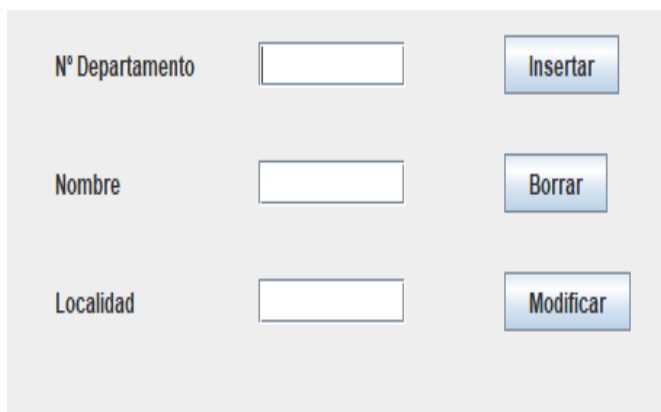


ACCESO A DATOS

UD2: MANEJO DE CONECTORES

ENUNCIADO 4: Insertar, Borrar, Modificar Departamentos.

Crea un formulario que permita insertar, borrar y modificar departamentos.



Formulario para insertar, borrar y modificar departamentos. El formulario está dividido en tres secciones:

- Nº Departamento:** Un campo de texto para ingresar el número de departamento, seguido de un botón "Insertar".
- Nombre:** Un campo de texto para ingresar el nombre del departamento, seguido de un botón "Borrar".
- Localidad:** Un campo de texto para ingresar la localidad del departamento, seguido de un botón "Modificar".

A la hora de insertar podrás hacerlo de 2 maneras:

- 1.- Comprueba previamente que el numero de departamento no existe.
- 2.- Realiza la inserción y si falla recoge el error. (1062)

A la hora de modificar no vamos a comprobar si existe el dato porque si no existe no da error sencillamente indica que se han modificado 0 registros.:

- 1.- Realiza la modificación sin realizar comprobación previa. Aqui no hay fallo ya que si no existiera devolvería 0 elementos modificados.

A la hora de borrar (en este caso borrado restringido) podrás hacerlo de 2 maneras:

- 1.- Comprueba previamente que el numero de departamento existe y que no tiene empleados.
- 2.- Realiza el borrado sin realizar comprobación previa y si falla recoge el error (1451).

ACCESO A DATOS

UD2: MANEJO DE CONECTORES

ENUNCIADO 5: Insertar, borrar y modificar Empleados

Realiza ahora un proceso que permita insertar empleados. Para ello tendrás en cuenta:

- a) Se deben introducir todos los datos.
- b) El salario debe ser positivo
- c) El director (introducirás un número) debe ser un empleado que ya exista en la empresa.
- d) El salario debe ser positivo.
- e) La fecha de alta no la ves en el formulario y será la que indique el sistema.
- e) El departamento lo seleccionarás de un combo previamente cargado con los departamentos.

El diseño de la pantalla es tal y como sigue:



Formulario de gestión de empleados:

- Número:
- Apellido:
- Oficio:
- Director:
- Salario:
- Departamento:

Botones de acción:

