JAVA EXAMPLES - CONNECT TO A DATABASE

 Copyright © tutorialspoint.com

Problem Description:

How to connect to a database using JDBC? Assume that database name is testDb and it has table named employee which has 2 records.

Solution:

Following example uses getConnection, createStatement & executeQuery methods to connect to a database & execute queries.

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) {
      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
      }
      catch(ClassNotFoundException e) {
         System.out.println("Class not found "+ e);
      }
      System.out.println("JDBC Class found");
      int no_of_rows = 0;
      try {
         Connection con = DriverManager.getConnection
         ("jdbc:derby://localhost:1527/testDb","username",
         "password");
         Statement stmt = con.createStatement();
         ResultSet rs = stmt.executeQuery
         ("SELECT * FROM employee");
         while (rs.next()) {
            no_of_rows++;
         }
         System.out.println("There are "+ no_of_rows
         + " record in the table");
      }
      catch(SQLException e){
         System.out.println("SQL exception occured" + e);
      }
   }
}
```

Result:

The above code sample will produce the following result.The result may vary. You will get ClassNotfound exception if your JDBC driver is not installed properly.

```
JDBC Class found
There are 2 record in the table
```

JAVA EXAMPLES - EDIT TABLE

 Copyright © tutorialspoint.com

Problem Description:

How to edit*Addorupdate* columns of a Table and how to delete a table?

Solution:

Following example uses create, alter & drop SQL commands to create, edit or delete table

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) throws Exception{
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection con = DriverManager.getConnection
      ("jdbc:derby://localhost:1527/testDb","username",
      "password");
      Statement stmt = con.createStatement();
      String query ="CREATE TABLE employees
      (id INTEGER PRIMARY KEY,
      first_name CHAR(50),last_name CHAR(75))";
      stmt.execute(query);
      System.out.println("Employee table created");
      String query1 = "aLTER TABLE employees ADD
      address CHAR(100) ";
      String query2 = "ALTER TABLE employees DROP
      COLUMN last_name";
      stmt.execute(query1);
      stmt.execute(query2);
      System.out.println("Address column added to the table
      & last_name column removed from the table");
      String query3 = "drop table employees";
      stmt.execute(query3);
      System.out.println("Employees table removed");
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
Employee table created
Address column added to the table & last_name
column removed from the table
Employees table removed from the database
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js

JAVA EXAMPLES - RETRIEVE CONTENTS FROM MANY TABLES

 Copyright © tutorialspoint.com

Problem Description:

How to join contents of more than one table & display?

Solution:

Following example uses inner join sql command to combine data from two tables. To display the contents of the table getString method of resultset is used.

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) throws Exception{
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection con = DriverManager.getConnection
      ("jdbc:derby://localhost:1527/testDb","username",
      "password");
      Statement stmt = con.createStatement();
      String query ="SELECT fname,lname,isbn from author
      inner join books on author.AUTHORID = books.AUTHORID";
      ResultSet rs = stmt.executeQuery(query);
      System.out.println("Fname  Lname   ISBN");
      while (rs.next()) {
         String fname = rs.getString("fname");
         String lname = rs.getString("lname");
         int isbn = rs.getInt("isbn");
         System.out.println(fname + "  " + lname+"   "+isbn);
      }
      System.out.println();
      System.out.println();
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
Fname  Lname   ISBN
john  grisham   123
jeffry  archer   113
jeffry  archer   112
jeffry  archer   122
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js

JAVA EXAMPLES - USE OF PREPARED STATEMENT

 Copyright © tutorialspoint.com

Problem Description:

How to use Prepared Statement in java?

Solution:

Following example uses PrepareStatement method to create PreparedStatement. It also uses setInt & setString methods of PreparedStatement to set parameters of PreparedStatement.

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) throws Exception{
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection con = DriverManager.getConnection
      ("jdbc:derby://localhost:1527/testDb","name","pass");
      PreparedStatement updateemp = con.prepareStatement
      ("insert into emp values(?,?,?)");
      updateemp.setInt(1,23);
      updateemp.setString(2,"Roshan");
      updateemp.setString(3, "CEO");
      updateemp.executeUpdate();
      Statement stmt = con.createStatement();
      String query = "select * from emp";
      ResultSet rs =  stmt.executeQuery(query);
      System.out.println("Id Name    Job");
      while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         String job = rs.getString("job");
         System.out.println(id + "  " + name+"   "+job);
      }
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
Id Name    Job
23  Roshan   CEO
```

