

Bases de datos

Unidade 4: Modificación da estrutura de BD relacionais

1. A02. Modificación da estrutura de bases de datos relacionais

1.1 Actividade

1.1.1 Modificación dunha base de datos

A sintaxe da sentenza que permite cambiar as características globais dunha base de datos é:

```
ALTER {DATABASE | SCHEMA} nome_da_base [opcións_a_modificar] ...
```

Consideracións sobre a sintaxe anterior:

- As opcións para modificar son as mesmas que as opcións de creación:
[DEFAULT] CHARACTER SET [=] nome_xogo_carácteres
[DEFAULT] COLLATE [=] nome_sistema_ordenación
- O cambio do xogo de caracteres ou as opcións de cotexamento para unha base de datos na que xa existan táboas, non afecta a estas e só afecta ás que se creen no futuro.



Tarefa 1. Modificar bases de datos empregando a sentenza ALTER.

1.1.2 Modificación do esquema dunha táboa

Co paso do tempo é normal que se teñan que facer adaptacións no esquema das táboas das bases de datos, pola aparición de novos requirimentos, novas restricións, ou a desaparición dalgunhas das existentes. Algunhas veces, ter que facer cambios débese a non terlle dedicado o tempo suficiente á fase de deseño conceptual e lóxico; é moi recomendable pararse a facer un bo deseño antes de empezar a escribir código para crear a base de datos.

A sentenza ALTER TABLE permite facer modificacións no esquema dunha táboa que xa existe na base de datos. Sintaxe:

```
ALTER TABLE nome_táboa  
[especificación_alter [, especificación_alter] ...]
```

A *especificación_alter* pode ser:

```
opcións de táboa  
| ADD [COLUMN] nome_columna definición_columna [FIRST | AFTER nome_columna]  
| ADD [COLUMN] (nome_columna definición_columna, ...)  
| CHANGE [COLUMN] nome_columna nome_novo nova_definición_columna  
| [FIRST|AFTER nome_columna],  
| MODIFY [COLUMN] nome_columna <nova_definición_columna> [FIRST | AFTER nome_columna],  
| ALTER [COLUMN] nome_columna {SET DEFAULT valor | DROP DEFAULT},  
| ADD {INDEX|KEY} [nome_índice] (columnas_índice),  
| ADD [CONSTRAINT [nome_restrición]] PRIMARY KEY (lista_columnas),  
| ADD [CONSTRAINT [nome_restrición]] UNIQUE {INDEX|KEY} [nome_índice] (lista_columnas),  
| ADD [CONSTRAINT [nome_restrición]] FOREIGN KEY [nome_índice] (lista_columnas)  
| REFERENCES nome_táboa (lista_de_columnas) [ON DELETE opción] [ON UPDATE opción]  
| DROP [COLUMN] nome_columna,  
| DROP {INDEX | KEY} nome_índice,  
| DROP PRIMARY KEY,
```

```
| DROP FOREIGN KEY nome_restrición,  
| RENAME [TO | AS] nome_táboa_nova  
| CONVERT TO CHARACTER SET xogo_carácteres [COLLATE sistema_colación]
```

Consideracións sobre a sintaxe:

- As opcións de táboa afectan ás características da táboa e son as mesmas que se utilizan na sentenza CREATE TABLE. Algunhas opcións de táboa son:

```
[DATA DIRECTORY= 'directorio']  
[INDEX DIRECTORY= 'directorio']  
[{ENGINE | TYPE} = {ISAM, MyISAM, InnoDB, ...}]  
[[DEFAULT] CHARACTER SET nome_xogo_carácteres] [COLLATE nome_sistema_colación]]  
[AUTO_INCREMENT = número]
```

