



# Select y From, inseparables

# 04

## Sumario

Tema 4: Select y From, inseparables.....	1
4.1 SELECT Y FROM: JUNTOS PERO NO REVUELTOS.....	2
4.1.1 La cláusula FROM.....	2
4.1.2 Pequeños trucos a seguir en la escritura de SQL.....	3
4.2 NUEVAS PRACTICAS EN SQLITE STUDIO.....	3
4.2.1 Escribiendo una primera consulta formal.....	4
4.2.2 Ejecutando la consulta.....	4
4.2.3 Comentarios en una sentencia.....	5
4.3 GUARDANDO LA CONSULTA.....	6
4.3.1 Guardar la sentencia generada.....	6
4.4 ABRIENDO SENTENCIAS DE CONSULTAS GUARDADAS.....	7
4.4.1 Ejecutando dos archivos.....	7
4.5 FORMATOS DE EXPORTACION.....	8
4.5.1 Exportando resultados en formato CSV.....	8
4.6 NUEVAS CONSULTAS.....	9
4.6.1 Creación, ejecución y guardado de consultas.....	10
4.6.2 Conveniencia de formatear el código SQL.....	11
4.7 EL CARACTER COMODIN ASTERISCO.....	13
4.7.1 Ahorro de trabajo con esta característica.....	13
4.8 BASES DE DATOS TRANSPARENTES.....	14
4.8.1 Bases de datos Conectadas y Desconectadas.....	14
4.9 HISTORIAL DE CONSULTAS.....	15
4.9.1 Recordando Consultas pasadas.....	16
4.9.2 Eliminar el historial.....	16

Descarga Materiales para prácticas: <https://tinyurl.com/01-Modulo01-BdsPlanas-DEMO>



## 4.1 SELECT Y FROM: JUNTOS PERO NO REVUELTOS

Una vez vista la parte teórica de las ramas de comandos de **SQLite** a nivel de introducción, comenzaremos a estudiar dos de las instrucciones principales a la hora de desarrollar nuestros conocimientos en este Gestor de Base de Datos Relacional.

### 4.1.1 La cláusula FROM

De ahora en adelante, el comando **SELECT** no puede obviar la cláusula **FROM**. Es decir, la cláusula **FROM** permite indicar en qué tabla o tablas (también en que consulta o consultas) se encuentra el o los campos especificados en el comando **SELECT**.

Estas tablas o consultas se separan por medio de comas (,), y, si sus nombres están compuestos por más de una palabra, éstos se escriben entre corchetes ([nombre de campo]). Veamos las muestras siguientes y su “traducción”:

```
SELECT Apellido1, Apellido2, Nombre, TelefonoDeContacto
FROM Clientes;
```

```
SELECT [Primer Apellido], [Segundo Apellido], Nombre, [Telefono de Contacto]
FROM Clientes
```

Ambas mostrarían los datos de los **Apellidos y Nombre**, además del **Teléfono de contacto**, que estarán situados en la tabla Clientes.

Las dos ejecuciones son iguales, lo único que cambia es que en la segunda muestra, al haber nombres de campo con espacios, éstos se tendrán que encerrar entre corchetes.

También sería correcta la siguiente muestra:

```
SELECT Clientes.Apellido1, Clientes.Apellido2, Clientes.Nombre, Clientes. Ciudad
FROM Clientes
```

En ésta, se antepuso el nombre de la tabla (separada por un punto) a cada uno de los nombres de campo.

Desglosemos las partes de la **SELECT** por separado:

**SELECT** <campo\_1>[,<campo\_2>,...<campo\_n>]

Relación de campos a recuperar de la tabla. Si queremos recuperar todos los campos, usaremos el símbolo \* (asterisco) pero, en este caso, el orden de los campos no será especificado por nosotros. Los campos pueden ser expresiones aritméticas formadas por campos de las tablas, constantes u otras expresiones aritméticas.

**FROM** <tabla\_1> [alias\_tabla\_1][,<tabla\_2> [alias\_tabla\_2],... <tabla\_n> [alias\_tabla\_n]]



Indicamos la(s) tabla(s) que intervienen en la consulta. Podemos “renombrar” una tabla mediante un alias para evitar teclear el nombre completo de las mismas.

[**ORDER BY** <campos\_de\_ordenación> [**DESC/ASC**]]

Ordena las filas o registros recuperados según los <campos\_de\_ordenación>. Si se indica más de una columna, los resultados se ordenan primero por el primer campo, después por el segundo, etc... La subcláusula **DESC** indica orden descendente y **ASC** indica orden ascendente (que es la opción por defecto).

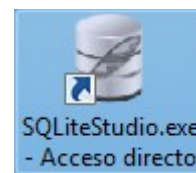
#### 4.1.2 Pequeños trucos a seguir en la escritura de SQL

Recomiendo, bajo mi humilde punto de vista, seguir las recomendaciones siguientes, con el fin de realizar la escritura en este lenguaje de la manera más eficiente. Por ello no estaría mal tener en cuenta los ocho puntos siguientes:

- Los nombres de comandos y cláusulas, es decir las palabras reservadas del lenguaje, siempre las escribo con Mayúsculas. Por ejemplo **SELECT, FROM, WHERE, GROUP BY...**
- Lo mismo recomiendo para operadores lógicos (**AND, OR, NOT**), Operadores de comparación (**LIKE, BETWEEN, IN**), Funciones de agregado, etc.
- A las tablas le suelo poner el prefijo **tbl** en minúsculas y el comienzo del nombre en mayúsculas. Por ejemplo **tblClientes, tblPedidos, tblStock...**
- Si es un nombre de tabla compuesto, utilizo la metodología **tblPedidosDeClientes, tblControlDeStock**, sin dejar espacios entre sus nombres.
- En este curso generalmente a las **Consultas (queries)**, al guardarlas les pondremos el prefijo **qry**, seguido de unos números y las guardaremos en formato **TXT** o **SQL** para abrirlas fácilmente mediante el **Bloc de Notas de Windows**. Por ejemplo **qry0101-01-01.txt** (las cuatro primeras cifras corresponden al número de módulo y número de base de datos; las restantes, al número de tema y número de query). Para las **Vistas**, utilizo el prefijo **vie**, como se verá más adelante.
- Para los nombres de campo, nunca uso espacios en blanco (si son compuestos). Mi norma de escritura es: **TipoDeClientes, FormaDePago, CodigoDeProveedor, ClasificacionEdad...**
- Si voy a utilizar consultas de varias tablas, tengo la costumbre a los nombres de campo ponerles una terminación especial (de tres o cuatro caracteres en mayúsculas) para saber a qué tabla me refiero. Por ejemplo, si tengo que realizar consultas para relacionar las tablas **tblPelículas** y **tblGeneros**, mediante los campos **IdGenero**, éstos los especificaré como **IdGeneroPEL** e **IdGeneroGEN**. En todo momento se sabe de qué campo y tabla se está hablando.
- Intento utilizar lo menor posible los acentos, “ñ” y espacios, así como otros caracteres especiales en nombres de tablas, campos y consultas (el lenguaje SQL está escrito en inglés). Si los usamos, al exportar a otros sistemas estos caracteres (e incluso espacios en blanco) podrían provocar más de un quebradero de cabeza.

#### 4.2 NUEVAS PRACTICAS EN SQLITE STUDIO

Aunque ya hemos visto un primer avance sencillo en el tema anterior con la cláusula **SELECT**, entramos de lleno en la programación de este lenguaje. Para ello, utilizaremos nuestra aplicación **SQLiteStudio**, que deberemos ejecutar debidamente a través del acceso directo situado en el **Escritorio** de nuestro equipo.





### 4.2.1 Escribiendo una primera consulta formal

En esta primera incursión se pretende visualizar toda la información de los **Empleados** de nuestra empresa (situada en la tabla **tblEmpleados**), mostrando únicamente los campos que se indiquen.

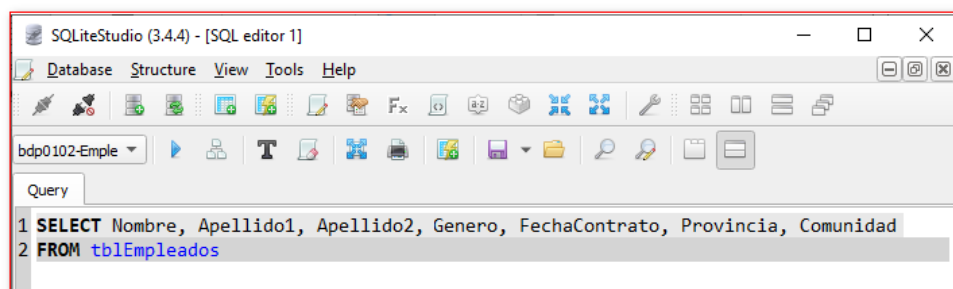


Para nuestro curso, se entiende por **query** o **consulta** cualquier petición de datos hacia una o varias tablas de una **Base de datos**. Estas **sentencias** suelen escribirse en lenguaje **SQL** (Standard Query Language) y pueden realizar operaciones de acceso y manipulación de la información almacenada.

En las prácticas a realizar de momento trabajaremos con **SELECT** y **FROM**.

