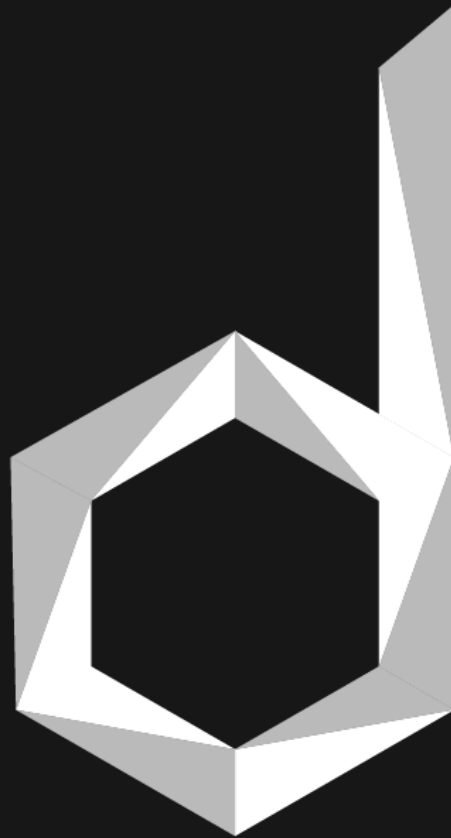


INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

PROGRAMACIÓN: DESARROLLO BACKEND

SQL

Bases de Datos y SQL

Contenido

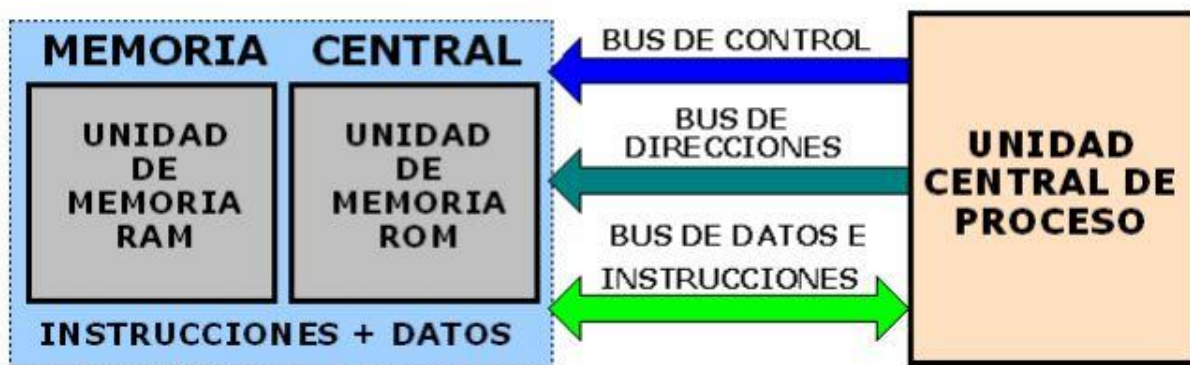
Introducción a las Bases de Datos	2
Tipos de Bases de Datos.....	2
Representación de las Bases de Datos: Nomenclatura de Chen	3
Diagrama ER (Entidad-Relación)	3
Diagrama Físico	7
Normalización: Tabla de Datos a RDB	9
Ejemplo de Base de Datos: Blog Post	12
Instalación del Manejador de Base de Datos	15
Referencias.....	18



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.

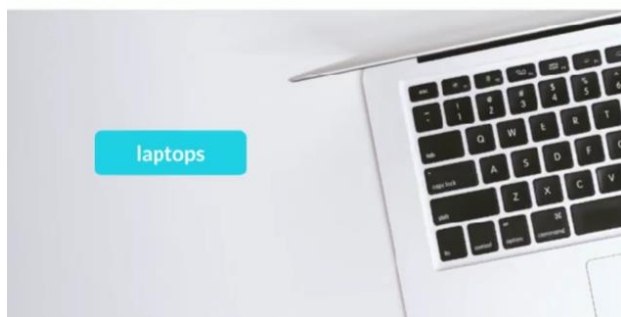
Representación de las Bases de Datos: Nomenclatura de Chen

