



Primeros pasos en Sql

03

Sumario

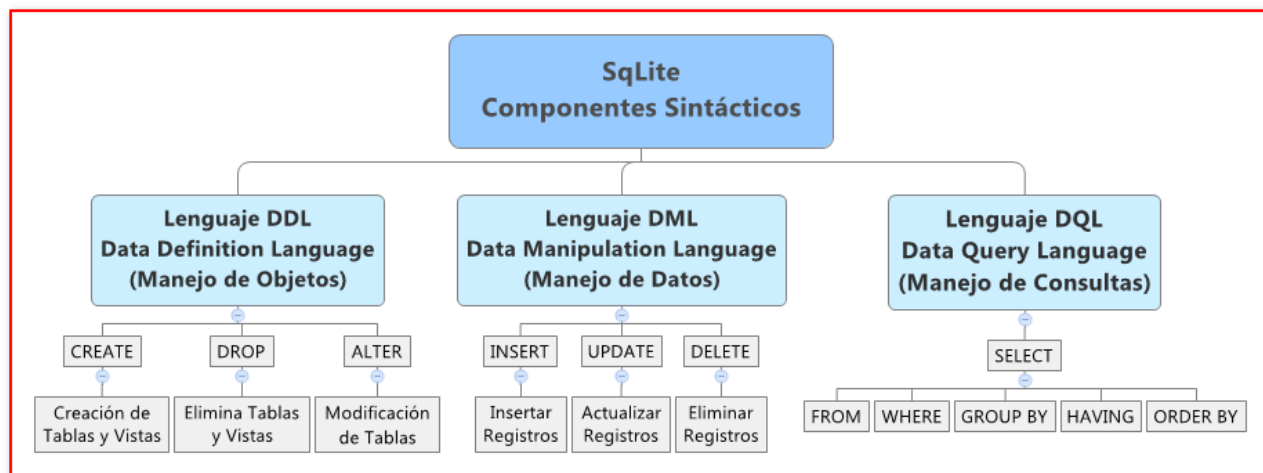
Tema 3: Primeros pasos en Sql.....	1
3.1 COMPONENTES DE SQL BAJO SQLITE.....	2
3.1.1 Breve introducción al lenguaje de Manipulación de Datos (DML).....	2
3.1.2 Breve introducción al lenguaje de Definición de datos (DDL).....	3
3.1.3 Breve introducción al lenguaje de Consulta de Datos (DQL).....	3
3.1.4 Las cláusulas.....	3
3.2 FUNCIONES DE AGREGADO.....	3
3.2.1 Consultas con predicado.....	3
3.3 COMPOSICION DQL.....	5
3.4 EL COMANDO SELECT.....	6
3.5 EL EDITOR SQL.....	6
3.5.1 Formas de abrir el Editor de consultas.....	7
3.5.2 Escribir y ejecutar consultas SQL.....	8
3.5.3 Crear y abrir varias ventanas de Editor de consultas.....	8
3.6 UN EJEMPLO SENCILLO DE SQL.....	9
3.6.1 Primeras consultas básicas.....	9
3.6.2 La cláusula AS en SQL.....	10
3.7 LA BD EMPLEADOS.....	11
3.7.1 Duplicar la Base de datos.....	11
3.7.2 La Tabla tblEmpleados.....	12
3.8 CARPETAS PARA GUARDAR CONSULTAS.....	14
3.8.1 Localización de dichas carpetas.....	14

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



3.1 COMPONENTES DE SQL BAJO SQLITE

Vistas las primeras prácticas de ordenaciones y filtrado de datos, vamos a comenzar realmente aquí con nuestro curso de **SQLite**, comentando que este **Sistema de Gestión de Bases de Datos Relacional** está compuesto por comandos, cláusulas, operadores y funciones de agregado. Estos elementos se combinan en las instrucciones para crear, actualizar y manipular las bases de datos. Veamos un resumen de los sublenguajes:



Los tipos principales de comandos se dividen en cuatro grupos:

- Ordenes del **Lenguaje de Definición de Datos (DDL, Data Definition Language)**, que permiten crear, eliminar y definir nuevas bases de datos, campos e índices.
- Ordenes del **Lenguaje de Manipulación de Datos (DML, Data Manipulation Language)**, que permiten insertar, modificar/actualizar y eliminar registros de las tablas en la Base de Datos.
- Ordenes del **Lenguaje de Consulta de Datos (DQL, Data Query Language)**, que permiten modificar y generar, así como ordenar, filtrar y extraer datos de las tablas de la Base de Datos.
- Y para terminar, el **DCL (Data Control Language)**, que se utilizaría para el Control de de Accesos y Transacciones, no es posible, en **SQLite**. La razón es que una base de datos SQLite está contenida en un solo archivo.

Todos ellos los iremos desarrollando a lo largo del curso, así como las distintas órdenes y comandos, cláusulas etc.

Con el fin de no atiborrar de información al alumnado, se prefiere ir desmenuzando poco a poco cada una de estas características.

3.1.1 Breve introducción al lenguaje de Manipulación de Datos (DML)

Con estas órdenes podrán realizarse funciones de:

- Actualización, Modificación,
- Borrado,
- Inserción.

Siendo las principales:



UPDATE	DELETE	INSERT
--------	--------	--------

3.1.2 Breve introducción al lenguaje de Definición de datos (DDL)

Es un lenguaje de programación para definir estructuras de datos. Este lenguaje es proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos así como de los procedimientos o funciones que permitan consultarlos.

Las tres órdenes o comandos son:

CREATE	DROP	ALTER
--------	------	-------

3.1.3 Breve introducción al lenguaje de Consulta de Datos (DQL)

Con SQLite, el usuario puede extraer los datos especificando qué quiere, y no cómo lo quiere. El usuario no tiene por qué conocer la disposición de los datos en el almacenamiento físico.

Con una sentencia **SQLite** se puede acceder a:

- Una o más columnas de una fila de una tabla.
- Una o más columnas de múltiples filas de una tabla.
- Una o más columnas de múltiples filas de múltiples tablas.
- Una o más columnas calculadas a partir de otras columnas.

La orden principal es la que se muestra y permite la recuperación de datos:

SELECT

3.1.4 Las cláusulas

Son condiciones de modificación utilizadas para definir los datos que desea seleccionar o manipular.

