



# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

## RELACIONES – PROFUNDICEMOS

Estimados aprendices, ya sabemos insertar y consultar registros de nuestras tablas, ahora nos corresponde actualizar y eliminar todos aquellos registros que previamente hemos trabajado.

Primero que todo, por favor ingresen a este link y actualicen sus datos por favor

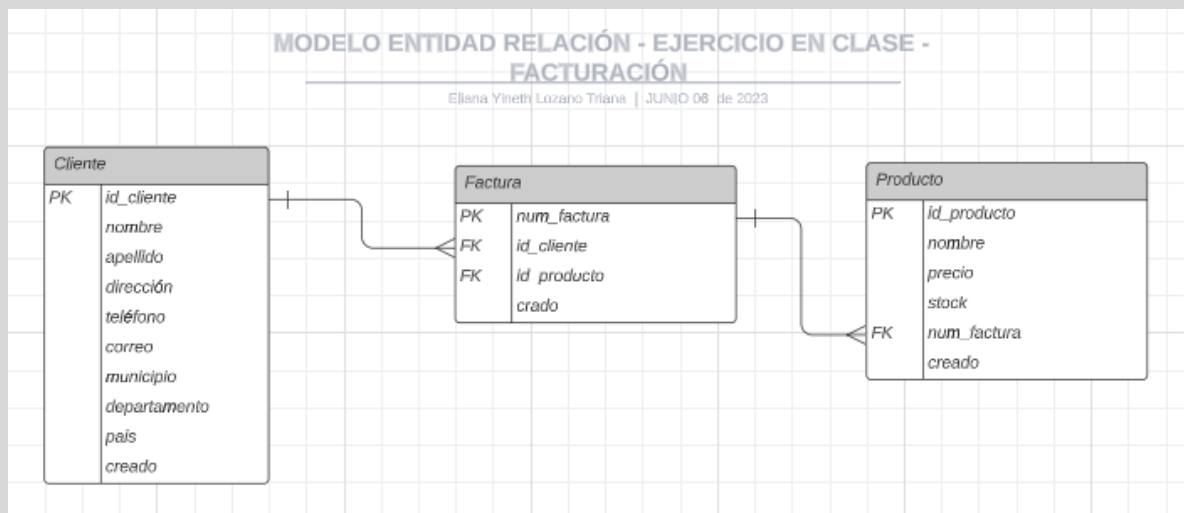
<https://docs.google.com/spreadsheets/d/10xBBKTDBVOXlx0ndr9ymsYKs7QU0ahTy/edit?usp=sharing&oid=106229533260816064003&rtpof=true&sd=true>

Organicemos la base en la que vamos a trabajar.

----- Base de datos de demostración – trabajaremos con la de la guía pasada ----

**CARGAR EL ARCHIVO .SQL Y EL WORD CON LOS PANTALLAZOS AL REPO COMO RELACION\_SQL EN UN ARCHIVO ZIP**

- **MODELO – MAQUETADO.**



- **CÓDIGO**

```
CREATE DATABASE FACTURACIÓN;
USE FACTURACIÓN;

CREATE TABLE cliente(
    id_cliente VARCHAR (30) UNIQUE PRIMARY KEY,
    nombre VARCHAR (25) UNIQUE NOT NULL,
    apellido VARCHAR (25) UNIQUE NOT NULL,
    direccion VARCHAR(20) NOT NULL,
    telefono VARCHAR(20) NOT NULL,
    correo VARCHAR(50) UNIQUE NOT NULL,
    municipio VARCHAR(20),
    departamento VARCHAR(20),
```



# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

```
    pais VARCHAR (20) CHECK (pais='Colombia'), -- CHECK permite limitar a
una respuesta
    creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

CREATE TABLE factura(
    num_factura VARCHAR (20) UNIQUE PRIMARY KEY,
    id_cliente VARCHAR (30),
    id_producto VARCHAR (30),
    creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (id_cliente) REFERENCES cliente(id_cliente),
);

CREATE TABLE productos(
    id_producto VARCHAR (30) UNIQUE PRIMARY KEY,
    nombre VARCHAR (25) NOT NULL,
    precio INT (25) NOT NULL,
    stock INT (25) NOT NULL,
    num_factura VARCHAR (20),
    creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (num_factura) REFERENCES factura(num_factura)
);

ALTER TABLE factura ADD FOREIGN KEY(id_producto) REFERENCES
productos(id_producto);
```

