

UNIDAD 5

MODELO LÓGICO RELACIONAL

BASES DE DATOS 22/23
CFGS DAW

EJERCICIO DE REPASO. HOSPITAL.

Autores:

Abelardo Martínez y Pau Miñana

Fecha: 22/12/22

Licencia Creative Commons



Reconocimiento - NoComercial - CompartirIgual (by-nc-sa): No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

ÍNDICE DE CONTENIDO

1. HOSPITAL.....	3
1.1 Paso a modelo relacional.....	3
1.2 Normalización.....	5
1.3 Paso a tablas.....	11

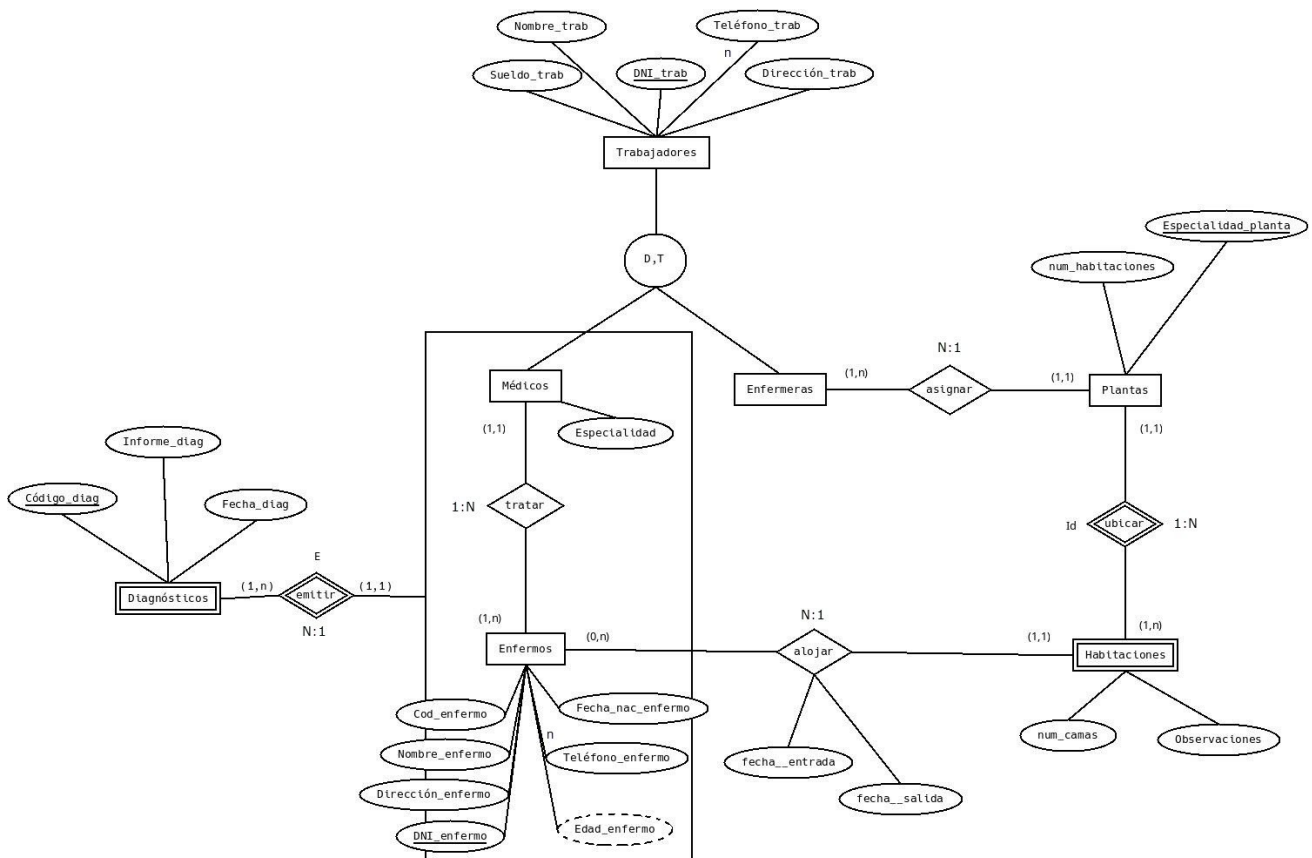
1. HOSPITAL

1.1 Paso a modelo relacional

Transforma a Modelo Relacional el siguiente diagrama E-R:

Observaciones:

- Los atributos nombre y dirección son compuestos tanto en la entidad Trabajadores como en la entidad Enfermos.
- El atributo “cod_enfermo” tiene valor único.
- Falta el atributo “número” como clave primaria de la entidad Habitaciones.



