

[5 PUNTOS] EJERCICIO 1: MODELADO LÓGICO RELACIONAL**ANTES DE NORMALIZAR****CENTRO** (id_centro, nombre_centro, direccion)

CP: id_centro

UK: {nombre_centro, direccion}

HORARIO (hora_inicio, hora_fin, dia)

CP: {hora_inicio, hora_fin, dia}

Pérdida semántica → dia >='2025-03-03' y dia <='2025-03-18'

Pérdida semántica → hora_fin - hora_inicio <= (+)2 horas

ID_DIGITAL (id, password, dir_tipo, dir_nombre, dir_piso, dir_numero, dir_localidad, dir_circunscripcion)

CP: id

Pérdida semántica → No hay ID_DIGITAles que NO pertenezcan a un votante

(Alternativa: Incluir el campo (redundante) único y no nulo dni* → VOTANTE {dni})**VOTANTE** (dni, nom_comp, telefono, id_dig*)

CP: dni

CAj: id_dig → ID_DIGITAL {id}

UK: id_dig

VNN: id_dig

PIDE_C (dni_votante, hora_inicio, hora_fin, dia, id_centro)

CP: {dni_votante, hora_inicio, hora_fin, dia, id_centro}

CAj: dni_votante → VOTANTE {dni}

CAj: {hora_inicio, hora_fin, dia} → HORARIO {hora_inicio, hora_fin, dia}

CAj: id_centro → CENTRO {id_centro}

SENADOR (dni, circ_presenta*, partido)

CP: dni

CAj: dni → VOTANTE {dni}

VNN: circ_presenta

Pérdida semántica → Valores para partido: 'ASIR', 'DAM', 'DAW', 'PHT', 'PRO'

VOTA_SEN (id_voto, dni_senador, num_votos)

CP: {id_voto, dni_senador}

CAj: id_voto → ID_DIGITAL {id}

CAj: dni_senador → SENADOR {dni}

Pérdida semántica → La circ_presenta del SENADOR y la dir_circunscripcion del la ID_DIGITAL deben coincidir.

Pérdida semántica → Máximo de votos por id_voto=3.

Pérdida semántica → num_votos almacena la cantidad de veces que un dni_senador aparece en esta tabla.

INTERVENTOR (dni, sueldo* centro_asignado*, supervisor)

CP: dni

CAj: dni → VOTANTE {dni}

CAj: centro_asignado → CENTRO {id_centro} A:C B:C

VNN: centro_asignado

CAj: supervisor → INTERVENTOR {dni}

VNN: sueldo

ACCION (dni_int, num_reg, descripcion)

CP: {dni_int, num_reg}

CAj: dni_int → INTERVENTOR {dni} A:C B:C

NORMALIZACIÓN 1FN

Se ha pedido expresamente que se mantenga num_votos (VOTA_SEN), aunque se puede sacar con consultas a la tabla vota_sen NO SE ELIMINA.

NORMALIZACIÓN 2FN

En VOTA_SEN el num_votos depende del senador solo, no de la clave completa, se mueve a la tabla de senadores.

NORMALIZACIÓN 3FN

Quien haya puesto el dni del votante en la ID_DIGITAL puede ver esto más fácilmente, pero los datos de la dirección se corresponden mejor a la tabla de VOTANTE que a la ID_DIGITAL, sobretodo en el momento que queremos que esta sea anónima. Se mueven a esa tabla.

La circunscripción depende transitivamente de la localidad, ya que cada localidad pertenece a una circunscripción concreta. Se crea una tabla de localidades por circunscripción (LOCAL_CIRC) y se eliminan éstas de la dirección.

No obstante, puesto que el diseño necesita conservar la circunscripción de la ID_DIGITAL para controlar los SENADORES que se pueden votar, se debería mantener allí la circunscripción de forma redundante. Este campo copiaría su valor de la tabla LOCAL_CIRC en el momento de crear el registro del votante (pérdida semántica) y no se volvería a modificar/borrar. Esto es necesario para asegurar la integridad de los datos. Para asegurar que las ID_DIGITAL adquieren una circunscripción de este modo le añadimos a dir_localidad(VOTANTE) una restricción VNN.

