

## **Comandos clase online SQL**

Select

From

[where] → Opcional

```
select * from dept
```

```
desc dept
```

```
desc emp
```

```
select * from emp
```

```
select ename,job, sal from emp
```

```
select ename,job, sal
```

```
from emp
```

```
where deptno=20
```

## **Expresiones aritméticas:**

Operadores aritméticos

+, -, \*, /

```
select ename,job, sal*12
```

```
from emp
```

## **Encabezados de columna:**

Select "columna" "nombre a mostrar"

```
select ename nom_emp, job trabajo, sal*12 salario_anual
```

```
from emp
```

Obtener los empleados cuya comisión es mayor al 25% de su salario

```
select deptno, ename, job, sal
```

```
from emp
```

```
where comm >sal * .25
```

## **Múltiples condiciones**

```
select deptno, ename, job, sal
```

```
from emp
```

```
where sal >= 500
```

```
and job='CLERK'
```

```
select deptno, ename, job, sal
```

```
from emp
```

```
where job not in ('PRESIDENT', 'MANAGER')
```

```
select deptno, ename, job, sal
```

```
from emp
```

where (deptno=10 or deptno=20) and job='MANAGER'

### **Operador between:**

Sintaxis

Where columna between valor\_min and valor\_max //inclusivo

Where columna not between valor\_min and valor\_max

```
select deptno, ename, job, sal
from emp
where sal between 1250 and 1600
```

```
select deptno, ename, job, sal
from emp
where sal not between 1250 and 1600
```

### **Valores Nulos**

Null significa ausencia de valor.

Sintaxis

Where columna is null

Where columna is not null

```
Select ename, job, comm from emp
Where comm is null
```

```
select deptno, ename, job, comm
from emp
where comm is not null
```

### **Operador Like**

Sintaxis

Where columna like 'patron'

Where columna not like 'patron'

Carácter	Descripción
_ (Underline)	Cualquier carácter sencillo
%	Cualquier secuencia de cero o más caracteres

```
Select ename from emp
Where ename like '%S'
```

```
Select ename
From emp
Where ename like '___R%'
```

```
select ename, deptno, job
from emp
where ename like '%A%I%'
```

```
Select deptno, dname, substr(dname,1,3) Code from dept
select deptno,lower(dname) from dept
select upper(ename), initcap(ename),ename from emp
ltrim
rtrim
Select deptno, length(dname) from dept
select ename,empno, job from emp
where mod(empno,2)=1
```

Round  
Trunc  
Power  
Nvl  
Add\_months  
Last\_day  
To\_char  
To\_date  
To\_number

Funciones agregadas o de agrupamiento  
select funcion\_agregada (atributo)  
from relacion

SUM  
AVG  
MIN  
MAX  
COUNT  
STDDEV  
VARIANCE

```
select sum(sal) from emp
select avg(sal) from emp
select min(sal) from emp
select count(ename) from emp
select count(job) from emp
group by job
select count(distinct job) from emp
select max(hiredate) from emp
select count(*) from emp
```

where deptno =20

Clausula group by  
select columnas\_del\_group\_by o func\_agregadas  
From tabla1,...,tablan  
Group by column1, column2

```
select deptno, count(*)  
from emp  
group by deptno
```

```
select deptno, sum(sal), avg(sal)  
from emp  
group by deptno
```

Contar la cantidad de personas que desempeña cada trabajo por depto

```
select deptno, job, count(*)  
from emp  
group by deptno, job
```

Obtener el promedio salarial para analistas y vendedores por depto

```
select deptno, avg(sal), count(*)  
from emp  
where job in ('ANALYST', 'SALESMAN')  
group by deptno
```

Clausula Having

```
select columnas_del_group_by o func_agregadas  
From tabla1,...,tablan  
Group by column1, column2  
having condicion
```

```
select deptno, job, count(*)  
from emp  
group by deptno, job  
having count(*) >=2
```

```
select deptno, job, count(*)  
from emp  
where job='ANALYST'  
group by deptno, job
```

Cuales son los deptos con por lo menos dos vendedores

```
select deptno, count(*)
```

```
from emp
where job='SALESMAN'
group by deptno
having count(*) >=2
```

Consultas anidadas

```
select atributo(s)
from relacion(es)
where columna = (select columna from relacion
                  [where condicion])
```

```
select ename, job
from emp
where job = (select job from emp
             where ename='JONES')
```

Nombre, trabajo y fecha\_cont de la primer persona contratada

```
select ename, job, hiredate
from emp
where hiredate = (select min(hiredate) from emp)
```

Nombre y trabajo de los empleados que trabajan en chicago

```
select * from emp
select * from dept
```

```
select ename, job
from emp
where deptno=(select deptno from dept
               where loc='CHICAGO')
```

Obtener nombre, trabajo y salario de los empleados del depto 20 que tengan el mismo trabajo que los empleados del depto 30

```
select ename, job, sal
from emp
where deptno =20
and job in (select job from emp
            where deptno=30)
```

Operador ANY - ALL

Cuando la subconsulta interna devuelve mas de un valor y necesitamos comparar usando los operadores de comparación

```
select columna(s)
from relacion
```

where columna operador\_comp ANY (subconsulta)

```
select distinct sal, ename, job, deptno
from emp
where sal > any (select sal from emp
                 where deptno=30)
```

```
select distinct sal, ename, job, deptno
from emp
where sal > all (select sal from emp
                 where deptno=30)
```

Obtener nombre, trabajo y salario de los empleados del depto 20 que tienen el mismo trabajo que los empleados del depto de ventas

```
select ename, job, sal
from emp
where deptno=20
and job in (select job from emp
            where deptno = (select deptno from dept
                             where dname='SALES'))
```

### **Consultas correlacionadas**

La ejecución de una subconsulta se puede correlacionar con el valor de un campo en la consulta principal.

