|  |
| --- |
| [Nombre del autor]  Curso: 2019/ 2020  2ºCurso 2ºCuatrimestre |

 Lunes, 20 de abril de 2020

|  |
| --- |
| **Bases de datos** |
| Práctica 3: Structured Query Language (SQL) |
| Grado en Ingeniería Informática |



Tabla de contenido

[Introducción 3](#_Toc38304632)

[Práctica 3 4](#_Toc38304633)

[Objetivos 4](#_Toc38304634)

[Enunciado 4](#_Toc38304635)

[Ejercicio 1. 5](#_Toc38304636)

[Diagrama Entidad-Relación 5](#_Toc38304637)

[Diseño Lógico 5](#_Toc38304638)

[Diferencias 6](#_Toc38304639)

[Ejercicio 2. 7](#_Toc38304640)

[¿Qué tipo de operación: a) producto cartesiano o b) concatenación, se está realizando en la cláusula WHERE tanto a través de la igualdad (d.cod\_gru=g.cod\_grupo) como del predicado IN? 7](#_Toc38304641)

[Proponer otra solución haciendo uso del predicado EXISTS 7](#_Toc38304642)

[¿Por qué predicado, de los estudiados en la parte II del SQL, podrías sustituir el igual en esta parte de la sentencia: g.nombre = ‘U2’? 7](#_Toc38304643)

[Consultas 8](#_Toc38304644)

[Tareas de administración 33](#_Toc38304645)

[Conclusiones 34](#_Toc38304646)

[Referencia bibliográfica 35](#_Toc38304647)

# Introducción

En esta memoria se plasma el trabajo realizado en la práctica 3 que se basa en la utilización del lenguaje SQL, haciendo uso del software propietario de BD Access de Microsoft.

Esta práctica abarca la parte del lenguaje de manipulación de datos de selección.

Para ello se aporta una base de datos con todas las tablas ya implementadas y se deben realizar consultas para obtener la información que se pide en cada una de ellas. Estas consultas deben seguir el estándar de SQL.

La base de datos que se usa en esta práctica, se trata de un SI basado en el mundo de la música. Se ha denominado Audiotek y en el se recogen las canciones, grupos de música, clubes de fans, discos grabados, artistas y compañías discográficas.

# Práctica 3

## Objetivos

El principal objetivo de esta práctica es conocer mejor y profundizar en el uso del lenguaje *Structured Query Language (SQL)*. Para ello se van a realizar una serie de consultas sobre la BD que se ha aportado.

A pesar de la existencia de más de una sentencia en este lenguaje, se ha hecho uso simplemente del predicado SELECT ya que todas las consultas que se piden son de selección.

## Enunciado

Partiendo de la BD que se va a explicar a continuación, se deben realizar las consultas que se van mostrando a lo largo de la memoria.

El SI denominado Audiotek consta de 8 tablas.

Los registros de la tabla Canción constan de un código, un título y una duración como campos. Dentro de un disco hay un conjunto de canciones, para conocer que canciones forman parte de un disco habrá que trabajar con la tabla esta donde ser guarda esta relación entre disco y canción*.* Esta tabla almacena tanto el código de canción como el de disco.

La tabla disco además del código que representa cada registro también se puede conocer el nombre y la fecha de publicación. También almacena el código de la compañía con la cual se ha publicado y el código del grupo que creo el disco.

Compañía, tiene un código, el nombre, dirección, fax y teléfono.

Grupo tiene al igual que todas, su código, además de un nombre, fecha de cuando se creó el grupo y país de origen. Un grupo está compuesto por una serie de artistas que tienen una función, a qué grupo pertenece cada artista y su función se almacena en la tabla pertenece, además del DNI del artista y el código del grupo.

El artista consta de un DNI y su nombre.

Como está relacionada toda esta BD es un factor importante y que se debe tener claro para poder comenzar con las consultas.

## Ejercicio 1.

### Diagrama Entidad-Relación

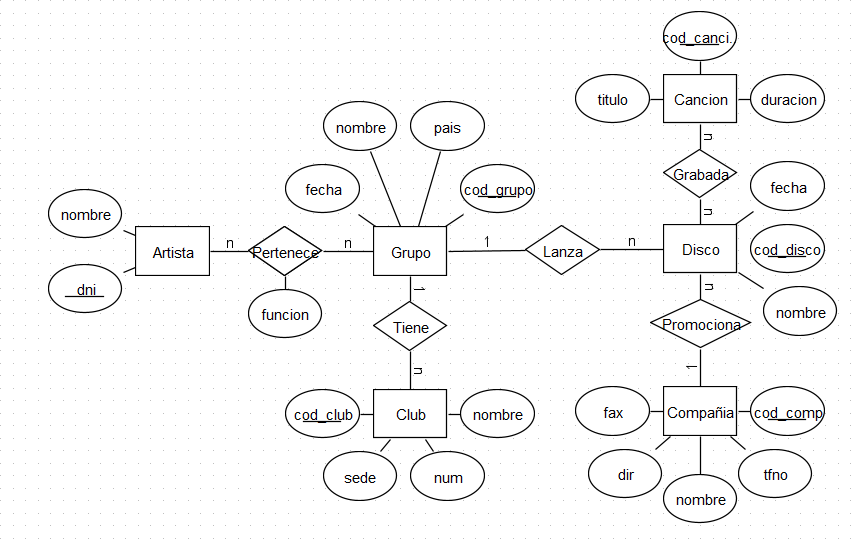


Ilustración 1. Diagrama Entidad-Relación AUDIOTEK

### Diseño Lógico

Cancion(cod\_cancion: dom\_codCancion, titulo: cod\_titulo, duración: cod\_duración)

CP:{cod\_cancion}

Disco(cod\_disco: dom\_codDisco, nombre: dom\_nombre, fecha: dom\_fecha, cod\_comp: dom\_codComp, cod\_gru: dom\_codGrupo)

CP:{cod\_disco}

CAj:{cod\_comp} 🡪 Compañía

CAj:{cod\_gru} 🡪 Grupo

Compañía(cod\_comp: dom\_codComp, nombre: dom\_nombre, fax: dom\_fax, tfno: dom\_tfno, dir: dom\_dir)

CP:{cod\_comp}

Grupo(cod\_gru: dom\_codGrupo, nombre: dom\_nombre, fecha: dom\_fecha, país:dom\_pais)

CP:{cod\_gru}

Club(cod\_club: dom\_codClub, nombre: dom\_nombre, sede: dom\_sede, num: dom\_num, cod\_gru: dom\_codGrupo)

CP:{cod\_club}

CAj:{cod\_gru} 🡪 Grupo

Artista(dni: dom\_dni, nombre: dom\_nombre)

CP:{dni}

Grabada(cod\_cancion: dom\_codCancion, cod\_disco: dom\_codDisco)

CP:{cod\_cancion, cod\_disco}

CAj:{cod\_cancion} 🡪 Cancion

CAj:{cod\_disco} 🡪 Disco

Pertenece(dni: dom\_dni, cod\_gru: dom\_codGrupo, función: dom\_funcion)

CP:{dni, cod\_gru}

CAj:{dni} 🡪 Artista

CAj:{cod\_gru} 🡪 Grupo

### Diferencias

En el diagrama de la memoria existen 8 relaciones y en cambio en la que se ha realizado hay 11. Esto se debe a las distintas cardinales, ya que en el diagrama que he realizado se tienen en cuenta todas las cardinales existentes y cada dos entidades debe existir una relación. En cambio, en el diagrama solo se reflejan aquellas que tienen cardinalidad n-n.

Esto es porque al ser cardinalidad n-n se debe almacenar la clave principal de cada entidad en una tabla, porque se debe guardar su relación de alguna manera. A diferencia de por ejemplo las cardinalidades 1-n en las cuales se almacena la clave primaria de la relación 1 en la entidad con la cardinalidad muchos y no almacena la clave primaria de la cardinalidad muchos en la otra entidad porque no es necesario y, por lo tanto, no es necesario hacer uso de otra tabla/relación para conocer la relación que hay entre las dos entidades.

