



# PECL2

Grado en Ingeniería de Computadores  
Bases de Datos – Laboratorio 8/10

Carlos Eguren Esteban – 09088570M

Pablo Ruiz Vidaurre – 05962602J

Alejandro Torres Pérez de Baños – 03213701A

## Contenido

<b>1. Análisis esquema relacional primera parte .....</b>	<b>2</b>
<b>1.1. Esquema Carlos y Pablo.....</b>	<b>2</b>
<b>1.2. Esquema Alejandro.....</b>	<b>2</b>
<b>2. Creación esquema intermedio .....</b>	<b>3</b>
<b>3. Consultas para importar los datos al esquema intermedio .....</b>	<b>4</b>
<b>4. Consultas para importar al esquema final.....</b>	<b>6</b>

## 1. Análisis esquema relacional primera parte

### 1.1. Esquema Carlos y Pablo

Las diferencias entre el esquema relacional entregado en la primera parte y el proporcionado con el guion en esta segunda parte son notables.

En el nuestro faltan las tablas intermedias en las relaciones [N,M], aunque después estén representadas en el fichero sql. Estas tablas son las llamadas 'Actuan' y 'guionizan' en el nuevo esquema proporcionado. Además, la tabla llamada 'géneros' no está presente en nuestro esquema a pesar de ser un atributo multivaluado.

### 1.2. Esquema Alejandro

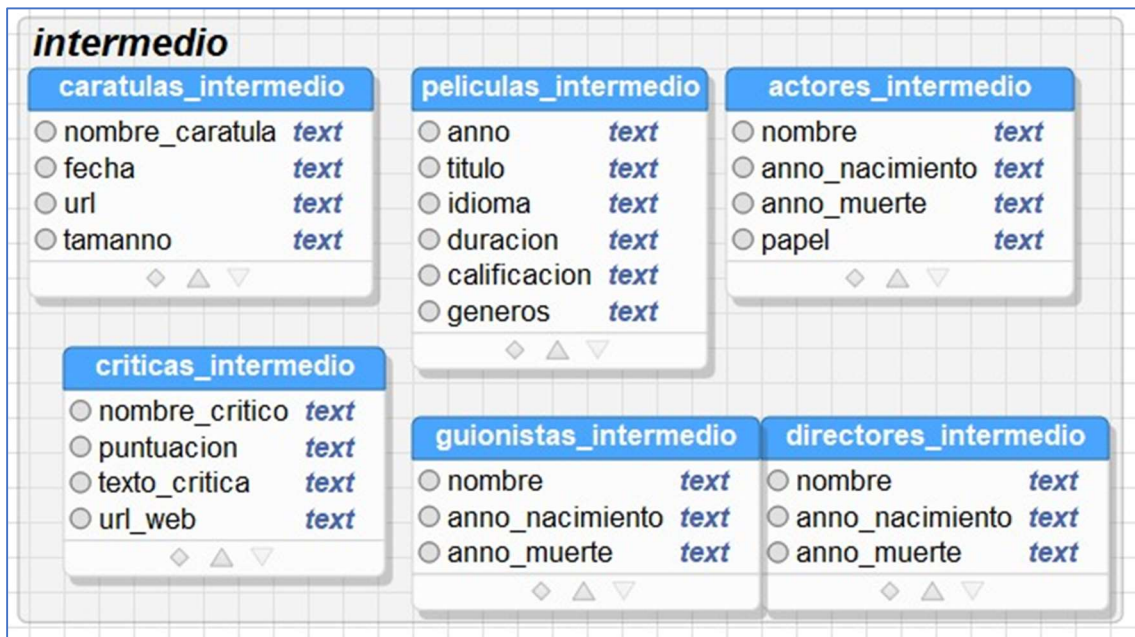
Comparando el esquema relacional hecho por Alejandro, también encontramos notables diferencias siendo la primera el formato, el esquema relacional había que realizarlo en pgmodeler pero al no ver esto, se realizó en Visio, además se utilizó el formato de diagrama de esquema y no lo visto en la carpeta Sesión 6 de laboratorio. Esto provoca sobre todo la falta de representación de las relaciones de cardinalidad, las claves foráneas están representadas como flechas que van de una entidad a otra a la que hacen referencia y las claves primarias no están señaladas.

Quitando los fallos de representación del esquema, encontramos 2 diferencias fundamentales. La primera es que no encontramos la entidad 'alojadas' que representaría una relación entre las carátulas y la página web. La segunda es que en el esquema proporcionado existe una entidad 'Dirige' intermedia entre 'director' y 'película' mientras que en el esquema proporcionado con el guión, esto se ha tratado como un atributo más de película, ya que cada película debe tener director y únicamente puede haber uno, con lo que su tratamiento como atributo es más correcto.

## 2. Creación esquema intermedio

En este apartado debemos crear un nuevo esquema que represente a los 6 archivos de extensión csv que se nos han proporcionado, creando una tabla por fichero con sus mismos atributos, todos de ellos tipo 'text'. Además, se pedía no añadir restricciones por lo que todas las PK y FK que debieran existir no se han creado para este esquema intermedio.

