

LAS CLÁUSULAS

SELECT	Selecciona los campos que nos interesa obtener
FROM	Selecciona las tablas de donde queremos obtener los datos
WHERE	Pone condiciones o relaciona las multitas
GROUP BY	Agrupar en grupos
HAVING	Pone condiciones al grupo
ORDER BY	Ordena la consulta por un campo determinado de forma ASC(ascendente) o DESC (descendente)

COUNT	Cuenta registros de una columna
MAX	Máximo valor de una columna
MIN	Mínimo valor de una columna
AVG	Valor medio de una columna
ROUND(número, decimales)	Redondear decimales

IMPORTANTE:

- ❑ Todas las **subconsultas** se hacen dentro del **WHERE**
- ❑ En caso de utilizar **join** la relación se hace en el **FROM**

EL ALIAS

Select count(*) AS Num_empleados

Select count(*) Num_empleados

Cualquiera de las opciones son válidas. Son lo mismo.

Ejemplos de la instrucción SELECT

Saca un listado con todos los datos

→ Select *

Saca un listado con los campos determinados

→ Select campo1, campo2, campo3...

Saca el número de tuplas de un campo

→ Select count(*)

Saca un listado de los nombres de empleados sin duplicados

→ Select distinct nombre from empleados

Saca el número de tuplas de un campo sin duplicados

→ select count(distinct(Nombre_equipo)) from jugadores;

Saca un listado con los empleados que **no tienen** fecha de nacimiento

→ SELECT *

```
FROM empleados
WHERE fecha_nac IS NULL;
```

Saca un listado con los empleados que **si tienen** fecha de nacimiento

```
→ SELECT *
    FROM empleados
    WHERE fecha_nac IS NOT NULL;
```

Atributos del WHERE:

BETWEEN: selecciona las tuplas entre dos valores

```
SELECT * FROM jugadores WHERE peso BETWEEN 50 AND 70;
SELECT (campos) FROM (tabla) WHERE (campo) BETWEEN (valor1) AND (valor2);
```

Sin usar BETWEEN

Tenistas que pesan mayor o igual a 80 y miden menos de 180

```
→ SELECT * FROM tenistas WHERE peso >= 80 AND altura < 180;
```

Saca los salarios comprendidos entre >=1000 y <=2000

```
→ WHERE salario >=1000 AND salario <=2000
```

Usando BETWEEN

Saca los salarios comprendidos entre >=1000 y <=2000

```
→ WHERE salario BETWEEN 1000 AND 2000
```

LIKE: filtra las tuplas

```
SELECT * FROM empleados WHERE nombre LIKE 'Juan %';
SELECT (campos) FROM (tabla) WHERE (campo) LIKE '(caracteres + % + _)';
```

Lo substituye por una cadena de caracteres > %

Lo substituye por un carácter > _

Sin usar LIKE

sintaxis: campo='valor'

```
→ where nombre='jose'
```

Usando LIKE

Saca todos los nombres que empiecen por: jo

```
→ where nombre like 'jo%'
```

Saca todos los nombres que acaben por: an

```
→ where nombre like '%an'
```

Saca todos los nombres compuestos

```
→ where nombre like 'jose %'
```

Saca todos los nombres que empiezan por "ma" de 5 letras

→ `where nombre like 'ma__'`

Encuentra una subcadena dentro de una cadena

→ `where nombre like '%jose%'`

IN: Permite poner un campo con varias condiciones

```
SELECT * FROM empleados WHERE ciudad IN ('Madrid', 'Barcelona',  
'Valencia', 'Sevilla');
```

```
SELECT (seleccion) FROM (tabla) WHERE (campo) IN ('valor1', 'valor2', 'valor3')
```

Sin usar IN

```
SELECT nombre, departamentos  
FROM empleados  
WHERE departamento=1 OR departamento=2 OR departamento=3  
ORDER BY nombre;
```

Usando IN

```
SELECT nombre, departamento  
FROM empleados  
WHERE departamento IN(2,5,8)  
ORDER BY nombre;
```

Algunas consultas se pueden resolver tanto por **subconsultas** como por **multitablas**
Aunque existen consultas que solo se pueden resolver por **subconsultas**

Estas dos consultas son la misma utilizando los dos métodos.

