

Cátedra Desarrollo de Software
Taller de Base de Datos

SQLite

SQL - SELECT

ESTRUCTURA BÁSICA DEL COMANDO **SELECT**

```
SELECT <columna1>, <columna2>, . . . .  
FROM <nombre_tabla>  
WHERE <restriccion1>  
<operador lógico> <restriccion2> . . . .  
ORDER BY <columna1> <tipo_orden>
```

ESTRUCTURA BÁSICA DEL COMANDO **SELECT**

```
SELECT <columna1>, <columna2>, . . . .  
FROM <nombre_tabla>  
WHERE <restriccion1>  
<operador lógico> <restriccion2> . . . .  
ORDER BY <columna1> <tipo_orden>
```

ESTRUCTURA BÁSICA DEL COMANDO **SELECT**

```
SELECT <columna1>, <columna2>, . . . .  
FROM <nombre_tabla>  
WHERE <restriccion1>  
<operador lógico> <restriccion2> . . . .  
ORDER BY <columna1> <tipo_orden>
```

ESTRUCTURA BÁSICA DEL COMANDO **SELECT**

```
SELECT <columna1>, <columna2>, . . . .  
FROM <nombre_tabla>  
WHERE <restriccion1>  
<operador lógico> <restriccion2> . . . .  
ORDER BY <columna1> <tipo_orden>
```

ESTRUCTURA BÁSICA DEL COMANDO **SELECT**

```
SELECT <columna1>, <columna2>, . . . .  
FROM <nombre_tabla>  
WHERE <restriccion1>  
<operador lógico> <restriccion2> . . . .  
ORDER BY <columna1> <tipo_orden>
```

ESTRUCTURA BÁSICA DEL COMANDO **SELECT**

```
SELECT <columna1>, <columna2>, . . . .  
FROM <nombre_tabla>  
WHERE <restriccion1>  
<operador lógico> <restriccion2> . . . .  
ORDER BY <columna1> <tipo_orden>
```

SELECT Tipos de Restricciones/Operadores de Comparación en el **WHERE**

Operadores de comparación: se utilizan para comparar valores y devolver resultados en función de si se cumple o no una determinada condición.

Los operadores de comparación incluyen:

"=", "<", ">", "<=", ">=" y "<>"

SELECT Tipos de Restricciones/Operadores de Comparación en el **WHERE**

Operadores lógicos: se utilizan para combinar múltiples condiciones y devolver un resultado en función de si se cumplen todas las condiciones o al menos una de ellas. Los operadores lógicos incluyen **"AND", "OR" y "NOT"**

SELECT Tipos de Restricciones/Operadores de Comparación en el **WHERE**

Operador de coincidencia de patrones: se utilizan para buscar patrones dentro de una cadena de texto. Los operadores de coincidencia de patrones incluyen:
"LIKE"

SELECT Tipos de Restricciones/Operadores de Comparación en el **WHERE**

Operadores de pertenencia: se utilizan para verificar si un valor está incluido en una lista de valores. Los operadores de pertenencia incluyen:
"IN" y "NOT IN "

SELECT Tipos de Restricciones/Operadores de Comparación en el **WHERE**

Operadores de comparación de nulos: se utilizan para comparar valores nulos. Los operadores de comparación de nulos incluyen:
"IS NULL" y "IS NOT NULL"

Tomando el contenido de la Tabla Personas con los siguiente datos

id	nombre	fecha	cantidad	importe
1	Juan	2023-03-01	10.5	125.25
2	Antonio	1960-07-22	20	245.11
3	José	1977-08-14	30	320.44
4	María	1944-04-26	12	128.0
5	Ramón	1968-11-27	145	1440.78
6	Nicolás	1980-10-10	13	134.87
7	Fernando	2000-10-10	20	200.0
8	Santiago	2016-12-24	421	4421.1
9	Candelaria	2002-11-24	728	8412.0
10	Luis	1960-08-11	1478	1854.22
11	Miguel	2005-10-11	13000	130000.0
12	Marcela	1964-12-10	14478.12	160637.2
13	Gabriel	2022-05-14	479	18725.3
14	Esteban	1978-03-03	78	478.0
15	Soledad	1971-07-06	4789	490000.0

SELECT Ejemplo con la tabla Personas

SELECT id, nombre, importe
FROM personas
WHERE id > 10
ORDER BY nombre asc;

id	nombre	importe
14	Esteban	478.0
13	Gabriel	18725.3
12	Marcela	160637.2
11	Miguel	130000.0
15	Soledad	490000.0

Tipo de Ordenamiento

asc: orden ascendente, se coloca después del nombre de la columna.

desc: orden descendente.

Operadores Similares

"=", "<", ">", "<=", ">=" y "<>"

SELECT Ejemplo con la tabla Personas

SELECT id, nombre, importe, fecha
FROM personas
WHERE id > 10 **AND** id <=12
ORDER BY fecha asc;

id	nombre	importe	fecha
12	Marcela	160637.2	1964-12-10
11	Miguel	130000.0	2005-10-11

Tipo de Ordenamiento

asc: orden ascendente, se coloca después del nombre de la columna.

desc: orden descendente.

