

نمی‌دونم تابحال به صورت جدی با SharePoint مایکروسافت کار کردید یا نه؟ این برنامه که عمدۀ کارهای خودش رو با SQL server انجام میده در طول یک روز ممکن است تا 80 گیگ log file اس‌کیوال سرور تولید کند و بعد از چند روز اگر به همین صورت به حال خود رها شود (که عوموما هم به همین صورت است!) ممکن است دیگر قادر به استفاده از سرور به دلیل پر شدن درایوی که لاغ فایل‌ها در آن ذخیره می‌شوند نباشد.

همچنین رشد tempdb نیز توسط این برنامه بسیار چشم‌گیر است. بنابراین همیشه به‌خاطر داشته باشید محل قرارگیری db و همچنین محل قرارگیری لاغ فایل‌ها (که هر دو قابل تنظیم هستند) را در درایوهایی قرار دهید که حداقل 100 گیگ فضای خالی در آنها موجود باشد.

با استفاده از اسکریپت زیر می‌شود حجم لاغ فایل‌های اس‌کیوال سرور را به حداقل رساند و نفس راحتی کشید! این مساله اگر جدی گرفته نشود واقعاً تبدیل به یک کابوس می‌شود!

اسکریپت زیر کلیه دیتابیس‌های موجود را یافته و shrink می‌کند. قسمت online و offline می‌کند. کردن آن هم به این خاطر است که ارتباط تمام کاربران متصل را به صورت آنی قطع می‌کند (یکی از چندین روش موجود برای kill کردن کاربران است). (یک stored procedure از آن درست کنید و با تعریف یک job جدید در اس‌کیوال سرور، این stored procedure را برای مثال هر روز ساعت 3 بامداد به صورت خودکار اجرا کنید)

```

Declare @database nvarchar(1000)
Declare @tsql nvarchar(4000)
Declare DatabaseCursor Cursor
Local
Static
For
select name from master.dbo.sysdatabases
open DatabaseCursor
fetch next from DatabaseCursor into @database

while @@fetch_status = 0
begin
print 'database:' + @database
if @database not in ('tempdb','master','model','msdb')
begin
SET @tsql = 'use master;
alter database [' + @database + '] set offline with rollback immediate;
alter database [' + @database + '] set online;
DECLARE @dbLogName nvarchar(500) ;
Use [' + @database + '];
select @dbLogName = rtrim(ltrim(name)) from sysfiles WHERE FILEID=2;
ALTER DATABASE [' + @database + '] SET SINGLE_USER ;
DBCC SHRINKFILE(@dbLogName , 2);
BACKUP LOG [' + @database + '] WITH TRUNCATE_ONLY ;
DBCC SHRINKFILE(@dbLogName , 2);
ALTER DATABASE [' + @database + '] SET MULTI_USER ;
exec(@tsql)
'
end
fetch next from DatabaseCursor into @database
end

close DatabaseCursor
deallocate DatabaseCursor

```

اسکریپت فوق، با SQL Server 2005 سازگار است اما در SQL Server 2008 منسخه شده است! (قسمت truncate کردن) نسخه سازگار با SQL server 2008 آن به صورت زیر است:

```

Declare @database nvarchar(1000)
Declare @tsql nvarchar(4000)
Declare DatabaseCursor Cursor
Local

```

```
Static
For
select name from master.dbo.sysdatabases
open DatabaseCursor
fetch next from DatabaseCursor into @database

while @@fetch_status = 0
begin
print 'database:' + @database
if @database not in ('tempdb','master','model','msdb')
begin
SET @tsql = 'use master;
    alter database [' + @database + '] set offline with rollback immediate;
    alter database [' + @database + '] set online;
DECLARE @dbLogName nvarchar(500) ;
Use [' + @database + '];
select @dbLogName = rtrim(ltrim(name)) from sysfiles WHERE FILEID=2;
ALTER DATABASE [' + @database + '] SET RECOVERY SIMPLE;
ALTER DATABASE [' + @database + '] SET SINGLE_USER ;
DBCC SHRINKFILE(@dbLogName , 2);
ALTER DATABASE [' + @database + '] SET MULTI_USER ;
ALTER DATABASE [' + @database + '] SET RECOVERY FULL;
exec(@tsql)
end
fetch next from DatabaseCursor into @database
end

close DatabaseCursor
deallocate DatabaseCursor
```

مأخذ اصلی مورد استفاده:

<http://go.microsoft.com/fwlink/?LinkId=111531&clcid=0x409>

نظرات خوانندگان

نوبسنده: Salar تاریخ: ۱۳۸۷/۰۸/۰۶ ۲۲:۴۶:۰۰

جناب نصیری، با اجازه به شما لینک دادم

نوبسنده: میثم تاریخ: ۱۳۸۸/۰۷/۰۴ ۱۷:۰۴:۴۶

خب چرا تو Properties دیتابیس Auto shrink رو نمیکنید؟

نوبسنده: وحید نصیری تاریخ: ۱۳۸۸/۰۷/۰۴ ۱۷:۲۰:۲۸

زمانیکه صحبت از دیتابیس شیرپوینت می‌شود یعنی دیتابیس بالای 50 گیگ.

و به 2 دلیل در اینجا نباید از auto shrink استفاده کرد:

- کاهش حجم قابل ملاحظه ذکر شده توسط auto shrink رخ نمی‌دهد.

- یکی از مواردی است که به شدت روی کارآیی دیتابیسی حجیم تأثیر منفی دارد و همچنین سبب ایجاد fragmentation می‌شود و در اکثر مقالات استفاده از آن به شدت نهی شده‌است.

آیا می‌دانید آخرین باری که از یک دیتابیس مفروض بک‌آپ گرفته شده، چه زمانی بوده است؟ (یا اینکه اصلاً چه اهمیتی دارد؟!)

```

USE master;
SELECT B.name AS Database_Name,
       ISNULL(STR(ABS(DATEDIFF(day, GetDate(),
       MAX(Backup_finish_date)))), 'NEVER') AS DaysSinceLastBackup,
       ISNULL(CONVERT(char(10), MAX(backup_finish_date), 101), 'NEVER') AS
       LastBackupDate
FROM   master.dbo.sysdatabases B
LEFT OUTER JOIN msdb.dbo.backupset A
ON     A.database_name = B.name
AND    A.type = 'D'
GROUP BY
       B.Name
ORDER BY
       B.name
  
```

	Database_Name	DaysSinceLastBackup	LastBackupDate
22	Northwind	NEVER	NEVER
23	outtime	5	10/31/2008

همانطور که مطلع هستید در SQL Server 2008 امکان فشرده سازی خودکار بک‌آپ‌ها هم فراهم شده است. با استفاده از اسکریپت زیر می‌توان از درصد فشرده سازی بک‌آپ‌ها گزارش گرفت:

```

SELECT bs.server_name,
       bs.database_name AS 'Database Name',
       CONVERT (BIGINT, bs.backup_size / 1048576 ) AS
       'Uncompressed Backup Size (MB)',
       CONVERT (BIGINT, bs.compressed_backup_size / 1048576 ) AS
       'Compressed Backup Size (MB)',
       CONVERT (NUMERIC (20,2), (CONVERT (FLOAT, bs.backup_size) /
       CONVERT (FLOAT, bs.compressed_backup_size))) AS 'Compression Ratio',
       DATEDIFF (SECOND, bs.backup_start_date, bs.backup_finish_date) AS
       'Backup Elapsed Time (sec)',
       bs.backup_finish_date AS 'Backup Finish Date'
FROM   msdb.dbo.backupset AS bs
WHERE  DATEDIFF (SECOND, bs.backup_start_date, bs.backup_finish_date) > 0
AND   bs.backup_size > 0
AND   bs.type = 'D' -- Change to L if you want Log backups
ORDER BY
       bs.backup_finish_date DESC
  
```

	server_name	Database Name	Uncompressed Backup Size (MB)	Compressed Backup Size (MB)	Compression Ratio	Backup Elapsed Time (sec)	Backup Finish Date
19	VAHID	RSSReader	42169	9871	4.27	929	2008-10-24 23:15:33.000

برای فعال سازی گزینه فشرده سازی بک‌آپ‌ها با استفاده از دستورات T-SQL می‌توان به صورت زیر عمل کرد (در SQL server :)(2008

```
USE master;
EXEC sp_configure 'backup compression default', '1';
RECONFIGURE WITH OVERRIDE;
```

ماخذ مورد استفاده:

<http://www.sqlservercentral.com>

چند روز قبل مشکلی رخ داده بود به این شرح!

سروری که SQL server بر روی آن نصب بود با خاطر SQL server، بیش از 50 درصد CPU usage مداوم پیدا کرده بود. عموماً مصرف CPU اس کیوال سرور روی سرورهای قوی بالا نیست و تداوم این حالت به این شدت یعنی بروز مشکل. اینجا است که این سؤال پیش می‌بادد، SQL Server الان داره چکار می‌کنه که تا این حد به صورت مداوم مصرف CPU آن بالا رفته؟ (حدوداً 2 ساعت تمام به صورت مداوم مصرف CPU بالای 50 درصد بود) با استفاده از کوئری زیر می‌شود، عبارات SQL ای را که هم اکنون در حال اجرا هستند را به صورت زنده مشاهده کرد: (در اس کیوال سرور 2005 به بعد)

```
USE master;
SELECT st.text, r.session_id, r.status, r.command, r.cpu_time, r.total_elapsed_time
FROM sys.dm_exec_requests r
CROSS APPLY sys.dm_exec_sql_text(sql_handle) AS st
```

مشکل از بروز یک loop بود که به این صورت دقیقاً کوئری مربوطه تشخیص داده شد و برطرف شد.

همچنین با استفاده از کوئری زیر می‌توان آخرین 50 کوئری اجرا شده در SQL server را به همراه زمان اجرای آنها گزارشگیری کرد:

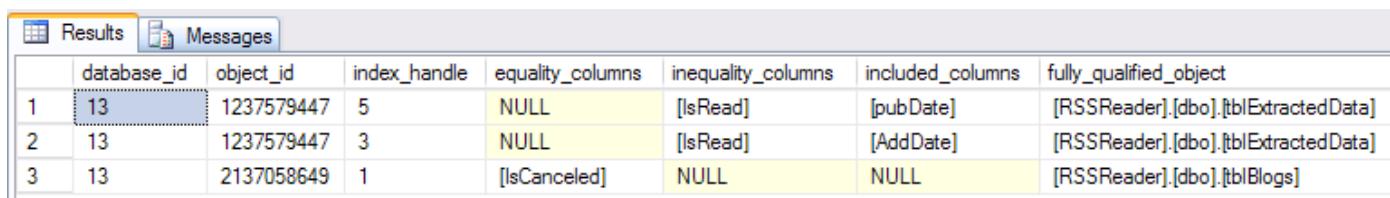
```
SELECT TOP 50
deqs.last_execution_time AS [Time],
dest.text AS [Query]
FROM sys.dm_exec_query_stats AS deqs
CROSS APPLY sys.dm_exec_sql_text(deqs.sql_handle) AS dest
ORDER BY
deqs.last_execution_time DESC
```

اگر علاقمند باشید در مورد این نوع کوئری‌ها اطلاعات بیشتری کسب کنید، مطالعه مقاله زیر توصیه می‌شود:

<http://www.sqlteam.com/article/dynamic-management-views>

در مطالب قبلی به اختصار در مورد dynamic management views که از 2005 SQL server به بعد ارائه شده‌اند مثال‌هایی کاربردی ارائه گشتند. یکی دیگر از قابلیت‌های فوق العاده مهم این DMV‌ها، پیشنهاد ایجاد ایندکس بر روی جداول است. این پیشنهادات بر اساس آمارهای جمع‌آوری شده توسط موتور بهینه ساز اجرای کوئری‌ها در اس کیوال سرور به شما ارائه خواهد شد. برای مثال کوئری زیر را در management studio اجر نمایید:

```
USE master;
SELECT d.database_id,
       d.object_id,
       d.index_handle,
       d.equality_columns,
       d.inequality_columns,
       d.included_columns,
       d.statement AS fully_qualified_object,
       gs.*
FROM   sys.dm_db_missing_index_groups g
JOIN   sys.dm_db_missing_index_group_stats gs
ON     gs.group_handle = g.index_group_handle
JOIN   sys.dm_db_missing_index_details d
ON     g.index_handle = d.index_handle
```



	database_id	object_id	index_handle	equality_columns	inequality_columns	included_columns	fully_qualified_object
1	13	1237579447	5	NULL	[IsRead]	[pubDate]	[RSSReader].[dbo].[tblExtractedData]
2	13	1237579447	3	NULL	[IsRead]	[AddDate]	[RSSReader].[dbo].[tblExtractedData]
3	13	2137058649	1	[IsCanceled]	NULL	NULL	[RSSReader].[dbo].[tblBlogs]

خروجی حاصل لیستی است که بر اساس تفاسیر موتور بهینه ساز اجرای کوئری‌ها بدست آمده است. بر equality_columns اساس حالت‌هایی مانند table.column = constant_value پیش‌بینی شده است. بر اساس حالت‌هایی مانند inequality_columns برای حالت‌هایی است که می‌خواهیم ایندکس ایجاد شده محدودیت اندازه 900 بایت را نداشته باشد، یا نوع داده‌ای مورد استفاده برای مثال max nvarchar و امثال آن باشد (text و ntext مجاز نیست) و مواردی از این دست.

هم مشخص می‌کند که این ایندکس دقیقاً باید بر روی چه دیتابیس و جدولی ایجاد شود.

تذکر: این آمارهای جمع‌آوری شده پس از هر بار ریاستارت سرور، صفر خواهند شد.

اکنون این سؤال مطرح می‌شود که چگونه از این اطلاعات استفاده کنیم؟ دقیقاً بر اساس INCLUSIVE_COLUMNS و EQUALITY_COLUMNS، INEQUALITY_COLUMNS عمل کرد:

```
CREATE NONCLUSTERED INDEX <unique index name>
ON <FULL_TABLE_NAME> (<EQUALITY_COLUMNS>,<INEQUALITY_COLUMNS>) -- exclude INEQUALITY_COLUMNS if NULL
INCLUDE (<INCLUDED_COLUMNS>); -- exclude INCLUDED_COLUMNS if NULL
```

خوب، پس از گزارشگیری، ممکن است لیست بلند بالای تهیه شود. کوئری زیر عبارات `create index` مورد نظر را بر اساس این قابلیت جدید تولید خواهد کرد:

```

SELECT mig.index_group_handle,
       mid.index_handle,
       migs.avg_total_user_cost AS AvgTotalUserCostThatCouldbeReduced,
       migs.avg_user_impact AS AvgPercentageBenefit,
       'CREATE INDEX missing_index_' + CONVERT (varchar, mig.index_group_handle)
       + '_' + CONVERT (varchar, mid.index_handle)
       + ' ON ' + mid.statement
       + ' (' + ISNULL (mid.equality_columns, '')
       + CASE
           WHEN mid.equality_columns IS NOT NULL AND mid.inequality_columns
           IS NOT NULL THEN ','
           ELSE ''
         END
       + ISNULL (mid.inequality_columns, '')
       + ')'
       + ISNULL (' INCLUDE (' + mid.included_columns + ')', '') AS
create_index_statement
FROM sys.dm_db_missing_index_groups mig
INNER JOIN sys.dm_db_missing_index_group_stats migs ON migs.group_handle = mig.index_group_handle
INNER JOIN sys.dm_db_missing_index_details mid ON mig.index_handle = mid.index_handle

```

	AvgPercentageBenefit	create_index_statement
1	78.51	CREATE INDEX missing_index_6_5 ON [RSSReader].[dbo].[tblExtractedData] ([IsRead]) INCLUDE ([pubDate])
2	76.67	CREATE INDEX missing_index_4_3 ON [RSSReader].[dbo].[tblExtractedData] ([IsRead]) INCLUDE ([AddDate])
3	89.32	CREATE INDEX missing_index_2_1 ON [RSSReader].[dbo].[tblBlogs] ([IsCanceled])

مزایای

ایجاد ایندکس‌های صحیح بر اساس نیازهای واقعی کاری: سریعتر شدن اجرای کوئری‌های جستجو در تعداد رکوردهای بالا مرتب سازی سریعتر نتایج (sorting) کوئری‌هایی که بر اساس عبارت GROUP BY ایجاد شده‌اند، سریعتر اجرا خواهند شد

نظرات خوانندگان

نویسنده: Alex

تاریخ: ۱۳۸۷/۰۸/۲۶ ۰۸:۳۲:۰۰

ممنون خیلی مفید بود. البته بنظر بنده ای حقیر تمام پستهای شما واسه من مفید بوده و هست. امیدوارم که بقیه اش هم اینجوری باشه.
همیشه موفق و سلامت باشید.

عنوان: پیدا کردن لیست SQL server های نصب شده در یک شبکه
نویسنده: وحید نصیری
تاریخ: ۱۴:۰۴:۳۸ ۱۳۸۷/۰۸/۲۸
آدرس: www.dotnettips.info
برچسبها: SQL Server

با آمدن 2008 SQL server استفاده از کتابخانه [SQL-DMO](#) برای انجام یک سری از امور بر روی اس کیوال سرور با استفاده از برنامه نویسی منسوخ شد. یکی از توانایی‌های این کتابخانه لیست کردن سرورهای اس کیوال (قابل دسترسی) موجود در شبکه بود. برای مثال توسط این کتابخانه به صورت زیر می‌توان اینکار را انجام داد:

در قطعه کد زیر فرض بر این است که ارجاعی به کتابخانه sqldmo را در برگه com مربوط به project->add reference اضافه کردید:

```
using SQLDMO;
using System.Collections.Generic;

public static List<string> GetSQLServersList2()
{
    List<string> result = new List<string>();
    ApplicationClass sqlApp = new ApplicationClass();
    NameList lst = sqlApp.ListAvailableSQLServers();
    for (int i = 1; i <= lst.Count; i++)
        result.Add(lst.Item(i));
    lst = null;
    sqlApp = null;
    return result;
}
```

با منسوخ شدن این کتابخانه COM (که تنها تا اس کیوال سرور 2005 پشتیبانی می‌شود)، در نگارش‌های جدید (و قدیم) اس کیوال سرور، با استفاده از قطعه کد زیر می‌توان لیست تمام SQL server های نصب شده در یک شبکه به همراه instance های آنها را بدست آورد.

```
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;

public class CListServers
{
    public static List<string> GetSQLServersList()
    {
        List<string> result = new List<string>();

        // Retrieve the enumerator instance and then the data.
        var instance = SqlDataSourceEnumerator.Instance;
        var table = instance.GetDataSources();

        // Display the contents of the table.
        foreach (DataRow row in table.Rows)
        {
            result.Add(string.Format("{0}\\"{1}", row[0], row[1]));
        }
        return result;
    }
}
```

راه دیگر:

کتابخانه COM یاد شده (SQL-DMO) در SQL server 2008 با کتابخانه [SMO](#) جایگزین شده است.

در این حالت خواهیم داشت:

```
using System.Collections.Generic;
using System.Data;
using Microsoft.SqlServer.Management.Smo;

public class CListServers
{
    public static List<string> GetSQLServersListSMO()
    {
        List<string> result = new List<string>();
        DataTable dt = SmoApplication.EnumAvailableSqlServers(false);
        if (dt.Rows.Count > 0)
        {
            foreach (DataRow dr in dt.Rows)
            {
                result.Add(dr["Name"].ToString());
            }
        }
        return result;
    }
}
```

تقریباً کلیه اعمالی که از طریق management studio قابل انجام هستند با کمک این کتابخانه نیز از طریق برنامه نویسی می‌توان به آنها پرداخت. برای مثال تهیه اسکریپت کلیه جداول، تریگرها و غیره.

عنوان: چرا نباید از کوئری‌های select * استفاده کرد؟

نویسنده: وحید نصیری

تاریخ: ۱۳۸۷/۰۹/۰۵ ۱۷:۱۷:۲۵

آدرس: www.dotnettips.info

برچسب‌ها: SQL Server

عموماً اولین پاسخی که به این سؤال داده می‌شود این است که این نوع کوئری‌ها اطلاعات زیادی را باز می‌گردانند و در نتیجه ترافیک شبکه بی‌جهت افزایش خواهد یافت. اما اگر طراحی دیتابیس صحیح بوده و اصول نرم‌السازی در آن پیاده سازی شده باشد، این پاسخ آنچنان صادق نخواهد بود (زیرا جداول اینگونه دیتابیس‌ها از تعداد فیلدهای بسیاری تشکیل نخواهند شد). برای مثال به نتیجه اجرای کوئری‌های زیر بر روی دیتابیس AdventureWorks با 89 هزار رکورد، دقت بفرمایید:

```
SELECT * FROM Production.TransactionHistoryArchive  
WHERE ReferenceOrderID < 100
```

```
SELECT ReferenceOrderLineID FROM Production.TransactionHistoryArchive  
WHERE ReferenceOrderID < 100
```

اختلاف ترافیک شبکه در این مثال تنها 15K یا حدوداً 10 درصد است (180K در مقابل 165K). هر چند ارزش بررسی و برطرف کردن را دارد اما تفاوت حاصل آنچنان قابل ملاحظه نیست.

مهمنترین دلیلی که اینجا باید به آن دقت داشت، تفاوت چشمگیر execution plan این دو کوئری (Ctrl-L) و بحث index coverage است. اس کیوال سرور برای اجرای بهینه کوئری‌ها از ایندکس‌های موجود استفاده خواهد کرد. اگر ایندکس تعریف شده از تمامی فیلدهای درخواستی شما تشکیل شده باشد، دیگر حتی به سراغ جدول هم نخواهد رفت (به این مفهوم، پوشش ایندکسی گفته می‌شود).

Query 1: Query cost (relative to the batch): 99%

```
SELECT * FROM [Production].[TransactionHistoryArchive] WHERE [ReferenceOrderID]<@1
```

Execution plan for Query 1:
SELECT * FROM [AdventureWorks].[Production].[TransactionHistoryArchive] WHERE [ReferenceOrderID]<@1
Clustered Index Scan
[AdventureWorks].[Production].[TransactionHistoryArchive]
Cost: 100 %

Query 2: Query cost (relative to the batch): 1%

```
SELECT [ReferenceOrderLineID] FROM [Production].[TransactionHistoryArchive] WHERE [ReferenceOrderID]<@1
```

Execution plan for Query 2:
SELECT [ReferenceOrderLineID] FROM [AdventureWorks].[Production].[TransactionHistoryArchive] WHERE [ReferenceOrderID]<@1
Index Seek
[AdventureWorks].[Production].[TransactionHistoryArchive]
Cost: 100 %

برای تولید تصویر فوق، کلیدهای Ctrl+L management studio را در فشار دهید.

این دیتابیس را از آدرس زیر می‌توانید دریافت کنید:

<http://www.codeplex.com/MSFTDBProdSamples>

کوئری اول از مزایای پوشش ایندکسی برخودار نخواهد بود (از روش جستجوی Clustered Index استفاده می‌کند) و در حالت دوم از Index Seek استفاده می‌گردد. حالت Index Seek یک صد بار بهینه‌تر از استفاده از Clustered Index عمل می‌کند زیرا در

حالت کوئری اول باید تمامی رکوردهای جدول بررسی شوند (این عدد از مقایسه نتایج execution plan بدست آمده است). تنها در صورتیکه بر روی تمامی فیلدهای جدول ایندکس تعریف کرده باشد (که اصلاً توصیه نمی‌شود)، کوئری اول توسط ایندکس‌ها پوشش داده شده و سریع اجرا خواهد شد.

بنابراین اگر از کندی اجرای کوئری‌ها با تعداد رکورد بالا شکایت دارید بهتر است نگاهی به نحوه تعریف آنها داشته باشید و تنها فیلدهایی را در کوئری تعریف کنید که به آنها نیاز دارید. در این حالت از مزایای پوشش ایندکسی برخودار شده، کوئری‌های سریعتری را خواهید داشت و همچنین در این حالت میزان مصرف CPU و حافظه نیز بر روی سرور کمتر خواهد بود.

همچنین در حالت کوئری‌هایی از نوع دوم ذکر شده، موتور بهینه ساز اس کیوال سرور پیشنهادات بهتری را برای ایجاد ایندکس‌های جدید و [گوشزد نمودن کمبود آنها](#) با ارائه included columns مناسب، ارائه می‌دهد.

علاوه مشخص ساختن تعداد دقیق فیلدهای مورد نیاز، نگهداری برنامه را ساده‌تر ساخته و فیلدهای اضافه شده آتی سبب تغییر رفتار کوئری‌ها برنامه نخواهند شد و استفاده نکردن از آن نشانه این است که هیچ برآورده از ابعاد واقعی کار در دست نیست.

ما آخذ:

[Speed Up Your Site! 8 ASP.NET Performance Tips](#)

[The real reason SELECT * queries are bad: index coverage](#)

نظرات خوانندگان

نویسنده: حمیدرضا

۱۳۸۷/۰۹/۰۵ ۱۹:۰۲:۰۰

تاریخ:

چند وقتیه که وبلاگ شما رو می‌خونم، استفاده می‌کنم و لذت می‌برم، گفتم حتی شده به قیمت یه نظر نابجا و بی‌ربط به موضوع هم شده، ازتون تشکر بکنم و برآتون آرزوی موفقیت.
موفق باشید.

با تکامل SQL server و بهبودهای حاصل شده، یک سری از ویژگی‌های موجود صرفاً جهت حفظ سازگاری با نگارش‌های قبلی ارائه می‌شوند. لیست کامل آنها را در آدرس زیر می‌توان مشاهده نمود:

[Deprecated Database Engine Features in SQL Server 2008](#)

لیست بلند بالایی است. اما در یک محیط کاری، نوع‌های زیر از سایر موارد ذکر شده بیشتر مورد استفاده قرار می‌گیرند:
منسوخ شده‌ها: `image` ، `text` ، `ntext` ، `varbinary` و `nvarchar` از نوع `max` دار

عموماً علت استفاده از نوع‌های `text` یا `ntext` (نمونه یونیکد) ، مشخص نبودن تعداد کاراکتری است که کاربر قرار است وارد کند. برای مثال یک سایت خبری ایجاد کرده‌اید و طول محتوای خبر ثبت شده در بانک اطلاعاتی از یک خبر به خبر دیگر کاملاً متفاوت است. در اینجا برای حل این مشکل از نوع‌های `text` یا `ntext` استفاده می‌شود (این مورد تا اس‌کیوال سرور 2000 توصیه می‌شود).

سرور 2000 تا `varchar max` 2,147,483,648 کاراکتر را می‌تواند ذخیره کند، یعنی تا 2 GB و `nvarchar max` تا نصف این مقدار را. در اس‌کیوال سرور 2000 محدودیت 8000 کاراکتر برای نوع `varchar` وجود داشت (و نوع `nvarchar` تا 4000 کاراکتر).

مزایای استفاده از نوع‌های `max` دار (از اس‌کیوال سرور 2005 به بعد) :

بهمود کارآئی کوئری‌های جستجو نسبت به نوع‌های `Text`
اگر **مطلوب** تشخیص کمبود ایندکس‌ها را دنبال کرده باشد، در آنجا ذکر شد که در قسمت `included columns` نمی‌توان از `text` و `ntext` استفاده کرد اما نوع‌های `max` دار متنی مجاز نمی‌شوند.

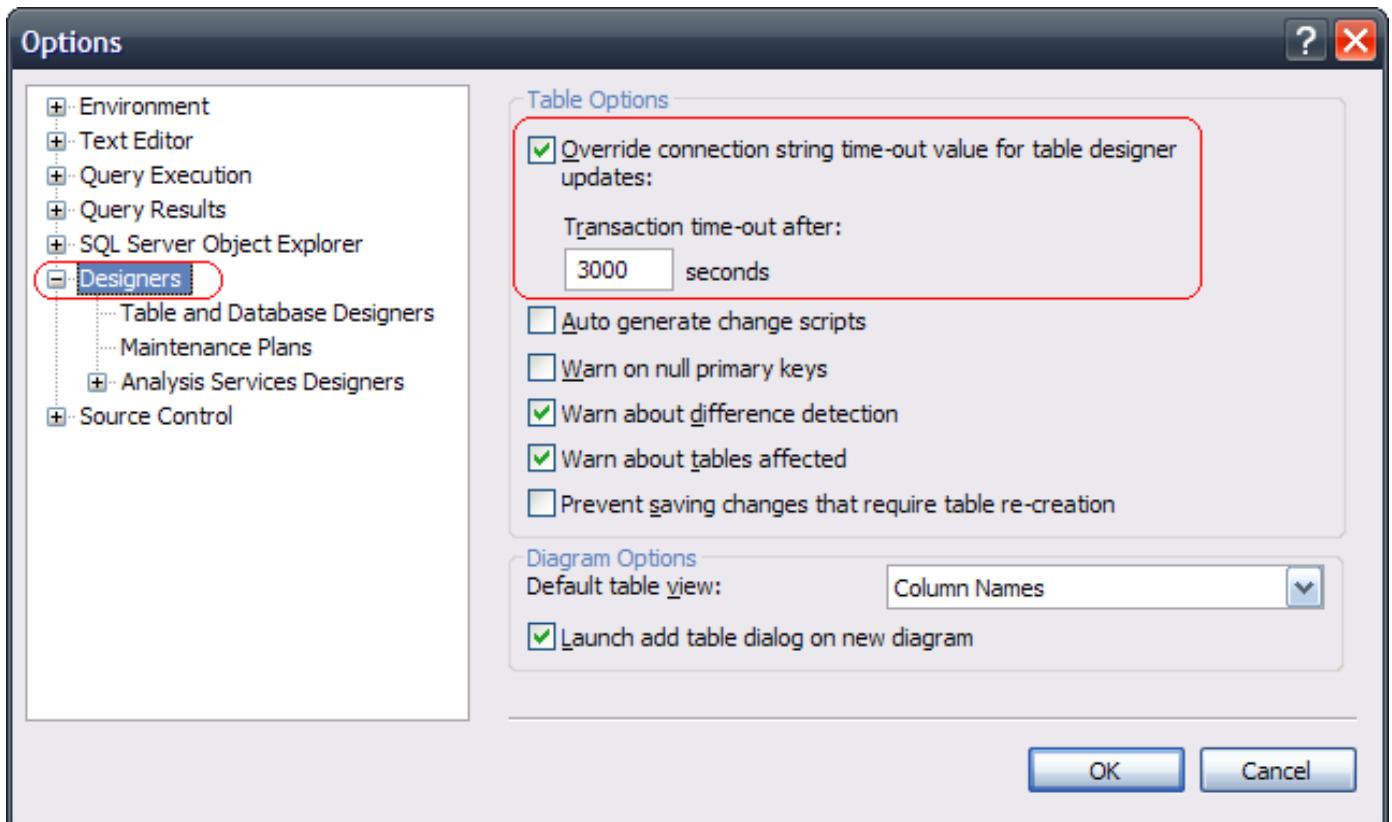
امکان استفاده از فیلد‌های `max` دار برای مرتب سازی کوئری مجاز است. (به شخصه با این مورد زیاد برخورد داشتم. برای مثال امکان سورت کردن یک گرید را در ASP.Net فراهم کرده‌اید و کاربر با کلیک بر روی سر ستون فیلدی از نوع `text` با یک خط متوقف خواهد شد)

امکان استفاده از نوع‌های `Text` به عنوان متغیر در رویه‌های ذخیره شده یا توابع T-SQL مهیا نیست اما این محدودیت در نوع‌های `max` دار برطرف شده است.

نوع‌های `text` را در توابع `CHARINDEX` ، `REPLACE` و `SUBSTRING` نمی‌توان بکار برد (برخلاف **نوع‌های متنی `max` دار**).
بعد از این توضیحات شاید علاقمند شده باشد که نوع‌های فیلد‌های قدیمی را به نوع‌های جدید تبدیل کنید (مراجعه به `management` ، `studio` ، تغییر نوع فیلد و کلیک بر روی دکمه ذخیره در نوار ابزار آن). اگر تعداد رکوردها بالا باشد، بدون شک با خطای زیر متوقف خواهید شد:

Timeout expired. The timeout period elapsed prior to completion of the operation or the server is not responding.

برای حل این مشکل می‌توان مقدار پیش فرض `timeout` را مطابق تصویر زیر تغییر داد (منوی `tools` گزینه `options`):

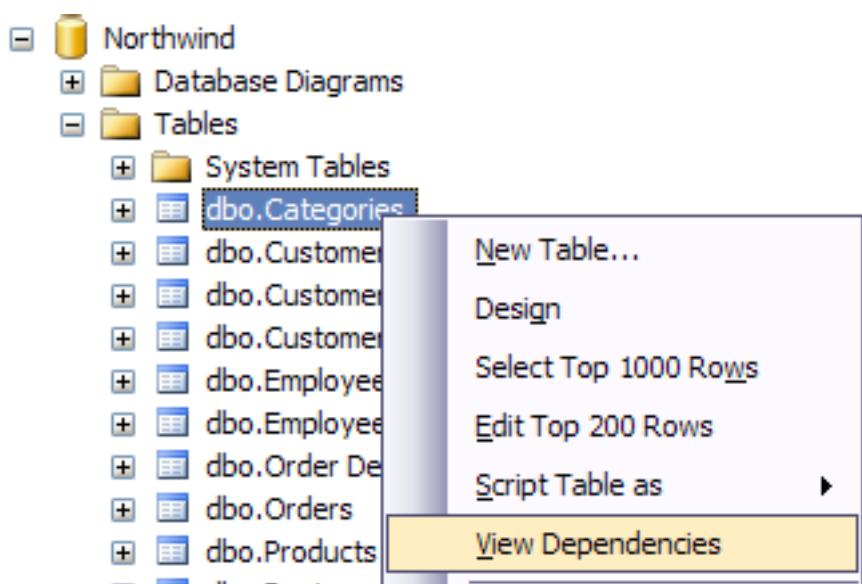


بعد از این تغییر به سادگی می‌توان عملیات ارتقاء را بدون نگرانی از بروز خطای فوق انجام داد.

با بالا رفتن تعداد اشیاء تعریف شده در SQL server، نگهداری آنها نیز مشکل‌تر می‌شود. در این حالت تغییر کوچکی در یکی از اشیاء ممکن است باعث از کار افتادن قسمتی از سیستم شود. بنابراین قبل از هر گونه تغییری در یک شیء، ابتدا باید سایر اشیاء وابسته به آن را یافت و در نظر داشت (dependencies). برای این منظور (impact analysis) راهکارهای مختلفی در SQL server وجود دارد که در ادامه به آن‌ها خواهیم پرداخت:

(الف) استفاده از [امکانات management studio](#) (اس کیوال سرور 2005 به بعد)

ساده‌ترین راه ممکن که گزارش مفصلی را نیز ارائه می‌دهد، کلیک بر روی یک شیء در [management studio](#) و انتخاب گزینه view dependencies است (شکل زیر).



در صفحه ظاهر شده می‌توان اشیایی را که شیء مورد نظر به آنها وابسته است، مشاهده نمود یا بر عکس (اشیایی که عملکرد آنها وابسته به شیء انتخابی است را نیز می‌توان [ملاحظه کرد](#)).

(ب) کوئری گرفتن از جداول سیستمی

امکانات قسمت قبل را با استفاده از اطلاعات جدول syscomments نیز می‌توان شبیه سازی کرد. در این جدول اطلاعات تعاریف کلیه view، رویه‌های ذخیره شده و غیره نگهداری می‌شود. برای مثال فرض کنید قصد داریم در جدول Orders دیتابیس Northwind، نام فیلد OrderDate را تغییر دهیم. قبل از این‌کار بهتر است [کوئری زیر را](#) اجرا کنیم تا نام اشیاء وابسته را بدست آوریم:

```

SELECT NAME
FROM syscomments c
JOIN sysobjects o
ON c.id = o.id
  
```

پیدا کردن وابستگی‌های اشیاء در SQL Server

```
WHERE TEXT LIKE '%OrderDate%'  
AND TEXT LIKE '%Orders%'
```

	name
1	Orders Qry
2	Quarterly Orders
3	Invoices
4	Sales by Category
5	CustOrdersOrders
6	SalesByCategory

این روش انعطاف پذیری بیشتری را نسبت به امکانات قسمت الف، ارائه می‌دهد. برای نمونه فرض کنید می‌خواهید در یک دیتابیس کلیه اشیایی که عملیات `delete` را انجام می‌دهند پیدا کنید (جستجوی اشیاء حاوی یک عبارت خاص). در این مورد خواهیم داشت:

```
SELECT NAME  
FROM syscomments c  
JOIN Northwind.dbo.sysobjects o  
ON c.id = o.id  
WHERE TEXT LIKE '%delete%'
```

ج) استفاده از رویه ذخیره شده سیستمی `sp_depends`

جدول سیستمی دیگری در اس کیوال سرور به نام `sysdepends` وجود دارد که اطلاعات وابستگی‌های اشیاء در آن‌ها نگهداری می‌شود. برای دسترسی به اطلاعات این جدول، اس کیوال سرور رویه ذخیره شده سیستمی `sp_depends` را ارائه داده است. برای مثال فرض کنید می‌خواهیم لیست اشیایی را که به جدول `Orders` در دیتابیس `Northwind` وابسته هستند، پیدا کنیم. در این حالت داریم:

```
USE Northwind  
EXEC sp_depends 'Orders'
```

	name	type
1	dbo.CustOrderHist	stored procedure
2	dbo.CustOrdersOrders	stored procedure
3	dbo.Employee Sales by Country	stored procedure
4	dbo.Invoices	view
5	dbo.Orders Qry	view
6	dbo.Product Sales for 1997	view
7	dbo.Quarterly Orders	view
8	dbo.Sales by Category	view
9	dbo.Sales by Year	stored procedure
10	dbo.Sales Totals by Amount	view
11	dbo.SalesByCategory	stored procedure
12	dbo.Summary of Sales by Quarter	view
13	dbo.Summary of Sales by Year	view

(د) استفاده از schema view

با استفاده از view سیستمی INFORMATION_SCHEMA.ROUTINES، که از ترکیب جداول sysobjects و syscolumns ایجاد شده است نیز می‌توان عملکرد sp_depends را شبیه سازی کرد اما جداول view ها از گزارش آن حذف شده‌اند.

```
SELECT routine_name,
       routine_type
  FROM INFORMATION_SCHEMA.ROUTINES
 WHERE routine_definition LIKE '%Orders%'
```

در جدول زیر مقایسه‌ای از امکانات و گزارش حاصل از این چهار روش با هم مقایسه شده‌اند:

Method	Type	Tables	Views	UDF	Stored Procedures	Triggers	CHECK Constraints
View Dependencies	X	X	X	X	X	X	X
syscomments query			X	X	X	X	X
schema view	X			X	X		
sp_depends	X		X	X	X		X

(۵) استفاده از برنامه SQL Dependency Tracker

نسخه آزمایشی برنامه ذکر شده را از [این آدرس](#) می‌توان دریافت کرد.

