



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC	2003-1	5046	Bases de Datos

PRÁCTICA No.	LABORATORIO DE	Bases de Datos	DURACIÓN (HORA)
9	NOMBRE DE LA PRÁCTICA	Funciones agregadas de Mysql	2

### 1 INTRODUCCIÓN

Las bases de datos son usadas a menudo para responder a preguntas como las siguientes: ¿Cuántas veces aparece un determinado dato en una tabla?, ¿Cuál es la edad máxima o mínima de un conjunto de personas? . Un gran número de pregunta como estas pueden ser contestadas gracias a las funciones que MySQL agrega, para manipular la información contenida en las tablas. Por ejemplo podemos utilizar la función count para contar un conjunto de filas y la función max para determinar el máximo valor contenido en una columna. En esta práctica estudiaremos algunas de las funciones agregadas mas importantes que MySQL soporta.

### 2 OBJETIVO (COMPETENCIA)

Utilizar las funciones adecuadas de Mysql, para dar respuesta a cuestionamientos que se presentan de manera práctica en la sección de desarrollo.

Formuló Ing. Alicia López Aguirre	Revisó M.C. Gloria Etelbina Chavez Valenzuela	Aprobó	Autorizó M.C. Miguel Ángel Martínez Romero
Maestro	Coordinador de la Carrera	Gestión de la Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 3 FUNDAMENTO

#### Funciones de agregados. (Aggregate functions)

A menudo queremos realizar operaciones sobre un conjunto de filas de una tabla. Por ejemplo, queremos sumar los contenidos de una columna Importe en toda la tabla, o sólo en las filas que cumplen cierta condición. Para eso existen ciertas funciones llamadas funciones de agregados (Aggregate functions). Veremos algunas de estas funciones.

#### Cálculos de fecha

MySQL ofrece muchas funciones que puedes usar para realizar cálculos con fechas, por ejemplo, para calcular edades o extraer partes de fechas. Por ejemplo para determinar cuantos años tiene una de tus mascotas, puedes calcular la edad como la diferencia entre la fecha de nacimiento y la fecha actual. Puedes hacerlo convirtiendo las dos fechas a días, coge la diferencia, y divídela por 365 (el número de días en un año) como se muestra en el sig. Ejemplo:

```
mysql> SELECT nombre, (TO_DAYS(NOW())-TO_DAYS(nacimiento))/365 FROM mascota;
```

nombre	(TO_DAYS(NOW())-TO_DAYS(nacimiento))/365
Bluffy	6.94
Claws	5.83
Buffy	10.68
Fang	9.39
Bowser	10.38
Chirpy	1.34
Whistler	2.10
Slim	3.71
Puffball	0.79

Aunque la consulta funcione, existen algunos puntos que podrían ser mejorados. Primero, el resultado podría ser revisado más fácilmente si las filas se presentaran ordenadas de alguna manera. Segundo, la cabecera de la columna edad no es muy significativa. El primer problema puede ser solucionado añadiendo una cláusula ORDER BY nombre para ordenar la salida por nombre. Para arreglar el tema del encabezamiento de columna, puedes darle un nombre a dicha columna de tal forma que aparezca una etiqueta diferente en la salida (esto es lo que se llama un alias de columna):

```
mysql> select nombre, (TO_DAYS(NOW())-TO_DAYS(nacimiento))/365 AS edad FROM mascota ORDER BY nombre;
```

Para ordenar la salida por edad en lugar de por nombre, puedes hacerlo usando simplemente una cláusula ORDER BY diferente:

```
mysql> select nombre, (TO_DAYS(NOW())-TO_DAYS(nacimiento))/365 AS edad FROM mascota ORDER BY edad;
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 3 FUNDAMENTO

Puede usarse una consulta similar para determinar la edad de la muerte para los animales que hayan muerto. Puedes determinar qué animales son estos comprobando si el valor muerte es NULL o no. Después, para aquellos que no tengan un valor NULL, calcular la diferencia entre los valores muerte y nacimiento:

```
mysql> select nombre, nacimiento, muerte,
-> (TO_DAYS(NOW())-TO_DAYS(nacimiento))/365 AS edad
-> FROM mascota WHERE muerte IS NOT NULL ORDER BY edad;
```

