

[5 PUNTOS] EJERCICIO 1: MODELADO LÓGICO RELACIONAL

Realiza el paso a tablas en tercera forma normal (3FN).

VOTANTE(dni, nom_comp, teléfono)

CP: dni

ID_DIGITAL(id, password, dir_tipo, dirección, dir_piso, dir_num, localidad, circunscripción, dni)

CP: id

CAj: dni → VOTANTE {dni}

VNN: dni

UK: dni

SENADOR(dni_sen, circ_presenta, partido, num_votos)

CP: dni_sen

CAj: dni_sen → VOTANTE {dni}

VNN: circ_presenta

Derivado: num_votos = filas VOTA_SEN donde aparece SENADOR.dni_sen

Valores de partido son: "ASIR", "DAM", "DAW", "PHT" y "PRO".

VOTA_SEN(id_dig, dni_sen)

CP: {id_dig, dni_sen}

CAj: id_dig → ID_DIGITAL {id}

CAj: dni_sen → SENADOR {dni}

Restricciones Adicionales:

ID_DIGITAL.circunscripción = SENADOR.circ_presenta

id_dig puede repetirse como mucho 3 veces en la tabla

HORARIO(h_ini, h_fin, día)

CP: {h_ini, h_fin, día}

Restricción adicional: día >= 03/03/2025 & día <= 18/03/2025

Restricción adicional: h_ini < h_fin & h_fin - h_ini <= 20000

CENTRO(id_centro, nom_centro, dir_centro)

CP: id_centro

UK: {nom_centro, dir_centro}

INTERVENTOR(dni_int, id_centro, sueldo, int_supervisado)

CP: dni_int

CAj: dni_int → VOTANTE {dni}

CAj: id_centro → CENTRO {id_centro} **B:C**

CAj: int_supervisado → INTERVENTOR {dni_int}

VNN: sueldo

VNN: id_centro

ACCIÓN(num_reg, desccpcion, dni_int)

CP: {num_reg, dni_int}

CAj: dni_int → Interventor {dni}

PIDE_CITA(dni_votante, día, h_ini, h_fin, id_centro)

CP: {dni_votante, día, h_ini, h_fin, id_centro}

CAj: dni_votante → VOTANTE {dni}

CAj: {día, h_ini, h_fin} → HORARIO {día, h_ini, h_fin}

CAj: id_centro → CENTRO {id_centro}

Restricciones:

- (1) num_votos → num_votos se calcula contando las veces que dni_sen aparece en la tabla **VOTA_SEN** (los votos que ha recibido dicho Senador). num_votos no se elimina porque se considera imprescindible para llevar la cuenta fácilmente, pero se traslada al Senador como atributo porque no depende de la **ID_DIG** que lo vote.
- (2) Una localidad sólo puede aparecer junto con una circunscripción. Pero una circunscripción puede aparecer con distintas localidades. (Es decir, una localidad pertenece sólo a una circunscripción).
- (3) La ID_DIGITAL.circunscripción y la SENADOR.circ_presenta deben ser la misma para que ID_DIGITAL.id y SENADOR.dni_sen aparezcan en **VOTA_SEN**.
- (4) En la tabla **VOTA_SEN** id_dig puede repetirse como mucho 3 veces (un mismo ID_DIG puede votar sólo 3 veces).
- (5) NOTA: En **HORARIO** las horas se representan en formato 'hhmmss' y esto hace que para representar la restricción de 2 horas de diferencia como máximo el valor a comparar sea 20000 (o lo que es lo mismo 2h 00 min y 00s).
- (6) **Pérdida semántica** → En principio no se produce pérdida semántica porque no se cumple ninguno de los tres supuestos en que ésta se da: no hay ninguna relación ternaria con alguna participación con mínima 1, no hay binarias con alguna participación (1,n), y no hay especializaciones distintas de parcial y solapada (de hecho, la que hay es parcial y solapada). Puesto que no existe pérdida semántica, no son necesarias restricciones de integridad.
- (7) NOTA: Tanto circunscripción como num_votos se habrían eliminado en el paso a 3FN (para circunscripción se habría creado una tabla LOCALIDAD(nom_localidad, circunscripción), y num_votos al tratarse de un derivado, se habría eliminado porque puede calcularse) pero ambos se han mantenido por conveniencia y acorde a los enunciados.

