

BASES DE DATOS

PECL2: SQL

Creadores:

Marcos Barroso Carmona (53996944M)- marcosbarroso@edu.uah.es

Alberto José Castillo Toma (09126517W)- albertojose.castillo@edu.uah.es

ÍNDICE

1. MODELO RELACIONAL	3 - 4
2. CARGA DE DATOS.....	5 - 8
3. REALIZACIÓN DE CONSULTAS	9 - 14

MODELO RELACIONAL

Hemos realizado las tablas de esta forma para que se puedan observar varias cosas:

1. Qué tipo de relación es.
2. Cuál es el nombre de cada tabla creada.
3. Cuáles son los atributos de cada tabla creada, y sus restricciones.
4. Cuales son sus PKs y FKs

E. FUERTE	Nombre tabla	Atributo 1	Atributo 2	Atributo 3	Atributo 4
Nombre tabla	<i>Disco</i>	Año_publicacion	Titulo	Url_Portada	Nombre_Grupo
PK/FK		PK	PK		FK
Restricciones		INTEGER	Text	Text	Text

E. FUERTE	Nombre tabla	Atributo 1	Atributo 2
Nombre tabla	<i>Grupo</i>	Nombre	URL_Grupo
PK/FK		PK	
Restricciones		Text	Text

E. FUERTE	Nombre tabla	Atributo 1	Atributo 2	Atributo 3	Atributo 4
Nombre tabla	<i>Usuario</i>	Nombre_Usuario	Nombre	Email	Contraseña
PK/FK		PK			
Restricciones		Text	Text	Text	Text

E. DÉBIL	Nombre tabla	Atributo 1	Atributo 2	Atributo 3	Atributo 5
Nombre tabla	<i>Canción</i>	Título	Duración	Año_Publicación_Disco	Título_Disco
PK/FK		PK		FPK	FPK
Restricciones		Text	Time	INTEGER	Text

E. DÉBIL	Nombre tabla	Atributo 1	Atributo 2	Atributo 3	Atributo 4	Atributo 5
Nombre tabla	<i>Edición</i>	Formato	Año_Edición	País	Año_Publicación_Disco	Título_Disco
PK/FK		PK	PK	PK	FPK	FPK
Restricciones		Text	INTEGER	Text	INTEGER	Text

RELACIÓN M:N	Nombre tabla	Atributo 1	Atributo 2	Atributo 3
Nombre tabla	<i>Desea</i>	Año_Publicacion_Disco	Titulo_Disco	Nombre_Usuario
PK/FK		FPK	FPK	FPK
Restricciones		INTEGER	Text	Text

RELACIÓN M:N	Nombre tabla	Atributo 1	Atributo 2	Atributo 3	Atributo 4	Atributo 5	Atributo 6	Atributo 7
Nombre tabla	<i>Tiene</i>	Formato_Edicion	Año_Edicion	País_Edicion	Año_Publicación_Disco	Título_Disco	Nombre_Usuario	Estado
PK/FK		FPK	FPK	FPK	FPK	FPK	FPK	
Restricciones		Text	INTEGER	Text	INTEGER	Text	Text	Text

MULTIEVALUADO	Nombre tabla	Atributo 1	Atributo 2	Atributo 3
Nombre tabla	<i>Géneros</i>	Año_Publicación_Disco	Título_disco	Género
PK/FK		FPK	FPK	PK
Restricciones		INTEGER	Text	Text

CARGA DE DATOS

Se han creado tablas temporales para cada uno de los archivos csv y con la misma estructura de sus atributos. En las tablas temporales todos los atributos son de tipo “text”.

