

## LA DIFERENCIA ENTRE SQL Y MySQL:

**SQL:** es un lenguaje de programación orientado a consultas de bases de datos (Structured Query Language).

**MySQL:** es un **sistema de administración de bases de datos** (Database Management System, DBMS) o también llamado motor de bases de datos.



Existen muchos tipos de bases de datos, desde un simple archivo hasta sistemas relacionales orientados a objetos. MySQL, como base de datos relacional, utiliza múltiples tablas para almacenar y organizar la información.

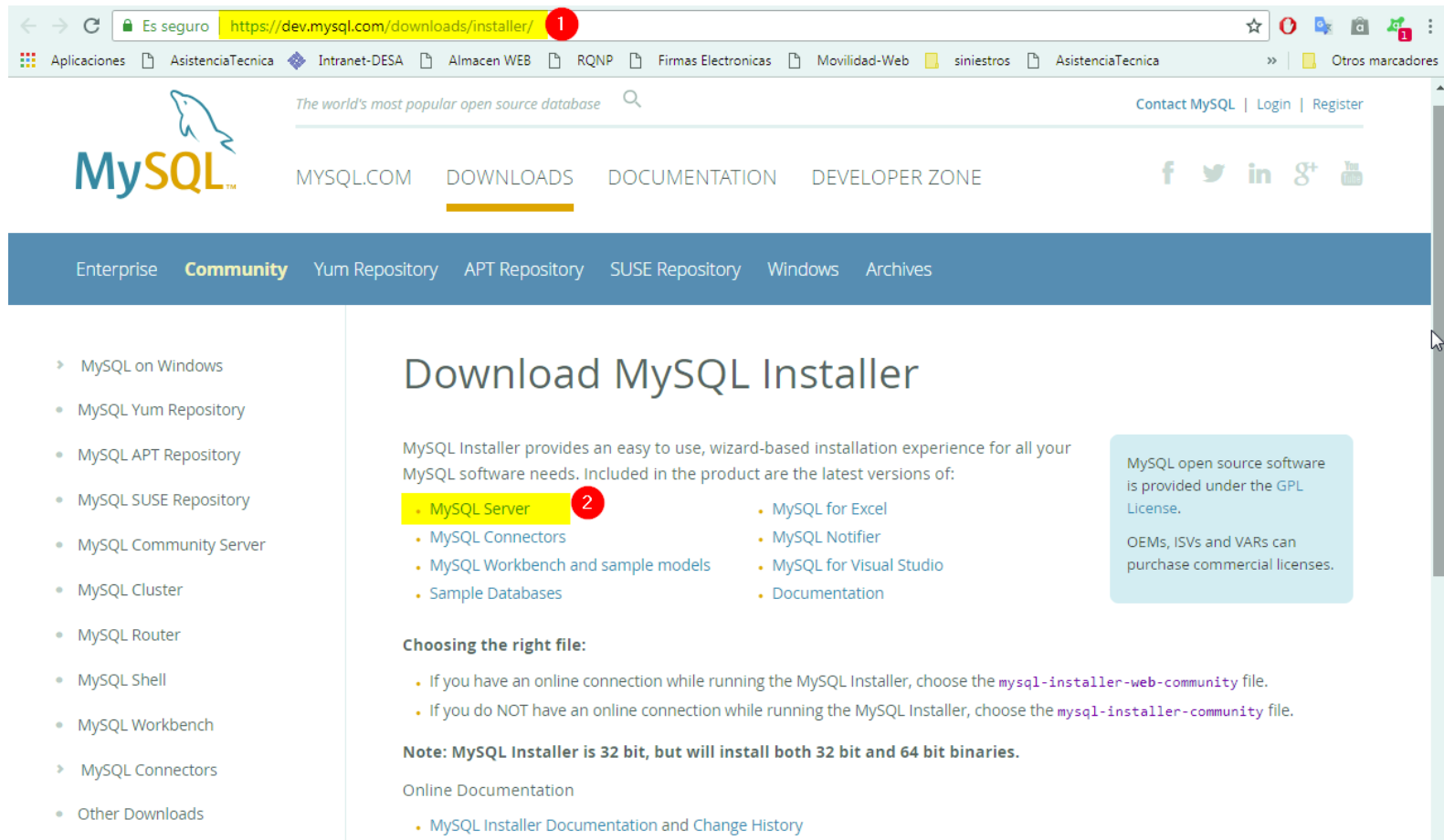
Fue escrito en C y C++ y se integra perfectamente con los lenguajes de programación más usados en todo el mundo.



## Instalación de MySQL

Para realizar la instalación de MySQL en tu pc lo primero que debes tener en cuenta es que debes hacer la verificación de la versión que quieres instalar y la distribución para tu sistema operativo. En este enlace encuentras el listado de las plataformas soportadas.

