

**[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:*

**\*Las tablas en 3FN están encuadradas.**

\*\*\*\*\*

**// Relación Entidad ACADEMIA + Entidad CURSO**

// Cardinalidad N:M entre ACADEMIA y CURSO

// Atención: Restricción de identidad de CURSO con respecto a ACADEMIA

**ACADEMIA**(cod\_acad, nombre, fecha\_ini, fecha\_fin)

CP: {cod\_acad}

**CURSO**(cod\_curso, cod\_acad, nombre, descripcion, asistentes)

CP: {cod\_curso, cod\_acad}

CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}

**OFERTA**(cod\_acad, cod\_curso, fecha\_ini, fecha\_fin)

CP: {cod\_acad, cod\_curso}

CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

**Restricciones:**

- (1) Asistentes derivado → Se calcula por el número de alumnos que asisten a un curso impartido por un profesor. Dado que cada curso tiene uno y solo un profesor, se podría calcular por la cantidad de veces que aparece un mismo cod\_curso en la tabla alumno.
- (2) Pérdida semántica → Todo curso debe tener una academia como mínimo y viceversa.
- (3) Pérdida semántica → Todo coste debe tener, al menos, una línea de coste.

**Tabla en 3FN.**

**ACADEMIA**(cod\_acad, nombre, fecha\_ini, fecha\_fin)

CP: {cod\_acad}

**1FN:** No hay atributos derivados, ni multivaluados, ni compuestos. La tabla ya está en 1FN.

**2FN:** La clave primaria es simple, con lo que no puede haber dependencias parciales. La tabla ya está en 2FN.

**3FN:** No hay dependencias transitivas. La tabla ya está en 3FN.

**OFERTA**(cod\_acad, cod\_curso, fecha\_ini, fecha\_fin)

CP: {cod\_acad, cod\_curso}

CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

**1FN:** No hay atributos derivados, ni multivaluados, ni compuestos. La tabla ya está en 1FN.

**2FN:** No hay dependencias parciales. La tabla ya está en 2FN.

**3FN:** No hay dependencias transitivas. La tabla ya está en 3FN.

\*\*\*\*\*

\*\*\*\*\*

**// Relación Entidad CURSO + Entidad MAESTRO**

// Cardinalidad N:1 entre CURSO y MAESTRO

// Atención: Se añade el campo id\_maestro a la tabla **CURSO**.

//Atención: Participación (1, 1) de **MAESTRO** en **CURSO**, por ello el VNN de id\_maestro en **CURSO**.

Giner Barrios, Pablo

21003297L

*Gracias por empezar cada ejercicio en una nueva cara*

**CURSO**(cod\_curso, cod\_acad, nombre, descripcion, asistentes, id\_maestro)

CP: {cod\_curso, cod\_acad}

CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}

CAj: {id\_maestro → **MAESTRO**(id)}

VNN: {id\_maestro}

**MAESTRO**(id, fecha\_grado, nom\_completo.nombre, nom\_completo.apellidos)

CP: {id}

**Tabla en 3FN.**

**CURSO**(cod\_curso, cod\_acad, nombre, descripcion, id\_maestro)

CP: {cod\_curso, cod\_acad}

CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}

CAj: {id\_maestro → **MAESTRO**(id)}

VNN: {id\_maestro}

**1FN:** El atributo asistentes en un atributo derivado, con lo que lo quitamos de la tabla y especificamos el modo de calcularlo en el apartado de restricciones.

**2FN:** No hay dependencias parciales. La tabla ya está en 2FN.

**3FN:** No hay dependencias transitivas. La tabla ya está en 3FN.

**MAESTRO**(id, nombre, apellidos, fecha\_grado)

CP: {id}

**1FN:** Tenemos el atributo nom\_completo como atributo compuesto. Quitamos la referencia al atributo compuesto y lo dejamos como atributos simples.

**2FN:** La clave primaria es simple, con lo que no puede haber dependencias parciales. La tabla ya está en 2FN.

**3FN:** No hay dependencias transitivas. La tabla ya está en 3FN.

\*\*\*\*\*  
\*\*\*\*\*

**// Relación Entidad CURSO + Entidad COSTE**

// Cardinalidad N:1 entre CURSO y COSTE

**// Atención: Restricción de Identidad**

**CURSO**(cod\_curso, cod\_acad, nombre, descripcion, id\_maestro)

CP: {cod\_curso, cod\_acad}

CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}

CAj: {id\_maestro → **MAESTRO**(id)}

VNN: {id\_maestro}

**COSTE**(linea, cod\_curso, concepto, importe)

CP: {linea, cod\_curso}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

**Restricciones:**

(1) Pérdida semántica → Todo curso debe tener, al menos, una línea de coste.

**Tabla en 3FN.**

**COSTE**(linea, cod\_curso, concepto, importe)

CP: {linea, cod\_curso}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

**1FN:** No hay atributos derivados, ni multivaluados, ni compuestos. La tabla ya está en 1FN.

**2FN:** No hay dependencias parciales. La tabla ya está en 2FN.

**3FN:** No hay dependencias transitivas. La tabla ya está en 3FN.

Giner Barrios, Pablo

21003297L

*Gracias por empezar cada ejercicio en una nueva cara*

\*\*\*\*\*  
\*\*\*\*\*

**// Relación Entidad CURSO + Entidad ALUMNO**

// Cardinalidad N:1 entre ALUMNO y CURSO

// Atención: Restricción de Identidad

**CURSO**(cod\_curso, cod\_acad, nombre, descripcion, asistentes, id\_maestro)

CP: {cod\_curso, cod\_acad}

CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}

CAj: {id\_maestro → **MAESTRO**(id)}

VNN: {id\_maestro}

**ALUMNO**(id, nom\_completo.nombr, nom\_completo.apellidos, {midiclorianos}n, fecha\_ini, fecha\_fin, cod\_curso)

CP: {id}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

VNN: {cod\_curso}

**Restricciones:**

(1) Pérdida semántica: Todo curso debe tener, al menos, un alumno.

**Tabla en 3FN.**

\*\*\*\*\*  
\*\*\*\*\*

**// Relación Entidad ALUMNO + Entidad ALUMNO**

// Cardinalidad N:1 entre ALUMNO y ALUMNO

// Atención: Añadimos un campo para cubrir la relación reflexiva.

**ALUMNO**(id, nom\_completo.nombre, nom\_completo.apellidos, {midiclorianos}n, fecha\_ini, fecha\_fin, cod\_curso, ayudado\_por)

CP: {id}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

CAj: {ayudado\_por → **ALUMNO**(id)}

VNN: {cod\_curso}

**Restricciones:**

--

**Tabla en 3FN.**

**1FN:** Tenemos el atributo nom\_completo como atributo compuesto y el atributo midiclorianos como multivaluado.

Quitamos la referencia al atributo compuesto dejándolos como atributos simples y quitamos el atributo midiclorianos para crear una nueva tabla que relacione los niveles de midiclorianos con los id de los alumnos.

**ALUMNO**(id, nombre, apellidos, fecha\_ini, fecha\_fin, cod\_curso, ayudado\_por)

CP: {id}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

CAj: {ayudado\_por → **ALUMNO**(id)}

VNN: {cod\_curso}

**MIDICLORIANOS**(midiclorianos, id\_alumno)

CP: {midiclorianos, id\_alumno}

CAj: {id\_alumno → **ALUMNO**(id)}

**2FN:** No hay claves primarias compuestas, por lo que no puede haber dependencias parciales. La tabla ya está en 2FN.

**3FN:** No hay dependencias transitivas. La tabla ya está en 3FN.

\*\*\*\*\*  
\*\*\*\*\*

**// Atención: Se trata de una especialización. La CP de PDAWAN, SENIOR y ALUMNO son la misma.**

**ALUMNO**(id, nom\_completo.nombre, nom\_completo.apellidos, {midiclorianos}n, fecha\_ini, fecha\_fin, cod\_curso, ayudado\_por)

CP: {id}

CAj: {cod\_curso → **CURSO**(cod\_curso)}

CAj: {ayudado\_por → **ALUMNO**(id)}

VNN: {cod\_curso}

**PDAWAN**(id, fecha\_entrada)

CP: {id}

CAj: {id → **ALUMNO**(id)}

**SENIOR**(id, especialidad, fecha\_nivel)

CP: {id}

CAj: {id → **ALUMNO**(id)}

**Restricciones:**

(1) Pérdida semántica → Todo alumno debe ser padawan o senior y no puede ser ambos a la vez.

**Tabla en 3FN.**

**PDAWAN**(id, fecha\_entrada)

CP: {id}

CAj: {id → **ALUMNO**(id)}

**1FN:** No hay atributos derivados, ni multivaluados, ni compuestos. La tabla ya está en 1FN.

**2FN:** La clave primaria es simple, con lo que no puede haber dependencias parciales. La tabla ya esta en 2FN.

**3FN:** Solo hay un atributo no perteneciente a la clave primaria, por lo que no puede haber dependencias transitivas. La tabla ya está en 3FN.

**SENIOR**(id, especialidad, fecha\_nivel)

CP: {id}

CAj: {id → **ALUMNO**(id)}

**1FN:** No hay atributos derivados, ni multivaluados, ni compuestos. La tabla ya está en 1FN.

**2FN:** La clave primaria es simple, con lo que no puede haber dependencias parciales. La tabla ya esta en 2FN.

**3FN:** No hay dependencias transitivas. La tabla ya está en 3FN.

\*\*\*\*\*

\*\*\*\*\*

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

Indica las sentencias necesarias para incorporar las tablas del ejercicio anterior en MySQL.

**ACADEMIA**(cod\_acad, nombre, fecha\_ini, fecha\_fin)  
CP: {cod\_acad}

```
CREATE TABLE academia (
cod_acad CHAR(6),
nombre VARCHAR(50),
fecha_ini DATE,
fecha_fin DATE,
CONSTRAINT aca_cod_pk PRIMARY KEY (cod_acad)
);
```

**CURSO**(cod\_curso, cod\_acad, nombre, descripcion, id\_maestro)  
CP: {cod\_curso, cod\_acad}  
CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}  
CAj: {id\_maestro → **MAESTRO**(id)}  
VNN: {id\_maestro}

```
CREATE TABLE curso (
cod_curso INTEGER,
cod_acad CHAR (6),
nombre CHAR (6),
descripción VARCHAR(60),
id_maestro INTEGER NOT NULL,
CONSTRAINT cur_cod_pk PRIMARY KEY (cod_curso, cod_acad),
CONSTRAINT cur_cod_fk FOREIGN KEY (cod_acad) REFERENCES academia (cod_acad),
CONSTRAINT cur_id_fk FOREIGN KEY (id_maestro) REFERENCES maestro (id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

**OFERTA**(cod\_acad, cod\_curso, fecha\_ini, fecha\_fin)  
CP: {cod\_acad, cod\_curso}  
CAj: {cod\_acad → **ACADEMIA**(cod\_acad)}  
CAj: {cod\_curso → **CURSO**(cod\_curso)}

```
CREATE TABLE oferta (
cod_acad CHAR(6),
cod_curso INTEGER,
fecha_ini DATE,
fecha_fin DATE,
CONSTRAINT ofe_cod_pk PRIMARY KEY (cod_acad, cod_curso),
CONSTRAINT ofe_aca_fk FOREIGN KEY (cod_acad) REFERENCES academia (cod_acad)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT ofe_cod_fk FOREIGN KEY (cod_curso) REFERENCES curso (cod_curso)
);
```

**MAESTRO**(id, nombre, apellidos, fecha\_grado)  
CP: {id}

```
CREATE TABLE maestro (
id INTEGER,
nombre VARCHAR(30),
apellidos VARCHAR (50),
fecha_grado DATE,
```

Giner Barrios, Pablo

21003297L

*Gracias por empezar cada ejercicio en una nueva cara*

CONSTRAINT mae\_id\_pk PRIMARY KEY (id)  
);

**COSTE**(linea, cod\_curso, concepto, importe)  
CP: {linea, cod\_curso}  
CAj: {cod\_curso → **CURSO**(cod\_curso)}

```
CREATE TABLE coste (
  linea INTEGER,
  cod_curso INTEGER,
  concepto VARCHAR(50),
  importe DOUBLE,
  CONSTRAINT cos_lin_pk PRIMARY KEY (linea, cod_curso),
  CONSTRAINT cos_cod_fk FOREIGN KEY (cod_curso) REFERENCES curso (cod_curso)
);
```

**ALUMNO**(id, nombre, apellidos, fecha\_ini, fecha\_fin, cod\_curso, ayudado\_por)  
CP: {id}  
CAj: {cod\_curso → **CURSO**(cod\_curso)}  
CAj: {ayudado\_por → **ALUMNO**(id)}  
VNN: {cod\_curso}

