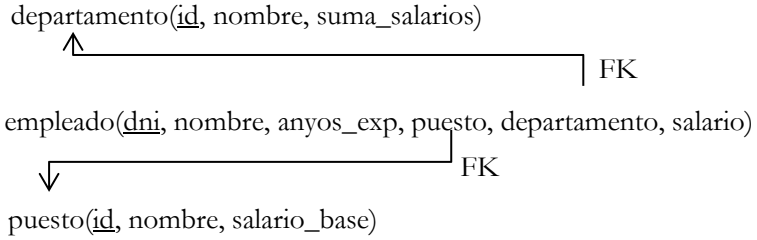
1. Disponemos de una base de datos llamada *grupos_alumnos* donde almacenamos información de los alumnos y los grupos a los que pertenecen. El código para generar la base de datos es el siguiente:

```
CREATE DATABASE grupos_alumnos; use
grupos_alumnos;

CREATE TABLE grupo (
id INT PRIMARY KEY,
nombre VARCHAR(50),
num_alumnos INT
);

CREATE TABLE alumno (
dni CHAR(9),     nombre
VARCHAR(50),     grupo
INT,
    FOREIGN KEY (grupo) REFERENCES grupo (id)
);

INSERT INTO grupo VALUES (1, "Pollitos", 0);
INSERT INTO grupo VALUES (2, "Tortuguitas", 0);
```

La columna *num_alumnos* de la tabla *grupo* almacena el número de alumnos que pertenecen a cada grupo. Se trata de un atributo derivado ya que su valor se puede calcular a partir de otros (a partir de la tabla *alumno*). Cada vez que se inserta, modifica o elimina un alumno en la tabla *alumno* debería actualizarse la columna *num_alumnos* para reflejar dicho cambio.

a. Se pide crear un *trigger* de forma que al insertar un alumno se actualice el número de alumnos en la tabla *grupo*.
b. Se pide crear un *trigger* de forma que al eliminar un alumno se actualice el número de alumnos en la tabla *grupo*.
c. Se pide crear un *trigger* de forma que al modificar el grupo de un alumno se actualice el número de alumnos en la tabla *grupo*.

2. Disponemos de una base de datos llamada *empresa* donde almacenamos información de los empleados, su puesto de trabajo y los departamentos a los que pertenecen. El grafo relacional de la base de datos es el siguiente:



- La columna *suma_salarios* de la tabla *departamento* es un atributo derivado, su valor es la suma de los salarios de todos los empleados del departamento.
- La columna *salario* de la tabla *empleado* también es un atributo derivado y se calcula de la siguiente forma: es el salario base asignado al puesto de trabajo que desempeña el empleado más un extra de 50€ por año de experiencia.

El código necesario para crear la base de datos es el siguiente:

```
CREATE DATABASE empresa;
USE empresa;
CREATE TABLE departamento (     id INT
PRIMARY KEY AUTO_INCREMENT,     nombre
VARCHAR(50),     suma_salarios
DECIMAL(10,2)
);

CREATE TABLE puesto (   id INT PRIMARY
KEY AUTO_INCREMENT,     nombre
VARCHAR(50),     salario_base DECIMAL
(10,2)
);

CREATE TABLE empleado (
dni CHAR(9) PRIMARY KEY,
nombre VARCHAR(50),
anyos_exp INT,     puesto
INT,     departamento INT,
salario DECIMAL (10,2),
    FOREIGN KEY (puesto) REFERENCES puesto (id),
    FOREIGN KEY (departamento) REFERENCES departamento (id)
);

INSERT INTO puesto VALUES (1, 'Administrativo', 1300),
                          (2, 'Técnico', 1200);

INSERT INTO departamento VALUES (1, 'Administración', 0),
                                (2, 'Informática', 0);
```

Por ejemplo, si insertamos un empleado con 5 años de experiencia y su puesto de trabajo es el número 1 (Administrativo) su salario será 1300 + (5 * 50) = 1550€.

a. Crea un *trigger* para que cada vez que se inserte un empleado se calcule automáticamente su salario según la formula indicada anteriormente: salario = salario base de su puesto + (anyos de experiencia * 50). El disparador también debe actualizar la suma total de salarios de la tabla departamento. Para evitar problemas utiliza un *trigger* BEFORE INSERT.
b. Crea un *trigger* para que, al modificar los años de experiencia de un empleado, se recalcule su salario. Deberá actualizarse también la suma total de salarios de la tabla departamento. Para evitar problemas utiliza un *trigger* BEFORE UPDATE.
c. Crea un *trigger* para que al borrar un empleado se actualice la suma total de la tabla departamento.

d. Crea un *trigger* para que al modificar el salario base de un puesto de trabajo, se recalcule el sueldo de todos los empleados con ese puesto. También habrá que recalcular la suma total de la tabla departamento.

3. Implementa una función en la base de datos Q3ERP que reciba como parámetro un precio mínimo y un precio máximo y devuelva el número de productos cuyo precio esté dentro del rango indicado.

4. Crea una *trigger* en la base de datos Q3ERP para que el precio de los productos que se inserten no pueda superar en 200€ al precio del producto más caro que exista en la base de datos. En el caso de que lo superen el precio que se guardará será el del producto más caro más 200€.

5. Crea una función que reciba como parámetro dos marcas, la función devolverá el identificador de aquella marca con el producto más caro. En caso de empate se devolverá cualquiera de los dos identificadores.

6. En la base de datos Q3ERP crea la siguiente tabla:

```
CREATE TABLE clientes (
id INT PRIMARY KEY,
nombre VARCHAR(50)
);
```

Crea un *trigger* en la tabla *clientes* que implemente la función de auto incremento, esto es, que cada vez que se inserte un cliente nuevo, su *id* deberá ser el número siguiente al *id* más alto.