DML2.pdf

melasudajaja

Bases de Datos

2º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga

1. Nombre y apellidos de los profesores del departamento de Lenguajes

```
select p.nombre, p.apellido1, p.apellido2
from profesores p, departamentos d
where d.codigo = p.departamento and upper(d.nombre) like '%LENGUAJES%';
```

2. Usando la función NVL extraiga un listado con el código y el nombre de las asignaturas de las que está matriculado 'Nicolas Bersabe Alba'. Proporcione además el número de créditos prácticos, pero caso de ser nulo, debe salir "No tiene" en el listado. Indicación: advierta que prácticos es NUMBER y el literal 'No tiene' es VARCHAR2

```
select distinct a.codigo "ASIGNATURA", a.nombre "NOMBRE, nvl(to_char(a.practicos), 'No tiene')
"CREDITOS PRACTICOS"
from asignaturas a join matricular m on (a.codigo = m.asignatura) join alumnos al on (m.alumno =
al.dni)
where al.nombre = 'Nicolas' and apellido1 = 'Bersabe' and apellido2 = 'Alba';
```

3. Para cada profesor perteneciente al departamento "Ingenieria de Comunicaciones", proporcione el número de semanas completas que lleva trabajando en el departamento y diga que día se cumple un ciclo de semana completa. Use las funciones TO_CHAR y NEXT_DAY. Tenga en cuenta que si el día de la semana donde cumple el ciclo es el día actual, NEXT_DAY le llevará a la siguiente semana, cuando debería indicarse que el ciclo se cumple hoy.

```
select p.nombre, p.apellido1, p.apellido2, trunc((sysdate, p.antiguedad)/7) "ANTIGUEDAD",
next_day(sysdate-1,to_char(antiguedad, 'day')) "Se cumple una semana"
from profesores p join departamentos d on (p.departamento = d.codigo)
where d.nombre = 'Ingenieria de Comunicaciones';
```

4. Alumnos que tengan aprobada la asignatura 'Bases de Datos'.

```
select * from alumnos al
join matricular m on (al.dni = m.alumno)
join asignaturas a on (m.asignatura = a.codigo)
where a.nombre = 'Base de Datos'
and m.calificacion in ('AP', 'NT', 'SB', 'MH');
```

5. Obtenga un listado en el que aparezcan el identificador de los profesores, su nombre y apellidos así como el código de las asignaturas que imparte y su nombre.

```
select p.id, p.nombre, p.apellido1, p.apellido2,a.codigo, a.nombre "ASIGNATURA"
from profesores p
join impartir im on (p.id = im.profesor)
join asignaturas a on (im.asignatura = a.codigo);
```

6. Nombre y edad de parejas de alumnos que tengan el mismo primer apellido. Téngase en cuenta que los apellidos podrían estar en mayúscula o en minúscula.

```
select distinct  a1.nombre "Nombre 1", trunc(months_between(sysdate, a1.fecha_nacimiento)/12)
"Edad 1", a2.nombre "Nombre 2", trunc(months_between(sysdate, a2.fecha_nacimiento)/12) "Edad
2"
from alumnos a1, alumnos a2
where a1.dni!=a2.dni AND a1.apellido1 = a2.apellido1;
```

7. Combinaciones de apellidos que se pueden obtener con los primeros apellidos de alumnos nacidos entre los años 1995 y 1996, ambos incluidos. Se recomienda utilizar el operador BETWEEN ... AND ... para expresar el rango de valores.

```
select al1.apellido1 "Primer Apellido", al2.apellido1 "Segundo Apellido"
from alumnos al1, alumnos al2
where al1.dni < al2.dni
and(to_char(al1.fecha_nacimiento, 'YYYY')between 1995 and 1996)
and(to_char(al2.fecha_nacimiento, 'YYYY')between 1995 and 1996);
```

