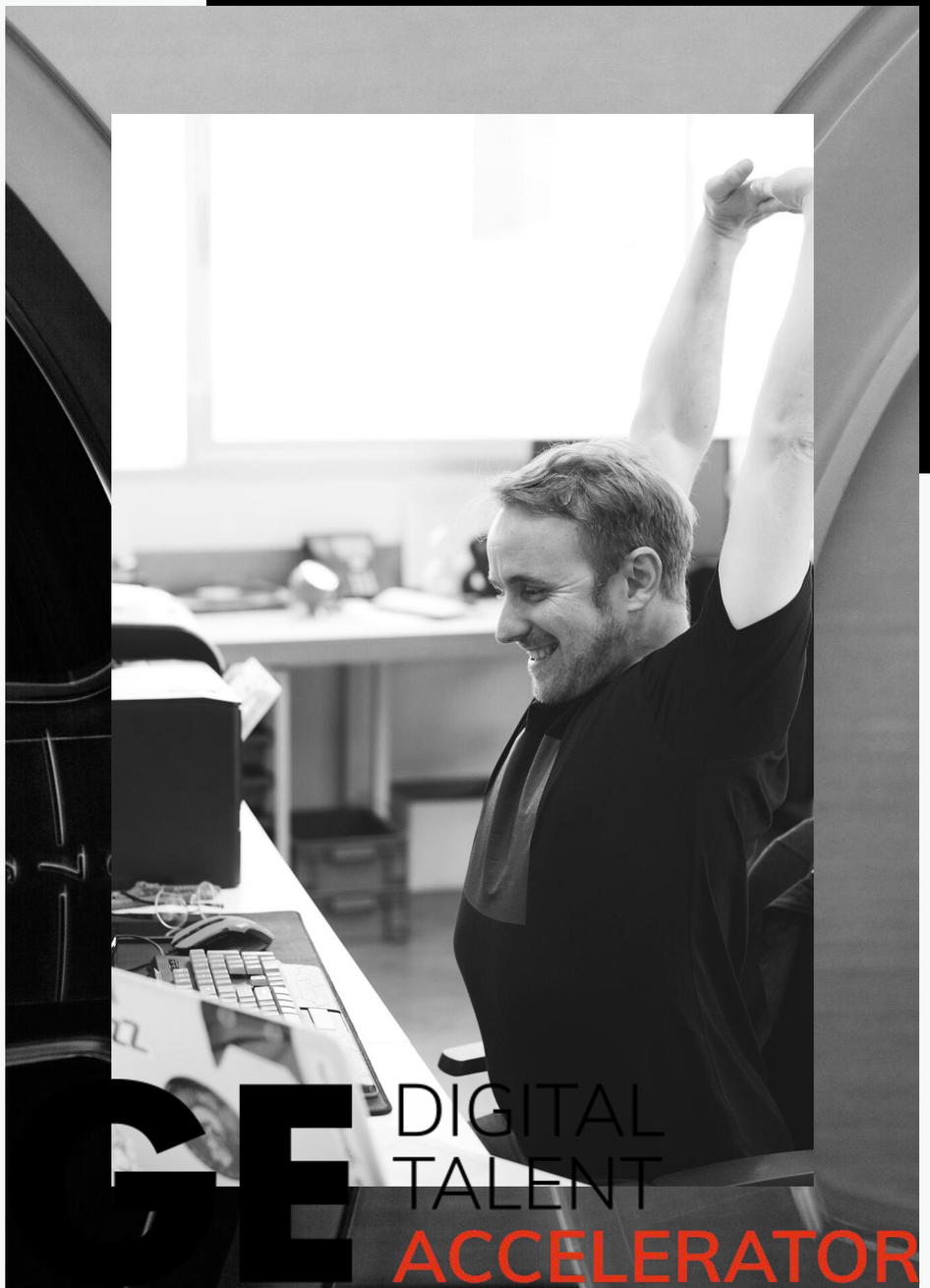
Bases de Datos Relacionales

Fundación ONCE | Inserta
Cofinanciado por la Unión Europea
MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL
Fondos Europeos

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Índice



THE BRIDGE DIGITAL TALENT ACCELERATOR

Base de datos

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Base de datos

Una base de datos es una colección de datos, que pueden estar representados por información acerca de cualquier tema. Un número telefónico, es un dato, un nombre, es un dato, un libro contiene datos... De qué tratan los datos en realidad no es importante en el contexto de la definición de datos.