## Ejercicio 2.

### ¿Qué tipo de operación: a) producto cartesiano o b) concatenación, se está realizando en la cláusula WHERE tanto a través de la igualdad (d.cod\_gru=g.cod\_grupo) como del predicado IN?

Se está realizando una operación de producto cartesiano, ya que se están combinando la información de dos tablas y se hace referencia a ellas en el FROM. [1][2][3][4]

### Proponer otra solución haciendo uso del predicado EXISTS

SELECT D1.nombre

FROM Disco D1

WHERE EXISTS (SELECT \*

FROM Grupo G, Disco D2

WHERE G.cod = D2.cod\_gru AND D2.cod = D1.cod AND G.nombre = 'U2')

### ¿Por qué predicado, de los estudiados en la parte II del SQL, podrías sustituir el igual en esta parte de la sentencia: g.nombre = ‘U2’?

Por el predicado LIKE porque la función de este predicado es la de expresas que cierta cadena se corresponda con el patrón que se expresa mediante una cadena de caracteres entre comillas. [1][2][3][4]

## Consultas

Antes de empezar con las consultas, se van a comentar una serie de aspectos.

Para empezar, cuando se quiere realizar la búsqueda de una cadena con el objetivo de obtener los registros que contienen esa cadena se puede realizar de dos formas distintas tanto con el operador =, como con el predicado LIKE. [1][2][3][4]

En cambio, cuando se quiere obtener la información de aquellos registros que no cumplen una cadena se puede hacer de tres formas. Puede construirse la operación donde se buscan aquellos registros que cumplan esa cadena y después hacer uso del operador booleano NOT, para obtener todos los registros que no cumplieran esa operación. Pero también se puede hacer uso del predicado NOT LIKE [1][2][3][4] o del operador <> [5] que exigen seleccionar aquellos registros que no contenga la cadena especificada. Se puede ver claramente las distintas posibilidades de realizarlo en la [primera consulta](#_Consulta_1._Obtener), a continuación de esta consulta, ya solo se mostrará una opción posible con respecto a este tema.

También se puede expresar de dos formas distintas cuando se quiere obtener los registros que superan un cierto valor numérico o no lo superen. Se puede poner que el valor debe ser mayor (>) que una cierta cifra o que el valor sea menor o igual (<=) y seleccionar todos los registros que no cumplan esa operación. Se puede ver una aplicación de esta explicación en la [consulta 2](#_Consulta_2._Seleccionar).

Puede darse el caso como en la [consulta 6](#_Consulta_6._Selecciona) que se pida obtener los registros cuyo contenido sea un valor que se encuentra dentro de un rango. Para ello se puede abarcar de dos formas de distintas. Se puede realizar dos operaciones unidas por el operador booleano AND que exige que se cumplan ambas operaciones, las operaciones deben ser que el valor del registro sea <= y la otra >=. La otra forma de realizarlo es mediante el predicado BETWEEN especificando los valores entre un AND. En este predicado están incluidos los valores que se indican, es decir, si se pone entre un rango de 2 a 4, sería BETWEEN 2 AND 4. Por lo tanto, si no quiere que se incluyan esos valores se deberá poner uno menos y uno más de la siguiente manera: BETWEEN 1 AND 3.

En el caso que se realice mediante dos operaciones, en vez de poner <= y >=, se pondrá < y >. [1][2][3][4][5]

Las fechas pueden ser expresadas de muchas maneras distintas, se puede o no especificar el año, pero si sí se especifica el año existen cuatro formatos para decir lo mismo. Pueden ser distintos por la forma en ser separar el día, mes y año; pudiendo hacer uso del guion (-) o de la barra (/). Pero también, por representar el mes mediante números o las tres primeras letras del mes en inglés. Un ejemplo de estas diferentes representaciones se puede ver en la [consulta 15](#_Consulta_15._Obtener). [6]

También tener en cuenta que cuando se hace uso del producto cartesiano, es decir, se hace referencia de dos o más tablas en la cláusula FROM, se obvia las palabras reservadas CROSS JOIN. [1][2][3][4]

En la cláusula SELECT puede ponerse a continuación ALL o DISTINCT, el ALL sirve para mostrar todos los datos recogidos y el DISTINCT para si hay datos repetidos no mostrarlo más de una vez. Por defecto, se realiza el SELECT ALL pero no es necesario especificarlo y por ello no se va a especificar a lo largo de la práctica.

Como se puede ver en la [consulta 18](#_Consulta_18._Seleccionar), cuando se hace uso del predicado IN se obtiene el mismo resultado que = ANY porque tienen la misma función, aunque se expresen de diferente forma.

Además, recordar que el SGBD no discrimina entre mayúsculas y minúsculas. Por lo tanto, se obtendrán los mismos resultados si por ejemplo pone ‘Palabra’ que ‘palabra’.

#### Consulta 1. Obtener el nombre de los grupos de música que no sean de España.

En este caso se quiere obtener el nombre de todos los registros de la tabla grupo donde el país no contenga la cadena ‘España’, pudiendo ser cualquier otra.

Como se ha dicho al principio cuando se quiere obtener aquellos registros que contengan una cadena o que no la contengan existen varias posibilidades, se van a mostrar en este caso como ejemplo.

El SELECT y FROM no varían en ningún caso, pero el WHERE varían en los tres casos que se muestran.

* 1er Caso

|  |
| --- |
| SELECT nombre  FROM grupo  WHERE NOT pais = 'España'; |

Para conseguir los registros que no contienen la cadena España, se buscan todos aquellos que si la contienen con el operador =, y posteriormente cuando se finaliza esa búsqueda se obtendrá el nombre de todos aquellos que no cumplieran esa operación, gracias al operador booleano NOT.

* 2º Caso

|  |
| --- |
| SELECT nombre  FROM grupo  WHERE pais NOT LIKE 'España'; |

En este caso se ha hecho uso del predicado NOT LIKE para obtener los registros cuyo campo país no contiene la cadena especificada.

* 3erCaso

|  |
| --- |
| SELECT nombre  FROM grupo  WHERE pais <> 'España'; |

El operador <> es utilizado en este tercer caso para conseguir todos los registros que contengan en el campo país una cadena distinta a ‘España’. [5]

A partir de estos tres casos, la solución para todos es la misma:

| **Consulta 1** |
| --- |
| **nombre** |
| U2 |
| Simple Minds |
| Mike + The Mechanics |
| Genesis |

Tabla 1. Consulta 1

#### Consulta 2. Seleccionar sólo el título de todas las canciones que duren más de 5 minutos.

Esta consulta requiere obtener el título de todas las canciones que duren más de 5 minutos de la tabla canción.

Hay dos formas de afrontar esta consulta. Al igual que la consulta anterior el SELECT y FROM no varían.

* 1er Caso

|  |
| --- |
| SELECT titulo  FROM cancion  WHERE duracion > 5; |

Se exige que el campo duración de los registros sea mayor que el número 5.

* 2º Caso

|  |
| --- |
| SELECT titulo  FROM cancion  WHERE NOT duracion <= 5; |

En este caso se obtienen todos los registros cuya duración sea inferior o igual a 5 y una vez realizada esta operación se cogen todos aquellos que no cumplan esa condición.

Los resultados obtenidos en ambos casos se pueden ver en la tabla a continuación (Tabla 2).

| **Consulta 2** |
| --- |
| **titulo** |
| 7 Deadly Sins |
| Lemon |
| So Cruel |
| Zooropa |

Tabla 2. Consulta 2

#### Consulta 3. Obtener el número total de discos de la base de datos.

Para esta consulta no va a ser necesaria la cláusula WHERE ya que con las cláusulas SELECT y FROM bastará. Se quiere obtener el número total de discos que hay y esta información se va a obtener de la tabla disco.

Para poder obtener el número total de discos, hay que obtener el número total de registros que hay en la tabla disco y para ello se hace uso de la función COUNT donde se pone \* entre los paréntesis para que cuente los registros que tiene. [7]

|  |
| --- |
| SELECT COUNT (\*)  FROM disco; |

| **Consulta 3** |
| --- |
| **Expr1000** |
| 18 |

Tabla 3. Consulta 3

#### Consulta 4. Obtener cuántos minutos dura la canción más larga.

Para esta consulta tampoco será necesario el uso de la cláusula WHERE.