A continuación, se han cargado los correspondientes csv a sus tablas.

```
1 \copy DiscoTemp FROM 'csv/discos.csv' DELIMITER ';' CSV HEADER ENCODING 'UTF8' NULL 'NULL';
2 \copy UsuarioTemp FROM 'csv/usuarios.csv' DELIMITER ';' CSV HEADER ENCODING 'UTF8' NULL 'NULL';
3 \copy CancionTemp FROM 'csv/canciones.csv' DELIMITER ';' CSV HEADER ENCODING 'UTF8' NULL 'NULL';
4 \copy EdicionTemp FROM 'csv/ediciones.csv' DELIMITER ';' CSV HEADER ENCODING 'UTF8' NULL 'NULL';
5 \copy DeseaTemp FROM 'csv/usuario_desea_disco.csv' DELIMITER ';' CSV HEADER ENCODING 'UTF8' NULL 'NULL';
6 \copy TieneTemp FROM 'csv/usuario_tiene_edicion.csv' DELIMITER ';' CSV HEADER ENCODING 'UTF8' NULL 'NULL';
```

Después se han realizado las distintas consultas para transferir los datos de las tablas temporales a las definitivas que se detallan a continuación para cada tabla definitiva:

TABLA GRUPO

```
1 INSERT INTO Grupo (Nombre, URL_Grupo)
2 SELECT DISTINCT Nombre_Grupo, URL_Grupo
3 FROM DiscoTemp;
```

De la tabla discoTemp se han seleccionado los nombres de los grupos que sean distintos (para que no haya repeticiones) y sus respectivas URL.

TABLA DISCO

```
1 INSERT INTO Disco (Ano_Publicacion, Titulo_Disco, Url_Portada, Nombre_Grupo)
2 SELECT DISTINCT ON (Ano_Publicacion, Titulo) Ano_Publicacion::INTEGER, Titulo, Url_Portada, Nombre_Grupo
3 FROM DiscoTemp;
```

De la tabla discoTemp se han seleccionado los discos distintos junto a su título, url de la portada, el nombre del grupo y el año de su publicación, que ha sido convertido a tipo INTEGER para realizar de manera adecuada las consultas posteriores.

TABLA USUARIO

```
1 INSERT INTO Usuario (Nombre_Usuario, Nombre, Email, Contraseña)
2 SELECT DISTINCT Nombre_Usuario, Nombre_Completo, Email, Contraseña
3 FROM UsuarioTemp;
```

De la tabla usuarioTemp se ha seleccionado su nombre, su nombre de usuario, su email y su contraseña.

TABLA CANCIÓN

```
1 INSERT INTO Cancion (Titulo_Cancion, Duracion, Titulo_Disco, Ano_Publicacion_Disco)
2 SELECT DISTINCT ON (CancionTemp.Titulo, Ano_Publicacion, DiscoTemp.Titulo) CancionTemp.Titulo,
3     MAKE_INTERVAL(mins => SPLIT_PART(CancionTemp.Duracion, ':', 1)::INTEGER,
4     secs => SPLIT_PART(CancionTemp.Duracion, ':', 2)::INTEGER)::TIME,
5     DiscoTemp.Titulo,
6     DiscoTemp.Ano_Publicacion::INTEGER
7 FROM CancionTemp
8 JOIN DiscoTemp ON CancionTemp.id_disco = DiscoTemp.id_disco;
```

Para insertar los datos en la tabla definitiva de canción, se ha hecho un JOIN con la condición de que el id del disco coincida en la tabla canciónTemp y discoTemp. De la tabla canciónTemp se ha seleccionado el título y la duración y de la tabla discoTemp se ha seleccionado el título del disco y el año de publicación, convirtiéndolo a INTEGER. Para pasar de un TEXT a formato TIME se ha dividido la cadena por el carácter “:” en el que la primera parte de la cadena son minutos y la segunda son los segundos. Cada parte de la cadena se convierte a INTEGER y con ellas se hace un INTERVAL, el cual se pasa a formato TIME.