#### ☐ PRACTICA 04.01.- Escribiendo sentencias en SQL

- 01. Abra la aplicación **SqliteStudio**.
- 02. La BD **bdp0102-Empleados-TRABAJO** deberá ser la activa.
- 03. La tabla **tblEmpleados** deberá estar seleccionada.
- 04. Cierre, si fuera necesario, la ventana donde aparecen las etiquetas **Data**, **Structure**, **Constraints**, etc..., es decir, **SqliteStudio** deberá aparecer con la zona central en gris.
- 05. Abra el **Editor SQL**, situado en la **Barra de Herramientas** principal.
- 06. En este punto aparecerá una etiqueta en la parte inferior con el nombre **SQL editor 1** (también aparece esta información en la **Barra de título** de la aplicación).
- 07. Con el fin de ver mejor la futura ejecución de la sentencia que vamos a escribir, haga clic en el icono **Results below the query**.
- 08. Escriba la muestra siguiente en la zona interior:



### 4.2.2 Ejecutando la consulta

En este momento queremos visualizar el resultado de la consulta. Si al ejecutarla aparece un mensaje especial de color rojo en la zona inferior de la aplicación (**Status**) de **SqliteStudio**, significa que algo mal se ha escrito (generalmente muchos de los errores son eso, mal especificados los nombres de campo, o tabla, olvido de alguna coma, etc.). Pero si por el contrario se muestra el resultado, eso significa que todo ha ido bien, así como además se muestra el tiempo de respuesta.



Recuerde: pulsando la tecla indicada, puede ejecutar la consulta.

**F9**

Para ejecutar la consulta (query) se deberá hacer clic en el icono ideal representado para ello. En ese momento y en la etiqueta inferior denominada como **Grid View**, se verá el resultado de la misma. En nuestro caso, se





visualizarán los **7** campos (columnas) elegidos mediante la **SELECT**, así como los datos obtenidos (**100** registros).

#### ☐ PRACTICA 04.02.- Vista de resultados

- 01. Una vez escrita la consulta, para ver su resultado en pantalla, deberá hacer clic en el icono comentado.
- 02. Aparecerá una nueva vista con la consulta: en la parte inferior de la sentencia **SQL** se pueden apreciar las columnas (nombres de campo) así como los datos que se han obtenido.
- 03. Observe el resultado de la misma, navegando por la información obtenida.
- 04. Aprecie también, el número de registros conseguidos.
- 05. Tenga en cuenta también, el tiempo que ha tardado en ejecutarse dicha consulta.

	Nombre	Apellido1	Apellido2	Genero	FechaContr	Provincia	Comunidad
1	Eduardo	Esquivel	Baeza	M	1999-02-24	Zamora	Castilla Leon
2	Cesar	Vergara	Garica	M	2019-03-11	Tarragona	Catalunya
3	Luis Miguel	Otero	Solis	M	2005-04-30	Soria	Castilla Leon
4	Fatima	Avalos	Valencia	F	2010-02-03	Sevilla	Andalucia
5	Lidia	Irizarry	Romo	F	2019-06-01	Zaragoza	Aragon
6	Santiago	Cazares	Bustamante	M	2011-12-21	Soria	Castilla Leon
7	Estrella	Oquendo	Centeno	F	2016-04-06	Teruel	Aragon
8	Damaso	Diaz	Pizarro	M	2014-08-22	Zaragoza	Aragon
9	Enrique	Romero	Olivas	M	2018-05-15	Valencia	Valencia
10	Marcela	Madrid	Suarez	F	2015-06-30	Oviedo	Asturias
11	Carlota	Delarosa	Sepulveda	F	2015-07-07	Alava	Pais Vasco
12	Alexis	Olvera	Perez	M	2010-05-17	Toledo	Castilla la Mancha
13	Casia	Briseno	Madrigal	F	2005-04-06	Valladolid	Castilla Leon
14	Amador	Chapa	Gurule	M	2010-05-05	Burgos	Castilla Leon

#### 4.2.3 Comentarios en una sentencia

Antes de guardar la consulta, recomiendo realizar comentarios en la sentencia escrita. De esta manera y si más adelante recuperamos dicha sentencia, al tenerla perfectamente “documentada” sabremos en todo momento lo que hace. Veamos un ejemplo:

```

1 -- Creación de mi primera sentencia formal en SQL
2 /* Se pretende visualizar la información de 7 columnas
3 situadas en la tabla tblEmpleados, con 100 registros */
4
5 SELECT Nombre, Apellido1, Apellido2, Genero, FechaContrato, Provincia, Comunidad
6 FROM tblEmpleados
  
```

Para inserta comentarios, deberemos seguir los siguientes pasos:

- Cuando sea una sola línea, se utilizarán los guiones simples “--” (sin comillas), como inicio.
- Y cuando sean múltiples líneas, lo natural es utilizar la simbología “/\*” de apertura y para el cierre ésta: “\*/” (sin comillas para ambas).

#### ☐ PRACTICA 04.03.- Documentando una sentencia

- 01. Dentro de la **Query**, posicione el puntero del ratón en la primera línea y antes del **SELECT**.
- 02. Pulse como mínimo tres veces [Enter] para dejar líneas en blanco.
- 03. Tomando como ejemplo las líneas documentadas de la imagen anterior, explique esta primera sentencia, escribiendo dicha información en el interior de la misma.



### 4.3 GUARDANDO LA CONSULTA

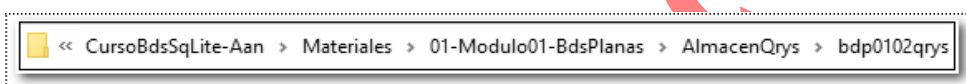
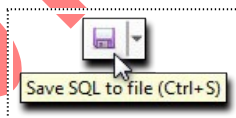
En cualquier trabajo informático, siempre se espera guardar en un archivo, la sentencia y si fuera posible, guardar además el resultado. Para la primera opción **SqliteStudio** facilita el guardado de la query en dos formatos: **SQL** y **TXT**. Para este curso, se optará por ambas opciones, puesto que así se podrá abrir mediante el **Bloc de Notas** de **Windows** y comprobar rápidamente el código para futuros estudios.

Más adelante también (y en próximas prácticas) también se aprenderá a exportar el resultado de cualquier query en los formatos que **SqliteStudio** tiene disponibles: **PDF**, **HTML**, **CSV**, etc.

#### 4.3.1 Guardar la sentencia generada

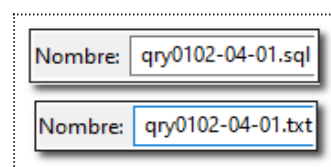
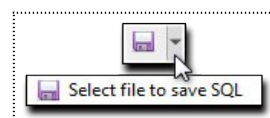
Se podrá guardar la citada sentencia con el fin de utilizarla posteriormente. Es decir, si una vez generada la query deseamos tenerla a nuestra disposición en el futuro, localizando y haciendo clic en el icono ideal, nos preguntará ubicación de carpeta y nombre de archivo.

Aprovecharemos para guardar esta primera sentencia en los formatos ya comentados (**.txt** y **.sql**) para conocer ambas posibilidades (recordemos que se puede tener un archivo con el mismo nombre pero con extensiones diferentes).



#### ☐ PRACTICA 04.04.- Guardar la sentencia en archivos diferentes

- 01. Haga clic en el icono citado.
- 02. Ya en el Cuadro de diálogo que aparece, llamado *Save to file*, localice la carpeta principal del curso.
- 03. Acceda en la carpeta **Materiales**.
- 04. Siga la ruta **01-Modulo01-BdsPlanas→AlmacenQrys**.
- 05. Dentro de ella, entre en la carpeta **bdp0102qrys**.
- 06. En el Cuadro de texto, **Nombre de archivo**, escriba **qry0102-04-01.sql**.
- 07. Guarde los cambios.
- 08. Ahora, haga clic en el desplegable de este icono para seleccionar la nueva opción que se muestra.
- 09. Guarde este archivo en formato de texto puro (**qry0102-04-01.txt**).
- 10. Mediante el **Explorador de Windows**, localice ambos archivos dentro de la carpeta **bdp0102qrys**.
- 11. Haga doble clic sobre el archivo de texto: se abrirá el **Bloc de Notas** con el código escrito de la consulta y mostrará su contenido. Ciérrelo.
- 12. Haga doble clic sobre el archivo en formato **sql**: también se abrirá el **Bloc de Notas** con el código escrito de la consulta y mostrará su contenido. Ciérrelo.
- 13. Regrese nuevamente a **SqliteStudio**.
- 14. Sin salir de la aplicación cierre la ventana de la consulta: aparecerá una gran zona gris indicativa de que no hay ninguna actividad ejecutándose.