Se pide obtener de la tabla canción obtener la máxima duración, es decir, obtener el registro del campo duración cuyo valor numérico sea mayor. Para poder obtener este valor se hace uso de la función MAX.[7]

|  |
| --- |
| SELECT MAX (duracion)  FROM cancion; |

| **Consulta 4** |
| --- |
| **Expr1000** |
| 6 |

Tabla 4. Consulta 4

#### Consulta 5. Según los datos en la base de datos, obtener las distintas funciones que un artista puede desempeñar dentro de un grupo.

Se hace uso de las cláusulas SELECT y FROM únicamente.

En este caso nos están pidiendo que se obtenga la función de un artista, esta función se encuentra en la tabla pertenece. Lo que se pide es obtener las distintas funciones para ello existe la palabra clave DISTINCT con la cual se permite observar todos los posibles que pueden adoptar los registros en un campo.

|  |
| --- |
| SELECT DISTINCT funcion  FROM pertenece; |

| **Consulta 5** |
| --- |
| **funcion** |
| bajo |
| batería |
| guitarra |
| teclado |
| voz |

Tabla 5. Consulta 5

#### Consulta 6. Selecciona tanto el nombre como la sede de los clubes de fans con un número de socios entre 100 y 500.

Se pide que se muestre tanto el nombre y la sede de los clubes de fans. Esta información se encuentra en la tabla club y pone como condición que el número de fans debe encontrarse entre 100 y 500. Se ha tenido en cuenta que esos valores también se incluyen.

Para resolver esta consulta se va a mostrar dos formas:

* 1er Caso

|  |
| --- |
| SELECT nombre, sede  FROM club  WHERE num BETWEEN 100 AND 500; |

Se ha hecho uso del predicado BEETWEEN para poder seleccionar los registros cuyo campo num, su valor numérico, estuviera entre 100 y 500.

* 2º Caso

|  |
| --- |
| SELECT nombre, sede  FROM club  WHERE num >= 100 AND num <= 500; |

En este caso se hacen dos operaciones y se exige que ambas deben ser ciertas para tener en cuenta dicho registro.

La primera operación exige que la variable num sea mayor o igual que 100 y la segunda que esa misma variable a su vez sea menor o igual que 500.

Obteniendo como resultado:

| **Consulta 6** | |
| --- | --- |
| **nombre** | **sede** |
| Jardin Botanico | 203,Valencia 46004 |
| Bonoculture | 12, East Av. |
| Waterfront | C/Martin Blas 22 |
| Presuntos | C/Albacete 12, bajo |
| Los Culpables | C/Maria Cristina 67 |
| Che U2 | C/Almussafes 59 |

Tabla 6. Consulta 6

#### Consulta 7. Obtener los nombres de los clubes de fans que no tienen definida la dirección de su sede o el número de admiradores.

Piden el nombre de los clubes, por lo tanto, de la table club que cumplan la condición de que en el campo sede o num el valor sea nulo.

En este caso piden que bien un campo u otro sea nulo, para expresar ese o, se hace uso de la operación booleana OR. Si alguno de los dos campos es cierto entonces se mostrará su nombre en el resultado. En el único caso en el cual no se obtengan resultados será cuando ambos campos tengan un valor.

Que el campo tenga un valor no nulo se expresa con el predicado IS NULL.

|  |
| --- |
| SELECT nombre  FROM club  WHERE (sede OR num) IS NULL; |

Tras realizar esta consulta no se obtiene ningún resultado, por lo que se entiende que, en todos los clubes, sede, num o ambos tiene un valor no nulo.

#### Consulta 8. Obtener los nombres de los grupos por orden alfabético inverso de nombre (de la Z a la A).

Para realizar esta consulta se va a usar una nueva cláusula, ORDER BY. Esta cláusula permite ordenar en base a un campo, además se puede indicar si se quiere realizar en orden ascendente o descendente.

En este caso piden los nombres de los grupos, por lo tanto, de la tabla grupo y la ordenación se va a realizar mediante el campo nombre en sentido descendiente, es decir, en el orden inverso al del abecedario.

|  |
| --- |
| SELECT nombre  FROM grupo  ORDER BY nombre DESC; |

| **Consulta 8** |
| --- |
| **nombre** |
| U2 |
| Simple Minds |
| Radio Futura |
| Presuntos Implicados |
| Mike + The Mechanics |
| Genesis |

Tabla 7. Consulta 8

#### Consulta 9. Obtener el nombre de los grupos que sólo constan de una palabra.

Se tiene que obtener el nombre de los grupos que cumplan la condición de que el valor de los registros del campo nombre deben tener solo una palabra.

Para poder obtener estos registros se ha hecho uso del predicado NOT LIKE. Para ello se exigido que el valor del campo nombre será distinto de aquel que contenga más de una palabra. Esto se ha expresado mediante el uso de dos asteriscos y un espacio. Los asteriscos simulan la aparición de cero o más caracteres, por lo tanto, se exige que no se cojan las cadenas que tras aparecer una serie de caracteres haya un espacio y de nuevo otra serie de caracteres, para simular el espacio que hay cuando una cadena está formada por más de una palabra.

|  |
| --- |
| SELECT nombre  FROM grupo  WHERE nombre NOT LIKE '\* \*'; |

| **Consulta 9** |
| --- |
| **nombre** |
| U2 |
| Genesis |

Tabla 8. Consulta 9

#### Consulta 10. Obtener el nombre y número de integrantes de los clubes de fans cuyo nombre contiene la cadena “100%”.

Se desea obtener el nombre y el número de fans que hay en cada club, por lo tanto, se va a trabajar con la tabla club.

La condición que se debe cumplir ahora es que el nombre contenga la cadena especificada, para ver si el registro contiene dicha cadena se hace uso del predicado LIKE.

Para representar la cadena se ha vuelto a hacer uso de los asteriscos delante y detrás del 100% ya que si dice que contiene puede ser que tanto al principio o después del 100% haya algún carácter más. Además, de esto hay otra complejidad y es que el % tiene un significado más allá de ser un símbolo, para que pierda este valor se debe poner entre corchetes y ya solo se tendrá en cuenta como un simple símbolo. [5]

|  |
| --- |
| SELECT nombre, num  FROM club  WHERE nombre LIKE '\*100[%]\*'; |

No se obtiene ningún resultado, por lo tanto, no habrá ningún club que contenga la cadena ‘100%’ en su nombre.

#### Consulta 11. Seleccionar nombre y sede de los clubes de fans de grupos de España, así como el nombre y fecha de formación del grupo al que admiran.

A partir de esta consulta se van a ver más similares donde se hace uso del producto cartesiano del álgebra relacional, este se puede ver representado de dos formas distintas. En un caso se hace uso de las palabras reservadas CROSS JOIN y en el otro caso se obvian estas palabras y se hace uso de una coma. Este segundo caso será la forma que se usa a lo largo de la memoria. Esta es la forma básica de combinar la información de dos o más tablas, haciendo referencia de ellas en la componente FROM de la consulta a realizar. [1][2][3][4]

En esta consulta se requiere de la tabla club y de la tabla grupo, ya que piden el nombre y la sede de los clubes y el nombre y fecha de los grupos.

Una condición para que de verdad se combinen estas tablas y exista una relación entre ellas para la solución obtenida es igualar el código por el que están relacionadas. En este caso es el código del grupo en la tabla club y el código de la tabla grupo.

Además, también se debe cumplir otra operación por lo que se hace uso del operador booleano AND. Esta operación exige que los registros seleccionados serán aquellos cuyo país en el que fue creado sea España.

