

APRENDE **SQL** DE 0 A ANALISTA DE DATOS

Aprende, practica y resuelve
desafíos de datos con SQL.



Adrian Rodriguez

1. Bienvenida del Curso de SQL	6
Bienvenida	6
Motivación del libro	6
2. Introducción a SQL	7
¿Qué es SQL?	7
¿Por que es importante SQL?	7
Quiz	8
3. ¿Qué es una base de datos?	8
Tipo de bases de datos	8
Bases de datos SQL	8
Bases de datos NoSQL	9
Tablas	9
Columnas	10
Filas	10
Tipos de relaciones entre tablas	10
Primary Keys y Foreign Keys	11
¿Qué es una Primary Key?	11
¿Qué es una Foreign Key?	12
Quiz	12
4. Modelo Entidad-Relación	13
¿Que es un modelo entidad-relación?	13
Ejemplo de modelo entidad-relación	13
Quiz	15
5. Instalación DBeaver y base de datos de práctica	16
¿Qué es DBeaver?	16
Instalación DBeaver	17
Instalación de base de datos para prácticas	24
7. Explorando la base de datos Northwind. Tus primeras SELECT	27
8. Leyendo datos de una tabla con SELECT	32
Uso del comando SELECT	32
Uso de alias	33
Quiz	35
Ejercicios (usa la bbdd de Northwind)	36
9. Filtrando datos de una consulta con WHERE	36
La cláusula WHERE	36
Condiciones de filtrado múltiple. Uso de AND/OR	38
Uso de AND	38
Uso de OR	39
Uso de IN	39

Múltiples condiciones	40
Lista de operadores habituales	41
Quiz	41
Ejercicios (usa la bbdd Northwind)	42
10. Ordenando datos de una consulta. Uso de ORDER BY	42
Uso de ORDER BY	42
Ordenar por varios campos	44
Quiz	44
Ejercicios	45
11. Consultando datos de dos o más tablas. Uso y tipos de JOINS	45
INNER JOIN	46
LEFT JOIN	48
RIGHT JOIN	49
FULL OUTER JOIN	50
Leer y combinar datos de dos o más tablas	50
Quiz	51
Ejercicios (usa la bbdd Northwind)	52
13. Agregación de resultados. Uso de GROUP BY	52
Agregaciones	52
Uso de la cláusula GROUP BY	54
HAVING	55
Quiz	56
Ejercicios (usa la bbdd Northwind)	57
14. Subconsultas	57
Quiz	59
Ejercicios (usa la bbdd Northwind)	59
15. Funciones Ventana	60
Quiz	62
Ejercicios (usa la bbdd Northwind)	62
16. CTEs y Vistas	62
CTEs	62
Vistas	64
Quiz	67
Ejercicios (usa la bbdd Northwind)	67
17. Funciones	67
Funciones de fecha	68
YEAR()	68
MONTH()	68
DATEDIFF	69
Funciones de cadena	69
LEFT	70
RIGHT	70
CONCAT	71
Funciones matemáticas	72

Funciones Ventana (Ampliación)	73
ROW_NUMBER() y DENSE_RANK()	73
LEAD() y LAG()	75
Funciones ventana matemáticas	76
Quiz	77
Ejercicios (usa la bbdd Northwind)	78
18. CASE	78
Quiz	80
Ejercicios (usa la bbdd Northwind)	81
19. UNION y UNION ALL	81
UNION	81
UNION ALL	83
Quiz	83
Ejercicios (usa la bbdd Northwind)	83
20. Creando tablas	84
CREATE TABLE	84
INSERT INTO	85
AUTOINCREMENT	86
Quiz	87
Ejercicios (usa la bbdd Northwind)	88
21. Actualizando datos	89
Quiz	90
Ejercicios (usa la bbdd Northwind)	91
22. Borrando Datos	91
Quiz	93
Ejercicios (usa la bbdd Northwind)	93
23. Índices	93
Quiz	94
Ejercicios (usa la bbdd Northwind)	95
24. Bases de datos Operacionales vs Analíticas	95
Bases de datos operacionales	95
Bases de datos analíticas. Modelo en estrella	96
Tabla de hechos	96
Dimensiones	97
Quiz	99
Ejercicios (usa la bbdd Northwind)	99

Edición: Julio de 2026

@ Adrian Rodriguez Gutierrez, 2026

<https://www.linkedin.com/in/stackedanalyst>

Todos los derechos reservados

No se permite la reproducción total o parcial de este libro, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio, sea este electrónico, mecánico, por fotocopia, por grabación u otros métodos, sin el permiso previo del autor. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual.

