

name

Carlos

Melissa

Fernando

# Comandos de SQL

## Terminología

DDL

**Data Definition Language**

Create, Alter, Drop, Truncate

DML

**Data manipulation Language**

Insert, Delete, Update

TCL

**Transaction Control Language**

Commit, Rollback

DQL

**Data Query Language**

Select

# Aggregate Functions & Filtering

Resumen

## Aggregate functions

- Count
- Sum
- Max
- Min
- Group By
- Having
- Order By

## Filtrando data

- Like
- In
- Is Null
- Is Not Null
- Where
- And
- Or
- Between

# Estructura general de un Select

	<b>SELECT</b>	<b>distinct</b> *, campos, alias, funciones
<b>JOINS</b>	<b>WHERE</b>	Condición, condiciones, and, or, in, like
	<b>GROUP BY</b>	Campo agrupador, ALL
	<b>HAVING</b>	Condición
	<b>ORDER BY</b>	Expresión, ASC, DESC
	<b>LIMIT</b>	Valor, ALL
	<b>OFFSET</b>	Punto de inicio



**SELECT** \* **FROM** Tabla A  
**WHERE** ( Sub Query from Tabla B )

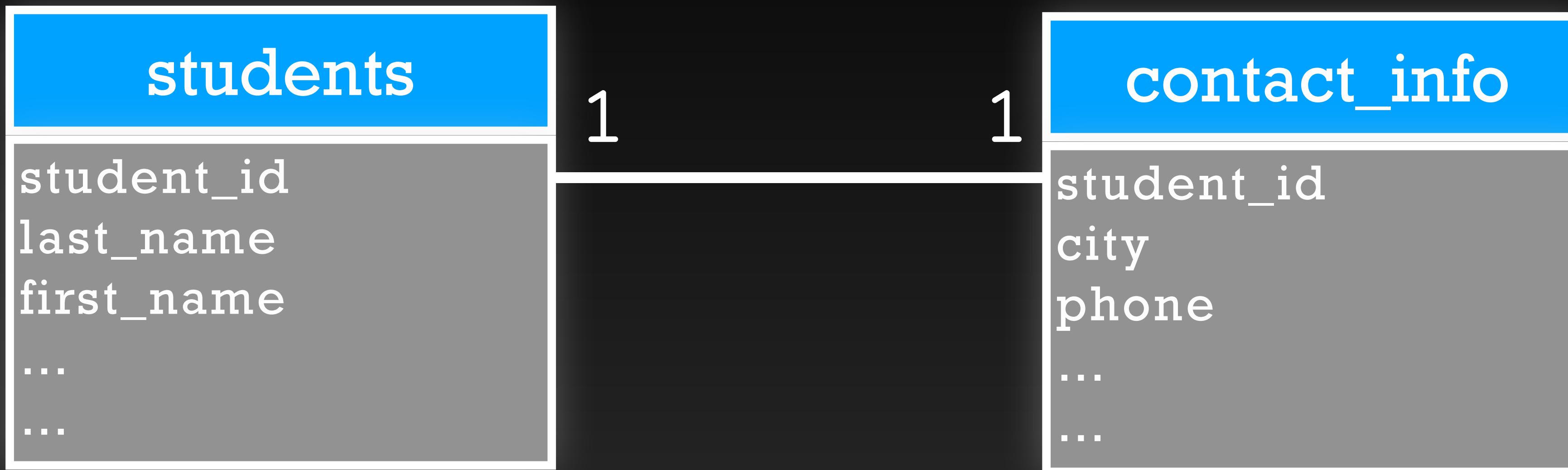
Tabla A
1...1,000,000

Tabla B
1...10

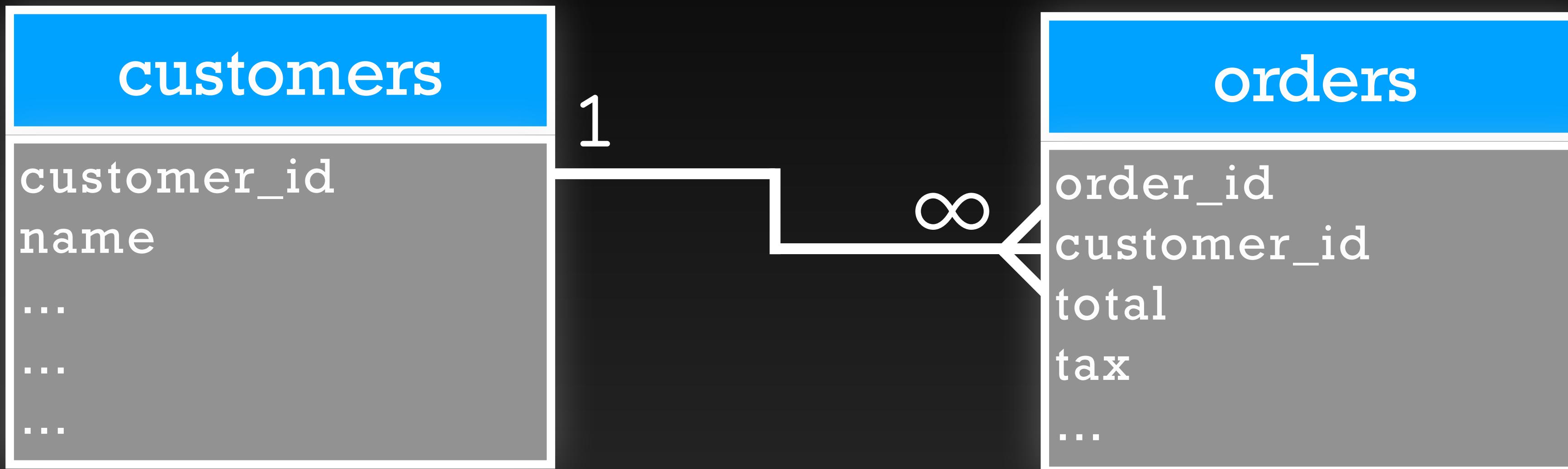
# Tipos de Relaciones

- **Uno a Uno** - One to One
- **Uno a muchos** - One to many
- **Relaciones a si mismas** - Self Joining relationships
- **Muchos a Muchos** - Many to Many