Por lo tanto, se seleccionarán todos los registros cuyo país donde fue creado el grupo fuera España y una vez se han obtenido se mostrará información de aquellos registros que tengan el mismo código que el de los registros que cumplen que España es el país.

|  |
| --- |
| SELECT C.nombre, C.sede, G.nombre, G.fecha  FROM club C, grupo G  WHERE G.pais LIKE 'España' AND C.cod\_gru = G.cod; |

| **Consulta 11** | | | |
| --- | --- | --- | --- |
| **C.nombre** | **sede** | **G.nombre** | **fecha** |
| Presuntos | C/Albacete 12, bajo | Presuntos Implicados | 01/11/1985 |
| Implicado | Torrejon de Ardoz 12 | Presuntos Implicados | 01/11/1985 |
| Los Culpables | C/Maria Cristina 67 | Presuntos Implicados | 01/11/1985 |
| Jardin Botanico | 203,Valencia 46004 | Radio Futura | 07/01/1980 |
| Futuristas | C/Alboraya 10 | Radio Futura | 07/01/1980 |

Tabla 9. Consulta 11

#### Consulta 12. Seleccionar el nombre de todos los artistas que pertenezcan a grupos españoles.

En este caso se pide el nombre de los artistas, por lo tanto, se necesita la tabla artista. Pero también se necesita la tabla grupo porque se deben seleccionar solos los artistas que pertenezcan a grupos cuyo país sea igual a España.

Pero estas dos tablas necesitan de otra tabla, pertenece, para poder relacionarse. Por lo tanto, una condición, deberá incluir una igualdad entre grupo y pertenece, y, artista y pertenece, donde se igualan los códigos necesarios para relacionarse.

Además, de estas igualdades también se debe cumplir que el país del grupo debe ser España.

|  |
| --- |
| SELECT A.nombre  FROM artista A, grupo G, pertenece P  WHERE G.pais LIKE 'España' AND P.cod = G.cod AND P.dni = A.dni; |

| **Consulta 12** |
| --- |
| **nombre** |
| J.L. Giménez |
| Soledad Giménez |
| Nacho Maño |
| Carlos Torero |
| Santiago Auseron |
| Luis Auseron |
| Enrique Sierra |

Tabla 10. Consulta 12

#### Consulta 13. Seleccionar los discos (nombre y fecha de publicación) que contengan alguna canción con una duración que sobrepase los 5 minutos.

Para realizar esta consulta se va a necesitar tanto la tabla canción como la tabla disco, pero también se necesita de tabla esta que relaciona disco y canción.

Como condición además de las operaciones con los códigos también se exige que la duración de la canción sea mayor de 5 minutos, como se indica en la primera operación del WHERE.

Se hace uso del DISTINCT para evitar resultados parecidos ya que, sin usarlo, el disco Zooropa aparecía dos veces y no aportaba ninguna información, por ser redundante.

|  |
| --- |
| SELECT DISTINCT D.nombre, D.fecha  FROM disco D, cancion C, esta E  WHERE C.duracion > 5 AND E.can = C.cod AND E.cod = D.cod; |

| **Consulta 13** | |
| --- | --- |
| **nombre** | **fecha** |
| Achtung baby | 09/12/1991 |
| Good news F.N. world | 12/11/1995 |
| Zooropa | 10/08/1994 |

Tabla 11. Consulta 13

#### Consulta 14. Obtener los nombres de las canciones que dan nombre al disco en el que aparecen, ordenadas alfabéticamente.

Esta consulta es muy similar a la anterior, pero varía en tres aspectos. La condición que se exige ahora no es la duración de la canción si no que el nombre del disco sea igual al nombre de la canción. Posteriormente, se debe mostrar el título de dichas canciones, pero deben estar ordenadas por el título de manera alfabética.

Se exige que sea en sentido ascendente para que no se haga en el orden inverso al alfabético, aunque tampoco sería necesario indicar que debe ser en este sentido ya que por defecto se hace siempre en orden ascendente.

|  |
| --- |
| SELECT C.titulo  FROM disco D, cancion C, esta E  WHERE D.nombre = C.titulo AND E.can = C.cod AND E.cod = D.cod  ORDER BY C.titulo ASC; |

| **Consulta 14** |
| --- |
| **titulo** |
| Alma de blues |
| De sol a sol |
| Invisible touch |
| Living years |
| October |
| Once Upon A Time |
| Ser de agua |
| The unforgettable fi |
| Word of mouth |
| Zooropa |

Tabla 12. Consulta 14

#### Consulta 15. Obtener los nombres y direcciones de las compañías que han grabado algún disco cuyo nombre empiece por ‘A’ publicado durante la década de los 80 (1980-1989).

En esta consulta se requieren dos tablas, la de compañía y la de disco. Para que exista una relación entre ellas se deberá igualar el código necesario. Pero además también se debe cumplir que el nombre del disco empiece por A, para ello tras la A puede aparecer cualquier carácter y además debe haber sido publicado en la década de los 80, es decir, desde el uno de enero de 1980 hasta el 31 de febrero de 1989.

Se puede representar la fecha de más de una forma, se puede separar el día, del mes del año mediante el primer y tercer caso, o mediante barras en el segundo y cuarto.

Además, el mes puede expresarse mediante números (1 y 12) o mediante las tres primeras letras de los meses en inglés (Jan y Dec).

* 1er Caso

|  |
| --- |
| SELECT C.nombre, C.dir  FROM companyia C, disco D  WHERE C.cod = D.cod\_comp  AND D.fecha BETWEEN #1-1-1980# AND #31-12-1989#  AND D.nombre LIKE 'A\*'; |

* 2º Caso

|  |
| --- |
| SELECT C.nombre, C.dir  FROM companyia C, disco D  WHERE C.cod = D.cod\_comp  AND D.fecha BETWEEN #01/01/1980# AND #31/12/1989#  AND D.nombre LIKE 'A\*'; |

* 3er Caso

|  |
| --- |
| SELECT C.nombre, C.dir  FROM companyia C, disco D  WHERE C.cod = D.cod\_comp  AND D.fecha BETWEEN #1-Jan-1980# AND #31-Dec-1989#  AND D.nombre LIKE 'A\*'; |

* 4º Caso

|  |
| --- |
| SELECT C.nombre, C.dir  FROM companyia C, disco D  WHERE C.cod = D.cod\_comp  AND D.fecha BETWEEN #1/Jan/1980# AND #31/Dec/1989#  AND D.nombre LIKE 'A\*'; |

El resultado obtenido es el mismo:

| **Consulta 15** | |
| --- | --- |
| **nombre** | **dir** |
| WEA | L Hoyos 42 |

Tabla 13. Consulta 15

#### Consulta 16. Obtener cuántos clubs de fans tiene cada grupo, junto con el nombre del grupo.

Se quiere obtener el total de clubs de fans que tiene cada grupo, por lo tanto, se hace uso de la función COUNT.

Agrupado según el nombre del grupo se obtiene el número de club de fans que tiene cada uno.

|  |
| --- |
| SELECT G.nombre, COUNT(C.cod\_gru)  FROM grupo G, club C  WHERE C.cod\_gru = G.cod  GROUP BY G.nombre; |

| **Consulta 16** | |
| --- | --- |
| **nombre** | **Expr1001** |
| Genesis | 2 |
| Mike + The Mechanics | 2 |
| Presuntos Implicados | 3 |
| Radio Futura | 2 |
| Simple Minds | 4 |
| U2 | 4 |

Tabla 14. Consulta 16

#### Consulta 17. Seleccionar el nombre de los discos que han sido grabados después de 1986 por grupos que tengan algún club de fans con más de 5000 socios.

Se quiere obtener el nombre de aquellos discos que hayan sido grabadas por grupos. Estos grupos deben cumplir que hayan sido creados después del 1986, es decir, a partir del 1 de enero de 1987. Esto se puede expresas o bien que la fecha sea mayor que el 31/12/1986 o que sea mayor o igual al 1/1/1987.

Además, el grupo debe tener al menos un club de fans que supere el valor de 5000.

|  |
| --- |
| SELECT DISTINCT D.nombre  FROM disco D, club C, grupo G  WHERE D.fecha > #31/12/1986#  AND C.cod\_gru = G.cod AND D.cod\_gru = G.cod  AND C.num >5000; |

| **Consulta 17** |
| --- |
| **nombre** |
| La canción de Jperro |
| Living years |
| We can't dance |
| Word of mouth |

Tabla 15. Consulta 17

#### Consulta 18. Seleccionar el nombre del club de fans que tiene más integrantes, indicando además cuántos fans lo componen.

