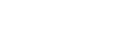


# **Taller de SQL**

202601 - ECON 1306

# CRUD

- C:  Crear tablas, conexiones entre tablas y registros.
- Create  Leer tablas.
- R: Read  Actualizar arquitectura de la BD y actualizar los registros.
- U:  Eliminar tablas y registros.
- Update 
- D: 
- Delete

# Crear servidor en Supabase



# CREATE

# Crear tablas

## Persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15

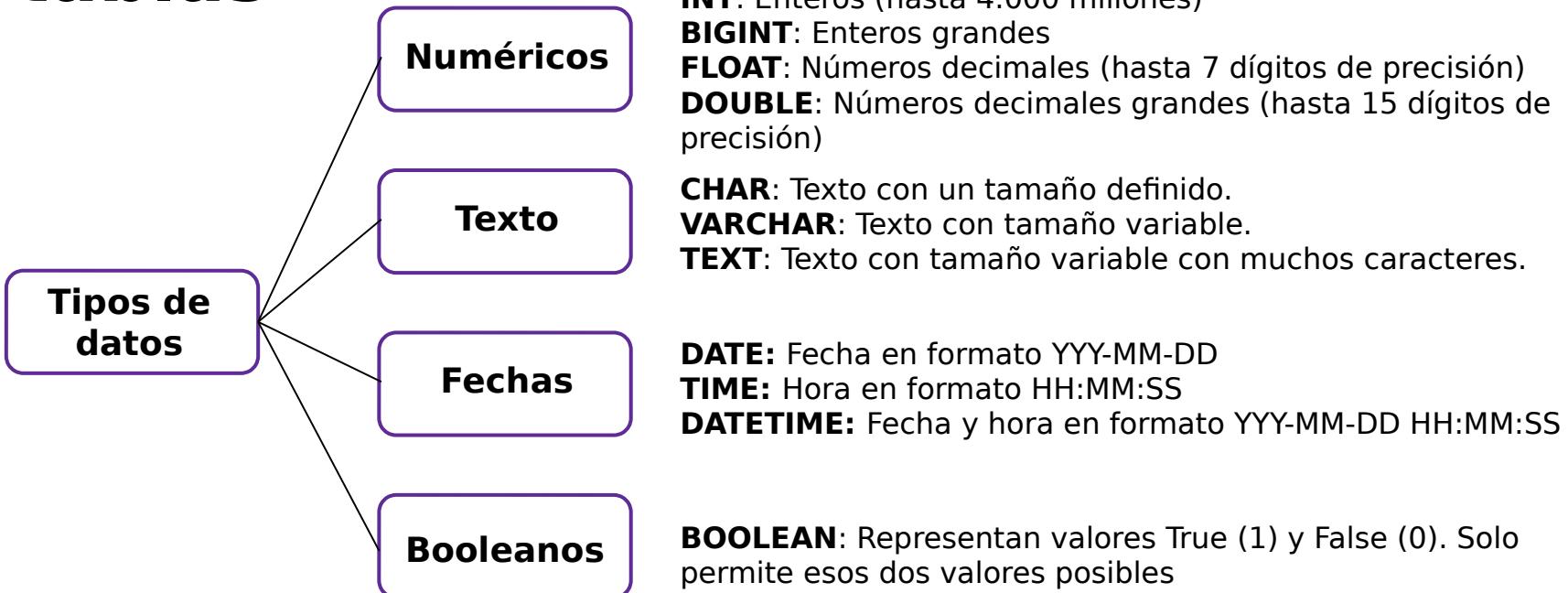
Se debe identificar

### 1. Atributos

- a. Nombre
- b. Tipo de dato
- c. Dominio
- d. ¿Puede estar vacío?

### 2. Identificador único (PK)

# Crear tablas



...

# Crear tablas: Identificar atributos y tipos de datos

persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15

## Atributos:

- id **INT**
- nombre **VARCHAR**
- nombre2 **VARCHAR**
- apellido **VARCHAR**
- sexo **CHAR**
- edad **INT**

# Crear tablas: Identificar dominios

¿Qué valores puede tomar?