- DATA DIRECTORY e INDEX DIRECTORY permiten cambiar as rutas absolutas nas que se almacenan os datos e os índices.
 - ENGINE permite cambiar o motor de almacenamento asociado á táboa. Ver o apartado "Motores de almacenamento en MySQL" que está máis adiante
 - CHARACTER SET e COLLATE permiten cambiar o conxunto de caracteres e o sistema de colación predeterminados para as columnas que se crean nesa táboa. Non afecta ás columnas que xa están creadas. Para cambiar o contido das columnas tipo cadea de caracteres, hai que utilizar a cláusula CONVERT TO CHARACTER SET.
 - AUTO_INCREMENT permite cambiar o número de comezo para a columna de tipo autoincremental.
- En xeral, as cláusulas ADD permiten engadir novas propiedades e as cláusulas DROP permiten eliminalas.
 - Se ao engadir unha columna (ADD) non se especifica a cláusula FIRST | AFTER, a nova columna engádese ao final da táboa.
 - Non é posible engadir unha columna tipo autoincremento se a táboa non está baleira.
 - Non se pode modificar NULL por NOT NULL se a táboa contén valores nulos para a columna a modificar; a operación inversa non presenta ningún problema.
 - Para cambiar a definición dunha columna, hai que utilizar a cláusula MODIFY, pero se ademais da definición tamén se quere cambiar o nome, entón hai que utilizar a cláusula CHANGE. O editor da actual versión de Workbench non reconece a cláusula MODIFY e marca a liña como un erro aínda que se executa correctamente.
 - A cláusula ALTER permite modificar o valor por defecto para unha columna.
 - A cláusula RENAME permite cambiar o nome da táboa.
 - A cláusula CONVERT TO CHARACTER SET permite cambiar o xogo de caracteres e o sistema de colación por defecto, e ademais, cambiar os valores de todas as columnas tipo cadea de caracteres (CHAR, VARCHAR e TEXT). Hai que ter en conta que non todos os sistemas de colación utilizan o mesmo número de bytes polo que pode ser necesario cambiar antes o tipo ou tamaño da columna.

Pódese facer a conversión só en algunha columna e non en todas as tipo cadea de caracteres, nese caso pódese utilizar a cláusula MODIFY. Exemplo:

```
alter table proba  
  modify column texto_latin1 varchar(200) character set utf8,
```

No manual pódense consultar todas as opcións que ten esta sentenza.

Exemplo de modificación da estrutura da táboa *fabricante* da base de datos *practicasl*:

```
alter table fabricante  
  add column pais varchar(60) default null,  
  add column enderezo varchar(200) not null after idFabricante,
```

```
add index idx_fabricante_nome (nome),  
engine = InnoDB;
```



Tarefa 2. Modificar táboas empregando a sentenza ALTER.

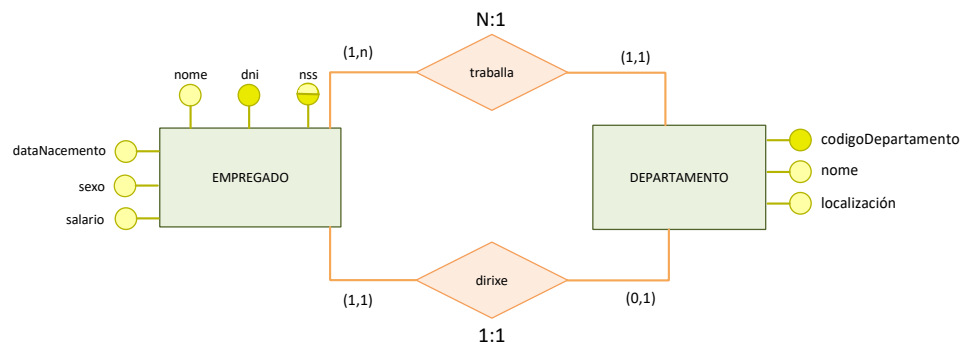
1.1.3 Engadir relacións e restricións de clave foránea

Pódense engadir relacións e restricións de clave foránea de dúas formas:

- Crear ao mesmo tempo as táboas e as relacións, empregando a sentenza CREATE TABLE. Neste caso hai que ter en conta a orde en que se crean as táboas, xa que non se pode crear unha táboa que conteña unha clave foránea se aínda non está creada a táboa á que fai referencia. Isto pode representar un problema no caso de relacións bidireccionais.
- Crear primeiro as táboas e establecer as relacións despois empregando sentenzas ALTER TABLE. Recoméndase esta segunda opción.

Por exemplo, supóñase que a entidade *empregado* está relacionada coa entidade *departamento* cunha relación *traballa* de tipo N:1, e *departamento* está relacionada con *empregado* coa relación *dirixe*, de tipo 1:1, con cardinalidade mínima 0. Ao crear primeiro a táboa *empregado* e despois *departamento*, provocaríase un erro porque na orde de creación da táboa *empregado* se define unha clave foránea que fai referencia á táboa *departamento*, que aínda non existe. O mesmo ocorre se empeza creando *departamento*.

En MySQL pódese solucionar o problema anterior desactivando a verificación de claves foráneas, poñendo o valor 0 ou OFF na variable *foreign_key_checks*.



Tarefa 3: Probar as posibilidades de creación de dúas táboas con interrelacións 1:N bidireccionais.



