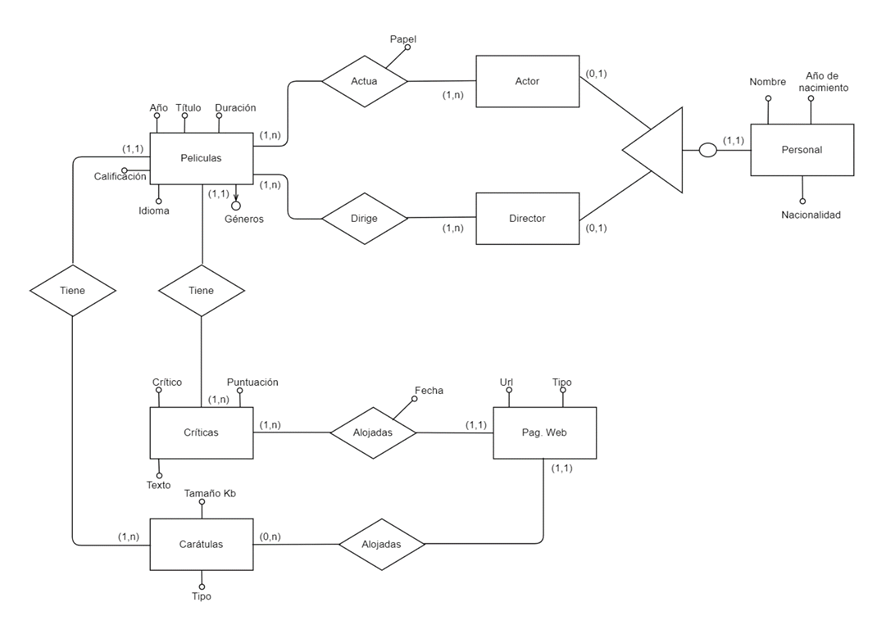
PL2

Bases de datos 2022-2023

Álvaro Camacho Martín DNI: 48083824R

Fátima Cristina de la Morena Rejón DNI: 71961234F





La base de datos dada tiene las siguientes entidades :

Personal

Como atributos tiene el nombre(varchar),el año de nacimiento(int),nacionalidad(varchar) y DNI(varchar),siendo DNI su PK

El personal debe ser Actor o Director

*Director*

*Actor*

Películas

Como atributos tiene duración(int),idioma(varchar),genero(varchar),año(int) y titulo(varchar),estos dos últimos atributos son la PK para esta entidad.

Criticas

Como atributos tiene critico(varchar),Id(int),puntuación(int) y texto(varchar),su PK es Id

Caratula

Como atributos tiene el tipo(varchar) ,tamaño(float) yId(int), su PK es el Id

Web

Como atributos tiene tipo(varchar) y url(varchar),la url será su PK

Relaciones:

alojadas

Con atributo como fecha

Es una relación entre web y crítica.

De uno a muchos.

Alojadas

Es una relación entre web y carátula.

De uno a muchos.

Tiene

Es una relación entre película y crítica.

De uno a muchos.

Tiene

Es una relación entre película y carátula.

De uno a muchos.

Actúa

Es una relación entre actor y película.

De muchos a muchos.

Dirige

Es una relación entre director y película.

De muchos a muchos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entidad/Relación | Atributos | Dominio | Restricción | PK/CK/Discriminante |
| Peliculas | Año | INT |  | PK |
| Titulo | CHAR(40) | [3] | PK |
| Duracion | Float |  |  |
| Idioma | CHAR(10) | [3] |  |
| Calificacion | INT | [1] |  |
| Generos | CHAR(40) | [3] |  |
| Personal | DNI | CHAR(8) | [2] | PK |
| Nombre | CHAR(40) | [3] |  |
| Año de nacimiento | int |  |  |
| Nacionalidad | CHAR(10) | [3] |  |
| Actor |  |  |  |  |
| Director |  |  |  |  |
| Criticas | Id | INT |  | PK |
| Critico | CHAR(40) | [3] |  |
| Puntuacion | INT | [1] |  |
| Texto | CHAR(400) | [3] |  |
| Caratulas | Id | INT |  | PK |
| Tamaño kb | INT |  |  |
| Tipo | CHAR(10) | [3] |  |
| PaginaWeb | Url | CHAR(100) |  | PK |
| Tipo | CHAR(10) | [3] |  |
| alojadas |  |  | 1,n |  |
| Alojadas | Fecha | DATE | 1,n |  |
| tiene |  |  | 1,n |  |
| Tiene |  |  | 1,n |  |
| Actua | Papel | CHAR(20) | 1,n |  |
| Dirige |  |  | 1,n |  |

[1] Es un entero dentro del rango [0-10]

[2] Dividimos el número completo del DNI entre 23, sin sacar decimales. Utilizamos el resto de la división para calcular la letra.