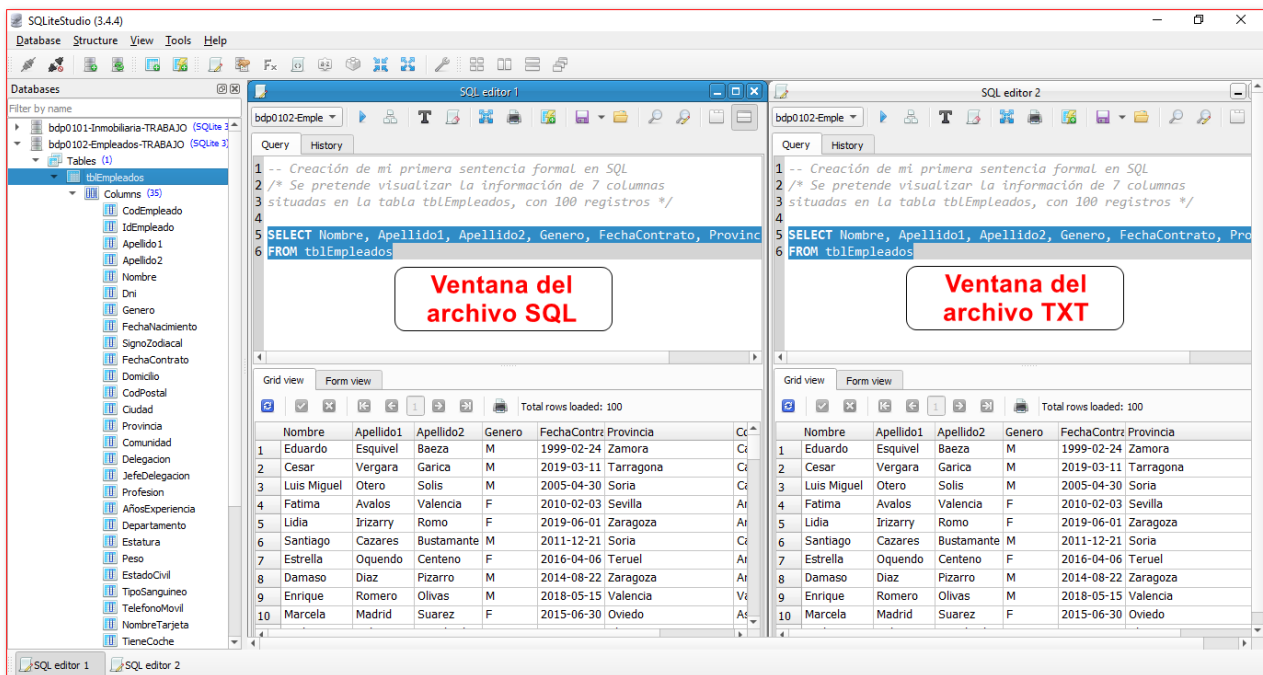
Para una consulta, su nombre **qry9999-99-99**, las cuatro primeras cifras de la izquierda corresponden al número de módulo y número de base de datos; las restantes, al número de tema y número de **query** para ese tema.

## 4.4 ABRIENDO SENTENCIAS DE CONSULTAS GUARDADAS

Teniendo **SQLiteStudio** activo, vamos a ver que fácil es generar las dos sentencias guardada en la carpeta correspondiente.

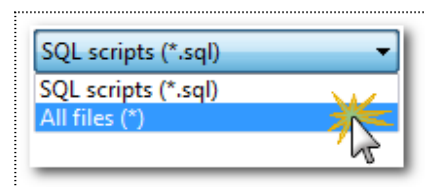
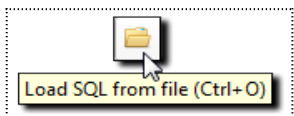
### 4.4.1 Ejecutando dos archivos

Además, veremos que ambos archivos guardados con el código **Sql** (con los tipos de archivo **Sql** y **Txt** respectivamente), se pueden abrir en dos ventanas separadas. Pero en primer lugar siempre se tendrá que abrir la ventana que utiliza el **Editor SQL**, haciendo clic en el icono asignado para tal fin. He aquí una representación gráfica de lo que se desea obtener:



## ☐ PRACTICA 04.05.- Abriendo archivos

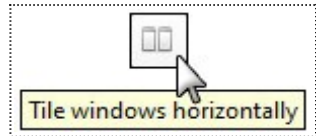
- 01. Abra el **Editor SQL**.
- 02. Abierta la ventana del mismo, localice y haga clic en el icono **Load SQL from file**.
- 03. Ya en el Cuadro de diálogo que aparece, llamado **Open file**, observe si ya le espera la carpeta **bdp0102qry**.
- 04. Si no fuera así, búsquela adecuadamente.
- 05. Verá que solamente aparece el archivo guardado en **SQL**.
- 06. Ábralo en pantalla.
- 07. Abra un nuevo **Editor SQL** y si es posible, maximícelo.
- 08. Nuevamente, haga clic en el icono **Load SQL from file** (o pulse la secuencia de teclas **[Ctrl]+[O]**).
- 09. En el desplegable situado en la zona derecha de **Nombre de archivo**, seleccione **All files (\*.\*)**. De esta manera aparecerán ambos archivos.







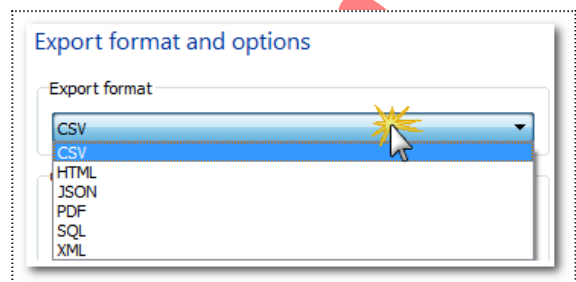
- 10. Observe como designa **SqliteStudio** a los archivos SQL: **Scripts**.
- 11. Haga doble clic en el archivo en formato **.txt**: aparecerá el script de la consulta guardada en este documentos de texto.
- 12. Como seguramente una ventana solapa a la anterior, localizando y haciendo clic en el icono **Tile windows horizontally**, podrá conseguir la visualización de ambos **scripts**.
- 13. Haga clic en el icono correspondiente para visualizar el resultado de ambas consultas.
- 14. Cierre una de las ventanas.



## 4.5 FORMATOS DE EXPORTACION

Una vez realizadas las practicas de guardado de archivos, veremos como se pueden obtener los datos de la única consulta que hemos realizado.

De esta manera, nuestra aplicación puede exportar los datos en varios formatos, siendo los habituales el **CSV** (que puede abrirse en **Excel**, por ejemplo), el ya muy utilizado **PDF**, creación de una página web en **HTML** y otros.



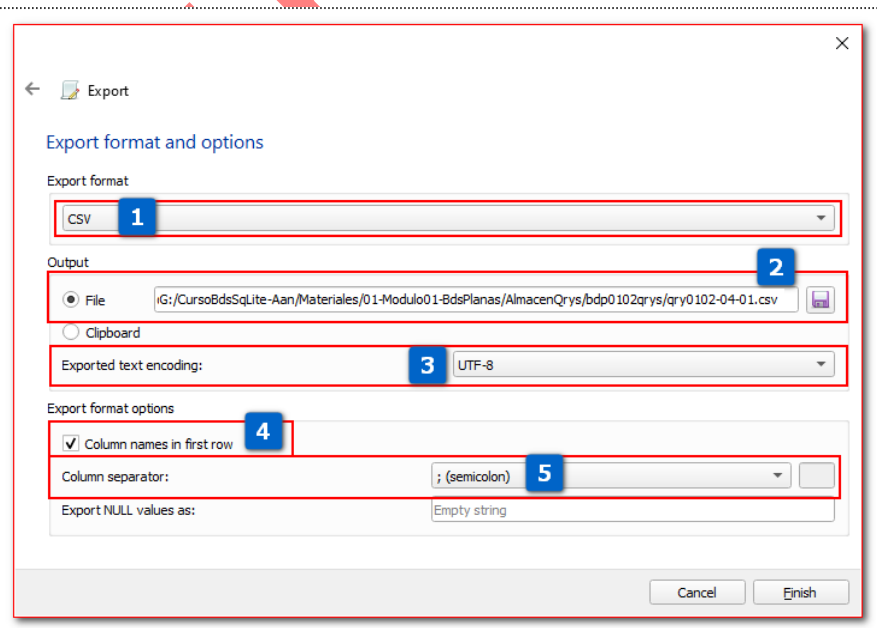
### 4.5.1 Exportando resultados en formato CSV

Localizando y haciendo clic en el icono adecuado, aparecerá una ventana inicial, en donde se muestra el origen de la base de datos correspondiente y el **script**.

En la segunda ventana, se deberá seleccionar (ver el punto 1) el tipo de archivo.

Mediante el punto 2, se deberá decidir en que carpeta y que nombre se aplicará al futuro archivo generado.

El paso 3 es el responsable de que el resultado de la consulta muestre el contenido del texto correctamente decodificado, sin errores ni caracteres extraños.



El paso 4 es interesante, puesto que los nombres de las columnas (**nombres de campos**) aparecerán en el archivo como encabezados.

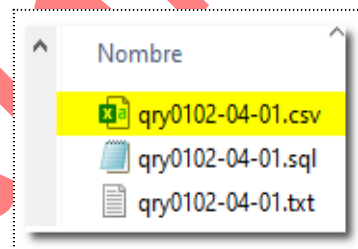
El paso 5 es muy importante para organizar los datos y adecuarlos correctamente en las columnas.





#### ☐ PRACTICA 04.06.- Exportación de datos

- 01. Obligatoriamente, el **Editor SQL** tiene que estar abierto y la consulta **qry0102-04-01.sql** ejecutada.
- 02. Localice y haga clic en el icono **Exports results**.
- 03. En la primera ventana que aparece, llamada **Export**, observe la información existente y haga clic en el botón **Next**.
- 04. En la siguiente ventana realice las modificaciones oportunas, de acuerdo a la información facilitada: guardar en formato **CSV** en las carpetas **AlmacenQrys/bdp0102qrys/** bajo el nombre de archivo **qry0102-04-01.csv**.
- 05. No olvide que el **UTF-8** deberá estar seleccionado en el desplegable perteneciente a **Export text encoding**.
- 06. En la zona **Export format options** deberán estar seleccionadas las opciones **Column names in first row** además en **Column separator** deberá seleccionar “; (semicolon)” como separador de columnas.
- 07. Haciendo clic en el botón **Finish**, y mediante el **Explorador de Windows**, localice en la carpeta comentada el archivo generado, el cual deberá abrirlo (si tiene **Excel** en tu equipo, observe como se abre esta aplicación con el archivo en el formato **.csv**).
- 08. Cierra el libro de **Excel** y abandona esta aplicación.
- 09. Ya en **SQLiteStudio**, cierre la consulta.



