Trabajo Practico Final (Consultorio)

alumno: Santillán Rafael Gabriel

Dias y horarios: martes y jueves de 19hs a 21hs

repositorio:https://github.com/santillan-rafael-gabriel/tpfinal-consultorio.git

Se presenta el croquis del esquema de la base de datos a crear en la pagina lucid.



Despues se empieza a armar el codigo para crear la base de datos del consultorio en el programa mysql:

primero el codigo:

primero creamos la base de datos que vamos a trabajar

create database consultorio;

-------------------------

despues empezamos a crear la tabla de medico

tabla de medico

create table Medico(

MedMat int(10) not null,

MedNom varchar(60) not null,

MedApe varchar(60) not null,

MedEsp varchar(120) not null,

primary key(MedMat)) Engine=InnoDB;

--------------------------

siguiente creamos la tabla de pacientes

tabla de paciente

create table paciente(

PacDni int(8) not null,

PacApe varchar(50) not null,

PacNom varchar(50) not null,

primary key(PacDni)) Engine=InnoDB;

--------------------------------------

y para finalizar creamos la tabla de historia clinica

tabla de hc

create table HC(

HcNum int(10) not null,

HcFec varchar(11) not null,

HcPacDni int(8) not null,

HcMedMat int(10) not null,

HcTrat varchar(120) not null,

primary key (HcNum),

foreign key (HcPacDni) references paciente(PacDni) on delete no action on update no action,

foreign key (HcMedMat) references medico(MedMat) on delete no action on update no action

) engine=InnoDB;

---------------------------------------------------------------

Se presenta el esquema de la base de datos de mysql de como queda terminada



despues empezamos a elaborar el codigo en eclipce para dar vida a la base de datos del consultorio y funcione.

el siguiente es el codigo del elaboracion del programa:

""pero primero tiene que tener instalado el driver de mysql-connector-java para que pueda funcionar el programa conectada a la base de datos""

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.InputMismatchException;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

public class main1{

Connection conexion = null;

Statement sentencia = null;

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

main1 m = new main1();

m.conectar();

boolean salir = false;

do {

switch (menuPrin()) {

case 1:

m.consultarPaciente();

break;

case 2: m.consultarMedico();

break;

case 3:

m.agregarPac();

break;

case 4:

m.agregarMed();

break;

case 5: m.consultarHistoria();

break;

case 6: m.agregarHC();

break;

case 0:

System.out.println("Vuelva pronto");

m.desconectar();

salir = true;

break;

default:

System.out.println("Opción incorrecta");

break;

}

} while (!salir);

}

//-------------------------------------------------------------------------------

//MENU PRINCIPAL:

private static int menuPrin() {

Scanner sc = new Scanner(System.in);

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

System.out.println("Conexión de bbdd MySQL");

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

System.out.println("1.MOSTRAR EL CONTENIDO DE LA TABLA PACIENTES");

System.out.println("2.MOSTRAR EL CONTENIDO DE LA TABLA MEDICOS");

System.out.println("3.INSERTAR UN REGISTRO EN LA TABLA PACIENTES");

System.out.println("4.INSERTAR UN REGISTRO EN LA TABLA MEDICOS");

System.out.println("5.MOSTRAR EL CONTENIADO DE LA HISTORIA CLINICA");

System.out.println("6.INSERTAR UN REGISTRO EN LA TABLA HISTORIA CLINICA");

System.out.println("0.SALIR");

System.out.println("\n Por favor, escoja una opción.");

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

return sc.nextInt();

}

//-----------------------------------------------------------------------------------------------

public void conectar() {

try {

Class.forName("com.mysql.jdbc.Driver");

conexion = DriverManager.getConnection("jdbc:mysql://localhost/consultorio", "root", "Santi$590");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" \* CONEXIÓN REALIZADA CORRECTAMENTE \* ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

} catch (Exception e) {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" \* NO SE HA PODIDO REALIZAR LA CONEXIÓN \* ");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

//-----------------------------------------------------------------------------------------------

private void desconectar() {

try {

conexion.close();

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

System.out.println("La conexion a la base de datos se ha terminado");

