

[5 PUNTOS] EJERCICIO 1: MODELADO LÓGICO RELACIONAL

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

Te mostramos un ejemplo de notación de modelado lógico que deberás cambiar por la de tu diagrama E-R:

Modelo lógico relacional (paso de E-R a tablas)

COCHE OFICIAL (matrícula, marca, modelo)

CP: matrícula

POLÍTICO (DNI, sueldo*, correo_elec, {teléfono}ⁿ, dir_comp.dir_postal, dir_comp.cod_postal, dir_comp.localidad, dir_comp.provincia, nom_comp.nombre, nom_comp.apellidos, matrícula)

CP: DNI

CAj: matrícula → Coche oficial (matrícula)

VNN: matrícula

VNN: sueldo

POLÍTICO-CARGO (DNI, nombre, comisión, complemento)

CP: DNI

CAj: DNI → Político (DNI)

POLÍTICO-CONCEJAL (DNI, oposición_sn, comp_sillón)

CP: DNI

CAj: DNI → Político (DNI)

ASESOR (DNI, nombre, estudios, teléfono, DNI_pol, dni_asesor_supervisor)

CP: DNI

CAj: DNI_pol → Político (DNI)

CAj: dni_asesor_supervisor → Asesor (DNI)

VNN: DNI_pol

REUNIÓN (Código, descripción, f_celebración*, asistentes)

Asistentes = número de políticos que participan en la reunión. Se cuentan los registros del DNI de cada político en cada reunión.

CP: código

VNN: f_celebración

DIETA (concepto, descripción)

CP: concepto

COBRAR (DNI_pol, código_reu, concepto_dieta)

CP: {DNI_pol, código_reu, concepto_dieta}

CAj: DNI_pol → Político (DNI)

CAj: código_reu → Reunión (código)

CAj: concepto_dieta → Dieta (concepto)

DESGLOSE (concepto, línea, cantidad, importe)

CP: {concepto, línea}

CAj: concepto → Dieta (concepto)

Restricciones:

(1) Pérdida semántica → Comprobar que todo **coche oficial** tiene asociado un **político**.

(2) Pérdida semántica → Comprobar que todo **político** debe ser sólo o **cargo** o **concejal**, nada más, pudiendo ser ambos a la vez (especialización total y solapada).

(3) Pérdida semántica → Comprobar que todo **político** tiene un asesor **asociado**.

(4) Atributo derivado → el atributo "**asistentes**" de la tabla **REUNIÓN** es derivado. Se eliminará este atributo, ya que su valor se puede obtener contando el número de DNIs de políticos que aparecen en la tabla **COBRAR** para una reunión dada.

(5) Pérdida semántica → En la relación **Cobrar**, comprobar que todo **político** tiene asociada una **reunión** y una **dieta** para cada una de sus participaciones. Hacer la misma comprobación con Dieta y Reunión, para comprobar que las 3 entidades están asociadas en cada participación que aparecen en la relación.

(6) Restricción de identidad → **Desglose** es una entidad débil, con restricción de identidad con respecto a **Dieta**. Desglose necesita la CP de dieta, junto con su propia CP para ser debidamente identificada. Incluye la restricción de existencia.

Modelo lógico relacional (3FN)

Se eliminan atributos derivados, multivaluados y compuestos, y se eliminan dependencias funcionales y transitivas. El modelo obtenido en 3FN a partir de las tablas anteriores es el siguiente:

COCHE OFICIAL (matrícula, modelo)

CP: matrícula

CAj: modelo → Marca (modelo)

MARCA (modelo, marca)

CP: modelo

POLÍTICO (DNI, sueldo*, correo_elec, dir_postal, localidad, nombre, apellidos, matrícula)

CP: DNI

CAj: localidad → Provincia (localidad)

CAj: matrícula → Coche oficial (matrícula)

CAj: dir_postal → Dirección (dir_postal)

