
RESOLUCIÓN DE CONSULTAS SQL

PENDIENTE DE REVISIÓN

Luis Egui

Contenidos

	Página
Problema 1	1
Problema 2	4

Problema 1

Dado el modelo relacional:

Pilotos(iniciales, nombre, escuderia, pais)

Circuitos(nombre, longitud, nVueltas)

Parrilla(piloto, circuito, posicion)

Tiempos(piloto, circuito, nVuelta, tiempo, paradaBoxes)

a) Realizando la consulta con un producto cartesiano, seria:

```
1  select p1.*
2  from Pilotos as p1, Pilotos as p2
3  where p1.iniciales != p2.iniciales — distinct
4  and p1.pais = p2.pais and p1.escuderia = p2.escuderia;
5
```

Haciendo uso del *join*, seria:

```
1  select p1.*
2  from Pilotos p1
3      join Pilotos p2 on p1.escuderia = p2.escuderia
4  where p1.pais = p2.pais and p1.iniciales != p2.iniciales;
5
```

b)

```
1  select distinct escuderia
2  from Pilotos
3      inner join Parrilla on iniciales = piloto
4  where posicion in ('1','2','3');
5
```

c)

```
1  select circuito, max(tiempo) as tiempo_max_vuelta
2  from Tiempos
3  group by circuito;
4
```

d)

```
1  select circuito , piloto , count(nVuelta) as vueltas_completadas ,
2      sum(tiempo) as tiempo_total
3  from Tiempos
4  where circuito = "Monza"
5  group by circuito , piloto
6  order by vueltas_completadas desc , tiempo_total asc;
7
```

e) Usando el operador *limit* de MySQL o PostgreSQL:

```
1  select nombre, longitud
2  from Circuitos
3  order by longitud desc
4  limit 1;
5
```

Para OracleSQL 12c R1 (12.1):

```
1  select nombre, longitud
2  from Circuitos
3  order by longitud desc
4  fetch first 1 rows only;
5
```

f)

```
1  select piloto , circuito , sum(tiempo) as tiempo_invertido ,
2      count(nVuelta) as vueltas_totales
3  from Tiempos
4      inner join Circuitos on circuito = nombre
5  group by piloto , circuito
6  having count(vueltas_totales) = nVueltas;
7
```

- g) Suponemos que *Tiempos(paradaBoxes)* se trata de un *boolean* cuyo valor es 1 cuando el piloto ha realizado la parada en la vuelta indicada en el registro; y que por defecto su valor es 0.

```
1  select circuito , avg(paradaBoxes) as media_paradas_boxes
2  from Tiempos
3  group by circuito;
4
```

Problema 2

Dado el modelo relacional:

Persona(dni, sexo, edad)

Habla(dniPersona, idioma, grado)

SolicitaCita(dniSolicitante, dniSolicitado, idioma)

a)

```
1  select idioma, count(*) as no_hablantes
2  from Habla
3  where grado = 3
4  group by idioma
5  order by no_hablantes desc;
```

b)

```
1  select avg(edad) as promedio_edad_hispanohablantes
2  from Habla h1
3      inner join Persona on h1.dniPersona = dni
4  where idioma = 'Espanol' and idioma not in (
5      select idioma
6      from Habla h2
7      where h1.dniPersona = h2.dniPersona and h2.idioma != 'Espanol');
8  — TODO: idioma = null?
```

c) Esta consulta no nos devuelve una unica fila, sino que para cada persona; nos devuelve el numero de idiomas que habla. AL estar ordenadas de mayor numero de idiomas hablados, a menor: obtendremos en primera posicion la persona que mas idiomas habla.

```
1  select dni, sexo, edad, count(idioma) as lenguas_habladas
2  from Habla
3      inner join Persona on dniPersona = dni
4  group by dni
5  order by lenguas_habladas desc;
6  — lenguas_habladas son filas unicas ya que idioma es unique.
```

En MySQL y PostgreSQL es sencillo poder limitar el numero de filas obtenidas en la anterior consulta, de la siguiente manera:

```
1  select dni, sexo, edad, count(idioma) as lenguas_habladas
2  from Habla
3       inner join Persona on dniPersona = dni
4  group by dni
5  order by lenguas_habladas desc
6  limit 1;
```

Para Oracle SQL 12c R1 (12.1):

```
1  select dni, sexo, edad, count(idioma) as lenguas_habladas
2  from Habla
3       inner join Persona on dniPersona = dni
4  group by dni
5  order by lenguas_habladas desc
6  fetch first 1 rows only;
```

Una manera mas trivial de conseguir el mismo resultado seria:

```
1  select dni, sexo, edad, max(lenguas_habladas) as max_lh
2  from (
3       select dni, sexo, edad, count(idioma) as lenguas_habladas
4       from Habla
5            inner join Persona on dniPersona = dni
6       group by dni
7       order by lenguas_habladas desc
8  )
9  where max_lh = lenguas_habladas;
```

d) De manera trivial:

```
1  select dni, sexo, edad, max(num_citas) as max_nc
2  from (
3      select p2.*, count(*) as num_citas
4      from SolicitaCita
5      inner join Persona p1 on dniSolicitante = p1.dni
6      inner join Persona p2 on dniSolicitado = p2.dni
7      where p1.sexo != p2.sexo
8      group by p2.dni
9      order by num_citas desc
10 )
11 where max_nc = num_citas;
```

Para MySQL y PostgreSQL:

```
1  select p2.*, count(*) as num_citas
2  from SolicitaCita
3      inner join Persona p1 on dniSolicitante = p1.dni
4      inner join Persona p2 on dniSolicitado = p2.dni
5  where p1.sexo != p2.sexo
6  group by p2.dni
7  order by num_citas desc
8  limit 1;
```

Para Oracle 12c R1 (12.1):

```
1  select p2.*, count(*) as num_citas
2  from SolicitaCita
3      inner join Persona p1 on dniSolicitante = p1.dni
4      inner join Persona p2 on dniSolicitado = p2.dni
5  where p1.sexo != p2.sexo
6  group by p2.dni
7  order by num_citas desc
8  fetch first 1 rows only;
```


- e) Recordamos que piden los idiomas hablados por el solicitante/solicitado pero que no han sido pedidos en la cita. Por lo que la consulta sería:

```
1  select h1.idioma — idiomas que habla el solicitante
2  from SolicitaCita
3      inner join Habla h1 on dniSolicitante = dniPersona
4  union — une filas de manera vertical, sin duplicarlas.
5  select h2.idioma — idiomas que habla el solicitado
6  from SolicitaCita
7      inner join Habla h2 on dniSolicitado = dniPersona
8  minus
9  select distinct idioma — idiomas pedidos en las citas
10 from SolicitaCita;
```

f)

```
1  select distinct h1.*
2  from SolicitaCita
3      inner join Habla h1 on dniSolicitante = h1.dniPersona
4      inner join Habla h2 on dniSolicitado = h2.dniPersona
5      inner join Persona on dniSolicitado = dni
6  where sexo = 'M' and edad < 25
7      and h1.idioma = 'Ingles' and h2.idioma = 'Ingles';
```

g)

```
1  select dni, sexo, edad, count(idioma) as num_idiomas_hablados
2  from (
3      select dni
4      from Persona
5      minus — personas nunca solicitadas
6      select distinct dniSolicitado
7      from SolicitaCita
8  )
9  inner join Habla on dni = dniPersona;
10 inner join Persona p on dni = p.dni;
```