

# Bases de Datos

## Ayudantía 5

# Tarea Chica 2

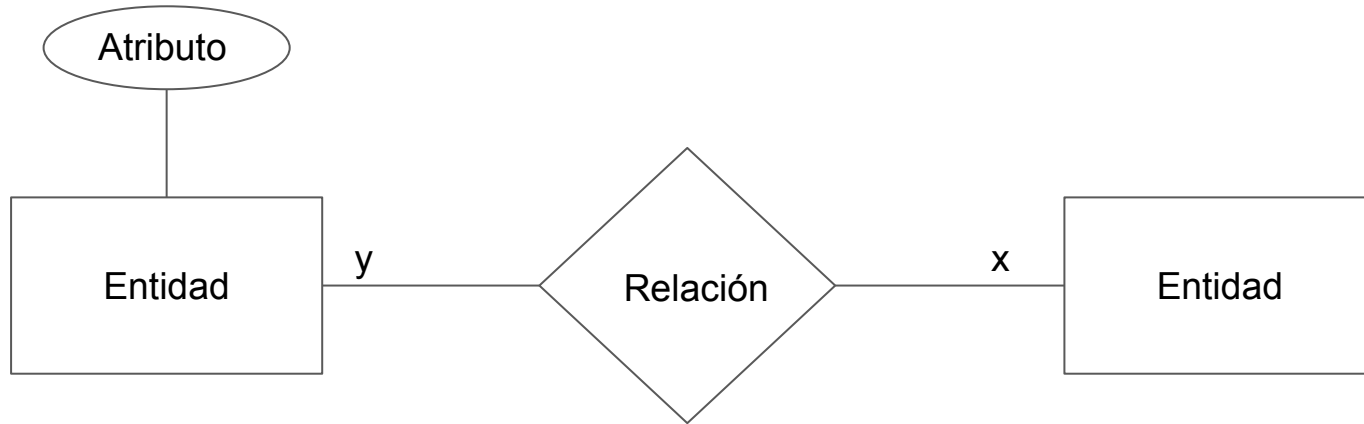
- Diagrama E/R
- Consultas SQL

# Diagrama de Entidad - Relación

# Elementos

- Entidades: objeto o concepto con existencia independiente y única.
- Atributos: características que identifican a una entidad.
  - Primary Key
  - Foreign Key
- Relación: dependencia o asociación entre entidades
  - Cardinalidad: indica la cantidad de entidades que pueden ir relacionadas con una entidad dada. (1:1, 1:N, N:M)

