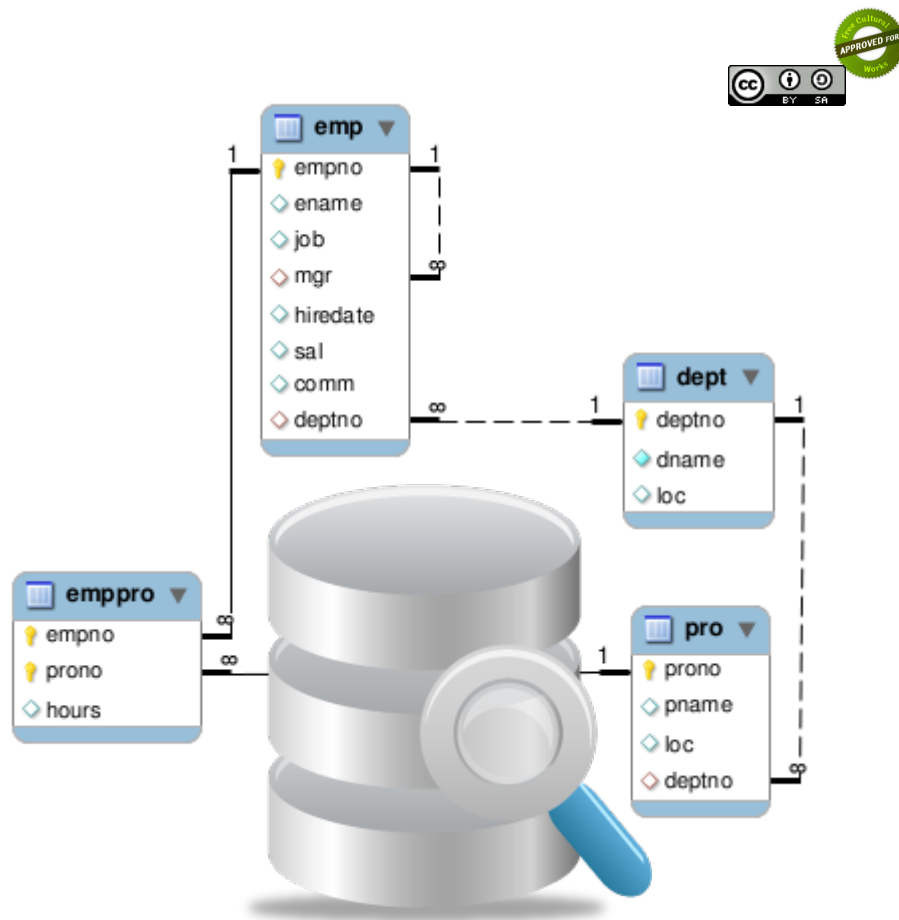


# Bases de datos 2019/2020

## Práctica 3: Structured Query Language (SQL)



**Autor@:**

Piedad Garrido ([piedad@unizar.es](mailto:piedad@unizar.es))

Grado en Ingeniería Informática (GII)

## 1. Objetivo de la práctica

Esta práctica tiene como objetivo profundizar en la utilización del lenguaje SQL (*Structured Query Language*). Se realizarán una serie de ejercicios, de consulta, sobre una Base de Datos (BD) ya existente que contiene información previamente introducida. Se utilizará exclusivamente la sentencia SELECT, mientras que otros predicados dedicados a la modificación de los datos incluidos en la BD (CREATE, INSERT, UPDATE, DELETE, ALTER, etc.) serán estudiados en prácticas posteriores, puesto que su complejidad suele ser más reducida. Las consultas más potentes y útiles, sin embargo, suelen requerir de la pericia del programador para extraer la información adecuada.