```
+-----+-----+-----+-----+
| nombre | nacimiento | muerte | edad |
+-----+-----+-----+-----+
| Bowser | 1989-08-31 | 1995-07-29 | 10.38|
+-----+-----+-----+-----+
```

La consulta usa muerte IS NOT NULL en lugar de muerte != NULL dado que NULL es un valor especial. Esto se explica más adelante. [Puedes consultar la sección [Working with NULL] del manual de MySQL.

¿Qué harías si quisieras saber qué animales cumplen años el mes que viene? Para este tipo de cálculos, año y día son irrelevantes, simplemente querrás extraer la parte mes de la columna nacimiento. MySQL ofrece muchas funciones de extracción de parte-de-fecha, como YEAR(),MONTH() y DAY(). La función apropiada para nuestro problema es MONTH(). Para ver cómo funciona, ejecuta una consulta rápida que muestre el valor de la fecha de nacimiento y el mes de nacimiento (MONTH(nacimiento)):

```
mysql> SELECT nombre, nacimiento, MONTH(nacimiento) FROM mascota;
```

Buscar animales que hayan nacido en el mes próximo es también sencillo de realizar. Suponte que Abril es el mes actual. Entonces el valor del mes es 4 y lo que buscas son animales nacidos en Mayo (mes 5):

```
mysql> SELECT nombre, nacimiento FROM mascota WHERE MONTH(nacimiento) = 5;
```

Existe una pequeña complicación si el mes actual es Diciembre, por supuesto. No puedes añadir simplemente uno al número de mes (12) y buscar animales nacidos en el mes 13, dado que no existe tal mes. En lugar de eso, debes buscar animales nacidos en Enero (mes 1).

Puedes escribir la consulta de tal forma que funcione independientemente del mes en el que estemos. De esa forma no tendrás que usar un número de mes en particular en la consulta. DATE\_ADD() te permite añadir un intervalo de tiempo a una fecha dada. Si añades un mes al valor de NOW(), y después extraes la parte del mes con MONTH(), el resultado produce el mes del cumpleaños que buscamos:



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 3 FUNDAMENTO

```
mysql> select NOW();
```

```
+-----+
| NOW() |
+-----+
| 2000-01-13 18:13:09 |
+-----+
```

```
mysql> SELECT nombre, nacimiento FROM mascota
```

```
-> WHERE MONTH(nacimiento) = MONTH(DATE_ADD(NOW(),INTERVAL 1 MONTH));
```

```
+-----+-----+
| nombre | nacimiento |
+-----+-----+
| Bluffy | 1993-02-04 |
+-----+-----+
```

Una manera diferente de conseguir los mismos resultados es añadir 1 al mes actual para conseguir el mes siguiente (tras usar la función módulo (MOD) para convertir el valor de mes actual en 0 si estamos en Diciembre (mes 12)):

```
mysql> SELECT nombre, nacimiento FROM mascota
```

```
-> WHERE MONTH(nacimiento) = MOD(MONTH(NOW()),12) +1;
```

```
+-----+-----+
| nombre | nacimiento |
+-----+-----+
| Bluffy | 1993-02-04 |
+-----+-----+
```

#### Contando filas

Las bases de datos son usadas a menudo para responder a la pregunta, ¿cuántas veces aparece un determinado tipo de datos en una tabla?. Por ejemplo, podrías querer saber cuántas mascotas tienes, o cuántas mascotas tiene cada propietario, o podrías querer realizar varios tipos de censos respecto a tus animales. Contar el número total de animales que tienes es lo mismo que preguntar ¿cuántas filas hay en la tabla mascota?, dado que hay sólo una fila por mascota. La función COUNT() cuenta el número de resultados no-NULL (N.T.: más adelante se indica una consulta SQL en la que se cuenta el número de animales por sexo, y aparece el símbolo NULL como parte de esa cuenta... lo cual invalida la afirmación que precede a este comentario!), así pues, la consulta a realizar para contar el número de animales tiene la siguiente forma:

```
mysql> SELECT COUNT(*) FROM mascota;
```

```
+-----+
| COUNT(*) |
+-----+
| 9 |
+-----+
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 3 FUNDAMENTO

Antes, conseguiste los nombres de las personas que poseen una mascota. Puedes usar COUNT() para averiguar cuántas mascotas tiene cada propietario:

```
mysql> SELECT propietario, COUNT(*) FROM mascota GROUP BY propietario;
```

Observa el uso de GROUP BY para agrupar todos los registros de cada propietario. Si no lo hubiéramos puesto, todo lo que conseguirías sería un mensaje de error:

```
mysql> SELECT propietario, COUNT(proprietario) FROM mascota;
```

ERROR 1140: Mixing of GROUP columns (MIN(),MAX(),COUNT()...) with no GROUP columns is illegal if there is no GROUP BY clause COUNT() y GROUP BY son útiles para la caracterización de tus datos de varias formas. Los siguientes ejemplos muestran difentes maneras para realizar operaciones de censo animal.