- ESTRUCTURA

```
mysql> desc factura;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default      | Extra      |
+-----+-----+-----+-----+-----+-----+
| num_factura | varchar(20) | NO   | PRI | NULL         |            |
| id_cliente  | varchar(30) | YES  | MUL | NULL         |            |
| id_producto | varchar(30) | YES  | MUL | NULL         |            |
| creado      | timestamp  | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> desc cliente;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default      | Extra      |
+-----+-----+-----+-----+-----+-----+
| id_cliente  | varchar(30) | NO   | PRI | NULL         |            |
| nombre      | varchar(25) | NO   | UNI | NULL         |            |
| apellido    | varchar(25) | NO   | UNI | NULL         |            |
| direccion   | varchar(20) | NO   |     | NULL         |            |
| telefono    | varchar(20) | NO   |     | NULL         |            |
| correo      | varchar(50) | NO   | UNI | NULL         |            |
| municipio   | varchar(20) | YES  |     | NULL         |            |
| departamento | varchar(20) | YES  |     | NULL         |            |
| pais        | varchar(20) | YES  |     | NULL         |            |
| creado      | timestamp  | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

mysql> desc productos;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default      | Extra      |
+-----+-----+-----+-----+-----+-----+
| id_producto | varchar(30) | NO   | PRI | NULL         |            |
| nombre      | varchar(25) | NO   |     | NULL         |            |
| precio      | int        | NO   |     | NULL         |            |
| stock       | int        | NO   |     | NULL         |            |
| num_factura | varchar(20) | YES  | MUL | NULL         |            |
| creado      | timestamp  | YES  |     | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

----- Base de datos de demostración



# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

Realizamos inserción de registros:

```
-- CLIENTE
```

```
INSERT INTO cliente (id_cliente, nombre, apellido, direccion, telefono, correo, municipio, departamento, pais)
VALUES ('900876543-0', 'JUAN', 'GOMEZ', 'CALLE 13', '8608978', 'USER1@USER1.COM', 'QUEBRADAS', 'RISARALDA', 'COLOMBIA');
INSERT INTO cliente (id_cliente, nombre, apellido, direccion, telefono, correo, municipio, departamento, pais)
VALUES ('800876543-0', 'CAMILO', 'VELEZ', 'CALLE 11', '87098765', 'USER2@USER2.COM', 'ORITO', 'PUTUMAYO', 'COLOMBIA');
INSERT INTO cliente (id_cliente, nombre, apellido, direccion, telefono, correo, municipio, departamento, pais)
VALUES ('1077654345', 'JUANA', 'DE ARCO', 'CALLE 10', '861876567', 'USER3@USER3.COM', 'BOGOTÁ', 'CUNDINAMARCA', 'COLOMBIA');
```

```
-- FACTURA
```

```
INSERT INTO factura (num_factura, id_cliente)
VALUES ('A-001', '900876543-0');
INSERT INTO factura (num_factura, id_cliente)
VALUES ('A-002', '1077654345');
INSERT INTO factura (num_factura, id_cliente)
VALUES ('A-003', '800876543-0');
```

```
-- No se insertan valores para la columna id_producto, debido a que, al ser llave foránea, estos valores deben primero crearse en la tabla original de productos.
```

```
--PRODUCTOS
```

```
INSERT INTO productos (id_producto, nombre, precio, stock, num_factura)
VALUES ('PRO-001', 'PORTÁTIL', '2800000', '70', 'A-001');
INSERT INTO productos (id_producto, nombre, precio, stock, num_factura)
VALUES ('PRO-002', 'TECLADO', '30000', '30', 'A-002');
INSERT INTO productos (id_producto, nombre, precio, stock, num_factura)
VALUES ('PRO-003', 'MOUSE', '15000', '20', 'A-003');
```

```
-- Ahora vamos a actualizar la columna id_producto de la tabla factura.
```

```
UPDATE factura SET id_producto = 'PRO-001' WHERE num_factura = 'A-001';
UPDATE factura SET id_producto = 'PRO-002' WHERE num_factura = 'A-002';
UPDATE factura SET id_producto = 'PRO-003' WHERE num_factura = 'A-003';
```



# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

Ahora tenemos lo siguiente:

```
mysql> SELECT * FROM cliente;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id_cliente | nombre | apellido | direccion | telefono | correo | municipio | departamento | pais | creado |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1077654345 | JUANA | DE ARCO | CALLE 10 | 861876567 | USER3@USER3.COM | BOGOTÁ | CUNDINAMARCA | COLOMBIA | 2023-06-20 20:20:39 |
| 800876543-0 | CAMILO | VELEZ | CALLE 11 | 87098765 | USER2@USER2.COM | ORITO | PUTUMAYO | COLOMBIA | 2023-06-20 20:20:36 |
| 900876543-0 | JUAN | GOMEZ | CALLE 13 | 8608978 | USER1@USER1.COM | QUEBRADAS | RISARALDA | COLOMBIA | 2023-06-20 20:03:24 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

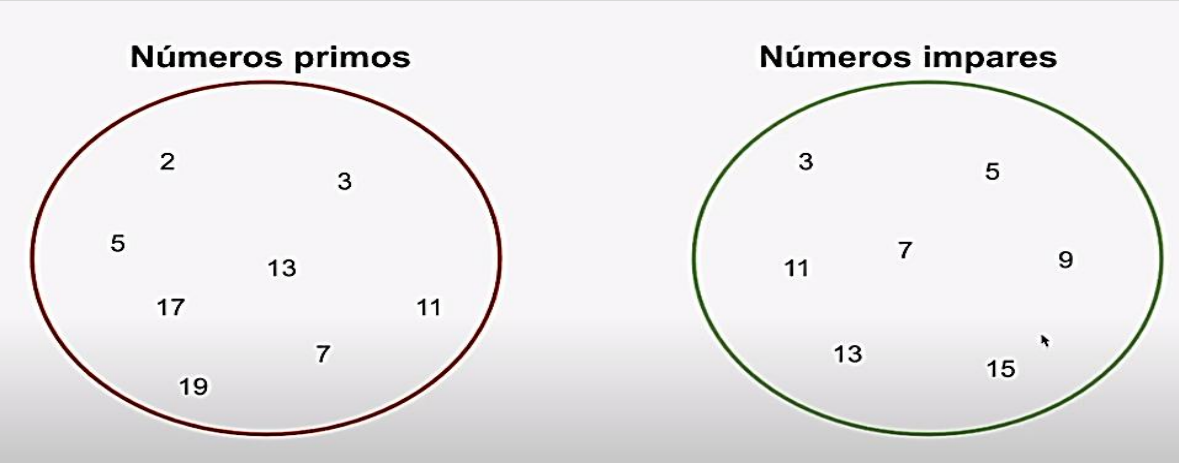
mysql> SELECT * FROM factura;
+-----+-----+-----+-----+
| num_factura | id_cliente | id_producto | creado |
+-----+-----+-----+-----+
| A-001 | 900876543-0 | PRO-001 | 2023-06-20 21:32:41 |
| A-002 | 1077654345 | PRO-002 | 2023-06-20 21:32:41 |
| A-003 | 800876543-0 | PRO-003 | 2023-06-20 21:32:43 |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT * FROM productos;
+-----+-----+-----+-----+-----+-----+
| id_producto | nombre | precio | stock | num_factura | creado |
+-----+-----+-----+-----+-----+-----+
| PRO-001 | PORTÁTIL | 2800000 | 70 | A-001 | 2023-06-20 21:45:46 |
| PRO-002 | TECLADO | 30000 | 30 | A-002 | 2023-06-20 21:45:46 |
| PRO-003 | MOUSE | 15000 | 20 | A-003 | 2023-06-20 21:45:47 |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

