

SQL Mastery

EXTRACTO DE MUESTRA (LECCIONES 1-3)

Índice de Contenidos

Introducción	3
1 - Bases de Datos Relacionales	4
2 - Consultas (SELECT FROM WHERE)	7
3 - Consultas II (Cláusulas SELECT FROM WHERE)	12
4 - Tipos de Datos	17
5 - Operadores Lógicos	24
6 - Funciones de Agregado	37
7 - Agrupación de Datos (GROUP BY)	47
8 - Filtros HAVING	58
9 - Ordenación de Datos	64
10 - LIKE y Valores NULL	70
11 - Síntesis Fundamentos SQL	78
12 - Producto Cartesiano	83
13 - Consultas III (Múltiples Tablas)	96
14 - Relaciones y Claves	106
15 - Reunión de Datos (JOIN)	119
16 - Modelo Entidad-Relación (E-R)	132
17 - Lenguaje de Definición de Datos (DDL)	147
18 - Lenguaje de Manipulación de Datos (DML)	161
19 - Síntesis Arquitectura SQL	168
20 - Funciones Nativas	173
21 - Subconsultas SELECT	190
22 - Subconsultas FROM y CTEs	199
23 - Subconsultas WHERE	221
24 - Funciones de Ventana I	240
25 - Funciones de Ventana II	254
26 - Vistas	277
27 - Cargas Masivas de Datos	286
28 - UPDATE con Subconsultas	294
29 - Funciones y Procedimientos	308
30 - Triggers	345
31 - Síntesis Ingeniería SQL	354
32 - Aplicación SQL	362

Introducción

Bienvenido a SQL Mastery

He diseñado este curso con un propósito muy claro: **romper la barrera del lenguaje técnico y hacer que el SQL sea más accesible y fácil de entender.**

A menudo, las bases de datos parecen complejas porque se explican con tecnicismos innecesarios. Mi objetivo con **SQL Mastery** es diferente: he convertido esa complejidad en conceptos sencillos, utilizando símiles cotidianos y un lenguaje cercano para que entiendas la lógica real detrás de los datos, sin importar tu nivel previo.

¿Qué hace diferente a este curso?

- **Claridad ante todo:** Aprenderás SQL de cero a experto, pero con un enfoque divulgativo que prioriza *entender* lo que haces, memorizar comandos no conduce a ningún sitio.
- **Práctica Real:** Simplificamos la teoría densa. Contarás con ejercicios resueltos y una **Consola SQL interactiva**, un *Query Lab* con **asistente de IA**, para que no te atasques resolviendo los ejercicios propuestos en cada lección. Con la Consola SQL podrás experimentar y *jugar* con el código de ejemplo del curso. Una herramienta fundamental para consolidar tus conocimientos con la práctica.
- **Aprendizaje Ameno:** He estructurado el contenido para evitar que sea tedioso. *Descubrirás que el SQL es mucho más intuitivo de lo que parece cuando lo abordamos con un lenguaje cercano.*

Un consejo antes de empezar

Tómatelo con paciencia y lee las lecciones con calma, asimilando lo que se explica. Es muy importante que trabajes los conceptos y te enfrentes a los ejercicios antes de solicitar la función de ayuda o mirar las soluciones. Recuerda: **equivocarse es parte del camino; hallar la solución es solo la confirmación del aprendizaje**.

Espero que disfrutes aprendiendo tanto como yo he disfrutado creando este contenido. Si logras comprender la lógica de las bases de datos y dominar SQL, habremos cumplido el objetivo.

Gracias y ¡mucho éxito!

1 - Bases de Datos Relacionales

Base de Datos

Las bases de datos existen desde que el ser humano empezó a almacenar datos en algún soporte. Si por datos entendemos dibujos, que lo son, entonces las primeras bases de datos fueron las paredes de las cuevas donde nuestros ancestros dibujaron las pinturas rupestres.