[3] Solo letras y espacio entre palabras, primera de cada palabra en mayúsculas

Modelo relacional

Películas(año(PK), título(PK), idioma, género, calificación, duración, DNI\_director)

Actor(DNI\_personal(PK))

Director(DNI\_personal(PK))

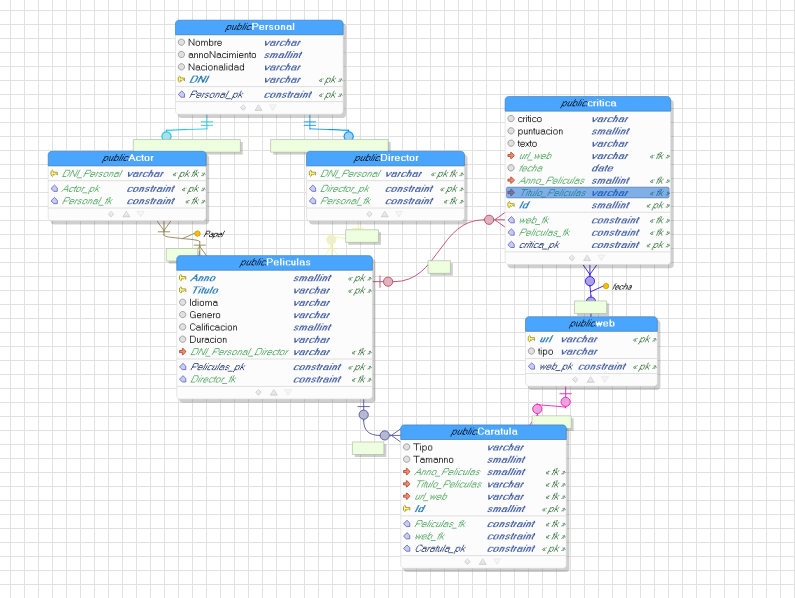
Personal(DNI(PK), nombre, año nacimiento, nacionalidad)

Crítica(id(PK), crítico, puntuación, texto, url\_págWeb(FK), año\_pelicula(FK), título\_película(FK))

Carátula(id(PK), tamaño, tipo, url\_págWeb(FK), año\_pelicula(FK), título\_película(FK))

Pág. Web(url(PK), tipo)

Películas\_Actor(DNI\_actor(FPK), año\_pelicula(FPK), titulo\_pelicula(FPK))



**Comandos usados durante la segunda parte de la PL2**

* Crear tablas intermedias:

CREATE TABLE actores\_pelis\_intermedio(año int, pelicula text, actor text);

CREATE TABLE actores\_ intermedio (nombre text, fecha\_nacimiento date);

CREATE TABLE directores\_ intermedio (nombre text, fecha\_nacimiento date);

CREATE TABLE películas\_ intermedio (año int, titulo text, generos text, puntuacion float, duracion text, idioma text, calificacion text);

CREATE TABLE pelis\_comentarios\_ intermedio (año int, titulo text, puntuacion float, autor text, contenido text, hash text, pagina\_web text);

CREATE TABLE pelis\_directores\_ intermedio (titulo text, año int, nombre text);

* Copiar datos de los CSV a las tablas intermedias:

COPY actores\_pelis\_ intermedio (año, pelicula, actor) from '/Users/alvarocamacho/Desktop/Bideoklub/Ficheros csv/actores\_pelis.csv' csv header delimiter E'\t' NULL 'NULL' encoding 'UTF8';

COPY actores\_ intermedio (nombre, fecha\_nacimiento) from '/Users/alvarocamacho/Desktop/Bideoklub/Ficheros csv/actores.csv' csv header delimiter E'\t' NULL 'NULL' encoding 'UTF8';

COPY pelis\_directores\_ intermedio (titulo, año, nombre) from '/Users/alvarocamacho/Desktop/Bideoklub/Ficheros csv/pelis\_directores.csv' csv header delimiter E'\t' NULL 'NULL' encoding 'UTF8'

COPY directores\_ intermedio (nombre, fecha\_nacimiento) from '/Users/alvarocamacho/Desktop/Bideoklub/Ficheros csv/directores.csv' csv header delimiter E'\t' NULL 'NULL' encoding 'UTF8'

COPY películas\_ intermedio (año, titulo, generos, puntuacion, duracion, idioma, calificacion) from '/Users/alvarocamacho/Desktop/Bideoklub/Ficheros csv/peliculas.csv' csv header delimiter E'\t' NULL 'NULL' encoding 'UTF8';

COPY pelis\_comentarios\_ intermedio (año, titulo, puntuacion, autor, contenido, hash, pagina\_web) from '/Users/alvarocamacho/Desktop/Bideoklub/Ficheros csv/pelis\_comentarios.csv' csv header delimiter E'\t' NULL NULL'

* Meter datos a las tablas finales

INSERT INTO peliculas.personal(nombre, año\_nacimiento)