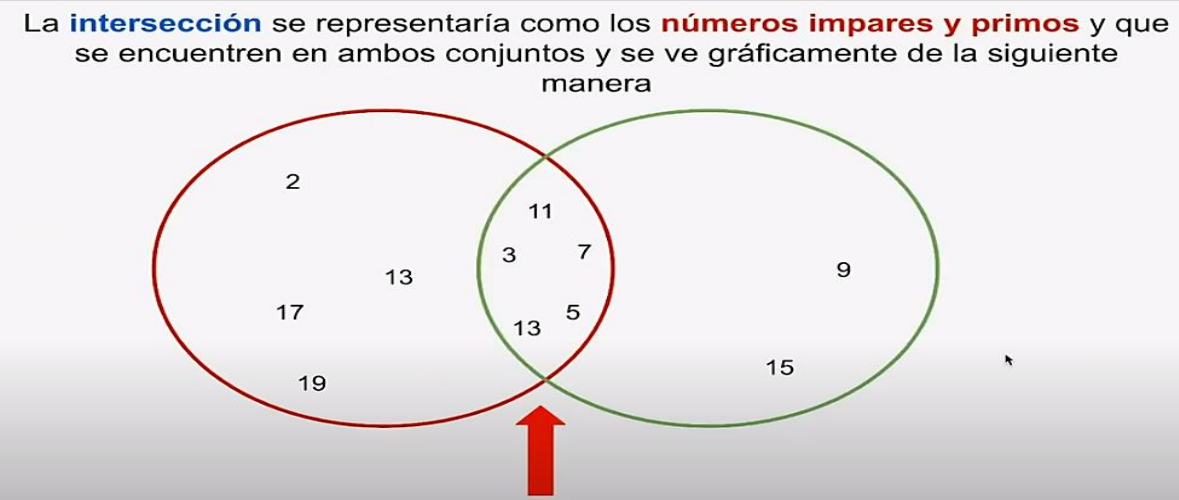
\*\*\* TODO LO ANTERIOR ES EL DESARROLLO DE LA GUÍA DE LA CLASE ANTERIOR, POR LO QUE TODO ESTO ES UN REPASO HASTA AQUÍ \*\*\*

## TEORIA DE CONJUNTOS

Ejemplos usando diagramas de BEN



### CASO 1.



SET DE DATOS RESULTANTE: [3, 5, 7, 11, 13]



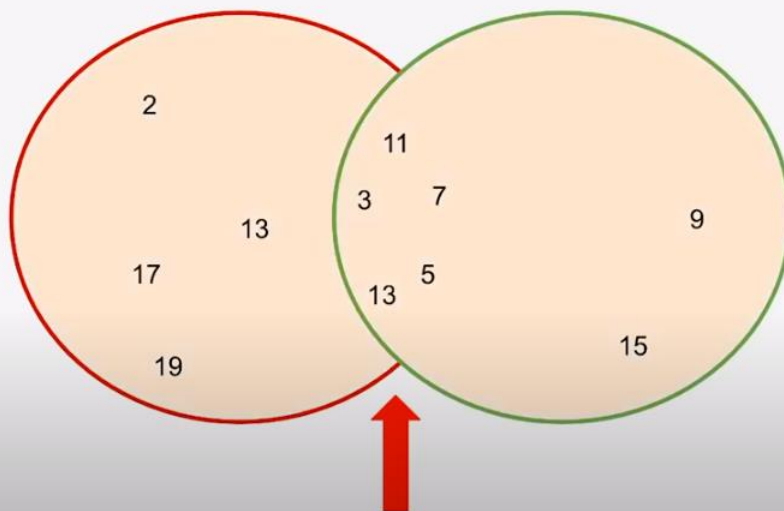
# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

## CASO 2.

La **unión** se representaría como los **todo los números de ambos conjuntos** y se ve gráficamente de la siguiente manera

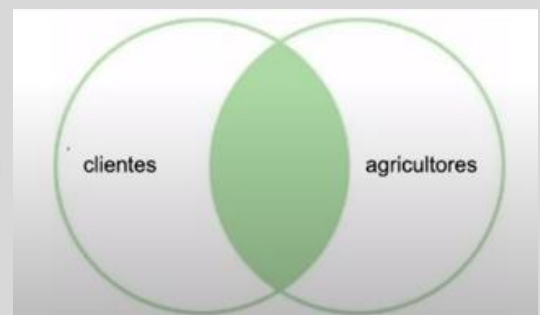


**SET DE DATOS RESULTANTE: [2, 3, 5, 7, 9, 11, 13, 15, 17, 19]**

Lo anterior nos explica cómo funcionan los JOIN, es decir la unión de tablas, de una manera visual, antes de pasar a SQL.