¿Esos valores tienen restricciones?

persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15



Todos los **VARCHAR** deben tener una restricción de dominio, que indique el máximo número de caracteres.

Solo puede tomar valor F o M

# Crear tablas: Identificar dominios

persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15

## Atributos:

- id **INT**
- nombre **VARCHAR (20)**
- nombre2 **VARCHAR (20)**
- apellido **VARCHAR (20)**
- sexo **CHAR (1), valores 'F' y 'M'**
- edad **INT**

# Crear tablas: ¿Puede estar vacío?

Hay campos que  
pueden estar vacíos y  
otros no.

Por ejemplo, una  
persona siempre tiene  
que tener un primer  
nombre, pero no  
siempre tiene segundo  
nombre

persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15

## Atributos:

- id **INT**  
**NOT NULL**
- nombre **VARCHAR (20)** **NOT NULL**
- nombre2 **VARCHAR (20)**
- apellido **VARCHAR (20)** **NOT NULL**
- sexo **CHAR (1)**, **valores 'F' y 'M'** **NOT NULL**
- edad **INT**  
**NOT NULL**

# Crear tablas: Identificador único

persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15

**IMPORTANTE:** Toda fila en una tabla debe ser única.

Si tuviéramos dos registros iguales no sabríamos cómo diferenciarlos.



**SOLUCIÓN:** Llave primaria.  
Un atributo (o un conjunto de atributos) que son únicos por cada registro de la tabla.

nombre	nombre2	apellido	sexo	edad
María	Lucía	Sánchez	F	11
María	Lucía	Sánchez	F	11

**Ej.** Número de cédula  
Código de la universidad

# Crear tablas

persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15

## Atributos:

- id **INT** **NOT NULL PK**
- nombre **VARCHAR (20)** **NOT NULL**
- nombre2 **VARCHAR (20)**
- apellido **VARCHAR (20)**. **NOT NULL**
- sexo **CHAR(1), valores 'F' y 'M'** **NOT NULL**
- edad **INT** **NOT NULL**

# Crear tablas

persona

id	nombre	nombre2	apellido	sexo	edad
1	María	Lucía	Sánchez	F	11
2	Juan		Perez	M	15

```
CREATE TABLE nombre_tabla (
    atributo0 tipo,
    atributo1 tipo NOT NULL,
    ...
    atributoN tipo,
    PRIMARY KEY (atributo0),
    CHECK (atributo1 in (val1, val2,
val3))
);
```

```
CREATE TABLE persona (
    id INT,
    nombre VARCHAR(20) NOT
    NULL,
    nombre2 VARCHAR(20),
    apellido VARCHAR(20) NOT
    NULL,
    sexo CHAR(1) NOT NULL,
    edad INT NOT NULL,
    PRIMARY KEY(id),
    CHECK (sexo IN ("F", "M"))
);
```

# Crear tablas

## Tipos de restricciones

### 1. PRIMARY KEY

### 2. FOREIGN KEY

### 3. CHECK

### 4. NOT NULL

### 5. DEFAULT: Valor por defecto que tiene una columna

### 6. AUTOINCREMENT

genero IN (“FEMENINO”, “MASCULINO”, “OTRO”)  
edad > 18, edad  $\geq$  18, edad  $\leq$  18, edad < 18, edad  $\neq$  18,  
edad BETWEEN 1 AND 150  
codigo LIKE “2023%”  Todos los códigos comienzan con 2023

dias\_en\_empresa INT NOT NULL **DEFAULT**  
0

Siempre que se cree un dato de un empleado va a tener número de días en la empresa igual a 0

# Crear tablas Ejercicio

[https://onecompiler.com/  
mysql/](https://onecompiler.com/mysql/)

cedula	nombre	apellido	genero	fecha_nacimiento	email
1234	María	Rodríguez	Femenino	2001-01-01	mrod@gmail.com
2345	Juan	Carmona	Otro	1994-07-09	jcarm@gmail.com
3456	Melissa	Robles	Femenino	2004-05-06	mrobles@gmail.com
6789	Andrés	Aguirre	Masculino	1970-12-05	aaguirre@gmail.com
7890	Daniel	Vanegas	Masculino	2001-02-02	dvanegas@gmail.com

\* No se permiten fechas antes de 1950.

