

# BASE DE DATOS

Continuamos con nuestra aventura digital!!

#### Casa del futuro

En este taller conocerán uno de los motores más populares y gratuitos de base de datos y aprenderán a crear, diseñar y modelar Bases de Datos usando los diagramas de Entidad Relación (DER) con MySQL Workbench.

DATE: 23/06/2023







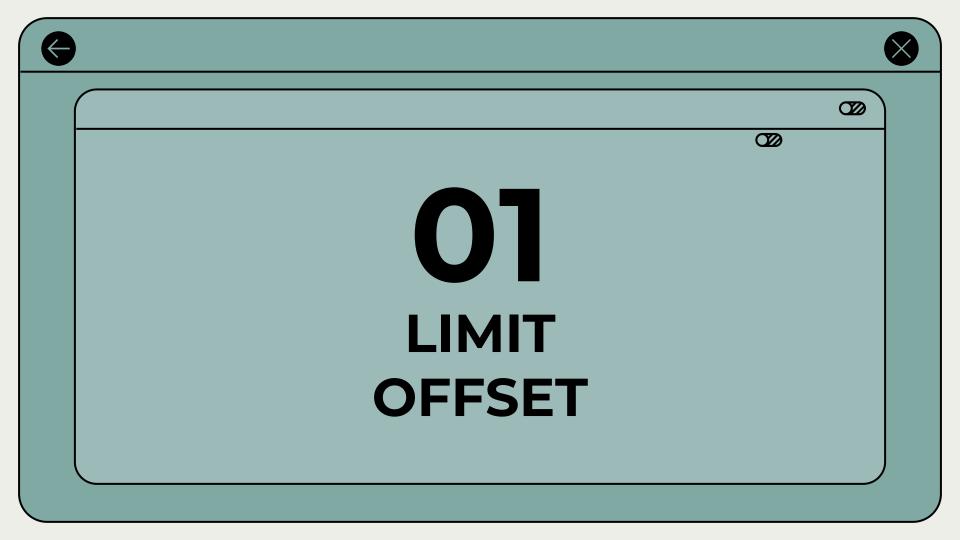


**ÍNDICE** 

01 02

LIMIT OFFSET

**ALIAS** 



#### Limit

Su funcionalidad es la de **limitar el número de filas** (registros/resultados) devueltas en las consultas SELECT. También establece el **número máximo** de registros a eliminar con DELETE.

```
SELECT nombre_columna1, nombre_columna2

SQL FROM nombre_tabla

LIMIT cantidad_de_registros;
```

#### Query de ejemplo

Teniendo una tabla **peliculas**, podríamos armar un top 10 con las películas que tengan más de 4 premios usando un **LIMIT** en la siguiente consulta:

```
SQL SELECT *
FROM peliculas
WHERE premios > 4
LIMIT 10;
```

#### Offset

- → En un escenario en donde hacemos una consulta de todas las películas de la base de datos, la misma nos devolvería muchos registros. Usando un **LIMIT** podríamos aclarar un límite de 20.
- → ¿Pero cómo haríamos si quisiéramos recuperar sólo 20 películas pero salteando las primeras 10 de la tabla?
- → OFFSET nos permite especificar a partir de qué fila comenzar la recuperación de los datos solicitados.

SELECT id, nombre, apellido

FROM alumnos

LIMIT 20

OFFSET 20:

**Seleccionamos** las columnas id, nombre y apellido.

SELECT id, nombre, apellido

FROM alumnos

de la tabla alumnos.

LIMIT 20

OFFSET 20;

SELECT id, nombre, apellido

FROM alumnos

LIMIT 20

OFFSET 20;

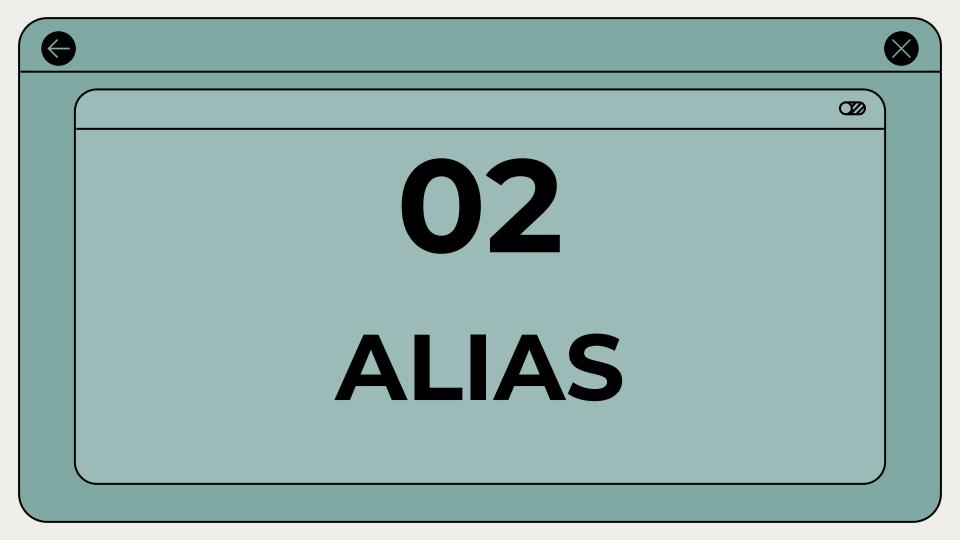
**Limitamos** los registros de la tabla resultante a **20** registros.

SELECT id, nombre, apellido FROM alumnos

LIMIT 20

OFFSET 20;

**Desplazamos** los resultados 20 posiciones para que se muestre desde la posición **21**.



#### **Alias**

Los **alias** se usan para darle un nombre temporal y más amigable a las **tablas**, **columnas** y **funciones**. Los **alias** se definen durante una consulta y persisten **solo** durante esa consulta.

Para definir un alias usamos las iniciales **AS** precediendo a la columna que estamos queriendo asignarle ese alias.

```
SQL SELECT nombre_columna1 AS alias_nombre_columna1 FROM nombre_tabla;
```

# Alias para una columna

SELECT razon\_social\_cliente AS nombre

SQL FROM cliente

ORDER BY nombre

#### Alias para una columna

SELECT razon\_social\_cliente AS nombre

FROM cliente

ORDER BY nombre;

Seleccionamos la columna

razon\_social\_cliente y le asignamos el **alias** nombre.

#### Alias para una columna

SELECT razon\_social\_cliente AS nombre

FROM cliente

ORDER BY nombre;

En el **FROM** elegimos tabla cliente.
Con el **ORDER BY** ordenamos los registros con la columna nombre.

# Alias para una tabla

SQL SELECT nombre, apellido, edad FROM alumnos\_comision\_inicial AS alumnos;

#### Alias para una tabla

SELECT nombre, apellido, edad

FROM alumnos\_comision\_inicial AS alumnos;

Seleccionamos las columnas nombre, apellido y edad.

#### Alias para una tabla

SELECT nombre, apellido, edad

FROM alumnos\_comision\_inicial AS alumnos;

Hacemos la consulta sobre la tabla alumnos\_comision\_inicial y le **asignamos** el **alias** alumnos.

No es recomendable asignar más de una palabra dentro de un alias. En el caso de necesitarlo, utilizar "\_". 44

De este modo, podemos darle alias a las columnas y tablas que vamos trayendo y hacer más legible la manipulación de datos, teniendo siempre presente que los alias no modifican los nombres originales en la base de datos.

