

Alumno:

| | | | | | |
|---|----|-----|----|-----|-------|
| 1 | 2a | 2b | 2c | 2d | TOTAL |
| 3 | 2 | 1.5 | 2 | 1.5 | |

1. Armar el diagrama de entidad-relación. Normalizado en 3era FN. Identificar PK, FK, UNIQUE.

El departamento de Informática de la Universidad de Almería desea diseñar una base de datos para gestionar los profesores que participan en los proyectos de investigación.

- De cada proyecto de investigación se desea almacenar un identificador único, nombre, presupuesto total, el programa de I+D que lo financia, fecha de inicio, fecha de finalización y una descripción.
- En los proyectos de investigación trabajan profesores del departamento durante un periodo de tiempo determinado por una fecha de inicio y una fecha de fin. Tenga en cuenta que un mismo profesor puede trabajar en el mismo proyecto en diferentes épocas.
- De cada profesor se desea almacenar un identificador único, nombre, apellidos, despacho y teléfono.
- Un profesor puede trabajar en varios proyectos a la vez y en un proyecto pueden trabajar varios profesores.
- Los profesores del departamento pueden ser *doctores* o *no doctores*.
- Un profesor *no doctor* debe ser supervisado por un profesor *doctor*. El tiempo de supervisión viene determinado por una fecha de inicio y una fecha de fin. Deberemos almacenar los profesores *doctores* que han supervisado a un profesor *no doctor* y durante qué periodos lo han sido.
- De todos los profesores que trabajan en el proyecto hay uno que es el investigador principal, que será el encargado de coordinar el proyecto. Es necesario almacenar quién es el investigador principal de cada uno de los proyectos. Tenga en cuenta que el investigador principal no puede cambiar a lo largo de la vida del proyecto, siempre será el mismo.
- El investigador principal de un proyecto tiene que ser un profesor *doctor*, en ningún caso podrá serlo un profesor *no doctor*.
- Los profesores *doctores* y *no doctores* escriben publicaciones. Una publicación consta de un código único y un título. Y una publicación puede ser de dos tipos, puede ser una publicación en una revista o en un congreso.
- Si la publicación es en una revista además del código único y el título vamos a almacenar el volumen, el número, la página de inicio y la página de fin.
- Si la publicación es en un congreso además del código único y el título vamos a almacenar el tipo de congreso, ciudad, país, fecha de inicio, fecha de fin y editorial.

2-. En base al siguiente modelo sobre el registro de películas, implementar las siguientes consultas SQL:

PERSONAS(IdPersona, NroDoc, Nombre, Sexo)

Contiene la información de los actores y directores. El tipo y número de documento que los identifica, su nombre y sexo (M o F).

PELICULAS (IdPelicula, Titulo, Genero, Duracion)

Contiene la información de las películas. Su identificador, título, género, duración en minutos y la identificación del director de la misma.

PARTICIPA(IdParticipa, IdPersona, IdPelicula, Rol, Minutos)

Contiene la información de los actores que protagonizaron cada película y por cuantos minutos. También se registran los directores de la misma, teniendo en cuenta que la misma persona puede ser director y actor en la misma película.

2. Resolver en SQL:

- a. La columna "rol" de la tabla PARTICIPA contiene los roles "ACTOR" O "DIRECTOR" que toma cada persona. Se pide crear una tabla Roles y realizar los scripts necesarios para realizar todos los cambios necesarios en la tabla PARTICIPA
- b. Listar la cantidad de veces que una persona fue actor o director en películas del género drama, con el siguiente formato:

| DNI | NOMBRE | ACTOR | DIRECTOR |
|----------|-----------------|-------|----------|
| 22333333 | Jose Campanella | 1 | 4 |
| 22333333 | Ricardo Darin | 5 | 0 |
| 22333333 | Rita Cortese | 2 | 2 |

- c. Mostrar la cantidad de películas en las que participó cada persona sin importar el rol que ocupó. Mostrar únicamente aquellas que superen la cantidad de películas del director "Jose Campanella".
- d. Crear un procedimiento almacenado que reciba por parámetro el nombre de una persona y un rol y devuelva las películas que NO tuvieron a esa persona con ese rol específico

```

create table rol
insert into rol
alter table participa add column add foreign
update participa
alter table participa drop column

```

```

select dni, nombre , count(case when rol=1 then 1 end) as actor, count(case when rol=2 then 1 end) as
director
from personas
inner participa
inner peliculas
where genero='drama'
group by IdPersona

```

```

select dni, nombre , count(distinct idpersona)
from personas
inner participa
inner peliculas
group by IdPersona
having count(*) >select count(*) from ... where nombre ='campanella'

```

```

create proc
select titulo
from peliculas
left join participa
left join persona on pe.id=pa.id and pe.nombre = nombre
left join rol on r.id = pa.idrol and r.nombre = rol
where pa.id is null

```

