**Clase n°13, ejercicios (guía ejercicio 13)**

**Autor: Vargas Diego**

Ejercicio 1:

package Conexion;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.util.ResourceBundle;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author diego vargas

\*/

public class Conex\_Bd {

private static Connection conec = null;

public static Connection devolver\_Conexion()

{

try

{

if (conec == null)

{

Runtime.getRuntime().addShutdownHook(new MiShDwnHook());

ResourceBundle rb = ResourceBundle.getBundle("Conexion.Datos\_Conexion");

String driver= rb.getString("driver");

String url= rb.getString("url");

String user= rb.getString("user");

String pass= rb.getString("pass");

Class.forName(driver);

conec = DriverManager.getConnection(url, user, pass);

JOptionPane.showMessageDialog(null, "Se establecio la conexion a la BD");

}

return conec;

}

catch (ClassNotFoundException| SQLException ex)

{

ex.printStackTrace(System.out);

throw new RuntimeException("no se pudo realizar la conexion con la bd", ex);

}

}

static class MiShDwnHook extends Thread {

@Override

public void run() {

try

{

conec = Conex\_Bd.devolver\_Conexion();

conec.close();

}

catch (SQLException ex)

{

ex.printStackTrace(System.out);

throw new RuntimeException(ex);

}

}

}

}

Datos archive .properties

driver: com.mysql.cj.jdbc.Driver

url: jdbc:mysql://localhost:3306/qatar2022?autoReconnect=true&useSSL=true

user:root

pass:123456

import Conexion.Conex\_Bd;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author diego vargas

\*/

public class Consultas\_Bd\_App {

public static void main(String[] args) {

Empleado empleado\_1= new Empleado(30789234,"juan","coceres","peruana",1);

Departamento depto= new Departamento("compras");

insertar\_Empleado(empleado\_1);

Empleado empleado\_2= new Empleado(30789234,"juan","coceres","brasilera",1);

actualizar\_Empleado(empleado\_2);

eliminar\_Departamento(depto);

mostrar\_Empleados();

mostrar\_Departamentos();

}

public static boolean insertar\_Empleado(Empleado emp){

Connection con=null;

PreparedStatement ps=null;

try {

con = Conex\_Bd.devolver\_Conexion();

String sent\_sql = "insert into empleado(dni,nombre,apellido,nacionalidad,id\_depto)";

sent\_sql += "values (?,?,?,?,?) ";

ps = con.prepareStatement(sent\_sql);

ps.setInt(1, emp.getDni());

ps.setString(2, emp.getNombre());

ps.setString(3, emp.getApellido());

ps.setString(4, emp.getNacionalidad());

ps.setInt(5, emp.getDepartamento());

ps.execute();

return true;

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "error,los datos no fueron insertados en la bd" + ex.getMessage());

} finally {

if (ps != null && con!=null) {

ps.close();

con.close();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "se genero un error y nose pudo cerrar la conexion" + e.getMessage());

}

}

return false;

}

public static boolean actualizar\_Empleado(Empleado emp){

PreparedStatement ps = null;

Connection con = null;

try {

con = Conex\_Bd.devolver\_Conexion();

String sent\_sql = "update empleado set nacionalidad=? ";

sent\_sql += "where dni = '" + emp.getDni() + "'";

ps = con.prepareStatement(sent\_sql);

ps.setString(1, emp.getNacionalidad());

ps.executeUpdate();

return true;

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, " no sé actualizaron los datos en la base de datos"+ " " + ex.getMessage());

} finally {

try {

if (ps != null && con!=null) {

ps.close();

con.close();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "se genero un error y no sé pudo cerrar la conexion " + " " + e.getMessage());

}

}

return false;

}

public static boolean eliminar\_Departamento(Departamento depto) {

PreparedStatement ps = null;

Connection con = null;

try {

con = Conex\_Bd.devolver\_Conexion();