همانطور که ملاحظه می‌کنید این جستجوها بر روی اطلاعات ذخیره شده در اس کیوال سرور صورت می‌گیرند و اگر در کدهای خود در خارج از اس کیوال سرور مخلوطی از عبارات اس کیوال را داشته باشد، نگهداری آنها بسیار مشکل خواهد بود. بنابراین تا حد ممکن باید عملیات مرتبط را در دیتابیس و توسط اشیاء اس کیوال سرور مانند رویه‌های ذخیره شده، view ها و امثال آن‌ها انجام داد تا این جدا سازی بهخوبی صورت گرفته و در زمان نیاز به انجام تغییرات، ردگیری اشیاء وابسته به سادگی صورت گیرد.

عنوان: SQL Server 2005 SP3
نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۰۹/۲۷
آدرس: www.dotnettips.info
برچسبها: SQL Server

خبر! نگارش نهایی سرویس پک 3 اس کیوال سرور 2005 ارائه شد.

[دربافت](#)

[لیست باگ‌های فیکس شده](#)

نظرات خوانندگان

نویسنده: مهدی پایروند
تاریخ: ۱۳۸۷/۱۲/۱۸ ۱۰:۴۳:۰۰

سلام، اگه که سرویس پک های قبلی رو نصب نکرده باشیم هم میتوانیم سرویس پک ۳ رو نصب کنیم یا نه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۱۸ ۱۲:۲۸:۰۰

بله. مطابق مستندات رسمی آن:

SP3 can be used to upgrade from any previous instance of SQL Server 2005

<http://blogs.technet.com/dataplatforminsider/archive/2008/12/10/sql-server-2005-sp3-now-available-for-download.aspx>

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۲/۱۸ ۱۲:۲۹:۰۰

ضمینا برای sp3 هم آپدیت اومده که باید نصب شود:

<http://support.microsoft.com/kb/959195>

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۹/۲۸ ۱۴:۰۲:۰۰

[SQL Server 2005 SP4](#)

یکی دیگر از معایب کوئری‌های select* در SQL server این است که تغییرات حاصل در فیلدهای جداول یک بانک اطلاعاتی را در view های ساخته شده از این نوع کوئری‌ها منعکس نمی‌کند.

برای مثال جدول `tblTreeItems` را با سه فیلد `id`, `parent` و `title` در نظر بگیرید. فرض کنید بر این اساس view زیر را ساخته‌ایم:

```
CREATE VIEW GetData
as
SELECT * FROM tblTreeItems
```

اکنون به جدول فوق، فیلد جدید `isActive` را اضافه می‌کنیم. پس از این عملیات اگر کوئری ساده `SELECT * FROM GetData` را اجرا کنیم، فیلد جدید `isActive` را در آن نخواهیم دید (برخلاف انتظار که می‌باشد کوئری `select *` رکوردهای تمام فیلدهای جدول را برابر می‌گرداند). در اینجا ممکن است مدتی وقت صرف دیباگ کردن سیستم شود که چرا تغییرات جدید اعمال نشده و چرا سیستمی که تا چند لحظه پیش داشت کار می‌کرد الان از کار افتاد!).

باید در نظر داشت که هنگام ایجاد یک view، تصویری از تمامی فیلدهای مورد استفاده در آن زمان، جهت بالابردن کارآیی کوئری و عدم محاسبه مجدد فیلدها در جداول سیستمی ذخیره می‌گردد (* با نام فیلدهای همان زمان ایجاد (نه زمان فعلی)، جایگزین خواهد شد). این تصویر ایستا است و با تغییر فیلدهای یک جدول به روز نخواهد شد.

برای به روز کردن view ها و stored procedures پس از تغییرات ساختاری در جداول، باید مجدد آنها را کامپایل کرد. برای این منظور راه‌های زیادی وجود دارد، برای مثال `drop` کردن یک view و ایجاد مجدد آن. یا باز کردن آن view در management studio (حالت `alter query`) و سپس فشردن دکمه F5 جهت اجرای مجدد کوئری که این‌بار بر اساس اطلاعات جدید به روز خواهد شد. یا استفاده از رویه‌های سیستمی `sp_recompile` و `sp_refreshview` که برای کامپایل مجدد view ها و رویه‌های ذخیره شده بکار می‌روند.

برای مدیریت ساده‌تر این موارد، اسکریپت زیر تمامی view ها و رویه‌های ذخیره شده یک دیتابیس را به صورت خودکار یافته و آنها را مجدد کامپایل می‌کند: (جهت مشاهده آن نیاز به ثبت نام دارد و رایگان است)

[Refreshing Views and Recompiling Stored Procs](#)

امروز یکی از برنامه‌ها (برنامه ASP.NET) با مشکل زیر مواجه شده بود:

پیغام خطای:

اتصال با سرور اس کیوال قطع شده است. لطفاً با مسئول مربوطه هماهنگ نمائید.
 SQLERR: 4060

این خطای معنای عدم امکان باز کردن دیتابیس است.

در طی این مدت با موارد زیادی از این دست (مشکلات مختلف عدم امکان برقراری ارتباط با اس کیوال سرور) برخورد داشتم که خلاصه تمام آن‌ها تابع زیر شده است:

```

public void CheckSQLServerStat(Exception ex)
{
  try
  {
    SqlException ar = (SqlException) ex;
    switch (ar.Number)
    {
      case 2:
      case 11:
      case 17:
      case 40:
      case 4060:
      case 1326:
      case 17142:
      case 18456:
        HttpContext.Current.Response.Write("<br/>" + "اتصال با سرور اس کیوال قطع شده است." +
        "لطفاً با مسئول مربوطه هماهنگ نمائید" + "<br/> SQLERR:" + ar.Number + "<br/>");
        break;
    }
  }catch{}
}
  
```

هنگام رخدادن یک خطای (در توابعی که با اس کیوال سرور کار می‌کنند)، exception حاصل را به این تابع پاس کرده و خطای حاصل را به صورت مختصر و مفید به کاربر نشان می‌دهم. کاربر متوجه می‌شود که یک مشکل اساسی در سیستم رخداده است. برنامه نویس هم با جستجو در مورد شماره خطای می‌تواند مشکل یابی کند. تابع Response.Write هم بالاتر از هر المان دیگری در صفحه، این خطای شکل واضحی نمایش می‌دهد. (کلا ریخت صفحه را به هم می‌ریزد که از لحاظ روانی لازم است! چون عملای در این حالت سیستم از کار افتاده است)

به management studio اس کیوال سرور که مراجعه کردم، علامت خاصی کنار نام دیتابیس نبود فقط برخلاف سایر دیتابیس‌ها که آیکون + مربوط به باز شدن tree آن وجود دارد، این یک مورد آنرا نداشت. بر روی نام دیتابیس کلیک راست کردم و انتخاب خواص، خطای زیر نمایش داده شد:

An exception occurred while executing a Transact-SQL statement or batch.
 (Microsoft.SqlServer.ConnectionInfo)

Database 'dbName' cannot be opened due to inaccessible files or insufficient memory or disk space. See the SQL Server errorlog for details. (Microsoft SQL Server, Error: 945)

بله! دیتابیس قابل باز شدن نبود چون درایو مربوطه پر شده بود. بعد از خالی کردن درایو و باز کردن فضای لازم، باز هم دیتابیس به حالت اجرایی و قابل استفاده برگشت. یک راه این است که کل سرویس مربوط به اس کیوال سرور را استاپ و استارت کرد. که البته این مورد سبب قطع ارتباط کل مجموعه می‌شود. یا راه دیگر اجرای چند سطر زیر است که دیتابیس را مجدد راه اندازی خواهد کرد (بدون نیاز به ری استارت سرویس)

```
use master;
alter database dbName set OFFLINE;
alter database dbName set online;
```

به این صورت دیتابیس مربوطه به حالت عادی بازگشت و قابل استفاده شد.

عنوان: مشکل اتصال به اس کیوال سرور 2000 از طریق 2008 management studio
نوبتندہ: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۱۵ ۱۱:۱۱:۳۵
آدرس: www.dotnettips.info
برچسب‌ها: SQL Server

هر چند مدتی است که اس کیوال سرور 2008 ارائه شده یا سرویس پک 3 اس کیوال سرور 2005 اخیرا منتشر گردیده، اما هنوز هم هاست‌های ما زحمت ارتقاء را به خود نداده‌اند. (چرا باید پول خرج کنند؟!)
اس کیوال سروری را که الان برای برنامه نویسی از آن استفاده می‌کنم 2008 است و چند روز قبل قصد داشتم به اس کیوال سرور 2000 هاست محترم از طریق 2008 management studio متصل شوم تا تغییرات جدید را به دیتابیس سایت اعمال کنم.
اولین تلاش برای اتصال با خطای زیر متوقف شد!

TITLE: Microsoft SQL Server Management Studio

Failed to retrieve data for this request. (Microsoft.SqlServer.Management.Sdk.Sfc)

ADDITIONAL INFORMATION:

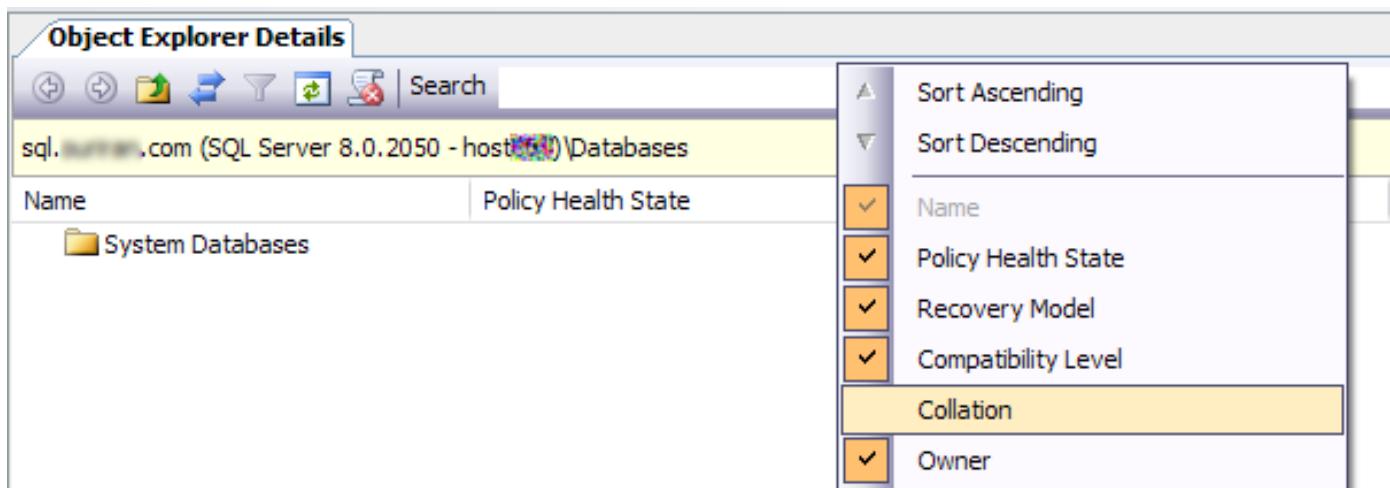
An exception occurred while executing a Transact-SQL statement or batch.
(Microsoft.SqlServer.ConnectionInfo)

(Server user 'hostXYZ' is not a valid user in database 'hostZXY'. (Microsoft SQL Server, Error: 916

فقط هم 2008 management studio این مشکل را دارد. با 2005 تست کردم مشکلی نبود.

بس از مدتی جستجو ، نتیجه کار به صورت زیر است:

به منوی View گزینه Object Explorer Details را در مراجعه کنید (در management studio اس کیوال سرور 2008). روی هدر صفحه‌ای که ظاهر می‌شود کلیک راست کرده و تیک collation را بردارید (شکل زیر). ممکن است collation دیتابیس آفلاین، نال گزارش شود و این مورد مشکل درست می‌کند. تیک collation را که بردارید این مورد بررسی نشده و لیست دیتابیس‌ها نمایش داده می‌شود.



نظرات خوانندگان

نوبتند: msj تاریخ: ۱۳۸۹/۰۷/۲۹ ۱۵:۰۱:۲۲

فکر می کنم که این مشکل در SQL 2008 Service Pack 1 که برای این مشکل در ۱۳۸۹/۰۷/۲۹ ۱۵:۰۱:۲۲

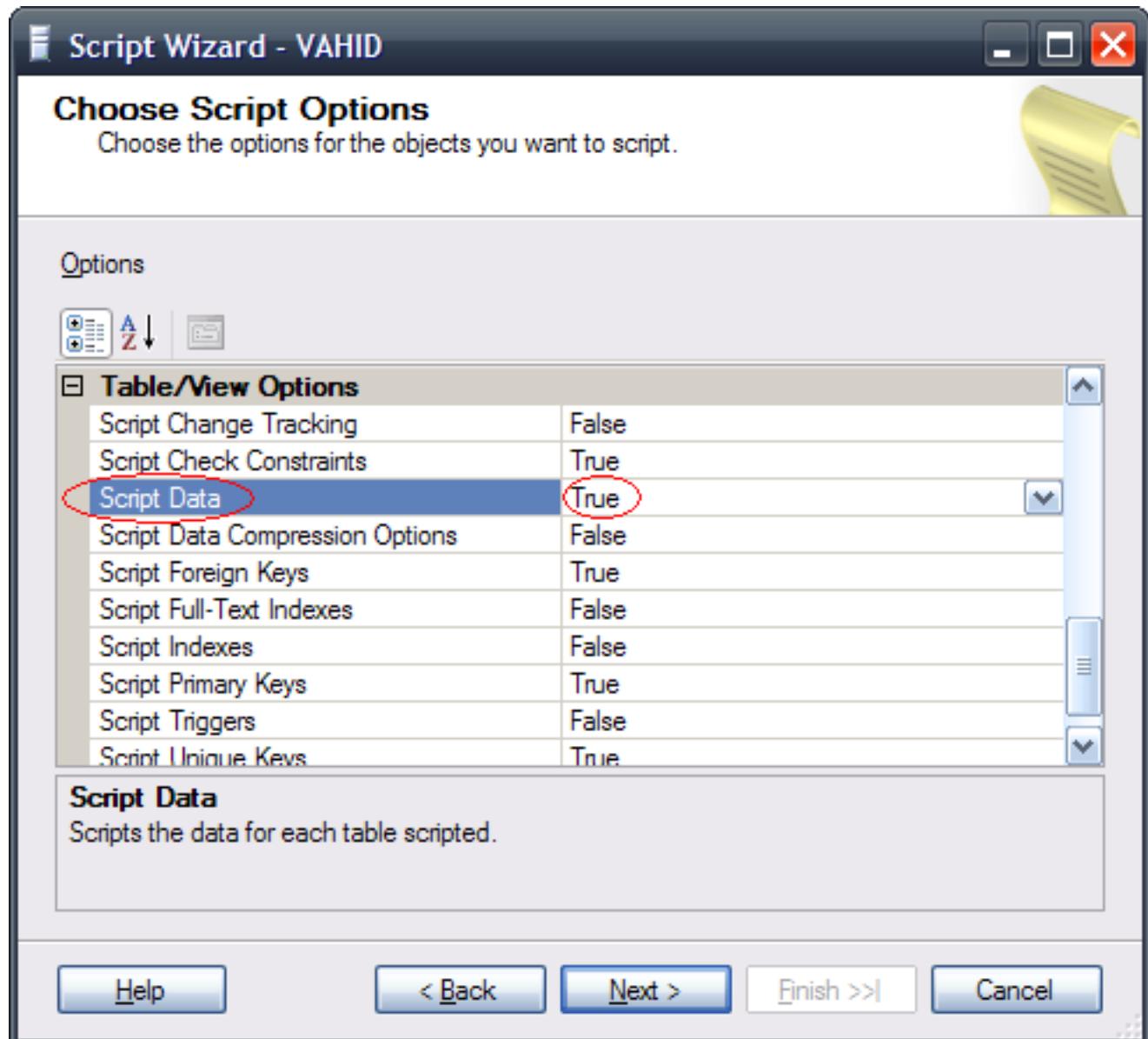
افزونه‌ای برای SQL server 2005 به نام [Database Publishing Wizard](#) وجود داشت/دارد که توسعه‌ی آن به ظاهر برای SQL server 2008 متوقف شده است. توسط این افزونه [می‌توان](#) رکوردهای یک دیتابیس را به صورت عبارات T-SQL درآورد (هر رکورد را به صورت خودکار تبدیل به یک دستور `insert` می‌کند). به این صورت کار انتقال دیتا خصوصاً به هاست‌هایی که دسترسی مستقیم `restore` کردن داده را نمی‌دهند، به سادگی صورت می‌گیرد. تنها کافی است خروجی کار یکبار بر روی دیتابیس مقصد اجرا شود تا رکوردهایی دقیقاً با همان اطلاعات دیتابیس منبع در آن ایجاد گردد.

این قابلیت اکنون جزئی از management studio اس کیوال سرور 2008 است.

برای استفاده از آن بر روی دیتابیس مورد نظر در 2008 management studio کلیک راست کرده و گزینه زیر را انتخاب کنید:

Tasks -> Generate scripts...

در صفحه ویزاردی که ظاهر می‌شود، بر روی next کلیک کرده و در صفحه‌ی بعدی گزینه `script data` را یافته و مقدار آن را به تنظیم نمائید (شکل زیر).



سپس بر روی next کلیک کرده و در صفحه بعد گزینه tables را انتخاب کنید. در ادامه با کلیک بر روی دکمه next ، در صفحه‌ی بعدی می‌توان تمامی جداول و یا تنها جداول مورد نظر را جهت تهیه اسکریپت داده‌های آنها انتخاب نمود و در آخر عملیات تهیه اسکریپت insert داده‌ها به صورت خودکار صورت خواهد گرفت.
برای مشاهده تصاویر بیشتری از این عملیات ساده می‌توان به [این مقاله](#) رجوع نمود.

شایان ذکر است این قابلیت با نگارش‌های پائین‌تر اس کیوال سرور نیز کار می‌کند (برای مثال اتصال به اس کیوال سرور 2000 از طریق management studio 2008).

دیروز به من اطلاع دادند که در یکی از برنامه‌ها دو تا گروه "تاسیسات مکانیکی" پیدا شده!!
 تاسیسات مکانیکی
 تاسیسات مکانیکی

استاندارد این شرکت، استفاده از `kbdfa.dll` مخصوص و نسبتاً قدیمی است. بنابراین استاندارد مورد استفاده همان [ی و ک عربی](#) است. (کاری ندارم خوب است یا بد، یا باید اینطور باشد یا نه، بحث این است که فعل اینطور است و قرار نیست چیزی عوض بشود!) در مثال فوق، [ی و ک](#) عبارت دوم فارسی است. یعنی نصب `kbdfa.dll` روی ویندوز تازه نصب شده، فراموش شده بود.

راه حل‌ها:

(الف) قبل از ثبت، یکسان سازی صورت گیرد. یعنی اجرای متدهای شبیه به متدهای زیر بر روی هر ورودی متنی فارسی:

```
public string SafeFarsiStr(string input)
{
    return input.Replace("ک", "ك").Replace("ی", "ي");
}
```

ب) خوب، الان که این یکسان سازی صورت نگرفته چه باید کرد؟ اسکریپتی را تهیه کرده‌ام (مخصوص SQL Server 2005 به بعد) به صورت زیر که این تبدیل را برای شما انجام می‌دهد. به صورت خودکار تمامی فیلداتی متنی کلیه جداول دیتابیس جاری شما را یافته و [ی و ک](#) آن‌ها را یکسان می‌کند. البته همانطور که عرض شد، مطابق استاندارد این شرکت و استفاده از فایل `kbdfa.dll` قدیمی مورد استفاده، تمام [ی و ک](#) های فارسی به عربی تبدیل می‌شوند.

اسکریپتی برای یک دست سازی [ی و ک](#) در تمامی رکوردهای تمامی جداول دیتابیس جاری--
 اسکریپت زیری و [ک فارسی را به عربی تبدیل می‌کند--
 در صورت نیاز به حالت عکس، جای مقادیر عددی یونیکد را تعویض نمائید--](#)

```
USE TestDb;

DECLARE @Table NVARCHAR(MAX),
        @Col NVARCHAR(MAX)

DECLARE Table_Cursor CURSOR
FOR
    پیدا کردن تمام فیلداتی متنی تمام جداول دیتابیس جاری-
    SELECT a.name, --table
           b.name --col
    FROM   sysobjects a,
           syscolumns b
    WHERE  a.id = b.id
           AND a.xtype = 'u' --User table
           AND (
                  b	xtype = 99 --ntext
                  OR b	xtype = 35 -- text
                  OR b	xtype = 231 --nvarchar
                  OR b	xtype = 167 --varchar
                  OR b	xtype = 175 --char
                  OR b	xtype = 239 --nchar
            )
OPEN Table_Cursor FETCH NEXT FROM Table_Cursor INTO @Table,@Col
WHILE (@@FETCH_STATUS = 0)
BEGIN
```

```

EXEC (
    'update [' + @Table + '] set [' + @Col +
    ']= REPLACE(REPLACE(CAST([' + @Col +
    '] as nvarchar(max)) , NCHAR(1740), NCHAR(1610)),NCHAR(1705),NCHAR(1603)) '
)
FETCH NEXT FROM Table_Cursor INTO @Table,@Col
END CLOSE Table_Cursor DEALLOCATE Table_Cursor

```

توضیحات و نکاتی در مورد اسکریپت فوق:

- (الف) برای آشنایی با انواع XType Datatype مورد استفاده در کوئری فوق به [این آدرس](#) مراجعه نمائید.
- (ب) همانطور که در مطالب قبلی این وبلاگ نیز [ذکر شد](#)، امکان استفاده از تابع replace بر روی فیلدهای text و ntext وجود ندارد. هیچ اشکالی ندارد! تمام آنها به `cast max` دار شده و این مشکل به این صورت حل می‌شود.
- (ج) اس کیوال سرور اجازه تعریف یک جدول یا فیلد را به صورت متغیر بکار رفته در یک کوئری T-SQL نمی‌دهد. برای حل این موضوع باید عبارت SQL مورد نظر را به صورت یک رشته درآورد و سپس exec کرد.
- (د) مجبور شدم از معادلهای عددی برای دقت بیشتر کار استفاده کنم

The screenshot shows a SQL query window with the following code:

```

50
51 SELECT NCHAR (1740) AS N'ی فارسی',
52           NCHAR (1610) AS N'ی عربی',
53           NCHAR (1705) AS N'ک فارسی',
54           NCHAR (1603) AS N'ک عربی'

```

The results grid below shows the output:

	ی فارسی	ی عربی	ک فارسی	ک عربی
1	ی	ی	ک	ک

(در کل از تابع UNICODE اس کیوال سرور برای بدست آوردن این اعداد می‌توان استفاده کرد)

تذکر: این اسکریپت بر روی یک دیتابیس کاری تست شده و نتیجه رضایت بخش بوده؛ اما اگر شما روزی خواستید از آن استفاده کنید، حتما full backup را قبل از اجرای آن فراموش نکنید و پس از اجرا، تابع SafeFarsiStr فوق را برای ادامه کار حتما لحاظ نمائید یا از یک [kbedfa.dll](#) هماهنگ استفاده کنید.

نظرات خوانندگان

نویسنده: Alex
تاریخ: ۱۳۸۷/۱۰/۲۵ ۰۷:۵۲:۰۰

خیلی به درد بخور بود.

نویسنده: افشار
تاریخ: ۱۳۸۷/۱۰/۲۵ ۰۸:۱۵:۰۰

بعضی نسخه‌های kbdfa.dll بر اساس استاندارد ۲۹۰۱ هستند. استاندارد ۲۹۰۱ توسط استاندارد جدیدتر ۹۱۴۷ باطل اعلام گردید. البته اختلاف این دو تا استاندارد خیلی تولی چشم نمی‌زند. یک درایور خوب هم برای استاندارد ۹۱۴۷ وجود دارد که من در ویندوز ویستا و چندین ویندوز ایکس پی از آن استفاده می‌کنم بدون این که مشکلی برایم به وجود آمده باشد.

آدرس درایور: <http://prdownloads.sourceforge.net/farsitools/persiankeyboard.zip?download>

مطالعه استاندارد: <http://fa.farsiweb.ir/mediawiki-fa/images/a/a9/Isiri-9147.pdf>

لینک استاندارد در سایت سازمان استاندارد: <http://www.isiri.org/asp/account/checklog.asp?ID=9147.pdf>

باید از زبان فارسی حمایت کنیم.

نویسنده: hajloo
تاریخ: ۱۳۸۷/۱۰/۲۶ ۱۴:۳۶:۰۰

مطلوب خوبی بود اما من هم با افشار موافقم که باید از فارسی حمایت کرد و بهتره که بجای جایگزینی حروف فارسی با عربی، بهتره که از فونتهای استاندارد استفاده بشه و حروف عربی به فارسی تبدیل بشن - از مطلب خوبتون هم واقعاً ممنونم

نویسنده: وحید نصیری
تاریخ: ۱۳۸۷/۱۰/۲۶ ۱۶:۰۴:۰۰

اگر این مورد به نحوی در خود برنامه‌های پایگاه داده حل بشود، این مسایل دیگر وجود نخواهد داشت. مثلاً اس کیوال سروری وی را به یک صورت پردازش کند و جدا از هم درنظر نگیرد. این مورد الان در جستجوی گوگل هم مشهود است و مشکل زا. شما با یک سری جواب می‌گیرید و با یک سری دیگر. این مورد مهم است!

نویسنده: معنوی
تاریخ: ۱۳۸۷/۱۱/۰۷ ۱۶:۵۸:۰۰

تو برنامه تون (دادت نتی) با استفاده از MessageFilter می‌تونید کلیه ورودی‌ها رو تحت کنترل داشته باشید.

نویسنده: as
تاریخ: ۱۳۸۹/۰۵/۲۷ ۰۰:۱۷:۰۹

مرسى از کمکت عالی بود...

نویسنده: msj

تاریخ: ۱۴:۵۳:۳۶ ۱۳۸۹/۰۷/۲۹

با سلام و تشکر از مطالب مفیدتون.

استاندارد سیستم من فارسی هست و میخواه همون هم ذخیره بشه.

من این اسکریپت رو اجرا کردم (البته با تغییر در replace و جایه جا کردن فارسی و عربی) و دیتابیس رو درست کرد.

فقط الان یه مشکلی دارم (البته فکر می کنم قبل از این اسکریپت هم بوده): هر insert که می زنم ، خود SQL "ی" فارسی رو به "ی" عربی تبدیل می کنه و ذخیره می کنه. یعنی من 1740 می فرستم ولی 1610 ذخیره میشه. این مشکل در مورد ک وجود نداره و همون 1705 فرستاده و ذخیره میشه.

من از Collation Arabic_CI_AS و SQL 2005 express استفاده می کنم.

نویسنده: وحید نصیری

تاریخ: ۱۶:۵۲:۱۲ ۱۳۸۹/۰۷/۲۹

مشکل از SQL Server نیست. مشکل از درایور صفحه کلید شما است که به نظر ایراد دارد. باید ورودی کدهای خودتون رو در تمام قسمت‌هایی که insert دارد بررسی کنید (قبل از ورود به دیتابیس این تصحیح باید انجام شود مانند استفاده ازتابع SafeFarsiStr فوق).

نویسنده: وحید نصیری

تاریخ: ۱۷:۰۶:۰۲ ۱۳۸۹/۰۷/۲۹

اثبات این مدعاهم که مشکل از SQL Server نیست ساده است (فیلد با collation عربی ایجاد شده و ی فارسی در آن ثبت شده است و سپس گزارشگیری):

```
DECLARE @tbl AS TABLE (f1 NVARCHAR(50) COLLATE Arabic_CI_AS NULL
((INSERT INTO @tbl(f1) VALUES(NCHAR(1740
SELECT * FROM @tbl
```

نویسنده: سامانی

تاریخ: ۱۱:۲۵ ۱۳۹۱/۰۴/۲۷

واقعاً عالی بود

در این مقاله قصد داریم نحوه مونیتور کردن میزان مصرف CPU توسط اس کیوال سرور را بررسی کنیم. برای بدست آوردن میزان CPU مصرفی اس کیوال سرور می‌توان به اسکریپت زیر رجوع کرد:

```
DECLARE @CPU_BUSY int, @IDLE int
SELECT @CPU_BUSY = @@CPU_BUSY, @IDLE = @@IDLE WAITFOR DELAY '000:00:01'
SELECT (@@CPU_BUSY - @CPU_BUSY)/((@@IDLE - @IDLE + @@CPU_BUSY - @CPU_BUSY) *1.00) *100 AS 'CPU
Utilization by sqlsvr.exe'
```

ماخذ

در ادامه قصد داریم، هر 5 دقیقه به صورت خودکار بررسی کنیم که آیا میزان مصرف CPU در اس کیوال سرور بالای 50 درصد است؟ و اگر بله، ایمیلی را به مسؤول مربوطه جهت بررسی ارسال کنیم.

بنابراین اولین کاری که باید صورت گیرد، فعال سازی Database Mail در اس کیوال سرور است که به صورت پیش فرض غیرفعال است. برای این منظور تنها کافی است اسکریپت زیر را بر روی سرور اجرا کنید:

```
USE [master]
GO
sp_configure 'show advanced options',1
GO
RECONFIGURE WITH OVERRIDE
GO
sp_configure 'Database Mail XPs',1
GO
--RECONFIGURE
GO
-- Create a New Mail Profile for Notifications
EXECUTE msdb.dbo.sysmail_add_profile_sp
    @profile_name = 'DBA_Notifications',
    @description = 'Profile for sending Automated DBA Notifications'
GO
-- Set the New Profile as the Default
EXECUTE msdb.dbo.sysmail_add_principalprofile_sp
    @profile_name = 'DBA_Notifications',
    @principal_name = 'public',
    @is_default = 1 ;
GO
-- Create an Account for the Notifications
EXECUTE msdb.dbo.sysmail_add_account_sp
    @account_name = 'SQLMonitor',
    @description = 'Account for Automated DBA Notifications',
    @email_address = 'nasiri@site.net', -- Change This
    @display_name = 'SQL Monitor',
    @mailserver_name = 'mail.site.net' -- Change This
GO
-- Add the Account to the Profile
EXECUTE msdb.dbo.sysmail_add_profileaccount_sp
    @profile_name = 'DBA_Notifications',
    @account_name = 'SQLMonitor',
    @sequence_number = 1
```

ماخذ

این اسکریپت برای اس کیوال سرورهای 2005 به بعد طراحی شده و تنها دو سطر آنرا پیش از اجرا باید ویرایش کنید. سطر مربوط به `mailserver_name` و `email_address`. آدرس ایمیل در حقیقت آدرس ایمیل قسمت `from` پیغام ارسالی را تشکیل می‌دهد. نام سرور میل هم، منظور آدرس `smtp server` شما در شبکه است.

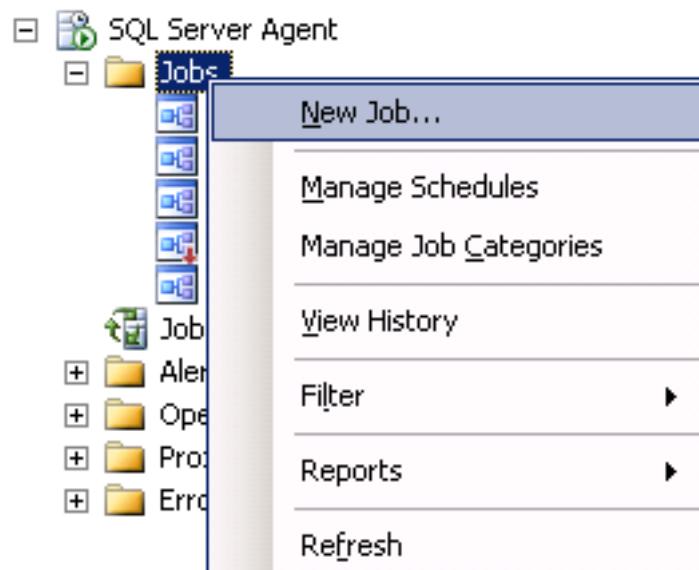
یا اگر علاقمند بودید که این کار را توسط ویژاردهای `management studio` انجام دهید (که در نهایت هیچ تفاوتی با اسکریپت فوق نخواهد داشت)، می‌توان به [این مقاله](#) رجوع کرد.

پس از اجرای اسکریپت فوق، برای بررسی صحت عملکرد فوق می‌توان دستور زیر را اجرا کرد:

```
--test  
EXECUTE msdb.dbo.sp_send_dbmail  
    @recipients = 'nasiri@site.net', -- Change This  
    @Subject = 'Test Message generated from SQL Server DatabaseMail',  
    @Body = 'This is a test message from SQL Server DatabaseMail'
```

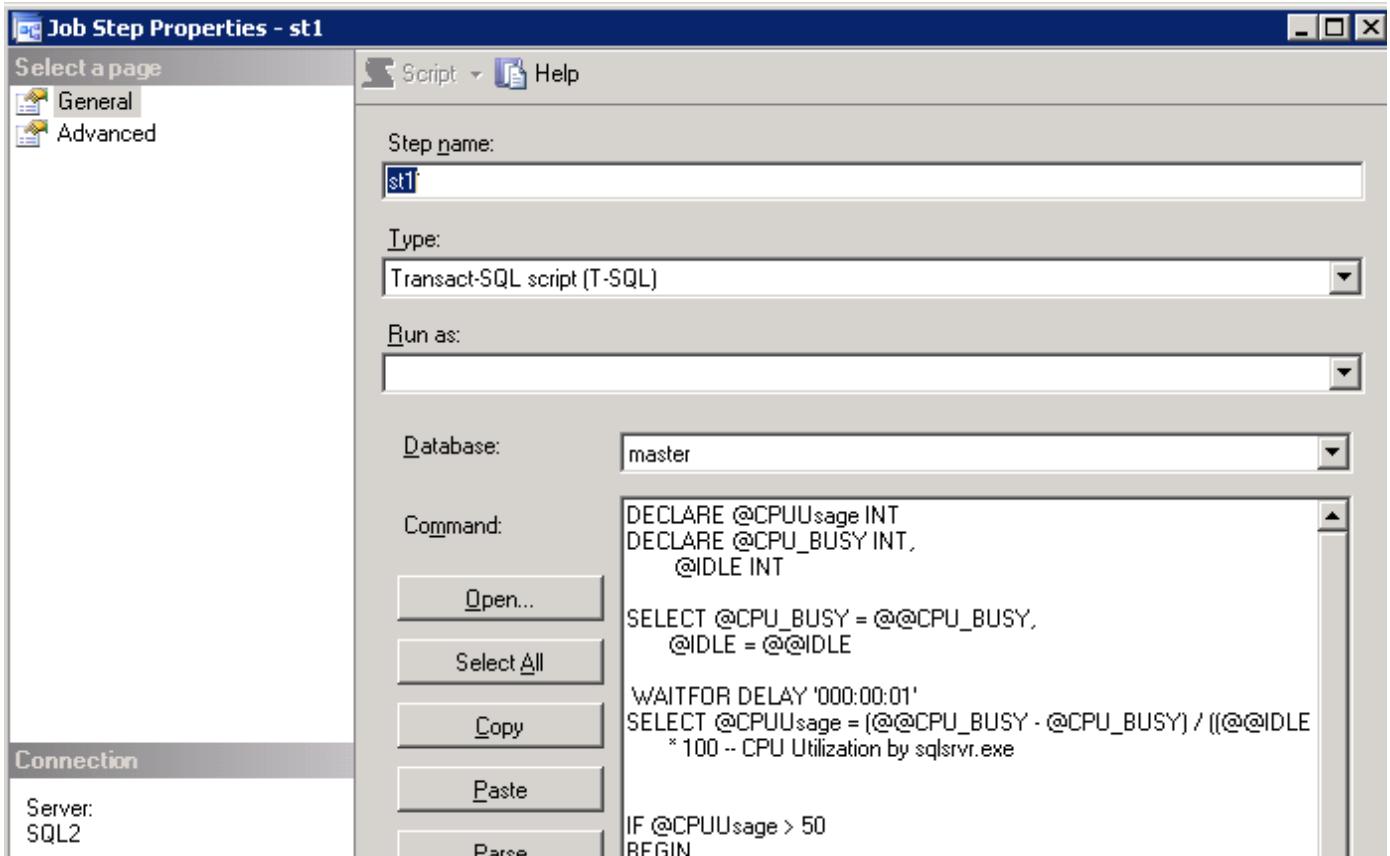
سایر پارامترهای این دستور را در [MSDN](#) می‌توان ملاحظه نمود.

