Dt : 4/3/2025

Note:

=>we use getConnection()-method is from 'DriverManager' to create implementation Object for 'Connection' interface, because getConnection()-method internally holding 'Anonymous Local InnerClass as implementation class of Connection interface' and which generate Connection-Implementation Object

Method Signature of getConnection():

```
public static java.sql.Connection getConnection
          (java.lang.String, java.lang.String, java.lang.String)
                              throws java.sql.SQLException;
```

syntax:

```
Connection con = DriverManager.getConnection("DB-URL","DB-UName","DB-PWord");
```

DB-URL => jdbc:oracle:thin:@localhost:1521:XE

DB-UName=> system

DB-PWord=> tiger

```
Connection con = DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:xe","system","tiger");
```

****imp***

JDBC statements:

Venkatesh Maipathii

=>JDBC statements will specify the type of operation to be performed on DB Product.

=>These JDBC statements are categorized into three types:

1.Statement

2.PreparedStatement

3.CallableStatement

1.Statement:

=>'Statement' is an interface from java.sql package and which is used to execute normal queries without IN-Parameters.

(Normal queries means Create,Insert,Select,Update and delete)

=>we use createStatement()-method from 'Connection' interface to create implementation object for 'Statement' interface,because this createStatement()-method internally holding 'Anonymous Local InnerClass as implementation class of Statement-Interface' and which generate Statement-Object.

Method Signature of createStatement();

```
public abstract java.sql.Statement createStatement() throws java.sql.SQLException;
```

syntax:

```
Statement stm = con.createStatement();
```

=>The following are two important methods of 'Statement' interface:

(a)executeQuery()

(b)executeUpdate()

Venkatesh Maipathii

(a)executeQuery():

=>executeQuery()-method is used to execute select-queries

Method Signature of executeQuery():

```
public abstract java.sql.ResultSet executeQuery(java.lang.String)
                                        throws java.sql.SQLException;
```

syntax:

```
ResultSet rs = stm.executeQuery("select-query");
```

(b)executeUpdate():

=>executeUpdate()-method is used to execute NonSelect-Queries

Method Signature of executeUpdate:

```
public abstract int executeUpdate(java.lang.String) throws java.sql.SQLException;
```

syntax:

```
int k = stm.executeUpdate("NonSelect-Query");
```

----

*imp

Creating JDBC Application Using IDE Eclipse:

step-1 : Open IDE Eclipse,while opening name the WorkSpace and click 'Launch'

step-2 : Create Java Project

step-3 : Add DB-Jar file to Java-Project through 'Build path'

