracle database is oracle.jdbc.driver.OracleDriver.

Connection URL: The connection URL for the oracle10G database is jdbc:oracle:thin:@localhost:1521:xe where jdbc is the API, oracle is the database, thin is the driver, localhost is the server name on which oracle is running, we may also use IP address, 1521 is the port number and XE is the Oracle service name. You may get all these information from the tnsnames.ora file.

Username: The default username for the oracle database is system.

Password: It is the password given by the user at the time of installing the oracle database.

Create a Table

Before establishing connection, let's first create a table in oracle database. Following is the SQL query to create a table.

```
create table emp(id number(10),name varchar2(40),age number(3));
```

Example to Connect Java Application with Oracle database

In this example, we are connecting to an Oracle database and getting data from emp table. Here, system and oracle are the username and password of the Oracle database.

```
import java.sql.*;
class OracleCon{
public static void main(String args[]){
try{
//step1 load the driver class
Class.forName("oracle.jdbc.driver.OracleDriver");

//step2 create the connection object
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");

//step3 create the statement object
Statement stmt=con.createStatement();

//step4 execute query
ResultSet rs=stmt.executeQuery("select * from emp");
while(rs.next())
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));

//step5 close the connection object
con.close();
```

```
}catch(Exception e){ System.out.println(e);}

}

}
```

The above example will fetch all the records of emp table.

To connect java application with the Oracle database ojdbc14.jar file is required to be loaded.

download the jar file ojdbc14.jar

Two ways to load the jar file:

paste the ojdbc14.jar file in jre/lib/ext folder
set classpath

1) paste the ojdbc14.jar file in JRE/lib/ext folder:

Firstly, search the ojdbc14.jar file then go to JRE/lib/ext folder and paste the jar file here.

2) set classpath:

There are two ways to set the classpath:

temporary
permanent

How to set the temporary classpath:

Firstly, search the ojdbc14.jar file then open command prompt and write:

```
C:>set classpath=c:\folder\ojdbc14.jar;.
```

Java Database Connectivity with MySQL

To connect Java application with the MySQL database, we need to follow 5 following steps.

In this example we are using MySql as the database. So we need to know following informations for the mysql database:

Driver class: The driver class for the mysql database is com.mysql.jdbc.Driver.

Connection URL: The connection URL for the mysql database is jdbc:mysql://localhost:3306/sonoo where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name. We may use any database, in such case, we need to replace the sonoo with our database name.

Username: The default username for the mysql database is root.

Password: It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

Let's first create a table in the mysql database, but before creating table, we need to create database first.

```
create database sonoo;
```

```
use sonoo;  
create table emp(id int(10),name varchar(40),age int(3));
```

Example to Connect Java Application with mysql database

In this example, sonoo is the database name, root is the username and password both.

```
import java.sql.*;  
class MysqlCon{  
    public static void main(String args[]){  
        try{  
            Class.forName("com.mysql.jdbc.Driver");  
            Connection con=DriverManager.getConnection(  
                "jdbc:mysql://localhost:3306/sonoo","root","root");  
            //here sonoo is database name, root is username and password  
            Statement stmt=con.createStatement();  
            ResultSet rs=stmt.executeQuery("select * from emp");  
            while(rs.next())  
                System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));  
            con.close();  
        }catch(Exception e){ System.out.println(e);}  
    }  
}
```

The above example will fetch all the records of emp table.

To connect java application with the mysql database, mysqlconnector.jar file is required to be loaded.

download the jar file mysql-connector.jar

Two ways to load the jar file:

Paste the mysqlconnector.jar file in jre/lib/ext folder

Set classpath

1) Paste the mysqlconnector.jar file in JRE/lib/ext folder:

Download the mysqlconnector.jar file. Go to jre/lib/ext folder and paste the jar file here.

2) Set classpath:

There are two ways to set the classpath:

temporary

permanent

How to set the temporary classpath

open command prompt and write:

```
C:>set classpath=c:\folder\mysql-connector-java-5.0.8-bin.jar;.
```

How to set the permanent classpath

Go to environment variable then click on new tab. In variable name write classpath and in variable value paste the path to the mysqlconnector.jar file by appending mysqlconnector.jar;.; as C:\folder\mysql-connector-java-5.0.8-bin.jar;.

Connectivity with Access without DSN

There are two ways to connect java application with the access database.

Without DSN (Data Source Name)

With DSN

Java is mostly used with Oracle, mysql, or DB2 database. So you can learn this topic only for knowledge.

Example to Connect Java Application with access without DSN

In this example, we are going to connect the java program with the access database. In such case, we have created the login table in the access database. There is only one column in the table named name. Let's get all the name of the login table.

```
import java.sql.*;
class Test{
public static void main(String ar[]){
try{
String database="student.mdb";//Here database exists in the current directory

String url="jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};
          DBQ="+ database + ";DriverID=22;READONLY=true";

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection c=DriverManager.getConnection(url);
Statement st=c.createStatement();
ResultSet rs=st.executeQuery("select * from login");

while(rs.next()){
System.out.println(rs.getString(1));
}

}catch(Exception ee){System.out.println(ee);}

}}
```

Example to Connect Java Application with access with DSN

Connectivity with type1 driver is not considered good. To connect java application with type1 driver, create DSN first, here we are assuming your dsn name is mydsn.

```
import java.sql.*;
class Test{
public static void main(String ar[]){
```

```
try{
    String url="jdbc:odbc:mydsn";
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    Connection c=DriverManager.getConnection(url);
    Statement st=c.createStatement();
    ResultSet rs=st.executeQuery("select * from login");

    while(rs.next()){
        System.out.println(rs.getString(1));
    }

} catch(Exception ee){System.out.println(ee);}

}}
```