**Actividad Repaso examen 1ª evaluación**

**1.2. Paso a modelo relacional normalizado**

**Usuario** (DNI, nombre\_completo, {telf.}n)

CP: DNI

**Paciente** (DNI)

CP: DNI

CAj: DNI 🡪 Usuario (DNI)

**Médico** (DNI, nºcolegiado)

CP: DNI

CAj: DNI 🡪 Usuario (DNI)

**Clínica** (IDClinica, nombre, dir\_comp)

CP: IDClinica

**Recetar** (DNI\_pac, DNI\_med, IDClinica, fecha\_vol)

CP: {DNI\_pac, DNI\_med, IDClinica, fecha\_vol}

CAj: DNI\_pac 🡪 Paciente (DNI)

CAj: DNI\_med 🡪 Medico (DNI)

CAj: IDClinica 🡪 Clínica (IDClinica)

**Análisis** (código, fecha, observaciones)

CP: código

**Asignar** (código\_an, DNI\_pac, DNI\_med, IDClinica, fecha\_vol)

CP: { código\_an, DNI\_pac, DNI\_med, IDClinica, fecha\_vol }

CAj: { DNI\_pac, DNI\_med, IDClinica, fecha\_vol) 🡪 Recetar

CAj: código\_an 🡪 Análisis (código)

UK: cod\_an

VNN: código\_an

**Linea\_analisis** (Número, valor, código\_an, código\_ind)

CP: {codígo\_an, numero)

CAj: código\_an 🡪 Análisis (código)

CAj: código\_in 🡪 Indicador (código)

**Indicador** (código, nombre, min\_ref, max\_ref, código\_cat)

CP: código

CAj: código\_cat 🡪 Categoría (código)

**Categoría** (código, nombre)

CP: código

**1FN**

Usuario no está en 1FN, porque Teléfono es multivaluado. Hay que crear una nueva tabla para teléfono.

**Usuario** (DNI, nombre\_completo)

CP: DNI

**Teléfono** (DNI, teléfono)

CP: {DNI, teléfono}

CAj: DNI 🡪 Usuario (DNI)

Todo lo demás ya está en 1FN

**Paciente** (DNI)

CP: DNI

CAj: DNI 🡪 Usuario (DNI)

**Médico** (DNI, nºcolegiado)

CP: DNI

CAj: DNI 🡪 Usuario (DNI)

**Clínica** (IDClinica, nombre, dir\_comp)

CP: IDClinica

**Recetar** (DNI\_pac, DNI\_med, IDClinica, fecha\_vol)

CP: {DNI\_pac, DNI\_med, IDClinica, fecha\_vol}

CAj: DNI\_pac 🡪 Paciente (DNI)

CAj: DNI\_med 🡪 Medico (DNI)

CAj: IDClinica 🡪 Clínica (IDClinica)

**Análisis** (código, fecha, observaciones)

CP: código

**Asignar** (código\_an, DNI\_pac, DNI\_med, IDClinica, fecha\_vol)

CP: { código\_an, DNI\_pac, DNI\_med, IDClinica, fecha\_vol }

CAj: { DNI\_pac, DNI\_med, IDClinica, fecha\_vol) 🡪 Recetar

CAj: código\_an 🡪 Análisis (código)

UK: cod\_an

VNN: código\_an

**Linea\_analisis** (Número, valor, código\_an, código\_ind)

CP: {codígo\_an, numero)

CAj: código\_an 🡪 Análisis (código)

CAj: código\_in 🡪 Indicador (código)

**Indicador** (código, nombre, min\_ref, max\_ref, código\_cat)

CP: código

CAj: código\_cat 🡪 Categoría (código)

**Categoría** (código, nombre)

CP: código

**2FN**

Todas las tablas ya están en 2FN

**3FN**

**Usuario** (DNI, nombre, apellidos)

CP: DNI

**Teléfono** (DNI, teléfono)

CP: {DNI, teléfono}

CAj: DNI 🡪 Usuario (DNI)

**Provincia** (provincia)

CP: provincia

**Ciudad** (ciudad, provincia)

CP: {ciudad, provincia}

CAj: provincia 🡪 Provincia (Provincia)

**Médico** (DNI, nºcolegiado)

CP: DNI

CAj: DNI 🡪 Usuario (DNI)

