All about DATETIME

Madhivanan
SQL Server MVP

Professional Association for SQL Server

- Two 4-byte integers
    - First 4-bytes represents number of days from the base date (January 01, 1900)
    - Second 4-bytes represents time which is number of milliseconds after midnight

Internal Storage

```
declare @mydate datetime
set @mydate='2009-12-10 18:32:55:873'
select
    @mydate as source_date,
    datediff(day, '1900-01-01',@mydate) as no_of_days,
    convert(char(15),@mydate,114) as time_part,
    datediff(millisecond, '1900-01-01',convert(char(15),@mydate,114))
    as number_of_milliseconds
```

Result

source_date	no_of_days	time_part	milliseconds
2009-12-10 18:32:55.873	40155	18:32:55:873	66775873

```
select
    dateadd(day,40155, '1900-01-01') as date,
    dateadd(millisecond,66775873,dateadd(day,40155, '1900-01-
    01')) as source_date
```

```
date                                    source_date
-----------------------------------------------------------------------
2009-12-10 00:00:00.000      2009-12-10 18:32:55.873
```

- Datetime
- Minimum value : January 1, 1753
- Maximum value :December 31, 9999

- Smalldatetime
- Minimum value : January 1, 1900
- Maximum value :June 6, 2079

Date Ranges

Rounding

Datetime value is rounded to

0.000,0.003 or 0.007 milliseconds

Example

```
select cast('2010-01-01 12:45:34.755' as datetime)
```

Result

```
-----------------------
2010-01-01 12:45:34.757
```

Datetime

Rounding numbers

Last digit of the millisecond	Rounded value
0 or 1	0
2,3 or 4	3
5,6,7 or 8	7
9	0 (increase previous digit)

Examples

Datetime value	Rounded value
01/01/98 23:59:59.990 or 01/01/98 23:59:59.991	1998-01-01 23:59:59.990
01/01/98 23:59:59.992, 01/01/98 23:59:59.993, or 01/01/98 23:59:59.994	1998-01-01 23:59:59.993
01/01/98 23:59:59.995, 01/01/98 23:59:59.996, 01/01/98 23:59:59.997, or 01/01/98 23:59:59.998	1998-01-01 23:59:59.997
01/01/98 23:59:59.999	1998-01-02 00:00:00.000

Rounding
SmallDatetime value is rounded to 1 minute

Values of 29.998 seconds or less are rounded down to the nearest minute; values of 29.999 seconds or more are rounded up to the nearest minute.

Example

```
select
    cast('2010-01-01 12:45:24.755' as smalldatetime),
    cast('2010-01-01 12:45:34.755' as smalldatetime)
```

Result

```
-----------------------------------------------------------
2010-01-01 12:45:00          2010-01-01 12:46:00
```

SmallDatetime

Millisecond expression

```
select
        cast('2010-01-01 12:45:34.79' as datetime),
        cast('2010-01-01 12:45:34:79' as datetime)
```

Result

```
------------------------------------------------------------------------
2010-01-01 12:45:34.790          2010-01-01 12:45:34.080
```

Date '1900-01-01 00:00:00.000‘ is equal To number 0

SELECT CAST(0 as DATETIME)
Result

1900-01-01 00:00:00.000

SELECT CAST(1 as DATETIME)
Result

1900-01-02 00:00:00.000

Date as number

Usual method

```
declare @date datetime
set @date='2010-01-01  12:45:34.79‘
--Convert  to style MM/dd/yyyy
select convert(varchar(10),@date,101)

Result
----------
12/15/2010

--Convert   back to datetime
Select convert(datetime,convert(varchar(10),@date,101))
Result
------------------------------
2010-12-15 00:00:00.000
```

Remove time part

Efficient method

```
Select dateadd(day,datediff(day,0,@date),0)
```

Result

```
------------------------------
2010-12-15 00:00:00.000
```

Method 1

(String handling)

```
declare @date datetime
set @date='2010-03-12 12:36:45'
select cast(cast(year(@date) as char(4))+'-
    '+right('00'+cast(month(@date) as varchar(2)),2)+'-01' as datetime)
```

Result

```
----------------------
2010-03-01 00:00:00.000
```

Method 2

(Date handling)

```
select dateadd(month,datediff(month,0,@date),0)
```

Result

```
-----------------------
2010-03-01 00:00:00.000
```

Find first day of the year

Method 1
(String handling)

```
declare @date datetime
set @date='2010-03-12 12:36:45'
select cast(cast(year(@date) as char(4))+'-01-01' as datetime)
```

Result

```
----------------------
2010-01-01 00:00:00.000
```

Method 2

(Date handling)

```
select dateadd(year,datediff(year,0,@date),0)
```

Result

```
-----------------------
2010-01-01 00:00:00.000
```

Find data added on yesterday

Assumption : Table has a datetime column with default value getdate()

Wrong method

Select columns from table

Where date_col=getdate()-1

Sample Queries

Correct Methods

Method 1

```
Select columns from table
Where datediff(day,date_col,getdate())=1
```

Method 2

```
Select columns from table
Where
date_col>=dateadd(day,datediff(day,0,getdate()),-1) and
date_col<dateadd(day,datediff(day,0,getdate()),0)
```

Find data added on last month

Wrong method

Select columns from table
Where date_col=dateadd(month,-1,getdate())

Correct Methods

Method 1

```
Select columns from table
Where datediff(month,date_col,getdate())=1
```

Method 2

```
Select columns from table
Where
date_col>=dateadd(month,datediff(month,0,getdate())-1,0) and
date_col<dateadd(month,datediff(month,0,getdate()),0)
```

Unambiguous formats

YYYYMMDD HH:MM:SS

YYYY-MM-DDTHH:MM:SS

Ambiguous formats

DD/MM/YYYY

MM/DD/YYYY

DD-MM-YYYY

MM-DD-YYYY

YYYY-MM-DD

Etc

Date Formats

```
Declare @test table(dates datetime)
insert into @test
select '12/03/2010'

select dates from @test
```

Result

```
dates
-----------------------------
2010-12-03 00:00:00.000
```

```
Declare @test table(dates datetime)
insert into @test
select '27/03/2010'

select dates from @test
```

- Msg 242, Level 16, State 3, Line 3
- The conversion of a char data type to a datetime data type resulted in an out-of-range datetime value.
- The statement has been terminated.

