Artículo **Revista de Tecnología Informática**
Diciembre, 2018 Vol.2 No.7 16-22

Implementación de una función en Mysql para determinar la edad

Implementation of a function in Mysql to determine the age

ARRIETA-ZUÑIGA, Juan†*, LEON-OLIVARES, Eric, MARTINEZ-PAGOLA, Salvador, MAGGI-NATALE, Carlos y JIMENEZ-GARCÍA, Lizandro

Tecnológico Nacional de México/Instituto Tecnológico de Pachuca

ID 1er Autor: *Juan, Arrieta-Zuñiga* / **ORC ID:** 0000-0001-9698-788x, **Researcher ID Thomson**: M-9232-2018, **arXiv:** jaaz1967)

ID 1er Coautor: *Eric, Leon-Olvares* / **ORC ID:** 0000-0002-6342-3719, **Researcher ID Thomson**: L-7445-2018, **iralis ID:** MXINF2704

ID 2do Coautor: *Salvador, Martinez-Pagola* / **ORC ID:** 0000-0003-4937-0996, **Researcher ID Thomson**: 6239-2018, **arXiv:** smpagola2000·1

ID 3er Coautor: *Carlos, Maggi-Natale* / **ORC ID:** 0000-0002-3042-4109, **Researcher ID Thomson**: P-4747-2018, **NCBI**: zgatomoteado

ID 4to Coautor: *Lizandro, Jiménez-García* / **ORC ID:** 0000-0001-8932-747X, **Researcher ID Thomson**: O-8745-2018, **arXiv:** lizandro030795

Recibido: 03 de Octubre, 2018; Aceptado 07 de Diciembre, 2018

Resumen

En algunos sistemas informáticos es necesario calcular la edad o el tiempo transcurrido entre dos fechas para llevar a cabo otros cálculos o emisión de reportes. Como se sabe este dato es un campo calculado ya que es dinámico con relación a la fecha. Actualmente no existe una función de librería en MySql para calcular este valor de manera directa, por esta razón este artículo muestra el desarrollo y la implementación de dicha función. Para ello se debe primero investigar las funciones de fecha incluidas en MySql y determinar si alguna de ellas cumple con nuestra necesidad para hallar la diferencia en años entre dos fechas, posteriormente crear al menos cuatro fórmulas, para después llevar a cabo varias pruebas en tablas con uno, dos y tres millones de registros para determinar cual de ellas es más eficiente con relación al tiempo empleado para el calculo, también es importante mencionar que los valores de fecha deben ser validados previamente.

Cálculo edad, Funciones de fecha, Librería Mysql

Abstract

In some computer systems it is necessary to calculate the age or time elapsed between two dates to carry out other calculations or issuance of reports. As you know this data is a calculated field because it is dynamic in relation to the date. Currently there is no library function in MYSQL to calculate this value directly, for this reason this article shows the development and implementation of this function. To do this, first investigate the date functions included in MYSQL and determine if any of them meet our need to find the difference in years between two dates, then create at least four formulas, and then carry out several Testing in tables with one, two, and three million records to determine which of them is more efficient in relation to the time spent calculating, it is also important to mention that the date values must be pre-validated.

Calcúlate age, Date functions, MySQL Libraries

Citación: ARRIETA-ZUÑIGA, Juan, LEON-OLIVARES, Eric, MARTINEZ-PAGOLA, Salvador, MAGGI-NATALE, Carlos y JIMENEZ-GARCÍA, Lizandro. Implementación de una función en Mysql para determinar la edad. Revista de Tecnología Informática. 2018. 2-7: 16-22

* Correspondencia al autor (correo electrónico: alejandroarrieta2010@hotmail.com)
† Investigador contribuyendo como primer autor

Introducción

Actualmente las organizaciones dependen en gran medida del manejo de su información, ya que esto les da ventaja competitiva o ayuda en la toma de decisiones.

Para llevar a cabo dicha administración se utilizan sistemas informáticos los cuales usan bases de datos relacionales o no relacionales para el almacenamiento y proceso de la información. Existen varios manejadores de bases de datos relacionales comerciales y libres, para este articulo usaremos Mysql, el cual es un software libre y popular para el desarrollo sistemas.

