



UNIVERSIDAD DE MÁLAGA



E.T.S. INGENIERÍA
INFORMÁTICA

UNIVERSIDAD DE MÁLAGA

Relación 3 (DML)

Base de Datos

Álvaro Pino Ruiz
Grado en Ingeniería Informática
Curso 2021/2022

Índice general

1. Agrupación 1	3
1.1. Ejercicio 1	3
1.2. Ejercicio 2	3
1.3. Ejercicio 3	3
1.4. Ejercicio 4	3
1.5. Ejercicio 5	4
1.6. Ejercicio 6	4
2. Agrupación 2	5
2.1. Ejercicio 7	5
2.2. Ejercicio 8	5
2.3. Ejercicio 9	5
2.4. Ejercicio 10	5
2.5. Ejercicio 11	6
3. Agrupación 3	7
3.1. Ejercicio 12	7
3.2. Ejercicio 13	7
3.3. Ejercicio 14	7
3.4. Ejercicio 15	8
3.5. Ejercicio 16	8
4. Negativas 1	9
4.1. Ejercicio 17	9
4.2. Ejercicio 18	9
4.3. Ejercicio 19	9
4.4. Ejercicio 20	10
4.5. Ejercicio 21	10
4.6. Ejercicio 22	10
5. Complementarias 1	11
5.1. Ejercicio 23	11
5.2. Ejercicio 24	11
5.3. Ejercicio 25	11
5.4. Ejercicio 26	12
5.5. Ejercicio 27	12
5.6. Ejercicio 28	12
5.7. Ejercicio 29	12
5.8. Ejercicio 30	12

Capítulo 1

Agrupación 1

1.1. Ejercicio 1

Calcular el número de profesores de cada departamento. Muestre el nombre del departamento y el número de profesores.

```
select d.nombre "DEPARTAMENTO", count (*) "NUMERO PROFESORES" from
departamentos d join profesores p on (d.codigo = p.departamento) group by d.nombre;
```

1.2. Ejercicio 2

Calcular el número de créditos asignados a cada departamento. Se consideran los créditos establecidos para cada asignatura del departamento, no si la imparten o no profesores del mismo, es decir, sume directamente los créditos de las asignaturas y reúna asignaturas y departamentos directamente, sin utilizar ninguna otra tabla.

```
select d.nombre "DEPARTAMENTO", sum(creditos) "NUMERO CREDITOS" from
departamentos d join asignaturas a on (a.departamento = d.codigo) group by d.nombre;
```

1.3. Ejercicio 3

Calcular el número de alumnos matriculados por curso (cada alumno debe contar una sola vez por curso aunque esté matriculado de varias asignaturas). Utilice COUNT (DISTINCT ...).

```
select a.curso, count (distinct al.dni) "NUMERO ALUMNOS" from asignaturas a join
matricular m on (a.codigo = m.asignatura) join alumnos al on (al.dni = m.alumno) where
a.curso is not null group by a.curso;
```

1.4. Ejercicio 4

Por cada número de despacho, indicar el total de créditos impartidos por profesores ubicados en ellos.

```
select p.despacho, sum(im.carga_creditos) "TOTAL CREDITOS" from profesores p join
impartir im on (p.id = im.profesor) where p.despacho is not null group by p.despacho;
```

1.5. Ejercicio 5

Calcular, por cada asignatura, qué porcentaje de sus alumnos son mujeres. Mostrar el código de la asignatura y el porcentaje.

```
select m.asignatura "ASIGNATURA", sum(decode(al.genero,'FEM',1,'MASC',
0))/count(al.dni)*100 "PORCENTAJE" from matricular m join alumnos al on (m.alumno =
al.dni) group by m.asignatura order by m.asignatura;
```

1.6. Ejercicio 6

Mostrar la población de cada provincia española: nombre de provincia y suma de hombres y mujeres de todos sus municipios.

```
select prov.nombre, sum(mun.hombres + mun.mujeres) "SUMA" from provincia prov join
municipio mun on (mun.cpro = prov.codigo) group by prov.nombre order by prov.nombre;
```

