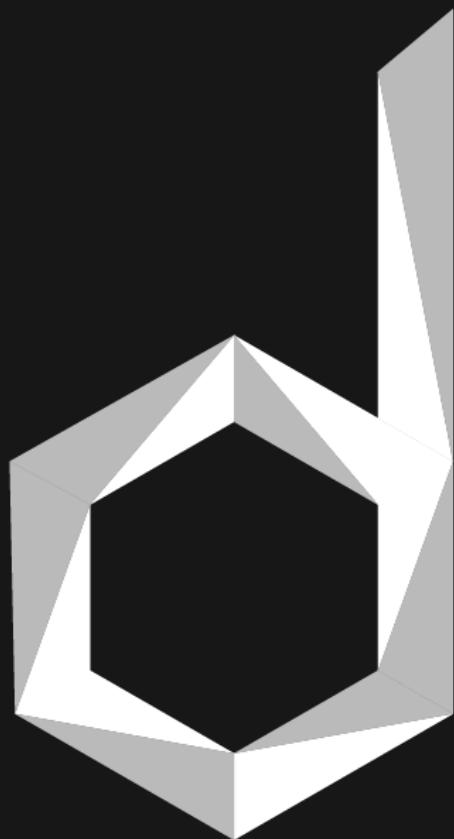


INGENIERÍA MECATRÓNICA



DI\_CERO

DIEGO CERVANTES RODRÍGUEZ

PROGRAMACIÓN: DESARROLLO BACKEND

SQL

Bases de Datos y SQL

# Contenido

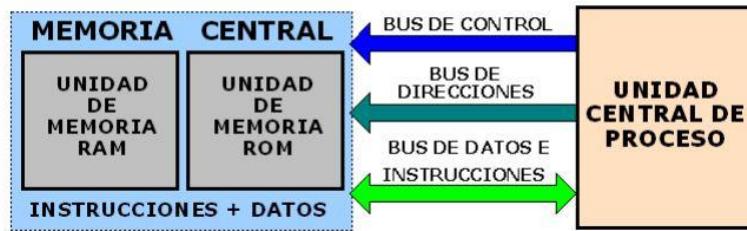
<b>Introducción a las Bases de Datos .....</b>	2
Tipos de Bases de Datos.....	2
Representación de las Bases de Datos: Nomenclatura de Chen .....	3
Diagrama ER (Entidad-Relación) .....	3
Diagrama Físico .....	7
Normalización: Tabla de Datos a Base de Datos Relacional (RDB)	9
<b>Ejemplo del Diagrama de una Base de Datos: Blog Posts .....</b>	12
<b>Manejador de Base de Datos Relacional.....</b>	15
Codificación de una Base de Datos en MySQL Workbench .....	18
<b>Base de Datos con Servicios Administrados (Nube) .....</b>	21
Google Cloud Services.....	22
<b>Lenguaje de Programación SQL .....</b>	26
Sub-lenguajes de SQL: <b>DDL (Data Definition Language)</b> .....	26
<b>CREATE:</b> .....	26
Crear una <b>Base de Datos</b> con SQL .....	26
Crear una <b>Tabla</b> con SQL.....	27
Crear una <b>Vista</b> con SQL.....	30
<b>ALTER:</b> .....	30
Alterar una <b>Tabla</b> con SQL.....	30
<b>DROP:</b> .....	32
Borrar una <b>Tabla, Columna o Base de Datos</b> con SQL .....	32
Sub-lenguajes de SQL: <b>DML (Data Manipulation Language)</b> .....	33
Referencias.....	34



# Introducción a las Bases de Datos

Las bases de datos ayudan a complementar la **arquitectura de Von Neumann**, que es la arquitectura utilizada en ordenadores, la cual a diferencia de la arquitectura Harvard utilizada en microcontroladores, **utiliza una memoria centralizada para realizar sus funciones**. La necesidad de extender la capacidad de la memoria central es la de conservar los datos más allá de la memoria RAM o ROM, ya que en la arquitectura Von Neumann si se contempla el procesamiento de datos, pero no el almacenamiento de datos persistentes, por eso es que es de suma importancia la utilización de las bases de datos.

## ARQUITECTURA VON NEUMANN



Para resolver esta situación, donde se busca que de una forma fácil se pudiera guardar y extraer la información, se concluyeron dos soluciones:

- **Bases de datos basadas en archivos:** Este método de almacenamiento de datos persistentes consiste en guardar datos en un archivo de texto plano, hojas de cálculo, etc. usualmente separados por comas o de alguna otra forma ordenada.
- **Bases de datos basadas en documentos:** En este tipo de base de datos, la unidad básica de almacenamiento es el documento, que puede contener datos en forma de texto, números, listas, objetos JSON (JavaScript Object Notation) y a veces incluso otros documentos anidados.

## Tipos de Bases de Datos

Los diferentes tipos de bases de datos existentes son los siguientes:

- **Relacionales o RDB:** Son bases de datos basadas en documentos que se rigen por las 12 reglas de Edgar Codd, que dan como resultado el álgebra relacional, a través de las cuales se indican las reglas con las que los datos de las RDB se pueden mezclar o relacionar entre sí.
  - **Privadas:** Microsoft SQL Server, Oracle, etc.
  - **Open Source:** PostgreSQL, MySQL, MariaDB, etc.

### Ejemplos de bases de datos relacionales



- **No relacionales:**
  - Memcached, Cassandra (Facebook), DynamoDB, ElasticSearch, BigQuery, Neo4j (GraphQL), MongoDB, Firestore (Firebase).

## Bases de datos no relacionales



- **Auto Administradas:** En este tipo de bases de datos se instala, actualiza y mantiene el software en un ordenador de forma local y la consistencia de datos se realiza de forma manual.
- **Administradas:** Este tipo de base de datos se ofrece por las nubes modernas como las proporcionadas por Amazon, Google, Azure (Microsoft), para ello la instalación no se realiza de forma local y, por lo tanto, no se mantiene la consistencia de datos de forma manual, sino que se realiza de forma automática por el servicio de la nube.

Representación de las Bases de Datos: Nomenclatura de Chen

### Diagrama ER (Entidad-Relación)

- **Entidad:** Una entidad es algo muy similar a un objeto, el cual se puede asociar con ciertos **atributos (características)**, de la misma forma como se maneja en POO.
  - **Atributo:** Se representa por medio de un **óvalo simple** cuando la entidad solo posee **uno solo de ese atributo**, si cuenta con más de uno, esto se indica con **dos óvalos anidados** que rodeen el nombre del atributo, a esto se le llama **atributo multivalor**.
    - **Ejemplo 1:** Cualquier automóvil posee un solo volante, pero varias llantas, por lo cual el atributo "volante" será rodeado por un óvalo simple y el atributo "llantas" se rodeará de un óvalo doble.
    - **Ejemplo 2:** Ahora se representará a través de un diagrama de Chen las entidades (objetos) laptops, donde cabe mencionar que **los atributos donde se subraye su nombre** se llaman **atributos clave** y diferencian cada laptop individualmente (instancia), además los atributos que tengan un **óvalo con línea punteada** representan los **atributos derivados**, que corresponden a datos que se pueden obtener a través de otros y además que los atributos pueden tener otros atributos relacionados.

## Entidades



## Atributos

no de serie	color	año	pantalla
LKJ789JKAS	gris	2017	AX4829i
KCO3100KJH	negro	2019	AX4930i
NSDJOIH128	negro	2018	AX4930i
09KSIHBD71	gris	2017	AX4829i

- **Tipos de atributos clave:** Los atributos clave pueden ser naturales, esto significa que son pertenecientes al objeto y no se pueden remover y los atributos clave artificiales, que se asignan de manera arbitraria.
- **Entidades fuertes:** No dependen de otra entidad para existir, estas se rodean de un cuadrado simple.
  - **Entidades débiles:** Si dependen de otra entidad para existir, estas se rodean de un cuadrado doble, así como los atributos multivalor. Además, existen dos tipos de debilidad:
    - **Debilidad por identidad:** Que para que se puedan diferenciar, deben tomar el atributo clave de la entidad de la que dependen.
    - **Debilidad por existencia:** Que pueden tener un identificador propio, pero aun así dependen de otra identidad para existir.

## Entidades débiles



## Entidades débiles: identidad

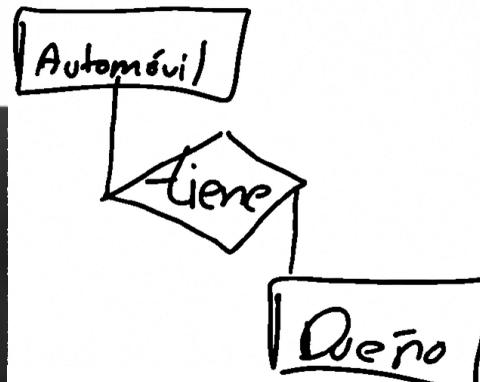
Libros			Ejemplares		
id	título	...	libro_id	localización	edición
LKJ789JKAS	Viaje al cent...	...	LKJ789JKAS	pasillo 1	1
KCO3100KJH	El señor de ...	...	KCO3100KJH	pasillo 1	1
NSDJOIH128	De la tierra...	...	NSDJOIH128	pasillo 1	3
09KSIHBD71	Amor en tie...	...	09KSIHBD71	pasillo 1	1

## Entidades débiles: existencia

Libros			Ejemplares		
id	título	...	id	localización	edición
LKJ789JKAS	Viaje al cent...	...	JKE7823CLK	pasillo 1	1
KCO3100KJH	El señor de ...	...	JKFE1093JD	pasillo 1	1
NSDJOIH128	De la tierra...	...	82938ISHDIK	pasillo 1	3
09KSIHBD71	Amor en tie...	...	838439JHDUI	pasillo 1	1

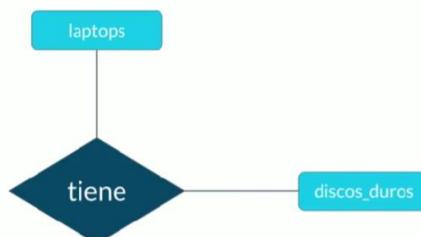
- **Relación:** Es la conexión con las que se ligan las diferentes entidades entre sí, para ello dentro de las relaciones se utilizan verbos que conecten una relación con la otra.

## Relaciones



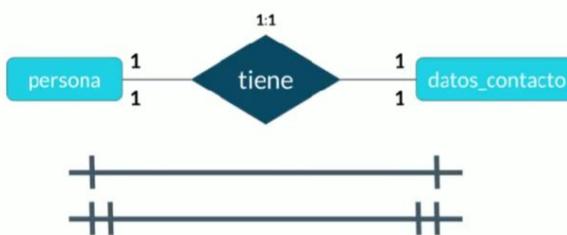
- **Cardinalidad:** Una peculiaridad de las relaciones es que a través de ellas se deben separar los atributos multivaluados, ya que cada uno puede tener características específicas y se relacionan con el concepto de cardinalidad porque este se relaciona con el número de veces que se repite un atributo en una entidad.
  - Cuando se utiliza la cardinalidad se utiliza todo el concepto del verbo de conexión que utiliza la relación para indicar cual es el número de entidades con las que cuenta una entidad, de esta manera es como se puede separar los atributos multivaluados.

## Relaciones

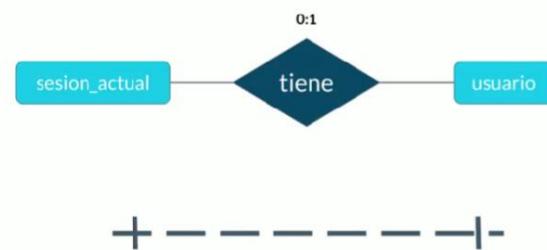


- De igual manera se obtiene la cardinalidad de ambos lados, ya que esto puede variar cuando se ve desde perspectivas distintas. Además, se maneja cierta nomenclatura en el diagrama para denotarlo.

