### Ejercicio 1: Insertar un Registro en la Tabla

Supongamos que tienes una tabla llamada Estudiantes en tu base de datos con los siguientes campos: id, nombre, y edad.

#### Código:

Este ejercicio inserta un nuevo estudiante en la tabla.

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class InsertarEstudiante {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/tu\_base\_de\_datos";

String usuario = "root";

String contraseña = "tu\_contraseña";

try (Connection con = DriverManager.getConnection(url, usuario, contraseña)) {

String sql = "INSERT INTO Estudiantes (nombre, edad) VALUES (?, ?)";

PreparedStatement pstmt = con.prepareStatement(sql);

// Definir los valores para el nuevo estudiante

pstmt.setString(1, "Carlos");

pstmt.setInt(2, 21);

int filasInsertadas = pstmt.executeUpdate();

System.out.println("Filas insertadas: " + filasInsertadas);

} catch (SQLException e) {

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

}

}

}

### Explicación:

* PreparedStatement: Permite ejecutar una consulta SQL con parámetros. La consulta INSERT INTO Estudiantes (nombre, edad) VALUES (?, ?) tiene dos marcadores de posición (?) que se reemplazarán por valores.
* pstmt.setString(1, "Carlos"): Establece "Carlos" como el primer parámetro (nombre).
* pstmt.setInt(2, 21): Establece 21 como el segundo parámetro (edad).
* executeUpdate(): Ejecuta la instrucción SQL y devuelve el número de filas afectadas (en este caso, una inserción).

### Ejercicio 2: Leer y Mostrar Todos los Registros de la Tabla

Este ejercicio lee todos los registros de la tabla Estudiantes y muestra la información de cada estudiante.

#### Código:

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

import java.sql.SQLException;

public class LeerEstudiantes {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/tu\_base\_de\_datos";

String usuario = "root";

String contraseña = "tu\_contraseña";

try (Connection con = DriverManager.getConnection(url, usuario, contraseña);

Statement stmt = con.createStatement()) {

String sql = "SELECT \* FROM Estudiantes";

ResultSet rs = stmt.executeQuery(sql);

while (rs.next()) {

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

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

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

System.out.println("ID: " + id + ", Nombre: " + nombre + ", Edad: " + edad);

}

} catch (SQLException e) {

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

}

}

}

### Explicación:

* Statement: Utilizado para ejecutar consultas SQL simples sin parámetros.
* executeQuery(): Ejecuta la consulta SELECT \* FROM Estudiantes y devuelve un ResultSet con los datos de la tabla.
* ResultSet rs: Contiene los resultados de la consulta. Se usa el método rs.next() para recorrer cada fila.
* rs.getInt("id")**,** rs.getString("nombre")**,** rs.getInt("edad"): Obtiene los valores de cada columna en la fila actual.

### Ejercicio 3: Actualizar un Registro en la Tabla

Este ejercicio actualiza la edad de un estudiante en la tabla Estudiantes dado un id específico.

#### Código:

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

public class ActualizarEstudiante {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/tu\_base\_de\_datos";

String usuario = "root";

String contraseña = "tu\_contraseña";

try (Connection con = DriverManager.getConnection(url, usuario, contraseña)) {

String sql = "UPDATE Estudiantes SET edad = ? WHERE id = ?";

PreparedStatement pstmt = con.prepareStatement(sql);

// Definir los valores de actualización

pstmt.setInt(1, 22); // Nueva edad

pstmt.setInt(2, 1); // ID del estudiante a actualizar

int filasActualizadas = pstmt.executeUpdate();

System.out.println("Filas actualizadas: " + filasActualizadas);

} catch (SQLException e) {

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

}

}

}

### Explicación:

* PreparedStatement: Define la consulta SQL UPDATE Estudiantes SET edad = ? WHERE id = ? con dos parámetros (?), uno para edad y otro para id.
* pstmt.setInt(1, 22): Establece el nuevo valor de edad (22) en el primer parámetro.
* pstmt.setInt(2, 1): Establece el valor del id (1) en el segundo parámetro para especificar el registro que se va a actualizar.
* executeUpdate(): Ejecuta la actualización y devuelve el número de filas afectadas.