En dichos sistemas de información es muy común trabajar con fechas (de nacimiento, caducidad, vencimientos, entre otros) y nos provee de fórmulas para el manejo de datos de tipo fecha, por ejemplo obtener año, mes y día, diferencia en días de dos fechas, sumar días a una fecha entre otras.

Pero una función que nos devuelva la edad de una persona en base a su fecha de nacimiento no existe. Además tenemos el problema que la edad es un campo calculado ya que cambia día a día, razón por la cual no se almacena.

Objetivo

Implementar una función en Mysql para el cálculo de la edad en años de una persona en base a su fecha de nacimiento.

Tipos de datos para almacenar fechas

MySQL posee determinados tipos de datos, los cuales te permiten almacenar fechas en la base de datos [2, 3, 4], ya sea guardando únicamente la parte de la fecha (año, mes y día), o también la parte de la hora correspondiente.

A continuación se describen las funciones a las cuales se hizo referencia anteriormente, se excluyen aquellas donde la parte del día, mes y años no se almacenan.

- *DATE*: Tipo de dato en MySQL que se usa para valores únicamente con la parte del año, mes y día de la fecha. El formato utilizado para almacenar los valores es YYYY-MM-DD [4].

- *DATETIME*: A diferencia del tipo anterior, este permite guardar la fecha completa, teniendo la posibilidad de almacenar además la hora con hasta 6 dígitos correspondientes a la fración exacta de segundo, mediante el uso de un punto decimal como separador entre los segundos y la parte fraccionaria. El formato que se usa es YY-MM-DD HH:MM:SS[.fracción] [4].

- *TIMESTAMP*: El funcionamiento de este es similar al de DATETIME, sin embargo, en este caso las fechas son convertidas y almacenadas en la base de datos utilizando el formato UTC (en el formato YYYY-MM-DD HH:MM:SS), y convertidas al formato específico de la zona actual del servidor al recuperarlas. [4].

Cuando MySQL no puede representar una fecha ingresada, esta es convertida a un valor “cero” (0000-00-00 00:00:00) [3, 4].

Tipo de dato	Rango
DATE	‘1000-01-01 a 9999-12-31’
DATETIME	‘1000-01-01 00:00:00.000000’ a ‘9999-12-31 23:59:59.999999’
TIMESTAMP	‘1970-01-01 00:00:01.000000’ a ‘2038-01-19 03:14:07.999999’

Tabla 1 Rango de valores para almacenar fechas en MySQL

Funciones de fecha y hora en MySQL

MySQL también proporciona un conjunto de funciones a través de las cuales es posible manipular valores temporales, como lo son de tipo fecha [8].

A continuación se describen algunas de dichas funciones, las cuales permiten desde obtener la fecha actual, convertir esta a algún formato específico válido, realizar la comparación entre ambas fechas y posteriormente, recuperar dicho resultado.

CURDATE y *CURRENT_DATE*: Funciones que retornan la fecha actual como un valor en formato ‘YYYY-MM-DD’ o YYYYMMDD, dependiendo de si la función es usada en un contexto numérico o como cadena. [4].

ISSN: 2531-2197

CURRENT_TIMESTAMP,*LOCALTIME*, NOW Y *LOCALTIMESTAMP*: Regresan como resultado una constante de tipo *TIME*, la cual indica la fecha y hora exacta en la cual se empezó a ejecutar la instrucción, o el momento en que se mandó llamar al procedimiento almacenado, función o trigger que contiene dicha instrucción. Tambien es posible obtener los microsegundos, si se ingresa un parámetro de 1 a 6 entre los paréntesis de la función. [8].

SYSDATE: Retorna la fecha y hora en que la instrucción se ejecutó.

Usa el formato 'YYYY-MM-DD HH:MM:SS' en un contexto de cadena y el formato YYYYMMDDHHMMSS en un formato numérico. En un procedimiento almacenado, trigger, función, esta devuelve el tiempo en que se ejecutó cada una de las llamadas a *SYSDATE*. [8].

DATEDIFF: Retorna la diferencia en días entre dos expresiones de tipo DATE o DATETIME, es decir. En el caso de ingresar expresiones DATETIME como parámetro, la parte de la hora será ignorada. [8].