JAVA EXAMPLES - RETRIEVE TABLE CONTENTS

 Copyright © tutorialspoint.com

Problem Description:

How to retrieve contents of a table using JDBC connection?

Solution:

Following example uses getString,getInt & executeQuery methods to fetch & display the contents of the table.

```
import java.sql.*;

public class jdbcResultSet {
   public static void main(String[] args) {
      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
      }
      catch(ClassNotFoundException e) {
         System.out.println("Class not found "+ e);
      }
      try {
         Connection con = DriverManager.getConnection
         ("jdbc:derby://localhost:1527/testDb","username",
         "password");
         Statement stmt = con.createStatement();
         ResultSet rs = stmt.executeQuery
         ("SELECT * FROM employee");
         System.out.println("id  name    job");
         while (rs.next()) {
            int id = rs.getInt("id");
            String name = rs.getString("name");
            String job = rs.getString("job");
            System.out.println(id+"   "+name+"    "+job);
         }
      }
      catch(SQLException e){
         System.out.println("SQL exception occured" + e);
      }
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
id  name   job
1   alok   trainee
2   ravi   trainee
```

JAVA EXAMPLES - USE OF SAVEPOINT & ROLLBACK

 Copyright © tutorialspoint.com

Problem Description:

How to make a Savepoint & Rollback in java?

Solution:

Following example uses Rollback method of connection to Rollback to a previously saved SavePoint

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) throws Exception{
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection con = DriverManager.getConnection
      ("jdbc:derby://localhost:1527/testDb","name","pass");
      Statement stmt = con.createStatement();
      String query1 = "insert into emp values(5,'name','job')";
      String query2 = "select * from emp";
      con.setAutoCommit(false);
      Savepoint spt1 = con.setSavepoint("svpt1");
      stmt.execute(query1);
      ResultSet rs = stmt.executeQuery(query2);
      int no_of_rows = 0;
      while (rs.next()) {
         no_of_rows++;
      }
      System.out.println("rows before rollback statement = "
      + no_of_rows);
      con.rollback(spt1);
      con.commit();
      no_of_rows = 0;
      rs = stmt.executeQuery(query2);
      while (rs.next()) {
         no_of_rows++;
      }
      System.out.println("rows after rollback statement = "
      + no_of_rows);
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
rows before rollback statement = 4
rows after rollback statement = 3
```

JAVA EXAMPLES - SEARCH TABLE CONTENTS

 Copyright © tutorialspoint.com

Problem Description:

How to Search contents of a table?

Solution:

Following method uses where & like sql Commands to search through the database.

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) throws Exception{
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection con = DriverManager.getConnection
      ("jdbc:derby://localhost:1527/testDb","username",
      "password");
      Statement stmt = con.createStatement();
      String query[] ={"SELECT * FROM emp where id=1",
      "select name from emp where name like 'ravi_'",
      "select name from emp where name like 'ravi%'"};
      for(String q : query){
         ResultSet rs = stmt.executeQuery(q);
         System.out.println("Names for query "+q+" are");
         while (rs.next()) {
            String name = rs.getString("name");
            System.out.print(name+"  ");
         }
         System.out.println();
      }
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
Names for query SELECT * FROM emp where id=1 are
ravi
Names for query select name from emp where name like 'ravi_' are
ravi2  ravi3
Names for query select name from emp where name like 'ravi%' are
ravi  ravi2  ravi3  ravi123  ravi222
```

JAVA EXAMPLES - SORT CONTENTS OF A TABLE

 Copyright © tutorialspoint.com

Problem Description:

How to sort contents of a table?

Solution:

Following example uses Order by SQL command to sort the table.