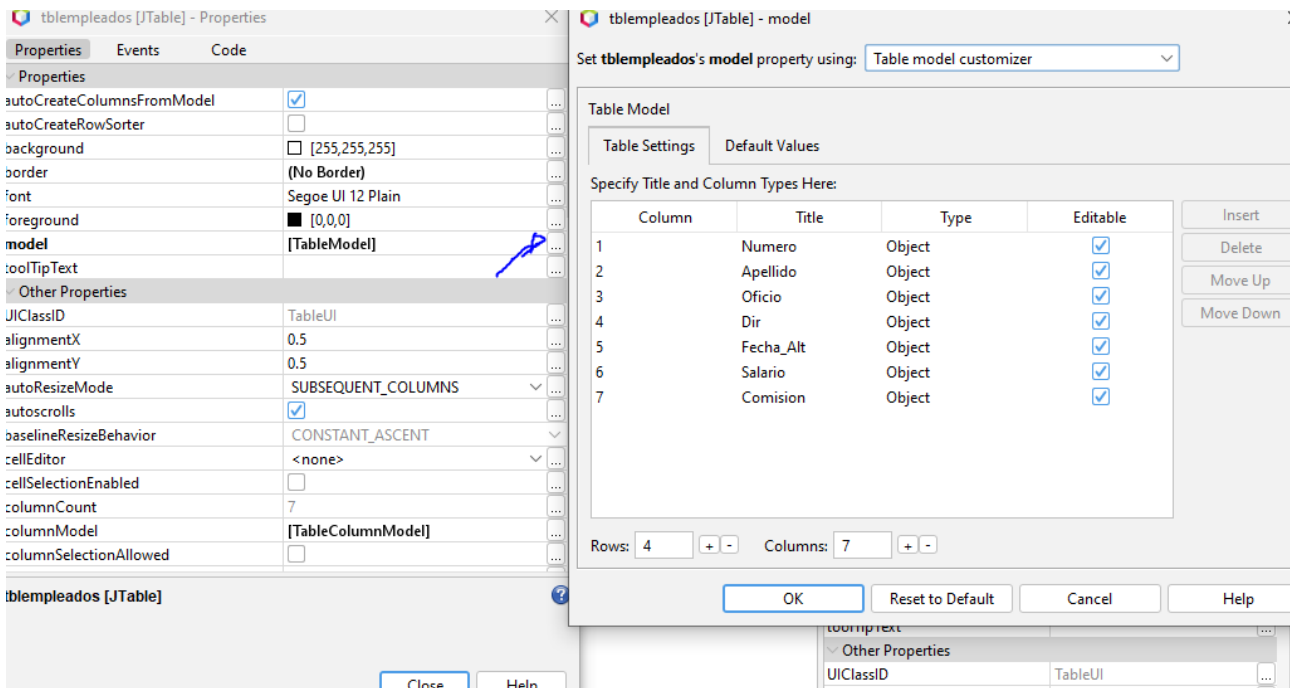
- Insertar
- Borrar
- Modificar

(Nota: en la solución que te propongo dejo los datos cargados en una tabla para ver los datos de los empleados correctamente.)

Para Crear una tabla debes establecer el modelo. Por ejemplo

ACCESO A DATOS

UD2: MANEJO DE CONECTORES



En el código del controlador defines el modelo
 static DefaultTableModel modelotabla = new DefaultTableModel();

y al iniciar dices que esa variable modelotabla es el modelo de la tabla que has diseñado previamente.

```
static DefaultComboBoxModel modelocombo = new DefaultComboBoxModel();
static DefaultTableModel modelotabla = new DefaultTableModel();

public static void iniciar() {
    ventana.setVisible(b: true);
    ventana.setLocationRelativeTo(c: null);
    ventana.getCmbDepartamento().setModel(aModel: modelocombo);
    modelotabla = (DefaultTableModel) ventana.getTblempleados().getModel();
}
```

Para cargar la tabla el proceso es como sigue:

- 1.- Se indica al modelo que no tiene registros. Se pone a 0
- 2.- Se realiza la consulta y si tiene empleados pasamos a recorrerlo.
- 3.- Incrementamos el número de fila en el valor actual (getRowCount()+1)
- 4.- Se carga cada columna con el valor de la consulta.