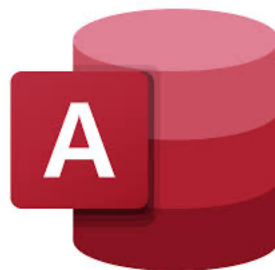
## 2. Entorno de trabajo

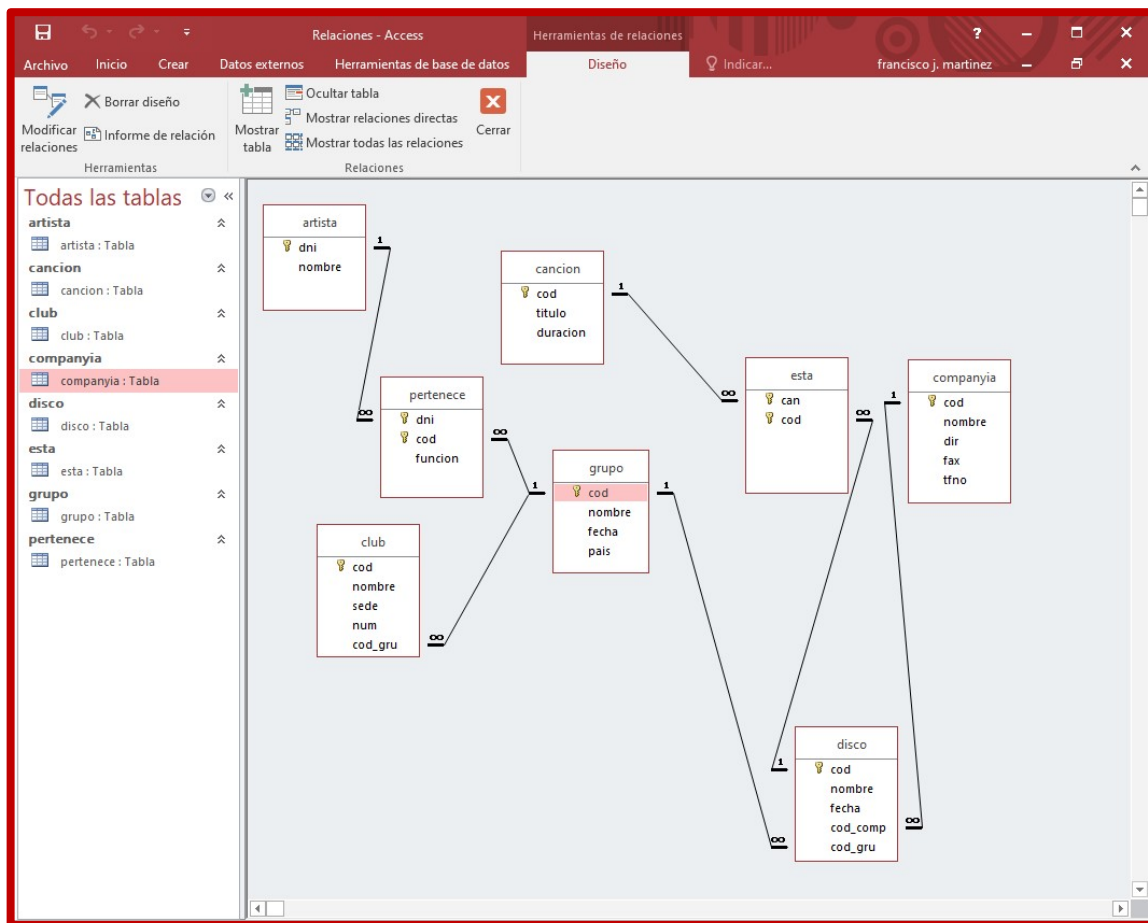
La base de datos, a utilizar en esta práctica, está construida empleando el software propietario de Bases de Datos (BD) Access de Microsoft o su homólogo gratuito y abierto OpenOffice DB. Este tipo de software no es necesario ejecutarlo en una arquitectura Cliente/Servidor (C/S), donde el equipo que hace de servidor contiene la BD y permite la conexión de múltiples clientes que actúan sobre ella, sino que se puede trabajar con él como una sencilla aplicación de escritorio y su proceso de instalación es rápido. Versiones de ambos tipos de software pueden descargarse desde los siguientes enlaces, para plataformas de 32 y 64 bits de los distintos sistemas operativos existentes en el mercado:

<https://www.openoffice.org/es/producto/base.html>

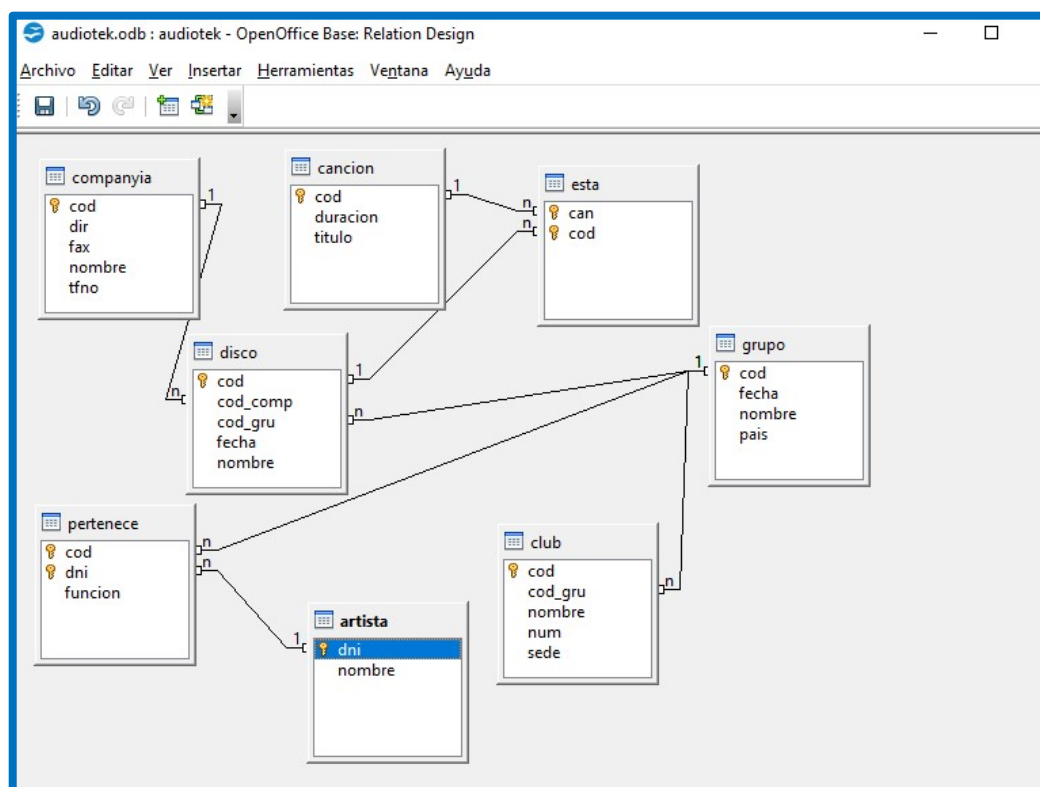
<https://www.microsoft.com/es-es/download>

