Note-1: If select SQL query execution gives bunch of records (0 or more records) to resultset object then we should use while(rs.next()) to process the result set.

Note-2: If select SQL query execution gives 0 or single record to result set object then we should use If(rs.next()) to process the result set.

Ex:- select \* from emp where eno=10;

Note-3: if select SQL query execution gives single record to result set object then we don’t need to use while(rs.next()) , if(rs.next()) . Just use rs.next() method.

Ex:- select count(\*) from emp;

1.Write JDBC application to Retrieve Records from Emp table.

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** Demo {

**public** **static** **void** main(String[] args) {

Connection con=**null**;

Statement s1=**null**;

ResultSet r1=**null**;

**try** {

Class.*forName*("oracle.jdbc.driver.OracleDriver");

con=DriverManager.*getConnection*("jdbc:oracle:thin:@localhost:1521:XE","sukumar","sukumar");

**if**(con!=**null**)

s1=con.createStatement();

**if**(s1!=**null**)

r1=s1.executeQuery("Select \* from emp");

**if**(r1!=**null**)

{

**while**(r1.next()) {

System.***out***.println(r1.getInt(1)+" "+r1.getString(2));

}

}

}

**catch**(SQLException e)

{

System.***out***.println(e.getMessage());

}

**catch**(ClassNotFoundException e)

{

System.***out***.println(e.getMessage());

}

**finally** {

**try** {

r1.close();

s1.close();

con.close();

} **catch** (SQLException e) {

e.getMessage();

}

}

}

}

Output:

---------

Eno Emp-Name

---------------------

1 suku

2 sv

Note:- The JDBC application has to close objects in the following specified order.

1. Close the resultSet

The JDBC application can’t process result set further.

1. Close the Statement

The JDBC application can’t send SQL Query further.

1. Close the connection

The JDBC application can’t interact with db further.