# Formato Diagrama E/R

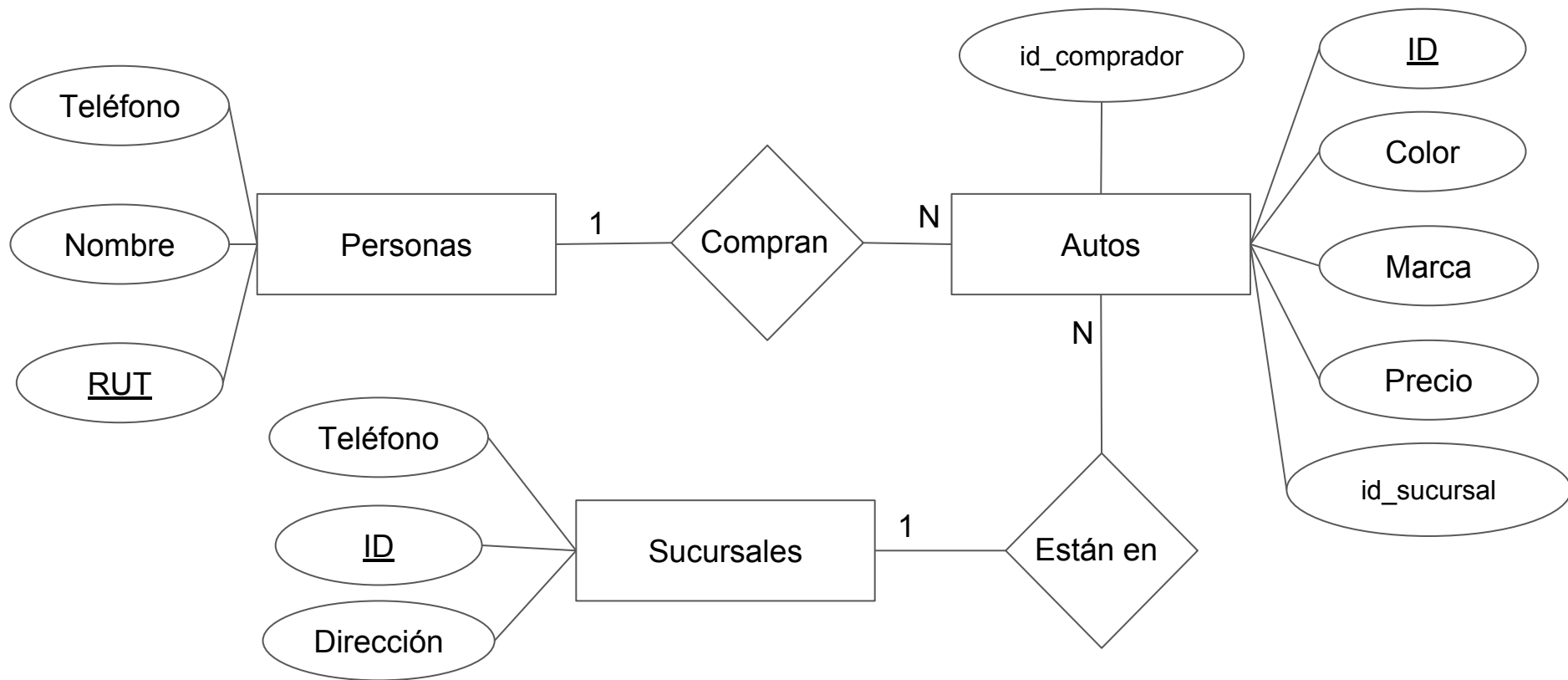


# Actividad

La empresa *DCCar* de venta de autos que dado a su gran éxito necesita implementar una base de datos de compra de autos. Debe guardar información sobre el auto (identificador, color, precio, marca), sobre los compradores (RUT, nombre, teléfono) y sucursales (dirección, teléfono).

Haga un diagrama de clases que represente la base de datos de la empresa *DCCar*. Considere en cómo se relacionan todas las entidades. No olvide agregar la cardinalidad a las relaciones.