Cómo usar este libro

Este libro sigue una serie de capítulos de niveles mas basico a avanzado sobre SQL, lo que te servira para ganar conocimientos de teoria sobre bases de datos donde podras encontrar ademas de las explicaciones ejemplos de codigo resaltados con **este formato** es decir en blanco y fondo negro para diferenciarlo del resto del texto. Así mismo al final de cada capítulo encontrarás un quiz con preguntas que te servirán para probar tu aprendizaje y afianzar tus conocimientos, a lo que también al final de cada capítulo encontrarás unos ejercicios que puedes practicar con el software y la base de datos que podrás instalar siguiendo las instrucciones en capítulos posteriores (Instalación DBeaver y base de datos para practicas).

1. Bienvenida del Curso de SQL

2 de Julio 2026 [Adrian Rodriguez](#)

Bienvenida

Hola, bienvenida/o a este curso de SQL. Me llamo Adrian y soy un apasionado de la tecnología, siendo así que llevo mas de 10 años trabajando en el campo de los datos, siendo SQL la herramienta que más útil, duradera, y que uso en mi dia a dia durante todo este tiempo.

He creado este libro/curso de SQL por que al igual que mi vida dio un cambio y mejoro cuando di por casualidad hace muchos años con un PDF de SQL sobre fundamentos de este lenguaje (aunque en inglés), lo que me proporcionó la oportunidad de conseguir mi primer trabajo de datos (por aquellos tiempos incluso aún es cuando se empezaba a hablar de la ciencia de datos y ni siquiera apenas existían este tipo de puestos, y menos en España, de donde soy). Si bien el PDF estaba bien pero era muy básico y tuve que esforzarme mucho por aprender rápido, ya sea por mi cuenta o preguntando a otros compañeros más senior, superando múltiples retos, por ello este libro está pensado para evitarte años de prueba y error o que incluso puedas empezar tu carrera profesional con SQL en el campo de los datos con buen nivel.

Motivación del libro

Este es un curso de SQL que recoge los fundamentos del lenguaje SQL de bases de datos así como conceptos bastante avanzados para su uso en el día a día como analista de datos. Con el contenido de este curso de SQL podrás ganar los conocimientos y la confianza necesarias para poder seguir creciendo en el campo de la tecnología y los datos, pues nunca hay que dejar de aprender, el mundo evoluciona rápido, y el conocimiento es un activo que siempre estará con nosotros y nos permitirá alcanzar nuestras metas.

El campo de la tecnología no deja de evolucionar constantemente, sin embargo SQL es un lenguaje de consulta de bases de datos que ha mejorado pero sigue siendo durante años uno de los principales elementos a la hora de trabajar con datos, ya sea a la hora de desarrollar aplicaciones empresariales de gestión o para poder realizar análisis de datos.

SQL es muy útil y necesario tanto para desarrolladores como para profesionales de datos, siendo así que tanto si eres analista de datos, científico de datos o ingeniero de datos, muy seguramente tendrás que usar SQL para trabajar con bases de datos.

2. Introducción a SQL

¿Qué es SQL?

SQL es el acrónimo de Structured Query Language, que traducido al Español significa lenguaje estructurado de consultas, en este caso de consultas a bases de datos.

SQL sigue un estándar ANSI, lo que quiere decir que muchos de sus comandos son comunes en los diferentes sabores de SQL, como puede ser T-SQL, MySQL, PL/SQL, PostgreSQL o SQLite, entre otros.

Fue desarrollado por IBM en los años 70 para el manejo de base de datos y se ha convertido en uno de los lenguajes más importantes a la hora de gestión y análisis de datos. Desde los años 70 SQL ha ido evolucionando de manera que ha ido adoptando nuevas capacidades para trabajar con datos.

