



Introducción a Bases de Datos y SQL



Octavio Robleto



octavio.robleto@gmail.com

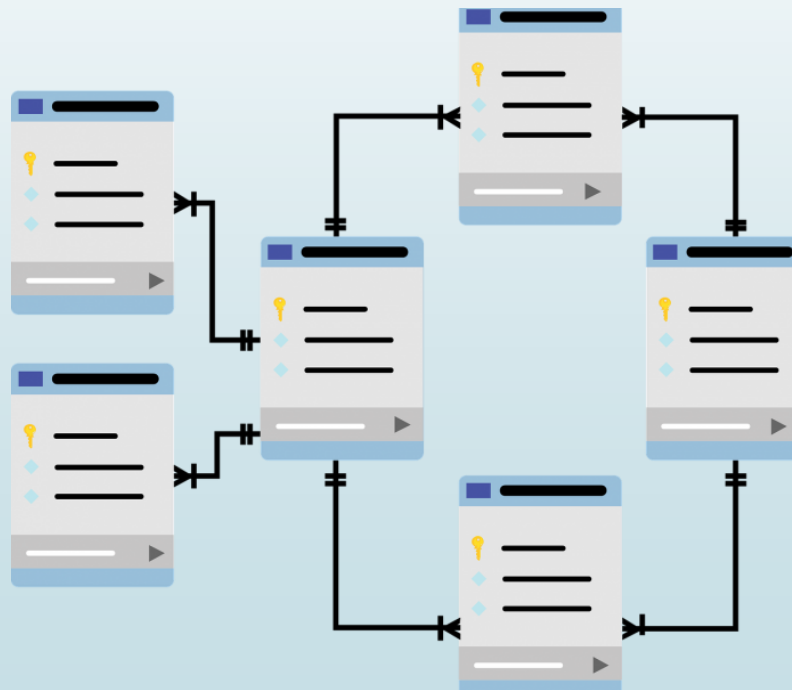


<http://octaviorobleto.com/>



Base de Datos Relacional

- Conjunto de datos relacionados entre sí y que tienen un significado implícito.
- **Modelo Relacional:** consistente en el almacenamiento de datos en tablas compuestas por filas, o tuplas, y columnas o campos.



Motor de Base de Datos

- Un motor de Bases de Datos es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos , además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.
- Los usuarios pueden acceder a la información usando herramientas específicas de consulta y / o de generación de informes, o bien mediante aplicaciones al efecto.
- En base a esto, podemos definir que los motores de base de datos sirven para **definir** , **construir** y **manipular** una base de datos .



Que podemos hacer con esto?



- **1- Definir una base de datos:** consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán .
- **2- Construir una base de datos:** es el proceso de almacenar los datos sobre algún medio de almacenamiento .
- **3- Manipular una base de datos:** incluye funciones como consulta , actualización, etc. de bases de datos.

