

Bases de Datos

- Sentencias SQL
- Registros duplicados Distinct
- Patrón regexp
- Alias

Lenguaje SQL

Tabla clientes

codcliente	apellido	nombre	codpostal
100	Perez	Gabriel	5500
101	Gonzalez	Catalina	5501
102	Barroso	Lautaro	5500
103	Balmaceda	Juan	5502
104	Baez	Juan	5502
105	Baez	Carlos	5501
106	Balmaceda	Juan	5502
107	Fabres	Juan	1001
110	Gonzalez		1001
111	Cabo	Andres	5500

Lenguaje SQL

distinct

Con la cláusula "distinct" se especifica que los registros con ciertos datos duplicados sean obviados en el resultado. **Por ejemplo,** queremos conocer todos los clientes de nombre "Juan" si utilizamos esta sentencia:

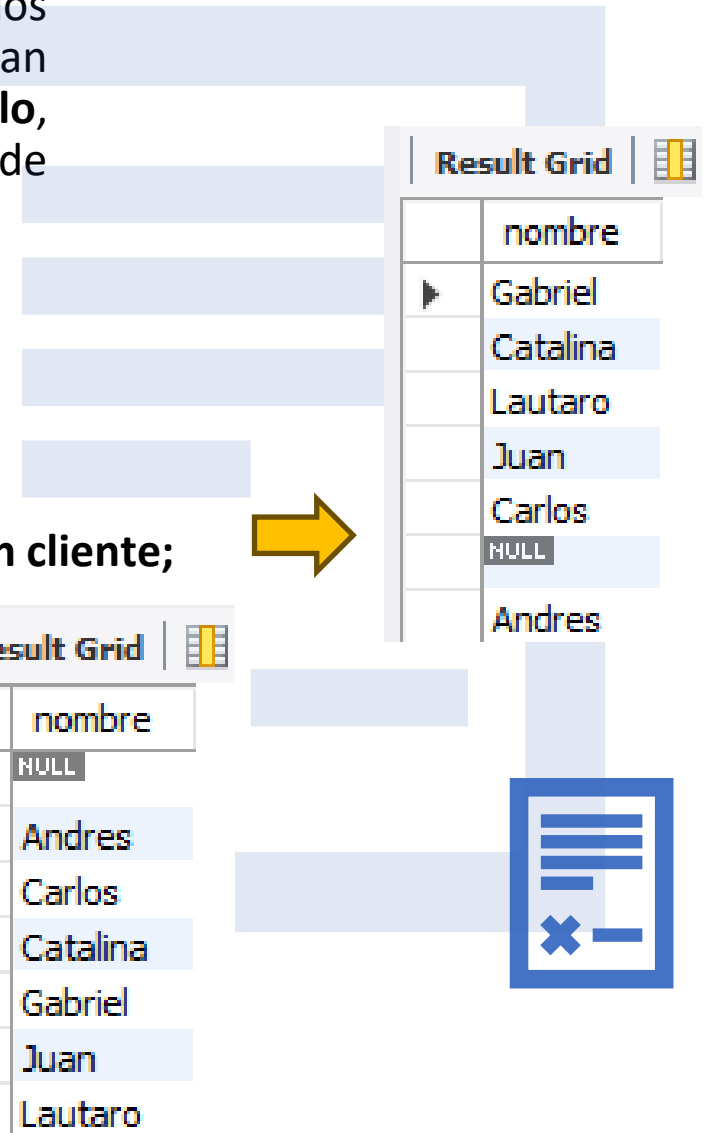
```
select nombre from cliente;
```

Aparecen repetidos. Para obtener la lista de nombres sin repetición usamos:

```
select distinct nombre from cliente;
```

También podemos tipear y producir el mismo resultado:

```
select nombre from cliente  
group by nombre;
```



Lenguaje SQL

Note que en los tres casos anteriores aparece "null" como un valor para "nombre". Si sólo queremos la lista de nombres conocidos, es decir, no queremos incluir "null" en la lista, podemos utilizar la sentencia siguiente:

```
select distinct nombre from clientes  
where nombre is not null;
```



Result Grid		
	nombre	
▶	Gabriel	
	Catalina	
	Lautaro	
	Juan	
	Carlos	
	Andres	

Para contar los distintos nombres, sin considerar el valor "null" usamos:

```
select count(distinct nombre)  
from clientes;
```



Result Grid			Filter Rows:
	count(distinct nombre)		
▶	6		

Note que, si contamos los nombres sin "distinct", no incluirá los valores "null" pero si los repetidos:

```
select count(nombre)  
from clientes;
```



Result Grid			Filter Rows:
	count(nombre)		
▶	9		

Lenguaje SQL

Búsqueda de patrones **regexp**

Los operadores "regexp" y "not regexp" busca patrones de modo similar a "like" y "not like".

Para buscar nombres que contengan la cadena "al" usamos:

```
select nombre  
from clientes  
where nombre regexp 'al';
```



Result Grid	
	nombre
▶	Catalina

Para buscar los nombres que tienen al menos una "h" o una "k" o una "w" tipeamos:

```
select nombre from clientes  
where nombre regexp '[hkwl]';
```



Result Grid	
	nombre
▶	Gabriel
	Catalina
	Carlos



Lenguaje SQL

Para buscar los nombres que tienen por lo menos una de las letras de la "a" hasta la "d", es decir, "a,b,c,d", usamos:

```
select nombre from clientes  
where nombre regexp '[a-d]';
```

Para ver los apellidos que comienzan con "A" tipeamos:

```
select apellido from clientes  
where apellido regexp '^A';
```

Para ver los apellidos que terminan en "AN" usamos:

```
select apellido from clientes  
where apellido regexp 'AN$';
```

Para buscar apellido que contengan una "a" luego un caracter cualquiera y luego una "e" utilizamos la siguiente sentencia:

```
select apellido from clientes  
where apellido regexp 'a.e';
```

El punto (.) identifica cualquier caracter.



Lenguaje SQL

Podemos mostrar los apellidos que contienen una "a" seguida de 2 caracteres y luego una "e":

```
select apellido from clientes  
where apellido regexp 'a..e';
```

Para buscar apellidos que tengan 6 caracteres exactamente usamos:

```
select autor from libros  
where autor regexp '^.....$';
```



Lenguaje SQL

Para buscar apellidos que tengan al menos 6 caracteres usamos:

```
select apellido from clientes  
where apellido regexp '.....';
```

Para buscar apellidos que contengan 2 letras "a" usamos:

```
select apellido from clientes  
where apellido regexp 'a.*a';
```

El asterisco indica que busque el caracter inmediatamente anterior, en este caso cualquiera porque hay un punto.

Lenguaje SQL

Alias



Un "**alias**" se usa como nombre de un campo o de una expresión o para referenciar una tabla cuando se utilizan más de una tabla (*tema que veremos más adelante*).

Cuando usamos una función de agrupamiento, por ejemplo:

```
select count(*)  
from clientes  
where apellido like '%Borges%';
```

la columna en la salida tiene como encabezado "**count(*)**", para que el resultado sea más claro podemos utilizar un alias:

```
select count(*) as ClientesBorges  
from clientes  
where apellido like '%Borges%';
```

Un alias puede tener hasta 255 caracteres, acepta todos los caracteres. La palabra clave "as" es opcional en algunos casos, pero es conveniente usarla. Si el alias consta de una sola cadena las comillas no son necesarias, pero si contiene más de una palabra, es necesario colocarla entre comillas.

Gracias...