FROM	WHERE	ORDER BY	GROUP BY	HAVING
------	-------	----------	----------	--------

3.2 FUNCIONES DE AGREGADO

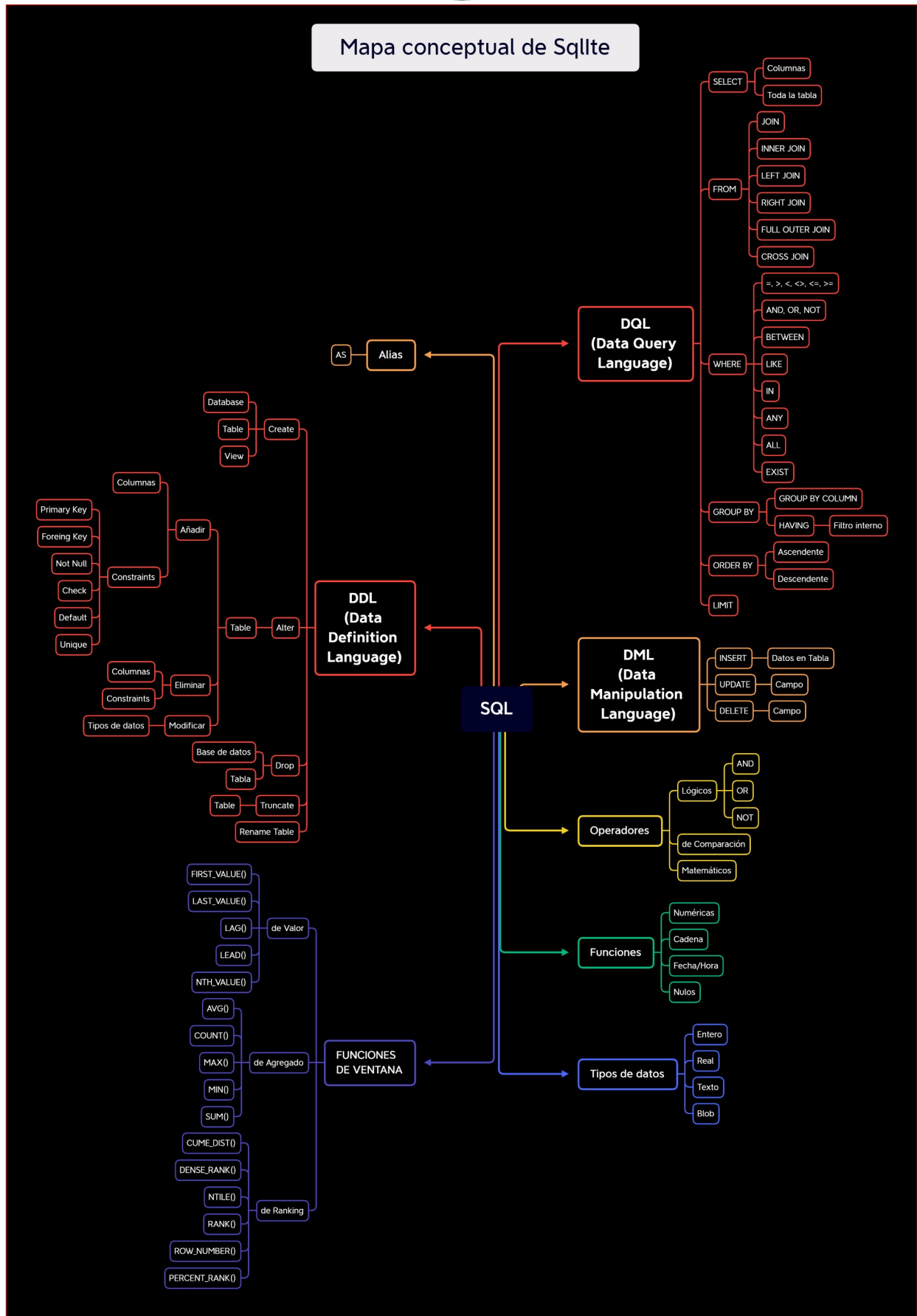
Se usan dentro de una cláusula SELECT en grupos de registros para devolver un único valor que se aplica a un grupo de registros y que son:

AVG	COUNT	SUM	MAX	MIN
-----	-------	-----	-----	-----

3.2.1 Consultas con predicado

El predicado se incluye entre la cláusula y el primer nombre del campo a recuperar, siendo los posibles predicados:

ALL	LIMIT	DISTINCT
-----	-------	----------





3.3 COMPOSICION DQL

Comenzamos por la que más se va a utilizar. El lenguaje **SQL** se compone de una serie de sentencias y cláusulas muy reducidas en número, pero potenciales en recuperación de información. A continuación, se muestra una de ellas como ejemplo, estando formada por **una sentencia** y **tres cláusulas**:

SELECT ListaDeCampos
FROM ListaDeTablas
[WHERE criterios
[ORDER BY ListaDeCampos]]



A lo largo de este módulo, tendremos como referencia un proyecto de Base de datos plana, sobre películas de cine ficticias, con la finalidad de ir mostrando ejemplos. Contendrá únicamente una tabla, llamada **tblPelículas**. **USTED NO TENDRÁ QUE HACER NADA**, únicamente ver los ejemplos como muestra y fijarse en las explicaciones. Observe parte del dataset, en la imagen siguiente:

IdPelícula	TituloPelícula	DuracionMinutos	ClasificacionEdad	Genero	DiasProyeccion	Sesion	HoraProyeccion	FechaEstreno	FechaFinFase	CineProyeccion	SalaProyeccion	Espectadores	Puntuacion
1	1 Buenos y regulares	115	Familia	Aventuras	Lun/Mie/Vie	Tarde	16:45	2023-01-10	2023-02-18	La Plaza	Sala D	13345	7.02
2	2 Buenas noches, mi amor	110	18 Años	Terror	Sab	Noche	11:45	2023-01-11	2023-02-16	San Luis	Sean Connery	7333	5
3	3 Risas a toda prisa	95	18 Años	Comedia	Sab/Dom	Noche	22:30	2023-01-12	2023-02-26	La Plaza	Sala B	12078	6.74
4	4 Dama del sur	101	18 Años	Accion	Jue/Vie	Noche	22:45	2023-01-15	2023-02-17	Puerta Norte	Oberan	10267	6.5
5	5 Caballitos en el desfile	118	Nenes	Animacion	Dom	Tarde	19:00	2023-01-18	2023-02-22	Puerta Sur	Lauren Bacall	1650	2
6	6 Las aventuras de Juanin	95	Familia	Animacion	Sab	Matinal	12:00	2023-01-20	2023-03-05	Santiago Apostol	Este	15227	7.95
7	7 Pookoo en el Amazonas	NULL	Todos los publicos	Animacion	Dom	Matinal	11:30	2023-01-22	2023-03-02	Plaza Grande	Sala D	8549	5.87
8	8 Transformados III	96	Familia	Aventuras	Sab	Matinal	12:00	2023-01-23	2023-03-04	San Luis	Robert De Niro	5646	4.69
9	9 Ahí te quiero ver	101	Todos los publicos	Comedia	Sab	Matinal	12:00	2023-01-23	2023-03-06	San Luis	Paul Newman	4163	3.89
10	10 Enviados al ayer	119	12 Años	Ficcion	Dom	Tarde	17:45	2023-01-24	2023-03-08	Molinos	Tom Cruise	18073	9.1
11	11 Romeo en el mar	95	Todos los publicos	Ficcion	Dom	Tarde	19:00	2023-01-24	2023-03-06	Plaza Grande	Sala E	6066	4.97
12	12 Social data	95	Mayores de 12	Accion	Sab/Dom	Matinal	12:00	2023-01-24	2023-03-03	Plaza Grande	Sala F	5320	4.01
13	13 Lapices parlantes	86	12 Años	Accion	Sab	Matinal	11:30	2023-01-25	2023-02-26	Puerta Norte	Futura	8819	5.92
14	14 El lago de las espinas	NULL	Nenes	Animacion	Sab/Dom	Matinal	12:00	2023-01-25	2023-02-25	Monterrey	Londres	5647	4.69
15	15 Anoecer en la isla	99	18 Años	Belica	Mar/Jue	Tarde	18:45	2023-01-29	2023-03-01	Puerta Sur	Tom Cruise	7859	5.02
16	16 Transformados II	93	Todos los publicos	Aventuras	Sab/Dom	Matinal	11:00	2023-01-30	2023-03-04	Monterrey	Lisboa	9441	6.12
17	17 Increiblemente divertidos	103	Nenes	Aventuras	Lun/Mie/Vie	Tarde	17:00	2023-02-01	2023-03-18	Argentina	Nicolas Cage	7481	5
18	18 Resguardarse de la lluvia	90	Familia	Accion	Dom	Tarde	18:30	2023-02-06	2023-03-23	Santiago Apostol	Sur	13521	7.08
19	19 El diablo está en la cuna	102	18 Años	Terror	Lun/Mie	Tarde	19:00	2023-02-10	2023-03-26	Monterrey	Mexico	3567	2.91
20	20 Compañeros de fatigas	96	Todos los publicos	Comedia	Dom	Matinal	12:00	2023-02-11	2023-03-19	Las Americas	Grace Kelly	8878	5.95
21	21 Revoluciones silenciosas	107	18 Años	Drama	Lun/Mie/Vie	Tarde	18:45	2023-02-16	2023-03-19	Monterrey	Buenos Aires	4726	4.01
22	22 Consejos familiares	99	Nenes	Animacion	Dom	Tarde	18:00	2023-02-16	2023-03-22	Plaza Grande	Sala H	2361	2.47
23	23 A santo de que	94	18 Años	Comedia	Dom	Tarde	16:00	2023-02-17	2023-04-01	San Luis	Charles Chaplin	3589	3.72
24	24 El cielo espera un angel	98	12 Años	Accion	Mar/Jue	Tarde	18:45	2023-02-18	2023-03-21	Puerta Norte	Novax	1110	1.99
25	25 Funky "El musical"	92	Mayores de 12	Ficcion	Dom	Tarde	17:45	2023-02-19	2023-04-01	Plaza Grande	Sala A	13930	7.16