8. Nombre y apellidos de parejas de profesores cuya diferencia de antigüedad (en valor absoluto) sea inferior a dos años y pertenezcan al mismo departamento. Muestre la antigüedad de cada uno de ellos en años.

```
select p1.nombre, p1.apellido1, p1.apellido2,  trunc ((sysdate - p1.antiguedad)/365) "Antiguedad1",
p2.nombre, p2.apellido1, p2.apellido2,  trunc ((sysdate - p2.antiguedad)/365) "Antiguedad2" from
profesores p1, profesores p2
where p1.departamento = p2.departamento and p1.id < p2.id and trunc (abs(p1.antiguedad -
p2.antiguedad)/365) <2;
```

9. Construya un listado en el que se muestren todos los posibles emparejamientos heterosexuales que se
pueden formar entre los alumnos matriculados en la asignatura de código 112 donde la nota de la
mujer es mayor que la del hombre y ambos se matricularon en la misma semana. En el listado muestre primero el nombre de la mujer y a continuación el del hombre. Etiquete las columnas como
"Ella" y "El" respectivamente. Para el cálculo de la semana use la función de conversión TO_CHAR.

```
select al1.nombre ||' '|| al1.apellido1 ||' '|| al1.apellido2 "Ella", al2.nombre ||' '|| al2.apellido1 ||' '||
al2.apellido2 "El" from alumnos al1 join matricular m1 on (al1.dni = m1.alumno), alumnos al2 join
matricular m2 on (al2.dni = m2.alumno) where m1.asignatura = 112 and m2.asignatura = 112 and
al1.genero = 'FEM' and al2.genero = 'MASC'
and to_char(al1.fecha_prim_matricula, 'WW') = to_char(al2.fecha_prim_matricula, 'WW')
and decode(m1.calificacion, 'MH', 10, 'SB', 9, 'NT', 8, 'AP', 6, 'SP', 4, 0) >  decode(m2.calificacion, 'MH',
10, 'SB', 9, 'NT', 8, 'AP', 6, 'SP', 4, 0);
```

10.Tríos de asignaturas pertenecientes a la misma materia. Debe presentarse el nombre de las 3
asignaturas seguido del código de la materia a la que pertenecen.

```
select a1.nombre "Asignatura 1", a2.nombre "Asignatura 2", a3.nombre "Asignatura 3", a1.cod_materia "Materia"
from asignaturas a1, asignaturas a2, asignaturas a3
where a1.cod_materia = a2.cod_materia and a1.cod_materia = a3.cod_materia and a1.codigo < a2.codigo and a2.codigo < a3.codigo;
```

11.Muestre el nombre, apellidos, nombre de la asignatura y las notas obtenidas por todos lo alumnos con más de 22 años. Utilice la función DECODE para mostrar la nota como (Matricula de Honor, Sobresaliente, Notable, Aprobado, Suspenso o No Presentado). Ordene por apellidos y nombre del alumno

```
select a.nombre, a.apellido1, a.apellido2, asi.nombre asignatura,
decode(m.calificacion, 'MH', 'Matricula de Honor', 'SB', 'Sobresaliente', 'NT', 'Notable', 'AP', 'Aprobado', 'SP', 'Suspenso', 'NP','No presentado') "Notas"
from alumnos a, asignaturas asi, matricular m
where months_between(sysdate, a.fecha_nacimiento)>12*22
and m.alumno = a.dni and m.asignatura = asi.codigo order by 2,3,1;
```

12.Nombre y apellidos de todos los alumnos a los que les de clase Enrique Soler. Tenga en cuenta que hay que utilizar los atributos ASINGNATURA, GRUPO y CURSO de las tablas IMPARTIR y MATRICULAR. Cada alumno debe aparecer una sola vez. Ordénelos por apellidos, nombre.

```
select al.nombre, al.apellido1, al.apellido2
from impartir im natural join matricular m join alumnos al on (al.dni = m.alumno)
join profesores p on (im.profesor = p.id) where p.nombre = 'Enrique' and p.apellido1 = 'Soler' order by al.apellido1, al.apellido1, al.nombre;
```