[2 PUNTOS] EJERCICIO 2: MODELADO FÍSICO DDL (CREACIÓN DE METADATOS)

Indica las sentencias necesarias para incorporar las tablas del ejercicio anterior en MySQL.
Recuerda tener cuidado con el orden de las instrucciones.

```
CREATE TABLE votante(  
dni VARCHAR(9) PRIMARY KEY,  
nom_comp VARCHAR(90),  
telefono VARCHAR(9)  
);
```

```
CREATE TABLE horario(  
h_ini TIME,  
h_fin TIME,  
dia DATE,  
CONSTRAINT hor_dia_ck CHECK (dia >= 20250303 AND dia <= 20250318),  
CONSTRAINT hor_ifd_pk PRIMARY KEY (h_ini, h_fin, dia),  
CONSTRAINT h_inifin_ck CHECK  
(h_ini < h_fin AND h_fin - h_ini <= 20000)  
);
```

```
CREATE TABLE centro(  
id_centro VARCHAR(10) PRIMARY KEY,  
nom_centro VARCHAR(30),  
dir_centro VARCHAR(50),  
CONSTRAINT cen_nomdir_uk UNIQUE (nom_centro, dir_centro)  
);
```

```
CREATE TABLE id_digital(  
id VARCHAR(10) PRIMARY KEY,  
password VARCHAR(16),  
dir_tipo VARCHAR(20),  
direccion VARCHAR(30),  
dir_piso INTEGER,  
dir_num INTEGER,  
localidad VARCHAR(30),  
circunscripcion VARCHAR(30),  
dni VARCHAR(9) NOT NULL,  
CONSTRAINT iddi_dni_fk FOREIGN KEY (dni)  
REFERENCES votante (dni),  
CONSTRAINT iddi_dni_uk UNIQUE(dni)  
);
```

```
CREATE TABLE senador(  
dni_sen VARCHAR(9) PRIMARY KEY,  
circ_presenta VARCHAR(30) NOT NULL,  
partido VARCHAR(4),  
num_votos INTEGER,  
CONSTRAINT sen_dni_fk FOREIGN KEY (dni_sen)  
REFERENCES votante (dni),  
CONSTRAINT sen_par_ck CHECK ( partido IN ('ASIR', 'DAM', 'DAW', 'PHT', 'PRO') )  
);
```

```
CREATE TABLE vota_sen(  
id_dig VARCHAR(10),  
dni_sen VARCHAR(9),  
CONSTRAINT vose_iddni_pk PRIMARY KEY (id_dig, dni_sen),  
CONSTRAINT vose_idig_fk FOREIGN KEY (id_dig)  
REFERENCES id_digital (id),  
CONSTRAINT vose_dnis_fk FOREIGN KEY (dni_sen)  
REFERENCES senador (dni_sen)  
);
```

```
CREATE TABLE interventor(  
dni_int VARCHAR(9) PRIMARY KEY,  
id_centro VARCHAR(10) NOT NULL,  
sueldo INTEGER NOT NULL,  
int_supervisado VARCHAR(9),  
CONSTRAINT int_dni_fk FOREIGN KEY (dni_int)  
REFERENCES votante (dni),  
CONSTRAINT int_icen_fk FOREIGN KEY (id_centro)  
REFERENCES centro (id_centro) ON DELETE CASCADE,  
CONSTRAINT int_isup_fk FOREIGN KEY (int_supervisado)  
REFERENCES interventor (dni_int)  
);
```

```
CREATE TABLE accion(  
num_reg INTEGER,  
descripcion VARCHAR(200),  
dni_int VARCHAR(9),  
CONSTRAINT acc_numdni_pk PRIMARY KEY (num_reg, dni_int),  
CONSTRAINT acc_dint_fk FOREIGN KEY (dni_int)  
REFERENCES interventor (dni_int)  
);
```

```
CREATE TABLE pide_cita(  
dni_votante VARCHAR(9),  
h_ini TIME,  
h_fin TIME,  
dia DATE,  
id_centro VARCHAR(10),  
CONSTRAINT pcit_dvdihf_pk PRIMARY KEY (dni_votante, h_ini, h_fin, dia, id_centro),  
CONSTRAINT pcit_dni_fk FOREIGN KEY (dni_votante)  
REFERENCES votante (dni),  
CONSTRAINT pcit_dhif_fk FOREIGN KEY (h_ini, h_fin, dia)  
REFERENCES horario (h_ini, h_fin, dia),  
CONSTRAINT pcit_icen_fk FOREIGN KEY (id_centro)  
REFERENCES centro (id_centro)  
);
```

Restricciones:

(1) num_votos → num_votos se calcula contando las veces que dni_sen aparece en la tabla **VOTA_SEN** (los votos que ha recibido dicho Senador). num_votos no se elimina porque se considera imprescindible para llevar la cuenta fácilmente, pero se traslada al Senador como atributo porque no depende de la **ID_DIG** que lo vote.