TIMESTAMPDIFF: Retorna la diferencia entre dos expresiones como un entero, en base a la unidad ingresada como parámetro.

En caso de ingresar una o ambas expresiones del tipo *DATE* como parámetro, la parte faltante será cubierta con 00:00:00. [8].

DATE: Extrae la parte de la fecha de una expresión tipo DATE o DATETIME. [8].

DATE_FORMAT: Da formato al valor de fecha de acuerdo a la cadena ingresada como formato.

Existe una gran cantidad de especificadores que pueden ser usados en la cadena de formato. Es necesario anteponer el carácter % antes de un especificador. [8].

Especificador	Descripción
%a	Día de la semana abreviado (Sun, Mon, …, Sat)
%b	Mes abreviado (Jan, Feb, …, Dec)
%c	Mes (0, 1, …, 12)
%d	Día del mes (00, 01, …, 31)
%e	Día del mes (0, 1, …, 31)
%j	Día del año (001, 002, …, 366)
%M	Mes (January, February,, December)
%m	Número del mes (1, 2, …, 12)
%W	Día de la semana (Sunday, Monday, …, Saturday)
%Y	Año numérico de 4 digitos
%y	Año numérico de 2 digitos

Tabla 2 Algunos de los especificadores válidos para la función DATE_FORMAT

DAY y *DAYOFMONTH*: Retornan el día en formato numérico, en un rango de 1 a 31 (ó 0 para fechas que tienen una parte de día "cero"). [8].

DAYOFYEAR: Retorna el día del año para un DATE o DATETIME, en un rango de 1 a 366. [8].

MONTH: Retorna el mes de una fecha específica, en un formato numérico de 1 a 12. [8].

YEAR: Retorna el año de un DATE, en un rango de 1000 a 9999 ó 0 para una fecha "cero". [8].

FROM_DAYS: Convierte un número de días a la fecha correspondiente. [8].

TO_DAYS: Convierte una fecha en formato DATE al correspondiente número de día para dicha fecha. Tomando en cuenta desde el año 0. [8].

Alternativas para obtener la edad en años

Tal como se menciona en la documentación oficial de MySQL, provee una gran cantidad de funciones que se pueden usar para realizar cálculos de edades. [1]

Dicho lo anterior, a continuación se analizan 6 diferentes funciones propuestas, las cuales permiten llevar a cabo esta tarea.

Función número 1

FLOOR((CURDATE() - FEC_CVE_FECHA) / 10000)

ISSN: 2531-2197

Descripción de la función

La fórmula anterior es meramente intuitiva. El funcionamiento de esta se describe a continuación.

1. Primero, obtenemos la fecha actual mediante la función *CURDATE*.
2. Calculamos la diferencia entre ambas fechas, la actual (obtenida en el paso 1) y la de nacimiento FEC_CVE_FECHA. Como el contexto de la consulta es numérico, el resultado de la función será en el formato YYYYMMDD, esto como valor numérico.
3. Dividimos el resultado del paso anterior entre 10000, lo cual pasa la parte del mes y el día después del punto decimal.
4. Como resultado del paso anterior, tenemos la edad como un valor entero, con la parte fraccionaria de la misma como valor decimal. Mediante la función *FLOOR[6]* obtenemos la parte de los años, que es el resultado de la consulta.

Función número 2

FLOOR(DATEDIFF(CURDATE(), FEC_CVE_FECHA) / 365.2)

Descripción de la función

La consulta anterior es similar a la primera. El procedimiento se describe a continuación.

1. Obtenemos la fecha actual mediante la función nativa *CURDATE*.
2. Calculamos el número de días que hay entre ambas fechas, mediante la función *DATEDIFF*.
3. Dividimos el resultado obtenido en el paso anterior entre la cantidad de días que hay en 1 año (para mayor exactitud, establecemos 365.2 como parámetro).
4. Como resultado del paso anterior, tenemos la edad, tomando en cuenta una porción decimal de la misma. Por lo que, mediante la función *FLOOR[6]*, obtenemos la parte entera de la misma.
5. Como resultado de la consulta, tenemos entonces, la edad exacta como un valor entero.