Observe un ejemplo del lenguaje, para entrar en materia, sobre una tabla de Películas:

Mediante la sentencia...

```
1 SELECT TituloPelícula, DuracionMinutos, Genero, Sesion, FechaEstreno
2 FROM tblPelículas
3 WHERE Genero = "Aventuras" AND Sesion = "Tarde"
4 ORDER BY TituloPelícula
```

...se ha
conseguido
el resultado:

	TituloPelícula	DuracionMinutos	Genero	Sesion	FechaEstreno
1	Buenos y regulares	115	Aventuras	Tarde	2023-01-10
2	Campos dorados	96	Aventuras	Tarde	2023-04-01
3	Increiblemente divertidos	103	Aventuras	Tarde	2023-02-01
4	Sombras del presente	103	Aventuras	Tarde	2023-02-21
5	Tomtoon se va del cole	96	Aventuras	Tarde	2023-03-02



La “traducción” sería:

Se **seleccionarán** los campos **TituloPelicula**, **DuracionMinutos**, **Genero**, **Sesion**, **FechaEstreno**
De la tabla **tblPeliculas**

Donde el **Género** a localizar es **Aventuras** y la **Sesión** sea por la **Tarde**

Ordenado por **TituloPelicula**, en Ascendente.

Como se aprecia, la sintaxis de SQL es muy similar al lenguaje hablado y sus aspectos principales son fácilmente asimilables.

3.4 EL COMANDO SELECT

Este comando **DQL**, se utilizará para seleccionar los campos de los cuales saldrá la consulta, es decir, que establece los campos que se visualizarán o compondrán la consulta. El parámetro **ListaDeCampos** estará compuesto por uno o más nombres de campos, separados por comas, pudiéndose especificar también el nombre de la tabla a la cual pertenecen, seguidos de un punto y del nombre del campo correspondiente.

Si el nombre del campo o de la tabla está compuesto de más de una palabra, este nombre tendrá de escribirse entre **corchetes** como por ejemplo **[Nombre del Cliente]**.

3.5 EL EDITOR SQL

La ventana del **Editor SQL** es aquella que se utiliza para escribir en código **SQL**, una consulta. En la parte superior de esta ventana del **Editor SQL** se encuentra su propia **Barra de herramientas**.

Iconos en la Barra de herramientas del Editor SQL



1	Cuadro combinado para elegir la base de datos de trabajo actual para la ventana del Editor.
2	Ejecuta la consulta escrita, mostrando el resultado en la zona correspondiente.
3	Ejecuta la instrucción EXPLAIN para la consulta actual.
4	Formatea y muestra en Sql mejorado, las consultas escritas.
5	Elimina el historial de ejecución de consultas.
6	Para exportar los resultados de la consulta escrita.
7	Para imprimir el código SQL de la consulta escrita.
8	Crea una vista en código SQL a partir de la consulta ejecutada.
9	Guarda el código SQL escrito de la consulta, en un archivo ,SQL o .TXT
10	Para cargar el contenido SQL del archivo seleccionado en el editor.
11-12	Para Buscar y Reemplazar información
13-14	Configura el modo de presentación de resultados del Editor SQL.

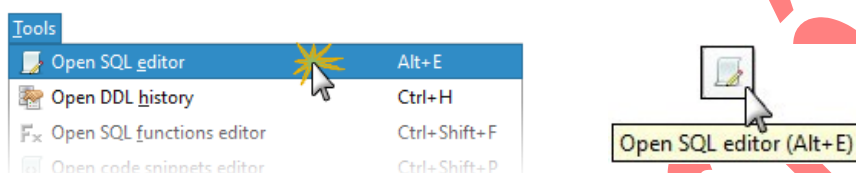


Existe una lista desplegable en esta **Barra de herramientas** que contiene una serie de bases de datos que están actualmente en curso, si se cargaron debidamente en **SqLiteStudio**. La base de datos seleccionada en esa lista se utiliza para ejecutar una consulta escrita. Puede cambiar entre las bases de datos existentes desplegando y seleccionando de la lista, pensando además que dicha lista se actualiza cada vez que se abre o cierra una base de datos.