Diagrama ER (Entidad-Relación)

- **Entidad:** Una entidad es algo muy similar a un objeto, el cual se puede asociar con ciertos **atributos (características)** y **métodos (funciones)**, 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



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:** Sí 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

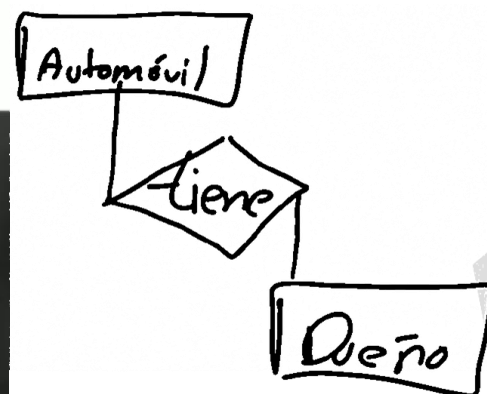
Libros			Ejemplares		
id	título	...	libro_id	localización	edición
LKJ789JKAS	Viaje al cent...	...	LKJ789JKAS	pasillo 1	1
KCO3100KJH	El señor de	KCO3100KJH	pasillo 1	1
NSDJOIH128	De la tierra...	...	NSDJOIH128	pasillo 1	3
09KSIHBD71	Amor en tie...	...	09KSIHBD71	pasillo 1	1

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 ligán las diferentes entidades entre sí, para ello dentro de las relaciones se utilizan verbos que conecten una relación con la otra.

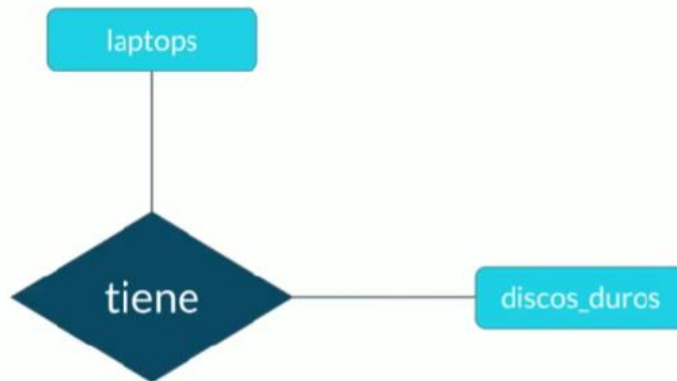
Relaciones



- **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.

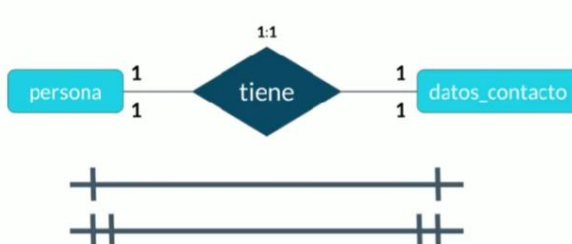
- Cuando se utiliza la cardinalidad se utiliza todo el concepto del verbo de conexión que utiliza la relación para indicar cual es el número de entidades con las que cuenta una entidad, de esta manera es como se puede separar los atributos multivaluados.

Relaciones

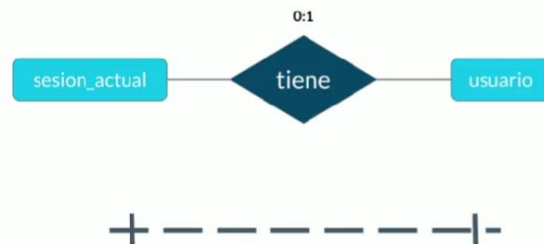


- De igual manera se obtiene la cardinalidad de ambos lados, ya que esto puede variar cuando se ve desde perspectivas distintas. Además, se maneja cierta nomenclatura en el diagrama para denotarlo.