Función número 3

YEAR(FROM_DAYS(TO_DAYS(CURDATE()) - TO_DAYS(FEC_CVE_FECHA)))

Descripción de la función

a) Obtenemos la fecha actual, mediante la función *CURDATE()*.
b) Convertimos ambas fechas, la actual y la de nacimiento a días, esto usando la función nativa de MySQL*TO_DAYS*.
c) Calculamos la diferencia en días que hay entre ambas fechas anteriormente mencionadas.
d) Volvemos a convertir, ahora a un valor tipo fecha, el resultado anteriormente obtenido, esto mediante la función *FROM_DAYS*.
e) Extraemos la parte del año de la fecha obtenida.

Función número 4

YEAR(CURDATE()) - YEAR(FEC_CVE_FECHA) - (DATE_FORMAT(CURDATE(),'%m%d') < DATE_FORMAT(FEC_CVE_FECHA, '%m%d'))

Descripción de la función

a) En la fórmula anterior, primero extraemos la parte del año de la fecha actual y de nacimiento, esto mediante la función *YEAR*.
b) Obtenemos la diferencia entre ambos años anteriormente obtenidos, este valor aún no es la edad deseada, pues no toma en cuenta, los meses y días de ambas fechas.
c) Mediante la función *DATE_FORMAT* extraemos la parte del mes y el día de ambas fechas.
d) Comparamos si el mes y el día de la fecha actual es menor que los valores de la fecha de nacimiento. Esto es igual a comparar, si ya se alcanzó el día de la fecha de nacimiento, o si está aún por cumplirse. Puesto que el contexto de la fórmula es numérico, el resultado será 0 ó 1 respectivamente de si es falsa o verdadera la condición.
e) Del valor obtenido en el paso b, restamos el valor obtenido en la condición anterior, lo cual es igual, ahora si, a la edad deseada.

ISSN: 2531-2197

Función 5

YEAR(CURDATE()) - YEAR(FEC_CVE_FECHA) - (RIGHT(CURDATE(),5)<RIGHT(FEC_CVE_FECHA, 5))

Descripción de la función

La fórmula anterior es similar a la número 4, la única diferencia entre ambas es al extraer el mes y el día de ambas fechas. En este caso se utiliza una función llamada *RIGHT* [7] para tratado de cadenas, la cual retorna la subcadena de longitud m, comenzando en la posición n de la cadena original. Esto tomando en cuenta dos parámetros que recibe la función (n y m) respectivamente.

Función número 6

TIMESTAMPDIFF(YEAR, FEC_CVE_FECHA, CURDATE()) [1]

Descripción de la función

La presente consulta, además de ser la más directa, está propuesta en la documentación oficial de MySQL en las últimas versiones. [1]

El resultado es, efectivamente, la edad exacta entre ambas fechas.

Generación de una base de datos con 1,000,000 de registros

Para poder evaluar el desempeño de cada una de las fórmulas anteriormente mencionadas, es necesaria la creación de una base de datos con al menos 1,000,000 de registros.

Dicho esto, se utilizó la herramienta para bases de datos MySQLWorkbench, junto con el paquete de software XAMPP, el cual contiene Apache, como distribución de servidor web, además del interprete PHP, como lenguaje de programación, entre otros.

```
-- INSTRUCCIÓN PARA LA CREACION DE LA BASE DE DATOS
CREATE DATABASE CALCULO_EDAD;

-- SELECCIONAMOS LA BASE DE DATOS A UTILIZAR
USE CALCULO_EDAD;

-- TABLA DONDE SE ALMACENARÁ EL MILLON DE REGISTROS
CREATE TABLE FECHA(
    FEC_CVE_FECHA INT PRIMARY KEY,
    FEC_FECHA DATE NOT NULL
);
```

Figura 1 Script para la creación de la base de datos

Primero, y utilizando la herramienta MySQLWorkbench, diseñamos la estructura de la base de datos.