A continuación, se muestran unas capturas de pantalla de las partes más importantes de las herramientas a utilizar.





**Figura 1** Visores de los diagramas de relaciones de la BD Audiotek en Microsoft Access y OpenOfficeDB



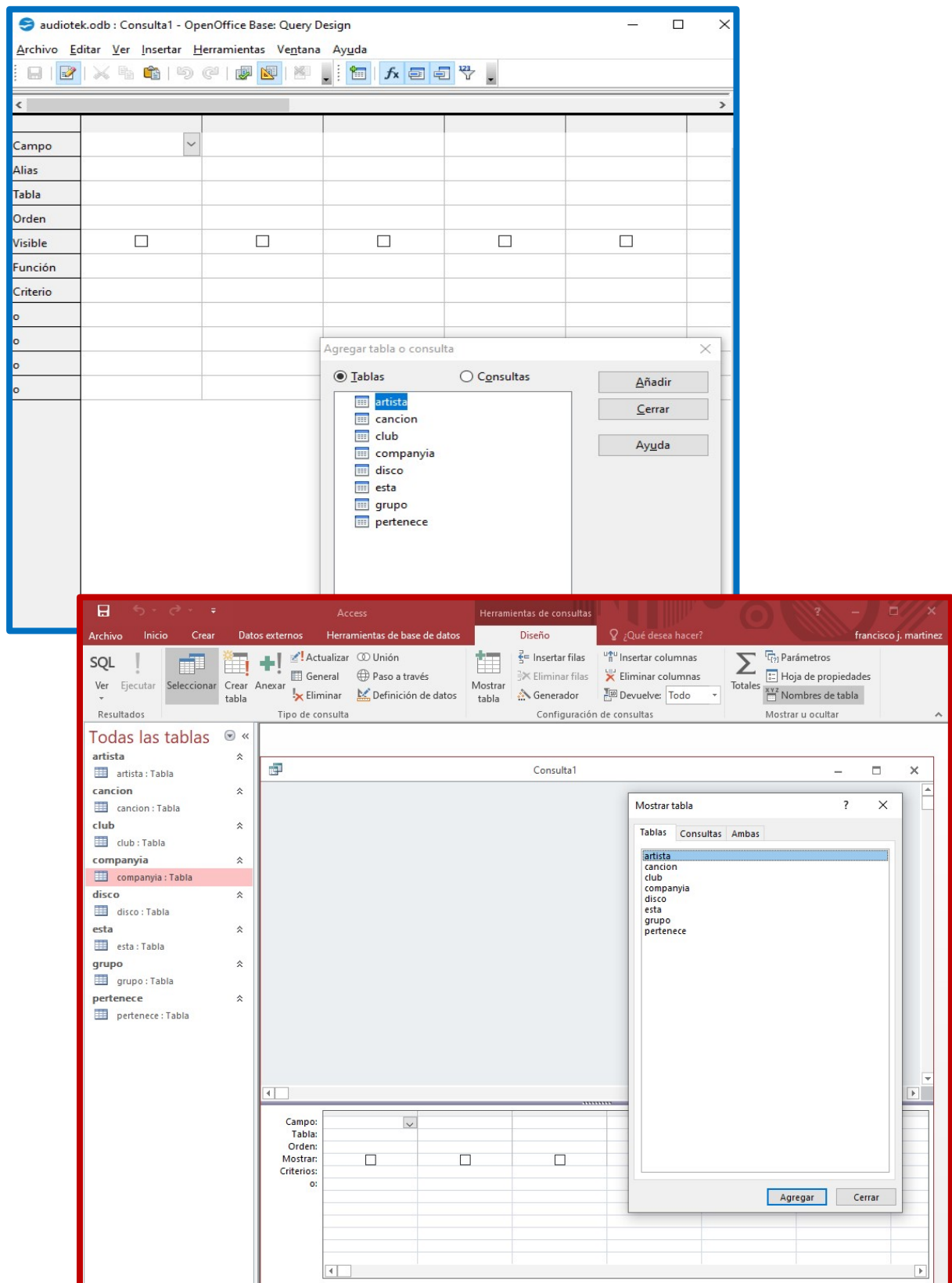


Figura 2 Herramientas de consultas (Diseño gráfico)