Subconsulta

Salarios más altos que pepe el propietario

```
select e.nombre, salario, d.nombre  
from empleados e, departamentos d  
where e.departamento=d.iddepartamento and salario >  
      (select salario from empleados  
       where nombre="pepe el propietario")  
order by salario;
```

Multitabla

Salarios más altos que pepe el propietario

```
select e1.nombre, e1.salario  
from empleados e1, empleados e2  
where e1.salario > e2.salario and e2.nombre="pepe el propietario";
```

Detalles a tener en cuenta:

- ❑ Los datos que podemos obtener con un **GROUP BY**
Son por ejemplo: la media del salario, el salario máximo, el salario mínimo.
Es decir, las condiciones que afectan al grupo.
En un **GROUP BY** no podemos obtener el nombre de un empleado, ya que estaría cogiendo al primero de la tabla.
- ❑ **Diferencia** entre **count(*)** y **count(columna)**
count(*): número de tuplas seleccionadas, incluyen los nulls.
count(columna): número de tuplas seleccionadas que tiene valor en esa columna, no incluyen los nulls.
- ❑ Dentro de la instrucción **SELECT** se pueden utilizar **expresiones**, en particular las **aritméticas**. También se usan en las cláusulas **WHERE**.

```
SELECT precio+(precio*iva) AS precio_netto FROM productos;
```

Para añadir: (En proceso)

operadores relacionales > >= < <= <>

operadores matemáticos + - * /

operadores lógicos: AND OR NOT

HAVING count(*) BETWEEN 200 AND 500 --> quedate con los grupos con la cantidad de tuplas entre 200 y 500

sum: obtener el valor de suma de las tuplas seleccionadas

avg(columna): valor medio de las tuplas seleccionadas

max(columna): el número más alto

min(columna): el número más bajo

Exemplo con HAVING

```
SELECT "nombre1_columna", SUM("nombre2_columna")  
FROM "nombre_tabla"  
GROUP BY "nombre1_columna"  
HAVING (condición de función aritmética)
```

Exemplos de ejercicios de consultas

/*1 Listado de ciudades que tengan algún equipo, sin duplicados.*/

```
SELECT distinct ciudad  
FROM equipos;
```

```
select distinct ciudad, nombre  
from equipos;
```

```
select DISTINCT ciudad from equipos where ciudad is not null
```

/*2 Selecciona los jugadores cuyo peso sea inferior a 270 y superior a 300 libras, con operador between, ordenado primero descendente por peso y ascendentemente por nombre.*/

```
USE NBA;  
SELECT *  
FROM jugadores  
WHERE peso not BETWEEN 270 and 300  
ORDER BY peso desc, nombre asc;
```

3 Jugadores de los Hornets.*/

```
SELECT nombre  
FROM jugadores  
WHERE nombre_equipo='hornets';
```

4 Listar jugadores que se llamen Mark

```
select Nombre from jugadores where nombre like 'Mark %'
```

/*5 Listado de jugadores en los que la segunda letra de su nombre sea 'a'.*/

```
SELECT *  
FROM jugadores  
WHERE nombre LIKE '_a%';
```

/*6 Listado de jugadores que jueguen en la posición G.*/

```
Select nombre from jugadores where posicion Like '%G%'
```

/*7 Listado de jugadores con nombres de cuatro caracteres, nombres con exactamente cuatro caracteres.*/

```
select nombre from jugadores where nombre like '____ %';
```

example

```
select *from jugadoreswhere procedencia='kansas' and substring_index(altura,'-',1)*30.48 +
substring_index(altura,'-',-1)*2.54>200;
```

example

8 Listado de jugadores de los Celtics, con sus nombres, altura en cm y peso en kg.
Indica en el encabezado de cada columna la unidad de medida correspondiente.*/

Incorrecto

```
SELECT nombre,altura as altura_cm,(peso*0.4535) as peso_kg
FROM jugadores;
```

Correcto pero inútil

```
select nombre,altura, substring_index(altura,'-',1)*30.48 +
substring_index(altura,'-',-1)*2.54 AltCm, (peso*0.4535)
from jugadores;
```

9 Listado de jugadores procedentes de Kansas cuya altura sea superior a 200 cm.

11 Datos del jugador más bajo, con su altura en cms.

12Listado de conferencias y divisiones, sin duplicados y ordenado por conferencia y división.

```
SELECT distinct conferencia,division
FROM equipos;
```

13 Listado de las diferentes procedencias de los jugadores.

```
SELECT distinct procedencia
FROM jugadores;
```

```
select nombre, Procedencia from jugadores
```

14 Listado de equipos que tienen jugadores procedentes de Arizona.

```
select DISTINCT nombre_equipo
from jugadores
where Procedencia='Arizona';
```

Numero de jugadores en la liga
`select count(*) from jugadores;`

`SELECT COUNT(codigo)
FROM jugadores;`

`select count(*) NumJugadoresLiga
from jugadores;`

Peso medio, peso máximo y peso mínimo de los jugadores de la liga
`SELECT MAX(peso),MIN(peso)
FROM jugadores;`