Tarefa 4. Utilizar a sentenza ALTER TABLE para a creación das relacións entre táboas.

1.1.4 Tarefa integradora sobre a LDD

A existencia desta tarefa está xustificada pola necesidade de realizar consecutivamente varias operacións sobre a base de datos que integre a maior parte das operacións en LDD estudadas ata o momento nesta unidade didáctica, como son: creación, borrado e modificación de bases de datos; creación, borrado e modificación de táboas e aplicación de restricións.



Tarefa 5. Tarefa integradora sobre LDD.

1.2 Tarefas

As tarefas propostas son as seguintes:

- Tarefa 1. Modificar bases de datos empregando a sentenza ALTER.
- Tarefa 2. Modificar táboas empregando a sentenza ALTER.
- Tarefa 3. Probar as posibilidades de creación de dúas táboas con interrelacións 1:N bidireccionais.
- Tarefa 4. Utilizar a sentenza ALTER TABLE para a creación das relacións entre táboas.
- Tarefa 5. Tarefa integradora sobre LDD.

1.2.1 Tarefa 1. Modificar bases de datos empregando a sentenza ALTER

A tarefa consiste en cambiar o xogo de caracteres para a base de datos *practicasl*. O novo xogo de caracteres será latin1, co sistema de colación ‘general’.

Solución

```
alter database practicasl
character set latin1
collate latin1_general_ci;
```

1.2.2 Tarefa 2. Modificar táboas empregando a sentenza ALTER

A tarefa consiste en realizar modificacións en táboas empregando a sentenza ALTER en varios casos:

- Tarefa 2.1. Facer os seguintes cambios na táboa *fabricante* da base de datos *practicasl*:
 - Engadir a columna de tamaño variable *email*, asignándolle como valor por defecto o valor nulo.
 - Engadir a columna de tamaño fixo *cif*, que non admite valores nulos, e colocala na segunda posición, despois da columna *idFabricante*.
 - Engadir un índice asociado á columna *cif*.
 - Cambiar o motor de almacenamento que ten actualmente, por Innodb.
- Tarefa 2.2. Facer os seguintes cambios na táboa *fabricante* da base de datos *practicasl*:

- Cambiar a clave primaria da táboa *fabricante*. A nova clave primaria pasa a ser a columna *cif*.
- Eliminar a columna *idFabricante*.
- Eliminar o índice asociado á columna *nome*.
- Tarefa 2.3. Facer os seguintes cambios na táboa *grupo* da base de datos *practicasl*:
 - Cambiarlle o nome a columna *cache*. O novo nome é *custo*.
 - Asignarlle á columna *dataFormación* como valor por defecto o valor cero.

Solución

■ Tarefa 2.1

```
/*cambios na táboa de fabricante*/
alter table fabricante
  add column email varchar(150) default null,
  add column cif char(9) not null after idFabricante,
  add index idx_fabricante_cif (cif),
  engine = InnoDB;
```

■ Tarefa 2.2

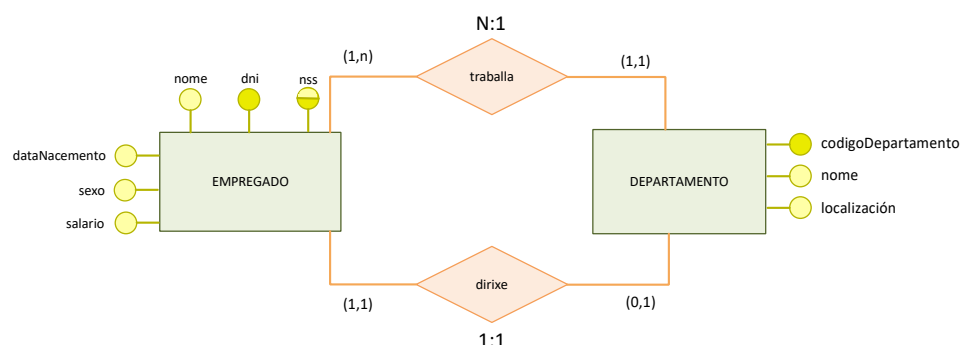
```
/*cambios na táboa de fabricante*/
alter table fabricante
  add primary key (cif),
  drop column idFabricante,
  drop index idx_fabricante_nome;
```

■ Tarefa 2.3

```
/*cambios na táboa grupo*/
alter table grupo
  change cache custo mediumint unsigned,
  alter dataFormacion set default 0;
```

1.2.3 Tarefa 3: Probar as posibilidades de creación de dúas táboas con interrelacións 1:N bidireccionais

A tarefa consiste en crear na base de datos *practicasl*, as táboas *empregado* e *departamento* que están relacionadas como se indica no seguinte diagrama E/R:



Baixo os seguintes supostos:

- Tarefa 3.1. Crear a táboa *empregado* só sentenzas CREATE TABLE.

