

# SQL

—

## Introducción



# Pre examen – SQL (teoría)

## Índice

1. Introducción
2. Tipos de Bases de Datos
  - 2.1. Relacional
  - 2.2. No Relacional
  - 2.3. Relacional vs No Relacional
3. Sublenguajes SQL
4. Motor de base de datos
5. Tipos de datos básicos: string, numéricos, fechas



# 1. Introducción

## Introducción a SQL

- **Structured Query Language**
- Lenguaje de **programación**
- **Consultas** en una base de datos **relacional**.
- **Sentencias simples** y declarativas.

Aprendiendo SQL también aprenderás **conceptos aplicables** a cualquier otro **sistema de almacenaje de datos**.



# 2. Tipos de bases de datos

Hay dos tipos fundamentales de sistemas de gestión de datos:

- las bases de **datos relacionales**
- las bases de **datos no relacionales** o NoSQL (Not Only SQL).

Cada sistema tiene sus propias características, ventajas e inconvenientes que los hacen adecuados para diferentes usos y aplicaciones.



# 2.1. Tipos de bases de datos – Relacional

Una base de datos SQL (**S**tructured **Q**uery **L**anguage) es un ejemplo de base de datos relacional.

La información se organiza en una estructura de tablas con filas y columnas.

Se utiliza un lenguaje de consulta estructurado (SQL) para acceder y gestionar los datos.

Son ideales cuando se necesita una estructura de datos clara y relaciones predefinidas entre elementos.

Siguen el principio ACID (**A**tomicity, **C**onsistency, **I**solation, **D**urability).

Algunos ejemplos populares de sistemas de gestión de bases de datos relacionales incluyen Oracle, MySQL, Microsoft SQL Server, PostgreSQL, DB2.



# 2.1. Tipos de bases de datos – Relacional

## **Ventajas** de las Bases de datos Relacionales:

- Estructura sólida y coherente
- Integridad de datos garantizada
- Amplio soporte y herramientas disponibles

## **Desventajas** de las Bases de Datos Relacionales:

- Escalabilidad limitada en comparación con NoSQL
- Requieren una estructura de datos definida previamente



# 2.1. Tipos de bases de datos – Relacional

Los datos se relacionan entre sí conectando las diferentes tablas.

Para establecer las relaciones existe la **Primary key** y la **Foreign key**.



## Primary key

- Es el ID **único** de cada fila en una tabla.
- No puede tener valores **NULL**.
- Cada tabla sólo puede contener **una** PK.
  - Puede ser **simple** o **compuesta**.

## Foreign key

- Facilita las conexiones entre las tablas.
- Es una columna en una tabla que en otra tabla es la PK.
- La tabla con la FK se conoce como tabla hija.
- La tabla en la que esa FK es la PK se denomina tabla padre.



# 2.1. Tipos de bases de datos – Relacional

## Relaciones



- **One-to-one:**
  - a. una entidad tiene una relación exclusiva con otra entidad y viceversa.
  - b. Número de la seguridad social – DNI.
- **One-to-many:**
  - a. una entidad tiene una relación exclusiva con otra entidad, pero no a la inversa.
  - b. Jugadores – Equipo.
- **Many-to-many:**
  - a. una entidad tiene relación con varias entidades más y también a la inversa.
  - b. Películas – Actores.
  - c. Va a necesitar una tabla intermedia para poderse relacionar.





# 2.1. Tipos de bases de datos – Relacional

Las tablas, vistas e índices de la base de datos están separadas de la estructura de almacenamiento físico.

Si se cambia el nombre de un archivo de la base de datos su estructura y nombre de sus tablas no se ven alterados.

El modelo relacional es sencillo pero muy potente.

Ejemplos de uso:

- rastrear inventarios
- procesar transacciones de comercio electrónico
- administrar cantidades enormes de información de clientes

Las relaciones se gestionan de forma segura conforme a unas normas y de manera uniforme.

Estas normas son las **reglas de integridad**.

