



Este tutorial está basado en los 40 video tutoriales del curso de SQL desde 0, en el siguiente código QR podrás acceder.

También podrás acceder haciendo clic [aquí](#).

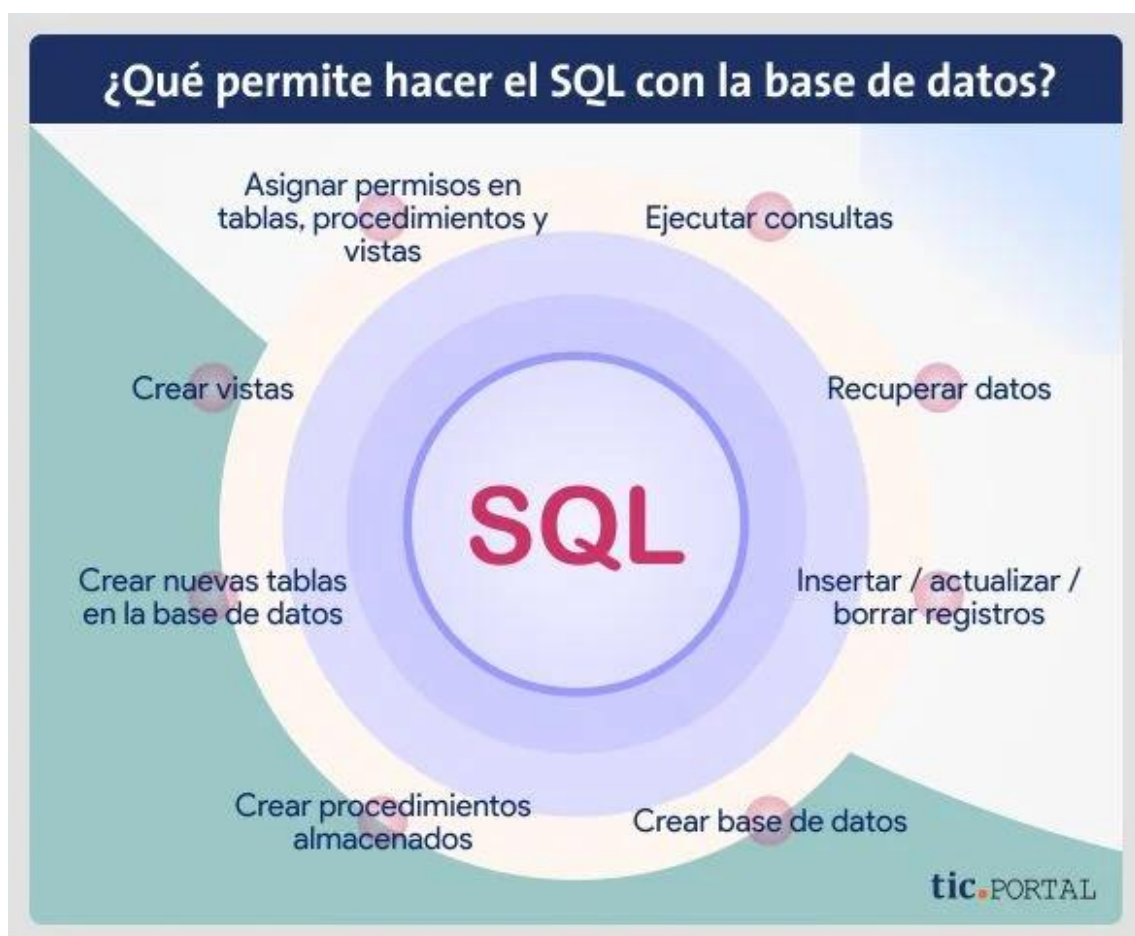
SQL DESDE 0
CANAL YOUTUBE DE DANISABLE

PERE MANEL VERDUGO ZAMORA

Web: www.peremanelv.com
Email: pereverdugo@gmail.com

Contenido

1.- ¿Qué es SQL?	3
2.- Instalación de herramientas en Windows.....	4
3.- Como Conectar XAMPP y MySQL Workbench.....	13
4.- Como crear una base de datos	24
5.- Como Crear un Tabla en una base de datos.....	25
6.- ¿Qué son las Sentencias de SQL (DML y DDL)?	26
7.- Identificadores de llaves primarias.....	27
8.- Tipos de datos.....	29
9.- Insertar Datos con INSERT.....	30
10.- Como obtener todos los datos en SEECT	32
11.- Como obtener un campo con SELECT	33
12.- UPDATE – Actualizar datos.....	34
13.- DELETE – Eliminar datos	36
14.- Comentarios	39
15.- Operadores.....	40
16.- Operador LIKE	42
17.- Valor NULL.....	44
18.- Comando DISTINCT	45
19.- Operadores Lógicos AND / OR	47
20.- Comando NOT	48
21.- Comando BETWEEN	50
22.- Comando ORDER BY.....	51
23.- ALIAS en SQL (AS).....	53
24.- TRUNCATE TABLE - Eliminar datos de Tablas	55
25.- DROP TABLE – Eliminar tablas.	56
26.- DROP DATABASE – Eliminar bases de datos.....	57
27.- EJERCICIO FINAL I (Modelos)	58
28.- EJERCICIO FINAL II (Creando la BD).....	59
29.- EJERCICIO FINAL III (Creando la tabla CONTRATO).....	61
30.- EJERCICIO FINAL IV (Insertando Datos)	62
31.- EJERCICIO FINAL V (Insertando datos Tabla CONTRATO).....	64
31.- EJERCICIO FINAL VI: Consultas Sencillas.....	68
32.- EJERCICIO FINAL VII: Consultas específicas	70
33.- EJERCICIO FINAL VIII: Consultas con Operadores Lógicos	71
34.- EJERCICIO FINAL IX: Alias o Sobrenombres	72



1.- ¿Qué es SQL?

SQL significa LENGUAJE ESTRUCTURADO DE CONSULTAS por sus siglas en inglés “Structured Query Language”.

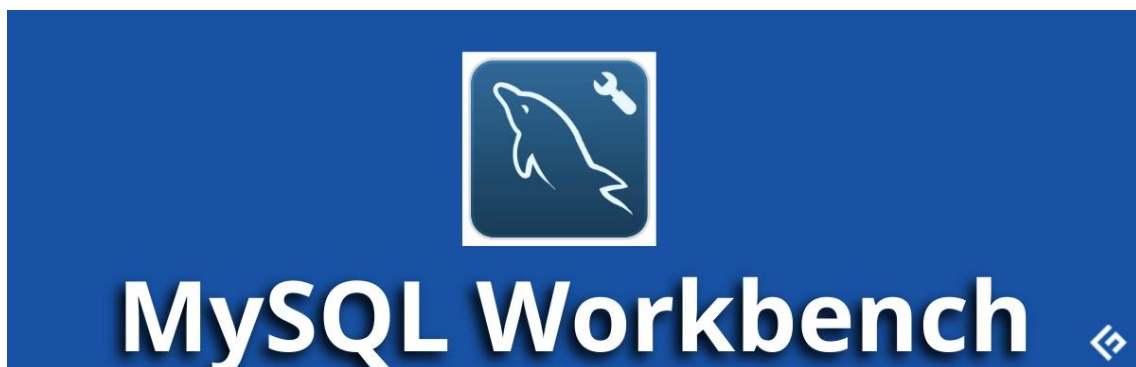
Se utiliza para definir, gestionar y manipular la información contenida en una Base de datos Relacional.

Es un lenguaje sencillo y potente que se emplea para la gestión de base de datos a distintos niveles de utilización, programadores y administradores de la base de datos.

HERRAMIENTAS



Vamos a utilizar un servidor local dentro de nuestro ordenador y para ello vamos a utilizar el programa XAMPP y para realizar nuestros modelos de base de datos y realizar la administración de la base de datos utilizaremos MySQL Workbench.



2.- Instalación de herramientas en Windows

Desde un navegador por ejemplo Google:



XAMPP

apachefriends.org
https://www.apachefriends.org > ...

XAMPP Installers and Downloads for Apache Friends

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de **XAMPP** ha sido ...

Visitaste esta página el 6/02/23.

Apache Friends Descargar Alojamiento Comunidad Acerca de

Buscar... Buscar ES

XAMPP Apache + MariaDB + PHP + Perl

¿Qué es XAMPP?

XAMPP es el entorno más popular de desarrollo con PHP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.



Descargar
Pulsa aquí para otras versiones

XAMPP para Windows
8.2.0 (PHP 8.2.0)

XAMPP para Linux
8.2.0 (PHP 8.2.0)

XAMPP para OS X
8.2.0 (PHP 8.2.0)

XAMPP es el entorno de desarrollo más popular con PHP, es una distribución de Apache totalmente gratuita y fácil de instalar.

Contiene MariaDB, PHP y Perl.

XAMPP es un servidor Apache que nos va a permitir instalar un servidor local para que nosotros podamos instalar dentro de este servidor nuestras bases de datos de esta manera nosotros vamos a emular una conexión real a través de MySQL.

De este modo en nuestro ordenador tenemos un servidor, es decir un ordenador externo y dentro de este servidor vamos a instalar nuestra base de datos.

Empieza la descarga.

Mientras se realiza la descarga vamos a buscar MySQL.



mysql workbench



mysql.com

<https://www.mysql.com> > products · Traducir esta página

MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs.

MySQL Workbench provides data modeling, SQL development, ...

MySQL :: MySQL Workbench

mysql.com/products/workbench/

Aplicaciones

The world's most popular open source database

MySQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE

Products Cloud Services Partners Customers Why MySQL? News & Events How to Buy

MySQL HeatWave

MySQL Enterprise Edition

- Datasheet (PDF)
- Technical Specification
- MySQL Database
- MySQL Document Store
- Oracle Enterprise Manager
- Enterprise Monitor

MySQL Workbench

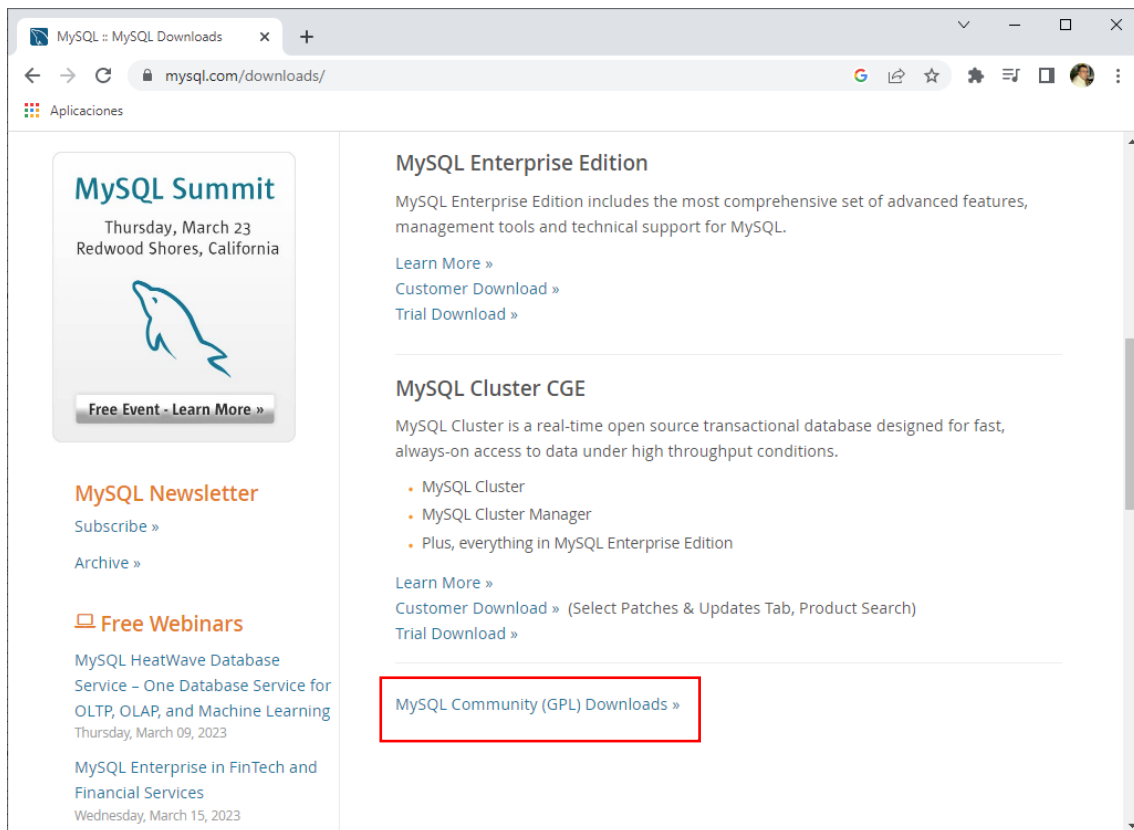
Enhanced Data Migration

Download Now »

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and server administration tools for server configuration, user administration, backup, and recovery.

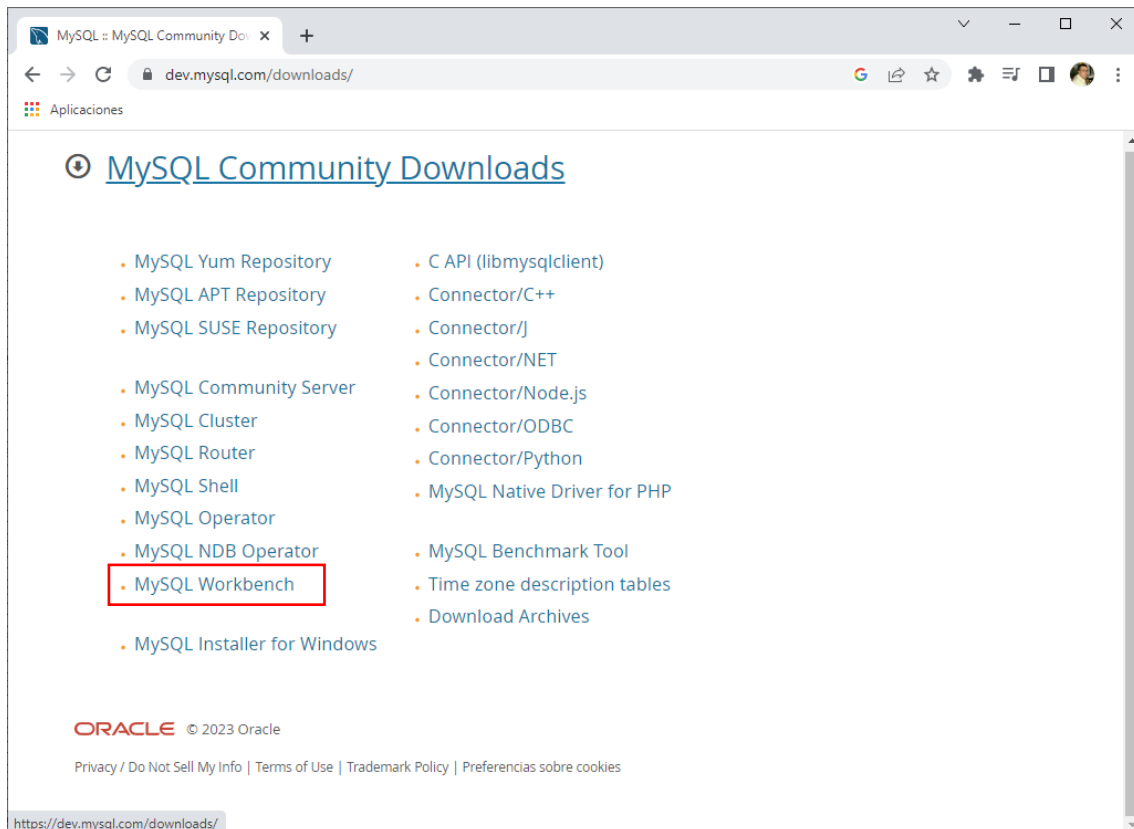
Start chat

Vamos a Downloads.



The screenshot shows the MySQL Downloads page on the official website. The page is divided into several sections. On the left, there is a sidebar with a 'MySQL Summit' announcement for Thursday, March 23 in Redwood Shores, California, featuring a blue fish logo and a 'Free Event - Learn More »' button. Below this is a 'MySQL Newsletter' section with 'Subscribe »' and 'Archive »' links. Further down is a 'Free Webinars' section listing two events: 'MySQL HeatWave Database Service - One Database Service for OLTP, OLAP, and Machine Learning' (Thursday, March 09, 2023) and 'MySQL Enterprise in FinTech and Financial Services' (Wednesday, March 15, 2023). The main content area on the right features 'MySQL Enterprise Edition' with a description and links for 'Learn More », 'Customer Download », and 'Trial Download ». Below this is the 'MySQL Cluster CGE' section, also with 'Learn More », 'Customer Download » (Select Patches & Updates Tab, Product Search)', and 'Trial Download »' links. At the bottom of the main content area, a red rectangular box highlights the link 'MySQL Community (GPL) Downloads »'.

Descarga para uso no comercial, en nuestro caso será académico.



The screenshot shows the MySQL Community Downloads page on the developer website. The page has a header with the title 'MySQL Community Downloads' and a list of download links organized in two columns. The links include: MySQL Yum Repository, MySQL APT Repository, MySQL SUSE Repository, MySQL Community Server, MySQL Cluster, MySQL Router, MySQL Shell, MySQL Operator, MySQL NDB Operator, MySQL Workbench (highlighted with a red rectangular box), MySQL Installer for Windows, C API (libmysqlclient), Connector/C++, Connector/J, Connector/NET, Connector/Node.js, Connector/ODBC, Connector/Python, MySQL Native Driver for PHP, MySQL Benchmark Tool, Time zone description tables, and Download Archives. At the bottom of the page, there is an 'ORACLE © 2023 Oracle' footer, a privacy policy link, and a URL bar showing 'https://dev.mysql.com/downloads/'.

MySQL :: Download MySQL Workbench

dev.mysql.com/downloads/workbench/

MySQL Workbench 8.0.32

Select Operating System:

Microsoft Windows

Recommended Download:

MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

Other Downloads:

Windows (x86, 64-bit), MSI Installer (mysql-workbench-community-8.0.32-winx64.msi)	8.0.32	45.7M	Download
		MD5: b515207df1b3187965eaaccb46a26e3a	Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

MySQL :: Begin Your Download

dev.mysql.com/downloads/file/?id=516912

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

[Login »](#)
using my Oracle Web account

[Sign Up »](#)
for an Oracle Web account

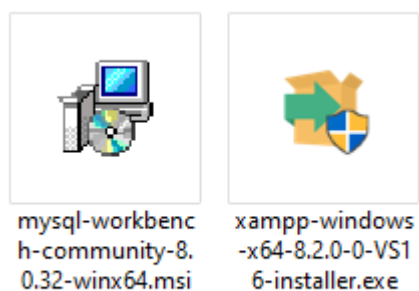
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

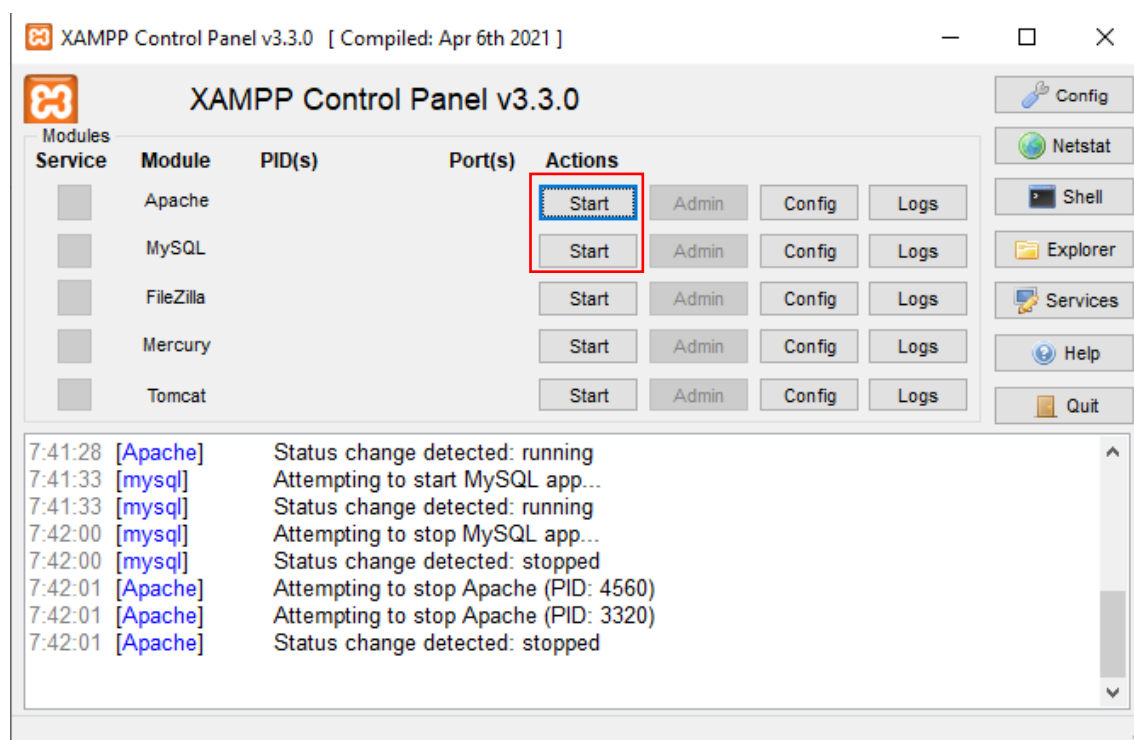
ORACLE © 2023 Oracle

[Privacy / Do Not Sell My Info](#) | [Terms of Use](#) | [Trademark Policy](#) | [Preferencias sobre cookies](#)

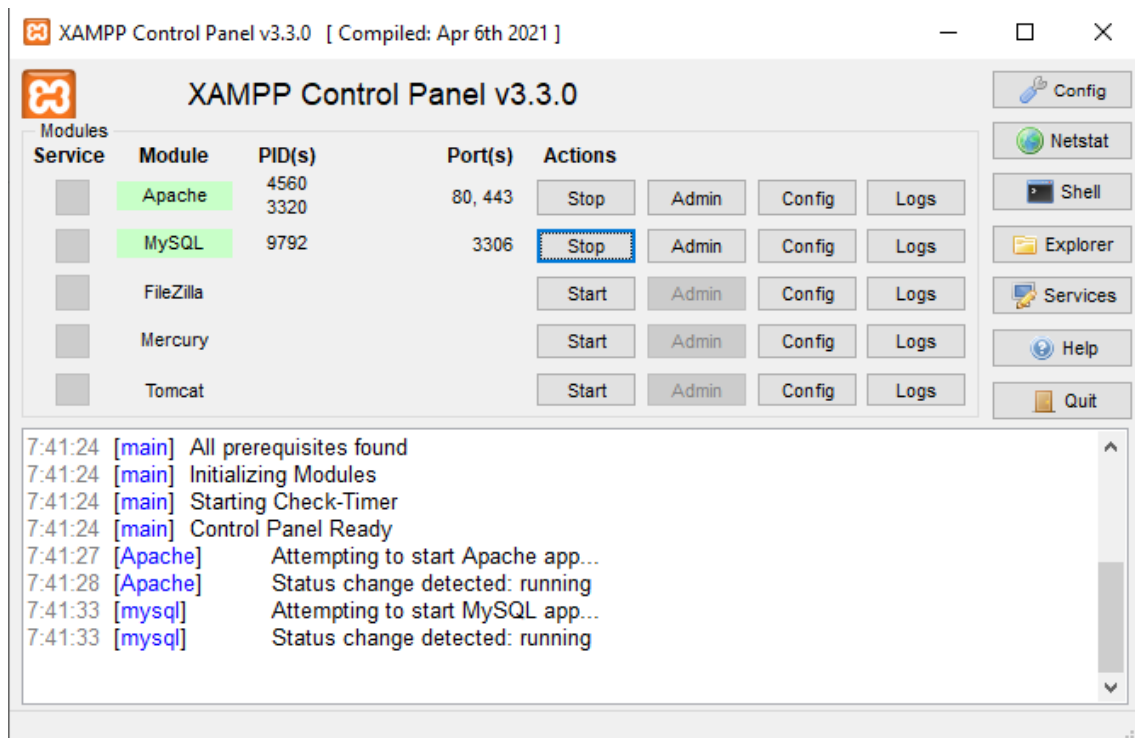
Ya tenemos las descargas.