Creamos entonces una tabla, la cual contiene un id, y una fecha aleatoriamente generada.

```
<?php
    require_once("conexion.php");
    $conexion = new Conexion();
    $link = $conexion -> conexionBd();
    echo mysqli_error($link);

    for ($i=0; $i<100; $i++) {
        for ($j=0; $j<10000; $j++) {
            $year   = rand(1900, 2018);
            $month  = rand(1, 12);
            $day    = rand(1, 28);
            $date   = $year."-".$month."-".$day;

            $query  = mysqli_query($link, "INSERT FECHA(FEC_CVE_FECHA)
                VALUES('$date')");
        }

        set_time_limit(10);
    }
?>
```

Figura 2 Código para la inserción del millón de registros en MySQ

En el código de programación, escrito en el lenguaje de programación PHP se observa cómo, mediante un ciclo el cual itera un millón de veces, y una variable llamada *$date*, (la cual contiene la fecha aleatoriamente generada, obtenida después de concatenar el resultado de las variables *$year*, *$month* y *$day*, que contienen el año, mes y día respectivamente) se ejecuta la consulta "*INSERT FECHA (FEC_CVE_FECHA) VALUES('$date')*".

Para asegurarnos que todas las fechas sean válidas, tomamos en cuenta como número máximo de día el 28, el cual todos los meses tienen.

Como resultado, se tiene una base de datos llamada CALCULO_EDAD, que contiene una tabla FECHA, con 1 millón de tuplas {id, fecha} en ella.

Resultados

Para la evaluación de las 6 funciones descritas anteriormente se utilizaron 3 equipos con diferentes características de hardware. Sin embargo, todas las funciones utilizadas, están basadas en la documentación oficial de MySQL para la versión 8.0 y probadas en equipos que tenían instalada la versión 5.7.23. Dicho lo anterior, se asegura el funcionamiento correcto de todo lo que se describe en el presente para ambas versiones.

Para cada uno de los equipos, la prueba consistió en ejecutar cada una de las 6 consultas (por separado) en 3 conjuntos de datos de 1, 2 y 3 millones de registros respectivamente.

ISSN: 2531-2197

Características primer equipo

– Memoria RAM 8 GB
– Disco Duro Sólido 120 GB
– Procesador Intel Core i3-5005U CPU 2.00 Ghz x 4
– Sistema Operativo Ubuntu 16.04 x64
– MySQL-server versión 5.7.23

Resultados

Función	Millones de registros		
	1	**2**	**3**
1	0.83	1.54	2.32
2	0.82	1.64	2.39
3	0.52	1.06	1.54
4	0.81	1.67	2.38
5	0.66	1.20	1.74
6	0.55	0.98	1.48

Tabla 3 Tiempo de ejecución (en segundos) para el primer equipo

En este caso, se observa claramente como, la función número 6 (*en amarillo*) obtiene los resultados más rápidos para 2 y 3 millones, respectivamente. La segunda consulta más rápida es la número 3 (*en azul*), con resultados muy próximos a la función ya mencionada para 2 y 3 millones de registros y además, respuesta más rápida para un millón de datos.

Características segundo equipo

– Memoria RAM 4 GB
– Disco Duro Rígido 1 TB
– Procesador Intel Celeron n3060 CPU 1.60 Ghz
– SistemaOperativoWindows10 Home Edition x64
– MySQL-server versión 8.0

Resultados

Función	Millones de registros		
	1	**2**	**3**
1	3.21	6.13	10.77
2	4.24	9.11	9.89
3	1.95	4.55	6.52
4	5.17	8.64	12.25
5	4.88	6.91	15.27
6	4.33	7.98	9.46

Tabla 4 Tiempo de ejecución (en segundos) para el segundo equipo

En este caso, como se puede ver en la tabla, la función más rápida fue la número 3 (*en amarillo*), la cual tuvó los mejores tiempos para 1, 2 y 3 millones de registros, superando por mucho a las demás funciones.

En este caso la segunda consulta más rápida fue la número 1 (*en azul*),

Características tercer equipo

– Memoria RAM 4 GB
– Disco Duro Sólido 120 GB
– Procesador Intel Celeron n2840 CPU 2.16 Ghz * 2
– Sistema Operativo Ubuntu 18.04 x64
– MySQL-server versión 5.7.23

Resultados

Función	Millones de registros		
	1	**2**	**3**
1	1.78	3.60	5.23
2	1.86	3.84	5.41
3	1.16	2.42	3.41
4	1.95	3.98	5.75
5	1.38	2.80	3.97
6	1.03	2.09	3.06

Tabla 5 Tiempo de ejecución (en segundos) para el tercer equipo

Para este equipo, pudo observarse como, al igual que en el primero de ellos, la función número 6 (*en amarillo*) fue la que tardó menos tiempo en obtener el conjunto de datos resultado.