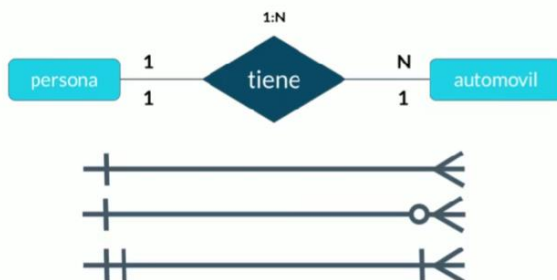
Cardinalidad: 1 a 1



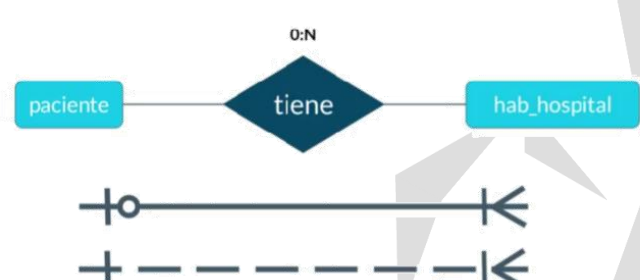
Cardinalidad: 0 a 1



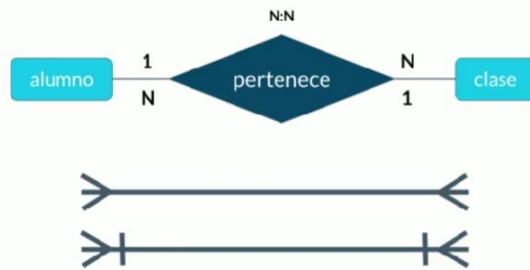
Cardinalidad: 1 a N



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:

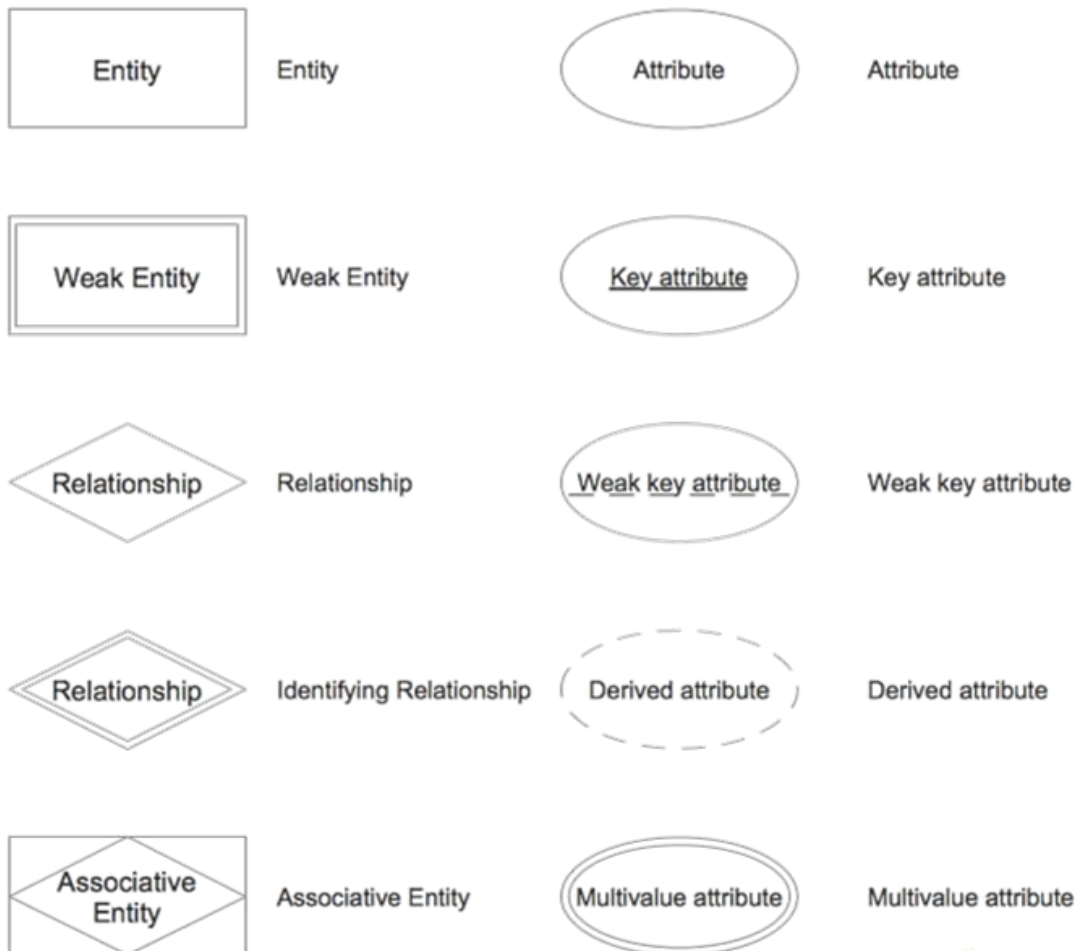


Diagrama Físico

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.
 - **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 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

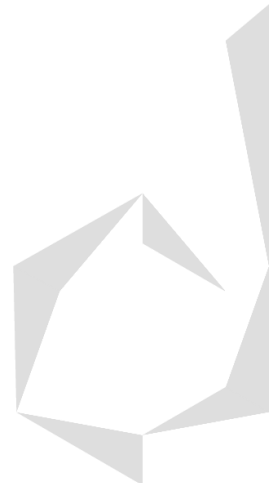
alumnos				
alumno_id	alumno	nivel_curso	nombre_curso	materia
1	Juanito	Maestría	Data engineering	MySQL
1	Juanito	Maestría	Data engineering	Python
2	Pepito	Licenciatura	Programación	MySQL
2	Pepito	Licenciatura	Programación	Python

- **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			cursos		
alumno_id	alumno	curso_id	curso_id	nivel_curso	nombre_curso
1	Juanito	1	1	Maestría	Data engineering
2	Pepito	2	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 de Base de Datos: Blog Post

Para crear una base de datos, primero que nada, debemos pensar en las **entidades** que se utilizaran 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:

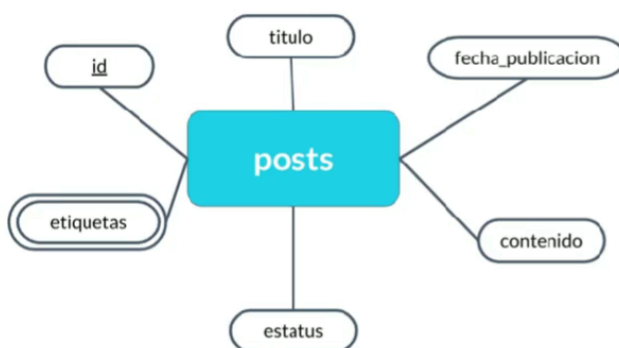
- **Entidades:**
 - Posts (Publicaciones).
 - Usuarios.
 - Comentarios.
 - Categorías.
 - Etiquetas.

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**).