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



The screenshot shows the MySQL Downloads page in a web browser. The address bar shows the URL <https://dev.mysql.com/downloads/installer/> with a red circle '1' next to it. The page header includes the MySQL logo, the tagline 'The world's most popular open source database', and navigation links for MySQL.COM, DOWNLOADS (highlighted with an orange bar), DOCUMENTATION, and DEVELOPER ZONE. Below the header is a blue navigation bar with links for Enterprise, Community (highlighted), Yum Repository, APT Repository, SUSE Repository, Windows, and Archives. The main content area is titled 'Download MySQL Installer' and describes the MySQL Installer as a wizard-based installation experience. It lists the latest versions of MySQL Server (highlighted with a red circle '2'), MySQL Connectors, MySQL Workbench and sample models, Sample Databases, MySQL for Excel, MySQL Notifier, MySQL for Visual Studio, and Documentation. A sidebar on the left lists various MySQL products and downloads. A blue box on the right states that MySQL open source software is provided under the GPL License and that OEMs, ISVs, and VARs can purchase commercial licenses. The page also includes a 'Choosing the right file:' section with instructions on selecting the correct installer file based on online connection status, a note about 32-bit and 64-bit binaries, and links to online documentation and change history.

Es seguro <https://dev.mysql.com/downloads/installer/> 1

Aplicaciones AsistenciaTecnica Intranet-DESA Almacen WEB RQNP Firmas Electronicas Movilidad-Web siniestros AsistenciaTecnica Otros marcadores

MySQL™ The world's most popular open source database Contact MySQL | Login | Register

MYSQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE

Enterprise **Community** Yum Repository APT Repository SUSE Repository Windows Archives

### Download MySQL Installer

MySQL Installer provides an easy to use, wizard-based installation experience for all your MySQL software needs. Included in the product are the latest versions of:

- MySQL Server 2
  - MySQL Connectors
  - MySQL Workbench and sample models
  - Sample Databases
- MySQL for Excel
- MySQL Notifier
- MySQL for Visual Studio
- Documentation

**Choosing the right file:**

- If you have an online connection while running the MySQL Installer, choose the `mysql-installer-web-community` file.
- If you do NOT have an online connection while running the MySQL Installer, choose the `mysql-installer-community` file.

**Note: MySQL Installer is 32 bit, but will install both 32 bit and 64 bit binaries.**

Online Documentation

- MySQL Installer Documentation and Change History

MySQL open source software is provided under the GPL License. OEMs, ISVs and VARs can purchase commercial licenses.

- MySQL on Windows
- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Workbench
- MySQL Connectors
- Other Downloads

# MySQL Installer

## MySQL. Installer

Adding Community

License Agreement

Choosing a Setup Type

Select Products and Features

Check Requirements

Installation

Product Configuration

Installation Complete

## Check Requirements

The following products have failing requirements. The installer will attempt to resolve some of this automatically. Requirements marked as manual cannot be resolved automatically. Click on those items to try and resolve them manually.

For Product	Requirement	Status
✓ MySQL Workbench 6.2.4	Microsoft Visual C++ 2013 Runtime...	