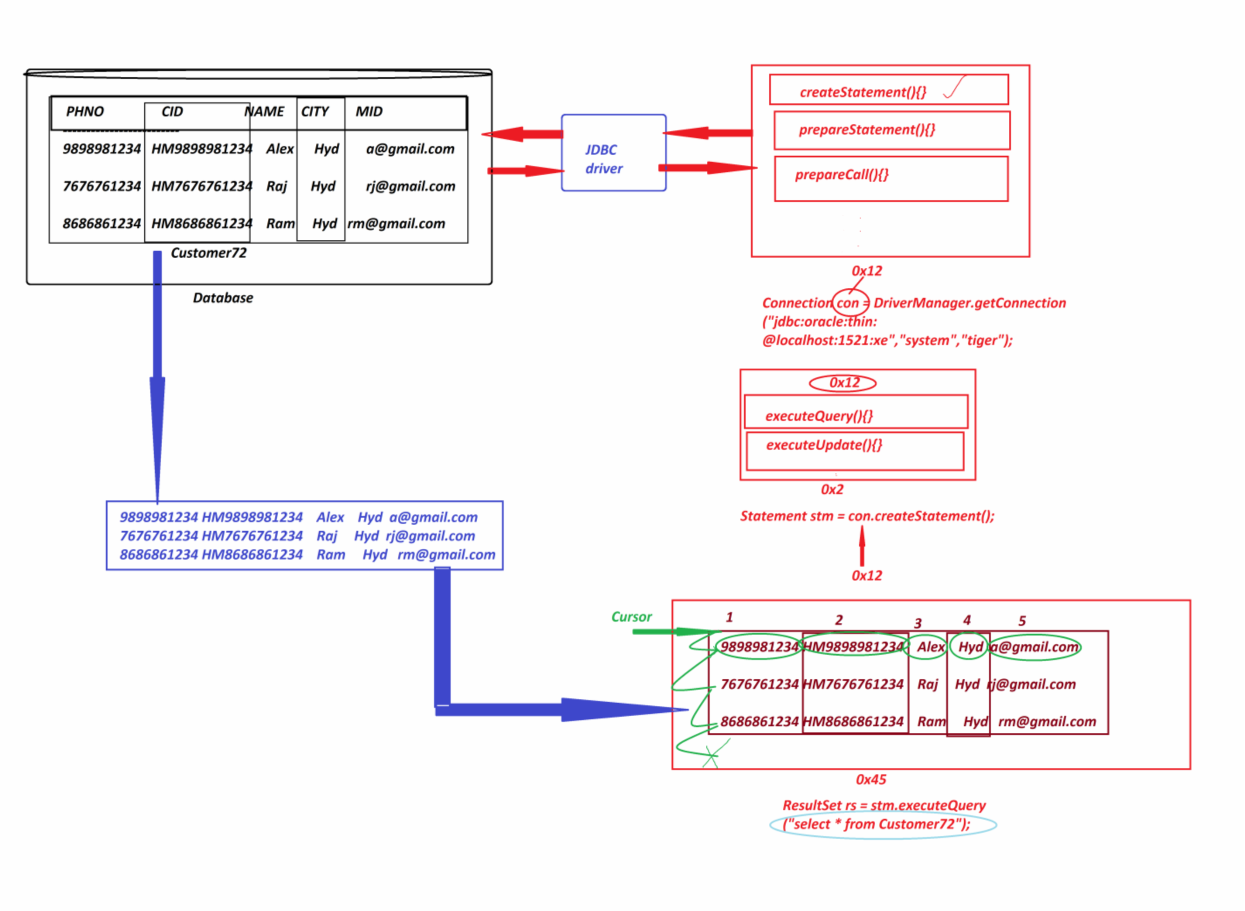
RightClick on Project->Build Path->Configure Build Path->Libraries->select 'Classpath' and click 'Add External JARs'->Browse and select DB-Jar file from user defined folder->Open->

Venkatesh Maipathii

Apply->Apply and Close.

step-4 : Create package in 'src'

step-5 : Create class(JDBC Program) in package and write JDBC-code to display all Customer details



PHNO	CID	NAME	CITY	MID
9898981234	HM9898981234	Alex	Hyd	a@gmail.com
7676761234	HM7676761234	Raj	Hyd	rj@gmail.com
8686861234	HM8686861234	Ram	Hyd	rm@gmail.com

```
Connection con = DriverManager.getConnection
("jdbc:oracle:thin:
@localhost:1521:xe","system","tiger");
```

```
Statement stm = con.createStatement();
```

```
ResultSet rs = stm.executeQuery
("select * from Customer72");
```

Customer72
Database
9898981234 HM9898981234 Alex Hyd a@gmail.com
7676761234 HM7676761234 Raj Hyd rj@gmail.com
8686861234 HM8686861234 Ram Hyd rm@gmail.com
JDBC driver
createStatement(){}
prepareStatement(){}
prepareCall(){}
0x12
0x12
executeQuery(){}
executeUpdate(){}
0x2
0x12
Cursor
1
2
3
4
5
9898981234 HM9898981234 Alex Hyd a@gmail.com
7676761234 HM7676761234 Raj Hyd rj@gmail.com
8686861234 HM8686861234 Ram Hyd rm@gmail.com
0x45

DBCon1.java

```
package test;
import java.sql.*;
public class DBCon1
{
	public static void main(String[] args)
	{
	  try
	  {
```

```
            //step-1 : Loader driver
            Class.forName("oracle.jdbc.driver.OracleDriver");
            //step-2 : Creating Connection to Database Product
            Connection con = DriverManager.getConnection
                        ("jdbc:oracle:thin:@localhost:1521:xe",
                                    "system","tiger");
           //step-3 : preparing JDBC-statement
            Statement stm = con.createStatement();
           //step-4 : Executing the query
            ResultSet rs = stm.executeQuery("select * from Customer72");
            while(rs.next())
            {
                  System.out.println(rs.getLong(1)+"\t"
                              +rs.getString(2)+"\t"+
                              rs.getString(3)+"\t"+
                              rs.getString(4)+"\t"+
                              rs.getString(5));
            }//end of loop
            //step-5 : Closing the connection from Database
            con.close();
        }//end of try
        catch(Exception e)
        {
            e.printStackTrace();
        }
      }
}
```

Venkatesh