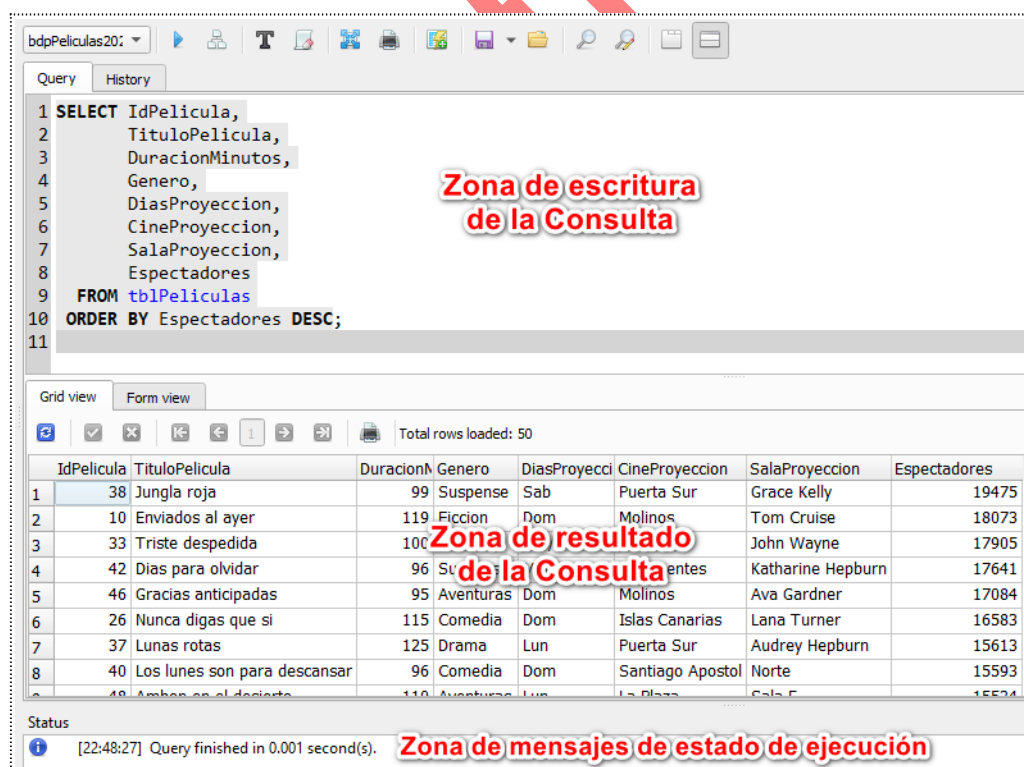
3.5.1 Formas de abrir el Editor de consultas

Para la apertura del Editor, puede:

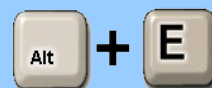
- Localizar la orden en el **Menu Tools→Open SQL Editor**
- o bien haciendo clic en el icono correspondiente situado en la **Barra de herramientas** principal de la aplicación



El **Editor SQL** tiene una zona de edición de texto en la parte superior de la ventana para escribir consultas **SQL** y una vista inferior para mostrar los resultados de la misma, así como la zona de mensajes para el estado de ejecución:



Pulsando la secuencia de teclas indicada, puede abrir una ventana vacía de Editor SQL rápidamente en pantalla. Si sigue realizando esta operación varias veces, tendrá a la vista varias ventanas vacías abiertas del editor.





Si prefiere tener el campo de edición de consultas en una página y los resultados en otra página, puede cambiarlo muy rápidamente desde la **Barra de herramientas**, utilizando los iconos correspondientes.

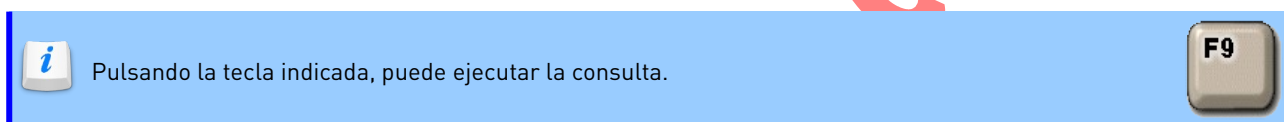
3.5.2 Escribir y ejecutar consultas SQL

El **Editor SQL** se utiliza para crear consultas, pero también será útil para la creación de **Vistas** (Views) y **Activadores** (Triggers), como se verá más adelante.

El editor SQL proporciona las siguientes características:

- Resaltado de sintaxis: funciona utilizando colores y fuentes definidos en la ventana de Configuración.
- Resalta los nombres de las tablas correctamente.
- Mostrar errores de escritura sobre la marcha.
- Autocompletado de sintaxis.
- Formateado de código de lectura fácil.

Una vez escrita una consulta en el Editor, para ejecutarla se localizará el icono correspondiente

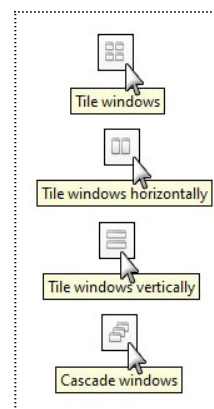
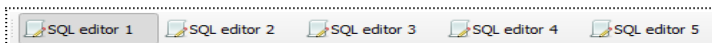


3.5.3 Crear y abrir varias ventanas de Editor de consultas

Una de las características más notables de **SQLiteStudio** es la posibilidad de trabajar con varias ventanas abiertas. Esto es muy interesante, puesto que gracias a esta posibilidad se pueden tener múltiples consultas a una tabla o varias, ejecutándose individualmente. Veamos un adelanto a esta forma de trabajo:

☐ PRACTICA 03.01.- Apertura multiventanas

- 01. En la BD actual **bdp0101-Inmobiliaria-TRABAJO.db** haga que la tabla **tblInmobiliaria2023** esté seleccionada en el panel izquierdo.
- 02. Abra el **Editor de SQL** al menos cinco veces usando las técnicas descritas para dicho fin.
- 03. Observe como en la zona inferior aparecen también en modo de pestañas numeradas, la representación gráfica de sus ventanas.
- 04. En la **Barra de herramientas** de la aplicación, localice los cuatro iconos situados lo más a la derecha en dicha barra: se utilizarán para organizar dichas ventanas en la zona central de la aplicación.
- 05. Haga clic libremente en cada uno de ellos y observe lo que acontece en pantalla.
- 06. Como en cualquier aplicación, **minimice**, **maximice** y **restaure** cada una de las cinco ventanas para familiarizarse a estas vistas.
- 07. Haga clic en cada uno de los titulares individuales que encabezan estas ventanas con el fin de que queden activas (solamente una de ellas será la activa, identificándose por su color azul).





- 08. Ahora haga clic en su interior, en la línea gris que acompaña a la cifra 1.
- 09. Localice y haga clic en las etiquetas nombradas como **SQL Editor 1**, **SQL Editor 2**, etc. en la zona inferior de la aplicación.
- 10. Observe como se identifican estos nombres en las ventanas correspondientes.
- 11. Finalmente, cierre cada una de las ventanas: la zona central quedará en color gris.

