

# DAW/DAM. UD REPASO. EJERCICIOS DE REPASO PARA EL EXAMEN

## DAW/DAM. Bases de datos (BD)

### UD REPASO. EJERCICIOS DE REPASO PARA EL EXAMEN

Diagrama E-R, modelo lógico relacional y paso a tablas

Abelardo Martínez y Pau Miñana

Basado y modificado de Sergio Badal ([www.sergiobadal.com](http://www.sergiobadal.com)) y Raquel Torres.

Curso 2023-2024

# 1. Laboratorio

## 1.1. Diagrama E-R

**Dado el siguiente enunciado, diseña el diagrama E-R, con las posibles debilidades y aclaraciones necesarias anexadas en lenguaje natural al pie del diagrama.**

Se desea modelar una BD para un laboratorio donde se realizan análisis de sangre. Cada análisis tiene asociado un único “volante”, compuesto por los datos del paciente (DNI, nombre completo y teléfonos asociados), el médico (los del paciente más el número de colegiado) y la clínica que lo solicita (IDClínica, nombre, dirección completa) de manera que no se aceptan análisis en el sistema si no se dispone de los tres datos. Es decir, un paciente no puede presentarse en el laboratorio y pedir un análisis si no viene de parte de un médico y de una clínica.

Médicos y pacientes se almacenan de manera conjunta como “usuarios” en la BD del laboratorio, existiendo otros tipos de usuario además de esos dos que no necesitamos modelar ahora.

De cada análisis, además del volante, se necesita saber el código del análisis, la fecha y las observaciones que se puedan añadir. Cada análisis tiene la estructura que puedes ver en la página siguiente:

CÓD. ANÁLISIS: 987546; FECHA: 12/12/2021; OBSERVACIONES: Revisión anual  
 PACIENTE: GARCÍA PÉREZ, ANA (23231245A); MÉDICO: SUBIRATS COLOMER, LUIS (2344); CLÍNICA: SOROLLA

### HEMATOLOGÍA

1 HEMATÍES 5,04 (Referencia 4,50 - 6,40)

2 LINFOCITOS 34,8 (Referencia 20 - 40)

[...]

### BIOQUÍMICA

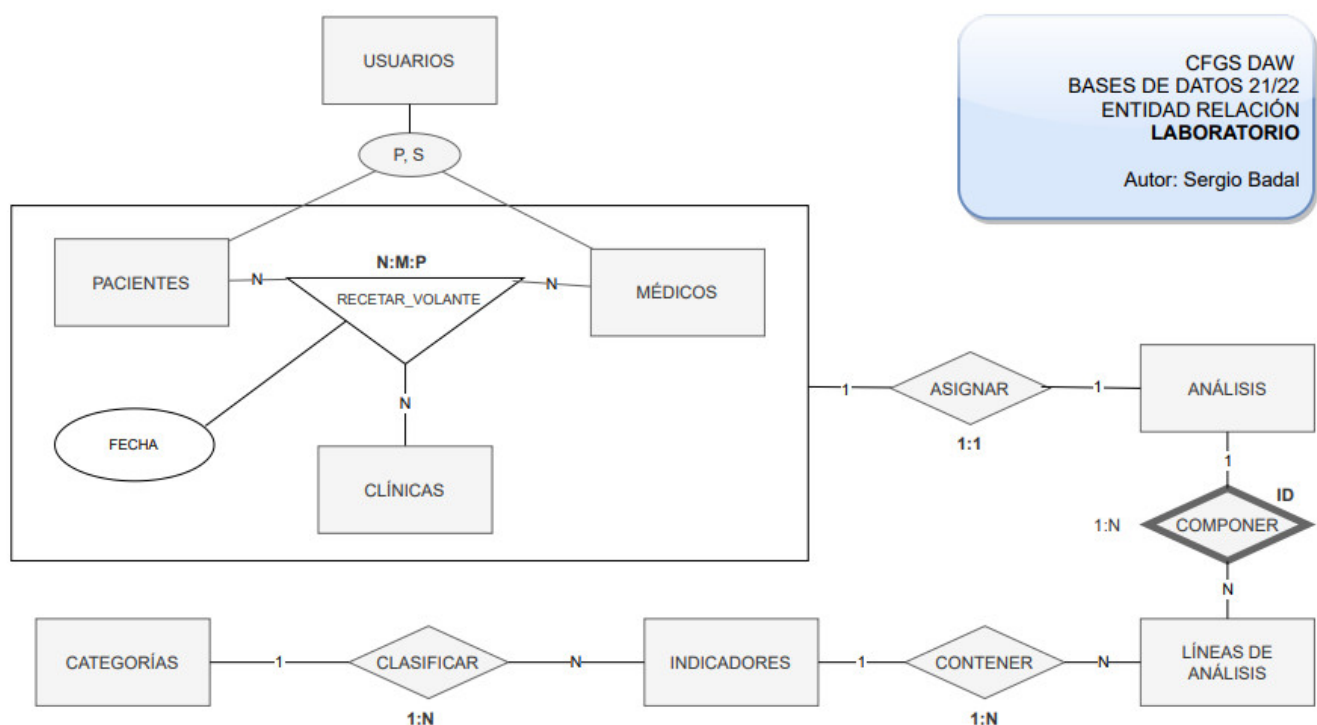
12 CALCIO 9,40 (Referencia 8,50 – 10,50)

13 CREATINA 0,90 (Referencia 0,50 – 1,30)

De los diferentes indicadores (CALCIO, CREATINA, etc.) queremos saber su código, nombre, valor mínimo de referencia y máximo de referencia, así como la categoría en la que se clasifica (HEMATOLOGÍA, BIOQUÍMICA, etc.) de la que queremos saber su código y nombre. A su vez, cada indicador se ordena de manera secuencial en cada análisis en forma de líneas del análisis (1, 2, etc.) de las que solo queremos saber el valor obtenido y el número de orden para poder colocarlo en el listado.

## Ejercicio 1.1. Solución

Diagrama E-R:



Esta es una propuesta de solución donde:

- Se han omitido los atributos para una mayor claridad del diagrama (están en el paso a tablas).
- Asume que las participaciones mínimas son todas cero.

La agregación puede cubrir también a la superentidad **USUARIOS** y el significado de la misma no variaría, ya que lo importante es que una agregación solo englobe a una única relación (en este caso ternaria). También sería válido eliminar la relación **ASIGNAR** y la entidad **ANÁLISIS** y relacionar directamente la agregación con la entidad débil **LÍNEAS DE**

ANÁLISIS, poniendo el código de análisis, las observaciones y la fecha como atributos de la relación ternaria o, más extraño pero válido, como atributos de la propia agregación.