< Back

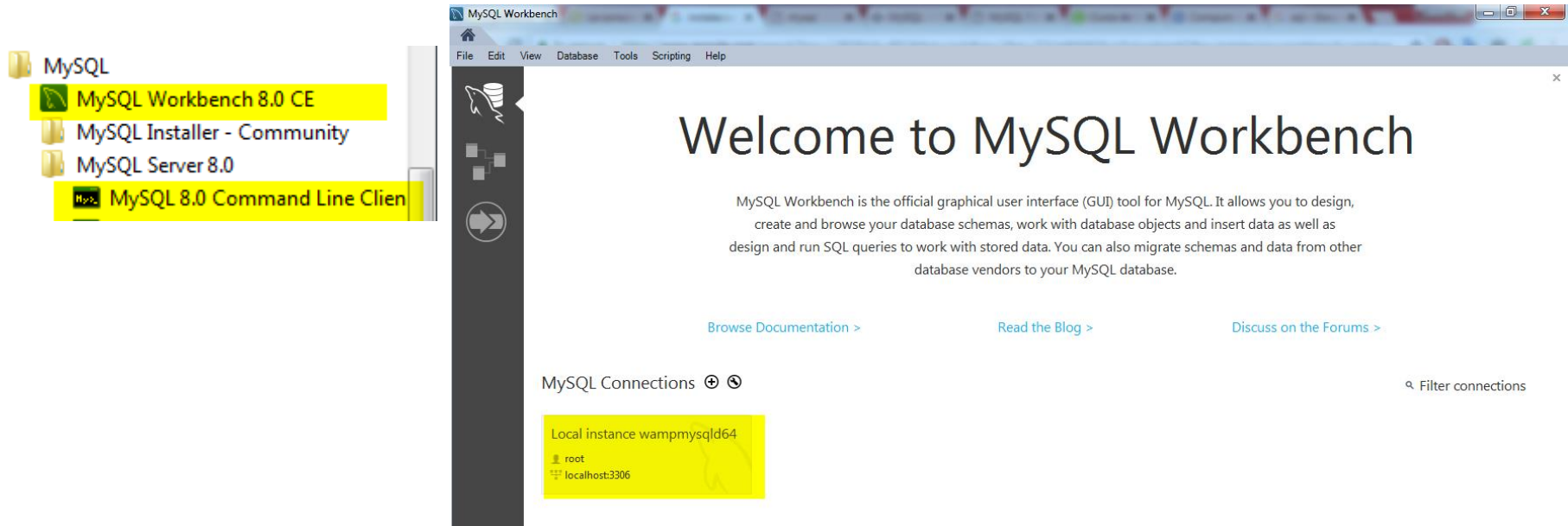
Next >

Cancel

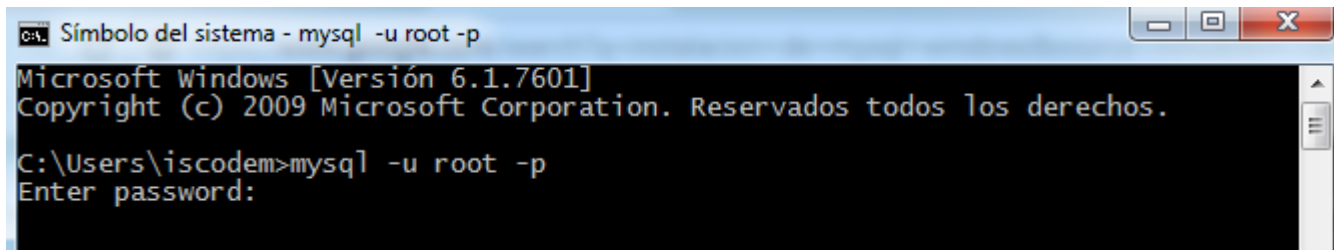
## Iniciar MySQL

### Por Interface grafica:

Se requiere instalar la herramienta WORKBENCH de Oracle y que trabaja particularmente en MySQL.



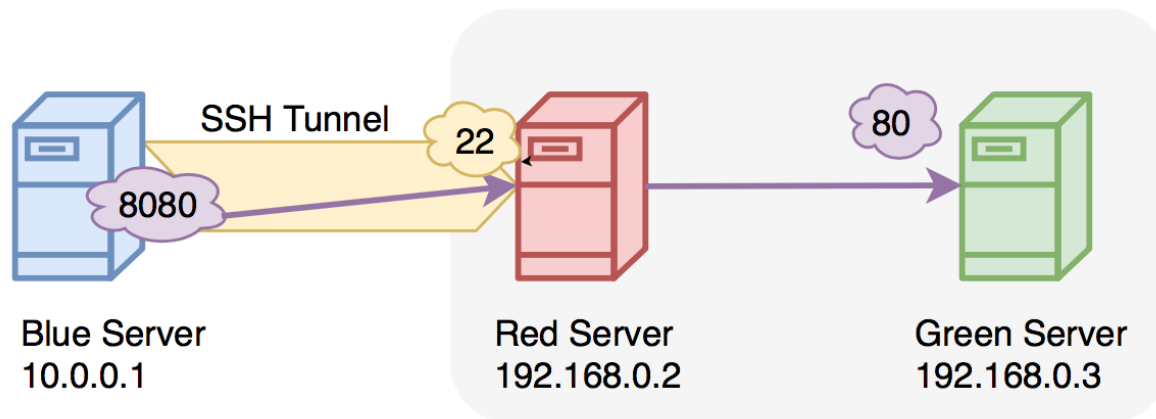
### Por línea de comandos:



La forma de conectarnos a nuestro servidor MySQL a través de nuestra terminal podría ser:

```
$ mysql -h <dirección_de_nuestro_servidor> -u <usuario> -  
pmi_clave -P <puerto>
```

Una situación muy normal en entornos de producción es tener que acceder a nuestro servidor a través de un túnel SSH, es decir, primero accedemos a una máquina que podría ser la entrada de nuestro cluster y después, desde ahí, accederíamos a nuestro servidor MySQL.



# Comandos básicos en MySQL

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/
MariaDB [mysql]> SHOW DATABASES;
+-----+
| Database |
+-----+
| db_iscodem |
| information_schema |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)
```

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/
MariaDB [mysql]> SELECT USER();
+-----+
| USER() |
+-----+
| root@localhost |
+-----+
1 row in set (0.00 sec)
```

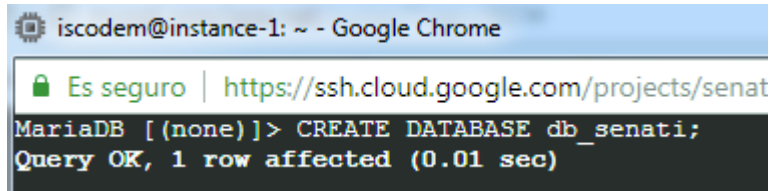
```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/
MariaDB [mysql]> SELECT CURRENT_DATE();
+-----+
| CURRENT_DATE() |
+-----+
| 2018-06-11 |
+-----+
1 row in set (0.00 sec)
```

