

DML

Lenguaje de Manipulación de Datos

- SELECT



¿Qué es SQL?

- **Structured Query Language** (Lenguaje de Consulta Estructurado)
- **DDL** (Lenguaje de Definición de Datos): Es el lenguaje encargado de la definición, alteración y eliminación de objetos en la base de datos. (CREATE, ALTER, DROP)
- **DML** (Lenguaje de Manipulación de Datos): Es el lenguaje encargado de la manipulación de los datos. (SELECT, INSERT, UPDATE, DELETE)
- **DCL** (Lenguaje de Control de Datos): Es el lenguaje que controla el acceso a las bases de datos. (GRANT, REVOKE)
- **TCL** (Lenguaje de Control de Transacciones): Es el lenguaje que controla la ejecución de comandos DML; indispensable para la consistencia e integridad de los datos.

Comando SELECT

El comando **select** permite mostrar los registros de las tablas. En álgebra relacional es equivalente a la selección (σ) y proyección (π) según sea el caso

Sintaxis 1: para listar todas las columnas de una tabla:

SELECT * FROM nombre_tabla;

Ejemplos:

select * from estado; $\rightarrow \sigma$ (estado)

select * from cancion; $\rightarrow \sigma$ (cancion)

select * from idioma; $\rightarrow \sigma$ (idioma)

select * from pais; $\rightarrow \sigma$ (pais)

Comando SELECT

Sintaxis 2: para listar todos los datos de algunas columnas de una tabla

SELECT col1,col2,...colN **FROM** nombre_tabla;

$\pi_{\text{col1,col2}}$ (R)

Ejemplos:

select nombre from disquera; $\rightarrow \pi_{\text{nombre}}$ (disquera)

select titulo,precio from disco; $\rightarrow \pi_{\text{titulo,precio}}$ (disco)

select nombre,apellido_paterno,correo from cliente;

$\pi_{\text{nombre,apellido_paterno, correo}}$ (cliente)

Comando SELECT

Sintaxis 3: Ordenar la listar datos según el orden especificado

SELECT col1,col2 **FROM** nombre_tabla **ORDER BY** Nombre_columna [asc | desc];

SELECT col1,col2 **FROM** nombre_tabla **ORDER BY** Numero_columna [asc | desc];

Ejemplos:

1. Orden ascendente (default)

```
select id_cliente,nombre,apellido_paterno,apellido_materno  
from cliente order by id_cliente;
```

Equivalente a:

```
select id_cliente,nombre,apellido_paterno,apellido_materno  
from cliente order by id_cliente asc;
```

Ordenar por número de columna:

```
select id_cliente,nombre,apellido_paterno,apellido_materno  
from cliente order by 3; -- ordena ascendentemente por la columna 3 (apellido_paterno)
```

Comando SELECT

Ejemplos:

2. Orden descendente

```
select id_cliente,nombre,apellido_paterno,apellido_materno  
from cliente order by id_cliente desc;
```

```
select id_cliente,nombre,apellido_paterno,apellido_materno  
from cliente order by 3 desc;
```

Uso de DISTINCT para listar valores únicos

3. Listar todos los apellidos paternos únicos

```
select distinct apellido_paterno from cliente; -- distinct no muestra valores  
repetidos, sólo valores distintos
```

Comando SELECT – WHERE

Sintaxis 4: para listar datos de una tabla según la condición establecida (clausula where)

SELECT * FROM nombre_tabla WHERE condición;

$\sigma_{\text{campo} = \text{<valor>}}$ (tabla)

Ejemplos:

1. Seleccionar todos los clientes que sean mujeres

```
select id_cliente,nombre,apellido_paterno,apellido_materno,sexo from  
cliente where sexo = 'M';
```

$\sigma_{\text{sexo} = \text{'M'}}$ (cliente)

Comando SELECT – WHERE

2. Seleccionar todos los clientes donde apellido paterno sea igual a 'Sanchez'

```
select * from cliente where apellido_paterno = 'Sanchez';
```

$\sigma_{\text{apellido_paterno} = \text{'Sanchez'}}$ (cliente)

3.

$\Pi_{\text{id_cliente, nombre, apellido_paterno, apellido_materno}}$ ($\sigma_{\text{apellido_materno} = \text{'Hernandez'}}$ (cliente))

¿Cuál es la instrucción SQL?

Comando SELECT – WHERE

4. Seleccionar todos los nombres de clientes que comiencen con 'M'
`select * from cliente where nombre like 'M%';`

5. Seleccionar todos los apellidos paternos de clientes que terminen con 'ez'
`select * from cliente where apellido_paterno like '%ez';`

6. Seleccionar todos los nombres de clientes que contengan una 'u'
`select nombre from cliente where nombre like '%u%';`

7. Seleccionar todos los nombres de clientes que tengan una 'a' en la segunda letra
`select nombre from cliente where nombre like '_a%';`

8. Seleccionar todos los apellidos paternos de clientes que no sean 'López'
`select * from cliente where apellido_paterno <> 'López';`

Comando SELECT – WHERE ... AND, OR

9. Seleccionar todos los apellidos paternos de clientes que terminen con 'ez' y que sean hombres

```
select * from cliente where apellido_paterno like '%ez' and sexo = 'H';
```

10. Seleccionar todos los nombres de clientes que contengan una 'a' en su nombre **y** que su apellido materno sea 'Hernández'

```
select * from cliente where nombre like '%a%' and apellido_materno like 'Hernández';
```

11. Seleccionar todos los nombres de clientes que empiecen con 'E' **o** que su apellido materno sea 'Hernández'

```
select * from cliente where nombre like 'E%' or apellido_materno like 'Hernández';
```

Comando SELECT (sobre campos de fechas)

Aunque las fechas se consideran como cadenas, también se pueden manejar como tal, pero no se recomienda ya que existen funciones específicas para fechas como year o month.

12. Seleccionar todos los discos que fueron lanzados en 2010

```
select * from disco where fecha_lanzamiento like '2010-%';
```

13. Seleccionar todos los discos que fueron lanzados en los años 2012 y 2018

```
select * from disco where fecha_lanzamiento like '2012-%' OR  
fecha_lanzamiento like '2018-%';
```

14. Seleccionar todos los discos que fueron lanzados en marzo del 2020

```
select * from disco where fecha_lanzamiento like '2020-03-%' ;
```

Comando SELECT (sobre campos numéricos)

15. Seleccionar todos los discos que cuesten menos de \$300

```
select * from disco where precio < 300;
```

16. Seleccionar todos los discos que cuesten más de \$500

```
select * from disco where precio > 500;
```

17. Seleccionar todos los discos que cuesten \$300 o más pero menos de \$500

```
select * from disco where precio >= 350 and precio < 500;
```

Comando SELECT – IN - BETWEEN

18. Seleccionar los discos con cantidad mayor o igual que 500 y menor o igual que 700

```
select * from disco where cantidad_disponible >= 500 and  
cantidad_disponible <=700;
```

Equivalente:

```
select titulo,precio,cantidad_disponible from disco where  
cantidad_disponible between 500 and 700;
```

19. Seleccionar todos los discos que tengan cantidad disponible 1, 300 y 855

```
select titulo,precio,cantidad_disponible from disco where  
cantidad_disponible = 1 or cantidad_disponible = 300 or cantidad_disponible  
= 855;
```

Equivalente:

```
select titulo,precio,cantidad_disponible from disco where  
cantidad_disponible in (1,300,855);
```

¿Preguntas?