

Trabajamos QUERYS sobre esquema SCOTT:

1. Ver la tabla empleados:

```
SELECT *  
FROM EMP
```

2. Seleccionar nombre, oficio, sueldo y comisión de los empleados:

```
SELECT ename, job, sal, comm  
FROM EMP
```

Alias:

```
SELECT ename as empleado,  
       job as empleo,  
       sal as salario,  
       comm as comision  
FROM EMP
```

3. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo sueldo sea mayor que 1000:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE sal > 1000
```

4. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo sueldo sea mayor que 1000, ordenado por sueldo:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE sal > 1000  
ORDER BY sal
```

Por defecto ordena Ascendente.

5. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo sueldo sea mayor que 1000, ordenado por sueldo descendente:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE sal > 1000  
ORDER BY sal desc
```

6. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo oficio sea CLERK:

```
SELECT ename,  
       job,
```

```

        sal,
        comm
FROM EMP
WHERE job='CLERK'

```

7. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo oficio no sea CLERK:

```

SELECT ename,
        job,
        sal,
        comm
FROM EMP
WHERE job!='CLERK'

```

8. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo oficio no sea CLERK y su sueldo sea mayor que 1500:

```

SELECT ename,
        job,
        sal,
        comm
FROM EMP
WHERE job!='CLERK'
      AND sal > 1500

```

9. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) que tengan asignada alguna comisión:

```

SELECT ename,
        job,
        sal,
        comm
FROM EMP
WHERE comm is not null

```

¿Que vemos aqui?. Solo los SALESMAN

10. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) que no tengan asignada ninguna comisión:

```

SELECT ename,
        job,
        sal,
        comm
FROM EMP
WHERE comm is null

```

11. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo nombre empieza por A

```

SELECT ename,
        job,
        sal,
        comm
FROM EMP
WHERE ename LIKE 'A%'

```

12. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo nombre contenga una A en cualquier posición:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE ename LIKE '%A%'
```

13. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo nombre NO contenga una A:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE ename NOT LIKE '%A%'
```

14. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo nombre empieza por una vocal:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE ename LIKE 'A%'  
      OR ename LIKE 'E%'  
      OR ename LIKE 'I%'  
      OR ename LIKE 'O%'  
      OR ename LIKE 'U%'
```

15. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo sueldo este comprendido entre 1000 y 2000:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE sal >= 1000  
      AND sal <=2000
```

O:

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE sal between 1000 AND 2000
```

16. Hacer un listado de empleados (Nombre, trabajo, sueldo, comisión) cuyo sueldo sea igual a 1000,2000,3000,4000 o 5000

```
SELECT ename,  
       job,  
       sal,  
       comm  
FROM EMP  
WHERE sal=1000  
       OR sal=2000  
       OR sal=3000  
       OR sal=4000  
       OR sal=5000
```

17. Calcular el sueldo neto de cada empleado y su retención de IRPF sabiendo que esta es un 19% para el sueldo y un 15% para la comisión.Redondear a dos decimales:

```
SELECT ename,  
       job,  
       sal,  
       IFNULL(comm,0) Comision,  
       ROUND(sal*0.19+IFNULL(comm,0)*0.15, 2) Retencion_IRPF,  
       sal+IFNULL(comm,0)-ROUND(sal*0.19+IFNULL(comm,0)*0.15, 2)  
       Sueldo_Neto  
FROM EMP
```

18. Obtener el máximo Salario, el mínimo y la media de todos los empleados:

```
SELECT max(sal),  
       min(sal),  
       avg(sal)  
FROM EMP
```

19. Obtener el máximo Salario, el mínimo y la media de todos los empleados por departamentos:

```
SELECT DEPTNO,min(sal),max(sal),avg(sal)  
FROM EMP  
GROUP BY DEPTNO
```

20. Obtener el número de empleados por departamento ordenado de mayor a menor:

```
SELECT DEPTNO,count(*)  
FROM EMP  
GROUP BY DEPTNO
```