- Tarefa 3.2. Crear primeiro as dúas táboas, sen claves foráneas, e despois engadir as claves foránea que establecen a relación entre elas coa axuda da sentenza ALTER TABLE. Borrar primeiro as táboas, se xa existiran.
- Tarefa 3.3. Desactivar a verificación de claves foráneas, coa variable *foreign_key_checks*, e despois facer a creación das táboas. Borrar primeiro as táboas, se xa existiran.

Solución

▪ Tarefa 3.1

Este primeiro suposto vai dar lugar a unha mensaxe de erro, debido a que cando se crea a táboa empregado aínda non existe a táboa departamento.

```
use practicas1;
drop table if exists empregado;
drop table if exists departamento;
create table empregado
(
dni char(9) not null,
nss char(10) not null,
nome varchar(80) not null,
dataNacemento date,
sexo enum('h','m') comment 'h = home, m = muller',
salario decimal(11,2) unsigned,
primary key (dni),
index idx_empleado_nss (nss),
codigoDepartamento smallint unsigned,
constraint fk_empleado_departamento foreign key (codigoDepartamento)
references departamento(codigoDepartamento)
on delete set null
on update cascade
);
create table departamento
(
codigoDepartamento smallint unsigned auto_increment not null,
nome varchar(60) not null,
localizacion varchar(40),
dniXefe char(9),
primary key (codigoDepartamento),
constraint fk_departamento_empleado foreign key (dniXefe) references empleado(dni)
on delete set null
on update cascade
);
```

Cando se executa a sentenza *create table empleado* prodúcese un erro porque a clave foránea *codigoDepartamento* fai referencia a unha táboa que aínda non existe. Se mostra a seguinte mensaxe de erro:

```
Error Code: 1215. No puede adicionar clave extranjera constraint
```

▪ Tarefa 3.2

Este segundo suposto solucionaría o problema anterior creando primeiro as táboas e despois engadindo as relacións coa axuda da sentenza ALTER.

```
use practicas1;
drop table if exists empleado;
drop table if exists departamento;
/*creación das táboas*/
create table empleado
(
```

```

dni char(9) not null,
nss char(10) not null,
nome varchar(80) not null,
dataNacemento date,
sexo enum('h','m') comment 'h = home, m = muller',
salario decimal(11,2) unsigned,
codigoDepartamento smallint unsigned,
primary key (dni),
index idx_empregado_nss (nss)
);
create table departamento
(
codigoDepartamento smallint unsigned auto_increment not null,
nome varchar(60) not null,
localizacion varchar(40),
dniXefe char(9),
primary key (codigoDepartamento)
);
/*engadir as restriccións de clave foránea par repretar as relacións */
alter table empregado
add constraint fk_empregado_departamento foreign key (codigoDepartamento)
references departamento(codigoDepartamento)
on delete set null
on update cascade;
alter table departamento
add constraint fk_departamento_empregado foreign key (dniXefe)
references empregado(dni)
on delete set null
on update cascade;

```

■ Tarefa 3.3

Este terceiro suposto solucionaría o problema desactivando a verificación de claves foráneas antes da creación das táboas.

```

/*desactivar a verificación de claves foráneas */
set FOREIGN_KEY_CHECKS = 0;
use practicas1;
drop table if exists empregado;
drop table if exists departamento;
create table empregado
(
dni char(9) not null,
nss char(10) not null,
nome varchar(80) not null,
dataNacemento date,
sexo enum('h','m') comment 'h = home, m = muller',
salario decimal(11,2) unsigned,
primary key (dni),
index idx_empregado_nss (nss),
codigoDepartamento smallint unsigned,
constraint fk_empregado_departamento foreign key (codigoDepartamento)
references departamento(codigoDepartamento)
on delete set null
on update cascade
);
create table departamento
(
codigoDepartamento smallint unsigned auto_increment not null,
nome varchar(60) not null,
localizacion varchar(40),

```



```

dniXefe    char(9),
primary key (codigoDepartamento),
constraint fk_departamento_empregado foreign key (dniXefe)
references empregado(dni)
on delete set null
on update cascade
);
/*activar a verificación de claves foráneas*/
set FOREIGN_KEY_CHECKS = 1;

```

1.2.4 Tarefa 4. Utilizar a sentenza ALTER TABLE para a creación das relacións entre táboas

A tarefa consiste en crear o esquema dunha base de datos creando primeiro as táboas e definindo despois as relacións entre eles en dous supostos:

- Tarefa 4.1. Crear a base de datos *eleccionModulos* e crear nela as táboas *profesor*, *grupo*, *modulo* e *imparte*, segundo a seguinte descrición:
 - Táboa *profesor*

Nome	Tipo	Null	Clave	Observacións
idProfesor	char(15)	Non	Primar y	Nome curto do profesor
corpo	enum	Non		Corpo ao que pertence. Valores permitidos: S = Profesorado Secundaria T = Profesorado Técnico
nrp	char(20)	Non	Única	Número de Rexistro Persoal
nome	varchar(30)	Non		
apelidos	varchar(60)	Non	Indice	
horasLectivas	smallint (sen signo)			Nº de horas de clase que imparte o profesor. Por defecto: 18

- Táboa *grupo*

Nome	Tipo	Null	Clave	Observacións
idGrupo	char(15)	Non	Primar y	Clave para identificar o grupo
descripcion	varchar(100)	Non		Descrición do nome do grupo
horasTitoria	smallint (sen signo)			Nº de horas de redución para o profesor titor
idProfesor	char(15)			Identificador do profesor que é titor do grupo

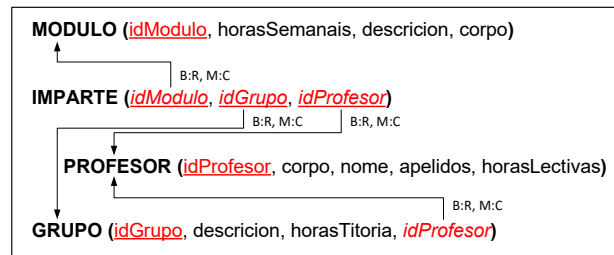
- Táboa *modulo*

Nome	Tipo	Null	Clave	Observacións
idModulo	char(6)	Non	Primar y	Clave para identificar o módulo
horasSemanais	tinyint (sen signo)			Descrición do nome do grupo
descripcion	varchar(100)			Nº de horas de redución para o profesor titor
corpo	enum	Non		Corpo dos profesores que poden impartir o módulo. Valores permitidos: S = Profesorado Secundaria T = Profesorado Técnico

- Táboa *imparte* (clave primaria composta)

Nome	Tipo	Null	Clave	Observacións
idModulo	char(6)	Non	Primar y	Clave para identificar o módulo
idGrupo	char(15)	Non		Clave para identificar o grupo
idProfesor	char(15)	Non		Nome curto do profesor

- Tarefa 4.2. Definir as relacións entre esas táboas tendo en conta o seguinte grafo relacional.



Solución

- Tarefa 4.1

```

/*borrado da base de datos, se existe (só cando se están facendo probas)*/
drop database if exists eleccionModulos;
/*creación da base de datos se non existe*/
create database if not exists eleccionModulos
    character set utf8
    collate utf8_spanish_ci;
/* ou tamén */
create schema if not exists eleccionModulos
    character set utf8
    collate utf8_spanish_ci;
/*activar vase de datos*/
use eleccionModulos;
/* TÁBOA profesor */
create table profesor (
    idProfesor char(15) not null,
    corpo enum('S','T') not null,
    nrp char(20) not null,
    nome varchar(30) not null,
    apellidos varchar(60) not null,
    horasLectivas smallint unsigned default 18,
    primary key (idProfesor),
    unique key (nrp)
)engine=InnoDB;
/* TÁBOA grupo*/
create table grupo (
    idGrupo char(15),
    descripcion varchar(100) not null,
    horasTitoria smallint unsigned,
    idProfesor char(15),
    primary key (idGrupo)
)engine=InnoDB;
/* TÁBOA modulo*/
create table modulo (
    idModulo char(6),
    horasSemanais smallint unsigned,
    descripcion varchar(100) not null,
    corpo enum('S','T'),
    primary key (idModulo)
)engine=InnoDB;
/* TÁBOA imparte */
create table imparte (
    idModulo char(6),
    idGrupo char(15),

```

```
idProfesor char(15),
primary key (idGrupo,idModulo,idProfesor)
)engine=InnoDB;
```

