

[4 PUNTOS] EJERCICIO 1: MODELADO LÓGICO RELACIONAL

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

Tarjetas: {idtarjeta, dni_bookier}

PK: {idtarjeta}

FK: {dni_bookier} -> Bookiers(dni)

VNN: {dni_bookier}

Bookiers: {dni, nombre, apellidos, fecha_nacimiento, id-centro}

PK: {dni}

FK: {id-centro} -> Centro(codigo-centro)

VNN: {id-centro}

Restricción: Toda fila de Bookiers debe tener como mínimo una fila asociada en Tarjetas.

Toda fila de Bookiers debe tener como mínimo una fila asociada en Centros.

Centro {codigo-centro, ciudad, nombre}

PK: {codigo-centro}

//Tabla creada para conseguir 3FN en Bookiers

BooikierEstudiante: {dni_bookier}

PK: {dni_bookier}

FK: {dni_bookier} -> Bookiers(dni)

Restricción: Todo Bookier debe ser Estudiante o Docente o ambos a la vez.

BookierDocente: {}

PK: {}

EstuVentas: {idestuventas, idventas, dni_bookier}

PK: {idestuventas}

FK: {dni_bookier} -> Bookiers(dni)

FK: {idventa} -> Ventas

VNN: {dni_bookier}

Restricción: 1. Toda venta debe ser Estuventas o DocVentas

2. Atributo derivado imp_total_estuventas, se deberá agregar posteriormente, este almacenará el valor total de todas las ventas de EstuVentas.

3. Toda línea en EstuVentas debe tener como mínimo una fila asociada en Líneas Estudiantes.

Ventas: {idventa, fecha, ip}

PK: {idventa}

Restricción: Atributo derivado imp_total_ventas, se deberá agregar posteriormente, este almacenará el valor total de todas las ventas de la tabla

DocVentas {}

PK: {}

Líneas Estudiantes: {n_linea, idestuventas, id_material, cantidad}

PK: {n_linea, idestuventas}

FK: {idestuventas} -> EstuVentas

FK: {id_material} -> Matestudiantes

VNN: {id_material}

MatEstudiantes: {id_material_estu}

PK: {id_matertial_estu}

FK: {id_matertial_estu} -> Materiales(id_material)

Restricción: 1. Todo material debe ser MatEstudiantes o MatDocentes

2. Toda línea en Matestudiantes debe tener como mínimo una línea asociada en Líneas Estudiantes

MatDocentes: {}

PK: {}

Materiales: {id_material, precio, nombre}

PK: {id_material}

Recomienda: {id_materia, id_material_recomentado}

PK: {id_materia, id_material_recomentado}

FK: {id_material} -> Materiales

FK: {id_material_recomentado} -> Matertiales

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

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

```
create database Evaluable2;  
use Evaluable2;
```

```
create table centro(  
codigo_centro smallint,  
ciudad varchar (25),  
nombre varchar (30),  
constraint centro_pk primary key (codigo_centro));
```

```
create table bookiers(  
dni varchar(9),  
nombre varchar(15),  
apellidos varchar(30),  
id_centro smallint,  
constraint bookiers_pk primary key (dni),  
constraint bookiers_fk foreign key (id_centro) references centro(codigo_centro)  
on delete cascade on update cascade);
```

```
create table tarjetas(  
idtarjeta smallint,  
dni_bookier varchar(9) not null,  
constraint tarejtas_pk primary key (idtarjeta),  
constraint tarjetas_fk foreign key (dni_bookier) references bookiers(dni)  
on delete cascade on update cascade);
```

```
create table bookierEstudiante(  
dni_bookier varchar(9),  
constraint bookierestudiante_pk primary key (dni_bookier),  
constraint bookierestudiante_fk foreign key (dni_bookier) references bookiers(dni)  
on delete cascade on update cascade);
```

```
create table ventas(  
idventas mediumint,  
fecha date,  
ip varchar(15),  
constraint ventas_pk primary key (idventas));
```

```
create table estuventas(  
idestuventas smallint,  
idventas mediumint,  
dni_bookier varchar(9) not null,  
constraint estuventas_pk primary key (idestuventas),  
constraint estuventas_dni_fk foreign key (dni_bookier) references bookiers(dni)  
on delete cascade on update cascade,  
constraint estuventas_idventa_fk foreign key (idventas) references ventas(idventas))
```

on delete cascade on update cascade);

```
create table materiales(  
  id_material smallint,  
  precio float,  
  nombre varchar(50),  
  constraint materiales_pk primary key (id_material));
```

```
create table matestudiantes(  
  id_material_estu smallint,  
  constraint matestudiantes_pk primary key (id_material_estu),  
  constraint matestudiantes_fk foreign key (id_material_estu) references materiales(id_material));
```

```
create table lineas_estuventas(  
  n_linea smallint,  
  idestuventas smallint,  
  id_material_estu smallint not null,  
  cantidad smallint,  
  constraint lineas_estudiantas_pk primary key (n_linea, idestuventas),  
  constraint lineas_estud_ventas_fk foreign key (idestuventas) references estuventas(idestuventas)  
  on delete cascade on update cascade,  
  constraint lineas_estud_material_fk foreign key (id_material_estu) references matestudiantes(id_material_estu)  
  on delete cascade on update cascade);
```

```
create table recomienda_material(  
  id_material smallint,  
  id_material_recomendado smallint,  
  constraint material_recomendado_pk primary key (id_material, id_material_recomendado),  
  constraint material_fk foreign key (id_material) references materiales(id_material)  
  on delete cascade on update cascade,  
  constraint material_recomendado_fk foreign key (id_material_recomendado) references materiales(id_material)  
  on delete cascade on update cascade);
```

[2 PUNTOS] EJERCICIO 3: MODELADO FÍSICO DDL (MODIFICACIÓN DE METADATOS)

Indica las sentencias necesarias para realizar estas modificaciones en MySQL:

- Modificar la tabla principal (BOOKIERS / DAWERS) 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.

Alter table bookiers add constraint Bookiers_UK unique key (nombre, apellidos);

- Cambiar la participación del lado izquierdo de la relación TENER de 1,1 a 0,1.

Para esta modificación, deberemos eliminar la propiedad VNN de la tabla Tarjetas.

Alter table tarjetas modify dni_bookier varchar(9);

Con esto modificamos la columna dándole de nuevo el mismo tipo de dato pero sin la propiedad VNN

- Eliminar la relación FORMAR_E o FORMAR_D del diagrama (según tu modelo) sin eliminar ninguna tabla.

Eliminaremos de la tabla Lineas_estudiantes la FK la columna id_material_estu (que proviene de la entidad Matestudiantes)

Alter table lineas_estudiantes drop foreign key lineas_estud_material_fk;

Alter table lineas_estudiantes drop column id_material_estu;

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

Esto lo conseguimos creando la propiedad On Update Cascade tras la creación de las constraint FK correspondientes a esa columna en las demás tablas:

Eliminamos primero la constraint para crearla 0:

Alter table matestudiantes drop constraint matestudiantes_fk;

Y la creamos con la nueva propiedad:

Alter table matestudiantes add constraint matestudiantes_fk foreign key (id_material_estu) references materiales(id_material) on update cascade;

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

Indica las sentencias necesarias para realizar estas modificaciones en MySQL:

Introducimos los materiales en la tabla "materiales":

```
insert into materiales values (101, 14, "bases de datos");  
insert into materiales values (102, 15, "Lenguaje de Marcas");  
insert into materiales values (103, 17, "Diagramas E-R Resueltos");  
insert into materiales values (104, 16, "Diseño Avanzado de Páginas Web");
```

Introducimos los materiales como materiales de estudiantes en "matestudiantes":

```
insert into matestudiantes values (101);  
insert into matestudiantes values (102);  
insert into matestudiantes values (103);  
insert into matestudiantes values (104);
```

Introducimos el centro al que perteneciera la bookier:

```
insert into centro values(15123, "Valencia", "CeedCV");
```

Introducimos a la estudiante en la tabla "bookiers":

```
insert into bookiers values ("24365471P", "María", "Pérez López",15123);
```

Introducimos a la bookier como estudiante en "bookierestudiante":

```
insert into bookierestudiante values ("24365471P");
```

Introducimos la venta de los dos productos (aunque ambos se harán en una misma venta) en la tabla "ventas":

```
insert into ventas values(1542, "2021-09-20", "44:153:0:22");
```

Introducimos la venta en "estuventas":

```
insert into estuventas values(133, 1542, "24365471P");
```

Una vez creados todos los datos necesarios, creamos la venta en sí en "lineas_estuventas":

```
insert into lineas_estuventas values (1, 133, 101, 2);  
insert into lineas_estuventas values (2, 133, 102, 1);
```

Introducimos ahora la recomendacion de material en "recomienda_material":

```
insert into recomienda_material values (101,103);  
insert into recomienda_material values (102,104);
```

Realizamos la ultima compra:

```
insert into ventas values (1601, "2021-10-15", "55:132:86:1");  
insert into estuventas values (134, 1601, "24365471P");  
insert into lineas_estuventas values(1, 134, 104, 1);  
insert into lineas_estuventas values(2, 134, 103, 6);
```