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) throws Exception{
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection con = DriverManager.getConnection
      ("jdbc:derby://localhost:1527/testDb","name","pass");
      Statement stmt = con.createStatement();
      String query = "select * from emp order by name";
      String query1="select * from emp order by name, job";
      ResultSet rs = stmt.executeQuery(query);
      System.out.println("Table contents sorted by Name");
      System.out.println("Id Name    Job");
      while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         String job = rs.getString("job");
         System.out.println(id + "  " + name+"   "+job);
      }
      rs = stmt.executeQuery(query1);
      System.out.println("Table contents after sorted
      by Name & job");
      System.out.println("Id Name    Job");
      while (rs.next()) {
         int id = rs.getInt("id");
         String name = rs.getString("name");
         String job = rs.getString("job");
         System.out.println(id + "  " + name+"   "+job);
      }
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
Table contents after sorting by Name
Id Name    Job
1  ravi   trainee
5  ravi   MD
4  ravi   CEO
2  ravindra   CEO
2  ravish   trainee
Table contents after sorting by Name & job
Id Name    Job
4  ravi   CEO
5  ravi   MD
1  ravi   trainee
2  ravindra   CEO
2  ravish   trainee
```

JAVA EXAMPLES - UPDATE TABLE CONTENTS

 Copyright © tutorialspoint.com

Problem Description:

How to update*delete*, *insertorupdate* contents of a table using JDBC connection?

Solution:

Following method uses update, delete & insert SQL commands to edit or delete row contents.

```
import java.sql.*;

public class updateTable {
   public static void main(String[] args) {
      try {
         Class.forName("org.apache.derby.jdbc.ClientDriver");
      }
      catch(ClassNotFoundException e) {
         System.out.println("Class not found "+ e);
      }
      try {
         Connection con = DriverManager.getConnection
         ("jdbc:derby://localhost:1527/testDb","username",
         "password");
         Statement stmt = con.createStatement();
         String query1="update emp set name='ravi' where id=2";
         String query2 = "delete  from emp where id=1";
         String query3 = "insert into emp values
         (1,'ronak','manager')";
         stmt.execute(query1);
         stmt.execute(query2);
         stmt.execute(query3);
         ResultSet rs = stmt.executeQuery("SELECT * FROM emp");
         System.out.println("id  name    job");
         while (rs.next()) {
            int id = rs.getInt("id");
            String name = rs.getString("name");
            String job = rs.getString("job");
            System.out.println(id+"   "+name+"    "+job);
         }
      }
      catch(SQLException e){
         System.out.println("SQL exception occured" + e);
      }
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
id  name    job
2   ravi    trainee
1   ronak   manager
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js

JAVA EXAMPLES - COMMIT A STATEMENT

 Copyright © tutorialspoint.com

Problem Description:

How to commit a query?

Solution:

Following example uses connection.commit method to execute a query.

```
import java.sql.*;

public class jdbcConn {
   public static void main(String[] args) throws Exception{
      Class.forName("org.apache.derby.jdbc.ClientDriver");
      Connection con = DriverManager.getConnection
      ("jdbc:derby://localhost:1527/testDb","name","pass");
      Statement stmt = con.createStatement();
      String query = "insert into emp values(2,'name1','job')";
      String query1 ="insert into emp values(5,'name2','job')";
      String query2 = "select * from emp";
      ResultSet rs = stmt.executeQuery(query2);
      int no_of_rows = 0;
      while (rs.next()) {
         no_of_rows++;
      }
      System.out.println("No. of rows before commit
      statement = "+ no_of_rows);
      con.setAutoCommit(false);
      stmt.execute(query1);
      stmt.execute(query);
      con.commit();
      rs = stmt.executeQuery(query2);
      no_of_rows = 0;
      while (rs.next()) {
         no_of_rows++;
      }
      System.out.println("No. of rows after commit
      statement = "+ no_of_rows);
   }
}
```

Result:

The above code sample will produce the following result.The result may vary.

```
No. of rows before commit statement = 1
No. of rows after commit statement = 3
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js