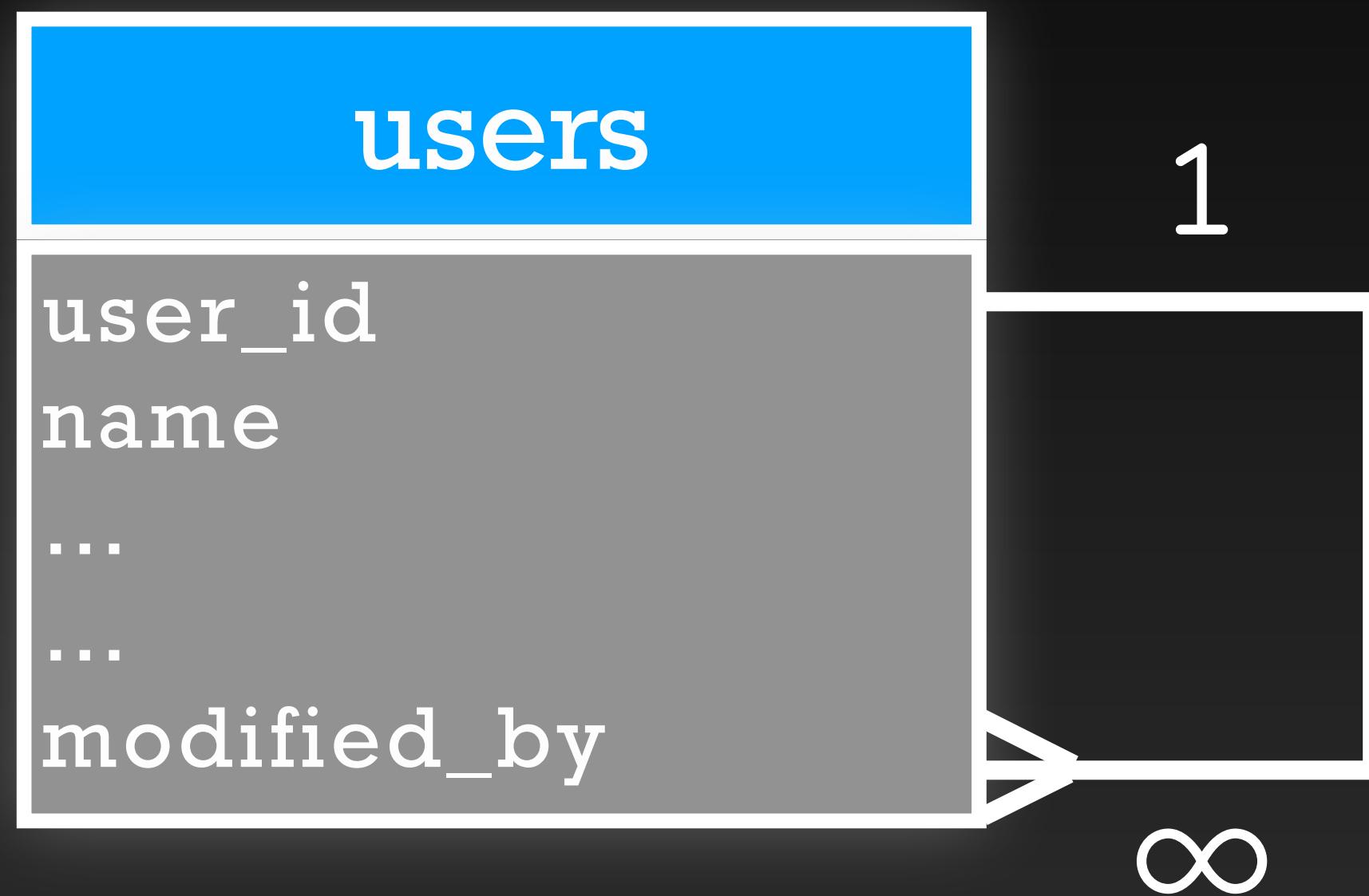
# Uno a Uno



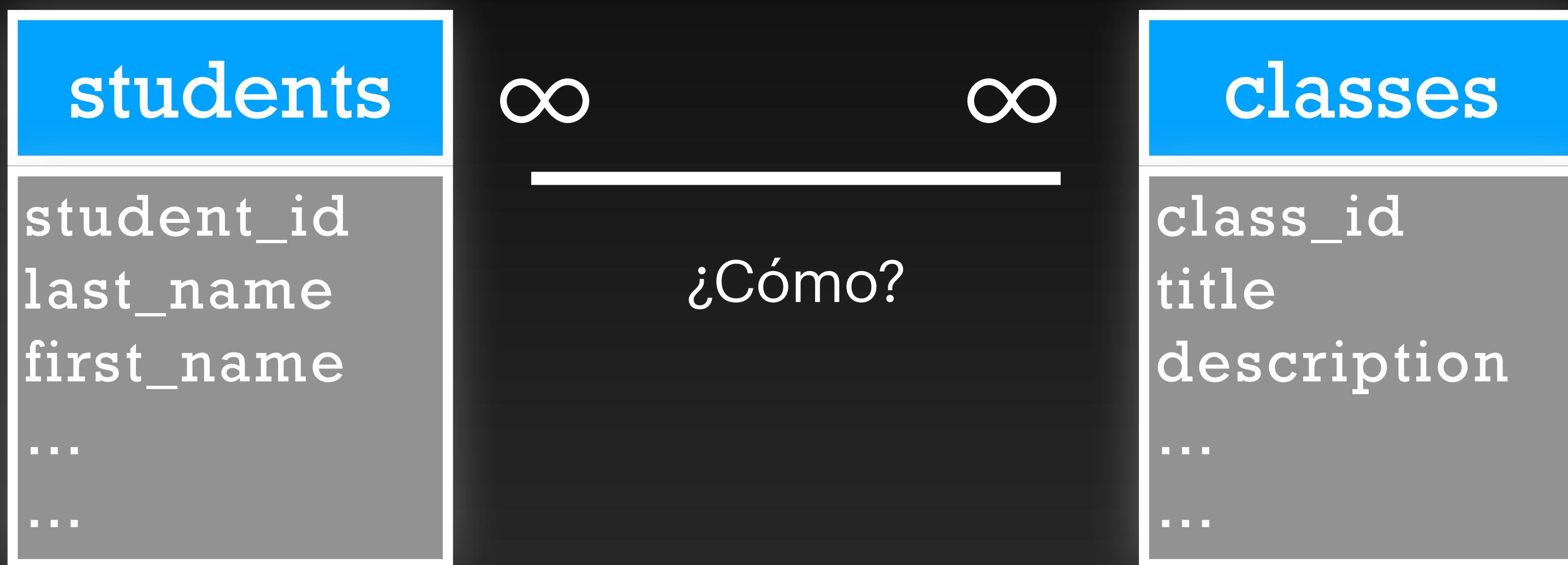
# Uno a muchos



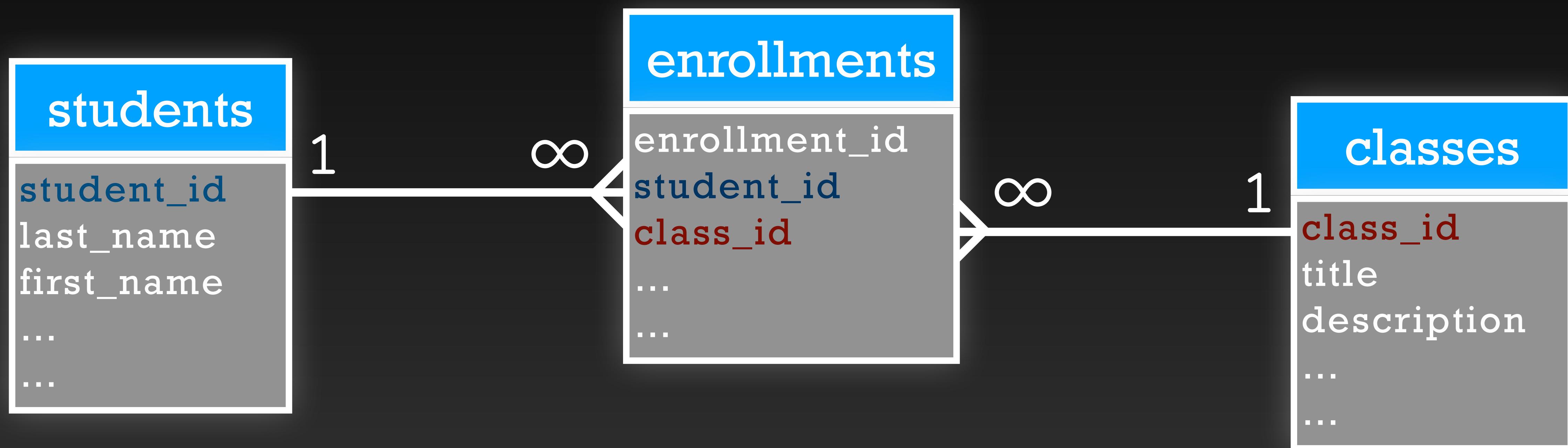
# Relación a si misma



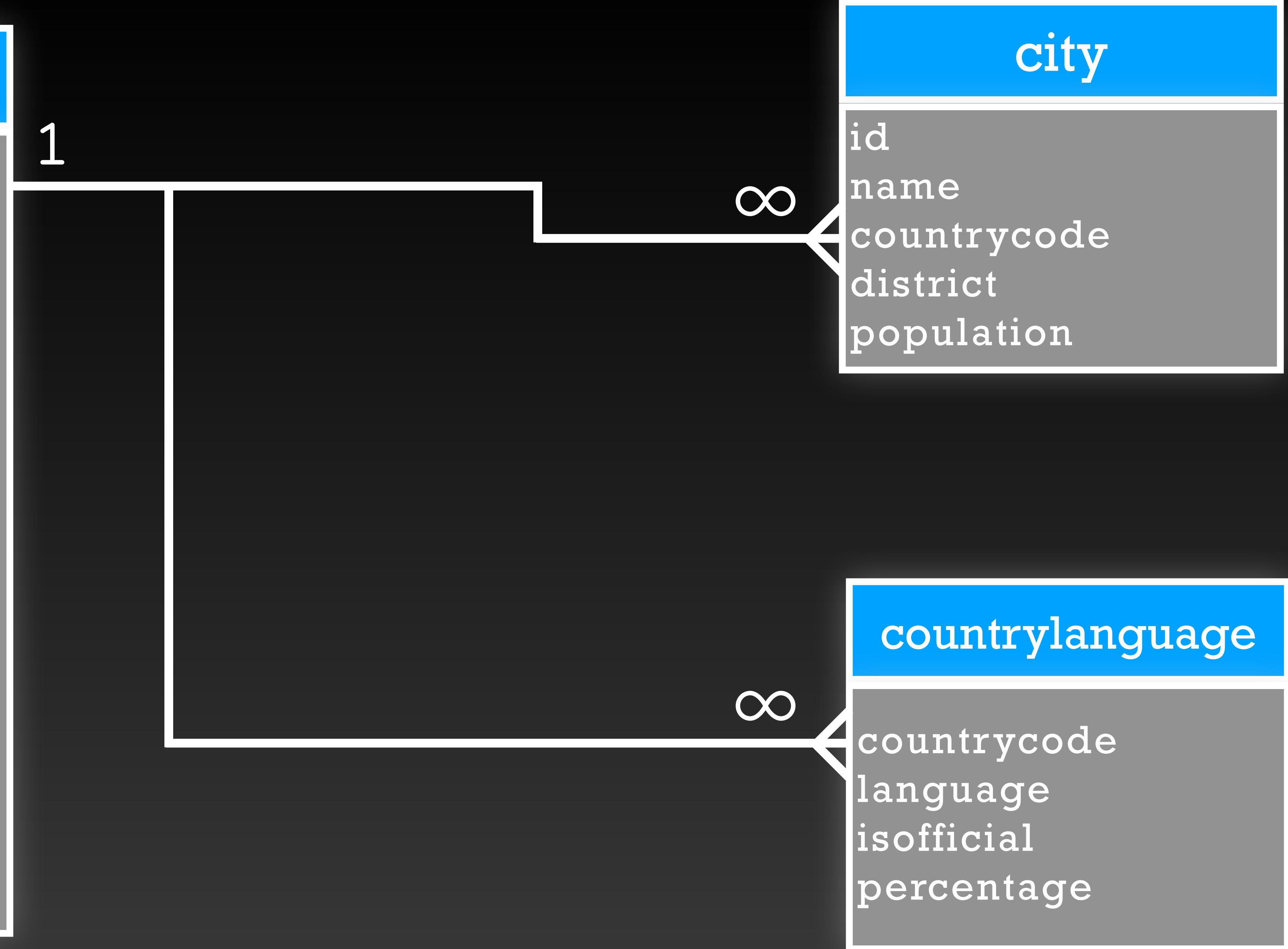