## 4.6 NUEVAS CONSULTAS

**SQL**, o lenguaje de consulta estructurado, es el lenguaje estándar para trabajar con Bases de datos relacionales. Se utiliza para realizar tareas de interacción de datos como:

- Crear tablas e insertar datos en ellas.
- Consulta y filtrado de datos.
- Modificación y actualización de datos.
- Eliminar datos o tablas.

Ya hemos visto como se trabaja en la creación de una consulta **SQL**: es una serie de instrucciones que se utiliza para extraer datos de una tabla en una Base de datos.

Pues bien, de ahora en adelante vamos a realizar muchísimas consultas, comenzando por las más sencillas hasta alcanzar cotas altas de aprendizaje y así dominar este lenguaje.

SQL editor 1

bdpPelículas20:

Query History

```

1 -- Número de películas por Género
2
3 SELECT Género AS [Género Cinematográfico],
4       COUNT( * ) AS [Núm. Películas por Género]
5 FROM tblPelículas
6 GROUP BY Género
7 HAVING COUNT(Género) >= 1;
8
9

```

Grid view Form view

Total rows loaded: 9

Género Cinematográfico	Núm. Películas por Género
1 Acción	7
2 Animación	7
3 Aventuras	11
4 Belica	3
5 Comedia	9
6 Drama	3
7 Ficción	4
8 Suspense	3
9 Terror	3

Tal y como se viene repitiendo, por ahora nuestras prácticas están enfocadas a Bases de datos con una sola tabla para este primer Módulo de aprendizaje. Pero llegará un momento (a partir del Módulo 2) que abordaremos Bases de datos más complejas con dos, tres y más tablas en su interior.



### 4.6.1 Creación, ejecución y guardado de consultas

Utilizando las mismas técnicas descritas, vamos a realizar las siguientes consultas. Véase el modelo siguiente que tomamos como referencia para futuras prácticas:

 <b>Supuesto</b>	¿Qué empleados tenemos cuyo sueldo es mayor o igual a 1000€ y menor o igual a 2500, ordenado por la columna Sueldo en descendente?	<input type="checkbox"/> qry9999-99-99
<b>SELECT</b>	Nombre, Apellido1, Apellido2, SueldoMes, EstadoCivil, Provincia, Profesion, Dni, FechaContrato	
<b>FROM</b>	tblEmpleados	
<b>WHERE</b>	SueldoMes BETWEEN 1000 AND 2500	
<b>ORDER BY</b>	SueldoMes DESC	

Script de ejemplo



Recuerde de nuevo: para una consulta, su nombre **qry9999-99-99**, las cuatro primeras cifras por la izquierda corresponden al número de módulo y número de base de datos; las restantes, al número de tema y número de query para ese tema.

Podrá apreciar que en la zona **Supuesto**, aparecerá una descripción de lo que se desea obtener (que aprovecharemos para documentar estos **scripts**).

Más a la derecha aparecerá el nombre de la consulta (éste último servirá para guardar cada una de las queries que va a realizar).

Un poco más abajo, se especifica el código **SQL** que deberá escribir en la aplicación **SqliteStudio**. Y para que sus resultados sean perfectos, se expondrá siempre parte de una captura de pantalla con ellos mostrados para que compare sus habilidades.

En este libro, tiene a tu disposición un cuadro en cada query, para que lo marque cada vez que las vaya completando (observe la figura situada encima del nombre de la query).

☐  
☒

- |                          |                                     |
|--------------------------|-------------------------------------|
| <input type="checkbox"/> | <b>PRACTICA 04.07.- Nueva query</b> |
|--------------------------|-------------------------------------|
- 01. Escriba en **SqliteStudio** la sentencia completa que abarca desde el **SELECT** hasta el **FROM** que se muestra debajo del **Supuesto**.
  - 02. Una vez ejecutada, guárdela en la carpeta **AlmacenQrys/bdp0102qrys/** bajo el nombre de archivo **qry0102-04-02.txt**.

 <b>Supuesto</b>	Mostrar los datos de los siguientes campos de la tabla, escribiéndolos de uno en uno.	<input type="checkbox"/> qry0102-04-02
<b>SELECT</b>	Dni, Apellido1, Apellido2, Nombre, Genero, SignoZodiacal, CodEmpleado, Provincia	
<b>FROM</b>	tblEmpleados	



Como se observa en la figura, no aparecen todos los datos: habrá que navegar hacia abajo para visualizarlos en la aplicación **SQLiteStudio**.

También hay que tener en cuenta que al seleccionar únicamente 8 campos, los datos son precisamente los que aparecen sobre los elegidos, ni más ni menos.

Y como no ha habido ninguna restricción más, aparecen los **100** registros que contiene esta tabla de Empleados.

De ahora en adelante, cree, ejecute y guarde las consultas siguientes con sus correspondientes nombres: **qry0102-04-03.sql** y **qry0102-04-04.sql**.

Query History

```
1 /* Mostrar Los datos de Los siguientes campos de La tabla, escribiéndolos de uno en uno.*/
2
3 SELECT Dni, Apellido1, Apellido2, Nombre, Genero, SignoZodiacal,
4 CodEmpleado, Provincia
5 FROM tblEmpleados
```

Grid view Form view

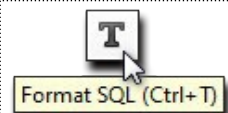
Total rows loaded: 100

	Dni	Apellido1	Apellido2	Nombre	Genero	SignoZodiacal	CodEmple	Provincia
1	728764956B	Esquivel	Baeza	Eduardo	M	Libra	INFO06	Zamora
2	372201536Q	Vergara	Garica	Cesar	M	Cancer	VENT04	Tarragona
3	296693355R	Otero	Solis	Luis Miguel	M	Cancer	FACT04	Soria
4	846911868R	Avalos	Valencia	Fatima	F	Acuario	INFO05	Sevilla
5	456654468Q	Irizarry	Romo	Lidia	F	Acuario	SEGU04	Zaragoza
6	722832058G	Cazares	Bustamante	Santiago	M	Aries	ALMA02	Soria
7	693931389Q	Oquendo	Centeno	Estrella	F	Piscis	CONT05	Teruel
8	714090472P	Diaz	Pizarro	Damaso	M	Virgo	LOGI04	Zaragoza
9	117868365T	Romero	Olivas	Enrique	M	Tauro	PERS05	Valencia
10	163559727Z	Madrid	Suarez	Marcela	F	Tauro	FINA01	Oviedo
11	778539878S	Delarosa	Sepulveda	Carlota	F	Leo	SEGU01	Alava
12	241442562T	Olvera	Perez	Alexis	M	Escorpio	COMP05	Toledo
13	587688030O	Briseno	Madrigal	Casia	F	Acuario	FINA06	Valladolid
14	583690787L	Chapa	Gurule	Amador	M	Geminis	COME01	Burgos
15	543009319H	Ojeda	Muniz	Libertad	F	Tauro	MANT05	Vizcaya
16	623034666G	Castaneda	Garrido	Carlos	M	Cancer	INFO09	Cordoba
17	712147286U	Cisneros	Zamudio	Pablo	M	Acuario	MARK02	Valencia
18	717823795L	Perez	Vallejo	Manuel	M	Geminis	COC101	Albacete

#### 4.6.2 Conveniencia de formatear el código SQL

Cuando se inicia el aprendizaje de **SQL**, a menudo las personas principiantes no prestan mucha atención al formateo de su código.