El atributo en la relación ternaria nos permite que un mismo médico prescriba un análisis a un mismo paciente desde una misma clínica pero en diferentes fechas. Es complicado de ver pero es la manera más sencilla de permitir esta situación. Al poder repetirse las tres claves primarias (PK) de las tres entidades que forman la relación, el atributo FECHA tendría que ser el cuarto elemento de la clave primaria de esa ternaria.

Otra alternativa sería que FECHA fuera un atributo de ANÁLISIS y la participación de ANÁLISIS fuera de N. De esta manera, un mismo volante (paciente-médico-clínica) podría generar diferentes análisis con sus fechas y sus observaciones.

Por último, la opción más lógica sería que cada volante tuviera un número de identificación único, lo que permitiría que un mismo médico recete un análisis a un mismo paciente desde una misma clínica pero con diferente número de volante. De este modo, la clave primaria sería simple y estaría formada únicamente por el número de volante.

## 1.2. Paso a modelo relacional normalizado

Pasa a tablas siguiendo el modelo lógico relacional, indicando las pérdidas semánticas y detallando los cambios necesarios para obtener unas tablas normalizadas en 3FN.

No olvides incluir las restricciones o anotaciones recopiladas en el modelo anterior.

### Ejercicio 1.2. Solución

#### Modelo lógico

- Las participaciones mínimas se asumen a CERO (para simplificar la solución).
- No hay pérdidas semánticas y sería conveniente explicar la debilidad.

**Usuarios** (DNI, nombre, apellidos, (teléfono)<sup>n</sup>)

CP: DNI

**Pacientes** (DNI)

CP: DNI

CAj: DNI → Usuarios (DNI)

No necesita más campos, ya que comparte los atributos de la superentidad. El enunciado no lo dice, pero podríamos considerar almacenar su dirección. De este modo tendríamos:

**Pacientes** (DNI, direccióncompleta.dirección\_postal, direccióncompleta.ciudad, direccióncompleta.provincia)

CP: DNI

CAj: DNI → Usuarios (DNI)

**Médicos** (DNI, ncolegiado)

CP: DNI

CAj: DNI → Usuarios (DNI)

UK: ncolegiado → aunque no se dice en el enunciado, el n.º de colegiado es único

**Clínicas** (IDclínica, nombre, direccióncompleta.dirección\_postal, direccióncompleta.ciudad, direccióncompleta.provincia)

CP: IDclínica

**Volante** (DNI\_P, DNI\_M, IDclínica, fecha\_v)

CP: {DNI\_P, DNI\_M, IDclínica, fecha\_v}

CAj: DNI\_P → Pacientes (DNI)

CAj: DNI\_M → Médicos (DNI)

CAj: IDclínica → Clínicas (IDclínica)

El atributo de la relación "fecha\_v" entra en la PK, pero no siempre es así (ver el caso de los BALNEARIOS aquí <http://di002.edv.uniovi.es/~fcano/bdatos/materiales/paso-a-tablas.pdf>)

**Análisis** (cod\_análisis, observaciones, fecha\_a)

CP: cod\_análisis

**Asignar** (DNI\_P, DNI\_M, IDclínica, fecha\_v, cod\_análisis)

CP: {DNI\_P, DNI\_M, IDclínica, fecha\_v}

CAj: {DNI\_P, DNI\_M, IDclínica, fecha\_v} → Volante (DNI\_P, DNI\_M, IDclínica, fecha\_v)

CAj: cod\_análisis → Análisis (cod\_análisis)

UK: cod\_análisis

VNN: cod\_análisis → no tiene sentido asignar un volante sin un análisis

Otra opción válida sería poner como PK el "cod\_análisis" y como UK+VNN los cuatro campos que ahora son PK.

**Categorías** (cod\_cat, nombre)

CP: cod\_cat

**Indicadores** (cod\_indicador, nombre, minref, maxref, cod\_cat)

CP: cod\_indicador

CAj: cod\_cat → Categorías (cod\_cat)

**Líneas análisis** (cod\_análisis, orden, cantidad, cod\_indicador)

CP: {cod\_análisis, orden}

CAj: cod\_análisis → Análisis (cod\_análisis)

CAj: `cod_indicador` → `Indicadores (cod_indicador)`

UK: {`cod_análisis`, `cod_indicador`} → esta restricción se deduce al entender cómo funciona un análisis y no surge del paso a tablas, por lo que incluirla es totalmente opcional, ya que no lo indica el enunciado.



### 1.2.1. Primera Forma Normal (1FN)

#### 1) Relación **Usuarios**

**Usuarios** (DNI, nombre, apellidos, (teléfono)<sup>n</sup>)

CP: DNI

El atributo teléfono es multivaluado. La relación no está en 1FN, por lo que procedemos a transformarla:

**Usuarios** (DNI, nombre, apellidos)

CP: DNI

**Teléfonos** (DNI, teléfono)

CP: {DNI, teléfono}

CAj: DNI → Usuarios (DNI)

#### 2) Relación **Pacientes**

**Pacientes** (DNI)

CP: DNI

CAj: DNI → Usuarios (DNI)

No necesita más campos, ya que comparte los atributos de la superentidad. El enunciado no lo dice, pero podríamos considerar almacenar su dirección. De este modo tendríamos:

**Pacientes** (DNI, direccióncompleta.dirección\_postal, direccióncompleta.ciudad, direccióncompleta.provincia)

CP: DNI

CAj: DNI → Usuarios (DNI)

La "direccióncompleta" es un atributo compuesto. En este caso, la relación no está ni en 1FN y tenemos que separar los atributos en varios campos.

**Pacientes** (DNI, dirección\_postal, ciudad, provincia)

CP: DNI

CAj: DNI → Usuarios (DNI)

### 3) Relación Clínicas

**Clínicas** (IDclínica, nombre, direccióncompleta.dirección\_postal, direccióncompleta.ciudad, direccióncompleta.provincia)

CP: IDclínica

La "direccióncompleta" es un atributo compuesto. En este caso, la relación no está ni en 1FN y tenemos que separar los atributos en varios campos.

**Clínicas** (IDclínica, nombre, dirección\_postal, ciudad, provincia)

CP: IDclínica

#### 1.2.2. Segunda Forma Normal (2FN)

Las tablas que tienen una clave primaria simple ya están en 2FN. El resto también se encuentran en 2FN, puesto que no hay dependencias parciales.

### 1.2.3. Tercera Forma Normal (3FN)

#### 1) Relación **Pacientes**

No está en 3FN, puesto que existe una dependencia transitiva provincia → ciudad. Debemos crear una nueva tabla para tratar eliminar esta dependencia transitiva.