SQL está ligado al concepto de base de datos relacional, esto es, un servidor de almacén de datos estructurados, que tienen como base en un RDMS, o Relational Database Management System. Con el tiempo estos servidores según el fabricante han ido adquiriendo nuevas capacidades para el manejo de datos, como pueden ser no muy recientemente la capacidad de la paralelización de consultas, lo que permite extraer datos de muy grandes volúmenes de datos de manera mucho más rápida y eficiente.

¿Por que es importante SQL?

La importancia de SQL radica entre otras razones porque hoy se ha convertido en una forma muy potente de poder gestionar y acceder a altos volúmenes de datos que se generan en el día a día. Muchas son las empresas en todo el mundo que almacenan sus datos en formato relacional, o mejor dicho, en forma de datos estructurados, y SQL es la forma de acceder y manipular estos datos.

SQL permite leer, organizar y agrupar datos, lo que para el análisis de datos o la programación lo convierte en una herramienta/lenguaje ideal para muchos cometidos en la gestión y almacenaje de datos.

SQL a dia de hoy es uno de los lenguajes más demandados en el mercado laboral, y es que si se generan muchos datos por parte de empresas y organizaciones, alguien ha de ser capaz también de poder extraer y analizar esos datos para obtener un valor que muchas veces se convierte en mejores decisiones y mejora de procesos de negocio, permitiendo a las empresas ser más eficientes y ganar en

ventajas competitivas, algo crucial en el mercado global y competitivo en el que vivimos hoy en día.

Quiz

1. ¿Qué significan las siglas SQL?

- a) Structured Query Language
- b) Simple Question Language
- c) System Query Logic
- d) Standard Query Language

2. ¿Qué es un RDMS?

- a) Relational Database Management System
- b) Random Data Memory Storage
- c) Remote Database Module Service
- d) Relational Data Model Schema

3. SQL está ligado al concepto de:

- a) Base de datos relacional
- b) Hoja de cálculo
- c) Archivo de texto
- d) Cache en memoria

3. ¿Qué es una base de datos?

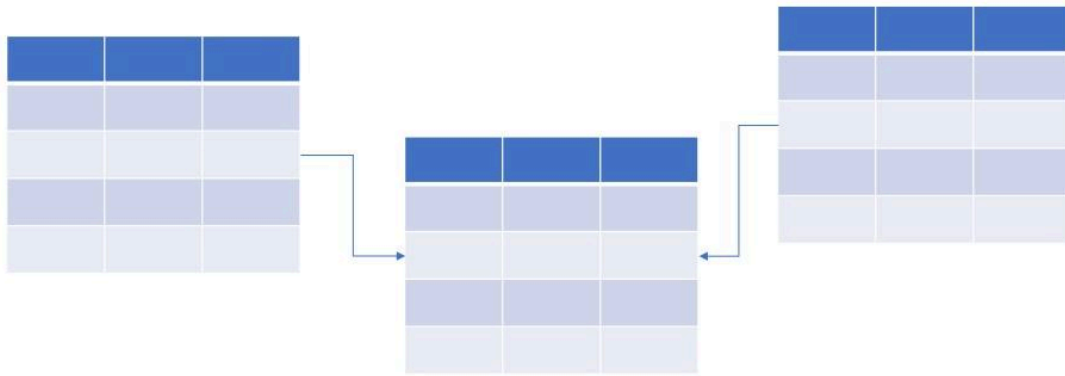
Una base de datos es un sistema desarrollado mediante software, que permite el almacenamiento, manipulación y recuperación de datos.

Tipo de bases de datos

Hay que diferenciar entre bases de datos SQL y bases de datos NoSQL. Este curso trata sobre las bases de datos SQL.

Bases de datos SQL

Son aquellas en las que la información es estructurada, esto es, en formato de tablas con columnas y filas. Un ejemplo claro de tabla puede ser una hoja de cálculo de excel. Una base de datos SQL, también denominada comúnmente como base de datos relacional, debe su nombre a la posibilidad de relacionar las diferentes tablas que la componen.