**Clínica** (IDClinica, nombre, dir\_postal, ciudad)

CP: IDClinica

CAj: ciudad 🡪 Ciudad (ciudad)

**Recetar** (DNI\_pac, DNI\_med, IDClinica, fecha\_vol)

CP: {DNI\_pac, DNI\_med, IDClinica, fecha\_vol}

CAj: DNI\_pac 🡪 Paciente (DNI)

CAj: DNI\_med 🡪 Medico (DNI)

CAj: IDClinica 🡪 Clínica (IDClinica)

**Análisis** (código, fecha, observaciones)

CP: código

**Asignar** (código\_an, DNI\_pac, DNI\_med, IDClinica, fecha\_vol)

CP: { código\_an, DNI\_pac, DNI\_med, IDClinica, fecha\_vol }

CAj: { DNI\_pac, DNI\_med, IDClinica, fecha\_vol) 🡪 Recetar

CAj: código\_an 🡪 Análisis (código)

UK: cod\_an

VNN: código\_an

**Linea\_analisis** (Número, valor, código\_an, código\_ind)

CP: {codígo\_an, numero)

CAj: código\_an 🡪 Análisis (código)

CAj: código\_in 🡪 Indicador (código)

**Indicador** (código, nombre, min\_ref, max\_ref, código\_cat)

CP: código

CAj: código\_cat 🡪 Categoría (código)

**Categoría** (código, nombre)

CP: código

**1.3. Paso a tablas. Indicar sentencias**

CREATE TABLE usuarios (

dni CHAR(9),

nombre VARCHAR(20),

apellidos VARCHAR(30),

CONSTRAINT usu\_dni\_pk PRIMARY KEY (dni)

);

CREATE TABLE telefonos (

dni CHAR(9),

telefono VARCHAR(12),

CONSTRAINT tel\_dnitel\_pk PRIMARY KEY (dni, telefono)

CONSTRAINT tel\_dni\_fk FOREIGN KEY (dni) REFERENCES usuarios (dni)

);

CREATE TABLE provincia (

provincia VARCHAR(20)

CONSTRAINT pro\_pro\_pk PRIMARY KEY (provincial)

);

CREATE TABLE ciudad (

ciudad VARCHAR(20),

provincia VARCHAR(20),

CONSTRAINT ciu\_ciuprov\_pk PRIMARY KEY (ciudad, provincial),

CONSTRAINT ciu\_pro\_fk FOREIGN KEY (provincial) REFERENCES provincia (provincia)

);

CREATE TABLE pacientes (

dni CHAR(9),

direccion\_postal VARCHAR(30),

ciudad VARCHAR(20),

CONSTRAINT pac\_dni\_pk PRIMARY KEY (dni),

CONSTRAINT pac\_dni\_fk FOREIGN KEY (dni) REFERENCES usuarios (dni),

CONSTRAINT pac\_ciu\_fk FOREIGN KEY (ciudad) REFERENCES ciudad (ciudad)

);

CREATE TABLE medicos (

dni CHAR(9),

ncolegiado VARCHAR(10),

CONSTRAINT med\_dni\_pk PRIMARY KEY (dni),

CONSTRAINT med\_dni\_fk FOREIGN KEY (dni) REFERENCES usuarios (dni),

);

CREATE TABLE clinicas (

idclinica VARCHAR(10),

nombre VARCHAR(20),

direccion\_postal VARCHAR(30),

ciudad VARCHAR(20),

CONSTRAINT cli\_idc\_pk PRIMARY KEY (idclinica),

CONSTRAINT cli\_ciu\_fk FOREIGN KEY (ciudad) REFERENCES ciudad (ciudad)

);

CREATE TABLE volante (

dni\_pac CHAR(9),

dni\_med CHAR(9),

idclinica VARCHAR(10),

fecha\_vol DATE,

CONSTRAINT vol\_dnipdnimidcfec\_pk PRIMARY KEY (dni\_pac, dni\_med, idclinica, fecha\_vol),

CONSTRAINT vol\_dnip\_fk FOREIGN KEY (dni\_pac) REFERENCES pacientes (dni),

CONSTRAINT vol\_dnim\_fk FOREIGN KEY (dni\_med) REFERENCES medicos (dni),

CONSTRAINT vol\_idc\_fk FOREIGN KEY (idclinica) REFERENCES clinicas (idclinica)

);