(2) Una localidad sólo puede aparecer junto con una circunscripción. Pero una circunscripción puede aparecer con distintas localidades. (Es decir, una localidad pertenece sólo a una circunscripción).

(3) La ID_DIGITAL.circunscripción y la SENADOR.circ_presenta deben ser la misma para que ID_DIGITAL.id y SENADOR.dni_sen aparezcan en **VOTA_SEN**.

(4) En la tabla **VOTA_SEN** id_dig puede repetirse como mucho 3 veces (un mismo ID_DIGITAL puede votar sólo 3 veces).

(5) NOTA: En **HORARIO** las horas se representan en formato 'hhmmss' y esto hace que para representar la restricción de 2 horas de diferencia como máximo el valor a comparar sea 20000 (o lo que es lo mismo 2h 00 min y 00s).

[1 PUNTO] EJERCICIO 3: MODELADO FÍSICO DDL (MODIFICACIÓN DE METADATOS)

Indica las sentencias necesarias para realizar estas modificaciones en MySQL. Recuerda tener cuidado con el orden de las instrucciones.

- Operación 1 (Eliminar la relación Dispone y cualquier referencia al votante en ID_DIGITAL a excepción de la circunscripción.)

```
// Para eliminar la relación "dispone", debemos eliminar la columna dni
// (y por tanto, la restricción de dni como Clave Ajena en ID_DIGITAL a dni en VOTANTE.
// Primero eliminamos la restricción de Clave Ajena dni → Votante {dni}.
// Después eliminamos la columna dni.
ALTER TABLE id_digital DROP CONSTRAINT iddi_dni_fk;
ALTER TABLE id_digital DROP dni;
```

- Operación 2 (Ahora todos los INTERVENTORES son supervisados, la participación pasa de (0,1) a (1,1).)

```
// Añadimos la restricción de VNN a la columna int_supervisado en la tabla INTERVENTOR.
// Lo hacemos modificando la definición de la columna (ahora contiene VARCHAR(9) pero sin nulos),
// ya que en MySQL VNN no se considera una restricción.
ALTER TABLE interventor MODIFY COLUMN int_supervisado VARCHAR(9) NOT NULL;
```

- Operación 3 (Si se elimina una ID_DIGITAL se borran sus votos también.)

```
// Añadimos la opción B:C (borrado en cascada) a id_dig en VOTA_SEN.
// Para ello primero eliminamos la restricción de Clave Ajena de id_dig en VOTA_SEN
// y la creamos de nuevo con el borrado en cascada:
ALTER TABLE vota_sen DROP CONSTRAINT vose_idig_fk;
ALTER TABLE vota_sen ADD CONSTRAINT vose_idig_fk FOREIGN KEY (id_dig) REFERENCES id_digital(id) ON DELETE CASCADE;
```

- Operación 4 (El sueldo en INTERVENTOR tiene un valor por defecto de 5000.)

```
// Añadimos el valor por defecto modificando la definición de la columna sueldo en INTERVENTOR
// Ahora la columna contiene datos de tipo INTEGER con valor por defecto 5000.
ALTER TABLE interventor MODIFY COLUMN sueldo INTEGER DEFAULT 5000;
```

- Operación 5 (El nombre_centro y direccion dejan de ser únicos en conjunto, ahora solo la direccion del centro es única.)

```
// Debemos cambiar la restricción de UNIQUE de ambas columnas a únicamente la columna direccion,
// Esto se hace eliminando dicha restricción y creándola de nuevo para direccion únicamente
ALTER TABLE centro DROP CONSTRAINT cen_nomdir_uk;
ALTER TABLE centro ADD CONSTRAINT cen_dir_uk UNIQUE (dir_centro);
```

[2 PUNTOS] EJERCICIO 4: MODELADO FÍSICO DML (MANIPULACIÓN DE DATOS)

Indica las sentencias necesarias para realizar estas modificaciones en MySQL. Recuerda tener cuidado con el orden de las instrucciones.