Número de animales por especies:

```
mysql> SELECT especie, COUNT(*) FROM mascota GROUP BY especie;
```

Número de animales por sexo:

```
mysql> SELECT sexo , COUNT(*) FROM mascota GROUP BY sexo;
```

```
+-----+-----+
| sexo  | COUNT(*) |
+-----+-----+
| NULL  |         1 |
| f     |         4 |
| m     |         4 |
+-----+-----+
```

(En este resultado, NULL indica "sexo desconocido")

El número de animales por combinación de especies y sexo:

```
mysql> SELECT especie , sexo, COUNT(*) FROM mascota GROUP BY especie, sexo;
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 3 FUNDAMENTO

No necesitas recuperar una tabla completa cuando uses COUNT(). Por ejemplo, la consulta anterior, cuando se realiza sólo sobre perros y gatos, se escribe así:

```
mysql> SELECT especie , sexo, COUNT(*) FROM mascota
-> WHERE especie = "perro" OR especie = "gato"
-> GROUP BY especie, sexo;
```

especie	sexo	COUNT(*)
gato	f	1
gato	m	1
perro	f	1
perro	m	2

O, si quieres conocer el número de animales por sexo sólo para animales de sexo conocido:

```
mysql> SELECT especie , sexo, COUNT(*) FROM mascota WHERE sexo IS NOT NULL GROUP BY especie, sexo;
```

COUNT ( DISTINCT Nombre\_de\_columna). Cuenta cuántos valores no NULL diferentes hay de la columna especificada, en las filas especificadas. El siguiente ejemplo produce el número de clientes distintos a los cuales se les han registrado ventas.

SUM(). Esta función totaliza una columna, dentro de las filas que cumplen una condición, o de todas las filas si no se especifica ninguna condición. En lo que sigue abreviaremos esta expresión y diremos simplemente "de las filas especificadas". Ejemplos: Sumar todos los importes de Ventas.

```
mysql> SELECT SUM(importe) FROM ventas;
```

SUM(importe)
1010.00



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 3 FUNDAMENTO

Sumar todos los importes de Ventas, con fecha del año 2003.

```
mysql> SELECT SUM(importe) FROM ventas WHERE YEAR(fecha)=2003;
```

```
+-----+
```

```
| SUM(importe) |
```

```
+-----+
```

```
|      NULL   |
```

```
+-----+
```

De manera similar podemos utilizar las siguientes funciones:

MAX(). Produce el máximo valor de una columna dentro de las filas especificadas.

MIN(). Produce el mínimo valor de una columna en las filas especificadas.

AVG(). Produce el promedio de los valores de una columna numérica, contando sólo las filas especificadas.

#### Problemas resueltos

1.- Obtener la suma de los salarios mayores a 11000 de un grupo de Empleados.

Solución:

Debemos obtener la suma de la columna Salario de aquellas filas de la tabla Empleados que tienen Salario mayor a 11000.

```
mysql> SELECT SUM(Salario) FROM Empleados WHERE Salario > 11000;
```

2.- Contar cuántas direcciones de clientes se tienen.

Solución:

Debemos obtener el número de filas de Clientes cuya columna Dirección no es NULL.

```
mysql>SELECT COUNT(Direccion) FROM Clientes;
```

Observar que como se especificó COUNT(Dirección) las direcciones NULL no se cuentan. En cambio SELECT COUNT(\*) FROM Clientes cuenta todas las filas, independientemente que tengan o no dirección (o alguna otra columna) NULL.

3.- ¿Cuál es el salario promedio de la empresa?

Solución:

```
mysql>SELECT AVG(Salario) FROM Empleados;
```

4.- ¿Cuál es el salario máximo pago por la empresa?

Solución:

```
mysql>SELECT MAX(Salario) FROM Empleados;
```



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 3 FUNDAMENTO

5.- ¿Cuáles son los empleados más antiguos de la empresa?

Solución:

Vamos a seleccionar todas las filas de la tabla Empleados con Fecha de ingreso igual a la fecha de ingreso mínima.