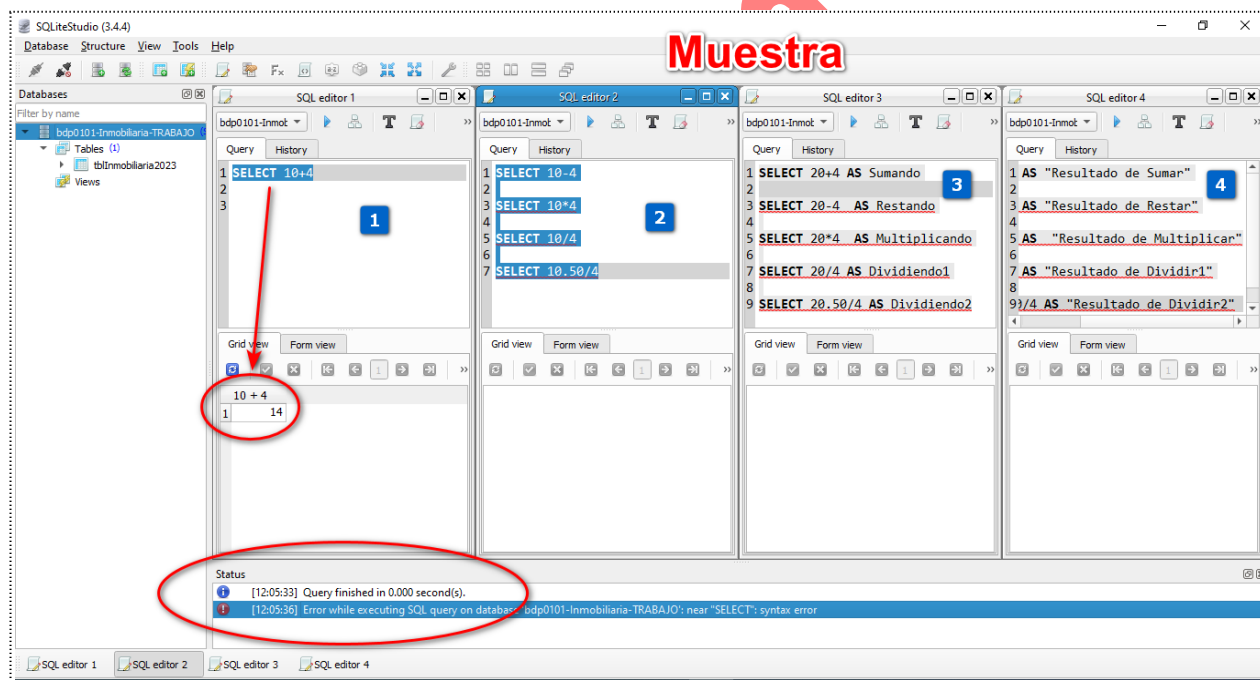
3.6 UN EJEMPLO SENCILLO DE SQL

La declaración **SELECT** es la que se ejecuta con más frecuencia; puede consultar tablas de una Base de datos y recuperar un conjunto de filas o incluso puede ejecutarse de la forma más sencilla con simples operaciones matemáticas, por ejemplo.

Cuando escriba una consulta **SQL** en el editor, resaltará la sintaxis **SQL**, pero también si escribe el nombre correcto de cualquier objeto (es decir, una tabla, índice, activador, vista, base de datos o nombre de columna existente), el nombre se resalta en azul color por defecto.

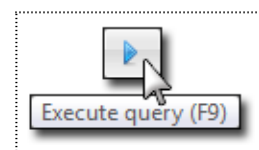
3.6.1 Primeras consultas básicas

En la ventana del **Editor SQL**, puede seleccionar (con el ratón o manteniendo presionada la tecla **[Mayús]** y usando las teclas de flecha) solo una parte del código escrito en la consulta **SQL** y hacer clic en el icono correspondiente para ejecutar solo la parte seleccionada del código.



☐ PRACTICA 03.02.- Mi primer SELECT

- 01. Cree cuatro ventanas de **Editor SQL**.
- 02. Organícelas en pantalla para que tengan el aspecto de la muestra.
- 03. Tal y como se representa la muestra, en la ventana 1, escriba, ejecute la consulta y observe su resultado en la parte inferior, en la pestaña **Grid view**.





- 04. Para la ventana **2**, escriba las cuatro líneas de código y observe como la consulta da un error al ejecutarla: no se pueden ejecutar en grupo las consultas.
- 05. En esta ventana, seleccione únicamente la primera línea que contiene **SELECT 10-4** y ahora ejecute nuevamente la consulta: funciona perfectamente.
- 06. Seleccione únicamente la segunda línea que contiene el código **SELECT 10*4** y ejecute la consulta para ver como individualmente funciona perfectamente.
- 07. Realice la selección de la tercera línea, la que contiene la división del **SELECT 10/4** y ejecútela para ver el resultado.
- 08. Haga lo propio con la última división de esta ventana.

3.6.2 La cláusula AS en SQL

Se utiliza en **SQL** para cambiar temporalmente el nombre a una o (unas columnas), así como en tablas en la ejecución de una consulta. Esta cláusula ayuda a mejorar la legibilidad del código y de su resultado, llamada también como "**alias**", existe solo mientras dura la consulta (de ahí su temporalidad).



Hay que tener en cuenta que el texto que sigue a la cláusula **AS**, si es una simple palabra, no hace falta que vaya "entrecomillada", pero si es un texto que contiene varias palabras, éstas deberán obligatoriamente ir "entre comillas simples" (o entre corchetes). Por ejemplo **Matemáticas** o '**Matemáticas aplicadas**' o [Matemáticas aplicadas].

En resumen, facilita la presentación de los resultados de la consulta y permite al desarrollador etiquetar los resultados con mayor precisión sin cambiar permanentemente el nombre de las columnas de la tabla o incluso la tabla misma.



PRACTICA 03.03.- Ejemplos con la cláusula AS

- 01. Para la ventana **3**, escriba las cinco líneas de código y ejecute la consulta globalmente: observe como da un error, ya que no se pueden ejecutar en grupo las consultas.
- 02. Ahora, seleccione únicamente la primera línea que contiene **SELECT 20+4 AS Sumando** y ejecute nuevamente la consulta: funciona perfectamente.
- 03. Observe el titular en la pestaña **Grid view**.
- 04. Seleccione la segunda línea con el código **SELECT 20-4 AS Restando** y ejecute la consulta para ver como individualmente funciona perfectamente.
- 05. Observe el titular en la pestaña **Grid view**.
- 06. Haga lo propio con las últimas líneas (siempre a nivel individual) para que el resultado sea el idóneo.
- 07. Para la ventana **4** escriba las cinco líneas de código.
- 08. Seleccione únicamente la primera línea que contiene **SELECT 30+4 AS "Resultado de Sumar"** y ejecute la consulta.
- 09. Observe el titular en la pestaña **Grid view**.
- 10. Ejecute las consultas restantes una a una, observando el titular de las mismas en la pestaña **Grid view**.
- 11. Finalmente, cierre cada una de las ventanas: la zona central quedará en color gris.
- 12. Abandone la aplicación **SqliteStudio**.