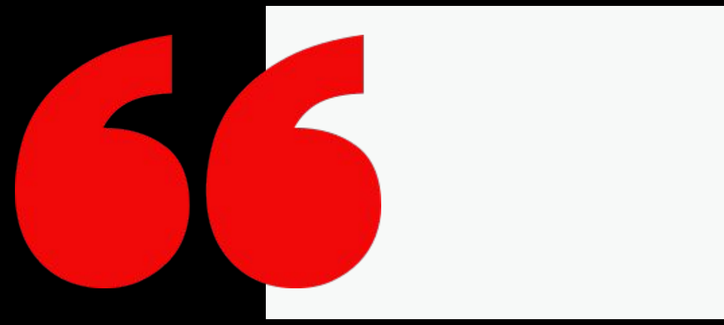
Por ejemplo, una agenda es conceptualmente una base de datos organizada por días. Y cada día contiene datos.

SQL

THE BRIDGE
DIGITAL TALENT ACCELERATOR

SQL

Structured Query Lenguage (SQL),es un lenguaje estandarizado que nos permite comunicarnos con las bases de datos relacionales.



¿Qué tipos de bases de datos SQL existen?

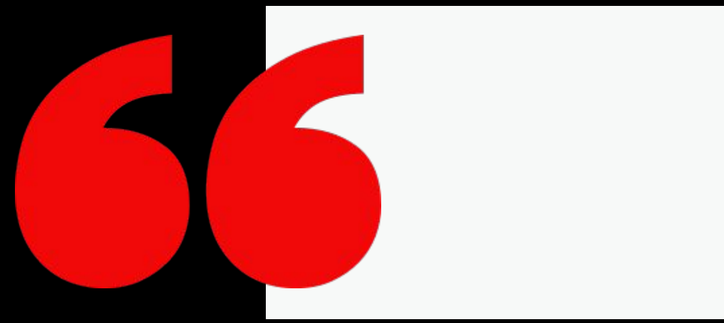
THE BRIDGE
DIGITAL TALENT ACCELERATOR

¿Qué tipos de bases de datos SQL existen?

Hay varios tipos de base de datos SQL (**MySQL** , **Oracle PLSQL**, **Microsoft SQL Server**, **PostgreSQL**, etc.)

Afortunadamente, **la mayoría son muy parecidas y la lógica es la misma**. Tienen alguna sentencia diferente, pero por lo general, **aprender un tipo de SQL te permitirá manejar cualquiera de las otras**.

¿Que es MySQL?



THE BRIDGE
DIGITAL TALENT ACCELERATOR

MySQL

MySQL es considerada la **base de datos de código abierto más popular del mundo** que utiliza el lenguaje de consulta estructurado (SQL).

Características MySQL

Basado en un "modelo relacional" de datos.

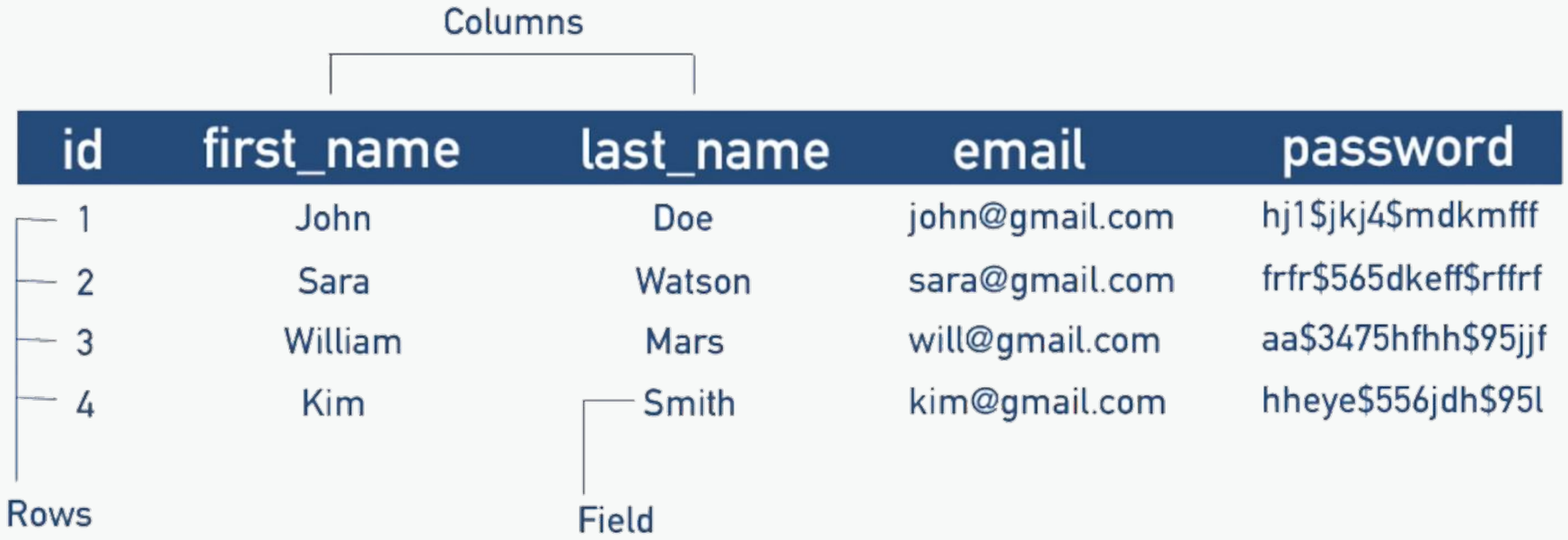
Utiliza "tablas" con "columnas" y "filas".

