MYSQL

**SELECT \* FROM nombredelatabla;** 🡪 Consulto todas las columnas de la tabla nombredelatabla.

# FROM.

Para consultar determinadas columnas, si son varias se separan por comas

**SELECT columna 1, columna2 FROM nombredelatabla;**

# ALIAS DE TABLAS.

Suelen utilizarse cuando consultamos varias tablas y hay nombres de columnas que coinciden.

**SELECT P.apellidos, P.edad 2 FROM personas P;**

A la tabla personas se le asigna el alias P.

# ALIAS DE COLUMNAS.

El alias se pone entre comillas simples o dobles a la derecha de la columna deseada.

**SELECT apellidos ‘ap’ FROM personas;**

# COLUMNAS CALCULADAS.

En el siguiente ejemplo se obtiene el año de nacimiento restando al año actual la edad de la tabla personas, el calculo lo devuelve en una columna con un nombre entre comillas.

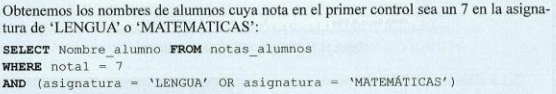
**SELECT apellidos, 2019-edad “Año nacimiento” FROM personas;**

Por cada nombre de alumno la nota media de los tres controles:

**SELECT nombre\_alumno, (nota1 + nota2 + nota3)/3 “nota media” FROM notas\_alumnos;**

Obtenemos los nombres de alumnos cuya nota media supere 6:

**SELECT Nombre\_Alumno FROM notas\_alumno WHERE (nota1 + nota2 + nota3) /3 > 6;**



# WHERE

Obtiene aquellas filas que cumplan la condición expresada.

**SELECT columna1, columna2 FROM nombredelatabla WHERE condición;**

**WHERE nota1 = 5**

**WHERE nota1= 6 OR nombre\_alumno LIKE ‘%Martinez%’**

**WHERE (nota1 > 5) AND (nota2 > 5 OR curso = 1)**

# ORDER BY

Sirve para ordenar los resultados. ASC es ascendentemente, es la opción por defecto, no es necesario escribirla, DESC es la ordenación descendente.

**SELECT….. FROM…… ORDEN BY Nombre\_alumno DESC;**

Se puede ordenar la consulta por la nota media del alumno ascedentemente:

**SELECT Nombre\_alumno,(nota1+nota2+nota3)/3 “Nota media” FROM notas\_alumnos ORDER BY (nota1+nota2+nota3)/3**

# DISTINCT

Recupera las filas que son distintas. Si no se pone, devuelve todas las veces que aparecen las asignaturas de esta forma, las repetidas no las devuelve.

**SELECT DISTINCT Asignatura FROM notas\_alumno**

# TEST DE VALOR NULO

**SELECT \* FROM …. WHERE nota1 IS NULL;**

**SELECT \* FROM …. WHERE nota1 IS NOT NULL;**

# BETWEEN

Comprueba si un valor está comprendido o no dentro de un rango de valores

**SELECT \* FROM …. WHERE edad BETWEEN 14 AND 17;**

**SELECT \* FROM …. WHERE edad NOT BETWEEN 14 AND 17;**

# IN

Permite comprobar si una expresión pertenece o no a un conjunto de valores.

Obtener los datos de los libros cuyas Unidades\_Vendidas sean 1200,1500 o 2000

**SELECT \* FROM libros WHERE unidades\_vendidas IN (1200,1500,2000);**

**SELECT \* FROM libros WHERE unidades\_vendidas not IN (1200,1500,2000);**

# OPERADOR LIKE

Consultas del tipo: Obtener los datos de los alumnos cuya columna Nombre\_alumno empiece por la letra C:

**SELECT \* FROM notas\_alumnos WHERE nombre\_alumno LIKE ’C%’;**

Obtener los datos de los alumnos que tengan el apellido ‘Sanchez’ dentro de la columna Nombre\_alumno:

**SELECT \* FROM notas\_alumno WHERE nombre\_alumno LIKE ‘%Sanchez%’;**

Aquellas filas cuya columna Nombre\_alumno contenga por lo menos una ‘M’:

**SELECT \* FROM notas\_alumno WHERE nombre\_alumno LIKE ‘%M%’**

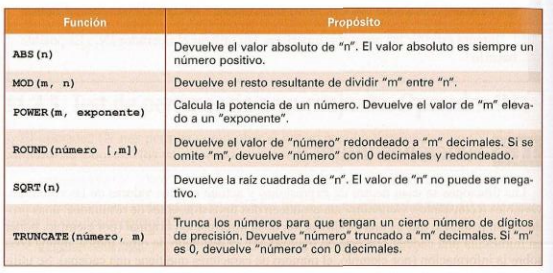
Aquellas filas cuya columna Nombre\_alumno contenga 3 caracteres y termine en M:

**SELECT \* FROM notas\_alumno WHERE nombre\_alumno LIKE ‘\_\_M’;**

Aquellas filas cuya columna Nombre\_alumno contenga cualquier cadena cuyo segundo carácter sea una ‘r’:

**SELECT \* FROM notas\_alumno WHERE nombre\_alumno LIKE ‘\_r%’;**

# FUNCIONES ARITMETICAS



Consulta que calcula el valor absoluto de -20, el resto de dividir 25 entre 4, la potencia de 2 elevado a 4 y la raíz cuadrada de 81:

**SELECT ABS (-20), MOD (25,4), POWER(2,4), SQRT(81);**

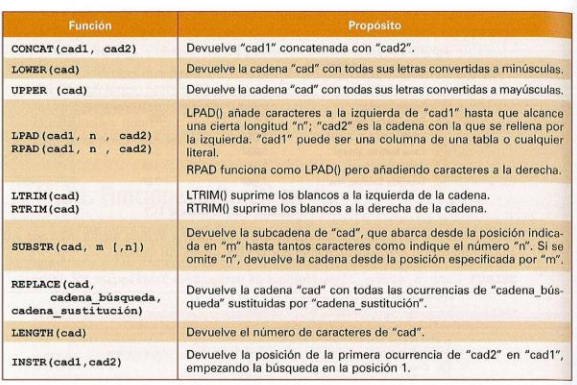
Obtener por cada alumno la nota media sin redondear y la nota media redondeada a dos posiciones ordenando ascendentemente por la nota media sin redondear:

**SELECT nombre\_alumno, (nota1+nota2+nota3)/3 “media no redondeada”,ROUND ((nota1+nota2+nota3)/3), 2) “media redondeada” FROM notas\_alumnos ORDER BY 2;**

Obtener de cada alumno la nota media sin truncar y la nota media truncada a 0 decimales ordenando descendentemente por la nota media sin truncar:

**SELECT nombre\_alumno, (nota1+nota2+nota3) /3 “media no truncada”, TRUNCATE ((nota1+nota2+nota3) /3, 0) “media truncada” FROM notas\_alumnos ORDER BY 2 DESC;**

# FUNCIONES DE CADENAS DE CARACTERES

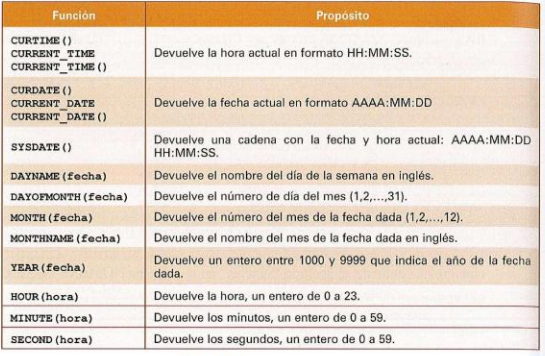


**select dni, concat(nombre,apellidos) 'nombre+apellido'**

Obtén los datos de los alumnos cuyo nombre de población tenga menos de 7 caracteres:

**select \* from alumnos where character\_length(poblacion)< 7;**

# FECHAS Y HORAS



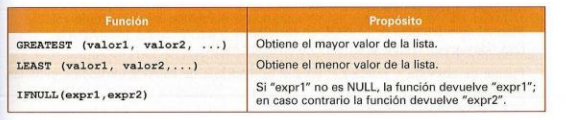
# EDAD A PARTIR DE UNA FECHA DE NACIMIENTO

Consultar el DNI, NOMBRE, APELLIDOS, CURSO, NIVEL, CLASE Y edad de todos los alumnos ordenado por APELLIDOS Y NOMBRE ascendentemente.

Se aplica la función timestampdiff(), esta función recibe como argumentos **la unidad de tiempo en la que queremos recibir la respuesta y las dos fechas entre las que queremos obtener la diferencia**.