# Muchos a Muchos



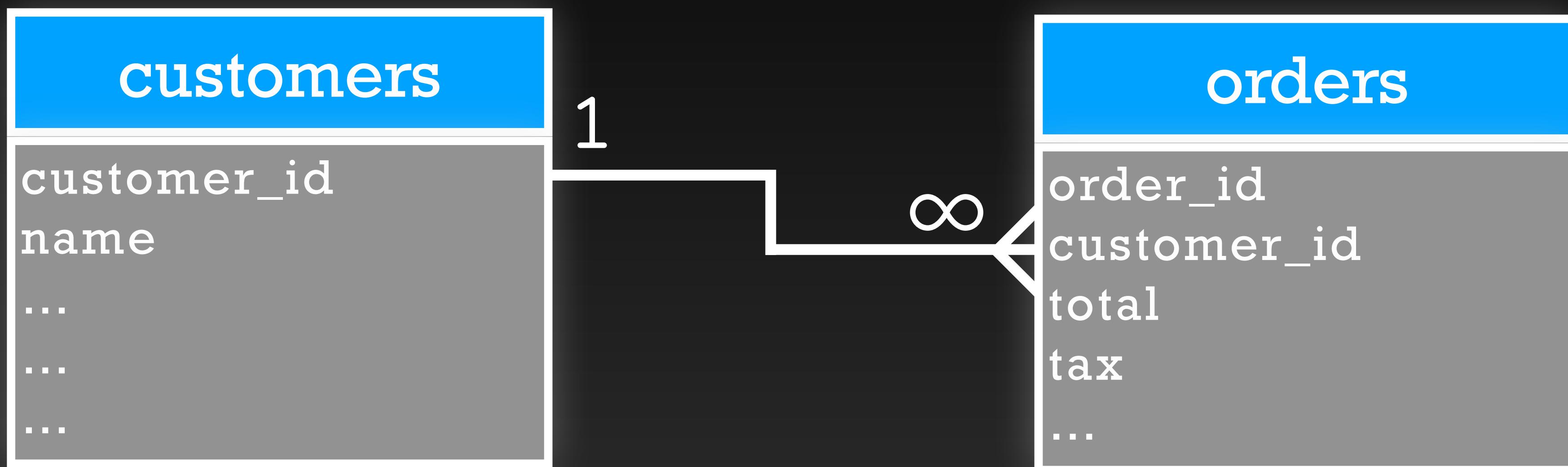
# Muchos a Muchos



country
code
name
continent
region
surfacearea
indepyear
population
lifeexpectancy
gnp
gnpold
localname
governmentform
headofstate
capital
code2



# Keys - Llaves



# Keys - Llaves

Primary  
Key

Super  
Key

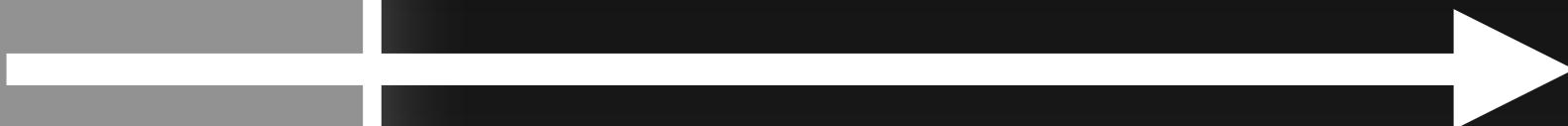
Candidate  
Key

Foreign  
Key

Composite  
Key

- Hay más, y todas las llaves sirven para identificar registros.
- Entre otras: Alternate Keys, Artificial Keys.

employee
employee_id
full_name
passport_number
license_number
email
SIN



## Primary Key

- Identifica un registro de forma única.
- Una tabla puede tener varios identificadores únicos.
- La llave primaria está basada en los requerimientos.

