

VideoClub Glob-Guster



Índice

1. Presentación del Problema.	3
2. Desglose del problema.....	3
3. Modelo Entidad/Relación.....	4
4. Paso a tabla.	4
5. Modelo Físico.	5
6. Datos.	6
7. Consultas, Procedimientos y Funciones.....	9

1. Presentación del Problema.

La cadena de Videoclubs Glob-Gusters ha decidido, para mejorar su servicio, emplear una base de datos para almacenar la información referente a las películas que ofrece en alquiler. Esta información es la siguiente:

- Una película se caracteriza por su título, nacionalidad, productora y fecha. Ej. Quo Vadis, Estados Unidos, M.G.M, 1955.
- En una Película pueden participar varios actores (nombre, nacionalidad, sexo), algunos de ellos como actores principales.
- Una película está dirigida por un director (nombre, nacionalidad).
- De cada película se dispone de uno o varios ejemplares diferenciados por un número de ejemplar y caracterizados por su estado de conservación.
- Un ejemplar se puede encontrar alquilado a algún cliente (nombre, dirección, teléfono). Se desea almacenar la fecha de comienzo del alquiler y la devolución.
- Cada socio puede alquilar como máximo 4 ejemplares. (Suponemos que a la vez).
- Un Socio tiene que ser avalado por otro socio que responda de él en caso de tener problemas con el alquiler.

2. Desglose del problema.

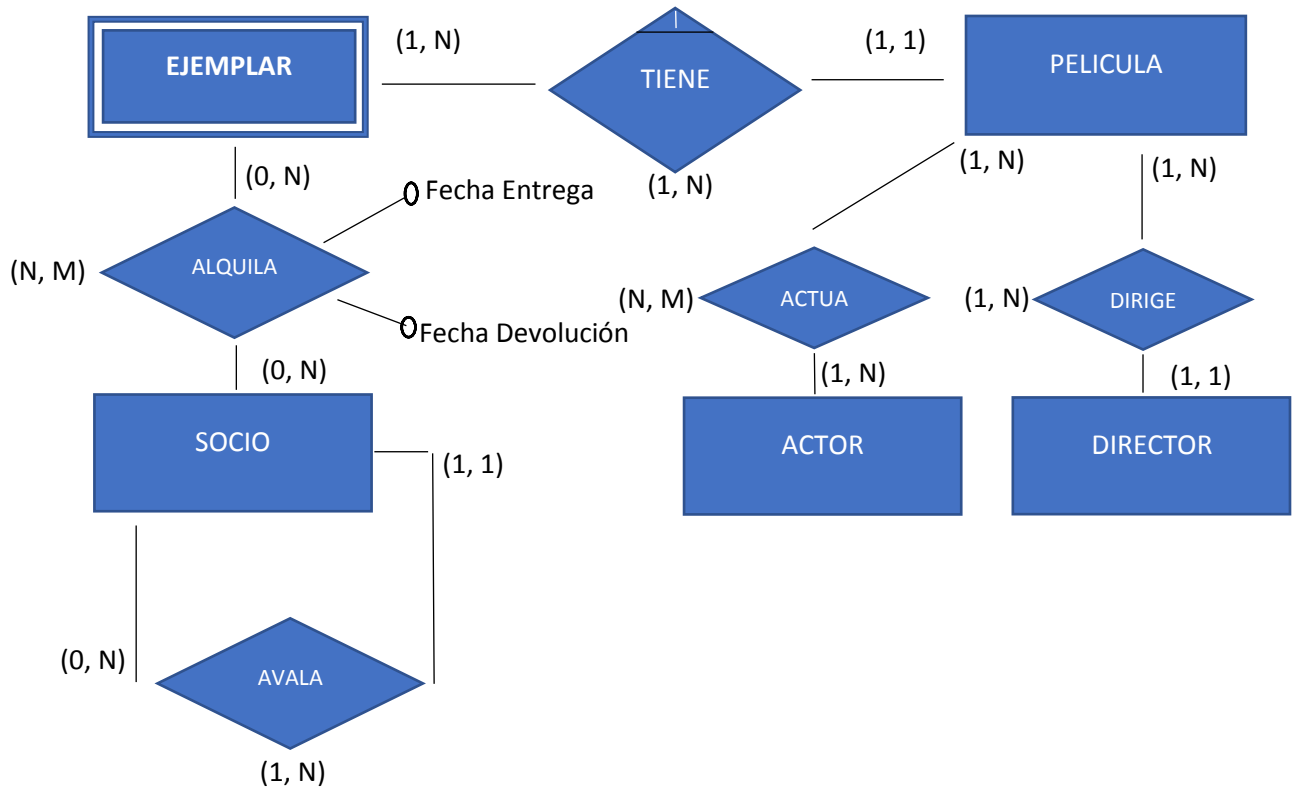
Podemos deducir que se deberá de almacenar:

