

EJERCICIO GUIADO. JAVA. ACCESO A BASE DE DATOS

Consultas de Acción. Modificaciones.

Las consultas SQL de acción son:

- INSERT INTO (permiten introducir nuevos registros en las tablas)
- DELETE (permiten eliminar registros de las tablas)
- UPDATE (permiten modificar los registros de las tablas)

Estas consultas SQL se ejecutan desde el programa a través del método *executeUpdate* propio del objeto *sentencia*.

Estas instrucciones SQL afectan al contenido de la base de datos y no devuelven ningún resultado. Es decir, no extraen datos, y por tanto no devuelven un *ResultSet*.

En las hojas anteriores se ha visto como ejecutar consultas INSERT INTO y DELETE. En esta hoja guiada veremos como ejecutar consultas de acción de modificación: UPDATE.

EJERCICIO GUIADO Nº 1

PLANTEAMIENTO

Se mejorará el proyecto realizado en la hoja anterior de forma que no solo permita añadir y borrar trabajadores de la tabla *trabajadores* de la base de datos MANEMPSA, sino que también permita realizar modificaciones en los datos de los trabajadores de dicha tabla.

1. Entre en NetBeans. Abre el proyecto llamado *GestionTrabajadores* que se programó en la hoja anterior.
2. Modifique la ventana principal añadiendo un nuevo panel con los siguientes componentes:

Nuevo panel

Ver Trabajadores

Alta Trabajador

DNI:

Nombre:

Apellidos:

Sueldo:

Fecha de Entrada:

Matrícula:

Eliminar Trabajador

DNI:

Modificación Trabajador

DNI:

Nombre:

Apellidos:

Sueldo:

Fecha de Entrada:

Matrícula:

Modificación Trabajador

DNI:

Nombre:

Apellidos:

Sueldo:

Fecha de Entrada:

Matrícula:

El panel se llamará *panelModificar*

El botón es *btnBuscar*

Los cuadros de texto se llaman respectivamente:

- txtModDNI*
- txtModNombre*
- txtModApellidos*
- txtModSueldo*
- txtModDia*
- txtModMes*
- txtModAnio*
- txtModMatricula*

El botón es *btnModificar*

3. La mejora que se pretende hacer es la siguiente:

- El usuario introducirá en el cuadro de texto *txtModDNI* el DNI del trabajador cuyos datos se quieren modificar.
- Al pulsar el botón *Buscar*, el resto de cuadros de texto se rellenarán con los datos del trabajador al que pertenece dicho DNI. (Si dicho DNI no existiera, se mostraría un mensaje de error)
- Cuando aparezcan los datos del trabajador en los cuadros de texto, el usuario podrá realizar las modificaciones pertinentes.
- Al pulsarse el botón *Efectuar Modificación* se modificarán los datos en la tabla.

4. Primero programaremos el botón de búsqueda. Entre en el *actionPerformed* de este botón y programe lo siguiente:

```
private void btnBuscarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO: Agregue su código aquí:  
    String caddni=""; //cadena del DNI  
    String consulta=""; //cadena para consulta  
    String caddia,cadmes,cadanio,cadefe; //cadenas para la fecha  
    try {  
        consulta="select * from trabajadores where DNI='"+txtModDNI.getText()+"'";  
        ResultSet r=sentencia.executeQuery(consulta);  
        if (!r.first()) {  
            JOptionPane.showMessageDialog(null,"No existe un trabajador con ese DNI");  
        } else {  
            txtModNombre.setText(r.getString("nombre"));  
            txtModApellidos.setText(r.getString("apellidos"));  
            txtModSueldo.setText(r.getString("sueldo"));  
            cadefe = r.getString("fecha");  
            caddia=cadefe.substring(8,10);  
            cadmes=cadefe.substring(5,7);  
            cadanio=cadefe.substring(0,4);  
            txtModDia.setText(caddia);  
            txtModMes.setText(cadmes);  
            txtModAño.setText(cadanio);  
            txtModMatricula.setText(r.getString("matricula"));  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null,"Hubo un error al buscar el trabajador");  
    }  
}
```

Estudiemos atentamente el código anterior:

Lo primero que hace el código es construir una consulta de selección que busque a todos los trabajadores que tengan el DNI introducido en el cuadro de texto *txtModDNI*.

Ejecuta la consulta y almacena el resultado en un típico *ResultSet*. Luego comprueba a través del siguiente if si no existe un trabajador con dicho DNI en el *ResultSet*:

```
if (!r.first()) {
```

En ese caso muestra un mensaje de error indicando que no se ha encontrado dicho trabajador.

En caso contrario muestra los datos del trabajador encontrado en los cuadros de texto. Para ello, se extrae del *ResultSet* cada campo y se coloca en el cuadro de texto correspondiente. Por ejemplo:

```
txtModNombre.setText(r.getText("nombre"));
```

Recoge el valor del campo *nombre*, y lo sitúa en el cuadro de texto *txtModNombre*.

En el caso de la fecha de entrada del trabajador, se extrae la fecha y luego se extrae de la cadena el día, el mes y el año, situando cada dato en su cuadro de texto correspondiente.

```
cadefe = r.getString("fecha");
caddia=cadefe.substring(8,10);
cadmes=cadefe.substring(5,7);
cadanio=cadefe.substring(0,4);
txtModDia.setText(caddia);
txtModMes.setText(cadmes);
txtModAnio.setText(cadanio);
```

Recuerda que cuando se extrae una fecha de un *ResultSet*, su formato es:

```
año-mes-día-hora:minutos
```

Por ejemplo: 2001-12-23-00:00

Así pues, si el DNI introducido pertenece a uno de los trabajadores de la tabla, sus datos aparecerán en los cuadros de texto cuando se pulse el botón *Buscar*.

