



# Introducción a Mysql

Otras funciones



# Clase 3

Temas

Puesta en común:

- Ejercicio práctico clase anterior

Teórico práctico:

- Where
- Joinear tablas (Tipos de Joins)
- In, and, or, between
- Distinct, count, max, min,
- Order by, group by, having



# Empecemos



# Where



## Where

El where es una cláusula condicional para filtrar los resultados. Usando esta cláusula, podemos especificar un criterio de selección para seleccionar los registros requeridos de una tabla.

Sintaxis:

```
SELECT * FROM <nombre_de_tabla> WHERE <condición/es>
```

También podemos seleccionar solo algunas columnas no hace falta traer todo siempre con el \*

```
SELECT <columna>, <columna> FROM <nombre_de_tabla> WHERE <condición/es>
```

## Where

**SELECT \* FROM jugador WHERE id\_jugador = 1**

id_jugador	nombre	apellid	sueldo_mensual	equipo_id_equipo	posicion_id_posicion	nacionalidad_id_nacionalidad
1	Lionel Andres	Messi	75000	1	1	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

#	Time	Action	Message	Duration / Fetch
1	16:41:38	SELECT * FROM jugador WHERE id_jugador = 1 LIMIT 0, 1000	1 row(s) returned	0,00047 sec / 0,000...

**SELECT nombre, apellido FROM jugador WHERE id\_jugador = 7**

nombre	apellido
Virgil	Van Dijk

#	Time	Action	Message	Duration / Fetch
1	16:42:05	SELECT nombre, apellido FROM jugador WHERE id_jugador = ...	1 row(s) returned	0,00052 sec / 0,000...



# Join



## Join

Antes de pasar a otros temas vamos a ver los joins.

Estos nos permiten unir y traer información desde distintas tablas.

Por ej: Yo no quiero ver el id del equipo, de la nacionalidad y la posicion, yo quiero ver el detalle.

id_jugador	nombre	apellido	sueldo_mensual	equipo_id_equipo	posicion_id_posicion	nacionalidad_id_nacionalidad
1	Lionel Andres	Messi	75000	1	1	1
2	Cristiano	Ronaldo	70000	2	2	2
3	Neymar	Junior	68000	3	2	3
4	Jan	Oblak	65000	4	3	4
5	Eden	Hazard	63000	5	2	5
6	Marc-Andre	ter Stegen	60000	1	3	6
7	Virgil	Van Dijk	58000	6	4	7
8	Paulo	Dybala	55000	2	6	1





## Join

En primer instancia tenemos que traer los datos de cada tabla poniendo adelante un alias para diferenciar desde donde vienen, por ej:

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad
```

Donde cada alias hace referencia a la tabla:

ju: jugador

e: equipo

p: posicion

n: nacionalidad



## Join

Luego declaramos nuestra tabla principal con su alias, en este caso jugador

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju
```



## Join

Y por último debemos relacionar todas las tablas por medio de su id, recuerda que la tabla jugador tiene las claves foraneas de equipo, posicion y nacionalidad

id_jugador	nombre	apellido	sueldo_mensual	equipo_id_equipo	posicion_id_posicion	nacionalidad_id_nacionalidad
1	Lionel Andres	Messi	75000	1	1	1
2	Cristiano	Ronaldo	70000	2	2	2
3	Neymar	Junior	68000	3	2	3
4	Jan	Oblak	65000	4	3	4
5	Eden	Hazard	63000	5	2	5
6	Marc-Andre	ter Stegen	60000	1	3	6
7	Virgil	Van Dijk	58000	6	4	7
8	Paulo	Dybala	55000	2	6	1



## Join

Es más o menos similar a cuando creamos las claves foráneas, relacionamos el id que está en la tabla principal con el id de la tabla al que pertenece.