Las tablas pueden relacionarse entre sí por claves.

THE BRIDGE

Tablas

Cada tabla de la base de datos tiene **columnas**, **filas** y **campos**. Las columnas especifican el tipo de datos, mientras que las filas contienen los datos reales en sí. A continuación se muestra cómo podría imaginar una tabla MySQL.



THE BRIDGE

Tipos de datos

MySQL DATA TYPES

DATE TYPE	SPEC
CHAR	String (0 - 255)
VARCHAR	String (0 - 255)
TINYTEXT	String (0 - 255)
TEXT	String (0 - 65535)
BLOB	String (0 - 65535)
MEDIUMTEXT	String (0 - 16777215)
MEDIUMBLOB	String (0 - 16777215)
LONGTEXT	String (0 - 4294967295)
LONGBLOB	String (0 - 4294967295)
TINYINT	Integer (-128 to 127)
SMALLINT	Integer (-32768 to 32767)
MEDIUMINT	Integer (-8388608 to 8388607)

DATA TYPE	SPEC
INT	Integer (-2147483648 to 214748-3647)
BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
FLOAT	Decimal (precise to 23 digits)
DOUBLE	Decimal (24 to 53 digits)
DECIMAL	"DOUBLE" stored as string
DATE	YYYY-MM-DD
DATETIME	YYYY-MM-DD HH:MM:SS
TIMESTAMP	YYYYMMDDHHMMSS
TIME	HH:MM:SS
ENUM	One of preset options
SET	Selection of preset options
BOOLEAN	TINYINT(1)

Copyright © mysqltutorial.org. All rights reserved.

THE BRIDGE

Instalación

MySQL Server

https://dev.mysql.com/downloads/windows/installer/8.0.html

MySQL Workbench

https://dev.mysql.com/downloads/workbench/

Video instalación

https://www.youtube.com/watch?v=9SFAjDC4Ies

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Trabajando con bases de datos

- Creamos nuestra base de datos y también podemos eliminarla si queremos.
- Luego seleccionamos la nueva base de datos.

```
// Crea la base de datos
CREATE DATABASE myFirstDB;
// Elimina la base de datos
DROP DATABASE myFirstDB;
// Selecciona la base de datos
USE myFirstDB;
// Elimina la tabla
DROP TABLE users;
// Vacía la tabla sin eliminarla
TRUNCATE TABLE;
```

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Creando tabla

- Ahora, finalmente podemos crear nuestra primera tabla.
- Además, podemos eliminar la tabla.

```
CREATE TABLE users(
   id INT AUTO_INCREMENT,
      first_name VARCHAR(100),
      last_name VARCHAR(100),
      email VARCHAR(50),
      password VARCHAR(20),
      register_date DATETIME,
    PRIMARY KEY(id)
);

DROP TABLE users;
```

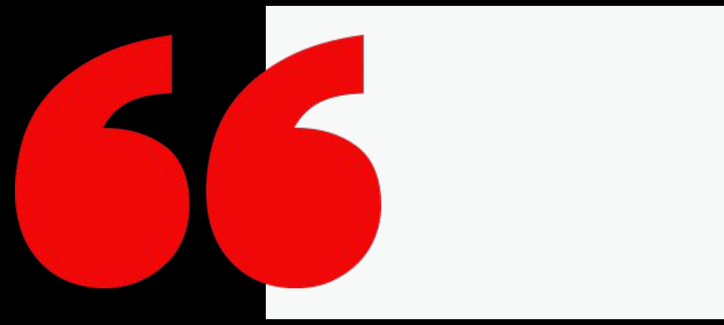
THE BRIDGE
DIGITAL TALENT ACCELERATOR

Cambiando tabla

- Puedes añadir una nueva columna, por ejemplo, la columna edad.
- Modificarla o eliminarla.

```sql
ALTER TABLE users ADD age VARCHAR(3);

ALTER TABLE users MODIFY COLUMN age
INT(3);

ALTER TABLE users DROP COLUMN age;
```

THE BRIDGE
DIGITAL TALENT ACCELERATOR



SELECT & INSERT

SELECT es la palabra clave SQL que permite que la base de datos sepa que desea recuperar datos.

INSERT inserta nuevas filas en una tabla existente.

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Insert & Select

- Creando nuestro primer usuario.
- También podemos añadir varios usuarios.
- Seleccionemos todos los usuarios con todo o con solo el nombre y apellido.