Peso máximo en kilos, 1 Libra=0.4535Kg
`select max(peso*0.45) as pesoMaximoenKG
from jugadores;`

Peso máximo en kilos, redondeando a dos decimales.
`select round(max(peso*0.45),2) as MaxKilos
from jugadores;`

Listado de las diferentes procedencias que hay en la tabla de jugadores
`SELECT distinct procedencia
FROM jugadores;`

¿Cuántos jugadores tienen procedencia conocida (no nula)?
`SELECT COUNT(procedencia)
FROM jugadores
WHERE procedencia is not null;`

Cuántas procedencias hay?
`SELECT COUNT(DISTINCT(procedencia))
FROM jugadores;`

Número de jugadores que juegan en la posición F
`select count(*) jugadoresjuegan from jugadores where posicion like '%F%';`

Listado de jugadores que juegan en la posicion F
`select nombre from jugadores where posicion like '%F';`

Numero de jugadores de los Celtics que juegan en la posicion C
`SELECT COUNT(*)
FROM jugadores
WHERE nombre_equipo='celtics' and posición LIKE '%c%';`

```
select count(nombre) jugCeltics from jugadores where nombre_equipo='Celtics' AND posicion like '%C%'
```

Listado con el nombre de equipo y el número de jugadores de cada equipo

```
SELECT nombre_equipo,COUNT(codigo)
FROM jugadores
GROUP BY nombre_equipo;
```

```
select count(nombre),nombre_equipo from jugadores group by nombre_equipo;
```

Listado con el número de jugadores por procedencia

```
SELECT procedencia,COUNT(codigo)
FROM jugadores
GROUP BY procedencia;
```

Listado con el número de jugadores por procedencia, procedencia no nula

```
SELECT procedencia,COUNT(codigo)
FROM jugadores
WHERE procedencia IS NOT NULL
GROUP BY procedencia;
```

Listado con el peso medio de cada equipo en Kg

```
SELECT nombre_equipo,AVG(peso*0.4535)
FROM jugadores
GROUP BY nombre_equipo;
```

Para cada equipo: peso medio redondeado a dos decimales y numero de jugadores

```
SELECT nombre_equipo,ROUND(AVG(peso*0.4535),2) AS peso_medio,COUNT(*) AS Num_jugadores
FROM jugadores
GROUP BY nombre_equipo;
```

17 - Listado con el número de equipos por ciudad. Listado de ciudades y el número de equipos de cada ciudad

```
SELECT COUNT(nombre) as 'Equipos por ciudad',ciudad
FROM equipos
GROUP BY ciudad;
```

18 Listado con el número de equipos por ciudad para aquellas ciudades que tengan más de un equipo

```
SELECT count(*),ciudad
FROM equipos
GROUP BY ciudad
HAVING count(*) > 1;
```

nop


```
SELECT COUNT(nombre) as 'Equipos por ciudad',ciudad
FROM equipos
WHERE COUNT(nombre) > 1
GROUP BY ciudad;
```

```
select count(nombre) as equiposCiudad, ciudad
from equipos
group by ciudad
having count(*) > 1;
```

19 Listado con el numero de equipos por conferencia y division y el total por conferencia

```
SELECT count(nombre),conferencia,division
FROM equipos
GROUP BY conferencia,division;
```

```
select conferencia, division, count(*)
from equipos
group by conferencia, division
union
select conferencia, '-' , count(*)
from equipos
group by conferencia;
```

```
select count(nombre), conferencia, division
from equipos
group by conferencia, division
UNION
select count(nombre), conferencia, '-'
from equipos
group by conferencia;
```

Jugadores por equipo que juegan en la posicion F

```
SELECT nombre,nombre_equipo,posicion
FROM jugadores
WHERE posicion like '%f%'
GROUP BY nombre_equipo;
```

Jugadores por equipo que juegan en la posicion F
Pero solo de los equipos que comiencen por la letra B

```
SELECT nombre,nombre_equipo,posicion
FROM jugadores
WHERE posicion like '%f%'
GROUP BY nombre_equipo
HAVING nombre_equipo like 'b%';
```

Jugadores por equipo que juegan en la posicion F

Pero solo para aquellos equipos donde haya más de 7 jugadores en esa posición

```
SELECT COUNT(*),nombre_equipo
FROM jugadores
WHERE posicion like '%f%'
GROUP BY nombre_equipo
HAVING COUNT(nombre)>7;
```

/*Listado de jugadores: nombre de jugador, ciudad del equipo donde juega*/

```
SELECT j.nombre,e.ciudad
FROM equipos e,jugadores j
WHERE e.nombre=j.nombre_equipo;
```