```
CREATE TABLE alumno (
  id CHAR(6),
  nombre VARCHAR(30),
  apellidos VARCHAR(50),
  fecha_ini DATE,
  fecha_fin DATE,
  cod_curso INTEGER NOT NULL,
  ayudado_por CHAR(6),
  CONSTRAINT alu_id_pk PRIMARY KEY (id),
  CONSTRAINT alu_cod_fk FOREIGN KEY (cod_curso) REFERENCES curso (cod_curso),
  CONSTRAINT alu_ayu_fk FOREIGN KEY (ayudado_por) REFERENCES alumno (id)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

**MIDICLORIANOS**(midiclorianos, id\_alumno)  
CP: {midiclorianos, id\_alumno}  
CAj: {id\_alumno → **ALUMNO**(id)}

```
CREATE TABLE midiclorianos (
  midiclorianos INTEGER,
  id CHAR(6),
  CONSTRAINT mid_mid_pk PRIMARY KEY (midiclorianos),
  CONSTRAINT mid_id_fk FOREIGN KEY (id) REFERENCES alumno (id)
);
```

**PADAWAN**(id, fecha\_entrada)  
CP: {id}  
CAj: {id → **ALUMNO**(id)}

```
CREATE TABLE padawan (
  id CHAR(6),
  fecha_entrada DATE,
  CONSTRAINT pad_id_pk PRIMARY KEY (id),
  CONSTRAINT pad_id_fk FOREIGN KEY (id) REFERENCES alumno (id)
  ON DELETE CASCADE ON UPDATE CASCADE
);
```

**SENIOR**(id, especialidad, fecha\_nivel)

CP: {id}

CAj: {id → **ALUMNO**(id)}

```
CREATE TABLE senior (
id CHAR(6),
especialidad VARCHAR(50),
fecha_nivel DATE,
CONSTRAINT sen_id_pk PRIMARY KEY (id),
CONSTRAINT sen_id_fk FOREIGN KEY (id) REFERENCES alumno (id)
ON DELETE CASCADE ON UPDATE CASCADE
);
```

**Restricciones:**

- (1) Pérdida semántica → Toda academia debe tener, al menos, un curso y todo curso debe pertenecer, al menos, a una academia.
- (2) Pérdida semántica → Todo curso debe tener, al menos, una línea de coste.
- (3) Pérdida semántica → Todo curso debe tener, al menos, un alumno.
- (4) Pérdida semántica → Todos los alumnos han de ser o padawan o senior y ninguno puede ser ambos a la vez.

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

Indica las sentencias necesarias para realizar estas modificaciones en MySQL:

Operación 1 (Modificar la tabla MAESTRO 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.)

// Añadir una restricción a la tabla estableciendo como UNIQUE KEY los campos nombre y apellidos.

```
ALTER TABLE maestro ADD CONSTRAINT mae_nom_uk UNIQUE KEY (nombre, apellidos);
```

Operación 2 (Cambiar la participación de la entidad alumno en la relación ASISTIR de (1,1) a (0,1).)

// Modificar la tabla ALUMNO para que el campo cod\_curso, que es la clave principal de la tabla CURSO no sea Valor No Nulo.

```
ALTER TABLE alumno MODIFY cod_curso VARCHAR(10);
```

Operación 3 (Permitir que se pueda modificar el ID de la tabla CURSO, propagando este cambio a las tablas dependientes de este dato.)

//Hay que eliminar las restricciones donde el campo cod\_curso funciona como clave ajena para volver a crearlas con el actualizado en cascada.

```
ALTER TABLE oferta DROP CONSTRAINT ofe_cod_fk;
```

```
ALTER TABLE oferta ADD CONSTRAINT ofe_cod_fk FOREIGN KEY (cod_curso) REFERENCES curso (cod_curso) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE coste DROP CONSTRAINT cos_cod_fk;
```

```
ALTER TABLE coste ADD CONSTRAINT cos_cod_fk FOREIGN KEY (cod_curso) REFERENCES curso (cod_curso) ON DELETE CASCADE ON UPDATE CASCADE;
```

```
ALTER TABLE alumno DROP CONSTRAINT alu_cod_fk;
```

```
ALTER TABLE alumno ADD CONSTRAINT alu_cod_fk FOREIGN KEY (cod_curso) REFERENCES curso (cod_curso) ON DELETE CASCADE ON UPDATE CASCADE;
```

Operación 3 (Modificar la tabla ACADEMIA para que por defecto la fecha de inicio sea la fecha del sistema.)

//Hay que modificar el campo fecha\_ini de la tabla academia para que el tipo de dato sea TIMESTAMP, que incorpora ya como default la fecha del sistema.

```
ALTER TABLE academia MODIFY fecha_ini TIMESTAMP;
```



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

Indica las sentencias necesarias para realizar estas modificaciones en MySQL:

**Operación 1** (Crea una academia con código “AMYEND”, nombre “Academia Maestro Yoda planetaEndor”).

// Hay que utilizar la sentencia para insertar datos en la tabla correspondiente, indicando el campo en el que introduces cada dato.

```
INSERT INTO academia (cod_acad, nombre)
--> VALUES ('AMYEND', 'Academia Maestro Yoda planetaEndor');
```

**Operación 2** (Crea un maestro Jedi con id “1”, nombre “Luke”, apellidos “Skywalker”).

// Del mismo modo que la operación anterior.

```
INSERT INTO maestro (id, nombre, apellidos)
--> VALUES (1, 'Luke', 'Skywalker');
```

**Operación 3** (Crea un curso para la academia creada anteriormente con código “7050”, nombre “CAPAD2”, descripción “Curso avanzado padawan nivel 2”. El maestro Jedi es Luke Skywalker.)

//Se crea el curso en la tabla curso se introducen los datos correspondientes y, después, relacionas el curso con la academia que se ha creado anteriormente en la tabla oferta, introduciendo el código de la academia y el código del curso.

```
INSERT INTO curso (cod_curso, nombre, descripcion, id_maestro)
--> VALUES (7050, 'CAPAD2', 'Curso avanzado padawan nivel 2', 1);
```

```
INSERT INTO oferta (cod_acad, cod_curso)
--> VALUES ('AMYEND', 7050);
```

**Operación 4** (Crea 3 alumnos asistentes al curso anterior con los siguientes datos – id “PAD001”, nombre “Rey”, apellidos “Skywalker”. La alumna es de tipo senior con especialidad “sable láser” y fecha de nivel 30/06/2022. id “PAD002”, nombre “Baby”, apellidos “Yoda”. El alumno es de tipo senior con especialidad “telequinesis” y fecha de nivel 31/07/2022. is “PAD003”, nombre “Chewbacca”, apellidos “Wookiee”. El alumno es de tipo senior con especialidad “Fuerza Bruta” y fecha de nivel 31/05/2022.

//Se crean los alumnos introduciendo los datos en la tabla alumnos y se introducen los datos de cada alumno a la tabla correspondiente padawan o senior.

```
INSERT INTO alumno (id, nombre, apellidos, cod_curso)
--> VALUES ('PAD001', 'Rey', 'Skywalker', 7050),
--> ('PAD002', 'Baby', 'Yoda', 7050),
--> ('PAD003', 'Chewbacca', 'Wookiee', 7050);
```

```
INSERT INTO senior (id, especialidad, fecha_nivel)
values --> ('PAD001', 'Sable Láser', '2022/06/30'),
--> ('PAD002', 'Telequinesis', '2022/07/31'),
--> ('PAD003', 'Fuerza Bruta', '2022/05/31');
```

**Operación 5** (El alumno Chewbacca es ayudado por la alumna Rey.)

//Hay que modificar la tabla alumno donde el nombre sea Chewbacca para añadir el dato de la id de rey.

```
UPDATE alumno  
SET ayudado_por='PAD001'  
WHERE nombre='Chewbacca';
```

**Operación 6** (El curso impartido genera 2 costes: “Salario profesorado” con importe 8 créditos imperiales y “Rotura de sables” con importe 1,5 créditos imperiales, respectivamente.)

//se introducen los datos en la tabla coste.

```
INSERT INTO coste (linea, cod_curso, concepto, importe)  
--> VALUES (1, '7050', 'Salario profesorado', 8),  
--> (2, '7050', 'Rotura sables', 1.5);
```

**Operación 7** (Actualiza la fecha de inicio del curso impartido a 1/09/2022.)

//Se introducen los datos en la tabla curso\_academia.

```
INSERT INTO curso_academia (cod_curso, cod_acad, fecha_ini)  
--> VALUES ('7050', 'AMYEND', '2022/09/1');
```

**Operación 8** (Borra al alumno Chewbacca.)

//Se elimina, de la tabla alumno, donde el nombre del alumno sea Chewbacca.

```
DELETE FROM alumno WHERE nombre = 'Chewbacca';
```