Una vez instalado xampp lo ejecutamos.

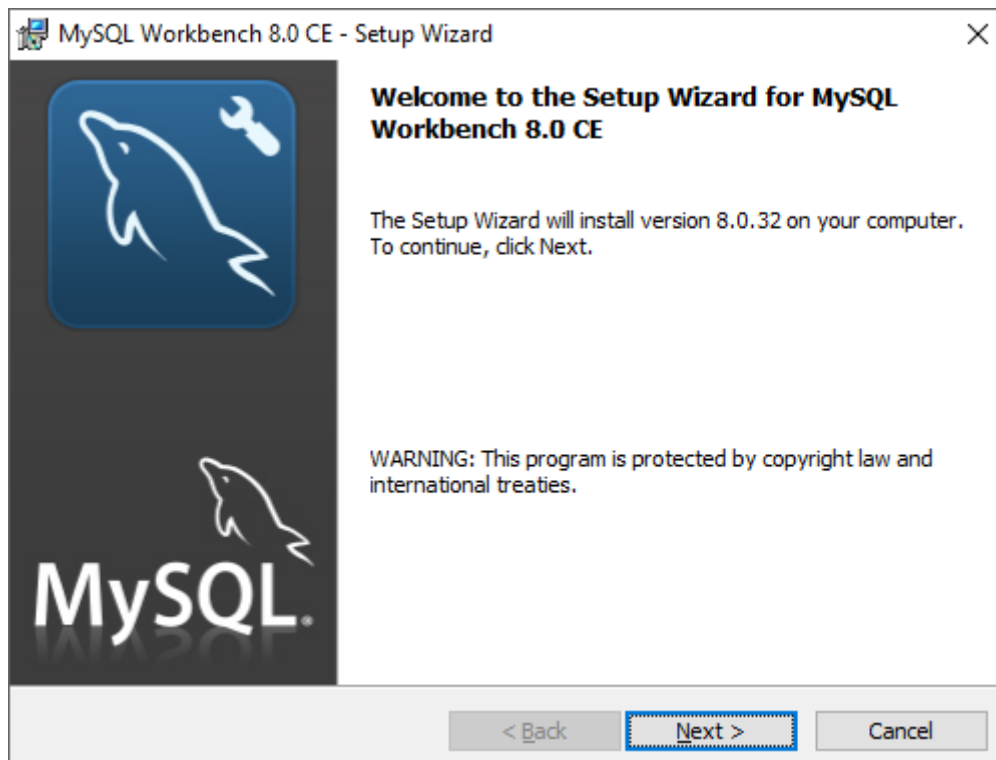


Ejecutamos Apache y MySQL.

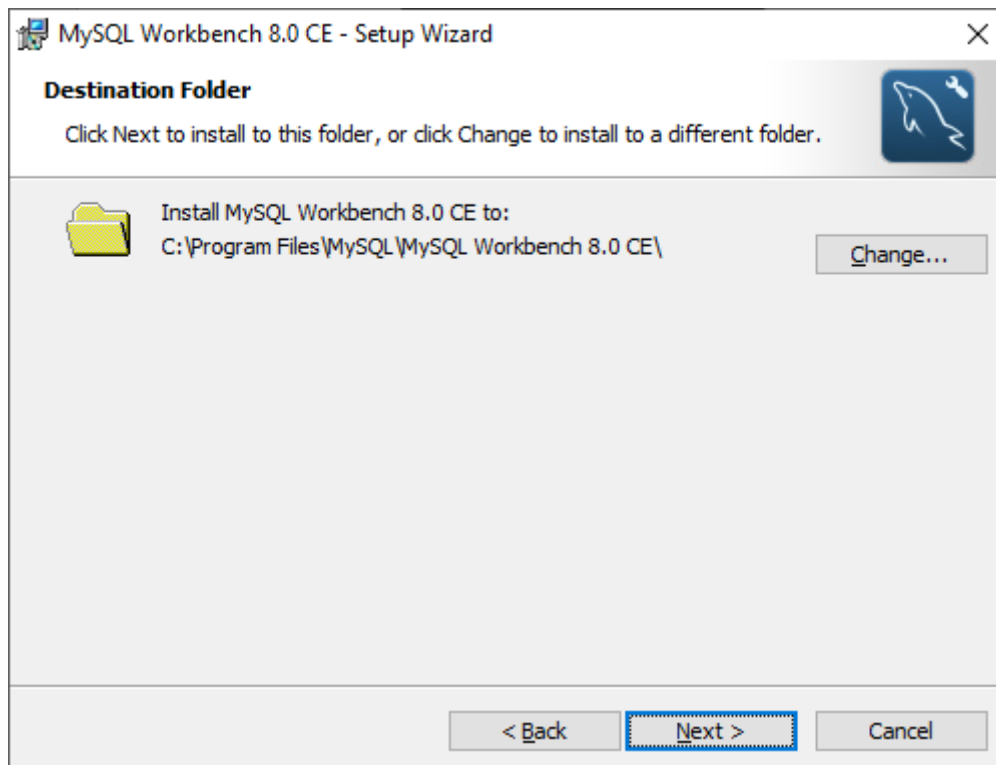


Ya está instalado Apache y MySQL.

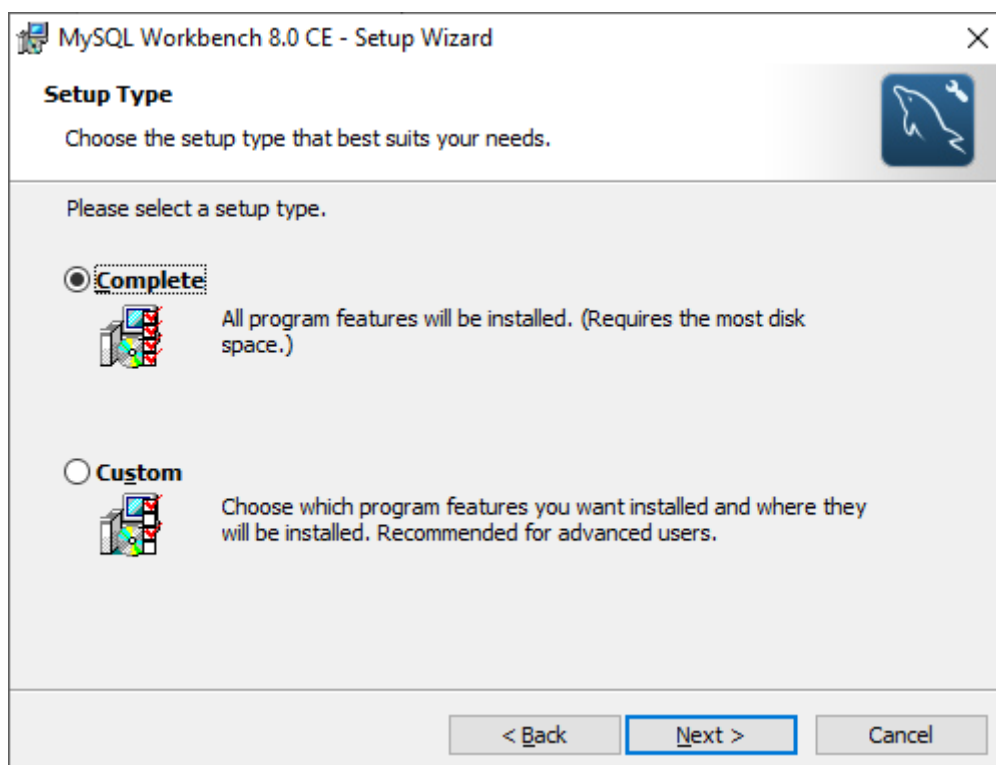
Vamos a instalar mysql workbench.



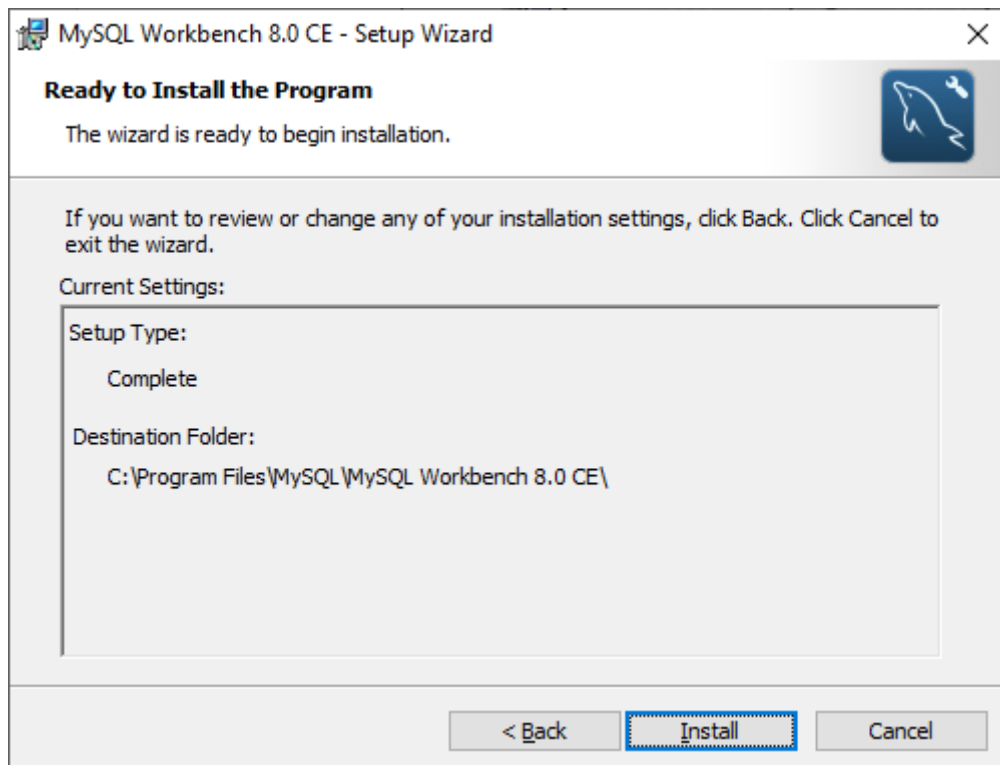
Le damos a Next.



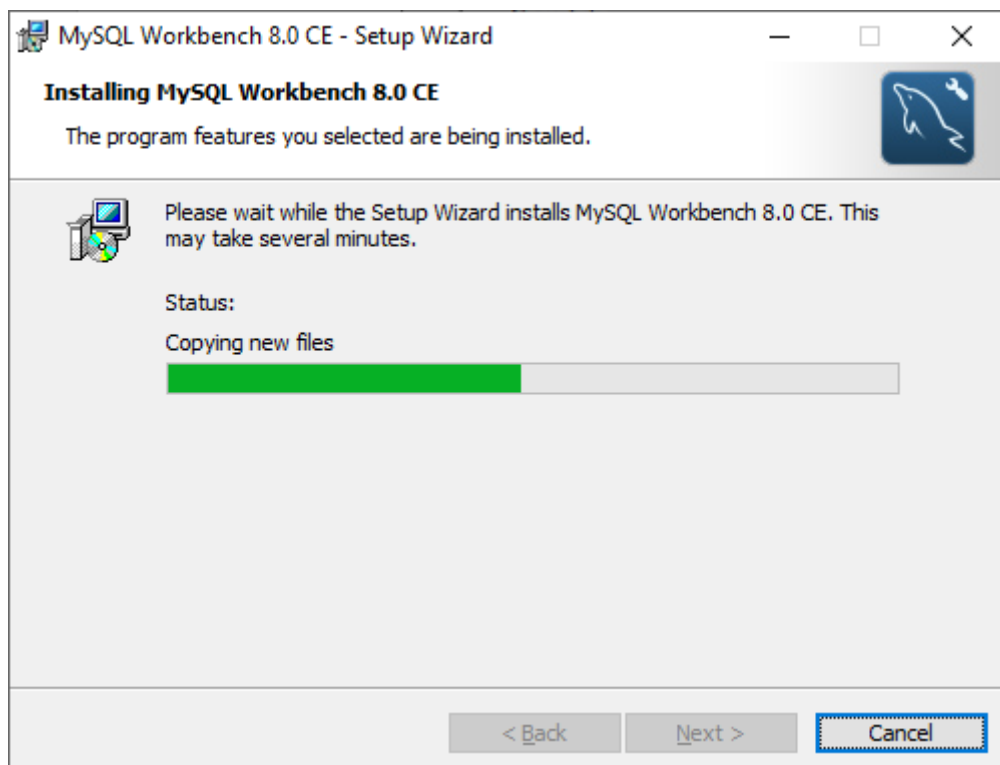
Le damos a Next.

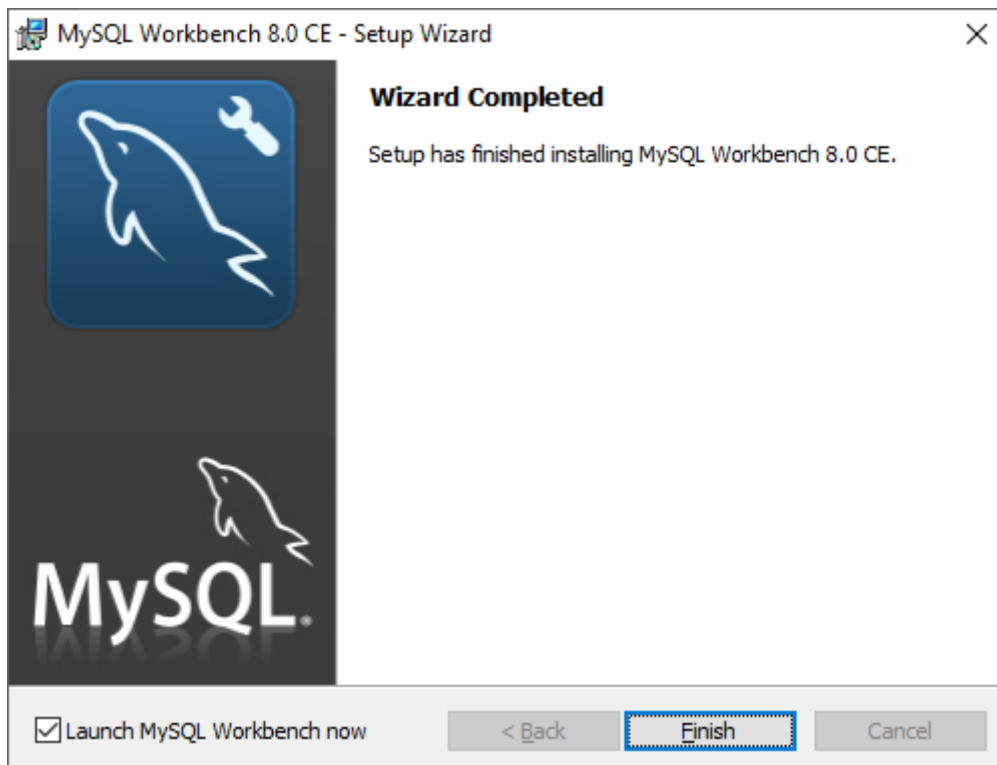


Que realice la instalación completa, seguido de Next.

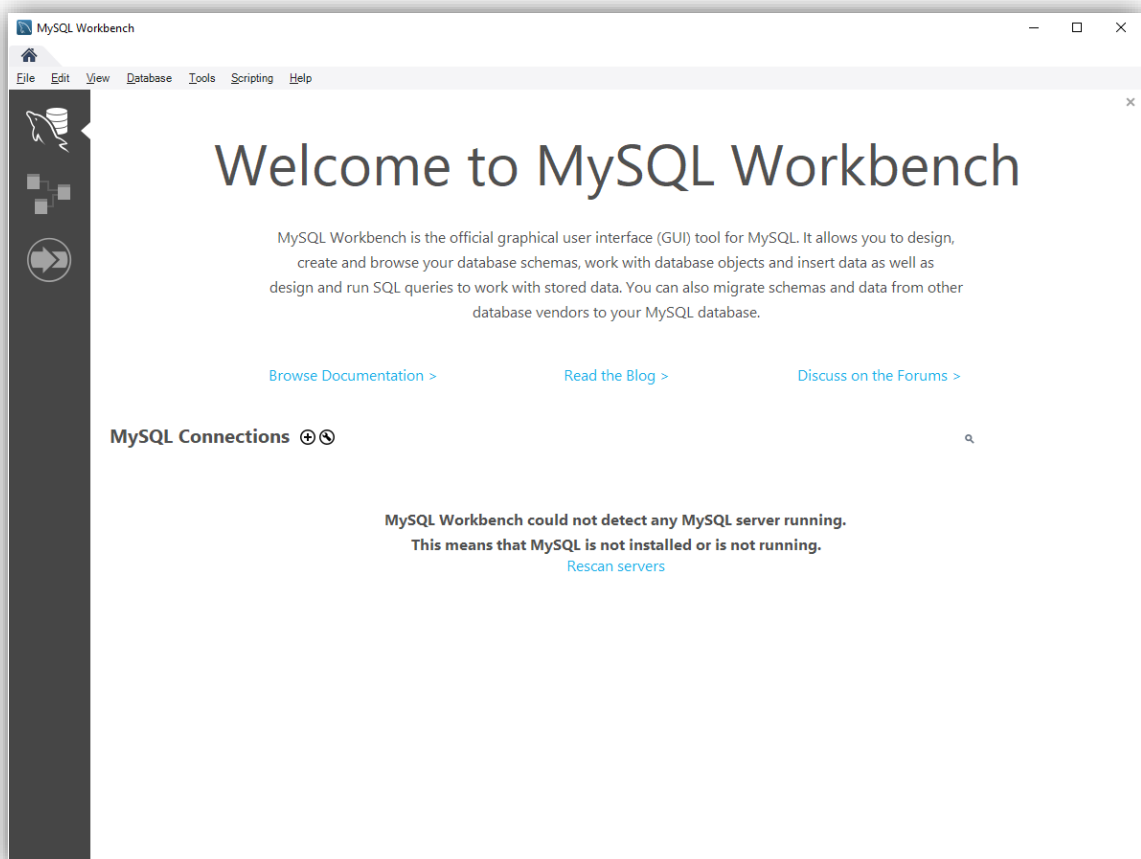


Le damos a Install.





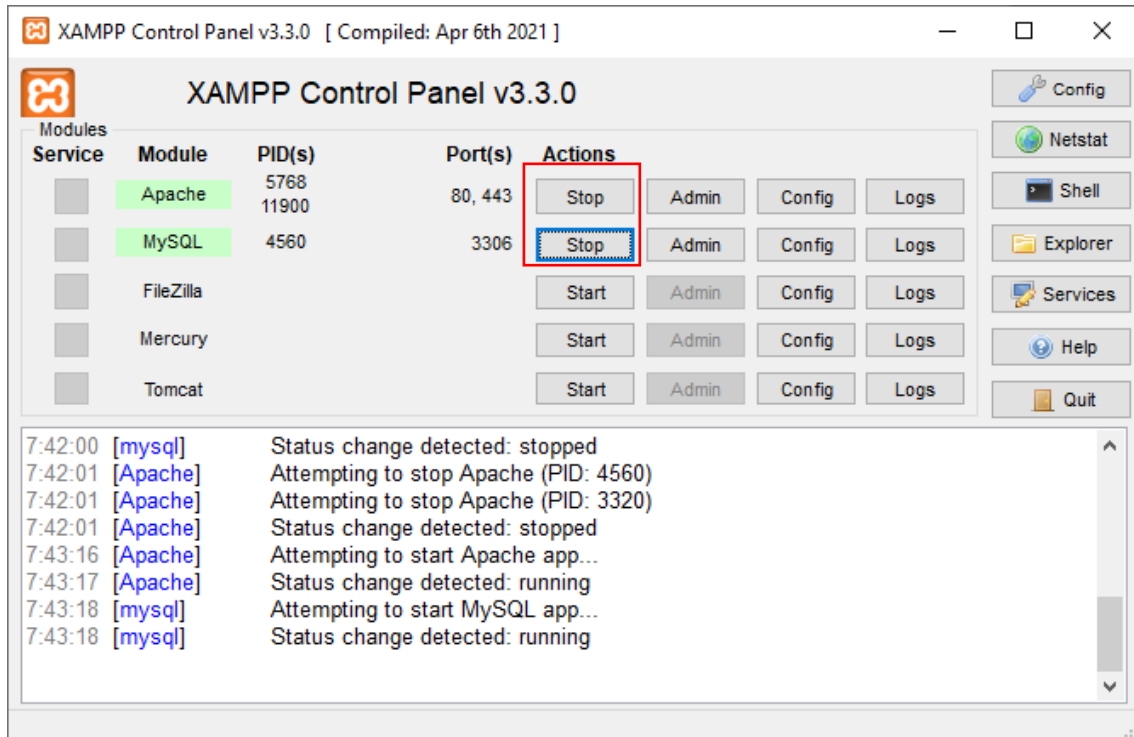
Seleccionamos Finish.



