

先上一张对话截图，本次分享的起因，皆因此对话而起

如果都没加设置成数据库默认生成呢
魏吉贺(魏吉贺) 2013-06-20 04:37:02
如果都没加 设置成数据库默认生成呢
来爱清(来爱清) 2013-06-20 04:37:06
数据库不知道设置默认值了没
杨双(杨双) 2013-06-20 04:37:23
sql会报错的
来爱清(来爱清) 2013-06-20 04:37:17
不过前端调用时已经报错了
杨双(杨双) 2013-06-20 04:37:28
默认001-什么的
来爱清(来爱清) 2013-06-20 04:37:24
溢出|
杨双(杨双) 2013-06-20 04:38:46
年不能小于1750
魏吉贺(魏吉贺) 2013-06-20 04:39:26
1970-1-1
杨双(杨双) 2013-06-20 04:39:45

魏吉贺(魏吉贺) 2013-06-20 04:39:43
默认当期日期吧

可以看下时间，在做慈展会项目的时候，来工等几个写后台方法的时候，前台的日期文本框，如果第一次访问该页面，后台不给前台的日期文本框准确的赋值，默认的初始时间是 0001-1-1，也就是 clr 中 DateTime 的初始值，我记得好像是这个，没事，这不是主因。这个时候，如果不改默认时间，直接提交，那么后台会报错。错误是来工说的，datetime 数值溢出问题

这个问题我相信很多人都碰到过，但是，原因或许并不明了，那么本次我将着重对 datetime 数值溢出问题做一下实验，以便帮大家弄清楚溢出临界值，为何会出现溢出

说到这，我想起我在做新人第四次作业的时候，用 excel 专家组件，组件曾经也从内部抛出过此异常，新人在做作业的时候，如果碰到溢出问题，可以先移步 <http://192.168.10.90:81/canyouwiki/index.php?doc-view-2461>，这个是 excel 专家组件抛 时间值溢出的说明，有需要的时候，再看本文。

好了，废话不多说，开始我们的实验吧

简便起见，我建立一个控制台程序，代码如下

```

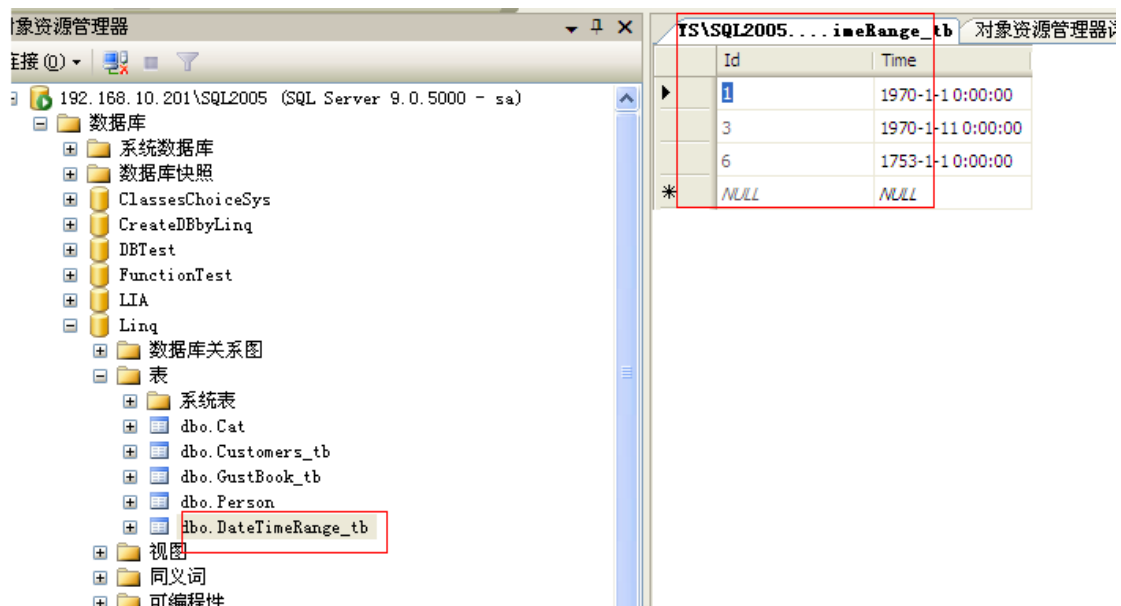
class Program
{
    public static readonly string connStr = @"Server=192.168.10.201\sql2005;Database=Linq;User Id=sa;";
    static void Main(string[] args)
    {
        string datetime = string.Empty;
        DateTime temptime = default(DateTime);

        while (!string.IsNullOrEmpty(datetime=Console.ReadLine()))
        {
            try
            {
                temptime = Convert.ToDateTime(datetime);
            }
            catch (Exception e)
            {
                Console.WriteLine(e);
                continue;
            }
            using (SqlConnection connection = new SqlConnection(connStr))
            {
                connection.Open();
                SqlCommand command = connection.CreateCommand();
                command.CommandText = "insert into DateTimeRange_tb(Time) values('" + temptime + "')";
                try
                {
                    int result = command.ExecuteNonQuery();
                    if (result > 0)
                    {
                        Console.WriteLine("日期插入成功");
                    }
                    else
                    {
                        Console.WriteLine("日期插入失败");
                    }
                }
                catch (SqlException e)
                {
                    Console.WriteLine(e);
                    Console.WriteLine();
                }
            }
        }
    }
}