			
CREATE	READ	UPDATE	DELETE
C	R	U	D

			
CREATE	DELETE	UPDATE	READ
A	B	M	C

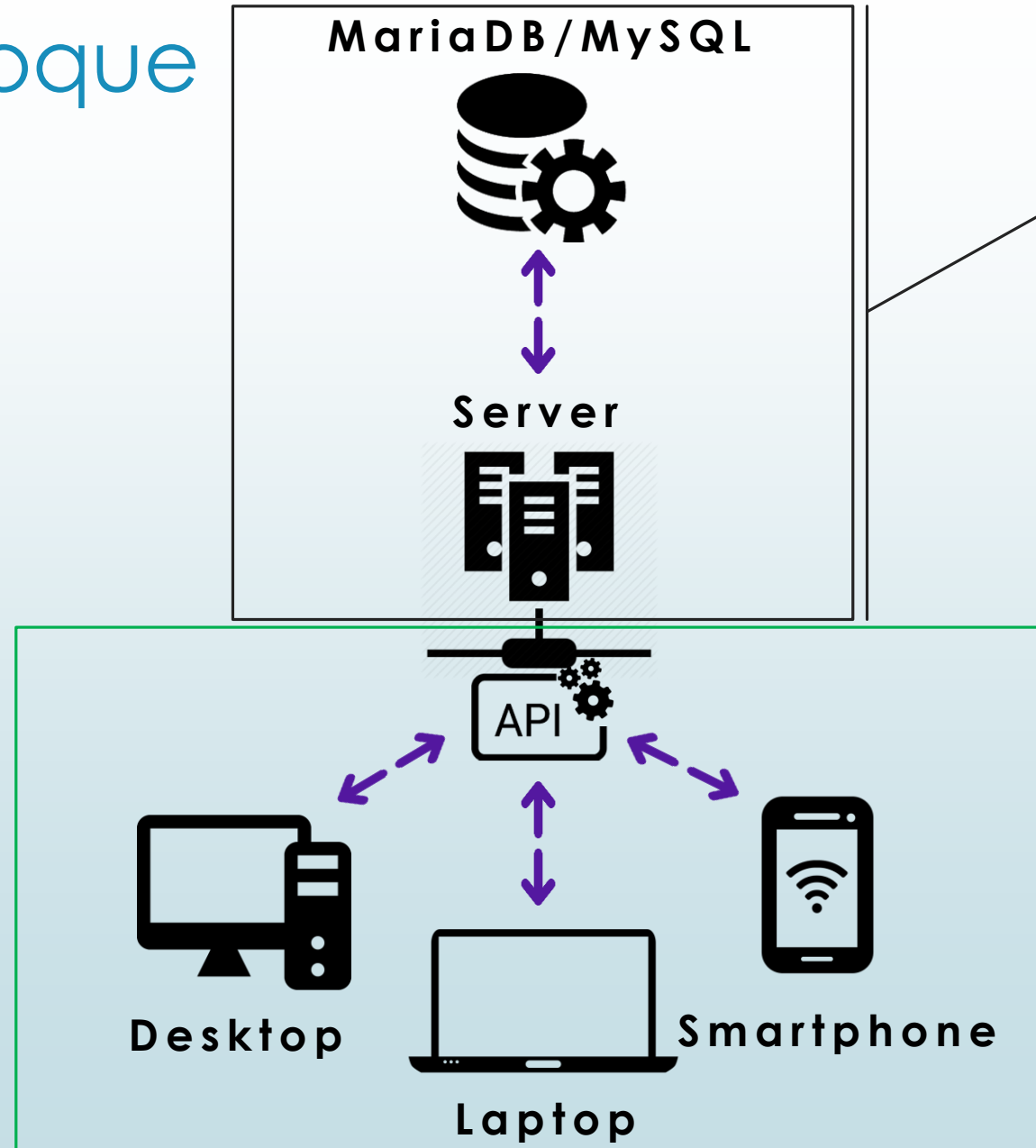


Control de Concurrency

- Se refiere al hecho de que los DBMS (Sistemas de Administración de Bases de Datos) permiten que muchas transacciones accedan a una misma base de datos a la vez. Esto para asegurar que las transacciones concurrentes no interfieran entre sí



Enfoque



Una aplicación servidor es el elemento de la comunicación que responde a las peticiones de los clientes, proporcionando el servicio requerido

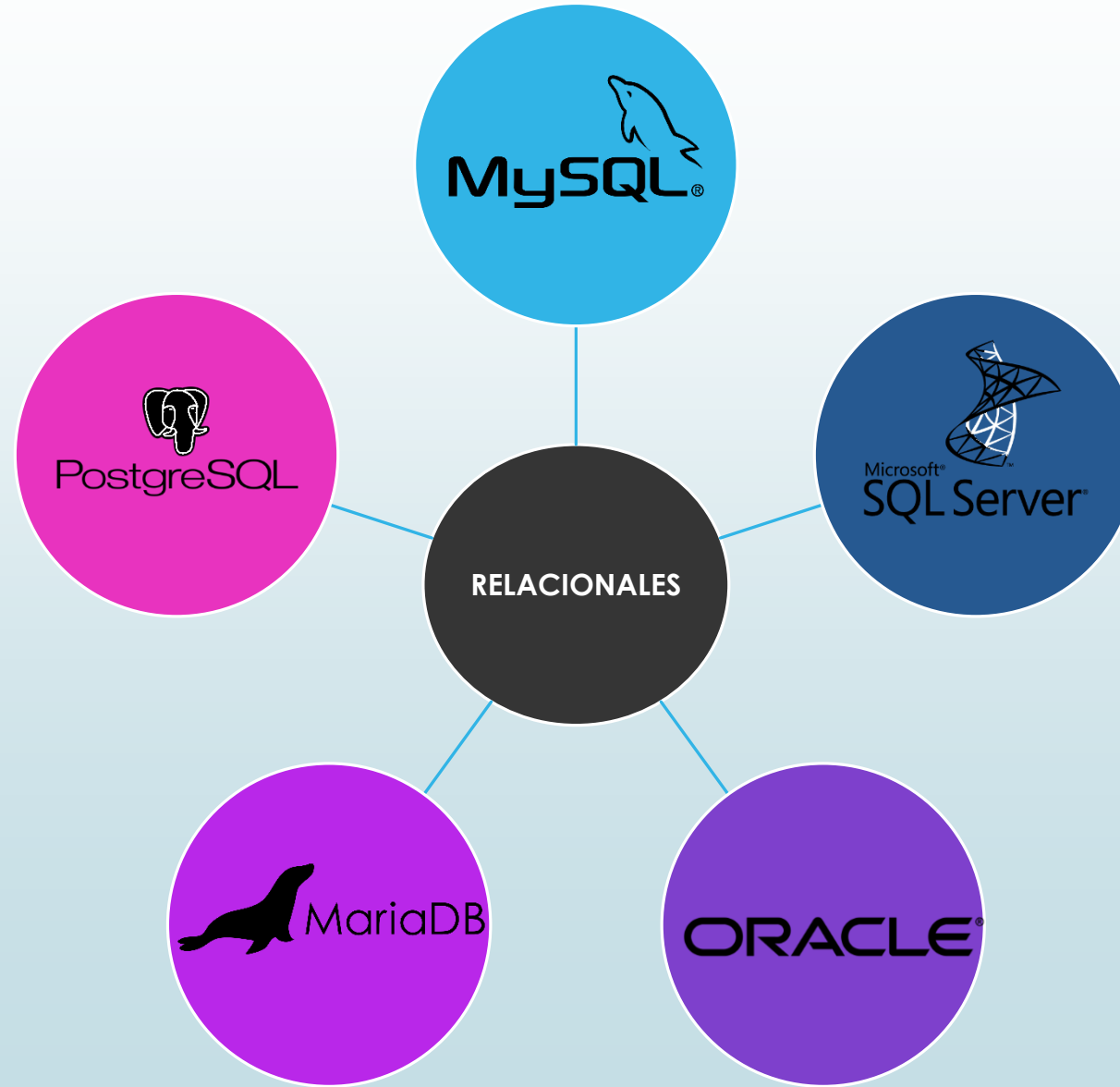
- Una aplicación cliente es el elemento de la comunicación que pide o solicita un servicio de red.