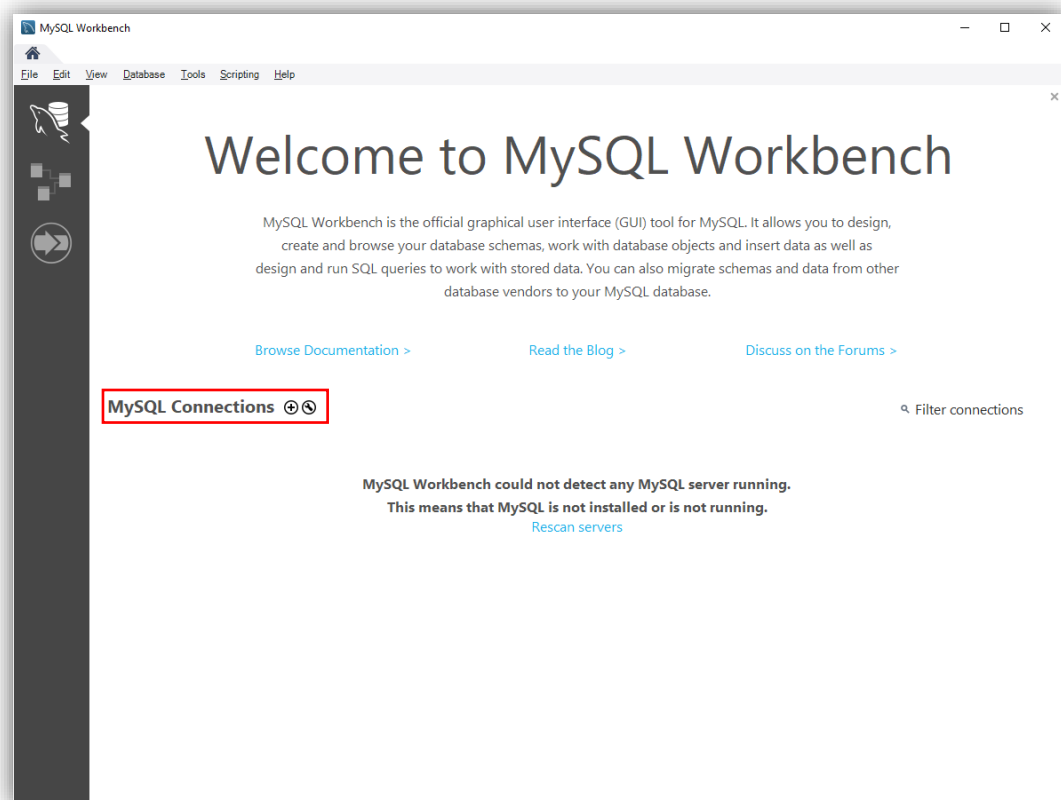
Donde podemos realizar las conexiones, nuestros modelos, estamos preparados para comenzar el curso.

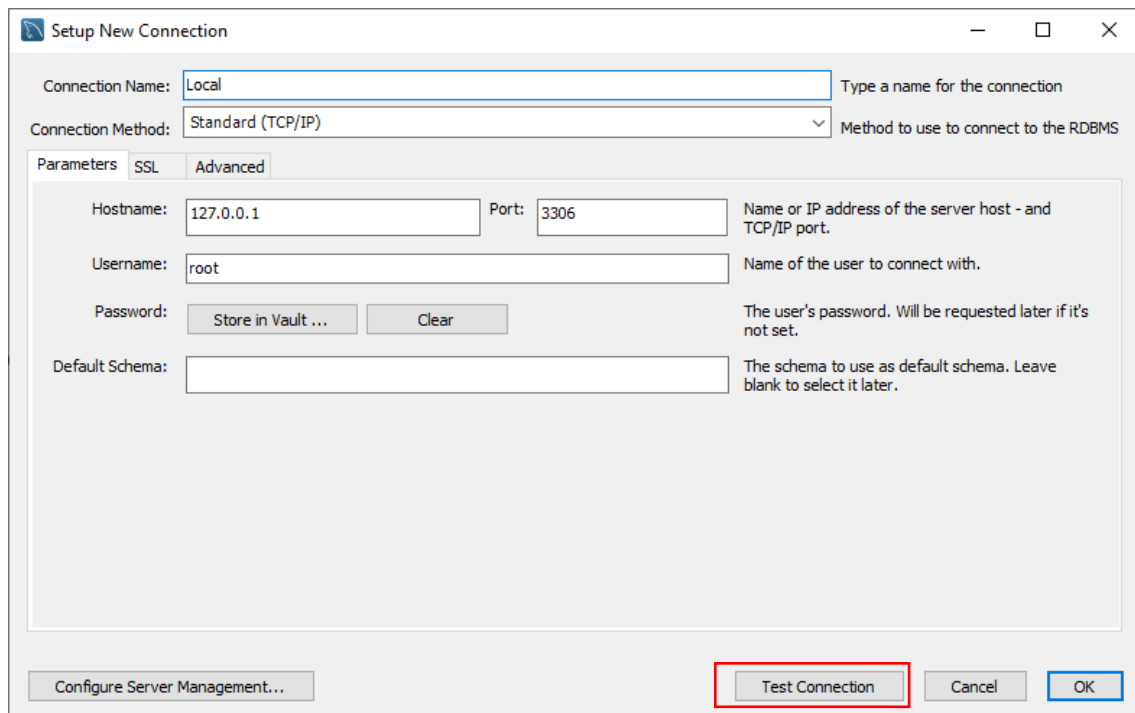
3.- Como Conectar XAMPP y MySQL Workbench

Abrimos nuestro servidor Apache

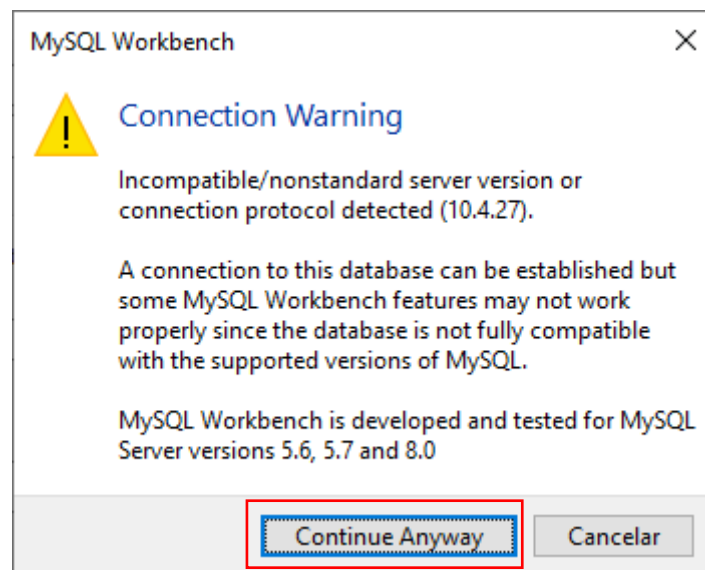


Vamos a iniciar Apache y MySQL y minimizamos la ventana.

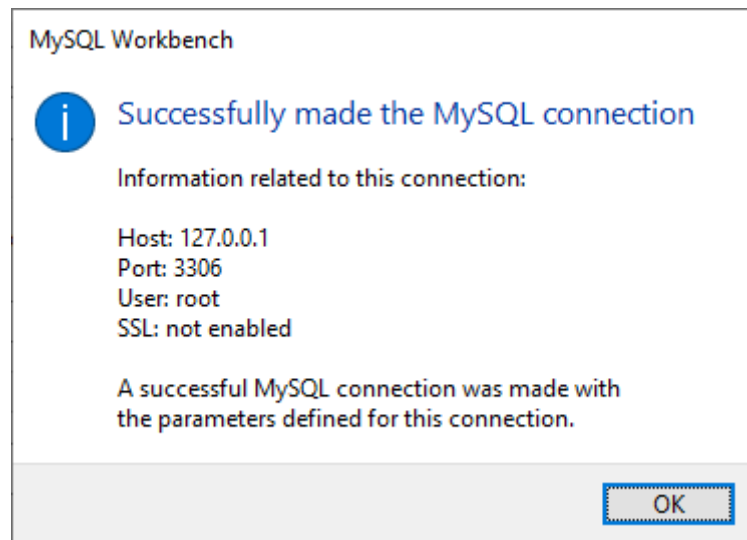




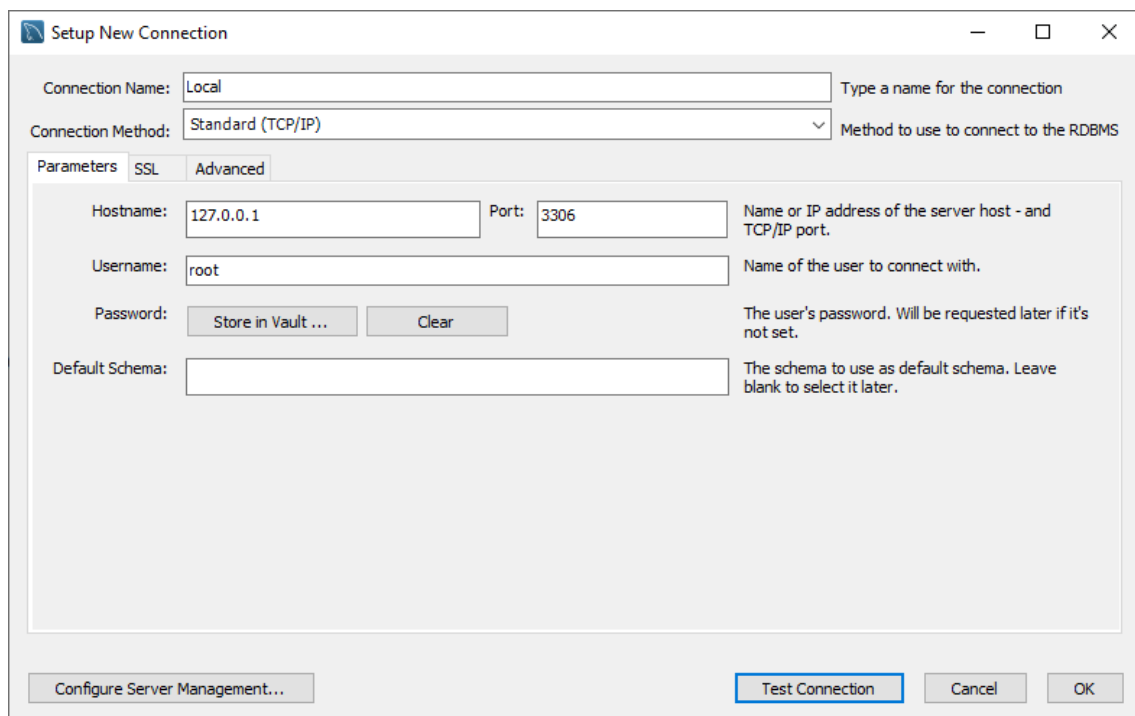
Como nombre Connection Name le pondremos local y a continuación seleccionaremos el botón Text Connection.



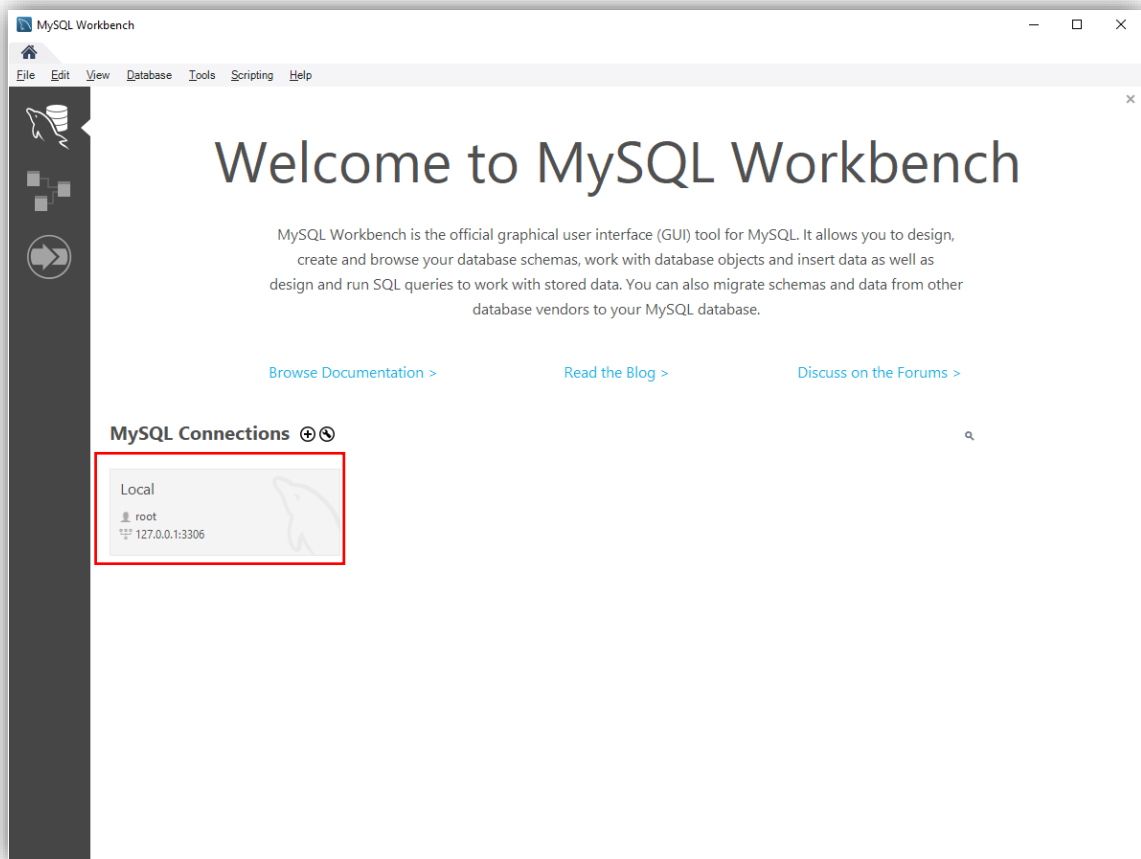
Le damos en Continue.



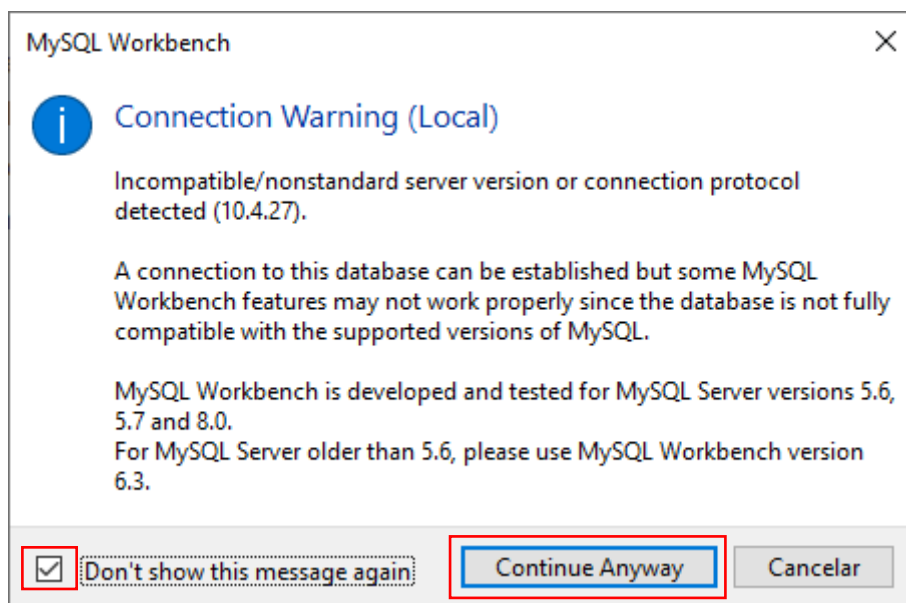
Le damos a OK.



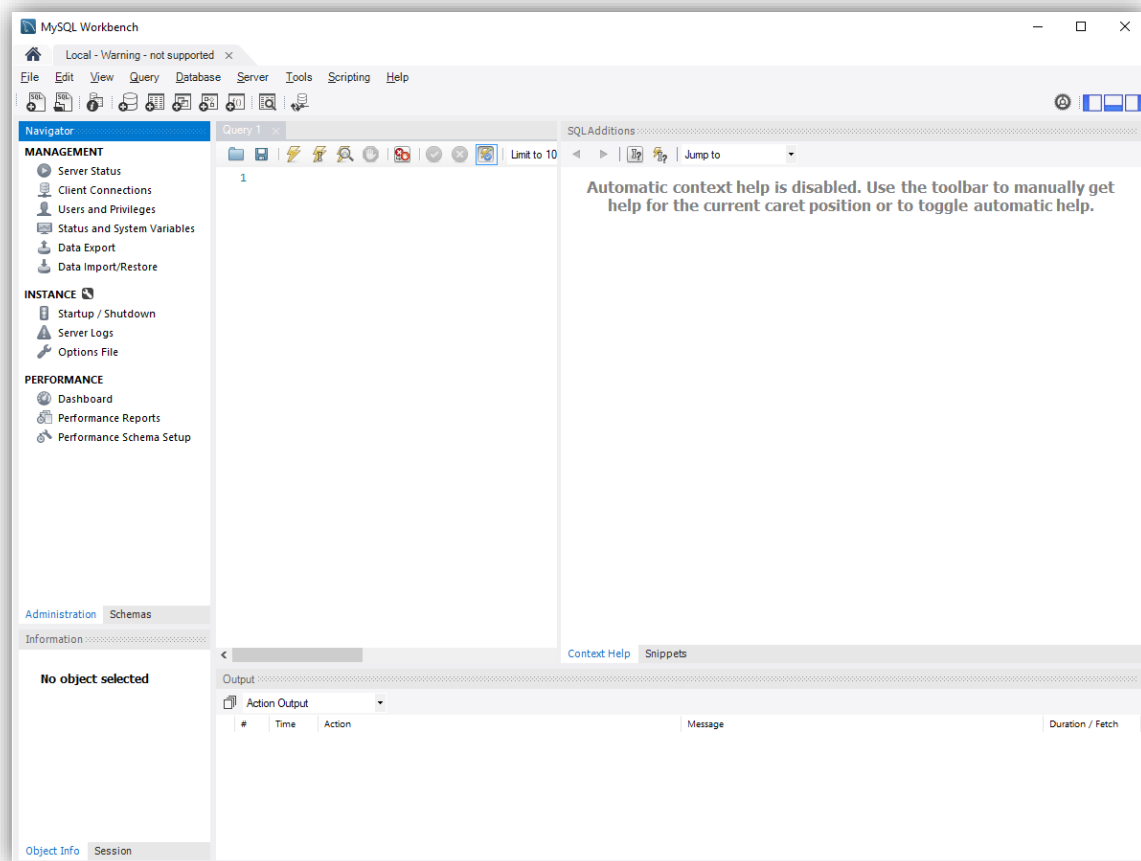
Le damos a OK.



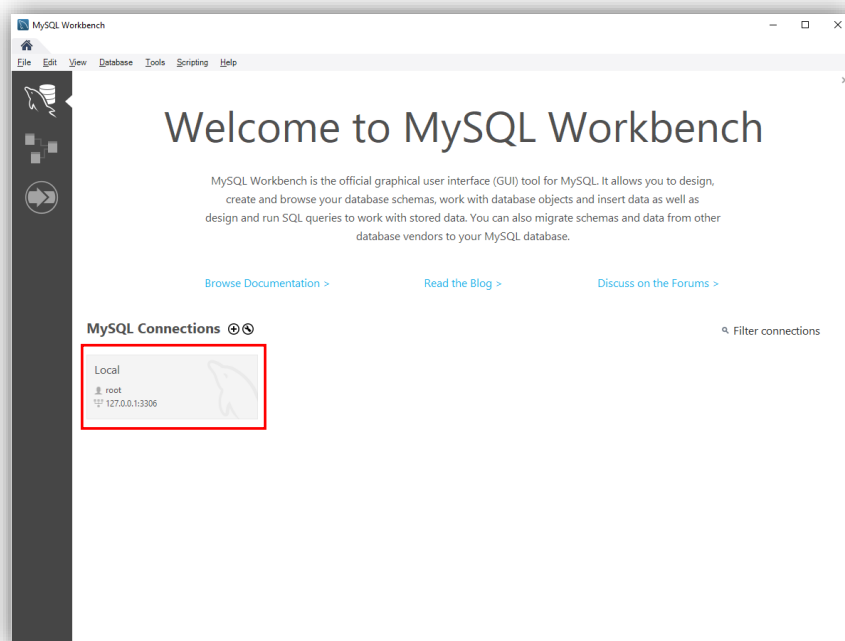
Seleccionamos la nueva conexión.



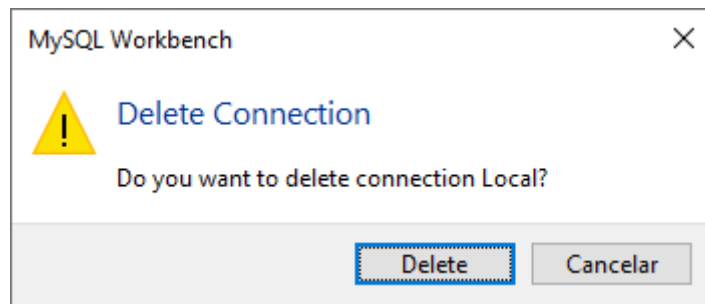
No queremos que salga de nuevo este mensaje y le damos a continuar.



Ya estamos listos para empezar el curso. Si tienen algún problema al configurar el servidor. Cerramos la aplicación sql y la volvemos a abrir.



Botón derecho y eliminar esta conexión.

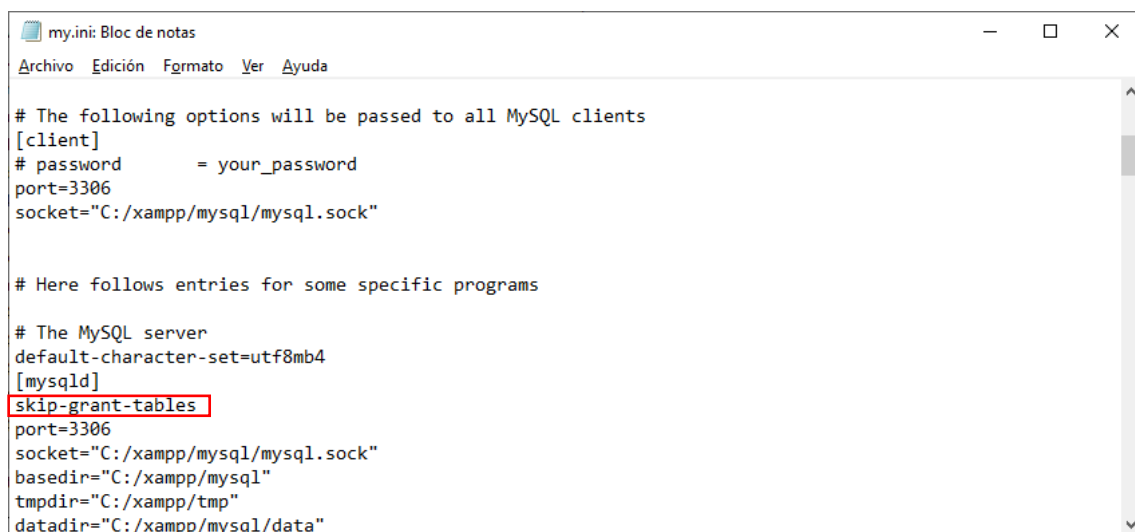


Le damos a Delete.

