

### NORMALIZACIÓN [1,5 PUNTOS]

Normaliza la siguiente relación hasta 3ª FN, para ello, rellena la tabla que se adjunta:

centro	<u>cod_act</u> <u>ividad</u>	nombre_ actividad	<u>dni mo</u> <u>onitor</u>	nombre_ monitor	sala	fecha	hora_ini cio	hora_fin
CuestaFit	01	Pilates	1122	Pilar	11	04/12/20	19:00	19:45
CuestaFit	02	Body pump	1122	Pilar	20	04/12/20	19:00	19:55
CuestaFit	01	Pilates	5544	Miguel	11	04/12/20	19:00	19:45
CuestaFit	03	Yoga	9988	Pablo	11	04/12/20	20:00	20:55

Donde:

- Centro es el nombre del centro donde se realiza la actividad y es único, ya que es una empresa y el nombre está registrado como único.

#### PRIMERA FORMA NORMAL

Indica la relación o relaciones que no cumplen esta FN  
Ya se encuentra en primera forma normal.

¿Por qué no se encuentra en 1ª FN?

Indica las relaciones resultantes del proceso de normalizar a 1ª FN:  
ACTIVIDADES\_DIRIGIDAS (cod actividad,dni monitor, sala, fecha, hora\_inicio, hora\_fin, nombre\_monitor, nombre\_actividad, centro)

#### SEGUNDA FORMA NORMAL

Indica la relación o relaciones que no cumplen esta FN  
ACTIVIDADES\_DIRIGIDAS (cod actividad,dni monitor, sala, fecha, hora\_inicio, hora\_fin, nombre\_monitor, nombre\_actividad, centro)

¿Por qué no se encuentra en 2ª FN? Indica las dependencias, si las hay  
Por que hay atributos que no son claves primarias que no dependen de la clave primaria al completo.

De dni\_monitor dependen= Nombre\_monitor

de cod\_actividad dependen= centro, hora\_inicio, hora\_fin, nombre\_actividad, sala

Indica las relaciones resultantes del proceso de normalizar a 2ª FN:

ACTIVIDADES\_DIRIGIDAS (cod actividad,dni monitor)

MONITOR (dni\_monitor , nombre\_monitor)

ACTIVIDAD (cod actividad, centro, hora\_inicio, hora\_fin, nombre\_actividad, fecha, sala)

### TERCERA FORMA NORMAL

Indica la relación o relaciones que no cumplen esta FN

ACTIVIDAD (cod actividad, centro, hora\_inicio, hora\_fin, nombre\_actividad, fecha, sala)

¿Por qué no se encuentra en 3ª FN? Indica las dependencias, si las hay

Por que hay atributos que no son claves primarias que dependen de otros atributos que no son clave primaria.

Indica las relaciones resultantes del proceso de normalizar a 3ª FN:

ACTIVIDADES\_DIRIGIDAS (cod actividad,dni monitor)

MONITOR (dni\_monitor , nombre\_monitor)

ACTIVIDAD (cod actividad, centro, hora\_inicio, hora\_fin, nombre\_actividad, fecha, sala)

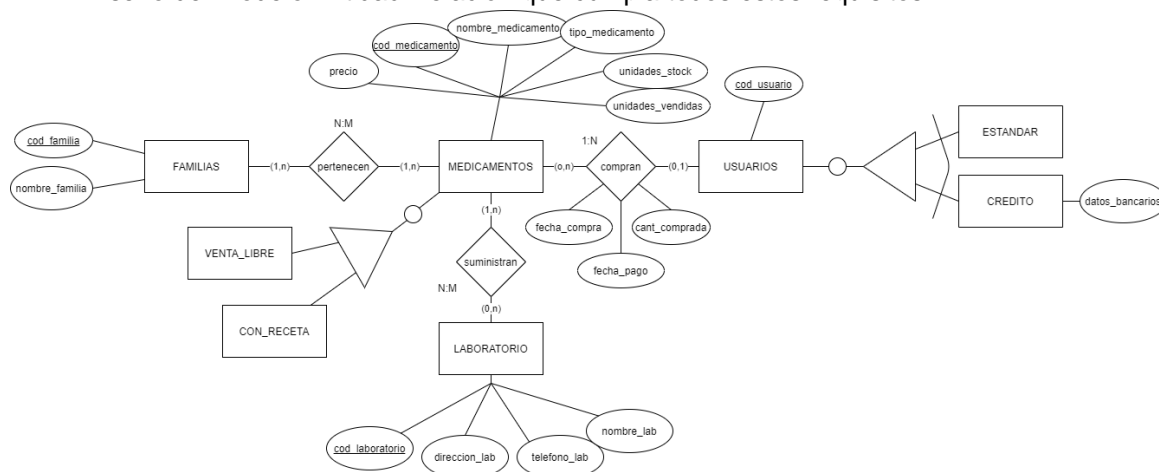
### ESQUEMA RELACIONAL NORMALIZADO

Indica las relaciones resultantes después de haber normalizado hasta 3ª FN:

La gestión de una farmacia requiere poder llevar control de los medicamentos existentes y de los que se van sirviendo, para lo cual se pretende diseñar una aplicación web que cumpla lo siguiente:

- En la **farmacia** se requiere una catalogación de todos los medicamentos existentes, para lo cual se almacenará un **código de medicamento**, **nombre del medicamento**, **tipo de medicamento** (jarabe, comprimido, pomada, etc.), **unidades en stock**, **unidades vendidas** y **precio**. Existen medicamentos de venta libre, y otros que sólo pueden dispensarse con receta médica.
- La **farmacia** adquiere cada medicamento a un **laboratorio**, o bien los fabrica ella misma. Se desea conocer el **código del laboratorio**, **nombre**, **teléfono y dirección**.
- Los **medicamentos** se agrupan en **familias**, un medicamento puede pertenecer a **una única familia o a varias**.
- De las familias de medicamentos guardaremos el **código y el nombre**.
- En la app habrá **dos tipos de usuarios**:
  - **Usuario estándar**: serán aquellos que van a la farmacia y pagan en el momento de llevarse los medicamentos.
  - **Usuario con crédito**: serán aquellos que realizan los pagos de sus pedidos a fin de cada mes.
- La farmacia quiere conocer las unidades de cada medicamento comprado (con o sin crédito) así como la **fecha de compra**. Además, es necesario tener los **datos bancarios de los clientes con crédito, así como la fecha de pago de las compras que realizan**.
- Aspectos a tener en cuenta:
  - Un **mismo medicamento puede ser distribuido por distintos laboratorios**.
  - **En la base de datos están todos los laboratorios de la zona, puede que alguno de ellos no suministre medicamentos a la farmacia**
  - Lo mismo ocurre con las familias de medicamentos, en la base de **datos están todas las familias de medicamentos que existen , pero puede ser que la farmacia no tenga stock de alguna familia**.

1 Diseño del Modelo Entidad Relación que cumpla todos estos requisitos



2 Explicar:

2.a Las cardinalidades de todas las relaciones que han considerado

Un MEDICAMENTO puede pertenecer a una o varias FAMILIA por tanto su cardinalidad es (1,n). Por otro lado una familia puede tener uno o varios medicamentos pertenecientes a ella (1,n). Por lo tanto la cardinalidad máxima será N:M.

Un LABORATORIO puede suministrar uno o varios medicamentos (1,n) después de todo puede que ese laboratorio se especialice en ese tipo de medicamentos y un laboratorio siempre suministra algún medicamento. Por el otro lado un medicamento puede ser suministrado por ningún laboratorio (en caso de que se fabrique en la farmacia) o de que suministre muchos, por lo que la cardinalidad será (0,n). Por tanto su cardinalidad máxima será N:M.

Un usuario puede comprar muchos medicamentos o ninguno (puede estar registrado y no comprar nada) cardinalidad: (0,n). Por otro lado un medicamento puede ser comprado por ningún usuario o solo por uno (0,1). Cardinalidad máxima: 1:N.

2.b El motivo de usar jerarquías o entidades débiles, si es el caso.

En el enunciado nos dicen que los usuarios son de dos tipos, estándar o acreditados, pero no pueden ser los dos tipos a la vez, por lo que he creado una jerarquía total exclusiva para clasificar el tipo de usuarios.

Por otra parte nos dicen que los medicamentos pueden ser de venta libre o con receta, para este caso he utilizado una jerarquía total solapada, pues un medicamento tiene que venderse siempre de alguna de las dos formas, pero hay casos en los que un medicamento puede venderse sin y con receta.