La tabla resultante es:



### 3. Consultas para importar los datos al esquema intermedio

```
\echo 'Cargando los datos'
\COPY intermedio.peliculas_intermedio FROM
'G:\Mi unidad\Universidad\2\1er Cuatrimestre\Bases de
Datos\Laboratorio\LaboratorioBBDD\LaboratorioBBDD\PL2-
SegundaParte\Archivos\Datos\peliculas.csv'
WITH (FORMAT csv, DELIMITER E'\t', NULL 'NULL', ENCODING 'UTF-8');

\COPY intermedio.guionistas_intermedio FROM
'G:\Mi unidad\Universidad\2\1er Cuatrimestre\Bases de
Datos\Laboratorio\LaboratorioBBDD\LaboratorioBBDD\PL2-
SegundaParte\Archivos\Datos\guionistas_peliculas.csv'
WITH (FORMAT csv, DELIMITER E'\t', NULL 'NULL', ENCODING 'UTF-8');

\COPY intermedio.directores_intermedio FROM
'G:\Mi unidad\Universidad\2\1er Cuatrimestre\Bases de
Datos\Laboratorio\LaboratorioBBDD\LaboratorioBBDD\PL2-
SegundaParte\Archivos\Datos\directores_peliculas.csv'
WITH (FORMAT csv, DELIMITER E'\t', NULL 'NULL', ENCODING 'UTF-8');

\COPY intermedio.criticas_intermedio FROM
'G:\Mi unidad\Universidad\2\1er Cuatrimestre\Bases de
Datos\Laboratorio\LaboratorioBBDD\LaboratorioBBDD\PL2-
SegundaParte\Archivos\Datos\criticas.csv'
WITH (FORMAT csv, DELIMITER E'\t', NULL 'NULL', ENCODING 'UTF-8');

\COPY intermedio.caratulas_intermedio FROM
'G:\Mi unidad\Universidad\2\1er Cuatrimestre\Bases de
Datos\Laboratorio\LaboratorioBBDD\LaboratorioBBDD\PL2-
SegundaParte\Archivos\Datos\caratulas.csv'
WITH (FORMAT csv, DELIMITER E'\t', NULL 'NULL', ENCODING 'UTF-8');

\COPY intermedio.actores_intermedio FROM
'G:\Mi unidad\Universidad\2\1er Cuatrimestre\Bases de
Datos\Laboratorio\LaboratorioBBDD\LaboratorioBBDD\PL2-
SegundaParte\Archivos\Datos\actores_peliculas.csv'
WITH (FORMAT csv, DELIMITER E'\t', NULL 'NULL', ENCODING 'UTF-8');
```

Todas las consultas cumplen el mismo formato:

```
\COPY intermedio.actores_intermedio FROM 'PATH'
```

\COPY que es el comando, *intermedio.peliculas\_intermedio* que es el nombre de la tabla del esquema intermedio que deseamos copiar, FROM junto con el PATH del csv. Habrá que modificar esta línea si deseamos importar otros archivos ya que está puesta la ruta absoluta.

```
WITH (FORMAT csv, DELIMITER E'\t', NULL 'NULL', ENCODING 'UTF-8');
```

FORMAT csv indica el formato del archivo que hemos seleccionado, DELIMITER el tipo de separador que hay entre datos, en el caso de nuestros csv es un tabulador, NULL 'NULL' indica cómo aparecen los valores NULL dentro del archivo y ENCODING es el tipo de codificación de caracteres.

## 4. Consultas para importar al esquema final

```
INSERT INTO cine.peliculas_final(anno,titulo, duracion, calificacionMPA,
idioma)
SELECT DISTINCT CAST(anno AS integer), titulo, NULLIF(duracion, 'NULL'),
NULLIF(calificacion, 'NULL'), NULLIF(idioma, 'NULL')
FROM intermedio.peliculas_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.actores_final(nombre)
SELECT DISTINCT nombre
FROM intermedio.actores_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.guionistas_final(nombre)
SELECT DISTINCT nombre
FROM intermedio.guionistas_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.directores_final(nombre)
SELECT DISTINCT nombre
FROM intermedio.directores_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.escribe_final(titulo_peliculas, nombre_guionista,
anno_peliculas)
SELECT DISTINCT titulo_peliculas, nombre, CAST(anno_peliculas AS integer)
FROM intermedio.guionistas_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.actua_final(titulo_peliculas, nombre_actor,
anno_peliculas)
SELECT DISTINCT titulo_peliculas, nombre, CAST(anno_peliculas AS integer)
FROM intermedio.actores_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.pagina_web_final(url_web)
SELECT DISTINCT url_web
FROM intermedio.caratulas_intermedio
UNION
SELECT DISTINCT url_web
FROM intermedio.criticas_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.criticas_final(anno_peliculas, titulo_peliculas,
critico, puntuacion, texto, url_web)
SELECT CAST(anno_peliculas AS integer), titulo_peliculas, nombre_critico,
CAST(puntuacion AS numeric(2,1)), texto_critica, url_web
FROM intermedio.criticas_intermedio ON CONFLICT DO NOTHING;
```

```
INSERT INTO cine.caratulas_final(anno_peliculas, titulo_peliculas,
nombre, tamanno)
SELECT CAST(anno_peliculas AS integer), titulo_peliculas,
nombre_caratula, tamanno
FROM intermedio.caratulas_intermedio ON CONFLICT DO NOTHING;
```

```

INSERT INTO cine.caratulas_WEB_final(anno_peliculas, titulo_peliculas,
url_web, fecha)
SELECT CAST(anno_peliculas AS integer), titulo_peliculas, url_web,
CAST(NULLIF(fecha, '\N') AS date)
FROM intermedio.caratulas_intermedio ON CONFLICT DO NOTHING;

INSERT INTO cine.generos_peliculas_final(genero, anno_peliculas,
titulo_peliculas)
SELECT DISTINCT regexp_split_to_table(generos, '\s+'), CAST(anno AS
integer), titulo
FROM intermedio.peliculas_intermedio ON CONFLICT DO NOTHING;