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad
```

## Join

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Lionel Andres	Messi	75000	FC Barcelona	Extremo derecho	Argentina
Cristiano	Ronaldo	70000	Juventus	Extremo izquierdo	Portugal
Neymar	Junior	68000	Paris Saint-Germain	Extremo izquierdo	Brasil
Jan	Oblak	65000	Atletico Madrid	Arquero	Eslovenia
Eden	Hazard	63000	Real Madrid	Extremo izquierdo	Belgica
Marc-Andre	ter Stegen	60000	FC Barcelona	Arquero	Alemania
Virgil	Van Dijk	58000	Liverpool	Defensa central	Holanda
Paulo	Dybala	55000	Juventus	Medio centro ofensivo	Argentina

#	Time	Action	Message	Duration / Fetch
1	17:02:56	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	8 row(s) returned	0,00039 sec / 0,000...



## Join

Ahora podría utilizar el where para traer todos los jugadores del Barcelona:

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE e.detalle_equipo = 'FC Barcelona'
```



## Join

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Lionel Andres	Messi	75000	FC Barcelona	Extremo derecho	Argentina
Marc-Andre	ter Stegen	60000	FC Barcelona	Arquero	Alemania

#	Time	Action	Message	Duration / Fetch
1	17:05:59	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	2 row(s) returned	0,00048 sec / 0,000...

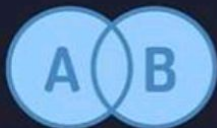
CLICK  
PARA  
MAS  
INFORMACION



SELECT \* FROM A  
**LEFT** JOIN B ON  
A.KEY = B.KEY



SELECT \* FROM A  
**LEFT** JOIN B ON  
A.KEY = B.KEY  
WHERE B.KEY IS  
NULL



SELECT \* FROM A  
**FULL OUTER** JOIN B  
ON A.KEY = B.KEY



SELECT \* FROM A  
**INNER** JOIN B ON  
A.KEY = B.KEY



SELECT \* FROM A  
**RIGHT** JOIN B ON  
A.KEY = B.KEY



SELECT \* FROM A  
**RIGHT** JOIN B ON  
A.KEY = B.KEY  
WHERE B.KEY IS  
NULL



SELECT \* FROM A  
**FULL OUTER** JOIN B  
ON A.KEY = B.KEY  
WHERE A.KEY IS  
NULL OR B.KEY IS  
NULL

# SQL JOINS





**In**



## In

El in nos permite aplicar una condición junto al where donde el filtro puede ser de varios ítems, por ejemplo:

Sintaxis:

```
SELECT * FROM <nombre_de_tabla> WHERE <valor_columna> IN (<valor>, <valor>...)
```

# In

En nuestro caso la query es más grande por los joins, pero la lógica es la misma:

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE e.detalle_equipo IN ('FC Barcelona', 'Real Madrid')
```

## In

Esa query va a ir a buscar a las tablas todos los jugadores que jueguen en el Barcelo y en el Madrid, haya o no haya datos para esos equipos, por ejemplo si agrego Bayern Munich, me va a devolver lo mismo porque no tengo jugadores en ese equipo.

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE e.detalle_equipo IN ('FC Barcelona', 'Real Madrid', 'Bayern Munich')
```



In

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Lionel Andres	Messi	75000	FC Barcelona	Extremo derecho	Argentina
Eden	Hazard	63000	Real Madrid	Extremo izquierdo	Belgica
Marc-Andre	ter Stegen	60000	FC Barcelona	Arquero	Alemania

#	Time	Action	Message	Duration / Fetch
1	17:27:43	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	3 row(s) returned	0,00095 sec / 0,000...



# Booleano



## Antes de pasar al **AND** y al **OR**

Hay que tener en claro que son los booleanos y cómo funciona cada una de esas opciones.

Si usamos un **AND** se tienen que cumplir todas las condiciones.

Si usamos un **OR**, con que se cumpla una, ya nos va a devolver datos la query.

A	B	A and B
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	FALSO
FALSO	VERDADERO	FALSO
FALSO	FALSO	FALSO

A	B	A or B
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	VERDADERO
FALSO	VERDADERO	VERDADERO
FALSO	FALSO	FALSO



**And**





## And

Nos permite usarlo junto al WHERE para realizar condiciones múltiples, donde todas las condiciones tienen que cumplirse para que la query nos devuelva datos

Sintaxis:

```
SELECT * FROM <nombre_de_tabla> WHERE <valor_columna> = '<valor>'  
AND <valor_columna> = '<valor>' ...
```



## And

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE e.detalle_equipo = 'FC Barcelona' AND p.detalle_posicion = 'Arquero'
```



## And

Si bien Messi juega en el Barcelona la query no me lo trae porque solo cumple con una de las condiciones, no es arquero

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Marc-Andre	ter Stegen	60000	FC Barcelona	Arquero	Alemania

#	Time	Action	Message	Duration / Fetch
1	18:41:49	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	1 row(s) returned	0,00078 sec / 0,000...



Or



## Or

Nos permite usarlo junto al WHERE para realizar condiciones múltiples, donde por lo menos una de las condiciones tiene que cumplirse para que la query nos devuelva datos

Sintaxis:

```
SELECT * FROM <nombre_de_tabla> WHERE <valor_columna> = '<valor>'  
OR <valor_columna> = '<valor>' ...
```



Or

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE e.detalle_equipo = 'FC Barcelona' OR p.detalle_posicion = 'Arquero'
```



## Or

Fijense que ahora me trae 3 jugadores porque cumplen por lo menos una de las condiciones:

- Messi juega en el **Barcelona** (No es arquero)
- Oblack es **Arquero** (No juega en el Barcelona)
- Ter Stegen es **Arquero** y juega en el **Barcelona**

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Lionel Andres	Messi	75000	FC Barcelona	Extremo derecho	Argentina
Jan	Oblak	65000	Atletico Madrid	Arquero	Eslovenia
Marc-Andre	ter Stegen	60000	FC Barcelona	Arquero	Alemania

#	Time	Action	Message	Duration / Fetch
1	18:43:55	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	3 row(s) returned	0,00097 sec / 0,000...



# Between





## Between

El between nos permite filtrar datos dentro de un rango, montos, números, fechas, etc.

Sintaxis:

```
SELECT * FROM <nombre_de_tabla> WHERE <valor_columna> BETWEEN <valor> AND <valor>
```



## Between

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE ju.sueldo_mensual BETWEEN 60000 and 70000
```



## Between

Por ejemplo traer todos los jugadores que cobran entre 60.000 y 70.000 mensuales.

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Cristiano	Ronaldo	70000	Juventus	Extremo izquierdo	Portugal
Neymar	Junior	68000	Paris Saint-Germain	Extremo izquierdo	Brasil
Jan	Oblak	65000	Atletico Madrid	Arquero	Eslovenia
Eden	Hazard	63000	Real Madrid	Extremo izquierdo	Belgica
Marc-Andre	ter Stegen	60000	FC Barcelona	Arquero	Alemania

#	Time	Action	Message	Duration / Fetch
1	18:50:04	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	5 row(s) returned	0,00036 sec / 0,000...



# Not Between



## Not Between

Podría agregar una negación con el **NOT** para que me traiga a todos los jugadores que no cobran entre 60.000 y 70.000

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE ju.sueldo_mensual NOT BETWEEN 60000 and 70000
```



## Not Between

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Lionel Andres	Messi	75000	FC Barcelona	Extremo derecho	Argentina
Virgil	Van Dijk	58000	Liverpool	Defensa central	Holanda
Paulo	Dybala	55000	Juventus	Medio centro ofensivo	Argentina

#	Time	Action	Message	Duration / Fetch
1	18:53:15	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	3 row(s) returned	0,00049 sec / 0,000.



## Not Between o un where + un or con dos condiciones

Puedo lograr el mismo resultado de esta forma:

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE ju.sueldo_mensual < 60000 or ju.sueldo_mensual > 70000
```



## Not Between o un where + un or con dos condiciones

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Lionel Andres	Messi	75000	FC Barcelona	Extremo derecho	Argentina
Virgil	Van Dijk	58000	Liverpool	Defensa central	Holanda
Paulo	Dybala	55000	Juventus	Medio centro ofensivo	Argentina

#	Time	Action	Message	Duration / Fetch
1	18:56:36	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	3 row(s) returned	0,00084 sec / 0,000..





# Distinct



## Distinct

Nos permite traer valores sin duplicar.

Sintaxis:

```
SELECT DISTINCT <columna> FROM <nombre_de_la_tabla>
```



## Distinct

Por ejemplo si consulto solo los equipos que hay relacionados con los jugadores me devuelve esto:

```
SELECT e.detalle_equipo  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo
```

detalle_equipo			
FC Barcelona			
FC Barcelona			
Juventus			
Juventus			
Paris Saint-Germain			
Atletico Madrid			
Real Madrid			
Liverpool			
#	Time	Action	Duration / Fetch
1	19:08:05	SELECT e.detalle_equipo FROM jugador ju INNER JOIN equipo ...	8 row(s) returned 0,00040 sec / 0,000.



## Distinct

En cambio si agrego el DISTINCT