Bases de datos NoSQL

Las bases de datos NoSQL se componen de información que no sigue un patrón (o mejor dicho esquema) de columnas y filas concretas. Por ejemplo, una base de datos NoSQL podría ser un sistema de un conjunto de archivos tipo clave-valor. La forma común de almacenar la información en una base de datos NoSQL es mediante archivos tipo Json. A continuación te muestro un ejemplo de documento de una base de datos NoSQL.

```
{
  "cliente_id": "12345",
  "nombre": "Juan Pérez",
  "email": "juan.perez@example.com",
  "telefono": "+123456789",
  "direccion": {
    "calle": "Av. Siempre Viva",
    "numero": 742,
    "ciudad": "Springfield",
    "pais": "USA"
  },
  "fecha_registro": "2022-05-15",
  "pedidos": [
```

Tablas

Una tabla en una base de datos es un objeto que se compone de filas y columnas (estas también llamadas comúnmente campos) al igual que podríamos tener, dado el ejemplo, en las hojas de cálculo (como Excel o las hojas de cálculo de Google).

Cada tabla en una base de datos representa una entidad, como pueden ser clientes, productos u órdenes de ventas¹.

	Columna 1	Columna 2	Columna 3
Fila 1			
Fila 2			
Fila 3			

Columnas

Una columna en una tabla de base de datos contiene la información de un atributo de la entidad que representa la tabla. Una columna tiene asociado un tipo de datos, como puede ser entero, decimal, cadena, binario o de tipo fecha. Por ejemplo una tabla en una base de datos que representa la entidad Empleados puede tener campos tales como Nombre, Fecha de incorporación, Dirección, Departamento, Responsable, Salario, etc, donde el tipo de dato de cada campo puede ser como sigue;

- Fecha de incorporación->fecha
- Dirección->cadena
- Departamento->cadena
- Responsable->cadena
- Salario->numérico

Filas

Las filas de una tabla de base de datos, también llamadas comúnmente registros, representan un evento o instancias específicas de una entidad. Por ejemplo en una tabla una fila puede representar un cliente con los datos de Nombre, Dirección, Identificador de cliente, Email, etc.

Tipos de relaciones entre tablas

¹ Dado el caso de un ejemplo de base de datos de un comercio. Una base de datos de este tipo suele recoger datos de los clientes, los productos que vende el comercio y así mismo un historico de ordenes de ventas.

Las relaciones en una base de datos son las conexiones entre las diferentes tablas que pueden componer la base de datos. Las relaciones nos permiten referir y combinar los datos entre las tablas. El tipo de relación entre tablas también se denomina cardinalidad.

Las relaciones entre tablas nos permiten organizar la información de manera coherente.

Existen 3 tipos principales de relaciones entre tablas (o cardinalidad):

- Uno a uno (1:1): Una relación uno a uno entre dos tablas significa que cada fila o registro de la primera tabla se relaciona con un único registro de la segunda tabla. Esto es, por ejemplo, una tabla Empleados donde podemos tener el nombre del empleados y los apellidos en dos columnas, se puede relacionar con otra tabla llamada direcciones donde cada empleado tendría una sola dirección en la segunda tabla de direcciones. Este tipo de relación en un diagrama entidad relación se representa por 1:1.
- Uno a muchos (1:n): Este tipo de relación se da cuando un registro de una tabla puede estar relacionado con uno o más registros de otra tabla, por ejemplo, cada registro de una tabla llamada clientes puede relacionarse con otra tabla llamada pedidos donde cada cliente puede tener uno o más pedidos.
- Muchos a muchos (n:n): Este tipo de relación se da cuando una tabla contiene diferentes registros que se relacionan también con diferentes registros de otra tabla. Por ejemplo, una tabla llamada estudiantes puede relacionar de manera muchos a muchos con otra tabla llamada asignaturas.

Primary Keys y Foreign Keys

Igual que hemos hablado de los tipos de relaciones entre tablas de una base de datos, debemos mencionar también lo que se denominan Primary keys y Foreign Keys. Estas son claves en una tabla, que nos permitirán también poder conocer cómo relacionar las tablas entre sí.