```

Estas han sido todas las consultas realizadas, comparten el mismo formato:

```

INSERT INTO cine.actua_final(titulo_peliculas, nombre_actor,
anno_peliculas)

```

INSERT INTO es la instrucción que inserta datos en el valor especificado, en uno de los ejemplos sería *cine.actua\_final(titulo\_peliculas, nombre\_actor, anno\_peliculas)*. Es decir, en las columnas que hay en el paréntesis de la tabla *actua\_final*.

```

SELECT DISTINCT titulo_peliculas, nombre, CAST(anno_peliculas AS integer)

```

Después se utiliza SELECT para indicar qué columnas deseamos de la tabla que queremos copiar, DISTINCT sirve para coger únicamente los datos que no están repetidos. Además, podemos cambiar el tipo de dato, como en *anno\_peliculas* que lo hemos modificado de TEXT, el original, a integer, esto se puede hacer con la función CAST.

```

FROM intermedio.actores_intermedio ON CONFLICT DO NOTHING;

```

Finalmente, FROM sirve para indicar cuál es la tabla o lugar del que deseamos coger los datos, en este caso *intermedi.actores\_intermedio*. ON CONFLICT sirve para decirle al código cómo queremos resolver los distintos problemas que se puede encontrar al importar los datos, con DO NOTHING estamos diciendo que si hay algún problema de duplicidad o algo, que no lo añada.



En el proceso de inserción de datos en las tablas finales nos hemos encontrado un problema con el atributo *nombres\_Personal\_Directores* de la tabla *peliculas*. Esta columna va a guardar los directores que trabajan en cada película, como puede ser más de uno, deberemos guardarlos separados por comas, esto lo hemos llegado a conseguir con la sentencia:

```
STRING_AGG(nombre, ', ')
```

Donde estamos diciendo que cada *nombre* (columna que representa los nombres de los directores en la tabla *directores\_intermedio* del esquema intermedio) encontrado lo separe por comas, pero agrupando todos en un mismo valor.

El problema es que esto generaba errores con la FK de *peliculas*, ya que aunque cada nombre por separado sí estaba referenciado, no existía un valor en la tabla *directores* que fuera un conjunto de 2 o más nombres separados por comas.

Ahora mostramos las 2 formas que hemos intentado:

```
--INSERT INTO cine.peliculas_final(anno,titulo, duracion, calificacionMPA,
idioma, nombre_personal_Directores)
--SELECT DISTINCT CAST(anno AS integer), titulo, NULLIF(duracion, 'NULL'),
NULLIF(calificacion, 'NULL'), NULLIF(idioma, 'NULL'), STRING_AGG(nombre, ',
')
--FROM intermedio.peliculas_intermedio JOIN intermedio.directores_intermedio
ON intermedio.directores_intermedio.titulo_peliculas =
intermedio.peliculas_intermedio.titulo
--GROUP BY anno, titulo, duracion, calificacion, idioma;
```

Esta instrucción nos eliminaba casi 600 valores de la tabla *peliculas*, donde ahora encontramos unas 1000 filas, con esta sentencia se nos quedaba en 407, provocado por la eliminación de ciertos valores y la mala codificación de la instrucción. Todo esto causaba errores con otras FK que referenciaban a ciertas películas que no habían conseguido ser añadidas.

```
--UPDATE cine.peliculas_final
--SET nombre_personal_Directores =
STRING_AGG(intermedio.directores_intermedio.nombre, ', ')
--FROM intermedio.directores_intermedio
--WHERE intermedio.directores_intermedio.titulo_peliculas =
cine.peliculas_final.titulo;
```

Lo otro que habíamos probado era a cargar la tabla sin el valor de esta columna y posteriormente actualizarla, pero causaba también error por mala codificación.

Por todo esto se ha decidido prescindir de esta columna y por lo tanto de la declaración de la FK que está comentada en el código para que el resto del programa se cargue con normalidad, de cara a la práctica 3 esperamos solventar el problema para realizarla correctamente.