



Nuestro compromiso es con el *futuro*.

# Bases de Datos

# Clase 2

# ¿Qué veremos hoy?

Hoy vamos a continuar el módulo de **Bases de Datos**, donde veremos mas sobre **Tablas**, los distintos **tipos de datos** de los que podemos disponer y cómo se pueden establecer las distintas **relaciones** entre estos objetos.

Seguiremos usando **Workbench**, viendo como editar, borrar o crear tablas; como así tambien, cómo visualizar scripts con esta herramienta.

# Base de datos. Tablas

Comencemos tratando de entender más en detalle lo que es una **tabla**:

Una tabla de base de datos es la estructura principal y más general que existe para almacenar información; una base de datos contendrá una o más tablas.

La forma de identificar cada tabla, será mediante un nombre único.

En concreto una tabla se divide en :

1. Las filas, que representan los registros de una base de datos
2. Cada fila tendrá su información dividida en columnas, con un nombre único dentro de una tabla.

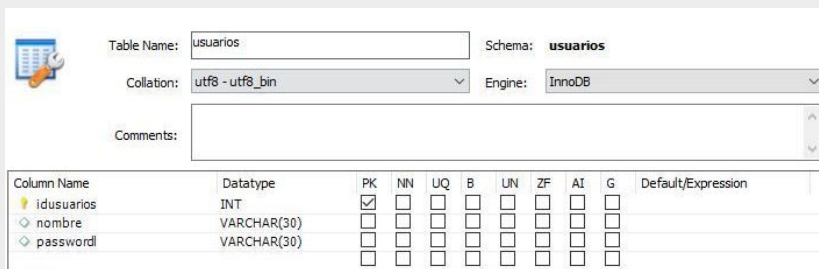


Table Name: usuarios Schema: usuarios

Collation: utf8 - utf8\_bin Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idusuarios	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombre	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

A continuacion, veremos algunos tipos de datos que podemos utilizar a la hora de crear nuestras **tablas**

# Tablas. Tipos de datos

## Datos **numéricos**

TINYINT  
SMALLINT  
MEDIUMINT  
INT o INTEGER  
BIGINT

## Datos para guardar cadenas de caracteres (**alfanuméricos**)

CHAR  
VARCHAR  
BINARY  
VARBINARY  
TINYBLOB  
TINYTEXT  
BLOB  
TEXT  
MEDIUMBLOB  
MEDIUMTEXT  
LONGBLOB  
LONGTEXT  
ENUM  
SET

## Datos para almacenar **fechas y horas**

DATE  
DATETIME  
TIME  
TIMESTAMP  
YEAR

## Atributos de los campos

NULL  
DEFAULT  
BINARY  
INDEX  
PRIMARY KEY  
AUTO\_INCREMENT  
UNIQUE  
FULLTEXT

# Tablas. Tipos de datos

Anteriormente, solo se mencionó de forma general, a los tipos de datos que podemos obtener. Veamos una descripción de al menos los más utilizados, exponiendo en este caso, los relacionados al tipo entero y los relacionados a cadenas de caracteres:

Nombre del Tipo [Nombre SQL]	Descripción
Nota [LONGVARCHAR]	Campo para texto muy grande (2GB)
Texto (fijo) [CHAR]	Texto de tamaño fijo
Texto [VARCHAR]	Texto de tamaño variable
Texto [VARCHAR_IGNORECASE]	Texto de tamaño variable que no diferencia entre mayúsculas y minúsculas

Nombre del Tipo [Nombre SQL]	Descripción
TinyInteger [TINYINT]	Entero de 3 cifras. (Soporta valores entre -128 y 127)
Small Integer [SMALLINT]	Enteros de 5 cifras. (soporta valores entre -32768 y 32767)
Integer [INTEGER]	Entero de 10 cifras. (soporta valores entre -2147483648 y 21473647)
Bigint [BIGINT]	Entero de 19 cifras

# Tablas. Tipos de datos

En lo que respecta a lo relacionado a fechas, tenemos:

Nombre del Tipo [Nombre SQL]	Descripción
Fecha [DATE]	Almacena valores del tipo día, mes y año
Hora [TIME]	Almacena valores del tipo hora, minuto y segundo
Fecha/Hora [TIMESTAMP]	Almacena valores del tipo día, mes, año, hora, minuto y segundo

Se ve claramente la distinción entre los distintos tipos de datos.

Una vez que se comprende los distintos tipos de datos que podemos usar, podemos pasar a ver cuales de ellos **identifican de forma unica una tupla**. Empecemos a ver el concepto de **clave primaria** y **clave foránea** de una tabla.