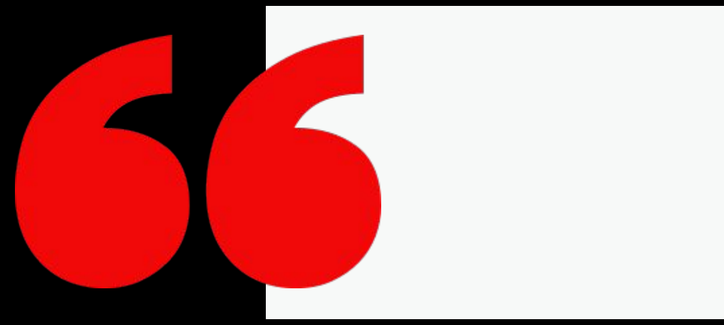
```
INSERT INTO users (first_name,
last_name, email, password,
register_date) values ('John', 'Doe',
'john@gmail.com', '123456', now());
```

```
INSERT INTO users (first_name,
last_name, email, password,
register_date) values ('Fred',
'Smith', 'fred@gmail.com', '123456',
now()), ('Sara', 'Watson',
'sara@gmail.com', '123456',now());
```

```
SELECT * FROM users; //Select all

SELECT first_name, last_name FROM
users;
```

THE BRIDGE
DIGITAL TALENT ACCELERATOR

WHERE



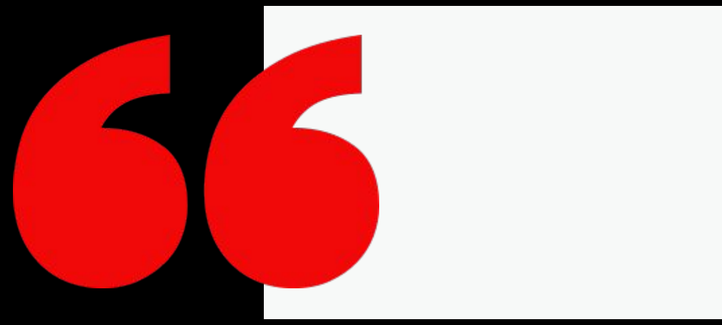
Una cláusula **WHERE** en SQL especifica que una declaración de lenguaje de manipulación de datos de SQL solo debe afectar a las filas que cumplen los criterios especificados.

WHERE

- Puedes filtrar usuarios para determinadas características, por ejemplo, por su edad.

```
SELECT * FROM users WHERE age=23;
```

THE BRIDGE
DIGITAL TALENT ACCELERATOR



DELETE & UPDATE

En **MySQL**, podemos eliminar datos usando las declaraciones **DELETE**.

La instrucción **UPDATE** se usa para cambiar el valor de las columnas en las filas seleccionadas de una tabla.

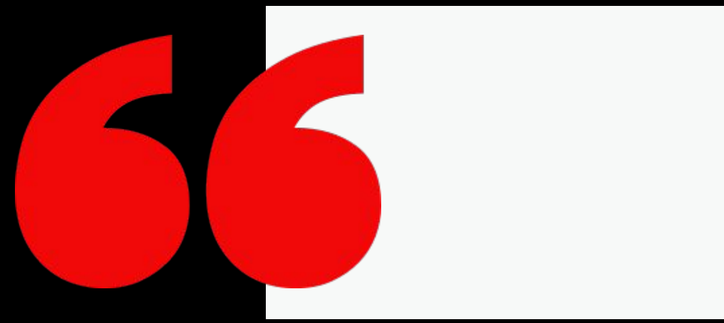
THE BRIDGE
DIGITAL TALENT ACCELERATOR

DELETE & UPDATE

- **Actualizar una fila de algún usuario**, puede hacerlo. En este caso, actualizamos la edad del usuario con id = 2.
- Eliminar toda la fila de un usuario.

```sql
UPDATE users SET age = 23 WHERE id = 2;

DELETE FROM users WHERE id = 3;
```

ORDER BY & BETWEEN



ORDER BY se utiliza para especificar el orden de clasificación del conjunto de resultados.

La condición **BETWEEN** de MySQL se usa para recuperar valores dentro de un rango en una instrucción SELECT, INSERT, UPDATE o DELETE.