CREATE TABLE analisis (

cod\_analisis VARCHAR(10),

observaciones VARCHAR(100),

fecha\_analisis DATE,

CONSTRAINT ana\_cod\_pk PRIMARY KEY (cod\_analisis)

);

CREATE TABLE asignar (

dni\_pac CHAR(9),

dni\_med CHAR(9),

idclinica VARCHAR(10),

fecha\_vol DATE,

cod\_analisis VARCHAR(10),

CONSTRAINT asi\_dnipdnimidcfecv\_pk PRIMARY KEY (dni\_pac, dni\_med, idclinica, fecha\_vol),

CONSTRAINT asi\_dnipdnimidclfecv\_fk FOREIGN KEY (dni\_pac, dni\_med, idclinica, fecha\_vol) REFERENCES volante (dni\_pac, dni\_med, idclinica, fecha\_vol),

CONSTRAINT asi\_codan\_fk FOREIGN KEY (cod\_analisis) REFERENCES analisis (cod\_analisis),

CONSTRAINT asi\_codan\_uk UNIQUE (cod\_analisis)

);

CREATE TABLE categorias (

cod\_cat VARCHAR(10),

nombre VARCHAR(20),

CONSTRAINT cat\_cod\_pk PRIMARY KEY (cod\_cat)

);

CREATE TABLE indicadores (

cod\_ind VARCHAR(10),

nombre VARCHAR(20),

minref FLOAT,

maxref FLOAT,

cod\_cat VARCHAR(10),

CONSTRAINT ind\_cod\_pk PRIMARY KEY (cod\_ind),

CONSTRAINT ind\_codcat\_fk FOREIGN KEY (cod\_cat) REFERENCES categorias (cod\_cat)

);

CREATE TABLE lineas\_analisis (

cod\_analisis VARCHAR(10),

orden INTEGER,

cantidad FLOAT,

cod\_ind VARCHAR(10),

CONSTRAINT lin\_codanord\_pk PRIMARY KEY (cod\_analisis, orden),

CONSTRAINT lin\_codan\_fk FOREIGN KEY (cod\_analisis) REFERENCES analisis (cod\_analisis),

CONSTRAINT lin\_codind\_fk FOREIGN KEY (cod\_ind) REFERENCES indicadores (cod\_ind)

);

**1.4. Modificación de metadatos**

ALTER TABLE medicos ADD CONSTRAINT med\_nco\_uk UNIQUE (ncolegiado);

ALTER TABLE analisis ADD aseguradora VARCHAR(20);

**1.5. Inserción de datos**

INSERT INTO provincia (provincia) VALUES

(‘Alicante’);

INSERT INTO ciudad (ciudad, provincia) VALUES

(‘Alicante’, ‘Alicante’);

INSERT INTO pacientes (dni, direccion\_postal, ciudad) VALUES

(‘23231245A’, ‘Avenida de la universidad, 40’, ‘Alicante’);

INSERT INTO medicos (dni, ncolegiado) VALUES

(‘16493202B’, ‘2344’);

INSERT INTO clínicas (idclinica, nombre, direccion\_postal, ciudad) VALUES

(‘SOR1829’, ‘Sorolla’, ‘Calle Argamasilla, 27’, ‘Alicante’);

INSERT INTO volante (dni\_pac, dni\_med, idclinica, fecha\_vol) VALUES

(‘23231245A’, ‘16493202B’, ‘SOR1829’, ‘2021-12-05’);

INSERT INTO análisis (cod\_analisis, observaciones, fecha\_analisis) VALUES

(‘987546’, ‘Revisión anual’, ‘2021-12-12’);

INSERT INTO asignar (dni\_pac, dni\_med, idclinica, fecha\_vol, cod\_analisis) VALUES

(‘23231245A’, ‘16493202B’, ‘SOR1829’, ‘2021-12-05’, ‘987546’);

INSERT INTO categorias (cod\_cat, nombre) VALUES

(‘HEM001’, ‘Hematología’),

(‘BIOQ001’, ‘Bioquímica’);

INSERT INTO indicadores (cod\_ind, nombre, minref, maxref, cod\_cat) VALUES

(‘Hemat001’, ‘Hematíes’, 4.50, 6.4, ‘HEM001’),

(‘Linfoc001’, ‘Linfocitos’, 20, 40, ‘HEM001’),

(‘Calc001’, ‘Calcio’, 8.5, 10.5, ‘BIOQ001’),

(‘Creat001’, ‘Creatinina’, 0.5, 1.30, ‘BIOQ001’);

