Al añadir esa tabla con una **clave foránea** (FOREIGN KEY) que referencia a la tabla persona, **sí hay varios cambios importantes** en tu proyecto, tanto en **la base de datos como en el código Java**.

**🧱 Cambios que implica esta estructura:**

**🧾 Tabla nueva:**

sql

КопироватьРедактировать

CREATE TABLE departamento (

id\_departamento INT PRIMARY KEY,

nombre VARCHAR(100),

id\_jefe INT,

FOREIGN KEY (id\_jefe) REFERENCES persona(dni)

);

* departamento ahora tiene un jefe (que es una persona).
* Se establece una relación **muchos departamentos → una persona** (como jefe).
* persona.dni debe existir antes de que se cree un departamento.

**🔄 Cambios necesarios en tu modelo Java:**

**1. Clase Departamento debe tener referencia al jefe:**

java

КопироватьРедактировать

public class Departamento {

private int idDepartamento;

private String nombre;

private Persona jefe; // Cambia aquí

public Departamento(int idDepartamento, String nombre, Persona jefe) {

this.idDepartamento = idDepartamento;

this.nombre = nombre;

this.jefe = jefe;

}

public int getIdDepartamento() { return idDepartamento; }

public String getNombre() { return nombre; }

public Persona getJefe() { return jefe; }

public void setJefe(Persona jefe) { this.jefe = jefe; }

@Override

public String toString() {

return nombre + " (Jefe: " + (jefe != null ? jefe.getNombre() : "Ninguno") + ")";

}

}

**2. Debes tener un método getDepartamentos() en SqlBdAccess que recupere los departamentos uniendo con persona:**

java

КопироватьРедактировать

public List<Departamento> getDepartamentos() {

List<Departamento> departamentos = new ArrayList<>();

String sql = "SELECT d.id\_departamento, d.nombre, p.nombre, p.apellido, p.dni, p.edad, p.sexo, p.fechaNacimiento, p.telefono, p.correo, p.direccion " +

"FROM departamento d LEFT JOIN persona p ON d.id\_jefe = p.dni";

try (Connection conn = SqlBdManager.getConnection();

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql)) {

while (rs.next()) {

int id = rs.getInt(1);

String nombreDept = rs.getString(2);

// Si el jefe es null, se crea sin jefe

String dni = rs.getString(5);

Persona jefe = null;

if (dni != null) {

jefe = new Persona(

rs.getString(3), // nombre

rs.getString(4), // apellido

dni,

rs.getInt(6), // edad

Sexo.valueOf(rs.getString(7)),

LocalDate.parse(rs.getString(8)),

rs.getString(9),

rs.getString(10),

rs.getString(11)

);

}

departamentos.add(new Departamento(id, nombreDept, jefe));

}

} catch (SQLException e) {

System.out.println("Error al obtener departamentos: " + e.getMessage());

}

return departamentos;

}

**3. Al insertar un nuevo departamento, ahora debes pasar el dni del jefe:**

java

КопироватьРедактировать

public void insertarDepartamento(Departamento departamento) {

String sql = "INSERT INTO departamento (id\_departamento, nombre, id\_jefe) VALUES (?, ?, ?)";

try (Connection conn = SqlBdManager.getConnection();

PreparedStatement stmt = conn.prepareStatement(sql)) {

stmt.setInt(1, departamento.getIdDepartamento());

stmt.setString(2, departamento.getNombre());

stmt.setString(3, departamento.getJefe().getDni()); // jefe como FK

stmt.execute();

} catch (SQLException e) {

System.out.println("Error al insertar departamento: " + e.getMessage());

}

}

**✅ ¿Qué debes cambiar en tu proyecto concretamente?**

| **Parte** | **Cambios específicos** |
| --- | --- |
| Base de datos | Ya lo hiciste: añadiste departamento con clave foránea a persona(dni). |
| Clase Departamento | Añadir atributo Persona jefe. |
| SqlBdAccess | Métodos nuevos: getDepartamentos(), insertarDepartamento(), editarDepartamento(). |
| Controlador JavaFX | Si tienes ComboBox para jefes, debes cargar personas en él y pasar la seleccionada. |
| Formularios | Al crear/editar Departamento, permitir elegir un jefe (por ejemplo con ComboBox). |