Sumando	
1	24
Restando	
1	16
Multiplicando	
1	80
Dividiendo1	
1	5

Resultado de Sumar	
1	34
Resultado de Restar	
1	26
Resultado de Multiplicar	
1	120
Resultado de Dividir1	
1	7
Resultado de Dividir2	
1	7.625



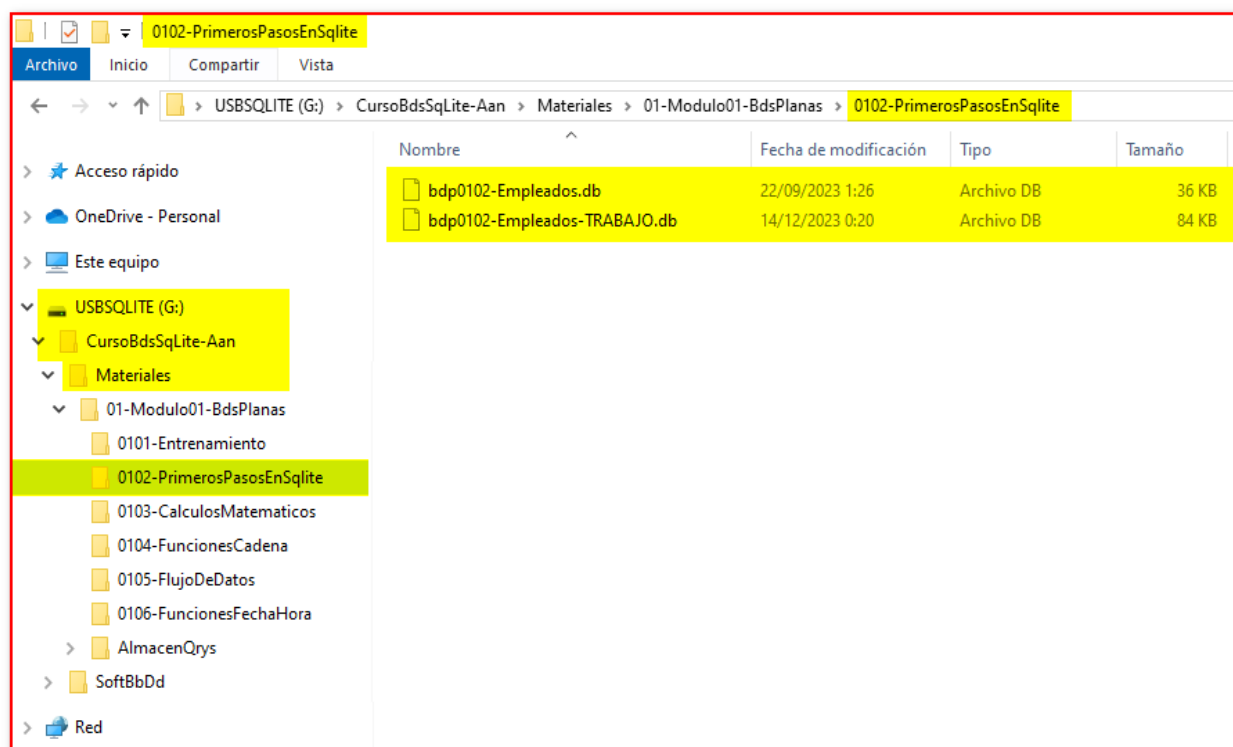
3.7 LA BD EMPLEADOS

La base de datos con la cual vamos a trabajar, se llama **bdp0102-Empleados.db**, que está situada en la carpeta **0102-PrimerosPasosEnSQL**. Solo contiene una tabla en su interior, llamada **tblEmpleados**, la cual contiene **100** registros y contempla la información de los trabajadores de una empresa ficticia.



3.7.1 Duplicar la Base de datos

Como hemos comentado en el tema anterior, siempre se deberá tener en cuenta, es que antes de trabajar con una Base de datos, **se deberá realizar una copia de la original**, con el fin de preservarla ante posibles catástrofes. De esta manera, siempre tendremos un respaldo ante una situación inesperada.



☐ PRACTICA 03.04.- Duplicar una base de datos por seguridad

- 01. Abra el **Explorador de Windows** y cerciórese de que tiene insertada su unidad USB de trabajo.
- 02. Localice en el interior de la carpeta **0102-PrimerosPasosEnSQL**, la única base de datos llamada **bdp0102-Empleados.db**.
- 03. Tendrá que realizar un duplicado de la misma y cambiar su nombre por el nuevo llamado **bdp0102-Empleados-TRABAJO.db**.
- 04. Obtenida la copia, **cierre el Explorador de Windows**.

☐ PRACTICA 03.05.- Primer contacto con la Bd Empleados

- 01. Abra **SQLiteStudio** haciendo doble clic en el icono del acceso directo situado en el **Escritorio**.



- 02. Haciendo clic en el icono correspondiente, aparecerá el Cuadro de diálogo **Database** en donde en primer lugar deberemos seleccionar el archivo de Bd que se ha facilitado y que nos está esperando en la carpeta conseguida anteriormente.
- 03. Seleccione el archivo de trabajo **bdp0102-Empleados-TRABAJO.db**, el cual será idóneo para seguir aprendiendo nuevos conocimientos en **Sql**. Este archivo comienza con los caracteres **bdp**, puesto que en su interior habrá una sola tabla como base de datos plana (de ahí sus iniciales).
- 04. Se hará clic en el botón para **testear** la conexión a nuestra bd.
- 05. Haciendo clic en el botón **OK**, se observará como se ha cargado perfectamente el archivo de la BD, estando posicionado su nombre en la **zona izquierda** de la aplicación.
- 06. Si ahora posiciona el puntero del ratón en el nombre de la BD, observará un panel indicativo de la ubicación, así como la versión y peso de la misma.
- 07. Y si hace doble clic en el nombre de la BD, observará dos únicos objetos como **Tables** y **Views**.
- 08. Haga doble clic en la palabra **Tables**: aparece un árbol de ramificaciones con el fin de ver todavía más los objetos, como son las **Columnas**, **Indexes** y **Triggers** de la única tabla existente llamada **tblEmpleados**.
- 09. Haga doble clic en el nombre de la tabla **tblEmpleados** situada en la zona izquierda: aparecen las vistas en la zona centra, como **Structure**, **Data**, **Constraints**, etc.
- 10. Ahora se verá como algunos iconos de la **Barra de Herramientas** principal están activos.
- 11. En la zona izquierda, haciendo clic sucesivamente en los triángulos correspondientes, verá como se expande/oculta la ramificación llegando a mostrar algunas de las columnas existentes (es decir los nombres de los campos de la tabla **tblEmpleados**) y mediante la **Barra de desplazamiento vertical**, podremos ver las restantes.
- 12. Posicionando el puntero del ratón en el nombre de la tabla existente, aparecerá en un panel indicativo, los nombres de las columnas así como el número de ellas.
- 13. En la zona izquierda de la aplicación, intente visualizar todos los nombres de las columnas de la tabla.
- 14. Abandone la aplicación **SqliteStudio**, haciendo clic en el botón superior de cierre, como en cualquier aplicación Windows.
- 15. Repita la apertura de la aplicación y observe como la última BD incorporada ya le está esperando, por defecto.
- 16. Haga que se muestre la estructura de la tabla **tblEmpleados**.
- 17. Vaya cambiando entre las etiquetas **Structure** y **Data** para las explicaciones siguientes.