5. Ejecuta el programa y comprueba su funcionamiento. Escribe un DNI en el cuadro del DNI de búsqueda y pulsa *Buscar*.

Modificación Trabajador

DNI:

Nombre:

Apellidos:

Sueldo:

Fecha de Entrada:

Matrícula:

Se introduce un DNI y luego se pulsa *Buscar*.

Si existe un trabajador con ese DNI, sus datos aparecerán en los cuadros de texto.

6. Se supone que cuando aparezcan los datos del trabajador en los cuadros de texto el usuario los modificará según le interese. Luego, al pulsar el botón *Efectuar Modificación* los nuevos datos serán actualizados en la tabla.
7. Para hacer esto, entre en el *actionPerformed* del botón *Efectuar Modificación* y programe lo siguiente:

```
private void btnModificarActionPerformed(java.awt.event.ActionEvent evt) {
// TODO: Agregue su código aquí:
String caddni=""; //cadena del DNI
String consulta=""; //cadena para consultas

try {
    consulta="update trabajadores set ";
    consulta+="nombre='"+txtModNombre.getText()+"',";
    consulta+="apellidos='"+txtModApellidos.getText()+"',";
    consulta+="sueldo='"+txtModSueldo.getText()+"',";
    consulta+="fecha=#"+txtModMes.getText()+"/"+
        txtModDia.getText()+"/"+
        txtModAño.getText()+"#,";
    consulta+="matricula='"+txtModMatricula.getText()+"'";
    consulta+=" where dni='"+txtModDNI.getText()+"'";
    sentencia.executeUpdate(consulta); //se ejecuta la modificación
    MostrarTodos(); //y se muestra la tabla de nuevo
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "Hubo un error al modificar el trabajador");
}

}
```

Se va a analizar este código. Estúdielo con detenimiento:

Debe observar que lo primero que se hace en este código es crear una consulta de actualización (UPDATE) a través de una concatenación de cadenas.

Si observa la concatenación, verá que el resultado es una instrucción SQL de tipo UPDATE en la que los datos que se actualizan son precisamente los datos introducidos en los cuadros de texto del panel de modificación.

Por ejemplo, si estos fueran los valores de los cuadros de texto del panel de modificación:

txtModDNI → 11.111.111-A
txtModNombre → Juan
txtModApellidos → Perez
txtModSueldo → 1100
txtModDia → 4
txtModMes → 12
txtModAnio → 2001
txtModMatricula → 1234-ABC

La instrucción ya construida por concatenación sería la siguiente:

```
update trabajadores  
set nombre='Juan',apellidos='Perez',sueldo=1100,fecha=#12/4/2001#, matricula='1234-ABC'  
where dni='11.111.111-A'
```

Esta instrucción SQL de tipo UPDATE incluye una cláusula WHERE con una condición de forma que solo se modifica aquel trabajador que tenga el DNI introducido en el cuadro de texto *txtModDNI*.

Una vez construida la consulta UPDATE, esta se ejecuta usando el objeto *sentencia* y luego se muestra el contenido de la tabla trabajadores en el panel a través de una llamada al método *MostrarTodos*.

Todo este código está rodeado de un try...catch para evitar errores inesperados.

8. Comprueba el funcionamiento del programa ejecutándolo. Debes introducir un DNI y pulsar el botón *Buscar* para que se rellenen los cuadros de texto con los datos del trabajador.

Luego cambia algún dato de los que han aparecido en los cuadros de texto y pulsa el botón *Efectuar Modificación*.

Si observas el panel de texto deberá aparecer la lista completa de trabajadores y en ella podrás observar la modificación que se ha realizado en el trabajador correspondiente.

22.222.222-y -- Eva Martínez Pérez -- 900,0 -- 10/11/2006 -- 5544-ERT
22.333.444-C -- Francisco López -- 1000,0 -- 01/06/2006 -- 1144-BBB
12.321.567-B -- Juan Pérez -- 1120,0 -- 04/05/2002 -- 4433-ABB
21.123.123-A -- Ana Ruiz -- 1200,0 -- 02/03/2002 -- 3322-ASR
11 -- assss b -- 11231,0 -- 01/01/2001 -- 22

Modificación Trabajador

DNI:

Nombre:

Apellidos:

Sueldo:

Fecha de Entrada:

Matrícula:

Se introduce un DNI y se le da a *Buscar*.

Los cuadros de texto se rellenan con los datos del trabajador.

Se realiza la modificación que se quiera en los cuadros de texto.

Finalmente se pulsa *Efectuar Modificación* para que se realice la modificación.

En el panel aparecerá la lista actualizada de los trabajadores.

CONCLUSIÓN

Se pueden ejecutar instrucciones SQL del tipo UPDATE (modificar registros) usando el objeto sentencia.

Para este tipo de instrucciones hay que usar el método *executeUpdate*.

Normalmente, será necesario construir una cadena de consulta UPDATE a través de la concatenación de subcadenas y datos introducidos por el usuario en cuadros de texto.

Este tipo de instrucciones SQL no devuelven ningún *ResultSet*, ya que no extraen datos de las tablas, sino que modifican el contenido de éstas.

Es una buena idea dar alguna opción de búsqueda que permita encontrar al registro a modificar y que sus datos aparezcan en varios cuadros de texto, de forma que el usuario tenga más facilidad a la hora de realizar los datos.