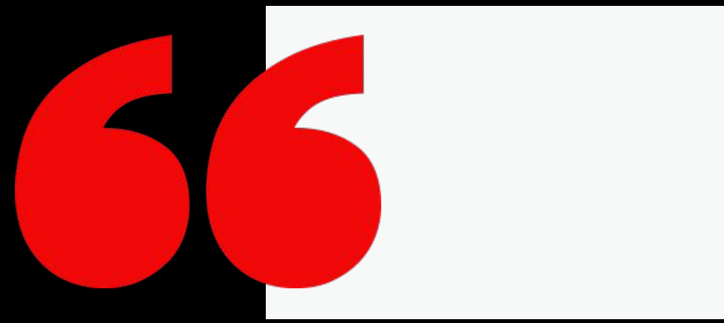
THE BRIDGE
DIGITAL TALENT ACCELERATOR

ORDER BY & BETWEEN

- Puedes **ordenar** tus usuarios por id, edad, etc... Con **ORDER BY ASC** o **ORDER BY DESC**.
- También podemos **buscar al usuario con una edad entre** los 20 y los 25 años.

```
//Order by ASC or DESC

//DESC

SELECT * FROM users ORDER BY id DESC;

//ASC

SELECT * FROM users ORDER BY id ASC;

//BETWEEN

SELECT * FROM users WHERE age BETWEEN
20 AND 25;
```

THE BRIDGE
DIGITAL TALENT ACCELERATOR



DISTINCT & CONCATENATE

DISTINCT va en contra de toda la fila de TODAS las columnas.

Con **CONCATENATE** puedes concatenar los valores mientras selecciona filas de la tabla.

THE BRIDGE
DIGITAL TALENT ACCELERATOR

CONCATENATE

- Ahora concatenamos first_name y last_name y nos devolverá el nombre completo.

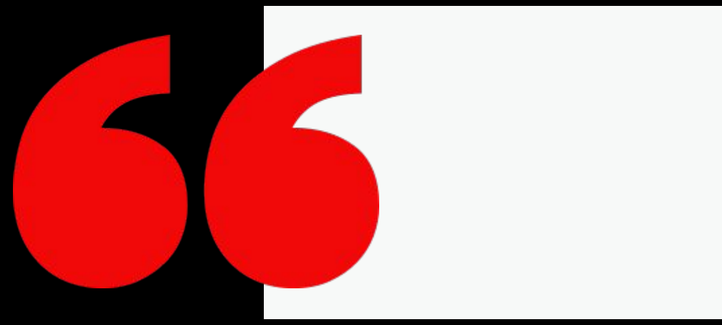
```
SELECT CONCAT(first_name, ' ', last_name) AS 'Name' FROM users;
```

DISTINCT

- Usamos **DISTINCT** con la columna de edad para que sólo devuelva las columnas que son diferentes.

```
SELECT DISTINCT age FROM users;
```

THE BRIDGE
DIGITAL TALENT ACCELERATOR



LIKE & NOT LIKE

El operador **LIKE & NOT LIKE** en SQL se usa en una columna que es de tipo varchar. Por lo general, se usa con%, que se usa para representar cualquier valor de string. La string que pasamos a este operador no distingue entre mayúsculas y minúsculas.

Se usa con **WHERE** para buscar un patrón específico en una columna.

THE BRIDGE
DIGITAL TALENT ACCELERATOR

LIKE & NOT LIKE

- Busca un usuario cuyo apellido **contenga** una 'a'.
- Busca un usuario cuyo first_name **comience** con 'm'.
- Busca un usuario cuyo first_name **termine con** 'd'.
- Busca un usuario cuyo apellido **no contenga** 'r'.

```sql
SELECT * FROM users WHERE last_name LIKE '%a%';

SELECT * FROM users WHERE first_name LIKE 'm%';

SELECT * FROM users WHERE first_name LIKE '%d';

SELECT * FROM users WHERE last_name NOT LIKE '%r%';
```

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Aggregate functions

- Una función agregada realiza un cálculo en varios datos y devuelve un solo valor.

```sql
SELECT COUNT(id) FROM users;

SELECT MAX(age) FROM users;

SELECT MIN(age) FROM users;

SELECT SUM(age) FROM users;
```

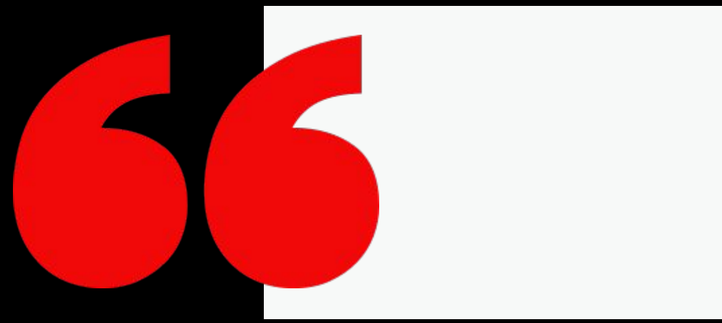
THE BRIDGE
DIGITAL TALENT ACCELERATOR

Group By

- La declaración GROUP BY **agrupa las filas que tienen los mismos valores** en filas de resumen.
- La instrucción GROUP BY se usa a menudo con funciones agregadas (COUNT, MAX, MIN, SUM, AVG) para agrupar el conjunto de resultados por una o más columnas.