System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

} catch (SQLException ex) {

System.out.println(ex.getMessage());

}

}

//----------------------------------------------------------------------------------------------

/\* METODO PARA REALIZAR UNA CONSULTA A LA TABLA HISTORIA CLINICA\*/

private void consultarHistoria() {

ResultSet r = buscar2("select HcNum,HcFec,HcPacDni,HcMedMat,HcTrat from hc");

try {

System.out.println("REGISTROS DE LA TABLA HITORIA CLINICA");

while (r.next()) {

System.out.println(r.getInt("HcNum") + " | " + r.getString("HcFec") + " | " + r.getInt("HcPacDni") + " | " + r.getString("HcMedMat") + " | " + r.getString("HcTrat"));

}

} catch (SQLException ex) {

Logger.getLogger(main1.class.getName()).log(Level.SEVERE, null, ex);

}

}

ResultSet buscar2(String sql) {

try {

sentencia = conexion.createStatement();

return sentencia.executeQuery(sql);

} catch (SQLException ex) {

Logger.getLogger(main1.class.getName()).log(Level.SEVERE, null, ex);

}

return null;

}

//----------------------------------------------------------------------------------------------------

/\*MÉTODO PARA REALIZAR UNA CONSULTA A LA TABLA PACIENTES\*/

private void consultarPaciente() {

ResultSet r = buscar("select PacDni,PacNom,PacApe from paciente");

try {

System.out.println("REGISTROS DE LA TABLA PACIENTES");

while (r.next()) {

System.out.println(r.getInt("PacDni") + " | " + r.getString("PacNom") + " | " + r.getString("PacApe"));

}

} catch (SQLException ex) {

Logger.getLogger(main1.class.getName()).log(Level.SEVERE, null, ex);

}

}

ResultSet buscar(String sql) {

try {

sentencia = conexion.createStatement();

return sentencia.executeQuery(sql);

} catch (SQLException ex) {

Logger.getLogger(main1.class.getName()).log(Level.SEVERE, null, ex);

}

return null;

}

////////////////////////////////////////////////////////////////////////////////

/\*MÉTODO PARA REALIZAR UNA CONSULTA A LA TABLA MEDICOS\*/

private void consultarMedico() {

ResultSet r = buscar1("select MedMat,MedNom,MedApe,MedEsp from medico");

try {

System.out.println("REGISTROS DE LA TABLA MEDICOS");

while (r.next()) {

System.out.println(r.getInt("MedMat") + " | " + r.getString("MedNom") + " | " + r.getString("MedApe") + " | " + r.getString("MedEsp"));

}

} catch (SQLException ex) {

Logger.getLogger(main1.class.getName()).log(Level.SEVERE, null, ex);

}

}

ResultSet buscar1(String sql) {

try {

sentencia = conexion.createStatement();

return sentencia.executeQuery(sql);

} catch (SQLException ex) {

Logger.getLogger(main1.class.getName()).log(Level.SEVERE, null, ex);

}

return null;

}

////////////////////////////////////////////////////////////////////////////

/\*MÉTODO PARA AGREGAR UN PACIENTE A LA BASE DE DATOS (CONSULTORIO) MYSQL\*/

private void agregarPac() {

String usuario = "root";

String password = "Santi$590";

Scanner sc = new Scanner(System.in);

try {

System.out.println("Escriba el DNI del Paciente: ");

int PacDni = sc.nextInt();

sc.nextLine();

System.out.println("Ingrese el nombre del Paciente: ");

String PacNom = sc.nextLine();

System.out.println("Ingrese el apellido del Paciente: ");

String PacApe = sc.nextLine();

//System.out.println("Ingrese el domicilio del Paciente: ");

// String PacDom = sc.nextLine();

if (PacNom.isEmpty() || PacApe.isEmpty()) {

throw new IllegalArgumentException("El nombre , el apellido o el domicilio estan vacios por favor ingrese de vuelta todo");

}

String sql = "insert into paciente (PacDni,PacNom,PacApe) values ('" + PacDni + "','" + PacNom + "','"+ PacApe+"')";

try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/consultorio", usuario, password);