Obtener los empleados que ganan más que el promedio salarial de su propio departamento.

```
select ename, deptno, sal
from emp
where sal > (promedio salarial del depto al que pertenece el empleado)
```

```
select ename, deptno, sal
from emp X
where sal > (select avg(sal) from emp
             where deptno=X.deptno)
```

Empleados que ganan igual o más que el salario de su supervisor

```
select ename, sal, deptno
from emp X
where sal >= (select sal from emp
              where empno=X.mgr)
```

## **Operador Exists**

Select  
From  
Where exists (subconsulta)

Obtener nombre, trabajo, deptno de los empleados que tienen otros empleados a su cargo.

```
select empno, ename, job, deptno  
from emp X  
where exists (select * from emp  
              where mgr=X.empno)  
order by empno
```

Obtener los departamentos que tienen empleados

```
select deptno, dname, loc  
from dept d  
where exists (select deptno from emp e  
              where d.deptno=e.deptno)
```

## **JOINS**

Permite enlazar dos tablas por el atributo común. Al relacionar mas de una tabla en la cláusula from automáticamente se realiza un producto cartesiano

Obtener el nombre, trabajo, salario y departamento del empleado y nombre del departamento

```
select ename, job, sal, d.deptno, dname  
from emp e, dept d  
where e.deptno=d.deptno
```

```
select ename, job, sal, d.deptno, dname  
from emp e inner join dept d  
on e.deptno=d.deptno
```

Obtener nombre, trabajo, deptno y la ubicacion de todos los CLERK

```
select ename, job, e.deptno, loc  
from emp e, dept d  
where job='CLERK'  
and e.deptno=d.deptno
```

```
select ename, loc  
from emp e left join dept d  
on e.deptno=d.deptno
```

```
select ename, loc  
from emp e, dept d  
where e.deptno=d.deptno(+)
```

```
select ename, loc  
from emp e right join dept d  
on e.deptno=d.deptno
```

```
select ename, loc  
from emp e, dept d  
where e.deptno(+)=d.deptno
```

Self join

Obtener el nombre y trabajo de cada empleado con el nombre y trabajo de su supervisor

```
select e.ename, e.job, e.mgr, s.empno, s.ename, s.job  
from emp e, emp s  
where e.mgr=s.empno
```

Compatibilidad de unión

Deben tener el mismo numero de atributos y cada atributo por posicion debe tener el mismo dominio

Union - Interseccion - Diferencia

Union: Deben ser union compatible

Registros de la primer consulta más registros de la segunda consulta, eliminando los duplicados

Union all → no elimina duplicados

Obtener nombre, trabajo y deptno de los empleados CLERK con los empleados en el deptno 10

```
select ename, job, deptno  
from emp  
where job='CLERK'  
union  
select ename, job, deptno  
from emp  
where deptno=10
```

Interseccion: Union compatible

Registros comunes de ambas consultas

Obtener nombre, trabajo y deptno de los empleados CLERK con los empleados del depto 10



```
select ename, job, deptno
from emp
where job='CLERK'
intersect
select ename, job, deptno
from emp
where deptno=10
```

Diferencia

Registros que se encuentran en la primer consulta

```
select ename, job, deptno
from emp
where job='CLERK'
minus
select ename, job, deptno
from emp
where deptno=10
```

```
select ename, job, sal
from emp
where sal>2000
minus
select ename, job, sal
from emp
where job='MANAGER'
```

```
select deptno, job, count(*)
from emp
group by deptno, job
union
select deptno, null, count(*)
from emp
group by deptno
union
select null, null, count(*)
from emp
```

Sintaxis

Create view nombre\_vista as

```
CREATE VIEW sales_staff AS
  SELECT empno, ename, deptno
```

```
FROM emp
WHERE deptno = 10
```

```
select * from sales_staff
select * from emp
```

```
insert into sales_staff values (7888, 'CRIS',10)
```

```
drop view nombre_vista
```

```
create or replace view sales_staff as
SELECT empno, ename, deptno
  FROM emp
 WHERE deptno = 10
 with read only
```

```
select * from sales_staff
```

```
delete from sales_staff
where empno=7782
```

Transacciones  
Transactions

Permite agrupar todas las instrucciones en una sola unidad para garantizar la consistencia e integridad de los datos

Sintaxis

Commit [work] → Finaliza la transaccion haciendo los cambios permanentes

Rollback [work] → finaliza la transaccion deshaciendo los cambios

Savepoint → Establece un punto dentro de la transacción para retornar en caso de rollback

Show autocommit → Muestra si el commit se ejecuta o no inmediatamente después de cada cambio.

Set autocommit [on|off] Establece cuando ocurre el cambio

```
create table bonus as
  select ename, job, sal, comm
  from emp
 where job='SALESMAN'
```

```
show autocommit
```

```
set autocommit off
```

desc bonus

```
insert into bonus (ename, job, sal)
select ename, job, sal
from emp
where job='ANALYST'
or job='MANAGER'
```

select \* from bonus

commit work

savepoint a

```
delete from bonus
where ename in('BLAKE', 'SCOTT')
```

select \* from bonus

rollback a

```
select b.brand_id, b.brand_name, avg(p.list_price)
from products p, brands b
where model_year=2018
and p.brand_id=b.brand_id
group by b.brand_id, b.brand_name
```