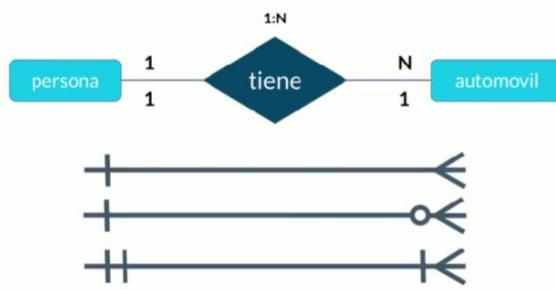
### Cardinalidad: 1 a 1



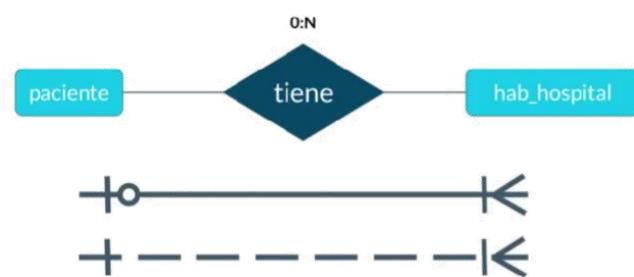
### Cardinalidad: 0 a 1



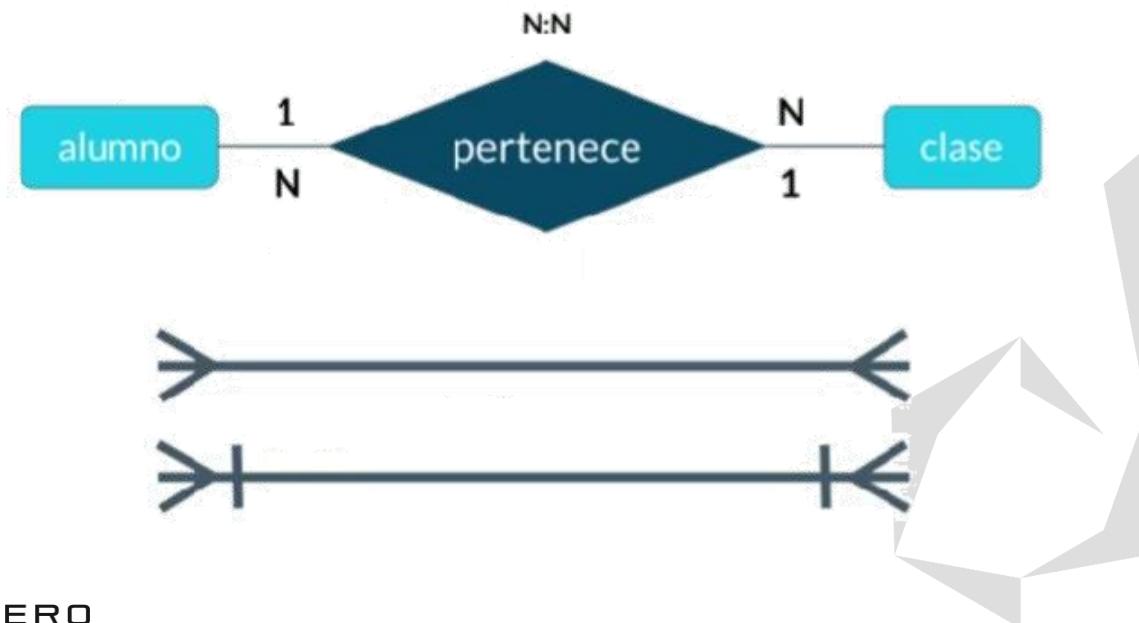
### Cardinalidad: 1 a N



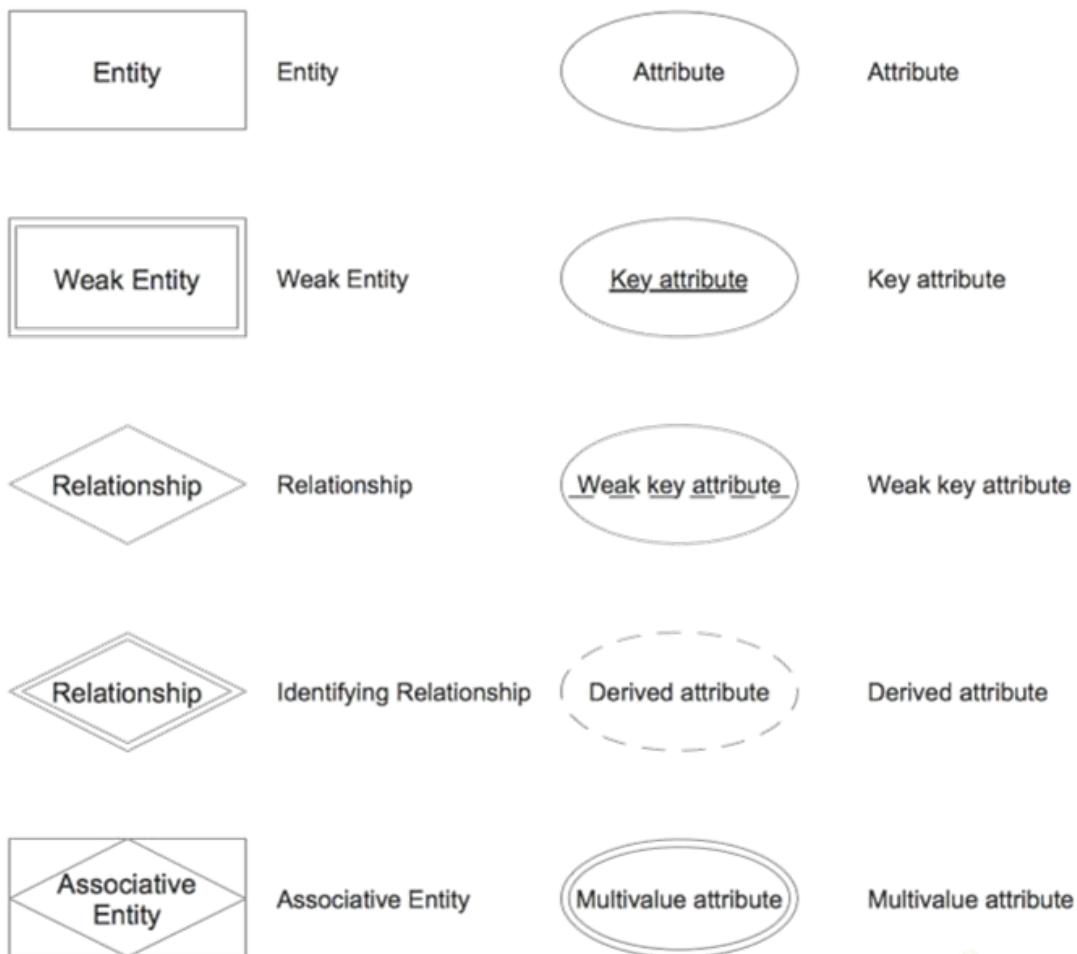
### Cardinalidad: 0 a N



### Cardinalidad: N a N



Todos los conceptos explicados previamente que describen los datos almacenados en una base de datos relacional se deben plasmar en un diagrama ER (entidad-relación), para ello se utiliza la siguiente simbología:



### Diagrama Físico

Además del **diagrama ER (entidad-relación)** existe otro llamado **diagrama físico** (que se deriva del ER), el cual es más específico ya que **menciona los tipos de datos**, que pueden ser los siguientes:

- **Texto:**
  - **Char(n):** Minimiza el espacio de memoria a solo los caracteres que ocupa el texto.
  - **VarChar(n):** Utiliza el espacio de memoria de forma dinámica, reservando como mínimo un espacio de memoria y extendiéndolo si es necesario hasta 255 caracteres.<sup>3</sup>
  - **Text:** Reserva el espacio de memoria para cadenas de caracteres (palabras u oraciones) muy grandes.
- **Números:**
  - **Enteros:** Integer, BigInt y SmallInt.
  - **Decimales:** Decimal(n, s) y Numeric(n,s ), donde n es el número y s indica cuantos decimales aparecen de dicho número.

- **Fecha/Hora:**
  - **Date:** Contiene año, fecha y día.
  - **Time:** Contiene solo la hora.
  - **Datetime y Timestamp:** Contienen la fecha y la hora.
- **Lógicos:**
  - **Boolean:** Puede adoptar valores true (1) o false (0).

## Tipos de dato

Texto	Números	Fecha/hora	Lógicos
CHAR(n)	INTEGER	DATE	BOOLEAN
VARCHAR(n)	BIGINT	TIME	
TEXT	SMALLINT	DATETIME	
	DECIMAL(n, s)	TIMESTAMP	
	NUMERIC (n, s)		

Además del tipo de dato, se indican las restricciones (reglas) de la base de datos que delimitan el tipo de dato que admite, cuántos datos admite, etc.

## Constraints (Restricciones)

Constraint	Descripción
NOT NULL	Se asegura que la columna no tenga valores nulos
UNIQUE	Se asegura que cada valor en la columna no se repita
PRIMARY KEY	Es una combinación de NOT NULL y UNIQUE
FOREIGN KEY	Identifica de manera única una tupla en otra tabla
CHECK	Se asegura que el valor en la columna cumpla una condición dada
DEFAULT	Coloca un valor por defecto cuando no hay un valor especificado
INDEX	Se crea por columna para permitir búsquedas más rápidas

- **Índice:** Es un elemento cuya ventaja es que permite realizar búsquedas de datos en la columna de una tabla de una base de datos, pero la desventaja que tiene es que hace lento el procesamiento de datos en esa columna. Por lo que su mayor utilidad es cuando en una base de datos se estarán realizando consultas constantes, pero no se introducirán datos nuevos de forma continua.

## Normalización: Tabla de Datos a Base de Datos Relacional (RDB)

El proceso de normalización permite obtener una base de datos a partir de una tabla de datos, separándolas en los componentes previamente explicados, como lo son las entidades, atributos, etc., para ello se aplican las 12 reglas del álgebra relacional de Codd, también llamadas formas normales o FN, que establecen la base de datos como relacional.

## Normalización



A continuación, se denotará este concepto con un ejemplo, donde partiendo de una tabla de datos, estos se organizarán para ser normalizados:

## Sin normalizar

alumno	nivel_curso	nombre_curso	materia_1	materia_2
Juanito	Maestría	Data engineering	MySQL	Python
Pepito	Licenciatura	Programación	MySQL	Python

Las formas normales que se siguen para normalizar la tabla son las siguientes:)

- **1FN (Primera Forma Normal) - Atributos atómicos:** Esta norma indica que no se pueden tener campos repetidos y sus atributos deben ser atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.

alumno	nivel_curso	nombre_curso	materia_1	materia_2
--------	-------------	--------------	-----------	-----------

Juanito	Maestría	Data engineering	MySQL	Python
---------	----------	------------------	-------	--------

Pepito	Licenciatura	Programación	MySQL	Python
--------	--------------	--------------	-------	--------

alumnos				
alumno_id	alumno	nivel_curso	nombre_curso	materia
1	Juanito	Maestría	Data engineering	MySQL
1	Juanito	Maestría	Data engineering	Python
2	Pepito	Licenciatura	Programación	MySQL
2	Pepito	Licenciatura	Programación	Python

- **2FN (Segunda Forma Normal) - Clave Única:** Esta norma indica que cada campo de la tabla debe depender de una clave única, si no es posible, se debe separar en entidades distintas.

alumnos				
alumno_id	alumno	nivel_curso	nombre_curso	materia
1	Juanito	Maestría	Data engineering	MySQL
1	Juanito	Maestría	Data engineering	Python
2	Pepito	Licenciatura	Programación	MySQL
2	Pepito	Licenciatura	Programación	Python

alumnos			
alumno_id	alumno	nivel_curso	nombre_curso
1	Juanito	Maestría	Data engineering
2	Pepito	Licenciatura	Programación

materias		
materia_id	alumno_id	materia
1	1	MySQL
2	1	Python
3	2	MySQL
4	2	Python



- **3FN (Tercera Forma Normal) - Campos Clave Sin Dependencias:** Esta norma indica que los campos clave no deben tener dependencias, osea que aquellos datos que no pertenecen a la entidad deben tener una independencia de las demás y un campo clave propio.

alumnos			
alumno_id	alumno	nivel_curso	nombre_curso
1	Juanito	Maestría	Data engineering
2	Pepito	Licenciatura	Programación

materias		
materia_id	alumno_id	materia
1	1	MySQL
2	1	Python
3	2	MySQL
4	2	Python

alumnos		
alumno_id	alumno	curso_id
1	Juanito	1
2	Pepito	2

cursos		
curso_id	nivel_curso	nombre_curso
1	Maestría	Data engineering
2	Licenciatura	Programación

materias		
materia_id	alumno_id	materia
1	1	MySQL
2	1	Python
3	2	MySQL
4	2	Python

- **4FN (Cuarta Forma Normal) - Campos Multivaluados:** Esta norma indica que los campos multivaluados deben ser identificados y separados por una clave única, evitando así que se repitan en cada entidad.

alumnos		
alumno_id	alumno	curso_id
1	Juanito	1
2	Pepito	2

cursos		
curso_id	nivel_curso	nombre_curso
1	Maestría	Data engineering
2	Licenciatura	Programación

materias	
materia_id	materia
1	MySQL
2	Python

materias_por_alumno		
mpa_id	materia_id	alumno_id
1	1	1
2	2	1
3	1	2
4	2	2

## Ejemplo del Diagrama de una Base de Datos: Blog Posts

Para crear una base de datos, primero que nada, debemos pensar en las **entidades** que se utilizarán en ella y posteriormente deberemos pensar en los **atributos** que le pertenecen, todo esto se coloca en un **diagrama ER (entidad-relación)** para modelar cómo los datos se relacionan entre sí. En el caso del blog post son los siguientes:

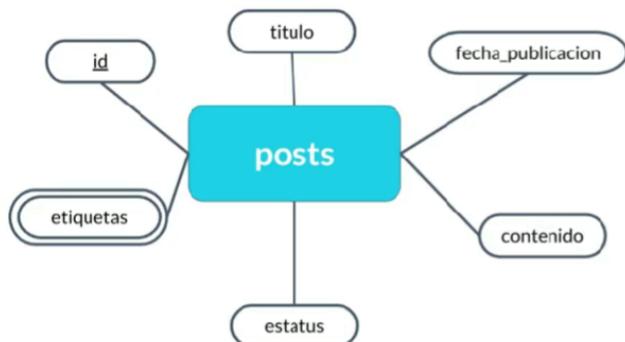
- **Entidades:**
  - Posts (Publicaciones).
  - Usuarios.
  - Comentarios.
  - Categorías.
  - Etiquetas.

### Diagrama ER: Platziblog



- **Atributos de las Entidades:**
  - **Atributos Entidad Posts:**
    - Título.
    - Fecha\_publicacion.
    - Contenido.
    - Estatus (Check Activo o Inactivo).
    - Etiquetas (Categoría interna).
    - Id (Clave única) (Primary Key o PK).

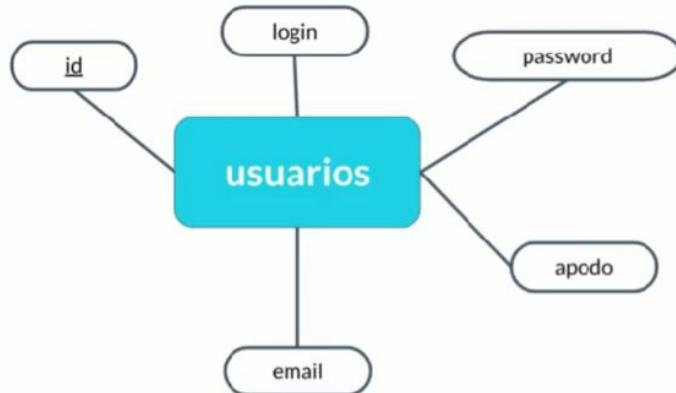
### Entidades Platziblog



- **Atributos Entidad Usuarios:**

- Login (Nombre de Usuario) (**Not Null o NN**).
- Password (**Not Null o NN**).
- Nickname (**Not Null o NN**).
- Email (**Not Null o NN y Unique**).
- Id (**Primary Key o PK**).

## Entidades Platziblog

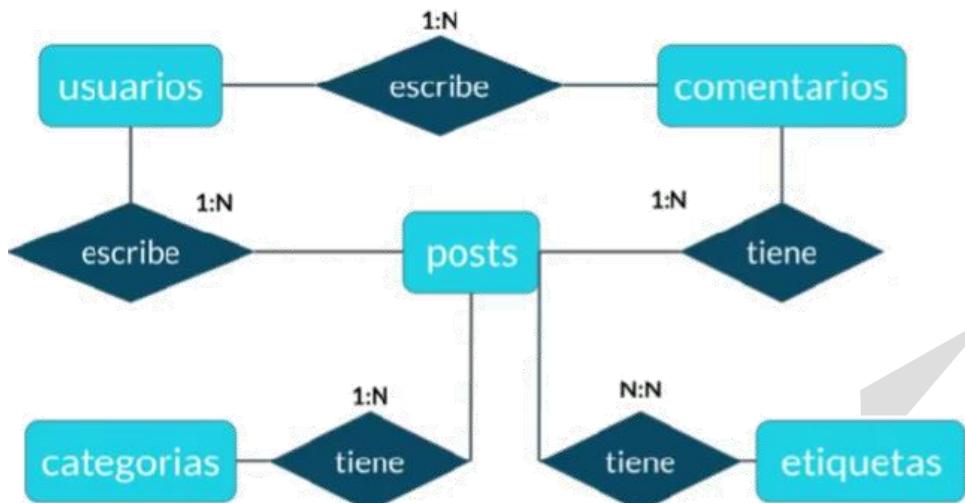


- **Diagrama ER (Entidad-Relación):**

- **Relaciones y Cardinalidad:**

- Un usuario tiene (puede escribir) varios posts.
- Un usuario tiene (puede escribir) varios comentarios.
- Un post tiene varios comentarios.
- Una categoría tiene (engloba) varios posts.
- Un post tiene varias etiquetas y una etiqueta tiene (engloba) varios posts.