Cliente - Servidor

- **Escalabilidad** : Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden agregar nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento** : Al estar distribuidas las funciones y responsabilidades entre varias computadoras independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por el cambio. Hoy día es frecuente también tener servidores en la nube
- **Seguridad**: Los DBMS permiten administrar permisos a TODO. Permisos a nivel de servidor, a nivel de base de datos, seguridad en tablas, permitir o no lectura de datos, escritura de datos, ejecución de procedimientos



Motores de Base de Datos



MariaDB y MySQL

La primera versión de **MariaDB** fue una bifurcación de **MySQL** con características adicionales, mejoras de rendimiento y correcciones de errores. Sin embargo, aunque ambos implementan el protocolo SQL, MariaDB y MySQL han evolucionado de manera diferente, convirtiéndose en bases de datos independientes.

MariaDB se ha convertido en una solución de base de datos completa con SQL distribuido, almacenamiento en columnas y tablas temporales, mientras que MySQL 8.0 fue una versión significativa con su diccionario de datos transaccionales y soporte JSON mejorado.

Ambos motores de base de datos son altamente robustos y óptimos y cada uno tiene una fortaleza específica pero para este tema cualquiera de los dos SMDB nos brindaran las herramientas adecuadas para persistir y recuperar la información de nuestros objetos.



Que necesitamos:



<https://dev.mysql.com/downloads/installer/>



<https://dev.mysql.com/downloads/workbench/>



<https://downloads.mariadb.org/>



<https://dbeaver.io/download/>

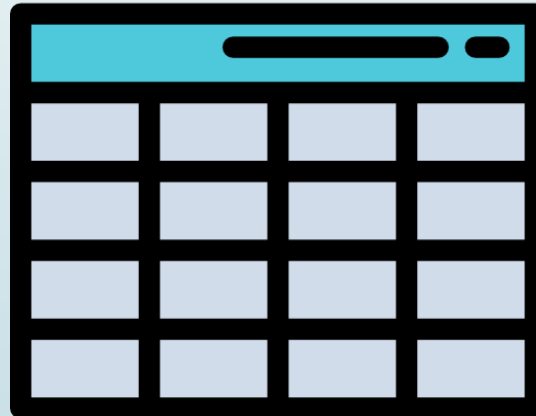


<https://www.apachefriends.org/es/download.html>



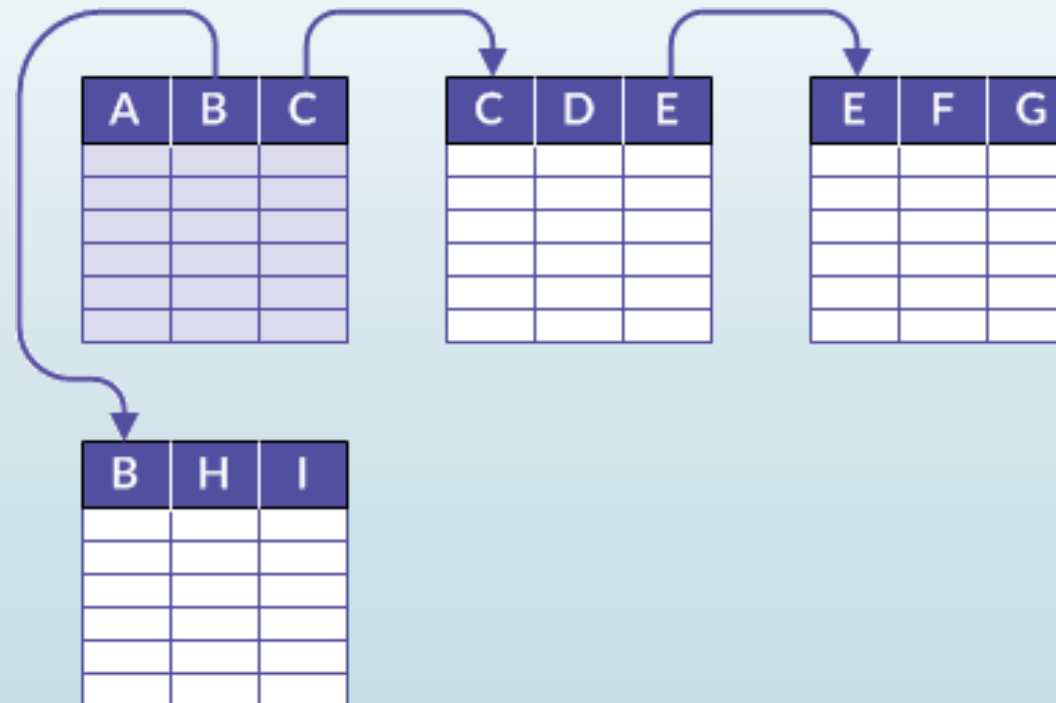
Tablas

- Las tablas son objetos de base de datos que contienen todos sus datos. En las tablas, los datos se organizan con arreglo a un formato de filas y columnas, similar al de una hoja de cálculo. Cada fila representa un registro que debería ser único y cada columna un campo dentro del registro.



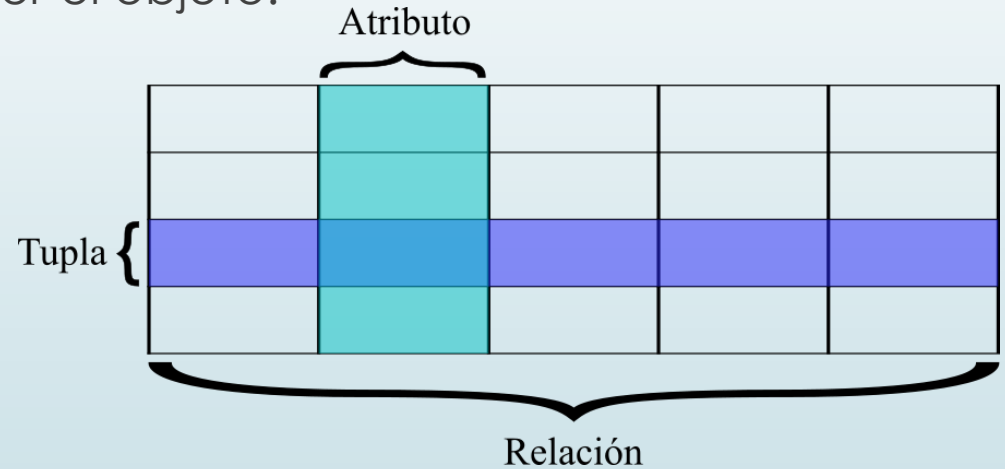
Relaciones

- Enlace entre dos tablas donde la clave principal de una tabla se asocia con la clave externa de otra tabla utilizando las relaciones de la base de datos

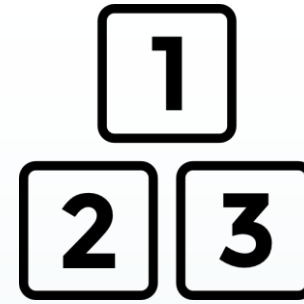


Tipos de Datos

- En SQL, cada columna, variable local, expresión y parámetro tiene un tipo de datos relacionado. Un tipo de datos es un atributo que especifica el tipo de datos que puede contener el objeto:
 - Numéricos exactos
 - Numéricos aproximados
 - Fecha y hora
 - Cadenas de caracteres
 - Cadenas de caracteres Unicode
 - Cadenas binarias
 - Otros tipos de datos