Si esta costumbre la usamos a diario, será fácil leer y será fácil corregir los posibles errores cuando aparezcan. En definitiva el formato adecuado del código SQL ayuda a los programadores a evitar errores. Obsérvese la siguiente muestra y saque sus propias conclusiones:



SQL editor 1

```
1 SELECT IdPelicula, TituloPelicula, DuracionMinutos, Genero, ClasificacionEdad
2 FROM tblPeliculas
3 WHERE Genero = "Aventuras" and ClasificacionEdad = "Familia"
4 ORDER BY TituloPelicula ASC;
5
```

**Sin formateo SQL**

Grid view Form view

Total rows loaded: 4

	IdPelicula	TituloPelicula	DuracionMinutos	Genero	ClasificacionEdad
1	1	Buenos y regulares	115	Aventuras	Familia
2	47	Piratas azules	100	Aventuras	Familia
3	30	Sombras del presente	103	Aventuras	Familia
4	8	Transformados III	96	Aventuras	Familia

SQL editor 2

```
1 SELECT
2     IdPelicula,
3     TituloPelicula,
4     DuracionMinutos,
5     Genero,
6     ClasificacionEdad
7 FROM
8     tblPeliculas
9 WHERE
10    Genero = "Aventuras" AND
11    ClasificacionEdad = "Familia"
12 ORDER
13    BY TituloPelicula ASC;
```

**Con formateo SQL**

Grid view Form view


Total rows loaded: 4

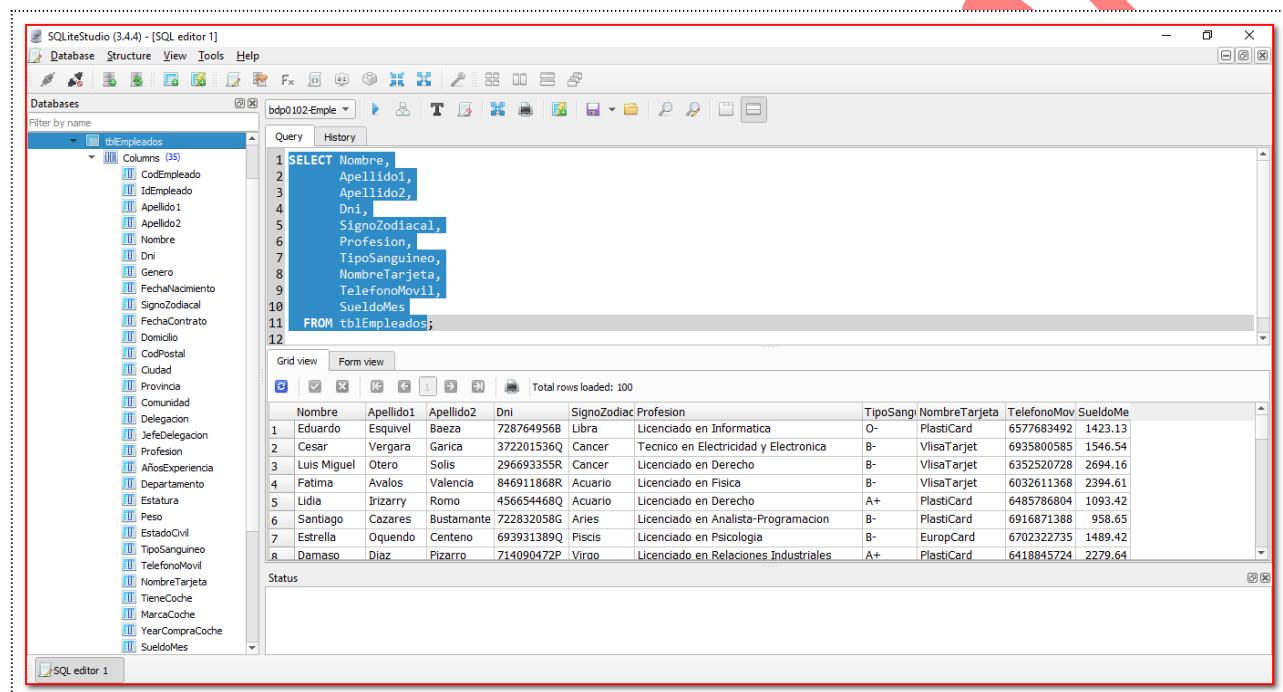
	IdPelicula	TituloPelicula	DuracionMinutos	Genero	ClasificacionEdad
1	1	Buenos y regulares	115	Aventuras	Familia
2	47	Piratas azules	100	Aventuras	Familia
3	30	Sombras del presente	103	Aventuras	Familia
4	8	Transformados III	96	Aventuras	Familia


También es verdad que para ahorrar espacio, en las muestras que expongo (capturas de pantalla) utilizaré la vieja usanza, pero la recomendación es usar asiduamente esta buena práctica.


☐ **PRACTICA 04.08.- Más consultas**

- 01. Escriba en **SQLiteStudio** las siguientes sentencias completas que abarca desde el **SELECT** hasta el **FROM** que se muestra debajo del Supuesto. Guárdelas con formato **SQL**.
- 02. Haga clic en el icono correspondiente para que el código quede formateado al estilo **SQL**.

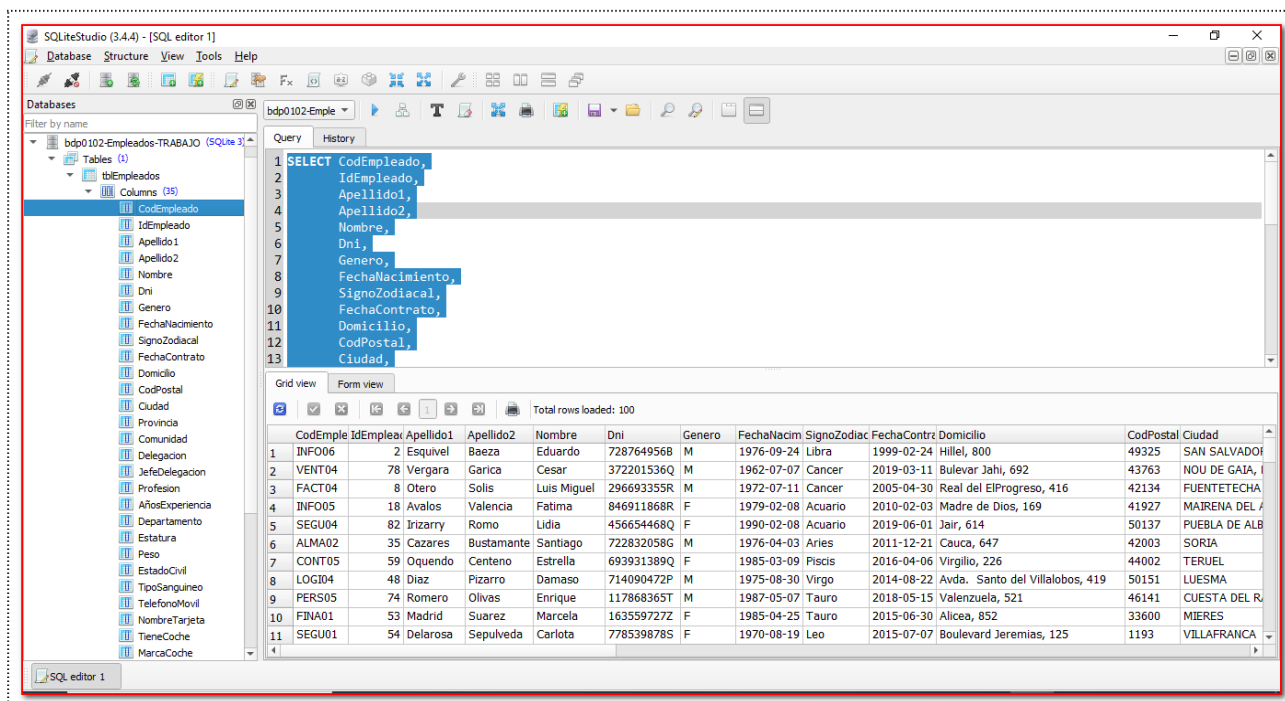
 <p><b>Supuesto</b></p>	Mostrar los datos de los siguientes campos de la tabla, escribiéndolos de uno en uno.	<input type="checkbox"/> <b>qry0102-04-03</b>
<b>SELECT</b>	Nombre, Apellido1, Apellido2, Dni, SignoZodiacal, Profesion, TipoSanguineo, NombreTarjeta, TelefonoMovil, SueldoMes	
<b>FROM</b>	tblEmpleados	



 <p><b>Supuesto</b></p>	Mostrar los datos de los siguientes campos de la tabla, escribiéndolos de uno en uno.	<input type="checkbox"/> <b>qry0102-04-04</b>
<b>SELECT</b>	CodEmpleado, IdEmpleado, Apellido1, Apellido2, Nombre, Dni, Genero, FechaNacimiento, SignoZodiacal, FechaContrato, Domicilio, CodPostal, Ciudad, Provincia, Comunidad, Delegacion, JefeDelegacion, Profesion, AñosExperiencia, Departamento, Estatura, Peso, EstadoCivil, TipoSanguineo, TelefonoMovil, NombreTarjeta, TieneCoche, MarcaCoche, YearCompraCoche, SueldoMes, TieneHijos, NumHijos, BonusAnualHijos, PuntualidadAnual, CocheEmpresa	
<b>FROM</b>	tblEmpleados	



Ha comprobado como ha tenido que escribir los **35** campos que componen la tabla **tblEmpleados** de la Base de datos **bdp0102-Empleados-TRABAJO.db**:



## 4.7 EL CARACTER COMODIN ASTERISCO

En la práctica anterior, la escritura de cada uno de los campos de las columnas de la tabla **tblEmpleados**, ha supuesto un enorme esfuerzo.


Pues bien, existe el comodín asterisco (\*) como carácter que se escribirá después del **SELECT** y que dicho símbolo sustituirá automáticamente a cada uno de los nombres de campos, con el consiguiente ahorro de escritura.

### 4.7.1 Ahorro de trabajo con esta característica

Veamos con que facilidad se pueden mostrar todos los campos de la tabla de trabajo para el caso que se nos presenta:

#### ☐ PRACTICA 04.09.- Mostrar todos los campos y sus datos con el comodín \*

- 01. Escriba en **SqliteStudio** la sentencia completa que abarca desde el **SELECT** hasta el **FROM** que se muestra debajo del Supuesto. Guárdela con formato **SQL**. Guárdela con formato **SQL**.