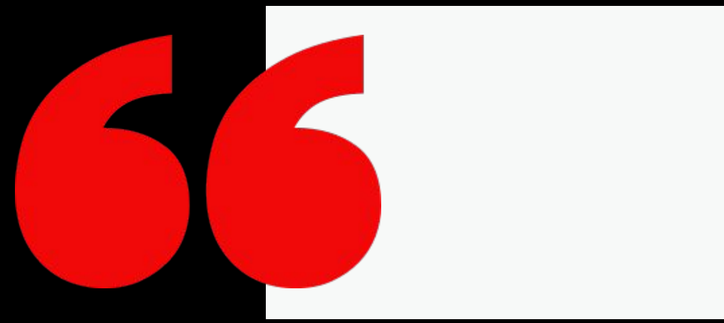
```sql
SELECT age, COUNT(age) FROM users GROUP BY age;
SELECT age, COUNT(age) FROM users WHERE age > 23 GROUP BY age;
```

Relaciones



THE BRIDGE
DIGITAL TALENT ACCELERATOR

Relaciones



Al crear una base de datos, el sentido común dicta que usemos tablas separadas para diferentes tipos de entidades. Algunos ejemplos son: clientes, pedidos, artículos, mensajes, etc ...

Pero también necesitamos tener relaciones entre estas tablas. Por ejemplo, los clientes realizan pedidos y los pedidos contienen artículos. Estas relaciones deben estar representadas en la base de datos.

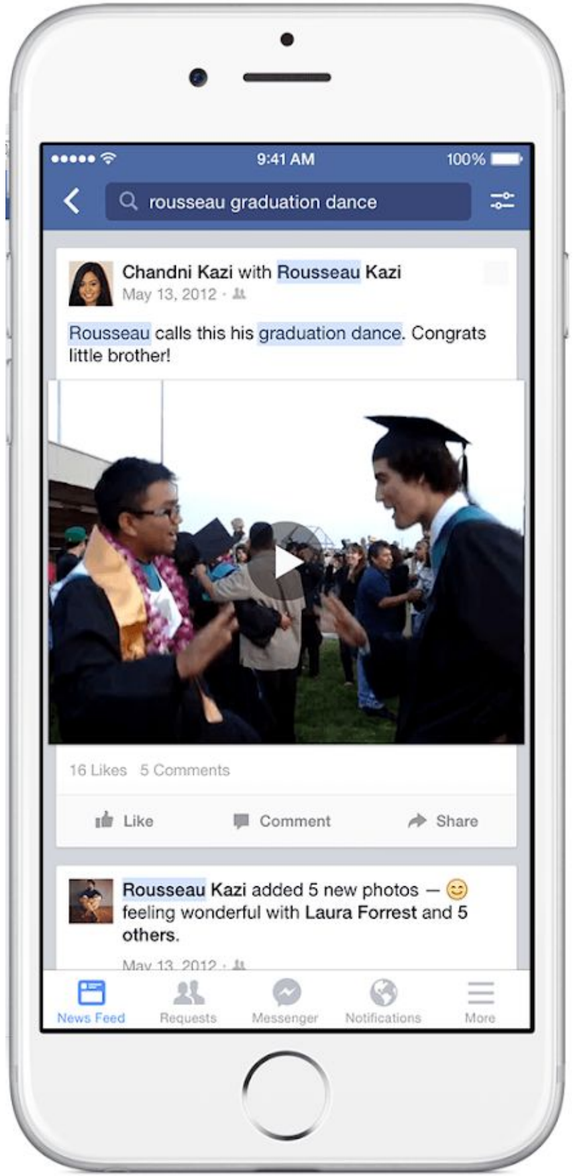
THE BRIDGE
DIGITAL TALENT ACCELERATOR

En Facebook hay usuarios y hay publicaciones.

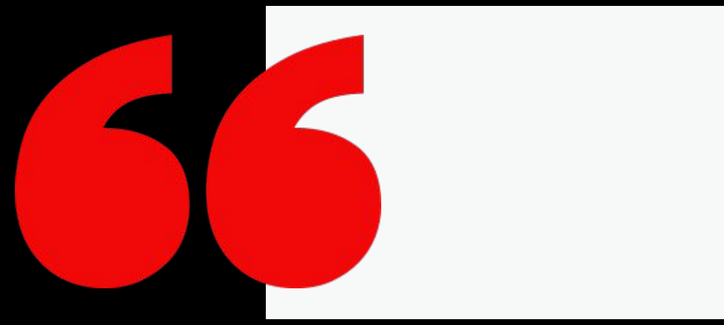
Post & users

En Facebook no vemos a los usuarios solos ni a las publicaciones solos.