\* Género solo tiene 3 valores posibles.

# Crear tablas

## PK compuestas

### articulos

<b>id_articulo</b>	<b>id_autor</b>	<b>titulo</b>	<b>autor</b>	<b>anio_publicacion</b>
1	11	A relational model of data for large shared data banks	E. F. Codd	1970
2	12	Comparative Study of Multi-query Optimization Techniques using Shared Predicate-based for Big Data	Radhya Sahal	2016
2	13	Comparative Study of Multi-query Optimization Techniques using Shared Predicate-based for Big Data	Mohamed H. Khafagy	2016

¿Llave  
primaria?

**PRIMARY KEY**(id\_articulo,  
id\_autor)

# Crear tablas

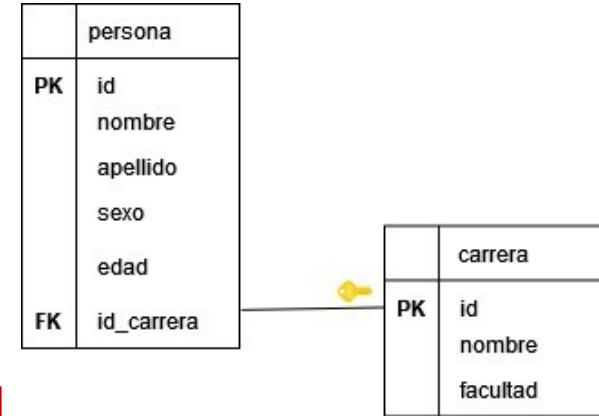
## Conexiones entre tablas

persona

id	nombre	nombre2	apellido	sexo	edad	id_carrera
1	María	Lucía	Sánchez	F	11	11
2	Juan		Perez	M	15	12

carrera

id	nombre	facultad
11	Economía	Economía
12	Matemáticas	Ciencias



Las llaves foráneas crean una restricción sobre las columnas. En el ejemplo, no podríamos tener un **id\_carrera** que no existiera en la tabla **carrera**.

**FOREIGN KEY(id\_carrera) references** carrera(id)

# Crear tablas

## Conexiones entre tablas

```
CREATE TABLE carrera(
```

```
    id INT,  
    nombre VARCHAR(20) NOT  
    facultad VARCHAR(20) NOT  
    NULL,  
    NULL,
```

```
    PRIMARY KEY(id)
```

```
);
```

```
CREATE TABLE persona (
```

```
    id INT,  
    nombre VARCHAR(20) NOT NULL,  
    nombre2 VARCHAR(20),  
    apellido VARCHAR(20) NOT NULL,  
    sexo CHAR(1) NOT NULL,  
    edad INT NOT NULL,  
    id_carrera INT NOT NULL,
```

```
    PRIMARY KEY(id),
```

```
    FOREIGN KEY(id_carrera) REFERENCES
```

```
carrera(id),
```

```
    CHECK (sexo IN ('F', 'M'))
```

```
);
```

	persona
PK	id
	nombre
	apellido
	sexo
	edad
FK	id_carrera

	carrera
PK	id
	nombre
	facultad

# Crear datos

```
INSERT INTO carrera  
VALUES (0, 'Matemáticas', 'Ciencias');
```

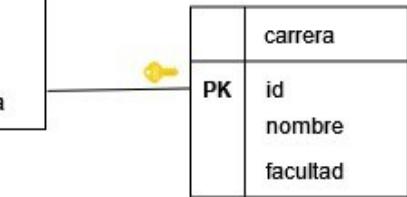
```
INSERT INTO persona  
VALUES (1, 'Melissa', 'Robles', 'F', 25, 0);
```