**Provincia** (provincia)

CP: provincia

**Ciudad** (ciudad, provincia)

CP: {ciudad, provincia}

CAj: provincia → Provincia (provincia)

**Pacientes** (DNI, dirección\_postal, ciudad)

CP: DNI

CAj: DNI → Usuarios(DNI)

CAj: ciudad → Ciudad (ciudad)

#### 2) Relación **Clínicas**

No está en 3FN, puesto que existe una dependencia transitiva provincia → ciudad. Debemos crear una nueva tabla para tratar eliminar esta dependencia transitiva.

Las tablas Provincia y Ciudad son las mismas que hemos creado anteriormente.

**Clínicas** (IDclínica, nombre, dirección\_postal, ciudad)

CP: IDclínica

CAj: ciudad → Ciudad (ciudad)

### 1.2.4. Resultado final

Y el resultado final:

**Usuarios** (DNI, nombre, apellidos)

CP: DNI

**Teléfonos** (DNI, teléfono)

CP: {DNI, teléfono}

CAj: DNI → Usuarios (DNI)

**Provincia** (provincia)

CP: provincia

**Ciudad** (ciudad, provincia)

CP: {ciudad, provincia}

CAj: provincia → Provincia (provincia)

**Pacientes** (DNI, dirección\_postal, ciudad)

CP: DNI

CAj: DNI → Usuarios(DNI)

CAj: ciudad → Ciudad (ciudad)

**Médicos** (DNI, ncolegiado)

CP: DNI

CAj: DNI → Usuarios (DNI)

UK: ncolegiado → aunque no se dice en el enunciado, el n.º de colegiado es único

**Clínicas** (IDclínica, nombre, dirección\_postal, ciudad)

CP: IDclínica

CAj: ciudad → Ciudad (ciudad)

**Volante** (DNI\_P, DNI\_M, IDclínica, fecha\_v)

CP: {DNI\_P, DNI\_M, IDclínica, fecha\_v}

CAj: DNI\_P → Pacientes (DNI)

CAj: DNI\_M → Médicos (DNI)

CAj: IDclínica → Clínicas (IDclínica)

El atributo de la relación "fecha\_v" entra en la PK, pero no siempre es así (ver el caso de los BALNEARIOS aquí <http://di002.edv.uniovi.es/~fcano/bdatos/materiales/paso-a-tablas.pdf>)

**Análisis** (cod\_análisis, observaciones, fecha\_a)

CP: cod\_análisis

**Asignar** (DNI\_P, DNI\_M, IDclínica, fecha\_v, cod\_análisis)

CP: {DNI\_P, DNI\_M, IDclínica, fecha\_v}

CAj: {DNI\_P, DNI\_M, IDclínica, fecha\_v} → Volante (DNI\_P, DNI\_M, IDclínica, fecha\_v)

CAj: cod\_análisis → Análisis (cod\_análisis)

UK: cod\_análisis

VNN: cod\_análisis → no tiene sentido asignar un volante sin un análisis

Otra opción válida sería poner como PK el "cod\_análisis" y como UK+VNN los cuatro campos que ahora son PK.

**Categorías** (cod\_cat, nombre)

CP: cod\_cat

**Indicadores** (cod\_indicador, nombre, minref, maxref, cod\_cat)

CP: cod\_indicador

CAj: cod\_cat → Categorías (cod\_cat)

**Líneas análisis** (cod\_análisis, orden, cantidad, cod\_indicador)

CP: {cod\_análisis, orden}

CAj: cod\_análisis → Análisis (cod\_análisis)

CAj: cod\_indicador → Indicadores (cod\_indicador)

UK: {cod\_análisis, cod\_indicador} → esta restricción se deduce al entender cómo funciona un análisis y no surge del paso a tablas, por lo que incluirla es totalmente opcional, ya que no lo indica el enunciado.

### 1.3. Paso a tablas

Indica las sentencias DDL necesarias, en el orden adecuado, para crear una BD en MySQL que pueda gestionar el modelo propuesto en los pasos anteriores.

No olvides incluir las restricciones o anotaciones recopiladas en los modelos anteriores.

### 1.4. Modificación de los metadatos (DDL)

Indica las sentencias DDL necesarias, en el orden adecuado, para:

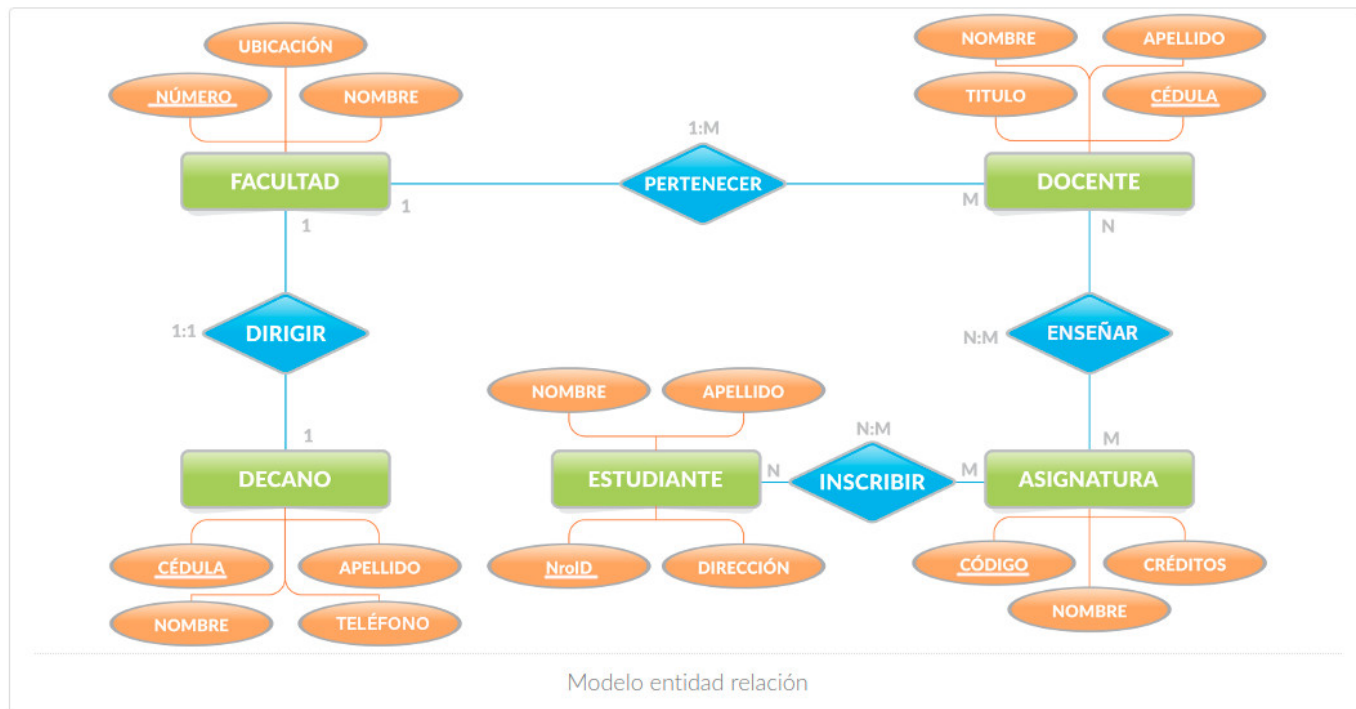
- Añadir que el número de colegiado sea único para todos los médicos almacenados
- Añadir la posibilidad de que un análisis sea asociado, opcionalmente, con una de las aseguradoras de salud (ASISA, SANITAS, ADESLAS, etc.).