Nos vamos a donde tenemos instalado XAMPP.

Vamos a la siguiente ruta: C:\xampp\mysql\bin

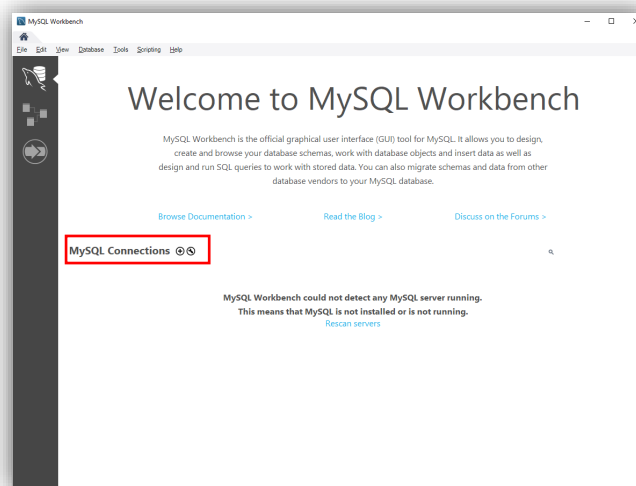
Y abrimos el archivo my.



Agregamos esta línea guardamos el archivo.

En Xampp paramos la conexión y volvemos a conectar.

Vamos a crear de nuevo la conexión.



Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

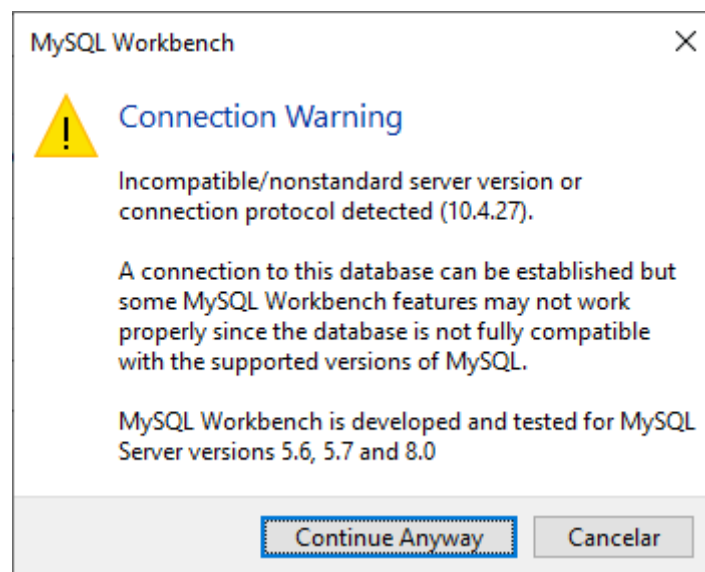
Username: Name of the user to connect with.

Password: Store in Vault ... Clear The user's password. Will be requested later if it's not set.

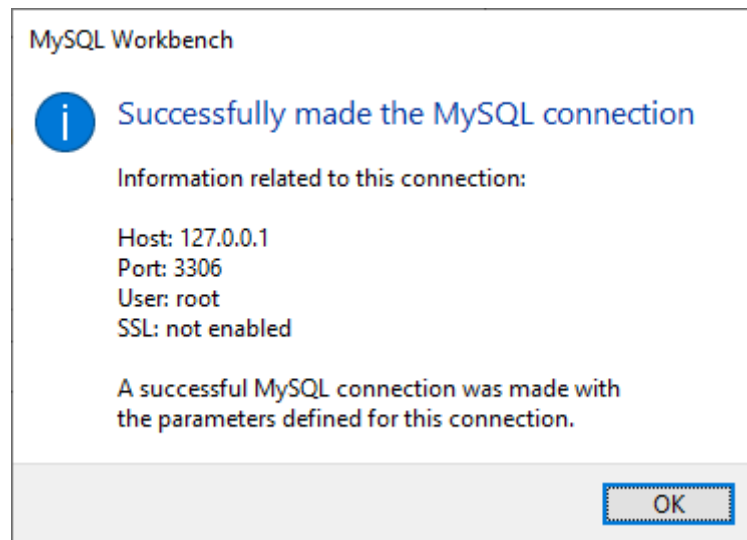
Default Schema: The schema to use as default schema. Leave blank to select it later.

Configure Server Management... Test Connection Cancel OK

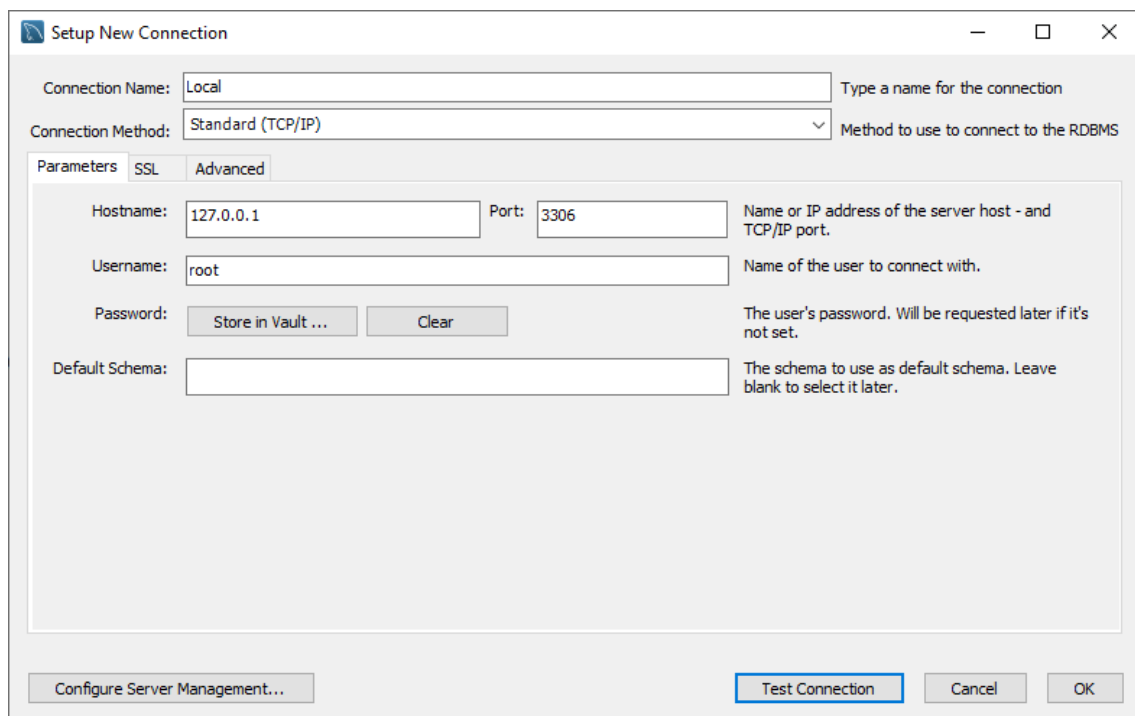
Le damos a Test Connection.



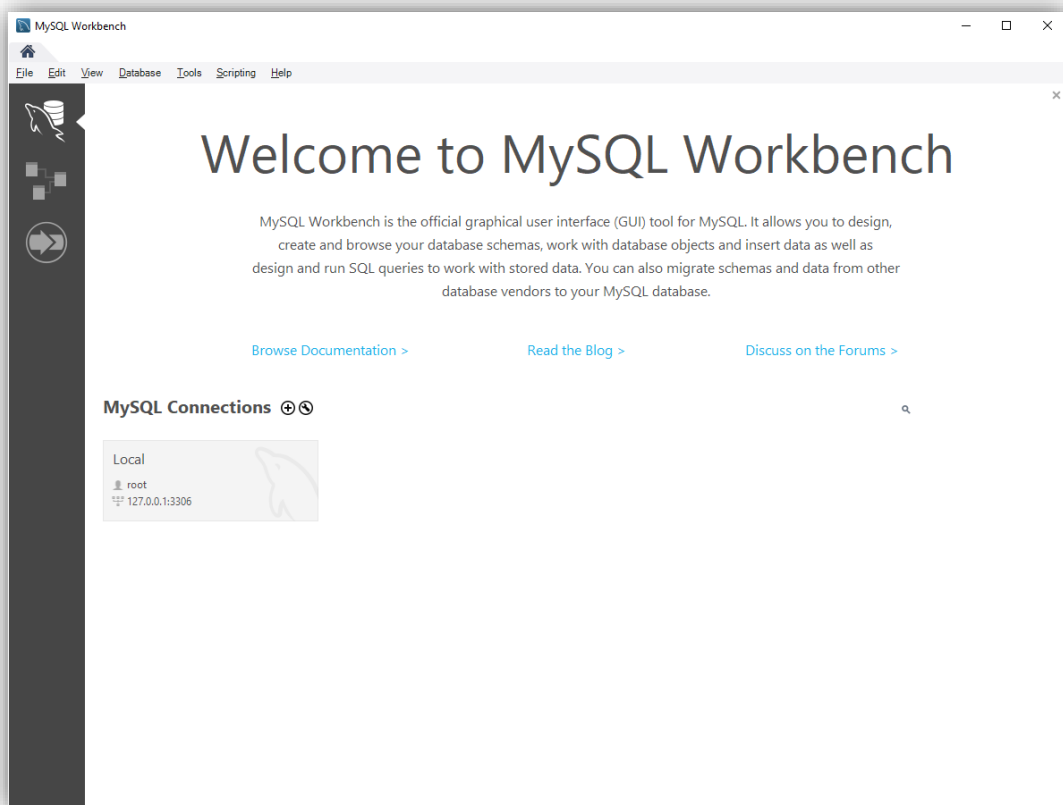
Nos avisa de una advertencia con el servidor, le damos a Continue.



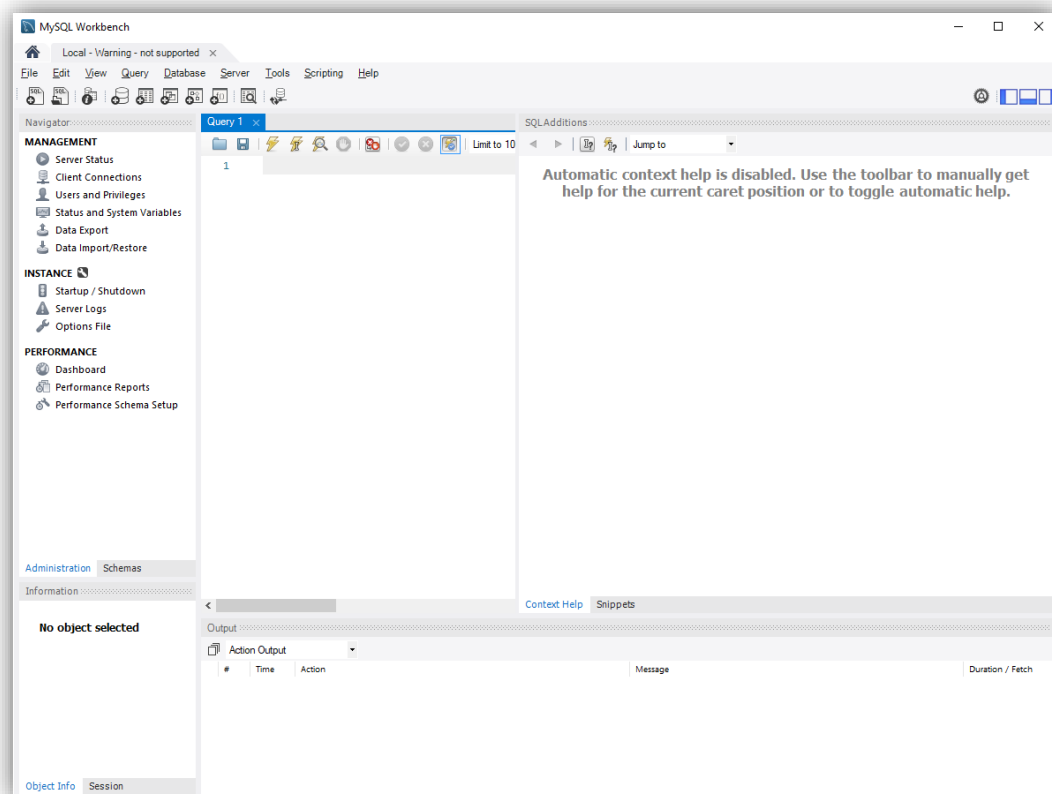
Le damos a OK.



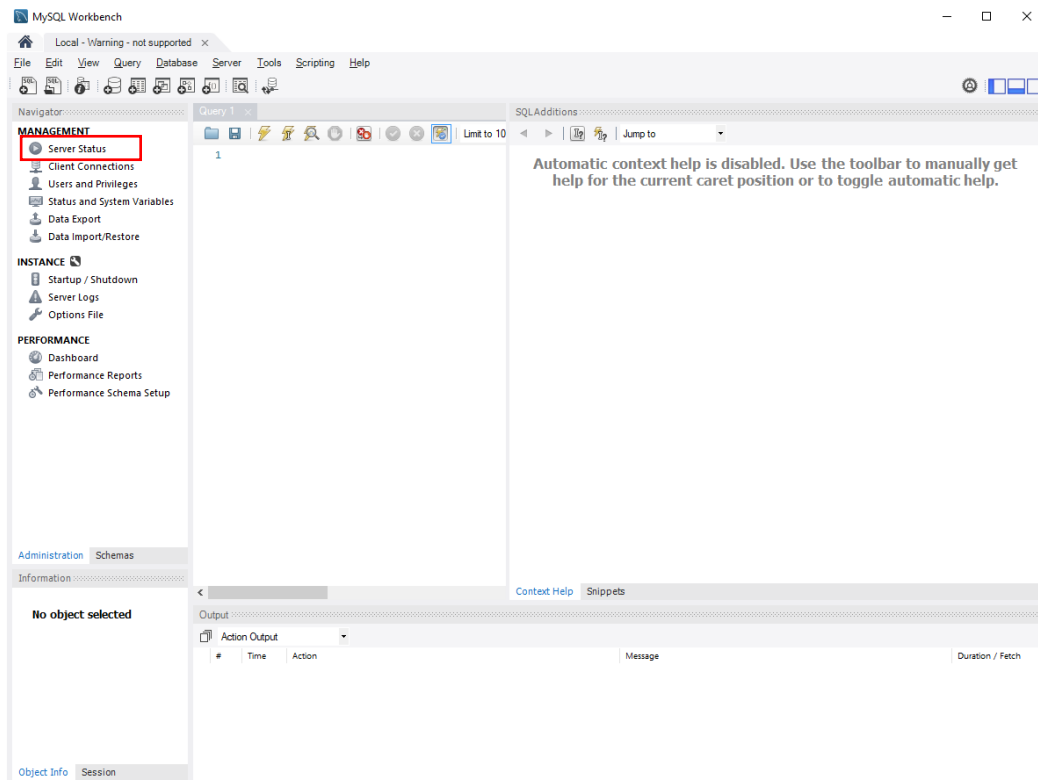
Le damos de nuevo a OK.



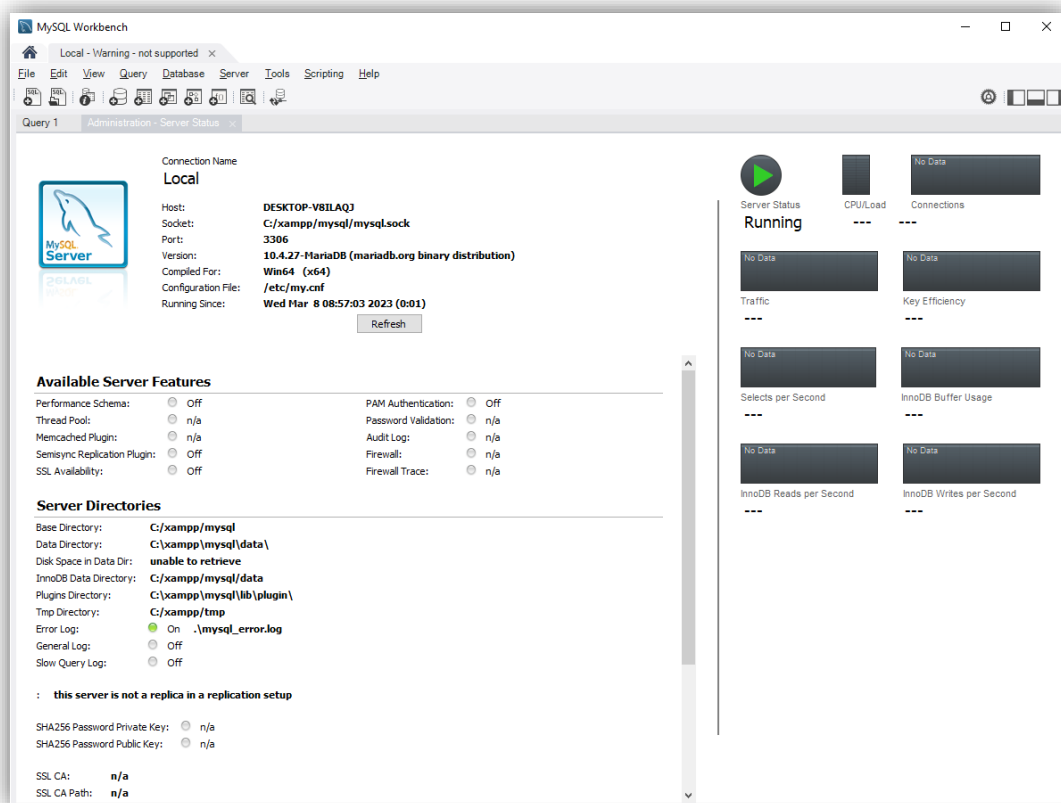
Ya tenemos creado nuestro servidor.



En los próximos capítulos ya vamos a empezar a programar en SQL, empezaremos con comandos básicos.



Seleccionaremos estado del servidor.



Si tienes problemas de conexión cambia la siguiente configuración:

Manage Server Connections

MySQL Connections

Local

Connection Name: Local

Connection Remote Management System Profile

Information about the server and MySQL configuration, such as path to the configuration file, command to start or stop it etc. You may pick a preset configuration profile or customize one for your needs.

System Type: Linux

Installation Type: CentOS 7 (MySQL Package)

Configuration File: /etc/my.cnf

Configuration File Section: mysqld

The following options specify how Workbench should perform certain management actions. Commands given here are the same as if they would be run in a system shell.

MySQL Management

Start MySQL: systemctl start mysqld.service

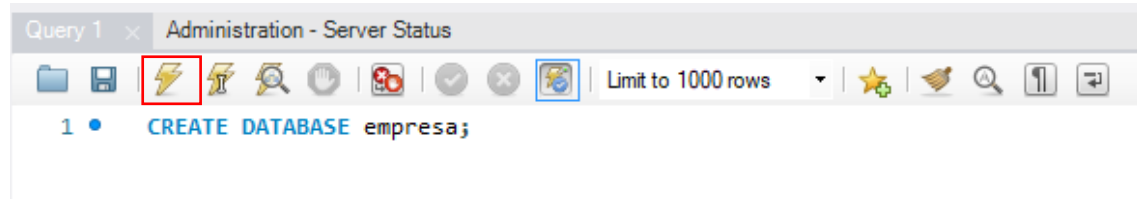
Stop MySQL: systemctl stop mysqld.service

☐ Acquire administrator rights to execute start/stop commands and write configuration data

New Delete Duplicate Move Up Move Down Test Connection Close

4.- Como crear una base de datos

Vamos a crear la base de datos.

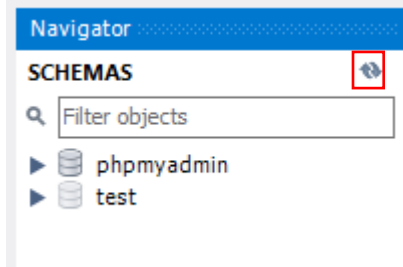


Para ejecutar le damos al rayo.

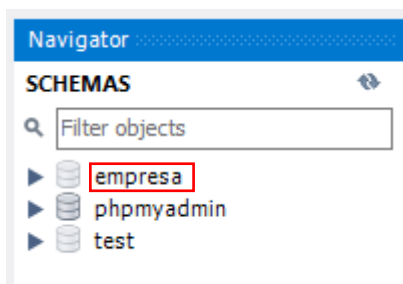
✓ 1 09:09:01 CREATE DATABASE empresa

Si le quitamos el punto y coma funciona igual y es porque solo hay un comando, si hubiera más de un comando hay que separarlos con el punto y coma.

Para ver la base de datos que hemos creado tenemos que actualizar.



Después de actualizar.

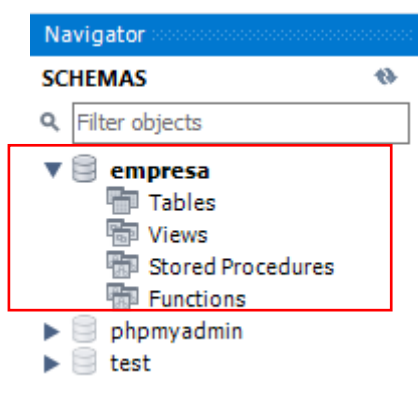


Vamos utilizar el comando use empresa.

1 USE EMPRESA;

Le estamos diciendo que queremos usar la base de datos EMPRESA.

Le damos a ejecutar.



Ahora muestra la estructura de la base de datos empresa, estamos listos a realizar comandos SQL y que afecten a esta base de datos.

5.- Como Crear un Tabla en una base de datos

Para crear una tabla vamos a utilizar los comandos CREATE TABLE nombreTabla y vamos a crear los campos.

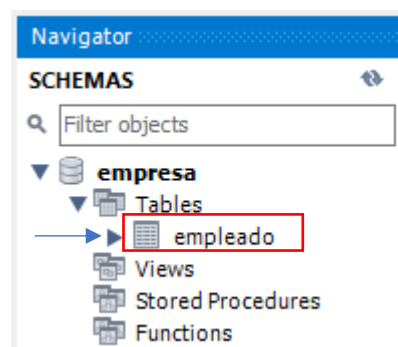
```
CREATE TABLE(id INTEGER,  
             nombre VARCHAR,  
             puesto VARCHAR,  
             );
```

```
1 CREATE TABLE EMPLEADO(idEmpleado integer,  
2                           nombre varchar(20),  
3                           puesto varchar(20));
```

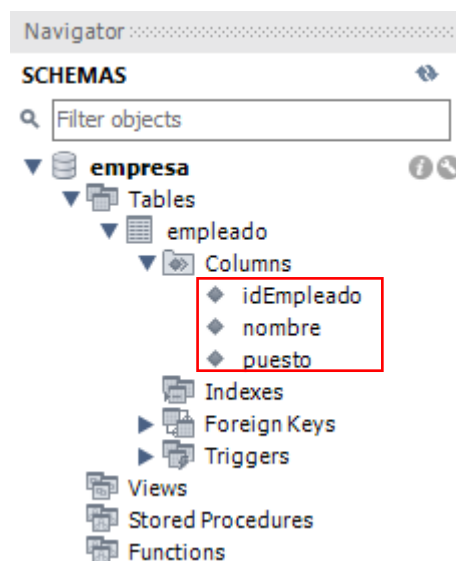
Vamos a ejecutar.

```
✓ 2 10:20:12 CREATE TABLE EMPLEADO(idEmpleado integer, nombre varchar(20), puesto varchar(20))
```

Nos dice que está correcto, si actualizamos.