# Candidate Key

- Un atributo o conjunto de ellos que identifican de forma única.
- Menos la llave primaria, los demás se consideran claves candidatas

employee

employee\_id  
full\_name  
passport\_number  
license\_number  
email  
SIN

employee

employee\_id

full\_name

passport\_number

license\_number

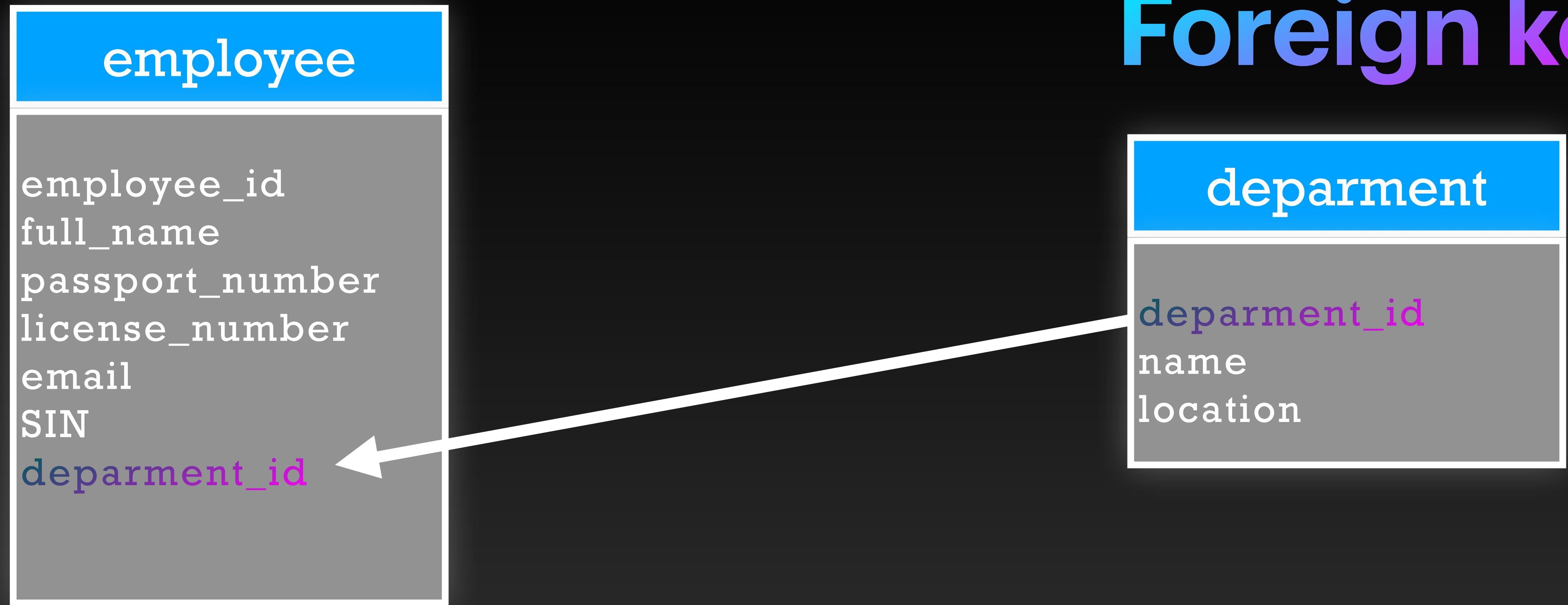
email

SIN

# SuperKey

- Es un conjunto de atributos que puede identificar de forma única.
- Es un superconjunto de una clave candidata.

# Foreign key



- Llaves foráneas son usadas para apuntar a la llave primaria de otra tabla.
- El department\_id en ambas tablas, deben de ser del mismo tipo de datos y longitud.

# Composite Key

- Cuando una clave primaria consta de más de un atributo, se conoce como clave compuesta.

employee

employee\_id

role

project\_id

...

...

...

...

# Index - índices



# Index - índices



# Index - índices



Bases de datos  
**Buenas prácticas**

# Preguntémonos:

- ¿Cuál es el objetivo de la misma?
- ¿Cómo usaremos la base de datos?

# En general un buen diseño

- Minimizar la redundancia.
- Proteger la precisión.
- Ser accesible.
- Cumplir las expectativas.