**Si no tiene todos los valores:**

```
INSERT INTO persona (id, nombre, id_carrera)  
VALUES (2, "Daniel", 0)
```

**persona**

	persona
<b>PK</b>	id
	nombre
	apellido
	sexo
	edad
<b>FK</b>	id_carrera



**carrera**

id	nombre	facultad
0	Matemáticas	Ciencias

id	nombre	apellido	sexo	edad	id_carrera
1	Melissa	Robles	F	25	0
2	Daniel				0

# Crear tablas

## Buenas prácticas

### Nombrado de atributos:

- Los atributos (columnas) deben llamarse en minúscula y para separar palabras se utiliza el guión bajo (\_).
- No utilizar caracteres especiales (puntos, comas, tildes, ñ)
- NO llamar las columnas “select” o “null” o cualquier palabra clave de SQL.

Ej. fecha\_nacimiento, nombre, dia\_hora, anio

### Restricciones:

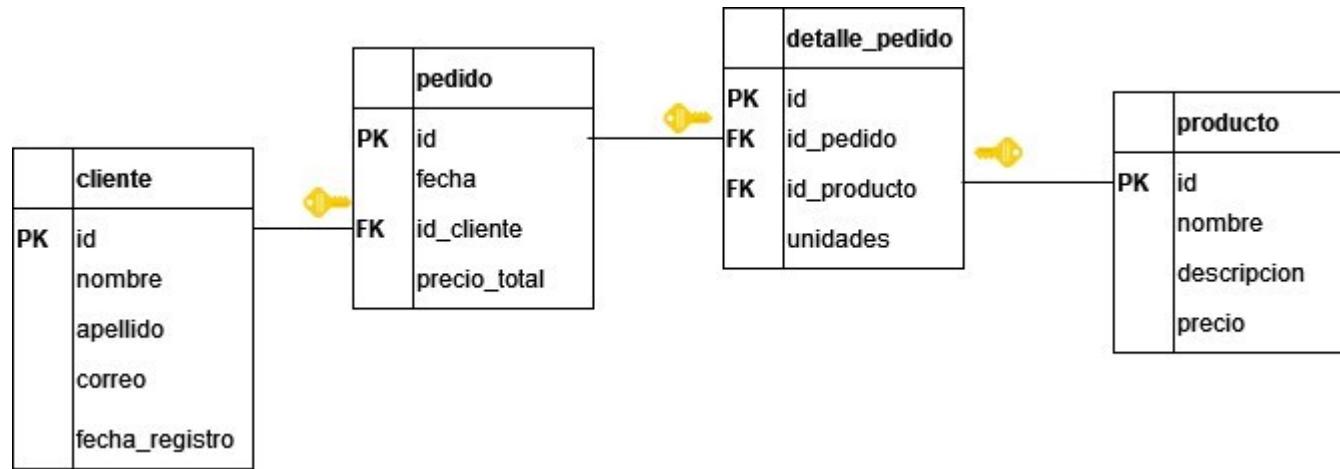
- Usar restricciones para garantizar (en lo posible) la limpieza de los datos y evitar inconsistencias.

### Espaciado/Indentación

# Crear tablas

## Conexiones entre tablas

### Ejercicio



# Crear datos

## Ejercicio

Con la base de datos que crearon en el ejercicio anterior, deben agregar los siguientes datos:

cliente				
id	nombre	apellido	correo	fecha_registro
1	Melissa	Robles	mv.robles@uniandes.edu.co	07/08/2023
2	Daniel	Aguirre	ed.aguirre@uniandes.edu.co	06/08/2023
3	Maria	Perez	mp@gmail.com	05/08/2023
4	Juan	Rodríguez	jr@gmail.com	05/08/2023

pedido			
id	fecha	id_cliente	precio_total
20	08/08/2023	1	14500
21	08/08/2023	2	1500

producto			
id	nombre	descripcion	precio
10	Lapiz	Lapiz	500
11	Cuaderno	Cuaderno 1 materia	7000
12	Borrador	Borrador marca x	1000

detalle_pedido			
id	id_pedido	id_producto	unidades
30	20	11	2
31	20	10	1
32	21	10	1
33	21	12	1