تا اینجا اس کیوال سرور برای ارسال ایمیل آماده شد. در ادامه قصد داریم یک job جدید در اس کیوال سرور ایجاد کنیم تا تمام موارد فوق را لحاظ کند.



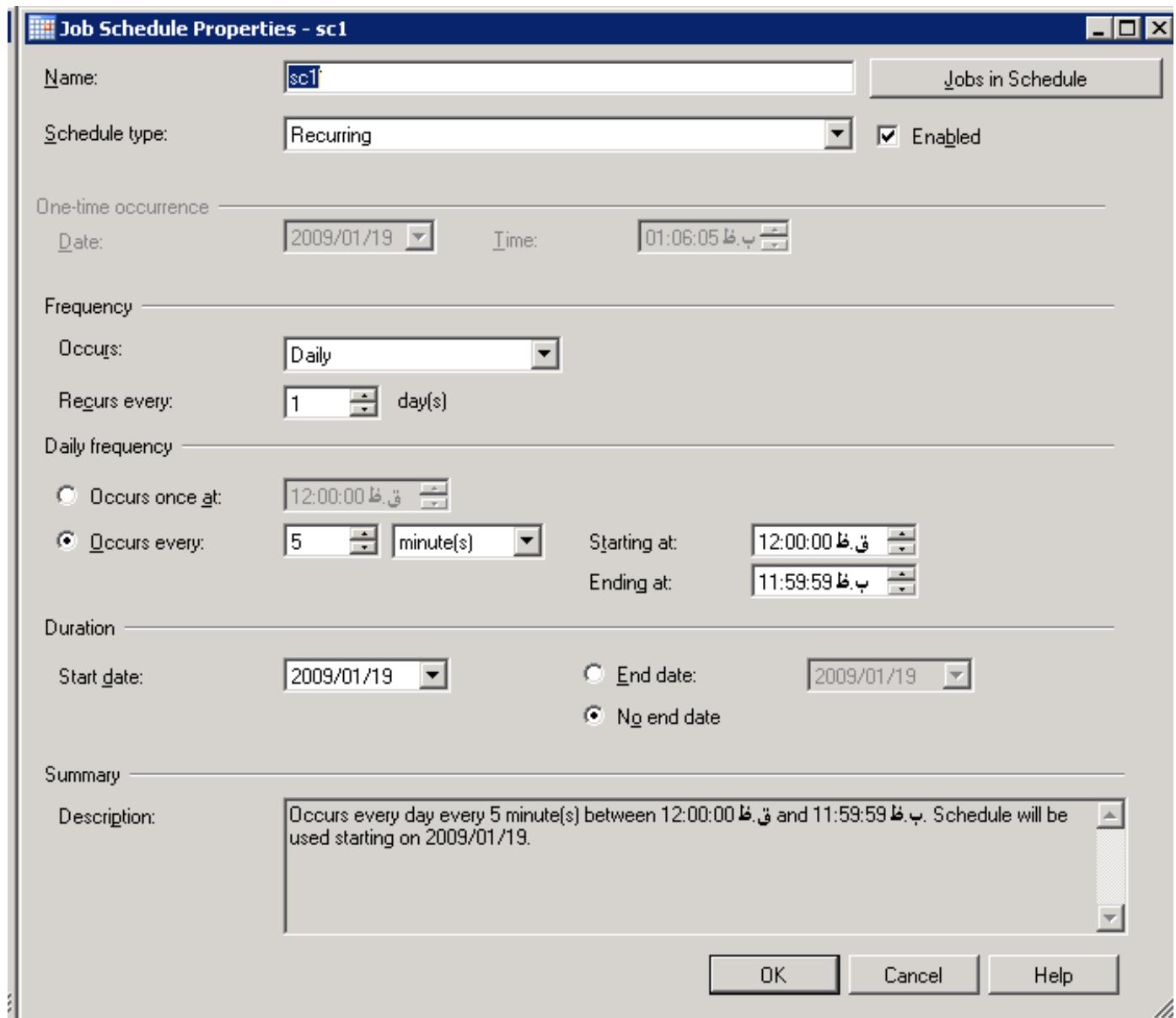
مطابق تصویر فوق ابتدا یک job جدید را آغاز خواهیم کرد.

مونیتور کردن میزان مصرف CPU در اس کیوال سرور



در ادامه اسکریپت زیر را جهت اجرا به آن معرفی می‌کنیم. توسط این اسکریپت، میزان جاری مصرف CPU اس کیوال سرور محاسبه شده و اگر این میزان بیشتر از 50 بود، یک ایمیل به مسؤول مربوطه با ذکر میزان CPU usage ارسال می‌گردد.

```
DECLARE @CPUUsage INT  
DECLARE @CPU_BUSY INT,  
    @IDLE INT  
  
SELECT @CPU_BUSY = @@CPU_BUSY,  
    @IDLE = @@IDLE  
  
WAITFOR DELAY '000:00:01'  
SELECT @CPUUsage = (@@CPU_BUSY - @CPU_BUSY) / ((@@IDLE - @IDLE + @@CPU_BUSY - @CPU_BUSY) * 1.00)  
* 100 -- CPU Utilization by sqlsvr.exe  
  
IF @CPUUsage > 50  
BEGIN  
    DECLARE @msg NVARCHAR(1000)  
    SET @msg = 'Please check SQL server, CPU usage is ' + CAST(@CPUUsage AS NVARCHAR(50))  
    + '%.'  
  
    EXECUTE msdb.dbo.sp_send_dbmail  
    @recipients = 'nasiri@site.net', -- Change This  
    @copy_recipients = 'nasiri@site.net', -- Change This  
    @Subject = 'CPU overload',  
    @Body = @msg  
    ,@importance = 'High'  
END
```



و در آخر زمان اجرای آن را به هر روز، هر 5 دقیقه یکبار تنظیم خواهیم کرد.

اگر نیاز به راه حلی پخته‌تر و بررسی متوسط چندین مقدار قبلی، مقایسه آن‌ها و سپس ارسال ایمیل داشتید، می‌توان به فصل 14 کتاب [Super SQL Server Systems](#) مراجعه کرد.

به صورت پیش فرض تنظیمات حافظه‌ی اس کیوال سرور به صورتی است که به آن اجازه می‌دهد تمامی حافظه‌ی مهیای سیستم عامل را مصرف کند! اگر شخصی با این مساله آشنا نباشد احتمالاً تصور خواهد کرد که اس کیوال سرور نشتی حافظه دارد یا کلا مشکلی در سیستم روی داده است که تا این حد مصرف حافظه بالا رفته است.
 برای مشاهده‌ی این تنظیمات روی instance سرور مورد نظر کلیک راست کرده و گزینه خواص را انتخاب کنید. سپس در صفحه‌ی باز شده گزینه memory را مشاهده نمایید.
 البته این کار را با استفاده از دستورات T-SQL هم می‌توان انجام داد:

```

EXEC sp_configure 'show advanced options', 1
RECONFIGURE WITH OVERRIDE

EXEC sp_configure 'max server memory (MB)'

EXEC sp_configure 'show advanced options', 0
RECONFIGURE WITH OVERRIDE
  
```

نتیجه‌ی آن بر روی سیستم برنامه نویسی من به صورت زیر است:

	name	minimum	maximum	config_value	run_value
1	max server memory (MB)	16	2147483647	1024	1024

که البته عمداً آن را بر روی یک گیگ تنظیم کرده‌ام تا عملکرد سیستم مختل نشود. (چون اگر غیر از این باشد، تعارفی نداشته و هر آنچه را که صلاح بداند مصرف می‌کند!)
 کمی در بلاگ‌های رسمی برنامه نویس‌های اس کیوال سرور در سایت MSDN در این‌باره جستجو کردم و نتیجه‌ی آن به صورت زیر است:

تنظیمات حداکثر حافظه‌ی مصرفی اس کیوال سرورهای 2005 و 2008 برای یک سیستم عامل 64 بیتی:

MaxMem Setting	Physical RAM
1500	2GB
3200	4GB
4800	6GB
6700	8GB
10600	12GB
14500	16GB
22400	24GB
30000	32GB
45000	48GB
59000	64GB

برای سروری 32 بیتی این اعداد حداکثر را تقریباً منهای 300 مگابایت نمائید. (و البته همانطور که مطلع هستید در سیستم عامل‌های سرور 32 بیتی، اگر بیشتر از 2 گیگابایت رم داشتید سوئیچ 3GB و اگر بیشتر از 4 گیگابایت رم مهیا بود، سوئیچ PAE باید در boot.ini تنظیم شوند. در غیر اینصورت هزینه‌ی سخت افزاری بیهوده‌ای را متحمل شده‌اید، زیرا استفاده‌ی بهینه‌ای از آن در یک سیستم عامل 32 بیتی نخواهد شد)

و این اعداد را همانطور که ذکر شد، در تنظیمات سرور می‌توان وارد نمود (از طریق management studio) و یا با استفاده از دستورات T-SQL نیز این امر میسر است: (در مثال زیر حداکثر حافظه مجاز مصرفی به 2300 مگابایت تنظیم می‌شود)

```
EXEC sp_configure 'show advanced options', 1
RECONFIGURE WITH OVERRIDE

EXEC sp_configure 'max server memory (MB)', 2300

EXEC sp_configure 'show advanced options', 0
RECONFIGURE WITH OVERRIDE
```

و البته لازم به ذکر است که اعداد فوق برای حالتی پیشنهاد شده است که سرور مورد نظر فقط یک کار و آن‌هم اجرا و سرویس دهی اس کیوال سرور را به عهده داشته باشد. بدیهی است اگر از این سرور استفاده‌های دیگری هم می‌شود این اعداد را باید کمتر نیز در نظر گرفت.

برای مطالعه‌ی بیشتر

<http://sqlnerd.blogspot.com/2006/07/memory-use-in-sql-server.html>

<http://blogs.msdn.com/axperf/archive/2008/03/10/welcome-database-configuration-checklist-part-1.aspx>

<http://technet.microsoft.com/en-us/library/bb124810.aspx>

نظرات خوانندگان

نوبنده: Meysam
تاریخ: ۱۳۸۹/۰۷/۲۲ ۰۳:۰۷:۲۲

اگر این سرور وظیفه‌ی وب سروری رو هم داشته باشه(IIS) چه درصدی رو پیشنهاد میکنید?
البته DNS هم هست طبیعتا!!

نوبنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۷/۲۲ ۰۵:۳۱

برای IIS حدود حداقل یک گیگ و اگر تعداد کاربر زیاد است تا یک و نیم گیگ را کنار بگذارید. DNS Server زیاد حافظه مصرف نمی‌کند. یا حداقل به اندازه‌ی IIS نیست. تا 300 مگ برای آن زیاد هم هست.
بنابراین اگر مثلا سرور شما 4 گیگ رم دارد و IIS و DNS و SQL Server و غیره را با هم دارد، حداکثر حافظه مصرفی SQL Server را به حدود 2.3 گیگ محدود کنید و گرنه خیلی زود صدای کاربران سایت(ها) در خواهد آمد که باز هم کند شد ... باز هم کند شد

...

یک سرویس کوچک ویندوز ان اتی نوشته‌ام که کارش این است که در پایان هر هفته، تمام دیتابیس‌های اس کیوال سرور موجود را یافته و اسکریپت تمام اشیاء آن‌ها را به صورت خودکار تولید می‌کند (از جداول گرفته تا تریگرها، رویه‌های ذخیره شده و غیره)، سپس کل مجموعه را فشرده کرده و سپس ایمیل می‌زند. این کار برای نگهداری تغییرات انجام شده در طول یک هفته لازم است. برنامه با استفاده از امکانات SMO تهیه شده است و اگر علاقمند بودید که این کار را انجام دهید، می‌توانید به مقاله‌های زیر رجوع کنید:

[Making a database clone using SMO](#)

[Using the SqlServer.Management.Smo](#)

[SQL Server: SMO Scripting Basics](#)

با آمدن اس کیوال سرور 2008، اشیاء SMO هم به روز شده‌اند و اگر با این اشیاء برنامه نویسی کرده باشد، برنامه بر روی سروری با اس کیوال سرور 2005 اجرا نخواهد شد و پیغام خطای زیر را دریافت خواهد کرد:

```
Could not load file or assembly 'Microsoft.SqlServer.Management.Sdk.Sfc, Version=10.0.0.0, Culture=neutral, PublicKeyToken=89845dc8080cc91' or one of its dependencies. The system cannot find the file specified.
```

خوبی‌خانه مایکروسافت این کتابخانه‌ها را به صورت مجزا هم برای دریافت قرار داده است و می‌توان آن‌ها را نصب نمود تا برنامه بدون اشکال اجرا شود. به صفحه زیر و قسمت Microsoft SQL Server 2008 Management Objects مراجعه نمایید:
[اینجا کلیک نمایید](#)

البته همانطور که در صفحه ذکر شده نیز عنوان گردیده است، به 6.0 MSXML هم نیاز می‌باشد که لینک دریافت آن در ابتدای صفحه فوق موجود است.

با توجه به در حال اجرا بودن 24 ساعته‌ی سرویس SQL server agent، استفاده‌های ارزنده‌ای از آن می‌توان کرد. برای مثال هر از گاهی بررسی کند که آیا هارد سرور پر شده یا نه؟ و اگر بله (کمبود میزان فضای خالی به حد خطرناکی رسیده)، یک ایمیل خودکار به مسؤول مربوطه ارسال کند.

عمده‌ی مطالبی که در این مقاله بررسی خواهند شد همانند مطلب [مونیتور کردن میزان مصرف CPU توسط اس کیوال سرور](#) است و از تکرار آن‌ها در اینجا صرفنظر خواهد شد (راه اندازی دیتابیس میل و همچنین تعریف یک job جدید که در مورد آن‌ها صحبت شد، همانند قبل است). تنها مطلب جدیدی که به آن اشاره خواهد شد، اسکریپت بررسی میزان فضای خالی و سپس ارسال ایمیل است که در یک job جدید همانند مقاله‌ی قبل باید به سرور اضافه شود. این اسکریپت به شرح زیر است:

```

DECLARE @DriveBenchmark INT
DECLARE @MachineName NVARCHAR(1000)
DECLARE @DiskFreeSpace INT
DECLARE @DriveLetter CHAR(1)
DECLARE @AlertMessage NVARCHAR(MAX)
DECLARE @MailSubject NVARCHAR(MAX)
DECLARE @NewLine CHAR(2)

SET @NewLine = CHAR(13) + CHAR(10)

SET @DriveBenchmark = 2048 -- 2GB
SET @MailSubject = 'Free space is low on ' + @@SERVERNAME
SET @AlertMessage = ''

IF EXISTS (
    SELECT *
    FROM tempdb..sysobjects
    WHERE id = OBJECT_ID(N'[tempdb]..[#disk_free_space]')
)
    DROP TABLE #disk_free_space

CREATE TABLE #disk_free_space
(
    DriveLetter CHAR(1) NOT NULL,
    FreeMB      INTEGER NOT NULL
)

/* Populate #disk_free_space with data */
INSERT INTO #disk_free_space
EXEC MASTER..xp_fixeddrives

DECLARE DriveSpace CURSOR FAST_FORWARD
FOR
    SELECT DriveLetter,
           FreeMB
    FROM #disk_free_space

OPEN DriveSpace
FETCH NEXT FROM DriveSpace INTO @DriveLetter, @DiskFreeSpace

WHILE (@@FETCH_STATUS = 0)
BEGIN
    IF @DiskFreeSpace < @DriveBenchmark
    BEGIN
        SET @AlertMessage = @AlertMessage + 'Drive ' + @DriveLetter + ' on ' + @@SERVERNAME
        + ' has only ' + CAST(@DiskFreeSpace AS VARCHAR) + ' MB left.' + @NewLine
    END

    FETCH NEXT FROM DriveSpace INTO @DriveLetter, @DiskFreeSpace
END
CLOSE DriveSpace
DEALLOCATE DriveSpace

DROP TABLE #disk_free_space

```

```
IF @AlertMessage <> ''  
BEGIN  
EXECUTE msdb.dbo.sp_send_dbmail  
@recipients = 'nasiri@site.net', -- Change This  
@copy_recipients = 'Administrator@site.net', -- Change This  
@Subject = @MailSubject,  
@Body = @AlertMessage  
,@importance = 'High'  
END
```

بررسی اسکریپت فوق:

همه چیز از رویه سیستمی [xp_fixeddrives](#) شروع می‌شود. حاصل اجرای این رویه، دریافت میزان فضای خالی هر درایو موجود در سرور خواهد بود. همانطور که در اسکریپت نیز مشخص است، برای ذخیره سازی خروجی این رویه، یک جدول موقتی (disk_free_space) ایجاد شده و خروجی آن به درون این جدول اضافه خواهد شد. سپس یک cursor ایجاد شده و تک تک رکوردهای حاصل با مقدار متغیر DriveBenchmark که در اینجا 2 گیگابایت در نظر گرفته شده است، مقایسه می‌گردد. سپس هر کدام از رکوردها که کمتر از 2 گیگابایت بود، متغیر AlertMessage را مقدار دهی خواهد کرد. در پایان اگر این متغیر مقدار دهی شده بود، یعنی مشکل حاصل شده و نتیجه‌ی بررسی به صورت یک ایمیل ارسال می‌گردد. بدیهی است که در صورت نیاز مقدار متغیر DriveBenchmark و آرگومان‌های recipients و copy_recipients کد فوق باید اصلاح شوند.

برای استفاده از آن یک job جدید تعریف کنید که مثلا هر سه ساعت یکبار اجرا شده و این اسکریپت را فراخوانی نماید.

به لطف امکانات سیستمی اس کیوال سرورهای 2005 به بعد و [DMV](#) های آن‌ها، آمارگیری از ریز اتفاقات رخ داده در یک اس کیوال سرور این روزها بسیار ساده شده است و نیازی به ابزارهای جانبی برای انجام این نوع عملیات نیست (یا کمتر هست). در ادامه مرواری خواهیم داشت بر یک سری کوئری که اطلاعات جالبی را در مورد وضعیت رویه‌های ذخیره شده‌ی دیتابیس‌های دیتابیس‌های شما ارائه می‌دهند. لازم به ذکر است که اکثر این آمارها با هر بار ری استارت سرور، صفر خواهند شد.

آیا می‌دانید در یک دیتابیس خاص کدامیک از رویه‌های ذخیره شده‌ی شما بیش از سایرین مورد استفاده بود و آماری از این دست؟

```

use dbName;
SELECT TOP(100) qt.text AS 'SP Name',
    qs.execution_count AS 'Execution Count',
    qs.execution_count / DATEDIFF(Second, qs.creation_time, GETDATE()) AS 'Calls/Second',
    qs.total_worker_time / qs.execution_count AS 'AvgWorkerTime',
    qs.total_worker_time AS 'TotalWorkerTime',
    qs.total_elapsed_time / qs.execution_count AS 'AvgElapsedTime',
    qs.max_logical_reads,
    qs.max_logical_writes,
    qs.total_physical_reads,
    DATEDIFF(Minute, qs.creation_time, GETDATE()) AS 'Age in Cache'
FROM sys.dm_exec_query_stats AS qs
CROSS APPLY sys.dm_exec_sql_text(qs.sql_handle) AS qt
WHERE qt.dbid = DB_ID() -- Filter by current database
ORDER BY
    qs.execution_count DESC

```

البته مرتب سازی پیش فرض این کوئری بر اساس تعداد بار اجرا است (رویه‌های ذخیره شده‌ی محبوب!), می‌شود آن را بر اساس total_physical_reads (فشار بر روی CPU سیستم)، total_logical_reads (فشار بر روی حافظه)، total_worker_time (فشار I/O کوئری‌ها)، total_logical_writes نیز مرتب کرد و نتایج جالب توجهی را بدست آورد.

آیا می‌دانید کدامیک از رویه‌های ذخیره شده‌ی شما بیش از سایرین کامپایل مجدد شده است؟

```

select top 50
    sql_text.text,
    sql_handle,
    plan_generation_num,
    execution_count,
    dbid,
    objectid
from
    sys.dm_exec_query_stats a
cross apply sys.dm_exec_sql_text(sql_handle) as sql_text
where
    plan_generation_num >1
order by plan_generation_num desc

```

آیا می‌دانید آخرین باری که رویه‌های ذخیره شده‌ی شما ویرایش شده‌اند چه زمانی بوده است؟

```

SELECT NAME,
    create_date,
    modify_date
FROM sys.objects

```

آیا از وضعیت رویه‌های ذخیره شده‌ی دیتابیس‌های اس کیوال سرور خود خبر دارید؟

```
WHERE  TYPE = 'P'  
ORDER BY  
    Modify_Date DESC,  
    NAME
```

آیا از وضعیت رویه‌های ذخیره شده‌ی دیتابیس‌های اس کیوال سرور خود خبر دارید؟

نظرات خوانندگان

نوبنده: مهرداد قاسمی
تاریخ: ۱۳۸۷/۱۲/۰۷ ۲۱:۱۵:۰۰

بسیار جالب و مفید ...
ممنون استاد

سؤال: چگونه می‌توان یک کوئری را تعیین اعتبار کرد بدون اینکه نتیجه‌ی اجرای آن برای ما اهمیتی داشته باشد یا دیتابیس را به هم بریزد؟

روش انجام این کار در اس کیوال سرور به صورت زیر است:

```
SET NOEXEC ON
GO
عبارات اس کیوال ما در اینجا قرار خواهد گرفت
GO
SET NOEXEC OFF
GO
```

مطابق راهنمای اس کیوال سرور، [SET NOEXEC](#) سبب کامپایل کوئری‌های پس از آن شده ولی آن‌هارا اجرا نخواهد کرد. همچنین فقط syntax [SET PARSEONLY ON](#) یک کوئری را بررسی کرده و خطاهای حاصل را نمایش می‌دهد (بدون اجرا کوئری، بدون کامپایل آن).

نظرات خوانندگان

نویسنده: Alex's Blog
تاریخ: ۱۳۸۷/۱۲/۱۲ ۰۷:۳۷:۰۰

این مورد خیلی بدردبهوره. ممنون

مطلوبی را روز قبل نوشتم در مورد تعیین اعتبار یک کوئری. این مورد از آنجایی حائز اهمیت می‌شود که برای مثال [تغییری در ساختار](#) یکی از جداول حاصل شود. اکنون می‌خواهیم بررسی کنیم آیا سیستم از کار افتاده یا نه؟
شما می‌توانید نام یک فیلد را تغییر دهید (حتی اگر این فیلد در یک رویه ذخیره شده استفاده شده باشد) و هیچ خطایی هم خواهید گرفت و این منشاء در دسرهای زیادی خواهد بود.
در حالت استفاده از `SET NOEXEC ON`، کوئری مورد نظر فقط کامپایل می‌شود و همچنین از لحاظ نحوی بررسی خواهد شد، اما این کافی نیست.
مثال زیر را در نظر بگیرید:

```
Create PROCEDURE Test1
AS
SELECT * FROM tblPIIDs1
```

جدول `tblPIIDs1` در دیتابیس مورد نظر وجود ندارد.
این کوئری قابل اجرا است. دکمه‌ی F5 را فشار دهید، بلافضله رویه ذخیره شده‌ی `Test1` برای شما ایجاد خواهد شد.
سپس کوئری زیر را اجرا کنید:

```
USE testdb
SET NOEXEC ON;
exec test1 ;
SET NOEXEC OFF;
```

بدون مشکل و بروز خطایی، پیغام زیر را نشان می‌دهد:

```
Command(s) completed successfully
```

ایرادی هم وارد نیست چون فقط عملیات `compile` و `parsing` صورت گرفته و نه اجرای واقعی رویه ذخیره شده. اینجا از لحاظ دستوری مشکلی وجود ندارد.

در این نوع موارد می‌توان از [SET FMTONLY ON](#) استفاده کرد. این مورد اجرای غیر واقعی یک کوئری را سبب می‌شود (تا ثیری روی دیتابیس موجود نخواهد داشت، برای مثال اگر در رویه ذخیره شما عبارت `insert` وجود داشت، دیتابیس `insert` خواهد شد) و تنها متادیتای حاصل را بازگشت می‌دهد. مثلاً نام ستون‌های یک کوئری را و همچنین در این حین اگر خطایی رخداده باشد، آن را نیز ارائه خواهد داد.

```
USE testdb
SET FMTONLY ON;
exec test1 ;
SET FMTONLY OFF;
```

با اجرای کوئری فوق خطای زیر ظاهر می‌شود:

```
Msg 208, Level 16, State 1, Procedure test1, Line 3
Invalid object name 'tblPIIDs1'.
```

برای اتوМАСИОН این توانایی می‌توان از کوئری زیر استفاده کرد:

```

USE testdb;
SET NOCOUNT ON;

DECLARE @name NVARCHAR(MAX),
        @sql NVARCHAR(MAX),
        @type CHAR(2), -- object type
        @type_desc NVARCHAR(60), -- object type description
        @params NVARCHAR(MAX) -- parameters

DECLARE @tblInvalid TABLE (
    -- invalid objects
    [type_desc] NVARCHAR(60),
    [name] NVARCHAR(MAX),
    [error_number] INT,
    [error_message] NVARCHAR(MAX),
    [type] CHAR(2)
);

DECLARE testSPs CURSOR FAST_FORWARD
FOR
    SELECT [name] = OBJECT_NAME(SM.[object_id]),
           [type] = SO.[type],
           SO.[type_desc],
           [params] =
    (
        SELECT (
            SELECT CONVERT(
                XML,
                (
                    SELECT STUFF(
                        (
                            SELECT ', ' + [name] +
                                '=NULL' AS
                                [text()]
                            FROM sys.parameters
                            WHERE [object_id] = SM.[object_id]
                            FOR XML PATH('')
                        ),
                        1,
                        1,
                        ''
                    )
                )
            )
        )
    )
    FOR XML RAW,
    TYPE
).value('/row[1]', 'varchar(max)')
FROM sys.sql_modules SM
JOIN sys.objects SO
    ON SO.[object_id] = SM.[object_id]
WHERE SO.[is_ms_shipped] = 0
    AND SO.[type] = 'P'

OPEN testSPs
FETCH NEXT FROM testSPs INTO @name, @type, @type_desc, @params

WHILE (@@FETCH_STATUS = 0)
BEGIN
    BEGIN TRY
        SET @sql = 'SET FMTONLY ON; exec ' + @name + ' ' + @params +
        '; SET FMTONLY OFF;';
        --PRINT @sql;
        EXEC (@sql);
    END TRY
    BEGIN CATCH
        PRINT @type_desc + ', ' + @name + ', Error: ' + CAST(ERROR_NUMBER() AS VARCHAR)
        + ', ' + ERROR_MESSAGE();
        INSERT INTO @tblInvalid
        SELECT @type_desc,
               @name,
               ERROR_NUMBER(),
               ERROR_MESSAGE(),
               @type
    END CATCH
END
;
```

```

    FETCH NEXT FROM testSPs INTO @name, @type, @type_desc, @params
END
CLOSE testSPs
DEALLOCATE testSPs

SELECT [type_desc],
       [name],
       [error_number],
       [error_message]
FROM   @tblInvalid
ORDER  BY
       CHARINDEX([type], ' U V PK UQ F TR FN TF P SQ '),
       [name];

```

توضیحات:

این کوئری، در دیتابیس جاری که در قسمت `use dbname` مشخص می‌شود، تمامی رویه‌های ذخیره شده را به صورت خودکار پیدا می‌کند. سپس لیست آرگومان‌های آن‌ها را نیز یافته و عبارت `exec` مربوطه را تشکیل می‌دهد. سپس با استفاده از `SET FMTONLY ON` سعی در شبیه سازی اجرای تک تک رویه‌های ذخیره شده می‌کند. اگر خطای در این بین رخ داد، آن‌ها را در یک جدول موقتی ذخیره کرده و در آخر نتیجه را نمایش می‌دهد.

ارزش این کوئری زمانی مشخص می‌شود که تعداد زیادی رویه ذخیره شده داشته باشد اما نمی‌دانید کدامیک از آن‌ها بر اساس آخرین تغییرات صورت گرفته، هنوز معتبر هستند یا نه. آیا به قول معروف، سیستم اوMD پایین یا خیر؟

نکته:

قسمتی که از XML استفاده شده جهت [concatenating](#) نتیجه حاصل از کوئری، مورد استفاده قرار گرفته و این روزها [بحث رایج](#) است که در بسیاری از سایتها در مورد آن می‌توان مطالب مفیدی را یافت. راه دیگر انجام آن استفاده از [COALESCE](#) می‌باشد.

ماخذ:

[Check Validity of SQL Server Stored Procedures](#)
[Which of your Stored Procedures are no longer Valid](#)
[SET FMTONLY ON](#)

نظرات خوانندگان

نویسنده: Alex's Blog

تاریخ: ۱۴:۱۲:۰۰ ۱۳۸۷/۱۲/۱۲

در مورد مطلب دیروزتون که فکر می کردم دنبال موردی بودم که امروز بهش اشاره کردین منظورم SET FMTONLY ON بود. این مورد بیشتر بکار میاد البته بنظر من. ممنون

پس از مدتی که از شروع به کار یک سیستم می‌گذرد، همانطور که تعریف ایندکس‌های مفید سرعت جستجوها را بالا می‌برد، ایجاد fragmentation در آن‌ها نیز تاثیر منفی در کارآیی خواهد داشت. به همین منظور نیاز است هر از چندگاهی بررسی شود میزان DBCC INDEXDEFRAG ایندکس‌ها چقدر است. اگر این میزان بیش از 30 درصد بود توصیه شده است که از دستور استفاده شود یا بازسازی مجدد (rebuild) ایندکس‌ها صورت گیرد.

یکی دیگر از امکانات dmv‌های اس‌کیوال سرورهای 2005 به بعد، ارائه آمار میزان fragmentation ایندکس‌ها است که کوئری آن به صورت زیر می‌تواند باشد:

```

USE dbName;
SELECT OBJECT_NAME(DMV.object_id) AS TABLE_NAME,
       SI.NAME AS INDEX_NAME,
       avg_fragmentation_in_percent AS FRAGMENT_PERCENT,
       DMV.record_count
  FROM sys.dm_db_index_physical_stats(DB_ID(), NULL, NULL, NULL, 'SAMPLED') AS DMV
  LEFT OUTER JOIN SYS.INDEXES AS SI
    ON DMV.OBJECT_ID = SI.OBJECT_ID
   AND DMV.INDEX_ID = SI.INDEX_ID
 WHERE avg_fragmentation_in_percent > 10
   AND index_type_desc IN ('CLUSTERED INDEX', 'NONCLUSTERED INDEX')
   AND DMV.record_count >= 2000
 ORDER BY TABLE_NAME DESC
  
```

باید در نظر داشت که اجرای این کوئری بر روی یک دیتابیس حجم زمان بر بوده و احتمالاً عملکرد سیستم را تحت تاثیر قرار می‌دهد. بنابراین استفاده از آن در خارج از ساعت کاری باید مد نظر باشد. بازسازی ایندکس‌ها نیز به همین صورت است.

برای بازسازی تمامی ایندکس‌های یک دیتابیس مفروض می‌توان از [کوئری زیر](#) استفاده کرد:

```

DECLARE @TableName VARCHAR(255)
DECLARE @sql NVARCHAR(500)
DECLARE @fillfactor INT
SET @fillfactor = 80
DECLARE TableCursor CURSOR
FOR
    SELECT OBJECT_SCHEMA_NAME([object_id]) + '.' + NAME AS TableName
    FROM sys.tables

OPEN TableCursor
FETCH NEXT FROM TableCursor INTO @TableName
WHILE @@FETCH_STATUS = 0
BEGIN
    SET @sql = 'ALTER INDEX ALL ON ' + @TableName +
               ' REBUILD WITH (FILLFACTOR = ' + CONVERT(VARCHAR(3), @fillfactor) + ')'
    EXEC (@sql)
    FETCH NEXT FROM TableCursor INTO @TableName
END
CLOSE TableCursor
DEALLOCATE TableCursor
  
```

فرض کنید یک دیتابیس آزمایشی دارید که می‌خواهید تمام رکوردهای آن را حذف کنید. اگر در این دیتابیس انواع و اقسام کلیدهای خارجی و تریگر و امثال آن وجود داشته باشد، صرفاً با یک دستور delete ساده کار به پایان نمی‌رسد و موفق به حذف رکوردها نخواهید شد (چون این قید و بندها به همین جهت طراحی شده‌اند تا یکپارچگی دیتابیس حفظ شود).
 اما اگر واقعاً این قیود در این لحظه مهم نبودند و نیاز بود تا تمام رکوردها را حذف کنیم، سریعترین راه حل موجود چیست؟

```
--Disable Constraints & Triggers
exec sp_MSforeachtable 'ALTER TABLE ? NOCHECK CONSTRAINT ALL'
exec sp_MSforeachtable 'ALTER TABLE ? DISABLE TRIGGER ALL'
--Perform delete operation on all table for cleanup
exec sp_MSforeachtable 'DELETE ?'
--Enable Constraints & Triggers again
exec sp_MSforeachtable 'ALTER TABLE ? CHECK CONSTRAINT ALL'
exec sp_MSforeachtable 'ALTER TABLE ? ENABLE TRIGGER ALL'
--Reset Identity on tables with identity column
exec sp_MSforeachtable 'IF OBJECTPROPERTY(OBJECT_ID('?''), ''TableHasIdentity'') = 1 BEGIN DBCC
CHECKIDENT ('?', RESEED,0) END'
```

اسکریپت فوق تمامی رکوردهای دیتابیس جاری را حذف کرده و همچنین فیلدی‌ای identity را نیز مجدداً به حالت اول باز می‌گرداند.

توضیحات:

یکی از رویه‌های ذخیره شده‌ی سیستمی اس کیوال سرور می‌باشد که مستند نشده است. اگر نیاز داشتید کدی را بدون نوشتن یک کرسر و امثال آن، بر روی تمامی جداول اجرا کنید می‌توان از آن استفاده نمود.
 امضای این رویه ذخیره شده به صورت زیر است:

```
exec @RETURN_VALUE=sp_MSforeachtable @command1, @replacechar, @command2,
@command3, @whereand, @precommand, @postcommand
```

که در آن:

Mقدار بازگشتی است.

اولین دستوری است که اجرا خواهد شد (به همین ترتیب سپس Command2 و بعد از آن Command3 اجرا خواهد گردید) کاراکتری است که در دستور T-SQL مورد نظر جایگزین نام جدول خواهد شد. مقدار پیش فرض آن ؟ است. Precommand پیش از پردازش عملیات روی هر جدولی اجرا می‌شود. Postcommand پس از اجرای کلیه دستورات روی تمامی جداول، اجرا خواهد شد.

چند مثال:

نمایش تمامی جداول دیتابیس جاری

```
EXEC sp_MSforeachtable "print '?'"
```

نمایش اندازه‌ی جداول یک دیتابیس

```
EXEC sp_msforeachtable 'sp_spaceused ''?'''
```


نظرات خوانندگان

نوبسند: مهدی
تاریخ: ۱۳۸۸/۰۵/۲۳ ۰۵:۵۷:۵۷

ممنون، مطلب خوبی بود.

عنوان:	مثال‌هایی در مورد استفاده از SQL Server 2005 Reporting Services
نوبسنده:	وحید نصیری
تاریخ:	۱۳۸۸/۰۱/۱۶ ۱۷:۲۱:۰۰
آدرس:	www.dotnettips.info
برچسب‌ها:	SQL Server

paper- SQL Server راه حلی مبتنی بر سرور جهت گزارشگیری‌های جامع با قابلیت ارائه گزارش‌های SQL Server 2005 Reporting Services است. (جهت آشنایی بیشتر می‌توان به مقاله "آقای SQL Server Reporting Service based ، ad hoc چیست؟" آقای رمضانی مراجعه کرد)

اخیرا مجموعه‌ای مثال در مورد کار با Reporting Services ارائه شده است که از آدرس زیر قابل دریافت است:

[SQL Server 2005 Report Packs](#)

این مجموعه شامل 7 بسته آموزشی و مثال‌های مرتبط می‌باشد:

SQL Server 2005 Integration Services Log Reports
 SQL Server 2005 Report Pack for Microsoft Dynamics Axapta 3.0
 SQL Server 2005 Report Pack for Microsoft Dynamics Navision 4.0
 SQL Server 2005 Report Pack for Microsoft Dynamics Great Plains 8.0
 SQL Server 2005 Report Pack for Microsoft Dynamics Great Plains 9.0
 SQL Server 2005 Report Pack for Microsoft Internet Information Services (IIS)
 SQL Server 2005 Report Pack for Financial Reporting

نظرات خوانندگان

نویسنده: Alex's Blog
تاریخ: ۱۳۸۸/۰۱/۱۷ ۰۸:۳۵:۰۰

سلام

امروز بعد از ۱۶ روز تعطیلی و نبود اینترنت ، به وبلاگتون که سرزدم بمباران اطلاعاتی شدم (:)

ممنون.