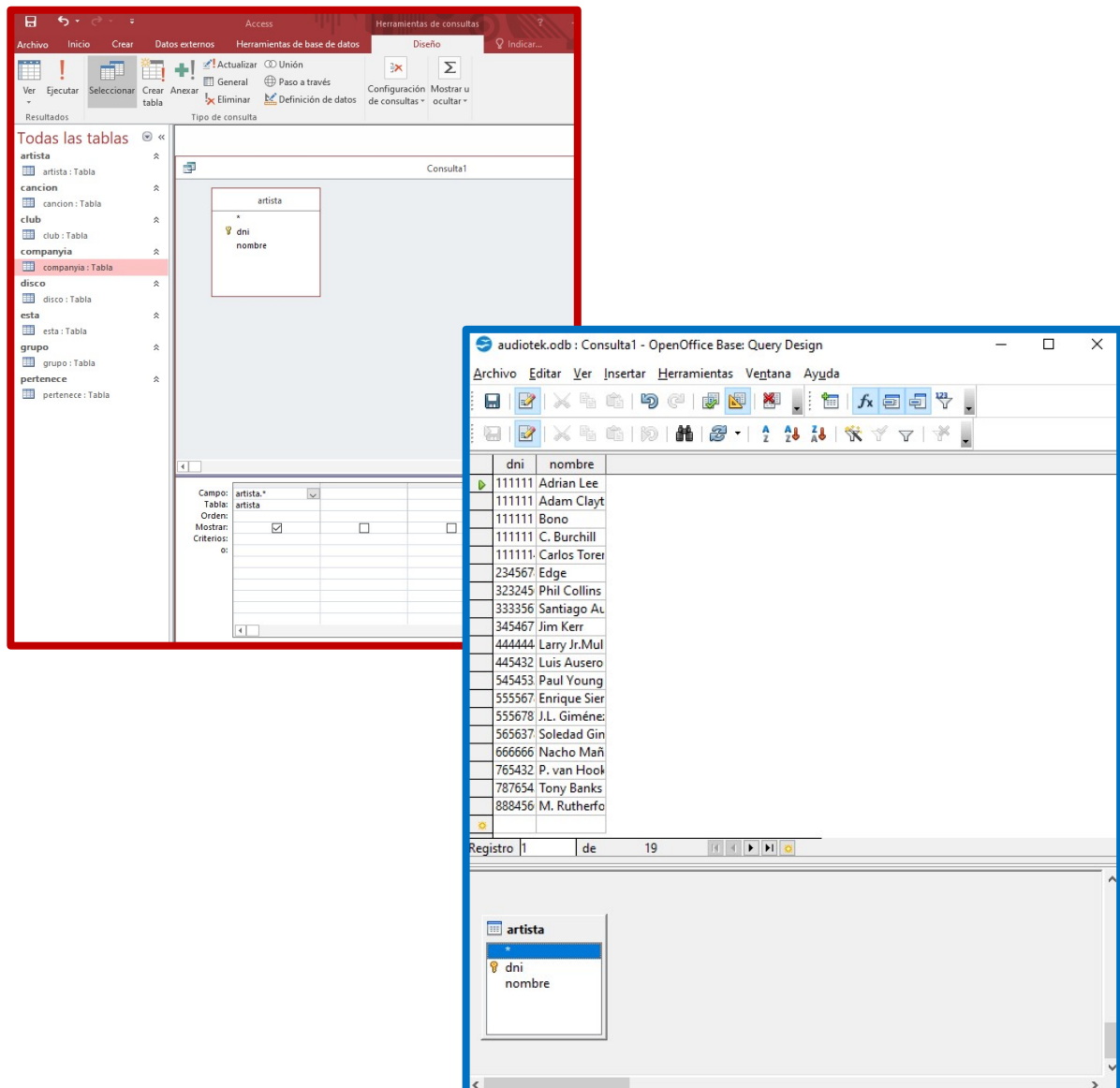
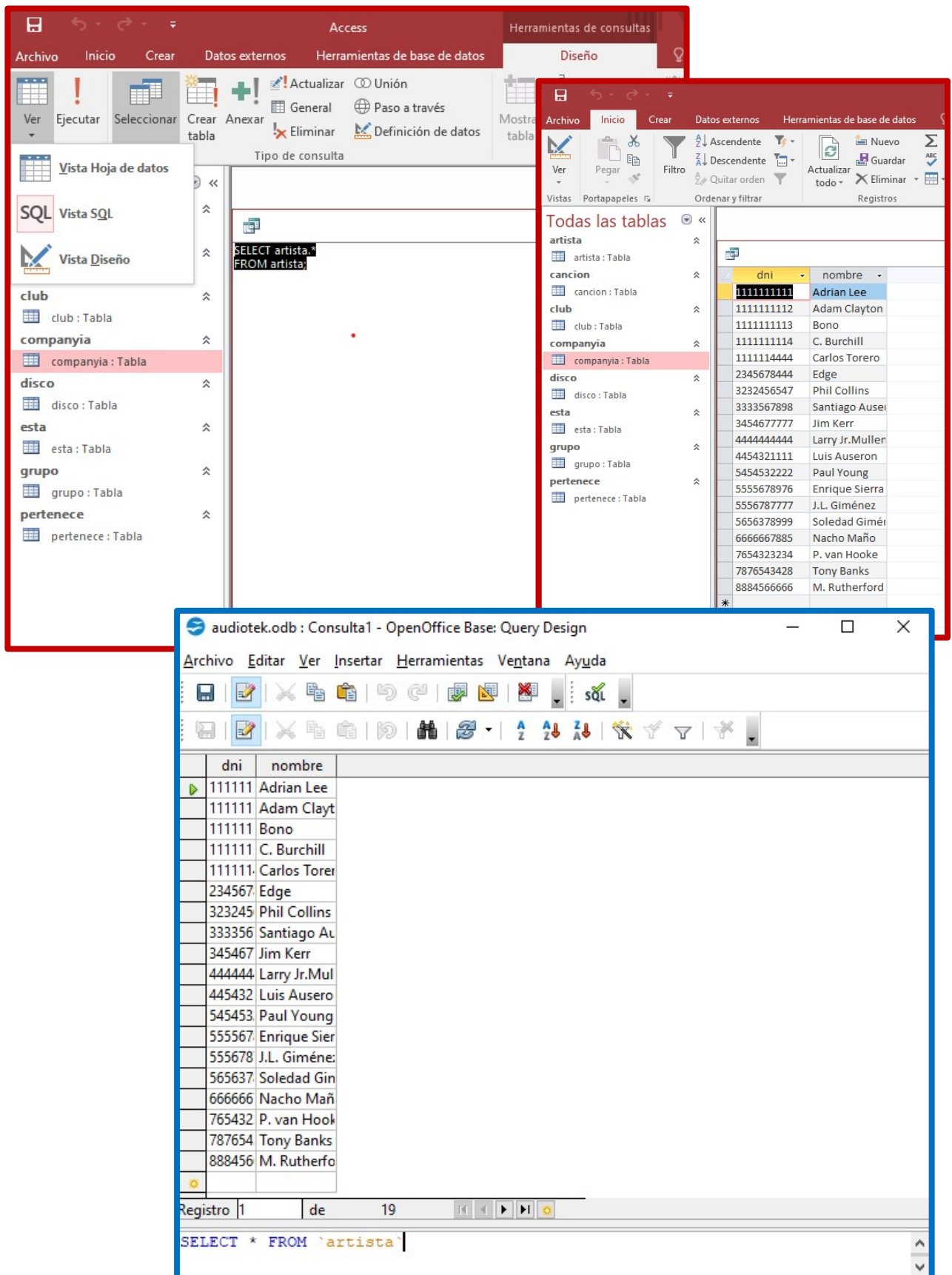