Numéricos Exactos

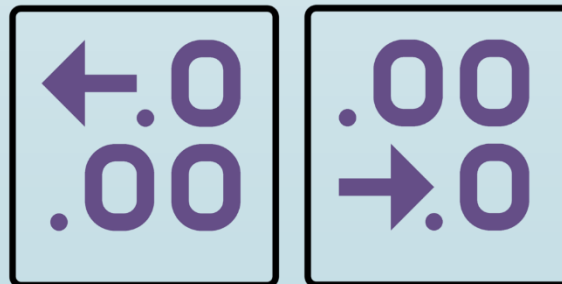


Tipo de Dato	Descripción	Longitud
BIT[(longitud)]	Longitud entre 1 y 64	1
TINYINT	Entero entre -128 a 127	1
BOOL, BOOLEAN	Rango entre 0 y 1. Se usa para valores de tipo falso (0) y verdadero (1)	1
SMALLINT[(longitud)] [UNSIGNED] [ZEROFILL]	Tipo entero. Rango entre -32,768 y 32,767 con signo. Longitud entre 0 y 65,535 sin signo	2
MEDIUMINT[(longitud)]	Tipo entero. Rango entre -8,388,608 y 8,388,607 con signo. Longitud entre 0 y 4,294,967,295 sin signo	3
INT[(longitud)] [UNSIGNED] [ZEROFILL]	Tipo entero. Rango entre -2,147,483,648 y 2,147,483,647 con signo. Longitud entre 0 y 4,294,967,295 sin signo	4
INTEGER[(longitud)] [UNSIGNED] [ZEROFILL]	Lo mismo que INT	4
BIGINT[(longitud)] [UNSIGNED] [ZEROFILL]	Tipo entero. Rango entre -9,223,372,036,854,775,808 y 9,223,372,036,854,775,807 con signo. Longitud entre 0 y 18,446,744,073,709,551,615 sin signo	8



Numéricos de Aproximados

Tipo de Dato	Descripción	Longitud
DECIMAL[(longitud,decimales)] [UNSIGNED] [ZEROFILL]	Número con coma flotante. Longitud de hasta 65 dígitos en la parte entera y hasta 30 dígitos en la parte decimal (por defecto 0)	Variable
DOUBLE[(longitud,decimales)] [UNSIGNED] [ZEROFILL]	Número con coma flotante (doble precisión). Longitud de - 1.7976931348623157 X 10308 hasta -2.2250738585072014 X 10-308 Ó De 2.2250738585072014 X 10-308 hasta 1.7976931348623157 X 10308	8



Fecha y hora

Tipo de Dato	Descripción	Longitud
DATE	Fecha con formato "YYYY-MM-DD"	3
DATETIME[(precisión)]	Fecha y hora con formato "YYYY-MM-DD HH:MM:SS.SSSSSS". El parámetro precisión define la precisión en la parte decimal de los segundos	5
TIMESTAMP[(precisión)]	El parámetro precisión representa la cantidad de dígitos que se usarán para representar una fecha y hora desde 1970 hasta 2037	4
TIME[(precisión)]	Hora, con formato "HH:MM:SS [.Precisión valores de S]"	3
YEAR[(2 4)]	Representa el año con formto de cuatro dígitos "YYYY" o dos dígitos "YY"	1



Cadenas de Textos



Tipo de Dato	Descripción	Longitud
CHAR[(longitud)]	Admite caracteres alfanuméricos. Longitud de 1 – 255 caracteres (por defecto 1). Si se asigna una cadena de longitud menor, esta se completará con espacios.	Longitud = bytes
VARCHAR[(longitud)]	Similar al CHAR salvo que es necesario especificar la longitud y no se autorrellena con espacios.	Longitud + 1 bytes
TINYTEXT,	TINYTEXT: Hasta 255 caracteres	Hasta 255 bytes (1 byte por cada caracter).
TEXT,	TEXT: Hasta 65,535 caracteres	Hasta 64 KB
MEDIUMTEXT,	MEDIUMTEXT: Hasta 16,777,215 caracteres	Hasta 16 MB
LONGTEXT	LONGTEXT: Hasta 4,294,967,295 caracteres	Hasta 4

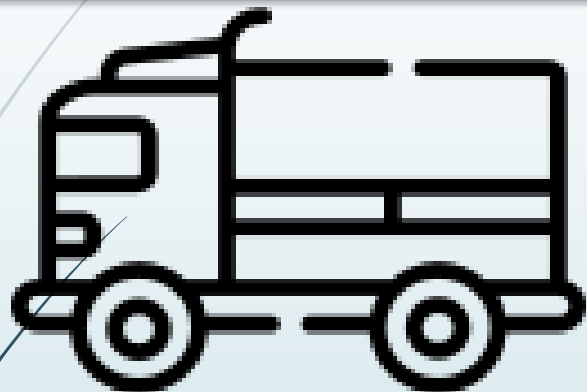


Otros Tipos de Datos



Tipo de Dato	Descripción	Longitud
BLOB	Objeto binario que puede almacenar cualquier tipo de información. Admite hasta 65,535 caracteres	Hasta 64 KB
TINYBLOB,	Hasta 255 caracteres	Hasta 255 bytes (1 byte por cada caracter).
MEDIUMBLOB,	Hasta 16,777,215 caracteres	Hasta 16 MB
LOB	Hasta 4,294,967,295 caracteres	Hasta 4 GB







