

15-06-2020

It is free edition of database —

Oracle Database Express Edition (XE) 18.4.0.0.0 (18c) : — for Windows x64. — Please download and install — (free edition)

Its size — (1.91GB) The necessary database driver classes for Oracle 18c XE are available in `ojdbc8.jar` file. So, using the jar file's location in classpath variable can connect the database through JDBC code.

Driver class Name for Oracle :

```
oracle.jdbc.driver.OracleDriver
```

Oracle 11g is also available for the same work to save memory. It will consume not more than 375 MB space — If you have less memory then you should

Database Connection URL for Oracle : — go for Oracle 11g XE Version

```
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 specify IP Address
`1521` is the port number.
`xe` is the Oracle Service Name.

MySQL Driver class name for JDBC —

```
com.mysql.jdbc.Driver
```

We will try to connect with Oracle 18c XE database through JDBC. Before doing this you should be aware about basic database commands which is provided by me now : — Please Remember!

SQL Commands (Used in JDBC)

P-2

Select Commands

(Queries)

(DQL) Data Query Language
Select * from employees;

Non Select Commands

(Queries)

(DML) Data Manipulation Language

insert
delete
update

Basic SQL Commands:-

① Creating a table:-

```
create table emp1 (empno. number, ename varchar2(15),  
post varchar2(10), address varchar2(30), basic number);
```

② To Delete/Drop table:-

```
Drop table emp1;
```

③ To insert rows into table:-

```
insert into emp1 values(100, 'Chandru', 'manager', 'Sasaram',
```

```
insert into emp1 values(101, 'Manish', 'Assistant', 'Dehri', 16000);
```

④ To update data:-

I want to replace post of empno=101 to

Associate. ~~manager~~

```
update emp1 set post='Associate' where empno=101;
```

⑤ To delete a row:-

```
delete from emp1 where empno=101;
```

⑥ To select data:-

```
Select * from emp1;
```

The above basic commands, you all need to know before doing JDBC programming.

Let us do the practical of making movies table P-3

as follows:-

Open Oracle database SQL Command prompt:-

Connect system/rranjan ←

Connected default username password specified while installing

SQL> create table movies (mno number, mname varchar2(10),
hero varchar2(10), heroine varchar2(10)); ←

After executing this command you will get message as -
Table created.

SQL> insert into movies values(1, 'Sultan', 'Salman', 'Anushka'); ←

(1) One row created

SQL> insert into movies values(2, 'Lagan', 'Amir', 'Padmini'); ←

1 row created

SQL> insert into movies values(3, 'Rees', 'Sharukh', 'Sunny'); ←

1 row created

SQL> Select * from movies; ←

MNO	MNAME	HERO	HEROINE
1	Sultan	Salman	Anushka
2	Lagan	Amir	Padmini
3	Rees	Sharukh	Sunny

SQL> Update movies set heroine = 'Gracy Singh' where
MNO=2; ←

SQL> Select * from movies; ←

You will see the current available data from table.

SQL> Delete from movies where mno=1; ←

1 row deleted. SQL> Select * from movies; ← you can see the result.

Now, let us take an example which creates a table named- emp1, with fields(columns)- empno, ename, post, address, basic using JDBC program.

```

import java.sql.*;
class Createtable {
public static void main(String args[]) throws Exception
{
String driver = "oracle.jdbc.driver.OracleDriver";
String url url = "jdbc:oracle:thin:@localhost:1521:xe";
String user = "system";
String pass = "rranjan";
Class.forName(oracle driver);
Connection con = DriverManager.getConnection(url, user, pass);
Statement st = con.createStatement();
st.executeUpdate("Create table emp1 (empno number, varchar(25), ename varchar2(25), post varchar2(15), address varchar2(35), basic number)");
st.executeUpdate("insert into emp1 values(100, 'Mohan', 'clerk', 'Gaya', 8500)");
ResultSet rs = st.executeQuery("commit");
rs = st.executeQuery("select * from emp1");
while(rs.next()) {
System.out.println(rs.getInt(1) + "\t" + rs.getString(2) + "\t" + rs.getString(3) + "\t" + rs.getString(4) + "\t" + rs.getInt(5));
}
con.close();
}
}

```

} // close of main function

} // close of class Creatable

P-5

Please remember — If we execute an SQL select command, a group of records we will get.

These ^(group) results of records as output can be handled by JDBC through ResultSet class object.

In case of Non-select Query results — (insert, delete, update)
output of insert — 1 row created

SQL > insert into movies values(4, 'Spider', 'Mahesh', 'Rani');
① row created.
Output of non-select query can be handled in JDBC as:—

These no. of row affected are handled by

SQL > delete from movies where mno=4;
① row deleted.

Three different methods to execute queries — in JDBC —

- ① executeQuery() — to execute select queries
returns → group of records (ResultSet)
- ② executeUpdate() — to execute insert, delete, update queries.
(Non-Select Queries)
- ③ execute() —

① public ResultSet executeQuery(String sql) throws SQLException;
eg:- ResultSet rs = st.executeQuery("Select * from movies");

② public int executeUpdate(String sql) throws SQLException;

eg → ~~int~~ int x = st.executeUpdate("delete from employees where esal > 50000);

System.out.println("The no. of employees deleted: " + x);

if we want to delete all records of employees table whose esal is greater than 50000. The no. of records deleted are stored into variable x.

Summary of all - JDBC Drivers (4.3)

The current version of JDBC is 4.3 - released since 21st September 2017.

<u>Property</u>	Type - I	Type - II	Type - III	Type - IV
1. Conversion	From JDBC call to ODBC call	From JDBC calls to Native library calls	From JDBC calls to middleware Server specific calls	From JDBC calls to database specific calls
2. Implemented in language	Only Java	Java & Native languages	Only Java	Only Java
3. Architecture follows	2-Tier	2-Tier	3-Tier	2-Tier
4. Is it platform independent driver?	NO	NO	Yes	yes
5. Is it database independent driver?	Yes	NO	Yes	NO
6. Is it thin or thick?	<u>Thick</u>	<u>Thick</u>	<u>thick</u>	Thin,
7. Also known as -	JDBC-ODBC Bridge Driver or Bridge Driver	Native API partly Java Driver or Native Driver	All Java Net Protocol Driver	All Java native Protocol Driver or Thin Driver

- END -