============================

**DataBase\_Con\_Select\_Command**

package SQLConnection;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import org.testng.annotations.Test;

/\*1) Create a connection

2) Create statement /Query

3) Execute statement /Query

4) Store the results in result set

5) Close connection and

6) After completion of this, if you want to write simple confirmation message - syso

\*/

public class DataBase\_Con\_Select\_Command {

@Test

public void DataBaseSelectCommand() throws SQLException {

// 1) Create a connection to the Database // here 'hr' = database . we will retrieve the data from hr database

Connection connect =DriverManager.getConnection("jdbc:mysql://localhost:3306/hr","root","mysql123");

//2 )Create statement /Query-

Statement statement=connect.createStatement();

//These query will return the data and I execute it, it will give RequiredSet or ResultSet

String s = "select \* from Customer"; // all data or

String s1 = "select column1, column 2 from Customer"; //Specific column or

String s2 = "select employee\_id,first\_name, last\_name from Customer"; // Capturing three different column

// to store the data we need to use a special class called as ResultSet

// 3) Execute statement /Queryand and 4)Store the results in result set

ResultSet resultset =statement.executeQuery("s");// =================1) First Query)

resultset.next();

// here 2 diff columns and n number of Rows.So I want to get all the columns and all the rows.

//that i will use loop statement

ResultSet rs1 =statement.executeQuery("s1"); // =================2) 2nd Query)

while(rs1.next()) {

rs1.getString("ColumnName1");

rs1.getString("ColumnName2");

} // if it returns true

ResultSet rs2 =statement.executeQuery("s2");//=================1) 3rd Query)

// This particular loop will be repeat multiple time till the last info and this rs contains data .

while(rs2.next()) {

int eid= rs1.getInt("EmployeeID");

String firstname= rs1.getString("First\_Name");

String lastName =rs1.getString("Last\_Name");

System.out.println(eid+" "+firstname+" "+lastName); // print as a singly

}

//4) Close connection {So connection name is 'connect'}

connect.close();

System.out.println("Query Executed ----");

}}

// In this loop we are reading the from this "rs"/ resultset/ rs1/ rs2. and printing the data into the consoles

========================================================

==============================================================================



====================

create table Employee

(

EmpID int,

EmpName Varchar (255),

Age int,

Phone int,

Addrress Varchar(255),

Email address Varchar(255)

);

insert into Employee values(1, "Tom", 25, 907548, "12th ave Brooklyn nyc", "tom@gmail.com");

insert into Employee values(2, "Toni", 29, 9075481, "13th ave Brooklyn nyc", "toni@gmail.com");

insert into Employee values(3, "Tono", 27, 9075482, "14th ave Brooklyn nyc", "tono@gmail.com");

insert into Employee values(4, "Tono", 20, 12075482, "15th ave Brooklyn nyc", "tono@gmail.com");

insert into Employee values(5, "Tom", 30, 9075948, "116th ave Brooklyn nyc", "tom@gmail.com");

insert into Employee values(5, "Tom", 30, 9075948, "116th ave Brooklyn nyc", "tom@gmail.com");

-- Asscending order and Dcending order

-- select \* from Employee order by EmpID ASC;

-- select \* from Employee order by age DESC;

-- select \* from Employee order by EmpName, Age;

-- And/or/not operator

-- select \* from Employee where Age>30 and EmpID > 2;

-- select \* from Employee where Age>25 and EmpID>2 ;

-- select \* from Employee where Age<22 or EmpID<5;

-- select \* from Employee where Age<22 or EmpID<5 and Phone=12075482;

select \* from Employee where Age<29 or EmpID<5 and Phone=12075482;

create table Employee

(

EmpId int,

EmpName Varchar (255),

Age int,

Phone int,

Addrress Varchar(255),

Email address Varchar(255)

);

insert into Employee values(1, "Tom", 25, 907548, "12th ave Brooklyn nyc", "tom@gmail.com");

insert into Employee values(2, "Toni", 29, 9075481, "13th ave Brooklyn nyc", "toni@gmail.com");

insert into Employee values(3, "Tono", 27, 9075482, "14th ave Brooklyn nyc", "tono@gmail.com");

insert into Employee values(4, "Tono", 22, 12075482, "15th ave Brooklyn nyc", "tono@gmail.com");

insert into Employee values(5, "Tom", 30, 9075948, "116th ave Brooklyn nyc", "tom@gmail.com");

insert into Employee values(5, "Tom", 30, 9075948, "116th ave Brooklyn nyc", "tom@gmail.com");

-- same value and everything in two or three or more . but it will give only one distict value not duplicate values

-- select \* from Employee;

-- select count(\*) from Employee;

-- select \* from employee where EmpName = "Tom";

-- select \* from employee where EmpName = "Ruma";

--

-- Two filter

-- select \* from employee where EmpName = "Tono" and age =27;

-- select \* from employee where Age> 27 and EmpName ="Tom";

-- select distinct \* from Employee;

-- select distinct count(\*) from Employee;

-- select count(distinct EmpName) from Employee;

-- select count(distinct EmpID) from Employee;

-- select \* from Employee order by EmpID ASC;

-- select \* from Employee order by EmpID DESC;

-- select \* from Employee order by Age DESC;

-- select \* from Employee order by Age DESC;

select \* from Employee order by EmpName, Age ;

Select , Distinct, Where ,

Java DataBase Connectivity

How JDBC works?

JDBC is a process to connect to the Database with Java.

At first we have to add a Jar file (mysql connector java/JDBC for oracle). As we know Java cannot connect with Database directly, for that we need a driver class as an interpreter.