TABLA EDICIÓN

```
1 INSERT INTO Edicion (Formato, Ano_Edicion, Pais, Ano_Publicacion_Disco, Titulo_Disco)
2 SELECT DISTINCT ON (Formato, Ano_Edicion, Pais, Ano_Publicacion, Titulo)Formato, Ano_Edicion::INTEGER, Pais, DiscoTemp.Ano_Publicacion::INTEGER, DiscoTemp.Titulo
3 FROM EdicionTemp
4 JOIN DiscoTemp ON EdicionTemp.id_disco = DiscoTemp.id_disco;
```

Para pasar las ediciones a su tabla definitiva, se hace un JOIN entre las tablas ediciónTemp y discoTemp en los cuales coinciden sus respectivos ID, u se selecciona el formato y año de edición de la tabla ediciónTemp y el año de publicación del disco y su título de la tabla discoTemp.

TABLA DESEA

```
1 INSERT INTO Desea (Ano_Publicacion_Disco, Titulo_Disco, Nombre_Usuario)
2 SELECT DISTINCT ON (Titulo_Disco, Nombre_Usuario) Disco.Ano_Publicacion::INTEGER, Disco.Titulo_Disco, DeseaTemp.Nombre_Usuario
3 FROM DeseaTemp JOIN Usuario ON DeseaTemp.Nombre_Usuario = Usuario.Nombre_Usuario JOIN Disco ON DeseaTemp.Titulo_Disco = Disco.Titulo_Disco;
```

Para pasar la tabla DeseaTemp a su tabla definitiva, realizamos un join de DeseaTemp con Usuario con la condición de que el nombre de usuario que existe en DeseaTemp sea el mismo que el que existe en Usuario. El resultado lo uniremos a la tabla Disco, con la condición de que el título del disco que existe en DeseaTemp sea el mismo que el que existe en Disco.

Así conseguiremos una tabla que contendrá, entre otros, atributos como el año en el que se publicó el disco, su título y el nombre del usuario que desea ese disco.

TABLA TIENE

```
1 INSERT INTO Tiene (Formato_Edicion, Ano_Edicion, Pais_Edicion, Ano_Publicacion_Disco, Titulo_Disco, Nombre_Usuario, Estado)
2 SELECT DISTINCT ON (Formato, Ano_Edicion, Nombre_Usuario, Pais, Ano_Publicacion_Disco, Titulo_Disco) Formato, Ano_Edicion::INTEGER, Pais, Ano_Publicacion_Disco::INTEGER, Titulo_Disco, TieneTemp.Nombre_Usuario, Estado
3 FROM TieneTemp JOIN Usuario ON TieneTemp.Nombre_Usuario = Usuario.Nombre_Usuario;
```

Para pasar los datos de las ediciones que tienen los usuarios se ha hecho un JOIN de las tablas tieneTemp y Usuario. Inicialmente no habría hecho falta ya que la tabla tieneTemp tiene todos los datos que debe tener la tabla Tiene definitiva, pero al haber errores en los nombres de usuario en los csv se ha hecho el JOIN con la condición de que el nombre de usuario de ambas tablas coincidan. (Había nombres de usuario en el csv Usuario_Tiene_Ediciones, que no aparecían en el csv de usuarios y viceversa).

TABLA GÉNEROS

```
1 INSERT INTO Generos (Ano_Publicacion_Disco, Genero, Titulo_Disco)
2 SELECT DISTINCT Ano_Publicacion::INTEGER, regexp_split_to_table(trim(both '[]' from Generos), ',\s*'), Titulo
3 FROM DiscoTemp;
4
```

Por último, nos encontramos con la tabla Generos, que contendrá los atributos “Ano_Publicacion_Disco, Genero y Titulo_Disco”.

Para ello, lo que hemos realizado es insertar en la tabla Generos los datos seleccionados de la tabla DiscoTemp, convirtiendo el año de publicación a entero, eliminando los corchetes de la columna Generos, dividiendo los géneros separados por comas en filas individuales y seleccionando el título del disco, asegurándonos de que los registros sean únicos.