- **De dominio**
- **Referencial**
- **De las entidades**



# 2.1. Tipos de bases de datos – Relacional

1. **Uniformidad de los datos:** ideal para mantener la uniformidad en todas las aplicaciones e instancias de la base de datos.
2. **Compromiso y atomicidad:** compromiso, conjunto de reglas y políticas sobre un cambio permanente en la base de datos. Ejemplo inventario.
3. **Procedimientos almacenados:** se pueden definir en la propia base de datos. Son bloques de código a los que se puede acceder con una simple llamada.
4. **Bloqueo y concurrencia:** pueden existir conflictos cuando varios usuarios o aplicaciones intentan cambiar los mismos datos a la vez. El bloqueo evita que usuarios/aplicaciones accedan a los datos mientras se modifican.
5. **Consideraciones al escoger una base de datos relacional:** ¿Necesitamos un gran nivel de precisión en los datos? ¿Habrá concurrencia? ¿Cuántos usuarios o aplicaciones van a necesitar acceder a la vez? ¿Qué compromisos de nivel de servicio tiene el proveedor?...



## 2.2. Tipos de bases de datos – No Relacional

### Base de datos NO Relacionales

Almacenamiento más flexible.

No requieren relaciones.

Grandes volúmenes de datos.

Altamente escalables.

Almacenamiento de datos:

- documentos
- pdf, xml, json, csv...
- columnas o grafos

Ejemplos:

- Amazon Dynamo DB
- Redis
- MongoDB



Ideales cuando se trabaja con datos no estructurados, o semiestructurados y se necesita escalabilidad.



## 2.2. Tipos de bases de datos – No Relacional

### Base de datos No Relacionales



**Ventajas** de las Bases de datos NoSQL:

- Escalabilidad horizontal sencilla
- Flexibilidad en la estructura de datos
- Manejo de grandes volúmenes de datos

**Desventajas** de las Bases de datos NoSQL:

- Menos consistencia en comparación con las bases de datos relacionales
- Falta de estandarización y soporte SQL uniforme



## 2.3. Relacional vs No Relacional

### Cuándo utilizar SQL o NoSQL

#### SQL

#### NoSQL

Los datos no crecen rápidamente	Esperas un crecimiento rápido de los datos
La estructura de datos es fija y predecible	Necesitas una escalabilidad horizontal
La integridad de los datos es fundamental	Los datos no tienen estructura definida o cambian con frecuencia



# 3. Sublenguajes SQL

## Tipos de sublenguajes

- **Data Definition Language (DDL):** se refiere a la parte del lenguaje que se utiliza para definir las estructuras para el almacenamiento de datos.
  - A. CREATE: crea nuevas tablas, campos, índices.
  - B. ALTER: modifica tablas existentes agregando campos o cambiando su definición.
  - C. DROP: elimina tablas e índices.
  - D. TRUNCATE: elimina todos los registros de una tabla.
  - E. RENAME: para renombrar objetos.



<https://dev.mysql.com/doc/refman/8.0/en/sql-data-definition-statements.html>



# 3. Sublenguajes SQL

## Tipos de sublenguajes

- **Data Manipulation Language (DML):** se pueden manipular los datos efectuando consultas para recuperarlos, modificarlos, borrarlos, etc. Las operaciones fundamentales de manipulación de datos son cuatro y consecuentemente cuatro los comandos DML que se necesitan para poder llevarlas a cabo.
  - A. **SELECT:** la sentencia por excelencia, se utiliza para ejecutar consulta de recuperación de datos, recibe el nombre de query.
  - B. **INSERT:** su finalidad es insertar filas nuevas en una tabla.
  - C. **UPDATE:** permite cambiar la información alojada en una o más filas.
  - D. **DELETE:** elimina una o más filas.



# 3. Sublenguajes SQL

## Tipos de sublenguajes

- **Data Control Language (DCL):** se compone de sentencias específicas para garantizar la seguridad de acceso a los datos, otorgando o denegando los permisos necesarios para operar sobre los elementos de la base de datos.
  - A. GRANT: otorga permisos.
  - B. REVOKE: deniega permisos.
- **Transaction Control Language (TCL):** permite administrar diferentes transacciones que ocurren dentro de la base de datos. Se comienza con **START TRANSACTION** y se separa cada instrucción con:
  - A. COMMIT: guarda los cambios y no se pueden deshacer.
  - B. ROLLBACK: revierte los cambios hasta el estado del último commit.