 <p><b>Supuesto</b></p>	<p>Mostrar todos los datos de la tabla, pero con el mínimo esfuerzo de escritura de los campos.</p>	<input type="checkbox"/> <p><b>qry0102-04-05</b></p>
<p><b>SELECT</b></p>	<p>*</p>	
<p><b>FROM</b></p>	<p>tblEmpleados</p>	



SQLiteStudio (3.4.4) - [SQL editor 1]

Database Structure View Tools Help

bdp0102-Emple

Query History

```
1 SELECT *
2 FROM tblEmpleados
```

Grid view Form view

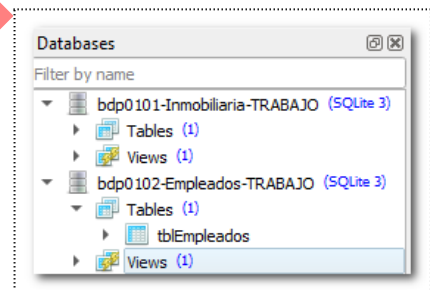
Total rows loaded: 100

	CodEmple	IdEmplear	Apellido1	Apellido2	Nombre	Dni	Genero	FechaNacim	SignoZodiac	FechaContr	Domicilio	CodPostal	Ciudad	Provincia
1	INFO06	2	Esquivel	Baeza	Eduardo	7287649568	M	1976-09-24	Libra	1999-02-24	Hillel, 800	49325	SAN SALVADOR DE PALAZUELO	Zamora
2	VENT04	78	Vergara	Garica	Cesar	372201536Q	M	1962-07-07	Cancer	2019-03-11	Bulevar Jahi, 692	43763	NOU DE GAJA, LA	Tarragona
3	FACT04	8	Otero	Solis	Luis Miguel	296693355R	M	1972-07-11	Cancer	2005-04-30	Real del ElProgreso, 416	42134	FUENTETecha	Soria
4	INFO05	18	Avalos	Valencia	Fatima	846911868R	F	1979-02-08	Acuario	2010-02-03	Madre de Dios, 169	41927	MAIRENA DEL ALJARAFE	Sevilla
5	SEGU04	82	Irizarry	Romo	Lidia	456654468Q	F	1990-02-08	Acuario	2019-06-01	Jair, 614	50137	PUEBLA DE ALBORTON	Zaragoza
6	ALMA02	35	Cazares	Bustamante	Santiago	722832058G	M	1976-04-03	Aries	2011-12-21	Cauca, 647	42003	SORIA	Soria
7	CONT05	59	Oquendo	Centeno	Estrella	693931389Q	F	1985-03-09	Piscis	2016-04-06	Virgilio, 226	44002	TERUEL	Teruel
8	LOGI04	48	Diaz	Pizarro	Damaso	714090472P	M	1975-08-30	Virgo	2014-08-22	Avda. Santo del Villalobos, 419	50151	LUESMA	Zaragoza
9	PERS05	74	Romero	Olivas	Enrique	117868365T	M	1987-05-07	Tauro	2018-05-15	Valenzuela, 521	46141	CUESTA DEL RATO	Valencia
10	FINA01	53	Madrid	Suarez	Marcela	163559727Z	F	1985-04-25	Tauro	2015-06-30	Alicea, 852	33600	MIERES	Oviedo
11	SEGU01	54	Delarosa	Sepulveda	Carlota	778539878S	F	1970-08-19	Leo	2015-07-07	Boulevard Jeremias, 125	1193	VILLAFRANCA	Alava
12	COMP05	26	Olvera	Perez	Alexis	241442562T	M	1988-11-08	Escorpio	2010-05-17	Boulevard Herodes, 414	45600	TALAVERA DE LA REINA	Toledo
13	FINA06	7	Briseno	Madrigal	Casia	5876880300	F	1964-01-21	Acuario	2005-04-06	Boulevard u, 279	47219	HONQUILANA	Valladolid
14	COME01	25	Chapa	Gurule	Amador	583690787L	M	1970-05-29	Geminis	2010-05-05	Isidoro, 918	9511	BERBERANA	Burgos
15	MAINT05	45	Ojeda	Muniz	Libertad	543009319H	F	1977-05-08	Tauro	2013-11-20	Alejandrina, 342	48140	GARBE	Vizcaya
16	INFO09	43	Castaneda	Garrido	Carlos	623034666G	M	1993-06-29	Cancer	2013-08-03	Avda. Suarez, 984	14970	VALDEARENAS	Cordoba
17	MARK02	76	Cisneros	Zamudio	Pablo	712147286U	M	1968-02-07	Acuario	2018-11-18	Loreto, 814	46017	VALENCIA	Valencia
18	COCIO1	42	Perez	Vallejo	Manuel	717823795L	M	1977-06-09	Geminis	2013-07-16	Avda. Cesar, 744	2240	MAHORA	Albacete
19	PUBL03	62	Galindo	Torrez	Miguel	271564501S	M	1968-11-22	Escorpio	2016-12-26	Avda. Santo del Cauca, 284	50152	AILES	Zaragoza

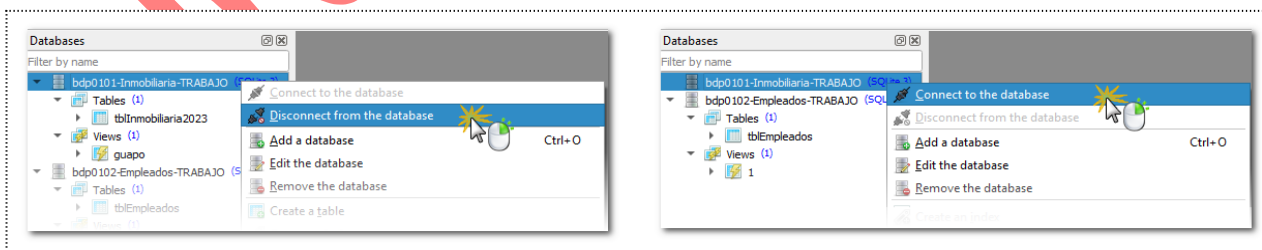
SQL editor 1

## 4.8 BASES DE DATOS TRANSPARENTES

Tal y como se está viendo, se pueden tener varias bases de datos abiertas sin problemas y realizar consultas individuales a ellas e incluso ejecutar declaraciones SQL que hagan referencia a múltiples bases de datos dentro de una sola consulta. Todo ello, gracias al mecanismo **transparente** de adjuntar bases de datos integrado en el **Editor SQL de SQLiteStudio**. En nuestro caso y de momento, tenemos activas dos de ellas como son la de **Inmobiliaria** y **Empleados**.



Existe la posibilidad de **desconectar** las bases de datos que se desee y dejar una activa para realizar las consultas pertinentes. Hay dos maneras de realizar esta operación: mediante el menú contextual que aparece al situarse sobre el nombre de cualquier BD, en el panel izquierdo **Databases** o utilizando los iconos **Connect to the database** y **Disconnect from the database**.



### 4.8.1 Bases de datos Conectadas y Desconectadas

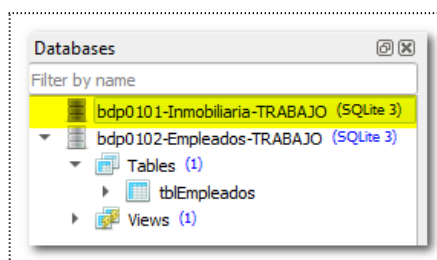
Cuando quedan "desconectadas", el icono representativo que acompaña al nombre de la BD (en el panel izquierdo de la aplicación) muestra un aspecto diferente y ésta quedará "dormida" momentáneamente: habrá que hacer clic en el icono correspondiente (o un doble clic si se desea) en el nombre de la BD para volver a hacer referencias a ella.





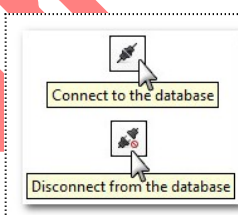
Veremos que esta opción (de **desconexión**) es muy útil ya que ayuda a centrarnos en la Bd en la que estamos realizando las consultas y que, además, próximamente existe una acción muy interesante que sin lo que estamos comentando, funcionaría. (Haremos referencia muy pronto).

En la **Barra de herramientas** principal de la aplicación también los iconos **Connect to the database** y **Disconnect from the database** actúan como un interruptor: se muestran encendidos o apagados (individualmente) según la situación.