Para realizar esta consulta se ha hecho uso de una subconsulta y se muestran dos formas distintas que solo radican en la forma de expresarlo porque tienen la misma función.

Se pide el nombre y número del club de fans con más número de integrantes, para conseguir el club que más número de fan tiene se ha hecho uso de la subconsulta, donde se consigue el registro con valor numérico mayor de todo el campo num. Una vez se obtiene se pide que el resultado que se muestre debe tener el mismo número de fans que el que se ha obtenido en la subconsulta y de esta forma se obtiene el grupo que mayor número de fans tiene.

|  |
| --- |
| SELECT nombre, num  FROM club  WHERE num IN (SELECT MAX (num)  FROM club); |

|  |
| --- |
| SELECT nombre, num  FROM club  WHERE num = ANY (SELECT MAX (num)  FROM club); |

| **Consulta 18** | |
| --- | --- |
| **nombre** | **num** |
| Genefans | 23412 |

Tabla 16. Consulta 18

#### Consulta 19. Seleccionar los títulos de las canciones tanto de mayor duración como de menor duración, indicando cuánto dura cada una.

Se quiere obtener el título y la duración de los registros de la tabla canción. Debe ser aquellas con mayor duración o con menor para ello se hace uso de una subconsulta como en el caso anterior para poder obtener el máximo y el mínimo de un campo.

La resolución de esta consulta se planteó una solución, pero posteriormente se vio que había una mejor. La primera propuesta que se propuso era que se realizarán dos consultas y al final se hiciera una unión de los resultados obtenidos.

|  |
| --- |
| SELECT titulo, duracion  FROM cancion  WHERE duracion IN (SELECT MAX (duracion)  FROM cancion)  UNION ALL  SELECT titulo, duracion  FROM cancion  WHERE duracion IN (SELECT MIN (duracion)  FROM cancion); |

Después se observó que se podía haber hecho que en el WHERE haya dos condiciones y cuando se cumpla una u otra se seleccionarán dichos registros.

|  |
| --- |
| SELECT titulo, duracion  FROM cancion  WHERE (duracion IN (SELECT MAX (duracion)  FROM cancion)) OR  (duracion IN (SELECT MIN (duracion)  FROM cancion)); |

Una respuesta con la otra varía respecto al orden en el cuál aparecen los registros, pero no en los datos.

| **Consulta 19** | |
| --- | --- |
| **titulo** | **duracion** |
| 7 Deadly Sins | 6 |
| Lemon | 6 |
| So Cruel | 6 |
| Zooropa | 6 |
| Guitarra y voz | 1 |
| Price | 1 |

Tabla 17. Consulta 19

#### *Consulta 20. Seleccionar el número de DNI y el nombre de los artistas que han pertenecido a más de un grupo de los registrados en la base de datos*.

Obtener el DNI y nombre de la tabla artista, pero para conocer si han pertenecido a más de un grupo se necesita obtener datos de la tabla pertenece.

Para ello se deberá obtener todos los DNI de la tabla pertenece que aparezcan más de una vez, para ello se agrupa por el DNI y se obtendrá todos los registros que aparezcan más de una vez. Después esos DNI obtenidos deberán ser igual al de la tabla de artista para poder mostrar su nombre.

Al principio esta consulta se afrontó con una subconsulta. Pero después se vieron dos formas más reducidas de realizarla, sin hacer uso de la subconsulta o haciendo uso de la subconsulta, pero si hacer uso de la cláusula GROUP BY y HAVING.

|  |
| --- |
| SELECT nombre, dni  FROM artista  WHERE dni IN (SELECT dni  FROM pertenece  GROUP BY dni  HAVING COUNT (\*) >1); |

|  |
| --- |
| SELECT A.nombre, A.dni  FROM artista A, pertenece P  WHERE A.dni = P.dni  GROUP BY A.dni, A.nombre  HAVING COUNT (\*) >1; |

En este caso, sin hacer uso de las cláusulas GROUP BY y HAVING, cambia la condición del WHERE. Cuando el valor devuelto por la subconsulta sea mayor que uno significa que hay un artista que pertenece a más de un grupo y por lo tanto se mostrará dicho registro. Si no se cumple que es mayor de 1 no ocurrirá nada.

|  |
| --- |
| SELECT A.nombre, A.dni  FROM artista A  WHERE 1 < (SELECT count(\*)  FROM pertenece P  WHERE P.dni =A.dni); |

| **Consulta 20** | |
| --- | --- |
| **nombre** | **dni** |
| M. Rutherford | 8884566666 |

Tabla 18. Consulta 20

#### Consulta 21. Seleccionar el nombre de todos los discos del grupo que fue fundado hace más tiempo.

Para este caso se tiene que obtener el grupo fundado hace más tiempo con la función MIN, una vez obtenido se deberán mostrar todos los discos gracias a la relación por el código.

Se planteó una forma con dos subconsultas. En la primera se obtiene el código del grupo fundado con menos tiempo y se dice que el código del disco debe ser igual.

|  |
| --- |
| SELECT nombre  FROM disco  WHERE cod\_gru IN (SELECT cod  FROM grupo  WHERE fecha IN (SELECT MIN (fecha)  FROM grupo)); |

En cambio, en el segundo caso se ha hecho con una subconsulta y un producto cartesiano. Se han igualado los códigos y se ha obtenido la fecha que tenga menor valor numérico. En el caso que se cumplan la igualdad del código y la mínima fecha se mostrará el nombre del registro.

|  |
| --- |
| SELECT D.nombre  FROM disco D, grupo G  WHERE (D.cod\_gru = G.cod) AND (G.fecha IN (SELECT MIN (G1.fecha)  FROM grupo G1)); |

| **Consulta 21** |
| --- |
| **nombre** |
| We can't dance |
| Invisible touch |
| Seconds out |

Tabla 19. Consulta 21

#### Consulta 22. Seleccionar el código, nombre y dirección de todas las compañías discográficas que no han grabado ningún disco de grupos ingleses.

Se va a buscar todos los registros que cumplan que su país es Inglaterra y después se tomarán aquellos que no lo cumplan.

Primero se afrontó con un predicado NOT IN, pero al hacer uso de este predicado puede darse el caso de que todos los registros no sean recorridos y, por lo tanto, se plantea otra forma haciendo uso del predicado NOT EXISTS, este recorrerá todos los registros independientemente de las condiciones que existan.

Em ambos casos se tomarán los registros que no cumplan las condiciones impuestas en la subconsulta, en este caso el país impuesto es Inglaterra por lo tanto al final serán todas las compañías que no sea Inglaterra el país pudiendo ser otro cualquiera.

En el primer caso, todos los códigos que no cumplan las condiciones de la subconsulta serán las que se mostrarán.

|  |
| --- |
| SELECT cod, nombre, dir  FROM compañía  WHERE cod NOT IN (SELECT D.cod\_comp  FROM disco D, grupo G  WHERE D.cod\_gru = G.cod AND G.pais LIKE 'Inglaterra'); |

En cambio, para este caso se está igualando el código en el WHERE de la subconsulta con diferencia al primer caso que se realizaba en el WHERE de la consulta principal.

|  |
| --- |
| SELECT cod, nombre, dir  FROM companyia C  WHERE NOT EXISTS (SELECT \*  FROM disco D, grupo G  WHERE D.cod\_gru = G.cod AND G.pais LIKE 'Inglaterra' AND  C.cod = D.cod\_comp); |

| **Consulta 22** | | |
| --- | --- | --- |
| **cod** | **nombre** | **dir** |
| 2 | ARIOLA | Aragon 204 |
| 3 | WEA | L Hoyos 42 |
| 6 | PoliDiscos | Cami de Vera |
| 7 | PoliDiscos | Polynesia St. |

Tabla 20. Consulta 22

Para resolver esta consulta no se podía hacer uso del predicado IN o EXISTS porque aparecen resultados distintos. Si se realiza con estos predicados las compañías PoliDiscos no aparecían.