Figura 3 Acceso al editor de consultas SQL



The image displays two software interfaces side-by-side, both showing the results of an SQL query. The top interface is Microsoft Access, and the bottom interface is OpenOfficeDB.

**Microsoft Access Interface:**

- Top Menu:** Archivo, Inicio, Crear, Datos externos, Herramientas de base de datos, Herramientas de consultas, Diseño.
- Left Navigation Pane:**
  - Vista Hoja de datos
  - Vista SQL (selected)
  - Vista Diseño
  - club: Tabla
  - compañía: Tabla
  - disco: Tabla
  - esta: Tabla
  - grupo: Tabla
  - pertenece: Tabla
- SQL View:**

```
SELECT artista.*
FROM artista;
```
- Right Pane (Todas las tablas):**
  - artista: Tabla
  - cancion: Tabla
  - club: Tabla
  - compañía: Tabla
  - disco: Tabla
  - esta: Tabla
  - grupo: Tabla
  - pertenece: Tabla
- Data Grid:**

dni	nombre
1111111111	Adrian Lee
1111111112	Adam Clayton
1111111113	Bono
1111111114	C. Burchill
1111114444	Carlos Torero
2345678444	Edge
3232456547	Phil Collins
3333567898	Santiago Auser
3454677777	Jim Kerr
4444444444	Larry Jr. Muller
4454321111	Luis Auseron
5454532222	Paul Young
5555678976	Enrique Sierra
5556787777	J.L. Giménez
5656378999	Soledad Giménez
6666667885	Nacho Maño
7654323234	P. van Hooke
7876543428	Tony Banks
8884566666	M. Rutherford

