

19. Muestra las matrículas de los vehículos que no tienen multas.

Select matricula from Matriculas where matricula NOT IN (select matricula from Multas);

También:

```
select matricula
from matriculas
where matricula <> all (select matricula
                        from multas);

select m.matricula
from matriculas m
where not exists (select *
                  from multas mu
                  where mu.matricula = m.matricula);
```

20. Muestra el campo nombre de las personas que no tienen vehículo, ordenado alfabéticamente.

```
select p.nombre
from personas p
where p.dni not in (select m.dni
                   from matriculas m)

order by 1;
```

21. Muestra la/s matrícula/s y el importe/s de la/s mayor/es multa/s.

```
select matricula, importe
from multas
where importe = (select max(importe)
                 from multas);
```

22. Muestra el campo nombre/s de la/s persona/s que haya/n tenido la multa/s de menor importe, y el importe.

```
select p.nombre, mu.importe
from personas p, matriculas m, multas mu
where p.dni = m.dni
      and m.matricula = mu.matricula
      and mu.importe = (select min(importe)
                       from multas);
```

23. Muestra, sin repetir, las matrículas que tienen más de una multa.

Mostraremos las matrículas de las multas si existe (**exists**) al menos otra multa (ref distinto) con esa misma matrícula.

```
select distinct mu.matricula
from multas mu
where exists (select *
              from multas mu2
              where mu2.matricula = mu.matricula
                    and mu2.referencia <> mu.referencia);
```

24. Muestra por cada multa que supera la media del total en importe, el importe y el nombre del infractor.

```
select mu.importe, p.nombre
```

```
from personas p, matriculas ma, multas mu
where p.dni = ma.dni
      and ma.matricula = mu.matricula
      and mu.importe > (select avg(importe)
                        from multas);
```

Recuerda

- Cuando no se usa **group by** (que veremos próximamente), en la línea de la **select** jamás podremos mezclar nombres de campos con funciones de columna.
- Las subselect correlacionadas son menos eficientes, ya que por cada fila de la **select** principal se vuelve a ejecutar la subselect. Por tanto, siempre que se pueda, se optará por una solución no corelacionada.
- Una subselect puede a su vez contener otra subselect, y así sucesivamente.
- Cuando comparamos un valor de la **select** con varios valores de la subselect, se ha de usar **all**, **some** o **any** (equivalentes), o **in**. Los campos a comparar no se han de llamar igual necesariamente, pero sí han de coincidir en tipo.
- La cláusula **exists** va acompañada de una subselect que siempre tendrá la forma '**select ***'. Se cumplirá si la subselect devuelve alguna fila.

Consultas con agrupamiento de filas de las tablas de multas.

El concepto del agrupamiento de filas nace de la necesidad de resolver consultas como ésta:

- Muestra, por cada matrícula, el número de multas que tiene.

Aparentemente es una consulta sencilla, pero con los elementos vistos hasta ahora no podemos resolverla. Podríamos mostrar el número de multas en total, o el número de multas de una matrícula concreta, pero no mostrar el listado que nos piden.

El agrupamiento de filas consiste en fusionar una o varias filas en una, cuando coinciden el campo o los campos por los que estamos agrupando. Además al agrupar podemos utilizar las funciones de columna para hacer cálculos que afectarán a cada uno de los grupos, como contar el número de filas agrupadas con **count(*)**, o por ejemplo sumar valores con **sum(importe)**.

Al igual que con la cláusula **where** seleccionamos las filas a mostrar, con la cláusula **having** podemos seleccionar los grupos que queremos mostrar. El orden de una select completa quedaría por tanto así:

```
select
from
where
group by
having
order by
```

Donde el **where** lo podremos omitir si no queremos eliminar filas, o el **having** en el caso de querer mostrar todos los grupos. **order by**, que también es opcional, siempre va al final.

25. Muestra el nombre de las personas que han tenido más de una multa en el mismo lugar.

```
select distinct p.nombre
from personas p, matriculas m, multas mu
where p.dni = m.dni
      and m.matricula = mu.matricula
      and exists (select *
                  from matriculas m2, multas mu2
                  where m2.matricula = mu2.matricula
                        and m.dni = m2.dni
                        and mu.lugar = mu2.lugar
                        and mu.ref <> mu2.ref);
```

26. Muestra, por cada matrícula, el número de multas que tiene.

```
select matricula, count(*)
from multas
group by matricula;
```

27. Muestra, por cada matrícula, el número de multas que tiene, cuando tenga más de una.

```
select matricula, count(*)
from multas
group by matricula
having count(*) > 1;
```

28. Muestra el nombre de las personas que han tenido más de una multa en el mismo lugar.

```
select p.nombre, mu.lugar
from personas p, matriculas m, multas mu
where p.dni = m.dni
      and m.matricula = mu.matricula
group by p.nombre, m.dni, mu.lugar
having count(*) > 1;
Nota: Al agrupar además por dni, nos aseguramos de no contar
personas distintas que casualmente se llamen igual.
```

29. Muestra el nombre de las personas, y número de vehículos que poseen, cuando su nombre (no apellido), empieza por 'V'.

```
select p.nombre, count(*)
from personas p, matriculas m
where p.dni = m.dni
      and p.nombre like 'V%'
group by p.dni, p.nombre;
```

30. Muestra cuál es el mayor número de vehículos que posee una persona.

Agrupando por **dni** en **matriculas** podemos contar por cada **dni** cuántas matrículas hay con **count(*)**. Finalmente con la

función **max()**, de todos los valores obtenidos nos quedaremos sólo con uno; el mayor. La función **count(*)** se ejecutará una vez por cada grupo obtenido, y la función **max()** sólo una en toda la **select**, una vez calculados todos los grupos.

```
select max(count(*)) MAXIMO
from matriculas
group by dni;
```

31. Muestra el nombre y número de vehículos que posee el propietario con mayor número de vehículos.

En la **select** calculamos por cada persona el número de vehículos que posee, y con la **subselect** nos quedamos con la que tiene un mayor número.

```
select p.nombre, count(*)
from personas p, matriculas m
where p.dni = m.dni
group by p.dni, p.nombre
having count(*) = (select max(count(*))
                  from matriculas
                  group by dni);
```

También se podría haber resuelto así:

```
select p.nombre, count(*)
from personas p, matriculas m
where p.dni = m.dni
group by p.dni, p.nombre
having count(*) >= all(select count(*)
                      from matriculas
                      group by dni);
```

32. Muestra el nombre de la persona que acumula un mayor importe en multas, y el importe.

En la **select** principal calculamos los importes acumulados de cada persona, y nos quedamos con aquélla cuya suma sea la máxima, que es calculada en la **subselect**, y sin necesidad de usar la tabla **personas**, ya que agrupando por **dni** obtenemos el mismo resultado.

```
select p.nombre, sum(importe)
from personas p, matriculas m, multas mu
where p.dni = m.dni
      and m.matricula = mu.matricula
group by p.dni, p.nombre
having sum(mu.importe) = (select max(sum(mu2.importe))
                        from matriculas m2, multas mu2
                        where m2.matricula = mu2.matricula
                        group by m2.dni);
```

- **IMPORTANTE.** Cuando agrupamos, en la línea **select**, sólo pueden aparecer campos por los que agrupamos, o funciones de columna.
- En el **having** no podremos hacer referencia a campos por los que no agrupamos.
- Se puede agrupar por campos que no son mostrados en la línea **select**.