21. Hacer un listado de departamentos con su número de empleados si estos son mayores de 5:

```
SELECT DEPTNO,count(*) AS "NUN_EMPLEADOS"  
FROM EMP  
GROUP BY DEPTNO  
HAVING NUN_EMPLEADOS > 5  
ORDER BY count(*) DESC
```

22. Listar los oficios distintos que hay en la tabla de empleados:

```
SELECT distinct(job)
FROM EMP
```

23. Listado de oficios con el número de empleados que los desempeñan ordenados por el número de estos de mayor a menor:

```
SELECT job, count(*)
FROM EMP
GROUP BY job
ORDER BY 2 DESC
```

24. Listado con el número de oficios distintos que aparecen en la tabla EMP:

```
SELECT job, count(*)
FROM EMP
GROUP BY job
ORDER BY 2 DESC
```

25. Listado con el tiempo que lleva contratado cada trabajador en la empresa:

```
SELECT ename,
       job,
       sal,
       comm ,
       CURDATE() "Fecha actual",
       HIREDATE "Contratacion",
       DATEDIFF(CURDATE(),HIREDATE) "Dias",
       FROM_DAYS( DATEDIFF(CURDATE(),HIREDATE) ) "Años"
FROM EMP
```

26. Obtener un listado con el nombre y la inicial de cada empleado.

```
SELECT ename,
       substring(ename,1,1) "inicial"
FROM EMP
```

27. Calcular el IRPF de cada empleado teniendo en cuenta que a los CLERK se les retiene un 15% y a los ANALYST un 20% y al resto un 19%.

```
SELECT ename,
       job,
       sal,
       comm ,
       case
       when job='CLERK' then sal*0.15
       when job='ANALYST' then sal*0.20
       else sal*0.19
       end
FROM EMP
```

28. Calcular el máximo sueldo ganado por un empleado:

```
SELECT max(sal)
FROM EMP
```

29. Quien es el empleado mejor pagado:

```
SELECT *  
FROM EMP  
WHERE sal IN (  
SELECT max(sal)  
FROM EMP)
```

JOINTS

1. Listar el nombre del empleado y el del departamento al que pertenece:

```
# INNER JOIN  
SELECT ename AS Empleado,  
       dname AS Departamento  
FROM EMP  
INNER JOIN DEPT ON EMP.deptno=DEPT.deptno
```

```
# INNER JOIN con alias  
SELECT ename AS Empleado,  
       dname AS Departamento  
FROM EMP e  
INNER JOIN DEPT d ON e.deptno=d.deptno
```

```
# INNER JOIN modo Oracle  
SELECT ename AS Empleado,  
       dname AS Departamento  
FROM EMP, DEPT  
WHERE EMP.deptno=DEPT.deptno
```

2. Listar el nombre del empleado, el código de este, el departamento y el código del departamento para cada empleado:

```
SELECT empno, ename, deptno, dname  
FROM EMP  
INNER JOIN DEPT ON EMP.deptno=DEPT.deptno
```

OJO da error.

```
SELECT EMP.empno, EMP.ename,  
       DEPT.deptno, DEPT.dname  
FROM EMP  
INNER JOIN DEPT ON EMP.deptno=DEPT.deptno
```

```
# Con alias_  
SELECT E.empno, E.ename,  
       D.deptno, D.dname  
FROM EMP E  
INNER JOIN DEPT D ON E.deptno=D.deptno
```

3. Listar el nombre del empleado y el de su jefe:
- ```
SELECT a.ename "Empleado", b.ename "Jefe"
FROM EMP a
INNER JOIN EMP b ON a.mgr=b.empno
```

# Modo Oracle

```
SELECT a.ename "Empleado", b.ename "Jefe"
FROM EMP a,
 EMP b
WHERE a.mgr=b.empno
```

4. Listar el nombre del empleado, su puesto, el de su jefe y su puesto:
- ```
SELECT a.ename "Empleado", a.job "Empleado job",
       b.ename "Jefe", b.job "Jefe job"
FROM EMP a
INNER JOIN EMP b ON a.mgr=b.empno
```

Modo Oracle

```
SELECT a.ename "Empleado", a.job "Empleado job",
       b.ename "Jefe", b.job "Jefe job"
FROM EMP a,
      EMP b
WHERE a.mgr=b.empno
```

Fijaos que no sale el presidente. No tiene jefe!!!