VNN: matrícula

VNN: sueldo

PROVINCIA (localidad, provincia)

CP: localidad

DIRECCIÓN (dir_postal, cod_postal)

CP: dir_postal

TELÉFONO (DNI, teléfono)

CP: {DNI, teléfono}

CAj: DNI → Político (DNI)

POLÍTICO-CARGO (DNI, nombre, comisión, complemento)

CP: DNI

CAj: DNI → Político (DNI)

POLÍTICO-CONCEJAL (DNI, oposición_sn, comp_sillón)

CP: DNI

CAj: DNI → Político (DNI)

ASESOR (DNI, nombre, estudios, teléfono, DNI_pol, dni_asesor_supervisor)

CP: DNI

CAj: DNI_pol → Político (DNI)

CAj: dni_asesor_supervisor → Asesor (DNI)

VNN: DNI_pol

López Morató, Marta

73590823T

Gracias por empezar cada ejercicio en una nueva cara

DIETA (concepto, descripción)

CP: concepto

REUNIÓN (Código, descripción, f_celebración*)

CP: código

VNN: f_celebración

COBRAR (DNI_pol, código_reu, concepto_dieta)

CP: {DNI_pol, código_reu, concepto_dieta}

CAj: DNI_pol → Político (DNI)

CAj: código_reu → Reunión (código)

CAj: concepto_dieta → Dieta (concepto)

DESGLOSE (concepto, línea, cantidad, importe)

CP: {concepto, línea}

CAj: concepto → Dieta (concepto)

Restricciones:

- (1) Pérdida semántica → Comprobar que todo **coche oficial** tiene asociado un **político**.
- (2) Pérdida semántica → Comprobar que todo **político** debe ser sólo o **cargo** o **concejal**, nada más, pudiendo ser ambos a la vez (especialización total y solapada).
- (3) Pérdida semántica → Comprobar que todo **político** tiene un asesor **asociado**.
- (4) Pérdida semántica → En la relación **Cobrar**, comprobar que todo **político** tiene asociada una **reunión** y una **dieta** para cada una de sus participaciones. Hacer la misma comprobación con Dieta y Reunión, para comprobar que las 3 entidades están asociadas en cada participación que aparecen en la relación.
- (5) Restricción de identidad → **Desglose** es una entidad débil, con restricción de identidad con respecto a **Dieta**. Desglose necesita la CP de dieta, junto con su propia CP para ser debidamente identificada. Incluye la restricción de existencia.

[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 marca (  
  modelo VARCHAR(30),  
  marca VARCHAR(20),  
  CONSTRAINT mar_mod_pk PRIMARY KEY (modelo)  
);
```

```
CREATE TABLE provincia (  
  localidad VARCHAR(30),  
  provincia VARCHAR(20),  
  CONSTRAINT prov_loc_pk PRIMARY KEY (localidad)  
);
```

```
CREATE TABLE direccion (  
  dir_postal VARCHAR(60),  
  cod_postal VARCHAR(6),  
  CONSTRAINT dir_dir_pk PRIMARY KEY (dir_postal)  
);
```

```
CREATE TABLE dieta (  
  concepto VARCHAR(20),  
  descripcion VARCHAR(80),  
  CONSTRAINT die_con_pk PRIMARY KEY (concepto)  
);
```

```
CREATE TABLE reunion (  
  codigo VARCHAR(6),  
  descripcion VARCHAR(60),  
  f_celebracion DATE NOT NULL,  
  CONSTRAINT reu_cod_pk PRIMARY KEY (codigo)  
);
```

```
CREATE TABLE desglose (  
  concepto VARCHAR(20),  
  linea INTEGER,  
  cantidad INTEGER,  
  importe FLOAT,  
  CONSTRAINT des_conlin_pk PRIMARY KEY (concepto, linea),  
  CONSTRAINT des_con_fk FOREIGN KEY (concepto) REFERENCES dieta (concepto)  
);
```

