

DML1.pdf



melasudajaja



Bases de Datos



2º Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga**

1. Hallar el nombre y dos apellidos de los profesores del departamento de código 1. Recordad que la igualdad en SQL se escribe con = y no con == como en C o C++.

```
select nombre, apellido1, apellido2 from docencia.profesores where departamento = 1;
```

2. Hallar el nombre y dos apellidos de los profesores de todos los departamentos salvo el de código 3. Para el operador "distinto" se puede utilizar != o bien <> .

```
select nombre, apellido1, apellido2 from docencia.profesores where departamento != 3;
```

3. Hallar el nombre y dos apellidos de los profesores cuyo correo está en el servidor "lcc.uma.es". En este caso hay que utilizar el operador LIKE. Recordad que nombre LIKE 'A%' será verdadero si nombre empieza por 'A'.

```
select nombre, apellido1, apellido2 from docencia.profesores where email like '%@lcc.uma.es';
```

4. Mostrar la lista de alumnos que no disponen de correo electrónico. Recordad el uso del operador IS NULL . Probad poniendo la expresión WHERE email = NULL . ¿Qué ocurre?

```
select nombre, apellido1, apellido2 from docencia.alumnos where email IS NULL;
```

5. Liste el nombre de las asignaturas de tercero, informando del total de créditos, de la proporción de teoría y de prácticas en tanto por ciento.

```
select nombre, creditos, (teoricos/creditos)*100 "Teoricos(%)", (practicos/creditos)*100  
"Practicos(%) " from docencia.asignaturas where curso = 3;
```

6. Muestre la lista de las notas de la asignatura 112 de la tabla MATRICULAR. Liste el código del alumno junto a su nota ordenado por el primero. Usad ORDER BY.

```
select alumno, calificacion from matricular where asignatura = 112 order by calificacion;
```

7. Haga la consulta 4 pero muestre la lista como: El alumno no dispone de Correo. Utilizad CONCAT o bien, el operador ||.

```
select 'El alumno '||nombre||' no dispone de correo electronico' from docencia.alumnos where  
email IS NULL;
```

8. Hallar el nombre y dos apellidos de los profesores que ingresaran antes de 1990. Es decir su fecha de ingreso es anterior a TO_DATE('01/01/1990', 'DD/MM/YYYY')

```
select nombre, apellido1, apellido2 from profesores where antiguedad <TO_DATE('01/01/1990',  
'DD/MM/YYYY');
```

9. Hallar el nombre y dos apellidos de los profesores que tengan menos de 30 años (use la fecha del sistema y la función MONTHS_BETWEEN).

```
select nombre, apellido1, apellido2, fecha_nacimiento, sysdate from profesores where  
ROUND(MONTHS_BETWEEN(SYSDATE, FECHA_NACIMIENTO)/12)<30;
```

10. Liste en mayúsculas el nombre y dos apellidos de los profesores que tienen más de 3 trienios. A un profesor se le concede un trienio cuando cumple tres años desde su ingreso. Pero si lleva 8 años y 11 meses solo tiene 2 trienios hasta que no cumpla los 9 años exactos. Use la función TRUNC para un cálculo correcto de los trienios. Muestre el número de trienios acumulados también. Renombre la columna de los trienios utilizando el Alias de columna.

```
select upper(nombre) NOMBRE, upper(apellido1) APELLIDO1, upper(apellido2) APELLIDO2,  
TRUNC((MONTHS_BETWEEN(SYSDATE, ANTIGUEDAD)/12)/3) "Trienios" from profesores where  
TRUNC((MONTHS_BETWEEN(SYSDATE, ANTIGUEDAD)/12)/3)>=(3*3);
```

11. Liste el nombre de todas las asignaturas que contienen en su nombre las palabras 'Bases de Datos'. Renombre dicha cadena en el listado como 'Almacenes de Datos'. Use la función REPLACE y el operador LIKE

```
select REPLACE(upper(NOMBRE), UPPER('Bases de Datos'), UPPER('Almacenes de Datos'))  
NOMBRE from asignaturas where UPPER(NOMBRE) LIKE '%BASES DE DATOS';
```

12. Muestre el nombre y créditos de todas las asignaturas obligatorias y optativas. Las asignaturas que no tienen asignado el valor de créditos debe poner NO ASIGNADO. Utilice la función NVL(expr1, expr2) que devuelve expr1 siempre que ésta no sea nula y expr2 en caso contrario. Aproveche que obligatorias y optativas comienzan ambas por el mismo carácter para simplificar la consulta (caracter LIKE 'O_').

```
select nombre, nvl(TO_CHAR(creditos), 'No asignado') "CREDITOS" from asignaturas where UPPER  
(caracter) like 'O_';
```

13. Informe de los alumnos que se han matriculado hace menos de dos meses

```
select * from alumnos where MONTHS_BETWEEN(SYSDATE, FECHA_PRIM_MATRICULA)<2;
```

14. Informe de los alumnos que entraron en la Universidad con menos de 18 años.

```
select * from alumnos where (MONTHS_BETWEEN(FECHA_PRIM_MATRICULA,  
FECHA_NACIMIENTO)/12)<18;
```

15. Informe de los alumnos que se matricularon en la universidad un lunes.

```
select * from alumnos where TO_CHAR(FECHA_PRIM_MATRICULA, 'DAY') like 'lunes%';
```