■ Tarefa 4.2

```
/* Definición das relacións entre as táboas: RESTRICIÓN DE CLAVE FORÁNEA */
alter table grupo
add constraint fk_grupo_profesor foreign key (idprofesor)
references profesor (idprofesor)
on delete restrict
on update cascade;
alter table imparte
add constraint fk_imparte_modulo foreign key (idmodulo)
references modulo (idmodulo)
on delete restrict
on update cascade,
add constraint fk_imparte_profesor foreign key (idprofesor)
references profesor (idprofesor)
on delete restrict
on update cascade,
add constraint fk_impate_grupo foreign key (idgrupo)
references grupo (idgrupo)
on delete restrict
on update cascade;
```

1.2.5 Tarefa 5: Tarefa integradora sobre LDD

A tarefa consiste en realizar as tarefas integradoras nos seguintes supostos:

- Tarefa 5.1. Na táboa *pelicula* da base de datos *practicasl* hai unha columna chamada *xenero* que garda o xénero da película. Decídese empregar un sistema de codificación de xéneros cinematográficos.

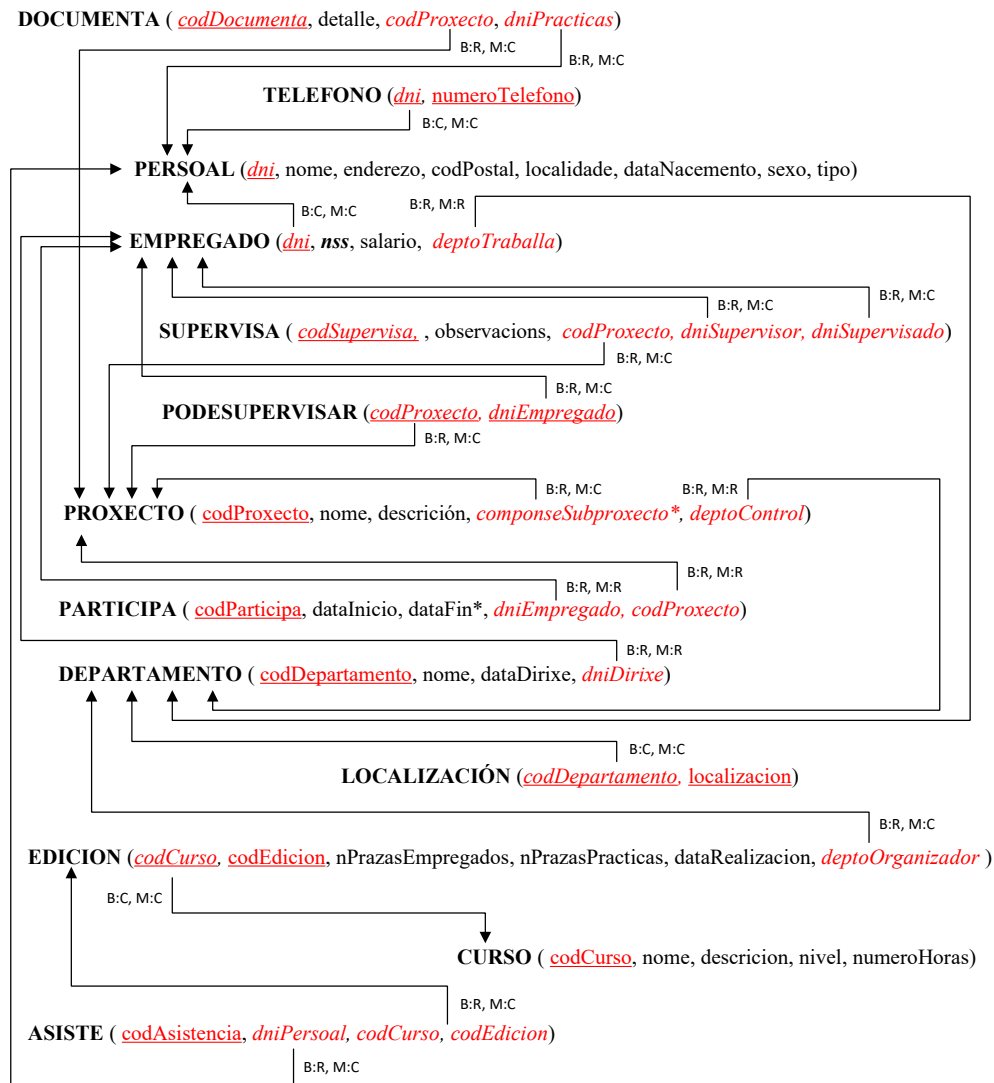
Se pide: crear unha táboa co sistema de codificación de xéneros, e modificar a columna *xenero* para utilizala como unha clave foránea que fai referencia a esa táboa.