Posteriormente los egipcios crearon grandes estructuras arquitectónicas que usaron, entre otras cosas, como soporte para almacenar datos y narrar la historia del antiguo Egipto en sus paredes. El tiempo transcurrió hasta el punto de que el significado de todos esos símbolos se perdió, sin embargo la base de datos perduró lo suficiente para que alguien consiguiera descifrar los jeroglíficos a tiempo, de modo que todos esos datos, esa faraónica base de datos, cobró de nuevo todo su sentido. De hecho el valor de toda esa información es mayor que todos los tesoros que pudiesen esconder tumbas y templos. Los arqueólogos esperan encontrar en los nuevos hallazgos, antes que objetos y tesoros, nuevos jeroglíficos que les permitan conocer algún episodio olvidado de la historia de esta fascinante civilización. En ocasiones es esa misma información la que proporciona las pistas para descubrir nuevos hallazgos.

En la actualidad las bases de datos informáticas han quitado todo el protagonismo a sus antecesoras, los archivos de papel, que aún se siguen usando en algunos ámbitos concretos. De bases de datos informáticas han existido de varios tipos, pero las que más han proliferado son las que se tratarán en este curso, las bases de datos relacionales. Mencionar que antes de estas últimas se usaron las bases de datos jerárquicas y posteriormente las bases de datos en red, actualmente sistemas en desuso.

Las bases de datos relacionales están plenamente asentadas en los sistemas de información informáticos, pero recientemente ha surgido un nuevo paradigma de bases de datos conocidas como NoSQL, un ejemplo es MongoDB, cuyo modo de funcionamiento es muy distinto al de las bases de datos relacionales que trata este curso. Estas bases de datos han sido impulsadas por las grandes compañías de Internet como Google, Facebook... ante la necesidad de procesar un enorme volumen de datos en tiempos aceptables.

Para encauzar el aprendizaje del lenguaje de consulta SQL empezaremos por conocer la estructura de almacenamiento que usa una base de datos relacional. En este caso no son paredes, ni montones de papel lo que se usa para almacenar la información, sino que se almacena en soportes informáticos bajo una estructura lógica de almacenamiento, como la tiene un archivo de papel, por ejemplo: edificio, planta, pasillo, ubicación, ficha. De este modo es posible recuperar la información que interesa de un modo ágil, gracias a los índices y la estructura organizada del archivo. A continuación se verá cómo estructura la información una base de datos relacional, pero antes, establezcamos unas pocas definiciones.

Base de Datos Relacional

Una base de datos ([BD](#)), o mejor dicho, un sistema gestor de bases de datos ([SGBD](#)), es un software que gestiona una o más bases de datos y nos permite explotar los datos almacenados en ellas de forma relativamente simple

mediante SQL.

Esta es una definición muy simplificada, pero para que el aprendizaje sea distendido lo supondremos así, de ese modo podemos centrarnos en aprender cómo y con qué propósito accedemos a los datos, dejando para el final cómo creamos, alimentamos o modificamos la BD.

Algunos ejemplos de SGBD son: [Oracle](#), [MariaDB](#), [MS SQL Server...](#)

En este curso se empleará un SGBD MariaDB, de modo que los ejemplos ejecutables y ejercicios propuestos están diseñados para MariaDB, y la Consola SQL accede a una base de datos MariaDB. **No se debe confundir con un curso para MariaDB**, no lo es, aplicar lo aprendido a uno u otro SGBD será cuestión únicamente de conocer la sintaxis de cada sistema y sus funcionalidades para interactuar con sus bases de datos.

Por ejemplo, si usted realiza un curso para escritores en castellano, donde aprende técnicas y trucos para escribir un thriller, es de esperar que no tenga que realizar el mismo curso en francés porque desea escribir su thriller en francés, para ello bastará con que sepa usted francés. Afortunadamente el estándar SQL empleado por los distintos SGBD es muy similar y en muchas cosas idéntico, no comparable a las diferencias que encontramos entre dos idiomas como puedan ser el castellano y el francés.

A lo largo del curso, especialmente en la tercera parte dedicada a conceptos avanzados, se comparan las extensiones que los principales SGBD implementan fuera del estándar SQL ANSI, mostrando las variaciones de sintaxis utilizadas por cada motor para realizar las mismas tareas.