### 1.5. Inserción de datos (DML)

Indica las sentencias DML necesarias, en el orden adecuado, para insertar el análisis que te mostramos junto al enunciado.

## 2. Universidad

Tenemos el siguiente diagrama E-R de una Universidad:



### 2.1. Paso a modelo relacional normalizado

Pasa a tablas el diagrama indicado, siguiendo el modelo lógico relacional, indicando las pérdidas semánticas y detallando los cambios necesarios para obtener unas tablas normalizadas en 3FN.

### 2.2. Inserción de datos (DML)

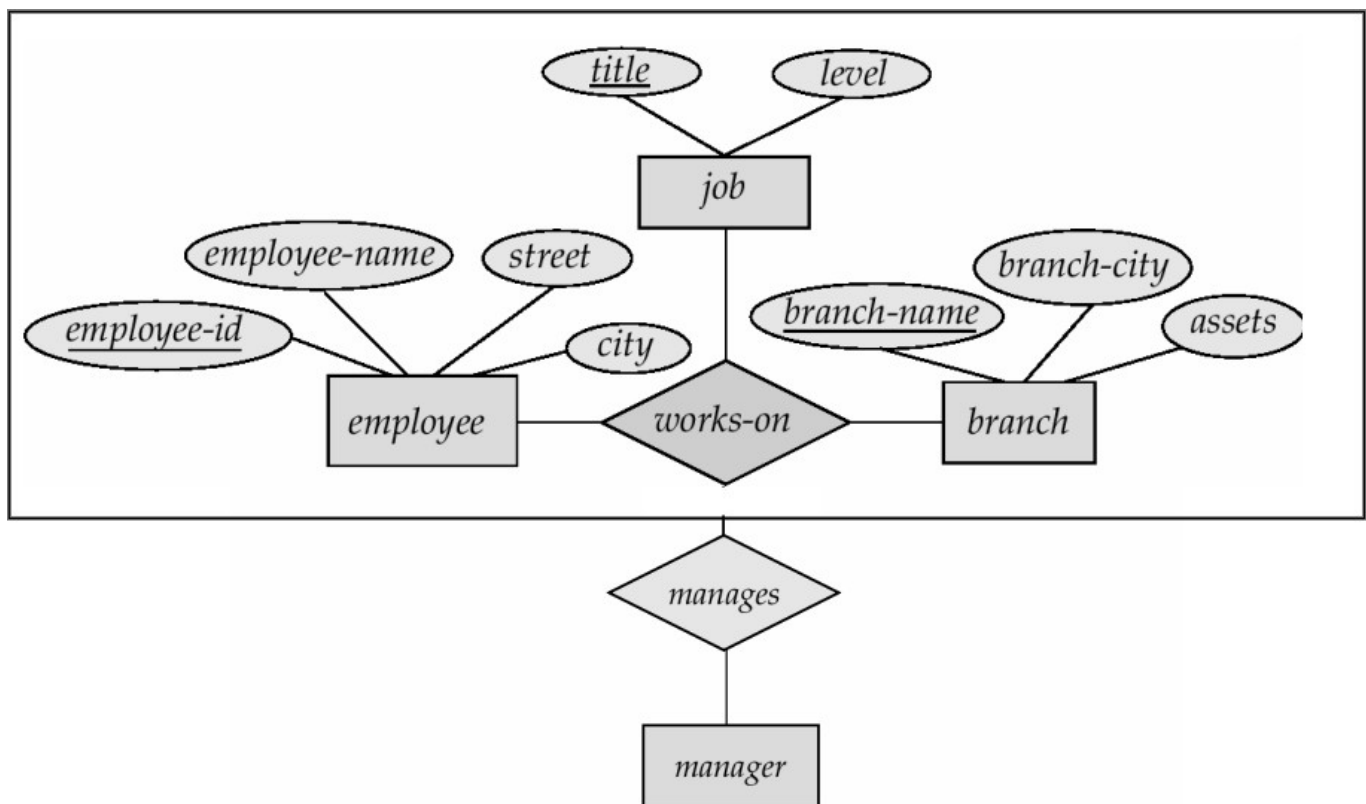
Indica las sentencias DML necesarias, en el orden adecuado, para insertar tres estudiantes, inscritos en tres asignaturas distintas, asociadas cada una a un docente distinto, de una facultad distinta con un decano distinto cada facultad.

### 3. Red de oficinas

Normaliza a 3FN las relaciones anteriores.

#### 3.1. Diagrama E-R

Tenemos el siguiente diagrama E-R de una red de oficinas:



Completa este diagrama E-R para añadir las cardinalidades y participaciones que consideres más apropiadas. También está en tu mano decidir los atributos de la entidad "manager" y una nueva relación binaria con una nueva entidad que actúe como entidad débil y que dependa de manager (a tu elección).

Recuerda justificar la debilidad que decidas implementar.



## 3.2. Paso a modelo relacional normalizado

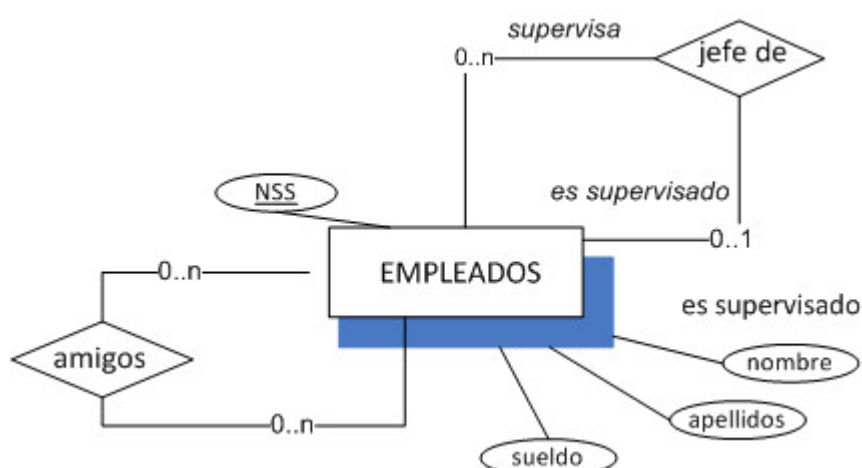
Pasa a tablas el diagrama indicado, siguiendo el modelo lógico relacional, indicando las pérdidas semánticas y detallando los cambios necesarios para obtener unas tablas normalizadas en 3FN.