- **Películas:** Título, Nacionalidad, productora y fecha.
- **Actores:** Nombre, Nacionalidad, Sexo.
- **Directores:** Nombre, Nacionalidad.
- **Ejemplares:** Nº de ejemplar, estado.
- **Socios:** ID Socio, nombre, Dirección, Teléfono.
- **Alquileres:** Socio, Fecha de Alquiler, Fecha de Devolución.

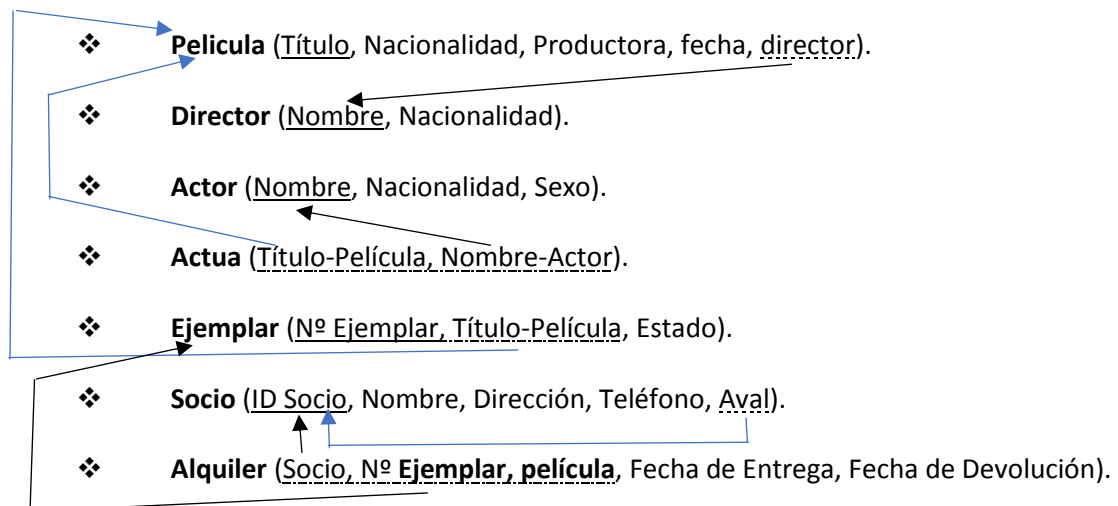
Con las siguientes restricciones:

- Cada Socio puede alquilar como **máximo 4 ejemplares** (suponemos que a la vez.).
- Un Socio **tiene que ser avalado** por otro socio.

3. Modelo Entidad/Relación.



4. Paso a tabla.



Leyenda:

- = Primary Key.
- = Foreign Key.
-: = Primary Key y Foreign Key.
- **Sdasd, dasdasd** = Misma clave.

5. Modelo Físico.

```

create table pelicula(
    titulo varchar(90) primary key,
    nacionalidad varchar(15) not null,
    productora varchar(15),
    ndirector varchar(30) foreign key references director
        on delete cascade on update cascade not null,
    fecha date
)

create table actor(
    nombre varchar(30) primary key,
    nacionalidad varchar(15) not null,
    sexo char check (sexo in ('f','m')) not null
)

create table director(
    nombre varchar(30) primary key,
    nacionalidad varchar(15) not null
)

create table actua(
    tpelicula varchar(90) foreign key references pelicula on delete cascade on
update cascade,
    nactor varchar(30) foreign key references actor on delete cascade on update
cascade
    constraint PK_actua primary key(tpelicula, nactor)
)

create table ejemplar(
    nejemplar tinyint identity(1,1) not null,

```

```

        npelicula varchar(90) foreign key references pelicula on delete cascade on
update cascade,
        estado varchar(9) check (estado in ('malo', 'regular', 'bueno', 'excelente'))
        constraint PK_ejemplar primary key(nejemplar, npelicula)
    )