CONSULTAS

1. Mostrar los discos que tengan más de 5 canciones. Construir la expresión equivalente en álgebra relacional.

```
1 \echo Consulta 1: Mostrar los discos que tengan más de 5 canciones
2 SELECT Titulo_Disco
3 FROM Disco
4 WHERE (SELECT COUNT(*)
5 FROM Cancion
6 WHERE Disco.Año_Publicacion = Cancion.Año_Publicacion_Disco AND Disco.Titulo_Disco = Cancion.Titulo_Disco) > 5;
```

La consulta obtiene los títulos de discos que tienen más de 5 canciones. Para cada disco en la tabla Disco, se cuenta cuántas canciones de la tabla Cancion coinciden con su título y año de publicación, usando una subconsulta. Si el conteo es mayor a 5, ese disco se incluye en los resultados.

En álgebra relacional:

S1 <- Disco $\bowtie_{(\text{Cancion.Año_Publicacion_Disco} = \text{Disco.Año_Publicacion})}$ Cancion

$\Pi_{\text{Titulo_disco}} \sigma_{(\text{Titulo_disco} \mathcal{G} \text{count}(\text{cancion}) > 5)}(\text{S1})$

2. Mostrar los vinilos que tiene el usuario Juan García Gómez junto con el título del disco, y el país y año de edición del mismo

```
1 \echo Consulta 2: Mostrar los vinilos que tiene el usuario Juan García Gómez junto con el título del disco, y el país y año de edición del mismo
2 SELECT Titulo_Disco, Pais_Edicion, Año_Edicion
3 FROM Tiene JOIN Usuario ON Tiene.Nombre_Usuario = Usuario.Nombre_Usuario
4 WHERE Usuario.Nombre = 'Juan García Gómez' AND Tiene.Formato_Edicion = 'Vinyl';
```

En esta consulta, teníamos que mostrar qué vinilos tenía Juan García Gómez, junto con el título, el país y el año de edición de cada uno.

Para ello, utilizamos un SELECT con el que mostraremos el título de cada disco, el país de cada edición y su año.

Los datos los cogemos de la unión que realizamos entre Tiene y Usuario, con la condición de que el nombre de usuario en Tiene sea el mismo que el que existe en Usuario.

Por último, mediante el WHERE filtramos, para que nos muestre solamente los vinilos del usuario Juan García Gómez.

3. Disco con mayor duración de la colección. Construir la expresión equivalente en álgebra relacional.

```
1 \echo Consulta 3: Disco con mayor duración de la colección.
2 WITH DuracionDiscos (Duracion,Titulo_Disco) AS
3 (SELECT SUM(Cancion.Duracion), Disco.Titulo_Disco
4 FROM Cancion JOIN Disco ON Disco.Titulo_Disco = Cancion.Titulo_Disco
5 GROUP BY Disco.Titulo_Disco)
6 SELECT Titulo_Disco, Duracion FROM DuracionDiscos WHERE Duracion = (SELECT MAX(Duracion) FROM DuracionDiscos);
```

Primero se realiza una vista temporal que almacena los títulos de los discos y su duración. Para obtener la duración se suman las duraciones de las canciones agrupadas por el título del disco.

Después se muestra el título del disco y su duración almacenada en la vista temporal en la cual la duración sea la máxima. Esto hace que solo se muestre el disco con mayor duración.

En álgebra relacional:

$\Pi_{Disco.Titulo_Disco} \mathcal{G}_{SUM(Cancion.Duracion), Disco.Titulo_Disco}(Cancion \bowtie_{Disco.Titulo_Disco = Cancion.Titulo_Disco} Disco) \text{ as } DuracionDiscos$

$\Pi_{Titulo_Disco, Duracion} (\sigma_{Disco.Titulo_Disco=Cancion.Titulo_Disco}(DuracionDiscos))$