RESULTADO FINAL

CENTRO (id_centro, nombre_centro, direccion)

CP: id_centro

UK: {nombre_centro, direccion}

HORARIO (hora_inicio, hora_fin, dia)

CP: {hora_inicio, hora_fin, dia}

Pérdida semántica → dia >='2025-03-03' y dia <='2025-03-18'

Pérdida semántica → hora_fin - hora_inicio <= (+)2 horas

ID_DIGITAL (id, password, circunscripcion)

CP: id

Pérdida semántica → No hay ID_DIGITALES que NO pertenezcan a un votante

(Alternativa: Incluir el campo (redundante) único y no nulo dni* → VOTANTE {dni})

Pérdida semántica → La circunscripción NO se puede actualizar/borrar.

LOCAL_CIRC (localidad, circunscripcion*)

CP: localidad

VNN: circunscripcion

VOTANTE (dni, nom_comp, dir_tipo, dir_nombre, dir_piso, dir_numero, dir_localidad*, telefono, id_dig*)

CP: dni

CAj: dir_localidad → LOCAL_CIRC {localidad}

CAj: id_dig → ID_DIGITAL {id}

UK: id_dig

VNN: id_dig

VNN: dir_localidad

Pérdida semántica → Se copia en circunscripcion (ID_DIGITAL) el valor de la circunscripcion (LOCAL_CIRC) correspondiente a la dir_localidad del VOTANTE al introducirlo.

PIDE_C (dni_votante, hora_inicio, hora_fin, dia, id_centro)

CP: {dni_votante, hora_inicio, hora_fin, dia, id_centro}

CAj: dni_votante → VOTANTE {dni}

CAj: {hora_inicio, hora_fin, dia} → HORARIO {hora_inicio, hora_fin, dia}

CAj: id_centro → CENTRO {id_centro}

SENADOR (dni, circ_presenta*, partido, num_votos)

CP: dni

CAj: dni → VOTANTE {dni}

VNN: circ_presenta

Pérdida semántica → num_votos almacena la cantidad de veces que dni_senador aparece en la tabla VOTA_SEN.

Pérdida semántica → Valores para partido: 'ASIR', 'DAM', 'DAW', 'PHT', 'PRO'

VOTA_SEN (id_voto, dni_senador)

CP: {id_voto, dni_senador}

CAj: id_voto → ID_DIGITAL {id}

CAj: dni_senador → SENADOR {dni}

Pérdida semántica → La circ_presenta del SENADOR y la circunscripcion del la ID_DIGITAL deben coincidir.

Pérdida semántica → Máximo de votos por id_voto=3.

INTERVENTOR (dni, sueldo*, centro_asignado*, supervisor)

CP: dni

CAj: dni → VOTANTE {dni}

CAj: centro_asignado → CENTRO {id_centro} A:C B:C

VNN: centro_asignado

CAj: supervisor → INTERVENTOR {dni}

VNN: sueldo

ACCION (dni_int, num_reg, descripcion)

CP: {dni_int, num_reg}

CAj: dni_int → INTERVENTOR {dni} A:C B:C

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

```
CREATE DATABASE IF NOT EXISTS db_elec_corrupoly
CHARACTER SET utf8mb4 COLLATE utf8mb4_es_0900_as_cs;
use db_elec_corrupoly
```

```
CREATE TABLE IF NOT EXISTS centro (
id_centro VARCHAR(5) PRIMARY KEY,
nombre_centro VARCHAR(60),
direccion VARCHAR(100),
CONSTRAINT cen_nomdir_uk UNIQUE(nombre_centro,direccion)
);
```

```
CREATE TABLE IF NOT EXISTS horario (
dia DATE,
hora_inicio TIME,
hora_fin TIME,
CONSTRAINT hor_diafin_pk PRIMARY KEY(dia,hora_inicio,hora_fin),
CONSTRAINT hor_dia_ck CHECK(dia >='2025-03-03' AND dia <='2025-03-18'),
CONSTRAINT hor_inifin_ck CHECK(TIMEDIFF(hora_fin,hora_inicio) <= TIME(20000) AND
TIMEDIFF(hora_fin,hora_inicio) > TIME(0))
);
```

```
CREATE TABLE IF NOT EXISTS id_digital (
id VARCHAR(10) PRIMARY KEY,
password CHAR(60),
circunscripcion VARCHAR(25)
);
```

Restricciones:

(1) Pérdida semántica → No hay ID_DIGITAles que NO pertenezcan a un votante
(Alternativa: Incluir el campo (redundante) único y no nulo dni* → VOTANTE {dni})