```

```

create table socio(
    idsocio int identity(1,1) primary key,
    nombre varchar(30) not null,
    direccion varchar(30) not null,
    telefono int check (telefono>999999999 and telefono<10000000000)not null,
    aval int foreign key references socio not null
)

```

```

create table alquilado(
    idsocio int foreign key references socio,
    ejemplar tinyint,
    pelicula varchar(90),
    fechaentrega date,
    fechadevolucion date
    constraint FK_ejemplar foreign key(ejemplar, pelicula) references ejemplar
        on delete cascade on update cascade,
    constraint PK_alquilado primary key (idsocio, ejemplar, pelicula)
)

```

6. Datos.

-- Directores

```

insert into director values ('Steve Bendelack', 'Britanico'), ('Alfonso Cuaron',
'Mexicano'),('Tim Burton', 'Estadounidense'), ('James Cameron', 'Canadiense'),
('Peter Jackson', 'Nuevozelandes'), ('Woody Allen', 'Estadounidense'),('Pedro
Almodobar', 'Español'),
('Luc Besson', 'Frances'), ('Sam Mendes', 'Britanico'), ('Mario de la Torre', 'Español')

```

-- Actores

insert into actor values ('Ian McKellen', 'Britanico', 'M'), ('Inma Cuestas', 'Española', 'F'), ('Maria Jimenez', 'Española', 'F'), ('Jim Carrey', 'Canadiense', 'M'), ('Rowan atkinson', 'Britanico', 'M'), ('Enma Watson', 'Britanica', 'F'), ('Robbie Coltrane', 'Escoces', 'M'), ('Michael Gambon', 'Irlandes', 'M'), ('Johnny Depp', 'Estadounidense', 'M'), ('Zoe zaldaña', 'Estadounidense', 'F')

--Películas

insert into pelicula values ('Charlie y la fabrica de Chocolate', 'Estadounidense', 'Plan B', 'Tim Burton', '2005-08-15'), ('Harry Potter: y el prisionero de Azkaban', 'Britanica', 'Heyday Films', 'Alfonso Cuaron', '2004-06-04'), ('La Condena', 'Española', 'Creta Pr', 'Mario de la Torre', '2009-10-12'), ('Las Vacaciones de Mr. Beans', 'Francesa', 'Gainsborough', 'Steve Bendelack', '2007-01-01'), ('Valerian y la ciudad de los mil planetas', 'Francesa', 'Belga Films', 'Luc Besson', '2017-08-26'), ('Julieta', 'Española', 'El Deseo', 'Pedro Almodobar', '2016-04-08'), ('El Hobbit: La desolacion de Smaug', 'Estadounidense', 'New Line Cinema', 'Peter Jackson', '2013-12-13'), ('El Señor de los anillos: Las dos torres', 'Nuevazelandes', 'WingNut Films', 'Peter Jackson', '2002-12-18')

--Ejemplares

insert into ejemplar values ('El Hobbit: La desolacion de Smaug', 'Excelente'), ('El Señor de los anillos: Las dos torres', 'Regular'), ('Harry Potter: y el prisionero de Azkaban', 'bueno'), ('El Señor de los anillos: Las dos torres', 'malo'), ('Las Vacaciones de Mr. Beans', 'bueno'), ('Las Vacaciones de Mr. Beans', 'bueno'), ('La Condena', 'excelente'),

('Charlie y la fabrica de Chocolate', 'regular')

--actua

insert into actua values ('El Señor de los anillos: Las dos torres', 'Ian McKellen'),

('El Hobbit: La desolacion de Smaug', 'Ian McKellen'),

('Charlie y la fabrica de Chocolate', 'Johnny Depp'),

('Harry Potter: y el prisionero de Azkaban', 'Enma Watson'),

('Harry Potter: y el prisionero de Azkaban', 'Robbie Coltrane'),

('Harry Potter: y el prisionero de Azkaban', 'Michael Gambon'),

('La Condena', 'Maria Jimenez'),

('Julieta', 'Inma Cuestas'),

('Las Vacaciones de Mr. Beans', 'Rowan atkinson')

--Socios

insert into socio values ('Jose Maria', 'C/nevas, 12', 635897454, 1),

('Mario', 'c/ falsa, 28', 568365789, 1),

('cristian', 'c/ algo, 6', 123456789, 1),

('maria', 'c/ bloque, 3', 789456123, 3),

('manu', 'c/ gondola, 5', 456789312, 2),

('antonio', 'c/ mac, 7', 852741963, 1),

('dani', 'c/ sierra, 20', 741852963, 1),

('maria', 'c/ para, 9', 564231897, 2)