# Tablas. Claves Primarias y foráneas

## Claves primarias

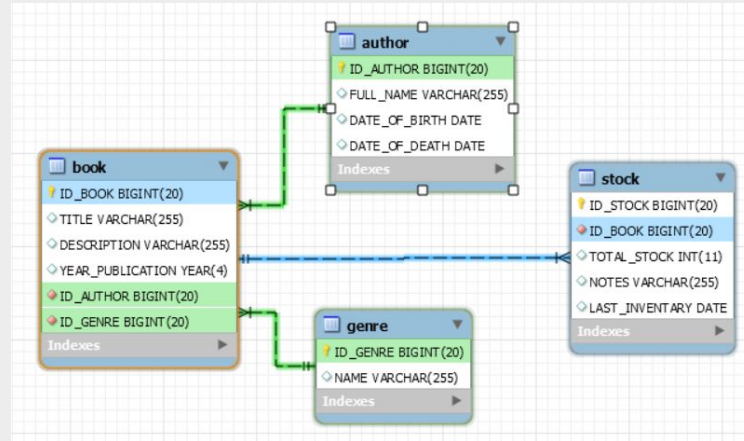
**Una clave primaria** es una columna o un conjunto de columnas en una tabla cuyos valores identifican de forma exclusiva una fila de la tabla. Una base de datos relacional está diseñada para imponer la exclusividad de las claves primarias permitiendo que haya sólo una fila con un valor de clave primaria específico en una tabla.

## Claves foráneas

**Una clave foránea** es una columna o un conjunto de columnas en una tabla cuyos valores corresponden a los valores de la clave primaria de otra tabla. Para poder añadir una fila con un valor de clave foránea específico, debe existir una fila en la tabla relacionada con el mismo valor de clave primaria.

# Tablas. Claves Primarias y foráneas

Veamos en base a los expuesto anteriormente, como se vería gráficamente en nuestra herramienta **Workbench**:



# Tablas. Claves Primarias y foráneas

De lo anterior, podemos ver que :

1. El nombre de la tabla será AUTHOR y dispondrá de los campos:
  - i. ID\_AUTHOR: BIGINT(20), Primary Key **(PK)**.
  - ii. FULL\_NAME: VARCHAR(255).
  - iii. DATE\_OF\_BIRTH: DATE.
  - iv. DATE\_OF\_DEATH: DATE.

Veremos que comunmente, se denota a la clave primaria como **PK**, siendo en este caso, ID\_AUTHOR

# Tablas. Claves Primarias y foráneas

1. Una tabla de los géneros que pueden tener los libros:
  - a. El nombre de la tabla será GENRE y dispondrá de los campos:
    - i. ID\_GENRE: BIGINT(20), Primary Key **(PK)**.
    - ii. NAME: VARCHAR(255).

Aquí nuevamente podemos observar que la **clave primaria** está dada por el campo ID\_GENRE.

# Tablas. Claves Primarias y foráneas

1. Una tabla de los libros:
  - a. El nombre de la tabla será BOOK y dispondrá de los campos:
    - i. ID\_BOOK: BIGINT(20), Primary Key (PK).
    - ii. TITLE: VARCHAR(255).
    - iii. DESCRIPTION: VARCHAR(255).
    - iv. YEAR\_PUBLICATION: YEAR.
    - v. ID\_AUTHOR: Foreign Key (FK) que se relacionará con el campo ID\_AUTHOR (PK) de la tabla AUTHOR.
    - vi. ID\_GENRE: Foreign Key (FK) que se relacionará con el campo ID\_GENRE (PK) de la tabla GENRE.

# Tablas. Claves Primarias y foráneas

Aquí comenzamos a ver algo diferente. Comenzamos viendo en las tablas GENRE y AUTHOR el termino **PK o clave principal**. Pero en la tabla BOOK vemos una relación entre la misma y las tablas mencionadas antes. Esto se logra a través de una clave **llamada FK o clave foránea**. Sigamos ahora analizando la tabla faltante, cuyo nombre es STOCK

# Tablas. Claves Primarias y foráneas

- a. La nombre de la tabla será STOCK y dispondrá de los campos:
  - i. ID\_STOCK: BIGINT(20), Primary Key **(PK)**.
  - ii. ID\_BOOK: Foreign Key **(FK)** que se relacionará con el campo ID\_BOOK **(PK)** de la tabla BOOK.
  - iii. TOTAL\_STOCK: INT(11).
  - iv. NOTES: VARCHAR(255).
  - v. LAST\_INVENTORY: DATE.