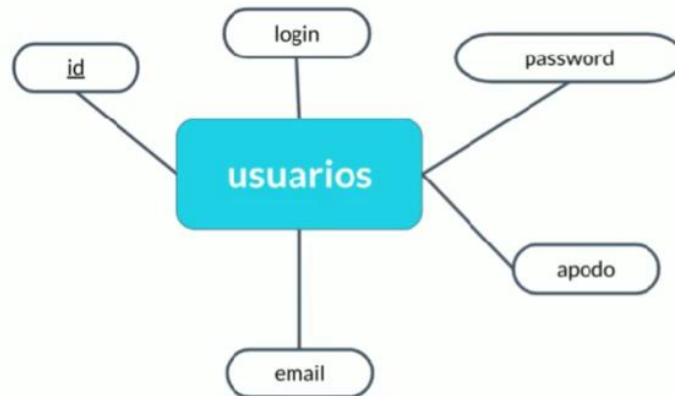
Entidades Platziblog



○ **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

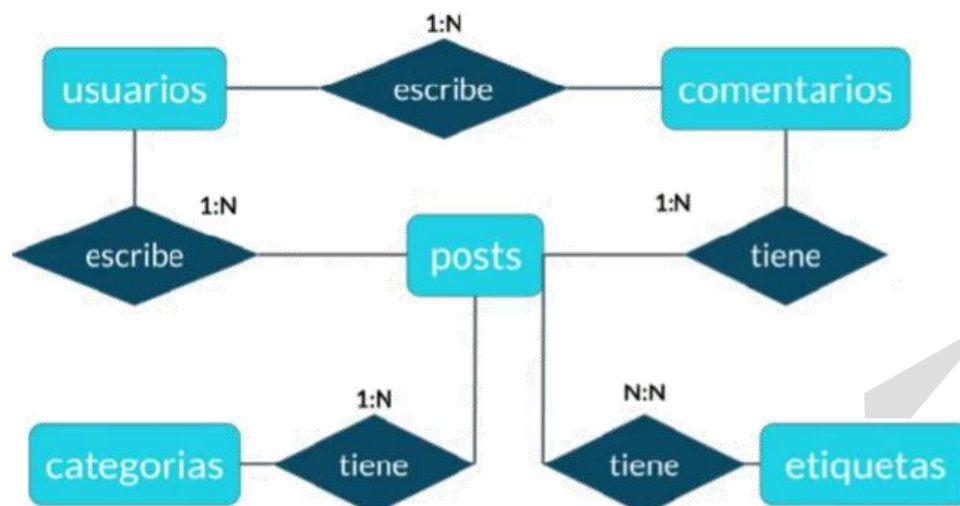


• **Diagrama ER (Entidad-Relación):**

○ **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:**

- **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.