```

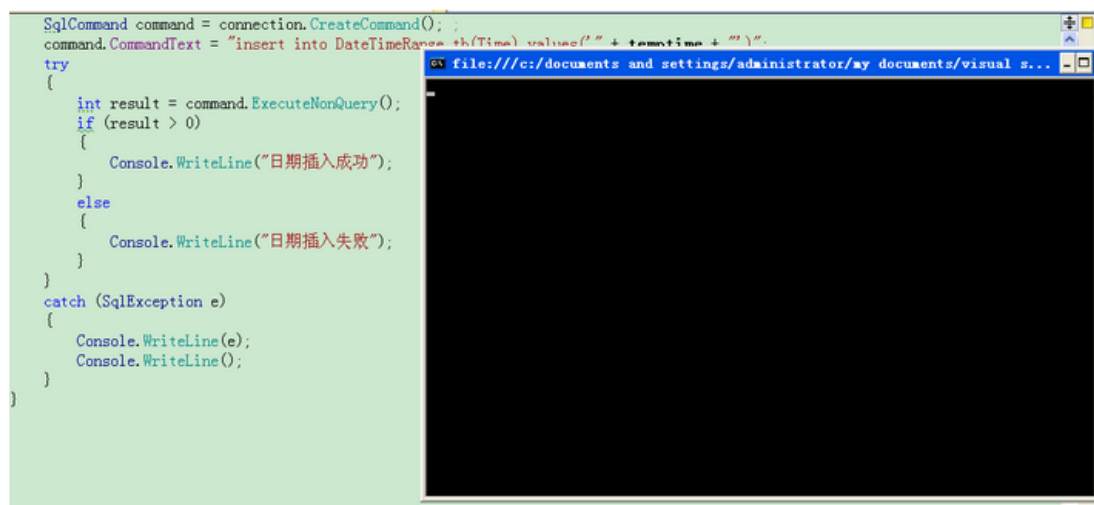
代码很简单，为了方便我测试，我不想每次都来启用控制台程序，所以将测试代码写成这个样子，我从控制台读取输入的字符串，将其转换为 clr 中定义的 datetime 格式，然后通过 ado.net 写进我的数据库表