# Tablas. Relaciones

Retomando el ejemplo presentado antes, teníamos las tablas:

- 1) GENRE (Genero)
- 2) BOOK (Libro)
- 3) STOCK (Referencia sobre cantidad de libros)
- 4) AUTHOR (Autor)

Nos enfoquemos en la tabla BOOK (libro). Esta tenía como **clave FK (clave foranea)** hacia la tabla AUTHOR por ejemplo. Tiene sentido decir que un autor puede escribir o tener varios libros?; si tomamos el caso de la tabla STOCK, podríamos pensar que la misma podría tener varios libros cierto ? Veamos en detalle estos puntos



# Tablas. Relaciones

## **Relación uno a uno:**

Relaciona un único registro de la tabla principal con uno solo de la tabla relacionada. Este tipo de relación produce el mismo resultado que si se unieran los campos de ambas tablas en una sola tabla.

## **Relación uno a varios**

Es el tipo de relación más frecuente. Un único registro de la tabla principal se puede relacionar con varios de la tabla relacionada.

# Tablas. Relaciones

## **Relación varios a varios**

Un registro de la tabla principal se relaciona con varios de la tabla relacionada y, además, un registro de la tabla relacionada se relaciona con varios de la tabla principal.

Este tipo de relaciones se puede transformar en dos relaciones de tipo uno a varios, creando una tabla intermedia de unión.

# Tablas. Relaciones

## Relación varios a varios

Un registro de la tabla principal se relaciona con varios de la tabla relacionada y, además, un registro de la tabla relacionada se relaciona con varios de la tabla principal.

Este tipo de relaciones se puede transformar en dos relaciones de tipo uno a varios, creando una tabla intermedia de unión.

Visto un poco más en detalle las relaciones entre tablas, veremos cómo editar/crear tablas con **Mysql Workbench.**

# Workbench

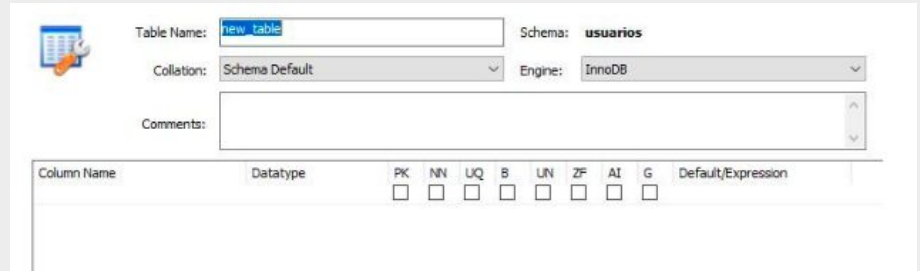


Le damos un nombre al esquema.

Ahora en esquemas veremos nuestro esquema.



**Primer paso**



**Segundo paso**

# Workbench

Table Name:  Schema: **usuarios**

Collation:  Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idusuarios	INT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
nombre	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
passwordl	VARCHAR(30)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm:

Lock Type:

```

1 CREATE TABLE `usuarios`.`usuarios` (
2   `idusuarios` INT NOT NULL,
3   `nombre` VARCHAR(30) NULL,
4   `passwordl` VARCHAR(30) NULL,
5   PRIMARY KEY (`idusuarios`))
6   ENGINE = InnoDB
7   DEFAULT CHARACTER SET = utf8
8   COLLATE = utf8_bin;
```

Tercer paso

Cuarto paso

# Workbench

Antes, se mostró el proceso que usualmente se sigue para crear una tabla haciendo uso de Workbench, herramienta que ya hemos instalado en la clase anterior.

En el **primer paso**, se crea un esquema. Dentro de este ultimo se agruparan distintos objetos, entre ellos, nuestras tablas.

En el **segundo paso**, creamos nuestra tabla

En el **tercer paso**, procedemos a agregar los campos a nuestra tabla

En el **último paso**, la herramienta nos ayuda a visualizar mediante lenguaje SQL el resultado de todo el proceso.

**No olvidemos que  
disponemos de  
clases de consulta!!**



ICARO Asociación Civil  
CUIT 30716564815  
[info@icaro.org.ar](mailto:info@icaro.org.ar)  
[www.icaro.org.ar](http://www.icaro.org.ar)

# Muchas gracias!



ICARO Asociación Civil  
CUIT 30716564815  
[info@icaro.org.ar](mailto:info@icaro.org.ar)  
[www.icaro.org.ar](http://www.icaro.org.ar)