- Operación 1 (Crea una ID_DIGITAL con los datos que quieras en la circunscripción "C.Valenciana" que haya votado al SENADOR del "PHT".)

```
// Se inserta una ID_DIGITAL con los valores siguientes teniendo en cuenta que circunscripcion sea 'C.Valenciana'.  
INSERT INTO id_digital (id, password, dir_tipo, direccion, dir_piso, dir_num, localidad, circunscripcion)  
VALUES('123456789', 'asdfgh', 'calle', 'villaroble', '3', '42', 'valencia', 'C.Valenciana');
```

```
// Antes de esta instrucción DML deberán crearse los senadores del siguiente paso, pero como es respuesta a este  
// enunciado, lo coloco aquí por conveniencia (pero en rojo porque no funcionará y dará un error).
```

```
INSERT INTO vota_sen(id_dig,dni_sen)  
VALUES ('123456789','22222222B');
```

- Operación 2 (Crea 2 SENADORES con los siguientes datos:
dni "11111111A", partido "PRO" circ_presenta "C.Valenciana".
dni "22222222B", partido "PHT" circ_presenta "C.Valenciana".)

```
// Para poder insertar cada SENADOR, primero debe existir como VOTANTE, puesto que la CAj dni_sen apunta a su dni  
// como VOTANTE.
```

```
INSERT INTO votante(dni,nom_comp,telefono)  
VALUES ('11111111A','Amancio','123456789');
```

```
INSERT INTO votante(dni,nom_comp,telefono)  
VALUES ('22222222B','Odiancio','987654321');
```

```
// Como segundo paso insertamos cada SENADOR con el mismo dni que el VOTANTE que es  
// Este es Amancio
```

```
INSERT INTO senador(dni_sen,circ_presenta,partido)  
VALUES ('11111111A','C.Valenciana','PRO');
```

```
//Este es Odiancio
```

```
INSERT INTO senador(dni_sen,circ_presenta,partido)  
VALUES ('22222222B','C.Valenciana','PHT');
```

```
// Una vez insertado el senador del PHT, sí se podría hacer el INSERT en VOTA_SEN del primer enunciado.  
// Lo vuelvo a colocar aquí que es donde iría realmente (en verde porque sí funcionará)
```

```
INSERT INTO vota_sen(id_dig,dni_sen)  
VALUES ('123456789','22222222B');
```