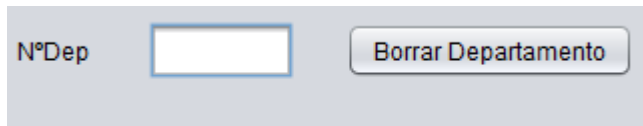
ACCESO A DATOS

UD2: MANEJO DE CONECTORES

```
public void cargartabla(Connection conn, Departamento dep, DefaultTableModel modelotabla) throws SQLException {  
    modelotabla.setRowCount(0);  
    String consulta = "Select * from empleados where dept_no=?";  
    PreparedStatement sentencia = conn.prepareStatement(sql:consulta);  
    sentencia.setInt(parameterIndex: 1, x: dep.getId());  
    ResultSet rs = sentencia.executeQuery();  
    while (rs.next()) {  
        modelotabla.setRowCount(modelotabla.getRowCount() + 1);  
        modelotabla.setValueAt(aValue: rs.getInt(columnIndex:1), modelotabla.getRowCount() - 1, column: 0);  
        modelotabla.setValueAt(aValue: rs.getString(columnIndex:2), modelotabla.getRowCount() - 1, column: 1);  
        modelotabla.setValueAt(aValue: rs.getString(columnIndex:3), modelotabla.getRowCount() - 1, column: 2);  
        modelotabla.setValueAt(aValue: rs.getInt(columnIndex:4), modelotabla.getRowCount() - 1, column: 3);  
        modelotabla.setValueAt(aValue: rs.getDate(columnIndex:5), modelotabla.getRowCount() - 1, column: 4);  
        modelotabla.setValueAt(aValue: rs.getDouble(columnIndex:6), modelotabla.getRowCount() - 1, column: 5);  
        modelotabla.setValueAt(aValue: rs.getDouble(columnIndex:7), modelotabla.getRowCount() - 1, column: 6);  
    }  
}
```

ENUNCIADO 6: Realiza un borrado en cascada de un departamento.

Crea un formulario que permita eliminar departamentos y lo empleados asociados al mismo. El proceso no debes hacerlo en la base de datos (no le cambies las opciones del on delete en la base de datos), debes hacerlo programando.



Introduce para realizar las pruebas un departamento nuevo (por ejemplo el 100) y crea al menos 2 empleados en este departamento. El introducir datos puedes hacerlo directamente sobre la BD.

Realiza un borrado en cascada del departamento 100, de tal manera que al borrarlo también elimine los empleados que tiene.

**Para realizar las pruebas puedes lanzar estos comandos que crear en departamento 100 de nombre borrame y 2 empleados denominados santo y seña **

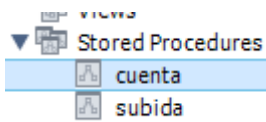
```
INSERT INTO `ejemplo`.`departamentos` (`dept_no`, `dnombre`, `loc`) VALUES ('100', 'Borrame', 'Wirtz');  
INSERT INTO `ejemplo`.`empleados` (`emp_no`, `apellido`, `dept_no`) VALUES ('100', 'Santo', '100');  
INSERT INTO `ejemplo`.`empleados` (`emp_no`, `apellido`, `dept_no`) VALUES ('200', 'Seña', '100');
```

ACCESO A DATOS

UD2: MANEJO DE CONECTORES

ENUNCIADO 7: Ejercicios con procedimientos almacenados.

Ejecuta el procedimiento almacenado cuenta proporcionado en la BD. Despliega Stored Procedures en el workbench



Observa que tiene el primer parámetro de salida y el segundo de entrada.

```
-- Routine DDL
--
DELIMITER $$

CREATE PROCEDURE `ejemplo`.`cuenta` (OUT salida int, IN valor int)
BEGIN
    select count(*) into salida from empleados where dept_no=valor;
END$$
```

El procedimiento lo que hace es recoger en la variable salida todos los empleados de un departamento dado.

Para ejecutar un procedimiento almacenado necesitamos ejecutar un CallableStatement

```
public void cuentaempleados(Connection conn, int numdep, JLabel salida) throws SQLException {
    salida.setText(text: "");
    String consulta = "{call ejemplo.cuenta(?,?)}";
    CallableStatement proc = conn.prepareCall(consulta);

    //Como el primer parámetro es de salida y entero lo indico
    proc.registerOutParameter(parameterIndex: 1, sqlType:Types.INTEGER);
    //El 2 parámetro es de entrada y un entero.
    proc.setInt(parameterIndex: 2, x: numdep);

    proc.executeQuery();

    salida.setText(proc.getInt(parameterIndex: 1) + "");
}
```

Observa que la ejecución del proc depende del orden y tipo de los parámetros. Puedes probar a cambiar el out salida que sea el parámetro 2 y el valor de entrada el parámetro 1. La ejecución cambiaría.