- Investigar previamente sobre diseños similares.
- Traiga las partes interesadas.
- Empárate del tema objetivo.

# Determinar los objetivos

- Mantén la simple.
- Usa estandarización.
- Considera futuras modificaciones.
- Mantén la deuda técnica a raya.
- Normalizar la data.
- Diseña a largo plazo.
- Crea documentación y diagramas.
- Prueba tu diseño

Principios a seguir

- No uses abreviaturas \*. Internationalization por i18n
- Se recomienda nombres tablas en singular.
- No re-inventes la rueda.
- Usa lo que el motor de base de datos te ofrece.
- Reglas, checks, llaves, indices, para evitar basura.
- Mantén la privacidad como prioridad.
- Nombres en inglés y evitar caracteres especiales.
- Todo en minúscula sin espacios.

## Principios a seguir

- Mantén la base de datos en su propio servidor.
- Mantén un modelado bajo versiones.
- Establece el tipo apropiado y precisión adecuada.
- No confíes en identificadores de terceros.
- Defina las llaves foráneas y relaciones.
- Si el esquema es muy grande, particiónalo.
- Evita nombres reservados. “user”, “table”, “create”

Principios a seguir

# Ideas a tener en mente

- Los nombres de tablas y campos vivirán más que las aplicaciones.
- Los nombres son contratos.
- La base de datos gobierna sobre los demás.

# Relaciones en singular

Tablas, vistas y cualquier relación en singular.

- Es posible tener una relación uno a uno, ¿seguiría esto siendo plural?
- En inglés, hay palabras que no tienen forma plural.  
“fish”, “species”, “series”
- Mucho software trabaja siguiendo esta regla de singular, y se usar mejor de esa forma. (Sentido semántico)

# Nombrado explícito

Evitar redundancia y lectura adicional

- person\_id vs id
- Las llaves foráneas deben de ser una combinación, por ejemplo:

```
CREATE TABLE team_member (
    team_id      bigint NOT NULL REFERENCES team(id),
    person_id    bigint NOT NULL REFERENCES person(id),
    CONSTRAINT team_member_pkey PRIMARY KEY (team_id, person_id));
```

- Indices deben de ser explícitos:

**person\_idx\_first\_name\_last\_name**

# Nombrado explícito

Así cuando algo falla, sabemos qué falló

```
CREATE TABLE team (
    id      bigserial PRIMARY KEY,
    name    text NOT NULL);

CREATE TABLE team_member (
    team_id  bigint REFERENCES team(id),
    person_id bigint REFERENCES person(id),
    CONSTRAINT team_member_pkey PRIMARY KEY (team_id, person_id));
```

PostgreSQL genera buenos  
nombres automáticamente

```
=# \d team_member
Table "public.team_member"
 Column | Type   | Modifiers
-----+--------+-----
 team_id | bigint | not null
 person_id | bigint | not null
Indexes:
"team_member_pkey" PRIMARY KEY, btree (team_id, person_id)
Foreign-key constraints:
"team_member_person_id_fkey" FOREIGN KEY (person_id) REFERENCES person(id)
"team_member_team_id_fkey" FOREIGN KEY (team_id) REFERENCES team(id)
```

# Ideas finales

- Si ya hay una estructura creada que sigue otras reglas, sigamos esas reglas. \*
- No mezclemos ideologías si no pensamos cambiar todo.
- Estos pasos son útiles para empezar nuevos diseños.

# Citas bibliográficas

RootSoft. "RootSoft/Database-Naming-Convention: Database Naming Conventions & Best Practices."

*GitHub*, [github.com/RootSoft/Database-Naming-Convention](https://github.com/RootSoft/Database-Naming-Convention). Accessed 18 July 2023.

Microsoft. Microsoft Support. (n.d.).

<https://support.microsoft.com/en-gb/office/database-design-basics-eb2159cf-1e30-401a-8084-bd4f9c9ca1f5>

Jefe Carlos

SubJefe Maria

SubJefe Pedro