4. De los discos que tiene en su lista de deseos el usuario Juan García Gómez, indicar el nombre de los grupos musicales que los interpretan.

```
1 \echo Consulta 4: De los discos que tiene en su lista de deseos el usuario Juan García Gómez, indicar el nombre de los grupos musicales que los interpretan.
2 SELECT Nombre_Grupo
3 FROM Disco JOIN Desea ON Disco.Ano_Publicacion = Desea.Ano_Publicacion_Disco AND Disco.Titulo_Disco = Desea.Titulo_Disco
4 JOIN Usuario ON Desea.Nombre_Usuario = Usuario.Nombre_Usuario
5 WHERE Usuario.Nombre = 'Juan García Gómez';
```

En esta consulta, teníamos que indicar el nombre de los grupos musicales que interpretan los discos que tiene en su lista de deseos el usuario Juan García Gómez. Para ello, utilizamos un SELECT con el que mostramos el nombre del grupo. Los datos los cogemos de la unión que realizamos entre las tablas Disco y Desea, y luego unimos con la tabla Usuario, con la condición de que el año de publicación y el título del disco en Disco sean los mismos que en Desea, y que el nombre de usuario en Desea sea el mismo que en Usuario. Por último, mediante el WHERE filtramos para que nos muestre solamente los discos deseados por el usuario Juan García Gómez.

5. Mostrar los discos publicados entre 1970 y 1972 junto con sus ediciones ordenados por el año de publicación.

```
1 \echo Consulta 5: Mostrar los discos publicados entre 1970 y 1972 junto con sus ediciones ordenados por el año de publicación.
2 SELECT Disco.Titulo_Disco, Disco.Ano_Publicacion, Edicion.Ano_Edicion
3 FROM Disco JOIN Edicion ON Disco.Titulo_Disco = Edicion.Titulo_Disco
4 WHERE Disco.Ano_Publicacion >= 1970 AND Disco.Ano_Publicacion <= 1972
5 ORDER BY Disco.Ano_Publicacion;
```

Se hace un JOIN entre disco y edición donde los títulos del disco son iguales y selecciona aquellos cuyo año de publicación está entre 1970 y 1972 y los ordena por año de publicación. Muestra el título del disco, el año de publicación y el año de cada una de sus ediciones .

6. Listar el nombre de todos los grupos que han publicado discos del género 'Electronic'. Construir la expresión equivalente en álgebra relacional.

```
1 \echo Consulta 6: Listar el nombre de todos los grupos que han publicado discos del género 'Electronic'.
2 SELECT DISTINCT Grupo.Nombre
3 FROM Grupo JOIN Disco ON Grupo.Nombre = Disco.Nombre_Grupo
4 JOIN Generos ON Disco.Titulo_Disco = Generos.Titulo_Disco
5 WHERE Generos.Genero LIKE '%Electronic%';
```

En esta consulta, teníamos que listar el nombre de todos los grupos que han publicado discos del género 'Electronic'. Para ello, utilizamos un SELECT con el que mostramos el nombre de cada grupo. Los datos los cogemos de la unión que realizamos entre las tablas Grupo y Disco, y luego unimos con la tabla Generos, con la condición de que el título del disco en Disco sea el mismo que el título del disco en Generos. Por último, mediante el WHERE filtramos para que nos muestre solamente los discos cuyo género contenga la palabra 'Electronic'.

En álgebra relacional:

$$\Pi_{\text{Grupo.Nombre}} (\sigma_{\text{Generos.Genero LIKE '%Electronic\%'}} (\text{Grupo} \bowtie_{\text{Grupo.Nombre = Disco.Nombre_Grupo}} \text{Disco} \bowtie_{\text{Disco.Titulo_Disco = Generos.Titulo_disco}} \text{Generos}))$$

7. Lista de discos con la duración total del mismo, editados antes del año 2000.

```
1 \echo Consulta 7: Lista de discos con la duración total del mismo, editados antes del año 2000.
2 SELECT Disco.Titulo_Disco, SUM(Cancion.Duracion)
3 FROM Disco JOIN Cancion ON Disco.Titulo_Disco = Cancion.Titulo_Disco
4 WHERE Disco.Ano_Publicacion < 2000 GROUP BY Disco.Titulo_Disco;
```

Se muestra el título del disco y la suma de la duración de sus canciones, para lo que se realiza un GROUP BY para agrupar las canciones que pertenezcan al mismo disco. También se unen las tablas de disco y canción y se seleccionan aquellos cuyo título de disco coincide y se filtran solo aquellos discos publicados antes del 2000.