اولین و اساسی‌ترین قدم در نگهداری یک سیستم مبتنی بر داده، تهیه پشتیبان‌های منظم و همچنین قابل اطمینان می‌باشد.
دستور T-SQL زیر بدون ریاستور کردن یک فایل بک آپ اس کیوال سرور، سعی در تعیین اعتبار آن می‌کند:

```
RESTORE VERIFYONLY
FROM DISK = 'C:\SQL_Backup\Test1'
WITH FILE = 1,
LOADHISTORY
```

این دستور

وضعیت کامل بودن پشتیبان و همچنین قابل خواندن بودن اطلاعات آن را بررسی می‌کند و در صورت سالم بودن بک آپ، پیغام زیر را نمایش خواهد داد:

```
The backup set on file 1 is valid.
```

عموماً عملیات تهیه پشتیبان در یک مکان مشخص در سرور صورت می‌گیرد (خصوصاً اگر یک job مختص به آن تعریف شده باشد که این کار را به صورت خودکار انجام دهد). بنابراین می‌توان عملیات اعتبار سنجی فوق را مکانیزه کرد. اسکریپت زیر مسیر آخرین بک آپ‌های گرفته شده در سرور را بر می‌گرداند:

```
SELECT DISTINCT physical_device_name
FROM msdb.dbo.backupmediafamily
ORDER BY
    physical_device_name
```

اکنون می‌توان مسیرهای فوق را در یک cursor جهت بررسی صحت تک تک موارد استفاده نمود:

```
DECLARE @path NVARCHAR(1000),
@msg NVARCHAR(MAX),
@NewLine CHAR(2),
@sql NVARCHAR(2000)

SET @NewLine = CHAR(13) + CHAR(10)
SET @msg = ''

DECLARE DATABASES_CURSOR CURSOR
FOR
    SELECT DISTINCT physical_device_name
    FROM msdb.dbo.backupmediafamily
    ORDER BY
        physical_device_name

OPEN DATABASES_CURSOR

FETCH NEXT FROM DATABASES_CURSOR INTO @path

WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT 'Verifying: ' + @path
    SET @sql = 'RESTORE VERIFYONLY FROM DISK = ''' + @path
    + ''' WITH FILE = 1, LOADHISTORY'

    EXEC sp_executesql @sql
    IF @@ERROR <> 0
    BEGIN
        SET @msg = @msg + 'Failed to verify: ' + @path + @NewLine
    END

    FETCH NEXT FROM DATABASES_CURSOR INTO @path
END
```

```
CLOSE DATABASES_CURSOR
DEALLOCATE DATABASES_CURSOR

IF @msg <> ''
BEGIN
    PRINT @msg
    -- send email
    EXECUTE msdb.dbo.sp_send_dbmail
    @recipients = 'nasiri@site.net', -- Change This
    @copy_recipients = 'Administrator@site.net', -- Change This
    @Subject = 'backup verification info.',
    @Body = @msg
    ,@importance = 'High'

END
```

اسکریپت فوق بر روی تمامی مسیرهای ثبت شده موجود که در آن‌ها پیشتر پشتیبان تهیه شده است، دستور RESTORE VERIFYONLY را اجرا می‌کند و در آخر اگر پیغامی حاصل شد، یعنی مشکلی پدید آمده و ایمیلی را به اشخاص مورد نظر ارسال می‌کند.

می‌توان بر روی این اسکریپت یک job تهیه کرد که هر روز پس از تهیه بک آپ خودکار، کار بررسی صحت عملیات را نیز انجام دهد.

عنوان: SQL Server 2008 سرویس پک یک
نوبتندہ: وحید نصیری
تاریخ: ۱۳۸۸/۰ ۱/۲۰ ۰۰:۳۹:۰۰
آدرس: www.dotnettips.info
برچسبها: SQL Server

نگارش نهایی سرویس پک یک SQL Server 2008 ارائه شد. (به بیان دیگر اکنون می‌توانید با خیال راحت از آن در محیط کاری استفاده کنید!)

دریافت

[توضیحات بیشتری در مورد این نگارش](#)
[لیست باگ‌های فیکس شده در سرویس پک یک](#)

این به روز رسانی به همراه [Feature Pack 2009](#) و [Report Builder 2.0](#) نیز می‌باشد.

اگر برنامه‌ی شما برای مثال با SMO مربوط به اس کیوال سرور 2008 کامپایل شود، روی سروری با SQL Server 2005 کار نخواهد کرد و پیغام می‌دهد که نگارش 10 اسembلی Microsoft.SqlServer.Management.Sdk.Sfc یافت نشد.

یک راه حل آن، نصب [Microsoft SQL Server 2008 Management Objects](#) بر روی سرور است، یا راه حل دوم، پیدا کردن اسembلی‌هایی که برنامه به آن‌ها ارجاع دارد و کپی کردن آن‌ها کنار فایل اجرایی برنامه در سرور. (درست کردن یک برنامه پرتاپل دات نتی، یا نسبتاً پرتاپل!)

برای این منظور کلاس زیر تهیه شده است که مسیر فایل اجرایی یا dll یک پروژه را دریافت کرده و [لیست تمام ارجاعات به آن را](#) به صورت بازگشتی پیدا می‌کند. (البته در قسمت مسیر اسembلی‌ها، اسembلی‌های سیستمی که با خود دات نت فریم ورک نصب می‌شوند، حذف شده است)

```

using System.Collections.Generic;
using System.Reflection;
using System.IO;

namespace App
{
    class CFindRef
    {
        #region Fields (2)

        /// <summary>
        /// لیستی جهت نگهداری نام اسembلی‌ها
        /// </summary>
        private readonly List<string> _assemblies = new List<string>();
        /// <summary>
        /// لیستی جهت نگهداری مسیر اسembلی‌ها
        /// </summary>
        private readonly List<string> _filePath = new List<string>();

        #endregion Fields

        #region Constructors (1)

        /// <summary>
        /// سازنده کلاس
        /// </summary>
        /// <param name="fileName">مسیر اولیه اسembلی مورد نظر</param>
        public CFindRef(string fileName)
        {
            ListReferences(fileName);
        }

        #endregion Constructors

        #region Properties (2)

        /// <summary>
        /// لیست مسیر اسembلی‌هایی که به آن‌ها ارجاعی وجود دارد منهای موارد سیستمی
        /// </summary>
        public List<string> ReferencedFiles
        {
            get
            {
                _filePath.Sort();
                return _filePath;
            }
        }

        /// <summary>
        /// لیست کامل اسembلی‌هایی که اسembلی ما به آن‌ها وابسته است
        /// </summary>
        public List<string> ReferencedNames
        {
            get

```

```

    {
        _assemblies.Sort();
        return _assemblies;
    }
}

#endregion Properties

#region Methods (1)

// Private Methods (1)

/// <summary>
/// متدی بازگشتی جهت یافتن لیست ارجاعات
/// </summary>
/// <param name="path">/param>
private void ListReferences(string path)
{
    آیا تکراری است؟//
    if (_assemblies.Contains(path))
        return;

    Assembly asm;
    آیا فایل است یا نام کامل اسembly //
    if (File.Exists(path))
    {
        // load the assembly from a path
        asm = Assembly.LoadFrom(path);
    }
    else
    {
        سعی در بارگذاری اسembly //
        try
        {
            asm = Assembly.Load(path);
        }
        catch
        {
           asm = null; جای بهبود دارد//
        }
    }

    if (asm == null) return;

    افزودن به لیست نامها
    _assemblies.Add(path);
    string asmLocation = asm.Location;
    حذف موارد سیستمی از لیست مسیر فایل‌ها//
    if (asmLocation != null && !asmLocation.Contains("\\System.")
        && !asmLocation.Contains("\\mscorlib"))
        _filePath.Add(asmLocation);

    پیدا کردن ارجاع‌ها
    AssemblyName[] imports = asm.GetReferencedAssemblies();
    // iterate
    foreach (AssemblyName asmName in imports)
    {
        فراخوانی بازگشتی جهت یافتن تمامی ارجاعات //
        ListReferences(asmName.FullName);
    }
}

#endregion Methods
}

```

مثالی در مورد نحوه استفاده از آن:

```

CFindRef cfr = new CFindRef(@"C:\App\test.exe");
foreach (var asmRef in cfr.ReferencedFiles)
{
    textBox1.Text += asmRef + Environment.NewLine;
    //Application.DoEvents();
}

```

برای نمونه، برنامه‌ای که از SMO مربوط به اس کیوال سرور 2008 استفاده می‌کند، این لیست ارجاعات را دارد:

```
C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.ConnectionInfo\10.0.0.0__89845dcd8080cc91\Microsoft.SqlServer.ConnectionInfo.dll  
C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.Dmf\10.0.0.0__89845dcd8080cc91\Microsoft.SqlServer.Dmf.dll  
C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.Management.Sdk.Sfc\10.0.0.0__89845dcd8080cc91\Microsoft.SqlServer.Management.Sdk.Sfc.dll  
C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.ServiceBrokerEnum\10.0.0.0__89845dcd8080cc91\Microsoft.SqlServer.ServiceBrokerEnum.dll  
C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.Smo\10.0.0.0__89845dcd8080cc91\Microsoft.SqlServer.Smo.dll  
C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.SqlClrProvider\10.0.0.0__89845dcd8080cc91\Microsoft.SqlServer.SqlClrProvider.dll  
C:\WINDOWS\assembly\GAC_MSIL\Microsoft.SqlServer.SqlEnum\10.0.0.0__89845dcd8080cc91\Microsoft.SqlServer.SqlEnum.dll
```

با کپی کردن همین فایل‌ها کنار فایل اجرایی برنامه‌ای که قرار است روی سروری با SQL Server 2005 کار کند، مشکل برطرف می‌شود و برنامه بدون مشکل کار خواهد کرد.

برنامه آماده هم در این زمینه موجود است، برای مثال CheckAsm

[مشاهده سایت](#)

نظرات خوانندگان

نویسنده: Sirasad

تاریخ: ۰۱/۲۵/۱۳۸۸ ۰۳:۳۷:۰۰

ببخشید آقای نصیری که این رو اینجا مطرح کردم. من به سفارش شما تو شرکت از SVN استفاده کردم . بدین نحو که اول خود SVN رو نصب کردم بعد TortoiseSVN . بعده هم که برنامه رو سرور مثل سعت کار میکنه ، ولی رو کلاینت ها مورد داره . پروژه رو هم رو سرور share کردم و کلاینت هم اون رو Map کردند . حالا وقتی Solution رو باز می کنیم برای هر پروژه این warning میده : Project Location Is not Trusted

وقتی هم که روی یک button کلیک میکنیم این error رو میده : Soure file f:\solution\project1 does not belong to the .project being debugged

وقتی هم میخاییم commit رو solution کنیم error میده که : Unable to open an ra_local url.unable to open repository البته خیلی من رو ببخشید که اندازه یه تاپیک پست دادم.

نویسنده: وحید نصیری

تاریخ: ۰۱/۲۵/۱۳۸۸ ۰۴:۴۹:۰۰

سلام

نیازی نبود سورسی را Map کنید یا اصلاً نباید این کار را می‌کردید. (اگر منظور مپ کردن سورس بوده نه فایل اجرایی برنامه) SVN وظیفه مدیریت و به اشتراک گذاشتن پروژه رو داره، نه شبکه ویندوزی یا لینوکسی. (حتماً از visual svn server استفاده کنید تا این موارد را برای شما ساده کند) کلاینت‌ها هر کدام نسخه‌ی کامل و لوکال خودشون رو باید داشته باشند (از طریق check out مخزن کد این پروژه لوکال باید تشکیل شود نه کپی دستی). سپس مثل اینکه دارند لوکال کار می‌کنند (نه از روی شبکه در حالت مپ شده). کاملاً حالت معمولی و قطع از شبکه. SVN برای مدیریت پروژه روی اینترنت هم بکار میره. نه چیزی map میشه و نه لازم هست کاربر همیشه به شبکه وصل باشه.

نسخه کد شما که روی سرور هست و توسط SVN نگهداری می‌شود، مخزن اصلی است که تغییرات با آن هماهنگ می‌شود. برای انتقال کد به مخزن، باید عملیات check in صورت گیرد.

بعد هر کدام از اعضای تیم زمانیکه check out می‌کنند ، یک نسخه‌ی محلی دریافت می‌کنند و این فolder تحت کنترل SVN قرار می‌گیرد، حالا مباحثه commit و update و غیره کار می‌کند. فقط هر بار که می‌خواهند commit کنند باید اول update کنند بیبنند کسی چیزی را تغییر داده، تصادمی هست یا نه؟ بعد commit کنند به سرور. (یعنی ارتباط با شبکه فقط در همین چند لحظه کوتاه است و بسیار سریع هم خواهد بود)

فصل دوم کتابچه‌ای را که من تهیه کردم لطفاً مطالعه کنید، گردش کاری آن توضیح داده شده است.

<http://www.dotnettips.info/2008/10/subversion.html>

در فصل یک هم توضیح دادم که چه پورتی را باید روی سرور باز کنید تا SVN توسط فایروال بلاک نشود.

حتماً توصیه می‌کنم اگر با VS.Net کار می‌کنید افزونه SVN را نصب کنید تا راحت و بدون دردسر کار کنید .

خطایی رو که توضیح دادید مربوط هست به اشتراک گذاشتن فایل اجرایی یک برنامه دات نتی روی شبکه که این خطایی رو می‌گیرید:

Project Location Is not Trusted

این خطایی رو با دادن full trust به برنامه می‌توانید حل کنید که اینجا قدم به قدم توضیح داده شده:

[http://msdn.microsoft.com/en-us/library/bkws2c\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/bkws2c(VS.80).aspx)

همچنین کامنت‌های اون رو لطفاً بخونید. مثلاً ممکن هست فایل بلاک شده باشه که با کلیک راست و unblock کردن مشکل حل میشه.

خطای زیر بیشتر مربوط به حالتی است که الف) هنوز مخزن کد ایجاد نشده و ب) عملیات `check in` `initial import` یا `check out` صورت نگرفته .
نحوی سند: `Unable to open an ra_local url. unable to open repository`
تاریخ: ۱۳۸۸/۰۱/۲۶ ۰۹:۰۰:۰۰
حتماً کتابچه فوق را مطالعه نمایید.

نویسنده: SirAsad
تاریخ: ۱۳۸۸/۰۱/۲۶ ۰۹:۰۰:۰۰

خیلی ممنون آقای نصیری ... ولی یه قضیه برام حل نشد . شما گفتید که: "کلاینت‌ها هر کدام نسخه‌ی کامل و لوکال خودشون رو باید داشته باشند (از طریق `check out` مخزن کد این پروژه لوکال باید تشکیل شود نه کپی دستی). سپس مثل اینکه دارند لوکال کار می‌کنند (نه از روی شبکه در حالت مپ شده)." چجوری باید بدون اینکه اونا سرور رو مپ کنند و یا اینکه فایل‌ها رو خودشون کپی کنند به سورس‌ها دسترسی داشته باشند ؟

من الان دارم `Viusal SVN Server` رو دانلود میکنم ، اما قبل این هم از `Ankh-SVN-2.0.6347` استفاده میکردم که کار هم باهش بسیار آسان بود ... اگر لطفی بکنید و در مورد ساخت نسخه مخصوص کلاینت‌ها یه توضیح بدید ممنون میشم .

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۱/۲۶ ۱۱:۱۴:۰۰

اولاً قرار نیست همه به سورس‌ها دسترسی داشته باشند. هر شخصی به پروژه‌ای که شما بهش دسترسی خواهید داد باید دسترسی داشته باشد. مدیریت اینکار با `visual svn server` و یکپارچه بودنش با سیستم `windows authentication` بسیار ساده و در حد چند کلیک است.

یک `repository` در این سرور درست می‌کنید. بعد نسخه‌ی اولیه پروژه در آن `check in` و `import` می‌شود. حالا سرور راه اندازی شده و آماده سرویس دهی است.

ثانیاً توزیع کد از طریق `svn` انجام می‌شه. کلاینت‌ها مثلاً با استفاده از `TortoiseSVN` یا افزونه‌ای که شما نصب می‌کنید با سرور ارتباط خواهند داشت و همین. نیازی نیست چیزی مپ بشه.

آنرا با افزونه‌ای مثل `Ankh-SVN` اشتباہ نگیرید. `Ankh-SVN` یک کلاینت `SVN` است و به خودی خود کاری ازش ساخته نیست. این افزونه به سرور وصل می‌شود اطلاعات جدید را می‌گیرد یا اطلاعات موجود را با سرور هماهنگ می‌کند. برای توضیحات بیشتر لطفاً کتابچه ذکر شده را مطالعه کنید. کمی تست کنید بعد پیاده سازی

<http://www.dotnettips.info/2008/10/subversion.html>

نویسنده: Sirasad
تاریخ: ۱۳۸۸/۰۱/۲۷ ۲۲:۴۳:۰۰

آقای خیلی ممنون . آقا مشکل ما حل شد .

دست گل تون درد نکنه

اگر با زبان C و مشتقات آن آشنايی داشته باشيد، حتما با عملگرهای ترکيبی آنها که جهت خلاصه نويسی بكار می روند، نيز کار کرده ايد. برای مثال:

```
int i =5;  
i += 15; // i = i + 15;
```

اس کیوال سرور 2008 نيز از اينگونه عملگرها پشتيباني به عمل می آورد . برای نمونه:

```
DECLARE @x1 int = 27;  
SET @x1 += 2 ;  
SELECT @x1 AS Added_2;
```

در دستورات T-SQL فوق دو نکته قابل توجه است:

الف) امكان تعريف و مقدار دهی همزمان یک متغير (مقدار دهی همزمان با تعريف، تا قبل از اس کیوال سرور 2008 پشتيباني نمی شد)

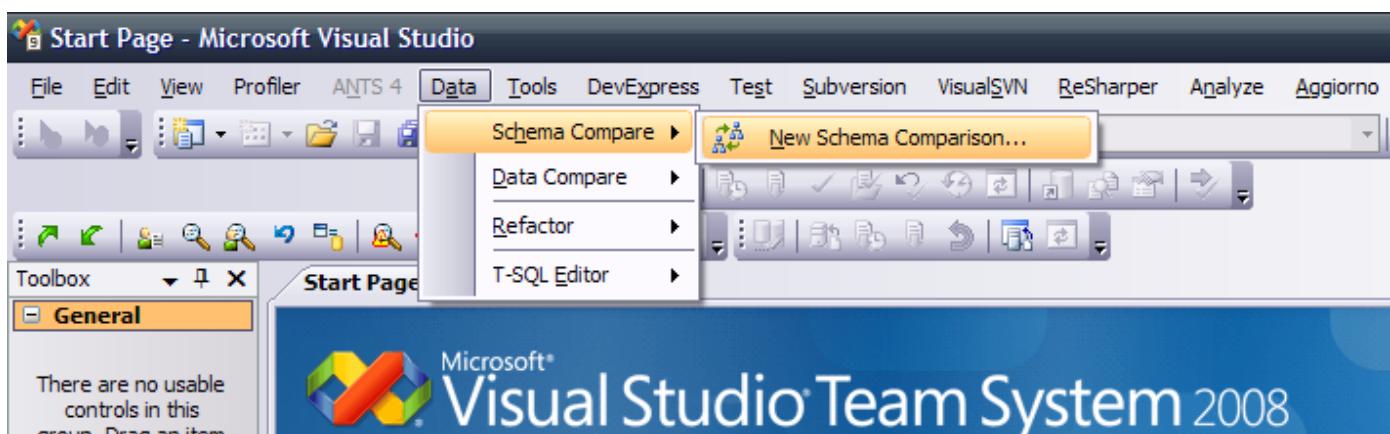
ب) امكان استفاده از عملگرهای C مانند در عبارات T-SQL

لیست این عملگرهای جدید به شرح زیر است:

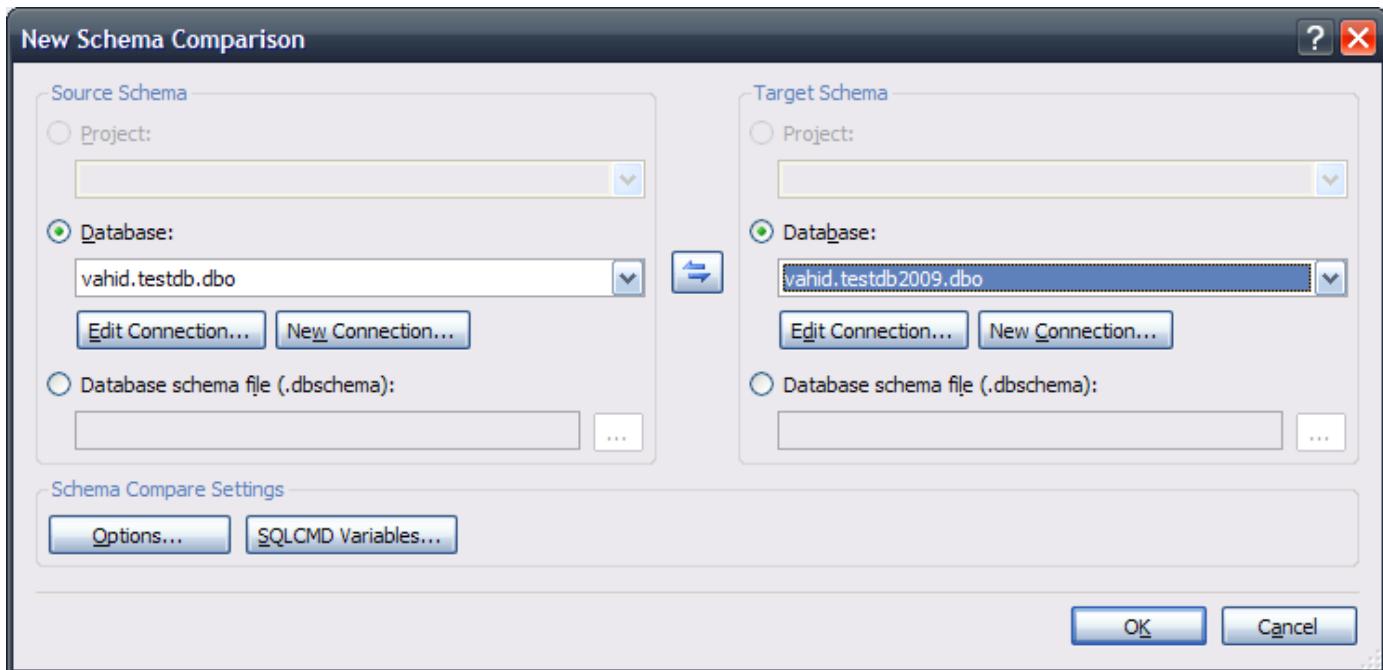
```
+= (Add EQUALS) (Transact-SQL)  
-= (Subtract EQUALS) (Transact-SQL)  
*= (Multiply EQUALS) (Transact-SQL)  
/= (Divide EQUALS) (Transact-SQL)  
%= (Modulo EQUALS) (Transact-SQL)  
&= (Bitwise AND EQUALS) (Transact-SQL)  
^= (Bitwise Exclusive OR EQUALS) (Transact-SQL)  
|= (Bitwise OR EQUALS) (Transact-SQL)
```

یکی از مواردی که در محیط کاری زیاد پیش می‌آید بحث همگام نبودن دیتابیس توسعه با دیتابیس کاری است. منظور از دیتابیس توسعه، همان دیتابیسی است که برای برنامه نویسی و آزمایش از آن استفاده می‌شود و دیتابیس کاری هم مشخص است (برای مثال بر روی یک سرور در اینترنت داخلی یک شرکت و یا بر روی یک سرور اینترنتی قرار دارد). عادت‌های مختلفی هم این‌جا ممکن است وجود داشته باشد، برای مثال تغییرات جدید بر روی دیتابیس کاری اعمال شود و سپس فراموش شود که همان‌ها نیز باید به دیتابیس توسعه هم اعمال شوند تا در تغییرات بعدی برای آزمایش دچار مشکل نشویم و بر عکس. بعد از یک مدت هم تبدیل به کابوس می‌شود؛ نمی‌دانیم الان دیتابیس کاری جدیدتر است یا دیتابیس توسعه؛ و یا اینکه کلاً دو دیتابیس مفروض چه تفاوت‌های ساختاری با هم دارند (بديهی است بحث ديتا در اينجا در درجه‌ی اول اهمیت قرار ندارد). فرست اين هم وجود ندارد که تک تک جداول، ویووها، رویه‌های ذخیره شده و خلاصه تمامی اشیاء مرتبط را بررسی کنیم که چه اختلافی با هم دارند. اينجا مستندات هم کمک نخواهند کرد چون صحبت از یک جدول با ۵ فیلد در میان نیست که موارد را سریع و به صورت دستی تطابق دهیم. همچنین این مشکل عموماً زمانی رخ می‌دهد که یکی از دو طرف در حال حاضر مستندات کامل و به روزی ندارد. اکنون چه باید کرد؟

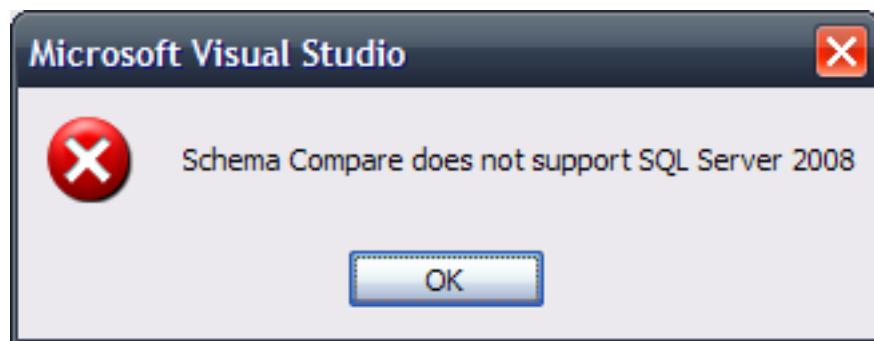
اولین فکري که به ذهن خطور می‌کند مراجعي به ابزارهای جانبی است (مثلًا [Red Gate's SQL Compare](#) چند صد دلاری) غافل از اينکه خود Visual studio 2008 (نگارش‌های تیمی و دیتابیسی) این قابلیت را نیز ارائه می‌دهد (شکل زیر).



پس از انتخاب new schema comparison، در صفحه‌ای که ظاهر می‌شود، بر روی new connection کلیک کرده و دیتابیس‌های مبداء و مقصد را جهت مقایسه ساختاری انتخاب نمایید و سپس بر روی دکمه Ok کلیک کنید.



اگر اس کیوال سرور 2008 را نصب کرده باشد، با پیغام زیر روبرو خواهید شد:



برای رفع این مشکل باید بسته به روز رسانی زیر را نصب کرد تا این نگارش نیز پشتیبانی شود:

[Microsoft® Visual Studio Team System 2008 Database Edition GDR](#)

(برای نصب حتما باید SP1 مربوط به 2008 VS.Net پیشتر نصب شده باشد)

پس از کلیک بر روی دکمه Ok، کار آنالیز دو دیتابیس شروع شده و تفاوت‌ها گزارش داده می‌شوند:

مقایسه ساختاری دو دیتابیس

The schema comparison is complete.

Status	vahid.testdb.dbo (Source Database)	Update Action	vahid.testdb2009.dbo (Target Database)
+ Equal	<unnamed>	Skip	<unnamed>
Tables			
+ Different definition	dbo.tblDates	Update	dbo.tblDates
+ Equal	dbo.tblFarsiTest	Skip	dbo.tblFarsiTest
+ Equal	dbo.tblMax	Skip	dbo.tblMax
+ Different definition	dbo.tblPIIDs	Update	dbo.tblPIIDs
+ Equal	dbo.tblTest	Skip	dbo.tblTest
+ Equal	dbo.tblTimes	Skip	dbo.tblTimes
+ Equal	dbo.tblTransItems	Skip	dbo.tblTransItems

Object Definitions

Source Object:

```

1 CREATE TABLE [dbo].[tblPIIDs] (
2     [p_id] NVARCHAR (50) NULL
3 );
4
5

```

Target Object:

```

1 CREATE TABLE [dbo].[tblPIIDs] (
2     [p_id] NVARCHAR (50) NULL,
3     [f1] INT NULL
4 );
5
6

```

New Text Different Text Missing Text

همچنین جهت سهولت کار، اسکریپت SQL T-SQL ایی را نیز به نام schema update script تولید می‌کند که با اجرای آن به سادگی کار به روز رسانی دیتابیس مقصد صورت خواهد گرفت.

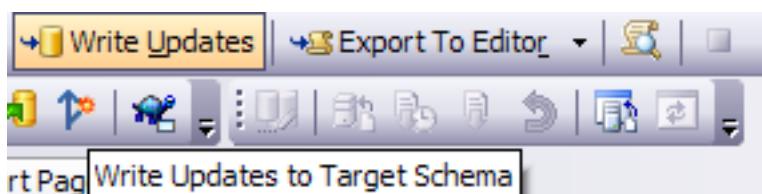
Schema Update Script - [vahid.testdb2009.dbo]

```

67
68 GO
69 ALTER TABLE [dbo].[tblPIIDs] DROP COLUMN [f1];
70
71
72 GO
73 PRINT N'Refreshing dbo.crossJoin...';
74

```

در پایان یا می‌توان اسکریپت تولید شده را ذخیره کرد و در زمانی دلخواه اجرا و اعمال نمود و یا می‌توان بلافاصله بر روی دکمه write updates که در نوار ابزار ظاهر شده است کلیک کرد تا دیتابیس مقصد از لحاظ ساختاری با دیتابیس مبداء یکی شود.



نظرات خوانندگان

نوبنده: Anonymous
تاریخ: ۱۳۸۸/۰۲/۰۱ ۲۱:۳۹:۰۰

من هر روز مطالب شما را پیگیری می کنم. مثل همیشه عالی بود.

نوبنده: salarblog
تاریخ: ۱۳۸۸/۰۲/۰۲ ۰۱:۳۰:۰۰

بسیار نکته کاربردی بود
همانطور که گفتید از reg gate استفاده می کردم، غافل از این امکان!

نوبنده: مهدی یوسفی
تاریخ: ۱۳۸۸/۰۲/۰۲ ۰۸:۰۸:۰۰

من هم برنامه های زیادی رو تست کردم
برنامه رایگان در این زمینه وب سایت کلور کامپوننت هست:
<http://www.clevercomponents.com/downloads/dbcomparer/dbcdownload.asp>
البته این برنامه هم اشکالاتی دارد ولی به نظر من که نسخه تیم سوییت رو ندارم خوب بود.

نوبنده: reza
تاریخ: ۱۳۸۸/۰۲/۰۲ ۱۹:۴۷:۰۰

salam
man az in estedage mikonam
<http://opendbiff.codeplex.com>

نوبنده: مهدی
تاریخ: ۱۳۸۸/۰۲/۰۵ ۱۳:۱۶:۰۰

سلام
ولی من Redgate رو ترجیح میدم؛ چون کرکش دم دسته!

نکته‌ی کوچکی در مورد ارسال ایمیل فارسی توسط رویه ذخیره شده سیستمی sp_send_dbmail اس کیوال سرور وجود دارد که شبیه به insert داده‌های فارسی در دیتابیسی است که پس از ثبت، به صورت ؟؟؟ ذخیره می‌شوند. (این مورد با تنظیم collation تقریباً قابل حل است)

اگر هنگام ثبت collation عربی یا فارسی (در اس کیوال سرور 2008) انتخاب شود، مشکلی در ثبت نخواهد بود. اگر به collation اهمیت نمی‌دهید باید اس کیوال سرور را مجبور کرد که داده را یونیکد ذخیره کند و اینکار با اضافه کردن یک N به ابتدای رشته صورت می‌گیرد و همچنین انتخاب نوع داده‌های n دار مانند nvarchar و امثال آن (n در اینجا به معنای national و احصار آن می‌باشد):

Insert into tblTest(f1,f2) values(1,N'متن فارسی')

دقیقاً همین نکته هم درباره‌ی ارسال ایمیل از طریق اس کیوال سرور صادق است. اگر N به ابتدای رشته اضافه نشود، رشته ارسالی را با فرمت ANSI ارسال می‌کند و داده‌های پونیکد متن تخریب خواهد شد؛ مثلاً چیزی شبیه به حالت زیر:

این مشکل به صورت زیر قابل حل است:

```
DECLARE @msg NVARCHAR(max)
SET @msg=N'متن فارسی'
```

برای ردگیری وضعت این مدل‌های ارسالی، هم می‌توان از کوئری‌های زیر استفاده نمود:

```
SELECT * from sysmail_allitems  
SELECT * from sysmail_faileditems  
SELECT * from sysmail_event_log
```

عنوان: محدود کردن دسترسی به اس کیوال سرور بر اساس IP
 نویسنده: وحید نصیری
 تاریخ: ۲۱:۰۹:۰۰ ۱۳۸۸/۰۲/۱۰
 آدرس: www.dotnettips.info
 برچسبها: SQL Server

عموماً محدود کردن دسترسی بر اساس IP بهتر است بر اساس راه حل های مانند [فایروال](#) ، [IPSec](#) و یا [RRAS IP Filter](#) صورت گیرد که جزو بهینه ترین و امن ترین راه حل های ممکن هستند.

در ادامه قصد داریم این محدودیت را با استفاده از امکانات خود اس کیوال سرور انجام دهیم (بلاک کردن کاربران بر اساس IP های غیر مجاز). مواردی که در ادامه ذکر خواهد شد در مورد اس کیوال سرور 2005 ، سرویس پک 2 به بعد و یا اس کیوال سرور 2008 صادق است.

اس کیوال سرور این قابلیت را دارد که می توان بر روی کلیه لاغین های صورت گرفته در سطح سرور [تیریگر تعريف کرد](#) . به این صورت می توان تمامی لاغین ها را برای مثال لاغ کرد (جهت بررسی مسائل امنیتی) و یا می توان هر لاغینی را که صلاح ندانستیم rollback ایجاد محدودیت روی لاغین در سطح سرور.

لاغ کردن کلیه لاغین های صورت گرفته به سرور

ایجاد جدولی برای ذخیره سازی اطلاعات لاغین ها:

```

USE [master]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

SET ANSI_PADDING ON
GO

CREATE TABLE [dbo].[Logging](
[id] [int] IDENTITY(1,1) NOT NULL,
[LogonTime] [datetime] NULL,
[LoginName] [nvarchar](max) NULL,
[ClientHost] [varchar](50) NULL,
[LoginType] [varchar](100) NULL,
[AppName] [nvarchar](500) NULL,
[FullLog] [xml] NULL,
CONSTRAINT [PK_IP_Log] PRIMARY KEY CLUSTERED
(
[id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO

ALTER TABLE [dbo].[Logging] ADD CONSTRAINT [DF_IP_Log_LogonTime] DEFAULT (getdate()) FOR [LogonTime]
GO
  
```

در ادامه یک تیریگر لاغین را جهت ذخیره سازی اطلاعات کلیه لاغین ها به سرور ایجاد می نمائیم:

```

USE [master]
GO

CREATE TRIGGER LogonTrigger
ON ALL SERVER
FOR LOGON
  
```

محدود کردن دسترسی به اس کیوال سرور بر اساس IP

```
AS
BEGIN
    DECLARE @data XML
    SET @data = EVENTDATA()

    INSERT INTO [Logging]
    (
        [LoginName],
        [ClientHost],
        [LoginType],
        [AppName],
        [FullLog]
    )
    VALUES
    (
        @data.value('/EVENT_INSTANCE/LoginName')[1], 'nvarchar(max)'),
        @data.value('/EVENT_INSTANCE/ClientHost')[1], 'varchar(50)'),
        @data.value('/EVENT_INSTANCE/LoginType')[1], 'varchar(100)'),
        APP_NAME(),
        @data
    )
END
```

اکنون برای مثال از آخرین 100 لگین انجام شده، به صورت زیر می‌توان گزارشگیری کرد:

```
SELECT TOP 100 * FROM [master].[dbo].[Logging] ORDER BY id desc
```

و بدیهی است در تریگر فوق می‌توان روی هر کدام از آیتم‌های دریافتی مانند ClientHost و غیره فیلتر ایجاد کرد و تنها موارد مورد نظر را ثبت نمود.