SELECT nombre, fecha\_nacimiento

FROM actores\_intermedio

*Solo metemos en las tablas finales las columnas comunes como en este caso q la tabla peliculas.personal también contiene nacionalidad,pero como en la tabla intermedia no,no la añadimos.*

INSERT INTO peliculas.peliculas(año, titulo, duracion, idioma, calificacion) SELECT distinct año, titulo, duracion, idioma, calificacion FROM peliculas\_intermedio;

INSERT INTO peliculas.actua(nombre\_personal\_actor, año\_peliculas, titulo\_peliculas) SELECT distinct actor, año, pelicula FROM actores\_pelis\_intermedio;

INSERT INTO peliculas.actor(nombre\_personal) SELECT distinct actor FROM actores\_pelis\_intermedio;

INSERT INTO peliculas.criticas(critico, puntuacion, texto, año\_peliculas, titulo\_peliculas, url\_pag\_web) SELECT distinct autor, puntuacion, contenido, año, titulo, pagina\_web FROM pelis\_comentarios\_intermedio;

*Falta fecha porque en la tabla intermedia no tienen fecha las criticas*

INSERT INTO peliculas.director(nombre\_personal) SELECT distinct nombre FROM directores\_intermedio;

INSERT INTO peliculas.generos(genero, año\_peliculas, titulo\_peliculas) SELECT distinct generos, año, titulo FROM peliculas\_intermedio;

*Falta caratulas y pag\_web porque no hay como rellenarlos con las tablas intermedias*

* Género

En las peliculas tenemos el atributo género,el cual es multievaluado,este tipo de de atributos crean una tabla a mayores,nosotros para tratarlos usamos:

SELECT distinct regexp\_split\_to\_table(genero, '\s+'),

año::int,titulo

FROM public.peliculas

Este comando separa los generos en diferentes tuplas ,de esta manera podemos cargar la tabla género.

PARTE 3

select count(\*) from peliculas.personal; 1534

select count(\*) from peliculas.actor; 997

select count(\*) from peliculas.director; 560

select count(\*) from peliculas.actua; 172

select count(\*) from peliculas.peliculas; 1126

select count(\*) from peliculas.generos; 3431

select count(\*) from peliculas.criticas; 1129

select count(\*) from peliculas.pag\_web; 1129

CONSULTAS

1. Mostar el nombre de las personas que aparezcan en la base de datos como actor y como director. Construir la expresión equivalente en álgebra relacional.

SELECT nombre

FROM actor

union

SELECT nombre

FROM director

2. Mostrar el número total de minutos de todas las películas en las que actúa Tom Cruise. Agrupar el resultado por género.

SELECT sum(duracion)

FROM peliculas inner join actores on peliculas actor=Tom Cruise

group by genero

3. Mostrar las personas que actúan o dirigen películas catalogadas en el género de Horror. Construir la expresión equivalente en álgebra relacional.

SELECT nombre.actor

FROM peliculas inner join actores

WHERE genero=Horror

-----------------------------------------------------------------------------------------------------------------

4. Mostrar el nombre de los actores y del director que aparecen en la película "The Lord of the Rings: “The Return of the King”?.

SELECT nombre.actor , nombre.director

FROM pelicula inner join (actores inner join directores)

WHERE nombre.pelicula = "The Lord of the Rings: “The Return of the King”?

-----------------------------------------------------------------------------------------------------------------

5. Mostrar los nombres de los directores junto con las películas que hayan dirigido para aquellos que a su vez sean actores.

SELECT nombre.director , nombre.pelicula

FROM peliculas inner join (actores inner join directores on nombre.actor=nombre.director)

6. Mostrar los actores que hayan nacido posteriormente al 31 de diciembre de 1980. Construir la expresión equivalente en álgebra relacional.

SELECT nombre

FROM actores

WHERE año de nacimiento>1980

-----------------------------------------------------------------------------------------------------------------

7. Mostrar todos los géneros junto con el número total de películas catalogadas en ese género, ordenadas por los géneros que tienen más películas a los que tienen menos.

SELECT nombre.genero , count(añoTitulo)

FROM peliculas inner joinr genero

ORDER by count(añoTitulo) desc

8. ¿Qué películas comparten el mismo título, pero con diferente año? Ordenar el resultado por orden alfabético.

SELECT distinct Titulo

FROM peliculas

9. Mostrar las películas que tienen las críticas con la máxima puntuación. Construir la expresión equivalente en álgebra relacional.

SELECT Titulo

FROM peliculas inner join criticas

WHERE puntuacion=10

-----------------------------------------------------------------------------------------------------------------

nose

10. Mostrar el título de las películas que tienen la mínima puntuación media. La puntuación media de cada película se toma como la media de todas las puntuaciones que tiene dicha película.

SELECT Titulo

FROM peliculas inner join critica

WHERE avg(puntuacion)

Luego lo miro