**Select dni, nombre, apellidos, curso, nivel, clase, timestampdiff (year, Fecha\_nac, curdate()) as edad from alum2006 order by apellidos, nombre;**

# FUNCIONES DE COMPARACION



Obtener el nombre de alumno, la mayor nota de las tres que tiene y la menor:

**SELECT nombre\_alumno, GREATEST (nota1, nota2, nota3) “Mayor nota”, LEAST (nota1, nota2, nota3) “Menor nota” FROM notas\_alumnos;**

En la siguiente consulta muestra las tres notas de los alumnos. Si alguna nota tiene valor nulo, se visualizará en su columna la palabra ‘NP’:

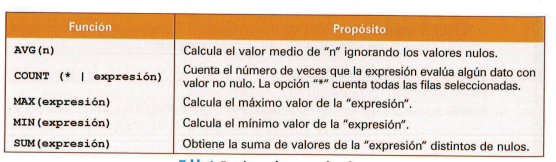
**SELECT nombre\_alumno, IFNULL(nota1, ‘NP’) “Nota 1”,**

**IFNULL (nota2, ‘NP’) “Nota 2”,**

**IFNULL (nota3, ‘NP’) “Nota 3”**

**FROM notas\_alumnos;**

# CONSULTAS RESUMEN



Cálculo del salario medio de los empleados:

**SELECT AVG(SALARIO) “Salario medio” FROM empleados;**

Cálculo del salario medio de los empleados del departamento 10:

**SELECT AVG(salario) FROM empleados WHERE dept\_no = 10;**

Cálculo del número de filas de la tabla;

**SELECT COUNT(\*) “Filas” FROM empleados;**

Cálculo del máximo salario:

**SELECT MAX(salario) FROM empleados;**

Obtener el apellido mínimo (alfabéticamente) de los empleados del departamento 20:

**SELECT MIN(apellido) FROM empleados WHERE dept\_no = 20;**

Cálculo de la suma de salarios de todos los empleados:

**SELECT SUM(salario FROM empleados;**

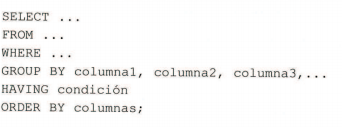
# DISTINCT + COUNT

Distinct realiza una selección de filas cuyos valores en la columna especificad no estén duplicados. El formato es:

**COUNT (DISTINCT oficio) “oficios” FROM empleados;**

# CONSULTAS AGRUPADAS GROUP BY

Esta cláusula sirve para calcular propiedades de uno o más conjuntos de filas. También existe una condición de búsqueda para grupos de filas, **HAVING**, se emplea para controlar cuál de los conjuntos de filas se visualiza.



Obtener el número de empleados que hay en cada departamento visualizando la columna DEPT\_NO y el número de empleados.

**SELECT dept\_no, COUNT(\*) FROM empleados GROUP BY dept\_no;**

Obtener los departamentos cuyo número de empleados es mayor que 3:

**SELECT dept\_no, COUNT(\*) FROM empleados GROUP BY dept\_no HAVING COUNT(\*) > 3;**

La consulta anterior ordenando descendentemente por número de empleados:

**SELECT dept\_no, COUNT(\*) FROM empleados GROUP BY dept\_no HAVING COUNT(\*) > 3 ORDER BY COUNT(\*) DESC;**

# COMBINACION EXTERNA

Permite seleccionar algunas filas de una tabla, aunque éstas no tengan correspondencia con las filas de la otra tabla con la que se combina.

Una consulta donde se visualice el número de departamento, el nombre y el número de empleados que tiene.

**SELECT d.dept\_no, dnombre, COUNT(e.emp\_no) FROM empleados e, departamentos d, WHERE e.dept\_no = d.dept\_no GROUP BY d,dept\_no, dnombre.**

Con esta sentencia solo aparecen los departamentos que tienen empleados, el departamento 40 se pierde, para que aparezca utilizamos la siguiente sentencia.

**SELECT d.dept\_no, dnombre, COUNT(e.emp\_no) FROM departamentos d LEFT JOIN empleados e, ON e.dept\_no = d.dept\_no GROUP BY d,dept\_no, dnombre.**

**Daría el mismo resultado:**

**SELECT d.dept\_no, dnombre, COUNT(e.emp\_no) FROM empleados e RIGHT JOIN departamentos d, ON e.dept\_no = d.dept\_no GROUP BY d,dept\_no, dnombre.**

En esta combinación por cada fila de la primera tabla de la izquierda que no se combine con ninguna fila de segunda tabla, se añade una fila al resultado con los datos de la primera fila y suponiendo un valor NULL para la segunda tabla.

# SUBCONSULTAS

Obtener el apellido, el salario y el departamento de los empleados con el mismo departamento que ‘ARROYO’ y con el salario menor que el salario de ‘ARROYO’.

**SELECT apellido, salario, dept\_no FROM empleados where dept\_no = (SELECT dept\_no FROM empleados WHERE =’ARROYO’) AND salario < (SELECT salario FROM empleados WHERE apellido =’ARROYO’);**

Obtener los datos de los departamentos que NO tengan empleados.

**SELECT \* FROM departamentos WHERE dept\_no NOT IN (SELECT DISTINCT dept\_no From empleados);**