Información

Tipo Documento	Numero de Documento	Nombre	Dirección	Correo	Fecha de Nacimiento	Carrera	Curso
DNI	0001	Octavio Robleto	Capital Federal - Villa Devoto	octavio.robleto@gmail.com	15/3/1983	Java	Java NP, Base de Datos, Spring
PAS	0003	Mariana Bracho	Provincia de Buenos Aires - Pilar	mariana.bracho@gmail.com	6/6/1989	.NET	C#, Base de Datos
CI	0002	Sabrina Castagno	Provincia de Buenos Aires - Tigre	sabrina.castagno@gmail.com	5/4/1987	Front End	Java Script, Base de Datos



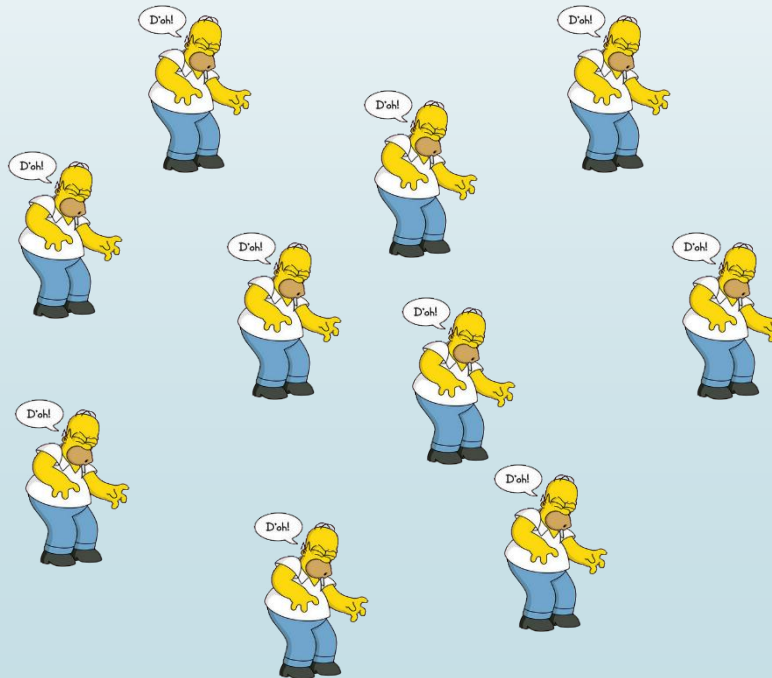
Normalización

- El proceso de normalización de una base de datos relacional consiste en aplicar una serie de reglas para evitar a futuro realizar queries, o consultas innecesariamente complejas. En otras palabras están enfocadas en eliminar redundancias e inconsistencias de dependencia en el diseño de las tablas



Para que?

- Evitar la redundancia de datos
- Proteger la integridad de los datos
- Evitar problemas de actualización de los datos en las tablas



Tipo Documento	Numero de Documento	Nombre	Dirección	Correo	Fecha de Nacimiento	Carrera	Curso
DNI	0001	Octavio Robleto	Capital Federal - Villa Devoto	octavio.robleto@gmail.com	15/3/1983	Java	Java NP
DNI	0001	Octavio Robleto	Capital Federal - Villa Devoto	octavio.robleto@gmail.com	15/3/1983	Java	Base de Datos
DNI	0001	Octavio Robleto	Capital Federal - Villa Devoto	octavio.robleto@gmail.com	15/3/1983	Java	Spring
PAS	0003	Mariana Bracho	Provincia de Buenos Aires - Pilar	mariana.bracho@gmail.com	6/6/1989	.NET	C#
PAS	0003	Mariana Bracho	Provincia de Buenos Aires - Pilar	mariana.bracho@gmail.com	6/6/1989	.NET	Base de Datos
CI	0002	Sabrina Castagno	Provincia de Buenos Aires - Tigre	sabrina.castagno@gmail.com	5/4/1987	Front End	Java Script
CI	0002	Sabrina Castagno	Provincia de Buenos Aires - Tigre	sabrina.castagno@gmail.com	5/4/1987	Front End	Base de Datos



Tipo Documento	Numero de Documento	Nombre	Apellido	Provincia	Urbanización	Correo	Fecha de Nacimiento	Carrera	Curso
DNI	0001	Octavio	Robleto	Capital Federal	Villa Devoto	octavio.robledo@gmail.com	15/3/1983	Java	Java NP
DNI	0001	Octavio	Robleto	Capital Federal	Villa Devoto	octavio.robledo@gmail.com	15/3/1983	Java	Base de Datos
DNI	0001	Octavio	Robleto	Capital Federal	Villa Devoto	octavio.robledo@gmail.com	15/3/1983	Java	Spring
PAS	0003	Mariana	Bracho	Provincia de Buenos Aires	Pilar	mariana.bracho@gmail.com	6/6/1989	.NET	C#
PAS	0003	Mariana	Bracho	Provincia de Buenos Aires	Pilar	mariana.bracho@gmail.com	6/6/1989	.NET	Base de Datos
CI	0002	Sabrina	Castagno	Provincia de Buenos Aires	Tigre	sabrina.castagno@gmail.com	5/4/1987	Front End	Java Script
CI	0002	Sabrina	Castagno	Provincia de Buenos Aires	Tigre	sabrina.castagno@gmail.com	5/4/1987	Front End	Base de Datos



