

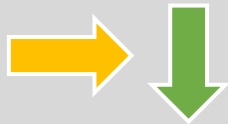
## CRUD 2 PARTE

### SELECT (CONSULTAS PERSONALIZADAS)

La instrucción **SELECT** se utiliza para seleccionar datos de una base de datos.

Los datos devueltos se almacenan en una tabla de resultados, denominada conjunto de resultados.

La instrucción **SELECT \* FROM**, se utiliza para traer todos los registros insertados en una tabla.



**SELECT** *column1, column2, ...*

**FROM** *table\_name*;

Aquí, *columna1, columna2, ...* son los nombres de campo de la tabla de la que desea seleccionar datos.

----- Base de datos de demostración

**POR FAVOR CREAR UNA BASE CON LAS SIGUIENTES CARACTERÍSTICAS PARA QUE REALICEN CADA UNO DE LOS PASOS Y CASOS DE ESTA GUÍA, CARGAR EL ARCHIVO .SQL Y EL WORD CON LOS PANTALLAZOS AL REPO COMO CRUD2\_SQL EN UN ARCHIVO ZIP**

1. **CREATE DATABASE** sena;
2. **USE** sena;
3. **CREATE TABLE** aprendices(  
    id INT (20) UNIQUE PRIMARY KEY,  
    nombre\_apellido VARCHAR (50) UNIQUE NOT NULL,  
    correo VARCHAR(50) UNIQUE NOT NULL,  
    edad INT UNSIGNED NOT NULL,  
    direccion VARCHAR(20) NOT NULL,  
    ciudad VARCHAR(20) NOT NULL,  
    estado ENUM('Activo', 'Inactivo') DEFAULT 'Inactivo',  
    creado TIMESTAMP DEFAULT CURRENT\_TIMESTAMP  
);
4. Realizar inserción de 15 registros.  
    **INSERT INTO** aprendices (id, nombre\_apellido, correo, edad,  
    direccion, ciudad, estado)  
    **VALUES ...**

----- Base de datos de demostración

```
MySQL 8.0 Command Line Client
mysql> CREATE TABLE aprendices(
-> id INT (20) UNIQUE PRIMARY KEY,
-> nombre_apellido VARCHAR (50) UNIQUE NOT NULL,
-> correo VARCHAR(50) UNIQUE NOT NULL,
-> edad INT UNSIGNED NOT NULL,
-> direccion VARCHAR(20) NOT NULL,
-> ciudad VARCHAR(20) NOT NULL,
-> estado ENUM('Activo', 'Inactivo') DEFAULT 'Inactivo',
-> creado TIMESTAMP DEFAULT CURRENT_TIMESTAMP
-> );
Query OK, 0 rows affected, 1 warning (0.05 sec)

mysql> INSERT INTO aprendices (id, nombre_apellido, correo, edad, direccion, ciudad, estado)
-> VALUES (1075861807, 'Eliana Lozano', 'eylozano@sena.edu.co', 31, 'Calle 15', 'Garzón', 'Activo');
Query OK, 1 row affected (0.03 sec)

mysql> INSERT INTO aprendices (id, nombre_apellido, correo, edad, direccion, ciudad, estado)
-> VALUES (1075678987, 'Juan Pérez', 'jp@hotmail.com', 20, 'Calle 13', 'Cali', 'Inactivo');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO aprendices (id, nombre_apellido, correo, edad, direccion, ciudad, estado)
-> VALUES (12187654, 'Maria Pepita', 'mariapepita1@gmail.com', 24, 'Calle 23', 'Barranquilla', 'Inactivo');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO aprendices (id, nombre_apellido, correo, edad, direccion, ciudad, estado)
-> VALUES (5678345, 'Ramón Trujillo', 'donramon@hotmail.com', 20, 'Calle del ocho', 'Jamundí', 'Activo');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO aprendices (id, nombre_apellido, correo, edad, direccion, ciudad, estado)
-> VALUES (1075678987, 'Juan Pérez', 'jp@hotmail.com', 20, 'Calle 13', 'Cali', 'Inactivo');
```

CASO 1

Realicemos una consulta para visualizar los registros de la columna ID de la tabla aprendices.