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/senatiproj/zones/us-east1-b/
MariaDB [mysql]> SELECT VERSION(), CURRENT_DATE, NOW();
+-----+
| VERSION() | CURRENT_DATE | NOW() |
+-----+
| 10.1.26-MariaDB-0+deb9u1 | 2018-06-11 | 2018-06-11 06:44:33 |
+-----+
1 row in set (0.00 sec)
```

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/proj
MariaDB [mysql]> QUIT;
Bye
root@instance-1:/home/iscodem#
```

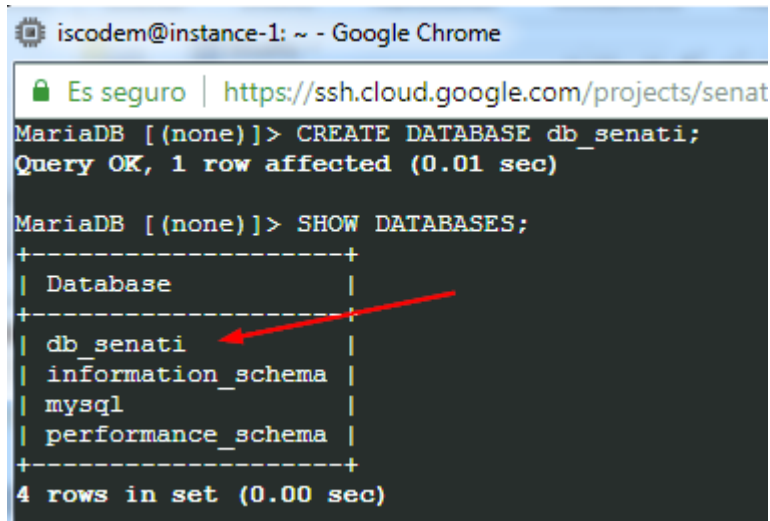
# CREAR BASES DE DATOS

## 1. Crear DATABASE



A terminal window titled 'iscodem@instance-1: ~ - Google Chrome' showing a MariaDB prompt. The user enters the command 'CREATE DATABASE db\_senati;'. The output is 'Query OK, 1 row affected (0.01 sec)'.

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/senat
MariaDB [(none)]> CREATE DATABASE db_senati;
Query OK, 1 row affected (0.01 sec)
```

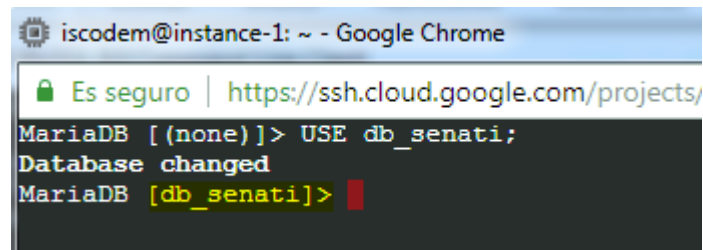


A terminal window titled 'iscodem@instance-1: ~ - Google Chrome' showing a MariaDB prompt. The user enters the command 'SHOW DATABASES;'. The output is a table listing four databases: 'db\_senati', 'information\_schema', 'mysql', and 'performance\_schema'. A red arrow points to 'db\_senati'.

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/senat
MariaDB [(none)]> CREATE DATABASE db_senati;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| db_senati |
| information_schema |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.00 sec)
```

## 2. Seleccionar DATABASE para trabajar en ella.



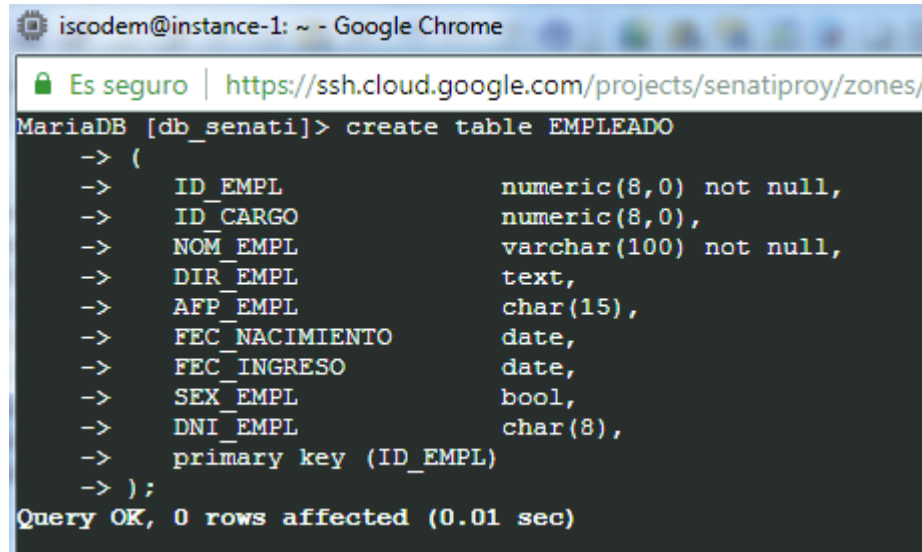
A terminal window titled 'iscodem@instance-1: ~ - Google Chrome' showing a MariaDB prompt. The user enters the command 'USE db\_senati;'. The output is 'Database changed'. The prompt changes to 'MariaDB [db\_senati]>'.