محدود کردن کاربران بر اساس IP

ایی که در رخداد لگین فوق بازگشت داده می‌شود همان IP کاربر راه دور است. برای فیلتر کردن IP های غیرمجاز، ابتدا در دیتابیس مستر یک جدول برای ذخیره سازی IP های مجاز ایجاد می‌کنیم و IP های کلیه کلاینت‌های معتبر خود را در آن وارد می‌کنیم:

```
USE [master]
GO
CREATE TABLE [IP_RESTRICTION](
    [ValidIP] [varchar](15) NOT NULL,
CONSTRAINT [PK_IP_RESTRICTION] PRIMARY KEY CLUSTERED
(
    [ValidIP] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

سپس تریگر لگین ما برای منع کاربران غیرمجاز بر اساس IP، به صورت زیر خواهد بود:

```
USE [master]
GO

CREATE TRIGGER [LOGIN_IP_RESTRICTION]
ON ALL SERVER
FOR LOGON
AS
BEGIN
    DECLARE @host NVARCHAR(255);
    SET @host = EVENTDATA().value('/EVENT_INSTANCE/ClientHost')[1], 'nvarchar(max)');

    IF (
        NOT EXISTS(
            SELECT *
            FROM   MASTER.dbo.IP_RESTRICTION
                WHERE ValidIP = @host)
    )
        RAISERROR('User %s from IP %s is blocked.', 16, 1, @host, @host)
```

```
        WHERE  ValidIP = @host
    )
BEGIN
    ROLLBACK;
END
END;
```

اخطار مهم!

تریگر فوق خطرناک است! ممکن است خودتان هم دیگر نتوانید لاگین کنید!! (حتی با اکانت ادمین)
بنابراین قبل از لاگین حتما IP لوکال و یا ClientHost لوکال را هم وارد کنید.

اگر گیر افتادید به صورت زیر می‌شود رفع مشکل کرد:

تنها حالتی که تریگر لاگین را فعال نمی‌کند Dedicated Administrator Connection است یا DAC هم به آن گفته می‌شود. به صورت پیش فرض برای ایجاد این اتصال اختصاصی باید به کامپیوتری که اس کیوال سرور بر روی آن نصب است به صورت لوکال لاگین کرد و سپس در خط فرمان دستور زیر را صادر کنید (حرف A آن باید بزرگ باشد):

```
C:\>sqlcmd -A -d master -q "insert into IP_RESTRICTION(validip) values('<local machine>')"
```

به این صورت local machine به جدول IP های مجاز اضافه شده و می‌توانید لاگین کنید!
این نوع تریگرهای در قسمت server objects در management studio ظاهر می‌شوند.

نظرات خوانندگان

نویسنده: مجید
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۱:۵۶

سلام جناب نصیری
مقاله بسیار جالبی بود.(حتی با اینکه متعلق به سه سال پیش بود)
بر اساس مقاله، تریگری برای لاغ کردن تمامی لاگین ها به سرور ایجاد کردم. اما به یک مشکل برخوردم:اگر لاگین sysAdmin باشد مشکلی نخواهیم داشت اما اگر تنها public باشد در هنگام لاگین با خطأ مواجه خواهیم شد:

```
Logon failed for login 'm' due to trigger execution.
```

```
Changed database context to 'master'.
```

```
Changed language setting to us_english. (Microsoft SQL Server, Error: 17892)
```

آیا راه حلی وجود دارد؟متاسفانه من نتوانستم راه حلی پیدا کنم.
با تشکر...

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۲۹ ۱۲:۲۶

باید دسترسی به سرور شما داشت تا بشود اظهار نظر کرد. دسترسی‌ها به چه بانک‌های اطلاعاتی دسترسی دارند و مسایلی از این دست که از راه دور قابل بررسی نیست.
ولی در کل در اینجا از دیتابیس سیستمی master برای ساخت جدول Logging استفاده شده. این را تغییر بدید به یک دیتابیس دیگر با سطح دسترسی عمومی.

```
USE [dbName]
go
INSERT INTO [dbName].schemaName.[Logging]
...
```

از اس کیوال سرور 2005 به بعد تابع [HashBytes](#) نیز به مجموعه توابع قابل استفاده در دستورات T-SQL اس کیوال سرور اضافه شده است که الگوریتم‌های MD2 | MD4 | MD5 | SHA1 را پشتیبانی می‌کند. برای مثال:

```
DECLARE @str1 VARCHAR(4),
        @str2 NVARCHAR(4)

-- متن یونیکد اینجا ناقص ذخیره می‌شود
SET @str1 = 'وحید'

SET @str2 = N'وحید'

SELECT hashbytes('md5', @str1) --C82A7D721AAE517AD76EF1B871BC33CE
SELECT hashbytes('md5', @str2) --7D883091B80F3CD20B872CABBFDDACDF
```

اگر این نتایج را بخواهیم با استفاده از فضای نام استاندارد `System.Security.Cryptography` تولید کنیم، باید به رشته `encoding` دریافتی حتماً دقت داشت؛ در غیر اینصورت نتایج یکسان نخواهند بود. مهم‌ترین `encoding`‌های پشتیبانی شده در دات نت در جدول زیر بر شمرده شده‌اند:

تعداد بیت هر کاراکتر	Encoding
هر کاراکتر آن 7 بیت است	ASCII
هر کاراکتر آن 7 بیت است	UTF7
هر کاراکتر آن 8 بیت و یا یک بایت است	UTF8
هر کاراکتر آن 16 بیت و یا دو بایت است	Unicode (UTF-16)
هر کاراکتر آن 32 بیت و یا 4 بایت است	UTF32

نوع `nvarchar` در اس کیوال سرور همانند حالت `Encoding.Unicode` دات نت است و هر کاراکتر آن 2 بایت می‌باشد. این نکته هنگام استفاده از این توابع بسیار حائز اهمیت است. برای مثال اگر تابع `HashBytes` اس کیوال سرور را بخواهیم در دات نت پیاده سازی کنیم، به کلاس زیر خواهیم رسید:

```
using System.Text;
using System.Security.Cryptography;

class CHash
{
    public static string GetMD5Hash(string input, Encoding encoding)
    {
        byte[] bytes = new MD5CryptoServiceProvider().ComputeHash(encoding.GetBytes(input));
        StringBuilder chars = new StringBuilder();
        foreach (byte chr in bytes)
        {
            chars.Append(chr.ToString("x2"));
        }
        return chars.ToString();
    }
}
```

در اینجا تنها حالت زیر با هش تولید شده یک فیلد یا متغیر از نوع `nvarchar` توسط تابع `HashBytes` اس کیوال سرور معادل است:

```
string result = CHash.GetMD5Hash("وحید", Encoding.Unicode);
```

پ.ن.

احتمالاً عده‌ای را دیده‌اید که هر چقدر تلاش می‌کنند با سی شارپ متون ایران سیستم تحت داس را به نمونه‌های ویندوزی تبدیل کنند، کمتر موفق می‌شوند؛ علت را با توجه به جدول encoding فوق و عدم اطلاع از آن بهتر می‌توان بررسی کرد.

نظرات خوانندگان

نویسنده: Anonymous

تاریخ: ۱۳۸۸/۰۲/۱۵ ۰۰:۱۴:۰۰

با عرض سلام

آقای نصیری حدودا ۳ سال پیش با خوندن کتاب شما و آقای هاشمیان با دات نت آشنا شدم بعد از مدت‌ها و بلاگ شما را بصورت اتفاقی پیدا کردم (ضمنا همچنان دنبال و بلاگ و ... از آقای هاشمیان هستم) شما ۲ نفر حق معلمی گزدن من دارید امیدوارم بتونم یه روز جبران کنم

قبل از هر چیز از شما بخاطر بروز نگه داشتن و بلاگتون تشکر می‌کنم

و اما سوال و درخواستی دارم

سوال

۱: برای هش کردن پسورد از همین تابعی که معرفی کردید (در قالب تریگر) استفاده کنم یا از توابع دات نت

پیشنهاد

۲: اگه امکانش هست درباره وب سرویس‌ها هم تاپیک داشه باشد مخصوصا تکنولوژی جدید دات نت WCF

ممnon از توجهتون

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۲/۱۵ ۰۰:۵۲:۰۰

سلام،

- شما لطف دارید.

- بر روی `select` نمی‌شود تریگر تعریف کرد بنابراین ...

و در کل فرقی نمی‌کند. یا یک رویه ذخیره شده بنویسید و کل عملیات را در آن پیاده کنید و سپس مثلًا یک ۰ یا ۱ یا `true` یا `false` بازگشت دهید که شخص یوزر نیم و پسورد درستی وارد کرده (در این حالت می‌شود از روش اس کیوال سرور استفاده کرد برای هش کردن پسورد وارد شده و سپس مقایسه)

یا پسورد را هش کنید (توسط کلاینت دات نت) و سپس با نمونه موجود در دیتابیس مقایسه کنید.

هر دو راه قابل انجام است و تفاوتی هم ندارد.

روش رویه ذخیره شده بهینه‌تر است و بدلیل کامپایل شدن سریعتر هم خواهد بود.

نویسنده: Mohammad Shams Javi

تاریخ: ۱۳۸۸/۰۲/۱۵ ۰۹:۰۹:۰۰

متشرکم

جالب بود

نویسنده: فاتحی

تاریخ: ۱۳۸۸/۰۲/۱۵ ۱۲:۱۲:۰۰

البته در استفاده از رویه ذخیره شده سمت `sql` این امکان وجود دارد که رمزعبور توسط SQL Profiler رصد شود!

و کلا پسورد هش نشده به دیتابیس پاس می‌شود.

جا داره منم از زحمات و مطالب شما تشکر و سپاس گذاری کنم.

نویسنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۲/۱۵ ۱۴:۳۷:۰۰

مقایسه نتایج الگوریتم‌های هش کردن اطلاعات در اس کیوال سرور و دات نت

البته کسی که دسترسی به SQL Profiler دارد یعنی دسترسی در حد برنامه نویس یا مدیر سیستم به او اعطاء شده و کلا باید به این دو اطمینان کرد چون در غیر اینصورت واقعاً کارکردن مشکل خواهد شد.

نوبنده: حمیدرضا

تاریخ: ۱۳۸۸/۰۲/۱۵ ۱۷:۵۷:۰۰

اگه اشتباه نفهمیده باشم تعداد بایتها یک کاراکتر توی استاندارد UTF-8 حداقل یک بایته (متغیره)، توی همون لینکی دادید نوشته ۱ تا ۴ بایت.

نوبنده: وحید نصیری

تاریخ: ۱۳۸۸/۰۲/۱۵ ۱۹:۴۸:۰۰

درسته. جدول حداقل و حداکثر رو میشه در سایت زیر هم دید:

http://unicode.org/faq/utf_bom.html

ولی برای این حداقل‌ها و حداکثرها، اما و اگرهای زیادی هست (در مورد عدم تداخل با یکدیگر) که در لینک‌های زیر توضیح داده شده:

<http://en.wikipedia.org/wiki/UTF-8>

<http://tools.ietf.org/html/rfc3629>

رشته‌ها در دات نت فریم ورک از نوع 16-UTF هستند و برای اینکه به صورت صحیحی تبدیل به آرایه‌ای از بایت‌ها شده و در الگوریتم‌های مورد نظر استفاده شوند باید به این نکته دقت داشت.

اس کیوال سرور خود را به 2008 ارتقاء داده‌اید؟ آیا می‌دانید که مجاز هستید از تمامی امکانات جدید آن در دیتابیس‌های موجود خود استفاده کنید یا خیر؟
به همین منظور ابتدا کوئی زیر را اجرا نمائید:

```
Use master;
SELECT name, compatibility_level FROM sys.databases
```

لیست دیتابیس‌های موجود به همراه درجه سازگاری آن‌ها نمایش داده می‌شود. هر کدام که درجه سازگاری مساوی با 100 نداشت یعنی مجاز به استفاده از تمامی امکانات سرور جدید نیست و با آن همانند یک دیتابیس قدیمی بر اساس درجه سازگاری آن رفتار می‌شود.
تفسیر این اعداد مطابق اطلاعات زیر است:

```
60 = SQL Server 6.0
65 = SQL Server 6.5
70 = SQL Server 7.0
80 = SQL Server 2000
90 = SQL Server 2005
100= SQL Server 2008
```

برای تغییر این درجه سازگاری می‌توان از یکی از دستورات T-SQL زیر استفاده کرد:

```
-- Old way:
EXEC sp_dbcmptlevel 'AdventureWorks', 100

-- New way:
ALTER DATABASE 'AdventureWorks' SET COMPATIBILITY_LEVEL = 100 ;
```

که روش اول منسوخ شده در نظر گرفته شده و روش دوم پیشنهاد می‌شود.
برای مثال کوئی زیر عبارات Alter مورد نظر را جهت ارتقاء دیتابیس‌های موجود که درجه سازگاری آن‌ها 100 نیست (هنوز به اس کیوال سرور 2008 ارتقاء پیدا نکرده‌اند)، تولید می‌کند:

```
SELECT 'ALTER DATABASE ' + NAME + ' SET COMPATIBILITY_LEVEL = 100;' TSQL
FROM   sys.databases
WHERE  compatibility_level <> 100
```

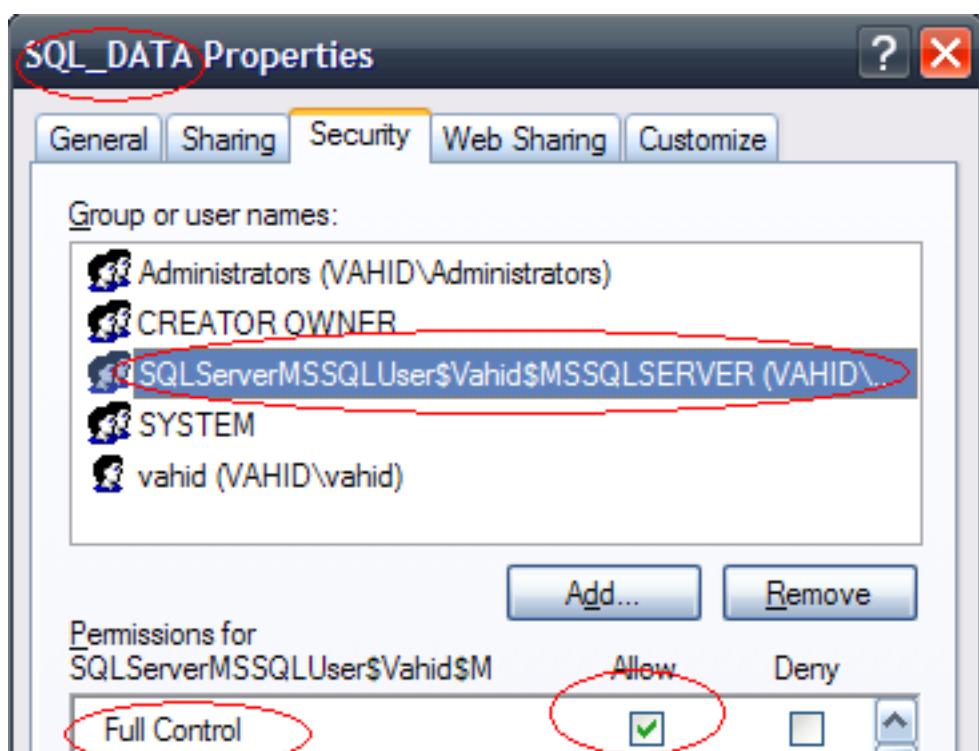
روز قبل نیاز بود تا فایل‌های `ldf` و `mdf` دیتابیس‌ها جابجا شوند (یک هارد بزرگتر و از این مسایل). برای جابجا کردن این فایل‌ها هم روش معمول `detach` و `attach` است. ابتدا روی دیتابیس کلیک راست کرده و `detach`. حالا فایل‌ها را جابجا می‌کنید و سپس `attach`. یا می‌شود بک آپ کامل گرفت و بعد ری استور کرد. عموماً هم نمی‌توان دیتابیس در حال استفاده را `detach` کرد. باید دیتابیس ابتدا `single user` شود و بعد می‌توان این کار را انجام داد.

تا اینجای کار متداول است. همه چیز به خوبی انجام شد. سپس در لحظه `attach`، دیتابیس‌ها به صورت `read only` ا teg شدند با آیکونی سیاه رنگ در `management studio`. (و رنگ من هم بلافاصله به همین رنگ متمایل شد!) بعد از مدتی جستجو مشخص شد که در اس کیوال سرور 2008 برای کاهش سطح حمله به سرور، از یک سری یوزر با دسترسی کم برای نصب اس کیوال سرور استفاده می‌شود (بسیار هم خوب) و اس کیوال سرور 2008، یک سری یوزر مخصوص را هم در هین نصب ایجاد می‌کند که به صورت خودکار بر روی پوشیده دیتابیس شما دسترسی `full control` دارد برای اینکه بتواند کارش را انجام دهد.

حال اگر شما در جای دیگری پوشیده‌ای درست کردید و این دیتابیس‌ها را منتقل نمودید، مجدداً پیش از هر کاری باید این دسترسی را برقرار کنید و گرنه اس کیوال سرور مجوز `write` نخواهد داشت؛ به همین جهت دیتابیس به صورت `read only` در `management studio` با رنگ مشکی ظاهر می‌شود.

نام این کاربر مخصوص به صورت زیر است:

`SQLServerMSSQLUser$ComputerName$MSSQLSERVER`



پس از برقراری دسترسی هم مشکل برطرف نمی‌شود. باید دستور زیر را نیز اجرا نمود:

```
ALTER DATABASE myDB SET READ_WRITE
```

اجرای این دستور نیز، نیاز به حالت `single user` دارد.

پ.ن.

می‌توان دسترسی یوزر سرویس اس کیوال سرور 2008 را نیز مانند نگارش‌های قبلی به حالت `local system` تغییر داد (یا هر اکانت دیگری با دسترسی بالا) تا این مشکلات نباشد؛ ولی بدیهی است سطح حمله به سرور نیز به همین اندازه افزایش می‌یابد.

نظرات خوانندگان

نویسنده: SirAsad | تاریخ: ۱۳۸۸/۰۲/۲۴ ۰۹:۰۲:۰۰

من به شخصه برای انتقال دیتابیس اول میام سرویس دیتابیس رو stop می کنم بعد منتقل میکنم و بعد دوباره سرویس رو Start می کنم . آیا این روش موردی داره ؟
حتی تو یه برنامه مجبور شدم این کار رو توسط برنامه نویسی هم انجام بدم ... چونکه مجبور از فایل دیتابیس کپی بگیرم .

نویسنده: وحید نصیری | تاریخ: ۱۳۸۸/۰۲/۲۴ ۱۰:۲۰:۰۰

- البته منظور از انتقال در اینجا بحث move و copy بود نه cut . بنابراین در حالت move ، سرور به نحوی باید از محل جدید مطلع بشه و نیاز به attach هست.
- ضمنا اگر به دنبال امنیت هستید، کپی کردن mdf و ldf ممکن است کار دست شما بده و یک وقت خیلی ساده attach نشه. ولی بک آپ این مشکل رو نداره.

نویسنده: وحید نصیری | تاریخ: ۱۳۸۸/۰۲/۲۴ ۱۴:۱۶:۰۰

این عدم دسترسی ممکن است هنگام attach سبب بروز خطاهای زیر هم شود (جهت تکمیل بحث):
Database cannot be upgraded because it is read-only or has read-only files. Make the database or files - writeable, and rerun recovery

CREATE FILE encountered operating system error 5 (failed to retrieve text for this error. Reason: 1815) while - attempting to open or create the physical file

گاهی از اوقات یک سری از امکانات جدید در دسترس هستند اما فراموش می‌شوند. برای مثال روش یافتن رکوردهای غیر یکسان دو جدول یکسان. مثلاً یک دیتابیس قدیمی دارید دقیقاً مشابه دیتابیس کاری فعلی با همان ساختار (ری استور شده از یک بک آپ). اکنون می‌خواهید بدانید در طول این مدت چه رکوردهایی به دیتابیس کاری اضافه شده که در دیتابیس قدیمی ری استور شده موجود نیست و کلاً کدام رکوردها با هم متفاوتند. چه باید کرد؟

مثال:

دو جدول موقتی یکسان زیر را در نظر بگیرید.

```

CREATE TABLE #tableA
(
column1  INT,
column2  INT
)

INSERT INTO #tableA
VALUES
(1,1)
,(1, 2)
,(1, 3)
,(2, 1)

SELECT column1,
       column2
FROM    #tableA

CREATE TABLE #tableB
(
column1  INT,
column2  INT
)

INSERT INTO #tableB
VALUES
(1,1)
,(1, 3)
,(2, 2)

SELECT column1,
       column2
FROM    #tableB

```

یک سری دیتای دلخواه به آن‌ها اضافه شده است. (از روش اضافه کردن چندین رکورد توسط [یک عبارت insert](#) که در اس کیوال سرور 2008 معرفی شده، استفاده گردیده است)

```

#tableA
column1  column2
-----  -----
1          1
1          2
1          3
2          1

#tableB
column1  column2
-----  -----
1          1
1          3
2          2

```

اکنون می‌خواهیم رکوردهایی از جدول A را که در جدول B نیستند، پیدا کنیم. روش متدائل انجام این‌کار در اس کیوال سرور 2000 به صورت زیر است:

```
SELECT column1,  
       column2  
  FROM  #tableA  
 WHERE NOT EXISTS (  
           SELECT *  
             FROM  #tableB  
            WHERE #tableA.column1 = #tableB.column1  
                  AND #tableA.column2 = #tableB.column2  
      )
```

column1	column2
1	2
2	1

و یا روش زیباتر انجام این‌کار که از اس کیوال سرور 2005 به بعد معرفی شده، به صورت زیر می‌باشد:

```
SELECT column1, column2 FROM #tableA  
EXCEPT  
SELECT column1, column2 FROM #tableB
```

column1	column2
1	2
2	1

Except

رکوردهای منحصر‌بفردی از کوئری سمت چپ را که در کوئری سمت راست وجود ندارند، بر می‌گرداند.
در این حالت تعداد ستون‌های در نظر گرفته شده برای مقایسه باید یکسان و یک نوع باشند.

همچنین اگر می‌خواهید رکوردهایی از جدول A را که در جدول B وجود دارند بیابید، می‌توان از [intersect](#) استفاده کرد.

نظرات خوانندگان

نوبسند: Anonymous
تاریخ: ۰۵:۵۴:۰۴ ۱۳۸۸/۰۳/۰۵

جالب بود ولی این ها فقط برای SQL Server نیستند. بلکه دستورات کلی SQL هستند!

نوبسند: وحید نصیری
تاریخ: ۰۹:۴۳:۵۰ ۱۳۸۸/۰۳/۰۵

من عموما مطالبی رو که توسط یک ناشناس طرح میشه رو حذف میکنم. محض اطلاع.

+

اپراتور except جزو استاندارد ANSI SQL است ولی تا اس کیوال سرور 2005 در این محصول پشتیبانی نمیشود. (درک ضمنی این مطلب از مقاله فوق کار مشکلی نیست)

علاوه بر فشرده سازی خودکار بک آپ‌ها که پیشتر در مورد آن‌ها صحبت شد، اس کیوال سرور 2008 دو نوع فشرده سازی دیگر را نیز پشتیبانی می‌کند:

: Row Compression

حالت row compression نحوه ذخیره سازی فیزیکی داده‌ها را تغییر می‌دهد. فعال سازی آن اثرات زیر را خواهد داشت:

- (الف) متادیتای هر رکورد را حداقل می‌کند (منظور از متادیتا اطلاعاتی مانند اطلاعات ستون‌ها، طول و آفست و غیره است)
- (ب) داده‌های عددی و رشته‌هایی با طول ثابت، به صورت اطلاعاتی با طول متغیر ذخیره خواهند شد، درست مانند varchar ها.

برای ایجاد جدولی که row compression در آن به صورت پیشفرض فعال است، می‌توان مانند مثال زیر عمل کرد:

```

CREATE TABLE MyTable
(
    ID int identity Primary key,
    Name char(100),
    Email char(100)
)
WITH (DATA_COMPRESSION = Row);

GO
  
```

و اگر جدول موجودی را می‌خواهید تغییر داده و این خاصیت را بر روی آن فعال نمایید، روش زیر را اعمال کنید:

```
Alter TABLE MyTable REBUILD WITH (DATA_COMPRESSION=Row, MAXDOP=2);
```

در اینجا MAXDOP مشخص می‌کند که از چند CPU باید برای فشرده سازی استفاده شود. (اگر جدولی حجمی دارد، به این صورت می‌توان عملیات فشرده سازی را سریعتر به پایان رساند)

: Page Compression

در روش دوم فشرده سازی اطلاعات در اس کیوال سرور 2008، که مهمترین حالت موجود نیز می‌باشد، اطلاعات مشترک، بین سطرهای یک صفحه به اشتراک گذاشته می‌شوند. این روش از فناوری‌های زیر استفاده می‌کند:

- (الف) روش row compression که در مورد آن صحبت شد جزئی از این روش است.
- (ب) Prefix Compression : به ازای هر ستون در یک صفحه، Prefix های تکراری یافت شده و در هدر مخصوص فشرده سازی ذخیره می‌شوند (محل این هدر پس از هدر صفحه است). سپس هرجایی که به این Prefix ها اشاره شده باشد، عدد منحصر بفرد شناسایی کننده آن‌ها نسبت داده می‌شود.
- (ج) Dictionary Compression : در این حالت مقادیر تکراری یک صفحه جستجو شده و در هدر فشرده سازی صفحه ذخیره می‌شوند. حالت Prefix Compression فقط به یک ستون منحصر می‌شود اما Dictionary Compression به کل صفحه اعمال می‌گردد.

برای فعال سازی آن در یک جدول جدید به روش زیر می‌توان عمل نمود:

```

CREATE TABLE MyTable
(
    ID int identity Primary key,
    Name char(100),
    Email char(100)
)
WITH (DATA_COMPRESSION = Page);
  
```

```

    Name char(100),
    Email char(100)
)
WITH (DATA_COMPRESSION = Page);

```

و برای اعمال آن به جدولی موجود از روش زیر می‌توان استفاده کرد:

```
Alter TABLE MyTable REBUILD WITH (DATA_COMPRESSION=Page, MAXDOP=2);
```

یک سری رویه‌های ذخیره شده سیستمی جدید نیز برای محاسبه حجم جداول، پیش و پس از فشرده سازی (بدون فشرده سازی واقعی) نیز در این نگارش گنجانده شده‌اند که به شرح زیر هستند:

```
-- بررسی اینکه چه میزان فضای با اعمال فشرده سازی صفحات قابل صرفه جویی خواهد بود
EXEC sp_estimate_data_compression_savings 'schemaname', 'TableName', NULL, NULL, 'PAGE';

-- بررسی اینکه چه میزان فضای با اعمال فشرده سازی ردیف‌ها قابل صرفه جویی خواهد بود
EXEC sp_estimate_data_compression_savings 'schemaname', 'TableName', NULL, NULL, 'ROW';
```

بنابراین قبل از اینکه فشرده سازی را فعال نمایید، ابتدا بررسی کنید آیا واقعاً میزان قابل توجهی اطلاعات فشرده خواهند شد و نتیجه حاصل رضایت‌بخش است یا خیر. همچنین باید درنظر داشت که جداول و یا ایندکس‌هایی که read و write بالای دارند برای این منظور مناسب نیستند. برای یافتن آن‌ها کوئری زیر را اجرا کنید:

```

USE dbName;
SELECT objectname = OBJECT_NAME(s.object_id),
       indexname = i.name,
       i.index_id,
       reads = range_scan_count + singleton_lookup_count,
       'leaf_writes' = leaf_insert_count + leaf_update_count + leaf_delete_count,
       'leaf_page_splits' = leaf_allocation_count,
       'nonleaf_writes' = nonleaf_insert_count + nonleaf_update_count +
                           nonleaf_delete_count,
       'nonleaf_page_splits' = nonleaf_allocation_count
FROM   sys.dm_db_index_operational_stats (DB_ID(), NULL, NULL, NULL) AS s
INNER JOIN sys.indexes AS i
        ON i.object_id = s.object_id
WHERE  OBJECTPROPERTY(s.object_id, 'IsUserTable') = 1
       AND i.index_id = s.index_id
ORDER BY
       leaf_writes DESC,
       nonleaf_writes DESC

```

و جهت تکمیل مبحث می‌توان به مقاله بسیار جامع زیر که اخیراً توسط مایکروسافت منتشر شده است رجوع نمود:

[Data Compression: Strategy, Capacity Planning and Best Practices](#)

SQL Server express edition

نگارش مجازی و ساده شده اس کیوال سرور است. این نگارش مجازی فاقد SQL Server agent برای زمان بندی انجام امور تکراری، برای مثال تهیه بک آپ‌های خودکار است. این مورد در کل ایرادی محسوب نمی‌شود زیرا می‌توان این عملیات را با استفاده از سیستم استاندارد scheduled tasks ویندوز نیز پیاده سازی کرد.

برنامه خط فرمان سورس بازی به نام [ExpressMaint](#) موجود است که می‌تواند از دیتابیس‌های اس کیوال سرور اکسپرس (و غیر اکسپرس) بک آپ تهیه کند. فقط کافی است این برنامه را به عنوان یک scheduled task ویندوز معرفی کنیم تا در زمان‌های تعیین شده در مکان‌های مشخص، بک آپ تهیه کند. همچنین این برنامه فایل‌های بک آپ تهیه شده را نیز تعیین اعتبار می‌کند. با پارامترهای خط فرمان آن [در اینجا](#) می‌توانید آشنا شوید. خلاصه کاربردی آن را به صورت چند دستور در ادامه مرور خواهیم کرد.

الف) یک فایل bat را با محتوای زیر درست کنید :

```
C:\backup\expressmaint.exe -S (local)\sqlexpress -D ALL_USER -T DB -R C:\backup -RU WEEKS -RV 2 -B  
C:\backup -BU DAYS -BV 2 -V -C
```

توضیحات: در این فایل bat ، مسیر فایل اجرایی برنامه حتماً باید دقیقاً ذکر شود و گرنۀ scheduled task ویندوز درست کار نخواهد کرد. همچنین instance اس کیوال سرور اکسپرس در اینجا local\sqlexpress فرض شده است. این دستور از تمامی دیتابیس‌های غیرسیستمی در مسیر C:\backup می‌گیرد (به ازای هر دیتابیس یک پوشه مجزا درست خواهد کرد و هر فایل را بر اساس تاریخ و ساعت مشخص می‌سازد). همچنین لاغ عملیات را نیز در همان پوشه تهیه می‌کند (نتیجه اعتبار سنجی صورت گرفته بر روی بک آپ‌های تهیه شده در این فایل‌ها ثبت می‌شود). مطابق پارامترهای بکار گرفته شده، بک آپ‌های قدیمی‌تر از دو روز به صورت خودکار حذف شده و لاغ فایل‌ها به مدت 2 هفته نگهداری می‌شوند.

ب) برای اجرای زمان بندی شده این فایل bat تهیه شده، دستورات زیر را در خط فرمان اجرا کنید (فرض بر این است که فایل bat تهیه شده در مسیر مشخص شده C:\backup\backup.bat قرار دارد) :

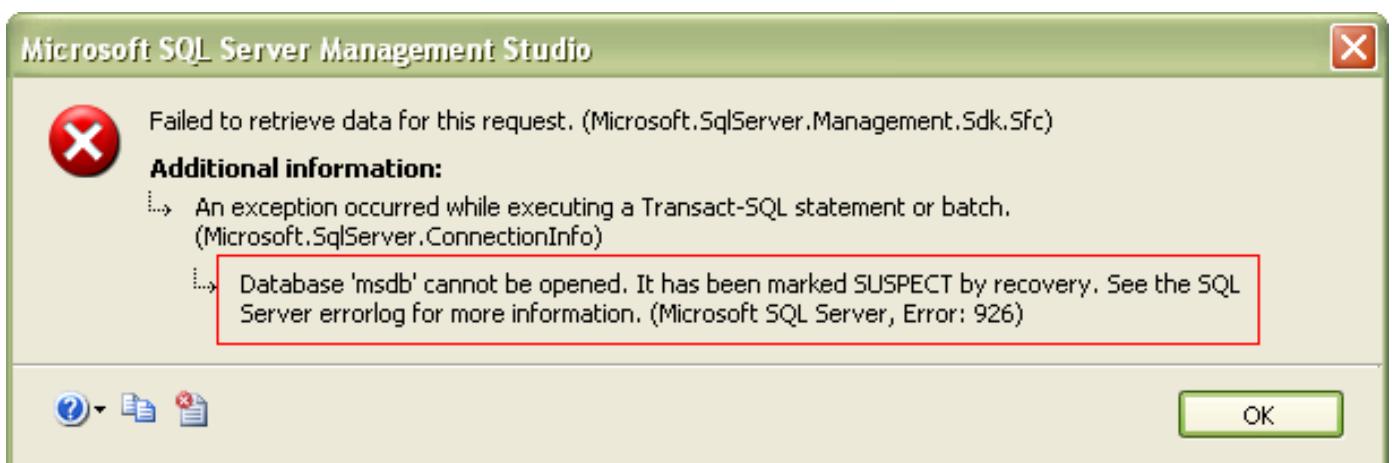
```
AT 23:30 /EVERY:m,t,w,th,f,s,su C:\backup\backup.bat  
AT 11:30 /EVERY:m,t,w,th,f,s,su C:\backup\backup.bat
```

به این صورت ویندوز هر روز، دوبار در طول روز از کلیه دیتابیس‌ها به صورت خودکار بک آپ تهیه می‌کند.

روشی که در اینجا ذکر شد منحصر به نگارش express نیست و با کلیه نگارش‌های SQL Server سازگار است.

عنوان: بازسازی msdb تخریب شده
نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۴/۲۸ ۱۲:۲۷:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: SQL Server

حاصل قطع برق و یا یک ری استارت دستی ناصحیح را در نظر بگیرید:



Database 'msdb' cannot be opened. It has been marked SUSPECT by recovery. See the SQL Server errorlog for more information. (Microsoft SQL Server, Error: 926)

Msdb از نوع دیتابیس‌های سیستمی است و نمی‌شود مطابق روال متداول دیتابیس‌های SUSPECT شده آنرا بازیابی کرد. این روش متداول به صورت زیر است:

```
ALTER DATABASE DBName SET EMERGENCY
DBCC checkdb('DBname')
ALTER DATABASE DBName SET SINGLE_USER WITH ROLLBACK IMMEDIATE
DBCC CheckDB ('DBName', REPAIR_ALLOW_DATA_LOSS)
ALTER DATABASE DBName SET MULTI_USER
```

در ابتدای کار دیتابیس در حالت اورژانسی قرار می‌گیرد. بعد وضعیت و میزان تخریب نمایش داده شده، سپس تک کاربره می‌شود. در ادامه به اس کیوال سرور اجازه داده می‌شود که دیتابیس را با هر وضعی (حتی به قیمت از دست رفتن تعدادی رکورد) ترمیم کند و در آخر دیتابیس مجدداً به حالت چند کاربره بازگشت داده می‌شود. این روشی است که سال قبل با قطعی‌های مکرر برق زیاد کاربرد داشت.

اما دیتابیس سیستمی msdb را نمی‌شود در حالت اورژانسی قرار داد؛ بنابراین باید به دنبال راه چاره‌ی دیگری بود. پس از مدتی جستجو در وبلاگ‌های msdn، راه حل زیر یافت شد و کاملاً عملی است (تست شده!):

روش زیر در مورد اس کیوال سرور 2008، 2005 و حتی 2000 نیز قابل استفاده است.

ابتدا خونسردی خودتان را حفظ کنید! الان فقط دیگر با management studio نمی‌توانید دیتابیس‌ها را مرور کنید و همچنین تمام job های تعریف شده شما نابود شده‌اند! اما سرور به کار عادی خودش می‌تواند ادامه دهد. سپس :

الف) تمام سرویس‌های مربوط به اس کیوال سرور را stop کنید. به کنسول سرویس‌ها مراجعه کرده و هر آنچه که در نام آن sql را مشاهده می‌کنید، stop کنید.

ب) با استفاده از خط فرمان، ابتدا به مسیر زیر وارد شوید:

```
cd "C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Binn\"
```

و سپس دستور زیر را اجرا نمایید:

```
start sqlservr.exe -c -m -T3608
```

به این ترتیب اس کیوال سرور در یک حالت حداقل که بتوان دیتابیس msdb تخریب شده را detach کرد راه اندازی می‌شود.

(پرچم 3608 مجوز detach کردن این دیتابیس را می‌دهد)

ج) این management studio را اجرا کنید. زمانیکه پنجره کانکت ظاهر می‌شود آنرا کنسل کرده و در نوار ابزار بالای صفحه روی دکمه new query کیک کنید (چون حالت راه اندازی سرور در حالت تک کاربره است نمی‌خواهیم اتصال دیگری برقرار شود و در کار اخلال کند). با کلیک بر روی new query پنجره connect to server ظاهر می‌شود. در همین پنجره بر روی دکمه options کلیک کرده در برگه connection properties در قسمت master نام connect to database را وارد نمود و اکنون بر روی دکمه connect کلیک نمایید.

ج) سپس دستور زیر را وارد کنید تا دیتابیس msdb را بتوان detach کرد.