Statement sentencia = con.createStatement()) {

int m = sentencia.executeUpdate(sql);

if (m == 1)

System.out.println("Se realizó correctamente la inserción: " + sql);

else

System.out.println("Falló la inserción");

} catch (SQLException e) {

System.out.println("Error al conectar con la base de datos.");

e.printStackTrace();

}

} catch (InputMismatchException e) {

System.out.println("Error: Debe ingresar un número para el DNI.");

sc.nextLine();

} catch (IllegalArgumentException e) {

System.out.println("Error: " + e.getMessage());

}

}

////////////////////////////////////////////////////////////////////////////

/\*MÉTODO PARA AGREGAR UN MEDICO A LA BASE DE DATOS (CONSULTORIO) MYSQL\*/

private void agregarMed() {

String usuario = "root";

String password = "Santi$590";

Scanner sc = new Scanner(System.in);

try {

System.out.println("Escriba la matricula del medico: ");

int MedMat = sc.nextInt();

sc.nextLine();

System.out.println("Ingrese el nombre del Medico: ");

String MedNom= sc.nextLine();

System.out.println("Ingrese el apellido del Medico: ");

String MedApe = sc.nextLine();

System.out.println("Ingrese la especialidad del medico: ");

String MedEsp = sc.nextLine();

if (MedNom.isEmpty() || MedApe.isEmpty()||MedEsp.isEmpty()) {

throw new IllegalArgumentException("El nombre , el apellido o la especialidad estan vacios");

}

String sql = "insert into medico (MedMat,MedNom,MedApe,MedEsp) values ('" + MedMat + "','" + MedNom + "','" + MedApe + "','" + MedEsp + "')";

try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/consultorio", usuario, password);

Statement sentencia = con.createStatement()) {

int m = sentencia.executeUpdate(sql);

if (m == 1)

System.out.println("Se realizó correctamente la inserción: " + sql);

else

System.out.println("Falló la inserción");

} catch (SQLException e) {

System.out.println("Error al conectar con la base de datos.");

e.printStackTrace();

}

} catch (InputMismatchException e) {

System.out.println("Error: Debe ingresar un número de matricula.");

sc.nextLine();

} catch (IllegalArgumentException e) {

System.out.println("Error: " + e.getMessage());

}

}

//-----------------------------------------------------------------------------------------------------

/\*MÉTODO PARA AGREGAR UN historia clinica A LA BASE DE DATOS (CONSULTORIO) MYSQL\*/

private void agregarHC() {

String usuario = "root";

String password = "Santi$590";

Scanner sc = new Scanner(System.in);

try {

System.out.println("Escriba el numero de historia clinica: ");

int HcNum = sc.nextInt();

sc.nextLine();

System.out.println("Escriba el DNI del Paciente: ");

int PacDni2 = sc.nextInt();

sc.nextLine();

System.out.println("Ingrese fecha: ");

String HcFec = sc.nextLine();

System.out.println("ingrese matricula del medico: ");

String HcMedMat = sc.nextLine();

System.out.println("ingrese tratamiento: ");

String HcTrat = sc.nextLine();

if (HcMedMat.isEmpty() || HcTrat.isEmpty()) {

throw new IllegalArgumentException("El nombre , el apellido o el domicilio estan vacios por favor ingrese de vuelta todo");

}

String sql = "insert into hc (HcNum,HcFec,HcPacDni,HcMedMat,HcTrat) values ('" + HcNum + "','" + HcFec + "','" + PacDni2 + "','" + HcMedMat +"','" + HcTrat +"')";

try (Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/consultorio", usuario, password);

Statement sentencia = con.createStatement()) {

int m = sentencia.executeUpdate(sql);

if (m == 1)

System.out.println("Se realizó correctamente la inserción: " + sql);

else

System.out.println("Falló la inserción");

} catch (SQLException e) {

System.out.println("Error al conectar con la base de datos.");

e.printStackTrace();

}

} catch (InputMismatchException e) {

System.out.println("Error: Debe ingresar un número para el DNI.");

sc.nextLine();

} catch (IllegalArgumentException e) {

System.out.println("Error: " + e.getMessage());

}

}

}