```
CREATE TABLE coche_oficial (  
  matricula CHAR(7),  
  modelo VARCHAR(30),  
  CONSTRAINT coc_mat_pk PRIMARY KEY (matricula),  
  CONSTRAINT coc_mod_fk FOREIGN KEY (modelo) REFERENCES marca (modelo)  
);
```

López Morató, Marta

73590823T

Gracias por empezar cada ejercicio en una nueva cara

```
CREATE TABLE politico (  
dni CHAR(9),  
nombre VARCHAR(20),  
apellidos VARCHAR(30),  
sueldo FLOAT NOT NULL,  
correo_elec VARCHAR(40),  
dir_postal VARCHAR(60),  
localidad VARCHAR(30),  
matricula CHAR(7) NOT NULL,  
CONSTRAINT pol_dni_pk PRIMARY KEY (dni),  
CONSTRAINT pol_loc_fk FOREIGN KEY (localidad) REFERENCES provincia (localidad),  
CONSTRAINT pol_mat_fk FOREIGN KEY (matricula) REFERENCES coche_oficial (matricula),  
CONSTRAINT pol_dir_fk FOREIGN KEY (dir_postal) REFERENCES direccion (dir_postal)  
);
```

```
CREATE TABLE politico_cargo (  
dni CHAR(9),  
nombre VARCHAR(20),  
comision VARCHAR(5),  
complemento FLOAT,  
CONSTRAINT polcar_dni_pk PRIMARY KEY (dni),  
CONSTRAINT polcar_dni_fk FOREIGN KEY (dni) REFERENCES politico (dni)  
);
```

```
CREATE TABLE politico_concejal (  
dni CHAR(9),  
oposicion_sn CHAR(2),  
comp_sillon FLOAT,  
CONSTRAINT polcon_dni_pk PRIMARY KEY (dni),  
CONSTRAINT polcon_dni_fk FOREIGN KEY (dni) REFERENCES politico (dni)  
);
```

```
CREATE TABLE telefono (  
dni CHAR(9),  
telefono VARCHAR(15),  
CONSTRAINT poltel_dnitel_pk PRIMARY KEY (dni, telefono),  
CONSTRAINT poltel_dni_fk FOREIGN KEY (dni) REFERENCES politico (dni)  
);
```

```
CREATE TABLE cobrar (  
dni_pol CHAR(9),  
codigo_reu VARCHAR(6),  
concepto_dieta VARCHAR(20),  
CONSTRAINT cob_dnicodcon_pk PRIMARY KEY (dni_pol, codigo_reu, concepto_dieta),  
CONSTRAINT cob_dni_fk FOREIGN KEY (dni_pol) REFERENCES politico (dni),  
CONSTRAINT cob_cod_fk FOREIGN KEY (codigo_reu) REFERENCES reunion (codigo),  
CONSTRAINT cob_con_fk FOREIGN KEY (concepto_dieta) REFERENCES dieta (concepto)  
);
```

```
CREATE TABLE asesor (  
dni CHAR(9),  
nombre VARCHAR(20),  
estudios VARCHAR(30),  
telefono VARCHAR(15),  
dni_pol CHAR(9) NOT NULL,  
dni_asesor_supervisor CHAR(9),  
CONSTRAINT ase_dni_pk PRIMARY KEY (dni),  
CONSTRAINT ase_dnipol_fk FOREIGN KEY (dni_pol) REFERENCES politico (dni),
```