INSERT INTO lineas\_analisis (cod\_analisis, orden, cantidad, cod\_ind) VALUES

(‘987546’, 1, 5.04, ‘Hemat001’),

(‘987546’, 2, 34.8, ‘Linfoc001’),

(‘987546’, 12, 9.40, ‘Calc001’),

(‘987546’, 13, 0.90, ‘Creat001’);

**2. Universidad**

**2.1. Paso a modelo relacional normalizado**

\* Se asumen la participaciones mínimas a 0, salvo que el ejercicio especifique lo contrario.

Diagrama E-R inicial:

**Facultad** (número, ubicación, nombre)

CP: número

**Docente** (cédula, nombre, apellido, título, número\_fac)

CP: cédula

CAj: número\_fac 🡪 Facultad (número)

**Asignatura** (código, nombre, créditos)

CP: código

**Enseñar** (cédula, código)

CP: {cédula, código}

CAj: cédula 🡪 Docente (cédula)

CAj: código 🡪 Asignatura (código)

**Estudiante** (nºId, nombre, apellidos, dirección)

CP: nºId

**Inscribir** (nºId, código)

CP: {nºId, código}

CAj: nºId 🡪 Estudiante (nºId)

CAj: código 🡪 Asignatura (código)

**Decano** (cédula, nombre, apellidos, teléfono)

CP: cédula

**Dirigir** (número\_fac, cédula\_dec)

CP: {número\_fac, cédula\_dec}

CAj: número\_fac 🡪 Facultad (número)

CAj: cédula\_dec 🡪 Decano (cédula)

Único: cédula\_dec

**Paso a 3FN**

El diagrama ya está en 3FN.

2.2. Inserción de datos (DML)

Insertar:

* 3 estudiantes
* Inscritos en 3 asignaturas
* Asociadas cada una a un docente distinto (3 docentes)
* De una facultad distinta (3 facultades)
* Con un decano distinto cada facultad (3 decanos)

INSERT INTO decano (cedula, nombre, apellidos, telefono) VALUES

(‘11111111A’, ‘Pepe’, ‘Garcia Morales’, ‘645997621’),

(‘22222222B’, ‘Maria’, ‘Fernandez Lopez’, ‘643975632’),

(‘33333333C’, ‘Luisa’, ‘Garcia Perez’, ‘640516985’);

INSERT INTO facultad (numero, nombre, ubicación) VALUES

(‘1654’, ‘Ciencias’, ‘Campus Palo Alto’),

(‘6495’, ‘Economia’, ‘Campus Villa verde’),

(‘1649’, ‘Medicina’, ‘Campus Palo Alto’);

INSERT INTO dirigir (numero\_fac, cedula\_dec) VALUES

(‘1654’, ‘11111111A’),

(‘6495’, ‘22222222B’),

(‘1649’, ‘33333333C’);

INSERT INTO docente (cedula, nombre, apellido, título, numero\_fac) VALUES

(‘25252525A’, ‘Enrique’, ‘Gonzalez’, ‘Profesor Asociado’, ‘1654’),

(‘15151515B’, ‘Ana’, ‘Mendoza’, ‘Directora’, ‘6495’),

(‘95959595C’, ‘Pilar’, ‘Lopez’, ‘Profesora titular’, ‘1649’);

INSERT INTO asignatura (codigo, nombre, creditos) VALUES

(‘MAT001’, ‘Matemáticas’, 12),

(‘FIS001’, ‘Fisica’, 12),

(‘ING001’, ‘Inglés’, 12);

INSERT INTO enseñar (cedula, codigo) VALUES

(‘25252525A’, ‘MAT001’),

(‘15151515B’, ‘FIS001’),

(‘95959595C’, ‘ING001’);

INSERT INTO estudiante (nºId, nombre, apellidos, dirección) VALUES

(‘1680’, ‘Pilatos’, ‘Garcia’, ‘Calle mayor, Sevilla’),

(‘2468’, ‘Fernando’, ‘Quiroga’, ‘Calle de arriba, Sevilla’),

(‘1720’, ‘Sara’, ‘Medina’, ‘Calle ancha, Sevilla’);