<https://dev.mysql.com/doc/refman/8.0/en/sql-transactional-statements.html>

<https://dev.mysql.com/doc/refman/8.0/en/sql-server-administration-statements.html>





# 4. Motor base de datos

## Relational Database Management System

RDBMS = Relational Database Management System

Es el software que se utiliza para almacenar, administrar, consultar y recuperar datos guardados en una base de datos relacional.

El **objetivo** del motor SQL es crear, leer, actualizar y eliminar datos.

Eso son las operaciones **CRUD** (Create, Read, Update, Delete).

Hay varios motores para gestionar bases de datos SQL:

- **MySQL**: el que vamos a utilizar
- Microsoft SQL Server
- MariaDB
- SQLite
- Oracle
- PostgreSQL



# 4. Motor base de datos: WORKBENCH

## MySQL WORKBENCH

Es una herramienta de gestión y diseño de bases de datos.



### Características:

- Conexión y gestión de instancias de bases de datos.
- Soporte para personalizar mediante complementos (add-ons)
- Editor de SQL con **resaltado de sintaxis** y análisis de declaraciones.
- Capacidad para trabajar con múltiples conjuntos de resultados editables.
- Modelado visual de datos con **diagramas entidad-relación** y funciones de arrastrar y soltar.
- Ingeniería inversa y directa a partir de guiones SQL y bases de datos en funcionamiento.
- Administración de bases de datos, incluyendo inicio y detención de instancias, configuración, gestión de cuentas, exploración de variables de instancia y ficheros de registros, así como exportación e importación de datos a gran escala.



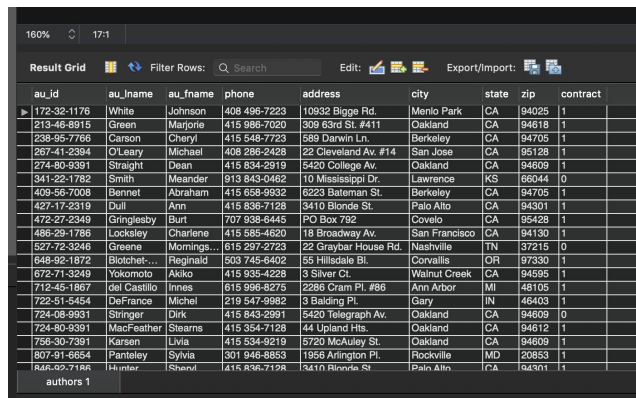
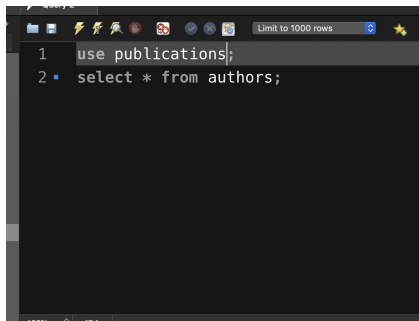
# 4. Motor base de datos: WORKBENCH

## Realizar una consulta

Es tan sencillo como **escribir el código** en el cuadro de texto y ejecutarlo con el botón del rayo o la combinación de teclas **ctrl + enter** (cmd + enter para mac).

Así nos dará la respuesta, en la parte inferior de la pantalla, en forma de tabla.

No te preocupes, si tu código no funciona te dará alguna pista del motivo.



