

Índice.

1. Formato básico de una consulta.....	2
2. Filtros en una consulta.....	2
2.1. Expresiones para filtros.....	2
2.2. Tipos de filtros.....	3
2.2.1 Filtros con operador de pertenencia a conjuntos.....	3
2.2.2 Filtros con operador de rango.....	3
2.2.3 Filtros con test de valor nulo.....	3
2.2.4 Filtros con test de patrón.....	4
2.2.5 Filtros con límite de número de registros.....	4
2.2.6 Filtros con expresiones regulares.....	5
3. Ordenación de una consulta.....	5

1. Formato básico de una consulta.

El formato básico para hacer una consulta es:

```
SELECT [DISTINCT] expresión_select [,expresión_select]  
[FROM tabla]
```

La selección de todos los datos de las personas almacenadas es:

```
SELECT *  
FROM persona ;
```

2. Filtros de una consulta.

Los filtros son condiciones que cualquier gestor de base de datos interpreta para seleccionar registros y mostrarlos como el resultado de una consulta.

En SQL la palabra clave para realizar filtros es la cláusula WHERE.

```
SELECT [DISTINCT] expresión_select [,expresión_select]  
[FROM tabla]  
[WHERE filtro]
```

2.1. Expresiones para filtros.

Un **filtro** es una expresión que indica la condición o condiciones que deben satisfacer todos los registros para ser seleccionados.

- Los filtros se construyen con expresiones.
- Una expresión es una combinación de operadores, operandos y funciones para producir un resultado.

Los elementos que forman parte de las expresiones son:

- Operandos → constantes, números enteros, números reales, cadenas de caracteres, fechas, variables u otras expresiones.
- Operadores aritméticos → +, -, *, /, %.
- Operadores relacionales → >, <, <>, >=, <=, =.
- Operadores lógicos → AND, OR, NOT.
- Paréntesis → ()
- Funciones → cada sistema gestor de bases de datos incorpora su propio repertorio de funciones.

2.2. Tipos de filtros.

Los filtros que se emplean en SQL pueden ser de los siguientes tipos:

- Filtros con operador de pertenencia a conjuntos.
- Filtros con operador de rango.
- Filtros con test de valor nulo.
- Filtros con test de patrón.
- Filtros con límite de número de registros.

2.2.1. Filtros con operador de pertenencia a conjuntos.

El operador IN permite comprobar si una columna tiene un valor igual que cualquiera de los que están incluidos dentro del paréntesis.

```
nombre_columna IN (valor1, valor2, ...)
```

Selección de todos los datos de las personas cuya ciudad sea o Pontevedra o Vigo:

```
SELECT *  
FROM persona  
WHERE ciudad IN ('Pontevedra','Vigo');
```

2.2.2. Filtros con operador de rango.

El operador BETWEEN permite seleccionar los registros que estén incluidos en un rango.

```
nombre_columna BETWEEN valor1 AND valor2
```

Selección de todos los datos de las personas cuya altura esté entre 160 y 180 cm:

```
SELECT *  
FROM persona  
WHERE altura BETWEEN 160 AND 180;
```

2.2.3. Filtros con test de valor nulo.

Los operadores IS e IS NOT permiten verificar si un campo es o no nulo, respectivamente.

```
nombre_columna IS [NOT] NULL
```

Selección de todos los datos de las personas cuya altura se desconozca:

```
SELECT *  
FROM persona  
WHERE altura IS NULL;
```

2.2.4. Filtros con test de patrón.

El operador LIKE permite seleccionar los registros a partir de un patrón.

Los filtros asociados a los test de patrón son los siguientes:

- % → busca coincidencias de cualquier número de caracteres, incluso de cero caracteres.
- _ → busca coincidencias de exactamente un carácter.

Estos filtros permiten seleccionar los registros que cumplan una serie de características.