A táboa de xéneros terá tres columnas, unha co código numérico (actualmente hai uns 35 códigos recoñecidos), outra cunha descrición curta, e outra cunha descrición máis detallada do xénero.

- Tarefa 5.2. Cambiar a clave primaria na táboa de *empregado* da base de datos *practicasl*. Actualmente é a columna *dni*, e se quere cambiar pola columna *nss*.

Hai que ter en conta que esa clave primaria é referenciada pola clave foránea *dniXefe* da táboa *departamento*, polo que será necesario facer unha serie de operacións previas ao cambio de clave primaria. Na táboa *departamento* xa non interesa ter o *dni* do xefe, senón o seu número de seguridade social (*nss*).

- Tarefa 5.3. Modificar a estrutura da base de datos *proxectos* (creada na actividade anterior), engadindo as columnas, e as restricións de integridade referencial necesarias, tendo en conta o seguinte grafo relacional:



Solución

■ Tarefa 5.1

```
use practicas1;
drop table if exists xenero;
/*creacion da táboa xenero*/
create table xenero
(
  codXenero tinyint unsigned zerofill not null,
  descricionBreve char(15) not null,
  descricionLarga varchar(150) not null,
  primary key (codXenero)
);
/*facen as modificacións necesarias na táboa pelicula*/
alter table pelicula
modify column xenero tinyint unsigned zerofill,
add constraint fk_pelicula_xenero foreign key (xenero)
references xenero(codXenero)
on delete restrict
on update cascade;
```

■ Tarefa 5.2

```
/*activación da base de datos practicas1*/
use practicas1;
/*eliminar as claves foráneas que fan referencia á clave primaria que se vai cambiar*/
```

```

alter table departamento
drop foreign key fk_departamento_empregado;
/*cambiar a clave primaria na táboa empregado*/
alter table empregado
drop primary key,
add primary key (nss);
/*cambiar a definición da clave foránea que fai referencia á nova clave primaria
para que pertencen ao mesmo dominio, e volver a crear a clave foránea*/
alter table departamento
change dniXefe nssXefe char(10),
add constraint fk_departamento_empregado foreign key (nssXefe)
references empregado(nss)
on delete set null
on update cascade;

```

No caso de intentar borrar a clave primaria na táboa de empregado antes de borrar as claves foráneas que fan referencia a ela, móstrase a seguinte mensaxe de erro:

```

/*Error Code: 1025. Error en el renombrado de '.\practicass1\#sql-e80_2' a '.\practicass1\empregado' (Error: 150 - Foreign key constraint is incorrectly formed)

```

■ Tarefa 5.3

```

use proxectos;
-- RESTRICIÓN DE INTEGRIDADE REFERENCIAL. ATRIBUTOS MULTIVALUADOS
-- Atributo multivaluado telefono na entidade persoal
alter table telefono
add constraint fk_telefono_persoal foreign key (dni)
references persoal (dni)
on delete cascade
on update cascade;
-- Atributo multivaluado localizacion na entidade departamento
alter table localizacion
add constraint fk_localizacion_departamento foreign key (codDepartamento)
references departamento (codDepartamento)
on delete cascade
on update cascade;

-- RESTRICIÓN DE INTEGRIDADE REFERENCIAL. INTERRELACIÓN 1:N
-- Relación traballa (1:N). Creación da columna e definición de clave foránea
alter table empregado
add column deptoTraballa smallint unsigned,
add constraint fk_empregado_departamento foreign key (deptoTraballa)
references departamento (codDepartamento)
on delete restrict
on update restrict;
-- Relación organiza (1:N). Creación da columna e definición de clave foránea
-- edición é entidade débil: Defición de clave foránea coa entidade forte
alter table edicion
add column deptoOrganizador smallint unsigned not null,
add constraint fk_edicion_curso foreign key (codCurso)
references curso (codCurso)
on delete cascade
on update cascade,
add constraint fk_edicion_departamento foreign key (deptoOrganizador)
references departamento (codDepartamento)
on delete restrict
on update cascade;
-- Relación controla (1:N). Creación da columna e definición de clave foránea
-- Relación componse (1:N). Creación da columna e definición de clave foránea
alter table proxecto
add column deptoControl smallint unsigned not null,

