

Práctica de Martin

1. Listar los datos de los empleados que tienen como jefe a alguno de los empleados cuyo jefe directo es el PRESIDENTE de la empresa. Haz una versión con subconsultas y otra con join. OJO. No mostrar datos de más en el join. 2 puntos.

```
SELECT *  
FROM empleados  
WHERE director IN (SELECT emp_no  
                   FROM empleados  
                   WHERE director = (SELECT emp_no  
                                     FROM empleados  
                                     WHERE oficio = 'Presidente'));
```

```
SELECT e.*  
FROM empleados e JOIN empleados j ON e.director = j.emp_no  
     JOIN empleados p ON j.director = p.emp_no  
WHERE p.oficio = 'Presidente';
```

2. Mostrar el número de empleado, el nombre y el apellido separados en columnas, de los empleados cuyo oficio es DIRECTOR, incluyendo una columna con el total del sueldo (SALARIO + COMISION) que cobran los empleados que dependen de ellos.

```
SELECT emp_no, SUBSTRING_INDEX(nomap, ' ', 1) nombre, SUBSTRING_INDEX(nomap, ' ', -1)  
apellido,  
       (SELECT SUM(salario + IFNULL(comision, 0))  
        FROM empleados e2  
        WHERE e2.director = e.emp_no) total_sueldo_empleados  
FROM empleados e  
WHERE oficio = 'Director';
```

3. Obtener, para cada empleado perteneciente al departamento de VENTAS, el número de UNIDADES de cada producto que vendió.

```
SELECT emp_no, nomap, producto_no, SUM(unidades) uds_vendidas  
FROM empleados JOIN clientes ON emp_no = vendedor_no JOIN pedidos USING (cliente_no)  
WHERE dep_no = (SELECT dep_no  
                FROM departamentos  
                WHERE dnombre = 'Ventas')  
GROUP BY emp_no, producto_no;
```

4. Seleccionar, para cada departamento, los empleados que tiene, mostrando un 0 en caso de no tener ninguno. Haz una versión con join y otra en la que no se use join. 2 puntos.

```
SELECT dep_no, COUNT(emp_no) empleados  
FROM departamentos LEFT JOIN empleados USING (dep_no)  
GROUP BY dep_no;
```

```

SELECT dep_no, COUNT(*)
FROM empleados
GROUP BY dep_no
UNION
SELECT dep_no, 0
FROM departamentos
WHERE dep_no NOT IN (SELECT dep_no FROM empleados);

```

5. Mostrar, para todos los productos, las ganancias generadas (calculadas multiplicando las UNIDADES vendidas por el PRECIO_ACTUAL en cada uno) y el stock resultante obtenido al restarle al STOCK_DISPONIBLE las unidades vendidas en los diferentes pedidos.

```

SELECT producto_no, SUM(unidades) * precio_actual ganancias, stock_disponible -
SUM(unidades) stock
FROM productos JOIN pedidos USING (producto_no)
GROUP BY producto_no;

```

6. Encontrar los datos de los clientes que tienen como vendedor al empleado que menos gana en cada departamento. (Subconsulta)

```

SELECT *
FROM clientes
WHERE vendedor_no IN (SELECT emp_no
FROM empleados e
WHERE salario = (SELECT MIN(salario)
FROM empleados e2
WHERE e.dep_no = e2.dep_no));

```

7. Mostrar los datos de los empleados que tienen mayor SALARIO que el resto de empleados de su departamento. (Subconsultas)

```

SELECT *
FROM empleados e1
WHERE salario = (SELECT MAX(salario)
FROM empleados e2
WHERE e1.dep_no = e2.dep_no);

```

8. Obtener todos los datos del departamento que más dinero le cuesta a la empresa, es decir, donde la suma de los sueldos

```

CREATE OR REPLACE VIEW total_sueldos_dep AS
SELECT dep_no, SUM(salario + IFNULL(comision, 0)) total_sueldos
FROM departamentos JOIN empleados USING (dep_no)
GROUP BY dep_no;
SELECT *
FROM departamentos
WHERE dep_no = (SELECT dep_no
FROM total_sueldos_dep
WHERE total_sueldos = (SELECT MAX(total_sueldos)
FROM total_sueldos_dep));

```