```
Use master;
sp_detach_db 'msdb'
```

مراحلی که عنوان شد مهم است. اگر به این صورت عمل نکنید با پیغام خطای زیر مواجه خواهید شد:

```
Cannot detach an opened database when the server is in minimally configured mode
```

اگر به این خطا برخوردید، یکبار دیگر از صفر شروع کنید. تمام سرویس‌های مرتبط با sql را استاپ کنید (حتی در صورت نیاز کارت شبکه سرور را نیز غیرفعال کنید). و از مرحله الف مجدد شروع نمایید تا حتماً حالت تک کاربره اتصال برقرار شود.

(همچنین پنجره کوئری جدیدی را نیز باز نکنید چون در این حالت فقط و فقط یک اتصال مجاز است)

تا اینجا موفق شدیم که دیتابیس msdb را detach کنیم. اکنون به پوشیده دیتابیس‌ها مراجعه کرده و mdf و ldf این دیتابیس تخریب شده را rename کنید (به هر اسمی که مایل بودید).

د) اکنون نوبت بازسازی مجدد این دیتابیس است.

محفویات فایل instmsdb.sql را که در مسیر C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\install قرار دارد، در پنجره کوئری تک کاربره‌ای که در مرحله قبل بازکرده‌ایم، copy/paste کرده و دکمه F5 را فشار دهید. پس از مدتی دیتابیس msdb باز سازی شده و مشکل برطرف می‌شود.

۵) اکنون سرور را stop و start کنید یا کلا کامپیوتر سرور را restart کنید تا تمامی سرویس‌های stop شده راه اندازی مجدد شوند.

نظرات خوانندگان

نویسنده: نیما
تاریخ: ۱۳۸۸/۰۴/۲۸ ۱۶:۲۴:۴۲

سلام استاد نصیری
باز هم یک نکته بی نظیر. از اینکه ما رو هم در دانسته هاتون شریک میکنین ممنونم

نویسنده: mojtaba
تاریخ: ۱۳۸۸/۰۷/۱۲ ۱۷:۳۵:۰۷

سلام
من در همان مرحله اول با این پیغام مواجه میشم:
error getting instance ID from name

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۷/۱۲ ۱۸:۱۳:۵۵

سلام
دو علت می‌توانه داشته باشند
۱- یوزری که اس کیوال سرور تحت آن اجرا می‌شود دسترسی به مسیر رجیستری زیر ندارد:
HKLM\Software\Microsoft\Microsoft SQL Server\Instance Names
۲- نام و هله (InstanceName) مورد نظر را باید دقیقاً ذکر کنید:
start sqlservr.exe -c -SServername\InstanceName -T3608

چندین روش برای انجام مقایسه حساس به حروف کوچک و بزرگ (case sensitive) در SQL Server وجود دارد که در ادامه آن‌ها را مرور خواهیم کرد:

ابتدا جدول موقتی زیر را جهت آزمایشات بعدی در نظر بگیرید

```
CREATE TABLE #tblTest
(
    f1 NVARCHAR(50)
)
INSERT INTO #tblTest (f1) VALUES('Test1')
INSERT INTO #tblTest (f1) VALUES('TEST1')
```

الف) استفاده از collation صحیح

عموماً هنگام نصب اس کیوال سرور از collation غیرحساس به کوچکی و بزرگی حروف استفاده می‌شود و این مورد سبب می‌شود که پیش فرض ایجاد دیتابیس‌ها نیز به همین صورت باشد (هر چند کاملاً قابل کنترل و تنظیم است). به صورت پویا می‌توان این collation را در کوئری‌ها نیز اعمال نمود. برای مثال:

```
SELECT f1 FROM #tblTest WHERE f1 COLLATE SQL_Latin1_General_CI_AS = 'Test1'
```

در این مثال اگر از collation حساس به کوچکی و بزرگی حروف استفاده نمی‌شد، خروجی هر دو رکورد ثبت شده می‌بود.

ب) استفاده از تابع BINARY_CHECKSUM اس کیوال سرور (نوعی الگوریتم ویژه، شبیه به امضای دیجیتال و هش کردن اطلاعات است)

```
SELECT f1 FROM #tblTest WHERE BINARY_CHECKSUM(f1) = BINARY_CHECKSUM('Test1')
```

و یا استفاده از امکانات هش کردن اطلاعات در اس کیوال سرور: (امضای دیجیتال دو رشته با هم مقایسه می‌شوند)

```
SELECT f1 FROM #tblTest WHERE hashbytes('md5',f1) = hashbytes('md5',N'Test1')
```

ج) مقایسه در حالت بایناری

```
SELECT f1 FROM #tblTest WHERE convert(varbinary(50),f1) = convert(varbinary(50),N'Test1')
```

متن زیر یک سری نکات و یا شاید توهمناتی را مطرح می‌کند که در مورد رویه‌های ذخیره شده در اس کیوال سرور رایج هستند.

۱- رویه‌های ذخیره شده در مقابل SQL Injection مقاوم هستند. کوئری‌های Ad hoc همیشه این آسیب پذیری را به همراه دارند. نادرست است! رویه‌های ذخیره شده‌ای که رشتہ‌ها را به صورت پارامتر دریافت کرده و آن‌ها را به صورت یک عبارت `sql` اجرا می‌کنند، آسیب پذیر هستند. اگر هنگام استفاده از کوئری‌های Ad hoc از پارامترها استفاده شود، در برابر حملات SQL Injection مصنوع خواهد بود.

۲- رویه‌های ذخیره شده کش می‌شوند اما این Plan برای کوئری‌های Ad hoc هر بار محاسبه و تولید می‌گردد. نادرست است! اس کیوال سرور تا این اندازه بی‌هوش نیست! اگر execution plan ای موجود باشد حتماً استفاده خواهد شد و برای موتور اس کیوال سرور اصلاً اهمیتی ندارد که کوئری در حال اجرا از یک رویه ذخیره شده صادر شده است یا از یک کوئری Ad hoc . رویه‌های ذخیره شده پیش کامپایل شده نیستند و مانند تمامی کوئری‌های دیگر در زمان اجرا کامپایل می‌شوند.

۳- زمانیکه از رویه ذخیره شده استفاده می‌کنید همه چیز را در یک مکان به صورت متمرکز و مجتمع خواهید داشت (مدیریت بهتر)

نادرست است! در یک مکان متمرکز در اختیار شما نیستند. برنامه جای خود را دارد و رویه‌های ذخیره شده در دیتابیس در جای دیگری قرار دارند و برای مثال اگر قرار باشد یک پارامتر را به رویه ذخیره شده خود اضافه کنید، کدهای شما نیز باید تغییر کنند.

۴- می‌توان از یک رویه ذخیره شده استفاده مجدد کرد (در نقاط مختلف یک کد) و اعمال تغییرات تنها در یک مکان (دیتابیس) باید صورت گیرد.

هر چند این مورد درست است، اما باید دقت داشت که اگر چندین برنامه از این رویه ذخیره شده استفاده می‌کنند نباید تغییرات شما باعث از کار افتادن سایر برنامه‌ها شوند.

۵- می‌توان رویه ذخیره شده را بدون نیاز به توزیع مجدد برنامه تغییر داد.
این مورد تا حدودی صحیح است. اگر تنها بحث بهینه سازی و امثال آن مطرح باشد صحیح است اما اگر واقعاً نیاز به تغییر یک کوئری در رویه ذخیره شده وجود داشته باشد به احتمال زیاد برنامه نیز باید دستخوش تغییراتی گردد تا این دو با هم هماهنگ شوند.

نظر شما چیست؟

نظرات خوانندگان

نویسنده: نیما

تاریخ: ۲۱:۵۳:۳۱ ۱۳۸۸/۰۵/۱۳

سلام استاد نصیری

در رابطه با نکته دوم من فکر کنم که هر بار هر کوئری میاد پارس بشه و بعد برash Execution Plan در نظر گرفته بشه. یادمde خودم امتحان کردم استور سریعتر بود. به شخصه ترجیح میدم که داخل برنامه از دستورات اس کیو ال استفاده نکنم چون مدیریتش سخت تره ممنون از مطلبتون

نویسنده: محمد

تاریخ: ۲۳:۱۴:۳۳ ۱۳۸۸/۰۵/۱۳

مرسی از مطلبتون،

منظورتون از کوئری‌های Ad Hoc کوئری‌هایی است که در زمان اجرا ساخته می‌شوند یا در سمت کلاینت قرار دارند هست؟ یا تعریف دیگری دارد؟

در مورد نکته‌ی دوم من این مطلب رو جای خوندم که کوئری‌های Ad Hoc هر بار کامپایل میشن و Exec Plan شون بدست میاد و.... البته اون مطلب در مورد 2000 SQL Server بود.

نویسنده: وحید نصیری

تاریخ: ۲۳:۴۵:۲۴ ۱۳۸۸/۰۵/۱۳

بله. به کوئری‌هایی گفته می‌شود که توسط برنامه بر اساس نیاز آنی کاربر ساخته می‌شوند.

http://it.toolbox.com/wiki/index.php/Ad_hoc_query

البته اینجا بیشتر منظور همان کوئری‌های مخلوط با سورس کدهای شماست.

نویسنده: افشار محبی

تاریخ: ۰۷:۴۶:۰۴ ۱۳۸۸/۰۵/۱۴

من شخصاً ترجیح می‌دهم به جای استفاده از SP و امثال‌هم از یک ORM خوب مثل NHibernate استفاده کنم. این طوری می‌توانم بخش زیادی از منطق را در داخل زبان مورد استفاده مثل C# پیاده‌سازی کنم.

نویسنده: Anonymous

تاریخ: ۱۰:۴۹:۳۲ ۱۳۸۸/۰۵/۱۴

منم با آقای محبی موافقم. با این تفاوت که LINQ رو NHibernate ترجیح میدم.

با SP ها درسته که تقرایبا کارها مدیریت بهتری دارن ولی مثلا زمانی که لازم باشه SP رو تغییر بدهید باید خیلی چیزها باز بینی بشن.

این مشکل وقتی با DataSet ها کار می‌کنی بیشتر هم میشه.

موفق باشید

نویسنده: وحید نصیری

تاریخ: ۱۳:۳۱:۱۸ ۱۳۸۸/۰۵/۱۵

چند مورد را در مورد نکته دوم جهت تکمیل بحث اضافه کنم:

در اس کیوال سرور 2000 اگر از ad hoc کوئری استفاده شود احتمال recompile شدن بسیار زیاد است اما این مورد از اس کیوال سرور 2005 به بعد به شدت بهبود یافته. برای مطالعه بیشتر

Execution Plan Caching and Reuse

<http://technet.microsoft.com/en-us/library/ms181055%28SQL.90%29.aspx>

نکته مهمی که در ad hoc کوئری‌ها ممکن است سبب recompile شدن plan اجرایی آن شود بحث تغییر نوع یا اندازه‌ی پارامترهای مورد استفاده است. اگر این موارد به صورت صریح ذکر شوند و از پارامترهای ADO.Net استفاده شوند، تشخیص نوع و اندازه پارامترها برای اس کیوال سرور بسیار ساده شده و حتماً از کش بجای recompile استفاده خواهد کرد. برای مطالعه بیشتر:

Parameters and Execution Plan Reuse

<http://technet.microsoft.com/en-us/library/ms175580%28SQL.90%29.aspx>

دقیقاً همین نکته را اگر از NHibernate استفاده می‌کنید نیز باید رعایت کنید:

NHibernate queries & sql server execution plans

<http://testdrivendev.wordpress.com/2009/03/10/nhibernate-queries-sql-server-execution-plans>

خلاصه این مقاله به این صورت است که اس کیوال سرور را وادار نکنیم که از recompile به جهت مشخص نبودن طول یا اندازه پارامترها، هر بار استفاده نماید.

نویسنده: Saeed Zarinfam

۰۸:۳۵:۱۰ ۱۳۸۸/۰۵/۱۸

تاریخ:

سلام

به چه مبحث خوبی اشاره کردید.

یکی دیگر از نکاتی که مهم است، از یکپارچگی درآمدن تیم پیاده سازی است که از لحاظ مدیریت پروژه بسیار مهم است. بدین معنی که اعضای تیم پروژه باید مفهوم جدیدی به نام sp را آموخت و بینند که این امر باعث کاهش سرعت توسعه می‌شود در حالی که با استفاده از یک ORM برنامه نویسان در لایه DAL با همان زبان برنامه نویسی که در لایه BLL برنامه می‌نویسنند کار می‌کنند. از طرف دیگر استفاده از sp ممکن است شما را به استفاده از پایگاه داده خاصی محدود کند.

همانطور که [مطلع هستید](#) قابلیت تهیه عبارات Insert از جداول یک دیتابیس، به صورت استاندارد به 2008 اضافه شده است. برای استفاده از این قابلیت از طریق برنامه نویسی به صورت زیر می‌توان عمل نمود:

الف) سه ارجاع را به اسمبلی‌های زیر اضافه نمایید:

```
Microsoft.SqlServer.ConnectionInfo
Microsoft.SqlServer.Management.Sdk.Sfc
Microsoft.SqlServer.Smo
```

ب) اکنون کدی که عملیات Script Data را با استفاده از قابلیت‌های SMO ارائه می‌دهد به صورت زیر خواهد بود:

```
using System;
using System.Text;
using Microsoft.SqlServer.Management.Smo;
using Microsoft.SqlServer.Management.Common;
using Microsoft.SqlServer.Management.Sdk.Sfc;

///<summary>
///<summary> تهیه اسکریپت رکوردها
///</summary>
///<param name="dbName">نام دیتابیس مورد نظر</param>
///<param name="table">نام جدولی که باید اسکریپت شود</param>
///<param name="instance">وهله سرور</param>
///<param name="userName">نام کاربری جهت اتصال</param>
///<param name="pass">کلمه عبور جهت اتصال به سرور</param>
///<returns>اسکریپت تهیه شده</returns>
public static string ScriptData(string dbName, string table, string instance, string userName,
string pass)
{
    try
    {
        ServerConnection serverConnection = new ServerConnection(instance, userName, pass);
        Server server = new Server(serverConnection);
        Database database = server.Databases[dbName];
        if (database == null) return string.Empty;
        Table tb = database.Tables[table];
        Scripter scripter = new Scripter(server) { Options = { ScriptData = true } };
        if (tb == null) return string.Empty;
        StringBuilder sb = new StringBuilder();
        foreach (string s in scripter.EnumScript(new Urn[] { tb.Urn }))
            sb.AppendLine(s);
        return sb.ToString();
    }
    catch (Exception ex)
    {
        //todo: log ...
        return string.Empty;
    }
}
```

نکته:

اسمبلی‌های SMO به همراه مجموعه SQL Server 2008 نصب می‌شوند. بنابراین متدهای برنامه‌ی فوق بر روی سروری با این مشخصات بدون مشکل اجرا خواهد شد. اما اگر قصد توزیع برنامه خود را دارید باید [Microsoft SQL Server 2008 Management Objects](#) را نیز به همراه برنامه خود نصب نمایید.

هر برنامه نویسی در طول عمر کاری خود حداقل یکبار با این مساله مواجه خواهد شد: "چگونه یک سری رکورد تکراری موجود را باید حذف کرد؟"

ابتدا جدول موقتی زیر را که در آن یک سری رکورد تکراری ثبت شده است در نظر بگیرید:

```

CREATE TABLE #Employee
(
ID      INT,
FIRST_NAME NVARCHAR(100),
LAST_NAME NVARCHAR(300)
)

INSERT INTO #Employee VALUES ( 1, 'Vahid', 'Nasiri' );
INSERT INTO #Employee VALUES ( 2, 'name1', 'lname1' );
INSERT INTO #Employee VALUES ( 3, 'name2', 'lname2' );
INSERT INTO #Employee VALUES ( 2, 'name1', 'lname1' );
INSERT INTO #Employee VALUES ( 3, 'name2', 'lname2' );
INSERT INTO #Employee VALUES ( 4, 'name3', 'lname3' );

```

روش‌های حذف رکوردهای تکراری جدول موقتی Employee

الف) استفاده از یک جدول موقتی دیگر بر اساس تمامی فیلدهای موجود

```

SELECT DISTINCT *
FROM #Employee

SELECT * INTO #DuplicateEmployee
FROM #Employee

INSERT #DuplicateEmployee
SELECT DISTINCT *
FROM #Employee

BEGIN TRAN
DELETE #Employee
INSERT #Employee
SELECT *
FROM #DuplicateEmployee

COMMIT TRAN

DROP TABLE #DuplicateEmployee

SELECT DISTINCT *
FROM #Employee

```

ب) استفاده از یک جدول موقتی دیگر بر اساس یک مجموعه از فیلدهای موجود

```

SELECT DISTINCT * FROM #Employee

SELECT * INTO #DuplicateEmployee FROM #Employee

INSERT #DuplicateEmployee
SELECT ID,
       FIRST_NAME,
       LAST_NAME

```

روش‌هایی برای حذف رکوردهای تکراری

```
FROM      #Employee
GROUP BY
    ID,
    FIRST_NAME,
    LAST_NAME
HAVING COUNT(*) > 1

BEGIN TRAN
DELETE #Employee
FROM   #DuplicateEmployee
WHERE  #Employee.ID = #DuplicateEmployee.ID
    AND #Employee.FIRST_NAME = #DuplicateEmployee.FIRST_NAME
    AND #Employee.LAST_NAME = #DuplicateEmployee.LAST_NAME

INSERT #Employee
SELECT *
FROM   #DuplicateEmployee

COMMIT TRAN
DROP TABLE #DuplicateEmployee

SELECT DISTINCT * FROM  #Employee
```

ج) استفاده از rowcount بر اساس یک مجموعه از فیلد های موجود

```
SELECT DISTINCT *
FROM   #Employee

SET ROWCOUNT 1
SELECT 1
WHILE @@rowcount > 0
    DELETE #Employee
    WHERE 1 < (
        SELECT COUNT(*)
        FROM  #Employee a2
        WHERE #Employee.ID = a2.ID
            AND #Employee.FIRST_NAME = a2.FIRST_NAME
            AND #Employee.LAST_NAME = a2.LAST_NAME
    )
SET ROWCOUNT 0

SELECT DISTINCT *
FROM   #Employee
```

د) استفاده از Analytical Functions بر اساس یک مجموعه از فیلد های موجود

```
SELECT DISTINCT *
FROM   #Employee;

WITH #DeleteEmployee AS (
    SELECT ROW_NUMBER()
    OVER(PARTITION BY ID, First_Name, Last_Name ORDER BY ID) AS
    RNUM
    FROM  #Employee
)

DELETE
FROM   #DeleteEmployee
WHERE  RNUM > 1

SELECT DISTINCT *
FROM   #Employee
```

۵) استفاده از یک فیلد identity جدید بر اساس یک مجموعه از فیلد های موجود

```
SELECT DISTINCT *
```

روش‌هایی برای حذف رکوردهای تکراری

```
FROM #Employee;
ALTER TABLE #Employee ADD UNIQ_ID INT IDENTITY(1, 1)
DELETE
FROM #Employee
WHERE UNIQ_ID < (
    SELECT MAX(UNIQ_ID)
    FROM #Employee a2
    WHERE #Employee.ID = a2.ID
        AND #Employee.FIRST_NAME = a2.FIRST_NAME
        AND #Employee.LAST_NAME = a2.LAST_NAME
)
ALTER TABLE #Employee DROP COLUMN UNIQ_ID
SELECT DISTINCT *
FROM #Employee
```

و در آخر

```
DROP TABLE #Employee
```

نظرات خوانندگان

نوبسند: [salarblog](#) | تاریخ: ۱۳۸۸/۰۵/۲۸ | ۱۵:۵۶:۵۹

بسیار عالی، تشکر.

نوبسند: [مهران زند](#) | تاریخ: ۱۳۸۸/۰۵/۳۱ | ۰۹:۱۱:۴۶

جناب نصیری من دقیقا کد های شما را نخواندمو اما من از تیکه کد زیر برای حذف رکورد های تکراری استفاده میکنم :

```
DELETE  
FROM MyTable  
WHERE ID NOT IN  
(  
SELECT MAX(ID)  
FROM MyTable  
GROUP BY COLUMN1, COLUMN2
```

که در اینجا ستون ها ستون هایی هستند که امکان یونیک بودن داده ها در آن وجود دارد.

با سپاس

اگر به ساز و کار شیرپوینت مایکروسافت دقت کنید، همه چیز را داخل دیتابیس ذخیره می‌کند (از اطلاعات رکوردها گرفته تا فایل‌ها و غیره). حال شاید این سؤال مطرح شود که برای ذخیره سازی فایل‌هایی با تعداد بیش از یک میلیون عدد، استفاده از دیتابیس مناسب است یا فایل سیستم متدائل. برای پاسخ به این سؤال باید به نکات ذیل توجه داشت:

- هر نوع عملیاتی که بر روی فایل‌ها صورت گیرد، بستن، بازکردن و غیره، نیازمند اعمالی در سطح سیستم عامل است (برای مثال بررسی سطح دسترسی لازم برای انجام این کارها).
- هر گونه عملیاتی بر روی فایل‌ها نیازمند یک حداقل قفل گذاری بر روی آن‌ها است که این نیز مصرف CPU قابل توجهی را سبب خواهد شد.
- تمامی اعمال ذکر شده کل سرور و تمامی سرویس‌های در حال اجرا را تحت تاثیر قرار داده و بازدهی آن‌هارا کاهش می‌دهند.
- حتی سیستم عامل‌ها نیز از یک file system database جهت مدیریت اعمال خود استفاده می‌کنند اما این روش برای مدیریت میلیون‌ها و میلیاردها فایل بهینه سازی نشده است.
- ذخیره سازی میلیون‌ها و میلیاردها فایل به تدریج سبب ایجاد fragmentation قابل توجهی شده و این مورد نیز بر روی کارآیی تاثیر منفی خواهد گذاشت (همچنین این مورد بر روی طول عمر تجهیزات ذخیره سازی داده‌ها تاثیر منفی دارد).
- تهیه پشتیبان و بازگرداندن میلیون‌ها فایل بسیار زمانگیر است (برای مثال جابجایی یک فایل یک مگابایتی بسیار سریعتر است از جابجایی 100 فایل 10 کیلوبایتی).
- مدیریت تغییرات و همچنین بررسی اینکه چه شخصی چه فایلی را قرار داده، حذف کرده یا تغییر داده است در حالت استفاده از file system مشکل است.
- به صورت پیش فرض عموماً مباحث replication و امثال آن توسط روش استفاده از file system خصوصاً با تعداد بالای فایل، پشتیبانی نمی‌شود.
- در حالت استفاده از file system، برنامه‌های وب باید دسترسی write بر روی یک سری پوشیده داشته باشند که این مورد همیشه از دیدگاه امنیتی مساله ساز بوده و مشکل آفرین.
- کرش file system مساوی است با کرش سیستم عامل و بازگشت این‌ها زمان برخواهد بود.

با توجه به این نکات استفاده از دیتابیس برای ذخیره سازی تعداد زیادی فایل، مزایای زیر را به همراه خواهد داشت:

- اکثر سیستم‌های دیتابیسی امروزی برای کار با حجم عظیمی از داده‌ها به حد بلوغ خود رسیده‌اند.
- هنگام استفاده از دیتابیس برای ذخیره سازی فایل‌ها دیگر سر و کار ما با میلیون‌ها فایل نخواهد بود و حداقل چند فایل دیتابیس و ملحقات آن مانند لاغ فایل، کل سیستم را تشکیل می‌دهند.
- فایل‌های دیتابیس برای مثال SQL Server، همیشه توسط SQL در حالت باز قرار داشته و مباحث قفل‌گذاری بر روی فایل‌های دیتابیس و بررسی سطح دسترسی و غیره توسط سیستم عامل در اینجا به حداقل خود می‌رسد.
- در این حالت بار سیستم عامل شما تنها سیستم است که مشغول سرویس دهی اطلاعات دیتابیس‌های شما است.
- جستجوی فایل‌ها، حتی جستجو در محتوای این فایل‌های ذخیره شده در یک دیتابیس بسیار سریعتر از روش full text search می‌باشد. امکان استفاده از کوئری‌های SQL انعطاف‌پذیری خاصی را به این سیستم‌ها خواهد داد (برای مثال قابلیت search مربوط به SQL server امکان جستجو بر روی رکوردهایی با محتوای pdf را نیز پس از انجام اندکی تنظیمات، دارا می‌باشد).
- هنگام کار با دیتابیس مباحث تراکنشی نقش بسیار حائز اهمیتی را بازی می‌کنند اما عموماً سیستم عامل‌ها در این زمینه نیازمند کار و برنامه نویسی قابل توجهی هستند (این قابلیت به ویندوز وینتا اضافه شده است).
- کرش یک دیتابیس عموماً سبب کرش سیستم عامل یا حتی کرش سایر دیتابیس‌های موجود خواهد شد.
- امکان تهیه پشتیبان از دیتابیس‌ها و بازیابی آن‌ها ساده است. (حداقل از بازیابی میلیون‌ها فایل ساده‌تر است)
- امکانات replication به صورت پیش فرض در اکثر سیستم‌های دیتابیسی امروزی مهیا است.

ذخیره سازی فایل‌ها در دیتابیس یا استفاده از فایل سیستم متدائل؟

- امکان ثبت و قایع و مدیریت اطلاعات افزوده شده به دیتابیس، از طریق نرم افزارهایی که برای این کار نوشته خواهد شد (یا حتی امکانات توکار این برنامه‌ها) از هر لحاظ نسبت به روش `file system` برتری دارد.
- امکانات سوئیچ کردن به دیتابیسی دیگر در شبکه در صورت کرش یک نود، مهیا است و پیش بینی شده است.
- برای استفاده از یک دیتابیس توسط یک برنامه وب، نیازی به داشتن دسترسی `write` بر روی هیچ فولدری وجود ندارد که این خود یک مزیت امنیتی مهم است و همچنین امکان محدود کردن سطوح دسترسی به فایل‌های ذخیره شده در دیتابیس با برنامه‌های نوشته شده نیز ساده‌تر است. (البته در اینجا مسلماً منظور از دیتابیس، دیتابیس `Access` نیست و MySQL یا SQL Server مد نظر هستند)

نظرات خوانندگان

نوبنده: **salarblog** | تاریخ: ۱۳۸۸/۰۶/۱۰ ۲۲:۱۶:۳۴

مطلوب جالبی بود.
این سوال برای من هم پیش اومده بود که چرا از SQL Server برای Team Foundation Server استفاده کردن، در حالی که این حجم زیاد تراکنش و فایل وجود دارد. ولی باید اقرار کنم که خیلی سریع تر SourceControl 2005 هست.

با این حال استفاده از دیتابیس مسئله قابلیت حمل رو پیش می‌آرد.

نوبنده: **macromediax** | تاریخ: ۱۳۸۸/۰۶/۱۹ ۱۷:۱۲:۵۳

اما معایب استفاده از دیتابیس برای فایل‌ها رو ننوشتی که فکر می‌کنم در انتهای مقاله این مورد رو هم باید بررسی می‌کردی. مثلاً فرض کن یک سرویس نوشتی که فایل‌های آهنگ و عکس‌های کاربرانشو می‌گیره و باید ذخیره کنه و چندین بار پخش کنه یا نمایش بده. مسلماً نمایش هر عکس اگر بخواهد از دیتابیس لود بشه، فشار زیادی را روی dbms می‌ذاره.. بخصوص وقتی تعداد عکس‌ها و درخواست نمایش بالاست..

نوبنده: **وحید نصیری** | تاریخ: ۱۳۸۸/۰۶/۱۹ ۱۷:۴۷:۴۰

مقاله کلا در این مورد بود که اگر تعداد عکس‌ها بالا باشد روش سنتی ذخیره سازی روی هارد به دلایلی که ذکر شده مشکل زا خواهد شد و اس کیوال سرور اینقدر توانمند و پخته است که بتواند این مساله را حل و فصل کند. (نمونه استفاده در مقیاس سازمانی از آن شیرپوینت است که همه اطلاعات را داخل دیتابیس قرار داده و سپس لود می‌کند) یا برای نمونه YouTube نیز تمام ویدیوهای خودش را درون دیتابیس ذخیره کرده و صد الته برای مقیاس پذیری آن روش‌های کش کردن اطلاعات پیشرفت‌های را نیز توسعه داده‌اند. شیرپوینت هم برای این منظور چیزی به نام BlobCache دارد.

نوبنده: **Saeed** | تاریخ: ۱۳۸۹/۰۳/۲۲ ۱۷:۰۳:۴۸

خیلی ممنون از مطلب خوبتون.
خیلی وقت بود که به دنبال جواب این سوال بودم که برای ذخیره کردن فایلها در سمت سرور از کدوم روش استفاده کنم. الته به این نتیجه رسیده بودم که فایلها رو در بانک اطلاعاتی ذخیره کنم. بعد از انجام این کار متوجه شدم که برای ذخیره کردن حدود 20 مگابایت عکس در سمت سرور، به اندازه فایل‌های بانک اطلاعاتی من در حدود 80 مگابایت افزوده می‌شود. که این امر من رو برای ذخیره و نگهداری حجم و تعداد زیاد فایل نگران کرده بود.

عنوان: به روز رسانی فیلدهای XML در SQL Server
نویسنده: حمید نصیری
تاریخ: ۱۳۸۸/۰۶/۱۳ ۱۸:۱۷:۰۰
آدرس: www.dotnettips.info
برچسبها: SQL Server

از 2005 SQL server به بعد، پشتیبانی کاملی از XML توسط این محصول صورت می‌گیرد. در ادامه مروری خواهیم داشت بر نحوه‌ی به روز رسانی مقادیر فیلدهایی از نوع XML در SQL Server . در ابتدا جدول موقتی زیر را که شامل یک رکورد از نوع XML است، در نظر بگیرید:

```
DECLARE @tblTest AS TABLE (xmlField XML)

INSERT INTO @tblTest
(
    xmlField
)
VALUES
(
    '<Sample>
        <Node1>Value1</Node1>
        <Node2>Value2</Node2>
        <Node3>OldValue</Node3>
    </Sample>'
)
```

می‌خواهیم OldValue را به مقداری دیگر تغییر دهیم.

سعی اول:

```
DECLARE @newValue VARCHAR(50)
SELECT @newValue = 'NewValue'
UPDATE @tblTest
SET    xmlField.modify('replace value of (/Sample/Node3)[1] with ' + @newValue)
```

این سعی با خطای زیر متوقف می‌شود:

```
The argument 1 of the XML data type method "modify" must be a string literal.
```

بنابراین از روش String concatenation برای معرفی مقدار متغیر مورد نظر در اینجا نمی‌شود استفاده کرد.

سعی دوم:

```
DECLARE @newValue VARCHAR(50)
SELECT @newValue = 'NewValue'

UPDATE @tblTest
SET    xmlField.modify(
        'replace value of (/Sample/Node3)[1] with sql:variable("@newValue")'
    )
```

روش معرفی صحیح یک متغیر را در اینجا می‌توان مشاهده کرد. اما این سعی نیز با خطای زیر متوقف می‌شود:

```
XQuery [@tblTest.xmlField.modify()]: The target of 'replace value of' must be a non-metadata attribute or an element with simple typed content, found 'element(NodeThree,xdt:untyped) ?'
```

```

DECLARE @newValue VARCHAR(50)
SELECT @newValue = 'NewValue'

UPDATE @tblTest
SET    xmlField.modify(
      'replace value of (/Sample/Node3/text())[1]
       with sql:variable("@newValue")'
)
SELECT xmlField.value('(/Sample/Node3)[1]', 'varchar(50)') FROM @tblTest

```

و بله. کار می‌کنه!

XML ای را که در ابتدا استفاده کردیم از نوع un-typed XML محسوب شده و هیچ schema ای را برای آن در نظر نگرفته‌ایم، به همین جهت باید دقیقاً مشخص کنیم که قصد داریم text این node را ویرایش نمائیم.

مشکل بعدی!

در ابتدا مثال زیر را در نظر بگیرید:

```

DECLARE @tblTest AS TABLE (xmlField XML)

INSERT INTO @tblTest
(
  xmlField
)
VALUES
(
  '<Sample>
    <Node1>Value1</Node1>
    <Node2>Value2</Node2>
    <Node3></Node3>
  </Sample>'
)

DECLARE @newValue VARCHAR(50)
SELECT @newValue = 'NewValue'

UPDATE @tblTest
SET    xmlField.modify(
      'replace value of (/Sample/Node3/text())[1]
       with sql:variable("@newValue")'
)
SELECT xmlField.value('(/Sample/Node3)[1]', 'varchar(50)') FROM @tblTest

```

این عبارات T-SQL، خلاصه بحث ما تا به اینجا هستند اما با یک تفاوت. نود 3 در اینجا خالی است.

اگر اسکریپت را اجرا کنید، هیچ تغییری را مشاهده نخواهید کرد. به عبارت دیگر به روز رسانی صورت نمی‌گیرد. در اینجا چون text این نود خالی است، فرض SQL Server بر این خواهد بود که وجود ندارد، بنابراین این نود را به روز رسانی نخواهد کرد. به همین منظور باید برای به روز رسانی این نود، عبارت جدید را در جایی که insert ندارد text replace کرد (و نه insert).

```

DECLARE @newValue VARCHAR(50)
SELECT @newValue = 'NewValue'

UPDATE @tblTest
SET    xmlField.modify(
      'replace value of (/Sample/Node3/text())[1]
       with sql:variable("@newValue")'
)

UPDATE @tblTest
SET    xmlField.modify(
      'insert text{sql:variable("@newValue")} into
       (/Sample/Node3)[1] [not(text())]'
)

```

```
SELECT xmlField.value('/Sample/Node3[1]', 'varchar(50)') FROM @tblTest
```

نظرات خوانندگان

نویسنده: Alex's Blog
تاریخ: ۱۳۸۸/۰۶/۲۲

سلام
عالی بود. ممنون

مفهومی در SQL Server وجود دارد به نام parameter sniffing که شرح آن به صورت زیر است.
ابتدا رویه ذخیره شده زیر را در نظر بگیرید:

```
create procedure test (@pid int)
as
select * from Sales.SalesOrderDetail
where ProductID = @pid
```

استفاده از کوئری‌های پارامتری یکی از بهترین تمرین‌های کاری با SQL server است؛ از آنجائیکه در این حالت plan تهیه شده مجدداً مورد استفاده قرار گرفته، همچنین از SQL injection نیز جلوگیری خواهد کرد، زیرا برای نمونه در مثال فوق تنها pid از نوع int پذیرفته می‌شود و نه هر ورودی خطرناک دیگری.
اما این نوع کوئری‌ها یک مشکل را نیز به همراه خود دارند. این plan تهیه شده به ازای اولین ورودی رویه ذخیره شده تهیه می‌شود (parameter sniffing) و الزامی ندارد که برای دومین ورودی و فراخوانی‌های بعدی، بهترین plan باشد.

برای حل این مشکل راه‌های زیادی هست:
الف) انتساب پارامترهای یک رویه ذخیره شده به متغیری محلی

```
create procedure test (@pid int)
as
Declare @mpid int
Set @mpid = @pid
select * from Sales.SalesOrderDetail
where ProductID = @mpid
```

در اینجا پارامتر ورودی مستقیماً در کوئری استفاده نشده است و SQL Server این متغیر محلی را sniff نخواهد کرد.

ب) استفاده از گزینه RECOMPILE که سبب خواهد شد به ازای هر ورودی یک plan بهینه تهیه شود. این مورد مصرف CPU بالایی را به همراه خواهد داشت.

ج) راه حل ارائه شده در SQL Server 2005 استفاده از علاوه اضافه کردن گزینه کمکی زیر به انتهای اسکریپت فوق

```
OPTION (OPTIMIZE FOR(@pid = 544))
```

در اینجا فرض بر این است که می‌دانیم pid=544 بسیار مورد استفاده قرار خواهد گرفت، بنابراین این معرفی را به موتور بهینه ساز SQL Server ارائه خواهیم کرد.

د) راه حل ارائه شده در SQL Server 2008 با استفاده از Optimize for unknown که در اس کیوال سرور 2008 معرفی شده است، مزیت استفاده از کوئری‌های پارامتری همانند استفاده مجدد از plan تهیه شده، حفظ گشته اما این plan تهیه شده اولیه بر اساس اولین مقدار پاس شده، تهیه نگردیده و حالت عمومی‌تر و بهینه‌تری را برای اکثر مقادیر پاس شده خواهد داشت.

