### Requisitos Previos

Para hacer este ejercicio se supone que:

* Tienes una base de datos MySQL llamada miBaseDatos.
* En la base de datos tienes una tabla llamada usuarios con la siguiente estructura:

Para crearte la tabla “usuarios”:

CREATE TABLE usuarios (

id INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(255) NOT NULL,

email VARCHAR(255) NOT NULL

);

### Ejercicio Completo de operaciones CRUD con JDBC

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class CrudJDBCExample {

// Configuración de la base de datos

static final String DB\_URL = "jdbc:mysql://localhost:3306/miBaseDatos";

static final String USER = "root";

static final String PASSWORD = "tu\_contraseña";

public static void main(String[] args) {

try {

// Crear (Insert)

insertarUsuario("Juan Pérez", "juan.perez@gmail.com");

insertarUsuario("Maria López", "maria.lopez@gmail.com");

// Leer (Select)

leerUsuarios();

// Actualizar (Update)

actualizarUsuario(1, "Juan Actualizado", "juan.actualizado@gmail.com");

// Eliminar (Delete)

eliminarUsuario(2);

// Leer nuevamente para verificar los cambios

leerUsuarios();

} catch (SQLException e) {

e.printStackTrace();

}

}

// Método para crear un nuevo usuario (Create)

public static void insertarUsuario(String nombre, String email) throws SQLException {

String insertSQL = "INSERT INTO usuarios (nombre, email) VALUES (?, ?)";

try (Connection con = DriverManager.getConnection(DB\_URL, USER, PASSWORD);

PreparedStatement pstmt = con.prepareStatement(insertSQL)) {

pstmt.setString(1, nombre);

pstmt.setString(2, email);

int filasAfectadas = pstmt.executeUpdate();

System.out.println("Usuario insertado, filas afectadas: " + filasAfectadas);

}

}

// Método para leer todos los usuarios (Read)

public static void leerUsuarios() throws SQLException {

String selectSQL = "SELECT \* FROM usuarios";

try (Connection con = DriverManager.getConnection(DB\_URL, USER, PASSWORD);

PreparedStatement pstmt = con.prepareStatement(selectSQL);

ResultSet rs = pstmt.executeQuery()) {

System.out.println("Lista de usuarios:");

while (rs.next()) {

int id = rs.getInt("id");

String nombre = rs.getString("nombre");

String email = rs.getString("email");

System.out.println("ID: " + id + ", Nombre: " + nombre + ", Email: " + email);

}

}

}

// Método para actualizar un usuario (Update)

public static void actualizarUsuario(int id, String nuevoNombre, String nuevoEmail) throws SQLException {

String updateSQL = "UPDATE usuarios SET nombre = ?, email = ? WHERE id = ?";

try (Connection con = DriverManager.getConnection(DB\_URL, USER, PASSWORD);

PreparedStatement pstmt = con.prepareStatement(updateSQL)) {

pstmt.setString(1, nuevoNombre);

pstmt.setString(2, nuevoEmail);

pstmt.setInt(3, id);

int filasAfectadas = pstmt.executeUpdate();

System.out.println("Usuario actualizado, filas afectadas: " + filasAfectadas);

}

}

// Método para eliminar un usuario (Delete)

public static void eliminarUsuario(int id) throws SQLException {

String deleteSQL = "DELETE FROM usuarios WHERE id = ?";

try (Connection con = DriverManager.getConnection(DB\_URL, USER, PASSWORD);

PreparedStatement pstmt = con.prepareStatement(deleteSQL)) {

pstmt.setInt(1, id);

int filasAfectadas = pstmt.executeUpdate();

System.out.println("Usuario eliminado, filas afectadas: " + filasAfectadas);

}

}

}