

```

/*          SOLUCIONES EJERCICIOS DML. LAS SUBCONSULTAS
*/

USE GestionSimples;

--1. Listar los numclie y nombres de los clientes que tienen asignado el
representante Juan Gris.

SELECT numclie, nombre
FROM clientes
WHERE repclie = ANY (SELECT numemp
                     FROM empleados
                     WHERE nombre = 'Juan Gris');

SELECT numclie, nombre
FROM clientes
WHERE repclie IN ( SELECT numemp
                  FROM empleados
                  WHERE nombre = 'Juan Gris');

SELECT numclie, nombre
FROM clientes
WHERE EXISTS ( SELECT *
              FROM empleados
              WHERE nombre = 'Juan Gris' AND numemp = repclie);

--2. Listar todos los productos (código, descripción y stock) del fabricante ACI y
cuyas existencias superan las existencias del producto ACI-41004.

SELECT idfab, idproducto, descripcion, existencias
FROM productos
WHERE idfab = 'ACI'
AND existencias > (SELECT existencias
                  FROM productos
                  WHERE idfab = 'ACI' AND idproducto = '41004');
-- En este caso no hace falta utilizar comparación cuantificada porque la
subconsulta devolverá como mucho una sólo fila por el WHERE que incluye.

--3. Listar los empleados (numemp, nombre y oficina) que trabajan en oficinas
"buenas" (las que han vendido más que su objetivo).

SELECT numemp, nombre, oficina
FROM empleados
WHERE oficina IN (SELECT oficina
                 FROM oficinas
                 WHERE ventas > objetivo);

SELECT numemp, nombre, oficina
FROM empleados
WHERE oficina = ANY (SELECT oficina
                    FROM oficinas
                    WHERE ventas > objetivo);

SELECT numemp, nombre, oficina
FROM empleados
WHERE EXISTS(SELECT *
            FROM oficinas
            WHERE ventas > objetivo AND
            oficinas.oficina=empleados.oficina);

```

--4. Listar los empleados (numemp, nombre y oficina) que no trabajan en oficinas dirigidas por el empleado 108.

```
SELECT numemp, nombre, oficina
FROM empleados
WHERE oficina IS NULL -- Para que salgan los empleados que no tienen oficina
OR oficina NOT IN (SELECT oficina
                   FROM oficinas
                   WHERE dir = 108);

SELECT numemp, nombre, oficina
FROM empleados
WHERE NOT EXISTS (SELECT *
                  FROM oficinas
                  WHERE dir = 108 and empleados.oficina = oficinas.oficina);
```

--5. Listar los clientes (nombre) que han remitido pedidos del fabricante ACI y n°producto que empiece por 4100, entre enero y el 15 de abril de 1990.

```
SELECT numclie, nombre
FROM clientes
WHERE numclie IN (SELECT clie FROM pedidos
                  WHERE fab = 'ACI'
                  AND producto LIKE '4100%' AND fechapedido >=
                  '01/01/90' AND fechapedido < '16/04/90');

SELECT numclie, nombre
FROM clientes
WHERE EXISTS (SELECT clie FROM pedidos
              WHERE numclie = clie AND fab = 'ACI'
              AND producto LIKE '4100%' AND fechapedido >=
              '01/01/90' AND fechapedido < '16/04/90');
```

--6. Listar los productos (clave y descripcion) para los cuales no se ha recibido ningún pedido de 10 euros o más.

```
SELECT idfab, idproducto, descripcion
FROM productos
WHERE NOT EXISTS (SELECT * FROM pedidos
                  WHERE idfab = fab
                  AND idproducto = producto AND importe >= 10)
```

--7. Listar los clientes (numclie, nombre) asignados a Juan que no han remitido un pedido superior a 300 euros.

```
SELECT numclie, nombre
FROM clientes
WHERE repclie IN (SELECT numemp FROM empleados
                  WHERE nombre LIKE 'Juan%')
AND NOT EXISTS (SELECT * FROM pedidos
                WHERE numclie = clie AND importe > 300)
```

--8. Listar las oficinas (oficina, ciudad) que tienen al menos un empleado cuya cuota represente más del 55% del objetivo de la oficina.

```
SELECT oficina, ciudad
FROM oficinas
WHERE EXISTS (SELECT * FROM empleados
              WHERE oficinas.oficina = empleados.oficina AND cuota > (.55
* objetivo))
```

--9. Listar los empleados (numemp, nombre) que han realizado un pedido que represente más del 1% de su cuota.

```
SELECT numemp, nombre
FROM empleados
WHERE (cuota * .01) <= ANY (SELECT importe FROM pedidos
                          WHERE rep = numemp)
```

```
SELECT numemp, nombre
FROM empleados
WHERE EXISTS (SELECT * FROM pedidos
              WHERE rep = numemp AND importe > cuota * 0.01)
```

```
SELECT numemp, nombre
FROM empleados
WHERE numemp IN (SELECT rep FROM pedidos
                WHERE importe > cuota* 0.01)
```