INSERT INTO inscribir (nºId, código) VALUES

(‘1680’, ‘MAT001’),

(‘2468’, ‘FIS001’),

(‘1720’, ‘ING001’);

**3. Red de oficinas**

**3.2. Pasa a modelo relacional el diagrama E-R**

**Employee** (emp\_id, name, Street, city)

CP: emp\_id

**Job** (title, level)

CP: title

**Branch** (name, city)

CP: name

**Works\_on** (emp\_id, title, branch\_name)

CP: {emp\_id, title, branch\_name)

CAj: emp\_id 🡪 Employee (emp\_id)

CAj: titles 🡪 Job (title)

CAj: branch\_name 🡪 Branch (name)

**Manager** (id, name, {emp\_id, title, branch\_name})

CP: id

CAj: {emp\_id, title, branch\_name} 🡪 works\_on

VNN: {emp\_id, title, branch\_name}

**Becario** (id, name, id\_man)

CP: {id, id\_man}

CAj: id\_man 🡪 Manager (id)

\* Becario tiene una restricción de identidad respecto a Manager, por eso se incluye la CP de manager en la CP de becario, como parte de su CP

**4. Empleados**

**4.1 Paso a modelo relacional normalizado**

**Empleados**(NSS, nombre, apellidos, sueldo, NSS\_supervisor)

CP: NSS

CAj: NSS\_supervisor 🡪 Empleados (NSS)

**Amigos** (NSS, NSS\_amigo)

CP: {NSS, NSS\_amigo}

CAj: NSS 🡪 Empleados(NSS)

CAj: NSS\_amigo 🡪 Empelados(NSS)

4.2. Inserción de datos (DML)

INSERT INTO empleados (nss, nombre, apellidos, sueldo, nss\_supervisor) VALUES

(‘935665’, ‘Lucia’, ‘Garcia Lorell’, 70000, NULL);

INSERT INTO empleados (nss, nombre, apellidos, sueldo, nss\_supervisor) VALUES

(‘566777’, ‘Ana’, ‘Obeno Gomez’, 44000, ‘935665’),

(‘123234’, ‘Miguel’, ‘Sanchez Lopez’, 40000, ‘935665’);

INSERT INTO amigos (nss, nss\_amigo) VALUES

(‘566777’, ‘935665’);

**5. Alumnos**

**5.1. Paso a modelo relacional normalizado**

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

CP: id\_alumno

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

**Ayudar** (id\_alumno\_ayudado, id\_alumno\_ayudante)

CP: {id\_alumno\_ayudado, id\_alumno\_ayudante}

CAj: id\_alumno\_ayudado 🡪 Alumno (id\_alumno)

CAj: id\_alumno\_ayudante 🡪 Alumno (id\_alumno)

**5.2. Inserción de datos (DML)**

INSERT INTO alumno (id\_alumno, nombre, id\_director) VALUES

(‘1111’, ‘Luke Skywalker’, NULL),

(‘4444’, ‘Baby Yoda’, NULL);

INSERT INTO alumno (id\_alumno, nombre, id\_director) VALUES

(‘2222’, ‘Kylo Ren’, ‘1111’),

(‘3333’, ‘Rey Skywalker’, ‘1111’);

INSERT INTO ayudar (id\_alumno\_ayudado, id\_alumno\_ayudante) VALUES

(‘2222’, ‘3333’),

(‘4444’, ‘1111’);

**6. Hospital**

**6.2. Paso a modelo relacional**

**Trabajadores** (dni\_trab, {telefono}n, sueldo, nombre.nombre, nombre.apellidos, dirección.postal, dirección.ciudad, dirección.provincia, dirección.país)

CP: dni\_trab

*\* Especialización disjunta y total. Todos los trabajadores deben ser o Médicos o Enfermeras, no pudiendo ser ambos a la vez, y no pudiendo ser de otro tipo.*

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

CP: dni\_enf

CAj: dni\_enf 🡪 Trabajadores (dni\_trab)

CAj: especialidad\_planta 🡪 Planta

VNN: especialidad\_planta

**Médicos** (dni\_med, especialidad)

CP: dni\_med

CAj: dni\_med 🡪 Trabajadores (dni\_trab)