**OpenOfficeDB Interface:**

- Title Bar:** audiotek.odt : Consulta1 - OpenOffice Base: Query Design
- Menu:** Archivo, Editar, Ver, Insertar, Herramientas, Ventana, Ayuda.
- Toolbar:** Includes icons for file operations, editing, and SQL execution.
- Data Grid:**

dni	nombre
111111	Adrian Lee
111111	Adam Clayt
111111	Bono
111111	C. Burchill
111111	Carlos Torer
234567	Edge
323245	Phil Collins
333356	Santiago Au
345467	Jim Kerr
444444	Larry Jr. Mul
445432	Luis Ausero
545453	Paul Young
555567	Enrique Sier
555678	J.L. Giméne
565637	Soledad Gin
666666	Nacho Mañ
765432	P. van Hook
787654	Tony Banks
888456	M. Rutherfo
- Footer:** Registro 1 de 19. SQL query: `SELECT * FROM `artista``

Figura 4 Resultados de la consulta de selección tanto en Microsoft Access como en OpenOfficeDB

### 3. Base de datos a consultar

La Base de Datos (BD) que se usará en esta práctica, contiene datos sobre un Sistema de Información (SI) basado en la música a la que se ha denominado Audiotek. Dicho SI está formado por las siguientes tablas.

**EJERCICIO:** Obtener tanto el esquema conceptual como el esquema lógico de la BD

Una vez obtenido el esquema conceptual o diagrama E-R, para cuya realización os debéis apoyar en la información suministrada en este boletín, identificar y comentar las diferencias encontradas entre vuestra solución y los diagramas proporcionados en la figura 1.

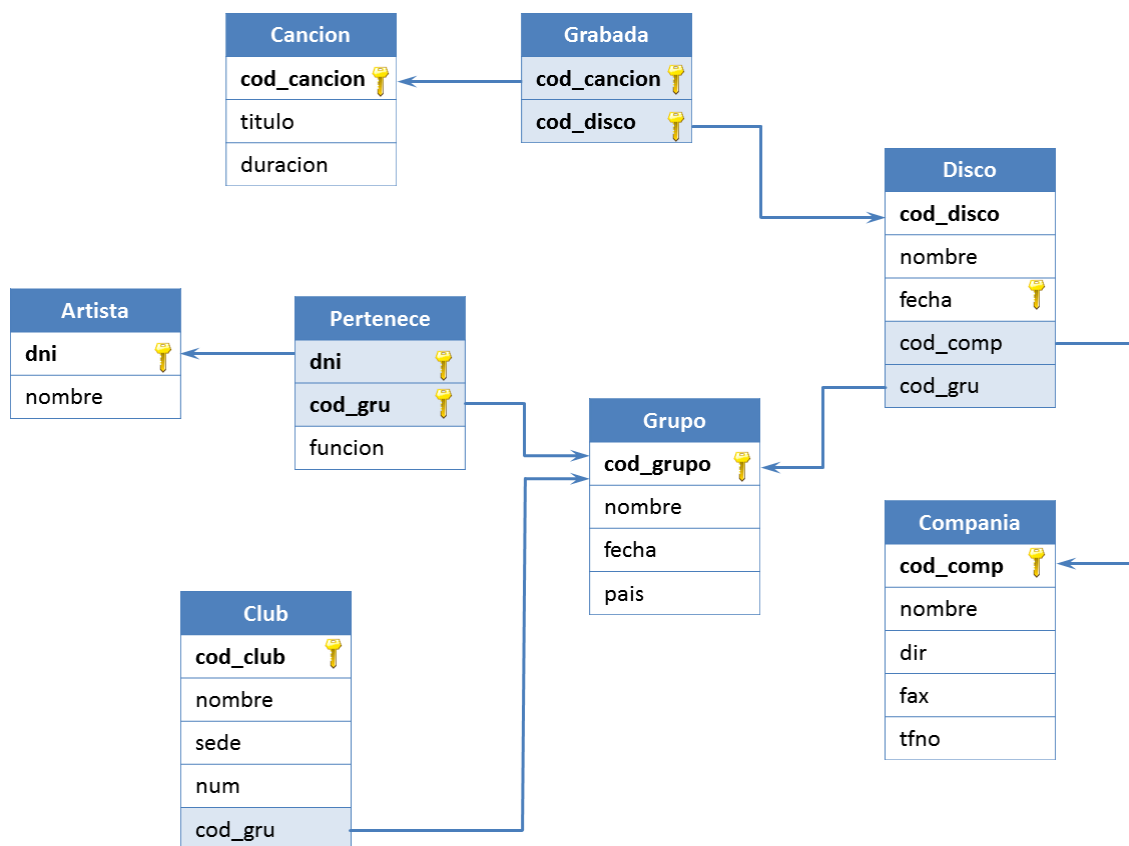


Figura 5 Diagrama de la BD

Los atributos de las tablas tienen el significado indicado a continuación:

#### Canción

- **cod\_cancion**: código de la canción (clave primaria).
- **titulo**: cómo se llama la canción.
- **duracion**: cuánto dura aproximadamente la canción (minutos).