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/
MariaDB [(none)]> USE db_senati;
Database changed
MariaDB [db_senati]>
```



# CREAR TABLAS EN UNA BASE DE DATOS MySQL

## 1. Crear TABLE



The screenshot shows a terminal window with a Google Chrome browser interface at the top. The address bar shows a secure connection to <https://ssh.cloud.google.com/projects/senatiproj/zones/>. The terminal prompt is `MariaDB [db_senati]>`. The user enters the command `create table EMPLEADO` followed by a multi-line definition for the table structure. The definition includes columns: `ID_EMPL` (numeric(8,0) not null), `ID_CARGO` (numeric(8,0)), `NOM_EMPL` (varchar(100) not null), `DIR_EMPL` (text), `AFP_EMPL` (char(15)), `FEC_NACIMIENTO` (date), `FEC_INGRESO` (date), `SEX_EMPL` (bool), and `DNI_EMPL` (char(8)). A primary key is defined on `ID_EMPL`. The command ends with `);`. The terminal response is `Query OK, 0 rows affected (0.01 sec)`.

```
MariaDB [db_senati]> create table EMPLEADO
-> (
->   ID_EMPL          numeric(8,0) not null,
->   ID_CARGO         numeric(8,0),
->   NOM_EMPL         varchar(100) not null,
->   DIR_EMPL         text,
->   AFP_EMPL         char(15),
->   FEC_NACIMIENTO   date,
->   FEC_INGRESO      date,
->   SEX_EMPL         bool,
->   DNI_EMPL         char(8),
->   primary key (ID_EMPL)
-> );
Query OK, 0 rows affected (0.01 sec)
```

```
CREATE TABLE `authors` (  
  `author_id`      int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `name`           varchar(100) NOT NULL,  
  `nationality`    varchar(100) DEFAULT NULL,  
  PRIMARY KEY (`author_id`),  
  UNIQUE KEY `uniq_author` (`name`)  
) ENGINE=InnoDB AUTO_INCREMENT=193 DEFAULT CHARSET=utf8;
```

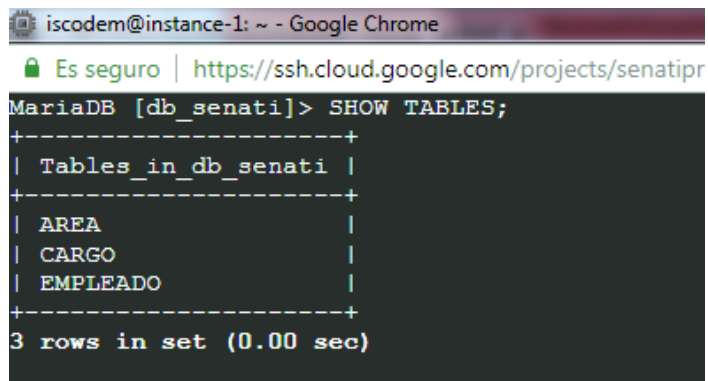
## 2. Mostrar el detalle del estado de la tabla

```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/senatiproj/zones/us-east1-b/instances/instance-1?authuser=0&hl=en_US&projectNumber=763226118344
MariaDB [db_senati]> SHOW TABLE STATUS;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Name      | Engine | Version | Row_format | Rows | Avg_row_length | Data_length | Max_data_length | Index_length | Data_free | Auto_increment | Create_time |
| Update_time | Check_time | Collation | Checksum | Create_options | Comment |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| AREA      | InnoDB | 10      | Compact    | 0    | 0              | 16384       | 0              | 0            | 0         | NULL           | -06-11 06:58:58 |
| NULL      | NULL    | NULL    | NULL       | utf8mb4_general_ci | NULL      |
| CARGO      | InnoDB | 10      | Compact    | 0    | 0              | 16384       | 0              | 0            | 0         | NULL           | -06-11 06:59:26 |
| NULL      | NULL    | NULL    | NULL       | utf8mb4_general_ci | NULL      |
| EMPLEADO   | InnoDB | 10      | Compact    | 0    | 0              | 16384       | 0              | 0            | 0         | NULL           | -06-11 06:59:37 |
| NULL      | NULL    | NULL    | NULL       | utf8mb4_general_ci | NULL      |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