Capítulo 2

Agrupación 2

2.1. Ejercicio 7

Visualizar, por cada departamento, el nombre del profesor más cercano a la jubilación (de mayor edad).

```
select d.nombre "DEPARTAMENTO", p.nombre || ' ' || p.apellido1 "PROFESOR" from
departamentos d join profesores p on (p.departamento = d.codigo) where (p.departamento,
p.fecha_nacimiento) in (select departamento, min(fecha_nacimiento) from profesores where
departamento = p.departamento group by departamento) order by d.codigo;
```

2.2. Ejercicio 8

Visualizar la asignatura de mayor número de créditos en que se ha matriculado cada alumno.

```
select distinct al.dni "ALUMNO", a.codigo, a.nombre from alumnos al join matricular m on
(al.dni = m.alumno) join asignaturas a on (a.codigo = m.asignatura) where (alumno,
creditos) in (select alumno, max(creditos) from matricular join asignaturas a on (a.codigo =
asignatura) group by alumno);
```

2.3. Ejercicio 9

Visualizar el profesor más antiguo de cada departamento.

```
select d.nombre "DEPARTAMENTO", p.nombre || ' ' || p.apellido1 "PROFESOR" from
departamentos d join profesores p on (p.departamento = d.codigo) where (p.departamento,
p.antiguedad) in (select departamento, min(antiguedad) from profesores where departamento
= p.departamento group by departamento) order by d.codigo;
```

2.4. Ejercicio 10

Visualizar para cada departamento, la asignatura con menos créditos.

```
select d.nombre "DEPARTAMENTO", a.nombre "ASIGNATURA" from departamentos d
join asignaturas a on (a.departamento = d.codigo) where (d.codigo, a.creditos) in (select
a.departamento, min (a.creditos) from asignaturas a group by a.departamento);
```

2.5. Ejercicio 11

Visualizar para cada asignatura, el alumno de menor edad matriculado en el curso 2014-2015.

```
select a.nombre "ASIGNATURA", al.nombre || ' ' || al.apellido1 || ' ' || al.apellido2
"ALUMNO", al.fecha_nacimiento "FECHA DE NACIMIENTO" from asignaturas a join
matricular m on (m.asignatura = a.codigo) join alumnos al on (m.alumno = al.dni) where
m.curso = '14/15' and not exists (select * from alumnos al2 join matricular m2 on
(m2.alumno = al2.dni) where al2.fecha_nacimiento > al.fecha_nacimiento and m2.curso =
m.curso and m2.asignatura = m.asignatura) order by a.nombre;
```


Capítulo 3

Agrupación 3

3.1. Ejercicio 12

Visualizar el profesor con mayor carga de créditos. Considere la carga de créditos como la suma de los créditos de las asignaturas que imparte dicho profesor. Nota: Tenga en cuenta que un profesor puede impartir sólo una parte de una asignatura, por lo que se debe utilizar los créditos de la tabla impartir.

```
select p.nombre || ' ' || p.apellido1 "PROFESOR" from profesores p where id = ( select
    profesor from impartir group by profesor having sum(carga_creditos) >= all( select
sum(carga_creditos) from impartir group by profesor having sum(carga_creditos) is not null));
```

3.2. Ejercicio 13

Visualizar el departamento con mayor número de asignaturas a su cargo.

```
select d.nombre from departamentos d join asignaturas a on (d.codigo = a.departamento)
group by d.nombre having count(*) = (select max(count(*)) from asignaturas group by
departamento);
```

3.3. Ejercicio 14

Muestre el listado de los profesores que imparten menos de 10 créditos en total. Indique el código del profesor y el número de créditos que imparte.

```
select p.id "PROFESOR", sum(im.carga_creditos) "CREDITOS" from profesores p join
impartir im on (p.id = im.profesor) group by p.id having sum(im.carga_creditos) < 10;
```