## INNER JOIN

clientes		agricultores	
	name text		name text
1	Alberto	1	Alejandro
2	Alejandra	2	Ana
3	Ana	3	Andrea
4	Andrea	4	Ari
5	Antonio	5	Bernardo
6	Cristina	6	Carlos
7	Diana	7	Clara
8	Diego	8	Cristobal
9	Elena	9	Daniel
10	Erica	10	David
11	Francisco	11	Diego



**INNER JOIN:** Devuelve todas las filas cuando hay al menos una coincidencia en ambas tablas.





# Bases de datos

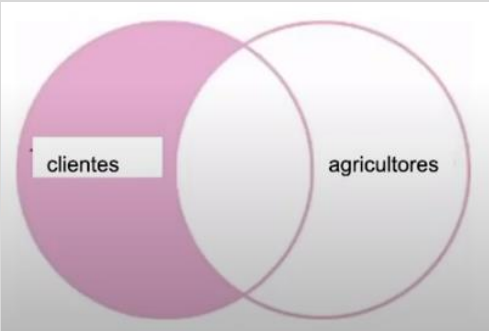
ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

## LEFT JOIN

Resultado

	cliente text	agricultor text
1	Alberto	[null]
2	Alejandra	[null]
3	Ana	Ana
4	Andrea	Andrea
5	Antonio	[null]
6	Cristina	[null]
7	Diana	[null]
8	Diego	Diego
9	Elena	[null]
10	Erica	[null]

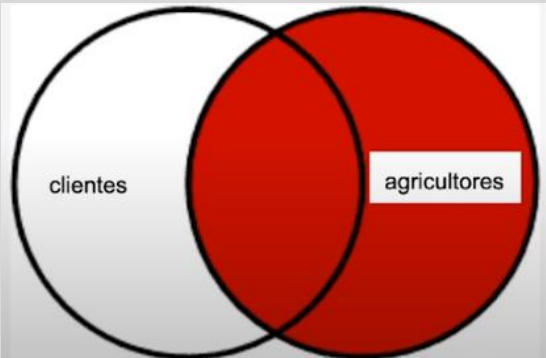


**LEFT JOIN:** Devuelve todas las filas de la tabla de la izquierda, y las filas coincidentes de la tabla de la derecha.

## RIGHT JOIN

Resultado

	cliente text	agricultor text
1	Ana	Ana
2	Andrea	Andrea
3	Diego	Diego
4	Jaime	Jaime
5	Jose	Jose
6	Juan	Juan
7	Maria	Maria
8	Marina	Marina
9	[null]	Clara
10	[null]	Cristobal

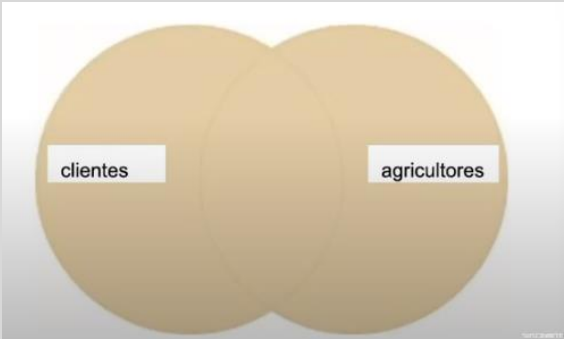


**RIGHT JOIN:** Devuelve todas las filas de la tabla de la derecha, y las filas coincidentes de la tabla de la izquierda.

## FULL OUTER JOIN

Resultado

	cliente text	agricultor text
1	Alberto	[null]
2	Alejandra	[null]
3	Ana	Ana
4	Andrea	Andrea
5	Antonio	[null]
6	Cristina	[null]
7	Diana	[null]
8	Diego	Diego
9	Elena	[null]
10	Erica	[null]
11	Francisco	[null]
...		
30	Rosario	[null]
31	Santiago	[null]
32	Sebastian	[null]
33	[null]	David
34	[null]	Alejandro
35	[null]	Ari
36	[null]	Tito
37	[null]	Pedro





# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

**OUTER JOIN:** Devuelve todas las filas de las dos tablas, la izquierda y la derecha. También se llama FULL OUTER JOIN.

Con las variantes **INNER, LEFT, RIGHT** y **FULL** de consultas multi-tabla somos capaces de obtener registros relacionados y los registros relacionados + los registros no relacionados en uno de los dos lados o en ambos.

### Ejercicio 1.

Obtener todos los clientes que tengan por lo menos una factura.