```
create procedure test (@pid int)
```

Optimize for unknown

```
as  
select * from Sales.SalesOrderDetail  
where ProductID = @pid  
OPTION(OPTIMIZE FOR (@pid UNKNOWN))
```

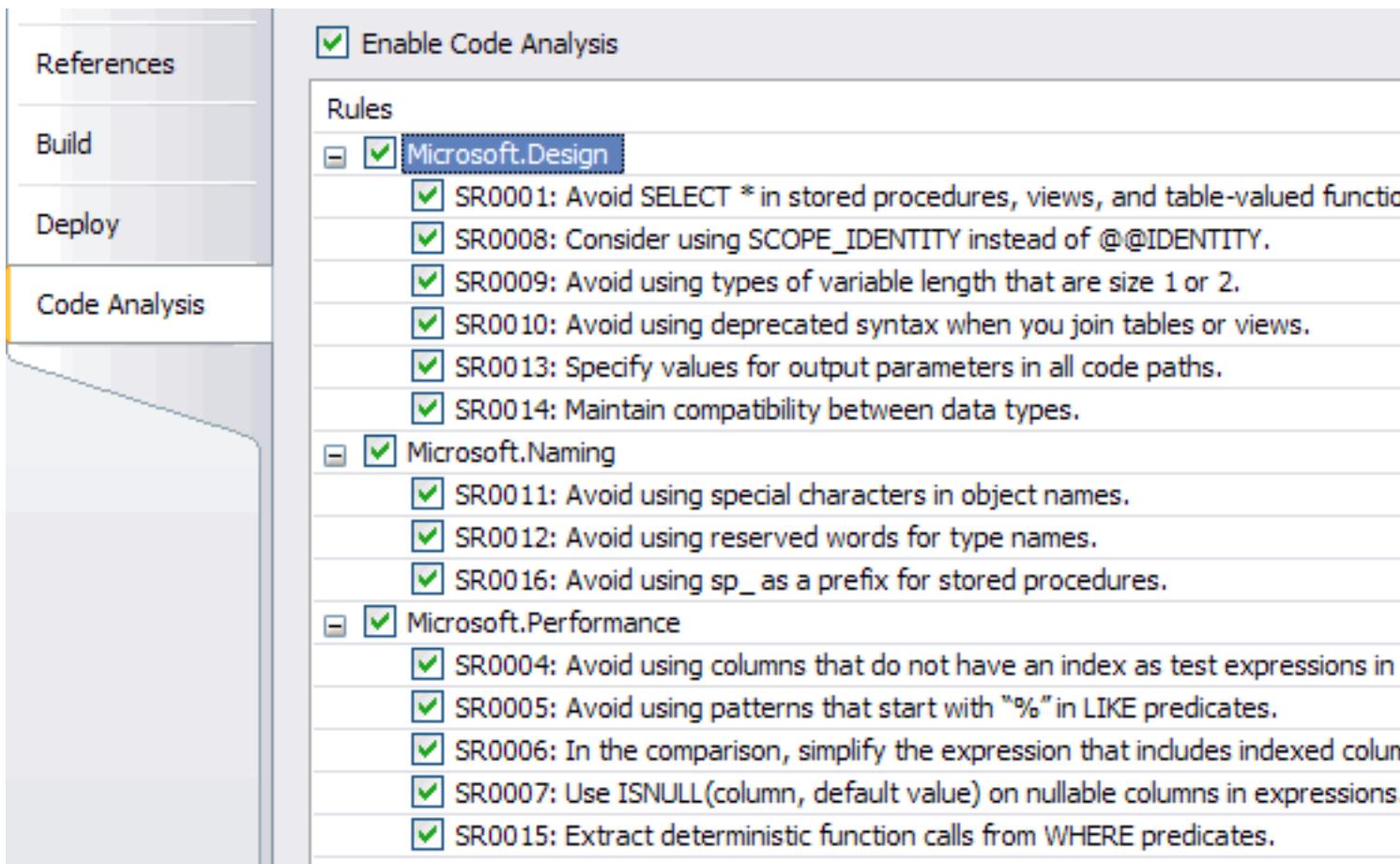
جهت مطالعه بیشتر ([+ و +](#))

نظرات خوانندگان

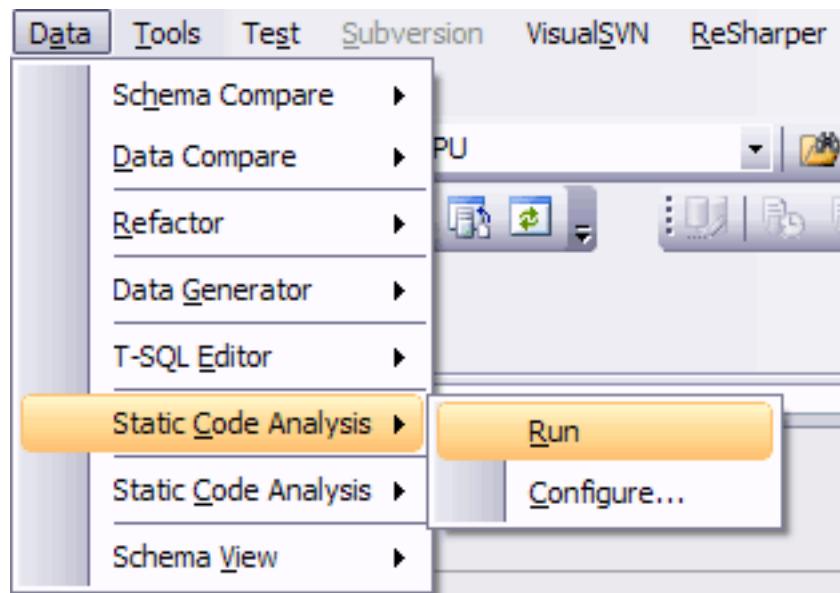
نوبسند: Amin
تاریخ: ۰۹:۳۳:۳۶ ۱۳۸۹/۰۲/۰۵

این مورد خیلی منو اذیت کرده بود که حلش کرده بودم اما امروز این پستتون رو دیدم!

حتما با FxCop که برای آنالیز اسمنبلی‌های برنامه‌های دات نتی بکار می‌رود آشنایی دارید. شبیه به این مورد به صورت افزونه‌ای برای Visual studio 2008 team system نیز موجود است. فقط کافی است Microsoft® Visual Studio Team System 2008 را نصب کرده و یک پروژه دیتابیس جدید را شروع کنید (نوع database wizard که یک دیتابیس کامل Database Edition GDR R2 را import می‌کند). سپس در برگه build تیک مربوط به code analysis را قرار دهید (شکل 1) و یکبار پروژه را build کنید. به این صورت در پنجره خروجی، اشکالات کدهای T-SQL شما گوشزد می‌شود (شکل 3). این کار را با استفاده از منوی Data نیز می‌توان انجام داد (شکل 2).



شکل ۱- فعال سازی تحلیل و بررسی کد



شکل 2- اجرای تحلیل و بررسی کد

Output

Show output from: Build

```
.SQL(475,15)Warning : SR0007 : Microsoft.Performance : Nullable columns can cause final results to be evaluated as NULL
.SQL(502,22)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from Char(157) to NVarChar(1000).
.SQL(506,15)Warning : SR0007 : Microsoft.Performance : Nullable columns can cause final results to be evaluated as NULL
.SQL(508,13)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from NVarChar(1023) to VarChar(128)
.SQL(518,15)Warning : SR0007 : Microsoft.Performance : Nullable columns can cause final results to be evaluated as NULL
30,7)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from Char(6) to VarChar(5).
33,7)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from Char(6) to VarChar(5).
102,8)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from VarChar(24) to VarChar(20).
104,8)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from VarChar(25) to VarChar(20).
122,8)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from VarChar(24) to VarChar(20).
124,8)Warning : SR0014 : Microsoft.Design : Data loss might occur when casting from VarChar(25) to VarChar(20).
SQL(8,9)Warning : SR0001 : Microsoft.Design : The shape of the result set produced by a SELECT * statement will change i
```

شکل 3- یک نمونه خروجی حاصل از تحلیل و بررسی کد

مطلوبی چندی قبل در مورد " ذخیره سازی فایل‌ها در دیتابیس یا استفاده از فایل سیستم متداول؟ " منتشر گردید، جهت برشمردن فواید ذخیره سازی فایل‌ها در دیتابیس ([+](#)). اما معایب این نوع ذخیره سازی بررسی نشدند:

- الف) اختصاص یافتن قسمتی از بافر SQL Server به این امر.
- ب) با توجه به قرار گرفتن داده‌های BLOB در دیتابیس ، transaction log قابل توجهی تولید خواهد شد. ([+](#))
- ج) بیش از 2GB را نمی‌توان در فیلد هایی از نوع varbinary(max) ذخیره کرد.
- د) به روز رسانی BLOB ها سبب ایجاد fragmentation می‌شود.

مايكروسافت برای رفع اين مشكلات در 2008 FileStream قابلیت جدیدی را ارائه داده است به نام که در طی مقالاتی به بررسی آن خواهیم پرداخت.

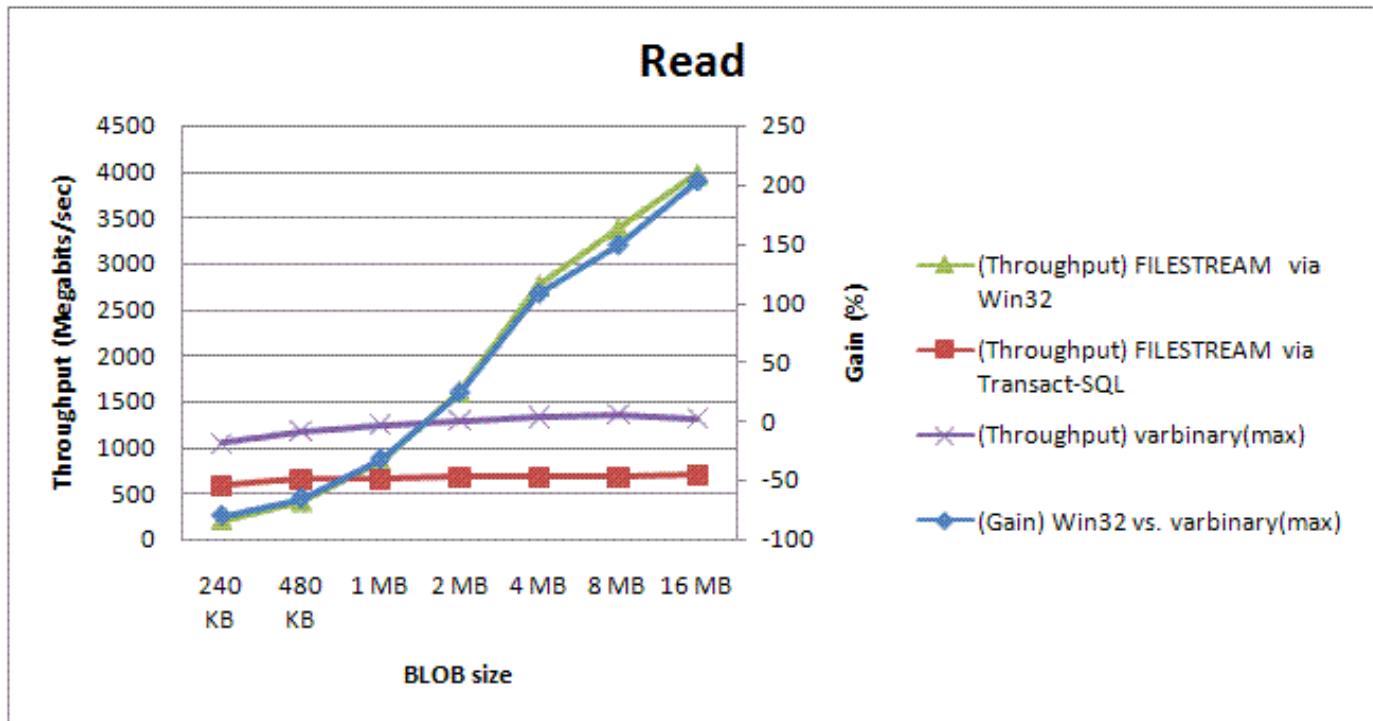
FILESTREAM موتور دیتابیس اس کیوال سرور را با سیستم فایل NTFS یکپارچه می‌کند؛ به این صورت که داده‌های BLOB از نوع varbinary(max) را به صورت فایل بر روی سیستم ذخیره خواهد کرد. سپس با استفاده از دستورات T-SQL می‌توان این فایل‌ها را ثبت، حذف، به روز رسانی، جستجو و بک آپ گیری کرد. این قابلیت نیز از فیلد هایی (max) varbinary استفاده می‌کند؛ اما اکنون ویژگی و برچسب FILESTREAM به این نوع فیلد ها الصاق خواهد شد. data باید در FILESTREAM filegroups ذخیره شوند. data containers در حقیقت همان پوشش های فایل سیستم می‌باشند. به آن ها نیز گفته می‌شوند که مرزی هستند بین ذخیره سازی داده‌ها در فایل سیستم و در دیتابیس.

مزایای سیستم FileStream چیست؟

- الف) سیستم transaction log مختص به خود را داشته، به همین جهت سبب رشد غیر منطقی حجم فایل دیتابیس اصلی نمی‌شوند.
- ب) هنگام به روز رسانی فیلد هایی از این دست، صرفا ایجاد یا حذف یک فایل ممکن است؛ بنابراین fragmentation ایجاد شده در این حالت بسیار کمتر از روش استفاده از فیلد هایی از نوع varbinary(max) می‌باشد.
- ج) استفاده از NT system cache جهت کش کردن اطلاعات که سبب بالا بردن بازدهی بانک اطلاعاتی خواهد شد.
- د) از buffer pool اس کیوال سرور در این حالت استفاده نشده (مطابق قسمت ج) و این حافظه جهت امور روزمره ای اس کیوال سرور کاملاً مهیا خواهد بود.
- ه) محدودیت 2GB فیلد هایی از نوع varbinary(max) در فایل سیستم، دیگر وجود نخواهد داشت.

چه زمانی بهتر است از FileStream استفاده شود؟

- الف) فایل هایی که ذخیره می‌شوند به طور متوسط بیش از یک مگابایت حجم داشته باشند. (برای کمتر از این مقدار varbinary(max) BLOBS کارآیی بهتری را ارائه می‌دهند). هر چند این مرز یک مگابایت مطابق اطلاعات books online است اما تجربیات کاری نشان می‌دهند که این سقف را باید 256 کیلوبایت درنظر گرفت.
- ب) قابلیت خواندن سریع اطلاعات فایل‌ها ممکن است (بررسی کارآیی مطابق تصویر زیر از MSDN). سیستم NTFS نسبت به SQL Server در خواندن فایل‌های حجمی سریعتر عمل می‌کند.
- ج) اگر از یک معماری middle tier در برنامه‌های خود در حال استفاده‌اید.
- د) زمانیکه نیاز باشد تا اطلاعات relational و non-relational مورد استفاده قرار گیرند.



نکاتی را که باید هنگام ذخیره سازی اطلاعات در FileStream در نظر داشت

- الف) هنگامی که یک جدول حاوی فیلدی از نوع FileStream میباشد، باید دارای فیلد ID منحصر بفرد نیز باشد.
- ب) data containers ایی که پیش از این در مورد آنها صحبت شد، باید تو در تو باشند.
- ج) FILESTREAM filegroups بر روی درایوهای فشرده شده نیز میتوانند قرار داشته باشند.

امنیت داده‌های FileStream در اس کیوال سرور دقیقاً همانند امنیت سایر اطلاعات ذخیره شده در دیتابیس است (دسترسی در حد جدول و یا فیلد). اگر کاربری دسترسی به فیلد FileStream در یک جدول داشته باشد، میتواند آن فایل را گشوده و استفاده کند. رمزنگاری بر روی این ستون‌ها پشتیبانی نمی‌شود. تنها اکانتی که اس کیوال سرور تحت آن در حال اجرا است دسترسی به آغاز و مشغول به کار می‌شود، اس کیوال سرور دسترسی به FILESTREAM data container را محدود خواهد کرد و دسترسی به این اطلاعات تنها از طریق دستورات T-SQL و یا OpenSqlFilestream API میسر خواهد بود. بدیهی است زمانیکه اس کیوال سرور متوقف شود، این اطلاعات بدون هیچگونه محدودیتی قابل دسترسی بوده و تنها محدودیت‌های سیستمی به آن‌ها اعمال خواهند شد (که این مورد باید مد نظر باشد).

نگهداری FileStream به صورت فیلهای varbinary(max) یکپارچه با دیتابیس ذخیره می‌شود؛ بنابراین نحوه تهیه پشتیبان از آن‌ها همانند روش‌های متداول است بدون هیچگونه تغییری (و این اطلاعات در بک آپ دیتابیس لحاظ می‌شوند). اگر نیاز بود هنگام تهیه پشتیبان از این نوع داده‌ها بک آپ گرفته نشود، میتوان از partial backup با پارامترهای مربوطه استفاده کرد.

ادامه دارد ...

نظرات خوانندگان

نویسنده: fahimi
تاریخ: ۱۳۹۱/۰۶/۰۷ ۲۲:۲

سلام
از آموزش بسیار عالی و کاربردی جنابعالی کمال تشکر را دارم
من آموزش شما را در C# & delphi پیاده کردم و به خوبی اجرا شد
سوال: آیا دستورات Update , Delete
علاوه بر تیبل فایل‌ها را هم تاثیر دار
به هر حال من دستور delete را اجرا کردم در تیبل رکورد حذف شد ولی فایل مرتبط با رکورد حذف نشد
با تشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۷ ۲۲:۱۰

همینطوره. این فایل‌ها سر فرصت توسط سیستم [FILESTREAM garbage collection](#) حذف خواهند شد.

نویسنده: حسین مرادی نیا
تاریخ: ۱۳۹۱/۰۶/۰۷ ۲۲:۵۷

سلام؛
اگر از Entity Framework در حالت CodeFirst استفاده نماییم میتوان از قابلیت‌های FileStream در آن استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۶/۰۷ ۲۳:۴۴

بله. خاصیت مورد نظر را به صورت byte array تعریف کنید:

```
public class Image
{
    public int Id { get; set; }
    public byte[] Data { get; set; }
}
```

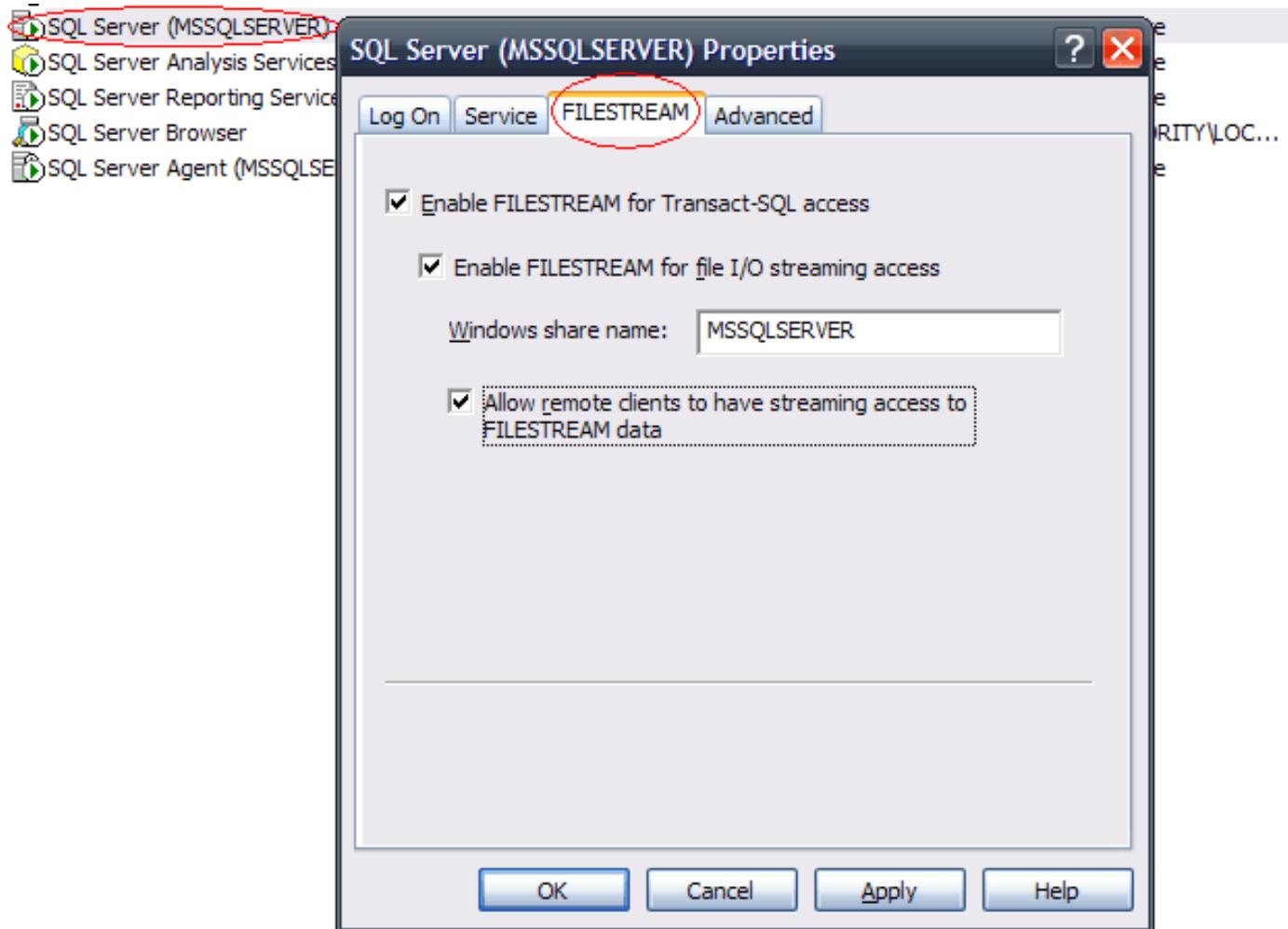
در این قسمت نحوه‌ی فعال سازی قابلیت FileStream را بررسی خواهیم کرد و در قسمت بعدی نحوه‌ی دسترسی به آن را از طریق برنامه نویسی مرور می‌نماییم.

فعال سازی قابلیت FileStream همانند اکثر قابلیت‌های اس کیوال سرور، فعال سازی FileStream نیز حداقل به دو صورت استفاده از GUI و قابلیت‌های management studio می‌سر است و یا استفاده از دستورات T-SQL (و البته کتابخانه‌ی SMO یا همان محصور کننده‌ی توانایی‌های FileStream نیز قابل استفاده است).

روش اول) استفاده از management studio به صورت پیش فرض غیرفعال است. برای فعال سازی آن به مسیر زیر مراجعه نمائید:

Start > All Programs > Microsoft SqlServer 2008 > Configuration Tools > SQL Server Configuration Manager

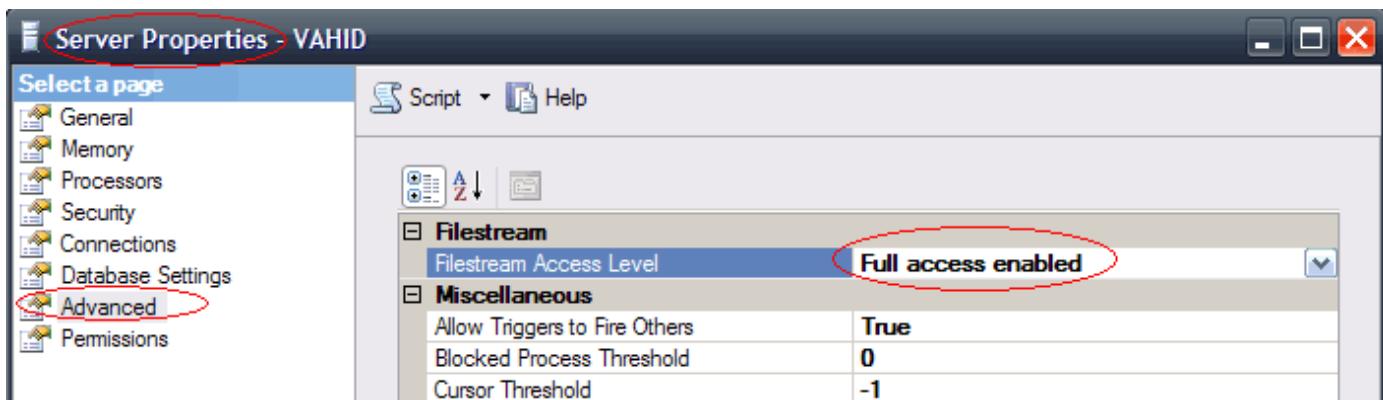
سپس در قسمت SQL Server services، وهله مربوط به SQL Server را یافته، کلیک راست و به برگه خواص آن مراجعه کرده (شکل زیر) و قابلیت FileStream را فعال کنید:



گزینه‌های مختلف آن به شرح زیر هستند:

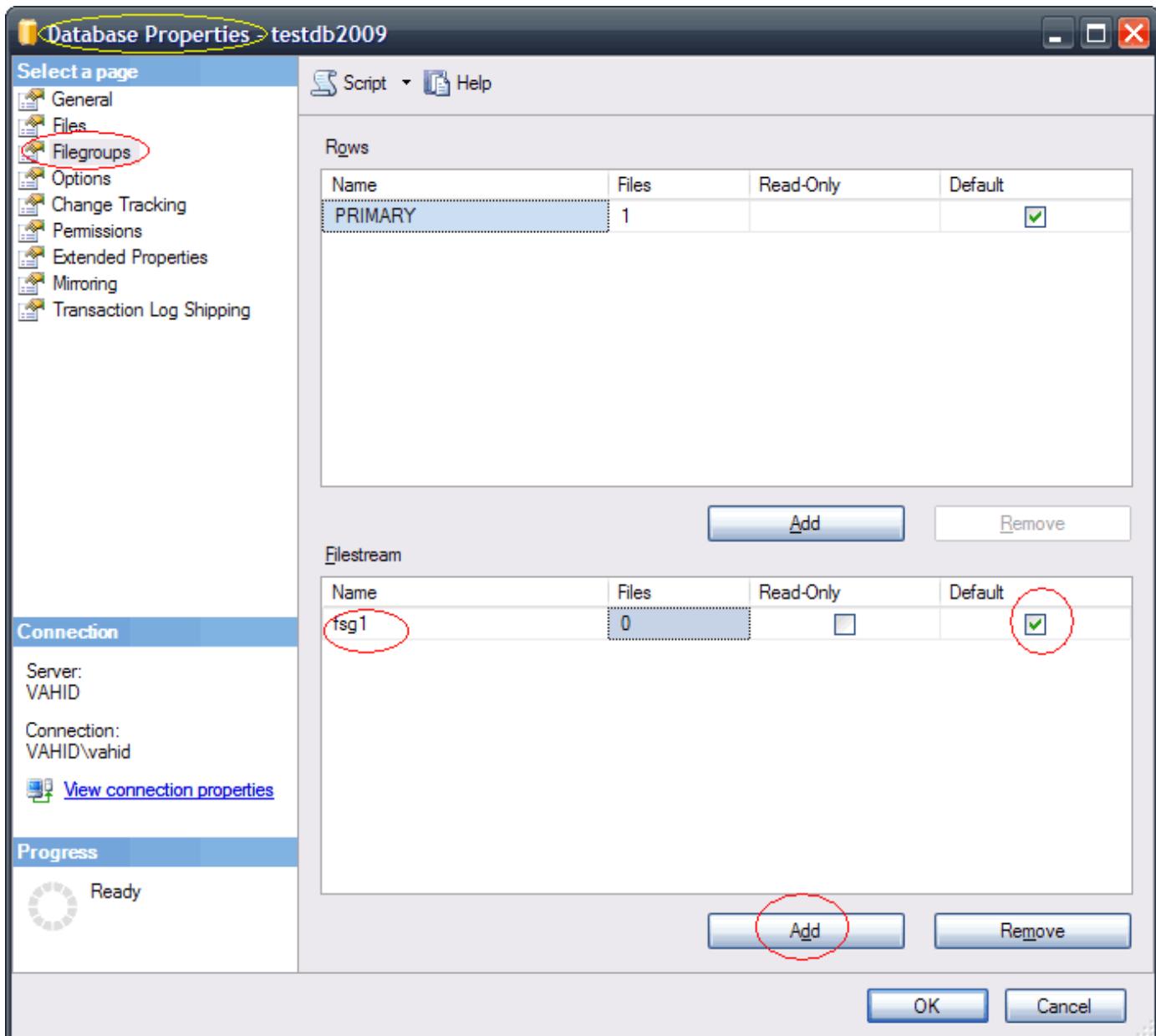
- امکان استفاده از دستورات T-SQL را جهت دسترسی به فایل‌ها فعال می‌سازد (یا بر عکس)
- امکان دسترسی به فایل‌ها با استفاده از Win32 streaming access
- اجازه دسترسی به کلاینت‌های راه دور جهت استفاده از FileStream

مرحله بعد، فعال سازی سطح دسترسی به سرور است. به management studio مراجعه نمایید. سپس بر روی وہله سرور مورد نظر کلیک راست نموده و به خواص آن مراجعه کنید (شکل زیر). سپس در قسمت advanced سطح دسترسی را بر روی Full قرار دهید.

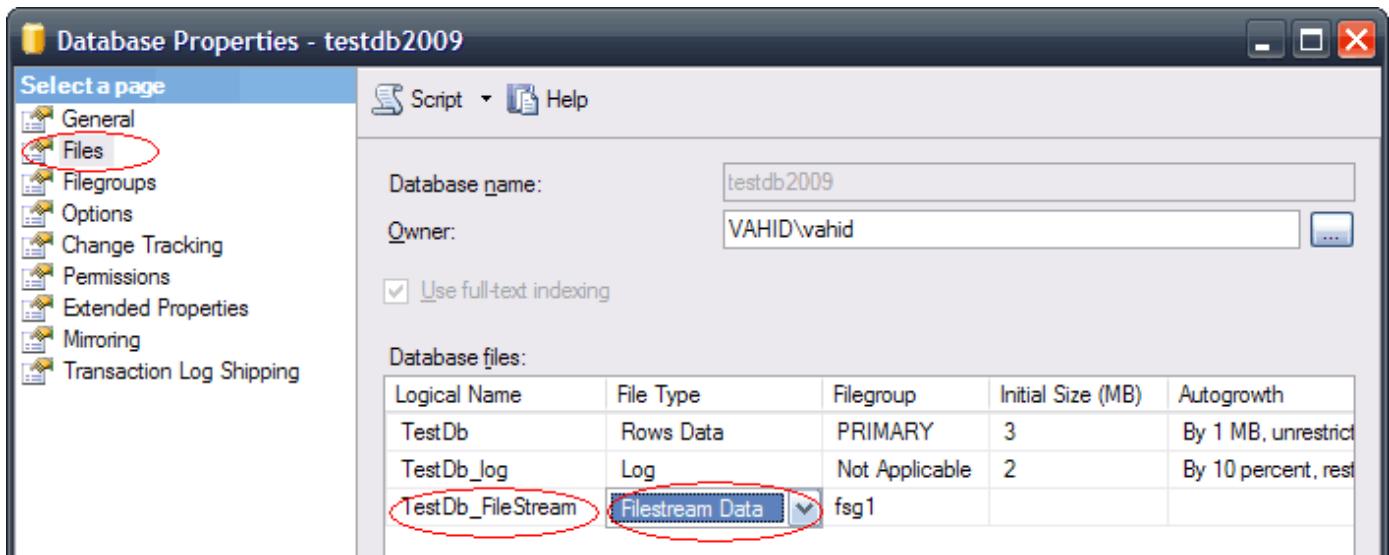


پس از این تنظیم به شما پیغام داده خواهد شد که باید دیتابیس سرور را یکبار راه اندازی مجدد نمایید تا تنظیمات مورد نظر، اعمال شوند.

در ادامه باید دیتابیسی را که نیاز است نوع داده FileStream را پیذیرد، تنظیم نمود.
بر روی دیتابیس مورد نظر کلیک راست کرده و در برگه خواص آن به گزینه‌ی Filegroups مراجعه کنید. سپس در اینجا یک گروه جدید را اضافه کرده، نامی دلخواه را وارد نموده و سپس تیک مربوط به default بودن آن را نیز قرار دهید (شکل زیر):



سپس در همین برگهی خواص دیتابیس که باز است، به گزینه‌ی Files مراجعه کنید. در اینجا سه کار را باید انجام دهید. ابتدا بر روی دکمه Add کلیک کرده و در قسمت logical name ردیف اضافه شده، نامی دلخواه را وارد کنید. سپس file type آن را بر روی FileStream قرار دهید. در ادامه به قسمت path در همین ردیف مراجعه نموده و مسیر ذخیره سازی را مشخص کنید. در پایان بر روی دکمه OK کلیک نمائید تا کار تنظیم دیتابیس به پایان رسد (شکل زیر):



روش دوم) استفاده از دستورات T-SQL
منهای قسمت تنظیمات SQL Server Configuration Manager که باید از طریق روش عنوان شده صورت گیرد، سایر موارد فوق را با استفاده از دستورات T-SQL نیز می‌توان انجام داد:

الف) تنظیم سطح دسترسی بر روی سرور

```
EXEC sp_configure filestream_access_level, 2 -- 0 : Disable , 1 : Transact Sql Access , 2 : Win IO
Access
GO
RECONFIGURE
GO
```

ب) تنظیمات دیتابیس

اگر نیاز باشد دیتابیس جدیدی ایجاد شود: (ایجاد گروه فایل مربوطه و سپس تنظیمات مسیر آن)

```
CREATE DATABASE Test_Db
ON
PRIMARY ( NAME = TestDb1,
    FILENAME = 'C:\DATA\Test_Db.mdf'),
FILEGROUP FileStreamGroup1 CONTAINS FILESTREAM( NAME = Testfsg1,
    FILENAME = 'C:\DATA\Learning_DbStream')
LOG ON ( NAME = TestDbLog1,
    FILENAME = 'C:\DATA\Test_Db.ldf')
GO
```

و یا ایجاد تغییرات بر روی دیتابیس موجود: (ایجاد گروه فایل مخصوص و سپس افزودن فایل مربوطه و تنظیمات آن)

```
--add filegroup
alter database TestDb
Add FileGroup FileStreamFileGroup1 contains FileStream
go

--Add FileGroup To DB
alter database TestDB
add file
(
    name = 'UserDocuments' ,
```

```
filename = 'C:\FileStream\UserDocuments'  
) to filegroup FileStreamFileGroup1
```

تعريف جدولی آزمایشی به همراه فیلدی از نوع `FileStream` :
تا اینجا سرور و همچنین دیتابیس جهت پذیرش این نوع داده آماده شدند. اکنون نوبت به استفاده از آن است:

```
CREATE TABLE [tblFiles]  
(  
    FileId     UNIQUEIDENTIFIER NOT NULL ROWGUIDCOL UNIQUE DEFAULT(NEWID()),  
    Title      NVARCHAR(255) NOT NULL,  
    SystemFile VARBINARY(MAX) FILESTREAM NULL  
)  
ON [PRIMARY] FILESTREAM_ON [fsg1]
```

توسط دستور T-SQL فوق جدولی که از نوع داده `FileStream` استفاده می‌کند، ایجاد خواهد شد. این جدول همانطور که مشخص است حتما باید دارای یک فیلد منحصر بفرد باشد (ر.ک. مقاله قبل) و همچنین برچسب فایل استریم به فیلدی از نوع `VARBINARY(MAX)` نیز الصاق شده است. به علاوه گروه فایل آن نیز باید به صورت صریح مشخص گردد؛ که در مثال ما مطابق تصاویر به `fsg1` تنظیم شده بود.

ادامه دارد ...

نظرات خوانندگان

نوبنده: Yahya
تاریخ: ۱۳۸۸/۰۹/۰۹ ۱۱:۲۲:۲۷

از اینکه به این خوبی توضیح داده اید بسیار سپاسگزارم

در انتهای قسمت قبل، نحوه ایجاد یک جدول جدید با فیلدی از نوع فایل استریم بررسی شد، حال اگر جدولی از پیش وجود داشت، نحوه افزودن فیلد ویژه مورد نظر به آن، به صورت زیر است:

```

alter table tbl_files set(filestream_on = 'default')

go
alter table tbl_files
add

[systemfile] varbinary(max) filestream null ,
FileId uniqueidentifier not null rowguidcol unique default (newid())
go

```

در ادامه جدول `tblFiles` قسمت قبل را در نظر بگیرید:

```

CREATE TABLE [tblFiles](
[FileId] [uniqueidentifier] ROWGUIDCOL NOT NULL,
[Title] [nvarchar](255) NOT NULL,
[SystemFile] [varbinary](max) FILESTREAM NULL,
UNIQUE NONCLUSTERED
(
[FileId] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] FILESTREAM_ON [fsg1]