# **Crear datos**

## **Ejercicio**

1. Agregue un nuevo producto “Escuadra” a la base de productos con el id=11. ¿Qué sucede? ¿Qué se debe modificar?
2. Cree un nuevo pedido con la fecha de hoy y un total de 18.000 pesos, para un cliente con id 5. ¿Qué sucede? ¿Qué debe modificar?

# READ

# Leer tablas Queries

**Consulta:** Instrucción o solicitud que se utiliza para recuperar datos de una base de datos.

Retorna una tabla con la información solicitada.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
id	id_encuesta	id_parentesco	sexo	edad	estado_civil	departamento_nacimiento																				
1	1	1	2	35	1	9																				
2	2	1	3	5	1	1																				
3	3	1	3	4	1	1																				
4	4	1	3	3	1	1																				
5	5	1	3	2	-1	1																				
6	6	2	2	25	2	1																				
7	7	2	3	4	1	1																				
8	8	2	3	-1	1	1																				
9	9	2	4	2	1	1																				
10	10	2	4	6	1	1																				
11	11	2	2	4	1	1																				
12	12	2	5	1	2	9																				
13	13	2	5	1	1	1																				
14	14	2	5	2	18	2																				
15	15	3	2	29	2	9																				
16	16	3	4	2	4	1																				
17	17	3	5	1	34	1																				
18	18	4	2	2	36	2																				
19	19	4	3	1	12	1																				
20	20	4	3	1	8	1																				
21	21	4	3	1	1	1																				
22	22	4	4	2	17	1																				
23	23	4	4	2	14	1																				
24	24	4	4	2	9	1																				
25	25	5	3	15	1	99																				
26	26	5	3	1	21	1																				

**SELECT** columna\_1,  
columna\_2  
**FROM** nombre\_tabla

**SELECT** \*  
**FROM** nombre\_tabla

“Obtener la  
información de la  
**columna\_1** y la  
**columna\_2** de la  
tabla  
**nombre\_tabla**”

“Obtener toda la  
información de la  
tabla  
**nombre\_tabla**”

# Leer tablas

## Queries

Consultas más complejas: condiciones.

**SELECT** columna\_1,  
columna\_2  
**FROM** nombre\_tabla  
**WHERE** condicion

Indica cuáles columnas quiero

De qué tabla obtengo la información

Indica cuáles registros quiero

**SELECT \***  
**FROM** persona  
**WHERE** edad  
 $\geq 18$

Seleccióna toda la información de las personas mayores de edad.

# Leer tablas

## Condiciones

**Table 2-1.** Comparison Operators

Operator	Meaning	Examples of True Statement
=	Equals	5=5, 'Junior' = 'Junior'
<	Less than	4<5, 'Ann' < 'Zebedee'
<=	Less than or equal to	4<=5, 5<=5
>	Greater than	5>4, 'Zebedee' > 'Ann'
>=	Greater than or equal to	5>=4, 5>=5
<>	Not equal	5<>4, 'Junior' <> 'Senior'

# Leer tablas

## Nulos

```
SELECT *  
FROM persona  
WHERE nombre2 IS NULL
```

```
SELECT *  
FROM persona  
WHERE nombre2 IS NOT  
NULL
```



Retorna los datos de personas que tienen un valor no nulo en nombre2.

(Personas con segundo nombre)

Los valores nulos son problemáticos

Una parte importante de analizar datos es ver cuántos valores nulos tiene una variable.