Vamos a ver la estructura de la tabla.



6.- ¿Qué son las Sentencias de SQL (DML y DDL)?

- Manipulación de datos (DML)
 - SELECT: Recuperar información dentro de la base de datos.
 - INSERT: Para añadir datos en las tablas.
 - DELETE: Para eliminar datos de una tabla.
 - UPDATE: Actualización o modificación de los datos.
- Definición de datos (DDL)
 - CREATE: Crea objetos dentro de la base de datos.
 - DROP: Eliminar objetos de la base de datos.
 - ALTER: Modificación de objetos de la base de datos.

7.- Identificadores de llaves primarias

IDENTIFICADORES o LLAVE PRIMARIA

ID_EMPLEADO	NOMBRE	APELLIDO_P	APELLIDO_M	EDAD	SEXO	SUELDO
001	Daniel	Pérez	Juárez	34	H	\$17,000
002	María	Hernández	Suarez	23	M	\$10,000
003	Erick	López	García	45	H	\$19,000
004	Martha	Pacheco	Ramírez	29	M	\$15,000

A través de este campo nosotros podremos acceder de todos y cada uno de los campos que están dentro de una tabla, si yo quisiera acceder al nombre completo de un empleado, pero solo tengo su identificador de empleado.

Si quiero buscar al empleado que tiene el identificador 003, es para lo que sirve las llaves primarias.

Es importante que en cada una de las tablas exista un identificador, porque este identificador no se va a repetir dentro de la tabla, no puede haber otro identificador con el mismo número por ese motivo no va a ayudar a identificar los datos.

Puede darse el caso de que exista otro empleado que se llame exactamente igual.

IDENTIFICADORES o LLAVE PRIMARIA

ID_EMPLEADO	NOMBRE	APELLIDO_P	APELLIDO_M	EDAD	SEXO	SUELDO
001	Daniel	Pérez	Juárez	34	H	\$17,000
002	María	Hernández	Suarez	23	M	\$10,000
003	Erick	López	García	45	H	\$19,000
004	Martha	Pacheco	Ramírez	29	M	\$15,000
005	Martha	Pacheco	Ramírez	28	M	\$15,000

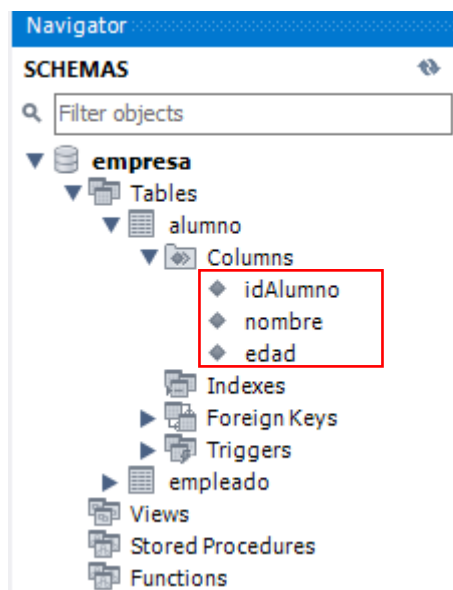
En el supuesto caso de que tengo dos empleados que se llaman exactamente igual, como puedo hacer referencia a una o a otra, pues por el identificador ya que este valor es único y no permite duplicados.

Vamos a crear otra base de datos llamada alumno con los campos idAlumno, nombre y edad, además como clave principal o identificador será el campo idAlumno.

```
1 CREATE TABLE alumno (idAlumno integer PRIMARY KEY,  
2                           nombre varchar(20),  
3                           edad integer);
```

Vamos a ejecutar.

✓ 3 11:00:16 CREATE TABLE alumno (idAlumno integer PRIMARY KEY, nombre varchar(20), edad integer)



Si seleccionamos el campo IdEmpleado de la tabla empleado esta será la información.

Column: idEmpleado
Definition:
idEmpleado int(11)

Ahora vamos a seleccionar el campo IdAlumno de la tabla alumno esta será la información.

Column: idAlumno
Definition:
idAlumno int(11) PK

PK → Significa Primary key.

8.- Tipos de datos

Un tipo de dato no es más que la clasificación de los datos, supongamos que estamos dentro de una tienda y vamos a la sección de frutas, encontramos una caja de manzanas donde dentro hay manzanas, al lado hay una caja de naranjas que dentro hay naranjas, así como otra caja de limones donde dentro hay limones.

Esto es lo que sucede con los datos, cada dato tiene un tipo de dato específico, en la tipología con las frutas no vamos a poner un limón en la caja de las naranjas porque este no es su sitio.

Pues los datos son de un tipo específico:

TIPOS:

- INT → Tipos de datos enteros.
- FLOAT → Tipo de datos flotante datos con decimales.
- VARCHAR → Tipo de datos caracteres simples como cadena de caracteres.
- DATE → Tipo de datos almacenará fecha en distintos formatos.

231	→ INT
"HOLA MUNDO"	→ VARCHAR
3.5	→ FLOAT
23/12/2018	→ DATE
"13 de Febrero"	→ VARCHAR
23 de Marzo del 2018	→ DATE
34	→ INT
23.56	→ FLOAT

9.- Insertar Datos con INSERT

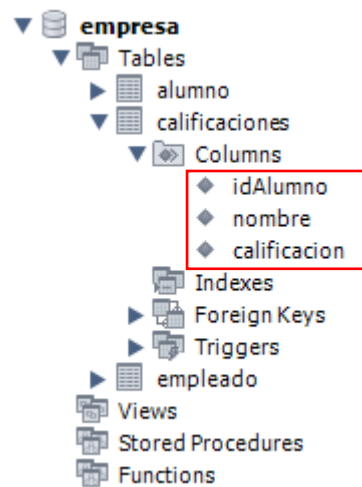
En este capítulo vamos ver como insertar datos en una tabla.

Pero primero vamos a crear una nueva tabla llamada Calificaciones.

```
1 CREATE TABLE Calificaciones (idAlumno integer PRIMARY KEY,  
2                               nombre varchar(20),  
3                               calificacion float);
```

Le damos a ejecutar.

```
✓ 4 12:10:53 CREATE TABLE Calificaciones (idAlumno integer PRIMARY KEY, nombre varchar(20), calificacion float)
```

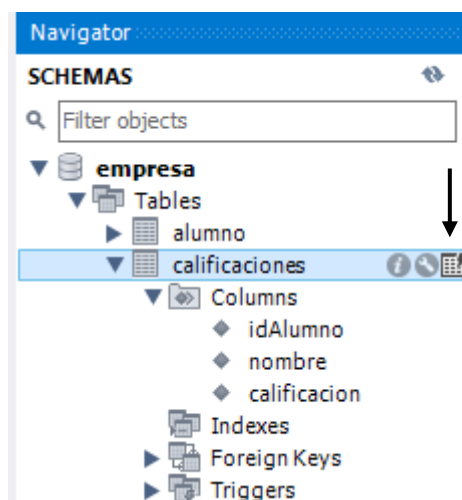


Vamos a insertar datos.

```
1 INSERT INTO calificaciones  
2 VALUES(1, 'Daniel', 9.5);
```

Le damos a ejecutar.

```
✓ 5 12:20:03 INSERT INTO calificaciones VALUES(1, 'Daniel', 9.5)
```



Le damos a mostrar tabla.

	idAlumno	nombre	calificacion
▶	1	Daniel	9.5
✱	NULL	NULL	NULL

Me muestra el registro que hemos introducido.

Agregamos otro registro:

```
1 • INSERT INTO calificaciones
2     VALUES(2, 'Ruben', 7.5);
```

Ejecutamos.

✓ 7 12:26:51 INSERT INTO calificaciones VALUES(2, 'Ruben', 7.5)

Lo hemos realizado con éxito.

Vamos a consultar la tabla:

	idAlumno	nombre	calificacion
▶	1	Daniel	9.5
	2	Ruben	7.5
✱	NULL	NULL	NULL

Ya tenemos el segundo alumno registrado.

10.- Como obtener todos los datos en SEECT

Antes vamos a ingresar otro alumno.

```
1 • INSERT INTO calificaciones
2     VALUES(3, 'ROBERTO', 10);
```

Ejecutamos.

```
✓ 9 12:32:43 INSERT INTO calificaciones VALUES(3, 'ROBERTO', 10)
```

Vamos a seleccionar todos los datos de una tabla.

```
1 • SELECT *
2     FROM calificaciones;
```

Ejecutamos.

	idAlumno	nombre	calificacion
▶	1	Daniel	9.5
	2	Ruben	7.5
	3	ROBERTO	10
*	NULL	NULL	NULL

11.- Como obtener un campo con SELECT

En este capítulo solo queremos saber los nombres de los alumnos que hay en la tabla.

```
1 • SELECT nombre
2 FROM calificaciones;
```

Le damos a ejecutar.

	nombre
▶	Daniel
	Ruben
	Roberto

Ahora queremos además ver su idAlumno.

```
1 • SELECT idAlumno, nombre
2 FROM calificaciones;
```

Le damos a ejecutar.

	idAlumno	nombre
▶	1	Daniel
	2	Ruben
	3	Roberto

12.- UPDATE – Actualizar datos

Vamos a ver como actualizar registros de las tablas de la base de datos.

Vamos a realizar una consulta para ver todo los datos de la tabla.

```
1 • SELECT *  
2 FROM calificaciones;
```

Le damos a ejecutar.

	idAlumno	nombre	calificacion
▶	1	Daniel	9.5
	2	Ruben	7.5
	3	Roberto	10

Vamos a añadir otro alumno.

```
1 • INSERT INTO calificaciones  
2 VALUES(4, 'Pedro', 5.6)
```

Vamos a ejecutar:

```
1 • SELECT *  
2 FROM calificaciones;
```

Vamos a consultar que este ultimo registro se ha añadido.

	idAlumno	nombre	calificacion
▶	1	Daniel	9.5
	2	Ruben	7.5
	3	Roberto	10
	4	Pedro	5.6

Nos hemos dado cuenta que la calificación de Pedro no es 5.6 sino 6.5 vamos a modificarlo.

```
1 • UPDATE calificaciones  
2 SET calificacion=6.5  
3 WHERE idAlumno=4;
```

Ejecutamos.

✓ 19 13:16:26 UPDATE calificaciones SET calificacion=6.5 WHERE idAlumno=4

Ahora hacemos una consulta para ver todos los campos y registros de la tabla calificaciones.

```
1 • SELECT * FROM calificaciones;
```

Ejecutamos.

	idAlumno	nombre	calificacion
▶	1	Daniel	9.5
	2	Ruben	7.5
	3	Roberto	10
	4	Pedro	6.5

Ya hemos actualizado el registro de Pedro.

Ahora quiero modificar a Daniel que tiene una calificación de 9.5 por Daniela.

```
1 • UPDATE calificaciones
2   SET nombre='Daniela'
3   WHERE idAlumno=1;
```

Le damos a ejecutar.

✓ 21 13:23:37 UPDATE calificaciones SET nombre='Daniela' WHERE idAlumno=1

Vamos a ver el cambio.

```
1 • SELECT * FROM calificaciones;
```

Vamos a ejecutar.

	idAlumno	nombre	calificacion
▶	1	Daniela	9.5
	2	Ruben	7.5
	3	Roberto	10
	4	Pedro	6.5

Hemos cambiado a Daniel por Daniela.

13.- DELETE – Eliminar datos

Con el comando DELETE podremos borrar registros.

Vamos a ver los registros de la tabla calificaciones.

```
1 • SELECT * FROM calificaciones
```

Ejecutamos.

	idAlumno	nombre	calificacion
▶	1	Daniela	9.5
	2	Ruben	7.5
	3	Roberto	10
	4	Pedro	6.5

El comando DELETE no nos va a permitir eliminar el valor de un campo específico, lo que va a eliminar es un registro completo.

```
1 • DELETE FROM calificaciones
2 WHERE idAlumno=3;
```

Vamos a ejecutar.

```
✓ 3 14:59:09 DELETE FROM calificaciones WHERE idAlumno=3
```

Vamos hacer si hemos eliminado a Roberto.

```
1 • SELECT * FROM calificaciones
```

Vamos a ejecutar.

	idAlumno	nombre	calificacion
▶	1	Daniela	9.5
	2	Ruben	7.5
	4	Pedro	6.5

Ya hemos eliminado a Roberto.

Vamos a agregar más registros.

```
1 • INSERT INTO calificaciones
2 VALUES(3, 'María', 7.6);
3 • INSERT INTO calificaciones
4 VALUES(5, 'Eduardo', 9.6);
```

Le damos a ejecutar.

```
✓ 5 15:07:28 INSERT INTO calificaciones VALUES(3, 'María', 7.6)
✓ 6 15:07:28 INSERT INTO calificaciones VALUES(5, 'Eduardo', 9.6)
```

Vamos a ver la tabla.

```
1 • SELECT * FROM calificaciones;
```

Vamos a ejecutar.

	idAlumno	nombre	calificacion
▶	1	Daniela	9.5
	2	Ruben	7.5
	3	María	7.6
	4	Pedro	6.5
	5	Eduardo	9.6

Vamos a cambiar la calificación de María a 7.5.

```
1 • UPDATE calificaciones
2   SET calificacion=7.5
3   WHERE idAlumno=3;
```

Vamos a ejecutar.

```
✓ 8 15:12:14 UPDATE calificaciones SET calificacion=7.5 WHERE idAlumno=3
```

Vamos a ver la tabla.

```
1 • SELECT * FROM calificaciones;
```

Ejecutamos.

	idAlumno	nombre	calificacion
▶	1	Daniela	9.5
	2	Ruben	7.5
	3	María	7.5
	4	Pedro	6.5
	5	Eduardo	9.6

Tenemos dos alumnos con la misma calificación.

Ahora queremos eliminar aquellos registros cuyos alumnos han obtenido la calificación de 7.5.

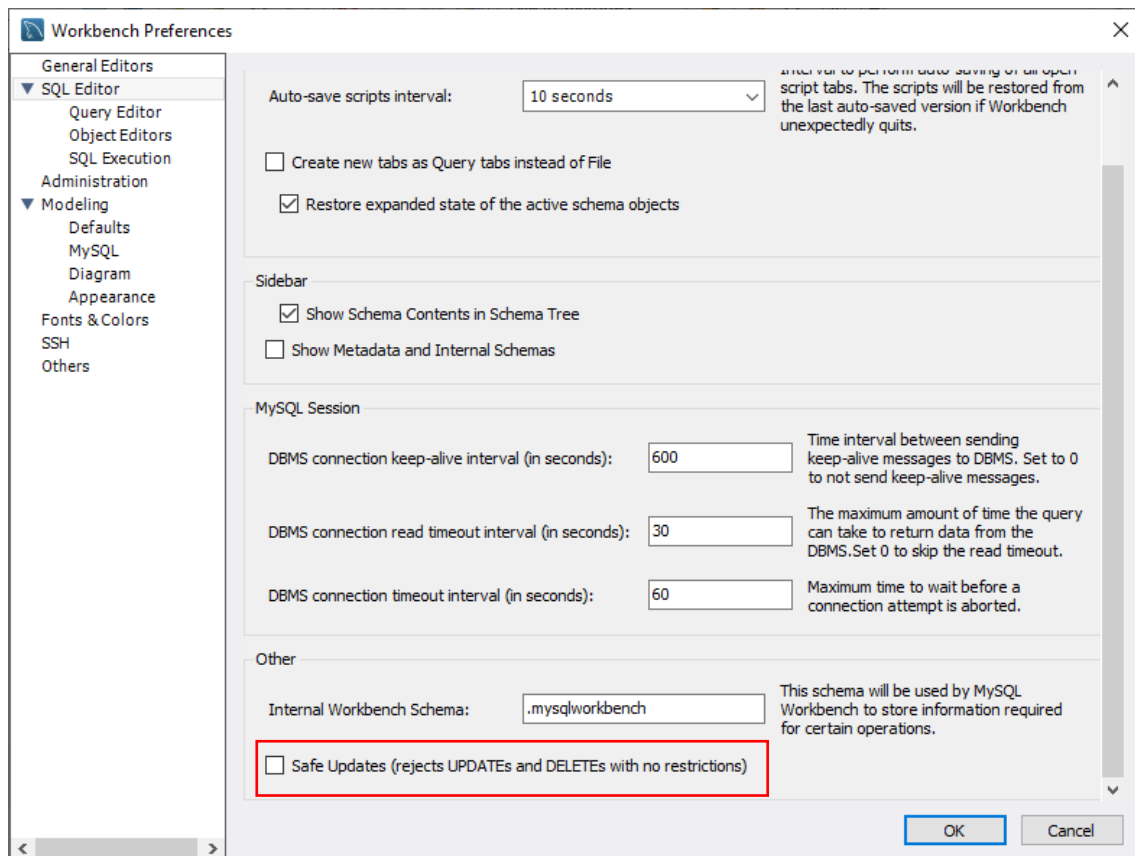
```
1 • DELETE FROM calificaciones
2   WHERE calificacion=7.5;
```

Le damos a ejecutar.

```
1 • DELETE FROM calificaciones
2   WHERE calificacion=7.5;
```

Al ejecutar me está dando error y no hay error de sintaxis, lo que pasa es que la poner como condición un campo que no es clave y evitar errores a la ejecución de esta sentencia, el programa impide su ejecución, pero como queremos eliminarlos de igual modo vamos a cambiar las preferencias del programa.

Del menú Edición seleccionamos Preferencias, en la opción SQL Editor desactivamos la opción remarcada en la siguiente imagen.



Ahora tenemos que cerrar la conexión y volver a entrar.

Una vez hemos seleccionado Ok, vamos a repetir la misma sentencia SQL.

```
1 • DELETE FROM calificaciones
2   WHERE calificacion=7.5;
3
4 • SELECT * FROM calificaciones;
```

Ejecutamos.

	idAlumno	nombre	calificacion
▶	1	Daniela	9.5
	4	Pedro	6.5
	5	Eduardo	9.6

14.- Comentarios

Los comentarios no es más herramienta que nosotros vamos a poder utilizar como programadores para agregar notas a nuestro código y así podernos recordar de lo que estamos haciendo o identificar una parte del código para saber lo que está realizando o que función tiene esta parte de código. No va a ser visto por el usuario final.

```
1 • SELECT idAlumno, calificacion
2   FROM calificaciones;
3
4   -- Comentario
5   # Comentario
6   /* Para iniciar el comentario
7   Para finalizar el comentario*/
```

Si queremos un comentario en una línea junto al código.

```
1 • SELECT idAlumno, calificacion -- Esto es un comentario
2   FROM calificaciones;
```

Otra forma de agregar comentario en la misma línea de código.

```
1 • SELECT idAlumno, calificacion # Esto es un comentario
2   FROM calificaciones;
```

Otra forma de comentario dentro del código.

```
1 • SELECT idAlumno, calificacion # Esto es un comentario
2   FROM /* Comentario */calificaciones;
```


15.- Operadores

Los operadores nos van a ayudar a poner condiciones dentro de nuestras consultas, para poder consultar determinados datos, es decir nosotros queremos consultar de la tabla calificaciones aquellas que tengan una calificación de 9.5.

```
1 • SELECT idAlumno, calificacion
2   FROM calificaciones
3  WHERE calificacion=9.5;
```

Mediante el WHERE ponemos la condición.

Si lo ejecutamos.

	idAlumno	calificacion
▶	1	9.5

Ahora ya estamos utilizando un operador ¿Cuál es? = de igual.

También hay otros operadores que nos van a ayudar a realizar determinadas consultas.

¿Qué operadores son?

=	Igual
<	Menor que
>	Mayor que
<=	Menor igual que
>=	Mayor igual que
!=	Diferente según gestor base de datos
<>	Diferente según gestor base de datos

<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion<9;</pre>	<table><tr><th></th><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>▶</td><td>4</td><td>Pedro</td><td>6.5</td></tr></table>		idAlumno	nombre	calificacion	▶	4	Pedro	6.5				
	idAlumno	nombre	calificacion										
▶	4	Pedro	6.5										
Mostrar las calificaciones menores de 9.													
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion>9;</pre>	<table><tr><th></th><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>▶</td><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td></td><td>5</td><td>Eduardo</td><td>9.6</td></tr></table>		idAlumno	nombre	calificacion	▶	1	Daniela	9.5		5	Eduardo	9.6
	idAlumno	nombre	calificacion										
▶	1	Daniela	9.5										
	5	Eduardo	9.6										
Mostrar las calificaciones mayores de 9.													

Vamos a agregar más valores.

```
1 • INSERT INTO calificaciones VALUE(6, 'Marco', 7);
2 • INSERT INTO calificaciones VALUE(7, 'María', 7.8);
3 • INSERT INTO calificaciones VALUE(8, 'Ruben', 6.9);
4 • INSERT INTO calificaciones VALUE(9, 'Daniel', 9);
5 • INSERT INTO calificaciones VALUE(10, 'Daniela', 8.5);
6
7 • SELECT * FROM calificaciones
```

Y al final que nos muestre todos los registros.

idAlumno	nombre	calificacion
1	Daniela	9.5
4	Pedro	6.5
5	Eduardo	9.6
6	Marco	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5

<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion<=9</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>6</td><td>Marco</td><td>7</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr></table>	idAlumno	nombre	calificacion	4	Pedro	6.5	6	Marco	7	7	María	7.8	8	Ruben	6.9	9	Daniel	9	10	Daniela	8.5			
idAlumno	nombre	calificacion																							
4	Pedro	6.5																							
6	Marco	7																							
7	María	7.8																							
8	Ruben	6.9																							
9	Daniel	9																							
10	Daniela	8.5																							
Mostrar las calificaciones menores o igual a 9																									
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion>=9;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	5	Eduardo	9.6	9	Daniel	9												
idAlumno	nombre	calificacion																							
1	Daniela	9.5																							
5	Eduardo	9.6																							
9	Daniel	9																							
Mostrar las calificaciones mayores o igual a 9																									
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion!=9;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>6</td><td>Marco</td><td>7</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	4	Pedro	6.5	5	Eduardo	9.6	6	Marco	7	7	María	7.8	8	Ruben	6.9	10	Daniela	8.5
idAlumno	nombre	calificacion																							
1	Daniela	9.5																							
4	Pedro	6.5																							
5	Eduardo	9.6																							
6	Marco	7																							
7	María	7.8																							
8	Ruben	6.9																							
10	Daniela	8.5																							
Mostrar las calificaciones que son distintas a 9																									
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion<>9;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>6</td><td>Marco</td><td>7</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	4	Pedro	6.5	5	Eduardo	9.6	6	Marco	7	7	María	7.8	8	Ruben	6.9	10	Daniela	8.5
idAlumno	nombre	calificacion																							
1	Daniela	9.5																							
4	Pedro	6.5																							
5	Eduardo	9.6																							
6	Marco	7																							
7	María	7.8																							
8	Ruben	6.9																							
10	Daniela	8.5																							
Mostrar las calificaciones son distintas a 9																									

Funcionan los dos operadores de distintos.