Vemos una publicación con el nombre del usuario que lo ha hecho o si vamos al perfil del usuario vemos las publicaciones que tiene.



THE BRIDGE



Primary key & Foreign key

La primary key consta de una o más columnas cuyos datos se utilizan para identificar de forma única cada fila de la tabla.

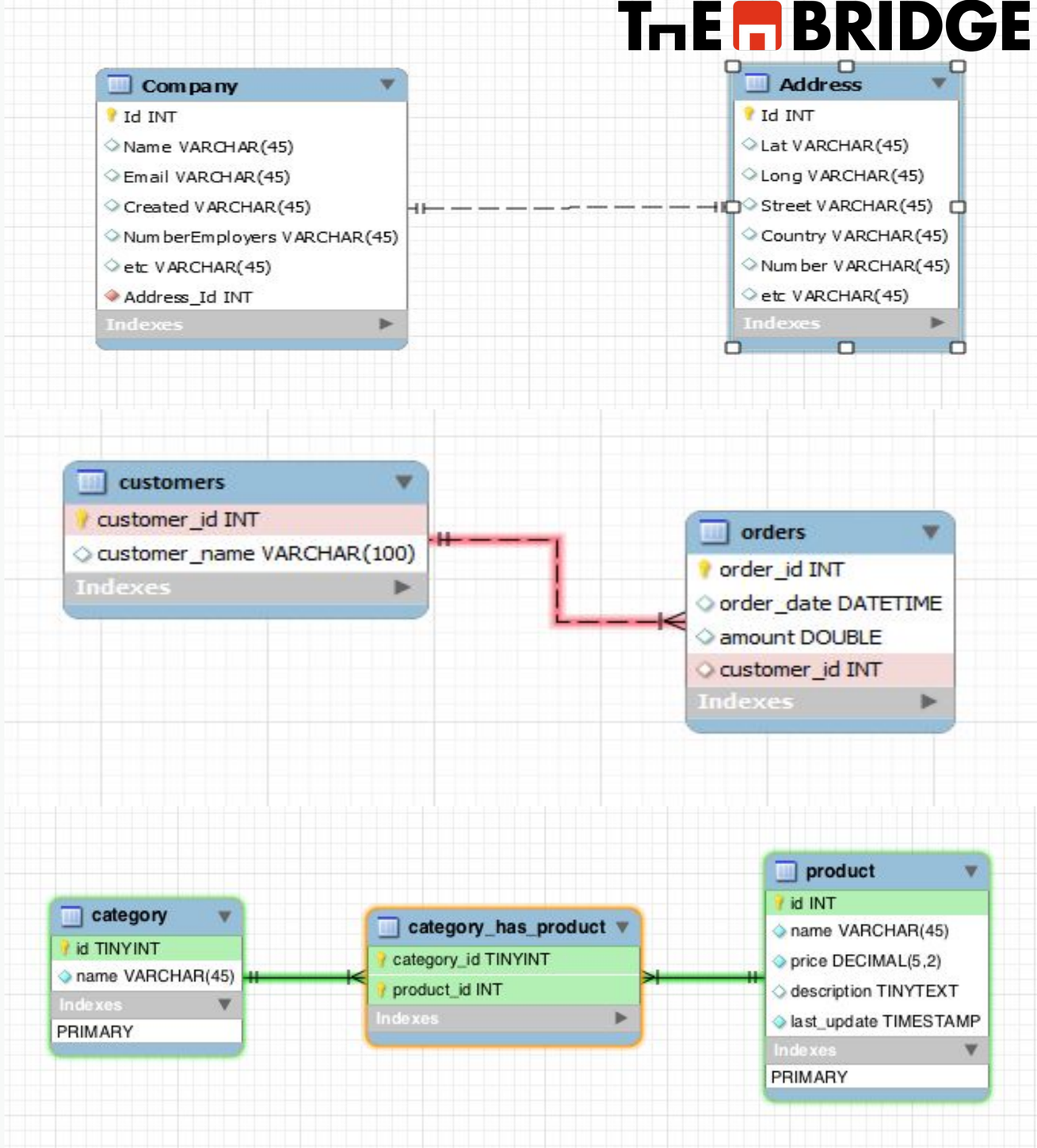
Una foreign key es un conjunto de una o más columnas en una tabla que hace referencia a la clave principal en otra tabla.

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Tipos relaciones SQL

En SQL, hay tres tipos de relaciones: uno a uno (1: 1), uno a muchos (1: N) o muchos a muchos (M: N) que se pueden modelar.

- **Uno a uno**. El dato de una tabla se relacionan con solo un dato de otra tabla
- **Uno a muchos**. El dato de una tabla se relaciona con muchos datos de otra tabla
- **Muchos a muchos**. Muchos datos de una tabla se relacionan con muchos datos de otra



¿Donde pongo la Foreign Key?