```
CONSTRAINT ase_est_ck CHECK (estudios IN ('basico', 'falsificado', 'ninguno')),  
CONSTRAINT ase_dniase_fk FOREIGN KEY (dni_asesor_supervisor) REFERENCES asesor (dni)  
);
```

Restricciones:

- (1) Pérdida semántica → Comprobar que todo **coche oficial** tiene asociado un **político**.
- (2) Pérdida semántica → Comprobar que todo **político** debe ser sólo o **cargo** o **concejal**, nada más, pudiendo ser ambos a la vez (especialización total y solapada).
- (3) Pérdida semántica → Comprobar que todo **político** tiene un asesor **asociado**.
- (4) Pérdida semántica → En la relación **Cobrar**, comprobar que todo **político** tiene asociada una **reunión** y una **dieta** para cada una de sus participaciones. Hacer la misma comprobación con Dieta y Reunión, para comprobar que las 3 entidades están asociadas en cada participación que aparecen en la relación.
- (5) Restricción de identidad → **Desglose** es una entidad débil, con restricción de identidad con respecto a **Dieta**. Desglose necesita la CP de dieta, junto con su propia CP para ser debidamente identificada. Incluye la restricción de existencia.

[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.

- Modificar la tabla POLÍTICO para que los campos “nombre” y “apellidos” sean únicos (en conjunto), es decir, que se pueda repetir el nombre y los apellidos por separado pero no puedan insertarse dos filas con los dos datos idénticos.

// Hay que añadir la restricción UNIQUE en la tabla Político, que englobe ambos campos, el nombre y los apellidos. La sentencia sería:

```
ALTER TABLE politico ADD CONSTRAINT pol_nomape_uk UNIQUE (nombre, apellidos);
```

- Cambiar la participación de la entidad POLÍTICO en la relación DISPONER de (1,1) a (0,1).

// Hay que quitar la restricción de NOT NULL del atributo matrícula. Como MySQL no la considera una restricción, si no una definición de columna, y no tiene un nombre (constraint) asociado para hacer la modificación, borramos el campo y lo volvemos a crear sin la definición de NOT NULL. Las sentencias serían:

```
ALTER TABLE politico DROP matricula;  
ALTER TABLE politico ADD matricula CHAR(7);
```

- Permitir que se pueda modificar el concepto de la tabla DIETA, propagando este cambio a las tablas dependientes de este dato.

// Hay que añadir la cláusula ON UPDATE CASCADE en las tablas donde el concepto de Dieta sea clave ajena. Como las restricciones no se pueden modificar, hay que eliminarlas y volverlas a crear con las nuevas restricciones que queramos. Las sentencias serían:

```
ALTER TABLE desglose DROP CONSTRAINT des_con_fk;  
ALTER TABLE desglose ADD CONSTRAINT des_con_fk FOREIGN KEY (concepto) REFERENCES dieta (concepto) ON  
UPDATE CASCADE;
```

```
ALTER TABLE cobrar DROP CONSTRAINT cob_con_fk;  
ALTER TABLE cobrar ADD CONSTRAINT cob_con_fk FOREIGN KEY (concepto_dieta) REFERENCES dieta (concepto) ON  
UPDATE CASCADE;
```

- Modificar la tabla REUNIÓN para que por defecto la fecha de celebración sea la fecha del sistema.

// Hay que modificar el atributo f_celebración. Los atributos se pueden modificar sin tener que quitarlos y volverlos a crear. La sentencia sería:

```
ALTER TABLE reunion MODIFY f_celebracion TIMESTAMP DEFAULT CURRENT_TIMESTAMP;
```

[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.

- Crea 3 coches oficiales con los siguientes datos:
 - Matrícula "1111AAA", marca "Mercedes".
 - Matrícula "2222AAB", marca "BMW".
 - Matrícula "3333AAC", marca "Lexus". Operación 1 (copiar enunciado)

// Como la tabla Coche_oficial sólo tiene los atributos "matrícula" y "modelo", y el atributo "marca" pertenece a la tabla Marcas, primero hay que crear las 3 marcas en la tabla Marcas, y luego ya crear los 3 coches oficiales. Las sentencias serían:

```
INSERT INTO marca (modelo, marca) VALUES  
(‘Mercedes GLS’, ‘Mercedes’),  
(‘BMW M6’, ‘BMW’),  
(‘LC deportivo’, ‘Lexus’);
```

```
INSERT INTO coche_oficial (matricula, modelo) VALUES  
(‘1111AAA’, ‘Mercedes GLS’),  
(‘2222AAB’, ‘BMW M6’),  
(‘3333AAC’, ‘LC deportivo’);
```

- Crea 3 políticos con los siguientes datos:
 - DNI '11111111A', nombre "Paco", apellidos 'Butxaca', sueldo 8000. Cargo: nombre 'alcalde'.
 - DNI '2222222B', nombre "Pepe", apellidos 'Meloquedo', sueldo 6000. Cargo: nombre 'teniente alcalde'.
 - DNI '3333333C', nombre "Armando", apellidos 'Nopuedo', sueldo 3000. Concejal: de la oposición.

// La tabla Político tiene 3 claves ajenas, matrícula (coche_oficial), dir_postal (direccion) y localidad (provincia). La tabla de Coche_oficial ya contiene los datos necesarios, pero las tablas de Direccion y Provincia están vacías, y si no añadimos datos, no podremos insertar políticos en la tabla Político. Posteriormente, para crear los cargos, hay que insertar datos en las tablas politico_cargo y politico_concejal. Las sentencias serían las siguientes, en el orden en el que aparecen:

```
INSERT INTO direccion (dir_postal, cod_postal) VALUES  
(‘calle mayor’, ‘46290’),  
(‘calle menor’, ‘47280’),  
(‘calle principal’, ‘44100’);
```

```
INSERT INTO provincia (localidad, provincia) VALUES  
(‘alcasser’, ‘valencia’),  
(‘silla’, ‘valencia’),  
(‘sueca’, ‘valencia’);
```

```
INSERT INTO politico (dni, nombre, apellidos, sueldo, correo_elec, dir_postal, localidad, matricula) VALUES  
(‘11111111A’, ‘Paco’, ‘Butxaca’, 8000, ‘pacobutxaca@gmail.com’, ‘calle mayor’, ‘alcasser’, ‘1111AAA’),  
(‘2222222B’, ‘Pepe’, ‘Meloquedo’, 6000, ‘pepemeloquedo@gmail.com’, ‘calle menor’, ‘silla’, ‘2222AAB’),  
(‘3333333C’, ‘Armando’, ‘Nopuedo’, 3000, ‘armandito@gmail.com’, ‘calle principal’, ‘sueca’, ‘3333AAC’);
```


López Morató, Marta

73590823T

Gracias por empezar cada ejercicio en una nueva cara

```
INSERT INTO politico_cargo (dni, nombre, comision, complemento) VALUES  
(‘11111111A’, ‘alcalde’, ‘33%’, 275),  
(‘22222222B’, ‘teniente alcalde’, ‘25%’, 290);
```

```
INSERT INTO politico_concejal (dni, oposicion_sn, comp_sillon) VALUES  
(‘33333333C’, ‘sí’, 2400);
```

- Crea 2 asesores con los siguientes datos:
 - DNI '44444444D', nombre "Eustaquio", estudios 'ninguno'. Asesora al alcalde (político con DNI '11111111A').
 - DNI '55555555E', nombre "Saruman", estudios 'falsificado'. Asesora al teniente de alcalde (político con DNI "22222222B").

// Como no se especifica si los asesores tienen otros asesores supervisores, el campo "dni_asesor_supervisor" se deja como nulo. Estas serían las sentencias:

```
INSERT INTO asesor (dni, nombre, estudios, telefono, dni_pol, dni_asesor_supervisor) VALUES  
(‘44444444D’, ‘Eustaquio’, ‘ninguno’, ‘659129577’, ‘11111111A’, NULL),  
(‘55555555D’, ‘Saruman’, ‘falsificado’, ‘631995468’, ‘22222222B’, NULL);
```

- Crea 3 reuniones con los siguientes datos:
 - Código "COM001", descripción "Comisión para subir impuestos".
 - Código "COM002", descripción "Comisión para asesorar a los asesores".
 - Código "PLE001", descripción "Pleno del Ayuntamiento".