(2) Pérdida semántica → La circunscripción NO se puede actualizar/borrar.

```
CREATE TABLE IF NOT EXISTS local_circ (
localidad VARCHAR(50) PRIMARY KEY,
circunscripcion VARCHAR(25) NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS votante (
dni VARCHAR(10) PRIMARY KEY,
nom_comp VARCHAR(200),
dir_tipo VARCHAR(4),
dir_nombre VARCHAR(60),
dir_piso TINYINT,
dir_numero SMALLINT,
dir_localidad VARCHAR(50) NOT NULL,
telefono VARCHAR(12),
id_dig VARCHAR(10) NOT NULL,
CONSTRAINT vot_dig_fk FOREIGN KEY (id_dig) REFERENCES id_digital(id),
CONSTRAINT vot_loc_fk FOREIGN KEY (dir_localidad) REFERENCES local_circ(localidad),
CONSTRAINT vot_dig_uk UNIQUE(id_dig)
);
```

Restricciones:

(1) Pérdida semántica → Se copia en circunscripcion (id_digital) el valor de la circunscripcion (local_circ) correspondiente a la dir_localidad(votante) al introducir cada votante.

```
CREATE TABLE IF NOT EXISTS pide_c (
dni_votante VARCHAR(10),
dia DATE,
hora_inicio TIME,
hora_fin TIME,
id_centro VARCHAR(5),
CONSTRAINT pid_all_pk PRIMARY KEY(dni_votante,dia,hora_inicio,hora_fin,id_centro),
CONSTRAINT pid_vot_fk FOREIGN KEY(dni_votante) REFERENCES votante(dni),
CONSTRAINT pid_hor_fk FOREIGN KEY(dia,hora_inicio,hora_fin)
REFERENCES horario(dia,hora_inicio,hora_fin),
CONSTRAINT pid_cen_fk FOREIGN KEY(id_centro) REFERENCES centro(id_centro)
);
```

```
CREATE TABLE IF NOT EXISTS senador (
dni VARCHAR(10) PRIMARY KEY,
circ_presenta VARCHAR(25) NOT NULL,
partido VARCHAR(4),
num_votos INTEGER,
CONSTRAINT sen_vot_fk FOREIGN KEY(dni) REFERENCES votante(dni),
CONSTRAINT sen_par_ck CHECK(partido IN ('ASIR', 'DAM', 'DAW', 'PHT', 'PRO'))
);
```

Restricciones:

(1) Pérdida semántica → En num_votos se copiará la cantidad de veces que dni_senador aparece en la tabla VOTA_SEN.

(2) Pérdida semántica → circ_presenta debe estar en la tabla circunscripcion(local_circ). Esta pérdida no había sido contemplada hasta ahora.

```
CREATE TABLE IF NOT EXISTS vota_sen (
id_voto VARCHAR(10),
dni_senador VARCHAR(10),
CONSTRAINT vos_digsen_pk PRIMARY KEY(id_voto,dni_senador),
CONSTRAINT vos_dig_fk FOREIGN KEY(id_voto) REFERENCES id_digital(id),
CONSTRAINT vos_sen_fk FOREIGN KEY(dni_senador) REFERENCES senador(dni)
);
```

Restricciones:

(1) Pérdida semántica → circ_presenta(senador) = circunscripcion(id_digital) para aceptar voto.

(2) Pérdida semántica → Máximo de votos por id_voto=3.

```
CREATE TABLE IF NOT EXISTS interventor (
dni VARCHAR(10) PRIMARY KEY,
sueldo DECIMAL(8,2) NOT NULL,
centro_asignado VARCHAR(5) NOT NULL,
supervisor VARCHAR(10),
CONSTRAINT int_vot_fk FOREIGN KEY(dni) REFERENCES votante(dni),
CONSTRAINT int_cen_fk FOREIGN KEY(centro_asignado) REFERENCES centro(id_centro) ON
UPDATE CASCADE ON DELETE CASCADE,
CONSTRAINT int_int_fk FOREIGN KEY(supervisor) REFERENCES interventor(dni)
);
```

```
CREATE TABLE IF NOT EXISTS accion (
dni_int VARCHAR(10),
num_reg TINYINT,
descripcion VARCHAR(256),
CONSTRAINT acc_intnum_pk PRIMARY KEY(dni_int,num_reg),
CONSTRAINT acc_int_fk FOREIGN KEY(dni_int) REFERENCES interventor(dni) ON UPDATE
CASCADE ON DELETE CASCADE
);
```