- Operación 3 (Crea 1 INTERVENTOR con los siguientes datos:
dni "12345678C", nom_comp "Juan Carlos Gurtelero Eresísimos", sueldo por defecto, supervisado por él mismo.
Pidió cita para votar el día 13 de marzo entre las 15:00 y las 17:30.
Registra una ACCION con descripcion "He votado por los jubilados de laresidencia La Viña".)

```
// Para poder insertar un INTERVENTOR debemos crear primero el VOTANTE que es y el CENTRO al que está asociado,  
// ya que tiene debilidad de existencia respecto a dicho CENTRO (y por tanto un B:C y un VNN en su CAj).
```

```
PARA ALMACENAR EL HORARIO HACE FALTA ALTER TABLE O CAMBIAR EL HORARIO
```

```
// Insertamos a Juan Carlos como VOTANTE:
```

```
INSERT INTO votante(dni, nom_comp, telefono)
```

```
VALUES ('12345678C', 'Juan Carlos Gurtelero Eresísimos', '654654654');
```

```
// Nos inventamos un CENTRO al que asociarlo:
```

```
INSERT INTO centro(id_centro, nom_centro, dir_centro)
```

```
VALUES ('GAIG42', 'Del Universo', 'C/Misco 33');
```

```
// Insertamos a Juan Carlos como INTERVENTOR con el mismo dni que el VOTANTE que es.
```

```
// Como se supervisa a sí mismo, no creamos otro INTERVENTOR y en int_supervisado ponemos su dni
```

```
INSERT INTO interventor(dni_int, id_centro, sueldo, int_supervisado)
```

```
VALUES ('12345678C', 'GAIG42', DEFAULT, '12345678C');
```

```
// Para la cita hay que tener en cuenta la restricción que introdujimos con respecto a la duración:
```

```
// La diferencia entre h_fin y h_ini es como mucho de 2 horas, restricción que no se cumple en esta cita.
```

```
// Para solventarlo tenemos tres opciones:
```

```
// opción 1 cambiar la restricción para que sí se acepte la cita (por ejemplo, 3 horas en lugar de 2)
```

```
// Primero borramos la restricción y la creamos de nuevo para que sea 3 horas
```

```
ALTER TABLE horario DROP CONSTRAINT h_inifin_ck;
```

```
ALTER TABLE horario ADD CONSTRAINT h_inifin_ck CHECK (h_ini < h_fin AND h_fin - h_ini <= 30000);
```

```
//Después de esto, podríamos introducir sin problemas el HORARIO y la PIDE_CITA (cita) mencionada:
```

```
// Insertamos el HORARIO
```

```
INSERT INTO horario(h_ini, h_fin, dia)
```

```
VALUES ('150000', '173000', '20250313');
```

```
// Insertamos la cita (PIDE_CITA)
```

```
INSERT INTO pide_cita(dni_votante, h_ini, h_fin, dia, id_centro)
```

```
VALUES ('12345678C', '150000', '173000', '20250313', 'GAIG42');
```

```
// Puesto que en esta sección se trata de utilizar sentencias DML, no elegiremos esta opción, vamos con las otras 2.
```

```
// opción 2 acotar la cita a la restricción que tenemos:
```

```
// Insertamos el HORARIO, cambiando la hora de fin para adaptarla a la restricción: 17:00 (justo 2 horas).
```

```
INSERT INTO horario(h_ini, h_fin, dia)
```

```
VALUES ('150000', '170000', '20250313');
```

```
// Insertamos la cita (PIDE_CITA) con dicho HORARIO.
```

```
INSERT INTO pide_cita(dni_votante, h_ini, h_fin, dia, id_centro)
```

```
VALUES ('12345678C', '150000', '170000', '20250313', 'GAIG42');
```

```
// Puesto que tampoco es lo que se pide, no elegiremos esta opción tampoco.
```

```
// opción 3 insertar DOS citas(PIDE_CITA) y que nuestro amigo Juan Carlos vote en la que quiera.
```

```
// Para ello creamos dos HORARIOS y dos citas(PIDE_CITA):
```

```
// Un HORARIO irá de 15:00 a 17:00 para cumplir las 2 horas y el otro con la media hora restante.
```

```
INSERT INTO horario(h_ini, h_fin, dia)
```

```
VALUES ('150000', '170000', '20250313');
```

```
INSERT INTO horario(h_ini, h_fin, dia)
```

```
VALUES ('170000', '173000', '20250313');
```

```
// Ahora insertamos las citas (PIDE_CITA) con los dos HORARIOS.
```

```
INSERT INTO pide_cita(dni_votante, h_ini, h_fin, dia, id_centro)
```

```
VALUES ('12345678C', '150000', '170000', '20250313', 'GAIG42');
```

```
INSERT INTO pide_cita(dni_votante, h_ini, h_fin, dia, id_centro)
```

```
VALUES ('12345678C', '170000', '173000', '20250313', 'GAIG42');
```


APELLIDOS, NOMBRE / COGNOMS, NOM

Sanchis Martínez, Roldán

Gracias por empezar cada ejercicio en una nueva cara

// Finalmente registramos la **ACCION** del **INTERVENTOR**:

```
INSERT INTO accion(num_reg, descripcion, dni_int)
```

```
VALUES (1,'He votado por los jubilados de la residencia La Viña','12345678C');
```

- Operación 4 (Actualiza la descripción de la ACCION registrada a "Colaboración con la residencia La Viña, para ayudar en el proceso de voto de sus residentes".)

// Únicamente necesitamos actualizar el contenido de la ACCION con num_reg = 1 (la que introducimos antes)

// Realizamos un UPDATE de dicha ACCION (num_reg = 1):

```
UPDATE accion
```

```
SET descripcion = 'Colaboración con la residencia La Viña, para ayudar en el proceso de voto de sus residentes'
```

```
WHERE num_reg = 1;
```

- Operación 5 (Sube el sueldo de todos los INTERVENTORES un 25%.)

// Para incrementar un 25% (multiplicar por 1.25) basta con hacer un UPDATE del sueldo a todos (sin WHERE):

```
UPDATE interventor
```

```
SET sueldo = sueldo * 1.25;
```

- Operación 6 (Borra la ID_DIGITAL que has creado)

// Dado que anteriormente añadimos un B:C en la CAj id_dig en **VOTA_SEN**, no tendremos ningún problema al

// intentar borrar la **ID_DIGITAL** (id = 123456789), únicamente nos quedaremos sin su voto.

```
DELETE FROM id_digital WHERE id = 123456789;
```