# Leer tablas

## Operadores lógicos

```
SELECT nombre, edad  
FROM persona  
WHERE edad > 18 AND  
sexo = "F"
```

```
SELECT nombre, edad  
FROM persona  
WHERE edad > 18 OR  
sexo = "F"
```

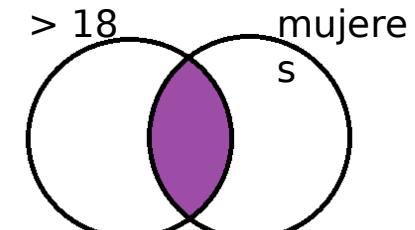
```
SELECT nombre, edad  
FROM persona  
WHERE NOT(edad > 18)
```

**AND**  
**OR**  
**NOT**

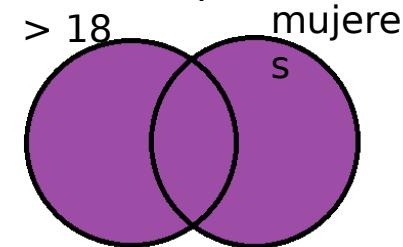
Selecciona el nombre y la edad de las mujeres mayores de edad.

Selecciona el nombre y la edad de las personas que son mujeres o que son mayores de edad.

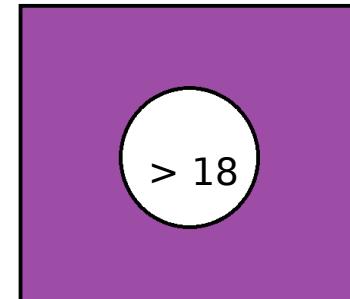
Selecciona el nombre y la edad de las personas que **no** tienen más de 18 años.



edad > 18 **AND** sexo = "F"

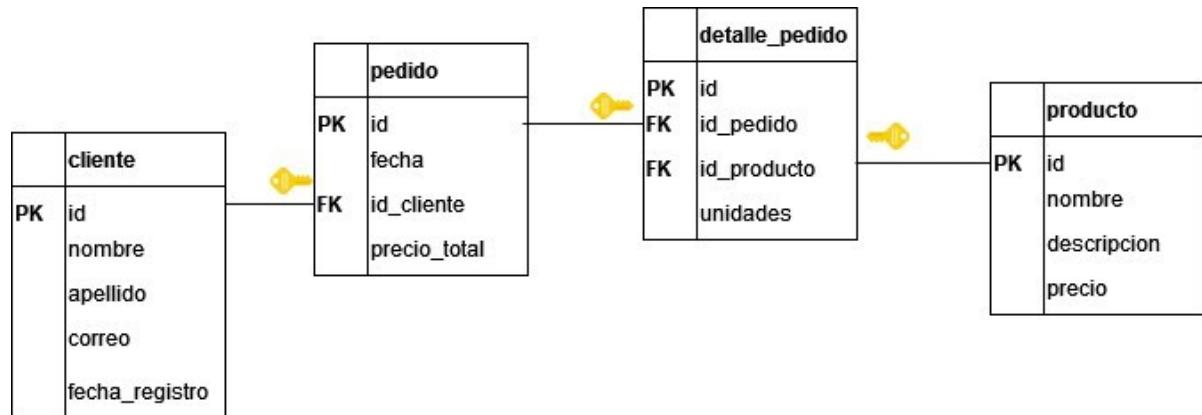


edad > 18 **OR** sexo = "F"



**NOT(edad > 18)**

# Leer tablas Operadores lógicos



1. Listar toda la información de los productos que cuestan más de 1000
2. Listar el precio de los pedidos que se realizaron entre 05/08/2023 y 06/08/2023

```
SELECT *  
FROM producto  
WHERE precio > 1000
```

```
SELECT precio_total  
FROM pedido  
WHERE fecha = "2023-08-05" OR fecha = "2023-08-06"
```