**No olvides incluir las restricciones o anotaciones recopiladas en el modelo anterior.**

## 4. Empleados

### 4.1. Paso a modelo relacional normalizado

Tenemos el siguiente diagrama E-R de empleados:



Pasa a tablas el diagrama indicado, siguiendo el modelo lógico relacional, indicando las pérdidas semánticas y detallando los cambios necesarios para obtener unas tablas normalizadas en 3FN.

### Ejercicio 4.1. Solución

#### Modelo lógico:

**Empleados** (NSS, sueldo, nombre, apellido, NSSSupervisor)

CP: NSS

CAj: NSSSupervisor → Empleados (NSS)

**Amigos** (NSSAmigo\_A, NSSAmigo\_B)

CP: {NSSAmigo\_A, NSSAmigo\_B}

CAj: NSSAmigo\_A → Empleados (NSS)

CAj: NSSAmigo\_B → Empleados (NSS)

## 4.2. Inserción de datos (DML)

Indica las sentencias DML necesarias, en el orden adecuado, para insertar estos empleados:

- 123234 MIGUEL SÁNCHEZ LÓPEZ 40.000€
- 566777 ANA OBENO GÓMEZ 44.000€
- 935665 LUCÍA GRACIA LORELL 70.000€

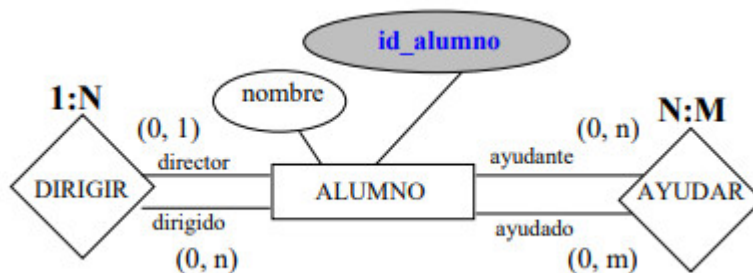
Con estas relaciones:

- Ana es amiga de Lucía
- Lucía es jefa de Ana y de Miguel
- Lucía no tiene jefes (es la gerente)

## 5. Alumnos

### 5.1. Paso a modelo relacional normalizado

Tenemos el siguiente diagrama E-R de alumnos:



Pasa a tablas el diagrama indicado, siguiendo el modelo lógico relacional, indicando las pérdidas semánticas y detallando los cambios necesarios para obtener unas tablas normalizadas en 3FN.

### Ejercicio 5.1. Solución

#### Modelo lógico:

**Alumno** (id\_alumno, nombre, id\_director)

CP: id\_alumno

CAj: id\_director → Alumno (id\_alumno)

**Ayudante** (id\_ayudanteA, id\_ayudanteB)

CP: {id\_ayudanteA, id\_ayudanteB}

CAj: id\_ayudanteA → Alumno (id\_alumno)

CAj: id\_ayudanteB → Alumno (id\_alumno)

## 5.2. Inserción de datos (DML)

Indica las sentencias DML necesarias, en el orden adecuado, para insertar estos alumnos:

- 1111 LUKE SKYWALKER
- 2222 KYLO REN
- 3333 REY SKYWALKER
- 4444 BABY YODA

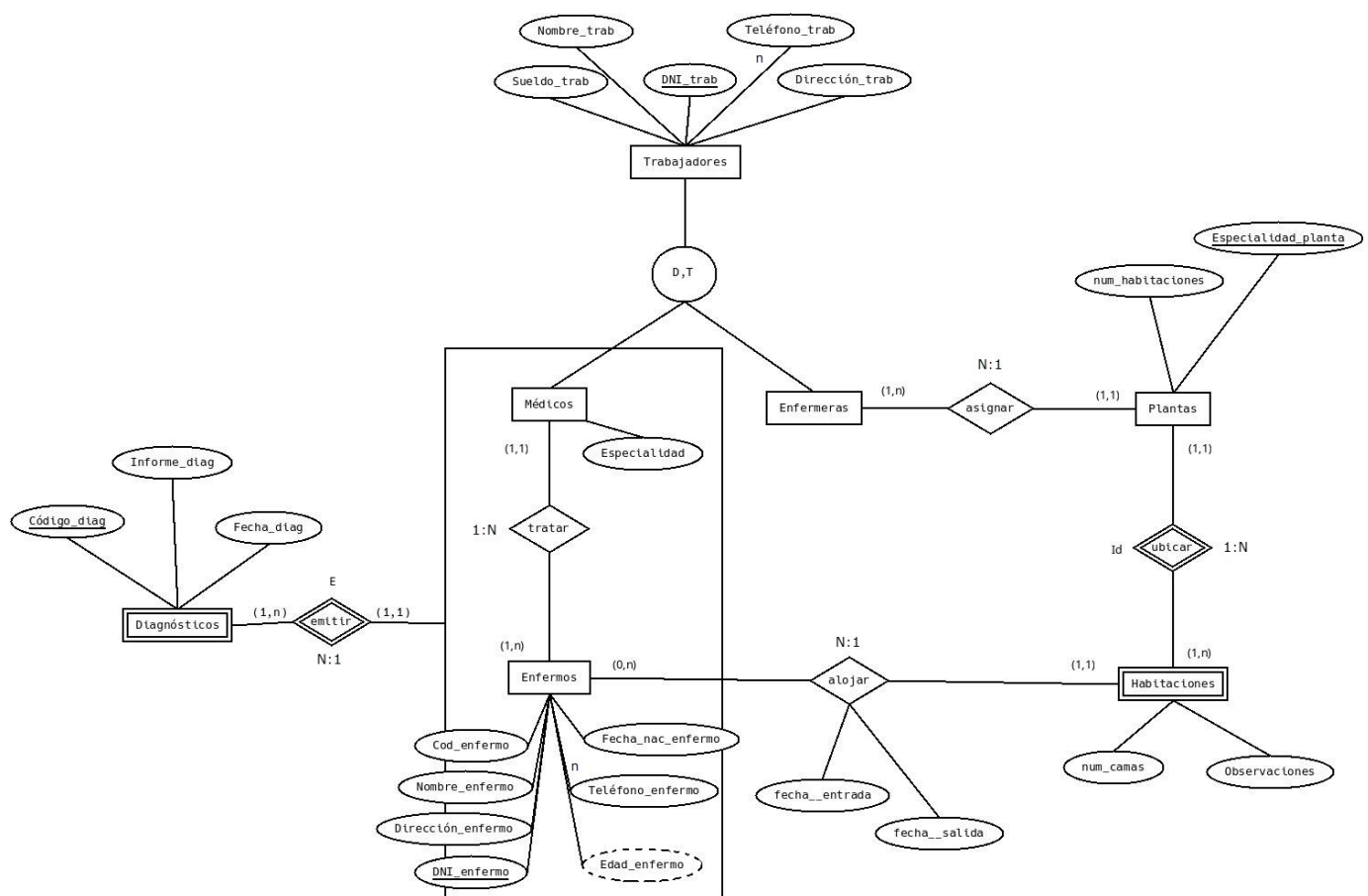
Con estas relaciones:

- Rey ayuda a Kylo Ren
- Luke dirige a Rey y Kylo Ren
- Baby Yoda no es dirigido por nadie pero es ayudado por Luke

## 6. Hospital

### 6.1. Diagrama E-R

Tenemos el siguiente diagrama E-R de un hospital:



#### Observaciones:

Los atributos nombre y dirección son compuestos tanto en la entidad Trabajadores como en la entidad Enfermos.

## 6.2. Paso a modelo relacional

Transforma a Modelo Relacional el diagrama E-R anterior.

### Ejercicio 6.2. Solución

**Trabajadores** (dni\_trab, nombre.nombre, nombre.apellidos, dirección.postal, dirección.ciudad, dirección.provincia, dirección.país, {teléfono}<sup>n</sup>, sueldo)

CP: dni\_trab

**Médicos** (dni\_trab, especialidad)

CP: dni\_trab

CAj: dni\_trab → Trabajadores (dni\_trab)

**Plantas** (especialidad\_planta, num\_habitaciones)

CP: especialidad\_planta

**Habitaciones** (especialidad\_planta, número, num\_camas, observaciones)

CP: {especialidad\_planta, número}

CAj: especialidad\_planta → Plantas (especialidad\_planta)

**Enfermeras** (dni\_trab, especialidad\_planta)

CP: dni\_trab

CAj: dni\_trab → Trabajadores (dni\_trab)

CAj: especialidad\_planta → Plantas (especialidad\_planta)

VNN: especialidad\_planta

**Enfermos** (dni\_enf, cod\_enf, nombre.nombre, nombre.apellidos, fecha\_nac, dirección.postal, dirección.ciudad, dirección.provincia, dirección.país, {teléfono}<sup>n</sup>, dni\_trab, especialidad\_planta, número, fecha\_entrada, fecha\_salida)

CP: dni\_enf

ÚNICO: cod\_enf

CAj: dni\_trab → Médicos (dni\_trab)

CAj: {especialidad\_planta, número} → Habitaciones (especialidad\_planta, número)

VNN: dni\_trab

VNN: {especialidad\_planta, número}

edad\_enf = fecha\_actual.Año – fecha\_nac.Año

**Diagnósticos** (cod\_diag, informe, fecha, dni\_trab, dni\_enf)

CP: cod\_diag

CAj: dni\_trab → Médicos (dni\_trab)

CAj: dni\_enf → Enfermos (dni\_enf)

VNN: dni\_trab

VNN: dni\_enf

### **Pérdidas semánticas:**

- (1) Todo trabajador del hospital es médico o enfermera, pero no ambos a la vez.
- (2) Toda planta debe tener asignada al menos una enfermera.
- (3) Toda planta debe tener al menos una habitación.
- (4) Todo médico debe tratar al menos a un enfermo.
- (5) Todo médico que trata a un enfermo debe emitir un diagnóstico.



## 6.2.1. Primera Forma Normal (1FN)

### 1) Relación **Trabajadores**

- Los atributos compuestos “nombre” y “dirección” se dividen en tantos campos como tenga la composición.

**Trabajadores** (dni\_trab, nombre, apellidos, dirección\_postal, ciudad, provincia, país, {teléfono}<sup>n</sup>, sueldo)

CP: dni\_trab

- El atributo multivaluado “teléfono” se sustituye por una relación en ER y se convierte a tablas.

**Trabajadores** (dni\_trab, nombre, apellidos, dirección\_postal, ciudad, provincia, país, sueldo)

CP: dni\_trab

**TrabTelf** (teléfono, dni\_trab)

CP: {teléfono, dni\_trab}

CAj: dni\_trab → Trabajadores (dni\_trab)

### 2) Relación **Enfermos**

- Los atributos compuestos “nombre” y “dirección” se dividen en tantos campos como tenga la composición.

**Enfermos** (dni\_enf, cod\_enf, nombre, apellidos, fecha\_nac, dirección\_postal, ciudad, provincia, país, {teléfono}<sup>n</sup>, dni\_trab, especialidad\_planta, número, fecha\_entrada, fecha\_salida)

CP: dni\_enf

ÚNICO: cod\_enf

CAj: dni\_trab → Médicos (dni\_trab)

CAj: {especialidad\_planta, número} → Habitaciones (especialidad\_planta, número)

VNN: dni\_trab

VNN: {especialidad\_planta, número}

$edad\_enf = fecha\_actual.Año - fecha\_nac.Año$

- El atributo multivaluado “teléfono” se sustituye por una relación en ER y se convierte a tablas.

**Enfermos** (dni\_enf, cod\_enf, nombre, apellidos, fecha\_nac, dirección\_postal, ciudad, provincia, país, dni\_trab, especialidad\_planta, número, fecha\_entrada, fecha\_salida)

CP: dni\_enf

ÚNICO: cod\_enf

CAj: dni\_trab → Médicos (dni\_trab)

CAj: {especialidad\_planta, número} → Habitaciones (especialidad\_planta, número)

VNN: dni\_trab

VNN: {especialidad\_planta, número}

$edad\_enf = fecha\_actual.Año - fecha\_nac.Año$

- El atributo derivado “edad” se elimina y se pone en observaciones.