```
INSERT INTO reunion (codigo, descripcion) VALUES  
(‘COM001’, ‘Comisión para subir impuestos’),  
(‘COM002’, ‘Comisión para asesorar a los asesores’),  
(‘PLE001’, ‘Pleno del Ayuntamiento’);
```

Crea 4 dietas con los siguientes datos:

- Concepto "Desplazamiento", descripción "Desplazamiento de la planta baja a la primera planta".
- Concepto "Productividad", descripción "Productividad por creación de problemas y burocracia".
- Concepto "Plus", descripción "Plus por cargo adicional".
- Concepto "Silla", descripción "Calentamiento de la silla".

```
INSERT INTO dieta (concepto, descripcion) VALUES  
(‘Desplazamiento’, ‘Desplazamiento de la planta baja a la primera planta’),  
(‘Productividad’, ‘Productividad por creación de problemas y burocracia’),  
(‘Plus’, ‘Plus por cargo adicional’),  
(‘Silla’, ‘Calentamiento de la silla’);
```

López Morató, Marta

73590823T

Gracias por empezar cada ejercicio en una nueva cara

- El asesor Eustaquio es supervisado por Saruman.

//Hay que actualizar la tabla Asesor para incluir el dni de Saruman en el registro de Eustaquio. Para ello, las sentencias serían:

```
UPDATE asesor
SET dni_asesor_supervisor = '55555555D'
WHERE nombre = 'Eustaquio';
```

- El Ayuntamiento celebra una REUNIÓN (pleno) con fecha 1/12/2023 y: °
 - Asisten los 3 políticos: Paco, Pepe y Armando.
 - Los 3 políticos cobran la dieta por desplazamiento.
 - El alcalde (Paco) genera los siguientes desgloses: línea 1, cantidad 1, importe 100 y línea 2, cantidad 2, importe 75.
 - El teniente de alcalde (Pepe) genera los siguientes desgloses: línea 1, cantidad 5, importe 50.
 - El concejal de la oposición (Armando) genera los siguientes desgloses: línea 1, cantidad 2, importe 30 y línea 2, cantidad 10, importe 5.

//Primero se crea la reunión con la fecha indicada:

```
INSERT INTO reunion (codigo, descripcion, f_celebracion) VALUES
('PLE002', 'Pleno del Ayuntamiento', STR_TO_DATE('2023-12-01', '%Y-%m-%d'));
```

//Luego se crean las dietas para cada político:

```
INSERT INTO dieta (concepto, descripcion) VALUES
('DESP_Paco', 'Desplazamiento de la planta baja a la primera planta'),
('DESP_Pepe', 'Desplazamiento de la segunda planta a la primera planta'),
('DESP_Armando', 'Desplazamiento de la tercera planta a la primera planta');
```

//Seguidamente se insertan los datos en la tabla Cobrar para que se relacione cada político con la reunión celebrada y su correspondiente dieta:

```
INSERT INTO cobrar (dni_pol, codigo_reu, concepto_dieta) VALUES
('11111111A', 'PLE002', 'DESP_Paco'),
('22222222B', 'PLE002', 'DESP_Pepe'),
('33333333C', 'PLE002', 'DESP_Armando');
```

//Finalmente se crean los desgloses para las dietas de cada político en la tabla Desglose:

```
INSERT INTO desglose (concepto, linea, cantidad, importe) VALUES
('DESP_Paco', 1, 1, 100),
('DESP_Paco', 2, 2, 75),
('DESP_Pepe', 1, 5, 50),
('DESP_Armando', 1, 2, 30),
('DESP_Armando', 2, 10, 5);
```

- El Ayuntamiento celebra una REUNIÓN (comisión para subir impuestos) con fecha 2/12/2023 y:
 - Asisten 2 políticos: Paco y Pepe.
 - El alcalde (Paco) cobra la dieta por plus y genera los siguientes desgloses: línea 1, cantidad 1, importe 500.
 - El teniente de alcalde (Pepe) cobra la dieta por plus y genera los siguientes desgloses: línea 1, cantidad 1, importe 300.

//Primero se crea la reunión con la fecha indicada:

```
INSERT INTO reunion (codigo, descripcion, f_celebracion) VALUES
('COM003', 'Comisión para subir impuestos', STR_TO_DATE('2023-12-02', '%Y-%m-%d'));
```

//Luego se crean las dietas para cada político:

```
INSERT INTO dieta (concepto, descripcion) VALUES
('PLUS_Paco', 'Plus por cargo adicional'),
('PLUS_Pepe', 'Plus por cargo adicional');
```

//Seguidamente se insertan los datos en la tabla Cobrar para que se relacione cada político con la reunión celebrada y su correspondiente dieta:

```
INSERT INTO cobrar (dni_pol, codigo_reu, concepto_dieta) VALUES
('11111111A', 'COM003', 'PLUS_Paco'),
('22222222B', 'COM003', 'PLUS_Pepe');
```

//Finalmente se crean los desgloses para las dietas de cada político en la tabla Desglose:

```
INSERT INTO desglose (concepto, linea, cantidad, importe) VALUES
('PLUS_Paco', 1, 1, 500),
('PLUS_Pepe', 1, 1, 300);
```

- Actualiza la fecha de celebración de la reunión del pleno a 5/12/2023.

// Se actualiza la tabla, especificando el registro que tenemos que actualizar (codigo = PLE002), para que sólo se actualice ese registro y no todos los de la tabla.

```
UPDATE reunion
SET f_celebracion = STR_TO_DATE('2023-12-05', '%Y-%m-%d')
WHERE codigo = 'PLE002';
```

- Borra el político Armando

// Tal como se han definido las tablas, no se han usado cláusulas de restricción para el borrado y actualización de datos y registros. Por tanto, si intentamos borrar a Armando directamente, no podremos hacerlo, ya que la CP de político (DNI), es clave ajena en estas tablas: politico_cargo, politico_concejal, telefono, cobrar, asesor. Primero hay que añadir la cláusula ON DELETE CASCADE en todas estas tablas, en la clave ajena DNI de político.

```
ALTER TABLE cobrar DROP CONSTRAINT cob_dni_fk;
ALTER TABLE cobrar ADD CONSTRAINT cob_dni_fk FOREIGN KEY (dni_pol) REFERENCES politico (dni) ON DELETE CASCADE;
```

López Morató, Marta

73590823T

Gracias por empezar cada ejercicio en una nueva cara

```
ALTER TABLE politico_cargo DROP CONSTRAINT polcar_dni_fk;
ALTER TABLE politico_cargo ADD CONSTRAINT polcar_dni_fk FOREIGN KEY (dni) REFERENCES politico (dni) ON DELETE CASCADE;
```

```
ALTER TABLE politico_concejal DROP CONSTRAINT polcon_dni_fk;
ALTER TABLE politico_concejal ADD CONSTRAINT polcon_dni_fk FOREIGN KEY (dni) REFERENCES politico (dni) ON
DELETE CASCADE;
```

```
ALTER TABLE telefono DROP CONSTRAINT poltel_dni_fk;
ALTER TABLE telefono ADD CONSTRAINT poltel_dni_fk FOREIGN KEY (dni) REFERENCES politico (dni) ON DELETE CASCADE;
```

```
ALTER TABLE asesor DROP CONSTRAINT ase_dnipol_fk;
ALTER TABLE asesor ADD CONSTRAINT ase_dnipol_fk FOREIGN KEY (dni_pol) REFERENCES politico (dni) ON DELETE CASCADE;
```

```
DELETE FROM politico
WHERE nombre = "Armando";
```

→→→→→→→→→→→→→→→→→→→→→→