Listado de jugadores de la conferencia Este: nombre de jugador, ciudad.

```
SELECT j.nombre,e.ciudad
FROM jugadores j,equipos e
WHERE e.nombre=j.nombre_equipo AND e.conferencia='east';
```

Listado de jugadores de la división atlántica de la conferencia este: nombre de jugador, ciudad

```
select j.nombre, ciudad, e.conferencia
from jugadores j, equipos e
where division='atlantic' and conferencia='east' and e.nombre=j.nombre_equipo;
```

```
SELECT j.nombre,e.ciudad
FROM jugadores j,equipos e
WHERE e.nombre=j.nombre_equipo and e.division='atlantic' AND e.conferencia='east';
```

Listado de jugadores de la division Atlantic y de la division Pacific

```
select j.*
from jugadores j, equipos e
where j.nombre_equipo=e.nombre and division in ('atlantic','pacific');
```

```
SELECT j.nombre
FROM jugadores j,equipos e
WHERE e.nombre=j.nombre_equipo and (e.division='atlantic' or division='pacific');
```

Listado de jugadores: nombre de jugador, temporada, media de puntos por partido

```
SELECT j.nombre,e.temporada,puntos_por_partido
FROM jugadores j,estadisticas e
WHERE j.codigo=e.jugador;
```

Listado con la siguiente información: Conferencia, numero de jugadores

```
select conferencia, count(codigo)
```

```
from equipos e, jugadores j
where e.nombre=j.nombre_equipo
group by conferencia;
```

```
SELECT e.conferencia,COUNT(j.nombre)
FROM jugadores j,equipos e
WHERE j.nombre_equipo=e.nombre
GROUP BY e.conferencia;
```

Peso medio de los jugadores por Conferencia y Division

```
select conferencia, division, avg(peso)
from jugadores j, equipos e
where j.nombre_equipo=e.nombre
group by conferencia, division
```

```
SELECT conferencia,division,AVG(peso)
FROM jugadores j,equipos e
WHERE j.nombre_equipo=e.nombre
GROUP BY conferencia,division;
```

Listado con la siguiente información: Media de las medias de los puntos por partido de cada jugador

```
SELECT AVG(puntos_por_partido)
FROM estadisticas
GROUP BY jugador;
```

```
select avg(puntos_por_partido), j.nombre
from jugadores j, estadisticas e
where j.codigo=e.jugador
group by j.codigo;
```

```
select nombre, avg(puntos_por_partido)
from jugadores j, estadisticas e
where j.codigo=e.jugador
group by e.jugador;
```

SUBCONSULTAS MALVADAS

http://www.mundoracle.com/subconsultas.html?Pg=sql_plsql_6.htm

Subconsultas Mono-registro

- Devuelven un único registro.
- Se utilizan operadores de comparación (=, >, >=, <, <= y <>).
- Ejemplo:

```
SQL> SELECT ename, job
      FROM emp
      WHERE job = (SELECT job FROM EMP
                   WHERE empno = 7369) → CLERK
      AND    sal > (SELECT sal FROM EMP
                   WHERE empno = 7876); → 1100
```

S.Q.L.

Subconsultas Multi-registro

- Devuelven más de un registro
- Se utilizan comparadores multiregistro:
 - IN → TRUE si se encuentra en la lista.
 - ANY (y sinónimo SOME) → TRUE si la condición se cumple con algún registro de la lista devuelta por la subconsulta.
 - ALL → TRUE si la condición se cumple con todos los registros de la lista devuelta por la subconsulta.
- El operador NOT puede ser utilizado con los operadores IN, ANY y ALL.

```
1 /*empleados que trabajan en el departamento direccion*/
```

```
select nombre
from empleados
where departamento = (select iddepartamento
                      from departamentos
```

```
where nombre='direccion');
```

```
2 /*empleados que ganan más dinero que pepe el propietario*/
```

```
SELECT nombre
FROM empleados
WHERE salario > (SELECT salario
                  FROM empleados
                  WHERE nombre='pepe el propietario');
```

```
3 /*Empleados que trabajan en el departamento dirección y administración*/
```

```
select *
from empleados
where departamento in (select iddepartamento
                        from departamentos
                        where nombre='direccion' or nombre='administracion');
```

```
SELECT *
FROM empleados
WHERE departamento in (SELECT iddepartamento
                        FROM departamentos
                        WHERE nombre='direccion' or
                        nombre='administracion');
```

```
4 /* contrario al anterior */
```

```
select *
from empleados
where departamento not in (select iddepartamento
                            from departamentos
                            where nombre='direccion' or nombre='administracion');
```

```
5 Empleados que ganan mais ou igual que a media
```

```
SELECT nombre
FROM empleados
WHERE salario >=(SELECT AVG(salario)
                  FROM empleados);
```

```
6 empleados que trabajan en el mismo departamento que pepe el propitario
```

```
SELECT nombre
FROM empleados
WHERE departamento=(SELECT departamento
                     FROM empleados
                     WHERE nombre like 'maria%');
```

```
* Modificar/actualizar el telefono del empleado on id 1 a 986986986 */
```

```
update empleados
```

```
set telefono='986986986'  
where idEmpleado=1;
```