# Ejercicio

<code>id</code>	<code>nombre</code>	<code>edad</code>	<code>salario</code>	<code>departamento</code>
1	Juan	28	50000.00	Ventas
2	Ana	35	60000.00	Ventas
3	Carlos	40	70000.00	TI
4	Elena	25	55000.00	Recursos Humanos
5	Miguel	32	75000.00	TI

1. Seleccionar empleados que trabajan en Ventas y tienen más de 30 años o ganan más de \$55,000
2. Seleccionar empleados que ganan entre \$50,000 y \$70,000 y tienen más de 30 años
3. Seleccionar empleados cuyo nombre comienza con 'A' y trabajan en un departamento que no sea Ventas ni Recursos Humanos

# UPDATE

# Actualizar tablas

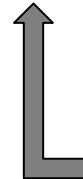
**ALTER TABLE** tabla  
**ADD** columna tipo



Agrego una columna a la tabla

**Ejemplo:** Queremos que la tabla **persona** tenga una nueva columna que sea **estado\_civil**.

**ALTER TABLE** persona  
**ADD** estado\_civil **VARCHAR(20)** **NOT NULL DEFAULT**  
'Soltero'



Crea la columna con un  
valor por Default.

# Actualizar datos

**UPDATE** tabla

**SET** columna 1= valor

**WHERE** condicion

**Ejemplo:** Queremos que la tabla **persona** el sexo en vez de ser “F” y “M”, diga “Femenino” y “Masculino”

**UPDATE** persona

**SET** sexo = “Femenino”

**WHERE** sexo = “F”;

**UPDATE** persona

**SET** sexo = “Masculino”

**WHERE** sexo = “M”;

# Actualizar datos

**UPDATE** tabla

**SET** columna 1= valor

**WHERE** condicion

**Ejemplo:** Digamos que tenemos una columna llamada **semestre** en nuestra tabla **persona**.

Es un nuevo semestre y queremos sumarle 1 al semestre de todos los estudiantes.

**UPDATE** persona

**SET** semestre = semestre + 1

# DELETE

# Eliminar datos

**DELETE FROM** tabla  
**WHERE** condicion



Elimina todos los registros de una tabla que cumplan con la condición indicada.

**Ejemplo:** En la tabla **persona** queremos únicamente la información de mujeres mayores de edad.

**DELETE FROM** persona  
**WHERE** sexo = "M" OR edad <= 18;



Eliminamos a los hombres y a los menores de edad. Así, quedamos únicamente con las mujeres mayores de edad.

# Eliminar tablas

**DROP TABLE** tabla

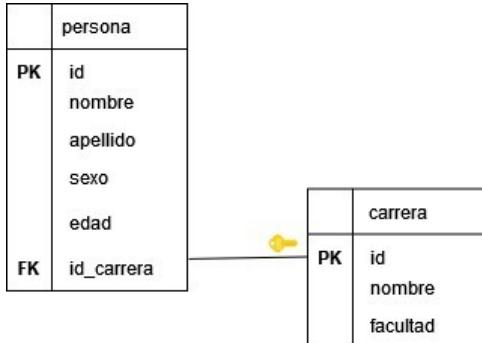


Elimina la tabla completa.

**Ejemplo:** Queremos borrar la tabla persona **DROP TABLE** persona

---

**¿Qué pasa si tenemos llaves foráneas?**



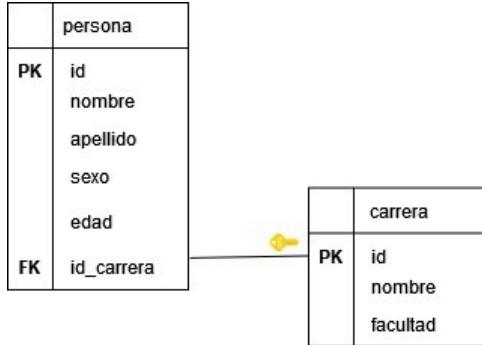
Recordemos que las llaves foráneas son restricciones.

No puedo tener un **id\_carrera** en la tabla **persona** si no existe en la tabla **carrera**.