```

```

add column compenseSubproxecto smallint unsigned not null,
add constraint fk_proxecto_departamento foreign key (deptoControl)
    references departamento (codDepartamento)
    on delete restrict
    on update restrict,
add constraint fk_proxecto_proxecto foreign key (compenseSubproxecto)
    references proxecto (codProxecto)
    on delete restrict
    on update cascade;

-- RESTRICIÓN DE INTEGRIDADE REFERENCIAL. INTERRELACIÓNS 1:1
-- Relación dirixe (1:1). Creación da columna e definición de clave foránea
alter table departamento
add column dniDirixe integer(8) unsigned not null,
add column dataDirixe date not null,
add constraint fk_departamento_empregado foreign key (dniDirixe)
    references empregado (dni)
    on delete restrict
    on update restrict;

-- RESTRICIÓN DE INTEGRIDADE REFERENCIAL. INTERRELACIÓNS N:M
-- Relación asiste (N:M). Definición de claves foráneas
alter table asiste
add constraint fk_asiste_persoal foreign key (dniPersoal)
    references persoal (dni)
    on delete restrict
    on update cascade,
add constraint fk_asiste_edición foreign key (codCurso,codEdicion)
    references edicion (codCurso,codEdicion)
    on delete restrict
    on update cascade;

-- Relación participa (N:M). Definición de claves foráneas
alter table participa
add constraint fk_participa_proxecto foreign key (codProxecto)
    references proxecto (codProxecto)
    on delete restrict
    on update restrict,
add constraint fk_participa_empregado foreign key (dniEmpregado)
    references empregado (dni)
    on delete restrict
    on update restrict;

-- Relación documenta (N:M). Definición de claves foráneas
alter table documenta
add constraint fk_documenta_proxecto foreign key (codProxecto)
    references proxecto (codProxecto)
    on delete restrict
    on update cascade,
add constraint fk_documenta_persoal foreign key (dniPracticas)
    references persoal (dni)
    on delete restrict
    on update cascade;

-- Relación podesupervisar (N:M). Definición de claves foráneas
alter table podesupervisar
add constraint fk_podesupervisar_proxecto foreign key (codProxecto)
    references proxecto (codProxecto)
    on delete restrict
    on update cascade,
add constraint fk_podesupervisar_empregado foreign key (dniEmpregado)
    references empregado (dni)
    on delete restrict
    on update cascade;

```

```

-- RESTRICIÓN DE INTEGRIDADE REFERENCIAL. INTERRELACIÓN 1:N:M
-- Relación supervisa (1:N:M). Definición de claves foráneas
alter table supervisa
add constraint fk_supervisa_proxecto foreign key (codProxecto)
references proxecto (codProxecto)
on delete restrict
on update cascade,
add constraint fk_supervisa_empregado_supervisor foreign key (dniSupervisor)
references empleado (dni)
on delete restrict
on update cascade,
add constraint fk_supervisa_empregado_supervisado foreign key (dniSupervisado)
references empleado (dni)
on delete restrict
on update cascade;

-- ESPECIALIZACIÓN:
alter table persoal
add column tipo enum('practicass','empleado') not null;
alter table empleado
add constraint fk_empleado_persoal foreign key (dni)
references persoal (dni)
on delete restrict
on update restrict;

/* INCLUSION. Para esta restricción pódense utilizar disparadores (triggers), asercións,
restriccións CHECK,ou definir claves foráneas. Como MySQL non ten implementada a cláusula
CHECK, e os disparadores e asercións estúdanse mais adiante, nesta actividade utilízase a
última das opcións (claves foráneas)*/

-- Relacións dirixe e traballa (definindo claves foráneas)
-- Comprobar que o empleado que dirixe o departamento traballa nese departamento.
alter table empleado
add index idx_departamento_dni_codDepartamento (dni,deptoTraballa);
alter table departamento
add constraint fk_departamento_empleado_traballa
foreign key (dniDirixe,codDepartamento)
references empleado (dni,deptoTraballa)
on delete restrict
on update restrict;

-- Relacións supervisa e podesupervisar (definindo claves foráneas)
-- Comprobar que o empleado que supervisa o proxecto pode supervisalo.
alter table podesupervisar
add index idx_podesupervisar_codProxecto_dniEmpleado (codProxecto,dniEmpleado);
alter table supervisa
add constraint fk_supervisa_podesupervisar foreign key (codProxecto,dniSupervisor)
references podesupervisar (codProxecto,dniEmpleado)
on delete restrict
on update restrict;

```