## Diagrama ER: Platziblog



- **Diagrama Físico (Tipo de Dato y Constraints):**

- **Tipos de Datos:**

Texto	Números	Fecha/hora	Lógicos
CHAR(n)	INTEGER	DATE	BOOLEAN
VARCHAR(n)	BIGINT	TIME	
TEXT	SMALLINT	DATETIME	
	DECIMAL(n, s)	TIMESTAMP	
	NUMERIC (n, s)		

- **Constraints (Restricciones):**

- **PK: Primary Key.**

- Esta clave debe estar ligada con una foreign key de alguna entidad que sea dependiente de ella.

- **FK: Foreign Key.**

- Esta clave debe estar ligada con una primary key para ver de qué entidad depende, para ello nos debemos fijar en la cardinalidad del diagrama.

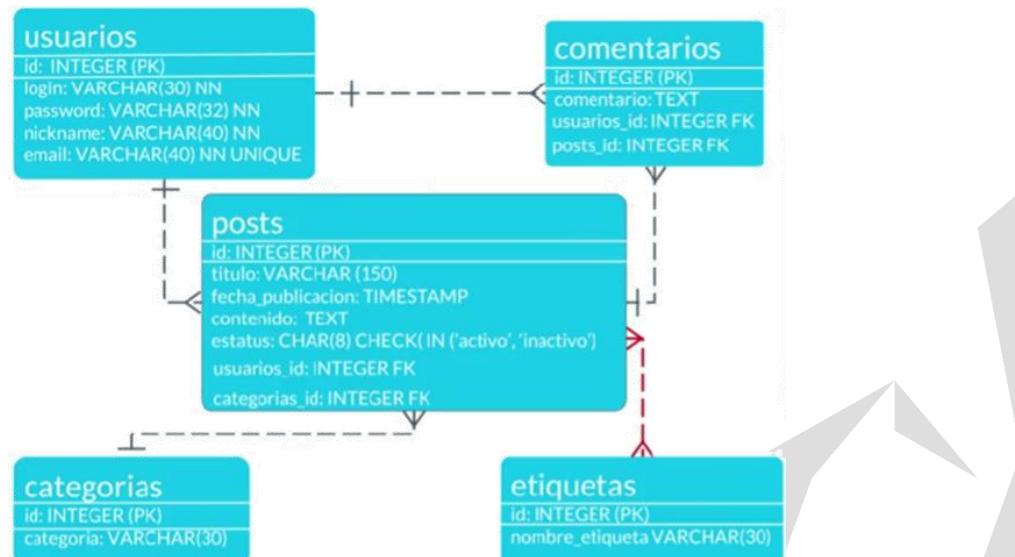
- **NN: Not Null.**

- **UNIQUE.**

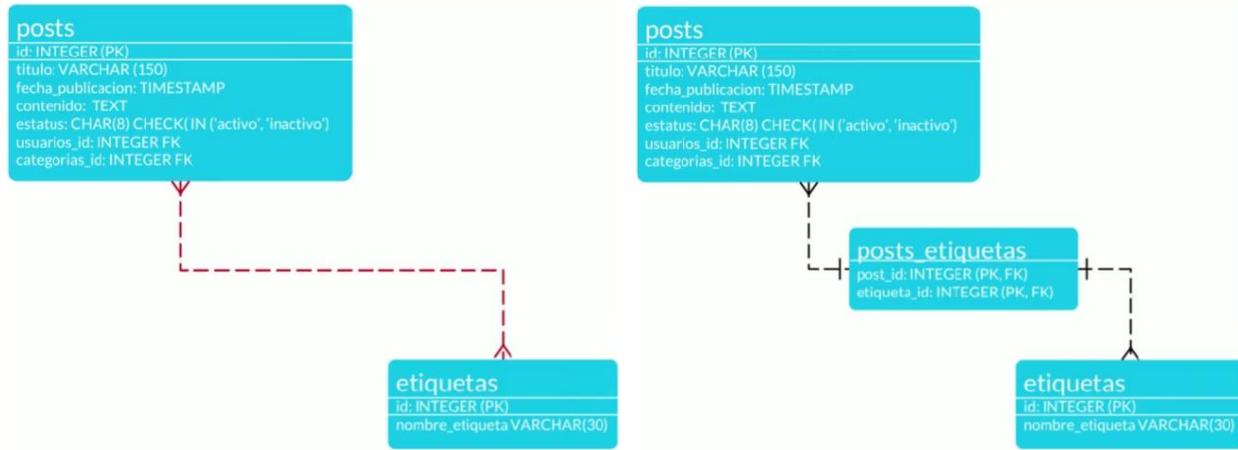
- **CHECK.**

Constraint	Descripción
NOT NULL	Se asegura que la columna no tenga valores nulos
UNIQUE	Se asegura que cada valor en la columna no se repita
PRIMARY KEY	Es una combinación de NOT NULL y UNIQUE
FOREIGN KEY	Identifica de manera única una tupla en otra tabla
CHECK	Se asegura que el valor en la columna cumpla una condición dada
DEFAULT	Coloca un valor por defecto cuando no hay un valor especificado
INDEX	Se crea por columna para permitir búsquedas más rápidas

- **Diagrama relacional global sin cardinalidad N:N:**



- **Diagrama N:N Intermedio o de Pivote:** Este se utiliza cuando se tiene una relación con cardinalidad de N:N entre dos entidades, para ello se debe agregar un diagrama intermedio que relacione ambos id. Es importante mencionar que para crear las llaves únicas de estas relaciones se deben combinar ambos id.



## Manejador de Base de Datos Relacional

Para poder experimentar con bases de datos se puede instalar un manejador de bases de datos relacionales (RDBMS o Relational Data Base Manager System) en nuestro sistema operativo Windows, aunque ya que se maneja una base de datos en producción, normalmente se utilizan servicios de nube.

MySQL es de los manejadores open source más populares del mercado, razón por la cual es el elegido para utilizarse, para ello debemos descargar la versión del instalador 5.7, que se encuentra disponible en el siguiente enlace:

<https://dev.mysql.com/downloads/windows/installer/5.6.html>

### ④ MySQL Community Downloads

◀ MySQL Installer

MySQL Installer 5.7.44

**Note:** MySQL 8.0 is the final series with MySQL Installer. As of MySQL 8.1, use a MySQL product's MSI or Zip archive for installation. MySQL Server 8.1 and higher also bundle MySQL Configurator, a tool that helps configure MySQL Server.

Select Version: 5.7.44

Select Operating System: Microsoft Windows

Windows (x86, 32-bit), MSI Installer	5.7.44	2.1M	Download
(mysql-installer-web-community-5.7.44.0.msi)	MDS: 6cc27e2a42a54b593a9d3544f2529a53   Signature		
Windows (x86, 32-bit), MSI Installer	5.7.44	373.7M	Download
(mysql-installer-community-5.7.44.0.msi)	MDS: e89af3ba9b4716ff5e647b0fd2edab2   Signature		



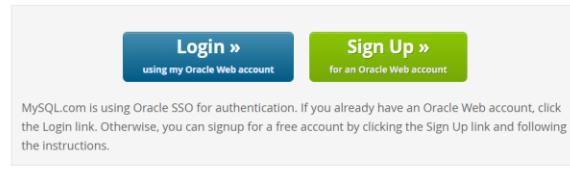
Después daremos clic en el botón de seguir con la descarga, o si queremos también nos podemos registrar a la plataforma de MySQL.

## MySQL Community Downloads

Login Now or Sign Up for a free account.

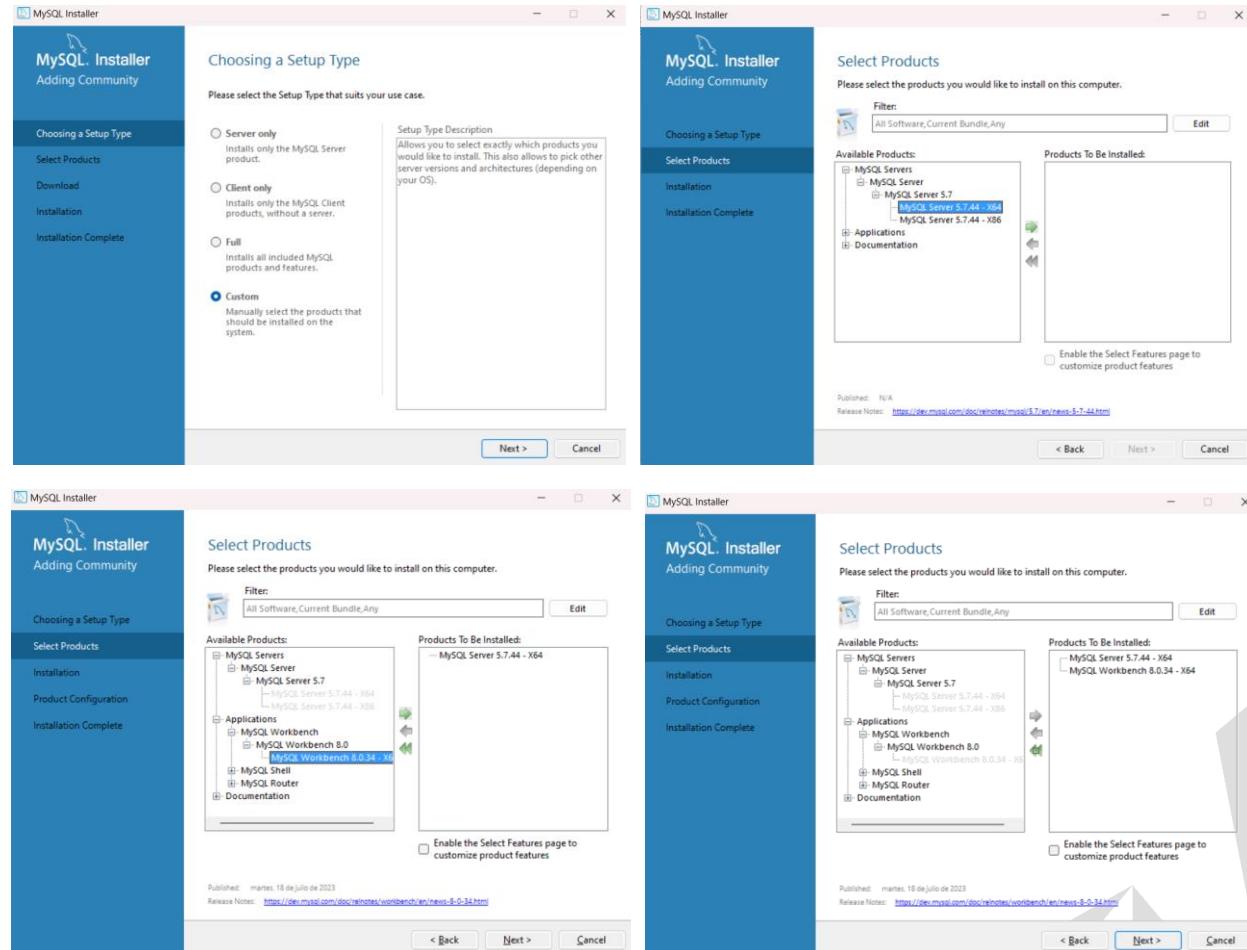
An Oracle Web Account provides you with the following advantages:

- 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

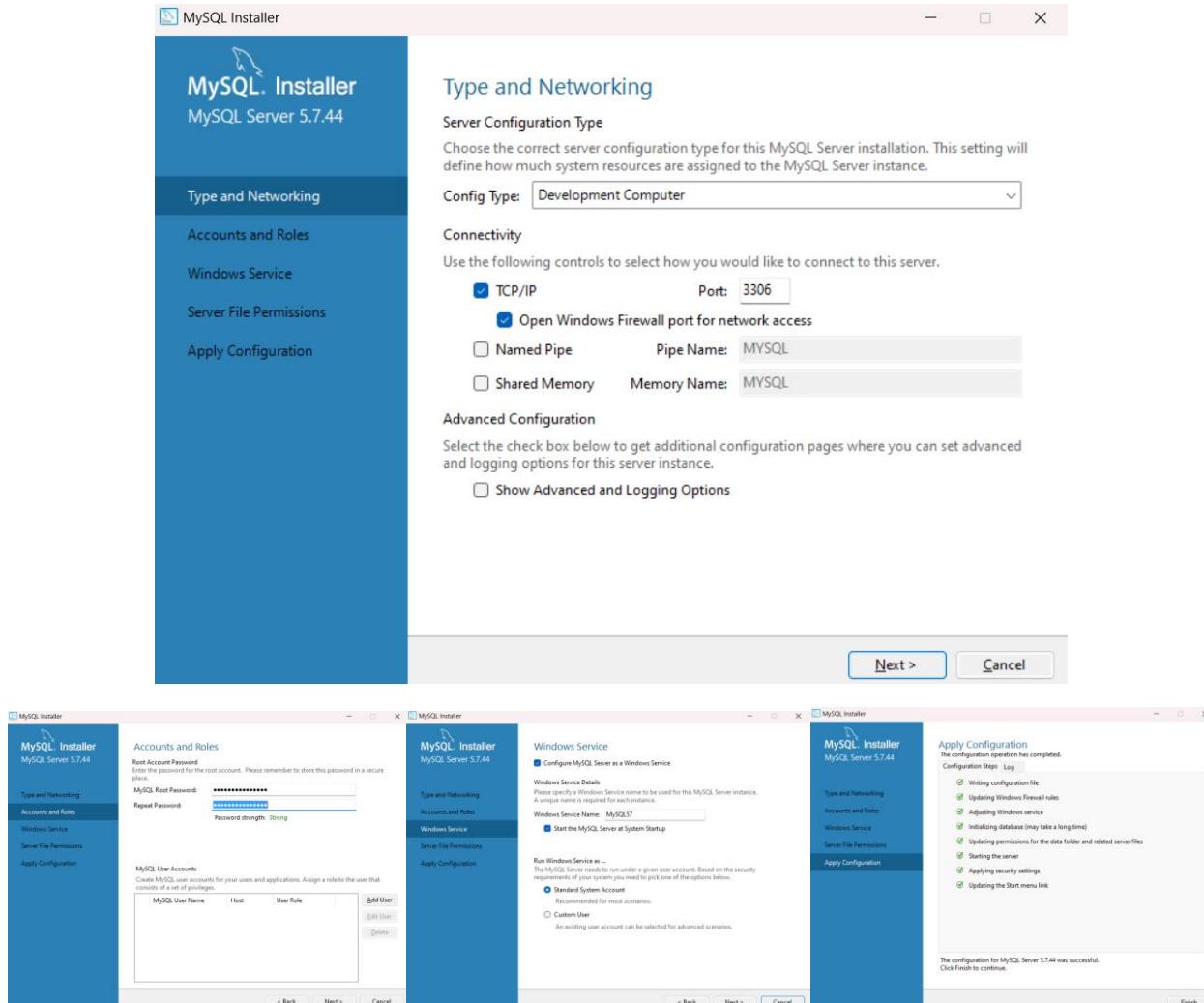


[No thanks, just start my download.](#)