*\* Comprobar que todo médico tiene un enfermo asociado.*

*\* Todo médico que trata a un enfermo debe emitir un diagnóstico.*

**Enfermos** (dni\_enfermo, cod\_enfermo, nombre\_enfermo.nombre, nombre\_enfermo.apellidos, dirección\_enfermo.postal, dirección\_enfermo.ciudad, dirección\_enfermo.provincia, dirección\_enfermo.país, fecha\_nac, {teléfono\_enfermo}n, edad\_enfermo, fecha\_entrada, fecha\_salida, dni\_med, {numero, especialidad\_planta})

CP: dni\_enfermo

Único: cod\_enfermo

edad\_enfermo = fecha\_actual - fecha\_nac

CAj: dni\_med 🡪 Médico (dni\_med)

CAj: {numero, especialidad\_planta} 🡪 Habitaciones {numero, especialidad\_planta}

VNN: dni\_med

VNN: {numero, especialidad\_planta}

**Plantas** (especialidad\_panta, num\_hab)

CP: especialidad\_planta

*\* Comprobar que toda planta tiene una enfermera asignada.*

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

CP: {numero, especialidad\_planta}

CAj: especialidad\_planta 🡪 Planta (especialidad\_planta)

\* Debilidad de identidad.

**Diagnósticos** (código\_diag, informe\_diag, fecha\_diag, dni\_med, dni\_enfermo)

CP: código\_diag

CAj: dni\_med 🡪 Médico (dni\_med)

CAj: dni\_enfermo 🡪 Enfermo (dni\_enfermo

VNN: dni\_med

VNN: dni\_enf

*\* Debilidad de existencia frente a la agregación Médico-enfermo.*

**1FN**

Trabajadores y enfermos 🡪 atributos compuestos y multivaluados. El atributo derivado “edad” de enfermo se elimina. Se crean nuevas tablas para los atributos multivaluados Teléfono

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

CP: dni\_trab

*\* Especialización disjunta y total. Todos los trabajadores deben ser o Médicos o Enfermeras, no pudiendo ser ambos a la vez, y no pudiendo ser de otro tipo.*

**Enfermos** (dni\_enfermo, cod\_enfermo, nombre, apellidos, dirección\_enfermo\_postal, ciudad\_enfermo, provincia\_enfermo, país\_enfermo, fecha\_nac, fecha\_entrada, fecha\_salida, dni\_med, {numero, especialidad\_planta})

CP: dni\_enfermo

Único: cod\_enfermo

CAj: dni\_med 🡪 Médico (dni\_med)

CAj: {numero, especialidad\_planta} 🡪 Habitaciones {numero, especialidad\_planta}

VNN: dni\_med

VNN: {numero, especialidad\_planta}

**Teléfono\_trabajadores** (dni\_trab, teléfono)

CP: {dni\_trab, teléfono}

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

**Teléfono\_enfermos** (dni\_enfermo, teléfono)

CP: {dni\_enfermo, teléfono}

CAj: dni\_enfermo 🡪 Enfermo (dni\_enfermo)

*El resto de tablas están ya en 1FN:*

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

CP: dni\_enf

CAj: dni\_enf 🡪 Trabajadores (dni\_trab)

CAj: especialidad\_planta 🡪 Planta

VNN: especialidad\_planta

**Médicos** (dni\_med, especialidad)

CP: dni\_med

CAj: dni\_med 🡪 Trabajadores (dni\_trab)

*\* Comprobar que todo médico tiene un enfermo asociado.*

*\* Todo médico que trata a un enfermo debe emitir un diagnóstico.*

**Plantas** (especialidad\_panta, num\_hab)

CP: especialidad\_planta

*\* Comprobar que toda planta tiene una enfermera asignada.*

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

CP: {numero, especialidad\_planta}

CAj: especialidad\_planta 🡪 Planta (especialidad\_planta)

\* Debilidad de identidad.

**Diagnósticos** (código\_diag, informe\_diag, fecha\_diag, dni\_med, dni\_enfermo)

CP: código\_diag

CAj: dni\_med 🡪 Médico (dni\_med)

CAj: dni\_enfermo 🡪 Enfermo (dni\_enfermo

VNN: dni\_med

VNN: dni\_enf

*\* Debilidad de existencia frente a la agregación Médico-enfermo.*

**2FN**

Todas las tablas están ya en 2FN

**3FN**

Todas las tablas, menos Trabajadores y Enfermos están ya en 3FN.

