

**Trabajo Práctico de Bases de Datos I**

Alumnos: Franco Villan y Lautaro Stangaferro

Materia: Base De datos.

Profesor Federico Malfasi.

Carrera: Tec. En Programacion.

Fecha maxima de entrega 04/11/2024

**CODIGO DE LARAGON (CMD)**

--Base de datos Practicoo

CREATE DATABASE Practicoo;

-- Tabla Usuarios

CREATE TABLE Usuarios (

id INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50) NOT NULL,

apellido VARCHAR(50) NOT NULL,

dni VARCHAR(15) UNIQUE NOT NULL,

telefono VARCHAR(15),

email VARCHAR(100),

creado\_el TIMESTAMP DEFAULT NOW(),

actualizado\_el TIMESTAMP DEFAULT NOW() ON UPDATE NOW(),

estado TINYINT DEFAULT 1

);

-- Tabla Géneros

CREATE TABLE Generos (

id INT PRIMARY KEY AUTO\_INCREMENT,

genero VARCHAR(50) UNIQUE NOT NULL

);

-- Tabla Libros

CREATE TABLE Libros (

id INT PRIMARY KEY AUTO\_INCREMENT,

nombre\_libro VARCHAR(100) NOT NULL,

autor VARCHAR(100),

fecha\_lanzamiento DATE,

id\_genero INT,

creado\_el TIMESTAMP DEFAULT NOW(),

actualizado\_el TIMESTAMP DEFAULT NOW() ON UPDATE NOW(),

estado TINYINT DEFAULT 1,

CONSTRAINT fk\_libros\_genero FOREIGN KEY (id\_genero) REFERENCES Generos(id)

);

-- Tabla Préstamos

CREATE TABLE Prestamos (

id INT PRIMARY KEY AUTO\_INCREMENT,

usuario\_id INT,

libro\_id INT,

fecha\_prestamo TIMESTAMP DEFAULT NOW(),

fecha\_devolucion\_estimada TIMESTAMP,

fecha\_devolucion\_real TIMESTAMP,

CONSTRAINT fk\_prestamos\_usuario FOREIGN KEY (usuario\_id) REFERENCES Usuarios(id),

CONSTRAINT fk\_prestamos\_libro FOREIGN KEY (libro\_id) REFERENCES Libros(id)

);

**CODIGO EN C#**

using System;

using MySql.Data.MySqlClient;

namespace GestorBiblioteca

{

class Program

{

static string connectionString = "server=localhost;database=practicoo;user=root;password=;";

static void Main(string[] args)

{

int option;

do

{

Console.WriteLine("Gestor de Biblioteca");

Console.WriteLine("1. Crear Usuario");

Console.WriteLine("2. Actualizar Estado de Usuario");

Console.WriteLine("3. Inactivar Usuario");

Console.WriteLine("4. Agregar Libro");

Console.WriteLine("5. Actualizar Estado de Libro");

Console.WriteLine("6. Inactivar Libro");

Console.WriteLine("7. Agregar Género");

Console.WriteLine("8. Actualizar Género");

Console.WriteLine("9. Crear Préstamo");

Console.WriteLine("10. Actualizar Fecha de Devolución del Préstamo");

Console.WriteLine("11. Salir");

Console.Write("Seleccione una opción: ");

if (!int.TryParse(Console.ReadLine(), out option))

{

Console.WriteLine("Opción no válida.");

continue;

}

switch (option)

{

case 1: CrearUsuario(); break;

case 2: ActualizarEstadoUsuario(); break;

case 3: InactivarUsuario(); break;

case 4: AgregarLibro(); break;

case 5: ActualizarEstadoLibro(); break;

case 6: InactivarLibro(); break;

case 7: AgregarGenero(); break;

case 8: ActualizarGenero(); break;

case 9: CrearPrestamo(); break;

case 10: ActualizarFechaDevolucionPrestamo(); break;

case 11: Console.WriteLine("Saliendo..."); break;

default: Console.WriteLine("Opción no válida."); break;