--10. Listar los empleados (numemp, nombre y edad) que no dirigen una oficina.

```
SELECT numemp, nombre, edad
FROM empleados
WHERE NOT EXISTS (SELECT * FROM oficinas WHERE dir = numemp)
```

--11. Listar los empleados (numemp, nombre y edad) que tienen una cuota por encima de la media.

```
SELECT numemp, nombre, edad
FROM empleados
WHERE cuota > (SELECT AVG(cuota) FROM empleados)
```

--12. Obtener cuántos empleados tienen una cuota por encima de la media, y la edad media de estos.

```
SELECT count(*) AS Cuantos, AVG(edad) AS [Edad Media]
FROM empleados
WHERE cuota > (SELECT AVG(cuota) FROM empleados)
```

--13. Listar todas las oficinas cuyos objetivos superan la suma de las cuotas de sus vendedores.

```
SELECT oficina, ciudad
FROM oficinas
WHERE objetivo > (SELECT SUM(cuota) FROM empleados WHERE oficina =
oficinas.oficina)
```

--14. Listar los empleados (numemp, nombre) mayores de 40 años que dirigen a un vendedor con superávit (ha vendido más que su cuota).

```
SELECT numemp, nombre
FROM empleados
WHERE edad > 40
      and numemp IN (SELECT jefe FROM empleados WHERE ventas > cuota )
```

--15. Listar los empleados (código de empleado) cuyo importe de pedido medio para productos fabricados por ACI es superior al importe medio global (de todos los pedidos de todos los empleados).

```
SELECT rep
FROM pedidos
WHERE fab = 'ACI'
GROUP BY rep
HAVING AVG(importe) > (SELECT AVG(importe) FROM pedidos)
```

--16. Listar los empleados (numemp, nombre, ventas) cuyas ventas son iguales o superiores al objetivo  
-- de las oficinas de una determinada ciudad (de todas las oficinas de esa ciudad).  
-- Las oficinas con objetivo nulo no se deben de tener en cuenta (como si no existiesen).  
-- Si no hay oficinas en la ciudad no queremos que salga ningún empleado.  
-- Intentar resolver la consulta utilizando >ALL

--16.1 Probar primero con A Coruña.

```
SELECT numemp, nombre, ventas
FROM empleados
WHERE ventas >= ALL (SELECT objetivo
                     FROM oficinas
                     WHERE ciudad = 'A Coruña');
```

-- 16.2 Ahora con Pamplona.  
--Pamplona tiene una oficina con objetivo nulo, en este caso no queremos que esa oficina cuente.  
-- Si la consulta devuelve algún nulo  
--no se cumple el ALL (Pamplona tiene un objetivo que cumple y uno nulo), añadimos un filtro.

```
SELECT numemp, nombre, ventas
FROM empleados
WHERE ventas >= ALL (SELECT objetivo
                     FROM oficinas
                     WHERE ciudad = 'Pamplona' and objetivo is not null);
```

```

-- 16.3 Para Barcelona.
--En este caso no tenemos oficinas en barcelona por lo que no queremos que salga
ningún empleado.
--Si no queremos que salgan tenemos que añadir una condición:
SELECT numemp, nombre, ventas
FROM empleados
WHERE ventas >= ALL (SELECT objetivo
                     FROM oficinas
                     WHERE ciudad = 'Barcelona' and objetivo is not null)
AND EXISTS (SELECT *
            FROM oficinas
            WHERE ciudad = 'Barcelona');

-- 16.4 Ahora para Madrid.
-- como en Madrid sólo hay una oficina y no tiene objetivo no queremos que salga
ningún empleado.

SELECT numemp, nombre, ventas
FROM empleados
WHERE ventas >= ALL (SELECT objetivo
                     FROM oficinas
                     WHERE ciudad = 'Madrid' and objetivo is not null)
AND EXISTS (SELECT *
            FROM oficinas
            WHERE ciudad = 'Madrid' and objetivo is not null);

-- Esta sería la consulta definitiva que nos serviría para cualquier situación.
-- Como se ve el modificador ALL puede darnos problemas
-- Otra solución:

-- 16.5 Sin ALL
SELECT numemp, nombre, ventas
FROM empleados
WHERE ventas >= (SELECT MAX(objetivo)
                 FROM oficinas
                 WHERE ciudad = 'Madrid');

--17. Listar las oficinas en donde todos los empleados tienen ventas que superan al
50% del objetivo de la oficina.

SELECT oficina, ciudad
FROM oficinas
WHERE (objetivo * .5) <= (SELECT MIN(ventas) FROM empleados WHERE
empleados.oficina = oficinas.oficina);

```