13.Nombre y apellidos de los alumnos matriculados en asignaturas impartidas por profesores del
departamento de 'Lenguajes y Ciencias de la Computación'. El listado debe estar ordenado alfabéticamente

```
select al.nombre, al.apellido1, al.apellido2 from impartir im natural join matricular m join alumnos al on (m.alumno = al.dni) join profesores p on (im.profesor = p.id) join departamentos d on (p.departamento = d.codigo) where d.nombre = 'Lenguajes y Ciencias de la Computacion'
order by al.apellido1, al.apellido2, al.nombre;
```

14.Listado con el nombre de las asignaturas, nombre de la materia a la que pertenece y nombre, apellidos y carga de créditos de los profesores que la imparten. El listado debe estar ordenado por código de materia y por orden alfabético inverso del nombre de asignatura.

```
select a.nombre "Asignatura", mat.nombre "Materia", p.nombre || ' ' || p.apellido1 || ' ' ||
p.apellido2 "Profesor", im.carga_creditos from asignaturas a
join materias mat on (a.cod_materia = mat.codigo) join impartir im on (a.codigo = im.asignatura) join
profesores p on (im.profesor = p.id) where im.carga_creditos is not null
order by a.cod_materia, a.nombre desc;
```

15.Listado con el nombre de asignatura, nombre de departamento al que está asignada, total de créditos y porcentaje de créditos prácticos, ordenado decrecientemente por el porcentaje de créditos prácticos. Aquellas asignaturas cuyo número de créditos totales, prácticos o teóricos no está especificado no deben salir en el listado.

```
select a.nombre "Asignatura", d.nombre "Departamento", a.creditos,  trunc
(a.practicos*100/a.creditos,2) "%Practicos" from asignaturas a
join departamentos d on (a.departamento = d.codigo)
where a.creditos is not null and a.practicos is not null and a.teoricos is not null
order by "%Practicos" desc;
```

16.Utilice las operaciones de conjuntos para extraer los códigos de las asignaturas que no son impartidas por ningún profesor.

```
 select codigo from asignaturas minus select asignatura from impartir ;
```

17.Muestre todos los emails almacenados en la base de datos (tablas de Profesores y Alumnos). Si un email aparece repetido en dos tablas distintas también deberá aparecer repetido en la consulta. Evite los NULL.

```
select email from profesores where email is not null
union all
select email from alumnos where email is not null;
```

18.Utilice las operaciones de conjuntos para buscar alumnos que puedan ser familia de algún profesor, es decir, su primer o segundo apellido es el mismo que el primer o segundo apellido de un profesor aunque no necesariamente en el mismo orden. Muestre simplemente los apellidos comunes.

```
(select upper(apellido1) from alumnos where apellido1 is not null union
select upper(apellido2) from alumnos where apellido2 is not null) intersect
(select upper(apellido1) from profesores where apellido1 is not null union  select upper(apellido2)
from profesores where apellido2 is not null);
```

19.Apellidos que contienen la letra elle ('ll') tanto de alumnos como de profesores.

```
(select apellido1 from alumnos where upper(apellido1) like '%LL%')
union
(select apellido2 from alumnos where upper(apellido2) like '%LL%')
union
(select apellido1 from profesores where upper(apellido1) like '%LL%')
union
(select apellido2 from profesores where upper(apellido1) like '%LL%');
```

20.Idem que la anterior pero sustituya la 'll' por una 'y'. Utilice REPLACE

```
(select replace(apellido1, 'll', 'y') from alumnos where upper(apellido1) like '%LL%')
union
(select replace(apellido2, 'll', 'y') from alumnos where upper(apellido2) like '%LL%')
union
(select replace(apellido1, 'll', 'y') from profesores where upper(apellido1) like '%LL%')
union
(select replace(apellido2, 'll', 'y') from profesores where upper(apellido2) like '%LL%');
```

21.Busque una incongruencia en la base de datos, es decir, asignaturas en las que el número de créditos teóricos + prácticos no sea igual al número de créditos totales. Muestre también los profesores que imparten esas asignaturas.

```
select  asi.nombre, profesor
from asignaturas asi left outer join impartir i on asi.codigo = i.asignatura
where nvl(creditos, 0)<>nvl(practicos, 0)+nvl(teoricos, 0); */
```