**Enfermos** (dni\_enf, cod\_enf, nombre, apellidos, fecha\_nac, dirección\_postal, ciudad, provincia, país, dni\_trab, especialidad\_planta, número, fecha\_entrada, fecha\_salida)

CP: dni\_enf

ÚNICO: cod\_enf

CAj: dni\_trab → Médicos (dni\_trab)

CAj: {especialidad\_planta, número} → Habitaciones (especialidad\_planta, número)

VNN: dni\_trab

VNN: {especialidad\_planta, número}

### Observaciones:

(1)  $edad\_enf = fecha\_actual.Año - fecha\_nac.Año$

### 6.2.2. Segunda Forma Normal (2FN)

Hay que eliminar las DF parciales.

- Las relaciones Trabajadores, Médicos, Plantas, Enfermeras, Enfermos y Diagnósticos no tienen claves principales compuestas, por lo que ya están en 2FN.
- Las relaciones TrabTelf y EnfermosTelf no tienen atributos que no sean clave primaria, por lo que ya están en 2FN.
- **Para la relación Habitaciones obtenemos el diagrama de DF para ver si hay dependencias parciales:** no hay dependencias parciales, ya que los campos “num\_camas” y “observaciones” dependen completamente de la clave primaria, por lo que está en 2FN.

### 6.2.3. Tercera Forma Normal (3FN)

Hay que eliminar las DF transitivas. Para ello es necesario calcular el diagrama funcional de cada relación:

#### 1) Relación **Trabajadores**

nombre, apellidos, dirección\_postal, ciudad, sueldo → dni\_trab

provincia → ciudad

país → provincia

**Provincias** (provincia, país)

CP: provincia

VNN: país

**Ciudades** (ciudad, provincia)

CP: ciudad

CAj: provincia → Provincias (provincia)

VNN: provincia

**Trabajadores** (dni\_trab, nombre, apellidos, dirección\_postal, ciudad, sueldo)

CP: dni\_trab

CAj: ciudad → Ciudades (ciudad)

#### 2) Relación **Enfermos**

cod\_enf, nombre, apellidos, dirección\_postal, ciudad, dni\_trab, especialidad\_planta,

número, fecha\_entrada, fecha\_salida → dni\_enf

provincia → ciudad

país → provincia

Procedemos de la misma forma que en Trabajadores, por lo que ya tenemos parte de las tablas.

**Enfermos** (dni\_enf, cod\_enf, nombre, apellidos, fecha\_nac, dirección\_postal, ciudad, dni\_trab, especialidad\_planta, número, fecha\_entrada, fecha\_salida)

CP: dni\_enf

ÚNICO: cod\_enf

CAj: ciudad → Ciudades (ciudad)

CAj: dni\_trab → Médicos (dni\_trab)

CAj: {especialidad\_planta, número} → Habitaciones (especialidad\_planta, número)

VNN: dni\_trab

VNN: {especialidad\_planta, número}

Y las de Provincias y Ciudades son las mismas.

### 3) Relación **TrabTelf**

**TrabTelf** (teléfono, dni\_trab)

Como no hay atributos no-clave, no puede haber dependencias transitivas, luego la relación ya está en 3FN.

### 4) Relación **EnfermosTelf**

**EnfermosTelf** (teléfono, dni\_enf)

Como no hay atributos no-clave, no puede haber dependencias transitivas, luego la relación ya está en 3FN.

### 5) Relación **Médicos**

**Médicos** (dni\_trab, especialidad)

Como solo hay un atributo no-clave, no puede haber dependencias transitivas, luego la relación ya está en 3FN.

### 6) Relación **Plantas**

**Plantas** (especialidad\_planta, num\_habitaciones)

Como solo hay un atributo no-clave, no puede haber dependencias transitivas, luego la relación ya está en 3FN.

### 7) Relación **Habitaciones**

**Habitaciones** (especialidad\_planta, número, num\_camás, observaciones)

Todos los atributos dependen directamente de la CP, luego la relación ya está en 3FN.

### 8) Relación **Enfermeras**

**Enfermeras** (dni\_trab, especialidad\_planta)

Como solo hay un atributo no-clave, no puede haber dependencias transitivas, luego la

relación ya está en 3FN.

### 9) Relación **Diagnósticos**

**Diagnósticos** (cod\_diag, informe, fecha, dni\_trab, dni\_enf)

Todos los atributos dependen directamente de la CP, luego la relación ya está en 3FN.

### 6.2.4. Resultado final

Y el resultado final:

**Provincias** (provincia, país)

CP: provincia

VNN: país

**Ciudades** (ciudad, provincia)

CP: ciudad

CAj: provincia → Provincias (provincia)

VNN: provincia

**Trabajadores** (dni\_trab, nombre, apellidos, dirección\_postal, ciudad, sueldo)

CP: dni\_trab

CAj: ciudad → Ciudades (ciudad)

**TrabTelf** (teléfono, dni\_trab)

CP: {teléfono, dni\_trab}

CAj: dni\_trab → Trabajadores (dni\_trab)

**Médicos** (dni\_trab, especialidad)

CP: dni\_trab

CAj: dni\_trab → Trabajadores (dni\_trab)

**Plantas** (especialidad\_planta, num\_habitaciones)

CP: especialidad\_planta

**Habitaciones** (especialidad\_planta, número, num\_camas, observaciones)

CP: {especialidad\_planta, número}

CAj: especialidad\_planta → Plantas (especialidad\_planta)

**Enfermeras** (dni\_trab, especialidad\_planta)

CP: dni\_trab

CAj: dni\_trab → Trabajadores (dni\_trab)

CAj: especialidad\_planta → Plantas (especialidad\_planta)

VNN: especialidad\_planta

**Enfermos** (dni\_enf, cod\_enf, nombre, apellidos, fecha\_nac, dirección\_postal, ciudad, dni\_trab, especialidad\_planta, número, fecha\_entrada, fecha\_salida)

CP: dni\_enf

ÚNICO: cod\_enf

CAj: ciudad → Ciudades (ciudad)

CAj: dni\_trab → Médicos (dni\_trab)

CAj: {especialidad\_planta, número} → Habitaciones (especialidad\_planta, número)

VNN: dni\_trab

VNN: {especialidad\_planta, número}

**EnfermosTelf** (teléfono, dni\_enf)

CP: {teléfono, dni\_enf}

CAj: dni\_enf → Enfermos (dni\_enf)

**Diagnósticos** (cod\_diag, informe, fecha, dni\_trab, dni\_enf)

CP: cod\_diag

CAj: dni\_trab → Médicos (dni\_trab)

CAj: dni\_enf → Enfermos (dni\_enf)

VNN: dni\_trab

VNN: dni\_enf

### Pérdidas semánticas:

- (1) Todo trabajador del hospital es médico o enfermera, pero no ambos a la vez.
- (2) Toda planta debe tener asignada al menos una enfermera.
- (3) Toda planta debe tener al menos una habitación.
- (4) Todo médico debe tratar al menos a un enfermo.
- (5) Todo médico que trata a un enfermo debe emitir un diagnóstico.