8. Lista de ediciones de discos deseados por el usuario Lorena Sáez Pérez que tiene el usuario Juan García Gómez.

```
1 \echo Consulta 8: Lista de ediciones de discos deseados por el usuario Lorena Sáez Pérez que tiene el usuario Juan García Gómez
2 WITH Deseados_Lorena(Formato, Ano_Edicion, Pais, Ano_Publicacion_Disco, Titulo_Disco ) AS
3 (SELECT Formato, Ano_Edicion, Pais, Edicion.Ano_Publicacion_Disco, Edicion.Titulo_Disco
4 FROM Desea
5 JOIN Usuario ON Desea.Nombre_Usuario = Usuario.Nombre_Usuario
6 JOIN Edicion ON Desea.Ano_Publicacion_Disco = Edicion.Ano_Publicacion_Disco AND Desea.Titulo_Disco = Edicion.Titulo_Disco
7 WHERE Desea.Nombre_Usuario = 'Lorena Sáez Pérez')
8
9 SELECT dl.Formato, dl.Ano_Edicion, Pais, dl.Ano_Publicacion_Disco, dl.Titulo_Disco
10 FROM Tiene
11 JOIN Usuario ON Tiene.Nombre_Usuario = Usuario.Nombre_Usuario
12 JOIN Deseados_Lorena dl ON dl.Ano_Publicacion_Disco = Tiene.Ano_Publicacion_Disco AND dl.Titulo_Disco = Tiene.Titulo_Disco
13 WHERE Tiene.Nombre_Usuario = 'Juan García Gómez';
```

Primero filtramos los discos que desea Lorena y los almacenamos en una vista temporal, para ello utilizamos un SELECT con el que mostramos el formato, el año de edición, el país, el año de publicación del disco y el título del disco. Los datos los cogemos de la unión que realizamos entre las tablas Edición y Desea, y luego unimos con la tabla Usuario, con la condición de que el año de publicación y el título del disco en Edición sean los mismos que en Desea, y que el nombre de usuario en Desea sea el mismo que en Usuario. Por último, mediante el WHERE filtramos para que nos muestre solamente las ediciones deseadas por el usuario Lorena Sáez Pérez.

Después unimos con un JOIN las tablas Tiene y Usuario con la condición de que el nombre de usuario sea el mismo en ambas. Y después se hace un JOIN con la vista temporal con la condición de que el año de publicación y el título del disco sean el mismo en ambas relaciones. Por último, se filtra con un WHERE los discos que tiene Juan. Y con el SELECT mostramos el formato, el año de edición, el país, el año de publicación del disco y el título del disco. Como resultado tenemos los discos que desea Lorena y que tiene Juan. No se obtiene ningún resultado ya que Lorena no desea ningún disco.

9. Lista todas las ediciones de los discos que tiene el usuario Gómez García en un estado NM o M. Construir la expresión equivalente en álgebra relacional.

```
1 \echo Consulta 9: Lista todas las ediciones de los discos que tiene el usuario Gómez García en un estado NM o M.
2 SELECT Formato_Edicion, Ano_Edicion, Pais_Edicion, Ano_Publicacion_Disco, Titulo_Disco
3 FROM Tiene
4 JOIN Usuario ON Tiene.Nombre_Usuario = Usuario.Nombre_Usuario
5 WHERE Usuario.Nombre LIKE '%Gómez García' AND (Estado = 'NM' OR Estado = 'M');
```

Se hace un JOIN de las tablas Tiene y Usuario con la condición de que coincidan los nombres de usuario. Se seleccionan aquellas filas que cuyo nombre contiene la cadena “Gomez García” y el estado es NM o M. Se muestra el formato, el año y el país de edición y el año de publicación y el título del disco.