**Estructura:**

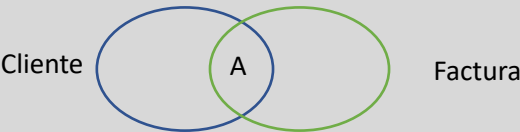
**SELECT**

\*

**FROM** <Nombre de la tabla de origen de datos>

**INNER JOIN** <Nombre de la tabla a unir> **ON** <tabla origen.campo en común>  
=<tabla a unir.campo en común >;

Teoría de conjuntos Intersección = A



```
SELECT
-> *
-> FROM cliente
-> INNER JOIN factura ON cliente.id_cliente = factura.id_cliente;
```

```
mysql> SELECT
-> *
-> FROM cliente
-> INNER JOIN factura ON cliente.id_cliente = factura.id_cliente;
```

id_cliente	nombre	apellido	direccion	telefono	correo	municipio	departamento	pais	creado	num_factura	id_cliente	id_producto	creado
1077654345	JUANA	DE ARCO	CALLE 10	861876567	USER3@USER3.COM	BOGOTÁ	CUNDINAMARCA	COLOMBIA	2023-07-17 16:36:27	A-002	1077654345	PRO-002	2023-07-17 16:36:27
800876543-0	CARILLO	VELEZ	CALLE 11	87098765	USER2@USER2.COM	ORITO	PUTUMAYO	COLOMBIA	2023-07-17 16:36:11	A-003	800876543-0	PRO-003	2023-07-17 16:36:11
900876543-0	JUAN	GOMEZ	CALLE 13	8608978	USER1@USER1.COM	QUEBRADAS	RISARALDA	COLOMBIA	2023-07-17 16:36:11	A-001	900876543-0	PRO-001	2023-07-17 16:36:11

3 rows in set (0.00 sec)

### Ejercicio 2.

Obtener todos los clientes con sus facturas, mostrando también los clientes que no tienen factura. (teoría de conjuntos, lado izquierdo)

**Estructura:**

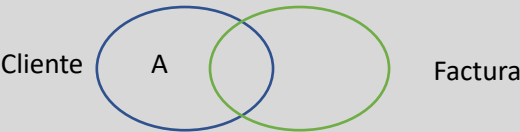
**SELECT**

\*

**FROM** <Nombre de la tabla de origen de datos>

**LEFT JOIN** <Nombre de la tabla a unir> **ON** <tabla origen.campo en común>  
=<tabla a unir.campo en común >;

Teoría de conjuntos lado izquierdo = A





# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

a) Obtener todos los clientes con sus facturas:

```
SELECT
*
FROM cliente
LEFT JOIN factura ON cliente.id_cliente = factura.id_cliente;
```

```
mysql> SELECT
-> *
-> FROM cliente
-> LEFT JOIN factura ON cliente.id_cliente = factura.id_cliente;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
id_cliente | nombre | apellido | direccion | telefono | correo | municipio | departamento | pais | creado | num_factura | id_cliente | id_producto | creado |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1077654345 | JUANA | DE ARCO | CALLE 10 | 861876567 | USER3@USER3.COM | BOGOTA | CUNDINAMARCA | COLOMBIA | 2023-07-17 16:36:27 | A-002 | 1077654345 | PRO-002 | 2023-07-17 16:36:39 |
800876543-0 | CAMILO | VELEZ | CALLE 11 | 87698765 | USER2@USER2.COM | ORITO | PUTUMAYO | COLOMBIA | 2023-07-17 16:36:11 | A-003 | 800876543-0 | PRO-003 | 2023-07-17 16:36:40 |
900876543-0 | JUAN | GOMEZ | CALLE 13 | 8690978 | USER1@USER1.COM | QUEBRADAS | RISARALDA | COLOMBIA | 2023-07-17 16:36:11 | A-001 | 900876543-0 | PRO-001 | 2023-07-17 16:36:39 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

b) Mostrando también los clientes que no tienen factura.

```
SELECT
*
FROM cliente
LEFT JOIN factura ON cliente.id_cliente = factura.id_cliente WHERE
factura.id_cliente IS NULL;
```

```
mysql> SELECT
-> *
-> FROM cliente
-> LEFT JOIN factura ON cliente.id_cliente = factura.id_cliente WHERE factura.id_cliente IS NULL;
Empty set (0.00 sec)
```

Como todos los clientes tienen factura, no nos retorna datos.

### Ejercicio 3.

Obtener todas las facturas de clientes, mostrando también las facturas que no tienen cliente asociado. (teoría de conjuntos, lado derecho)

Estructura:

```
SELECT
*
FROM <Nombre de la tabla de origen de datos>
RIGHT JOIN <Nombre de la tabla a unir> ON <tabla origen.campo en común>
=<tabla a unir.campo en común >;
```

Teoría de conjuntos lado derecho = A



a) Obtener todas las facturas de clientes.