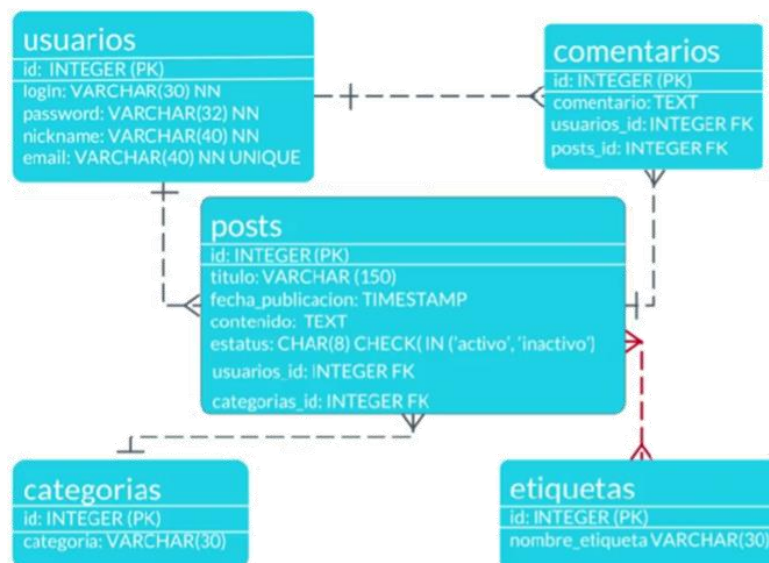
- **NN: Not Null.**

- **UNIQUE.**

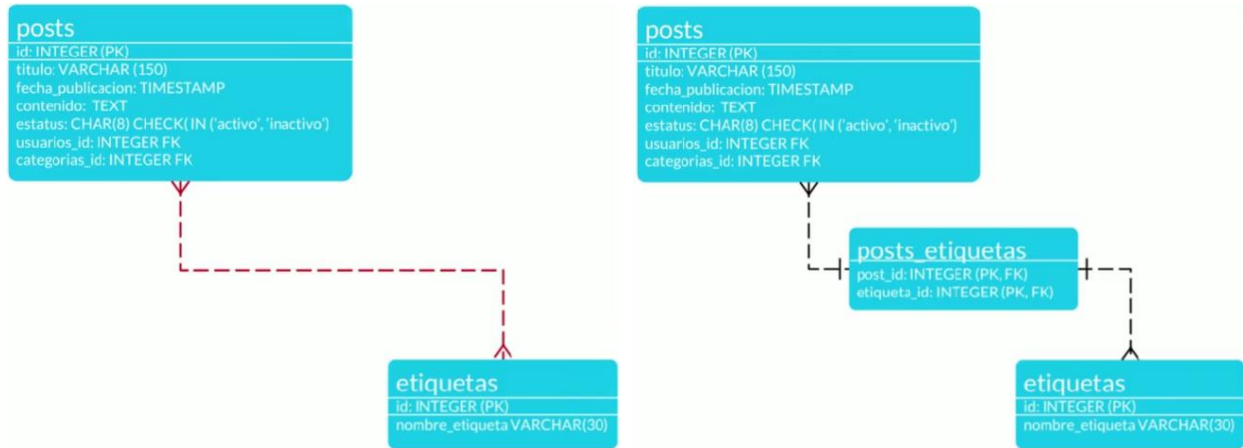
- **CHECK.**

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.



Instalación del Manejador de Base de Datos

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>

MySQL Community Downloads

MySQL Installer

General Availability (GA) Releases Archives

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:

Select Operating System:

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: e89af3ba9bb4716ff5e647b0f42edab2 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

Login »

using my Oracle Web account

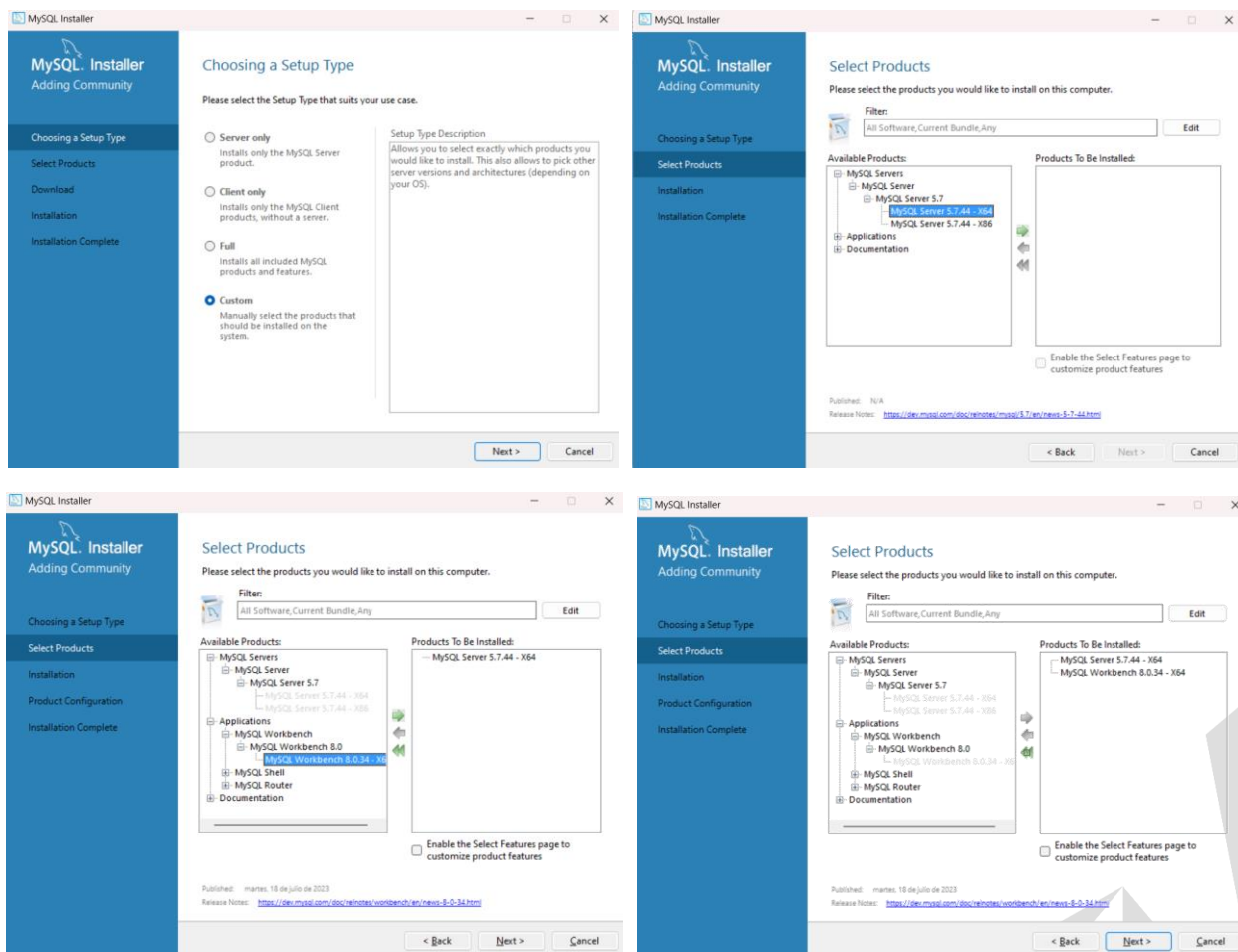
Sign Up »