http://www.w3schools.com/sql/sql_insert.asp

```
UPDATE table_name  
SET column1=value1,column2=value2,...  
WHERE some_column=some_value;
```

```
UPDATE Customers  
SET ContactName='Alfred Schmidt', City='Hamburg'  
WHERE CustomerName='Alfreds Futterkiste';
```

```
/* Enunciado 1 el empleado con idempleado 1 cobrará 400 */  
UPDATE empleados  
SET salario=400  
WHERE idempleado=1;
```

```
/* Enunciado 2 incrementar el salario del empleado 1 en 3 euros */  
update empleados  
set salario=salario+3  
where idempleado=1;
```

```
UPDATE empleados  
SET salario=salario+3  
WHERE idempleado=1;
```

```
/* Enunciado 3 incrementar el salario del empleado 1 un 3% */  
update empleados  
set salario=salario*1.03  
where idempleado=1
```

```
CALCULAR EL RESTO DE 10/3  
select 10%3;
```

```
/* Enunciado 4 incrementar un 10% el salario a aquellos empleados  
cuyo salario sea inferior a la media salarial de la empresa */  
UPDATE empleados  
SET salario=salario*1.10  
WHERE salario<(SELECT AVG(salario)  
FROM empleados);
```

```
update empleados
set salario=salario*1.02
where departamento = (select iddepartamento
                        from departamentos
                        where nombre='produccion');
```

```
select e.nombre, salario, salario*1.02 salarioIncrementado
from empleados e join departamentos on departamento=iddepartamento
where departamentos.nombre='produccion';
```

```
show tables;
```

```
select e.nombre NombreEmpleado, count(proyecto) NumProyectos
from empleados e, proyectos p, empleados_proyectos ep
where ep.empleado=e.idempleado and p.idproyecto=p.idProyecto
group by idempleado;
```

```
/*Mostramos la fecha y hora actuales*/
select now();
select curdate();
select current_time();
create table departamentos2 as
select * from departamentos;
select * from departamentos2;
/*actualizar fecha de director en departamento de extranjero con la fecha actual*/
update departamentos2
```

```
set fechaDirector=curdate()
where fechaDirector is null;
```

```
select dayofmonth(now()) dia, month(now()) mes, year(now()) anho;
select weekday(now());
```

```
select month(now()), minute(now()), second(now());
```

```
select date_add(now(), interval 1 day);
select date_add(curdate(), interval 1 day);
select date_sub(curdate(), interval 1 day);
select now()+interval 1 day;
```

```
select date_add(date_add(curdate(), interval 1 year), interval 1 day);
select curdate()+interval 1 year + interval 1 day;
```

```
select nombre, fecha_nac, datediff(curdate(),fecha_nac) diasNacido
from empleados;
```

```
select curdate(), date_format(curdate(), '%d-%m-%y');
```

```
select nombre, lower(nombre) from empleados;
```

```
select concat(nombre, '---->', nif, '---->', direccion)
from empleados;
```

```
select concat_ws('---->', nombre, nif, direccion)
from empleados;
```

```
select nombre, right(nombre,2),left(nombre,2), substring(nombre,4),
substring(nombre,4,2)
from empleados;
```

```
select minute(now()), second(now());
```

```
update empleados //ACTUALIZA EMPLEADOS
set departamento=(select iddepartamento //pon en departamento ventas
                  from departamentos
                  where nombre='ventas')
where nombre in ('ana','pepiño'); //cando se chaman ana e pepiño
```

```
update empleados //ACTUALIZA EMPLEADOS
set departamento=(select iddepartamento //pon en departamento ventas
                  from departamentos
```



```
        where nombre='ventas')  
where idempleado in (10,11,12,15,16); //Cando teñen estes id empleados
```

```
update empleados //aCTUALIZA EMPLEADOS  
set departamento=(select iddepartamento //pon en departamento ventas  
                   from departamentos  
                   where nombre='ventas')  
where idempleado >=10; //Cando o id de empleado e maior  
de 10
```