Estructura Mínima de Almacenamiento

- **Tabla:** Objeto de almacenamiento perteneciente a una BD. Es una estructura en forma de cuadrante donde se almacenan registros o filas de datos. Cada tabla tiene un nombre único en la BD.
- **Registro:** Cada una de las filas de una tabla, está compuesta por campos o atributos.
- **Campo:** Cada uno de los “cajoncitos” de un registro donde se guardan los datos. Cada campo tiene un nombre único para la tabla de la cual forma parte, además es de un tipo (naturaleza) determinado, por tanto no podemos guardar limones en el cajón de las naranjas, en términos informáticos y a modo de ejemplo, no encontraremos un dato alfanumérico (letras y números) en un campo diseñado para guardar datos numéricos. Dedicaremos una lección a los tipos de datos más adelante.

Por el momento estas son las definiciones que necesitamos, veamos ahora un ejemplo concreto de tabla.

Ejemplo de Tabla

Tabla EMPLEADOS:

ID_EMPLEADO	NOMBRE	APELLIDOS	F_NACIMIENTO	SEXO	CARGO	SALARIO
1	Carlos	Jiménez Clarín	1985-05-03	H	Mozo	1500.00
2	Elena	Rubio Cuestas	1978-09-25	M	Secretaria	1300.00
3	José	Calvo Sisman	1990-11-12	H	Mozo	1400.00
4	Margarita	Rodriguez Garcés	1992-05-16	M	Secretaria	1325.50

Cada registro o fila de datos contiene información de un empleado. En el ejemplo observamos que la tabla tiene un diseño de siete campos y que almacena cuatro registros. El nombre de cada campo viene dado por la fila de encabezado. El dato que contiene el campo ID_EMPLEADO identifica cada registro, pero por ahora no le demos importancia a esto.

Los registros o miembros de una tabla tienen en común sus atributos, no el dato en sí, que lo más probable es que difiera de un registro a otro, pero sí el hecho de que todos ellos poseen esos atributos. En el ejemplo los miembros de la tabla EMPLEADOS tiene en común que todos ellos son personas empleadas en una empresa, que tienen un nombre y un salario, una fecha de nacimiento, etc. Por lo tanto las tablas de una BD guardan información de individuos o unidades de una misma naturaleza con una serie de atributos en común.

Resumen

Una BD contendrá tablas que a su vez contendrán registros y en estos se encontrarán los datos distribuidos en una serie de campos. Cada registro de la tabla guarda la información particular de una unidad o miembro de un mismo grupo. El SGBD cumple la función de interfaz, entre el usuario y la BD, permitiéndonos interactuar con ella mediante **SQL**.

2 - Consultas (SELECT FROM WHERE)

Si este fuese un manual de SQL, o alguno de los muchos cursos SQL que se pueden encontrar por la web, ahora sería el momento de abordar la instalación de un SGBD para posteriormente crear una BD con algunas tablas de ejemplo con las que empezar a trabajar. Pero este curso pretende ser diferente. Si ahora se expusiera lo antes mencionado, corremos el riesgo de que usted pierda el interés por la materia; además, el SQL no incluye la instalación del SGBD ni la creación de tablas. Si usted, por ejemplo, desea aprender electricidad, ¿acaso ha de fabricar bombillas, cables y el generador eléctrico con el que poder trabajar? Obviamente no.

En este curso la BD y las tablas con las que trabajar las tiene accesibles mediante la **Consola SQL** (nuestro "laboratorio de pruebas"), así que empezaremos directamente por lo que será la tónica de este curso: las consultas SQL, que es digamos donde está la miga, y no será hasta más adelante, en la segunda parte del curso, donde veremos la creación de tablas y cómo modificar la información. A fin de cuentas no tiene demasiado sentido aprender a crear tablas cuando aún no sabe qué hacer con ellas.

Consultas SQL

Abordemos las consultas SQL con un caso práctico. Sobre la tabla EMPLEADOS se plantea la siguiente cuestión:

¿Qué empleados tienen un salario mayor a 1350?