for an Oracle Web account

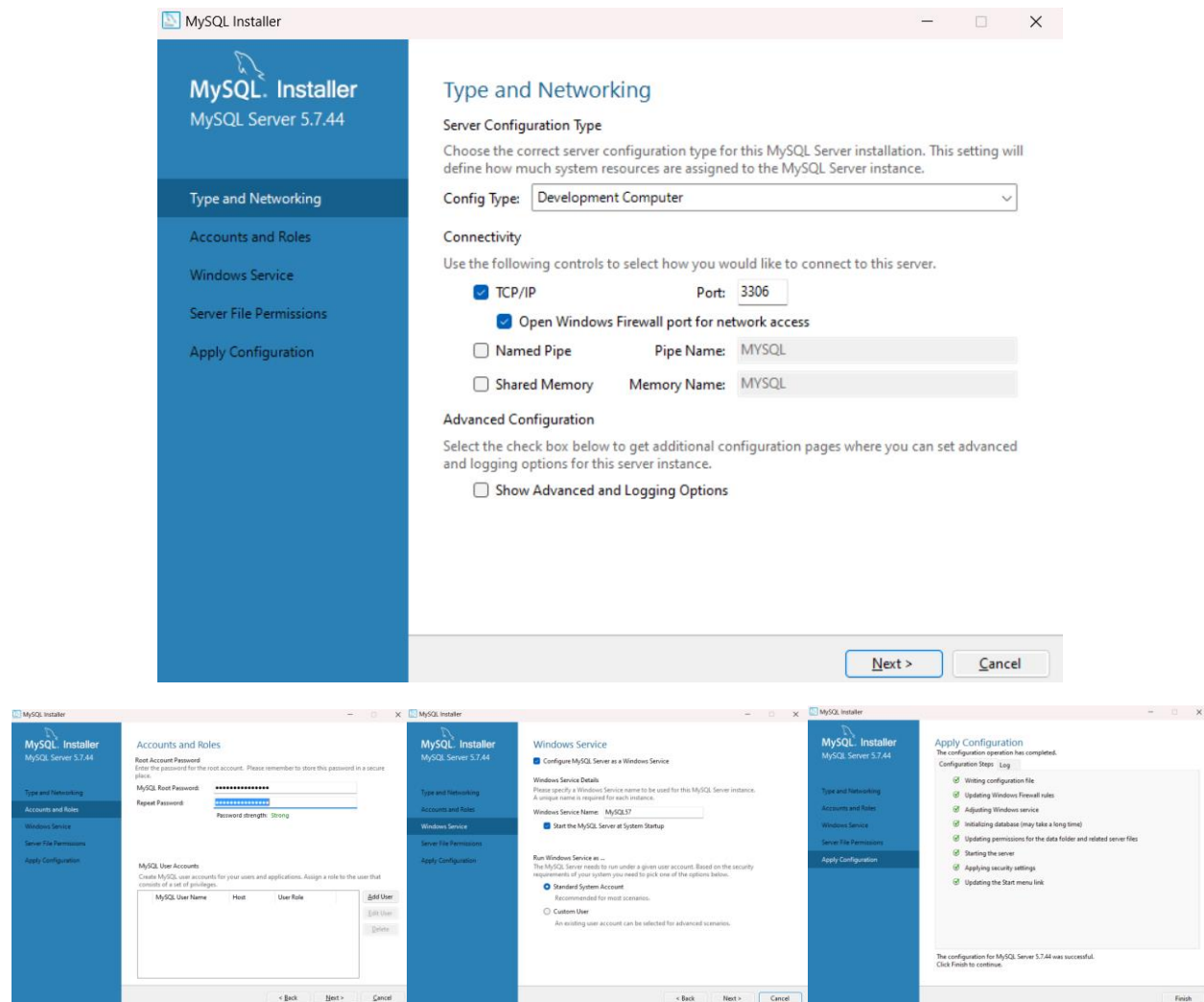
MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