¿Qué pasa si elimino un registro de la tabla **carrera**?

# Eliminar tablas

¿Qué pasa si tenemos llaves foráneas?



Recordemos que las llaves foráneas son restricciones.

No puedo tener un **id\_carrera** en la tabla **persona** si no existe en la tabla **carrera**.

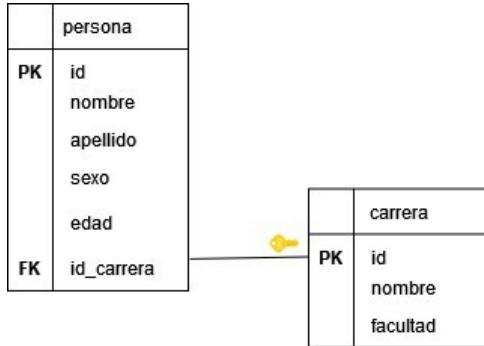
¿Qué pasa si elimino un registro de la tabla carrera?

Cannot delete or update a parent row: a foreign key constraint fails (`sandbox\_db`.`persona`, CONSTRAINT `persona\_ibfk\_1` FOREIGN KEY (`id\_carrera`) REFERENCES `carrera` (`id\_carrera`))

- En general **no se puede eliminar datos que están referenciados por una Foreign key**. Para poder eliminarlos, no debe tener una referencia explícita.
  - Ej. Si quiero borrar los registros de la carrera de Matemáticas, debo primero eliminar todas las personas que están en la carrera de matemáticas (o borrar toda la tabla persona).
- Otra forma: **DROP CASCADE**

# Eliminar tablas

¿Qué pasa si tenemos llaves foráneas?



## DELETE CASCADE

Elimina automáticamente los datos de la tabla que referencia a los registros eliminados.

**DELETE FROM** carrera **WHERE** nombre = "Matemáticas"

Modifica ambas tablas:

- Elimina el registro de Matemáticas de la tabla **carrera**.
- Elimina todos los registros de personas asociadas a la carrera de Matemáticas de la tabla **persona**.

# Eliminar tablas

```
CREATE TABLE persona (
    id INTEGER,
    nombre VARCHAR(20) NOT NULL,
    nombre2 VARCHAR(20) NOT NULL,
    apellido VARCHAR(20) NOT NULL,
    sexo CHAR(1) NOT NULL,
    edad INTEGER NOT NULL,
    id_carrera VARCHAR(20) NOT NULL,
    PRIMARY KEY(id),
    FOREIGN KEY(id_carrera) REFERENCES carrera(id) ON
    DELETE CASCADE,
    CHECK (sexo IN ("F", "M"))
);
```

## Pregunta:

¿Cómo hago esto sin tener que borrar la tabla y volver a crearla?

# Leer tablas

## Ejercicio



1. Explore la tabla `dane_hogares`
2. Escriba consultas para responder las siguientes preguntas:
  - ¿Cuál es el total de menores de 15 años en los departamentos del eje cafetero (Caldas, Risaralda, Quindío)?
  - ¿Cuáles departamentos tienen más de 500.000 hogares?
  - ¿Cuáles departamentos tienen más de 1.000.000 de menores de 15 años?
  - ¿Cuáles departamentos del eje cafetero tienen más de 300.000 hogares?
  - Un departamento se considera grande si tiene más de 1.000.000 de hogares. ¿Cuáles departamentos grandes tienen más de 1.000.000 de menores de 15 años?

# Actualizar

## Ejercicio

Con la base anterior, realizar las siguientes operaciones:

1. Agregar una nueva columna **hogares\_sin\_menores\_15\_anios**.
2. Realice un query que calcule el número de hogares de cada departamento que no tienen menores de 15 años y lo guarde en la columna creada.
3. Modifique la tabla para que el departamento “Archipiélago de San Andrés, Providencia y Santa Catalina” se llame únicamente “**San Andrés**”.