## 3. Mostrar la estructura de la tabla

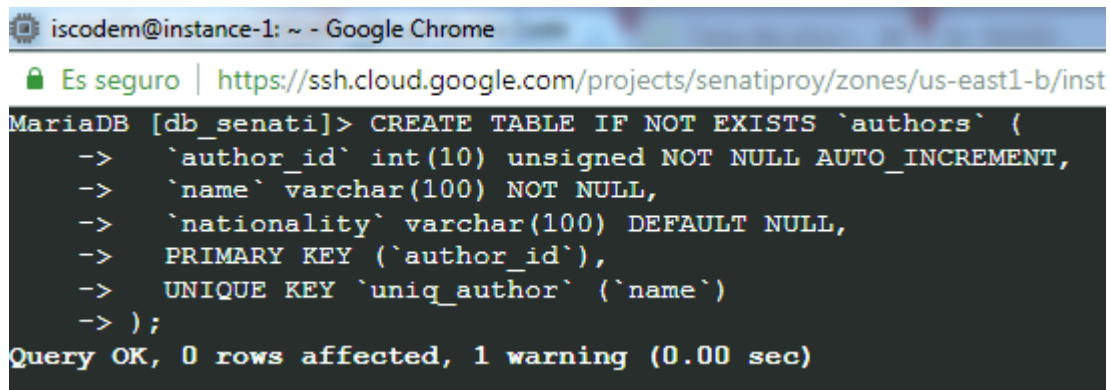
```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/senatiproj/zones/us-east1-b/instances/
MariaDB [db_senati]> DESC EMPLEADO;
+-----+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+-----+
| ID_EMPL    | decimal(8,0)  | NO   | PRI | NULL    |
| ID_CARGO    | decimal(8,0)  | YES  |     | NULL    |
| NOM_EMPL    | varchar(100)  | NO   |     | NULL    |
| DIR_EMPL    | text          | YES  |     | NULL    |
| AFP_EMPL    | char(15)      | YES  |     | NULL    |
| FEC_NACIMIENTO | date         | YES  |     | NULL    |
| FEC_INGRESO  | date         | YES  |     | NULL    |
| SEX_EMPL    | tinyint(1)    | YES  |     | NULL    |
| DNI_EMPL    | char(8)       | YES  |     | NULL    |
+-----+-----+-----+-----+-----+-----+-----+
9 rows in set (0.02 sec)
```

#### 4. Mostrar todas las tablas de una base de datos.



```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/senatipr
MariaDB [db_senati]> SHOW TABLES;
+-----+
| Tables_in_db_senati |
+-----+
| AREA                 |
| CARGO                 |
| EMPLEADO              |
+-----+
3 rows in set (0.00 sec)
```