En cuanto al segundo lugar, otra vez se encuentra la función 3 (*en azul*), cuyos tiempos de ejecución fueron menores que en las demás consultas, superados únicamente, por la función número 6.

Conclusiones

Después de analizar el desempeño de cada una de las diferentes consultas propuestas se pueden observar diferentes resultados entre el primer lugar, teniendo 2 diferentes funciones como primer lugar en las pruebas.

Dicho lo anterior, y para poder determinar cuál es mas rápida entre ellas, se decidió comparar ahora el tiempo de ejecución únicamente de estas dos funciones, en base a los resultados obtenidos para cada conjunto de datos y para cada equipo correspondientemente.

ISSN: 2531-2197

Esto ante la pregunta ¿Qué tanto fue más rápida la primera de la segunda? Y viceversa.

Función	Millones de registros		
	1	**2**	**3**
3	1.00	1.08	1.04
6	1.06	1.00	1.00

Tabla 6 Comparación de que función fue más rápida, para el equipo de prueba número 1

Función	Millones de registros		
	1	**2**	**3**
3	1.00	1.00	1.00
6	2.22	1.75	1.45

Tabla 7 Comparación de que función fue más rápida, para el equipo de prueba número 2

Función	Millones de registros		
	1	**2**	**3**
3	1.13	1.16	1.11
6	1.00	1.00	1.00

Tabla 8 Comparación de que función fue más rápida, para el equipo de prueba número 3

Aún y pese a que la función número 6 fue más rápida que la número 3 en la mayoría de las pruebas realizadas, en el caso de prueba del equipo número 2, los valores observados tuvieron la diferencia más grande observada, a diferencia de la consulta número 3, cuyos resultados siempre fueron casi iguales a los de la función más rápida para cada conjunto de datos.

Dicho lo anterior, se concluye que la mejor función (en base al comportamiento de los tiempos) para calcular la fecha es la número 3.

YEAR(FROM_DAYS(TO_DAYS(CURDATE()) - TO_DAYS(FEC_CVE_FECHA))).

Recomendaciones

Como se ha mencionado la edad es un campo calculado (no forma parte de la base de datos) y en varios sistemas informáticos se utiliza con frecuencia, como se observa dos funciones (3 y 6) fueron las que obtuvieron mejores resultados, por lo que se recomienda el uso de cualquiera de ellas, ya que normalmente las tablas solo tienen miles de registros y ahí seguramente no se notará el rendimiento por usar cualquiera de ellas.

Referencias

[1] MySQL. (2018). Date Calculations. 2018-08-17, de MySQL Sitio web: https://dev.mysql.com/doc/refman/8.0/en/date-calculations.html

[2] MySQL. (2018). Date and Time Type Overview. 2018-08-17, de MySQL Sitio web: https://dev.mysql.com/doc/refman/8.0/en/date-and-time-type-overview.html

[3] MySQL. (2018). Date and Time Types.2018-08-17, de MySQL Sitio web: https://dev.mysql.com/doc/refman/8.0/en/date-and-time-types.html

[4] MySQL. (2018). The DATE, DATETIME, and TIMESTAMP Types. 2018-08-17, de MySQL Sitio web: https://dev.mysql.com/doc/refman/8.0/en/datetime.html

[5] MySQL. (2018). Date and Time Functions.2018-08-17, de MySQL Sitio web: https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html

[6] MySQL. (2018). Numeric Functions and Operators. 2018-08-17, de MySQL Sitio web: https://dev.mysql.com/doc/refman/8.0/en/numeric-functions.html

[7] MySQL. (2018). String Functions. 2018-08-17, de MySQL Sitio web: https://dev.mysql.com/doc/refman/8.0/en/string-functions.html

[8] MySQL 5.0 Refernce Manual (2015). Estados Unidos:Oracle.

ISSN: 2531-2197