```
SELECT * FROM Empleados WHERE Fecha_ingreso = (SELECT MIN(Fecha_ingreso) FROM Empleados);
```

6.- Hallar los nombres de los empleados con salario máximo en la empresa.

Solución:

```
SELECT Nombre FROM Empleado WHERE Salario =(SELECT MAX(Salario) FROM Empleados);
```

7.-¿Cuántos salarios distintos se pagan en la empresa?

Solución:

```
SELECT COUNT(DISTINCT Salario) FROM Empleados;
```

#### La cláusula HAVING.

Después de un GROUP BY, todavía se pueden volver a seleccionar filas con una cláusula HAVING. Por ejemplo, podemos estar interesados, hipotéticamente, en las facturas con más de una línea. Entonces deberíamos usar

```
mysql>SELECT Factura, COUNT(*) FROM Lineas_Factura GROUP BY Factura HAVING COUNT(*) > 1;
```

Si no nos interesa ver el número de líneas podemos usar

```
mysql>SELECT Factura FROM Lineas_Factura GROUP BY Factura HAVING COUNT(*) > 1;
```

Otra variante más legible

```
mysql>SELECT factura, COUNT(*) AS cantidad FROM lineas_factura GROUP BY factura HAVING cantidad > 1;
```

El orden de las cláusulas vistas hasta ahora es SELECT.....FROM....., WHERE, GROUP BY, HAVING, ORDER BY

Recuerda que existen dos momentos en que se seleccionan filas, una con el WHERE antes de la agrupación y otro con el HAVING, después de la misma.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### 4 PROCEDIMIENTO (DESCRIPCIÓN)

<b>A EQUIPO NECESARIO</b>	<b>MATERIAL DE APOYO</b>
Computadora con acceso al servidor Mysql	Práctica impresa

### **B DESARROLLO DE LA PRÁCTICA**

- 1.- Introduce tu login y password para acceder al servidor de MySQL.
- 2.- Borra las bases de datos utilizadas en la práctica anterior si es que ya fueron revisadas por tu tutor.
- 2.- Crear la Base de datos llamada FACKBOOKXXX donde XXX es tu matrícula
- 3.- La base de datos contendra la siguiente tabla llamada CIA:

<b>name</b>	<b>region</b>	<b>area</b>	<b>population</b>	<b>gdp</b>
Yemen	Middle East	527970	14728474	23400000000
Zaire	Africa	2345410	44060636	18800000000
Zambia	Africa	752610	9445723	7900000000
Zimbabwe	Africa	390580	11139961	17400000000
Mexico	America	1346232	80000000	167340000000
EEUU	America	1568731	65000000	34647540000000

Donde:

Area: Es una medida en metros cuadrados

GDP: Es el producto interno bruto. Es una medida de la abundancia total generada por el país en Dolares en un año.

Esta tabla contiene en total 265 datos, por comodidad trabajaremos solo con el fragmento presentado arriba suponiendo que dicho fragmento representa a todos los países del mundo. Si desea obtener la tabla completa puede consultar la sig. dirección <http://www.cia.gov> que es la dirección de la agencia central de inteligencia conocida por sus siglas en ingles como CIA.

- 4.-Realice una consulta que muestre la población total del mundo.
- 5.-Enumere todas las regiones distintas que existen.
- 6.-Enumere los países que tienen un GDP mayor que el conjunto de Africa.
- 7.-Enumere cada región y los países que pertenecen a dicha región.
- 8.-Enumere las regiones con una población superior a 14 millones de habitantes.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA  
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)  
DOCUMENTO DEL SISTEMA DE CALIDAD**

## Formato para prácticas de laboratorio

### **B DESARROLLO DE LA PRÁCTICA**

- 9.-Para cada región, enumere región y número de países que contienen una población inferior a los 10 millones de habitantes.
- 10.-Obtenga el promedio del GPD mundial.
- 11.-Muestre el nombre de los países que tienen una superficie cuadrada mayor al promedio la superficie mundial.
- 12.-Muestre la región en la que se encuentran los países que tienen una superficie cuadrada menor al promedio de la superficie mundial.
- 13.-Muestre el nombre del país con mayor población.
- 14.-Muestre el nombre del país con menor población.
- 15.-Enumere para las diferentes regiones del mundo el promedio de población.
- 16.-Enumere los países que contienen un GDP mayor al conjunto de America.
- 17.-Obtenga el promedio de población para America.
- 18.-Obtenga el promedio de población para Africa.
- 19.-Muestre el nombre de la región con mayor población.
- 20.-Muestre el nombre de la región con menor población.

### **C CÁLCULOS Y REPORTE**

El alumno deberá entregar un reporte impreso que contenga las respuestas de cada uno de los puntos propuestos en la parte del desarrollo de esta práctica.

## **5 RESULTADOS Y CONCLUSIONES**

Al finalizar la práctica el alumno será capaz de manejar las principales funciones agregadas que soporta Mysql.

## **6 ANEXOS**