ID_EMPLEADO	NOMBRE	APELLIDOS	F_NACIMIENTO	SEXO	CARGO	SALARIO
1	Carlos	Jiménez Clarín	1985-05-03	H	Mozo	1500.00
2	Elena	Rubio Cuestas	1978-09-25	M	Secretaria	1300.00
3	José	Calvo Sisman	1990-11-12	H	Mozo	1400.00
4	Margarita	Rodriguez Garcés	1992-05-16	M	Secretaria	1325.50

La respuesta es simple: José y Carlos tienen un salario mayor a 1350, pero si tuviésemos 500 empleados nos llevaría más tiempo responder, y al final tampoco tendríamos la certeza de no habernos equivocado. El SQL nos permite responder estas preguntas de forma rápida y fiable, salvo error al construir la consulta o errores en los propios datos.

Vamos pues a construir la consulta que nos permita responder a esta cuestión.

Preguntas de Construcción

Para construir una consulta SQL debemos hacernos como mínimo tres preguntas:

1. ¿Qué datos nos están pidiendo?
2. ¿Dónde están esos datos?

3. ¿Qué requisitos deben cumplir los registros?

En este caso, la respuesta a esas preguntas sería:

- El nombre y los apellidos de los empleados.
- En la tabla EMPLEADOS.
- Que el sueldo del empleado sea superior a 1350.

Vamos a suponer por un momento que el SGBD, que es quien intermedia entre el usuario y la BD, fuese una persona; la BD un archivo de papel, y el jefe pide lo siguiente: "Necesito saber qué empleados cobran más de 1350 euros". Usted, que conoce bien el archivo (tablas) y qué datos contiene la ficha de un empleado (campos de la tabla EMPLEADOS), mentalmente se hace las preguntas de construcción y le dice a su ayudante, que siempre espera órdenes concretas:

Seleccióname el NOMBRE y los APELLIDOS
del archivo EMPLEADOS
cuyo SALARIO sea mayor a 1350

El ayudante sirve la petición y se la entrega para que finalmente usted se la facilite a su jefe. ¿Qué papel ocuparían hoy en una empresa moderna? El jefe sigue siendo el jefe, eso está claro, pero usted ha pasado a ser el informático, y su ayudante el SGBD.

Sintaxis SQL

En SQL la forma de operar es parecida, esta información se obtiene mediante la siguiente consulta:

```
select NOMBRE, APELLIDOS  
      from EMPLEADOS  
     where SALARIO > 1350
```

Obsérvese que en la consulta escribimos los nombres de los **objetos** de la base de datos (tablas y campos) en mayúsculas, mientras que las palabras reservadas (**SELECT, FROM, WHERE**) aparecen en minúsculas. Esto responde a un propósito meramente estético, con la intención de hacer el código más ordenado y legible. No obstante, tenga en cuenta que, dependiendo del SGBD y del sistema operativo (Windows, Linux, etc.) en el que se ejecute, el sistema puede ser sensible a mayúsculas y minúsculas en lo que respecta a los nombres de los objetos, aunque no así en las palabras reservadas.

Y el resultado que nos devuelve el SGBD es:

NOMBRE	APELLIDOS
Carlos	Jiménez Clarín
José	Calvo Sisman

Parecido a lo que nos entregaría nuestro ayudante en el archivo, una lista con la respuesta o solución a la cuestión planteada. Como ve, tanto el modo de solicitar la información al SGBD, sea clásico o informatizado, como el modo en que este nos muestra la información, son muy similares, al menos en este caso. No podemos afirmar lo mismo del tiempo que uno y otro tardan en facilitarnos la información solicitada.

Forma General

En general una consulta SQL simple tendrá la siguiente forma:

```
select CAMPOS(separados por comas)
      from TABLA
     where CONDICION
```

El SQL permite al usuario desentenderse de cómo el SGBD ejecuta la consulta, al igual que usted se desentiende de cómo su ayudante en el archivo de papel se las ingenia para facilitarle la información. Usted esperará pacientemente tras el mostrador a que su ayudante prepare su pedido y le entregue los datos. Dicho de otro modo, basta con saber cómo pedir la información y no cómo proceder a reunirla.