```
SELECT DISTINCT e.detalle_equipo  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo
```

detalle_equipo			
FC Barcelona			
Juventus			
Paris Saint-Germain			
Atletico Madrid			
Real Madrid			
Liverpool			

#	Time	Action	Message	Duration / Fetch
1	19:10:45	SELECT DISTINCT e.detalle_equipo FROM jugador ju INNER JOI...	6 row(s) returned	0,00085 sec / 0,000...



# Count



## Count

El count permite contar cuantas veces se repite algo

Sintaxis:

```
SELECT COUNT(<columna>) as <alias> FROM <nombre_de_la_tabla> WHERE <condición>
```



## Count

Por ejemplo cuántos jugadores de Argentina tengo cargados

```
SELECT count(n.detalle_nacionalidad) as total_Argentina
FROM jugador ju
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad
where n.detalle_nacionalidad = 'Argentina'
```



## Count

total\_Argentina

2

#	Time	Action	Message	Duration / Fetch
1	19:17:37	SELECT count(n.detalle_nacionalidad) as total_Argentina FRO...	1 row(s) returned	0,00035 sec / 0,000..





Max



## Max

Nos permite devolver el valor máximo de una columna puede usarse para mostrar el valor o ser usado como una condición en el where

Sintaxis:

```
SELECT MAX(<columna>) as <alias> FROM <nombre_de_la_tabla>
```

```
SELECT * FROM <nombre_de_la_tabla> WHERE <columna> = (SELECT MAX(<columna>) as  
<alias> FROM <nombre_de_la_tabla>)
```



## Max

```
SELECT MAX(sueldo_mensual) as sueldo_mas_alto FROM jugador
```

sueldo\_mas\_alto

75000

#	Time	Action	Message	Duration / Fetch
1	19:24:17	SELECT MAX(sueldo_mensual) as sueldo_mas_alto FROM jugador...	1 row(s) returned	0,00022 sec / 0,000...



## Max

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE ju.sueldo_mensual = (SELECT MAX(sueldo_mensual) as sueldo_mas_alto FROM jugador)
```



# Max

nombre	apellido	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Lionel Andres	Messi	75000	FC Barcelona	Extremo derecho	Argentina

#	Time	Action	Message	Duration / Fetch
1	19:25:15	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	1 row(s) returned	0,00058 sec / 0,000...



Min



## Min

Nos permite devolver el valor mínimo de una columna puede usarse para mostrar el valor o ser usado como una condición en el where

Sintaxis:

```
SELECT MIN(<columna>) as <alias> FROM <nombre_de_la_tabla>
```

```
SELECT * FROM <nombre_de_la_tabla> WHERE <columna> = (SELECT MIN(<columna>) as  
<alias> FROM <nombre_de_la_tabla>)
```



## Min

```
SELECT MIN(sueldo_mensual) as sueldo_mas_alto FROM jugador
```

suelo\_mas\_alto

55000

#	Time	Action	Message	Duration / Fetch
1	19:38:21	SELECT MIN(sueldo_mensual) as sueldo_mas_alto FROM jugad...	1 row(s) returned	0,00026 sec / 0,000...





## Min

```
SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_equipo, p.detalle_posicion,  
n.detalle_nacionalidad  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
INNER JOIN posicion p on p.id_posicion = ju.posicion_id_posicion  
INNER JOIN nacionalidad n on n.id_nacionalidad = ju.nacionalidad_id_nacionalidad  
WHERE ju.sueldo_mensual = (SELECT MIN(sueldo_mensual) as sueldo_mas_alto FROM jugador)
```



# Min

nombre	apellidc	sueldo_mensual	detalle_equipo	detalle_posicion	detalle_nacionalidad
Paulo	Dybala	55000	Juventus	Medio centro ofensivo	Argentina

#	Time	Action	Message	Duration / Fetch
1	19:39:30	SELECT ju.nombre, ju.apellido, ju.sueldo_mensual, e.detalle_e...	1 row(s) returned	0,00045 sec / 0,000...