}

} while (option != 11);

}

static void CrearUsuario()

{

Console.Write("Ingrese nombre: ");

string nombre = Console.ReadLine();

Console.Write("Ingrese apellido: ");

string apellido = Console.ReadLine();

Console.Write("Ingrese DNI: ");

string dni = Console.ReadLine();

Console.Write("Ingrese teléfono: ");

string telefono = Console.ReadLine();

Console.Write("Ingrese email: ");

string email = Console.ReadLine();

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "INSERT INTO Usuarios (nombre, apellido, dni, telefono, email) VALUES (@nombre, @apellido, @dni, @telefono, @correo)";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@nombre", nombre);

cmd.Parameters.AddWithValue("@apellido", apellido);

cmd.Parameters.AddWithValue("@dni", dni);

cmd.Parameters.AddWithValue("@telefono", telefono);

cmd.Parameters.AddWithValue("@correo", email);

cmd.ExecuteNonQuery();

Console.WriteLine("Usuario creado correctamente.");

}

}

static void ActualizarEstadoUsuario()

{

Console.Write("Ingrese ID del usuario a actualizar: ");

int id = int.Parse(Console.ReadLine());

Console.Write("Ingrese nuevo estado (1=activo, 0=inactivo): ");

int estado = int.Parse(Console.ReadLine());

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "UPDATE Usuarios SET estado = @estado, actualizado\_el = NOW() WHERE id = @id";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@estado", estado);

cmd.Parameters.AddWithValue("@id", id);

cmd.ExecuteNonQuery();

Console.WriteLine("Estado del usuario actualizado correctamente.");

}

}

static void InactivarUsuario()

{

Console.Write("Ingrese ID del usuario a inactivar: ");

int id = int.Parse(Console.ReadLine());

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "UPDATE Usuarios SET estado = 0, actualizado\_el = NOW() WHERE id = @id";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@id", id);

cmd.ExecuteNonQuery();

Console.WriteLine("Usuario inactivado correctamente.");

}

}

static void AgregarLibro()

{

Console.Write("Ingrese nombre del libro: ");

string nombreLibro = Console.ReadLine();

Console.Write("Ingrese autor: ");

string autor = Console.ReadLine();

Console.Write("Ingrese fecha de lanzamiento (YYYY-MM-DD): ");

string fechaLanzamiento = Console.ReadLine();

Console.Write("Ingrese ID del género: ");

int idGenero = int.Parse(Console.ReadLine());

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "INSERT INTO Libros (nombre\_libro, autor, fecha\_lanzamiento, id\_genero) VALUES (@nombre, @autor, @fecha, @id\_genero)";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@nombre", nombreLibro);

cmd.Parameters.AddWithValue("@autor", autor);

cmd.Parameters.AddWithValue("@fecha", fechaLanzamiento);

cmd.Parameters.AddWithValue("@id\_genero", idGenero);

cmd.ExecuteNonQuery();

Console.WriteLine("Libro agregado correctamente.");

}

}

static void ActualizarEstadoLibro()

{

Console.Write("Ingrese ID del libro a actualizar: ");

int id = int.Parse(Console.ReadLine());

Console.Write("Ingrese nuevo estado (1=activo, 0=inactivo): ");

int estado = int.Parse(Console.ReadLine());

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "UPDATE Libros SET estado = @estado, actualizado\_el = NOW() WHERE id = @id";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@estado", estado);

cmd.Parameters.AddWithValue("@id", id);

cmd.ExecuteNonQuery();

Console.WriteLine("Estado del libro actualizado correctamente.");

}

}

static void InactivarLibro()

{

Console.Write("Ingrese ID del libro a inactivar: ");

int id = int.Parse(Console.ReadLine());

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "UPDATE Libros SET estado = 0, actualizado\_el = NOW() WHERE id = @id";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@id", id);

cmd.ExecuteNonQuery();

Console.WriteLine("Libro inactivado correctamente.");