¿Por qué no aparecían? Esto puede ser porque la compañía quizás no ha sacado todavía un disco, pero eso no es lo que pregunta, en este caso pregunta que si ha publicado trabajos de grupos ingleses y esta compañía no lo ha hecho lo que significa que será un resultado más.

#### Consulta 23. Obtener el nombre de las compañías discográficas que sólo han publicado trabajos de grupos ingleses.

Para ello simplemente se va a modificar la consulta anterior eliminando el NOT del predicado IN de esta manera se obtendrá lo que se pide, que el país donde se crearon sea Inglaterra.

Se ha realizado mediante una subconsulta y sin hacer uso de subconsulta porque el predicado IN no era necesario.

Además, también se ha hecho mediante una subconsulta, pero utilizando el predicado EXISTS por el motivo que se ha comentado en la consulta anterior.

|  |
| --- |
| SELECT nombre  FROM companyia  WHERE cod IN (SELECT D.cod\_comp  FROM disco D, grupo G  WHERE D.cod\_gru = G.cod AND G.pais LIKE 'Inglaterra'); |

|  |
| --- |
| SELECT DISTINCT C.nombre  FROM companyia C, disco D, grupo G  WHERE C.cod = D.cod\_comp AND D.cod\_gru = G.cod AND G.pais LIKE 'Inglaterra'; |

|  |
| --- |
| SELECT nombre  FROM companyia C  WHERE EXISTS (SELECT \*  FROM disco D, grupo G  WHERE D.cod\_gru=G.cod AND G.pais LIKE 'Inglaterra' AND  C.cod = D.cod\_comp); |

| **Consulta 23** |
| --- |
| **nombre** |
| Island |
| Virgin |
| ATLANTIC |

Tabla 21. Consulta 23

#### Consulta 24. Seleccionar nombre y número de teléfono de aquellas compañías discográficas que hayan grabado todos los discos de alguno de los grupos.

Para resolver esta consulta se va a utilizar dos referencias a una tabla porque se quiere comparar información de una misma tabla, más concretamente la tabla disco. Se quiere observar si teniendo el mismo código de grupo, deberá tener el mismo código de compañía.

¿Por qué se hace uso del predicado NOT IN en vez del IN? Partiendo de la consulta a realizar, se va a explicar el motivo.

Las condiciones que se deben cumplir si se pusiera el predicado IN serían que ante un distinto código de compañía, tendría que ser que el código del grupo deberá ser distinto para que se cumpla el objetivo de la consulta. Por lo tanto, si por ejemplo dentro de un mismo grupo hay dos discos que son grabados por una compañía, pero uno que no lo está, como en un caso la condición será cierta, el código se obtendrá y se mostrará en los resultados.

En cambio, si se trabaja con el predicado NOT IN las condiciones serían que el código de compañía fuera distinto pero el de grupo fuera igual. Si se encuentra un caso en el cual no se cumple dicha condición no es seleccionado, pero si alguno cumple esa condición ese código no será mostrado, aunque haya 2 registros que lo cumplan y uno que no.

Al igual que los casos anteriores se ha hecho uso del predicado NOT EXISTS de forma que esta consulta sería el más correcto porqué se recorren todos los registros y con el NOT IN no y pueden olvidarse resultados, y, por lo tanto, no realizarse la consulta de manera correcta.

|  |
| --- |
| SELECT DISTINCT C.nombre, C.tfno  FROM companyia C, disco D  WHERE (D.cod\_comp = C.cod) AND  (C.cod NOT IN(SELECT D1.cod\_gru  FROM disco D1, disco D2  WHERE (D1.cod\_comp <> D2.cod\_comp) AND  (D1.cod\_gru = D2.cod\_gru))); |

|  |
| --- |
| SELECT DISTINCT C.nombre, C.tfno  FROM companyia C, disco D1  WHERE (D1.cod\_comp = C.cod) AND  NOT EXISTS (SELECT \*  FROM disco D2, disco D3  WHERE (D2.cod\_comp <> D3.cod\_comp) AND  (D2.cod\_gru = D3.cod\_gru) AND (D1.cod =D2.cod)); |

| **Consulta 24** | |
| --- | --- |
| **nombre** | **tfno** |
| ARIOLA | 913667890 |
| ATLANTIC | 5482312 |
| Island | 72724444 |
| Virgin | 20812446 |
| WEA | 932401213 |

Tabla 22. Consulta 24

#### Consulta 25. Obtener los nombres de los grupos en los que todos los integrantes tienen diferentes funciones.

Para realizar esta consulta es necesario tener dos referencias a una misma tabla, ya que se necesita comparar que, si los DNI son distintos y pertenecen al mismo grupo, es decir, tienen el mismo código de grupo, deberán tener distinta función.

Se ha hecho uso de un NOT IN y se comprueba que si alguno con distinto DNI en un mismo grupo tiene la misma función ese código no será seleccionado.

|  |
| --- |
| SELECT nombre  FROM grupo  WHERE cod NOT IN (SELECT P1.cod  FROM pertenece P1, pertenece P2  WHERE P1.dni <> P2.dni AND P1.cod = P2.cod AND P1.funcion = P2.funcion); |

Como en todas las consultas anteriores se ha realizado la consulta con el predicado NOT EXISTS, de tal forma, que esta sería la consulta más acertada.

|  |
| --- |
| SELECT G.nombre  FROM grupo G  WHERE NOT EXISTS (SELECT \*  FROM pertenece P1, pertenece P2  WHERE P1.dni <> P2.dni AND P1.cod = P2.cod AND  P1.funcion = P2.funcion AND G.cod = P1.cod); |

| **Consulta 25** |
| --- |
| **nombre** |
| U2 |
| Simple Minds |
| Mike + The Mechanics |
| Genesis |
| Presuntos Implicados |
| Radio Futura |

Tabla 23. Consulta 25

#### Consulta 26. Seleccionar el número total de fans de cada grupo español, además de su nombre, ordenados de mayor a menor número de fans.

Se pone como condición que el país del grupo sea España, se ordena por el número de fans, agrupado por el nombre de cada grupo.

Al agruparlo por el nombre grupo se suma los valores de los números de fans que tienen el mismo nombre.

|  |
| --- |
| SELECT G.nombre, SUM(num)  FROM club C, grupo G  WHERE C.cod\_gru = G.cod AND G.pais LIKE 'España'  GROUP BY G.nombre  ORDER BY SUM(num) DESC; |

| **Consulta 26** | |
| --- | --- |
| **nombre** | **Expr1001** |
| Radio Futura | 10207 |
| Presuntos Implicados | 617 |

Tabla 24. Consulta 26

a. OPCIONAL: modificar la consulta para que también aparezcan los grupos sin ningún club de fans.

Simplemente, se deben cumplir las dos premisas anteriores y después si se da la casualidad de que alguno no tenga ningún club de fans también se deberá mostrar para ello se hace uso de una concatenación externa, más concretamente RIGHT JOIN, donde se observa el rango abarcar claramente en la ilustración 2. Siguiendo la ilustración la tabla 1 sería el club y la tabla 2 el grupo.

En este caso dentro de los grupos solo se quieren los de España y además de ello se quieren los clubs de fans que no tienen ningún fan, es decir que la variable num tiene un valor nulo. Para poder abarcar tanto los grupos españoles como aquellos registros que tienen un valor nulo de número de fans se hace uso de la concatenación externa con la referencia de la tabla del operando de la derecha.