**Teléfono\_trabajadores** (dni\_trab, teléfono)

CP: {dni\_trab, teléfono}

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

**Teléfono\_enfermos** (dni\_enfermo, teléfono)

CP: {dni\_enfermo, teléfono}

CAj: dni\_enfermo 🡪 Enfermo (dni\_enfermo)

*El resto de tablas están ya en 1FN:*

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

CP: dni\_enf

CAj: dni\_enf 🡪 Trabajadores (dni\_trab)

CAj: especialidad\_planta 🡪 Planta

VNN: especialidad\_planta

**Médicos** (dni\_med, especialidad)

CP: dni\_med

CAj: dni\_med 🡪 Trabajadores (dni\_trab)

*\* Comprobar que todo médico tiene un enfermo asociado.*

*\* Todo médico que trata a un enfermo debe emitir un diagnóstico.*

**Plantas** (especialidad\_panta, num\_hab)

CP: especialidad\_planta

*\* Comprobar que toda planta tiene una enfermera asignada.*

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

CP: {numero, especialidad\_planta}

CAj: especialidad\_planta 🡪 Planta (especialidad\_planta)

\* Debilidad de identidad.

**Diagnósticos** (código\_diag, informe\_diag, fecha\_diag, dni\_med, dni\_enfermo)

CP: código\_diag

CAj: dni\_med 🡪 Médico (dni\_med)

CAj: dni\_enfermo 🡪 Enfermo (dni\_enfermo

VNN: dni\_med

VNN: dni\_enf

*\* Debilidad de existencia frente a la agregación Médico-enfermo.*

Trabajadores y Enfermos tienen dependencias transitivas con los atributos de la dirección. Se crean nuevas tablas.

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

CP: dni\_trab

*\* Especialización disjunta y total. Todos los trabajadores deben ser o Médicos o Enfermeras, no pudiendo ser ambos a la vez, y no pudiendo ser de otro tipo.*

**Enfermos** (dni\_enfermo, cod\_enfermo, nombre, apellidos, dirección\_enfermo\_postal, ciudad\_enfermo, fecha\_nac, fecha\_entrada, fecha\_salida, dni\_med, {numero, especialidad\_planta})

CP: dni\_enfermo

Único: cod\_enfermo

CAj: dni\_med 🡪 Médico (dni\_med)

CAj: {numero, especialidad\_planta} 🡪 Habitaciones {numero, especialidad\_planta}

VNN: dni\_med

VNN: {numero, especialidad\_planta}

*Observaciones*:

*edad = fecha\_actual - fecha\_nac*

**Ciudad** (Ciudad, Provincia)

CP: ciudad

CAj: Provincia 🡪 Provincia (Provincia)

VNN: Provincia

**Provincia** (provincia, País)

CP: Provincia

VNN: País

**6.3. Paso a tablas (DDL)**

CREATE TABLE provincias (

provincia VARCHAR(20),

país VARCHAR(20) NOT NULL,

CONSTRAINT pro\_pro\_pk PRIMARY KEY (provincia)

);

CREATE TABLE ciudades (

ciudad VARCHAR(20),

provincia VARCHAR(20) NOT NULL,

CONSTRAINT ciu\_ciu\_pk PRIMARY KEY (ciudad),

CONSTRAINT ciu\_pro\_fk FOREIGN KEY (provincia) REFERENCES provincias (provincia)

);

CREATE TABLE trabajadores (

dni\_trab CHAR(9),

nombre VARCHAR(20),

apellidos VARCHAR(40),

direccion\_postal VARCHAR(50),

ciudad VARCHAR(20),

sueldo FLOAT,

CONSTRAINT tra\_dni\_pk PRIMARY KEY (dni\_trab),

CONSTRAINT tra\_ciu\_fk FOREIGN KEY (ciudad) REFERENCES ciudades (ciudad)

);

CREATE TABLE telefono\_trabajador (

telefono VARCHAR(12),

dni\_trab CHAR(9),

CONSTRAINT tel\_teldnitrab\_pk PRIMARY KEY (telefono, dni\_trab),

CONSTRAINT tel\_dnitrab\_fk FOREIGN KEY (dni\_trab) REFERENCES trabajadores (dni\_trab)

);

CREATE TABLE medicos (

dni\_med CHAR(9),

especialidad VARCHAR(60),

CONSTRAINT med\_dni\_pk PRIMARY KEY (dni\_med),

CONSTRAINT med\_dni\_fk FOREIGN KEY (dni\_med) REFERENCES trabajadores (dni\_trab)