Selección de las personas cuyo nombre contenga 'mas':

```
SELECT *  
FROM persona  
WHERE nombre LIKE '%mas%';
```

2.2.5. Filtros con límite de número de registros.

Este filtro limita el número de registros que devuelve una consulta.

```
[LIMIT [desplazamiento,] filas]
```

- Filas especifica el número de filas que se devolverán
- Desplazamiento especifica a partir de qué file se empieza a contar el desplazamiento.

Selección de los datos de la primera persona de la lista:

```
SELECT *  
FROM persona LIMIT 1;
```

2.2.6. Filtros con expresiones regulares de patrones.

El operador REGEXP (o RLIKE) permite seleccionar los registros a partir de expresiones regulares.

Una expresión regular es una poderosa herramienta para determinar si una cadena de texto se ajusta a un determinado patrón.

```
SELECT *
FROM persona
WHERE nombre REGEXP expresión_regul;
```

La expresión regular estará contenida entre comillas (simples ', o dobles '"').

Los caracteres especiales que puede contener la expresión regular son los siguientes:

- **Comienzo y fin de la cadena.**
 - Indicador de comienzo de cadena: ^
Selección de las personas cuyo nombre empiece por H → `select * from persona where nombre REGEXP '^H';`
 - Indicador de fin de cadena: \$
Selección de las personas cuyo nombre acabe en z → `select * from persona where nombre REGEXP 'z$';`
- **Coincidencia con caracteres.**
 - Con cualquier carácter: .
Selección de las personas cuyo nombre empiece por cualquier letra → `select * from persona where nombre REGEXP '^.';`
 - Con una secuencia de caracteres: |
Selección de las personas cuyo nombre empiece por J, S o L → `select * from persona where nombre REGEXP '^J|^S|^L';`
- **Indicadores de repetición.**
 - El carácter que precede puede aparecer cero o una vez: ?
Selección de las personas cuyo nombre no tenga una L, o sólo una → `select * from persona where nombre REGEXP 'L?';`
 - El carácter que precede puede aparecer una o más veces: +
Selección de las personas cuyo nombre empiece tenga una L o más → `select * from persona where nombre REGEXP 'L+';`
 - El carácter que precede puede aparecer cero, una o más veces: *
Selección de las personas cuyo nombre no tenga una l, o una vez o varias → `select * from persona where nombre REGEXP 'l*';`
 - El carácter que precede aparecerá en un rango de valores mínimo y máximo: {min,max}
Selección de las personas cuyo nombre empiecen por l o ll → `select * from persona where nombre REGEXP '^l{1,2}';`
 - Los caracteres que preceden aparecerán en un rango de valores: (caracteres){min,max}
Selección de los nombres de persona cuyo NIF tenga un 1, uno o dos 0 y un 1 → `select * from persona where NIF REGEXP '1(0){1,2}1';`
- **Secuencia de caracteres.**
 - Coincidencia con un carácter en un conjunto: []
Selección de las personas cuyo nombre empiece por a hasta d → `select * from persona where nombre REGEXP '^[a-d]';`

Selección de las personas cuyo nombre no empiece por a hasta d → `select * from persona where nombre REGEXP '^[^a-d]';`
 - Coincidencia con la totalidad de los caracteres: ()
Selección de las personas cuyo nombre sea María o Ignacio → `select * from persona where nombre REGEXP '(María|Ignacio)';`

3. Ordenación de una consulta.

La ordenación de los datos mostrados en una consulta se realiza a través de la cláusula ORDER BY de la sentencia SELECT.

Esta cláusula permite ordenar el conjunto de resultados de forma ascendente o descendente, por una o varias columnas. Si no se indica nada, se ordenará ascendentemente.

La columna por la que se ordenará se expresará por el nombre de dicha columna, una expresión o por la posición numérica del campo.

```
SELECT [DISTINCT] expresión_select [,expresión_select]  
[FROM tabla]  
[WHERE filtro]  
[ORDER BY {nombre_columna | expresión | posición} [ASC | DESC] ], ...
```