**SELECT** id **FROM** aprendices;

```
mysql> SELECT id FROM aprendices;
+-----+
| id    |
+-----+
| 5678345 |
| 12187654 |
| 1075678987 |
| 1075861807 |
+-----+
4 rows in set (0.00 sec)
```

CASO 2.

Realicemos una consulta para visualizar los registros de las columnas: nombre\_apellidos y edad de la tabla aprendices.

**SELECT** nombre\_apellido, edad **FROM** aprendices;

```
mysql> SELECT nombre_apellido, edad FROM aprendices;
+-----+-----+
| nombre_apellido | edad |
+-----+-----+
| Ramón Trujillo  | 20   |
| Maria Pepita    | 24   |
| Juan Pérez      | 20   |
| Eliana Lozano   | 31   |
+-----+-----+
4 rows in set (0.00 sec)
```

CASO 3.

Realicemos una consulta para visualizar los registros de las columnas: id, nombre\_apellido y estado

**SELECT** nombre\_apellido, edad, estado **FROM** aprendices;

```
mysql> SELECT nombre_apellido, edad, estado FROM aprendices;
+-----+-----+-----+
| nombre_apellido | edad | estado |
+-----+-----+-----+
| Ramón Trujillo  | 20   | Activo  |
| María Pepita    | 24   | Inactivo|
| Juan Pérez      | 20   | Inactivo|
| Eliana Lozano   | 31   | Activo  |
+-----+-----+-----+
4 rows in set (0.00 sec)
```

--- ¿Qué pasa si debemos recuperar una cantidad extensa de datos y nos es difícil la lectura mediante la tabla de resultados, como se indicó al inicio de la guía?

Existen las denominadas cartas, que son otra presentación de estos datos para facilitar la lectura de los mismos, la sintaxis es la siguiente:

CASO 1.

**SELECT** id **FROM** aprendices\G;

```
mysql> SELECT id FROM aprendices\G;
***** 1. row *****
id: 5678345
***** 2. row *****
id: 12187654
***** 3. row *****
id: 1075678987
***** 4. row *****
id: 1075861807
4 rows in set (0.00 sec)
```

CASO 2.

**SELECT** nombre\_apellido, edad, estado **FROM** aprendices\G;

```
mysql> SELECT nombre_apellido, edad, estado FROM aprendices\G;
***** 1. row *****
nombre_apellido: Ramón Trujillo
      edad: 20
      estado: Activo
***** 2. row *****
nombre_apellido: María Pepita
      edad: 24
      estado: Inactivo
***** 3. row *****
nombre_apellido: Juan Pérez
      edad: 20
      estado: Inactivo
***** 4. row *****
nombre_apellido: Eliana Lozano
      edad: 31
      estado: Activo
4 rows in set (0.00 sec)
```

### CASO 3.

```
SELECT * FROM aprendices\G;
```

```
MySQL 8.0 Command Line Client

mysql> SELECT * FROM aprendices\G;
***** 1. row *****
      id: 5678345
nombre_apellido: Ramón Trujillo
      correo: donramon@hotmail.com
      edad: 20
  direccion: Calle del ocho
      ciudad: Jamundí
      estado: Activo
      creado: 2023-06-05 07:31:22
***** 2. row *****
      id: 12187654
nombre_apellido: Maria Pepita
      correo: mariapepita1@gmail.com
      edad: 24
  direccion: Calle 23
      ciudad: Barranquilla
      estado: Inactivo
      creado: 2023-06-05 07:31:22
***** 3. row *****
      id: 1075678987
nombre_apellido: Juan Pérez
      correo: jp@hotmail.com
      edad: 20
  direccion: Calle 13
      ciudad: Cali
      estado: Inactivo
      creado: 2023-06-05 07:31:22
***** 4. row *****
```

--- ¿Cómo se realizaría la consulta de un registro o conjunto de registros específicos o particulares de una tabla?