);

CREATE TABLE plantas (

especialidad\_planta VARCHAR(50),

num\_hab INTEGER,

CONSTRAINT pla\_esp\_pk PRIMARY KEY (especialidad\_planta)

);

CREATE TABLE habitaciones (

especialidad\_planta VARCHAR(50),

numero INTEGER,

numero\_camas INTEGER,

observaciones VARCHAR(100),

CONSTRAINT hab\_espnum\_pk PRIMARY KEY (especialidad\_planta, numero),

CONSTRAINT hab\_esp\_fk FOREIGN KEY (especialidad\_planta) REFERENCES plantas (especialidad\_planta)

);

CREATE TABLE enfermeras (

dni\_enf CHAR(9),

especialidad\_planta VARCHAR(50) NOT NULL,

CONSTRAINT enf\_dni\_pk PRIMARY KEY (dni\_enf),

CONSTRAINT enf\_esp\_fk FOREIGN KEY (especialidad\_planta) REFERENCES plantas (especialidad\_planta),

CONSTRAINT enf\_dni\_fk FOREIGN KEY (dni\_enf) REFERENCES trabajadores (dni\_trab)

);

CREATE TABLE enfermos (

dni\_enfermo CHAR(9),

cod\_enf VARCHAR(8),

nombre VARCHAR(20),

apellidos VARCHAR(40),

fecha\_nac DATE,

direccion\_postal VARCHAR(60),

ciudad VARCHAR(20),

dni\_med CHAR(9) NOT NULL,

especialidad\_planta VARCHAR(50) NOT NULL,

numero INTEGER NOT NULL,

fecha\_entrada DATE,

fecha\_salida DATE,

CONSTRAINT enf\_dnienfer\_pk PRIMARY KEY (dni\_enfermo),

CONSTRAINT enf\_cod\_uk UNIQUE (cod\_enf),

CONSTRAINT enf\_ciu\_fk FOREIGN KEY (ciudad) REFERENCES ciudades (ciudad),

CONSTRAINT enf\_dnimed\_fk FOREIGN KEY (dni\_med) REFERENCES medicos (dni\_med),

CONSTRAINT enf\_espnum\_fk FOREIGN KEY (especialidad\_planta, numero) REFERENCES habitaciones (especialidad\_planta, numero)

);

CREATE TABLE telefono\_enfermos (

dni\_enfermo CHAR(9),

telefono VARCHAR(12),

CONSTRAINT telenf\_dnienf\_pk PRIMARY KEY (dni\_enfermo, telefono),

CONSTRAINT telenf\_dnienf\_fk FOREIGN KEY (dni\_enfermo) REFERENCES enfermos (dni\_enfermo)

);

CREATE TABLE diagnosticos (

cod\_diag VARCHAR(8),

informe VARCHAR(20),

fecha DATE,

dni\_med CHAR(9) NOT NULL,

dni\_enfermo CHAR(9) NOT NULL,

CONSTRAINT día\_cod\_pk PRIMARY KEY (cod\_diag),

CONSTRAINT dia\_dnimed\_fk FOREIGN KEY (dni\_med) REFERENCES medicos (dni\_med),

CONSTRAINT dia\_dnienf\_fk FOREIGN KEY (deni\_enfermo) REFERENCES enfermos (dni\_enfermo)

);

**Examen año pasado**

1. **Crear un diagrama E-R**

A white text with green text

Description automatically generated

A close up of a text

Description automatically generated

A black text on a white background

Description automatically generated

\* 1. 🡪 comunicadores internos (tiene varios) 🡪 si tiene varios, indica que es **multivaluado**.

\* 7. Tiene pinta de agreagación, más que de relación ternaria. El “turno” será atributo de la relación “controla”.

1. **A partir de un diagrama E-R, hacer las sequencias DDL para crear las tablas y restricciones, e insertar ciertos datos (DML)**

\* Poner siempre primero el CREATE DATABASE + el “use database”

A diagram of a company

Description automatically generated

**Solución 1**

A diagram of a person's work flow

Description automatically generated

El atributo “Turno” iría en la parte de la agregación, en la tabla de Mandar (turno, etc)

2. Paso a tablas

A yellow and black text

Description automatically generated

A white paper with black text

Description automatically generated

A screenshot of a computer code

Description automatically generated