3.4. Ejercicio 15

Listar los profesores que tienen una carga de créditos superior a la media. Use clausula HAVING y anide funciones de agrupación.

```
select p.nombre, p.apellido1, p.apellido2 from profesores p join impartir im on (p.id =  
im.profesor) group by p.nombre, p.apellido1, p.apellido2 having sum(im.carga_creditos) >  
(select avg(sum(carga_creditos)) from impartir group by profesor);
```

3.5. Ejercicio 16

Visualizar aquellos profesores que imparten 2 o más asignaturas en el curso 15/16 con una carga de créditos inferior a 6.5 en cada una de ellas.

```
select profesor from impartir where curso = '15/16' and carga_creditos < 6.5 group by  
profesor having count (asignatura) >= 2 order by count(asignatura);
```

Capítulo 4

Negativas 1

4.1. Ejercicio 17

Dar el nombre de las asignaturas hueso. Una asignatura se dice hueso si ningún alumno la ha superado.

```
select distinct a.nombre from asignaturas a where not exists (select asignatura from
matricular where asignatura = a.codigo and calificacion != 'SP' and calificacion != 'NP' and
calificacion is not null group by asignatura having count (*) > 0);
```

4.2. Ejercicio 18

Listar el nombre de los departamentos que no tienen ninguna asignatura con más de 6 créditos.

```
select nombre from departamentos where codigo not in (select departamento from
asignaturas where creditos > 6);
```

4.3. Ejercicio 19

Listar alfabéticamente los profesores que están en la lista negra de los alumnos. Si un profesor está en la lista negra de los alumnos, da clase en alguna asignatura optativa y en ella los alumnos no se matriculan para evitarlo. Tenga en cuenta que si hay dos turnos de la optativa, los alumnos tienden a evitar al profesor de ese turno, pero no a los de los otros grupos.

```
select p.nombre, p.apellido1, p.apellido2 from profesores p join impartir im on (p.id =
im.profesor) join asignaturas a on (im.asignatura = a.codigo) where a.caracter = 'OP' and
(im.asignatura, im.curso, im.grupo) not in (select asignatura, curso, grupo from matricular)
order by p.apellido1, p.apellido2, p.nombre;
```

4.4. Ejercicio 20

Mostrar las parejas de profesores que no tienen ningún alumno en común

```
select distinct p1.nombre || ' ' || p1.apellido1 || ' ' || p1.apellido2 "PROFESOR 1", p2.nombre  
|| ' ' || p2.apellido1 || ' ' || p2.apellido2 "PROFESOR 2" from profesores p1, profesores p2  
where p1.id < p2.id and not exists ((select alumno from matricular natural join impartir  
where profesor = p1.id) intersect (select alumno from matricular natural join impartir where  
profesor = p2.id));
```

4.5. Ejercicio 21

Mostrar el listado de profesores que no comparten ninguna de sus asignaturas (dos profesores comparten asignatura si imparten la misma asignatura independientemente del turno).

```
select distinct p1.nombre || ' ' || p1.apellido1 || ' ' || p1.apellido2 "PROFESOR 1", p2.nombre  
|| ' ' || p2.apellido1 || ' ' || p2.apellido2 "PROFESOR 2" from profesores p1, profesores p2  
where p1.id < p2.id and not exists ((select asignatura from impartir where profesor = p1.id)  
intersect (select asignatura from impartir where profesor = p2.id));
```

4.6. Ejercicio 22

Mostrar los nombres de asignaturas que no tienen dos alumnos matriculados del mismo municipio.

```
select nombre from asignaturas where codigo not in (select distinct a.codigo from asignaturas  
a join matricular m on (a.codigo = m.asignatura) join alumnos al on (m.alumno = al.dni)  
group by a.codigo, al.cpro, al.cmun having count(*) >= 2);
```

Capítulo 5

Complementarias 1

5.1. Ejercicio 23

Listar los alumnos matriculados en alguna asignatura que no esté impartida por el profesor de mayor edad. No sacar duplicados.

```
select distinct alumno from matricular natural join impartir join profesores p on (profesor =  
    p.id) where p.fecha_nacimiento not in (select min(fecha_nacimiento) from profesores);
```