Resumen

Hemos visto cómo construir una consulta SQL simple y concreta, que nos da la solución a una cuestión concreta.

Se ha definido la forma general de una consulta SQL simple:

```
select CAMPOS(separados por comas)
      from TABLA
     where CONDICION
```

Destacar también la utilidad de las preguntas de construcción para ayudarnos a construir la consulta:

1. ¿Qué datos nos piden?
2. ¿Dónde están los datos?
3. ¿Qué requisitos debe cumplir los registros?

En una empresa moderna un informático cumple la función de encargado del archivo, y sus ayudantes son hoy los sistemas informatizados.

Ejercicio Propuesto

Intente hallar una consulta que devuelva el nombre, apellidos y la fecha de nacimiento de aquellos empleados que cobren más de 1350 euros.

Puede probar su funcionamiento en la Consola SQL .

Sugerencia: si envía el código SQL que sigue a este párrafo desde la consola al SGBD, este le reportará la lista de campos de la tabla EMPLEADOS bajo una columna titulada **Field** (término en inglés que significa campo). Esto le puede venir bien para tener presente los nombres de los campos de la tabla EMPLEADOS mientras desarrolla el ejercicio.

```
describe EMPLEADOS
```

La siguiente sentencia más simplificada es equivalente:

```
desc EMPLEADOS
```

SOLUCIÓN EJERCICIO

```
select NOMBRE, APELLIDOS, F_NACIMIENTO  
      from EMPLEADOS  
     where SALARIO > 1350
```

NOMBRE	APELLIDOS	F_NACIMIENTO
Carlos	Jiménez Clarín	1985-05-03
José	Calvo Sisman	1990-11-12

3 - Consultas II (Cláusulas SELECT FROM WHERE)

En la lección anterior hemos construido con éxito nuestra primera consulta:

```
select NOMBRE, APELLIDOS  
      from EMPLEADOS  
     where SALARIO > 1350
```

Las tres cláusulas y las Preguntas de Construcción

Fijémonos ahora en las tres cláusulas de la anterior consulta SQL y qué relación guardan con las preguntas de construcción:

- **Cláusula SELECT:** Donde indicamos los campos de la tabla que queremos obtener, separados por comas.
Responde a la pregunta: ¿Qué datos nos piden?
 - **Cláusula FROM:** Donde indicamos en qué tabla se encuentran estos campos. Responde a la pregunta: ¿Dónde están los datos?
 - **Cláusula WHERE:** Donde establecemos la condición que han de cumplir los registros de la tabla que serán seleccionados. Responde a la pregunta: ¿Qué requisitos deben cumplir los registros? Es, de hecho, donde se establece el filtro de registros, es decir, qué registros serán considerados para mostrar sus datos y cuáles no.
-

Modificando la Cláusula Where

¿Qué empleados tienen un sueldo comprendido entre 1350 y 1450?

ID_EMPLEADO	NOMBRE	APELLIDOS	F_NACIMIENTO	SEXO	CARGO	SALARIO
1	Carlos	Jiménez Clarín	1985-05-03	H	Mozo	1500.00
2	Elena	Rubio Cuestas	1978-09-25	M	Secretaria	1300.00
3	José	Calvo Sisman	1990-11-12	H	Mozo	1400.00
4	Margarita	Rodriguez Garcés	1992-05-16	M	Secretaria	1325.50

Si nos hacemos las preguntas de construcción:

1. ¿Qué datos nos piden?
2. ¿Dónde están los datos?
3. ¿Qué requisitos deben cumplir los registros?

Observamos que para las dos primeras preguntas las respuestas son idénticas a la anterior cuestión, pero para la tercera es distinta. Esto nos indica que las cláusulas `SELECT` y `FROM` no van a cambiar respecto a la anterior consulta, y solo lo hará la cláusula `WHERE`, así que podemos tomar la anterior consulta como patrón y modificarla para adaptarla a lo que se nos pide ahora.