3.7.2 La Tabla **tblEmpleados**

Dicha tabla contiene **35** columnas (campos) y cada campo de la tabla tiene en su interior datos. Para un mejor conocimiento de ellos se comentan uno a uno:

- El campo **CodEmpleado** sirve para identificar a cada persona que trabaja en nuestra empresa mediante un código único e irrepitable (en esta tabla, esta columna es, de tipo **Texto**).
- El campo **IdEmpleado** es un campo **N Numérico (Entero)**. Su función es controlar el número de empleados.
- Los campos **Apellidos** y **Nombre** son obvios. Campos de tipo **Texto**.
- En el campo de texto **Email**, se guardarán los datos del correo electrónico de cada persona.
- En el campo **Dni** estará el dato de este documento para cada empleado. Es un campo de tipo **Texto**.



- En el campo **Género** (que es **Texto**) aparecerá un solo carácter (F, para **Femenino** y M para **Masculino**).
- En el campo **FechaNacimiento** (de tipo **Texto**) guardaremos información de cada persona, en cuanto al **Año, Mes y Día** de Nacimiento.
- Cada empleado tiene un **SignoZodiacal**, que será **Texto**.
- En el campo **FechaContrato** se especificará la fecha de cuando nuestros empleados han empezado a trabajar en ella. Campo de tipo **Texto**.
- Los campos **Domicilio, CodPostal, Ciudad, Provincia y Comunidad** son obvios (todos en formato **Texto**).
- Nuestra empresa tiene **Delegaciones** por toda **España** (hay 7 delegaciones). En este campo (de formato **Texto**) se guardará la información de cada trabajador con respecto a su delegación.
- EN las Delegaciones, existen **Jefes de delegación**. Estos datos están en formato **Texto**.
- El campo **Profesion** (de formato **Texto**), es obvio: la profesión de cada persona. Pueden existir registros que no se hayan cumplimentado todavía.
- **AñosExperiencia** es un campo Numérico (Integer), en donde se guardarán los años que nuestros trabajadores tienen como experiencia en su profesión.
- Cada empleado trabaja en un **Departamento** de la empresa (hay 16). De formato **Texto**, se guardará la información de los nombres de dichos departamentos.
- En los campos **Estatura** y **Peso**, se guardará la información personal correspondiente. Son campos de tipo **Numérico**. Pueden existir personas que no hayan comunicado estos datos.
- En el campo **EstadoCivil**, de tipo **Texto**, (y que hay 4 formas) se guardará dicha información. Pueden existir personas que no tengan estos datos.
- Cada persona tiene un **TipoSanguineo** (hay 8 tipos). Campo de tipo **Texto**.
- Generalmente los empleados de nuestra empresa tienen un **TelefonoMovil** (campo de tipo **Numérico**). Puede haber personas que todavía no hayan comunicado este dato.
- En el campo **NombreTarjeta** (de tipo **texto**) se guardará el nombre de cualquiera de las 4 existentes. Puede haber personas que todavía no hayan comunicado este dato.
- Los campos **TieneCoche, MarcaCoche** y **YearCompraCoche** son aquellos en donde pondremos información al respecto. El primero será de tipo **Texto** y tendrá dos únicas posibles respuestas (S/N); el segundo de tipo **Texto** y el tercero **Numérico**. Para los dos últimos, podría darse el caso de que no se tuvieran datos todavía.
- En el campo **SueldoMes** (de tipo **Numérico**) se guardarán los datos de lo que cobra cada empleado (podría darse el caso de que algunos empleados no tuvieran este dato todavía).
- En los campos **TieneHijos** (de tipo **Texto**, con las posibilidades únicas de S/N) y **NumHijos** (de tipo **Numérico**), pueden existir algunos trabajadores que no hayan comunicado estos datos.
- El campo **BonusAnualHijos** corresponde a una ayuda por hijos del empleado/a.
- **PuntualidadAnual** es una cantidad económica que incentiva llegar a la hora a la empresa.
- En el campo **CocheEmpresa** se especificará si una persona conduce un automóvil para fines laborales.



PRACTICA 03.06.- Visualizar la Estructura de la tabla tblEmpleados

- 01. En la etiqueta **Structure**, navegue por esta vista y observe detenidamente la información existente.
- 02. Identifique los distintos nombres de campos (es decir las columnas) y los tipos de datos individuales.
- 03. Si ahora hace un doble clic en el nombre de un campo, aparecerá el Cuadro de diálogo **Column** con más características, pero únicamente para el campo seleccionado.

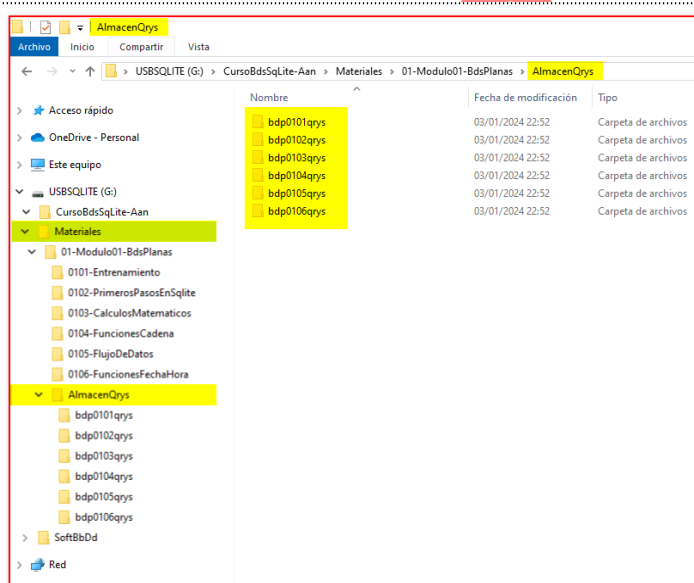


- 04. Cierre ese Cuadro de diálogo abierto.
- 05. Averigüe la información de otros campos restantes.
- 06. Compruebe como en esta etiqueta **Structure** existen diversos iconos para trabajar con la edición, modificación y puesta a punto de los campos de la tabla.
- 07. Tal y como se apreciará, de momento, no existe ningún campo clave (**Primary Key**) y no hay ninguna columna que organice un orden en la tabla.
- 08. Situándose nuevamente en la etiqueta **Data**, observe los datos de la tabla: de esta manera queda muy clara la diferencia entre la estructura de la tabla y los datos de la misma.
- 09. Finalmente, cierre esta ventana y abandone la aplicación.

3.8 CARPETAS PARA GUARDAR CONSULTAS

De ahora en adelante y a partir del **Tema04**, comenzaremos a guardar respectivamente en archivos **.SQL** y **.TXT** el código de las consultas que realicemos. Incluso cuando al final de los temas se procedan a realizar repases, nos referiremos a las carpetas que se muestran en la figura de la izquierda. Así, se trabajará con distintos **Datasets** para entender como en la variedad está el conocimiento.

Como se ha comentado, estas bases de datos son planas para este primer módulos del curso y siempre tienen una sola tabla en su interior. Esto se ha pensado para que de esta manera se concentre el estudio en un único objeto de datos.