#### 5. Crear una tabla verificando su existencia en la BD.



```
iscodem@instance-1: ~ - Google Chrome
Es seguro | https://ssh.cloud.google.com/projects/senatiproj/zones/us-east1-b/inst
MariaDB [db_senati]> CREATE TABLE IF NOT EXISTS `authors` (
  ->   `author_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  ->   `name` varchar(100) NOT NULL,
  ->   `nationality` varchar(100) DEFAULT NULL,
  ->   PRIMARY KEY (`author_id`),
  ->   UNIQUE KEY `uniq_author` (`name`)
  -> );
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

## Consideraciones importantes al crear tablas

**DATO IMPORTANTE:** MySql no detecta cuando los datos de AUTO\_INCREMENT han sido borrados, por tanto seguirá con el ultimo dato que guardo.

**NOT NULL** permite que un campo no pueda estar sin completar, es distinto NULO a EMPTY.

**INTEGER UNSIGNED** permite que no existan signos, de forma que los numeros siempre sean positivos.

**DEFAULT [valor por defecto]:** Si no introducimos ningun dato, mySQL automaticamente le dara al campo el valor de DEFAULT.

La diferencia entre CHAR y VARCHAR es que en el primer tipo de dato se asigna un valor en memoria por defecto dependiendo el tamaño de espacios que sean asignados, en cambio, VARCHAR va llenando esos campos en memoria conforme se tengan los datos en el campo.

**COMMENT:** Comentario a la columna que solo es visible para quien este manejando la base de datos

Para imagenes asignamos VARCHAR, NOTA: no se guarda la imagen como tal, guardamos el url del origen de la imagen

**FLOAT** es utilizado para calculos precisos, DOUBLE puede ser aplicado de forma simple a los precios de una libreria (en este caso)

**DOUBLE** (espacios que podremos llenar, espacios asignados para numeros decimales)

**TEXT** permite agregar texto, grandes cantidades de caracteres

**MYSQL** acepta minúsculas y mayúsculas para palabras reservadas, se recomienda usar MAYÚSCULAS.

Si necesitas renombrar una tabla puedes hacerlo con el comando **RENAME TABLE** author **TO** authors (Por mencionar un ejemplo).

**DROP TABLE:** Borra una tabla, el contenido y la estructura. Mucho cuidado con este comando.

**DESCRIBE** 'nombre de la tabla' : nos muestra las columnas de la tabla.

**SHOW FULL COLUMNS FROM** 'nombre de la tabla': muestra toda la información de la tabla.

La palabra **YEAR** es una palabra reservada, si queremos utilizar una palabra reservada utilizamos una tilde inversa, no es muy recomendado.

Es posible utilizar TINYINT como bandera booleana (1 ó 0).

## **MyISAM**

1. Bloqueo de tabla
2. Aumento del rendimiento si nuestra aplicación realiza un elevado número de consultas "Select".
3. Las tablas pueden llegar a dar problemas en la recuperación de datos.
4. Permite hacer búsquedas full-text
5. Menor consumo memoria RAM
6. Mayor velocidad en general a la hora de recuperar datos.
7. Ausencia de características de atomicidad ya que no tiene que hacer comprobaciones de la integridad referencial, ni bloquear las tablas para realizar las operaciones, esto nos lleva como los anteriores puntos a una mayor velocidad.

## **InnoDB**

1. Bloqueo de registros
2. Soporte de transacciones
3. Rendimiento
4. Concurrencia
5. Confiabilidad
6. Permite hacer búsquedas full-text (mysql >= 5.6)

## Ventajas y desventajas de usar: AUTO\_INCREMENT

### Ventajas

- Son fáciles de definir, ya que no requieren análisis profundo (y por ende tiene a descuidarse la consistencia).
- Ocupan menos espacio
- Se realizan JOINS rápidos (ya que la evaluación es a nivel de binarios), y por ende pueden obtenerse mejores performances en ciertos casos.

### Desventajas

- No se puede hacer integración ni consolidación de sistemas sin complejos procesos de migración de datos con validaciones muy elaboradas.
- **Poco flexibles:** la clave es la lógica de operaciones/negocio, si por cualquier motivo quieres cambiarla, tendrías que cambiar todas las tablas, lo cual será un proceso costoso.
- **Normalización:** las tablas no están completamente normalizadas porque al añadir un campo adicional (y realmente innecesario para identificar una fila) hay varias claves candidatas.

## TIPOS DE DATOS EN MySQL

### Tipos de dato numéricos

Listado de cada uno de los **tipos de dato numéricos en MySQL**, su ocupación en disco y valores.

- **INT (INTEGER)**: Ocupación de 4 bytes con valores entre -2147483648 y 2147483647 o entre 0 y 4294967295.
- **SMALLINT**: Ocupación de 2 bytes con valores entre -32768 y 32767 o entre 0 y 65535.
- **TINYINT**: Ocupación de 1 bytes con valores entre -128 y 127 o entre 0 y 255.
- **MEDIUMINT**: Ocupación de 3 bytes con valores entre -8388608 y 8388607 o entre 0 y 16777215.