```
SELECT
*
FROM cliente
RIGHT JOIN factura ON cliente.id_cliente = factura.id_cliente;
```





# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

```
mysql> SELECT
-> *
-> FROM cliente
-> RIGHT JOIN factura ON cliente.id_cliente = factura.id_cliente;
```

id_cliente	nombre	apellido	direccion	telefono	correo	municipio	departamento	pais	creado	num_factura	id_cliente	id_producto	creado
900876543-0	JUAN	GOMEZ	CALLE 13	85089778	USER1@USER1.COM	QUEBRADAS	RISARALDA	COLOMBIA	2023-07-17 16:36:11	A-001	900876543-0	PRO-001	2023-07-17 16:36:39
1077654345	JUANA	DE ARCO	CALLE 10	851876557	USER3@USER3.COM	BOGOTÁ	CUNDINAMARCA	COLOMBIA	2023-07-17 16:36:27	A-002	1077654345	PRO-002	2023-07-17 16:36:39
800876543-0	CAMILLO	VELEZ	CALLE 11	87098765	USER2@USER2.COM	ORITO	PUTUMAYO	COLOMBIA	2023-07-17 16:36:11	A-003	800876543-0	PRO-003	2023-07-17 16:36:40

3 rows in set (0.00 sec)

b) Obtener las facturas que no tienen cliente asociado.

```
SELECT
*
FROM cliente
RIGHT JOIN factura ON cliente.id_cliente = factura.id_cliente WHERE
factura.id_cliente IS NULL;
```

```
mysql> SELECT
-> *
-> FROM cliente
-> RIGHT JOIN factura ON cliente.id_cliente = factura.id_cliente WHERE factura.id_cliente IS NULL;
```

Empty set (0.00 sec)

La sintaxis de OUTER JOIN o FULL OUTER JOIN no existen en MySQL, pero se puede conseguir el mismo resultado de diferentes formas, esta es una:

```
SELECT
*
FROM cliente
LEFT JOIN factura ON cliente.id_cliente = factura.id_cliente;

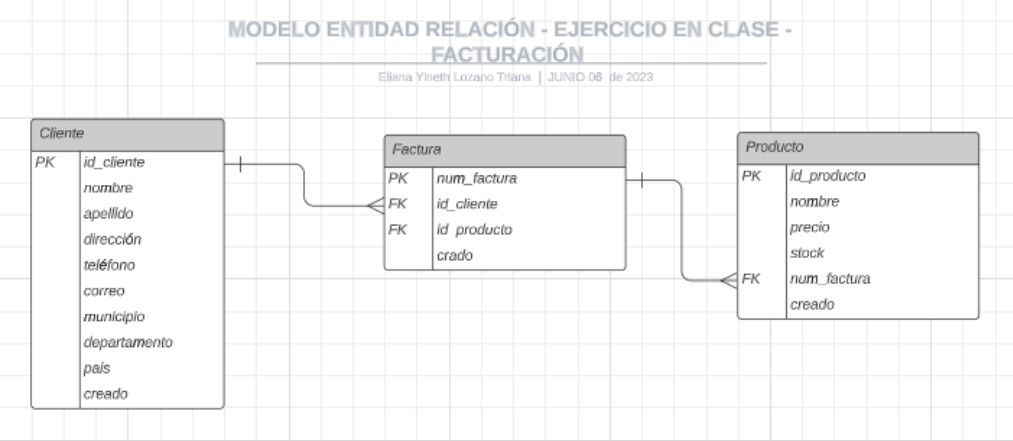
UNION

SELECT
*
FROM clientes
RIGHT JOIN factura ON cliente.id_cliente = factura.id_cliente;
```

## TALLER

1. Maquete un modelo entidad relación con por lo menos 4 tablas, declare llaves primarias y foráneas.

Ejemplo:



2. Desarrolle el modelo en la terminal.
3. Realice la inserción de por lo menos 15 registros, en cada una de las tablas.



# Bases de datos

ELIANA YINETH  
LOZANO TRIANA

[eylozano@sena.edu.co](mailto:eylozano@sena.edu.co)

4. Aplique teoría de conjuntos: Inner, Left y Right.
5. Cargue el archivo .sql y los pantallazos en Word o PDF.
6. Cargue a la plataforma en la evidencia correspondiente en comprimido, de igual manera cárguela en GitHub