au_id	au_name	au_fname	phone	address	city	state	zip	contract
172-32-1176	White	Johnson	408 496-7223	10932 Bigge Rd.	Menlo Park	CA	94025	1
213-46-8915	Green	Marjorie	415 986-7020	309 63rd St. #411	Oakland	CA	94618	1
238-95-7766	Carson	Cheryl	415 548-7723	589 Darwin Ln.	Berkeley	CA	94705	1
267-21-2394	O'Leary	Michael	408 236-8428	22 Cleveland Av. #14	San Jose	CA	95128	1
274-80-9391	Straght	Dean	415 834-2319	5420 College Av.	Oakland	CA	94609	1
341-22-1782	Smith	Meander	913 843-0462	10 Mississippi Dr.	Lawrence	KS	66044	0
409-56-7008	Bennet	Abraham	415 658-9932	6223 Bateman St.	Berkeley	CA	94705	1
427-17-2319	Dull	Ann	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	1
472-27-2349	Gringlesby	Burt	707 938-6445	PO Box 792	Covelo	CA	95428	1
486-29-1786	Locksley	Charlene	415 585-4620	18 Broadway Av.	San Francisco	CA	94130	1
527-72-3246	Greene	Mornings...	615 227-2723	22 Graybar House Rd.	Nashville	TN	37215	0
648-92-1892	Botchek	Reginald	503 745-6402	55 Hillside St.	Cornwall	OR	97330	1
672-71-3249	Yokomoto	Akiko	415 935-4228	3 Silver Ct.	Walnut Creek	CA	94598	1
712-45-1867	del Castillo	Innes	615 996-8275	2286 Cram Pl. #86	Ann Arbor	MI	48105	1
722-51-5454	DeFrance	Michel	219 547-9982	3 Balding Pl.	Gary	IN	46403	1
724-08-9931	Stringer	Dirk	415 843-2991	5420 Telegraph Av.	Oakland	CA	94609	0
724-80-9391	MacFeather	Stearns	415 354-7128	44 Upland Hts.	Oakland	CA	94612	1
755-30-7281	Karen	Livia	415 534-9219	5720 McCutley St.	Oakland	CA	94609	1
807-91-6854	Pantelley	Sylvia	301 946-8853	1566 Arlington Pl.	Rockville	MD	20853	1
848-02-7188	McIntire	Sheryl	415 836-7128	3410 Blonde St.	Palo Alto	CA	94301	1



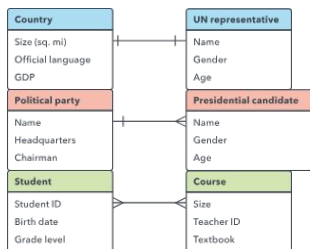
# 4. Motor base de datos: WORKBENCH

## Diagrama ERD

Muestra las entidades de la base de datos y cómo se relacionan.

Para facilitar su interpretación hay varios símbolos que facilitan visualmente la comprensión.

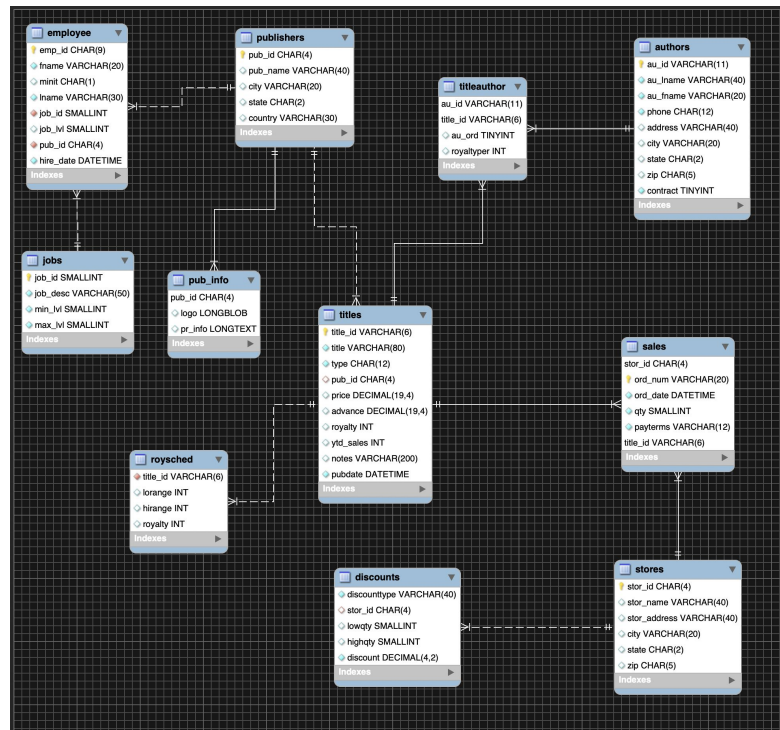
- Llaves para las claves (primarias y foráneas)
- Diamantes para los atributos
- Colores
- Flechas con extremos distintos para visualizar el tipo de relación entre las tablas



one-to-one

one-to-many

many-to-many



# 5. Tipos de datos básicos

## 1. Tipos de Datos Numéricos:

- ENTEROS: Incluyen TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT.
- DECIMALES: Incluyen DECIMAL y NUMERIC, que permiten almacenar números con decimales de precisión fija.

