http://www.sqlservercentral.com/articles/T-SQL/63351/
Printed 2009/03/29 09:06AM

Calculating Age

By Lynn Pettis, 2008/06/05

How do you calculate the age of an individual? This can become an extremely hot topic and the number of solutions provided can be numerous. Before attacking this “controversial” topic, it actually would help set some ground rules to be used in designing an aging algorithm.

The most difficult part of this, of course, is figuring out what to do with leap years. When does a person (or object or document for that matter) become another year older? For the purposes of this short article, I propose that this occurs on February 28th of non leap years.

Some Basic Date Arithmetic, SQL Style

Before diving into the aging calculation, let’s look at some very simple date arithmetic using SQL Server 2005.

First, let’s see what adding 1 year to February 28, 2008 and February 29, 2008 returns.

```
declare @date1 datetime,
        @date2 datetime
set @date1 = '20080228'
set @date2 = '20080229'
select dateadd(yy, 1, @date1), dateadd(yy, 1, @date2)
```

Result:

2009-02-28 00:00:00.000 2009-02-28 00:00:00.000

Interesting, adding 1 year to both dates results in the same date. Let’s see what happens when you now add 4 years.

```
declare @date1 datetime,
        @date2 datetime
set @date1 = '20080228'
set @date2 = '20080229'
select dateadd(yy, 4, @date1), dateadd(yy, 4, @date2)
```

Result:

2012-02-28 00:00:00.000 2012-02-29 00:00:00.000

Now that is interesting, both dates aren’t the same. This is what would be expected.

Now, let’s look at DATEDIFF and see how it works a little.

```
declare @date1 datetime,
        @date2 datetime
set @date1 = '20080229'
set @date2 = '20090228'
select datediff(yy, @date1, @date2)
```

Result:

1

But wait, what is the result of the following?

```
declare @date1 datetime,
        @date2 datetime
set @date1 = '20081231'
set @date2 = '20090101'
select datediff(yy, @date1, @date2)
```

Result:

1

Wait! That isn't right, there isn't a year between those two dates, something must be wrong. As you can see, the DATEDIFF function returned the number of year's difference between the year values of the two dates, not the actual number of years between the two dates.

We can use this information to create a simple algorithm:

Age = datediff(yy, date_of_birth, Today)

– (if Today < dateadd(yy, datediff(yy, date_of_birth, Today),date_of_birth)

then 1 else 0

Or, to put it in to T-SQL:

```
declare @dob datetime,
        @age int,
        @day datetime
set @day = '2008-02-28'
set @dob = '2007-03-01'
set @age = datediff(yy,@dob,@day) -
           case when @day < dateadd(yy,datediff(yy,@dob,@day), @dob) then 1 else 0 end
select @age
```

Conclusion

This is just one method to use to calculate the age of person in code. There are numerous other methods that can also be used. All I can suggest, is use the method that works best for you in your situation.