Consulta patrón:

```
select NOMBRE, APELLIDOS  
  from EMPLEADOS  
 where SALARIO > 1350
```

Antes el salario debía ser mayor a 1350, ahora debe estar comprendido entre 1350 y 1450, ambos inclusive. La cláusula `WHERE` la construiremos de la siguiente manera:

```
where SALARIO >= 1300 and SALARIO <= 1450
```

Y se lee así: donde el salario sea mayor o igual a 1350 y menor o igual a 1450.

La consulta quedaría:

```
select NOMBRE, APELLIDOS  
  from EMPLEADOS  
 where SALARIO >= 1300 and SALARIO <= 1450
```

Si comparamos las dos consultas, se observa cómo únicamente difieren en la cláusula `where`.

El Operador BETWEEN

Trataremos a fondo los operadores mas adelante, así que a modo de apunte comentar que existe otro modo de obtener el mismo resultado aprovechando más los recursos del SQL mediante el operador `BETWEEN` (entre). La consulta es equivalente y quedaría de la siguiente manera:

```
select NOMBRE, APELLIDOS  
  from EMPLEADOS  
 where SALARIO between 1350 and 1450
```

Es decir: donde el salario esté entre 1350 y 1450 ambos inclusive.

Y el resultado que nos devuelve el SGBD es:

NOMBRE	APELLIDOS
José	Calvo Sisman

Comentarios en el código SQL

A medida que nuestras consultas ganan complejidad, puede resultar útil añadir notas explicativas que ayuden a comprender la lógica del código. Estas anotaciones son ignoradas por el motor del SGBD durante la ejecución. En SQL disponemos de dos métodos para comentar:

- **Comentarios de línea:** Se utilizan dos guiones seguidos "--". El motor ignorará todo lo que se escriba desde los guiones hasta el final de esa línea.
- **Comentarios de bloque:** Se delimitan entre los símbolos "/*" para abrir y "*/" para cerrar. Son ideales para explicaciones largas o encabezados que ocupan varias líneas.

Veamos cómo aplicaríamos ambos tipos de comentarios a nuestra consulta:

```
/* Consulta que filtra empleados
   por un rango salarial específico */
select NOMBRE, APELLIDOS -- Campos a mostrar
      from EMPLEADOS          -- Origen de los datos
     where SALARIO between 1350 and 1450
```

El Motor SQL

Aunque en la lección anterior se dijo que el SQL nos permite desentendernos de cómo se reúnen los datos, en un futuro nos vendrá bien entender de forma lógica su manera de proceder. Veamos cómo ejecuta esta consulta el motor SQL del SGBD. Primero seleccionará los registros que cumplen la condición de la cláusula **WHERE**, para ello debe recorrer todos los registros de la tabla y decidir, en función de la condición, si lo toma en consideración o no. Al final reunirá los campos indicados en la cláusula **SELECT** de la tabla indicada en la cláusula **FROM** cuyos registros han sido seleccionados por la cláusula **WHERE**.

Resumen

Hemos dividido una consulta SQL concreta en tres cláusulas, se ha relacionado cada cláusula con las preguntas de construcción, hemos tomado como patrón una consulta para modificarla y adaptarla a los nuevos requisitos.

Ejercicio Propuesto

Intente hallar una consulta que devuelva el nombre y apellidos de los empleados que cobren menos de 1350 euros.

SOLUCIÓN EJERCICIO

```
select NOMBRE, APELLIDOS  
      from EMPLEADOS  
     where 1350 > SALARIO
```

O bien:

```
select NOMBRE, APELLIDOS  
      from EMPLEADOS  
     where SALARIO < 1350
```

Y el resultado que nos devuelve el SGBD es:

NOMBRE	APELLIDOS
Elena	Rubio Cuestas
Margarita	Rodriguez Garcés

¿Te gusta el Curso SQLMastery?

El PDF Completo contiene **29 lecciones más.**

Adquiere el Plan Maestría y descarga el Curso SQLMastery completo ahora.

[OBTENER VERSIÓN COMPLETA](#)