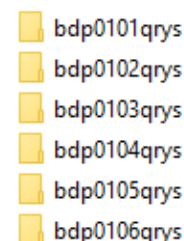


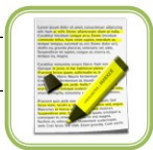
3.8.1 Localización de dichas carpetas

Se llama **Sentencia** a un conjunto de instrucciones almacenadas en un archivo para su ejecución. Y en ellas se diseñaran instrucciones en lenguaje **SQL** bajo **SqliteStudio**. Pues bien, con el fin de guardar esta información, se facilitaron una serie de ubicaciones cuando se entregaron los materiales para este primer módulo.

☐ PRACTICA 03.07.- Carpetas para guardar Consultas

- 01. Abra el **Explorador de Windows**.
- 02. Dentro de la carpeta principal **CursoBdsSqlite-Aan**, localice la carpeta **Materiales**.
- 03. Sitúese en la carpeta **01-Modulo01-BdsPlanas**.
- 04. Ahora investigue las carpetas que hay en el interior de la carpeta **AlmacenQrys**.
- 05. Observe que en el interior de cada una de ellas no hay ninguna información: se guardarán los archivos que se crearán en las consultas a realizar en los temas correspondientes.
- 06. Finalmente, cierre el **Explorador de Windows**.





RESUMEN DEL TEMA 03

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

01	Con el Lenguaje de Definición de Datos (DDL), se pueden crear Tablas.
02	El llamado DQL (Lenguaje de Consulta de Datos) permite extraer datos de las tablas.
03	Las siglas DML corresponden al Lenguaje de Manipulación de Datos.
04	Órdenes como UPDATE , DELETE e INSERT pertenecen al lenguaje DML .
05	CREATE , DROP y ALTER son órdenes que pertenecen al lenguaje DDL .
06	Con el comando SELECT se podrán seleccionar los campos de los cuales saldrá la consulta.
07	Se llama Sentencia a un conjunto de instrucciones almacenadas en una consulta para su ejecución.
08	El código de una consulta se puede guardar en archivos con formato .TXT y .SQL respectivamente.
09	El Editor SQL es una ventana en donde se pueden escribir sentencias y al ejecutar se podrán ver los resultados.
10	AS es una cláusula que se utiliza para cambiar temporalmente el nombre a una o unas columnas.



Toma Notas



Para saber más

Ventajas de utilizar el modelo relacional

- El modelo de base de datos relacional en DBMS es el modelo más utilizado para gestionar bases de datos. Ofrece una serie de ventajas sobre otros modelos, incluida precisión, facilidad de uso, colaboración, seguridad y categorización.
- Quizás la ventaja más importante del modelo relacional es su precisión. Debido a que los datos están organizados en tablas, es fácil realizar un seguimiento de la información y asegurarse de que esté actualizada. Esta precisión es esencial para las empresas que dependen de la información de la base de datos para tomar decisiones.
- El modelo relacional también es fácil de usar. Los usuarios pueden acceder fácilmente a los datos y ejecutar consultas sin necesidad de comprender conceptos complejos. Esta facilidad de uso hace que el modelo relacional sea ideal para empresas que necesitan recuperar rápidamente información de sus bases de datos.

Averiguar la versión de trabajo de SQLite

- Escribiendo en el **Editor Sql** el script **SELECT SQLITE_VERSION();** se consigue el número de versión en la aplicación **SQLite**. Para nuestro caso, estamos trabajando con la número **3.41.2**.



AUTOEVALUACION 03

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.

1	Los comandos INSERT, UPDATE y DELETE corresponden al lenguaje...	BIEN/MAL
<input type="checkbox"/> A	DLQ	
<input type="checkbox"/> B	DDL	
<input type="checkbox"/> C	DML	
<input type="checkbox"/> D	DQL	
2	La instrucción SELECT...	BIEN/MAL
<input type="checkbox"/> A	Generalmente va acompañada de la orden ORDER BY	
<input type="checkbox"/> B	Generalmente va acompañada de la orden FROM	
<input type="checkbox"/> C	Generalmente va acompañada de la orden WHERE	
<input type="checkbox"/> D	Ninguna respuesta de las anteriores es correcta	
3	DDL posee instrucciones como...	BIEN/MAL
<input type="checkbox"/> A	CREATE, DROP, ALTER, UPDATE	
<input type="checkbox"/> B	CREATE, DROP, ALTER	
<input type="checkbox"/> C	FROM, WHERE, HAVING	
<input type="checkbox"/> D	Todas las respuestas son correctas	
4	El código de una consulta se puede guardar en archivos con formato...	BIEN/MAL
<input type="checkbox"/> A	.DOC y .XLT	
<input type="checkbox"/> B	.PSD, .MP3 y .PNG	
<input type="checkbox"/> C	.TXT y .SQL	
<input type="checkbox"/> D	Ninguna respuesta es correcta	
5	La instrucción SELECT...	BIEN/MAL
<input type="checkbox"/> A	Se utilizará para seleccionar los campos de los cuales saldrá la consulta	
<input type="checkbox"/> B	Se utilizará para seleccionar los datos de los cuales saldrá la consulta	
<input type="checkbox"/> C	No tiene ninguna utilidad en el lenguaje SQL	
<input type="checkbox"/> D	Va unida generalmente a la instrucción UPDATE	
6	Se puede abrir el editor SQL, pulsando la secuencia de teclas...	BIEN/MAL
<input type="checkbox"/> A	[Ctrl]+[E]	
<input type="checkbox"/> B	[Mayúsc]+[E]	
<input type="checkbox"/> C	[Ctrl]+[Alt]+[E]	
<input type="checkbox"/> D	[Alt]+[E]	
7	Una consulta se puede ejecutar pulsando la o las teclas...	BIEN/MAL
<input type="checkbox"/> A	[F9]	
<input type="checkbox"/> B	[Ctrl]+[F9]	
<input type="checkbox"/> C	[Alt]+[F9]	
<input type="checkbox"/> D	Ninguna de las anteriores	
8	AS es una cláusula...	BIEN/MAL
<input type="checkbox"/> A	Que no existe en el lenguaje SQL	
<input type="checkbox"/> B	Para indicar que una persona sobresale sobre el resto	
<input type="checkbox"/> C	Que se utiliza para cambiar temporalmente el nombre a una o unas columnas	
<input type="checkbox"/> D	Ninguna respuesta es correcta	
9	Se pueden crear y abrir varias ventanas en el Editor SQL...	BIEN/MAL
<input type="checkbox"/> A	No	
<input type="checkbox"/> B	Si	
<input type="checkbox"/> C	Si, pero con limitaciones de memoria	
<input type="checkbox"/> D	Hay un máximo de cinco ventanas abiertas a la vez	
10	Se llama Sentencia a...	BIEN/MAL
<input type="checkbox"/> A	Un conjunto de instrucciones almacenadas en una consulta para su ejecución	
<input type="checkbox"/> B	Un conjunto de datos almacenadas en un archivo	
<input type="checkbox"/> C	Un conjunto de instrucciones almacenadas en un archivo .CSV o .PDF para su ejecución	
<input type="checkbox"/> D	Un conjunto de campos almacenadas en una Tabla	