## 2. Tipos de Datos de Texto:

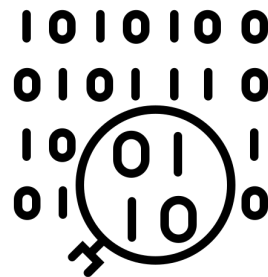
- CHAR: Almacena cadenas de texto de longitud fija.
- VARCHAR: Almacena cadenas de texto de longitud variable.
- TEXT: Utilizado para textos de longitud variable más larga.
- ENUM: Almacena uno de los valores predefinidos en una lista.

## 3. Tipos de Datos de Fecha y Hora:

- DATE: Almacena fechas (año, mes, día).
- TIME: Almacena horas, minutos, segundos.
- DATETIME: Combina fecha y hora.
- TIMESTAMP: Almacena una marca de tiempo.

## 4. Tipos de Datos Binarios:

- BINARY: Almacena datos binarios de longitud fija.
- VARBINARY: Almacena datos binarios de longitud variable.
- BLOB: Utilizado para datos binarios más largos, como imágenes o archivos.



# 5. Tipos de datos básicos

## 5. Tipos de Datos Booleanos:

- a. BOOLEAN o BOOL: Almacena valores verdaderos o falsos.

## 6. Tipos de Datos de Enumeración:

- a. ENUM: Almacena uno de los valores predefinidos en una lista.

## 7. Tipos de Datos Geoespaciales:

- a. GEOMETRY: Utilizado para datos geoespaciales.
- b. POINT: Almacena coordenadas de un punto.
- c. POLYGON: Almacena información de polígonos.

## 8. Tipos de Datos JSON:

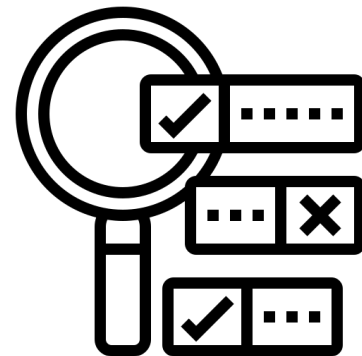
- a. JSON: Almacena datos en formato JSON.

## 9. Tipos de Datos de Identificación:

- a. AUTO\_INCREMENT: Usado para generar automáticamente valores únicos al insertar registros en una tabla.

## 10. Tipos de Datos de Texto Largos:

- a. TINYTEXT, MEDIUMTEXT, LONGTEXT: Almacenan texto de longitud variable, con capacidades crecientes.



[https://www.w3schools.com/mysql/mysql\\_datatypes.asp](https://www.w3schools.com/mysql/mysql_datatypes.asp)

<https://dev.mysql.com/doc/refman/8.0/en/storage-requirements.html>



# 5.1. Tipos de datos: STRING

Nombre	Almacenamiento	Rango (ancho de columna)
CHAR, CHARACTER o NCHAR	Longitud de la cadena, incluidos espacios en blanco anteriores o posteriores (si corresponde)	4 096 bytes
VARCHAR, CHARACTER VARYING o NVARCHAR	4 bytes + bytes totales por caracteres, donde cada carácter puede tener entre 1 y 4 bytes.	65 535 bytes (64K -1)
BPCHAR	Convertido a CHAR(256) de longitud fija.	256 bytes
TEXT	Convertido a VARCHAR(256).	260 bytes

Los datos de caracteres pueden almacenarse como cadenas de longitud fija o variable; la diferencia es que las cadenas de longitud fija se rellenan con espacios a la derecha y siempre consumen el mismo número de bytes, y las cadenas de longitud variable no se rellenan con espacios a la derecha y no siempre consumen el mismo número de bytes.