大概就是这样子，我的表是这个样子的

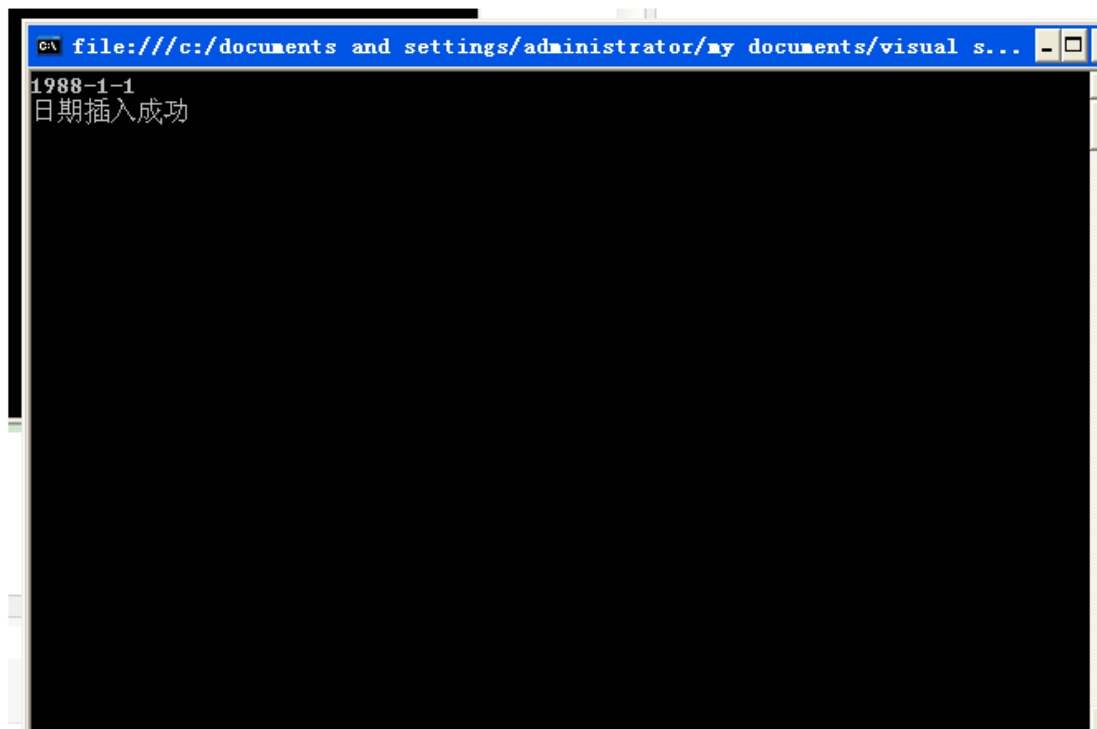


ok, 基本的就是这个样子

我们来测试吧, 跑其控制台程序, 如下



控制台在等待输入, 好输入数据



好，看到提示了，提示插入成功，不用怀疑，肯定插入进表了，真的如此么，看一下

YS\SQL2005....imeRange_tb 对象资源管理器详细信息		
	Id	Time
▶	1	1970-1-1 0:00:00
	3	1970-1-11 0:00:00
	6	1753-1-1 0:00:00
	7	1976-1-1 0:00:00
	8	1965-1-1 0:00:00
	9	1988-1-1 0:00:00
	10	1988-1-1 0:00:00
	11	1988-1-4 0:00:00
	12	1754-1-1 0:00:00
	13	1970-1-1 0:00:00
	14	1753-1-1 0:00:00
	18	1900-1-1 0:00:00
	19	1753-1-1 0:00:00
	21	1988-1-1 0:00:00
*	NULL	NULL

红色的是我们刚才插入的那条记录

接下来我来插入一条数据，那就是魏工在第一张对话图中提到的临界值，

1970-1-1，结果如何呢，见图

```
C:\ file:///c:/documents and settings/admini
1988-1-1
日期插入成功
1970-1-1
日期插入成功
-
```