Tipo Documento	Numero de Documento	Nombre	Apellido	Provincia	Urbanización	Correo	Fecha de Nacimiento	Carrera	ID Curso
DNI	0001	Octavio	Robleto	Capital Federal	Villa Devoto	octavio.robleto@gmail.com	15/3/1983	Java	JAVA01
DNI	0001	Octavio	Robleto	Capital Federal	Villa Devoto	octavio.robleto@gmail.com	15/3/1983	Java	BD01
DNI	0001	Octavio	Robleto	Capital Federal	Villa Devoto	octavio.robleto@gmail.com	15/3/1983	Java	JAVASP01
PAS	0003	Mariana	Bracho	Provincia de Buenos Aires	Pilar	mariana.bracho@gmail.com	6/6/1989	.NET	C01
PAS	0003	Mariana	Bracho	Provincia de Buenos Aires	Pilar	mariana.bracho@gmail.com	6/6/1989	.NET	BD01
CI	0002	Sabrina	Castagno	Provincia de Buenos Aires	Tigre	sabrina.castagno@gmail.com	5/4/1987	Front End	JS01
CI	0002	Sabrina	Castagno	Provincia de Buenos Aires	Tigre	sabrina.castagno@gmail.com	5/4/1987	Front End	BD01

ID	Curso
JAVA01	Java NP
BD01	Base de Datos
JAVASP01	Spring
C01	C#
JS01	Java Script



Tipo Documento	Numero de Documento	Nombre	Apellido	ID Provincia	ID Urbanización	Correo	Fecha de Nacimiento	Carrera	ID Curso
DNI	0001	Octavio	Robleto	1	1	octavio.robleto@gmail.com	15/3/1983	1	JAVA01
DNI	0001	Octavio	Robleto	1	1	octavio.robleto@gmail.com	15/3/1983	1	BD01
DNI	0001	Octavio	Robleto	1	1	octavio.robleto@gmail.com	15/3/1983	1	JAVASP01
PAS	0003	Mariana	Bracho	2	2	mariana.bracho@gmail.com	6/6/1989	2	C01
PAS	0003	Mariana	Bracho	2	2	mariana.bracho@gmail.com	6/6/1989	2	BD01
CI	0002	Sabrina	Castagno	2	3	sabrina.castagno@gmail.com	5/4/1987	3	JS01
CI	0002	Sabrina	Castagno	2	3	sabrina.castagno@gmail.com	5/4/1987	3	BD01

ID	Provincia
1	Capital Federal
2	Provincia de Buenos Aires

ID	Urbanización	ID Provincia
1	Villa Devoto	1
2	Pilar	2
3	Tigre	2

ID Carrera	Carrera
1	Java
2	.NET
3	Front End

ID	Curso
JAVA01	Java NP
BD01	Base de Datos
JAVASP01	Spring
C01	C#
JS01	Java Script



Tipo Documento	Numero de Documento	Nombre	Apellido	ID Provincia	ID Urbanización	Correo	Fecha de Nacimiento	Carrera
DNI	0001	Octavio	Robleto	1	1	octavio.robleto@gmail.com	15/3/1983	1
PAS	0003	Mariana	Bracho	2	2	mariana.bracho@gmail.com	6/6/1989	2
CI	0002	Sabrina	Castagno	2	3	sabrina.castagno@gmail.com	5/4/1987	3

ID	Provincia
1	Capital Federal
2	Provincia de Buenos Aires

ID	Urbanización	ID Provincia
1	Villa Devoto	1
2	Pilar	2
3	Tigre	2

ID Carrera	Carrera
1	Java
2	.NET
3	Front End

ID	Curso
JAVA01	Java NP
BD01	Base de Datos
JAVASP01	Spring
C01	C#
JS01	Java Script

Tipo Documento	Numero Documento	ID Curso
DNI	0001	JAVA01
DNI	0001	BD01
DNI	0001	JAVASP01
PAS	0003	C01
PAS	0003	BD01
CI	0002	JS01
CI	0002	BD01