Ilustración 2. Right Join (Consulta 26 a))

|  |
| --- |
| SELECT G.nombre, SUM(num)  FROM club C RIGHT JOIN grupo G  ON C.cod\_gru = G.cod  WHERE G.pais LIKE 'España' OR C.num IS NULL  GROUP BY G.nombre; |

| **Consulta 26 a)** | |
| --- | --- |
| **nombre** | **Expr1001** |
| Presuntos Implicados | 617 |
| Radio Futura | 10207 |

Tabla 25. Consulta 26 a)

#### Consulta 27. Seleccionar los grupos (nombre y número total de componentes) formados por más de tres artistas.

Se debe obtener de la tabla grupo el nombre y número de artistas que lo forman. Solo aquellos grupos que tengan más de tres artistas, por lo tanto, para saber si un grupo tiene más de tres artistas se necesita la tabla de pertenece. Para saber si tiene más de tres se agrupa cada artista por el grupo al cual pertenece y posteriormente se exige en el HAVING que ese número sea mayor de 3.

|  |
| --- |
| SELECT G.nombre, COUNT(P.cod)  FROM grupo G, pertenece P  WHERE G.cod = P.cod  GROUP BY G.nombre  HAVING COUNT(P.cod) > 3; |

| **Consulta 27** | |
| --- | --- |
| **nombre** | **Expr1001** |
| Mike + The Mechanics | 4 |
| Radio Futura | 4 |
| U2 | 4 |

Tabla 26. Consulta 27

#### Consulta 28. Obtener cuántas canciones han sido grabadas por cada compañía discográfica, junto con el nombre y la dirección de la compañía.

La complejidad de este enunciado se encuentra en mostrar la cantidad de canciones que tiene cada disco, pero para ello se hace uso de las operaciones = para obtener la relación de las tablas mediante códigos. No es necesario la tabla canción ya que la cantidad de canciones que tiene cada disco se encuentra en la tabla esta, por lo tanto, si se agrupa por el código se obtendrá las canciones correspondientes de cada disco y a su vez la relación de los discos con las compañías.

|  |
| --- |
| SELECT C.nombre, COUNT(E.can), C.dir  FROM esta E, disco D, companyia C  WHERE C.cod = D.cod\_comp AND D.cod = E.cod  GROUP BY C.nombre, C.dir; |

| **Consulta 28** | | |
| --- | --- | --- |
| **nombre** | **dir** | **Expr1002** |
| ARIOLA | Aragon 204 | 22 |
| ATLANTIC | 12, E St. | 54 |
| Island | 67, JB St. | 43 |
| Virgin | 2,23th St. | 34 |
| WEA | L Hoyos 42 | 31 |

Tabla 27. Consulta 28

#### Consulta 29. Obtener cuántos discos han sido grabados por los grupos que tienen más de 1000 fans (sumando los de todos sus clubes de admiradores), junto con el nombre del grupo.

Se obtiene el código del grupo, de aquellos grupos que tenga un total de fans mayor de 1000.

Cuando se sabe cuáles son los grupos, se agrupa según el nombre para poder obtener el número de clubes que tienen cada grupo con más de 1000 fans.

|  |
| --- |
| SELECT G.nombre, COUNT(D.cod\_gru)  FROM grupo G, disco D  WHERE (G.cod = D.cod\_gru) AND (G.cod IN (SELECT cod\_gru  FROM club  GROUP BY cod\_gru  HAVING SUM (num) > 1000))  GROUP BY G.nombre; |

Esta primera consulta puede ser expresada de diferente manera, sin hacer uso de 2 subconsultas. Solo se hace uso de una para obtener si la suma de número de fans es mayor a 1000. Además, en el primer caso se utiliza una cláusula de GROUP BY y HAVING en la subconsulta y en este caso posterior no es necesario.

|  |
| --- |
| SELECT G.nombre, COUNT(D.cod\_gru)  FROM disco D, grupo G  WHERE (D.cod\_gru = G.cod) AND (1000 < (SELECT SUM(CL.num)  FROM club CL  WHERE CL.cod\_gru = G.cod))  GROUP BY G.nombre; |

| **Consulta 29** | |
| --- | --- |
| **nombre** | **Expr1001** |
| Genesis | 3 |
| Mike + The Mechanics | 2 |
| Radio Futura | 2 |
| Simple Minds | 4 |
| U2 | 4 |

Tabla 28. Consulta 29

#### Consulta 30. Seleccionar los títulos y duración de todas las canciones grabadas por el grupo al que admira el club de fans con menor número de integrantes.

Una vez se obtiene el código del club de fans con menos integrantes, se ponen como condiciones la operación de igualdad entre los códigos correspondientes para al finar lograr mostrar el título y la duración.

|  |
| --- |
| SELECT DISTINCT CA.titulo, CA.duracion  FROM cancion CA, esta E, disco D, grupo G  WHERE (CA.cod = E.can) AND (E.cod = D.cod) AND (D.cod\_gru = G.cod) AND  (G.cod IN (SELECT cod\_gru  FROM club  WHERE num IN (SELECT MIN (num)  FROM club))); |

De esta consulta se ha conseguido una más simplificada, pasando de dos consultas a una, pero se añade una tabla al FROM de la consulta principal y se ha añadido una condición más al WHERE.

|  |
| --- |
| SELECT DISTINCT CA.titulo, CA.duracion  FROM cancion CA, esta E, disco D, grupo G, club CL  WHERE (CA.cod = E.can) AND (E.cod = D.cod) AND (D.cod\_gru = G.cod) AND  (G.cod = CL.cod\_gru) AND (CL.num IN (SELECT MIN(num)  FROM club)); |

| **Consulta 30** | |
| --- | --- |
| **titulo** | **duracion** |
| Beautiful day | 5 |
| Before | 4 |
| Black and blue | 3 |
| Blame | 4 |
| Don't | 4 |
| Every body gets... | 5 |
| Get up | 4 |
| Let's pretend... | 5 |
| Living years | 4 |
| My crime of passion | 3 |
| Nobody knows | 3 |
| Nobody's perfect | 3 |
| Poor boy down | 2 |
| Seeing is believing | 5 |
| Stop baby | 4 |
| Time and place | 4 |
| Way you look at me | 3 |
| Why me? | 3 |
| Word of mouth | 3 |
| Yesterday, today,.. | 2 |

Tabla 29. Consulta 30

#### Consulta 31. Seleccionar el título y duración de la canción más larga, pero sólo si es la única con esa duración.

Se obtiene la máxima duración que puede tener una canción y para que el resultado sea mostrado debe ser un registro solo el que tenga esa duración por ello COUNT de los registros debe ser igual a 1.

|  |
| --- |
| SELECT titulo, duracionFROM cancionWHERE duracion IN (SELECT duracionFROM cancionWHERE duracion IN (SELECT MAX (duracion)FROM cancion)GROUP BY duracionHAVING COUNT (\*) = 1); |

Pero esta consulta puede realizarse sin hacer uso del GROUP BY y se ha hecho uso del predicado EXISTS.

|  |
| --- |
| SELECT C1.titulo , C1.duracionFROM cancion C1WHERE EXISTS (SELECT \*FROM cancion C2WHERE (C2.cod=C1.cod) AND (C2.duracion IN(SELECT MAX(duracion)FROM cancion))AND (1 = (SELECT COUNT(\*)FROM cancion C3WHERE C2.duracion = C3.duracion))); |

Tras realizar la consulta no se obtiene ningún resultado por lo que se entiende que no existe una canción que tenga más duración que las demás, sino que hay más de una canción con la máxima duración que existe.

#### Consulta 32. Seleccionar el sexto club con mayor número de fans (es decir, que haya cinco clubes con más integrantes), indicando cuántos fans tiene. NOTA: se permite usar TOP en la sentencia.

Una vez se ha obtenido cuales son los tops 6, se coge aquel con menor número de fans entre esos seis para obtener el sexto club con mayor número de fans.

|  |
| --- |
| SELECT nombre, num  FROM club  WHERE num IN (SELECT MIN(num)  FROM club  WHERE num IN (SELECT TOP 6 num  FROM club  ORDER BY num DESC)); |

| **Consulta 32** | |
| --- | --- |
| **nombre** | **num** |
| Mentes Fuertes | 1984 |

Tabla 30. Consulta 32

#### Consulta 33. Obtener el nombre de los artistas que tengan la función de bajo en un único grupo y que además éste tenga más de dos miembros.