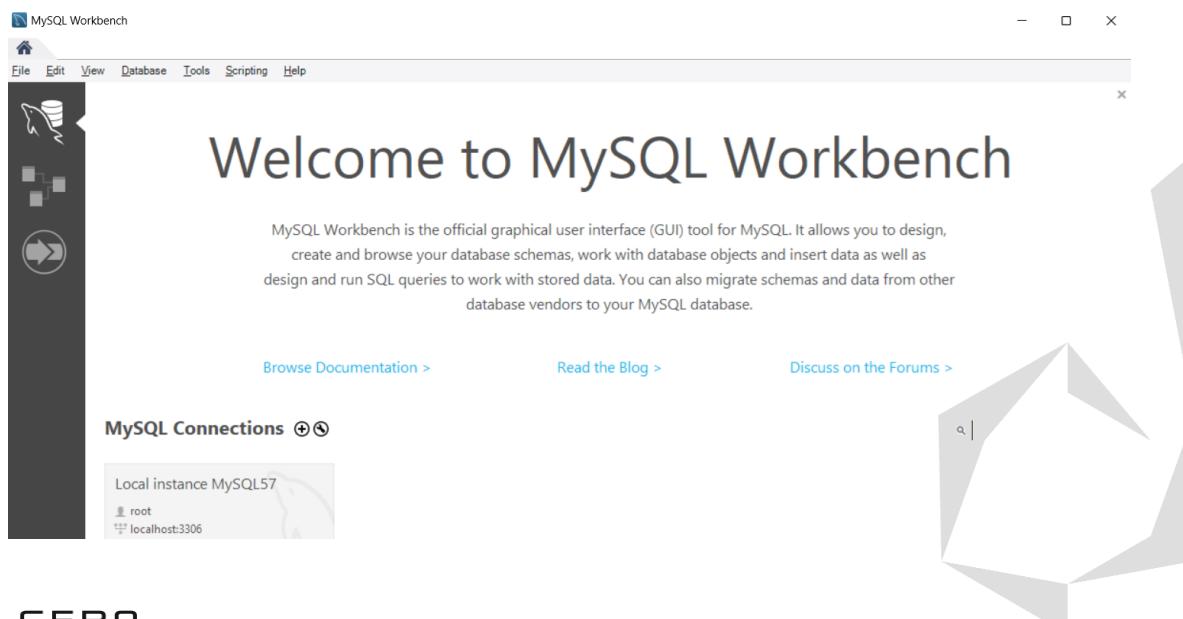
Luego daremos clic en la instalación Custom → MySQL Servers → MySQL Server 5.7 → MySQL Server 5.7 - X64 → → → Applications → MySQL Workbench → MySQL Workbench → Next → Execute.



Config Type → Development Computer → Port → Puerto Deseado, este lo deberemos recordar → Next → **MySQL Root Password:** Contraseña del usuario principal de la DB, tiene todos los permisos → Execute.

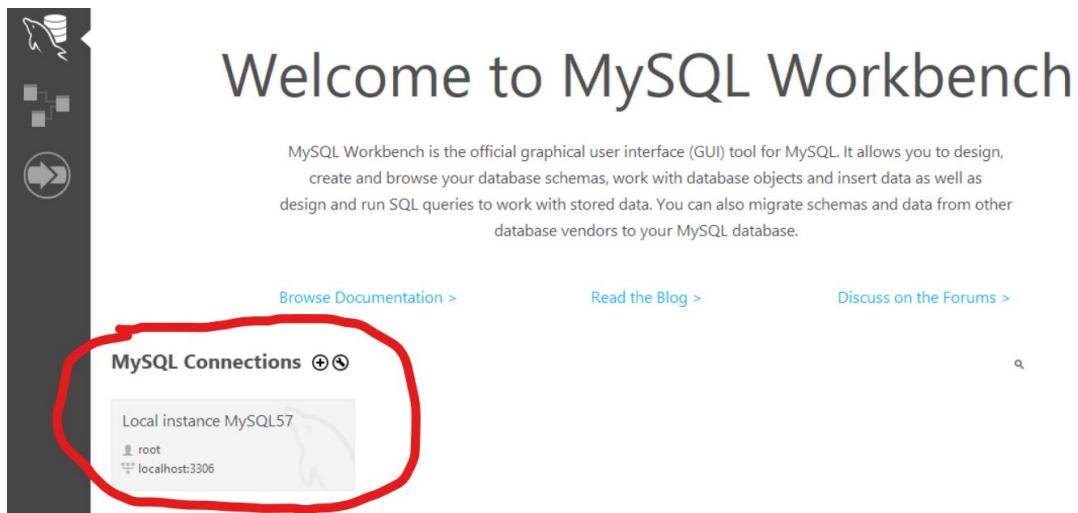


Al finalizar este procedimiento ya estaré instalado el sistema de manejo de la base de datos RDBMS llamado MySQL Workbench, el cual es llamado cliente gráfico y permite ver cómo funciona la base de datos internamente en forma de tablas.

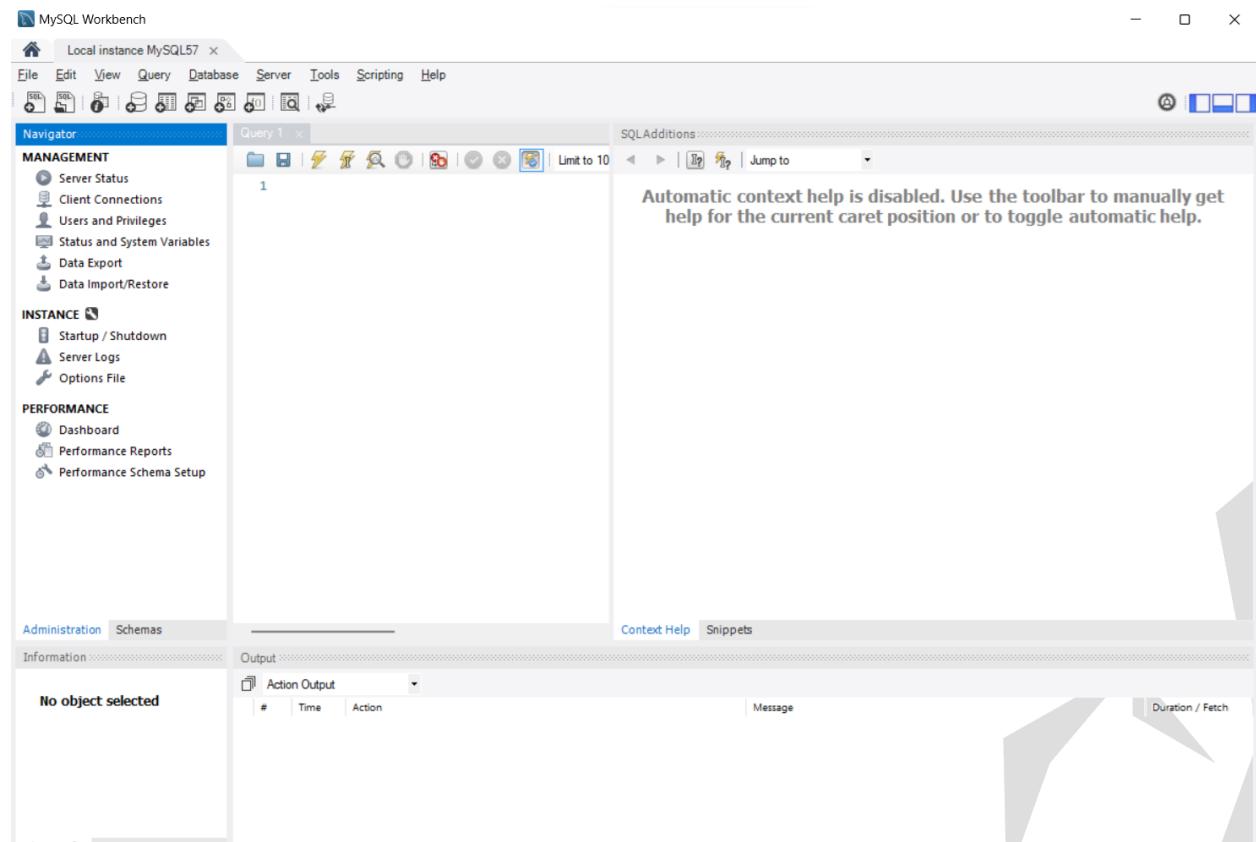


## Codificación de una Base de Datos en MySQL Workbench

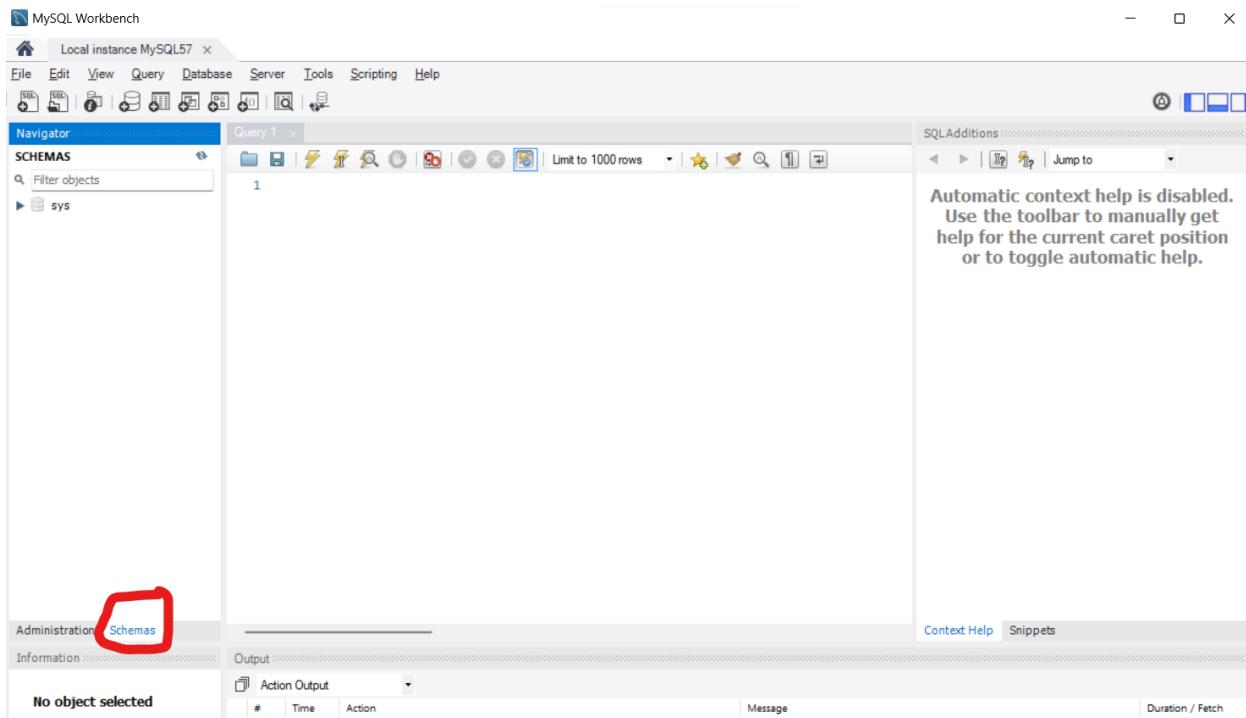
En la pantalla principal del RDBMS MySQL Workbench se puede observar una parte que dice MySQL Connections, en esta se muestran todas las bases de datos que se pueden manejar con este cliente gráfico, indicando su nombre, nombre de usuario (root) y puerto. En esta misma parte si se quisiera agregar una base de datos adicional, se podría realizar al dar clic en el símbolo de +.



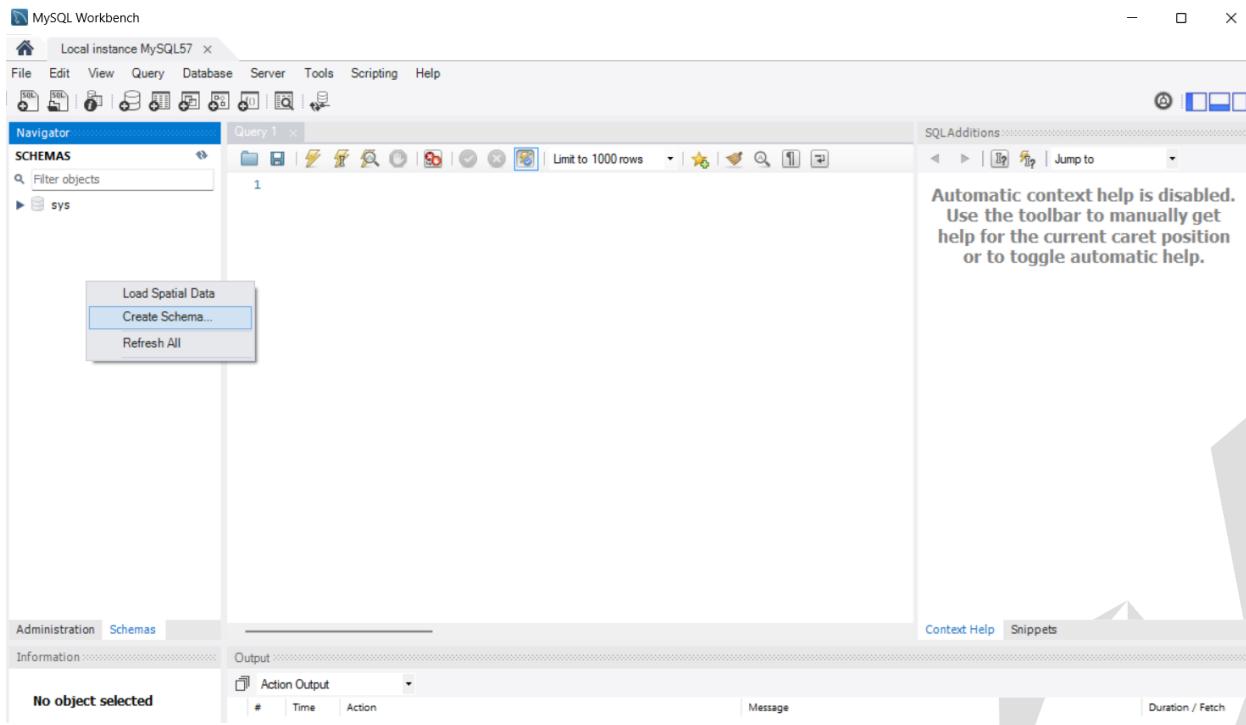
Al dar clic en estas conexiones a las bases de datos, me pedirá la contraseña de acceso y después de dársela podremos ya manejar la base de datos a través del lenguaje SQL.