En álgebra relacional:

$\Pi_{\text{Formato_Edicion, Ano_Edicion, Pais_Edicion, Ano_Publicacion_Disco, Titulo_Disco}}(\sigma_{(\text{Usuario.Nombre LIKE '%Gómez García'}) \cap (\text{Estado} = \text{'NM'} \cup \text{Estado} = \text{'M'})}(\text{Tiene} \bowtie_{\text{Tiene.Nombre_Usuario} = \text{Usuario.Nombre_Usuario}} \text{Usuario}))$

10. Listar todos los usuarios junto al número de ediciones que tiene de todos los discos junto al año de lanzamiento de su disco más antiguo, el año de lanzamiento de su disco más nuevo, y el año medio de todos sus discos de su colección.

```
1 \echo Consulta 10: Listar todos los usuarios junto al número de ediciones que tiene de todos los discos junto al año de lanzamiento de su disco más antiguo, el año de lanzamiento de su disco más nuevo, y el año medio de todos sus discos de su colección.
2 SELECT Nombre_Usuario, COUNT(*), MIN(Ano_Edicion), MAX(Ano_Edicion), AVG(Ano_Edicion)
3 FROM Tiene GROUP BY Nombre_Usuario;
```

En esta consulta, teníamos que listar todos los usuarios junto al número de ediciones que tienen de todos los discos, junto al año de lanzamiento de su disco más antiguo, el año de lanzamiento de su disco más nuevo, y el año medio de todos sus discos de su colección. Para ello, utilizamos un SELECT con el que mostramos el nombre de usuario, el número de ediciones, el año de edición más antiguo, el año de edición más nuevo y el año medio de edición. Los datos los cogemos de la tabla Tiene, agrupando los resultados por el nombre del usuario.

¡OJO! El csv incluye discos cuya fecha de lanzamiento es el año 0. Los hemos tenido en cuenta a la hora de calcular la media, pero realmente el csv no debería incluir discos con fechas de lanzamiento erróneas.

Se podría solucionar, básicamente, aplicando: “WHERE Ano_Edicion > 0”.

11. Listar el nombre de los grupos que tienen más de 5 ediciones de sus discos en la base de datos.

```
1 \echo Consulta 11: Listar el nombre de los grupos que tienen más de 5 ediciones de sus discos en la base de datos
2 SELECT Nombre_Grupo
3 FROM Disco JOIN Edicion ON Disco.Ano_Publicacion = Edicion.Ano_Publicacion_Disco AND Disco.Titulo_Disco = Edicion.Titulo_Disco
4 GROUP BY Nombre_Grupo HAVING COUNT(*) > 5;
```

Se realiza un JOIN entre Disco y Edición con la condición de que coinciden el año de publicación y el título del disco, se agrupa por nombre de grupo que se filtran aquellos grupos que tienen más de 5 discos. Se muestra únicamente el nombre del grupo.

12. Lista el usuario que más discos, contando todas sus ediciones tiene en la base de datos.

```
1 \echo Consulta 12: Lista el usuario que más discos, contando todas sus ediciones tiene en la base de datos
2 SELECT Nombre_Usuario
3 FROM Tiene GROUP BY Nombre_Usuario
4 HAVING COUNT(*) = (SELECT MAX(NumDiscos) FROM (SELECT COUNT(*) AS NumDiscos FROM Tiene GROUP BY Nombre_Usuario) AS NumDiscos);
```

En esta consulta, teníamos que listar el usuario que más discos, contando todas sus ediciones, tiene en la base de datos. Para ello, utilizamos un SELECT con el que mostramos el nombre de usuario. Los datos los cogemos de la tabla Tiene, agrupando los resultados por el nombre del usuario y contando el número de discos que tiene cada usuario. Luego, utilizamos una subconsulta para encontrar el número máximo de discos que tiene cualquier usuario. Por último, mediante el HAVING filtramos para que nos muestre solamente los usuarios que tienen esa cantidad máxima de discos.