# Actividad

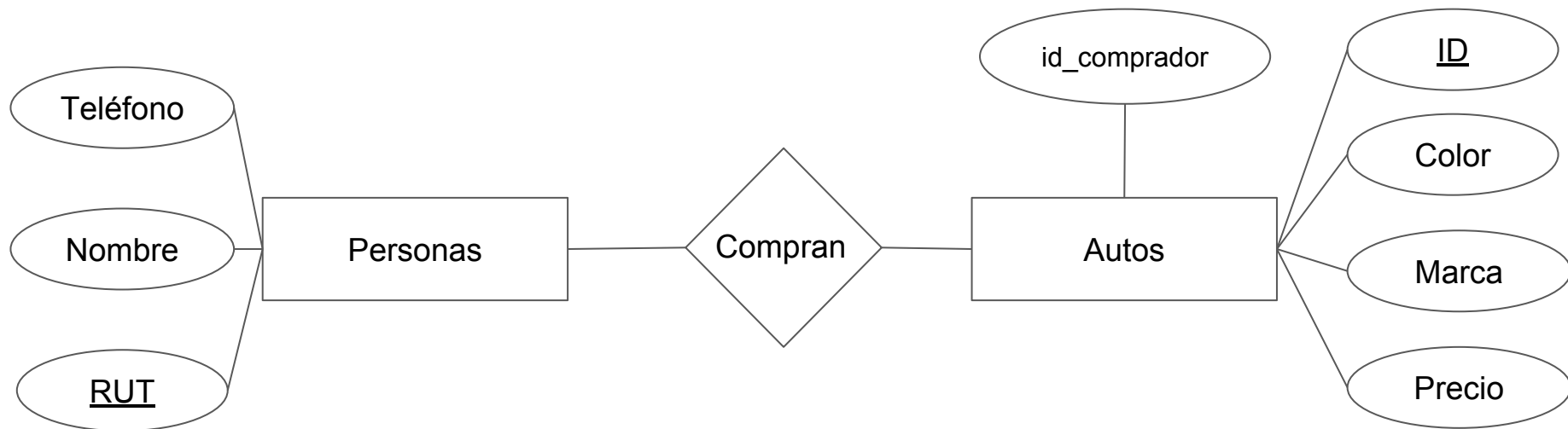


# SQL:

# Structured Query Language



# Bases de datos que usaremos



# Base de datos

## Autos

id	color	marca	precio(\$)	dueño
1	rojo	KIA	8000	7.654.184-1
2	verde	Mercedes	10000	15.904.874-5
3	rojo	Mercedes	20000	18.743.921-5
4	negro	Fiat	8000	17.928.4932-k
5	azul	Ford	15000	18.278.191-3

## Personas

rut	nombre	edad	teléfono
7.654.184-1	Juan	55	876326
15.904.874-5	Diego	32	873492
17.928.4932-k	Marcela	28	248273
18.278.191-3	Silvia	25	473492
18.743.921-5	Pablo	23	725897

# Base de datos

## Autos

id	color	marca	precio(\$)	dueño
1	rojo	KIA	8000	7.654.184-1
2	verde	Mercedes	10000	15.904.874-5
3	rojo	Mercedes	20000	18.743.921-5
4	negro	Fiat	8000	17.928.4932-k
5	azul	Ford	15000	18.278.191-3

## Personas

rut	nombre	edad	teléfono
7.654.184-1	Juan	55	876326
15.904.874-5	Diego	32	873492
17.928.492-k	Marcela	28	248273
18.278.191-3	Silvia	25	473492
18.743.921-5	Pablo	23	725897

# Base de datos

## Autos

id	color	marca	precio(\$)	dueño
1	rojo	KIA	8000	7.654.184-1
2	verde	Mercedes	10000	15.904.874-5
3	rojo	Mercedes	20000	18.743.921-5
4	negro	Fiat	12000	17.928.4932-k
5	azul	Ford	15000	18.278.191-3

## Personas

rut	nombre	edad	teléfono
7.654.184-1	Juan	55	876326
15.904.874-5	Diego	32	873492
17.928.4932-k	Marcela	28	248273
18.278.191-3	Silvia	25	473492
18.743.921-5	Pablo	23	725897



# Ahora trabajemos con sqlite

- 1) Descarga del siding la base de datos disponible.
- 2) En consola, posicionate en la carpeta donde tienes el archivo example.db.
- 3) Ejecuta sqlite3
- 4) Ahora abre la base de datos haciendo: **.open example.db**

# Comandos útiles

**.open:** Abre la base de datos en el directorio correspondiente.

**.tables:** Muestra las tablas que contiene la base de datos.

**.schema:** Muestra el esquema de la base de datos, vale decir, tablas con sus atributos, indicando el tipo de cada uno de ellos.

**.quit:** Cierra el programa.

# SQL básico

SELECT  
FROM  
WHERE

atributos  
relaciones (tablas)  
condiciones

# Para ver las tablas

```
SELECT *  
FROM Personas
```



# Consultas

Muestre todos los atributos, de todas las personas que se llamen Juan.

```
SELECT *  
FROM Personas  
WHERE nombre = "Juan"
```

# Consultas

Muestre el nombre y teléfono, de todas las personas que tengan menos de 30 años.

# Consultas

Muestre el nombre y teléfono, de todas las personas que tengan menos de 30 años.

```
SELECT    nombre, teléfono
FROM      Personas
WHERE     edad < 30
```

## Ordenar con ORDER BY

Muestre de manera ascendente, el nombre de la persona ordenadas según su edad.

```
SELECT    nombre
FROM      Personas
ORDER BY  edad
```

Por default el orden es ascendente.

# Joins en SQL

¿Qué pasa si queremos obtener el nombre de los dueños de los autos?

id	color	marca	precio(\$)	dueño
1	rojo	KIA	8000	7.654.184-1
2	verde	Mercedes	10000	15.904.874-5
3	rojo	Mercedes	20000	18.743.921-5
4	negro	Fiat	8000	17.928.4932-k
5	azul	Ford	15000	18.278.191-3

rut	nombre	edad	teléfono
7.654.184-1	Juan	55	876326
15.904.874-5	Diego	32	873492
17.928.4932-k	Marcela	28	248273
18.278.191-3	Silvia	25	473492
18.743.921-5	Pablo	23	725897

# Joins en SQL

¿Qué pasa si queremos obtener el nombre de los dueños de los autos?