¿Qué es una Primary Key?

Una Primary Key (PK), también llamada clave primaria, es una columna donde cada valor de los diferentes registros de las filas de este campo es único para esa columna. Por ejemplo una tabla de clientes con una columna IdCliente puede ser el

identificador de cada cliente dentro de la tabla. Una tabla por norma suele tener solo una primary key, si bien puede haber primary keys compuestas de varios atributos dados por otras columnas de la tabla.

¿Qué es una Foreign Key?

Una Foreign Key (FK), o clave foránea, es una clave externa hacia otra tabla. Por ejemplo, la misma tabla clientes que mencionamos arriba puede contener una clave foránea hacia otra tabla, en este caso podríamos imaginar que esta columna que apunta a otra tabla se trata del tipo de cliente, esta columna puede contener 3 posibles valores, Empresa pequeña, Empresa Mediana y Empresa Grande, donde los clientes de la tabla serán de cada uno de estos 3 tipos, y por contra tendremos la otra tabla que puede ser TipoDeCliente donde encontraremos la columna Tipo de Cliente con los 3 registros (o filas) para cada valor de este atributo, esto es, un registro para Empresa Pequeña, otro registro para Empresa Mediana y un tercer registro para Empresa Grande.

Para el ejemplo que estamos comentando, la relación entre las tablas TipoDeCliente y Cliente la relación sería de 1 a muchos (1:n), dado que en la primera tabla (TipoDeCliente) tendremos un registro único para cada tipo de cliente, mientras que en la tabla Clientes podemos tener muchos clientes pero cada uno pudiendo ser Empresa Pequeña, Empresa Mediana o Empresa Grande.

Quiz

1. ¿Qué permite una base de datos?

- a) Solo almacenar datos
- b) Almacenamiento, manipulación y recuperación de datos
- c) Solo consultar datos
- d) Solo borrar datos

2. Las bases de datos SQL tienen la información estructurada en:

- a) Documentos JSON
- b) Tablas con columnas y filas
- c) Archivos clave-valor
- d) Solo texto

3. ¿Qué es una Primary Key (clave primaria)?

- a) Una clave que apunta a otra tabla
- b) Una columna donde cada valor es único en la tabla

- c) Un tipo de relación entre tablas
- d) Un índice opcional

4. La relación 'uno a muchos' (1:n) significa que:

- a) Cada registro de una tabla se relaciona con un único registro de otra
- b) Un registro de una tabla puede relacionarse con uno o más de otra tabla
- c) Muchas tablas se fusionan en una
- d) Solo puede haber una relación entre tablas

5. ¿Qué es una Foreign Key (clave foránea)?

- a) La clave principal de la tabla
- b) Una clave que apunta o referencia a otra tabla
- c) Un tipo de dato de columna
- d) La primera columna de la tabla

4. Modelo Entidad-Relación

¿Que es un modelo entidad-relación?

En esta lección vamos a explicar con un ejemplo el modelo Entidad-Relación (ER), que no es otra cosa que utilizamos los profesionales de datos para representar el diseño, o modelo, de una base de datos de manera que podemos entender fácilmente los diferentes elementos que componen una base de datos, tales como sus entidades, sus columnas (atributos) y la manera en que se relacionan las diferentes tablas que la componen.

Un modelo entidad-relación, o diagrama de una base de datos, también es muy útil a la hora de diseñar una base de datos antes de su implementación.

Ejemplo de modelo entidad-relación

En un modelo de entidad-relación las tablas no muestran sus filas, de manera que transponen las columnas para verlas en vertical de manera apilada.

Por ejemplo, una base de datos sobre ventas de un comercio puede tener 3 tablas (normalmente una base de datos de ventas tiene más tablas, pero para el ejemplo

nos sirve), Esto es, una primera tabla de Órdenes de venta, una segunda tabla de Clientes y una tercera tabla de Productos.