Dbcc useroptions

Set options	Value
language	us_english
dateformat	mdy
datefirst	7

.

.

.

.

Dateformats

mdy → mm/dd/yyyy, mm.dd.yyyy, mm-dd-yyyy

dmy → dd/mm/yyyy, dd.mm.yyyy, dd-mm-yyyy

ymd → yyyy/mm/dd, yyyy.mm.dd, yyyy-mm-dd

ydm → yyyy/dd/mm, yyyy.dd.mm, yyyy-dd-mm

Etc

```
set dateformat dmy
Declare @test table(dates datetime)
insert into @test
select '12/03/2010'
union all
select '27/03/2009'
select dates from @test

dates
-----------------------
2010-03-12 00:00:00.000
2009-03-27 00:00:00.000
```

Date format dd-MMM-yyyy
Reliable only if the default language of the server is English

```
Set dateformat dmy
Declare @test table(dates datetime)
insert into @test
select '12-Mar-2010'
union all
select '27-Mar-2009'
select dates from @test

dates
-----------------------
2010-03-12 00:00:00.000
2009-03-27 00:00:00.000
```

```
set language german
Declare @test table(dates datetime)
insert into @test
select '12-Mar-2010'
union all
select '27-Mar-2009'
select dates from @test
```

```
Msg 241, Level 16, State 1, Line 2
Fehler beim Konvertieren einer Zeichenfolge in einen datetime-Wert.
```

```
set language german
Declare @test table(dates datetime)
insert into @test
select ‘20100312'
union all
select ‘20090327'
select dates from @test

dates
----------------------
2010-03-12 00:00:00.000
2009-03-27 00:00:00.000
```

```
set language german
Declare @test table(dates datetime)
insert into @test
select '2010-03-12'
union all
select '2009-03-27'
select dates from @test
```

Die Spracheneinstellung wurde auf Deutsch geändert.

Msg 242, Level 16, State 3, Line 3

Bei der Konvertierung eines char-Datentyps in einen datetime-Datentyp liegt der datetime-Wert außerhalb des gültigen Bereichs.

Die Anweisung wurde beendet.

```
set language german
Declare @test table(dates datetime)
insert into @test
select '2010-03-12T00:00:00'
union all
select '2009-03-27T00:00:00'
select dates from @test

dates
----------------------
2010-03-12 00:00:00.000
2009-03-27 00:00:00.000
```

Unreliable ISDATE() function

Isdate() is used to check if the date is a valid date but not always reliable

Reasons

It depends on the date settings

It does implicit convertion

```
set dateformat mdy
select
isdate('12/02/2009'),
isdate('28/02/2009'),
isdate('2009-12-12'),
isdate('20091219')

----------- ----------- ----------- -----------
1           0           1           1
```

Examples for isdate()

```
set dateformat dmy
select
isdate('12/02/2009'),
isdate('28/02/2009'),
isdate('2009-12-12'),
isdate('20091219')

----------- ----------- ----------- -----------
1           1           1           1
```

```
set dateformat mdy
select
isdate(2000),
isdate('2000'),
isdate(3000000/456)

----------- ----------- -----------
1           1           1
```

Isdate() does implicit convertion

```
select
cast(2000 as datetime),
cast('2000' as datetime),
cast(3000000/456 as datetime)
---------------------- ---------------------- ----------------------
1905-06-24 00:00:00.000 2000-01-01 00:00:00.000 1918-01-05
    00:00:00.000
```

```
declare @dates table (dates varchar(8))
insert into @dates(dates)
Select '20071201'
union all
Select '2007'
union all
Select 2007
union all
Select '1800'

select dates from @dates where ISDATE(dates) =1
```

```
Dates
----------------
20071201
2007
2007
1800
```

```
select dates from @dates
where ISDATE(dates) =1 and LEN(dates)=8


dates
--------
20071201
```

```
declare @d datetime
set @d='2008-10-12 16:32:18'
select
convert(varchar(10),@d,101),
convert(varchar(10),@d,103),
convert(varchar(30),@d,109),
convert(varchar(10),@d,112)

---------- ---------------------------------- ------------------------------ ---------

10/12/2008 12/10/2008 Oct 12 2008  4:32:18:000PM     20081012
```

Date formations

Don’t format dates using sql

Reasons

- Dates become Varchars and wont allow date related caluculations (dateadd, datediff,etc)
- Wont allow to make use of index (if defined) if formatted at where clause
- Web page, reports, etc treat it as varchars (calculations, Ordering,etc wont work properly)

Do format dates using sql

Cases

- Export results to text file with specific date format
- Import data from other sources where dates are in different formats
- Front end application can’t be changed but it needs specific date formats for display

Thank You

Write your questions @
chnsqlug@googlegroups.com