SELECT

nombre

FROM

Personas, Autos

WHERE

dueño = rut

# Joins en SQL

¿Qué pasa si queremos obtener el nombre de los dueños de los autos que sean de color rojo?

SELECT  
FROM  
WHERE  
AND

nombre  
Personas, Autos  
dueño = rut  
color = “rojo”

# Agregados

**SUM:** Suma los elementos de una columna.

**MIN:** Mínimo de una columna.

**MAX:** Máximo de una columna.

**AVG:** Promedio de una columna.



# Agregados

Encuentre el valor promedio de los precios de los autos de la marca Ford

SELECT	AVG(precio)
FROM	Autos
WHERE	marca = "Ford"

Los agregados se aplican a un solo atributo.

# Count

Cuenta el número de filas de una tabla

```
SELECT      COUNT(marca)
FROM        Autos
WHERE       precio > 10.000
```

## Group by

¿Qué ocurre si queremos calcular el precio promedio de cada marca, donde el precio es mayor a \$10.000 ?

```
SELECT      AVG(precio)
FROM        Autos
WHERE       precio > 10000
GROUP BY    marca
```

## Group by

- Primero se obtiene una tabla según lo que pusimos en el **WHERE** y **FROM**.

id	color	marca	precio(\$)	dueño
1	rojo	KIA	8000	7.654.184-1
2	verde	Mercedes	12000	15.904.874-5
3	rojo	Mercedes	20000	18.743.921-5
4	negro	Fiat	8000	17.928.4932-k
5	azul	Ford	15000	18.278.191-3
6	verde	KIA	12000	16.593.219-2
7	azul	Ford	20000	15.783.291-6

# Group by

- Primero se obtiene una tabla según lo que pusimos en el **WHERE** y **FROM**.

id	color	marca	precio(\$)	dueño
2	verde	Mercedes	12000	15.904.874-5
3	rojo	Mercedes	20000	18.743.921-5
5	azul	Ford	15000	18.278.191-3
6	verde	KIA	12000	16.593.219-2
7	azul	Ford	20000	15.783.291-6

# Group by

- Se agrupa según **GROUP BY**

id	color	marca	precio(\$)	dueño
2	verde	<u>Mercedes</u>	12000	15.904.874-5
3	rojo	<u>Mercedes</u>	20000	18.743.921-5
5	azul	<u>Ford</u>	15000	18.278.191-3
7	azul	<u>Ford</u>	20000	15.783.291-6
6	verde	<u>KIA</u>	12000	16.593.219-2

## Group by

- Se aplica función de agregado a cada grupo y se muestra según el select.

id	color	marca	precio(\$)	dueño
2	verde	<u>Mercedes</u>	12000	15.904.874-5
3	rojo		20000	18.743.921-5
5	azul	<u>Ford</u>	15000	18.278.191-3
7	azul		20000	15.783.291-6
6	verde	<u>KIA</u>	12000	16.593.219-2

## Group by

- Se aplica función de agregado a cada grupo y se muestra según el select.

marca	AVG(precio)
Mercedes	16000
Ford	17500
KIA	12000



# HAVING

- Podemos aplicar condiciones a la tabla resultante al hacer **GROUP BY**.

```
SELECT      AVG(precio) AS total
FROM        Autos
WHERE       precio > 10000
GROUP BY    marca
HAVING      total > 15000
```

# HAVING

- Podemos aplicar condiciones a la tabla resultante al hacer **GROUP BY**.

marca	total
Mercedes	16000
Ford	17500

# Atributos Duplicados

POKEMONS	
id	integer
name	varchar(255)
height	integer
weight	integer
base experience	integer

MOVES	
id	integer
name	varchar(255)
type_id	integer
power	integer
pp	integer
accuracy	integer

# Atributos Duplicados

POKEMONS	
id	integer
name	varchar(255)
height	integer
weight	integer
base experience	integer

MOVES	
id	integer
name	varchar(255)
type_id	integer
power	integer
pp	integer
accuracy	integer

# Atributos Duplicados

Para evitar este problema:

SELECT	Personas.nombre
FROM	Personas, Autos
WHERE	Autos.dueño = Personas.rut