我们看到 1970-1-1 顺利插入，魏工提到这个是临界值，那么是不是意味着，小于这个值，会报 datetime 值溢出异常呢？

我们来测试一下

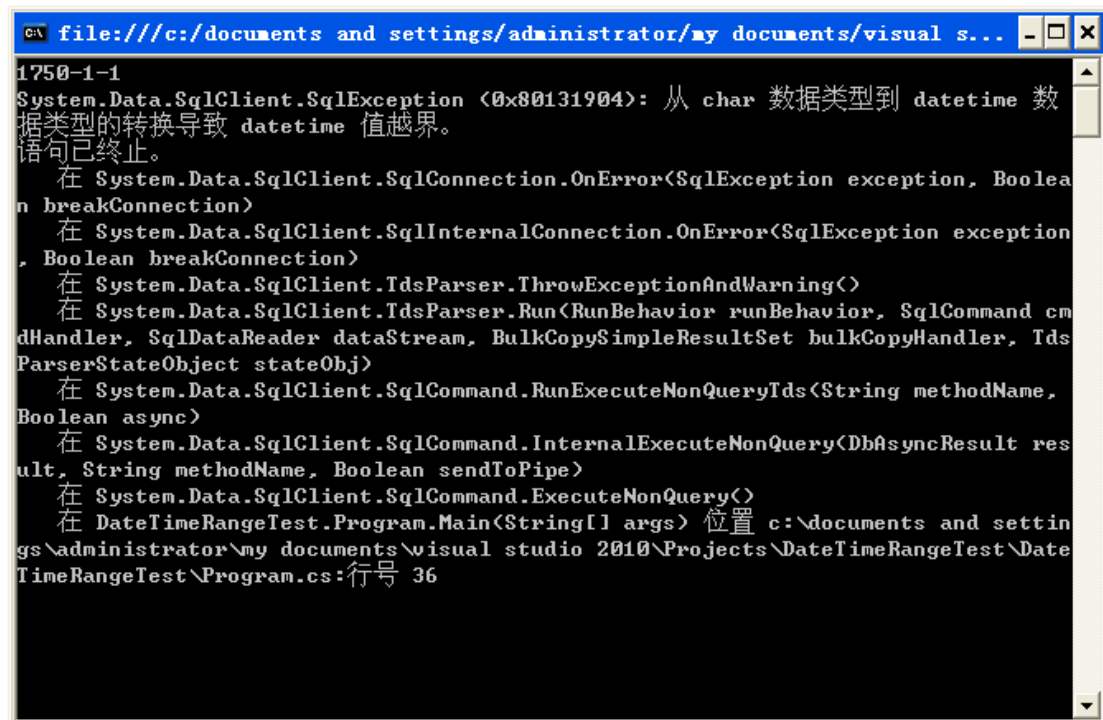
The screenshot shows the SQL Server Enterprise Manager interface. On the left, a table named 'imeRange_tb' is displayed with columns 'Id' and 'Time'. The 'Time' column contains various datetime values. On the right, a command window shows the results of inserting data. The values '1988-1-1', '1970-1-1', and '1950-1-1' are shown, each followed by '日期插入成功' (Date inserted successfully). The value '1950-1-1' is highlighted with a red box.

Id	Time
1	1970-1-1 0:00:00
3	1970-1-11 0:00:00
6	1753-1-1 0:00:00
7	1976-1-1 0:00:00
8	1965-1-1 0:00:00
9	1988-1-1 0:00:00
10	1988-1-1 0:00:00
11	1988-1-4 0:00:00
12	1754-1-1 0:00:00
13	1970-1-1 0:00:00
14	1753-1-1 0:00:00
18	1900-1-1 0:00:00
19	1753-1-1 0:00:00
21	1988-1-1 0:00:00
22	1970-1-1 0:00:00