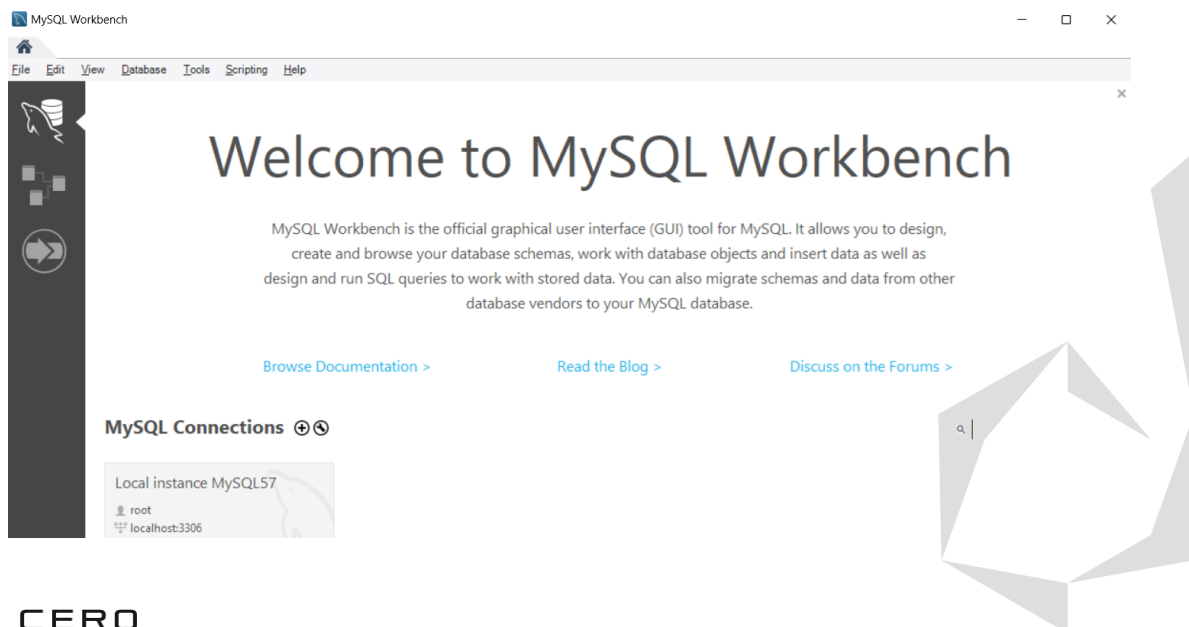
Luego daremos clic en la instalación Custom → MySQL Servers → MySQL Server 5.7 → MySQL Server 5.7 - X64 → → → Applications → MySQL Workbench → MySQL Workbench → Next → Execute.



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 estará 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.



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-/](https://platzi.com/new-home/clases/1566-bd/19781-bienvenida-conceptos-basicos-y-contexto-historico/)