5.2. Ejercicio 24

Visualizar aquellos alumnos matriculados en más de dos asignaturas a los que no les dé clase ningún profesor del departamento de 'Matematica Aplicada'.

```
select m.alumno from matricular m group by m.alumno having count(*) > 2 and m.alumno  
not in (select alumno from matricular m natural join impartir im join profesores p on (p.id =  
    im.profesor) join departamentos d on (p.departamento = d.codigo) where d.nombre =  
        'Matematica Aplicada');
```

5.3. Ejercicio 25

Visualizar la asignatura de más créditos de cada departamento entre aquellas no impartidas por un profesor nacido antes de 1970.

```
select a.codigo, a.departamento from asignaturas a where a.creditos = (select max(creditos)  
    from asignaturas where a.departamento = departamento) and a.codigo not in (select  
        im.asignatura from impartir im join profesores p on (im.profesor = p.id) where  
            to_char(p.fecha_nacimiento,'YYYY')<1970);
```

5.4. Ejercicio 26

Listar por orden de carga de créditos aquellos departamentos que tengan matriculados más de diez alumnos (en cualquiera de sus asignaturas).

```
select a.departamento, sum(carga_creditos) from asignaturas a join impartir im on (a.codigo = im.asignatura) where a.departamento in (select a2.departamento from asignaturas a2 join matricular m on (a2.codigo = m.asignatura) group by a2.departamento having count(m.alumno)>10) group by a.departamento order by sum(carga_creditos);
```

5.5. Ejercicio 27

Visualizar el número total de alumnos que posee cada profesor (si un alumno está matriculado en varias asignaturas de un profesor, contará varias veces), pero sólo de aquellos profesores que impartan alguna asignatura en la que el número de créditos prácticos es superior al de teóricos.

```
select im.profesor "profesor", count(*) "alumno" from impartir im join matricular m on (im.asignatura = m.asignatura) where profesor in (select im2.profesor from impartir im2 join asignaturas a on (im2.asignatura = a.codigo) where a.practicos > a.teoricos) group by im.profesor;
```

5.6. Ejercicio 28

De las asignaturas con más de 3 alumnos en el curso 15/16, visualizar el nombre del alumno más veterano (con la fecha de la primera matrícula más antigua).

```
select a.asignatura, m.alumno from (select m.asignatura from matricular m where m.curso = '15/16' group by m.asignatura having count(m.alumno) > 3) a join matricular m on (m.asignatura = a.asignatura) join alumnos al on (al.dni = m.alumno) where al.fecha_prim_matricula = (select min(al.fecha_prim_matricula) from alumnos al join matricular m on (m.alumno = al.dni) where m.asignatura = a.asignatura);
```

5.7. Ejercicio 29

De cada asignatura con menos de 16 alumnos matriculados en el curso 15/16 e impartida por el departamento más antiguo (con la menor fecha de creación), visualizar el nombre de la asignatura y el número de profesores que la imparte en el curso 15/16.

```
select a.nombre "ASIGNATURA", (select count(distinct im.profesor) from impartir im where im.asignatura = a.codigo and im.curso='15/16') "NÚMERO DE PROFESORES" from asignaturas a where a.codigo in (select m.asignatura from matricular m where m.curso = '15/16' group by m.asignatura having count(*)<16 and a.departamento in (select d.codigo from departamentos d where d.fecha_creacion in (select min(d.fecha_creacion) from departamentos d))) order by a.codigo;
```

5.8. Ejercicio 30

Visualizar por cada profesor que imparte al menos 2 asignaturas, a aquéllos que tienen a algún alumno nacido antes del año 1995.

```
select im.profesor "PROFESOR" from impartir im natural join matricular m where
  im.profesor in (select im.profesor from impartir im group by im.profesor having
    count(*) >= 2) and m.alumno in (select al.dni from alumnos al where
    to_char(al.fecha_nacimiento, 'YYYY') < '1995');
```