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

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

// Si se tienen datos de la dirección en ID_DIGITAL se deben usar ALTER.. DROP COLUMN para ELIMINARLOS y ALTER... ADD COLUMN para añadirlos a VOTANTE. También se debe eliminar el DNI del votante en caso de haberlo añadido a la tabla ID_DIGITAL. No es el caso de esta solución que ya ha tenido en cuenta estos detalles, en esta simplemente se elimina id_dig de VOTANTE, para ello se necesita eliminar primero la FOREIGN KEY que tiene el campo.

```
ALTER TABLE votante DROP CONSTRAINT vot_dig_fk;  
ALTER TABLE votante DROP COLUMN id_dig;
```

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

// Para ello, el campo supervisor no puede ser nulo, se modifica.

```
ALTER TABLE interventor MODIFY COLUMN supervisor VARCHAR(10) NOT NULL;
```

- Si se elimina una ID_DIGITAL se borran sus votos también.

// Se modifica la FOREIGN KEY (DROP+ADD) de VOTA_SEN para añadir Borrado en Cascada

```
ALTER TABLE vota_sen DROP CONSTRAINT vos_dig_fk;  
ALTER TABLE vota_sen ADD CONSTRAINT vos_dig_fk FOREIGN KEY(id_voto) REFERENCES  
id_digital(id) ON DELETE CASCADE;
```

- El sueldo en INTERVENTOR tiene un valor por defecto de 5000.

// Se modifica la definición del campo sueldo.

```
ALTER TABLE interventor MODIFY COLUMN sueldo DECIMAL(8,2) NOT NULL DEFAULT 5000.00;
```

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

// Se elimina la restricción y se crea una nueva, se puede hacer también modificando la columna direccion en vez de añadir una restricción.

```
ALTER TABLE centro DROP CONSTRAINT cen_nomdir_uk;  
ALTER TABLE centro ADD CONSTRAINT cen_dir_uk UNIQUE(direccion);
```

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

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

// Hay que crear al senador antes de poder votarlo, así que se debe realizar este subapartado antes que el que está primero en la tarea. Para poder crear un senador hay que insertarlo primero como votante; se pueden inventar sus datos o dejarlos a NULL (salvo dir_localidad). A su vez, para poder crear votantes hay que poner primero su localidad en LOCAL_CIRC. El senador puede ser de cualquier circunscripción, por facilidad, lo haremos de la C.Valenciana, la misma a la que se presentan ambos.

```
INSERT INTO local_circ VALUES ("Valencia","C.Valenciana");
INSERT INTO votante (dni, dir_localidad) VALUES ("11111111A", "Valencia"),("22222222B", "Valencia");
INSERT INTO senador VALUES ("11111111A", "C.Valenciana", "PRO", NULL), ("22222222B", "C.Valenciana", "PHT", NULL);
```

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

// Como ahora las identidades no están relacionadas con votantes se puede hacer sin limitaciones, más allá de la circunscripción, aunque esa limitación no está implementada. Creamos la ID_DIGITAL y el VOTA_SEN

```
INSERT INTO id_digital VALUES ("1234567890",NULL,"C.Valenciana");
INSERT INTO vota_sen VALUES ("1234567890","11111111A");
```

- 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 la residencia La Viña".

// De nuevo primero hay que insertarlo como votante, y esta vez necesitamos inventar un centro para que esté asignado. Se mantiene la localidad para ahorrar crear otra. Para ahorrarnos crear un supervisor, y muy en el espíritu de esta tarea, se supervisará a sí mismo.

```
INSERT INTO votante (dni, nom_comp, dir_localidad) VALUES ("12345678C", "Juan Carlos Gurtelero Eresísimos", "Valencia");
INSERT INTO centro VALUES ("CV001","CEEDCV",NULL);
INSERT INTO interventor VALUES ("12345678C", DEFAULT, "CV001", "12345678C");
INSERT INTO accion VALUES ("12345678C", 1, "He votado por los jubilados de la residencia La Viña");
```

- Actualiza la descripcion de la ACCION registrada a "Colaboración con la residencia La Viña, para ayudar en el proceso de voto de sus residentes".

```
UPDATE accion
SET descripcion="Colaboración con la residencia La Viña, para ayudar en el proceso de voto de sus residentes"
WHERE dni_int="12345678C" AND num_reg=1;
```

EV02.04. EVALUABLE MODELOS RELACIONAL Y FÍSICO. SOLUCIÓN

- Sube el sueldo de todos los INTERVENTORES un 25%

```
UPDATE interventor  
SET sueldo=sueldo*1.25;
```

- Borra la ID_DIGITAL que has creado.

```
DELETE FROM id_digital  
WHERE id="1234567890";
```