从图上我们可以很清晰的看见我们插入的数据 1950-1-1，这个条记录成功插入，

所以溢出临界值肯定不是 1970-1-1

那么溢出临界值是多少呢，我记得好像是 1750-1-1，试一下
我输入 1750-1-1



```
file:///c:/documents and settings/administrator/my documents/visual s...
1750-1-1
System.Data.SqlClient.SqlException (0x80131904): 从 char 数据类型到 datetime 数
据类型的转换导致 datetime 值越界。
语句已终止。
   在 System.Data.SqlClient.SqlConnection.OnError<SqlException exception, Boolean
breakConnection>
   在 System.Data.SqlClient.SqlInternalConnection.OnError<SqlException exception
, Boolean breakConnection>
   在 System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning<>
   在 System.Data.SqlClient.TdsParser.Run<RunBehavior runBehavior, SqlCommand cm
dHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, Tds
ParserStateObject stateObj>
   在 System.Data.SqlClient.SqlCommand.RunExecuteNonQueryTds<String methodName,
Boolean async>
   在 System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery<DbAsyncResult res
ult, String methodName, Boolean sendToPipe>
   在 System.Data.SqlClient.SqlCommand.ExecuteNonQuery<>
   在 DateTimeRangeTest.Program.Main(String[] args) 位置 c:/documents and settin
gs/administrator/my documents/visual studio 2010/Projects/DateTimeRangeTest/Date
TimeRangeTest/Program.cs:行号 36
```

已经抛出 datetime 值溢出了，所以，临界值也不是我说的 1750-1-1，那么临界
值是多少，我决定百度，找点资料看下

查到的资料如下

MS SQL Server 2005 有 datetime 和 smalldatetime 两种格式的日期时间数据类
型。注意：没有单独的日期或时间类型。 datetime
数据库内部用两个 4 字节的整数存储 datetime 数据类型的值。第一个 4 字节存
储基础日期（即 1900-1-1, base date）之前或之后的日期。日期范围为 1753-1-1
至 9999-12-31。当日期为 1900-1-1 时值为 0；之前的日期是负数；之后日期是
正 数。另外一个 4 字节存储以午夜后三又三分之一（10/3）毫秒数所代表的每
天的时间。精确度为百分之三秒（等于 3.33 毫秒或 0.00333 秒）。如下表 所示，
把值调整到.000、.003、或.007 秒的增量

这个是 sql server 内部对 datetime 数值范围的定义和处理。

再明白不过了，真是如此？ just do it

```

在 System.Data.SqlClient.SqlInternalConnection.OnError
, Boolean breakConnection)
在 System.Data.SqlClient.TdsParser.ThrowExceptionAndW
在 System.Data.SqlClient.TdsParser.Run<RunBehavior ru
dHandler, SqlDataReader dataStream, BulkCopySimpleResult
ParserStateObject stateObj>
在 System.Data.SqlClient.SqlCommand.RunExecuteNonQue
Boolean async)
在 System.Data.SqlClient.SqlCommand.InternalExecuteNo
ult, String methodName, Boolean sendToPipe>
在 System.Data.SqlClient.SqlCommand.ExecuteNonQuery()
在 DateTimeRangeTest.Program.Main<String[] args> 位置
gs\administrator\my documents\visual studio 2010\Project
TimeRangeTest\Program.cs:行号 36

```

1753-1-1
日期插入成功

1753-1-1 这调数据插入成功

再插入一条数据，注意睁大你的钛合金眼睛，见证奇迹的时刻

```

1752-12-31 23:59:59
System.Data.SqlClient.SqlException (0x80131904): 从 char 数据类型到 datetime 数
据类型的转换导致 datetime 值越界。
语句已终止。
在 System.Data.SqlClient.SqlConnection.OnError<SqlException exception, Boolean
n breakConnection>
在 System.Data.SqlClient.SqlInternalConnection.OnError<SqlException exception
, Boolean breakConnection>
在 System.Data.SqlClient.TdsParser.ThrowExceptionAndWarning<>
在 System.Data.SqlClient.TdsParser.Run<RunBehavior runBehavior, SqlCommand cm
dHandler, SqlDataReader dataStream, BulkCopySimpleResultSet bulkCopyHandler, Tds
ParserStateObject stateObj>
在 System.Data.SqlClient.SqlCommand.RunExecuteNonQueryTds<String methodName,
Boolean async>
在 System.Data.SqlClient.SqlCommand.InternalExecuteNonQuery<DbAsyncResult res
ult, String methodName, Boolean sendToPipe>
在 System.Data.SqlClient.SqlCommand.ExecuteNonQuery<>
在 DateTimeRangeTest.Program.Main<String[] args> 位置 c:\documents and settin
gs\administrator\my documents\visual studio 2010\Projects\DateTimeRangeTest\Date
TimeRangeTest\Program.cs:行号 36