### Compania

- **cod\_comp**: código de la compañía editora de discos (clave primaria).
- **nombre**: cómo se llama la compañía.
- **dir**: dirección postal de la compañía.
- **fax**: número de fax de contacto.
- **tfno**: número de teléfono de las oficinas.

### Disco

- **cod\_disco**: código que identifica cada disco (clave primaria).
- **nombre**: título del disco.
- **fecha**: fecha en que se editó el disco.
- **cod\_comp**: código de la compañía que ha editado el disco.
- **cod\_gru**: código del grupo que ha grabado el disco.

### Grabada

- Mantiene información sobre qué canciones hay grabadas en cada disco. Es decir, **cod\_cancion** es el código de la canción que está grabada en el disco de código **cod\_disco**.
- (**cod\_cancion**, **cod\_disco**): clave primaria.

### Grupo

- **cod\_grupo**: código del grupo (clave primaria).
- **nombre**: cómo se llama el grupo.
- **fecha**: cuándo se constituyó el grupo.
- **pais**: dónde surgió el grupo.

### Artista

- **dni**: nº de DNI del artista (clave primaria).
- **nombre**: cómo se llama el artista.

### Pertenece

- Mantiene información sobre quién forma parte de cada grupo. Es decir, el artista **dni** forma parte del grupo de código **cod\_gru** y realiza en éste la tarea especificada en **funcion**.
- (**dni**, **cod\_gru**): clave primaria.



**Club** (de fans)

- **cod\_club**: código del club de fans (clave primaria).
- **nombre**: cómo se llama el club.
- **sede**: dónde tiene la sede.
- **num**: cuántas personas pertenecen al club.
- **cod\_gru**: código del grupo que se admira.

**ALIAS**: Cuando aparece más de una tabla en el FROM, se hace uso del alias. Un carácter o varios caracteres que acompañan a los atributos, conforme van apareciendo en la consulta, agilizan su programación y permiten hacer uso de atributos que se denominen igual en distintas relaciones (tablas), entre otros usos.

#### 4. Sentencias a generar

Para completar la práctica, deberán generarse una serie de consultas SQL que permitan obtener la información indicada en cada caso. Por ejemplo, para un enunciado como:

**“Obtener los nombres de los discos del grupo U2”, debería proporcionarse la sentencia:**

```
SELECT d.nombre
FROM Disco d, Grupo g
WHERE d.cod_gru = g.cod_grupo AND g.nombre = 'U2'
```

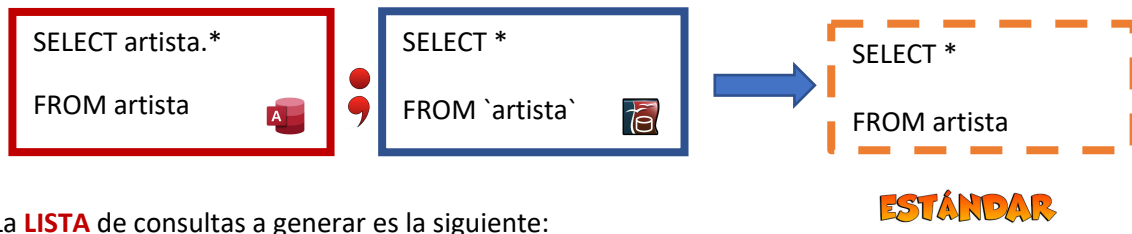
O también sería válida la siguiente solución, trabajando con subconsultas:

```
SELECT d.nombre
FROM Disco d
WHERE d.cod_gru IN (SELECT g.cod_grupo
                    FROM Grupo g AND g.nombre = 'U2')
```

**EJERCICIO:** ¡Vamos a trabajar un poquito más las consultas, contestando a las siguientes preguntas, antes de empezar con la práctica!:

1. ¿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?
2. Proponer otra solución haciendo uso del predicado EXISTS
3. ¿Por qué predicado, de los estudiados en la parte II del SQL, podrías sustituir el igual en esta parte de la sentencia:  $g.nombre = 'U2'$ ?

Antes de comenzar a programar en SQL, si os fijáis en la figura 4, el código de las consultas difiere en algunas cosas. Son simplemente pequeños detalles de sintaxis que, como se comentó en clase, utilizan las compañías de software para diferenciarse en el mercado. Es por ello que vuestras consultas, si hacéis uso del entorno gráfico para ayudaros o del wizard, deberéis pulirlas y dejarlas con notación estandarizada (ver ANEXO), para así garantizar su funcionalidad en cualquier software de bases de datos o Sistema Gestor de Bases de Datos.



