

DML 1: Consultas simples

En primer lugar se practica la proyección y selección. SELECT lista de atributos WHERE condición :

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 ocurre en C o C++.
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 <> .
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'.
4. Mostrar el nombre de alumnos que no disponen de correo electrónico. Recordad el uso del operador IS NULL . Probad poniendo la expresión WHERE email = NULL . ¿Qué ocurre?
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.
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.
7. Mostrar la población de cada municipio español: nombre de municipio y suma de hombres y mujeres de todos municipios.

Uso de Funciones:

8. Haga la consulta 4 del bloque anterior pero muestre la lista como: El alumno no dispone de Correo. Utilizad CONCAT o bien, el operador ||.
9. 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')
10. 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).
11. 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.
12. 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
13. 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 (carácter LIKE 'O_').
14. Informe de los alumnos que se han matriculado hace menos de dos meses
15. Informe de los alumnos que entraron en la Universidad con menos de 18 años.
16. Informe de los alumnos que se matricularon en la universidad un lunes.

--Practica 1

--Ejercicio 1

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

--Ejercicio 2

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

--Ejercicio 3

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

--Ejercicio 4

```
select nombre, apellido1, apellido2 from alumnos
where email is null;
```

--Ejercicio 5

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

--Ejercicio 6

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

--Ejercicio 7

```
select nombre, hombres+mujeres POBLACION from municipio;
```

--Ejercicio 8

```
select DECODE(Genero, 'MASC','El alumno ','FEM','La alumna ') ||nombre ||' '|| apellido1 ||' '|| apellido2|| 'no
tiene Correo.' Correos from alumnos
where email is null;
```

--Ejercicio 9

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

--Ejercicio 10

```
select nombre, apellido1, apellido2 from profesores
where trunc(months_between(sysdate, fecha_nacimiento)/12) < 30;
```

--Ejercicio 11

```
select upper(nombre), upper(apellido1), upper(apellido2), trunc((months_between(sysdate,
antiguedad)/12)/3) Trienios from profesores
where trunc((months_between(sysdate, antiguedad)/12)/3) > 3;
```

--Ejercicio 12

```
select replace(nombre, 'Bases de Datos', 'Almacenes de datos') Nombre from asignaturas
where nombre like '%Bases de Datos%';
```

--Ejercicio 13

```
select nombre, DECODE(Creditos, null, 'NO ASIGNADO', creditos) from asignaturas
where caracter like 'O%';
```

--Ejercicio 14

```
select dni, nombre, apellido1, apellido2, genero, direccion, telefono, email, fecha_nacimiento,
fecha_prim_matricula from alumnos
where trunc(months_between(sysdate, fecha_prim_matricula)) < 2;
```

--Ejercicio 15

```
select dni, nombre, apellido1, apellido2, genero, direccion, telefono, email, fecha_nacimiento,
fecha_prim_matricula from alumnos
where trunc(months_between(fecha_prim_matricula, fecha_nacimiento)/12) < 18;
```

--Ejercicio 16

```
select dni, nombre, apellido1, apellido2, genero, direccion, telefono, email, fecha_nacimiento,
fecha_prim_matricula from alumnos
where next_day(fecha_prim_matricula, 'Lunes') - fecha_prim_matricula = 7;
```