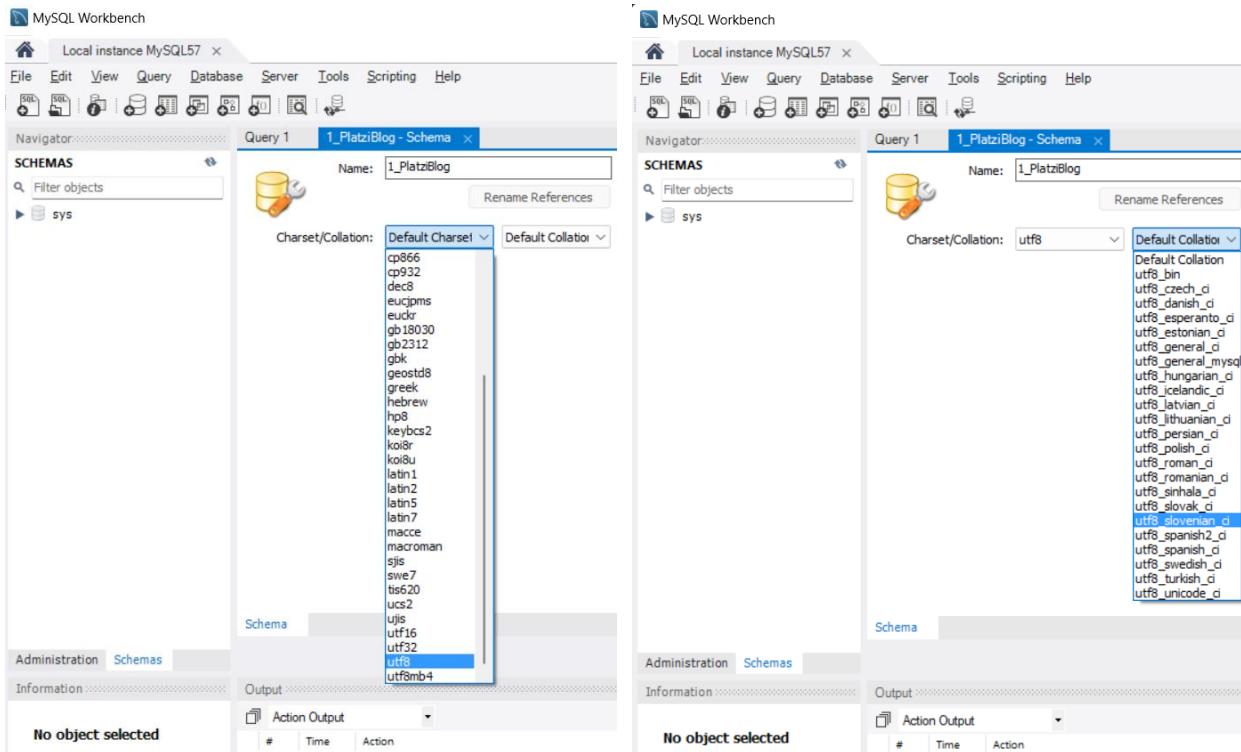
Para acceder a la información de las bases de datos, debemos dar clic en la pestaña de Schemas, ya que esta es una forma alterna de referirnos a ellas:



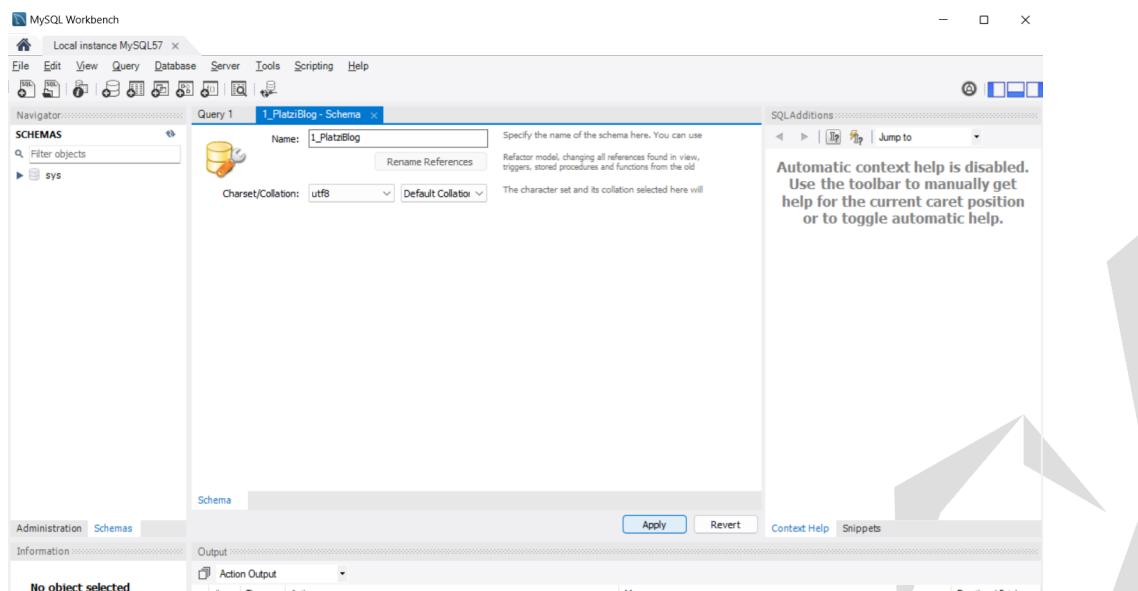
Para crear una base de datos nueva, daremos clic derecho sobre el área de trabajo de los Schemas y seleccionaremos la opción de Create Schema...



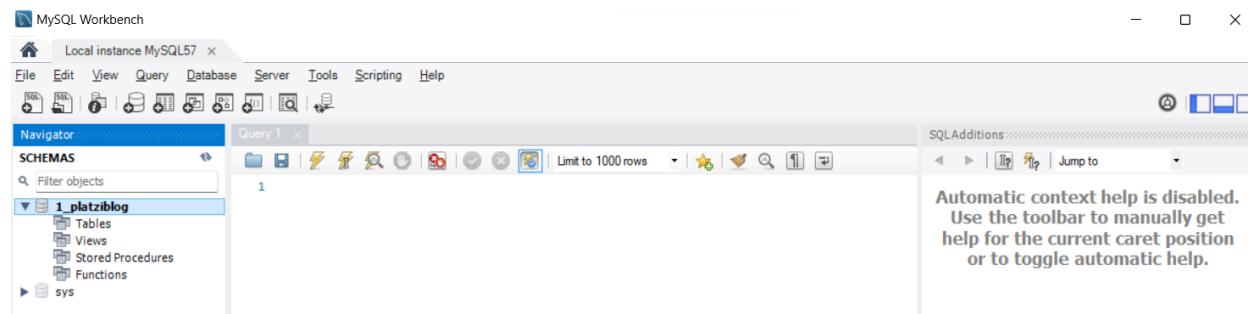
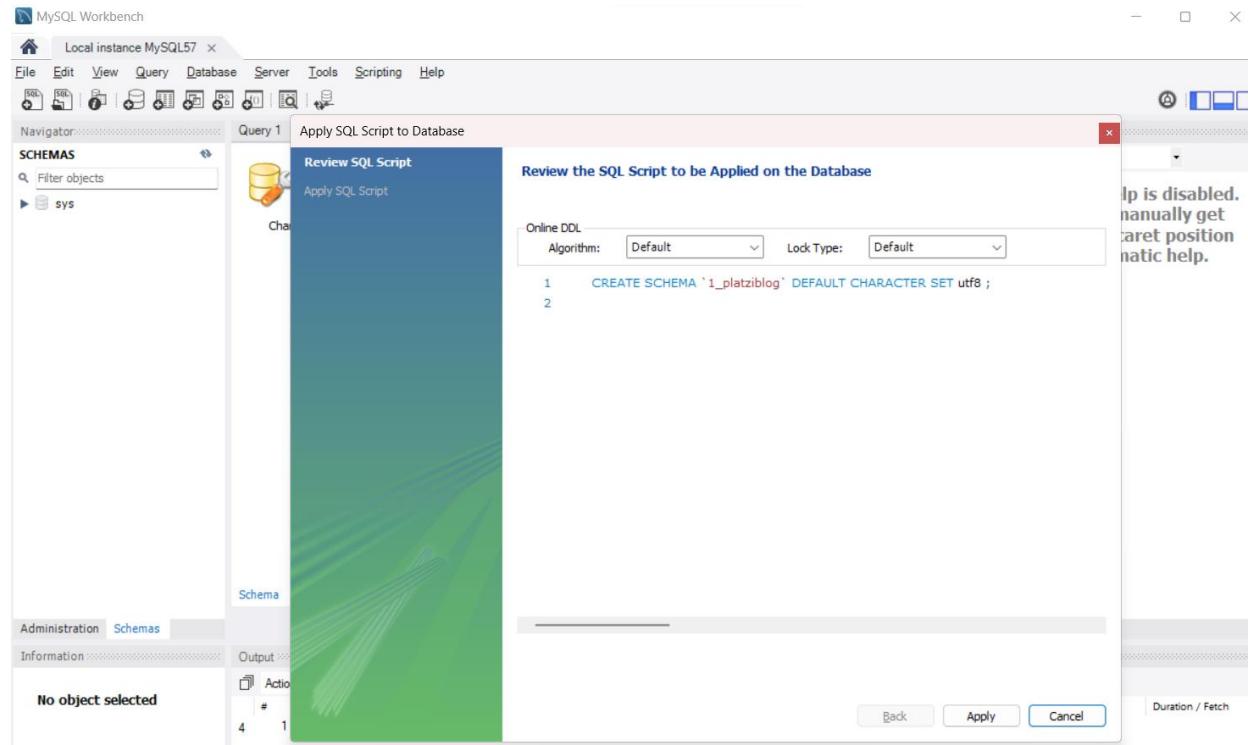
Luego asignaremos un nombre para la base de datos y seleccionaremos la opción de Charset/Collation para indicar la declaración de codificación o codificación de caracteres, donde al especificar la opción utf-8 (caracteres Unicode) como la codificación, nos aseguramos de que el archivo pueda contener caracteres especiales, letras acentuadas y otros caracteres no ASCII sin problemas (recordemos que esta misma codificación la indicábamos al inicio de los scripts creados en el lenguaje de programación Python), además podríamos hasta especificar el idioma que se utilizará en la base de datos, pero en este caso eso no se indicará.



Ya que se haya especificado la codificación, se dará clic en el botón de Apply.



Al haber dado clic en el botón de Apply, aparecerá una ventana donde se indica la configuración indicada a través de lenguaje SQL, a esta ventana se le da clic de nuevo en el botón de Apply → Finish y al hacerlo aparecerá una nueva pestaña en el área de trabajo de SCHEMAS con el nombre indicado.



## Base de Datos con Servicios Administrados (Nube)

Muchas empresas no utilizan un manejador de bases de datos relacionales local (RDBMS o Relational Data Base Manager System) para gestionar sus bases de datos, sino que utilizan algo llamado Servicio Administrado o Cloud, el cual se encuentra en la nube y por ello no se debe administrar de forma manual la base de datos en cuestiones de seguridad, actualizaciones, crecimiento, sistema operativo, redes, etc. solo se centran en su funcionalidad, pero no en su hardware o mantenimiento. Algunos ejemplos muy famosos de ello son AWS, Google Cloud Platform, Azure, etc. pero su desventaja es que la mayoría de ellos son servicios de pago, no open source para desarrolladores, razón por la cual están más dirigidos a empresas

## Google Cloud Services

Para este ejemplo se utilizará el periodo de prueba de Google Cloud, el cual se encuentra en el siguiente enlace, donde al ingresar me llevará por default al último proyecto de Google que haya creado, pero como este será un proyecto nuevo, deberé dar clic en el desplegable que tiene el nombre del proyecto y dar clic en el botón de PROYECTO NUEVO:

<https://console.cloud.google.com/>

The screenshot shows two views of the Google Cloud Console. The top view is the 'Selecciona un proyecto' (Select a project) screen, which lists existing projects: 'Asistente Virtual' (selected), 'My Project', and 'App Enerdrails'. Below this is the main dashboard with a 'Te damos la bienvenida' (Welcome) message and a 'Crea una VM' (Create a VM) button. The bottom view is the 'Proyecto nuevo' (New project) creation dialog, where a new project named '1-PlatziBlog DB' is being created. The dialog includes fields for project name, location ('Sin organización'), and quota management.

Ya que se haya creado y seleccionado el proyecto de Servicios Administrados del Platzi Blog, daremos clic en el menú de la esquina superior izquierda y seleccionaremos la opción de SQL.

The screenshot shows the Google Cloud Console with the navigation menu open on the left. The 'SQL' option is selected. The main area displays a welcome message and several quick actions: 'Crea una VM', 'Ejecuta una consulta en BigQuery', 'Crea un clúster de GKE', and 'Crea un bucket de almacenamiento'. A large, stylized 'SQL' logo is prominently displayed at the bottom right.