### Observaciones:

- (1) edad\_enf = fecha\_actual.Año – fecha\_nac.Año



## 6.3. Paso a tablas

Traduce a tablas el modelo relacional anterior utilizando el SGBDR MySQL.

### Ejercicio 6.3. Solución

```
CREATE DATABASE Hospital CHARACTER SET utf8 COLLATE utf8_spanish_ci;

USE Hospital;

CREATE TABLE Provincias (
  provincia VARCHAR(30),
  pais VARCHAR(20) NOT NULL,
  CONSTRAINT pro_prov_pk PRIMARY KEY (provincia)
);

CREATE TABLE Ciudades (
  ciudad VARCHAR(50),
  provincia VARCHAR(30) NOT NULL,
  CONSTRAINT ciu_ciu_pk PRIMARY KEY (ciudad),
  CONSTRAINT ciu_prov_fk FOREIGN KEY (provincia) REFERENCES Provincias (provincia)
);

CREATE TABLE Trabajadores (
  dni_trab VARCHAR(9),
  nombre VARCHAR(20),
  apellidos VARCHAR(60),
  direccion_postal VARCHAR(100),
  ciudad VARCHAR(50),
  sueldo FLOAT,
```

```

CONSTRAINT trab_dni_pk PRIMARY KEY (dni_trab),
CONSTRAINT trab_ciu_fk FOREIGN KEY (ciudad) REFERENCES Ciudades (ciudad)
);

CREATE TABLE TrabTelf (
telefono    VARCHAR(15),
dni_trab    VARCHAR(9),
CONSTRAINT trt_tel dni_pk PRIMARY KEY (telefono, dni_trab),
CONSTRAINT trt_dni_fk FOREIGN KEY (dni_trab) REFERENCES Trabajadores (dni_trab)
);

CREATE TABLE Medicos (
dni_trab    VARCHAR(9),
especialidad VARCHAR(20),
CONSTRAINT med_dni_pk PRIMARY KEY (dni_trab),
CONSTRAINT med_dni_fk FOREIGN KEY (dni_trab) REFERENCES Trabajadores (dni_trab)
);

CREATE TABLE Plantas (
especialidad_planta VARCHAR(20),
num_habitaciones    INTEGER,
CONSTRAINT pla_esp_pk PRIMARY KEY (especialidad_planta)
);

CREATE TABLE Habitaciones (
especialidad_planta VARCHAR(20),
numero              INTEGER,
num_camas           INTEGER,
observaciones       VARCHAR(200),
CONSTRAINT hab_espnum_pk PRIMARY KEY (especialidad_planta, numero),
CONSTRAINT hab_esp_fk FOREIGN KEY (especialidad_planta) REFERENCES Plantas (es
);

CREATE TABLE Enfermeras (
dni_trab    VARCHAR(9),
especialidad_planta VARCHAR(20) NOT NULL,
CONSTRAINT enf_dnitrab_pk PRIMARY KEY (dni_trab),

```

```

CONSTRAINT enf_dnitrab_fk FOREIGN KEY (dni_trab) REFERENCES Trabajadores (dni_trab),
CONSTRAINT enf_esp_fk FOREIGN KEY (especialidad_planta) REFERENCES Plantas (especialidad_planta);

```

```

CREATE TABLE Enfermos (
dni_enf          VARCHAR(9),
cod_enf          VARCHAR(10),
nombre          VARCHAR(20),
apellidos        VARCHAR(60),
fecha_nac        DATE,
direccion_postal VARCHAR(100),
ciudad           VARCHAR(50),
dni_trab         VARCHAR(9) NOT NULL,
especialidad_planta VARCHAR(20) NOT NULL,
numero          INTEGER NOT NULL,
fecha_entrada    DATE,
fecha_salida     DATE,
CONSTRAINT enf_dnienf_pk PRIMARY KEY (dni_enf),
CONSTRAINT enf_cod_uk UNIQUE (cod_enf),
CONSTRAINT enf_ciu_fk FOREIGN KEY (ciudad) REFERENCES Ciudades (ciudad),
CONSTRAINT enf_dnimed_fk FOREIGN KEY (dni_trab) REFERENCES Trabajadores (dni_trab),
CONSTRAINT enf_hab_fk FOREIGN KEY (especialidad_planta, numero) REFERENCES Haberes (especialidad_planta, numero);

```

```

CREATE TABLE EnfTelf (
telefono        VARCHAR(15),
dni_enf         VARCHAR(9),
CONSTRAINT ent_tel dni_pk PRIMARY KEY (telefono, dni_enf),
CONSTRAINT ent_dni_fk FOREIGN KEY (dni_enf) REFERENCES Enfermos (dni_enf) ON DELETE CASCADE;

```

```

CREATE TABLE Diagnósticos (
cod_diag        VARCHAR(10),
informe         VARCHAR(300),
dni_trab        VARCHAR(9) NOT NULL,
dni_enf         VARCHAR(9) NOT NULL,
CONSTRAINT dia_cod_pk PRIMARY KEY (cod_diag),

```

```
CONSTRAINT dia_dnitrab_fk FOREIGN KEY (dni_trab) REFERENCES Trabajadores (dni_  
CONSTRAINT dia_dnienf_fk FOREIGN KEY (dni_enf) REFERENCES Enfermos (dni_enf)  
);
```

### Pérdidas semánticas:

- (1) Todo trabajador del hospital es médico o enfermera, pero no ambos a la vez.
- (2) Toda planta debe tener asignada al menos una enfermera.
- (3) Toda planta debe tener al menos una habitación.
- (4) Todo médico debe tratar al menos a un enfermo.
- (5) Todo médico que trata a un enfermo debe emitir un diagnóstico.

### Observaciones:

- (1)  $\text{edad\_enf} = \text{fecha\_actual.Año} - \text{fecha\_nac.Año}$

## 7. Bibliografía

- Iván López, M.<sup>a</sup> Jesús Castellano. John Ospino. Bases de Datos. Ed. Garceta, 2a edición, 2017. ISBN: 978-8415452959
- Matilde Celma, Juan Carlos Casamayor y Laura Mota. Bases de datos relacionales. Ed. Prentice-Hall, 2003
- Cabrera Sánchez, Gregorio. Análisis y diseño detallado de aplicaciones informáticas de gestión. Ed. McGraw-Hill, 1st edition, 1999. ISBN: 8448122313
- Fernando Cano. Paso del E-R a tablas. <http://di002.edv.uniovi.es/~fcano/bdatos/materiales/paso-a-tablas.pdf>



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)