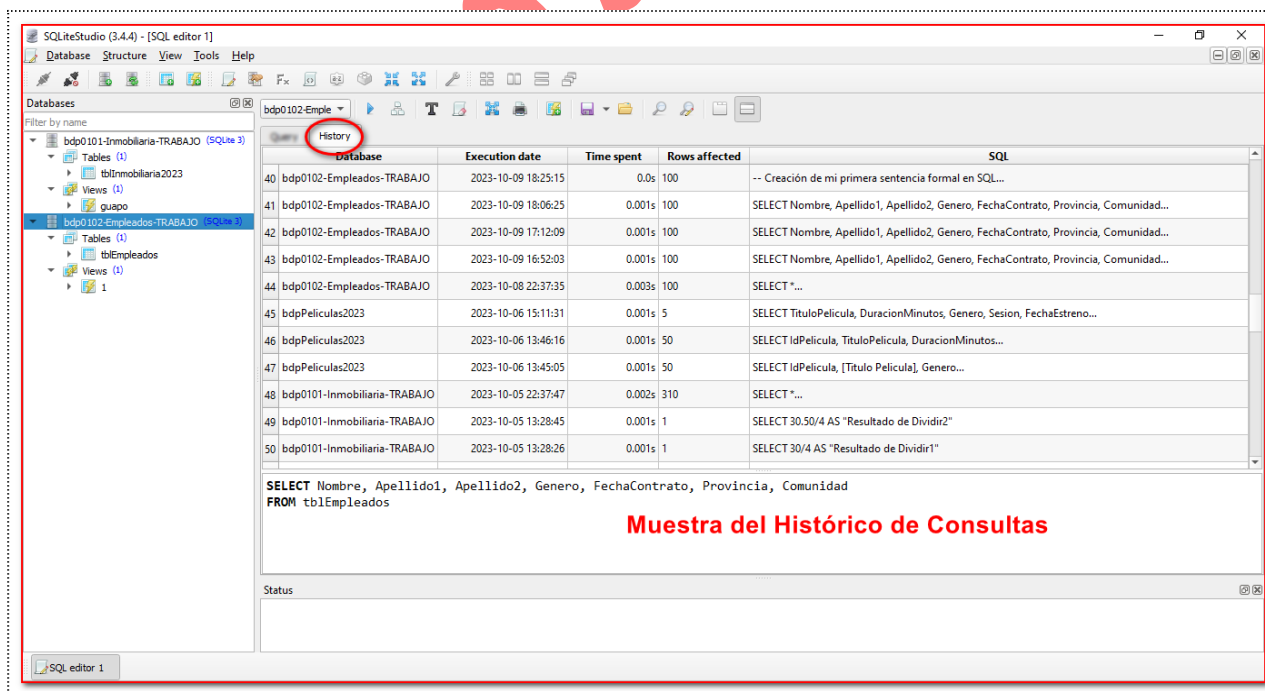
#### ☐ PRACTICA 04.10.- Desactivar y Activar Bases de datos

- 01. En **SQLiteStudio**, haga que las dos únicas Bases de datos de trabajo queden **desconectadas** en la aplicación, usando las opciones oportunas en el menú contextual correspondiente.
- 02. Observe los iconos que aparecen a la izquierda de los nombres de las Bd: tienen un aspecto diferente al habitual.
- 03. Seguidamente, elija una de ellas y vuelva a dejarla **"conectada"** utilizando los iconos que se muestran a la derecha.
- 04. Finalmente, ambas BD deberán estar conectadas.



## 4.9 HISTORIAL DE CONSULTAS

Nuestra aplicación no deja de sorprendernos habitualmente. Y una de las opciones un tanto curiosas, consiste en poder recordar en el tiempo, las consultas que se han realizado y además de visualizar el código, también ejecutarlas con sus correspondientes resultados. Y eso se consigue en la etiqueta **History**, que se encuentra en el **Editor SQL**.







Esta opción no tiene nada que ver con el guardado de consultas en archivos **SQL** y **TXT**: el historial queda administrado en **SqliteStudio** de manera temporal. Eso sí, existe un número limitado de queries guardadas, de acuerdo al número especificado en la **Configuración** de la aplicación, tal como se verá en temas futuros.

#### 4.9.1 Recordando Consultas pasadas

Tan fácil como ejecutar el **Editor SQL** y hacer clic en la etiqueta **History**. Una vez en esta vista, navegaremos por la lista y observaremos como la aplicación nos muestra en columnas, características como la Fecha y hora de ejecución, nombres de las Bds implicadas, además de una pequeña representación del código SQL aplicado.

#### ☐ PRACTICA 04.11.- Mostrar consultas pasadas y ejecutarlas

- 01. En el **Editor SQL**, localice y haga clic en la etiqueta **History**.
- 02. Observe la lista generada y navegue por ella para ver el número de consultas creadas hasta la fecha.
- 03. Haga clic en la la número **5** para seleccionarla. Compruebe el código escrito por usted.
- 04. Ejecute dicha consulta de la forma habitual, pero para ver el resultado en pantalla, haga clic en la etiqueta **Query**.
- 05. Seguidamente, elija individualmente en **History**, las consultas que desee, ejecútelas y visualice los resultados como se ha comentado en el paso anterior.

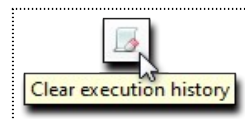
#### 4.9.2 Eliminar el historial

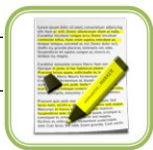
Si se desea, se puede realizar una limpieza de una, de varias y de todas las consultas que han quedado en esta zona ya conocida. Para selecciona a nivel individual, clic y suprimir; en grupo, seleccionarlas con **[Mayús]** si son consecutivas y suprimir. Pulsando **[Ctrl]** y clic para las no consecutivas y suprimir. Y para eliminarlas todas, existe un icono en la **Barra de herramientas** del Editor que se puede utilizar para este fin. Como es lógico, esta limpieza incumbe a todas las Bds de datos existentes.



#### ☐ PRACTICA 04.12.- Limpiar el historial de consultas

- 01. En el **Editor SQL**, localice y haga clic en la etiqueta **History** (por si no estuviera en ella).
- 02. Observe la lista generada de consultas creadas hasta la fecha.
- 03. Seleccione las consultas no consecutivas **1, 3, 5 y 7** de la lista. Elimínelas.
- 04. Ahora, elimine las consultas consecutivas de la **8** a la **11**.
- 05. Finalmente, elimine todas las consultas restantes, localizando y haciendo clic en el icono correspondiente. Observe el Cuadro de diálogo que aparece.
- 06. Como estamos de acuerdo en la eliminación, responda afirmativamente.
- 07. Para asegurarnos que todo ha ido bien, cierre el **Editor SQL**, vuelva a abrirlo y sitúese de nuevo en la etiqueta **History**: tendrá que estar vacío.
- 08. Abandone **SqliteStudio**.





## RESUMEN DEL TEMA 04

He aquí los 10 puntos más importantes a modo de sumario:

01	Con el comando <b>SELECT</b> se podrán seleccionar los campos de los cuales saldrá la consulta.
02	Se pueden desconectar las bases de datos que se desee y dejar una activa para realizar las consultas pertinentes.
03	Se llama <b>Script</b> a un conjunto de instrucciones almacenadas en un archivo para su ejecución.
04	La cláusula <b>FROM</b> permite indicar en qué tabla se encuentran los campos especificados en el comando <b>SELECT</b> .
05	Se podrá ejecutar una consulta pulsando la tecla de función <b>[F9]</b> .
06	Realizar comentarios en una consulta es para tenerla perfectamente "documentada" de lo que hace.
07	El código SQL de las consultas se pueden guardar externamente en archivos <b>SQL</b> y <b>TXT</b> respectivamente.
08	Se pueden exportar datos generados en una consulta en formato <b>CSV</b> y <b>PDF</b> entre otras opciones.
09	La idea de Formatear el código <b>SQL</b> es para que sea más fácil leer y corregir posibles errores cuando aparezcan.
10	El comodín asterisco (*) sustituirá automáticamente a cada uno de los nombres de campos después del <b>SELECT</b> .



## Toma Notas




## Para saber más

## Curiosidades de SQLiteStudio

- En **SQLiteStudio** se pueden escribir varias **SELECTS** y seleccionando una de ellas independientemente se puede ejecutar, dando el resultado de la **SELECT** seleccionada.
- Como se verá, **Autocompletar** es una pequeña ventana auxiliar que aparecerá cuando se pulsa la secuencia de teclas **[Ctrl]+[Espacio]**, o después de que el usuario escriba un carácter de punto y espera un momento. Esta ventana contiene una lista de valores propuestos que encajarían en la posición actual del cursor. Están ordenados en orden, de modo que los valores que el usuario más desee se colocan al principio.

## Curiosidades de SQL

- No es un lenguaje de programación al uso: aunque SQL tiene una sintaxis similar a los lenguajes de programación, en realidad es un **lenguaje de consulta** diseñado para administrar y manipular bases de datos relacionales.
- Es un lenguaje declarativo: esto significa que en SQL se describe qué resultados se desean obtener y no cómo se deben obtener. La base de datos se encarga de determinar la mejor forma de ejecutar la consulta.



## ACTIVIDADES DE REPASO

**1ra. Actividad – Abriendo Consultas en SqliteStudio, guardadas en archivos**

- 01. Abra y ejecute la consulta correspondiente al archivo **0102-01query.txt**.
- 02. Abra y ejecute la consulta correspondiente al archivo **0102-02query.txt**.
- 03. Abra y ejecute la consulta correspondiente al archivo **0102-03query.sql**.
- 04. Abra y ejecute la consulta correspondiente al archivo **0102-04query.sql**.
- 05. Abra y ejecute la consulta correspondiente al archivo **0102-05query.sql**.
- 06. Haciendo clic en cada uno de los iconos correspondientes situados en la **Barra View Toolbar**, observe la organización en pantalla de las ventanas de estos archivos.
- 07. Finalmente, cierre cada una de las ventanas existentes.

**2da Actividad – Filtrado de datos**

- 01. En la tabla **tblEmpleados** y mediante un filtro, localice todos los trabajadores de la **Delegación Norte**.
- 02. A continuación, muestre aquellas personas de nuestra empresa cuyo **Departamento** es **Seguridad**, de la **Comunidad de Murcia**.
- 03. Localice aquellas personas cuyo **Departamento** es **Compras**, cuya **Delegación** es **NorEste**, cuyo **SignoZodiacal** sea **Piscis**, cuya **Marca de coche** sea **Honda**.
- 04. ¿Cuáles son las personas que su **Grupo sanguíneo** es **O+**?
- 05. ¿Cuáles son las personas que viven en la **Provincia de Zaragoza** y en la **Ciudad de Zaragoza**?

