1. Dado el siguiente esquema, indica las sentencias DDL que se indican a continuación:
2. Introduce 5 campos de ejemplo por cada tabla:

/\*DEPARTAMENTOS\*/

INSERT INTO departamentos VALUES (1,"departamento1",2000.0,1000.0);

INSERT INTO departamentos VALUES (2,"departamento2",3000.0,500.0);

INSERT INTO departamentos VALUES (3,"departamento3",1000.0,100.0);

INSERT INTO departamentos VALUES (4,"departamento4",7000.0,5000.0);

INSERT INTO departamentos VALUES (5,"departamento5",10000.0,700.0);

/\*EMPLEADOS\*/

INSERT INTO empleado VALUES (001,"38712645A","nombre1","apellido1","apellido1",1);

INSERT INTO empleado VALUES (002,"38712645C","nombre2","apellido2","apellido2",1);

INSERT INTO empleado VALUES (003,"38712655B","nombre3","apellido3","apellido3",2);

INSERT INTO empleado VALUES (004,"38712645D","nombre4","apellido4","apellido4",4);

INSERT INTO empleado VALUES (005,"38712645E","nombre5","apellido5","apellido5",5);

1. Script de creación de la tabla departamento:

CREATE TABLE IF NOT EXISTS departamentos (

codigo INT(10) PRIMARY KEY,

nombre VARCHAR(100),

presupuesto DOUBLE,

gastos DOUBLE

);

1. Script de creación de la tabla empleado:

CREATE TABLE IF NOT EXISTS empleado (

codigo INT(10) PRIMARY KEY,

nif VARCHAR(9),

nombre VARCHAR(100),

apellido1 VARCHAR(100),

apellido2 VARCHAR(100),

codigo\_departamento INT(10),

FOREIGN KEY (codigo\_departamento) REFERENCES departamentos (codigo) ON UPDATE CASCADE ON DELETE CASCADE

);

1. Modificación de la tabla empleado para que el nif de los empleados no pueda repetirse entre empleados, pero sí se permita que existan empleados que no hayan proporcionado su nif

ALTER TABLE empleado MODIFY COLUMN nif VARCHAR(9) UNIQUE;

1. Modificación de la tabla empleado para que los empleados deban incluir al menos el nombre y primer apellido

ALTER TABLE empleado MODIFY COLUMN nombre VARCHAR(100) NOT NULL;

ALTER TABLE empleado MODIFY COLUMN apellido1 VARCHAR(100) NOT NULL;

1. Si se actualiza un departamento que tenga empleados asignados se deben actualizar los empleados.

ALTER TABLE empleado ADD FOREIGN KEY (codigo\_departamento) REFERENCES departamentos (codigo) ON UPDATE CASCADE ON DELETE CASCADE;

1. Crea el campo “dirección” como un varchar de tamaño 100 en la tabla empleado.

ALTER TABLE empleado ADD COLUMN direccion VARCHAR(100);

1. Sobre las tablas del ejercicio anterior realiza las siguientes acciones:
2. Devuelve un listado con los empleados y los datos de los departamentos donde trabaja cada uno

SELECT empleado.\*,departamentos.\* FROM empleado JOIN departamentos ON empleado.codigo\_departamento = departamentos.codigo;

1. Calcula la media del presupuesto de todos los departamentos.

SELECT AVG(presupuesto) FROM empleado RIGHT JOIN departamentos ON empleado.codigo\_departamento = departamentos.codigo;

1. Devuelve un listado donde sólo aparezcan aquellos empleados que no tienen ningún departamento asociado (mediante un JOIN)

INSERT INTO empleado VALUES (006,"38712645Z","nombre6","apellido6","apellido6",null,"");

SELECT empleado.nombre FROM empleado LEFT JOIN departamentos ON empleado.codigo\_departamento != departamentos.codigo WHERE empleado.codigo\_departamento IS null;

1. Devuelve los nombres de los departamentos que tienen empleados asociados. (Utilizando una subconsulta con IN o NOT IN).

SELECT nombre FROM departamentos WHERE codigo IN (SELECT codigo\_departamento FROM empleado);

1. Devuelve el nombre del departamento con mayor presupuesto y la cantidad que tiene asignada (Empleando una subconsulta).

SELECT nombre,presupuesto FROM departamentos WHERE presupuesto = (SELECT MAX(presupuesto) FROM departamentos);

1. Muestra cuántos empleados tiene cada departamento (incluyendo aquellos que no tengan empleados)

SELECT departamentos.nombre,COUNT(empleado.codigo) FROM departamentos LEFT JOIN empleado ON empleado.codigo\_departamento = departamentos.codigo GROUP BY departamentos.codigo;

1. Añade un 20% de presupuesto a aquellos departamentos que tengan menos de 10000 euros de presupuesto

UPDATE departamentos SET presupuesto=presupuesto+(presupuesto\*0.2) WHERE presupuesto < 10000;