22.Muestre en orden alfabético los nombres completos de todos los profesores y a su lado el de sus directores si es el caso (si no tenemos constancia de su director de tesis dejaremos este espacio en blanco, pero el profesor debe aparecer en el listado).

```
select p1.nombre "Nombre P1", p1.apellido1 "Apellido P1", p1.apellido2 "Apellido 2",
p2.nombre "Nombre P2", p2.apellido1 "Apellido P2", p2.apellido2 "Apellido 2"
from profesores p1 left outer join profesores p2
on p1.director_tesis = p2.id ORDER BY 2,3,1; */
```

23.Muestre el nombre y apellidos de cada profesor junto con los de su director de tesis y el número de tramos de investigación del director. Recuerde que el director de tesis de un profesor viene dado por el atributo DIRECTOR_TESIS y el número de tramos se encuentra en la tabla INVESTIGADORES. Los nombres de cada profesor y su director deben aparecer con el siguiente formato: 'El Director de Angel Mora Bonilla es Manuel Enciso Garcia-Oliveros'.

```
select 'El director de ' || p1.nombre || '' || p1.apellido1 || '' || p1.apellido2 || 'es' ||
p2.nombre || '' || p2.apellido1 || '' || p2.apellido2 "TESIS",
nvl (inn.tramos, 0) "Tramos
from profesores p1
join profesores p2 on (p1.director_tesis = p2.id)
left outer join investigadores inn on (p2.id = inn.id_profesor);
```

24.Liste el nombre de todos los alumnos ordenados alfabéticamente. Si dicho alumno tuviese otro alumno que se ha matriculado exactamente a la vez que él, muestre el nombre de este segundo alumno a su lado

```
select al1.nombre, al1.apellido1, al1.apellido2, al2.nombre, al2.apellido1, al2.apellido2,
from alumnos al1
left outer join alumnos al2 on (al1.fecha_prim_matricula = al2.fecha_prim_matricula and al1.dni <
al2.dni)
order by 2,3,1;
select * from sol_2_24;
```

25.Listado con el nombre de todas las asignaturas. En caso de que exista, para cada asignatura se muestra el curso, grupo y nombre y primer apellido del profesor que la imparte.

```
select a.nombre "asignatura", im.curso, im.grupo, p.nombre, p.apellido1 from asignaturas a
left outer join impartir im on (a.codigo = im.asignatura) left outer join profesores p on (im.profesor =
p.id) order by a.codigo;
```

26.Nombres e identificador de los profesores que nunca han impartido grupo.

```
select nombre, id from profesores where id not in  (select profesor from impartir);
```

27.Nombre y apellidos de 2 alumnas matriculadas de la asignatura de código 115 . Use ROWNUM para filtrar el número de tuplas que se desea (2 en este caso). Las tuplas repetidas deben filtrarse también.

```
select nombre, apellido1, apellido2 from alumnos where rownum <=2
and upper(genero) like 'FEM%'
and dni in(select alumno from matricular where asignatura=115);
```

28.Muestre todos los datos de los profesores que no son directores de tesis.

```
select *
from profesores where id not in(select director_tesis from profesores where director_tesis is not
null);
```

29.Liste el nombre y código de las asignaturas que tienen en su mismo curso otra con más créditos que ella

```
select a.nombre, a.codigo from asignaturas a
where exists (select * from asignaturas where nvl(creditos,0) > nvl(a.creditos,0)
and nvl(curso,0) = nvl(a.curso,1));
```

30.Use las operaciones de conjuntos y la consulta anterior para mostrar las asignaturas que tienen el máximo número de créditos de su curso.

```
(select nombre, codigo from asignaturas) minus
(select a.nombre, a.codigo from asignaturas a
where exists (select * from asignaturas where nvl(creditos,0) > nvl(a.creditos,0)  and nvl(curso,0) =
nvl(a.curso,1)));
```