Trabajadores (dni_trab, nombre.nombre, nombre.apellidos, dirección.postal, dirección.ciudad, dirección.provincia, dirección.país, {teléfono}n, sueldo)

CP: {dni_trab}

Médicos (dni_trab, especialidad)

CP: {dni_trab}

CAj: {dni_trab} → Trabajadores

Plantas (especialidad_planta, num_habitaciones)

CP: {especialidad_planta}

Habitaciones (especialidad_planta, número, num_camás, observaciones)

CP: {especialidad_planta, número}

CAj: {especialidad_planta} → Plantas

Enfermeras (dni_trab, especialidad_planta)

CP: {dni_trab}

CAj: {dni_trab} → Trabajadores

CAj: {especialidad_planta} → Plantas

VNN: especialidad_planta

Enfermos (dni_enf, cod_enf, nombre.nombre, nombre.apellidos, fecha_nac, dirección.postal, dirección.ciudad, dirección.provincia, dirección.país, {teléfono}n, dni_trab, especialidad_planta, número, fecha_entrada, fecha_salida)

CP: {dni_enf}

ÚNICO: cod_enf

CAj: {dni_trab} → Médicos

CAj: {especialidad_planta, número} → Habitaciones

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

CAj: {dni_enf} → Enfermos

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.

1.2 Normalización

Normaliza a 3FN las relaciones anteriores.

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}n, 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

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}n, dni_trab, especialidad_planta, número, fecha_entrada, fecha_salida)

CP: {dni_enf}

ÚNICO: cod_enf

CAj: {dni_trab} → Médicos

CAj: {especialidad_planta, número} → Habitaciones

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

CAj: {especialidad_planta, número} → Habitaciones

VNN: dni_trab

VNN: {especialidad_planta, número}

edad_enf = fecha_actual.Año - fecha_nac.Año

EnfermosTelf (teléfono, dni_enf)

CP: {teléfono, dni_enf}

CAj: {dni_enf} → Enfermos

- El atributo derivado “edad” se elimina.

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

CAj: {especialidad_planta, número} → Habitaciones

VNN: dni_trab

VNN: {especialidad_planta, número}

2FN

Hay que eliminar las DF parciales. Como las relaciones Trabajadores, Médicos, Plantas, Enfermeras, Enfermos y Diagnósticos no tienen claves principales compuestas, 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.

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

Trabajadores (dni_trab, nombre, apellidos, dirección_postal, ciudad, sueldo)

CP: {dni_trab}

CAj: {ciudad} → Ciudades

Provincias (provincia, país)

CP: {provincia}

VNN: país

Ciudades (ciudad, provincia)

CP: {ciudad}

CAj: {provincia} → Provincias

VNN: provincia

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

CAj: {dni_trab} → Médicos

CAj: {especialidad_planta, número} → Habitaciones

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_camas, 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.

Y el resultado final:

Trabajadores (dni_trab, nombre, apellidos, dirección_postal, ciudad, sueldo)

CP: {dni_trab}

CAj: {ciudad} → Ciudades

TrabTelf (teléfono, dni_trab)

CP: {teléfono, dni_trab}

CAj: {dni_trab} → Trabajadores

Provincias (provincia, país)

CP: {provincia}

VNN: país

Ciudades (ciudad, provincia)

CP: {ciudad}

CAj: {provincia} → Provincias

VNN: provincia

Médicos (dni_trab, especialidad)

CP: {dni_trab}

CAj: {dni_trab} → Trabajadores

Plantas (especialidad_planta, num_habitaciones)

CP: {especialidad_planta}

Habitaciones (especialidad_planta, número, num_camas, observaciones)

CP: {especialidad_planta, número}

CAj: {especialidad_planta} → Plantas

Enfermeras (dni_trab, especialidad_planta)

CP: {dni_trab}

CAj: {dni_trab} → Trabajadores

CAj: {especialidad_planta} → Plantas

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

CAj: {dni_trab} → Médicos

CAj: {especialidad_planta, número} → Habitaciones

VNN: dni_trab

VNN: {especialidad_planta, número}

EnfermosTelf (teléfono, dni_enf)

CP: {teléfono, dni_enf}

CAj: {dni_enf} → Enfermos

Diagnósticos (cod_diag, informe, fecha, dni_trab, dni_enf)

CP: {cod_diag}

CAj: {dni_trab} → Médicos

CAj: {dni_enf} → Enfermos

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.

1.3 Paso a tablas

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

```
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            DOUBLE,  
    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_teldni_pk PRIMARY KEY (telefono, dni_trab),  
    CONSTRAINT trt_dni_fk FOREIGN KEY (dni_trab) REFERENCES Trabajadores (dni_trab) ON DELETE  
    CASCADE ON UPDATE CASCADE  
);
```

```
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_camapas            INTEGER,
    observaciones          VARCHAR(200),
    CONSTRAINT hab_espnum_pk PRIMARY KEY (especialidad_planta, numero),
    CONSTRAINT hab_esp_fk FOREIGN KEY (especialidad_planta) REFERENCES Plantas
(especialidad_planta) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

```
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) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

```
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 Habitaciones
(especialidad_planta, numero) ON DELETE RESTRICT ON UPDATE CASCADE
);
```

```
CREATE TABLE EnfTelf (
telefono         VARCHAR(15),
dni_enf          VARCHAR(9),
CONSTRAINT ent_teldni_pk PRIMARY KEY (telefono, dni_enf),
CONSTRAINT ent_dni_fk FOREIGN KEY (dni_enf) REFERENCES Enfermos (dni_enf) ON DELETE
CASCADE ON UPDATE 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_trab)

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.