String sent\_sql = "delete from departamento where desc\_dept = '" + depto.getDescripcion\_depto() + "'";

ps = con.prepareStatement(sent\_sql);

ps.execute();

return true;

} catch (SQLException ex) {

JOptionPane.showMessageDialog(null, "no sé pudo realizar la consulta en la base de datos, registro sin modificaciones " + " " + ex);

} finally {

try {

if (ps != null && con!=null) {

ps.close();

con.close();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, " no sé pudo realizar el cierre de la conexion" + " " + e.getMessage());

}

}

return false;

}

public static boolean mostrar\_Empleados(){

PreparedStatement ps=null;

ResultSet rs=null;

Connection con =null;

try{

con=Conex\_Bd.devolver\_Conexion();

String sent\_sql= "select dni, nombre, apellido, nacionalidad, d.desc\_dept ";

sent\_sql += "from empleado as e inner join departamento as d on e.id\_depto= d.id\_departamento ";

sent\_sql += "where d.desc\_dept= 'logistica' ";

ps=con.prepareStatement(sent\_sql);

rs= ps.executeQuery();

JOptionPane.showMessageDialog(null, "a continuacion se muestran los empleados del departamento de logistica");

while(rs.next()){

System.out.println("mostrando empleados que pertenecen al departamento de logistica:" + " " + rs.getInt(1) + " " + rs.getString(2) + " " + rs.getString(3) +

" " + rs.getString(4) + " " + rs.getString(5));

}

JOptionPane.showMessageDialog(null, "no hay mas empleados que pertenezcan al departamento de logistica...");

return true;

}catch(SQLException ex){

JOptionPane.showMessageDialog(null, "no sé pudo realizar la consulta en la base de datos, registro sin modificaciones " + " " + ex);

} finally {

try {

if (ps != null && con!=null) {

ps.close();

con.close();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, " no sé pudo realizar el cierre del preparedStatement" + " " + e.getMessage());

}

return false;

}

}

public static boolean mostrar\_Departamentos(){

PreparedStatement ps=null;

ResultSet rs=null;

Connection con =null;

try{

con=Conex\_Bd.devolver\_Conexion();

String sent\_sql= "select desc\_dept ";

sent\_sql += "from departamento order by desc\_dept asc";

ps=con.prepareStatement(sent\_sql);

rs= ps.executeQuery();

JOptionPane.showMessageDialog(null, "a continuacion se muestran los departamentos ordenados alfabeticamente");

while(rs.next()){

System.out.println("mostrando los departamentos cargados en la bd:" + " " + rs.getString(1));

}

JOptionPane.showMessageDialog(null, "no hay mas departamentos para mostrar.....");

return true;

}catch(SQLException ex){

JOptionPane.showMessageDialog(null, "no sé pudo realizar la consulta en la base de datos, registro sin modificaciones " + " " + ex);

} finally {

try {

if (ps != null && con!=null) {

ps.close();

con.close();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, " no sé pudo realizar el cierre del preparedStatement" + " " + e.getMessage());

}

return false;

}

}

}

public class Departamento {

private String descripcion\_depto;

public Departamento(){}

public Departamento(String depto){

this.descripcion\_depto=depto;

}

public String getDescripcion\_depto() {

return descripcion\_depto;

}

}

public class Empleado {

private int dni;

private String nombre;

private String apellido;

private String nacionalidad;

private int departamento;

public Empleado(){}

public Empleado(int documento,String nomb,String ape,String nacionalidad,int depto){

this.dni=documento;

this.nombre=nomb;

this.apellido=ape;

this.nacionalidad=nacionalidad;

this.departamento=depto;

}

public int getDni() {

return dni;

}

public String getNombre() {

return nombre;

}

public String getApellido() {

return apellido;

}

public String getNacionalidad() {

return nacionalidad;

}

public int getDepartamento() {

return departamento;

}

public void setNacionalidad(String nacionalidad) {

this.nacionalidad = nacionalidad;

}

}

Enlace GitHub: