Step 3:

Create a statement

We can create a statement in Three ways:

1. Using Statement Interface
2. Using PreparedStatement Interface
3. Using CallableStatement Interface

1 and 2 are used to execute SQL Queries(Insert, Update, Delete, Select)

3 is used to execute either Procedure or function

Using Statement interface:

How to create a statement interface?

We have a predefined method called createStatement in Connection interface that returns Statement object

Statement stmt = con.createStatement()

Note:

1. createStatement method throws SQL Exception which is a checked exception
2. at the end of the program, we must close the object of statement using stmt.close()

Step 4: executing the query using Statement interface

**Step 4 🡪 Execute Sql queries :**

In order to execute the SQL commands on database, Statement or PreparedStatement interface provides three different methods:

|  |  |  |  |
| --- | --- | --- | --- |
| Object Reference | Method | Return type | Purpose |
| Statement | execute(String) | Boolean | used to write any SQL Query (select or non – select) |
| executeUpdate(String) | int | used to write execute DML queries(insert or delete or update) |
| executeQuery(String) | ResultSet | used to execute select statement |

How to get the data from SQL?

ResultSet rs = stmt.executeQuery(select-query);

executeQuery method returns ResultSet Obect

What is ResultSet?

* Is an Java Object
* Contains the output from SQL Query

Note:

1. executeQuery throws a checked exception called SQL Exception
2. at the end of the program, we must close the resultset object using rs.close() before closing the statement object

Step 5: Processing the Result Set

In this step, we are retrieving the values from each and every column using getX methods

Where X is the Java Data type of the respective column

**Step 5 🡪 Process the ResultSet :**

For the select operations, we use executeQuery() method. The executed query returns the data in the form of ResultSet object. To process the data we need to go through the ResultSet.

|  |  |
| --- | --- |
| Some of the Methods available in ResultSet Interface | |
| **int getInt(int ColumnIndex)**  **int getInt(String ColumnName)** | It is used to get the value of the specified column Index as an int data type. |
| **float getFloat(int ColumnIndex)**  **float getFloat(String ColumnName):** | It is used to get the value of the specified column Index as a float data type. |
| **java.sql.date getDate(int ColumnIndex)**  **java.sql.date getDate(String ColumnName)** | It is used to get the value of the specified column Index as a date value. |

How to navigate the cursor from one row to another row?

|  |  |
| --- | --- |
| **boolean next()** | Returns false, if the pointer reached to end of the list. Otherwise true. |

Example to process the result set

For Example:

|  |
| --- |
| ResultSet rs = stmt.executeQuery("select empno from emp");  while(rs.next()){  System.out.println(rs.getInt(1));  } |

JDBC program to retrieve the Employee information from Employee Table

**package** com.sssit.jdbcapp2;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** RetrieveAllEmployees {

**public** **static** **final** String ***URL*** = "jdbc:oracle:thin:@localhost:1521:orcl";

**public** **static** **final** String ***UNAME*** = "advdec23";

**public** **static** **final** String ***PASS*** = "advdec23";

**public** **static** **final** String ***SQLQUERY***="select empno,ename,job,sal from employee";

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

**try** {

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

System.***out***.println("Class Loaded successfully....");

Connection con = DriverManager.*getConnection*(***URL***, ***UNAME***, ***PASS***);

System.***out***.println("Connection Estd....");

Statement stmt = con.createStatement();

ResultSet rs = stmt.executeQuery(***SQLQUERY***);

**while**(rs.next()) {

System.***out***.printf("%10d%20s%10s%10.2f\n",

rs.getInt(1),

rs.getString("ename"),

rs.getString("job"),

rs.getFloat("sal")

);

}

rs.close();

stmt.close();

con.close();

} **catch** (ClassNotFoundException e) {

System.***out***.println("Failed to Load Driver class...");

} **catch** (SQLException e) {

System.***out***.println("Failed in executing SQL operation");

}

}

}

JDBC program to Insert a row to Employee table

**package** com.sssit.jdbcapp2;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** InsertEmployee {

**public** **static** **final** String ***URL*** = "jdbc:oracle:thin:@localhost:1521:orcl";

**public** **static** **final** String ***UNAME*** = "advdec23";

**public** **static** **final** String ***PASS*** = "advdec23";

**public** **static** **final** String ***SQLQUERY***=

"insert into employee values(8005,'Emp5','Clerk',8001,'28-Dec-23',5000,null,null)";

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

**try** {

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

System.***out***.println("Class Loaded successfully....");

Connection con = DriverManager.*getConnection*(***URL***, ***UNAME***, ***PASS***);

System.***out***.println("Connection Estd....");

Statement stmt = con.createStatement();

**int** result = stmt.executeUpdate(***SQLQUERY***);

**if**(result>0)

System.***out***.println("Inserted successfully.....");

**else**

System.***out***.println("Insertion Failed.....");

stmt.close();

con.close();

} **catch** (ClassNotFoundException e) {

System.***out***.println("Failed to Load Driver class...");

} **catch** (SQLException e) {

System.***out***.println("Failed in executing SQL operation");

}

}

}