

Boletín de ejercicios sobre Deseño Físico

1. Crea unha base de datos de nome “*practicass2*” empregando o xogo de caracteres latin1 e o idioma (colación) español.

```
-- Creación da base de datos practicas2
create database if not exists practicas2
character set latin1
collate latin1_spanish_ci;
/* ou tamén */
create schema if not exists practicas2
default character set latin1
collate latin1_spanish_ci ;
```

2. Poñer en uso a base de datos creada no exercicio anterior.

```
use practicas2
```

3. Borrar a base de datos “*practicass2*”.

```
drop database if exists practicas2;
```

4. Crea as seguintes táboas:

- a. Crea a base de datos “*practicass1*”, empregando o sistema de codificación utf8 en español e crear nesa base de datos a táboa fabricante co motor de almacenamento MyISAM. Debes poñer en uso a base de datos “*practicass1*” antes de crear as taboas A estrutura da táboa é a seguinte:

Nome columna	Tipo	Null	Clave	Observacións
idFabricante	char(5)	N	Primaria	Código que identifica a cada fabricante
nome	varchar(80)	N		

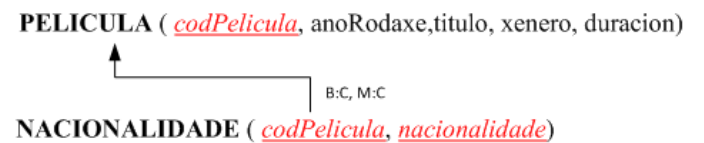
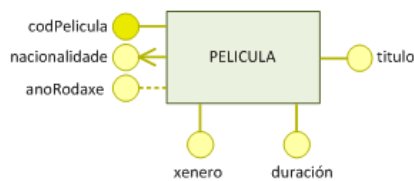
```
/*creación da base de datos practicas1*/
create database if not exists practicas1
default character set utf8
default collate utf8_spanish_ci;
/*Se podería crear a táboa sen necesidade de activar a
base de datos, utilizando a notación nomeBd.nomeTaboa: */
drop table if exists practicas1.fabricante;
create table practicas1.fabricante # nomeBd.nomeTaboa
(idFabricante char(5) not null,
nome varchar(80) not null,
primary key (idFabricante)
) engine = MyISAM;
Outra forma de facelo:
/*creación da base de datos practicas1 utilizando a opción schema*/
create schema if not exists practicas1
default character set utf8
default collate utf8_spanish_ci;
```

```

/*Activar a base de datos practicas1 */
use practicas1;
/*Borrar a táboa no caso de que exista coa sentença DROP e creala coa sentença CREATE
As operacións fanse sobre a base de datos activa, que é practicas1 */
drop table if exists fabricante;
create table fabricante
(idFabricante char(5) not null,
nome varchar(80) not null,
primary key (idFabricante)
) engine = MyISAM;

```

- b. Crear na base de datos practicas1 as táboas transacionais correspondentes á interpretación do seguinte diagrama E/R e ao seu grafo relacional:



Terase en conta que:

- O atributo *nacionalidade* é un atributo multivaluado da entidade pelicula.
- O xogo de caracteres para almacenar os datos nas columnas destas táboas será latin1 e a colación en español.
- O atributo *codPelicula* será un valor numérico que se irá incrementando, de forma automática, nunha unidade cada vez que insira unha nova fila na táboa *pelicula*.
- O atributo *titulo* ocupa como máximo 100 caracteres, e non admitirá valores nulos. Os valores desta columna almacenaranse utilizando o xogo de caracteres latin1 e o sistema de colación general.
- O atributo *duración* gardarase como unha cantidade numérica positiva, con dous dí-xitos enteiros e dous decimais.

```

/*activación da base de datos practicas1*/
use practicas1;
/*Creación da táboa pelicula, se non existe*/
create table if not exists pelicula
(codPelicula smallint unsigned auto_increment not null,
anoRodaxe date null,
titulo varchar(100) character set latin1 collate latin1_general_ci not null,
xenero varchar(30) not null,
duracion decimal(4,2) unsigned not null,
primary key (codPelicula)
) engine = InnoDB
default character set latin1 default collate latin1_spanish_ci;
/*Creación da táboa nacionalidade, se non existe*/
create table if not exists nacionalidade
(codPelicula smallint unsigned not null,
nacionalidade varchar(30) not null,
primary key (codPelicula,nacionalidade),
constraint fk_nacionalidade_pelicula foreign key (codPelicula)
references pelicula(codPelicula)
)

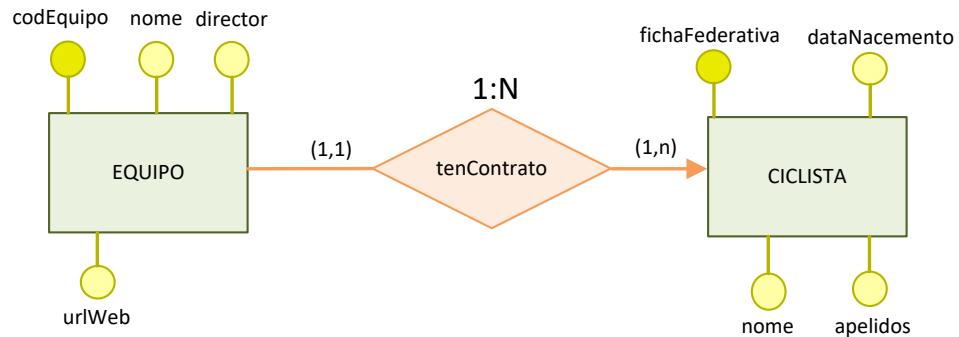
```

```

on delete restrict
on update cascade
)      engine = Innodb
      default character set latin1 default collate latin1_spanish_ci;

```

- c. Crear na base de datos practicas1, as táboas que correspondan ao seguinte diagrama E/R, interpretando a relación de tipo 1:N.



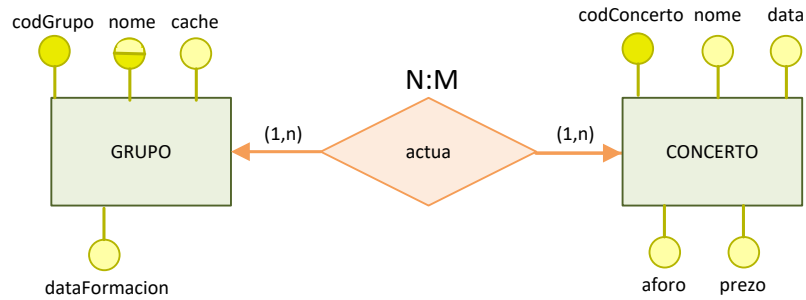
Terase en conta que o código do equipo está composto por unha combinación de letras e números cun tamaño fixo de 9 caracteres

```

/*activación da base de datos practicas1*/
use practicas1;
/*Creación da táboa equipo, se non existe*/
create table if not exists equipo
(
codEquipo char(9) not null,
nome varchar(50) not null,
director varchar(80) not null,
urlWeb varchar(150) not null,
primary key (codEquipo)
)      engine = Innodb;
/*Creación da táboa ciclista, se non existe*/
create table if not exists ciclista
(
fichaFederativa mediumint unsigned not null,
dataNacemento date not null,
nome varchar(40) not null,
apelidos varchar(80) not null,
codEquipo char(9) not null, # not null: cardinalidade mínima 1
primary key (fichaFederativa),
constraint fk_ciclista_equipo foreign key (codEquipo)
references equipo(codEquipo)
on delete restrict
on update cascade
)      engine = Innodb;

```

- d. Crear na base de datos practicas1, as táboas que correspondan ao seguinte diagrama E/R, interpretando a relación de tipo N:M.