En SQL, hay tres tipos de relaciones: uno a uno (1: 1), uno a muchos (1: N) o muchos a muchos (M: N):

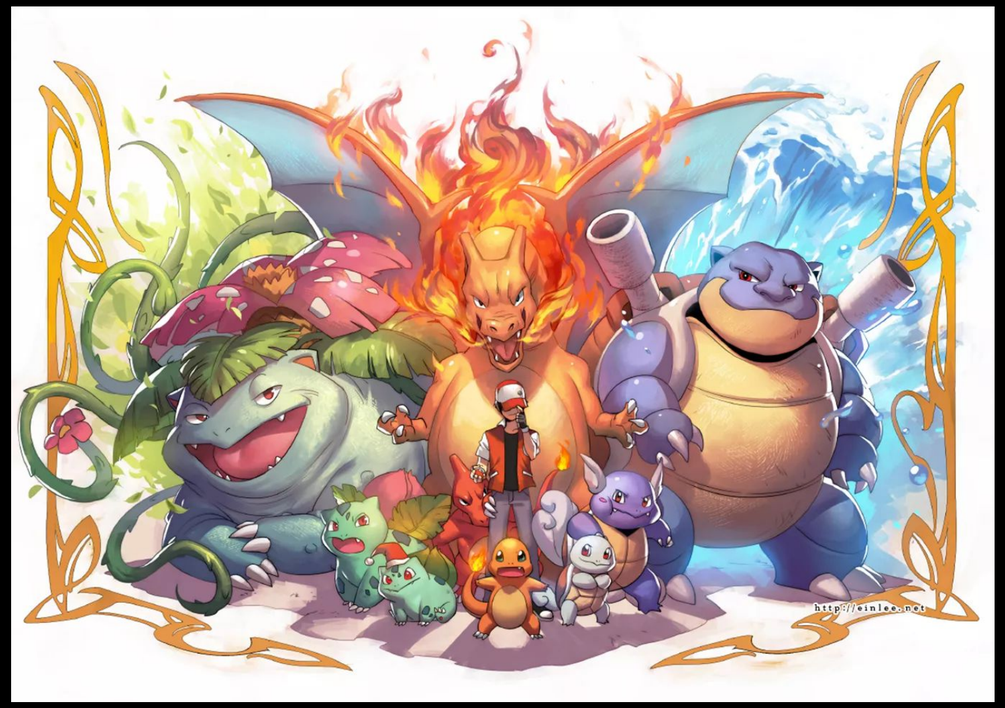
- **Uno a uno**. La foreign Key se pondrá en la tabla que podría ser el muchos. Por ejemplo **un estudiante solo puede tener un carnet** de universidad. Y un carnet solo puede pertenecer a un estudiante. La relación es uno a uno. ¿Donde pondrias la foreign Key? En los carnets, ya que, pueden caducar, por lo tanto un estudiante puede llegar a tener otro carnet nuevo.
- **Uno a muchos**. Imaginemos una red social, un usuario puede tener muchas publicaciones, pero una publicación solo puede ser de un usuario. En este caso el **“muchos” es publicaciones y por ello la foreign key iría en la tabla de publicaciones**.
- **Muchos a muchos**. Cuando tenemos una relación muchos a muchos, como por ejemplo pedidos y productos. **Se crearía una tabla intermedia y en esa tabla se guardarían las foreign keys**.

Ponte a prueba

THE BRIDGE

RELACIÓN POKEMON - TIPO

Hay muchos pokemon y cada uno es de un tipo(planta,agua,bicho,fuego,metal...)



THE BRIDGE
DIGITAL TALENT ACCELERATOR

Resulta que hemos descubierto que pueden haber pokemons con más de un tipo

THE BRIDGE
DIGITAL TALENT ACCELERATOR

RELACIÓN STORMTROOPERS-DARTH VADER

Darth Vader puede elegir cuantos soldados puede tener. Pero los soldados sólo tienen un Darth Vader.

Únete al lado oscuro

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Relación director - película

Cada película tiene un director y cada director puede dirigir una o varias películas. De las películas, sabemos su título, fecha de estreno y su género. Del director conocemos su nombre y dni.

THE BRIDGE
DIGITAL TALENT ACCELERATOR

Club - jugador

Un club puede tener jugando a varios jugadores, pero cada jugador solo puede jugar en un club

THE BRIDGE
DIGITAL TALENT ACCELERATOR

RELACIÓN JUGADOR EQUIPO DE LOL-PERSONAJES DEL JUEGO

Cada jugador de un equipo de League of Legends tiene la opción de elegir un personaje. Y un personaje solo puede ser elegido por un jugador.

Cada jugador(nombre, apellidos, DNI) sólo puede elegir un personaje(nombre, habilidad)

THE BRIDGE
DIGITAL TALENT ACCELERATOR