16.- Operador LIKE

Para que sirve este operador para aplicarlos en algún tipo de cadena. Supongamos que tenemos un apellido como Ramírez, López, Ruiz, entre otro con la sentencia LIKE se intenta localizar coincidencias.

Si queremos localizar todos los apellidos que empiezan por R será LIKE 'R%'.

Si queremos localizar todos los apellidos que termina por Z será LIKE '%Z'.

Si queremos localizar todos los apellidos que por parte del centro tiene MI será LIKE '%MI%'.

```
1 • SELECT *
2 FROM calificaciones
```

Vamos a ver el contenido de la tabla.

idAlumno	nombre	calificacion
1	Daniela	9.5
4	Pedro	6.5
5	Eduardo	9.6
6	Marco	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5

<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE nombre LIKE 'M%';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>6</td><td>Marco</td><td>7</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr></table>	idAlumno	nombre	calificacion	6	Marco	7	7	María	7.8									
idAlumno	nombre	calificacion																	
6	Marco	7																	
7	María	7.8																	
Queremos consultar por los nombres que empiezan por M.																			
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE nombre LIKE '%o';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>6</td><td>Marco</td><td>7</td></tr></table>	idAlumno	nombre	calificacion	4	Pedro	6.5	5	Eduardo	9.6	6	Marco	7						
idAlumno	nombre	calificacion																	
4	Pedro	6.5																	
5	Eduardo	9.6																	
6	Marco	7																	
Queremos consultar por los apellidos que terminan por o.																			
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE nombre LIKE '%D%';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	4	Pedro	6.5	5	Eduardo	9.6	9	Daniel	9	10	Daniela	8.5
idAlumno	nombre	calificacion																	
1	Daniela	9.5																	
4	Pedro	6.5																	
5	Eduardo	9.6																	
9	Daniel	9																	
10	Daniela	8.5																	
Queremos consultar por los nombre que tienen en alguna parte la letra d.																			
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE nombre LIKE '%NI%';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	9	Daniel	9	10	Daniela	8.5						
idAlumno	nombre	calificacion																	
1	Daniela	9.5																	
9	Daniel	9																	
10	Daniela	8.5																	
Queremos consultar por los apellidos que lleven la NI.																			

<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion LIKE '%9%';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	5	Eduardo	9.6	8	Ruben	6.9	9	Daniel	9						
idAlumno	nombre	calificacion																				
1	Daniela	9.5																				
5	Eduardo	9.6																				
8	Ruben	6.9																				
9	Daniel	9																				
Queremos consultar por la calificación que contenga el número 9.																						
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion LIKE '%.%';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	4	Pedro	6.5	5	Eduardo	9.6	7	María	7.8	8	Ruben	6.9	10	Daniela	8.5
idAlumno	nombre	calificacion																				
1	Daniela	9.5																				
4	Pedro	6.5																				
5	Eduardo	9.6																				
7	María	7.8																				
8	Ruben	6.9																				
10	Daniela	8.5																				
Queremos consultar por las calificaciones que tengan un punto.																						

El operador LIKE no diferencia entre mayúsculas y minúsculas.

17.- Valor NULL

Vamos a agregar varios registros:

- 1 • `INSERT INTO calificaciones VALUES(2, NULL, 7);`
- 2 • `INSERT INTO calificaciones VALUES(3, 'Javier', 10);`
- 3 • `SELECT * FROM calificaciones`

Este será el resultado:

idAlumno	nombre	calificacion
1	Daniela	9.5
2	NULL	7
3	Javier	10
4	Pedro	6.5
5	Eduardo	9.6
6	Marco	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5

- 1 • `SELECT *`
- 2 `FROM calificaciones`
- 3 `WHERE nombre IS NULL;`

idAlumno	nombre	calificacion
2	NULL	7

Vamos a consultar por los nombres que estén vacíos.

- 1 • `SELECT *`
- 2 `FROM calificaciones`
- 3 `WHERE nombre IS NOT NULL;`

idAlumno	nombre	calificacion
1	Daniela	9.5
3	Javier	10
4	Pedro	6.5
5	Eduardo	9.6
6	Marco	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5

Vamos a consultar por los nombres que no están vacíos.

18.- Comando DISTINCT

```
1 • SELECT * FROM calificaciones;
```

Vamos a ver el contenido de la tabla calificaciones.

idAlumno	nombre	calificacion
1	Daniela	9.5
2	NULL	7
3	Javier	10
4	Pedro	6.5
5	Eduardo	9.6
6	Marco	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5

Vamos a agregar registros:

```
1 • INSERT INTO calificaciones VALUES(11, 'Marcos', 10);
2 • INSERT INTO calificaciones VALUES(12, 'Pedro', 8);
3 • SELECT * FROM calificaciones;
```

idAlumno	nombre	calificacion
1	Daniela	9.5
2	NULL	7
3	Javier	10
4	Pedro	6.5
5	Eduardo	9.6
6	Marcos	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5
11	Marcos	10
12	Pedro	8

```
1 • SELECT nombre
2   FROM calificaciones;
```

nombre

Daniela
NULL
Javier
Pedro
Eduardo
Marcos
María
Ruben
Daniel
Daniela
Marcos
Pedro

La sentencia DISTINCT nos va a evitar sacar valores repetidos.

1 •	SELECT DISTINCT nombre	nombre
2	FROM calificaciones;	Daniela
		NULL
		Javier
		Pedro
		Eduardo
		Marcos
		María
		Ruben
		Daniel

Los repetidos no los muestra.

calificacion	1 •	SELECT DISTINCT calificacion	calificacion
9.5	2	FROM calificaciones;	9.5
7			7
10			10
6.5			6.5
9.6			9.6
7			7.8
7.8			6.9
6.9			9
9			8.5
8.5			8
10			
8			

El segundo 7 ya no sale.

19.- Operadores Lógicos AND / OR

Los operadores lógicos se van a utilizar dentro de la sentencia WHERE, lo utilizaremos cuando queremos controlar más de una condición.

1	●	SELECT *	idAlumno	nombre	calificacion
2		FROM calificaciones	1	Daniela	9.5
3		WHERE nombre='Daniela';	10	Daniela	8.5

Quiero consultar los alumnos que se llamen 'Daniela'.

1 •	SELECT *	
2	FROM calificaciones	
3	WHERE nombre='Daniela' AND calificacion>9;	

idAlumno	nombre	calificacion
1	Daniela	9.5

Quiero consultar con Daniela, y que tengan una calificación mayor de 9.

Vamos a agregar un nuevo registro.

```
1 • INSERT INTO calificaciones VALUES(13, 'Daniela', 9.2);
```

```
1 • INSERT INTO calificaciones VALUES(14, 'Daniela', 10);
```

1 •	SELECT *	idAlumno	nombre	calificacion
2	FROM calificaciones	1	Daniela	9.5
3	WHERE nombre='Daniela' AND calificacion>9;	13	Daniela	9.2
		14	Daniela	10

Repetimos la consulta anterior.

1 • `SELECT *`

2 `FROM calificaciones`

3 `WHERE nombre='Pedro' or calificacion<7;`

idAlumno	nombre	calificacion
4	Pedro	6.5
8	Ruben	6.9
12	Pedro	8

Quiero consultar por las personas que se llamen Pedro o han obtenido una calificación menor de 7.

20.- Comando NOT

Con este comando hacemos una negación, lo contrario de la condición.

<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE NOT nombre='María';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>3</td><td>Javier</td><td>10</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>6</td><td>Marcos</td><td>7</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr><tr><td>11</td><td>Marcos</td><td>10</td></tr><tr><td>12</td><td>Pedro</td><td>8</td></tr><tr><td>13</td><td>Daniela</td><td>9.2</td></tr><tr><td>14</td><td>Daniela</td><td>10</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	3	Javier	10	4	Pedro	6.5	5	Eduardo	9.6	6	Marcos	7	8	Ruben	6.9	9	Daniel	9	10	Daniela	8.5	11	Marcos	10	12	Pedro	8	13	Daniela	9.2	14	Daniela	10
idAlumno	nombre	calificacion																																						
1	Daniela	9.5																																						
3	Javier	10																																						
4	Pedro	6.5																																						
5	Eduardo	9.6																																						
6	Marcos	7																																						
8	Ruben	6.9																																						
9	Daniel	9																																						
10	Daniela	8.5																																						
11	Marcos	10																																						
12	Pedro	8																																						
13	Daniela	9.2																																						
14	Daniela	10																																						
Quiero consultar por los alumnos que no se llamen María.																																								
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE nombre='Daniela';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr><tr><td>13</td><td>Daniela</td><td>9.2</td></tr><tr><td>14</td><td>Daniela</td><td>10</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	10	Daniela	8.5	13	Daniela	9.2	14	Daniela	10																								
idAlumno	nombre	calificacion																																						
1	Daniela	9.5																																						
10	Daniela	8.5																																						
13	Daniela	9.2																																						
14	Daniela	10																																						
Quiero consultar por los alumnos que se llaman Daniela.																																								
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE NOT nombre='Daniela';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>3</td><td>Javier</td><td>10</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>6</td><td>Marcos</td><td>7</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>11</td><td>Marcos</td><td>10</td></tr><tr><td>12</td><td>Pedro</td><td>8</td></tr></table>	idAlumno	nombre	calificacion	3	Javier	10	4	Pedro	6.5	5	Eduardo	9.6	6	Marcos	7	7	María	7.8	8	Ruben	6.9	9	Daniel	9	11	Marcos	10	12	Pedro	8									
idAlumno	nombre	calificacion																																						
3	Javier	10																																						
4	Pedro	6.5																																						
5	Eduardo	9.6																																						
6	Marcos	7																																						
7	María	7.8																																						
8	Ruben	6.9																																						
9	Daniel	9																																						
11	Marcos	10																																						
12	Pedro	8																																						
Quiero consultar por los alumnos que no se llaman Daniela.																																								
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE NOT calificacion=10;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>2</td><td>NULL</td><td>7</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>6</td><td>Marcos</td><td>7</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr><tr><td>12</td><td>Pedro</td><td>8</td></tr><tr><td>13</td><td>Daniela</td><td>9.2</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	2	NULL	7	4	Pedro	6.5	5	Eduardo	9.6	6	Marcos	7	7	María	7.8	8	Ruben	6.9	9	Daniel	9	10	Daniela	8.5	12	Pedro	8	13	Daniela	9.2			
idAlumno	nombre	calificacion																																						
1	Daniela	9.5																																						
2	NULL	7																																						
4	Pedro	6.5																																						
5	Eduardo	9.6																																						
6	Marcos	7																																						
7	María	7.8																																						
8	Ruben	6.9																																						
9	Daniel	9																																						
10	Daniela	8.5																																						
12	Pedro	8																																						
13	Daniela	9.2																																						
Quiero consultar a los alumnos que no han obtenido un 10 en calificaciones.																																								
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE NOT nombre='Daniela' 4 AND calificacion=10;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>3</td><td>Javier</td><td>10</td></tr><tr><td>11</td><td>Marcos</td><td>10</td></tr></table>	idAlumno	nombre	calificacion	3	Javier	10	11	Marcos	10																														
idAlumno	nombre	calificacion																																						
3	Javier	10																																						
11	Marcos	10																																						

Quiero consular por aquellos alumnos que no son Daniela y además su calificación ha sido de 10.

```
1 • SELECT *
2 FROM calificaciones
3 WHERE NOT nombre='Daniela'
4 AND NOT calificacion=10;
```

idAlumno	nombre	calificacion
4	Pedro	6.5
5	Eduardo	9.6
6	Marcos	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
12	Pedro	8

Quiero consultar por aquellos alumnos que no son Daniela y además su calificación no ha sido de 10

21.- Comando BETWEEN

La traducción de BETWEEN significa entre.

<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion > 8 4 AND calificacion < 10;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr><tr><td>13</td><td>Daniela</td><td>9.2</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	5	Eduardo	9.6	9	Daniel	9	10	Daniela	8.5	13	Daniela	9.2												
idAlumno	nombre	calificacion																													
1	Daniela	9.5																													
5	Eduardo	9.6																													
9	Daniel	9																													
10	Daniela	8.5																													
13	Daniela	9.2																													
Imaginemos que queremos dar un reconocimiento a todas las personas que sacaron alguna calificación entre 8 y 10.																															
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion 4 BETWEEN 8 AND 10;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>3</td><td>Javier</td><td>10</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr><tr><td>11</td><td>Marcos</td><td>10</td></tr><tr><td>12</td><td>Pedro</td><td>8</td></tr><tr><td>13</td><td>Daniela</td><td>9.2</td></tr><tr><td>14</td><td>Daniela</td><td>10</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	3	Javier	10	5	Eduardo	9.6	9	Daniel	9	10	Daniela	8.5	11	Marcos	10	12	Pedro	8	13	Daniela	9.2	14	Daniela	10
idAlumno	nombre	calificacion																													
1	Daniela	9.5																													
3	Javier	10																													
5	Eduardo	9.6																													
9	Daniel	9																													
10	Daniela	8.5																													
11	Marcos	10																													
12	Pedro	8																													
13	Daniela	9.2																													
14	Daniela	10																													
Hay una pequeña diferencia y es que los valores 8 y 10 irán incluidos.																															
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion >= 8 4 AND calificacion <=10</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>3</td><td>Javier</td><td>10</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr><tr><td>11</td><td>Marcos</td><td>10</td></tr><tr><td>12</td><td>Pedro</td><td>8</td></tr><tr><td>13</td><td>Daniela</td><td>9.2</td></tr><tr><td>14</td><td>Daniela</td><td>10</td></tr></table>	idAlumno	nombre	calificacion	1	Daniela	9.5	3	Javier	10	5	Eduardo	9.6	9	Daniel	9	10	Daniela	8.5	11	Marcos	10	12	Pedro	8	13	Daniela	9.2	14	Daniela	10
idAlumno	nombre	calificacion																													
1	Daniela	9.5																													
3	Javier	10																													
5	Eduardo	9.6																													
9	Daniel	9																													
10	Daniela	8.5																													
11	Marcos	10																													
12	Pedro	8																													
13	Daniela	9.2																													
14	Daniela	10																													
Ahora sí que el resultado es el mismo.																															
<pre>1 • SELECT * 2 FROM calificaciones 3 WHERE calificacion 4 BETWEEN 6 AND 8 5 AND NOT nombre='Pedro';</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>6</td><td>Marcos</td><td>7</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr></table>	idAlumno	nombre	calificacion	6	Marcos	7	7	María	7.8	8	Ruben	6.9																		
idAlumno	nombre	calificacion																													
6	Marcos	7																													
7	María	7.8																													
8	Ruben	6.9																													
Queremos consultar con los alumnos cuyas notas están entre 6 y 8, y además que no sea Pedro.																															

22.- Comando ORDER BY

El comando de ORDER BY nos va a ordenar la consulta.

idAlumno	nombre	calificacion	1 • SELECT *	idAlumno	nombre	calificacion
1	Daniela	9.5	2 FROM calificaciones	2	NULL	7
2	NULL	7	3 ORDER BY nombre;	9	Daniel	9
3	Javier	10		1	Daniela	9.5
4	Pedro	6.5		13	Daniela	9.2
5	Eduardo	9.6		10	Daniela	8.5
6	Marcos	7		14	Daniela	10
7	María	7.8		5	Eduardo	9.6
8	Ruben	6.9		3	Javier	10
9	Daniel	9		6	Marcos	7
10	Daniela	8.5		11	Marcos	10
11	Marcos	10		7	María	7.8
12	Pedro	8		4	Pedro	6.5
13	Daniela	9.2		12	Pedro	8
14	Daniela	10		8	Ruben	6.9

Realizamos una consulta que ordenamos alfabéticamente por el nombre.

1 • SELECT *	idAlumno	nombre	calificacion
2 FROM calificaciones	8	Ruben	6.9
3 ORDER BY nombre DESC;	4	Pedro	6.5
	12	Pedro	8
	7	María	7.8
	6	Marcos	7
	11	Marcos	10
	3	Javier	10
	5	Eduardo	9.6
	13	Daniela	9.2
	10	Daniela	8.5
	1	Daniela	9.5
	14	Daniela	10
	9	Daniel	9
	2	NULL	7

Realizamos una consulta que ordenamos alfabéticamente por el nombre en modo descendente.

El orden puede ser ASC ascendente o DESC descendente por defecto si no se pone nada será ascendente.

1 • SELECT *	idAlumno	nombre	calificacion
2 FROM calificaciones	4	Pedro	6.5
3 ORDER BY calificacion ASC;	8	Ruben	6.9
	2	NULL	7
	6	Marcos	7
	7	María	7.8
	12	Pedro	8
	10	Daniela	8.5
	9	Daniel	9
	13	Daniela	9.2
	1	Daniela	9.5
	5	Eduardo	9.6

	<table><tr><td>11</td><td>Marcos</td><td>10</td></tr><tr><td>3</td><td>Javier</td><td>10</td></tr><tr><td>14</td><td>Daniela</td><td>10</td></tr></table>	11	Marcos	10	3	Javier	10	14	Daniela	10																																				
11	Marcos	10																																												
3	Javier	10																																												
14	Daniela	10																																												
Queremos realizar una consulta que la ordene por calificaciones en modo ascendente.																																														
<pre>1 • SELECT * 2 FROM calificaciones 3 ORDER BY calificacion DESC;</pre>	<table><tr><th>idAlumno</th><th>nombre</th><th>calificacion</th></tr><tr><td>14</td><td>Daniela</td><td>10</td></tr><tr><td>11</td><td>Marcos</td><td>10</td></tr><tr><td>3</td><td>Javier</td><td>10</td></tr><tr><td>5</td><td>Eduardo</td><td>9.6</td></tr><tr><td>1</td><td>Daniela</td><td>9.5</td></tr><tr><td>13</td><td>Daniela</td><td>9.2</td></tr><tr><td>9</td><td>Daniel</td><td>9</td></tr><tr><td>10</td><td>Daniela</td><td>8.5</td></tr><tr><td>12</td><td>Pedro</td><td>8</td></tr><tr><td>7</td><td>María</td><td>7.8</td></tr><tr><td>6</td><td>Marcos</td><td>7</td></tr><tr><td>2</td><td>NULL</td><td>7</td></tr><tr><td>8</td><td>Ruben</td><td>6.9</td></tr><tr><td>4</td><td>Pedro</td><td>6.5</td></tr></table>	idAlumno	nombre	calificacion	14	Daniela	10	11	Marcos	10	3	Javier	10	5	Eduardo	9.6	1	Daniela	9.5	13	Daniela	9.2	9	Daniel	9	10	Daniela	8.5	12	Pedro	8	7	María	7.8	6	Marcos	7	2	NULL	7	8	Ruben	6.9	4	Pedro	6.5
idAlumno	nombre	calificacion																																												
14	Daniela	10																																												
11	Marcos	10																																												
3	Javier	10																																												
5	Eduardo	9.6																																												
1	Daniela	9.5																																												
13	Daniela	9.2																																												
9	Daniel	9																																												
10	Daniela	8.5																																												
12	Pedro	8																																												
7	María	7.8																																												
6	Marcos	7																																												
2	NULL	7																																												
8	Ruben	6.9																																												
4	Pedro	6.5																																												
Ahora quiero las calificaciones en modo descendente.																																														

23.- ALIAS en SQL (AS)

Servirá para simplificar nuestras consultas.

Cuando nosotros queramos agregar un alias a una columna se hace de la siguiente forma:

```
1 • SELECT idAlumno, nombre, calificacion
2 FROM CALIFICACIONES;
```

idAlumno	nombre	calificacion
1	Daniela	9.5
2	NULL	7
3	Javier	10
4	Pedro	6.5
5	Eduardo	9.6
6	Marcos	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5
11	Marcos	10
12	Pedro	8
13	Daniela	9.2
14	Daniela	10
NULL	NULL	NULL

Ahora vamos a utilizar las alias.

```
1 • SELECT idAlumno, nombre as nom, calificacion
2 FROM CALIFICACIONES;
```

idAlumno	nom	calificacion
1	Daniela	9.5
2	NULL	7
3	Javier	10
4	Pedro	6.5
5	Eduardo	9.6
6	Marcos	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5
11	Marcos	10
12	Pedro	8
13	Daniela	9.2
14	Daniela	10

Es una forma de renombrar las columnas o nombre de los campos.

Y lo podemos hacer con todos los campos.

```
1 • SELECT idAlumno as Id, nombre as nom, calificacion as nota
2 FROM CALIFICACIONES;
```

Id	nom	nota
1	Daniela	9.5
2	NULL	7
3	Javier	10
4	Pedro	6.5
5	Eduardo	9.6
6	Marcos	7
7	María	7.8
8	Ruben	6.9
9	Daniel	9
10	Daniela	8.5
11	Marcos	10
12	Pedro	8
13	Daniela	9.2
14	Daniela	10

Ahora para tablas.

```

1 • SELECT *
2 FROM calificaciones AS cal
3 WHERE cal.calificacion>9;

```

idAlumno	nombre	calificacion
1	Daniela	9.5
3	Javier	10
5	Eduardo	9.6
11	Marcos	10
13	Daniela	9.2
14	Daniela	10

Es agregar un apodo a nuestros campos y tablas.

24.- TRUNCATE TABLE - Eliminar datos de Tablas

Este comando nos ayudara a eliminar registros , sin eliminar la tabla ni la estructura de la tabla.

También ayudará a resetear todo los contadores que sean auto incrementables a 0.

Dejamos al tabla vacía para volver a rellenarla.

Para este capítulo vamos a trabajar con la tabla empleado.

```
1 • INSERT INTO empleado VALUES(1, 'Daniel', 'Director');
2 • INSERT INTO empleado VALUES(2, 'Uriel', 'Subdirector');
3 • INSERT INTO empleado VALUES(3, 'Martín', 'Gerente');
4 • INSERT INTO empleado VALUES(4, 'Rosa', 'Gerente');
5 • INSERT INTO empleado VALUES(5, 'Marta', 'Gerente');
6
7 • SELECT * FROM empleado
```

Este será el resultado:

idEmpleado	nombre	puesto
1	Daniel	Director
2	Uriel	Subdirector
3	Martín	Gerente
4	Rosa	Gerente
5	Marta	Gerente

```
1 • TRUNCATE TABLE empleado;
```

Ejecutamos y a continuación consultamos los registros de la tabla.

```
1 • SELECT * FROM empleado;
```

Este será la información que tiene la tabla.

idEmpleado	nombre	puesto
------------	--------	--------

Está vacía.