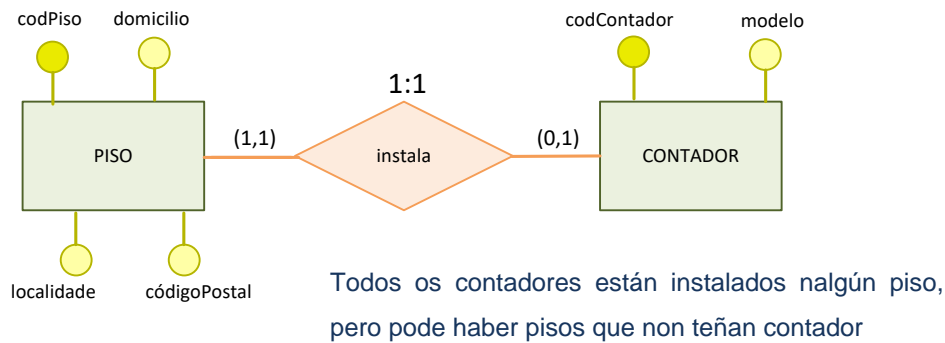
Terase en conta que:

- O caché do grupo é o mesmo para todos os concertos e suponse que non pode ser superior a 5 millóns de euros.
- Para facilitar as buscas debe crearse un índice para a columna nome da táboa grupo.
- O aforo representa o número máximo de espectadores que poden asistir ao concerto e como máximo será 100000 persoas.
- O prezo das entradas ven dado en euros con dous decimais e o prezo máximo é 200 euros.

```

/*activación da base de datos practicas1*/
use practicas1;
/*Creación da táboa grupo, se non existe*/
create table if not exists grupo
(
codGrupo smallint unsigned not null,
nome varchar(80) not null,
cache mediumint unsigned not null,
dataFormacion date not null,
primary key (codGrupo),
index idx_grupo_nome (nome)
) engine = InnoDB;
/*Creación da táboa concerto, se non existe*/
create table if not exists concerto
(
codConcerto mediumint unsigned not null,
nome varchar(100) not null,
data date not null,
aforo mediumint unsigned not null comment 'numero máximo de espectadores',
precio decimal(5,2) not null comment 'prezo da entrada',
primary key (codConcerto)
) engine = InnoDB;
/*Creación da táboa actuacion que sae da transformación
da relación N:M nunha nova táboa*/
create table if not exists actuacion
(
codConcerto mediumint unsigned not null,
codGrupo smallint unsigned not null,
primary key (codConcerto,codGrupo),
constraint fk_actuacion_concerto foreign key (codConcerto)
references concerto(codConcerto)
on delete restrict
on update cascade,
constraint fk_actuacion_grupo foreign key (codGrupo)
references grupo(codGrupo)
on delete restrict
on update cascade
) engine = InnoDB;
  
```

- e. Crear na base de datos practicas1, as táboas que correspondan ao seguinte diagrama E/R, interpretando a relación de tipo 1:1.



Terase en conta que só existen tres modelos de contadores que se identifican como 'a-eg235', 'samsung20', e 'aeg55'.

```

/*activación da base de datos practicas1*/
use practicas1;
/*Creación da táboa piso, se non existe*/
create table if not exists piso
(
    codPiso smallint unsigned not null,
    domicilio varchar(150) not null,
    localidade varchar(80) not null,
    codPostal char(5) not null,
    primary key (codPiso)
) engine = InnoDB;
/*Creación da táboa contador, se non existe*/
create table if not exists contador
(
    codContador mediumint unsigned auto_increment not null,
    modelo enum('aeg235','samsung20','aeg55') not null,
    codPiso smallint unsigned not null, # not null: cardinalidade mínima 1
    primary key (codContador),
    constraint fk_contador_piso foreign key (codPiso)
        references piso (codPiso)
        on delete restrict
        on update cascade
) engine = InnoDB;
  
```

5. Borrar la tabla fabricante creada en el ejercicio 4.a. Escribe la sentencia que necesitarías para borrar la tabla si la base de datos *practicas1* no estuviera en uso

```
use practicas1;
```

```
drop table if exists fabricante;
```