5. Listar el nombre del empleado, su puesto, el de su jefe y su puesto. Incluidos los que no tienen jefe:
- ```
SELECT a.ename "Empleado", a.job "Empleado job",
 b.ename "Jefe", b.job "Jefe job"
FROM EMP a
LEFT JOIN EMP b ON a.mgr=b.empno
```

# notación Oracle:

```
SELECT a.ename "Empleado", a.job "Empleado job",
 b.ename "Jefe", b.job "Jefe job"
FROM EMP a,
 EMP b
WHERE a.mgr(+) = b.empno
```

No funciona en Mysql:

6. Nombre del empleado, fecha de contratación de este, nombre de su jefe, fecha de contratación de este solo si la fecha de contratación del empleado es anterior a la de su jefe:
- ```
SELECT a.ename "Empleado", a.hiredate,
       b.ename "Jefe", b.hiredate
FROM EMP a
INNER JOIN EMP b ON a.mgr=b.empno
WHERE a.hiredate < b.hiredate
```

Nomenclatura Oracle

```
SELECT a.ename "Empleado", a.hiredate,  
b.ename "Jefe", b.hiredate  
FROM EMP a,  
      EMP b  
WHERE      a.mgr=b.empno  
          AND a.hiredate < b.hiredate
```

7. Listado de departamentos, su localización y empleados que trabajan en el:

```
SELECT dname,loc,ename  
FROM DEPT D  
INNER JOIN EMP E ON D.deptno=E.deptno
```

#OJO, OPERATIONS no sale en el listado:

```
SELECT dname,loc,ename  
FROM DEPT D  
LEFT JOIN EMP E ON D.deptno=E.deptno
```

8. Listado de nombres de empleados y sus departamentos para los empleados que ganan menos de 1000:

```
SELECT ename,sal,dname  
FROM EMP E  
LEFT JOIN DEPT D ON E.deptno=D.deptno  
WHERE sal < 1000
```

OJO a;

```
SELECT ename,sal,dname  
FROM EMP E  
RIGHT JOIN DEPT D ON E.deptno=D.deptno
```

Sobre IMDB

1. En ratings vemos el campo rating_time:
SELECT FROM_UNIXTIME(rating_time)
FROM ratings
2. Ver la película con más número de ratings:
SELECT movie_id,count(ratings)
FROM ratings r
GROUP BY movie_id
ORDER BY 2 DESC

Podemos sacar la media de las valoraciones:

```
SELECT movie_id,count(ratings),AVG(ratings)  
FROM ratings r  
GROUP BY movie_id
```


ORDER BY 2 DESC

Podemos ordenar por la media de las valoraciones:

```
SELECT movie_id,count(ratings),AVG(ratings)
FROM ratings r
GROUP BY movie_id
ORDER BY 3 DESC
```

Vamos a ver las que tengan más de 300 valoraciones:

```
SELECT movie_id,count(ratings),AVG(ratings)
FROM ratings r
GROUP BY movie_id
HAVING count(ratings) > 300
ORDER BY 3 DESC
```

Vemos cual es:

```
SELECT *
FROM movies
WHERE movie_id=50
```

3. Localiza el usuario que mas valoraciones ha hecho:

```
SELECT user_id,count(ratings)
FROM ratings r
GROUP BY user_id
ORDER BY 2 DESC
```

Vemos quien es:

```
SELECT *
FROM users u
WHERE user_id=405
```

#¿Valoro a Star Wars?

```
SELECT *
FROM ratings
WHERE user_id=405 and movie_id=50
```