25.- DROP TABLE – Eliminar tablas.

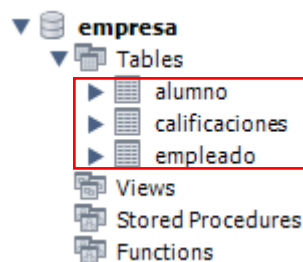
Vamos a agregar registros a la tabla empleado.

```
1 • INSERT INTO empleado VALUES(1 , 'Ruben', 'Director');
2 • INSERT INTO empleado VALUES(2 , 'Javier', 'Subdirector');
3 • INSERT INTO empleado VALUES(3 , 'Marcos', 'Gerente');
4 • INSERT INTO empleado VALUES(4 , 'María', 'Gerente');
5 • INSERT INTO empleado VALUES(5 , 'Alberto', 'Gerente');
6
7 • SELECT * FROM empleado;
```

Vamos a ejecutar.

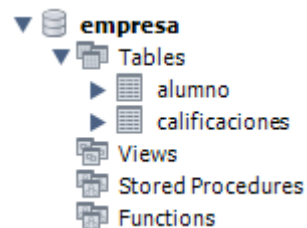
idEmpleado	nombre	puesto
1	Ruben	Director
2	Javier	Subdirector
3	Marcos	Gerente
4	María	Gerente
5	Alberto	Gerente

Estas son las tablas de la base de datos empresa.



```
1 • DROP TABLE empleado;
```

Ejecutamos.



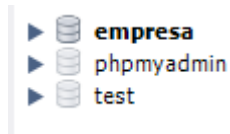
Ya hemos eliminado la tabla empleado.

Si la queremos utilizar de nuevo habrá que crearla de nuevo.

```
1 • CREATE TABLE EMPLEADO(idEmpleado integer,
2 •                             nombre varchar(20),
3 •                             puesto varchar(20));
```

26.- DROP DATABASE – Eliminar bases de datos

Utilizaremos el comando DROP DATABASE para eliminar base de datos.

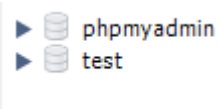


Vamos a eliminar la base de datos **empresa**, de este modo eliminamos todo su contenido.

```
1 • DROP DATABASE empresa;
```

Si ejecutamos.

```
✓ 2 06:47:14 DROP DATABASE empresa
```



Ya hemos eliminado la base de datos.

```
1 • DROP DATABASE IF EXISTS empresa;
```

Nos va a permitir identificar si la base de datos existe para eliminarla.

Vamos a ejecutar.

```
⚠ 3 06:50:19 DROP DATABASE IF EXISTS empresa
```

Nos envía una advertencia diciendo que esta base de datos no existe.

```
1 • DROP DATABASE empresa;
```

Si lo ejecutamos y la base de datos no existe.

Vamos a ejecutar.

```
✗ 4 06:51:32 DROP DATABASE empresa
```

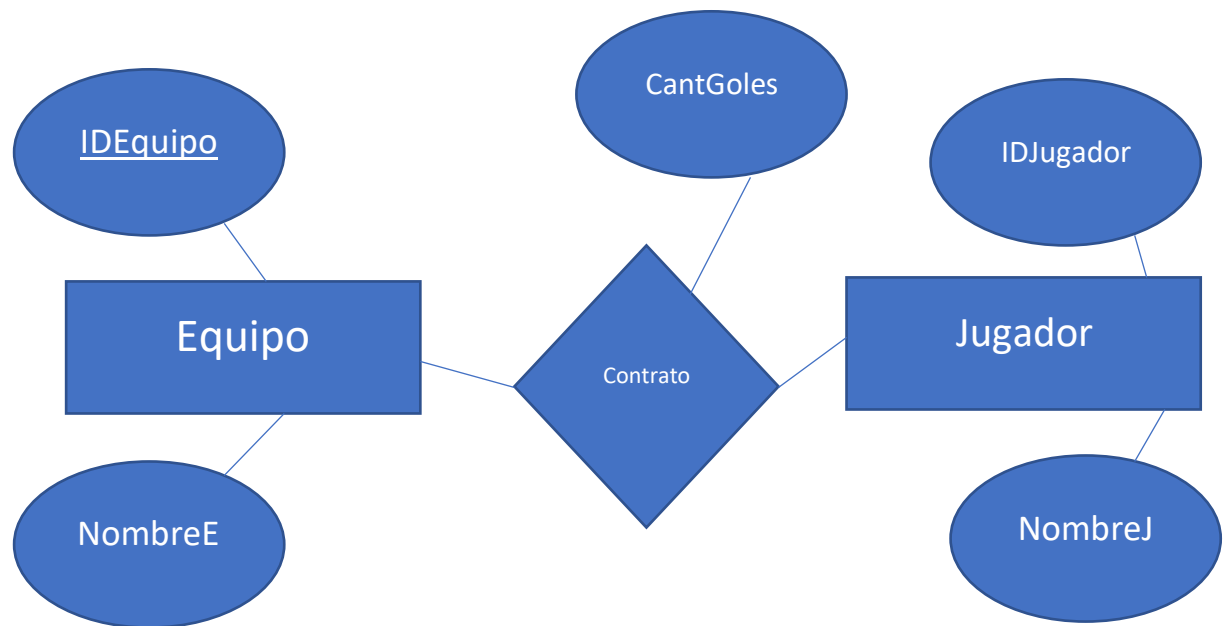
27.- EJERCICIO FINAL I (Modelos)

Planteamiento del problema:

El salón de la FAMA de la FIFA requiere tener un registro de la cantidad de goles que hizo un jugador a lo largo de su vida en cada equipo que jugó. Se conoce el nombre de los equipos y de los jugadores.

- JUGADOR nombre
- EQUIPO nombre

Modelo entidad relación:



Modelo relacional:

EQUIPO	
ID_EQUIPO (INTEGER)	NOMBRE_E (VARCHAR)
001	Real Madrid
002	Barcelona

CONTRATO		
ID_EQUIPO (INTEGER)	ID_JUGADOR (INTEGER)	CantGoles (INTEGER)

JUGADOR	
ID_JUGADOR (INTEGER)	NOMBRE_J (VARCHAR)
001	Cristiano Ronaldo
002	Lionel Messi

28.- EJERCICIO FINAL II (Creando la BD)

Creación de Base de Datos en SQL

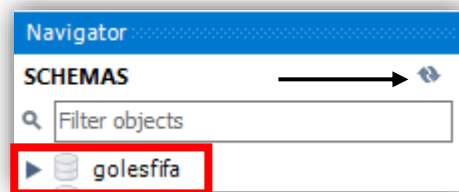
Vamos a crear la base de datos golesfifa.

A continuación vamos a crear las tablas EQUIPO Y JUGADOR.

```
1 -- Creando una base de datos llamada golesfifa
2 • CREATE DATABASE golesfifa;
```

Le damos a ejecutar.

✓ 1 09:09:50 CREATE DATABASE golesfifa



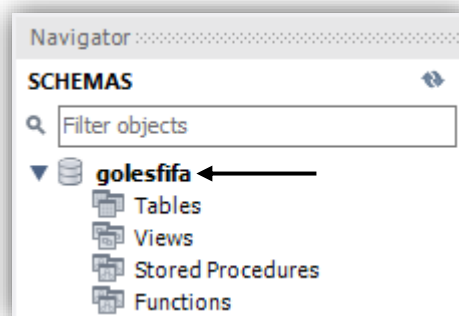
Ya hemos creado la base de datos.

Ahora vamos a llamar a la base de datos.

```
1 -- Usar la base de datos golesfifa
2 -- En este gestor de bases no funciona el USE DATABASE
3 • USE golesfifa;
```

Vamos a ejecutar.

✓ 4 09:15:14 USE golesfifa



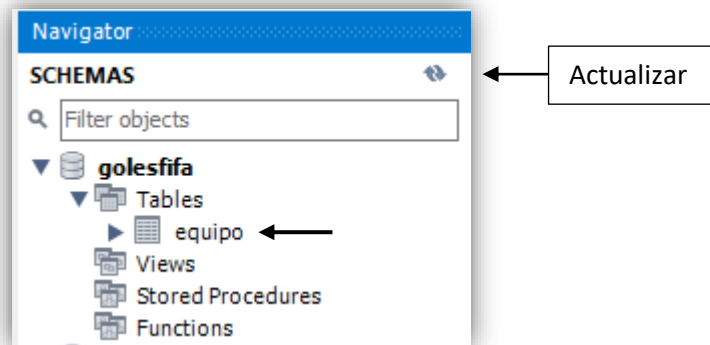
Al ponerse en negrita significa que ya está activa.

A continuación vamos a crear la tabla EQUIPO.

```
1 -- Crear tabla EQUIPO
2 • CREATE TABLE EQUIPO (ID_EQUIPO INTEGER PRIMARY KEY,
3   NOMBRE_E VARCHAR(30));
```

Ejecutamos.

✓ 5 09:19:14 CREATE TABLE EQUIPO (ID_EQUIPO INTEGER PRIMARY KEY, NOMBRE_E VARCHAR(30))

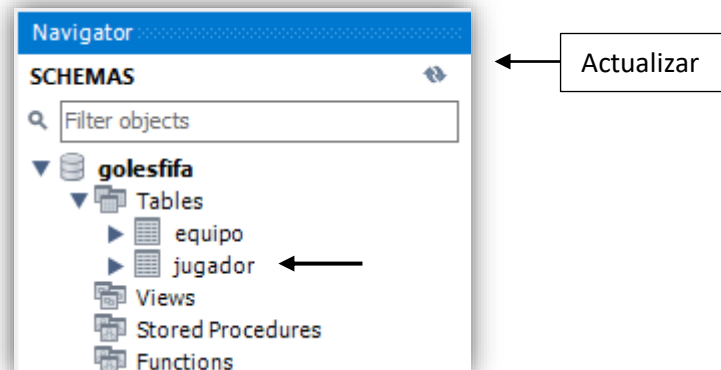


Ahora vamos a crear la tabla JUGADOR.

```
1 -- Crear tabla JUGADOR
2 • CREATE TABLE JUGADOR (ID_JUGADOR INTEGER PRIMARY KEY,
3   NOMBRE_J VARCHAR(50));
```

Vamos a ejecutar.

✓ 6 09:22:09 CREATE TABLE JUGADOR (ID_JUGADOR INTEGER PRIMARY KEY, NOMBRE_J VARCHAR(50))



29.- EJERCICIO FINAL III (Creando la tabla CONTRATO)

Vamos a crear la tabla CONTRATO que va a estar relacionada con la tabla EQUIPO y la tabla

```
1  -- Crear tabla CONTRATO
2  CREATE TABLE CONTRATO(
3      ID_EQUIPO INTEGER NOT NULL, -- Este campo ENETERO no puede estar vacío
4      ID_JUGADOR INTEGER NOT NULL, -- Este campo ENETERO no puede estar vacío
5      CANT_GOLES INTEGER, -- Campo para guardar el número de goles
6      PRIMARY KEY (ID_EQUIPO, ID_JUGADOR), -- La clave es al suma de
7                                          -- ID_EQUIPO + ID_JUGADOR
8      -- Vamos a relacionar el campo ID_EQUIPO con la tabla EQUIPO
9      FOREIGN KEY (ID_EQUIPO) REFERENCES EQUIPO(ID_EQUIPO),
10     -- Vamos a relacionar el campo ID_JUGADOR con la tabla JUGADOR
11     FOREIGN KEY (ID_JUGADOR) REFERENCES JUGADOR(ID_JUGADOR));
```

Vamos a ejecutar.

✓ 8 10:03:06 CREATE TABLE CONTRATO(ID_EQUIPO INTEGER NOT NULL, -- Este campo ENETERO no puede estar vacío

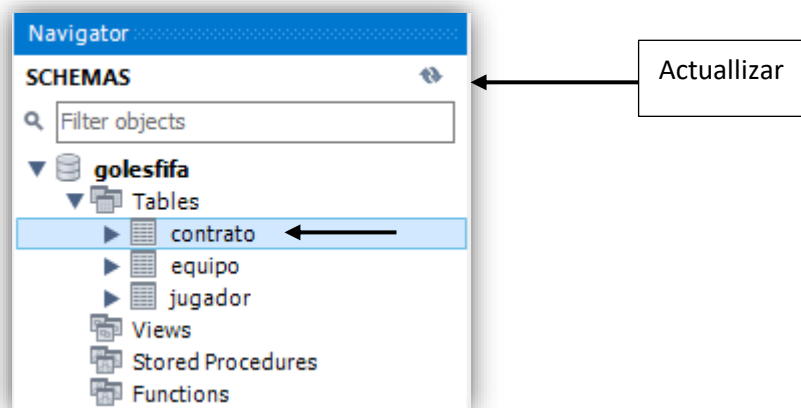


Table: **contrato**

Columns:

ID_EQUIPO	int(11) PK
ID_JUGADOR	int(11) PK
CANT_GOLES	int(11)

} PRIMARY KEY

Para poder agregar un registro tanto ID_EQUIPO como ID_JUGADOR no pueden estar vacíos.

30.- EJERCICIO FINAL IV (Insertando Datos)

En este capítulo solo vamos a añadir registros en las tablas EQUIPO y JUGADOR.

```
1  -- Vamos a añadir 5 registros.
2  •  INSERT INTO EQUIPO VALUES(1, 'REAL MADRID');
3  •  INSERT INTO EQUIPO VALUES(2, 'BARCELONA');
4  •  INSERT INTO EQUIPO VALUES(3, 'MANCHERTER UNITED');
5  •  INSERT INTO EQUIPO VALUES(4, 'AC MILAN');
6  •  INSERT INTO EQUIPO VALUES(5, 'JUVENTUS');
```

Le damos a ejecutar.

```
✓ 9 10:42:09 INSERT INTO EQUIPO VALUES(1, 'REAL MADRID')
✓ 10 10:42:09 INSERT INTO EQUIPO VALUES(2, 'BARCELONA')
✓ 12 10:50:50 INSERT INTO EQUIPO VALUES(3, 'MANCHERTER UNITED')
✓ 13 10:50:50 INSERT INTO EQUIPO VALUES(4, 'AC MILAN')
✓ 14 10:50:50 INSERT INTO EQUIPO VALUES(5, 'JUVENTUS')
```

Vamos a ver el contenido de la tabla EQUIPO.

```
1  •  SELECT * FROM EQUIPO;

✓ 11 10:43:06 SELECT * FROM EQUIPO LIMIT 0, 1000
```

ID_EQUIPO	NOMBRE_E
1	REAL MADRID
2	BARCELONA
3	MANCHERTER UNITED
4	AC MILAN
5	JUVENTUS

Ahora en la tabla JUGADOR.

```
1  -- Anadir 5 registros tabla JUGADOR
2  •  INSERT INTO JUGADOR VALUE(1, 'NEYMAR');
3  •  INSERT INTO JUGADOR VALUE(2, 'MESSI');
4  •  INSERT INTO JUGADOR VALUE(3, 'CRISTIANO');
5  •  INSERT INTO JUGADOR VALUE(4, 'AGUERO');
6  •  INSERT INTO JUGADOR VALUE(5, 'KANTE');
```

Los ejecutamos.

```
✓ 18 10:56:56 INSERT INTO JUGADOR VALUE(1, 'NEYMAR')
✓ 19 10:56:56 INSERT INTO JUGADOR VALUE(2, 'MESSI')
✓ 20 10:56:56 INSERT INTO JUGADOR VALUE(3, 'CRISTIANO')
✓ 21 10:56:56 INSERT INTO JUGADOR VALUE(4, 'AGUERO')
✓ 22 10:56:56 INSERT INTO JUGADOR VALUE(5, 'KANTE')
```

Vamos a ver el contenido de la tabla JUGADOR.

1 • `SELECT * FROM JUGADOR;`

✓ 23 10:58:10 `SELECT * FROM JUGADOR LIMIT 0, 1000`

ID_JUGADOR	NOMBRE_J
1	NEYMAR
2	MESSI
3	CRISTIANO
4	AGUERO
5	KANTE

31.- EJERCICIO FINAL V (Insertando datos Tabla CONTRATO)

En este capítulo vamos a añadir registros a nuestra tabla CONTRATO.

Table: contrato	
Columns:	
<u>ID_EQUIPO</u>	int(11) PK
<u>ID_JUGADOR</u>	int(11) PK
CANT_GOLES	int(11)

Esta tabla es una tabla compuesta de dos campos ID_EQUIPO y ID_JUGADOR que son los dos campos que van a componer la llave de esta tabla.

Tenemos una llave compuesta por dos campos ID_EQUIPO y ID_JUGADOR.

¿Cómo vamos a añadir registros en esta tabla?

Tenemos que tener en cuenta.

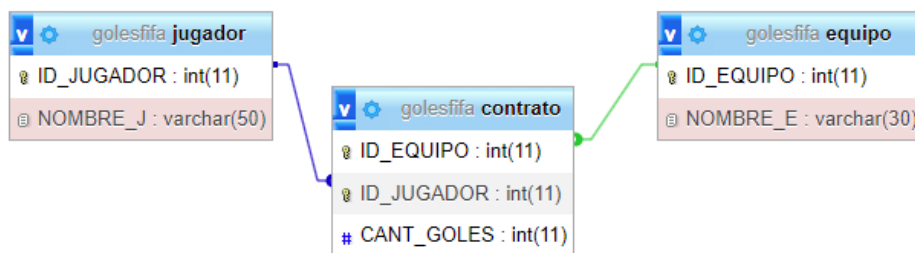


Tabla CONTRATO está relacionada con la tabla JUGADOR y EQUIPO por sus respectivos ID, esto significa que si introducimos en la tabla CONTRATO registro donde el campo ID_EQUIPO no está registrado en la tabla EQUIPO o con el ID_JUGADOR que no está registrado en la tabla JUGADOR, nos va a dar error.

Para insertar registros en esta tabla es interesante tener la relación de los jugadores y equipos para agregarlos correctamente.

1	SELECT * FROM EQUIPO;		1	SELECT * FROM JUGADOR;	
	ID_EQUIPO	NOMBRE_E		ID_JUGADOR	NOMBRE_J
	1	REAL MADRID		1	NEYMAR
	2	BARCELONA		2	MESSI
	3	MANCHERTER UNITED		3	CRISTIANO
	4	AC MILAN		4	AGUERO
	5	JUVENTUS		5	KANTE

A partir de ahora ya podemos agregar los registros en la tabla CONTRATO.

Por ejemplo si queremos que añadir un registro que diga en el REAL MADRID el jugador CRISTIANO ha marcado 50 goles.

```
1  INSERT INTO CONTRATO
2  -- 1 REAL MADRID y 3 CRISTIANO de sus respectivas tablas.
3  VALUES(1, 3, 50);
```

Vamos a ejecutar.

✓ 5 12:09:52 INSERT INTO CONTRATO -- 1 REAL MADRID y 3 CRISTIANO de sus respectivas tablas. VALUES(1, 3, 50)

Ahora vamos a consultar la tabla CONTRATO.

```
1 • SELECT * FROM CONTRATO;
```

Este será el resultado:

✓ 6 12:11:08 SELECT * FROM CONTRATO LIMIT 0, 1000

ID_EQUIPO	ID_JUGADOR	CANT_GOLES
1	3	50

Lo que hay que tener en cuenta que introducir otra vez en el Real Madrid a CRISTIANO otro número de goles, no se va a poder repetir esto nos dará un error.

```
1 • INSERT INTO CONTRATO VALUES(1,3,46);
```

Si lo ejecutamos.

✗ 7 12:17:20 INSERT INTO CONTRATO VALUES(1,3,46)

Error Code: 1062. Duplicate entry '1-3' for key 'PRIMARY'

Nos dice que no podemos duplicar la clave principal.

Si podremos agregar al mismo jugador con otro equipo.

```
1  -- Con el BARCELONA, CRISTIANO ha marcado 46 goles.
2 • INSERT INTO CONTRATO VALUES(2,3,46);
```

Vamos a ejecutar.

✓ 8 12:21:01 INSERT INTO CONTRATO VALUES(2,3,46) 1 row(s) affected

El registro se ha añadido.

Vamos a consultar la tabla CONTRATO.

```
1 • SELECT * FROM CONTRATO;
```

ID_EQUIPO	ID_JUGADOR	CANT_GOLES
1	3	50
2	3	46

Ahora al REAL MADRID el jugador NEYMAR ha marcado 16 goles.

```
1 • INSERT INTO CONTRATO VALUE(1,1,16);
```

Ejecutamos.

✓	10	12:26:31	INSERT INTO CONTRATO VALUE(1,1,16)	1 row(s) affected
---	----	----------	------------------------------------	-------------------

Vamos a ver los registros de la tabla CONTRATO.

```
1 • SELECT * FROM CONTRATO;
```

Ejecutamos.

✓	12	12:28:19	SELECT * FROM CONTRATO LIMIT 0, 1000	3 row(s) returned
---	----	----------	--------------------------------------	-------------------

ID_EQUIPO	ID_JUGADOR	CANT_GOLES
1	1	16
1	3	50
2	3	46

Vamos a agregar los siguientes registros:

- 1 • INSERT INTO CONTRATO VALUES(3, 4, 10);
- 2 • INSERT INTO CONTRATO VALUES(1, 5, 8);
- 3 • INSERT INTO CONTRATO VALUES(2, 4, 12);
- 4 • INSERT INTO CONTRATO VALUES(4, 3, 22);
- 5 • INSERT INTO CONTRATO VALUES(5, 5, 5);
- 6 • INSERT INTO CONTRATO VALUES(2, 5, 13);
- 7 • INSERT INTO CONTRATO VALUES(4, 4, 7);
- 8 • INSERT INTO CONTRATO VALUES(5, 2, 32);
- 9 • INSERT INTO CONTRATO VALUES(1, 1, 9);
- 10 • INSERT INTO CONTRATO VALUES(3, 4, 16);
- 2 • INSERT INTO CONTRATO VALUES(1, 4, 9);

Ejecutamos.