--alquilado

--NOTA: la restricción de máximo 4 no la hemos contemplado

insert into alquilado values (1, 1, 'El Hobbit: La desolacion de Smaug', '2018-02-11', '2018-02-13'),

(1, 3, 'Harry Potter: y el prisionero de Azkaban', '2018-02-13', '2018-02-15'),

(1, 6, 'Las Vacaciones de Mr. Beans', '2018-02-13', '2018-02-15'),

(1, 8, 'Charlie y la fabrica de Chocolate', '2018-02-13', '2018-02-15'),

(2, 2, 'El Señor de los anillos: Las dos torres', '2018-02-11', '2018-02-13'),
 (3, 5, 'Las Vacaciones de Mr. Beans', '2018-02-11', '2018-02-13'),
 (3, 8, 'Charlie y la fabrica de Chocolate', '2018-02-11', '2018-02-13'),
 (4, 8, 'Charlie y la fabrica de Chocolate', '2018-02-10', '2018-02-11')

7. Consultas, Procedimientos y Funciones.

--Muestra los datos de las peliculas disponibles

```
select * from pelicula where titulo in (select npelicula from ejemplar)
```

--Muestra la cantidad de ejemplares ordenados por estado

```
select estado, Count(*) as 'ejemplares' from ejemplar group by estado
```

--Muestra peliculas por nacionalidades ordenado de mas a menos

```
select nacionalidad, COUNT(*) as 'peliculas' from pelicula group by nacionalidad order by peliculas desc
```

--Muestra los referidos de los socios (cantidad de personas que han avalado)

```
select aval as 'ID Socio', COUNT(aval) as 'Referidos' from socio group by aval
```

--Muestra todos los datos de las peliculas con más de un ejemplar

```
select titulo, nacionalidad, productora, ndirector, fecha, COUNT(*) as 'ejemplares'
from ejemplar, pelicula group by npelicula, titulo, nacionalidad, productora, ndirector,
fecha
having titulo=npelicula and COUNT(*)>1
```

--una vista que Muestra el numero total del peliculas, actores, directores, ejemplares, socios

```
create view cantidadDeDatos as(
```

```
select (select COUNT(*) from pelicula) as 'Peliculas', (select COUNT(*) from ejemplar)
as 'Ejemplares', (select COUNT(*) from actor) as 'Actores', (select COUNT(*) from
director) as 'Directores', (select COUNT(*) from socio) as 'Socios', (select COUNT(*)
from alquilado) as 'Alquileres')
```

--NO se permiten tablas temporales en las vistas

```
select * from cantidadDeDatos
```

```
--Tabla temporal para las peliculas no devueltas
```

```
select idsocio, ejemplar, pelicula, fechaentrega into #nodevueltas from alquilado
where fechadevolucion is null
```

```
select * from #nodevueltas
```

```
drop table #nodevueltas
```

```
--Funcion que te dice las peliculas de director
```

```
create function peliculade(@director varchar(30))
```

```
returns varchar(200)
```

```
as begin
```

```
    declare @s varchar(200),
```

```
           @titulo varchar(90)
```

```
    set @s = 'El Director '+@director+' dirige:'
```

```
    declare cu CURSOR for
```

```
        select titulo from pelicula where ndirector = @director
```

```
    open cu
```

```
    fetch cu into @titulo
```

```
    while (@@FETCH_STATUS =0)
```

```
        begin
```

```
            set @s += char(10)+@titulo+', '
```

```
            fetch cu into @titulo
```

```
        end
```

```
    close cu
```

```
    deallocate cu
```

```
    return @s
```

```
end
```

```
print dbo.peliculade('Peter Jackson')
```

--Procedimiento para cambiar estado de pelicula en caso que exista si no existe da error.

```
create procedure cambiodeestado
```

```
    @nejemplar tinyint,
```

```
    @estadonuevo varchar(9)
```

```
as
```

```
    if exists(select * from ejemplar where nejemplar = @nejemplar)
```

```
        begin
```

```
            update ejemplar
```

```
            set estado = @estadonuevo
```

```
            where nejemplar = @nejemplar
```

```
        end
```

```
    else
```

```
        begin
```

```
            print 'El Ejemplar no existe.'
```

```
        end
```

```
select * from ejemplar
```

```
exec cambiodeestado 20, 'regular'
```

```
select * from ejemplar
```