

EJERCICIO GUIADO. JAVA. ACCESO A BASE DE DATOS

Consultas de Acción. Bajas.

En la hoja guiada anterior se vio como se podían ejecutar consultas SQL de acción del tipo INSERT INTO (para insertar registros en las tablas de la base de datos)

En esta hoja guiada se practicará con otras consultas SQL de acción. Concretamente con las del tipo DELETE (usadas para eliminar registros de las tablas de la base de datos)

Al igual que ocurre con las consultas INSERT INTO, las consultas DELETE haya que ejecutarlas usando el método *executeUpdate* del objeto *sentencia*.

```
sentencia.executeUpdate("delete . . . ");
```

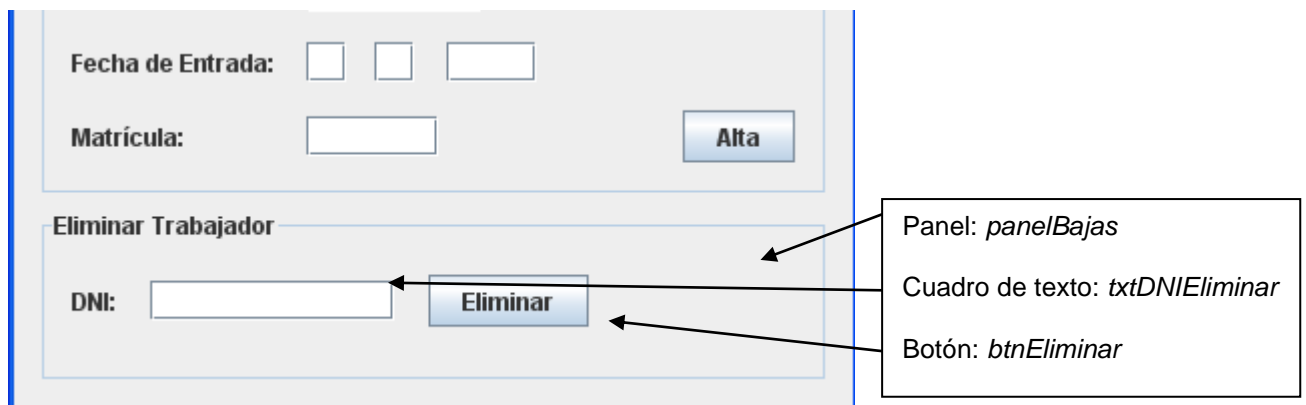
Al ejecutar estas consultas se cambiará el contenido de las tablas de la base de datos y no se devolverá ningún *ResultSet*.

EJERCICIO GUIADO Nº 1

PLANTEAMIENTO

Se mejorará el proyecto realizado en la hoja anterior de forma que no solo admita la inserción de nuevos trabajadores en la tabla trabajadores de la base de datos MANEMPSA, sino que también permite eliminar a trabajadores de la tabla.

1. Entre en NetBeans. Abre el proyecto llamado *GestionTrabajadores* que se programó en la hoja anterior.
2. Añade en la parte inferior de la ventana un nuevo panel:



3. La mejora que se pretende hacer es la siguiente:
 - El usuario introducirá en el cuadro de texto *txtDNIEliminar* el DNI del trabajador que quiere eliminar.
 - Al pulsar el botón *Eliminar* el trabajador con dicho DNI se eliminará de la tabla.
 - Antes de eliminar al trabajador, se le pedirá al usuario una confirmación, para evitar un borrado accidental.
 - Si el DNI introducido no se correspondiera con ningún trabajador de la tabla, entonces se mostrará un mensaje indicándolo.

4. Para hacer esto, entre en el *actionPerformed* del botón *Eliminar* y programe lo siguiente:

```
private void btnEliminarActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO: Agregue su código aquí:  
    String caddni=""; //cadena del DNI  
    String consulta=""; //cadena para consultas  
    int resp; //respuesta a pregunta de confirmación  
    try {  
        //Busco a los trabajadores que tengan dicho DNI  
        consulta="select * from trabajadores where DNI='"+txtDNIEliminar.getText()+"'";  
        ResultSet r=sentencia.executeQuery(consulta);  
        if (!r.first()) { //si no puedo situarme en el primero del resultset, es  
            //que no encontré a ninguno con ese DNI  
            JOptionPane.showMessageDialog(null,"No existe un trabajador con ese DNI");  
        } else { //en caso contrario existe, y lo puedo borrar  
            //pregunto antes  
            resp=JOptionPane.showConfirmDialog(null,"¿Confirma el borrado?",  
                "Borrar",JOptionPane.YES_NO_OPTION);  
            if (resp==JOptionPane.YES_OPTION) { //si se quiere borrar  
                consulta="delete from trabajadores where DNI='"+txtDNIEliminar.getText()+"'";  
                sentencia.executeUpdate(consulta); //se ejecuta la eliminación  
                MostrarTodos(); //y se muestra la tabla de nuevo  
            } else { //si no se quiere borrar  
                JOptionPane.showMessageDialog(null,"Borrado cancelado por el usuario");  
            }  
        }  
    } catch (Exception e) {  
        JOptionPane.showMessageDialog(null,"Hubo un error al eliminar el trabajador");  
    }  
}
```