# Order by



## Order by

Nos permite ordenar los valores según columna/s de forma ascendente o descendente

Sintaxis:

```
SELECT * FROM <nombre_de_tabla> ORDER BY <columna> ASC
```

```
SELECT * FROM <nombre_de_tabla> ORDER BY <columna> DESC
```



## Order by ASC

Por ejemplo ordenar los detalle\_equipo de la tabla equipo

**SELECT detalle\_equipo FROM equipo ORDER BY detalle\_equipo ASC**

detalle\_equipo

Atletico Madrid
Boca Juniors
FC Barcelona
Juventus
Liverpool
Manchester City
Paris Saint-Germain
Real Madrid
River Plate

#	Time	Action	Message	Duration / Fetch
1	19:46:42	SELECT detalle_equipo FROM equipo ORDER BY detalle_equipo...	9 row(s) returned	0,00054 sec / 0,000...



## Order by DESC

Por ejemplo ordenar los detalle\_equipo de la tabla equipo

**SELECT** detalle\_equipo **FROM** equipo **ORDER BY** detalle\_equipo **DESC**

detalle\_equipo

River Plate  
Real Madrid  
Paris Saint-Germain  
Manchester City  
Liverpool  
Juventus  
FC Barcelona  
Boca Juniors  
Atletico Madrid

#	Time	Action	Message	Duration / Fetch
1	19:48:44	SELECT detalle_equipo FROM equipo ORDER BY detalle_equipo...	9 row(s) returned	0,00041 sec / 0,000...



# Group by



## Group by

La sentencia group by sirve para agrupar datos, según un determinado criterio, y aplicar operaciones sobre los elementos del conjunto conformado.

Sintaxis:

```
SELECT * FROM <nombre_de_tabla> GROUP BY <condición>
```





## Group by

Supongamos que queremos que una consulta nos devuelva, todo el listado de equipos (sin repetidos) y una cuenta de cuantos jugadores hay por cada equipo, en este caso tengo que agrupar los datos por el nombre del equipo

```
SELECT e.detalle_equipo, count(e.detalle_equipo) as total
FROM jugador ju
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo
GROUP BY e.detalle_equipo
```



## Group by

detalle_equipo	tota
Atletico Madrid	1
FC Barcelona	2
Juventus	2
Liverpool	1
Paris Saint-Germain	1
Real Madrid	1

#	Time	Action	Message	Duration / Fetch
1	19:54:28	SELECT e.detalle_equipo, count(e.detalle_equipo) as total FRO...	6 row(s) returned	0,00079 sec / 0,000...



## Group by

Puedo jugar con el ORDER BY, por ejemplo, quiero el mismo listado pero que me ordene los nombres de los equipos de menor a mayor (A -Z) y la cuenta de jugadores de mayor a menor

```
SELECT e.detalle_equipo, count(e.detalle_equipo) as total
FROM jugador ju
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo
GROUP BY e.detalle_equipo
ORDER BY total DESC, detalle_equipo ASC
```



## Group by

detalle_equipo	tota
FC Barcelona	2
Juventus	2
Atletico Madrid	1
Liverpool	1
Paris Saint-Germain	1
Real Madrid	1

#	Time	Action	Message	Duration / Fetch
1	19:56:15	SELECT e.detalle_equipo, count(e.detalle_equipo) as total FRO...	6 row(s) returned	0,00075 sec / 0,000.



# Having



## Having

El having sirve como una condición de filtro para los datos que fueron agrupados

Sintaxis:

```
SELECT <columnas>, <columna>... FROM <nombre_de_tabla>  
GROUP BY <columna_que_agrupa>  
HAVING <filtro_sobre_los_datos_agrupados>
```



## Having

```
SELECT e.detalle_equipo, count(e.detalle_equipo) as total  
FROM jugador ju  
INNER JOIN equipo e on e.id_equipo = ju.equipo_id_equipo  
GROUP BY e.detalle_equipo  
HAVING total > 1  
ORDER BY total DESC, detalle_equipo ASC
```



## Having

Me devuelve solo dos valores, porque primero los agrupó, aplicó el filtro sobre el grupo (cuenta > 1) y luego los ordeno

detalle_equipo	tota
FC Barcelona	2
Juventus	2

#	Time	Action	Message	Duration / Fetch
1	20:06:41	SELECT e.detalle_equipo, count(e.detalle_equipo) as total FRO...	2 row(s) returned	0,00031 sec / 0,000...





## Para la próxima clase

- Releer la diapositiva y el material
- La próxima finalizamos con los temas “más complejos” pero los más útiles