Operadores Similares

"=", "<", ">", "<=", ">=" y "<>"

SELECT Ejemplo con la tabla Personas

SELECT id, nombre, importe, fecha
FROM personas
WHERE id > 10 and id <=12
ORDER BY fecha desc;

11	Miguel	130000.0	2005-10-11
12	Marcela	160637.2	1964-12-10

Tipo de Ordenamiento

asc: orden ascendente, se coloca después del nombre de la columna.

desc: orden descendente.

Operadores Similares

"=", "<", ">", "<=", ">=" y "<>"

SELECT Ejemplo con la tabla Personas

SELECT id, nombre, importe, fecha

FROM personas

WHERE nombre like '%a%'

ORDER BY fecha desc;

Formas De Utilizar Los Comodines

(%) remplaza múltiples caracteres.

(_) remplaza un carácter.

Nombre like '%a%'

Nombre like '%a'

Nombre like 'a%'

Nombre like '_a%'

Nombre like '%a_'

Tipo de Ordenamiento

asc: orden ascendente, se coloca después del nombre de la columna.

desc: orden descendente.

id	nombre	importe	fecha
1	Juan	125.25	2023-03-01
13	Gabriel	18725.3	2022-05-14
8	Santiago	4421.1	2016-12-24
9	Candelaria	8412.0	2002-11-24
7	Fernando	200.0	2000-10-10
14	Esteban	478.0	1978-03-03
15	Soledad	490000.0	1971-07-06
5	Ramón	1440.78	1968-11-27
12	Marcela	160637.2	1964-12-10
2	Antonio	245.11	1960-07-22
4	María	128.0	1944-04-26

SELECT Ejemplo con la tabla Personas

SELECT id, nombre, importe, fecha

FROM personas

WHERE not (nombre like '%a%')

ORDER BY fecha desc;

id	nombre	importe	fecha
11	Miguel	130000.0	2005-10-11
6	Nicolás	134.87	1980-10-10
3	José	320.44	1977-08-14
10	Luis	1854.22	1960-08-11

Tipo de Ordenamiento

asc: orden ascendente, se coloca después del nombre de la columna.

desc: orden descendente.

Formas De Utilizar Los Comodines

(%) reemplaza múltiples caracteres.

(_) reemplaza un carácter.

Nombre like '%a%'

Nombre like '%a'

Nombre like 'a%'

Nombre like '_a%'

Nombre like '%a_'

SELECT Ejemplo con la tabla Personas

SELECT *

FROM personas

WHERE id in (2,6 ,8, 12);

id	nombre	fecha	cantidad	importe
2	Antonio	1960-07-22	20	245.11
6	Nicolás	1980-10-10	13	134.87
8	Santiago	2016-12-24	421	4421.1
12	Marcela	1964-12-10	14478.12	160637.2

SELECT Ejemplo con la tabla Personas

SELECT *

FROM personas

WHERE id not in (2,6 ,8, 12);

id	nombre	fecha	cantidad	importe
1	Juan	2023-03-01	10.5	125.25
3	José	1977-08-14	30	320.44
4	María	1944-04-26	12	128.0
5	Ramón	1968-11-27	145	1440.78
7	Fernando	2000-10-10	20	200.0
9	Candelaria	2002-11-24	728	8412.0
10	Luis	1960-08-11	1478	1854.22
11	Miguel	2005-10-11	13000	130000.0
13	Gabriel	2022-05-14	479	18725.3
14	Esteban	1978-03-03	78	478.0
15	Soledad	1971-07-06	4789	490000.0

SELECT Ejemplo con la tabla Personas

SELECT *

FROM personas

WHERE id **IN** (2,6 ,8, 12)

OR id = 1

id	nombre	fecha	cantidad	importe
2	Antonio	1960-07-22	20	245.11
6	Nicolás	1980-10-10	13	134.87
8	Santiago	2016-12-24	421	4421.1
12	Marcela	1964-12-10	14478.12	160637.2
1	Juan	2023-03-01	10.5	125.25

SELECT Ejemplo con la tabla Personas

SELECT *

FROM personas

WHERE id **IN** (1,6 ,8, 12)

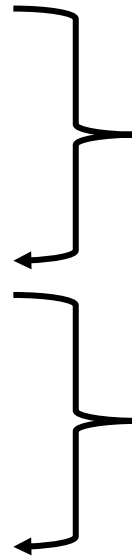
AND id = 1

id	nombre	fecha	cantidad	importe
1	Juan	2023-03-01	10.5	125.25

SELECT Ejemplo con la tabla Personas

SELECT *
FROM personas
WHERE nombre **IS NULL**;

SELECT *
FROM personas
WHERE nombre **IS NOT NULL**;



Por como está carga la tabla devuelve todos los registro de la tabla