Tamén se podería executar nunha única sentença, cualifinaco o nome da táboa

```
drop table if exists practicas1.fabricante;
```

6. Supoñendo que a columna nome da táboa **concerto** da base de datos *practicas1* utilízase frecuentemente como condición para localizar os datos dun concerto determinado, pídese crear un índice asociado a esta columna para que esas consultas tarden menos tempo en executarse.

```

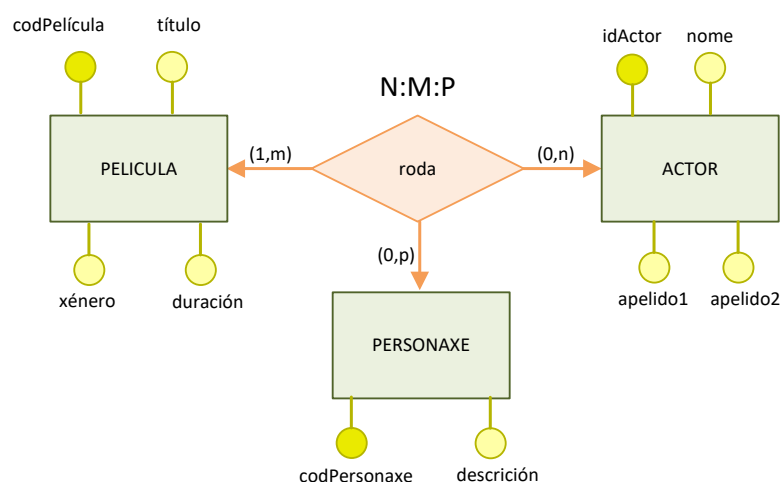
create index idx_concerto_nome
on concerto (nome);
  
```

7. Supoñendo que có paso do tempo chegouse á conclusión de que era mellor eliminar o índice asociado á columna nome da táboa concerto da base de datos practicas1. Pídese escribir a sentenza SQL que permite borrar ese índice.

```
drop index idx_ concerto_nome on concerto;
```

8. Escribir un script de sentenzas SQL para realizar as seguintes operacións na base de datos practicas1:

- Borrar a táboa película e todas as relacionadas con ela.
- Crear as táboas que correspondan ao seguinte diagrama E/R, interpretando a seguinte relación ternaria.



- Crear un índice asociado á columna título da táboa película, e outro índice composto polas columnas apelido1, apelido2 e nome da táboa actor.
- Borrar o índice composto creado no apartado anterior.
- Borrar a táboa roda.

