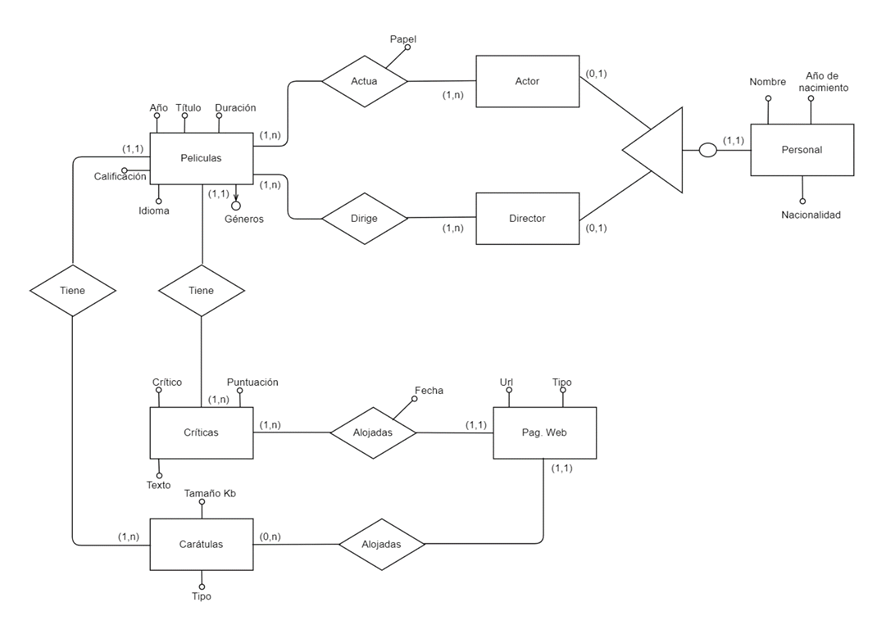
PL2

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))

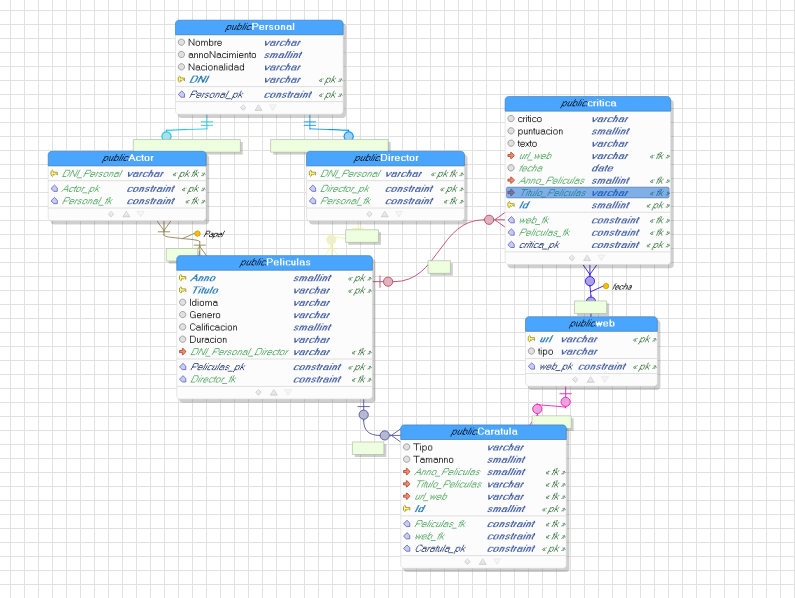
Personal(DNI(PK), nombre, año nacimiento, nacionalidad)

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

En nuestra base de datos:

Texto

Descripción generada automáticamente

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\_personal

FROM peliculas.actor

union

SELECT nombre\_personal

FROM peliculas.director

**σnombre\_personal(peliculas.director) ∪ σnombre\_personal(peliculas.actor)**

*Resultados en csv en llamado “Ejercicio1Profesor” y “Ejercicio1”*

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.peliculas

INNER JOIN peliculas.actua on peliculas.titulo = actua.titulo\_peliculas

INNER JOIN peliculas.generos on peliculas.año = generos.año\_peliculas

WHERE actua.nombre\_personal\_actor='Tom Cruise'

GROUP BY genero

Tabla

Descripción generada automáticamente con confianza media

*En nuestra base de datos trabajamos la duración como si fuese un text, por lo tanto nos imposible sumar las duraciones.*

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

SELECT peliculas.actua.nombre\_personal\_actor, peliculas.nombre\_personal\_director

FROM peliculas.actua

INNER JOIN peliculas.generos on actua.año\_peliculas = generos.año\_peliculas and generos.titulo\_peliculas = actua.titulo\_peliculas

INNER JOIN peliculas.peliculas on generos.año\_peliculas = peliculas.año and peliculas.titulo = generos.titulo\_peliculas

WHERE genero='Horror'

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

Nuestra base de datos:

Tabla

Descripción generada automáticamente

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

Base de datos proporcionada.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Álgebra relacional:

**π nombre\_personal\_actor, nombre\_personal\_director (σgenero=’Horror’(peliculas.actua ⨝ peliculas.generos ⨝ peliculas.peliculas))**

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\_personal\_actor, nombre\_personal\_director

FROM peliculas.peliculas

INNER JOIN peliculas.actua on actua.titulo\_peliculas = peliculas.titulo

WHERE titulo = 'The Lord of the Rings: The Return of the King'

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

Base de datos proporcionada:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Nuestra base de datos:

Tabla

Descripción generada automáticamente

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

SELECT nombre\_personal\_director, titulo

peliculas.peliculas

INNER JOIN peliculas.actua on peliculas.titulo = actua.titulo\_peliculas

WHERE nombre\_personal\_director is not NULL

Base de datos proporcionada:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Nuestra base de datos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

SELECT nombre

FROM peliculas.personal inner join peliculas.actor on actor.nombre\_personal = personal.nombre

WHERE f\_nacimiento < '12-31-1980'

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

*Incluidos en ficheros csv ya que las tablas son bastante extensas.(“Ejercicio6Prof” y “Ejercicio6”*

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 genero, count(titulo\_peliculas)

FROM peliculas.generos

GROUP BY genero

ORDER by count(titulo\_peliculas) desc

Base de datos proporcionada:

Tabla

Descripción generada automáticamente con confianza media

Nuestra base de datos:

Tabla

Descripción generada automáticamente con confianza media

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

SELECT titulo

FROM peliculas.peliculas

WHERE titulo = titulo and año <> año

ORDER BY titulo

Base de datos proporcionado:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Nuestra base de satos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

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

Base de datos proporcionada:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Nuestra base de datos:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Álgebra relacional:

**πtitulo(σpuntuacion=10(peliculas.pelicula⨯peliculas.critica))**

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.peliculas

INNER JOIN peliculas.criticas on peliculas.titulo = criticas.titulo\_peliculas

GROUP by titulo

HAVING avg(puntuacion)<5

Base de datos proporcionada:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Nuestra base de datos:

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente con confianza media

Los resultados e estas consultas también se pueden ver en los csv incluidos llamados “Ejercicio(*Número de ejercicio*)Prof.csv” y “Ejercicio(*Número de ejercicio*).csv”

*“Si me das pescado comeré hoy, si me enseñas a pescar comeré todos los días”*

*Proverbio chino*