**Introducción**

En la sesión pasada, conocimos el Sistema Gestor de Bases de Datos y describimos algunas tablas para conocer su estructura y posteriormente escribimos algunas consultas para obtener algunos datos de la base. En esta sesión continuaremos revisando la estructura de las consultas añadiendo más opciones que darán pie a un filtrado más especializado.

# Búsqueda de patrones

La [búsqueda de patrones](https://es.wikipedia.org/wiki/B%C3%BAsqueda_de_patrones) es una técnica ampliamente utilizada por distintos lenguajes de programación y Sistemas Gestores de Bases de Datos. En bases de datos, consiste en, dado un patrón representado mediante [cadenas](https://es.wikipedia.org/wiki/Cadena_de_caracteres) encontrar un registro dentro de la base de datos en donde el campo coincida con el patrón buscado.

Cuando una cadena caza con el patrón a buscar, se dice que este caza con el patrón buscado.

En MySQL, podemos hacer búsqueda de patrones mediante la restricción LIKE que puede usarse en combinación de la cláusula WHERE de una consulta. La sintaxis de esta restricción es la siguiente:

<expresión> LIKE <patrón>

La expresión representa la columna sobre la cual queremos realizar la búsqueda y el patrón es una cadena que incluye el patrón correspondiente. En MySQL los patrones que podemos encontrar, usan los siguientes símbolos:

* % Caza con cualquier cadena, de cualquier longitud, incluso cadenas vacías.
* \_ Caza con un único carácter.

Veamos algunos ejemplos:

SELECT nombre

FROM persona

WHERE apellido\_paterno LIKE ‘S%’;

Esta consulta, obtiene el campo nombre de la tabla persona siempre y cuando el apellido paterno inicie con la letra S. Recordar que el % indica que puede ser reemplazado por cualquier cadena. Por ejemplo, puede encontrar las cadenas ‘Sánchez’, ‘Soto’, ‘Saldivar’.

SELECT nombre

FROM persona

WHERE apellido\_materno LIKE ‘%am%’;

Esta consulta obtiene el campo nombre de la tabla persona siempre y cuando el apellido materno, contenga subcadena Mendoz. El símbolo \_ indica que puede ser reemplazado por cualquier caracter. Por ejemplo, puede encontrar las cadenas ‘Mendoza’, ‘Mendoze’, ‘Mendozo’, etc.

SELECT nombre

FROM persona

WHERE apellido\_paterno NOT LIKE ‘%a’;

Como puede apreciarse en la consulta anterior, podemos combinar la restricción LIKE con los operadores lógicos. En este caso se traen los nombres de aquellas personas cuyo apellido no termina con a.

**Funciones de agrupamiento**

En la sesión pasada, estudiamos cómo traer o proyectar campos de una tabla, sin embargo, también es posible aplicar distintas funciones a los campos con el fin de agruparlos para obtener valores específico. Para ello se tienen algunas funciones, en esta sesión veremos los siguientes:

* SUM suma todos los valores de una columna
* COUNT cuenta los valores de una columna
* MIN obtiene el valor mínimo de una columna
* MAX obtiene el valor máximo de una columna
* AVG obtiene el promedio de los valores de una columna

Veamos algunos ejemplos:

SELECT sum(salario) AS “Salario Total”

FROM empleados

WHERE salario > 5000;

Esta consulta obtiene la suma de los salarios de los empleados de una empresa y sólo considera aquellos salarios mayores a 5000. La columna resultante se nombra como sum(salario), sin embargo, este nombre no es claro, por lo que la restricción AS nos permite agregar un alias a la columna y llamarla en su lugar Salario Total.

SELECT count(ocupacion) AS “Número de ocupaciones”

FROM persona;

Esta consulta cuenta el número de ocupaciones que hay en la columna ocupación. Es importante mencionar que la función count únicamente cuenta registros que no son nulos.

SELECT max(edad) AS “Mayor”

FROM persona;

SELECT min(edad) AS “Mayor”

FROM persona;

Estas consultas obtienen la edad más grande y más pequeña, respectivamente, en un conjunto de personas.

SELECT avg(calificacion) “Promedio”

FROM estudiantes

WHERE calificacion >= 6.0;

La consulta anterior, calcula el promedio de los estudiantes cuya calificación fue aprobatoria.

Como puedes apreciar, en general, la sintaxis de una función es

funcion(parámetros)

En el caso de la funciones de agrupamiento, éstas sólo reciben un parámetro pues toman una columna, sin embargo existen muchas más funciones que tienen otras utilidades. Llamamos a estas funciones de agrupamientos, pues reducen los valores de una columna un único valor.

Enlace de consulta para funciones de MySQL

<https://www.techonthenet.com/mysql/functions/>

# Agrupamientos

Ahora que sabes cómo operan las funciones de agrupamiento, estás listo para hacer tus primeros agrupamientos. Un agrupamiento, toma valores distintos de una columna y permite hacer un análisis detallado mediante las funciones de agrupamiento.

Por ejemplo, supongamos que, existe una tabla que almacena la ocupación de un conjunto de personas y quisiéramos saber cuántas personas hay por cada ocupación. Bueno, pues podemos hacer un agrupamiento y auxiliarnos de la función count.

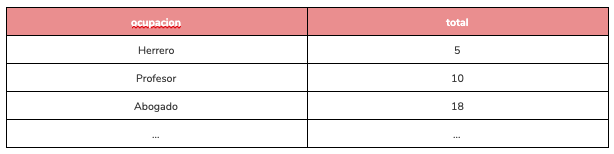
Para usar un agrupamiento, se debe usar la restricción GROUP BY que indica los campos por los cuales se realizará el agrupamiento. Los campos por los cuales se agrupará, deben aparecer en la consulta y debe haber al menos una función de agrupamiento proyectada.

SELECT ocupacion, count(\*) “Total”

FROM persona

GROUP BY ocupacion;

El símbolo \* en la función count, le indica a MySQL que se deben tomar todos los registros de la tabla. De esta forma el resultado luce como:



# Subconsultas

Una subconsulta consiste de consultas que incluyen otras consultas en su interior. Una subconsulta puede aparecer en la parte SELECT, en la parte FROM o en la parte WHERE de una consulta. Veamos algunos ejemplos.

SELECT nombre, correo

FROM contactos

WHERE direccion IN

(SELECT direccion

FROM ubicaciones

WHERE codigo\_postal = ‘01060’);

Esta consulta consta de una subconsulta, que se ejecuta primero. De esta forma, primero obtenemos todas las direcciones que se encuentran dentro del código postal 01060 y con los resultados obtenidos (un tabla), filtramos aquellos contactos que viven en esas direcciones.

SELECT (Calificacion / Alumnos) AS “Promedio”

FROM

(SELECT sum(calif) AS “Calificacion”, count(\*) AS “Alumnos”

FROM alumnos) AS subconsulta;

En esta consulta tomamos las columnas resultantes de la subconsulta. En este tipo de consultas MySQL pide poner un nombre a la subconsulta para poder referenciar, en caso de que sea necesario. Esta consulta primero calcula la suma de las calificaciones y el total de alumnos de una tabla y posteriormente se usan estos datos para calcular el promedio.

Notamos también que es posible usar operaciones aritméticas, como es el caso de la suma, resta, división y multiplicación.