En esta consulta se va comprobando cada condición impuesta, primero que la función sea de bajo y después que el grupo tenga más de dos miembros y a su vez el miembro debe pertenecer solo a un grupo.

|  |
| --- |
| SELECT A.nombre  FROM artista A, pertenece P  WHERE (A.dni = P.dni) AND (P.funcion LIKE 'bajo') AND  (P.dni IN (SELECT dni  FROM pertenece  WHERE cod IN (SELECT cod  FROM pertenece  GROUP BY cod  HAVING (COUNT(\*) >2))  GROUP BY dni  HAVING COUNT(\*) = 1)); |

Pero esta consulta puede ser simplificada en cuanto al uso de las cláusulas de GROUP BY y HAVING y haciendo uso del EXISTS se recorrerán todas las tablas por completo como ya se ha comentado.

|  |
| --- |
| SELECT A.nombre  FROM artista A  WHERE EXISTS (SELECT \*  FROM pertenece P1  WHERE (A.dni = P1.dni) AND (funcion LIKE 'bajo') AND  (1= (SELECT COUNT (\*)  FROM pertenece P2  WHERE (P1.dni=P2.dni) AND  (2 < (SELECT COUNT (\*)  FROM pertenece P3, pertenece P4  WHERE (P3.cod = P4.cod)))))); |

| **Consulta 33** |
| --- |
| **nombre** |
| Adam Clayton |
| Luis Auseron |
| Nacho Maño |

Tabla 31. Consulta 33

#### Consulta 34. Obtener el disco con menor duración total, indicando cuántos minutos dura.

Esta consulta tiene de especial que se realiza una subconsulta en el HAVING, con esta subconsulta se quieren obtener aquellos discos con menor duración, por lo tanto, se obtienen todas las duraciones mayores que todas las que hay, con esto se mostrarían todas las canciones que no tengan la duración mínima para después coger aquellas que no lo cumplían.

Por ejemplo, si se tiene los valores 1, 2, 3, 4, 5 y en la otra tabla también se tiene los mismos valores 1, 2, 3, 4, 5; y se exige que se tomen aquellos valores que sean mayores que los otros. 1 no es mayor que 1, 2, 3, 4 o 5, pero en cambio 2, si es mayor que 1 y 2; y 3 es es mayor que 1 y 2. Por lo tanto, esto es lo que se pretende lograr en esta subconsulta, pero claro el valor que se quiere obtener es el 1, por lo que si se exige que se obtenga todos los registros que no cumplan esto solo devolverá el 1 y, por lo tanto, al final se obtiene el valor mínimo.

|  |
| --- |
| SELECT D.nombre,SUM (C.duracion)  FROM esta E, cancion C, disco D  WHERE E.can = C.cod AND D.cod = E.cod  GROUP BY E.cod, D.nombre  HAVING NOT(sum(duracion) > ALL (SELECT SUM (C1.duracion)  FROM esta E1, cancion C1, disco D1  WHERE E1.can = C1.cod AND D1.cod = E1.cod  GROUP BY E1.cod)); |

| **Consulta 34** | |
| --- | --- |
| **nombre** | **Expr1001** |
| De sol a sol | 28 |
| The unforgettable fi | 28 |
| Once upon a time | 28 |
| Sister feelings call | 28 |

Tabla 32. Consulta 34

#### Consulta 35. Para formar nuevos grupos, se pretende crear dúos formados por un vocalista y un guitarrista que no hayan estado en el mismo grupo anteriormente. Generar todos los grupos posibles.

Primero se pondrá la tabla pertenece y artista dos veces para poder comparar en la misma tabla.

Para formar el dúo no deben pertenecer al mismo grupo, por lo tanto, los códigos deberán ser distintos. La función de uno será guitarrista y la del otro cantante.

Se igualará el DNI de pertenece a un artista y el otro al otro artista para poder mostrar los dúos que se obtienen finalmente.

|  |
| --- |
| SELECT A1.nombre, A2.nombre  FROM pertenece P1, pertenece P2, artista A1, artista A2  WHERE P1.cod <> P2.cod AND (P1.funcion LIKE 'guitarra' AND P2.funcion LIKE 'voz') AND  P1.dni = A1.dni AND A2.dni = P2.dni; |

Para mejorar la consulta se ha hecho uso del predicado EXISTS.

|  |
| --- |
| SELECT A1.nombre, A2.nombre  FROM artista A1, artista A2  WHERE EXISTS (SELECT \*  FROM pertenece P1, pertenece P2  WHERE P1.cod <> P2.cod AND (P1.funcion LIKE 'guitarra' AND  P2.funcion LIKE 'voz') AND P1.dni = A1.dni AND A2.dni = P2.dni); |

| **Consulta 35** | |
| --- | --- |
| **A1.nombre** | **A2.nombre** |
| C. Burchill | Bono |
| C. Burchill | Phil Collins |
| C. Burchill | Santiago Auseron |
| C. Burchill | Paul Young |
| C. Burchill | Soledad Giménez |
| Edge | Phil Collins |
| Edge | Santiago Auseron |
| Edge | Jim Kerr |
| Edge | Paul Young |
| Edge | Soledad Giménez |
| Enrique Sierra | Bono |
| Enrique Sierra | Phil Collins |
| Enrique Sierra | Jim Kerr |
| Enrique Sierra | Paul Young |
| Enrique Sierra | Soledad Giménez |
| J.L. Giménez | Bono |
| J.L. Giménez | Phil Collins |
| J.L. Giménez | Santiago Auseron |
| J.L. Giménez | Jim Kerr |
| J.L. Giménez | Paul Young |

Tabla 33. Consulta 35

## Tareas de administración

No ha sido necesario llevar a cabo las tareas de administración porque el software estaba instalado cuando se realizó la compra del portátil del que se ha hecho uso para realizar esta práctica. Por lo tanto, no se ha podido encontrado ningún problema en la instalación.

La versión con la que se ha realizado esta práctica es la 16.0 en el sistema operativo de Windows Pro y el tipo de licencia es Microsoft Office Profesional Plus 2016.

# Conclusiones

Gracias a esta práctica se ha conseguido conocer y aprender sobre una parte del lenguaje SQL, más concretamente el LMD de selección.

Para poder abarcar esta práctica en las circunstancias dadas se realizado una lectura profunda sobre la lección tanto de la documentación básica, complementaria y el curso que se propuso.

Esto es suficiente para poder abarcar las 20-25 primeras consultas. Para las siguientes, no sirve solo la lectura, sino que se exige de perspicacia, comprensión y tiempo para poder dar con la solución.

Esta práctica me ha servido para poder observar que con leer toda la documentación no me bastaba si no que debía realizar ejercicios y poner en práctica la teoría porque así es como de verdad he aprendido.

# Referencia bibliográfica

La referencia de la que se ha hecho uso para la realización de la Práctica 3 son los siguientes recursos:

1. ELMASRI, Ramez; NAVATHE, Shamkant B. Sistemas de bases de datos. Conceptos fundamentales. Addison Wesley Iberoamericana, 1997.
2. MOTA, L.; CELMA, M.; CASAMAYOR, J. C. Bases de datos relacionales: teoría y diseño. Universidad Politécnica de Valencia. SPUPV, 1994, vol. 94.
3. DATE, Chris J. Introducción a los sistemas de bases de datos. Pearson Educación, 2001.
4. PELZER, Peter Gulutzan&Trudy. SQL-99 Complete, Really. 1999.
5. Support.microsoft.com. 2020. Tabla De Operadores. [En línea] Disponible en: <https://support.microsoft.com/es-es/office/tabla-de-operadores-e1bc04d5-8b76-429f-a252-e9223117d6bd?ui=es-es&rs=es-es&ad=es>.
6. Support.microsoft.com. Consultas. [En línea] Disponible en: <https://support.microsoft.com/es-es/office/consultas-93fb69b7-cfc1-4f3e-ab56-b0a01523bb50?ui=es-ES&rs=es-ES&ad=ES>.
7. Deletesql.com. Curso SQL - Curso SQL Desde Cero. [En línea] Disponible en: <http://deletesql.com/viewforum.php?f=5&sid=7c65e10af34cca4197217bd8878c349e>.