**3ra. Actividad – Ordenaciones**

- 01. Realice una **Ordenación** en **Ascendente** por la columna **Departamento**.
- 02. Navegue por la información y observe el resultado obtenido.
- 03. Elimine dicha ordenación.
- 04. Ahora, realice una **Ordenación** por la columna **FechaNacimiento** en **Descendente**.
- 05. Navegue por la información y observe el resultado obtenido.
- 06. Elimine dicha ordenación.
- 07. Realice una **Ordenación** en **Ascendente** por la columna **Provincia** y por orden **Ascendente** del **Apellido1**.
- 08. Navegue por la información y observe el resultado obtenido.
- 09. Realice una **Ordenación** en **Descendente** por la columna **Genero** y **Orden Ascendente** por **IdEmpleado**.
- 10. Navegue por la información y observe el resultado obtenido.
- 11. Elimine dicha ordenación.
- 12. Realice una **Ordenación** en **Ascendente** por la columna **FechaContrato**.
- 13. Navegue por la información y observe el resultado obtenido.
- 14. Elimine dicha ordenación.



Atención: en las siguientes prácticas se trabajará con la Bd de **Inmobiliaria**.

#### 4ta. Actividad – Crear y Guardar una consulta

- 01. Desconecte la Bd que contiene en su interior la tabla de **tblEmpleados**.
- 02. Conecte la Bd que contiene en su interior la tabla **tblInmobiliaria2023**.
- 03. Cree una consulta con los campos: **IdNumRegistro, FechaAlta, RefInmueble, ActivoSN, TipoInmueble, Ciudad, Provincia, PrecioAlquilerMes, PrecioValorVenta**.
- 04. Ejecute la consulta y observe el resultado.
- 05. Realice la acción de **guardar** la consulta, en la carpeta **Materiales**.
- 06. Siga la ruta **01-Modulo01-BdsPlanas→AlmacenQrys**.
- 07. Dentro de ella, localice la carpeta **bdp0101qrys**.
- 08. Especifique el nombre a la consulta: se llamará **qry0101-04-01R.sql** (la “R” corresponde a práctica de **REPASO**).
- 09. Guarde los cambios y cierre el **Editor Sql**.
- 10. Localice y abra el archivo generado, fuera de **SqliteStudio**. Observe el código **SQL**.
- 11. Cierre dicho archivo.

#### 5ta. Actividad – Abrir, formatear, modificar, documentar y visualizar una consulta

- 01. Abra la consulta **qry0101-04-01R.sql**.
- 02. Aplíquelo **Formato SQL**.
- 03. Documente la consulta con la información: “**Esta consulta muestra los registros de la tabla tblInmobiliaria2023, pero solamente 10 campos**”, antes del **SELECT**, ocupando dos líneas (divídala por donde desee).
- 04. Añada el campo **Vendedor** a dicha consulta, antes del **FROM**.
- 05. Guarde las modificaciones realizadas, ejecute la consulta y observe el resultado.
- 06. Abra el archivo generado, fuera de **SqliteStudio**. Observe los cambios.
- 07. Cierre dicho archivo.

#### 6ta. Actividad – Crear, formatear y exportar datos de una consulta

- 01. Elija de la tabla **tblInmobiliaria2023**, los campos **IdNumRegistro, FechaAlta, RefInmueble, ActivoSN, TipoInmueble, Ciudad, Provincia**, para crear una nueva consulta.
- 02. Ejecútela y observe el resultado.
- 03. Guárdela en la misma zona de trabajo que la anterior, bajo el nombre **qry0101-04-02R.sql**.
- 04. Aplíquelo **Formato SQL**.
- 05. Añada los campos **PrecioValorVenta** y **PrecioAlquilerMes** a dicha consulta, antes del **FROM**.
- 06. Guarde las modificaciones realizadas, ejecute la consulta y observe el resultado.
- 07. Exporte el archivo de datos generado, en formato **CSV** en la misma zona de trabajo que las anteriores. El archivo se llamará **qry0101-04-02R.csv**.
- 08. Exporte el archivo de datos generado, en formato **HTML** en la misma zona de trabajo que las anteriores. El archivo se llamará **qry0101-04-02R.html**.
- 09. Exporte el archivo de datos generado, en formato **PDF**. en la misma zona de trabajo que las anteriores. El archivo se llamará **qry0101-04-02R.pdf**.
- 10. Abandone **SqliteStudio**.
- 11. Investigue la ubicación de los tres archivos generados en formato **CSV, HTML y PDF**.
- 12. Haga doble clic en cada uno de ellos para observar con que aplicaciones se abren.
- 13. Cierre cada uno de los archivos investigados.



## AUTOEVALUACION 04

Responda (marcando con una ☒ única) a diez preguntas relacionadas con el tema que ha estudiado para comprobar que sus conocimientos han sido realmente aprovechados.

**PUNTUACION:** se permiten 2 errores (nota mínima es un 8). Si tienen más de dos, debería repasar la lección.

<b>1</b>	<b>Se pueden desconectar las bases de datos que se desee y dejar una activa...</b>	BIEN/MAL
<input type="checkbox"/> A	Falso, no existe esta opción en SQLiteStudio	
<input type="checkbox"/> B	Verdadero, para crear tablas a la vez entre bases de datos diferentes	
<input type="checkbox"/> C	Para desproteger campos (columnas) de posibles modificaciones	
<input type="checkbox"/> D	Para realizar las consultas pertinentes	
<b>2</b>	<b>La cláusula FROM permite...</b>	BIEN/MAL
<input type="checkbox"/> A	Indicar en qué tabla se encuentran los campos especificados en el comando SELECT	
<input type="checkbox"/> B	Indicar en qué columna se encuentran los campos especificados en el comando SELECT	
<input type="checkbox"/> C	Ordenar en ascendente los datos al ejecutar una consulta	
<input type="checkbox"/> D	Mostrar los datos de todas las Bases de datos activas	
<b>3</b>	<b>Realizar comentarios en una consulta se utiliza para...</b>	BIEN/MAL
<input type="checkbox"/> A	Tenerla perfectamente "documentada" de lo que hace	
<input type="checkbox"/> B	No existe esta opción en Sql	
<input type="checkbox"/> C	Existen unos caracteres especiales como inicio y fin para esta característica	
<input type="checkbox"/> D	Las respuestas A y C son correctas	
<b>4</b>	<b>La idea de Formatear el código SQL es...</b>	BIEN/MAL
<input type="checkbox"/> A	Aligerar espacio al guardar un archivo de una consulta	
<input type="checkbox"/> B	Para facilitar la organización visual en pantalla	
<input type="checkbox"/> C	No existe esta opción en Sql	
<input type="checkbox"/> D	Ejecutar más rápido una consulta	
<b>5</b>	<b>Existen dos subcláusulas que acompañan a ORDER BY...</b>	BIEN/MAL
<input type="checkbox"/> A	La subcláusula DESC indica orden descendente	
<input type="checkbox"/> B	Y ASC indica orden ascendente (que es la opción por defecto)	
<input type="checkbox"/> C	Las anteriores respuestas son verdaderas	
<input type="checkbox"/> D	Las dos primeras respuestas son falsas	
<b>6</b>	<b>El comodín asterisco (*) sustituirá automáticamente...</b>	BIEN/MAL
<input type="checkbox"/> A	A cada uno de los datos dispuestos al ejecutar una consulta	
<input type="checkbox"/> B	No tiene ningún valor significativo en SQL	
<input type="checkbox"/> C	A cada uno de los nombres de campos después del FROM	
<input type="checkbox"/> D	A cada uno de los nombres de campos después del SELECT	
<b>7</b>	<b>Recordar en el tiempo las consultas que se han realizado se consigue...</b>	BIEN/MAL
<input type="checkbox"/> A	En la etiqueta History, que permite visualizar el código	
<input type="checkbox"/> B	También ejecutarlas con sus correspondientes resultados	
<input type="checkbox"/> C	Las respuestas anteriores son verdaderas	
<input type="checkbox"/> D	Las dos primeras respuestas son falsas	
<b>8</b>	<b>Los comentarios de un Script comienzan y terminan con la simbología...</b>	BIEN/MAL
<input type="checkbox"/> A	/? xxxxx ¿/	
<input type="checkbox"/> B	/* xxxxx */	
<input type="checkbox"/> C	/x xxxxx x/	
<input type="checkbox"/> D	Ninguna respuesta es correcta	
<b>9</b>	<b>Una consulta generada, puede exportar los datos en archivos...</b>	BIEN/MAL
<input type="checkbox"/> A	PDF, HTML, CSV	
<input type="checkbox"/> B	TXT, DBF, SQL	
<input type="checkbox"/> C	PDS, HTML, CSV	
<input type="checkbox"/> D	DOC, DOCX, XLS	
<b>10</b>	<b>Se llama Script a...</b>	BIEN/MAL
<input type="checkbox"/> A	Un conjunto de datos almacenados en un archivo	
<input type="checkbox"/> B	Un conjunto de instrucciones almacenadas en un archivo .CSV o .PDF para su ejecución	
<input type="checkbox"/> C	Un conjunto de instrucciones almacenadas en un archivo .TXT o .SQL para su ejecución	
<input type="checkbox"/> D	Un conjunto de campos almacenados en una Tabla	