```

我们插入 1752-12-31 23:59:59。让时间数据只少一秒钟，我们可以看到，哪怕只少一秒的时间，数据已经发生了溢出。

经过以上的测试实验，我们可以验证，sql 中 datetime 数据临界值是 1753-1-1，不是 1970-1-1。只要数值比那个临界值小，那么你在像 sql server 中插入数据的时候，就一定会发生 datetime 值溢出的问题。

好了，不知道大家有没有疑问，sql 中的 datetime 有临界值，那么 clr 中的临界值是多少呢

很简单，上 msdn 查下，结果如下



The screenshot shows the MSDN documentation for the `DateTime` type. The first line states: "DateTime 值类型表示值范围在公元（基普纪元）0001 年 1 月 1 日午夜 12:00:00 到公元（C.E.）9999 年 12 月 31 日晚上 11:59:59 之间的日期和时间". This line is highlighted with a red box. Below this, it says: "时间值以 100 毫秒为单元（该单元称为刻度）进行计量，而特定日期是自 GregorianCalendar 日历中公元（C.E.）0001 年 1 月 1 日午夜 12:00 以来的刻度数。例如，刻度值 312413760000000000L 表示 0100 年 1 月 1 日（星期五）午夜 12:00:00。DateTime 值始终在显式或默认日历的上下文中表示。". Below this, under "版本注意事项", it says: "在 .NET Framework 2.0 版以前，DateTime 结构包含一个 64 位字段，该字段由一个未使用的 2 位字段和一个私有字段 Ticks 串联组成，Ticks 字段是一个 62 位无符号字段，其中包含表示日期和时间的刻度数。Ticks 字段的值可通过 Ticks 属性获取。". Finally, it says: "从 .NET Framework 2.0 开始，DateTime 结构包含一个由私有字段 Kind 和 Ticks 字段串联组成的 64 位字段。Kind 字段是一个 2 位字段，它指示 DateTime 结构是表示本地时间、协调通用时间 (UTC) 还是 UTC 和本地时间都未指定。Kind 字段用于处理本地时间和 UTC 时间之间的转换，但不用于时间的比较或算术运算。Kind 字段的值可通过 Kind 属性获取。".

clr 中定义的 datetime 范围，我已经用红的圈起来了。

好了，相信本文可以让大家对 ms sql server 中 datetime 数值溢出异常有一个很充分的认识 and 了解，以后在开发过程中会注意这个问题。

全文写到这里，我想说明的一点是，老邓的一句说是极其有道理的，实践是检验真理的唯一标准。

任何主观的猜想和判断，只有经过实践的检验，才能靠得住，否则就是不可靠的。

不知道魏工看到我这篇分享，作何感想，如果大家对我的实验有异议，欢迎私底下下来找我讨论，非常欢迎，:)))

周末愉快~

魏：双遇到问题，在质疑中能够坚持去追根，挺不错的。

关于 1970-1-1 记得有那么一回事，后来找了一下具体来历，这个来源于 mysql （张冠李戴了 :))

<http://dev.mysql.com/doc/refman/5.0/en/date-and-time-type-overview.html>。

综合来看，不同数据库对不同的日期类型的可容许范围处理的确还是不一样的。