✓	12	12:28:19	SELECT * FROM CONTRATO LIMIT 0, 1000	3 row(s) returned
✓	13	12:36:50	INSERT INTO CONTRATO VALUES(3, 4, 10)	1 row(s) affected
✓	14	12:36:50	INSERT INTO CONTRATO VALUES(1, 5, 8)	1 row(s) affected
✓	15	12:36:50	INSERT INTO CONTRATO VALUES(2, 4, 12)	1 row(s) affected
✓	16	12:36:50	INSERT INTO CONTRATO VALUES(4, 3, 22)	1 row(s) affected
✓	17	12:36:50	INSERT INTO CONTRATO VALUES(5, 5, 5)	1 row(s) affected
✓	18	12:36:50	INSERT INTO CONTRATO VALUES(2, 5, 13)	1 row(s) affected
✓	19	12:36:50	INSERT INTO CONTRATO VALUES(4, 4, 7)	1 row(s) affected
✓	20	12:36:50	INSERT INTO CONTRATO VALUES(5, 2, 32)	1 row(s) affected
✓	23	12:38:23	INSERT INTO CONTRATO VALUES(1, 4, 9)	1 row(s) affected

Vamos a consultar la tabla.

```
1 • SELECT * FROM CONTRATO;
```

ID_EQUIPO	ID_JUGADOR	CANT_GOLES
1	1	16
1	3	50
1	4	9
1	5	8
2	3	46
2	4	12
2	5	13
3	4	10
4	3	22
4	4	7
5	2	32
5	5	5

31.- EJERCICIO FINAL VI: Consultas Sencillas

• <code>SELECT ID_EQUIPO FROM EQUIPO;</code>	<table><tr><th>ID_EQUIPO</th></tr><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>4</td></tr><tr><td>5</td></tr></table>	ID_EQUIPO	1	2	3	4	5						
ID_EQUIPO													
1													
2													
3													
4													
5													
De la tabla EQUIPO muéstrame todos los registros de la columna ID_EQUIPO.													
1 • <code>SELECT NOMBRE_E FROM EQUIPO;</code>	<table><tr><th>NOMBRE_E</th></tr><tr><td>REAL MADRID</td></tr><tr><td>BARCELONA</td></tr><tr><td>MANCHERTER UNITED</td></tr><tr><td>AC MILAN</td></tr><tr><td>JUVENTUS</td></tr></table>	NOMBRE_E	REAL MADRID	BARCELONA	MANCHERTER UNITED	AC MILAN	JUVENTUS						
NOMBRE_E													
REAL MADRID													
BARCELONA													
MANCHERTER UNITED													
AC MILAN													
JUVENTUS													
Muéstrame la columna NOMBRE_E de la tabla EQUIPO todos los registros.													
1 • <code>SELECT ID_EQUIPO, NOMBRE_E</code> 2 <code>FROM EQUIPO;</code>	<table><tr><th>ID_EQUIPO</th><th>NOMBRE_E</th></tr><tr><td>1</td><td>REAL MADRID</td></tr><tr><td>2</td><td>BARCELONA</td></tr><tr><td>3</td><td>MANCHERTER UNITED</td></tr><tr><td>4</td><td>AC MILAN</td></tr><tr><td>5</td><td>JUVENTUS</td></tr></table>	ID_EQUIPO	NOMBRE_E	1	REAL MADRID	2	BARCELONA	3	MANCHERTER UNITED	4	AC MILAN	5	JUVENTUS
ID_EQUIPO	NOMBRE_E												
1	REAL MADRID												
2	BARCELONA												
3	MANCHERTER UNITED												
4	AC MILAN												
5	JUVENTUS												
Muéstrame las columnas ID_EQUIPO y NOMBRE_E todos los registros de la tabla EQUIPO.													
1 • <code>SELECT * FROM EQUIPO;</code>	<table><tr><th>ID_EQUIPO</th><th>NOMBRE_E</th></tr><tr><td>1</td><td>REAL MADRID</td></tr><tr><td>2</td><td>BARCELONA</td></tr><tr><td>3</td><td>MANCHERTER UNITED</td></tr><tr><td>4</td><td>AC MILAN</td></tr><tr><td>5</td><td>JUVENTUS</td></tr></table>	ID_EQUIPO	NOMBRE_E	1	REAL MADRID	2	BARCELONA	3	MANCHERTER UNITED	4	AC MILAN	5	JUVENTUS
ID_EQUIPO	NOMBRE_E												
1	REAL MADRID												
2	BARCELONA												
3	MANCHERTER UNITED												
4	AC MILAN												
5	JUVENTUS												
Muéstrame todos los campos y registros de la tabla EQUIPO.													
Podemos repetir las mismas consultas con la tabla JUGADOR.													
1 • <code>SELECT ID_JUGADOR FROM JUGADOR;</code>	<table><tr><th>ID_JUGADOR</th></tr><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>4</td></tr><tr><td>5</td></tr></table>	ID_JUGADOR	1	2	3	4	5						
ID_JUGADOR													
1													
2													
3													
4													
5													
1 • <code>SELECT NOMBRE_J FROM JUGADOR;</code>	<table><tr><th>NOMBRE_J</th></tr><tr><td>NEYMAR</td></tr><tr><td>MESSI</td></tr><tr><td>CRISTIANO</td></tr><tr><td>AGUERO</td></tr><tr><td>KANTE</td></tr></table>	NOMBRE_J	NEYMAR	MESSI	CRISTIANO	AGUERO	KANTE						
NOMBRE_J													
NEYMAR													
MESSI													
CRISTIANO													
AGUERO													
KANTE													
1 • <code>SELECT ID_JUGADOR, NOMBRE_J</code> 2 <code>FROM JUGADOR;</code>	<table><tr><th>ID_JUGADOR</th><th>NOMBRE_J</th></tr><tr><td>1</td><td>NEYMAR</td></tr><tr><td>2</td><td>MESSI</td></tr><tr><td>3</td><td>CRISTIANO</td></tr><tr><td>4</td><td>AGUERO</td></tr><tr><td>5</td><td>KANTE</td></tr></table>	ID_JUGADOR	NOMBRE_J	1	NEYMAR	2	MESSI	3	CRISTIANO	4	AGUERO	5	KANTE
ID_JUGADOR	NOMBRE_J												
1	NEYMAR												
2	MESSI												
3	CRISTIANO												
4	AGUERO												
5	KANTE												

1 • <code>SELECT * FROM JUGADOR;</code>	<table><tr><th>ID_JUGADOR</th><th>NOMBRE_J</th></tr><tr><td>1</td><td>NEYMAR</td></tr><tr><td>2</td><td>MESSI</td></tr><tr><td>3</td><td>CRISTIANO</td></tr><tr><td>4</td><td>AGUERO</td></tr><tr><td>5</td><td>KANTE</td></tr></table>	ID_JUGADOR	NOMBRE_J	1	NEYMAR	2	MESSI	3	CRISTIANO	4	AGUERO	5	KANTE																											
ID_JUGADOR	NOMBRE_J																																							
1	NEYMAR																																							
2	MESSI																																							
3	CRISTIANO																																							
4	AGUERO																																							
5	KANTE																																							
Lo mismo podemos hacer con la tabla CONTRATO.																																								
1 • <code>SELECT ID_EQUIPO FROM CONTRATO;</code>	<table><tr><th>ID_EQUIPO</th></tr><tr><td>1</td></tr><tr><td>5</td></tr><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>4</td></tr><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>4</td></tr><tr><td>1</td></tr><tr><td>2</td></tr><tr><td>5</td></tr></table>	ID_EQUIPO	1	5	1	2	4	1	2	3	4	1	2	5																										
ID_EQUIPO																																								
1																																								
5																																								
1																																								
2																																								
4																																								
1																																								
2																																								
3																																								
4																																								
1																																								
2																																								
5																																								
1 • <code>SELECT ID_EQUIPO, ID_JUGADOR</code> 2 <code>FROM CONTRATO;</code>	<table><tr><th>ID_EQUIPO</th><th>ID_JUGADOR</th></tr><tr><td>1</td><td>1</td></tr><tr><td>5</td><td>2</td></tr><tr><td>1</td><td>3</td></tr><tr><td>2</td><td>3</td></tr><tr><td>4</td><td>3</td></tr><tr><td>1</td><td>4</td></tr><tr><td>2</td><td>4</td></tr><tr><td>3</td><td>4</td></tr><tr><td>4</td><td>4</td></tr><tr><td>1</td><td>5</td></tr><tr><td>2</td><td>5</td></tr><tr><td>5</td><td>5</td></tr></table>	ID_EQUIPO	ID_JUGADOR	1	1	5	2	1	3	2	3	4	3	1	4	2	4	3	4	4	4	1	5	2	5	5	5													
ID_EQUIPO	ID_JUGADOR																																							
1	1																																							
5	2																																							
1	3																																							
2	3																																							
4	3																																							
1	4																																							
2	4																																							
3	4																																							
4	4																																							
1	5																																							
2	5																																							
5	5																																							
1 • <code>SELECT ID_EQUIPO,</code> 2 <code>ID_JUGADOR,</code> 3 <code>CANT_GOLES</code> 4 <code>FROM CONTRATO;</code>	<table><tr><th>ID_EQUIPO</th><th>ID_JUGADOR</th><th>CANT_GOLES</th></tr><tr><td>1</td><td>1</td><td>16</td></tr><tr><td>1</td><td>3</td><td>50</td></tr><tr><td>1</td><td>4</td><td>9</td></tr><tr><td>1</td><td>5</td><td>8</td></tr><tr><td>2</td><td>3</td><td>46</td></tr><tr><td>2</td><td>4</td><td>12</td></tr><tr><td>2</td><td>5</td><td>13</td></tr><tr><td>3</td><td>4</td><td>10</td></tr><tr><td>4</td><td>3</td><td>22</td></tr><tr><td>4</td><td>4</td><td>7</td></tr><tr><td>5</td><td>2</td><td>32</td></tr><tr><td>5</td><td>5</td><td>5</td></tr></table>	ID_EQUIPO	ID_JUGADOR	CANT_GOLES	1	1	16	1	3	50	1	4	9	1	5	8	2	3	46	2	4	12	2	5	13	3	4	10	4	3	22	4	4	7	5	2	32	5	5	5
ID_EQUIPO	ID_JUGADOR	CANT_GOLES																																						
1	1	16																																						
1	3	50																																						
1	4	9																																						
1	5	8																																						
2	3	46																																						
2	4	12																																						
2	5	13																																						
3	4	10																																						
4	3	22																																						
4	4	7																																						
5	2	32																																						
5	5	5																																						
1 • <code>SELECT * FROM CONTRATO;</code>	Hace lo mismo que el anterior ejemplo.																																							

32.- EJERCICIO FINAL VII: Consultas específicas

Vamos a consultar las tres tablas:

Consusltar tabla EQUIPO.			
1 • <code>SELECT * FROM EQUIPO;</code>	ID_EQUIPO		NOMBRE_E
	1	REAL MADRID	
	2	BARCELONA	
	3	MANCHERTER UNITED	
	4	AC MILAN	
	5	JUVENTUS	
Consultar tabla JUGADOR.			
1 • <code>SELECT * FROM JUGADOR;</code>	ID_JUGADOR		NOMBRE_J
	1	NEYMAR	
	2	MESSI	
	3	CRISTIANO	
	4	AGUERO	
	5	KANTE	
Consultar tabña CONTRATO.			
1 • <code>SELECT * FROM CONTRATO;</code>	ID_EQUIPO	ID_JUGADOR	CANT_GOLES
	1	1	16
	1	3	50
	1	4	9
	1	5	8
	2	3	46
	2	4	12
	2	5	13
	3	4	10
	4	3	22
	4	4	7
	5	2	32
	5	5	5

¿Qué es una consulta específica? Porque dentro de ellas nosotros vamos a especificar un dato y obviamente debe de devolvernos un valor dependiendo de ese dato específico.

Quiero saber el ID_JUGADOR pariendo del nombre del jugador.

1 • <code>SELECT ID_JUGADOR FROM JUGADOR</code>	ID_JUGADOR
2 <code>WHERE NOMBRE_J='CRISTIANO';</code>	3
Vamos a hacer al revés, introduciendo el número de ID_JUGADOR nos retorne el nombre del jugador.	
1 • <code>SELECT NOMBRE_J FROM JUGADOR</code>	NOMBRE_J
2 <code>WHERE ID_JUGADOR=5;</code>	KANTE
1 • <code>SELECT * FROM JUGADOR</code>	ID_JUGADOR NOMBRE_J
2 <code>WHERE ID_JUGADOR=5;</code>	5 KANTE
	NULL NULL

33.- EJERCICIO FINAL VIII: Consultas con Operadores Lógicos

Vamos a crear una consulta que nos devuelva todos los nombres de los EQUIPOS que su ID sea mayor a 4.

1 •	SELECT * FROM EQUIPO	ID_EQUIPO	NOMBRE_E
2	WHERE ID_EQUIPO>4;	5	JUVENTUS

Si le indicamos los que si ID es menor a 4.

1 •	SELECT * FROM EQUIPO	ID_EQUIPO	NOMBRE_E
2	WHERE ID_EQUIPO<4;	1	REAL MADRID
		2	BARCELONA
		3	MANCHERTER UNITED

Queremos incluir que el ID 4 también se muestre.

1 •	SELECT * FROM EQUIPO	ID_EQUIPO	NOMBRE_E
2	WHERE ID_EQUIPO<=4;	1	REAL MADRID
		2	BARCELONA
		3	MANCHERTER UNITED
		4	AC MILAN

Queremos ver los registros cuyo Id sea mayor igual 4.

1 •	SELECT * FROM EQUIPO	ID_EQUIPO	NOMBRE_E
2	WHERE ID_EQUIPO>=4;	4	AC MILAN
		5	JUVENTUS

De la tabla CONTATO queremos ver cuando CANT_GOLES sea mayor a 30.

1 •	SELECT * FROM CONTRATO	ID_EQUIPO	ID_JUGADOR	CANT_GOLES
2	WHERE CANT_GOLES>30;	1	3	50
-		2	3	46
		5	2	32

34.- EJERCICIO FINAL IX: Alias o Sobrenombres

Utilizamos ALIAS para cambiar el nombre de las columnas.

<pre>1 • SELECT ID_EQUIPO, NOMBRE_E 2 FROM EQUIPO;</pre>	<table> <tr> <th>ID_EQUIPO</th><th>NOMBRE_E</th></tr> <tr><td>1</td><td>REAL MADRID</td></tr> <tr><td>2</td><td>BARCELONA</td></tr> <tr><td>3</td><td>MANCHERTER UNITED</td></tr> <tr><td>4</td><td>AC MILAN</td></tr> <tr><td>5</td><td>JUVENTUS</td></tr> </table>	ID_EQUIPO	NOMBRE_E	1	REAL MADRID	2	BARCELONA	3	MANCHERTER UNITED	4	AC MILAN	5	JUVENTUS
ID_EQUIPO	NOMBRE_E												
1	REAL MADRID												
2	BARCELONA												
3	MANCHERTER UNITED												
4	AC MILAN												
5	JUVENTUS												
<pre>1 • SELECT ID_EQUIPO AS EQUIPO, NOMBRE_E 2 FROM EQUIPO;</pre>	<table> <tr> <th>EQUIPO</th><th>NOMBRE_E</th></tr> <tr><td>1</td><td>REAL MADRID</td></tr> <tr><td>2</td><td>BARCELONA</td></tr> <tr><td>3</td><td>MANCHERTER UNITED</td></tr> <tr><td>4</td><td>AC MILAN</td></tr> <tr><td>5</td><td>JUVENTUS</td></tr> </table>	EQUIPO	NOMBRE_E	1	REAL MADRID	2	BARCELONA	3	MANCHERTER UNITED	4	AC MILAN	5	JUVENTUS
EQUIPO	NOMBRE_E												
1	REAL MADRID												
2	BARCELONA												
3	MANCHERTER UNITED												
4	AC MILAN												
5	JUVENTUS												
<pre>1 • SELECT ID_EQUIPO AS EQUIPO, 2 NOMBRE_E AS 'EQUIPO FUTBOL' 3 FROM EQUIPO;</pre>	<table> <tr> <th>EQUIPO</th><th>EQUIPO FUTBOL</th></tr> <tr><td>1</td><td>REAL MADRID</td></tr> <tr><td>2</td><td>BARCELONA</td></tr> <tr><td>3</td><td>MANCHERTER UNITED</td></tr> <tr><td>4</td><td>AC MILAN</td></tr> <tr><td>5</td><td>JUVENTUS</td></tr> </table>	EQUIPO	EQUIPO FUTBOL	1	REAL MADRID	2	BARCELONA	3	MANCHERTER UNITED	4	AC MILAN	5	JUVENTUS
EQUIPO	EQUIPO FUTBOL												
1	REAL MADRID												
2	BARCELONA												
3	MANCHERTER UNITED												
4	AC MILAN												
5	JUVENTUS												

Esto es muy útil cuando queremos realizar subconsultas dentro de SQL.

¿Como crear un alias para una tabla?

<pre>1 • SELECT eq.ID_EQUIPO, jg.ID_JUGADOR 2 FROM EQUIPO eq, JUGADOR jg;</pre>	<table> <tr> <th>ID_EQUIPO</th><th>ID_JUGADOR</th></tr> <tr><td>1</td><td>2</td></tr> <tr><td>2</td><td>2</td></tr> <tr><td>3</td><td>2</td></tr> <tr><td>4</td><td>2</td></tr> <tr><td>5</td><td>2</td></tr> <tr><td>1</td><td>3</td></tr> <tr><td>2</td><td>3</td></tr> <tr><td>3</td><td>3</td></tr> <tr><td>4</td><td>3</td></tr> <tr><td>5</td><td>3</td></tr> <tr><td>1</td><td>4</td></tr> <tr><td>2</td><td>4</td></tr> <tr><td>3</td><td>4</td></tr> <tr><td>4</td><td>4</td></tr> <tr><td>5</td><td>4</td></tr> <tr><td>1</td><td>5</td></tr> <tr><td>2</td><td>5</td></tr> <tr><td>3</td><td>5</td></tr> <tr><td>4</td><td>5</td></tr> <tr><td>5</td><td>5</td></tr> </table>	ID_EQUIPO	ID_JUGADOR	1	2	2	2	3	2	4	2	5	2	1	3	2	3	3	3	4	3	5	3	1	4	2	4	3	4	4	4	5	4	1	5	2	5	3	5	4	5	5	5
ID_EQUIPO	ID_JUGADOR																																										
1	2																																										
2	2																																										
3	2																																										
4	2																																										
5	2																																										
1	3																																										
2	3																																										
3	3																																										
4	3																																										
5	3																																										
1	4																																										
2	4																																										
3	4																																										
4	4																																										
5	4																																										
1	5																																										
2	5																																										
3	5																																										
4	5																																										
5	5																																										

35.- EJERCICIO FINAL X: UNIONES ENTRE TABLAS

Queremos obtener cuantos goles anotó CRISTIANO en el REAL MADRID.

Para esta consulta vamos a necesitar las tres tablas, (JUGADOR – EQUIPO – CONTRATO).

Vamos a comprobar si CRISTIANO está en la tabla de JUGADOR.

1 • <code>SELECT * FROM JUGADOR;</code> Cristiano en el jugador con el ID_JUGADOR es el 3.	<table><tr><th>ID_JUGADOR</th><th>NOMBRE_J</th></tr><tr><td>1</td><td>NEYMAR</td></tr><tr><td>2</td><td>MESSI</td></tr><tr><td>3</td><td>CRISTIANO</td></tr><tr><td>4</td><td>AGUERO</td></tr><tr><td>5</td><td>KANTE</td></tr></table>	ID_JUGADOR	NOMBRE_J	1	NEYMAR	2	MESSI	3	CRISTIANO	4	AGUERO	5	KANTE																											
ID_JUGADOR	NOMBRE_J																																							
1	NEYMAR																																							
2	MESSI																																							
3	CRISTIANO																																							
4	AGUERO																																							
5	KANTE																																							
1 • <code>SELECT * FROM EQUIPO;</code> Real Madrid en el equipo con el ID_EQUIPO es el 1.	<table><tr><th>ID_EQUIPO</th><th>NOMBRE_E</th></tr><tr><td>1</td><td>REAL MADRID</td></tr><tr><td>2</td><td>BARCELONA</td></tr><tr><td>3</td><td>MANCHERTER UNITED</td></tr><tr><td>4</td><td>AC MILAN</td></tr><tr><td>5</td><td>JUVENTUS</td></tr></table>	ID_EQUIPO	NOMBRE_E	1	REAL MADRID	2	BARCELONA	3	MANCHERTER UNITED	4	AC MILAN	5	JUVENTUS																											
ID_EQUIPO	NOMBRE_E																																							
1	REAL MADRID																																							
2	BARCELONA																																							
3	MANCHERTER UNITED																																							
4	AC MILAN																																							
5	JUVENTUS																																							
1 • <code>SELECT * FROM CONTRATO;</code> Con equipo Real Madrid y jugador 3 Cristiano ha marcado 50 goles.	<table><tr><th>ID_EQUIPO</th><th>ID_JUGADOR</th><th>CANT_GOLES</th></tr><tr><td>1</td><td>1</td><td>16</td></tr><tr><td>1</td><td>3</td><td>50</td></tr><tr><td>1</td><td>4</td><td>9</td></tr><tr><td>1</td><td>5</td><td>8</td></tr><tr><td>2</td><td>3</td><td>46</td></tr><tr><td>2</td><td>4</td><td>12</td></tr><tr><td>2</td><td>5</td><td>13</td></tr><tr><td>3</td><td>4</td><td>10</td></tr><tr><td>4</td><td>3</td><td>22</td></tr><tr><td>4</td><td>4</td><td>7</td></tr><tr><td>5</td><td>2</td><td>32</td></tr><tr><td>5</td><td>5</td><td>5</td></tr></table>	ID_EQUIPO	ID_JUGADOR	CANT_GOLES	1	1	16	1	3	50	1	4	9	1	5	8	2	3	46	2	4	12	2	5	13	3	4	10	4	3	22	4	4	7	5	2	32	5	5	5
ID_EQUIPO	ID_JUGADOR	CANT_GOLES																																						
1	1	16																																						
1	3	50																																						
1	4	9																																						
1	5	8																																						
2	3	46																																						
2	4	12																																						
2	5	13																																						
3	4	10																																						
4	3	22																																						
4	4	7																																						
5	2	32																																						
5	5	5																																						

```

1 • SELECT NOMBRE_J, NOMBRE_E, CANT_GOLES
2 FROM JUGADOR J, EQUIPO E, CONTRATO C
3 WHERE J.ID_JUGADOR=C.ID_JUGADOR
4 AND E.ID_EQUIPO=C.ID_EQUIPO

```

NOMBRE_J	NOMBRE_E	CANT_GOLES
NEYMAR	REAL MADRID	16
CRISTIANO	REAL MADRID	50
AGUERO	REAL MADRID	9
KANTE	REAL MADRID	8
CRISTIANO	BARCELONA	46
AGUERO	BARCELONA	12
KANTE	BARCELONA	13
AGUERO	MANCHERTER UNITED	10
CRISTIANO	AC MILAN	22
AGUERO	AC MILAN	7
MESSI	JUVENTUS	32
KANTE	JUVENTUS	5

Tenemos una única tabla con todos los jugadores, equipos y contratos.

```
1 • SELECT NOMBRE_J, NOMBRE_E, CANT_GOLES
2 FROM JUGADOR J, EQUIPO E, CONTRATO C
3 WHERE J.ID_JUGADOR=C.ID_JUGADOR
4 AND E.ID_EQUIPO=C.ID_EQUIPO
5 AND J.NOMBRE_J='CRISTIANO'
6 AND E.NOMBRE_E='REAL MADRID';
```

NOMBRE_J	NOMBRE_E	CANT_GOLES
CRISTIANO	REAL MADRID	50