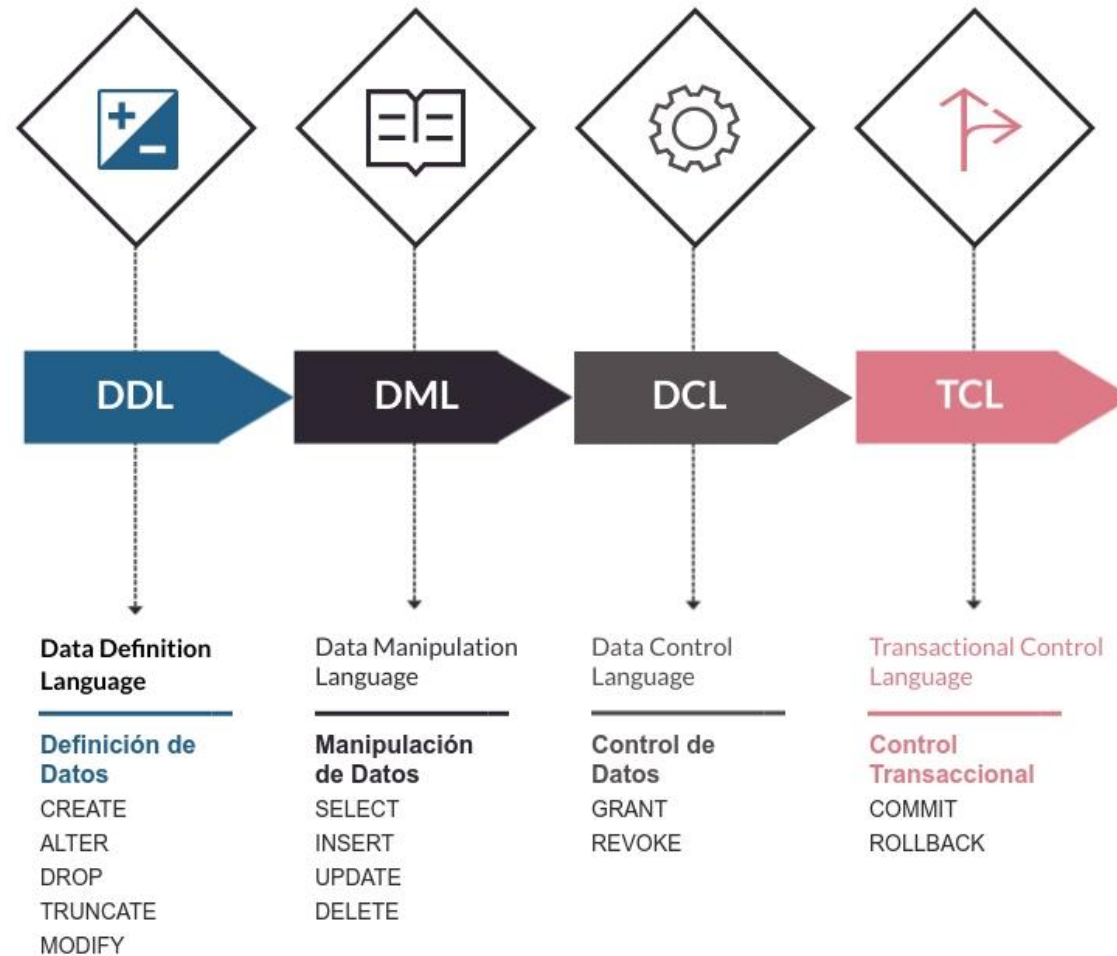
SQL

- Es un Lenguaje Declarativo estándar de alto nivel con el cual se comunica con las bases de datos relacionales.
- Su nombre en ingles “Structured Query Language”, que traducido al español significa : Lenguaje de Consultas Estructurado.





CLASIFICACIÓN DE SQL



CREADO POR
Octavio Robleto

<https://orobeto.github.io/octaviorobeto.github.io>



Prof. Octavio Robleto

Comentarios

Se utilizan para explicar secciones de sentencias SQL o para evitar la ejecución de sentencias SQL.

- Los comentarios de código multilínea están envueltos en `/* ... */`
- Los comentarios de una línea pueden comenzar con la almohadilla o numeral `#` o también con dos guiones consecutivos.

```
/*  
    1 - Cuando Necesito varias lineas  
    2 - ...  
*/  
  
-- Cuando Necesito una sola linea  
  
# Cuando Necesito una sola linea
```



DDL CREATE

- La instrucción se usa para crear objetos en la base de datos (Base de Datos, Tablas, ETC)

```
create database NombreBaseDeDatos;
```

```
create table NombreTabla(  
    nombreColumna1 tipoDato,  
    nombreColumna2 tipoDato,  
    nombreColumnan tipoDato,  
    clavePrimaria(NombreColumna),  
    claveForanea(NombreColumna)  
    referencia a nombreTabla (nombreColumna)  
);
```



DDL DROP

- La instrucción se usa para elimina objetos en la base de datos (Base de Datos, Tablas, ETC)

```
drop database NombreBaseDeDatos;
```

```
drop table NombreTabla;
```



DDL ALTER

- La instrucción se usa para agregar, eliminar y modificar columnas en una tabla.

```
alter table nombreTabla  
add nombreColumna tipoDato;
```

```
alter table nombreTabla  
drop nombreColumna;
```

```
alter table nombreTabla  
modify nombreColumna tipoDato;
```



Restricciones

- Clave Primaria: es uno o combinación de campos que le indica a la tabla que ese registro (fila) es único.
- Clave Foránea: es uno o combinación de campos que le indica a la tabla que debe existir