**Nota::** Recuerden que los id en sus registros no son los mismos que en la base de demostración de esta guía, es decir los registros de su base y la de esta guía no son los mismos, deben hacer el ajuste para seguir el paso a paso.

Para poder seleccionar una fila en particular de una tabla, se utiliza **WHERE**.

## CASO 1.

Si necesitas devolver todas las filas correspondientes al id = 1075678987, escribiríamos:

```
SELECT * FROM aprendices WHERE id = 1075678987;
```

```
mysql> SELECT * FROM aprendices WHERE id = 1075678987;
```

id	nombre_apellido	correo	edad	direccion	ciudad	estado	creado
1075678987	Juan Pérez	jp@hotmail.com	20	Calle 13	Cali	Inactivo	2023-06-05 07:31:22

```
1 row in set (0.01 sec)
```

```
mysql>
```

CASO 2.

Si necesitas devolver todas las filas correspondientes al correo = 'eylozano@sena.edu.co', escribiríamos:

**SELECT \* FROM** aprendices **WHERE** correo = 'eylozano@sena.edu.co';

```
mysql> SELECT * FROM aprendices WHERE correo = 'eylozano@sena.edu.co';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id      | nombre_apellido | correo           | edad | direccion | ciudad | estado | creado           |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1075861807 | Eliana Lozano   | eylozano@sena.edu.co | 31   | Calle 15  | Garzón | Activo | 2023-06-05 07:31:22 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

CASO 3.

Obtener todos los usuarios con edad mayor o igual a 24.

**SELECT \* FROM** aprendices **WHERE** edad >= 24;

```
mysql> SELECT * FROM aprendices WHERE edad >= 24;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id      | nombre_apellido | correo           | edad | direccion | ciudad | estado | creado           |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 12187654 | Maria Pepita    | mariapepita1@gmail.com | 24   | Calle 23  | Barranquilla | Inactivo | 2023-06-05 07:31:22 |
| 1075861807 | Eliana Lozano   | eylozano@sena.edu.co | 31   | Calle 15  | Garzón | Activo | 2023-06-05 07:31:22 |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

CASO 4.

Obtener todos los registros con fecha de creación el día de hoy: (2023-06-05 00:00:01).

**SELECT \* FROM** aprendices **WHERE** creado > '2023-06-05 00:00:01';

```
mysql> SELECT * FROM aprendices WHERE creado > '2023-06-05 00:00:01';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id      | nombre_apellido | correo           | edad | direccion | ciudad | estado | creado           |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 5678345 | Ramón Trujillo  | donramon@hotmail.com | 20   | Calle del ocho | Jamundí | Activo | 2023-06-05 07:31:22 |
| 12187654 | Maria Pepita    | mariapepita1@gmail.com | 24   | Calle 23  | Barranquilla | Inactivo | 2023-06-05 07:31:22 |
| 1075678987 | Juan Pérez     | jpp@hotmail.com | 20   | Calle 13  | Cali | Inactivo | 2023-06-05 07:31:22 |
| 1075861807 | Eliana Lozano   | eylozano@sena.edu.co | 31   | Calle 15  | Garzón | Activo | 2023-06-05 07:31:22 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

El campo fecha y hora de creación siempre tendrá esta estructura: **año-mes-día**  
**hora:min:seg**

CASO 5.

Obtener todos los registros con id: 1075678987, 5678345.

**SELECT \* FROM** aprendices **WHERE** id **IN** (1075678987, 5678345);

```
mysql> SELECT * FROM aprendices WHERE id IN (1075678987, 5678345);
+-----+-----+-----+-----+-----+-----+-----+-----+
| id      | nombre_apellido | correo           | edad | direccion | ciudad | estado | creado           |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 5678345 | Ramón Trujillo  | donramon@hotmail.com | 20   | Calle del ocho | Jamundí | Activo | 2023-06-05 07:31:22 |
| 1075678987 | Juan Pérez     | jpp@hotmail.com | 20   | Calle 13  | Cali | Inactivo | 2023-06-05 07:31:22 |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

El operador **IN** nos permite limitar la búsqueda a campos específicos o listas.