La tabla Ordenes tiene los siguientes campos (recuerda que en un modelo entidad-relación las columnas de una tabla están traspuestas a como las veríamos en su forma original con filas y columnas):

- IdOrden
- IdCliente
- FechaOrden
- IdProducto
- Cantidad
- Importe

Bien, como puedes ver no nos hace falta pintar la tabla con filas y columnas para entender cuál es la estructura de la tabla Ordenes, sino que hemos simplemente listado sus campos.

La segunda tabla Clientes, tendría los siguientes campos:

- IdCliente
- Nombre
- Apellidos
- Email

Y por último la tabla Productos, tendría los siguientes campos:

- IdProducto
- Categoria
- PrecioUnidad

Debemos hablar un poco sobre la lógica de negocio de esta base de datos de 3 tablas. La tabla Ordenes registra los datos de las ventas de manera que un cliente puede hacer un pedido de uno o más productos en una fecha determinada. La tabla Productos registra la información de los diferentes productos que se venden en el comercio. Y por último la tabla Clientes contendrá asimismo la información de los diferentes Clientes del negocio.

Ahora, si nos fijamos bien, te darás cuenta de que hay nombres de campos que se repiten entre las tablas, por ejemplo IdProducto en la tabla Productos e IdProducto en la tabla Ordenes. Te preguntarás, ¿por qué es esto así?, bien, esta es la forma más fácil de identificar como relacionar dos tablas. Cuando los nombres de los campos entre dos tablas se repiten esto te da pistas para saber cómo se relacionan estas, en este las tablas Ordenes y Producto se relacionan por el campo IdProducto. La tabla Productos contiene un registro único por cada producto, y la tabla Ordenes

contiene la información de las ventas en las que los clientes pueden hacer una o más compras de uno o varios productos en una o varias fechas.

Por lo tanto las relaciones entre las tablas se dan de la siguiente manera....

La relación entre la tabla Clientes y Ordenes será uno a muchos (1:n), y la relación entre las tablas Producto y Ordenes será también uno a muchos (1:n). Las tablas Productos y Clientes no se relacionan entre sí directamente sino a través de la tabla Ordenes (mas abajo podra ver la representación de este modelo entidad-relacion).

Recuerda que las tablas pueden tener una clave primaria (PK), y una o más claves foráneas (FK). Siguiendo el ejemplo de nuestra base de datos de ventas, la tabla Productos tendrá una clave primaria que se corresponde con el campo IdProducto, que será el identificador único de cada producto en nuestro sistema. Asimismo la tabla Clientes tendrá una clave primaria dada por el campo IdCliente para identificar de manera unívoca a cada cliente en nuestro sistema. ¿Y qué ocurre con la tabla Ordenes? Pues que esta contendrá las foreign keys o claves foráneas que apuntan a las tablas Productos y Clientes dadas por IdProducto e IdCliente respectivamente.

Si ilustramos todo lo que hemos comentado anteriormente, nuestro modelo entidad-relación de nuestra base de datos Ventas, esta se podría representar de la siguiente manera:



Quiz

1. ¿Para qué se usa el modelo Entidad-Relación (ER)?

- a) Para ejecutar consultas SQL
- b) Para representar el diseño de una base de datos y sus relaciones
- c) Para almacenar datos
- d) Para borrar tablas

2. En un diagrama ER, ¿cómo se suelen mostrar las columnas de una tabla?

- a) Solo los nombres de las tablas
- b) En vertical, apiladas
- c) Como filas de datos
- d) No se muestran

3. ¿Cómo identificamos por qué campos se relacionan dos tablas?

- a) Por el orden de las tablas
- b) Por los nombres de campos que se repiten entre tablas
- c) Por el tamaño de la tabla
- d) Por el color en el diagrama

4. La tabla Ordenes en el ejemplo tiene claves foráneas (FK) que apuntan a:

- a) Solo a Clientes
- b) Solo a Productos
- c) A Clientes y a Productos (IdCliente, IdProducto)
- d) A ninguna tabla

5. La relación entre la tabla Clientes y la tabla Ordenes en el ejemplo es:

- a) Muchos a muchos (n:n)
- b) Uno a uno (1:1)
- c) Uno a muchos (1:n)
- d) No están relacionadas