Como el servicio de base de datos de Google Cloud, después de cierto número de datos empieza a cobrar, pedirá que ingreses tus datos bancarios, aunque en un inicio no cobrará nada, solo hasta pasar cierto umbral de número de datos almacenados. Ya que se hayan ingresado, daremos clic al botón de Crear Instancia con Créditos Gratuitos → MySQL → Habilitar API.



### Elige tu motor de base de datos

MySQL Versiones: 8.0, 5.7, 5.6 <a href="#">Elegir MySQL</a>	PostgreSQL Versiones: 15, 14, 13, 12, 11, 10, 9.6 <a href="#">Elegir PostgreSQL</a>	SQL Server Versiones: 2022, 2019, 2017 <a href="#">Elegir SQL Server</a>
---	---	--

[¿Quieres obtener más información sobre los motores de base de datos de Cloud SQL?](#)  
[Más información](#)

Ahora daremos clic en Instance ID para asignar un nombre a la base de datos y luego indicaremos su contraseña, los demás parámetros como zona y región se refieren a la ubicación del servidor, que por el momento no es de suma importancia. Además, me obliga a que elija un plan de cobro dependiendo del servicio, pero cabe mencionar que la versión de prueba incluye 300 USD en forma de créditos, por lo que se deberá monitorear el uso de la DB, porque en máximo 90 días, podría empezar a realizar cobros.

Resumen	
Edición de Cloud SQL	Enterprise
Region	us-central1 (Iowa)
Versión de la base de datos	MySQL 8.0
CPU virtuales	2 CPU virtual(es)
Memoria	8 GB
Caché de datos	Inhabilitada
Almacenamiento	10 GB
Conexiones	IP pública
Copia de seguridad	Automatizada
Disponibilidad	Zona única
Recuperación de un momento determinado	Habilitada

Ya que haya dado clic al botón de CREAR INSTANCIA, aparecerá el dashboard de la base de datos MySQL, pero en este momento no tendrá ninguna base de datos creada, por lo cual deberemos acceder a ella en la parte lateral izquierda donde dice Bases de datos → CREAR BASE DE DATOS → Asignar nombre y codificación (utf8) → CREAR. Para que se pueda completar este proceso, deberemos dejarla acabar de crear la instancia.

The screenshot shows the Google Cloud SQL dashboard for the '1-Platziblog DB' project. On the left sidebar, under 'INSTANCIA PRINCIPAL', the 'Bases de datos' option is selected. In the main content area, a new instance 'platziblog-db' is being created. A message at the top states: 'Se está creando la instancia. Este proceso puede demorar algunos minutos. Mientras tanto, puedes seguir viendo información sobre la instancia.' Below this, there is a graph titled 'Uso de CPU' which displays a timeline from UTC-6 18:00 to 16:00. A note on the graph says: 'No hay datos disponibles para el período seleccionado.' At the bottom of the page, a link reads: 'Ir a Estadísticas de consultas para obtener información más detallada sobre las consultas y el rendimiento'.

The screenshot shows the 'Crear una base de datos' dialog box. It contains fields for 'Nombre de la base de datos' (set to 'platziblog-db'), 'Intercalación' (set to 'utf8'), and 'Grupo de caracteres' (set to 'utf8'). There are also dropdowns for 'Intercaleación' (set to 'Intercaleación predeterminada') and 'Más información' (link to MySQL identifier rules). At the bottom are 'CREAR' and 'CANCELAR' buttons.

Ahora volveremos a la pestaña principal de Descripción general para bajar en la ventana y dar clic en el botón de Conectarse a esta instancia → ABRIR CLOUD SHELL.

The screenshot shows the 'Descripción general' tab of the Google Cloud SQL dashboard. It features a 'Conectar a esta instancia' section with fields for 'Dirección IP pública' (34.171.9.248) and 'Nombre de la conexión' (platziblog-db-484922:us-central1:platziblog-db). Below this is a '¿Necesitas ayuda para conectarte?' section with a link to 'Más información'. The 'Configuración' section displays resource details: 'CPU virtuales' (2), 'Memoria' (8 GB), and 'Almacenamiento de SSD' (10 GB). At the bottom right, the 'Operaciones' section shows a log entry: 'Se reinició platziblog-db' at '17:01:15 GMT-6'.

Con esta terminal nos podremos comunicar con la base de datos recién creada, donde por default se nos proporciona el comando con el cual nos podemos conectar con la base de datos, que es el siguiente:

```
gcloud sql connect platziblog-db --user=root --quiet
```

Pero antes de poder acceder a ella, se debe activar una API que permite hacerlo y se encuentra en el siguiente link:

<https://console.cloud.google.com/apis/api/sqladmin.googleapis.com>

The screenshot shows the Google Cloud Platform interface for the Cloud SQL Admin API. At the top, there's a banner about a free trial credit. Below it, the navigation bar includes 'Google Cloud' and '1-PlatziBlog DB'. The main content area is titled 'Cloud SQL Admin API' and describes it as 'Google Enterprise API'. It has a 'PROBAR ESTA API' button. On the left, a sidebar shows 'INSTANCIA PRINCIPAL' with 'Descripción general' selected, along with other options like 'Estadísticas del sistema', 'Estadísticas de consultas', 'Conexiones', 'Usuarios', and 'Notas de versión'. The main panel shows 'Configuración' with settings for 'CPU virtuales' (2), 'Memoria' (8 GB), and 'Almacenamiento de SSD' (10 GB). Below this is a 'Conectar a esta instancia' section with an IP address input field containing '34.171.9.248'. A 'Terminal' tab is open at the bottom, showing a Cloud Shell session with the command: 'Welcome to Cloud Shell! Type "help" to get started. Your Cloud Platform project in this session is set to platziblog-db-404922. Use "gcloud config set project [PROJECT\_ID]" to change to a different project. diegorko06@cloudshell:~\$ gcloud sql connect platziblog-db --user=root --quiet'.

Cabe mencionar que la conexión solo durará 5 minutos por cuestiones de seguridad, al acabar el proceso de conexión se introducirá la contraseña y ya tendremos acceso a la base de datos.

This screenshot shows the same Google Cloud Platform interface as the previous one, but the Cloud Shell terminal output now shows a successful MySQL connection. The command 'gcloud sql connect platziblog-db --user=root --quiet' was run, followed by the MySQL prompt 'mysql>'. The terminal also displays copyright information for Oracle and a note about Oracle trademarks.

# Lenguaje de Programación SQL

Las siglas SQL de significan Structured Query Language, la función principal de este lenguaje de programación es la de realizar consultas a datos de una forma estandarizada no importando que base de datos se esté utilizando.

Además del lenguaje SQL existen los lenguajes NoSQL, cuyas siglas significan Not Only SQL, esto se utiliza más que nada en bases de datos no relacionales, donde, aunque sí se sigue utilizando en algunas partes el lenguaje SQL, se utilizan algunas variantes de este lenguaje. Algunos ejemplos de las bases de datos no relacionales que utilizan alguna variante de SQL son Cassandra, Big Query, etc.

## Sub-lenguajes de SQL: DDL (Data Definition Language)

Los dos sub-lenguajes más importantes del lenguaje SQL son llamados DDL y DML, primero se explicará el DDL que sirve para estructurar la base de datos:

- **DDL (Data Definition Language):** La función principal de este sub-lenguaje de SQL es la de crear la estructura de una base de datos. Esto se refiere a las entidades, atributos, relaciones, etc. que son descritos en un diagrama ER o en un diagrama físico. Los **3 comandos** con los que cuenta para llevar a cabo la estructuración de datos con el lenguaje DDL son los siguientes:
  - **CREATE:** Comando que ayuda a crear una base de datos, tabla, vista, índice, etc.
  - **ALTER:** Comando que ayuda a alterar o modificar una entidad, tabla, tipo de dato, etc.
  - **DROP:** Comando que ayuda a eliminar elementos de una base de datos. Esta se debe utilizar con mucho cuidado, ya que accidentalmente podríamos borrar nuestra db completa.

**Algunos de los elementos u objetos** que se van a manipular con los **comandos DDL** son los siguientes:

- **DATABASE:** Se refiere a la base de datos, a veces también se le llama **SCHEMA**.
- **TABLE:** Las tablas se refieren a la visualización de los datos después de haberlas modelado a través de un diagrama ER o diagrama físico.
- **VIEW:** Las vistas se refieren a la forma en la que se pueden interpretar y ordenar los datos extraídos de una database, de tal forma que puedan ser entendidos o de utilidad para el usuario, ya que, en las tablas de las bases de datos, muchas veces la información está segmentada y de esta forma la podemos filtrar y organizar.

### CREATE:

#### *Crear una Base de Datos con SQL*

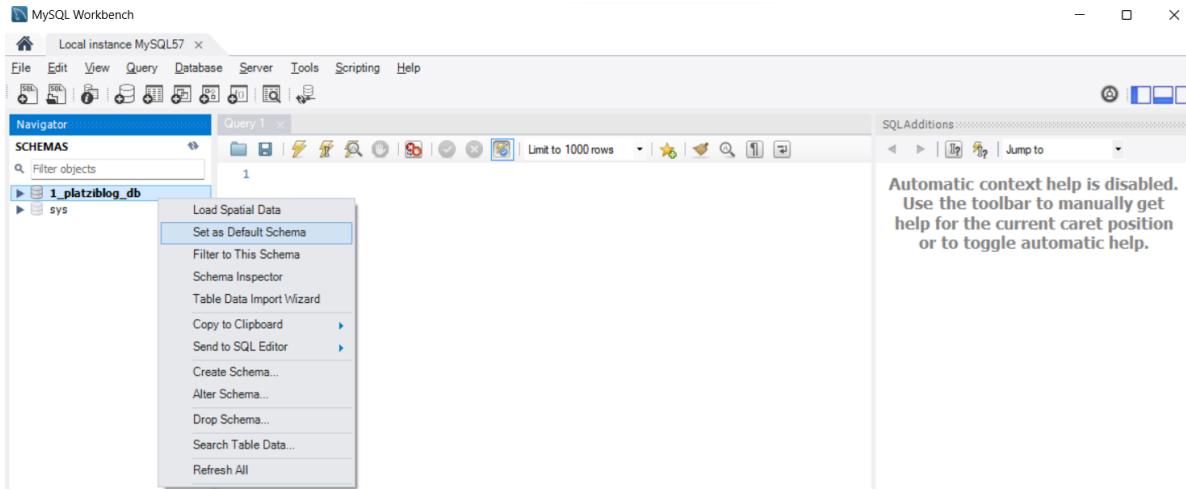
Siempre que se quiera crear una base de datos nueva se deberán ejecutar los siguientes comandos, ya sea en un Manejador de Bases de Datos Relacionales (RDBMS) o en una Base de Datos con Servicios Administrados (Nube o Cloud):

**CREATE DATABASE "Nombre\_Base\_de\_Datos";**

//La siguiente instrucción no siempre se incluye, porque se da por entendida.

```
USE      DATABASE      "Nombre_Base_de_Datos";
```

Para asegurarnos que el código SQL se está aplicando a una base de datos específica en MySQL Workbench, daremos clic derecho sobre ella y seleccionaremos la opción de Set as Default Schema.

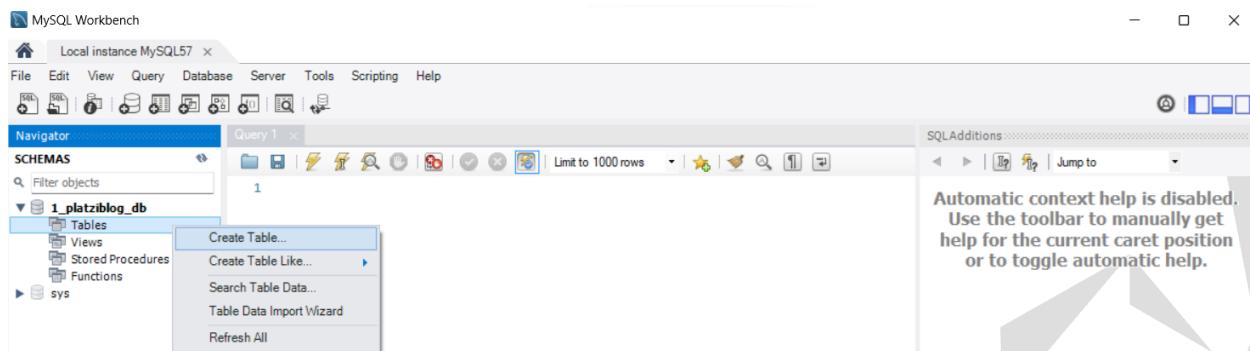


### **Crear una Tabla con SQL**

Cuando se quiera crear una nueva tabla en una base de datos se deberá ejecutar el siguiente comando, ya sea en un Manejador de Bases de Datos Relacionales (RDBMS) o en una Base de Datos con Servicios Administrados (Nube o Cloud), aunque cabe la pena mencionar que es importante previamente tener ya listo el diagrama ER y físico de la base de datos, para seguir dicha estructura:

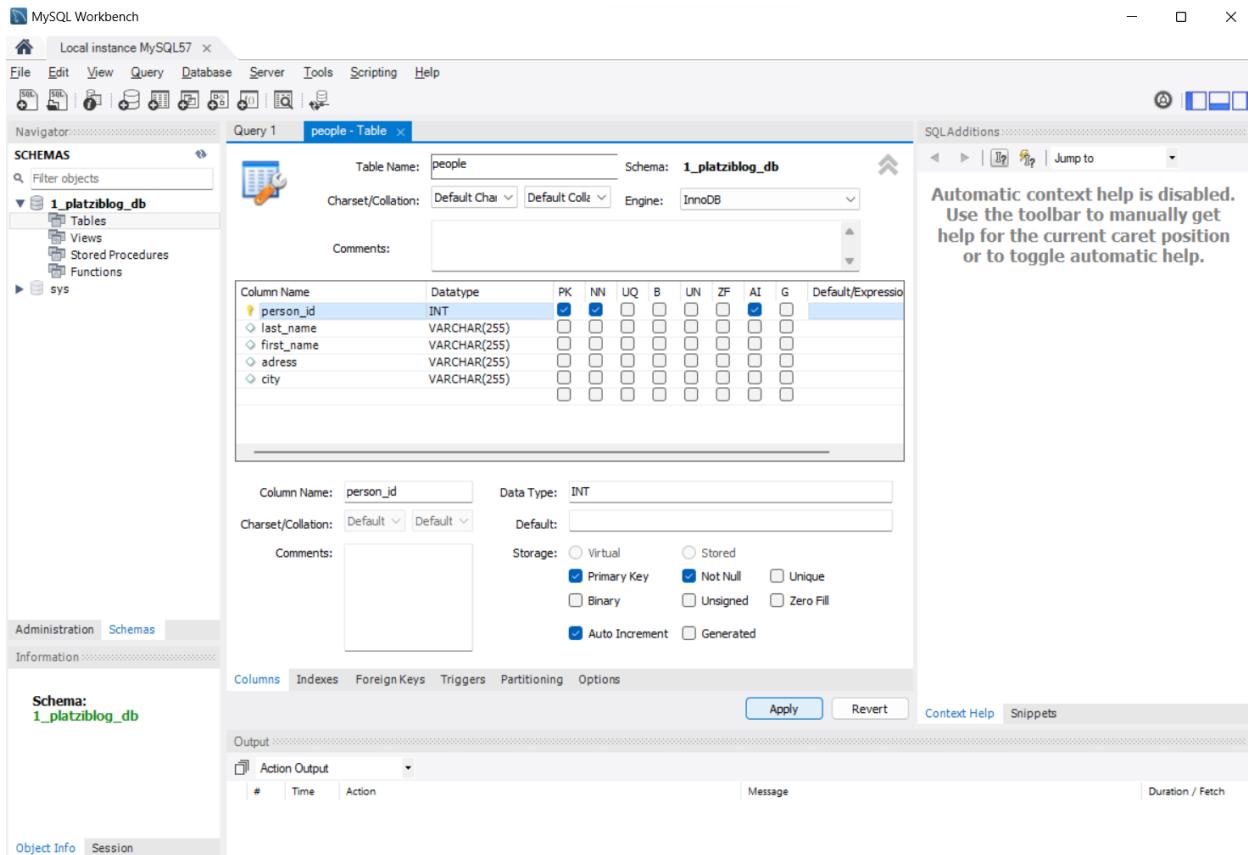
```
CREATE TABLE  "Nombre_Base_de_Datos". "Nombre_Entidad_o_Tabla"(  
    "Nombre_Atributo"      Tipo_de_Dato_y_Constraints ,  
    "Nombre_Atributo"      Tipo_de_Dato_y_Constraints  
);
```

Para crear una nueva tabla en una base de datos específica con código SQL en MySQL Workbench, desplegaremos la pestaña de la base de datos, daremos clic derecho sobre donde dice Tables y seleccionaremos la opción de Create Table...

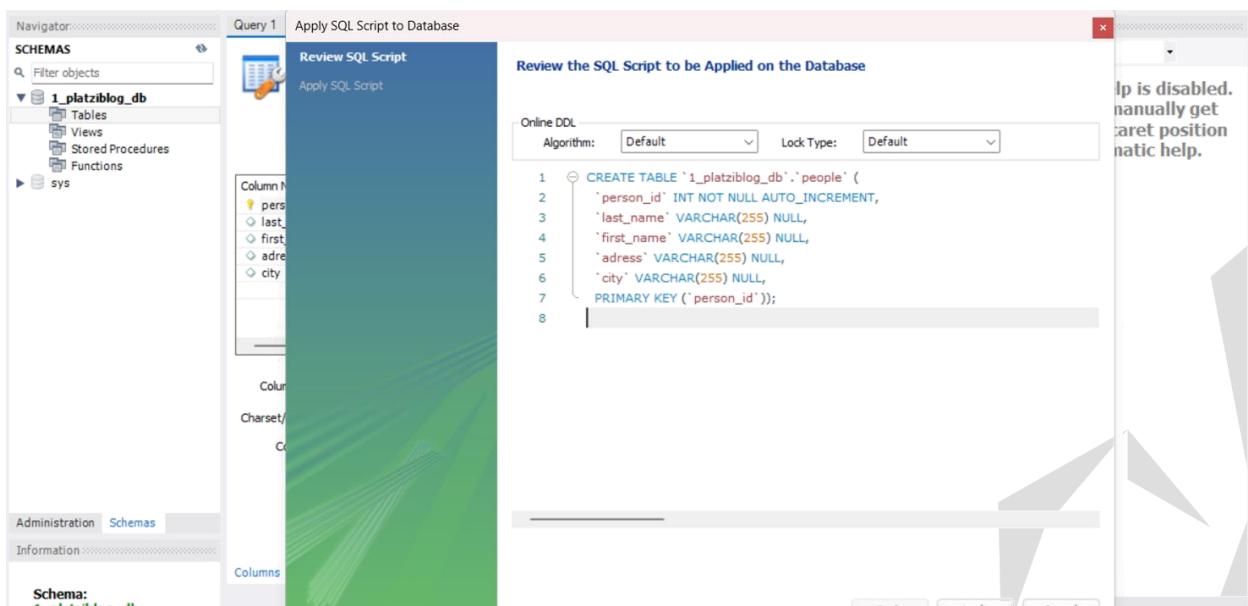


La ventaja de utilizar un cliente gráfico es que se puede visualizar el resultado del código.

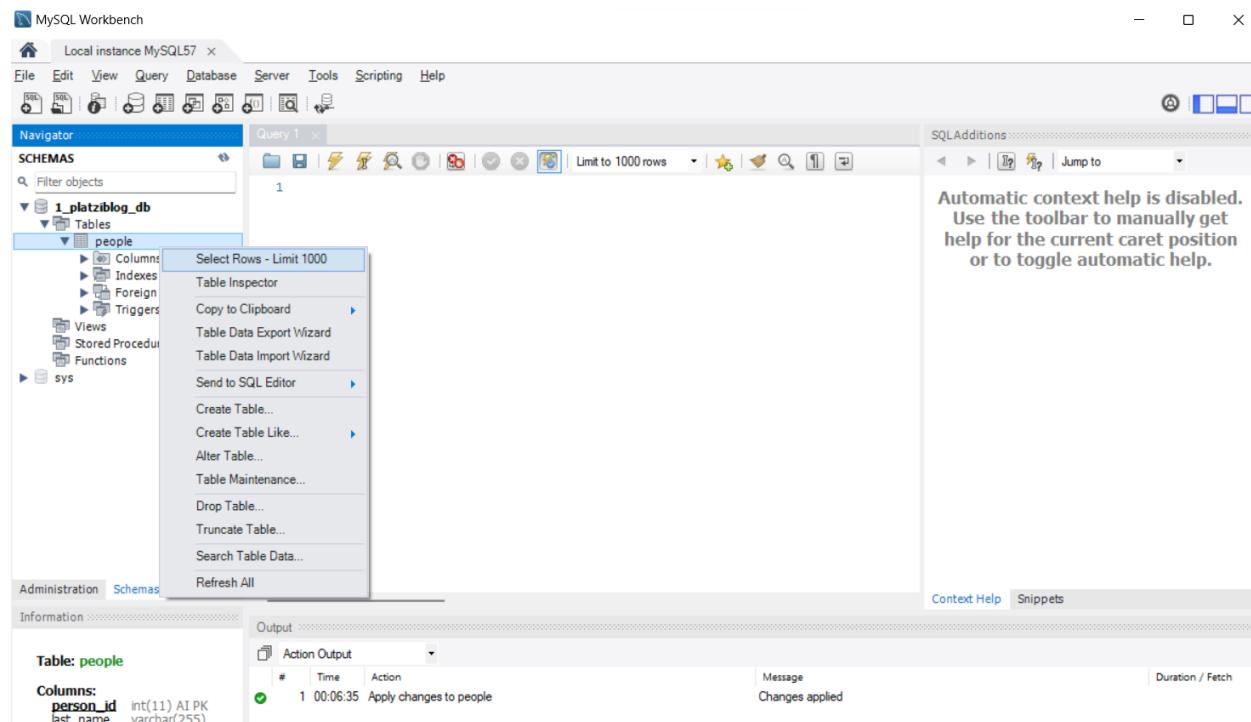
Ya que se hayan llenado todos los atributos de la tabla (entidad) que queremos crear, incluyendo sus **constraints (restricciones)**, daremos clic en el botón de Apply.



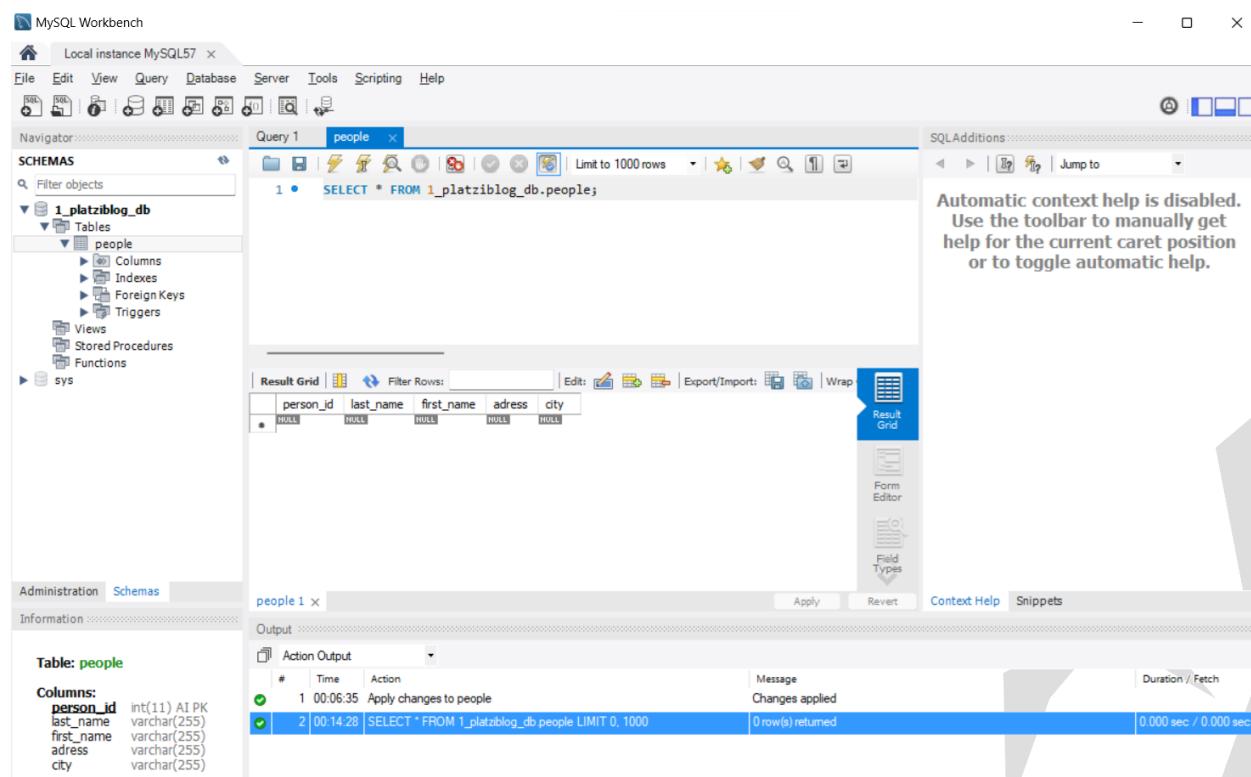
En este punto es donde veremos el código SQL que se ejecutará en la base de datos justo antes de que lo haga, y podremos observar que cumple la sintaxis descrita anteriormente. Ya que estemos satisfechos con el código mostrado, daremos de nuevo clic en el botón de Apply → Finish.



Ahora ya podremos visualizar los datos almacenados en esta **entidad** al ingresar dentro del desplegable llamado Tables, dar clic derecho sobre el nombre de la **tabla** creada y seleccionar la opción de Select Rows - Limit 1000.



Y veremos la **tabla** con los **atributos** que le agregamos en un formato de tabla.



### **Crear una Vista con SQL**

Cuando se quiera crear una nueva vista en una base de datos, la cual representa una extracción de datos que se encuentren filtrados y organizados, se deberá ejecutar el siguiente comando, ya sea en un Manejador de Bases de Datos Relacionales (RDBMS) o en una Base de Datos con Servicios Administrados (Nube o Cloud):

```
USE      DATABASE      "Nombre_Base_de_Datos";  
CREATE OR REPLACE  VIEW    "v_Nombre_Vista" AS  
//Filtro o tabla de la que se quieran obtener los datos de la vista.
```

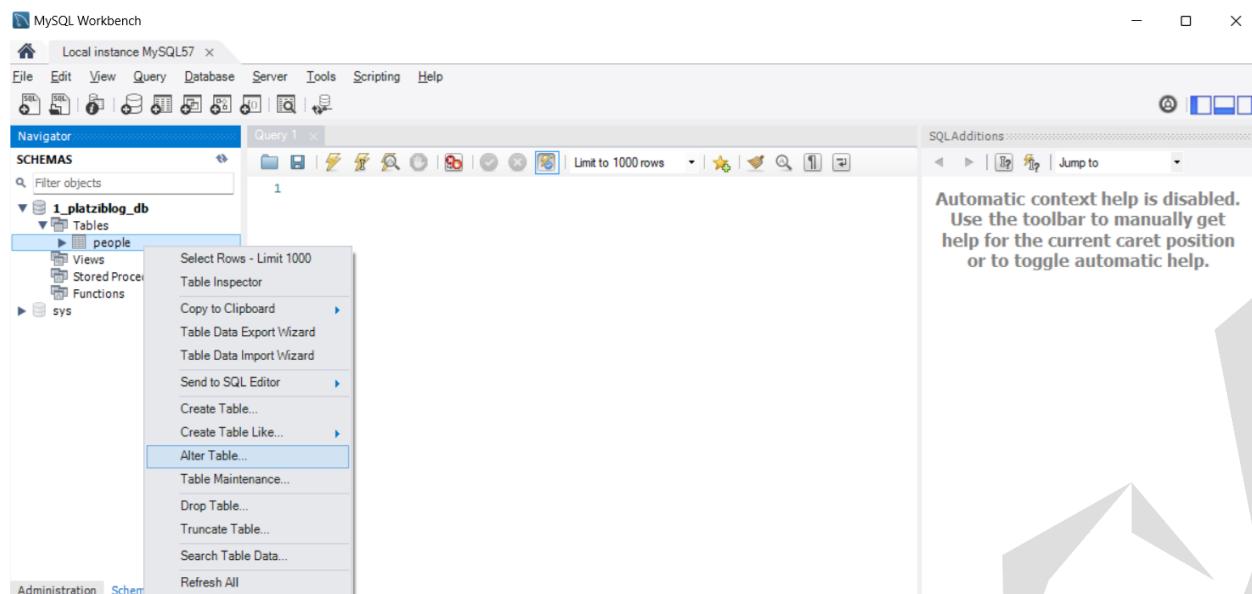
ALTER:

### **Alterar una Tabla con SQL**

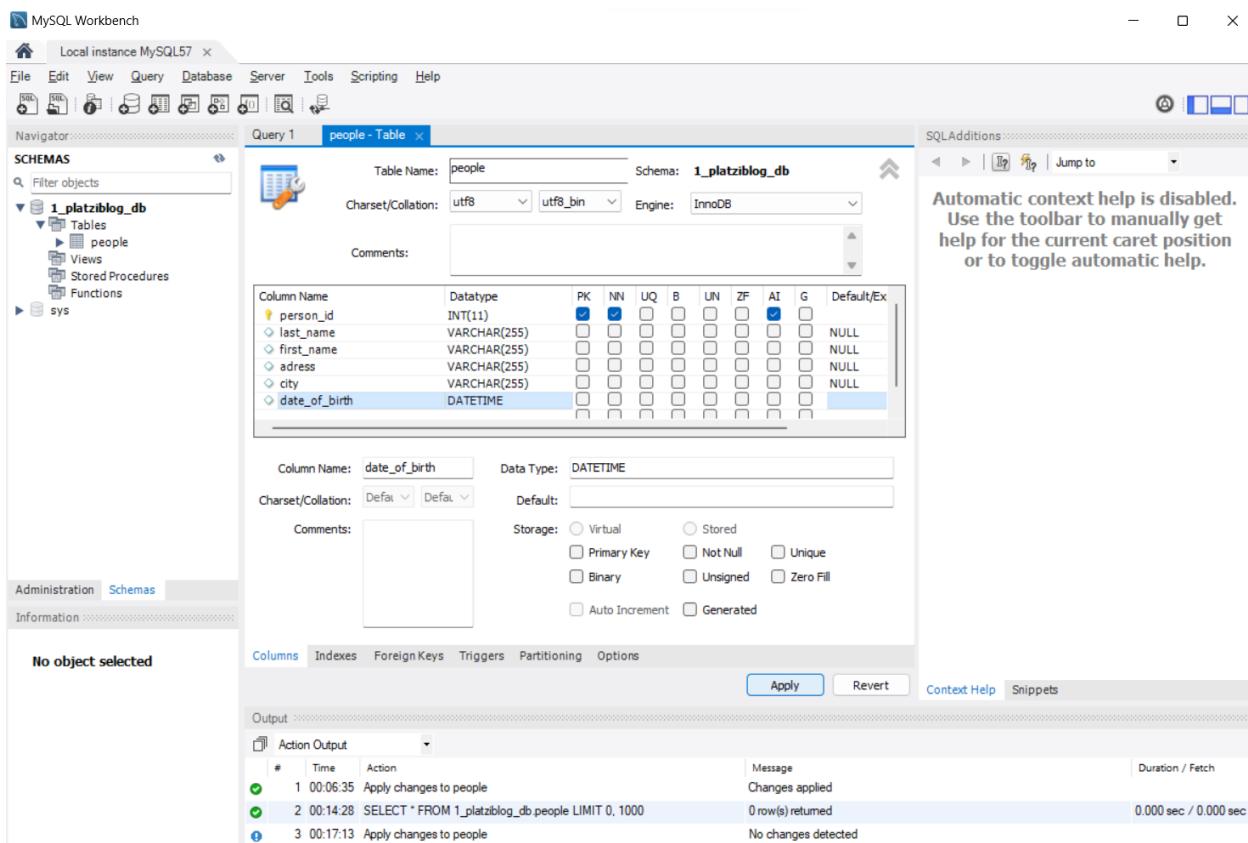
Cuando se quiera modificar una tabla en una base de datos se deberá ejecutar el siguiente comando, ya sea en un Manejador de Bases de Datos Relacionales (RDBMS) o en una Base de Datos con Servicios Administrados (Nube o Cloud):

```
ALTER  TABLE      "Nombre_Base_de_Datos". "Nombre_Entidad_o_Tabla";  
ADD    COLUMN     "Nombre_Atributo"      Tipo_de_Dato_y_Constraints;  
CHANGE COLUMN    " Columna_a_Modificar"  Tipo_de_Dato_y_Constraints;  
DROP    COLUMN     " Columna_a_Borrar";
```

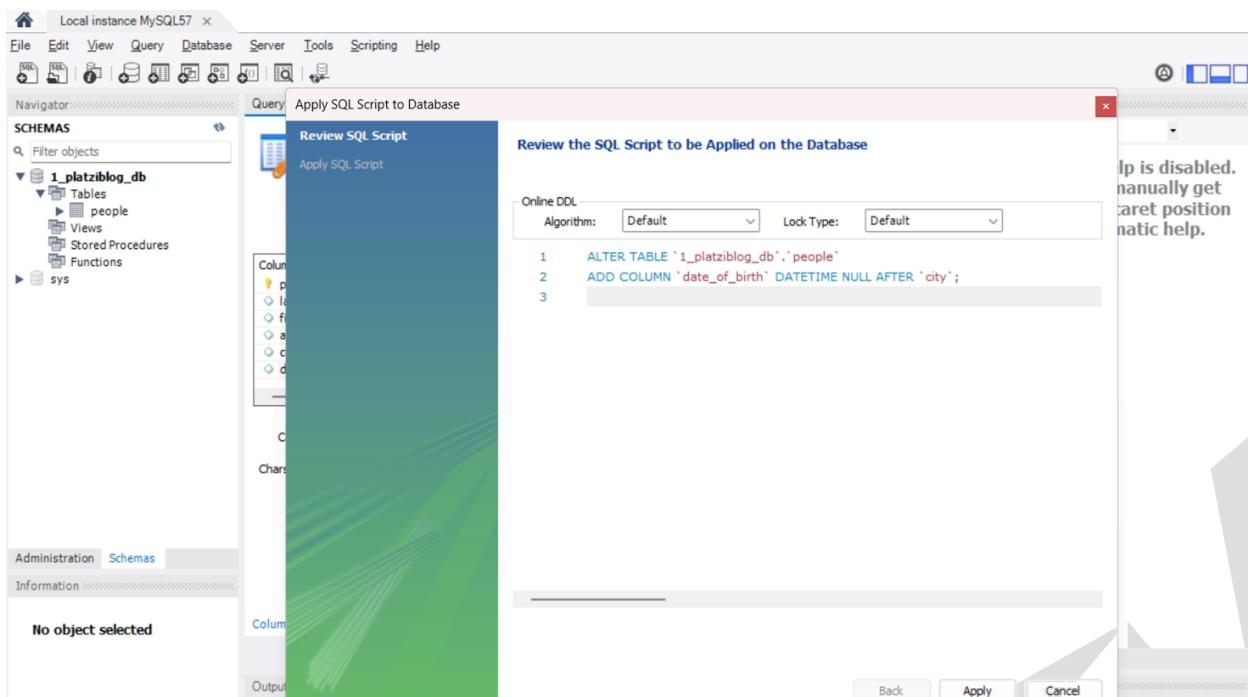
Para alterar una **entidad o tabla** en una base de datos específica con código SQL en MySQL Workbench, desplegaremos la pestaña de la base de datos → luego la de Tables → Daremos clic derecho sobre la entidad que se quiere modificar → Seleccionaremos la opción de Alter Table...



Todas las acciones de agregar, editar y borrar se ejecutarán de forma manual dentro del RDBMS.



Y al dar clic en el botón de Apply podremos ver su equivalente en formato de código SQL.



Recordemos que para ver el resultado en formato de tabla se debe dar clic derecho sobre el nombre de la **tabla** y seleccionar la opción de Select Rows - Limit 1000.

The screenshot shows the MySQL Workbench interface. In the Navigator pane, the '1\_platziblog\_db' schema is selected, and the 'Tables' section shows the 'people' table. The Query 1 window contains the SQL command: `SELECT * FROM 1_platziblog_db.people;`. The Result Grid shows the following columns: person\_id, last\_name, first\_name, adress, city, date\_of\_birth. All rows are currently null. On the right, a SQLAdditions panel displays a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

**DROP:**

### Borrar una Tabla, Columna o Base de Datos con SQL

Cuando se quiera modificar una tabla en una base de datos se deberá ejecutar el siguiente comando, ya sea en un Manejador de Bases de Datos Relacionales (RDBMS) o en una Base de Datos con Servicios Administrados (Nube o Cloud):

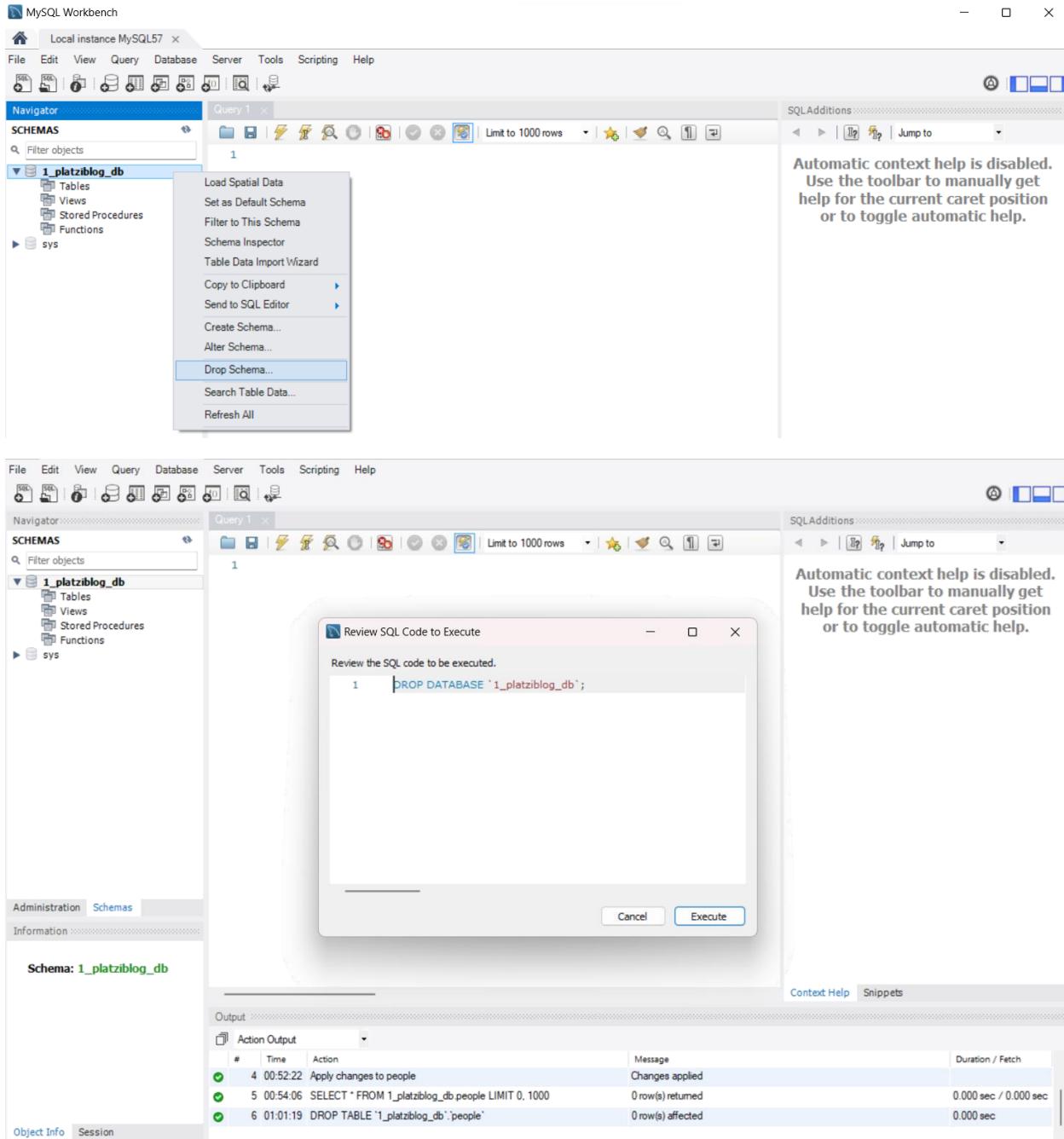
```

DROP DATABASE "Nombre_Base_de_Datos";
DROP TABLE      "Nombre_Base_de_Datos". "Nombre_Entidad_o_Tabla";
DROP COLUMN    "Columna_a_Borrar";

```

La misma secuencia que se aplica para alterar una tabla se aplica para borrar columnas, tablas o db.

The screenshot shows the MySQL Workbench interface. The 'people' table is selected in the Navigator pane. A context menu is open over the table name, with the 'Drop Table...' option highlighted. The Query 1 window shows the number 1. On the right, a SQLAdditions panel displays a message: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."



## Sub-lenguajes de SQL: DML (Data Manipulation Language)

Los dos sub-lenguajes más importantes del lenguaje SQL son llamados DDL y DML, ya se abordó el lenguaje DDL que sirve para estructurar la base de datos y ahora se explicará el DML que sirve para manipular los datos:

- **DML (Data Manipulation Language):** La función principal de este sub-lenguaje de SQL es la de manipular los datos de una base de datos, ya sea para crear vistas o para meter, actualizar,

borrar o extraer datos. Los **4 comandos** con los que cuenta para llevar a cabo la manipulación de datos con el lenguaje DML son los siguientes:

- **INSERT**: Comando que ayuda a crear una base de datos, tabla, vista, índice, etc.
- **UPDATE**: Comando que ayuda a alterar o modificar una entidad, tabla, tipo de dato, etc.
- **DELETE**: Comando que ayuda a eliminar elementos de una base de datos. Esta se debe utilizar con mucho cuidado, ya que accidentalmente podríamos borrar nuestra db completa.
- **SELECT**: Comando que ayuda a eliminar elementos de una base de datos. Esta se debe utilizar con mucho cuidado, ya que accidentalmente podríamos borrar nuestra db completa.

Algunos de los **elementos u objetos** que se van a manipular con los **comandos DDL** son los siguientes:

- **DATABASE**: Se refiere a la base de datos, a veces también se le llama **SCHEMA**.
- **TABLE**: Las tablas se refieren a la visualización de los datos después de haberlas modelado a través de un diagrama ER o diagrama físico.
- **VIEW**: Las vistas se refieren a la forma en la que se pueden interpretar y ordenar los datos extraídos de una database, de tal forma que puedan ser entendidos o de utilidad para el usuario, ya que, en las tablas de las bases de datos, muchas veces la información está segmentada y de esta forma la podemos filtrar y organizar.

## Referencias

Platzi, Israel Vázquez, "Curso de Fundamentos de Bases de Datos", 2018 [Online], Available: <https://platzi.com/new-home/clases/1566-bd/19781-bienvenida-conceptos-basicos-y-contexto-historico/>