- **BIGINT**: Ocupación de 8 bytes con valores entre -8388608 y 8388607 o entre 0 y 16777215.
- **DECIMAL (NUMERIC)**: Almacena los números de coma flotante como cadenas o string.
- **FLOAT (m,d)**: Almacena números de **coma flotante**, donde 'm' es el número de dígitos de la parte entera y 'd' el número de decimales.
- **DOUBLE (REAL)**: Almacena número de coma flotante con precisión doble. Igual que FLOAT, la diferencia es el rango de valores posibles.
- **BIT (BOOL, BOOLEAN)**: Número entero con valor 0 o 1.



## Tipos de dato con formato fecha

Listado de cada uno de los **tipos de dato con formato fecha en MySQL**, su ocupación en disco y valores.

- **DATE:** Válido para almacenar una fecha con año, mes y día, su rango oscila entre '1000-01-01' y '9999-12-31'.
- **DATETIME:** Almacena una fecha (año-mes-día) y una hora (horas-minutos-segundos), su rango oscila entre '1000-01-01 00:00:00' y '9999-12-31 23:59:59'.
- **TIME:** Válido para almacenar una hora (horas-minutos-segundos). Su rango de horas oscila entre -838-59-59 y 838-59-59. El formato almacenado es 'HH:MM:SS'.
- **TIMESTAMP:** Almacena una fecha y hora UTC. El rango de valores oscila entre '1970-01-01 00:00:01' y '2038-01-19 03:14:07'.
- **YEAR:** Almacena un año dado con 2 o 4 dígitos de longitud, por defecto son 4. El rango de valores oscila entre 1901 y 2155 con 4 dígitos. Mientras que con 2 dígitos el rango es desde 1970 a 2069 (70-69).

## Diferentes tipos de dato con formato string

- **CHAR:** Ocupación fija cuya longitud comprende de 1 a 255 caracteres.
- **VARCHAR:** Ocupación variable cuya longitud comprende de 1 a 255 caracteres.
- **TINYBLOB:** Una longitud máxima de 255 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- **BLOB:** Una longitud máxima de 65.535 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- **MEDIUMBLOB:** Una longitud máxima de 16.777.215 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- **LOBLOB:** Una longitud máxima de 4.294.967.298 caracteres. Válido para objetos binarios como son un fichero de texto, imágenes, ficheros de audio o vídeo. No distingue entre minúsculas y mayúsculas.
- **SET:** Almacena 0, uno o varios valores una lista con un máximo de 64 posibles valores.
- **ENUM:** Igual que **SET** pero solo puede almacenar un valor.
- **TINYTEXT:** Una longitud máxima de 255 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.
- **TEXT:** Una longitud máxima de 65.535 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.
- **MEDIUMTEXT:** Una longitud máxima de 16.777.215 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.
- **LONGTEXT:** Una longitud máxima de 4.294.967.298 caracteres. Sirve para almacenar texto plano sin formato. Distingue entre minúsculas y mayúsculas.

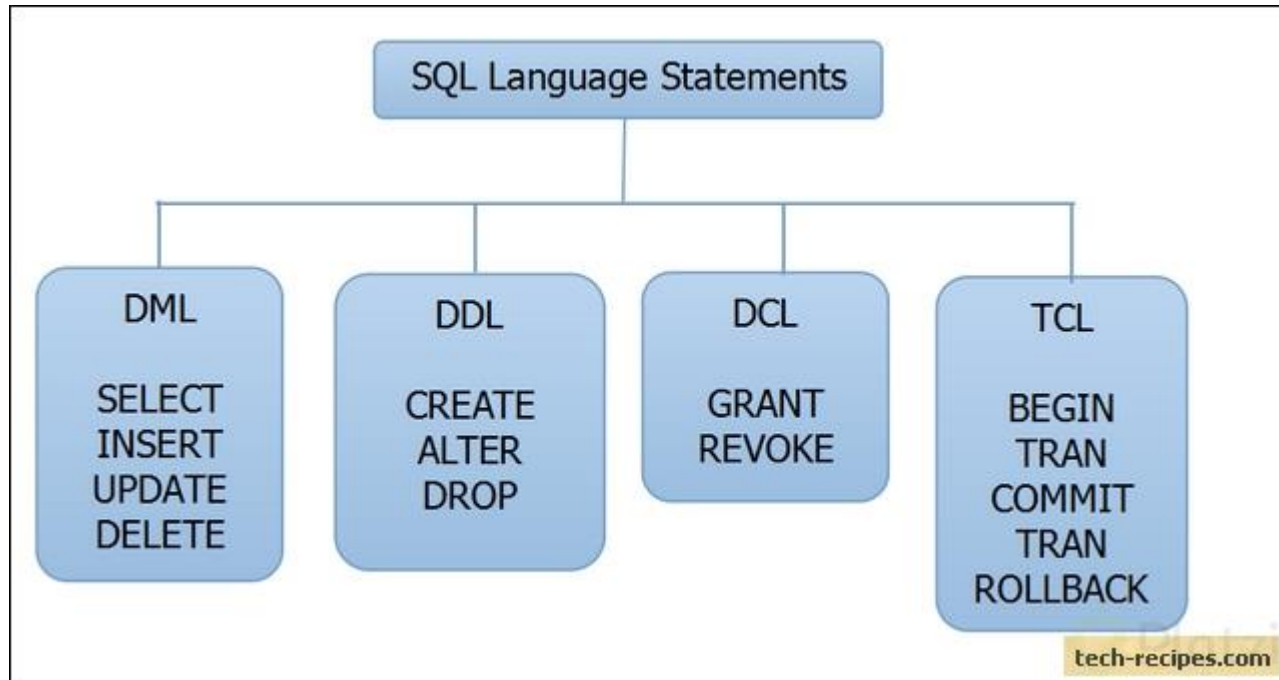
## OPERACIONES DE DATOS EN MySQL

Existen 3 cosas para poder hacer la descripción de una **Base de Datos**:

**DML** = Data Manipulation Language o Lenguaje de Manipulación de Datos.

**DDL** = Data Definition Language o Lenguaje de Definición de Datos.

**SQL** = Structured Query Language o Lenguaje de Consulta Estructurada.



## **Consideraciones al usar la sentencia SELECT**

**SELECT:** la herramienta más importante de SQL pues con ella logramos extraer los datos.

**WHERE:** agrega condicionantes a nuestro SELECT para traer datos más específicos.

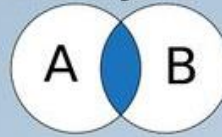
**LIKE:** busca coincidencias de texto.

**FUNCIONES:** realizan tareas ya programadas y muy puntuales.

**LIMIT n:** Trae un **n** número límite de coincidencias de nuestro select de forma aleatoria.

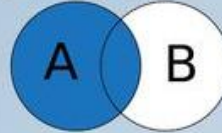
# SQL JOINS

## INNER JOIN



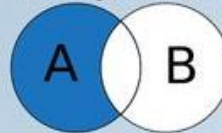
```
SELECT *  
FROM A  
INNER JOIN B ON A.key = B.key
```

## LEFT JOIN



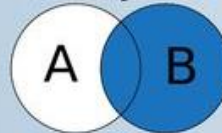
```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key
```

## LEFT JOIN (sans l'intersection de B)



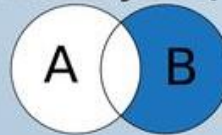
```
SELECT *  
FROM A  
LEFT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

## RIGHT JOIN



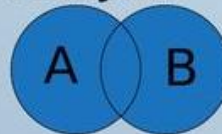
```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key
```

## RIGHT JOIN (sans l'intersection de A)



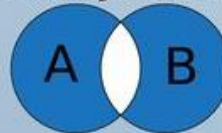
```
SELECT *  
FROM A  
RIGHT JOIN B ON A.key = B.key  
WHERE B.key IS NULL
```

## FULL JOIN



```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key
```

## FULL JOIN (sans intersection)



```
SELECT *  
FROM A  
FULL JOIN B ON A.key = B.key  
WHERE A.key IS NULL  
OR B.key IS NULL
```