}

}

static void AgregarGenero()

{

Console.Write("Ingrese nombre del género: ");

string genero = Console.ReadLine();

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "INSERT INTO Generos (genero) VALUES (@genero)";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@genero", genero);

cmd.ExecuteNonQuery();

Console.WriteLine("Género agregado correctamente.");

}

}

static void ActualizarGenero()

{

Console.Write("Ingrese ID del género a actualizar: ");

int id = int.Parse(Console.ReadLine());

Console.Write("Ingrese nuevo nombre del género: ");

string nuevoGenero = Console.ReadLine();

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "UPDATE Generos SET genero = @nuevoGenero WHERE id = @id";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@nuevoGenero", nuevoGenero);

cmd.Parameters.AddWithValue("@id", id);

cmd.ExecuteNonQuery();

Console.WriteLine("Género actualizado correctamente.");

}

}

static void CrearPrestamo()

{

Console.Write("Ingrese ID del usuario: ");

int usuarioId = int.Parse(Console.ReadLine());

Console.Write("Ingrese ID del libro: ");

int libroId = int.Parse(Console.ReadLine());

Console.Write("Ingrese fecha de devolución estimada (YYYY-MM-DD): ");

string fechaDevolucionEstimada = Console.ReadLine();

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

// Validación de estado de usuario

string checkUsuario = "SELECT estado FROM Usuarios WHERE id = @usuarioId";

MySqlCommand cmdCheckUsuario = new MySqlCommand(checkUsuario, connection);

cmdCheckUsuario.Parameters.AddWithValue("@usuarioId", usuarioId);

int estadoUsuario = Convert.ToInt32(cmdCheckUsuario.ExecuteScalar());

// Validación de estado de libro

string checkLibro = "SELECT estado FROM Libros WHERE id = @libroId";

MySqlCommand cmdCheckLibro = new MySqlCommand(checkLibro, connection);

cmdCheckLibro.Parameters.AddWithValue("@libroId", libroId);

int estadoLibro = Convert.ToInt32(cmdCheckLibro.ExecuteScalar());

if (estadoUsuario == 1 && estadoLibro == 1)

{

// Si ambos estados son activos, se crea el préstamo

string query = "INSERT INTO Prestamo (usuario\_id, libro\_id, fecha\_devolucion\_estimada, creado\_el) VALUES (@usuarioId, @libroId, @fechaDevolucion, NOW())";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@usuarioId", usuarioId);

cmd.Parameters.AddWithValue("@libroId", libroId);

cmd.Parameters.AddWithValue("@fechaDevolucion", fechaDevolucionEstimada);

cmd.ExecuteNonQuery();

Console.WriteLine("Préstamo creado correctamente.");

}

else

{

if (estadoUsuario == 0)

{

Console.WriteLine("No se puede crear el préstamo. El usuario está inactivo.");

}

if (estadoLibro == 0)

{

Console.WriteLine("No se puede crear el préstamo. El libro está inactivo.");

}

}

}

}

static void ActualizarFechaDevolucionPrestamo()

{

Console.Write("Ingrese ID del préstamo a actualizar: ");

int prestamoId = int.Parse(Console.ReadLine());

Console.Write("Ingrese nueva fecha de devolución (YYYY-MM-DD): ");

string nuevaFechaDevolucion = Console.ReadLine();

using (MySqlConnection connection = new MySqlConnection(connectionString))

{

connection.Open();

string query = "UPDATE Prestamo SET fecha\_devolucion\_estimada = @nuevaFecha, actualizado\_el = NOW() WHERE id = @prestamoId";

MySqlCommand cmd = new MySqlCommand(query, connection);

cmd.Parameters.AddWithValue("@nuevaFecha", nuevaFechaDevolucion);

cmd.Parameters.AddWithValue("@prestamoId", prestamoId);

cmd.ExecuteNonQuery();

Console.WriteLine("Fecha de devolución del préstamo actualizada correctamente.");

}

}

}

}