La **LISTA** de consultas a generar es la siguiente:

1. Obtener el nombre de los grupos de música que no sean de España.
2. Seleccionar sólo el título de todas las canciones que duren más de 5 minutos.
3. Obtener el número total de discos de la base de datos.
4. Obtener cuántos minutos dura la canción más larga.
5. Según los datos en la base de datos, obtener las distintas funciones que un artista puede desempeñar dentro de un grupo.
6. Selecciona tanto el nombre como la sede de los clubes de fans con un número de socios entre 100 y 500.
7. Obtener los nombres de los clubes de fans que no tienen definida la dirección de su sede o el número de admiradores.
8. Obtener los nombres de los grupos por orden alfabético inverso de nombre (de la Z a la A).
9. Obtener el nombre de los grupos que sólo constan de una palabra.
10. Obtener el nombre y número de integrantes de los clubes de fans cuyo nombre contiene la cadena "100%".
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.
12. Seleccionar el nombre de todos los artistas que pertenezcan a grupos españoles.
13. Seleccionar los discos (nombre y fecha de publicación) que contengan alguna canción con una duración que sobrepase los 5 minutos.
14. Obtener los nombres de las canciones que dan nombre al disco en el que aparecen, ordenadas alfabéticamente.
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).
16. Obtener cuántos clubs de fans tiene cada grupo, junto con el nombre del grupo.
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.
18. Seleccionar el nombre del club de fans que tiene más integrantes, indicando además cuántos fans lo componen.
19. Seleccionar los títulos de las canciones tanto de mayor duración como de menor duración, indicando cuánto dura cada una.

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.
21. Seleccionar el nombre de todos los discos del grupo que fue fundado hace más tiempo.
22. Seleccionar el código, nombre y dirección de todas las compañías discográficas que no han grabado ningún disco de grupos ingleses.
23. Obtener el nombre de las compañías discográficas que sólo han publicado trabajos de grupos ingleses.
24. Seleccionar nombre y número de teléfono de aquellas compañías discográficas que hayan grabado todos los discos de alguno de los grupos.
25. Obtener los nombres de los grupos en los que todos los integrantes tienen diferentes funciones.
26. Seleccionar el número total de fans de cada grupo español, además de su nombre, ordenados de mayor a menor número de fans.
  - a. OPCIONAL: modificar la consulta para que también aparezcan los grupos sin ningún club de fans.
27. Seleccionar los grupos (nombre y número total de componentes) formados por más de tres artistas.
28. Obtener cuántas canciones han sido grabadas por cada compañía discográfica, junto con el nombre y la dirección de la compañía.
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.
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.
31. Seleccionar el título y duración de la canción más larga, pero sólo si es la única con esa duración.
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.
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.
34. Obtener el disco con menor duración total, indicando cuántos minutos dura.
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.

## 5. Entrega de la práctica

La entrega de esta práctica será **INDIVIDUAL**, no por grupos, de forma que cada uno entregará una memoria que deberá contener, para cada uno de los casos del apartado anterior, la consulta SQL que devolvería la información requerida, así como una captura de pantalla del resultado.

La fecha de entrega será el **lunes, 20 de abril de 2020**, a través de una tarea habilitada en la plataforma Moodle (ADD de la UZ).

## ANEXO: Sentencia SELECT

```
SELECT [ALL | DISTINCT] comalista_ítem_seleccionado | *  
FROM tabla  
[WHERE expresión_condicional]  
[ORDER BY comalista_referencia_col]
```

- **ALL** : Permite la aparición de filas idénticas (valor por defecto).
- **DISTINCT**: No permite la aparición de filas idénticas.
- La expresión\_condicional está formada por un conjunto de predicados combinados con las conectivas lógicas AND, OR y NOT.
- Los predicados utilizados permiten comparar columnas:
  - predicados de comparación: =, <>, >, <, >=, <=.
  - predicado LIKE: permite comparar una tira de caracteres con un patrón.
  - predicado BETWEEN: permite comprobar si un escalar está en un rango.
  - predicado IN: permite comprobar si el valor está dentro de un conjunto.
  - predicado IS NULL: permite comprobar si el valor es nulo.

```
SELECT [ALL | DISTINCT] comalista_item_seleccionado | *  
FROM comalista_referencia_tabla  
[WHERE expresión_condicional]  
[GROUP BY comalista_referencia_col]  
[HAVING expresión_condicional]  
[ORDER BY comalista_referencia_col]
```

- comalista\_item\_seleccionado: información a obtener de la base de datos (si se indica el carácter \* se selecciona toda la información disponible).
  - Pueden usarse funciones para trabajar con conjuntos: COUNT (contar filas), SUM (suma de valores), MAX, MIN (máximo y mínimo de valores), AVG (valor medio).
- **FROM** comalista\_referencia\_tabla: especifica de qué tablas se obtiene la información buscada.
- **WHERE** expresión\_condicional: expresa una condición que deben cumplir las filas de la consulta resultante.
- **GROUP BY** comalista\_referencia\_col: permite formar consultas agrupadas para extraer información global sobre los grupos formados.
- **HAVING** expresión\_condicional: condición sobre los grupos formados.
- **ORDER BY** comalista\_referencia\_col: ordena por una o varias columnas.