Al definir una columna de caracteres, debe especificar el tamaño máximo de cualquier cadena que se vaya a almacenar en la columna.

Por ejemplo, si desea almacenar cadenas de hasta 20 caracteres de longitud, puede utilizar cualquiera de las siguientes definiciones:

- `char(20) -> /* de longitud fija */`
- `varchar(20) -> /* longitud variable */`

La longitud máxima de las columnas char es actualmente de 255 bytes, mientras que las columnas varchar pueden tener hasta 65.535 bytes. Si necesitas almacenar cadenas más largas (como correos electrónicos, documentos XML, etc.), entonces querrás utilizar uno de los tipos de texto (mediumtext y longtext).

En general, se debe utilizar el tipo char cuando todas las cadenas que se van a almacenar en la columna tienen la misma longitud, como las abreviaturas del estado, y el tipo varchar cuando las cadenas que se van a almacenar en la columna tienen longitudes diferentes. Tanto char como varchar se utilizan de forma similar en los principales servidores de bases de datos.



## 5.2. Tipos de datos: NUMÉRICOS

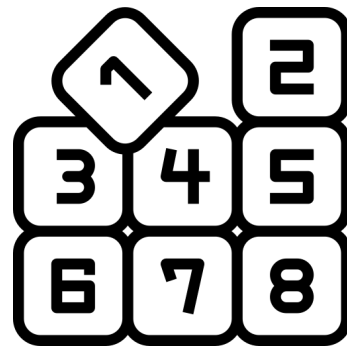
Type	Signed range	Unsigned range
<code>tinyint</code>	-128 to 127	0 to 255
<code>smallint</code>	-32,768 to 32,767	0 to 65,535
<code>mediumint</code>	-8,388,608 to 8,388,607	0 to 16,777,215
<code>int</code>	-2,147,483,648 to 2,147,483,647	0 to 4,294,967,295

<code>bigint</code>	$-2^{63}$ to $2^{63} - 1$	0 to $2^{64} - 1$
---------------------	---------------------------	-------------------

Type	Numeric range
<code>float( p , s )</code>	$-3.402823466\text{E}+38$ to $-1.175494351\text{E}-38$ and $1.175494351\text{E}-38$ to $3.402823466\text{E}+38$
<code>double( p , s )</code>	$-1.7976931348623157\text{E}+308$ to $-2.2250738585072014\text{E}-308$ and $2.2250738585072014\text{E}-308$ to $1.7976931348623157\text{E}+308$

Puede parecer razonable tener un único tipo de datos numéricos llamado “numérico”, pero en realidad hay diferentes tipos que reflejan las diversas formas en que se utilizan los números.

En función del rango van a necesitar mayor o menor cantidad de espacio de memoria. Al definirlos ese es un factor a tener en cuenta para no tener un espacio reservado que finalmente no va a ser utilizado.





## 5.3. Tipos de datos: FECHAS

Type	Default format	Allowable values
date	YYYY-MM-DD	1000-01-01 to 9999-12-31
datetime	YYYY-MM-DD HH:MI:SS	1000-01-01 00:00:00.000000 to 9999-12-31 23:59:59.999999
timestamp	YYYY-MM-DD HH:MI:SS	1970-01-01 00:00:00.000000 to 2038-01-18 22:14:07.999999
year	YYYY	1901 to 2155
time	HHH:MI:SS	-838:59:59.000000

Junto con las cadenas y los números, es 1000% seguro que trabajarás con información sobre fechas y/u horas. Este tipo de datos se denomina **temporal**, y algunos ejemplos de datos temporales en una base de datos son:

- La fecha futura en la que se espera que ocurra un evento concreto, como el envío del pedido de un cliente.
- La fecha en que se envió el pedido de un cliente.
- La fecha y hora en que un usuario modificó una fila concreta de una tabla.
- La fecha de nacimiento de un empleado.
- El año correspondiente a una fila de una tabla de hechos yearly\_sales en un almacén de datos.
- El tiempo transcurrido para completar un trabajo específico una línea de montaje de automóviles.