### CONSULTAS [4 PUNTOS]

Para realizar estas consultas, debes:

- Usar la BD que tienes a tu disposición (universidad.sql)
- Mostrar únicamente la información que se pide y en el orden pedido
- Seguir las reglas sintácticas y de formato que hemos visto en clase:
  - Uso de mayúsculas y minúsculas
  - Saltos de línea
- Copiar en cada apartado la consulta correspondiente

A partir de la base de datos "universidad", contesta:

- 1 Devuelve un listado con el nombre, primer apellido y segundo apellido de todos los alumnos. El listado deberá estar ordenado alfabéticamente de la A a la Z por el primer apellido, segundo apellido y nombre. **0,3**

```
SELECT nombre, apellido1, apellido2
FROM persona
WHERE tipo='alumno'
ORDER BY apellido1 ASC,
apellido2 ASC,
nombre ASC;
```

- 2 Muestra el nombre, el cuatrimestre y los créditos de las asignaturas que se imparten en el mismo cuatrimestre que la asignatura con id 10. Ordena el resultado por nombre de la Z a la A **0,5**

```
SELECT nombre, cuatrimestre, creditos
FROM asignatura
WHERE cuatrimestre=
(
SELECT cuatrimestre
FROM asignatura
WHERE id=10
)
ORDER BY nombre DESC;
```

- 3 Devuelve los datos que se enumeran a continuación de los alumnos que nacieron entre 01/01/1990 y 31/12/2000. Muéstralos ordenados por su año de nacimiento (de menor a mayor). Debes usar la función YEAR. **0,3**
- 3.a nombre
  - 3.b año de nacimiento

```
SELECT nombre, YEAR(fecha_nacimiento)
FROM persona
WHERE tipo='alumno'
AND
YEAR(fecha_nacimiento)>=1990
AND
YEAR(fecha_nacimiento)<=2000
ORDER BY YEAR(fecha_nacimiento) ASC;
```

- 4 Calcula la media de alumnos matriculados en cada asignatura. El resultado sólo debe mostrar dos columnas, una con el identificador de la asignatura y otra con el número de alumnos matriculados en ésta. Ponle un alias adecuado a la columna correspondiente al número de alumnos y muestra los datos ordenados de mayor a menos matriculaciones. **0,5**

- 5 Devuelve el nombre, apellidos, teléfono y nif de profesores que no han dado de alta su número de teléfono en la base de datos y además su nif termina en K. **0,3**

```
SELECT nombre, apellido1, apellido2, telefono, nif
FROM persona
WHERE tipo='profesor'
AND
telefono IS NULL
AND
nif LIKE '%k';
```

- 6 Calcula cuántos profesores hay en cada departamento. El resultado sólo debe mostrar dos columnas, una con el identificador del departamento y otra con el número de profesores que hay en ese departamento (esta columna la debes llamar 'num\_profesores'). El resultado deberá estar ordenado de mayor a menor número de profesores. **0,5**

- 7 Devuelve el listado de las asignaturas que se imparten en el primer cuatrimestre, en el tercer curso del grado que tiene el identificador 4. Debes mostrar: **0,3**

- 7.a nombre de la asignatura
- 7.b créditos
- 7.c tipo
- 7.d curso
- 7.e cuatrimestre

Ordenalas alfabéticamente por el nombre de la asignatura

```
SELECT nombre, credits, tipo, curso, cuatrimestre
FROM asignatura
WHERE cuatrimestre=1
AND
curso=3
AND
id_grado=4
ORDER BY nombre asc;
```

- 8 Devuelve todos los datos del alumno más joven. **0,5**

```
SELECT *
FROM persona
WHERE tipo='alumno'
AND fecha_nacimiento=
(
SELECT MAX(fecha_nacimiento)
FROM persona
);
```

- 9 Devuelve un listado con todos los datos de los departamentos que no tienen a ningún profesor asociado **0,8**

```
SELECT *
FROM departamento
WHERE id IN
(
SELECT id_departamento
FROM profesor
);
```



- 10 Muestra el nombre, el tipo y la ciudad de las personas que viven en la misma ciudad que el profesor con id 3 **0,5**

```
SELECT nombre, ciudad, tipo
FROM persona
WHERE ciudad=
(
SELECT ciudad
FROM persona
WHERE id=3
);
```

- 11 Devuelve un listado con el identificador de todos los grados existentes en la base de datos y el número de asignaturas que tiene cada uno. El resultado deberá estar ordenado de menor a mayor por el id del grado. Ponle un alias adecuado a la columna correspondiente al número de asignaturas. **0,5**