Se va a analizar este código. Estúdielo con detenimiento:

Lo primero interesante que encontrará es la construcción de una consulta de selección (SELECT) usando la típica concatenación de cadenas:

```
consulta="select * from trabajadores where DNI='"+txtDNIEliminar.getText()+"'";  
ResultSet r=sentencia.executeQuery(consulta);
```

Esto se hace para extraer aquellos trabajadores que tengan el DNI introducido en el cuadro de texto *txtDNIEliminar*. La idea es saber si existe en la tabla algún trabajador con dicho DNI. Al ejecutar esta consulta, el *ResultSet* *r* se llenará con trabajadores que tengan ese DNI.

Observe el código que viene a continuación:

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

Este if significa: "si no puedo colocarme en el primer elemento del *ResultSet* *r*".

Si no es posible colocarse en el primer elemento del resultado de la consulta será porque no hay un primer elemento, o dicho de otra forma, porque el *ResultSet* está vacío, o dicho de otra forma más, porque no se encontró a nadie que tuviera el DNI indicado.

Si esto fuera así, no existe un trabajador con dicho DNI y por tanto no puede ser borrado. Así pues se muestra un mensaje con un *JOptionPane* indicando dicha circunstancia.

En caso contrario, existe ese trabajador y por tanto podemos borrarlo. Observa el contenido del *else*:

```
resp=JOptionPane.showConfirmDialog(null,"¿Confirma el borrado?",
                                   "Borrar",JOptionPane.YES_NO_OPTION);
if (resp==JOptionPane.YES_OPTION) {
```

Este trozo de código le pregunta al usuario si quiere borrar realmente al trabajador. Si la respuesta es sí (*YES_OPTION*) se efectuará el borrado tal como se indica a continuación:

```
consulta="delete from trabajadores where DNI='"+txtDNIEliminar.getText()+"'";
sentencia.executeUpdate(consulta); //se ejecuta la eliminacion
MostrarTodos(); //y se muestra la tabla de nuevo
```

Como puede observar, se construye a través de una concatenación, una consulta de acción SQL del tipo *DELETE*, que nos permitirá borrar el trabajador con el DNI introducido.

Luego se ejecuta dicha instrucción usando el método *executeUpdate* del objeto *sentencia* (recuerde que usará siempre *executeUpdate* para ejecutar consultas de acción: altas, eliminación y modificación)

Finalmente se llama de nuevo al método que creamos en la hoja anterior que muestra todo el contenido de la tabla *trabajadores*, y de esta forma podremos comprobar el borrado del trabajador.

En caso de que el usuario no haya confirmado el borrado, el código muestra un mensaje "Borrado cancelado por el usuario".

Por otro lado, todo este código es susceptible de sufrir errores inesperados, por lo que está rodeado por un *try ... catch*.

5. Ejecute el programa y pruebe a eliminar trabajadores de la tabla. Se recomienda que pruebe a introducir DNIs inexistentes, para ver que ocurre. Pruebe también la confirmación del borrado: responda a veces que sí y otras veces que no.

11 -- a b -- 111,0 -- 01/01/2001 -- sin coche
22.222.222-y -- Eva Martínez -- 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

Ver Trabajadores

Alta Trabajador

DNI:

Nombre:

Apellidos:

Sueldo:

Fecha de Entrada:

Matrícula:

Alta

Eliminar Trabajador

DNI:

22.222.222-y

Eliminar

Introduzca el DNI del trabajador que quiere borrar y luego pulse *Eliminar*.

Al hacerlo el listado se actualizará y ya no mostrará el trabajador con dicho DNI ya que habrá sido borrado.

Si el DNI no existiera el borrado no se produciría y mostraría un mensaje de error.

CONCLUSIÓN

Se pueden ejecutar instrucciones SQL del tipo DELETE (borrar registros) usando el objeto sentencia.

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

Normalmente, será necesario construir una cadena de consulta DELETE 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.

Para realizar borrados será necesario habitualmente el comprobar que los registros a borrar existen en la tabla. Esto se hará ejecutando una consulta SELECT y comprobando si el *ResultSet* resultante contiene algún registro, usando el método *first*.

También se recomienda pedir confirmación al usuario antes de realizar borrados en las tablas.