CASO 6.

Obtener el nombre y correo electrónico del usuario de mayor edad.

Lo primero que debemos hacer es ordenar los registros con respecto a una columna que en este caso sería edad.

**SELECT \* FROM** aprendices **ORDER BY** edad;

```
mysql> SELECT * FROM aprendices ORDER BY edad;
```

id	nombre_apellido	correo	edad	direccion	ciudad	estado	creado
5678345	Ramón Trujillo	donramon@hotmail.com	20	Calle del ocho	Jamundí	Activo	2023-06-05 07:31:22
1075678987	Juan Pérez	jp@hotmail.com	20	Calle 13	Cali	Inactivo	2023-06-05 07:31:22
12187654	Maria Pepita	mariapepita1@gmail.com	24	Calle 23	Barranquilla	Inactivo	2023-06-05 07:31:22
1075861807	Eliana Lozano	eylozano@sena.edu.co	31	Calle 15	Garzón	Activo	2023-06-05 07:31:22

4 rows in set (0.00 sec)

Por Default el ordenamiento será ascendente, pero podemos especificar si es ascendente con **ASC** o descendente con **DESC**

**SELECT \* FROM** aprendices **ORDER BY** edad **DESC**;

```
mysql> SELECT * FROM aprendices ORDER BY edad DESC;
```

id	nombre_apellido	correo	edad	direccion	ciudad	estado	creado
1075861807	Eliana Lozano	eylozano@sena.edu.co	31	Calle 15	Garzón	Activo	2023-06-05 07:31:22
12187654	Maria Pepita	mariapepita1@gmail.com	24	Calle 23	Barranquilla	Inactivo	2023-06-05 07:31:22
5678345	Ramón Trujillo	donramon@hotmail.com	20	Calle del ocho	Jamundí	Activo	2023-06-05 07:31:22
1075678987	Juan Pérez	jp@hotmail.com	20	Calle 13	Cali	Inactivo	2023-06-05 07:31:22

4 rows in set (0.00 sec)

Ahora bien, para obtener el nombre y correo electrónico del usuario de mayor edad, lo que debemos hacer es establecer un limite para el primer registro que vendría a hacer el de mayor edad una vez ordenado descendentemente.

**SELECT** nombre\_apellido, correo **FROM** aprendices **ORDER BY** edad **DESC LIMIT 1**;

```
mysql> SELECT nombre_apellido, correo FROM aprendices ORDER BY edad DESC LIMIT 1;
```

nombre_apellido	correo
Eliana Lozano	eylozano@sena.edu.co

Con **LIMIT** lo que hacemos es delimitar a que registros queremos devolver, puede ser el primero, el segundo ... n.

CASO 7.

Se requiere obtener todos los correos que terminen en el sufijo **.com**.

**SELECT \* FROM** aprendices **WHERE** correo **LIKE** '%.com';

```
mysql> SELECT * FROM aprendices WHERE correo LIKE "%.com";
```

id	nombre_apellido	correo	edad	direccion	ciudad	estado	creado
5678345	Ramón Trujillo	donramon@hotmail.com	20	Calle del ocho	Jamundí	Activo	2023-06-05 07:31:22
12187654	Maria Pepita	mariapepita1@gmail.com	24	Calle 23	Barranquilla	Inactivo	2023-06-05 07:31:22
1075678987	Juan Pérez	jp@hotmail.com	20	Calle 13	Cali	Inactivo	2023-06-05 07:31:22

3 rows in set (0.00 sec)

CASO 8.

Se requiere obtener todos los correos que terminen en el prefijo **don**.

```
mysql> SELECT * FROM aprendices WHERE correo LIKE 'don%';
```

id	nombre_apellido	correo	edad	direccion	ciudad	estado	creado
5678345	Ramón Trujillo	donramon@hotmail.com	20	Calle del ocho	Jamundí	Activo	2023-06-05 07:31:22

1 row in set (0.00 sec)

El **LIKE** nos permite trabajar con cadenas de caracteres: TEXT o VARCHAR para prefijos o sufijos.