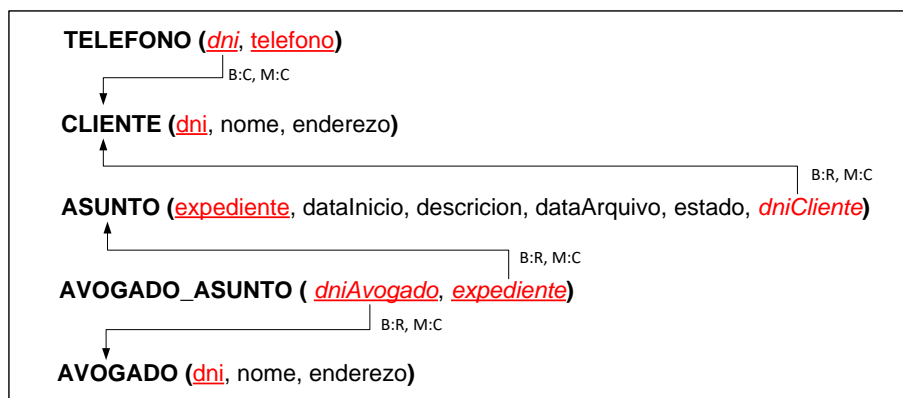
```
/*activación da base de datos practicas1*/
use practicas1;
/*borrado da táboa película e táboas relacionadas*/
drop table if exists nacionalidade;
drop table if exists pelicula;
/*Creación da táboa película, se non existe*/
create table if not exists pelicula
(
codPelícula smallint unsigned auto_increment not null,
título varchar(100) not null,
xenero varchar(30),
duracion decimal(4,2) unsigned,
primary key (codPelícula)
) engine = Innodb ;
/*Creación da táboa actor, se non existe*/
create table if not exists actor
(
idActor mediumint unsigned not null,
nome varchar(40) not null,
apelido1 varchar(40),
apelido2 varchar(40),
primary key (idActor)
) engine = Innodb;
```

```

/*Creación da táboa personaxe, se non existe*/
create table if not exists personaxe
(
codPersonaxe char(10) not null,
descripcion varchar(40) not null,
primary key (codPersonaxe)
) engine = InnoDB;
/*Creación da táboa roda que sae da transformación
da relación ternaria N:M:P nunha nova táboa, se non existe*/
create table if not exists roda
(
codRodaxe smallint unsigned auto_increment not null,
codPelicula smallint unsigned not null,
idActor mediumint unsigned not null,
codPersonaxe char(10) not null,
primary key (codRodaxe),
constraint fk_roda_pelicula foreign key (codPelicula)
references pelicula(codPelicula)
on delete restrict
on update cascade,
constraint fk_roda_actor foreign key (idActor)
references actor(idActor)
on delete restrict
on update cascade,
constraint fk_roda_personaxe foreign key (codPersonaxe)
references personaxe(codPersonaxe)
on delete restrict
on update cascade
) engine = InnoDB;
/*creación do índice composto*/
create index idx_actor_apellido1_apellido2_nombre on actor (apellido1,apellido2,nome);
/*borrado do índice composto*/
drop index idx_actor_apellido1_apellido2_nombre on actor;
/*borrado da táboa roda*/
drop table roda;

```

9. Crear a base de datos gabinete, tendo en conta o seguinte grafo relacional:



```

/*BORRADO DA BASE DE DATOS, NO CASO QUE EXISTA
(só cando se están facendo probas)*/
drop database if exists gabinete;

```

```

/* CREACIÓN DA BASE DE DATOS*/
create database if not exists gabinete
    character set utf8
    collate utf8_spanish_ci;
/* ou tamén */
create schema if not exists gabinete
    default character set utf8
    collate utf8_spanish_ci ;

/* ACTIVACIÓN DA BASE DE DATOS */
use gabinete;

/* CREACIÓN DAS TÁBOAS */
create table cliente (
    dni char(9) not null,
    nome varchar(100) not null,
    enderezo varchar(150) not null,
    primary key (dni),
    key nome (nome)
) engine=innodb ;

create table telefono (
    dni char(9) not null,
    telefono char(12) not null,
    primary key (dni,telefono),
    constraint fk_telefono_cliente foreign key (dni)
        references cliente (dni)
        on delete cascade
        on update cascade
) engine=innodb ;

create table avogado (
    dni char(9) not null,
    nome varchar(100) not null,
    enderezo varchar(150) not null,
    primary key (dni)
) engine=innodb ;

create table asunto (
    expediente int(10) unsigned not null auto_increment,
    dataInicio datetime not null default current_timestamp,
    descricion text not null,
    dataArquivo date default null,
    estado enum('recopilación','tramite','xuizo','arquivado') not null,
    dniCliente char(9) not null,
    primary key (expediente),
    key dniCliente (dniCliente),
    constraint fk_asunto_cliente foreign key (dniCliente)
        references cliente (dni)

```



```
        on delete restrict  
        on update cascade  
    ) engine=innodb ;
```

```
create table avogado_asunto (  
    dniAbogado char(9) not null,  
    expediente int(10) unsigned not null,  
    primary key (dniAbogado,expediente),  
    key expediente (expediente),  
    constraint fk_avogado_asunto_avogado foreign key (dniAbogado)  
        references avogado (dni)  
        on delete restrict  
        on update cascade,  
    constraint fk_avogado_asunto_asunto foreign key (expediente)  
        references asunto (expediente)  
        on delete restrict  
        on update cascade  
    ) engine=innodb ;
```