ALTER TABLE [dbo].[tblFiles] ADD DEFAULT (newid()) FOR [FileId]
GO

```

نحوه افزودن رکوردی جدید به جدول `tblFiles`:

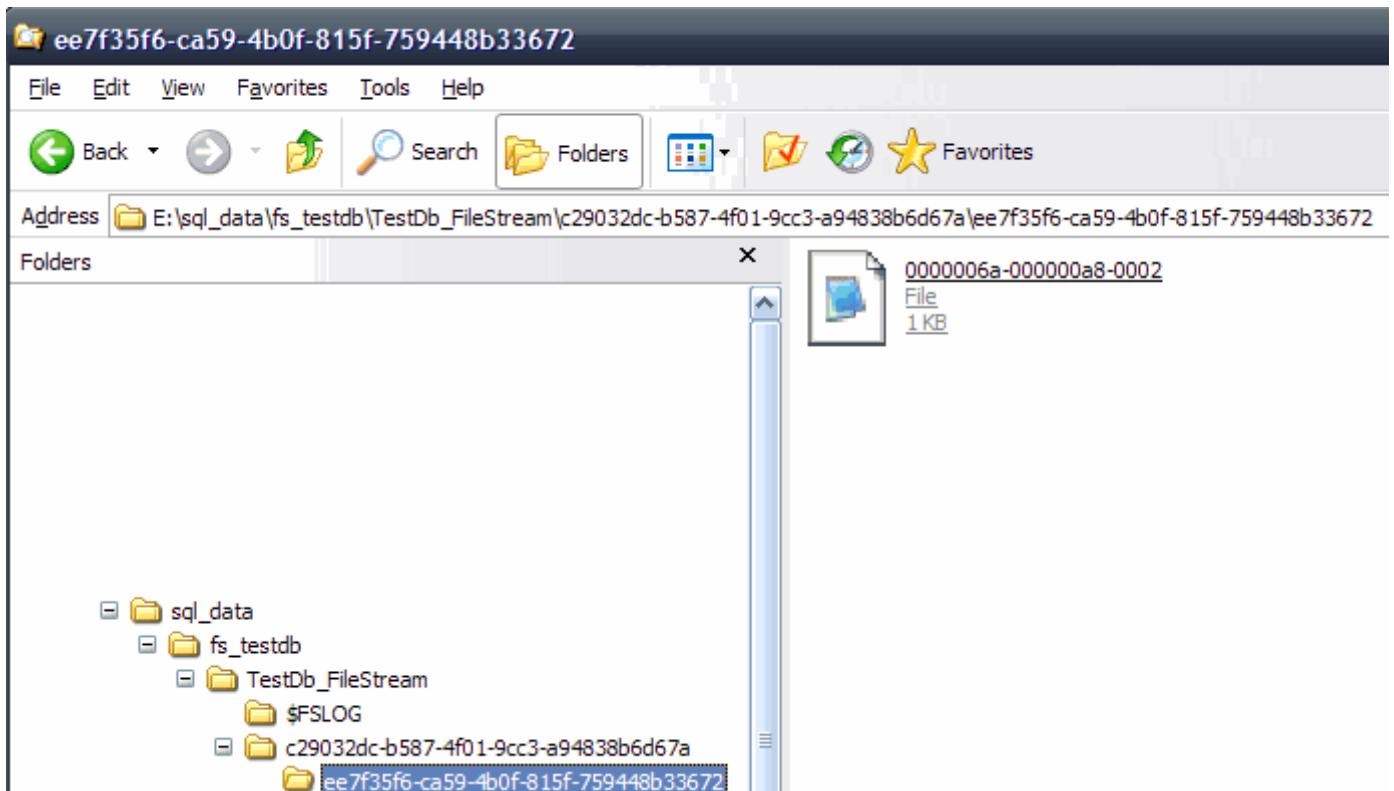
```

INSERT INTO [tblFiles]
(
[Title],
[SystemFile]
)
VALUES
(
'file-1',
CAST('data data data' AS VARBINARY(MAX))
)

```

در اینجا سعی کرده‌ایم یک رشته ساده را در فیلدی از نوع فایل استریم ذخیره کنیم که روش کار به صورت فوق است. از آنجائیکه مقدار پیش فرض `FileId` را هنگام تعریف جدول به `NEWID` تنظیم کرده‌ایم، نیازی به ذکر آن نیست و به صورت خودکار محاسبه و ذخیره خواهد شد.

اگر کنجکاو باشید که این فایل اکنون کجا ذخیره شده و نحوه مدیریت آن توسط اس کیوال سرور به چه صورتی است، فقط کافی است به مسیری که هنگام افزودن گروه فایل‌ها و فایل مربوطه در تنظیمات خواص دیتابیس در قسمت قبل مشخص کردیم، مراجعه کرد (شکل زیر).



بدیهی است افزودن یک رشته به این صورت کاربرد عملی ندارد و صرفا جهت یک مثال ارائه شد. در ادامه، نحوه ثبت محتویات یک فایل را در فیلدی از نوع فایل استریم و سپس خواندن اطلاعات آن را از طریق برنامه نویسی بررسی خواهیم کرد:

```
using System;
using System.IO;
using System.Data.SqlClient;
using System.Data;

namespace FileStreamTest
{
    class CFS
    {
        /// <summary>
        /// افزودن رکورد به جدول حاوی ستونی از نوع فایل استریم
        /// </summary>
        /// <param name="filePath">مسیر فایل</param>
        /// <param name="title">عنوانی دلخواه</param>
        public static void AddNewRecord(string filePath, string title)
        {
            // آیا فایل وجود دارد؟
            if (!File.Exists(filePath))
                throw new FileNotFoundException("لطفاً مسیر فایل معتبری را مشخص نمایید.", filePath);

            // خواندن اطلاعات فایل در آرایه‌ای از بایت‌ها
            byte[] buffer = File.ReadAllBytes(filePath);

            using (SqlConnection objSqlCon = new SqlConnection())
            {
                // کانکشن استرینگ باید از یک فایل کانفیگ خوانده شود
                objSqlCon.ConnectionString =
                    "Data Source=(local);Initial Catalog=testdb2009;Integrated Security = true";
                objSqlCon.Open();

                // شروع یک تراکنش
                using (SqlTransaction objSqlTran = objSqlCon.BeginTransaction())
                {
                    // ساخت عبارت افزودن پارامتری
                    using (SqlCommand objSqlCmd = new SqlCommand(
                        "INSERT INTO [tblFiles]([Title], [SystemFile]) VALUES(@title , @file)",
```

```

        objSqlCon, objSqlTran))
    {
        objSqlCmd.CommandType = CommandType.Text;
        //تعريف وضعيت پaramترها و مقدار دهی آنها
        objSqlCmd.Parameters.AddWithValue("@title", title);
        objSqlCmd.Parameters.AddWithValue("@file", buffer);
        //اجرای فرمان
        objSqlCmd.ExecuteNonQuery();
    }
    //پایان تراکنش
    objSqlTran.Commit();
}
}

/// <summary>
/// دریافت اطلاعات فایل ذخیره شده به صورت آرایه‌ای از بایت‌ها
/// </summary>
/// <param name="fileId"></param>
/// <returns></returns>
public static byte[] GetDataFromDb(string fileId)
{
    byte[] data = null;

    using (SqlConnection objConn = new SqlConnection())
    {
        //کوئری اس کیوال پارامتری جهت دریافت محتویات فایل
        string cmdText = "SELECT SystemFile FROM tblFiles WHERE FileId=@id";
        using (SqlCommand objCmd = new SqlCommand(cmdText, objConn))
        {
            //کانکشن استرینگ باید از یک فایل کانفیگ خوانده شود:
            objConn.ConnectionString =
                "Data Source=(local);Initial Catalog=testdb2009;Integrated Security = true";
            objConn.Open();

            //تنظیم کردن وضعيت و مقدار پaramتر تعريف شده در کوئری
            objCmd.Parameters.AddWithValue("@id", fileId);

            //اجرای فرمان و دریافت فایل
            using (SqlDataReader objread = objCmd.ExecuteReader())
            {
                if (objread != null)
                    if (objread.Read())
                    {
                        if (objread["SystemFile"] != DBNull.Value)
                            data = (byte[])objread["SystemFile"];
                    }
            }
        }
        return data;
    }
}
}

```

مثالی در مورد روش استفاده از کلاس فوق :

```

using System.IO;
namespace FileStreamTest
{
    class Program
    {
        static void Main(string[] args)
        {
            CFS.AddNewRecord(@"C:\filest05.PNG", "test1");
            //آی دی رکورد ذخیره شده در دیتابیس برای مثال
            byte[] data = CFS.GetDataFromDb("BB848D45-382C-4D95-BF4E-52C3509407D4");
            if (data != null)
            {
                File.WriteAllBytes(@"C:\tst.PNG", data);
            }
        }
    }
}

```

```
    } } }
```

روش فوق با روش متدالو افزودن یک فایل به دیتابیس اس کیوال سرور هیچ تفاوتی ندارد و اینجا هم بدون مشکل کار می‌کند. اطلاعات نهایی به صورت فایل‌های بر روی سیستم که توسط اس کیوال سرور مدیریت خواهد شد و با جدول شما یکپارچه‌اند، ذخیره می‌شوند.

در روش دیگری که در اکثر مقالات مرتبط مورد استفاده است، از شیء `SqlFileStream` کمک گرفته شده و نحوه انجام آن نیز به صورت زیر می باشد.

در ابتدا دو رویه ذخیره شده زیر را ایجاد می کنیم:

```
CREATE PROCEDURE [AddFile](@Title NVARCHAR(255), @filepath VARCHAR(MAX) OUTPUT)
AS
BEGIN
    SET NOCOUNT ON;

    DECLARE @ID UNIQUEIDENTIFIER
    SET @ID = NEWID()

    INSERT INTO [tblFiles]
    (
        [ fileId ],
        [ title ],
        [ SystemFile ]
    )
    VALUES
    (
        @ID,
        @Title,
        CAST('' AS VARBINARY(MAX))
    )

    SELECT @filepath = SystemFile.PathName()
    FROM   tblFiles
    WHERE  FileId = @ID
END
GO

CREATE PROCEDURE [GetFilePath](@Id VARCHAR(50))
AS
BEGIN
    SET NOCOUNT ON;

    SELECT SystemFile.PathName()
    FROM   tblFiles
    WHERE  FileId = @ID
END
```

در رویه ذخیره شده AddFile ، ابتدا رکوردی بر اساس عنوان دلخواه ورودی با یک فایل خالی ایجاد می‌شود. سپس مسیر سیستمی این فایل را در آرگومان خروجی filepath قرار می‌دهیم. SystemFile.PathName از اس کیوال سرور 2008 جهت فیلد های فایل استریم به اس کیوال سرور اضافه شده است. از این مسیر در برنامه خود جهت نوشتن بایت های فایل مورد نظر در آن توسط شیء SqlFileStream استفاده خواهیم کرد.

```
using System.Data.SqlClient;
using System.Data;
using System.Data.SqlTypes;
using System.IO;

namespace FileStreamTest
{
    class CFSqlFileStream
    {
        /// <summary>
```

```

افزودن رکورد به جدول حاوی ستونی از نوع فایل استریم
/// </summary>
/// <param name="filePath">مسیر فایل</param>
/// <param name="title">عنوانی دلخواه</param>
public static void AddNewRecord(string filePath, string title)
{
    آیا فایل وجود دارد؟//
    if (!File.Exists(filePath))
        throw new FileNotFoundException(
            "لطفاً مسیر فایل معتبری را مشخص نمایید", filePath);

    خواندن اطلاعات فایل در آرایه‌ای از بایت‌ها//
    byte[] buffer = File.ReadAllBytes(filePath);

    using (SqlConnection objSqlCon = new SqlConnection())
    {
        //todo: کانکشن استرینگ باید از یک فایل کانفیگ خوانده شود
        objSqlCon.ConnectionString =
            "Data Source=(local);Initial Catalog=testdb2009;Integrated Security = true";
        objSqlCon.Open();

        شروع یک تراکنش//
        using (SqlTransaction objSqlTran = objSqlCon.BeginTransaction())
        {
            استفاده از رویه ذخیره شده افزودن فایل//
            using (SqlCommand objSqlCmd = new SqlCommand(
                "AddFile", objSqlCon, objSqlTran))
            {
                objSqlCmd.CommandType = CommandType.StoredProcedure;

                مشخص ساختن وضعيت و مقدار پaramتر عنوان//
                SqlParameter objSqlParam1 = new SqlParameter("@Title", SqlDbType.NVarChar, 255);
                objSqlParam1.Value = title;

                مشخص ساختن پaramتر خروجی رویه ذخیره شده//
                SqlParameter objSqlParamOutput = new SqlParameter("@filepath",
                    SqlDbType.VarChar, -1);
                objSqlParamOutput.Direction = ParameterDirection.Output;

                افزودن پaramترها به شیء کامند//
                objSqlCmd.Parameters.Add(objSqlParam1);
                objSqlCmd.Parameters.Add(objSqlParamOutput);

                احرای رویه ذخیره شده//
                objSqlCmd.ExecuteNonQuery();

                و سپس دریافت خروجی آن//
                string Path = objSqlCmd.Parameters["@filepath"].Value.ToString();
            }
        }
    }
}

استفاده خواهیم کرد زمینه تراکنش فایل استریم موجود را دریافت کرده و از آن برای نوشتن محتویات فایل//
این مورد نیز یکی از تازه‌های اس کیوال سرور 2008 است//
using (SqlCommand objCmd = new SqlCommand(
    "SELECT GET_FILESTREAM_TRANSACTION_CONTEXT()", objSqlCon, objSqlTran))
{
    byte[] objContext = (byte[])objCmd.ExecuteScalar();
    using (SqlFileStream objSqlFileStream =
        new SqlFileStream(Path, objContext, FileAccess.Write))
    {
        objSqlFileStream.Write(buffer, 0, buffer.Length);
    }
}
objSqlTran.Commit();
}

دریافت اطلاعات فایل ذخیره شده به صورت آرایه‌ای از بایت‌ها
/// </summary>
/// <param name="fileId">کلید مورد استفاده</param>
/// <returns></returns>
public static byte[] GetDataFromDb(string fileId)
{
    byte[] buffer = null;

    using (SqlConnection objSqlCon = new SqlConnection())
    {
        //todo: کانکشن استرینگ باید از یک فایل کانفیگ خوانده شود
    }
}

```

```

        objSqlCon.ConnectionString =
            "Data Source=(local);Initial Catalog=testdb2009;Integrated Security = true";
        objSqlCon.Open();

        // شروع یک تراکنش
        using (SqlTransaction objSqlTran = objSqlCon.BeginTransaction())
        {
            // استفاده از رویه ذخیره شده دریافت مسیر فایل
            using (SqlCommand objSqlCmd =
                new SqlCommand("GetFilePath", objSqlCon, objSqlTran))
            {
                objSqlCmd.CommandType = CommandType.StoredProcedure;

                // مشخص ساختن پارامتر ورودی رویه ذخیره شده و مقدار دهی آن
                SqlParameter objSqlParam1 = new SqlParameter("@ID", SqlDbType.VarChar, 50);
                objSqlParam1.Value = fileId;
                objSqlCmd.Parameters.Add(objSqlParam1);

                // اجرای رویه ذخیره شده و دریافت مسیر سیستمی فایل استریم
                string path = string.Empty;
                using (SqlDataReader sdr = objSqlCmd.ExecuteReader())
                {
                    sdr.Read();
                    path = sdr[0].ToString();
                }
            }

            // زمینه تراکنش فایل موجود را دریافت کرده و از آن برای خواندن محتویات فایل
            // این مورد نیز یکی از تازه‌های اس کیوال سرور 2008 است
            using (SqlCommand objCmd = new SqlCommand(
                "SELECT GET_FILESTREAM_TRANSACTION_CONTEXT()", objSqlCon, objSqlTran))
            {
                byte[] objContext = (byte[])objCmd.ExecuteScalar();

                using (SqlFileStream objSqlFileStream =
                    new SqlFileStream(path, objContext, FileAccess.Read))
                {
                    buffer = new byte[(int)objSqlFileStream.Length];
                    objSqlFileStream.Read(buffer, 0, buffer.Length);
                }
            }

            objSqlTran.Commit();
        }

        return buffer;
    }
}

```

در پایان برای تکمیل بحث می‌توان به مقاله‌ی مرجع زیر مراجعه کرد:

FILESTREAM Storage in SQL Server 2008

نظرات خوانندگان

نوبسنده: Yahya
تاریخ: ۱۳۸۸/۰۹/۰۹ ۱۱:۲۲:۰۳

تشکر فراوان از مطلب خوب شما

نوبسنده: حمید
تاریخ: ۱۳۹۱/۰۵/۰۶ ۲۰:۱۳

سلام
ممnon از مطلب مفیدتون
یک سؤالی که برام پیش اومند اینه که ، این فایلها که در مسیر مشخص شده ذخیره شدن ، موندگار هستن یا موقعی اند ؟

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۶ ۲۰:۱۵

ماندگار هستند.

نوبسنده: حمید
تاریخ: ۱۳۹۱/۰۵/۰۶ ۲۱:۹

خب ، اگر این فایلها به هر دلیلی حذف بشن یا ویروسی بشن روی دیتا اصلی تأثیری داره ؟

نوبسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۵/۰۶ ۲۱:۵۲

دیتای اصلی همین فایلها هستند. نحوه دسترسی به آنها اینبار از طریق استریمها است که سریار کمی دارند نسبت به حالت معمولی که کل فایل، داخل بانک اطلاعاتی ذخیره می‌شود و برای دریافت آن pool buffer موتور بانک اطلاعاتی و حافظه سیستم کاملاً درگیر و مصرف می‌شوند. در دو قسمت قبل این بحث فایل استریم به مباحث تئوری آن پرداخته شده. لطفاً آن‌ها را مطالعه کنید.

قابلیت جالبی از SQL Server 2005 به بعد به این محصول اضافه شده است که امکان ایجاد یک وب سرویس بومی را بر اساس رویه‌های ذخیره شده و یا توابع تعریف شده در دیتابیس‌های موجود، فراهم می‌سازد. این قابلیت نیازی به IIS یا هر هاست دیگری برای اجرا ندارد و توسط خود اس کیوال سرور راه اندازی و مدیریت می‌شود.

توضیحات مفصل آن را در MSDN [می‌توانید ملاحظه کنید](#) و در اینجا یک مثال عملی از آن را با هم مرور خواهیم کرد:

الف) ایجاد یک جدول آزمایشی به همراه تعدادی رکورد دلخواه در آن

```
CREATE TABLE [tblWSTest](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [f1] [nvarchar](50) NULL,
    [f2] [nvarchar](500) NULL,
CONSTRAINT [PK_tblWSTest] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

SET IDENTITY_INSERT [tblWSTest] ON
INSERT [tblWSTest] ([id], [f1], [f2]) VALUES (1, N'a1', N'a2')
INSERT [tblWSTest] ([id], [f1], [f2]) VALUES (2, N'b1', N'b2')
INSERT [tblWSTest] ([id], [f1], [f2]) VALUES (3, N'c1', N'c2')
INSERT [tblWSTest] ([id], [f1], [f2]) VALUES (4, N'd1', N'd2')
INSERT [tblWSTest] ([id], [f1], [f2]) VALUES (5, N'e1', N'e2')
SET IDENTITY_INSERT [dbo].[tblWSTest] OFF
```

ب) ایجاد یک رویه ذخیره شده در دیتابیس جاری

```
CREATE PROCEDURE GetAllData
AS
SELECT f1,
       f2
FROM   tblWSTest
```

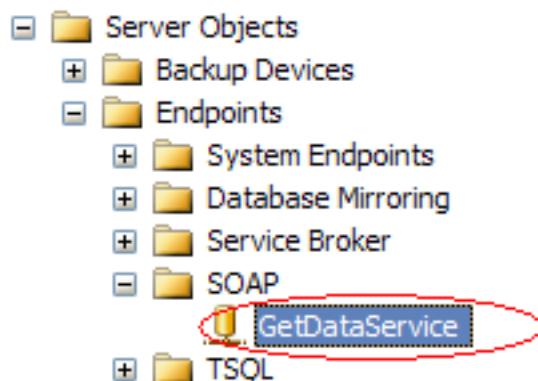
ج) ایجاد یک HTTP Endpoint

```
CREATE ENDPOINT GetDataService
    STATE = STARTED
AS HTTP(
    PATH = '/GetData',
    AUTHENTICATION = (INTEGRATED),
    PORTS = (CLEAR),
    CLEAR_PORT = 8080,
    SITE = '*'
)
FOR SOAP(
    WEBMETHOD 'GetAllData'
    (NAME = 'testdb2009.dbo.GetAllData'),
    WSDL = DEFAULT,
    DATABASE = 'testdb2009',
    NAMESPACE = DEFAULT
)
```

توضیحات:

در حالت clear و یا ssl می‌تواند باشد. همچنین برای اینکه با IIS موجود بر روی سیستم هم تداخل نکند PORT

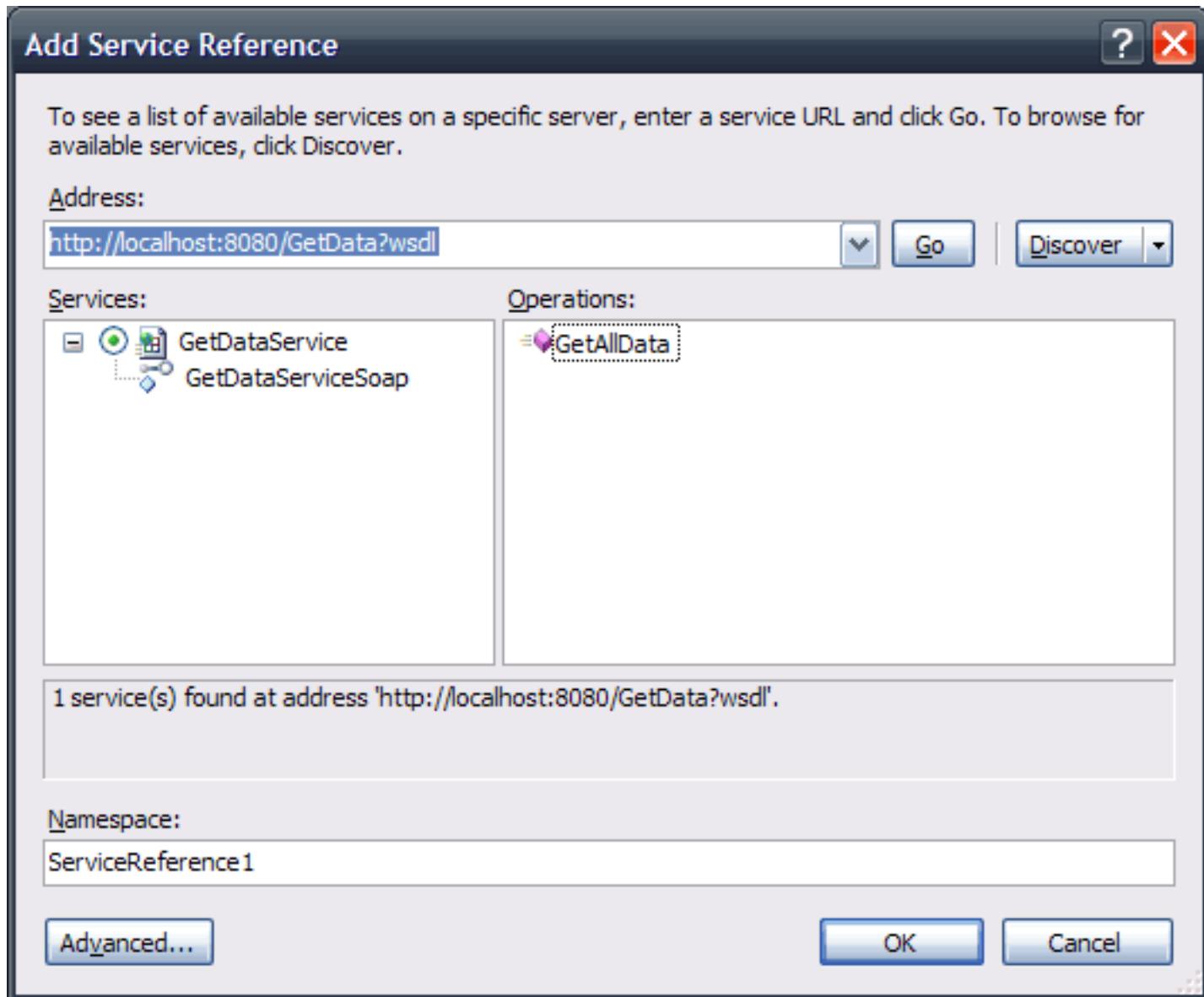
8080 تنظیم شده است. سایر پارامترهای آن بسیار واضح هستند. برای مثال تعیین دیتابیسی که این رویه ذخیره شده در آن قرار دارد و همچنین مسیر کامل دسترسی به آن دقیقا مشخص می‌گردد.



این وب سرویس هم اکنون آغاز به کار کرده است. برای مشاهده wsdl آن، آدرس زیر را در مرورگر وب خود وارد نمایید (PATH و `CLEAR_PORT` معرفی شده در endPoint اینجا بکار می‌رود):

<http://localhost:8080/GetData?wsdl>

د) استفاده از این وب سرویس در یک برنامه ویندوزی یک برنامه ساده winForms را شروع کنید. سپس یک DataGridView را بر روی فرم قرار دهید (بدیهی است این مورد می‌تواند یک برنامه ASP.Net هم باشد و موارد مشابه دیگر). سپس از منوی پروژه، یک service reference را در VS2008 بر اساس آدرس wsdl فوق اضافه کنید (شکل زیر):



برای اینکه این مثال در VS2008 درست کار کند باید فایل app.config ایجاد شده را کمی ویرایش کرد. قسمت آن را یافته و تغییرات زیر را با توجه به AUTHENTICATION مورد نیاز تغییر دهید:

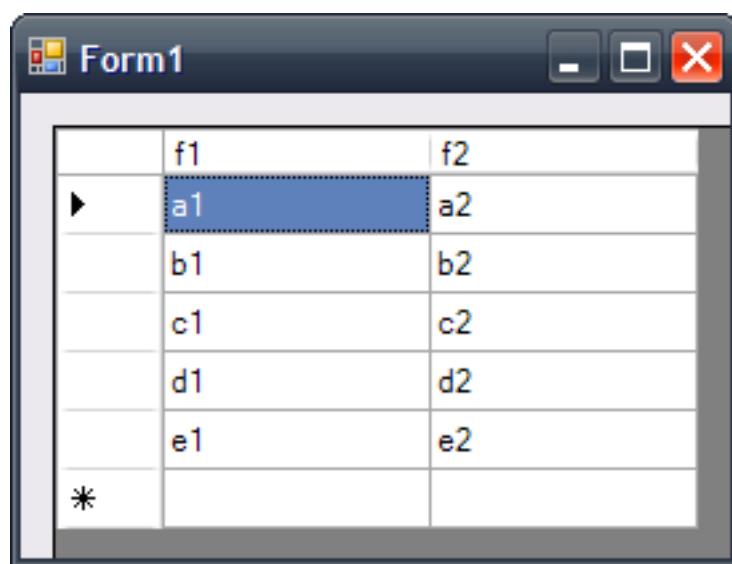
```
<security mode="TransportCredentialOnly">
    <transport clientCredentialType="Windows" proxyCredentialType="None"
        realm="" />
    <message clientCredentialType="UserName" algorithmSuite="Default" />
</security>
```

سپس کد برنامه ما به صورت زیر خواهد بود:

```
using System;
using System.Data;
using System.Windows.Forms;

namespace WebServiceTest
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

```
private void Form1_Load(object sender, EventArgs e)
{
    ServiceReference1.GetDataServiceSoapClient data =
        new ServiceReference1.GetDataServiceSoapClient();
    dataGridView1.DataSource = (data.GetAllData())[0] as DataSet;
}
}
```



نظرات خوانندگان

نویسنده: DotNetCoders
تاریخ: ۰۷/۱۲/۵۳:۱۵:۲۲

خیلی جالب بود و بسیار کارا !

فقط سوال اینجاست که میشه نوع خروجی رو مشخص کرد ؟

مثلا JSON ؟

نویسنده: وحید نصیری
تاریخ: ۰۷/۱۲/۴۰:۲۳

سلام

نه مسلم‌ما در حد WCF service های کامل نمی‌توانید ازش انتظار داشته باشید و خروجی فقط SOAP messages است.
البته می‌توانید FORMAT = NONE را در تعریف end point لحاظ کنید و بعد خروجی را آن طور که دوست دارید فرمت کرده و ارائه دهید. (باید کد نویسی کنید)

ولی کلا در این حد که نیاز به یک هاست را برطرف می‌کند و خیلی سریع بدون کد نویسی آنچنانی یک وب سرویس در اختیار شما خواهد بود، واقعاً کارآمد است.

نویسنده: Alex's Blog
تاریخ: ۰۷/۱۶/۵۴:۱۹:۱۴

واقعاً عالیه. خیلی بدرد بخوره.
ما اگه شما رو نداشتیم چیکار می‌کردیم :

نویسنده: امیر هاشم زاده
تاریخ: ۰۸/۱۴/۹۳:۱۹:۴۶

این امکان برای اوراکل 11جی هم وجود دارد؟

قسمت اول را در [این آدرس](#) می‌توانید مطالعه نمایید.

در ادامه قسمت اول، اگر بخواهیم نود جدیدی را به فیلد XML موجود اضافه کنیم، روش انجام آن به صورت زیر است (یکی از روش‌ها البته):

```
DECLARE @tblTest AS TABLE (xmlField XML)
INSERT INTO @tblTest
(
xmlField
)
VALUES
(
'<Sample>
  <Node1>Value1</Node1>
  <Node2>Value2</Node2>
  <Node3>OldValue</Node3>
</Sample>'
)
DECLARE @Name NVARCHAR(50)
SELECT @Name = 'Vahid'
UPDATE @tblTest
SET      xmlfield.modify(
           'insert element Node4 {sql:variable("@Name")}' as last into
(/Sample)[1]')
SELECT tt.xmlField
FROM   @tblTest tt
```

که حاصل آن (افزوده شدن یک المان جدید به نام Node4 بر اساس مقدار متغیر Name در انتهای لیست) به صورت زیر می‌باشد:

```
<Sample>
<Node1>Value1</Node1>
<Node2>Value2</Node2>
<Node3>OldValue</Node3>
<Node4>Vahid</Node4>
</Sample>
```

سؤال 1 :

اگر بخواهیم نام Node4 نیز متغیر باشد به چه صورتی باید مساله را حل کرد؟
در این حالت باید از کوئری‌های داینامیک استفاده کرد. باید یک رشته را ایجاد (کل عبارت update باشد یک رشته شود) و سپس از دستور exec کمک گرفت و البته باید وقت داشت در این حالت کار encoding کارکترهای غیرمجاز در XML را باید خودمان انجام دهیم.

سؤال 2 :

اگر بخواهیم نام نودها و مقادیر آنها را به صورت یک جدول معمولی بازگشت دهیم به چه صورتی باید عمل کرد؟

```
DECLARE @XML AS XML
```

```
SELECT @XML = tt.xmlField
FROM   @tblTest tt

SELECT t.c.value('local-name(..)', 'varchar(max)') AS ParentNodeName,
       t.c.value('local-name(.)', 'varchar(max)') AS NodeName,
       t.c.value('text()[1]', 'varchar(max)') AS NodeText
FROM   @XML.nodes('/**/*') AS t(c)
```

که پس از اجرای آن خواهیم داشت:

ParentNodeName	-	NodeName	-	NodeText
Sample		Node1		Value1
Sample		Node2		Value2
Sample		Node3		OldValue
Sample		Node4		Vahid

آیا می‌دانید که تا این تاریخ پس از ارائه سرویس پک یک اس کیوال سرور 2008، دقیقاً 5 به روز رسانی دیگر نیز در مورد آن منتشر شده‌اند؟

آیا می‌دانید پس از ارائه سرویس پک سه مربوط به اس کیوال سرور 2005، دقیقاً 10 مورد به روز رسانی دیگر آن نیز منتشر گردیده‌اند؟

آیا می‌دانید پس از سرویس پک 4 اس کیوال سرور 2000 دقیقاً چند مورد به روز رسانی مرتبط با آن منتشر شده‌اند؟ (72 مورد!)
آیا می‌دانید دقیقاً از چه نگارشی از SQL Server با کدام به روز رسانی‌ها استفاده می‌کنید؟

پاسخ دقیق به تمام این سؤالات را به صورت طبقه بندی شده و بسیار منظم، در وبلاگ زیر می‌توانید مشاهده نمایید:

<http://sqlserverbuilds.blogspot.com>

نظرات خوانندگان

نویسنده: Kianoosh
تاریخ: ۱۳۸۸/۰۸/۱۶ ۰۰:۳۳:۲۸

باتشکر از شما آقای مهندس نصیری

نویسنده: وحید نصیری
تاریخ: ۱۳۸۹/۰۸/۱۵ ۱۳:۳۳:۵۷

این سایت اطلاعات نگارش‌های مختلف را نگهداری می‌کند: (+)

اگر دیتابیس خود را در طی چند سال از یک نگارش به نگارشی دیگر یا از یک سرور به سروری دیگر منتقل کرده باشد، به احتمال زیاد به مشکلات Collations هم برخورده اید. یکی از فیلدهای Arabic_CI_AS است (بجا مانده از دوران قبل از 2008 SQL Server) در یک جدول و در جدولی دیگر فیلدی تازه‌ای با Collation Persian_100_CI_AS از نوع Collation تعريف شده است. ذخیره سازی و مقایسه رشته‌ها را کنترل می‌کنند. زمانیکه یک جدول جدید را در SQL Server ایجاد می‌کنیم، اگر فیلدهای Collation به صورت صریح ذکر نگردند، بر مبنای همان Collation پیش فرض دیتابیس تعريف خواهند شد.

بنابراین اگر پس از استفاده از 2008 SQL Server و تنظیم Collation پیش فرض دیتابیس به Persian_100_CI_AS، به این موارد دقت نکنیم، دیر یا زود دچار مشکل خواهیم شد.

عملیات مرتب سازی با وجود تداخلات Collations مشکل ساز نمی‌شود (خطای دریافت نمی‌کنید)، اما ممکن است الزاماً صحیح عمل نکند. مشکل از آنجایی آغاز می‌شود که قصد داشته باشیم داده‌ها را مقایسه کنیم یا join ای بین این دو جدول با فیلدهای ناهمگون از لحاظ Collation ایجاد نمائیم. در این حالت حتماً خطاهای تداخل Collation را دریافت کرده و کوئری‌های ما اجرا نخواهند شد.

Cannot resolve collation conflict for equal to operation

یک راه حل این است که در حین join به صورت صریح هر دو فیلد ذکر شده را به صورت یکسان ذکر کنیم که بیشتر یک مرهم مؤقتی است تا راه حل اصولی. برای مثال:

```

SELECT ID
FROM ItemsTable
INNER JOIN AccountsTable
WHERE ItemsTable.Collation1Col COLLATE DATABASE_DEFAULT
= AccountsTable.Collation2Col COLLATE DATABASE_DEFAULT
  
```

راه دیگر این است که مشخص کنیم که کدام فیلدها در دیتابیس Collation پیش فرض دیتابیس تطابق ندارند. سپس بر اساس این لیست شروع به تغییر Collations نمائیم. اسکریپت زیر تمام فیلدهای ناهمگون از لحاظ Collation دیتابیس جاری را برای شما لیست خواهد کرد:

```

DECLARE @defaultCollation NVARCHAR(1000)
SET @defaultCollation = CAST(
    DATABASEPROPERTYEX(DB_NAME(), 'Collation') AS NVARCHAR(1000)
)
SELECT C.Table_Name,
    Column_Name,
    Collation_Name,
    @defaultCollation DefaultCollation
FROM   Information_Schema.Columns C
    INNER JOIN Information_Schema.Tables T
        ON  C.Table_Name = T.Table_Name
WHERE  T.Table_Type = 'Base Table'
    AND RTRIM(LTRIM(Collation_Name)) <> RTRIM(LTRIM(@defaultCollation))
    AND COLUMNPROPERTY(OBJECT_ID(C.Table_Name), Column_Name, 'IsComputed') = 0
ORDER BY
    C.Table_Name,
    C.Column_Name
  
```

برای مثال جهت تغییر Collation فیلد Serial از جدول tblArchive از نوع nvarchar با طول 200 به Persian_100_CI_AS می‌توان از دستور T-SQL زیر استفاده کرد:

```
ALTER TABLE [tblArchive] ALTER COLUMN [Serial] nvarchar(200) COLLATE Persian_100_CI_AS not null
```


نظرات خوانندگان

نویسنده: reza
تاریخ: ۱۳۸۸/۰۹/۰۱ ۱۷:۲۲:۱۵

آریان

میشه اینطوری هم انجام داد

```
, SELECT sys.tables.name AS TableName , sys.columns.name AS ColumnName
      sys.columns.is_nullable , sys.columns.collation_name
                           FROM sys.columns
INNER JOIN sys.tables ON sys.columns.object_id=sys.tables.object_id
(((WHERE sys.columns.collation_name<>(CAST(DATABASEPROPERTYEX(DB_NAME() , 'Collation') AS NVARCHAR(1000
```

در مورد روش‌های کاهش حجم لاغ فایل‌های SQL Server در [این مطلب](#) بحث شد.
اما یکی از دیتابیس‌های قدیمی shrink نمی‌شد و پیغام خطای زیر را صادر می‌کرد:

```
Cannot shrink log file 2 because of minimum log space required.
```

یکی از علت‌هایی که اگر مطابق روش ذکر شده در مقاله یاده شده رفتار شود، سبب کاهش حجم لاغ فایل یک دیتابیس نمی‌شود، وجود تراکنش‌های کامل نشده است. جهت مشاهده‌ی وضعیت تراکنش‌های یک دیتابیس می‌توان دستور زیر را صادر کرد:

```
DBCC OPENTRAN
```

که نتیجه به صورت زیر بود:

```
Replicated Transaction Information:
```

```
Oldest distributed LSN      : (0:0:0)
```

(Oldest non-distributed LSN : (5291:25:1

وجود سطر مربوط به Oldest non-distributed LSN به این معنا است که هنوز یک replication نا تمام بر روی این دیتابیس موجود است. البته چون این دیتابیس از یک سرور دیگر به اینجا منتقل شده بود و هیچ نوع replication ای هم در این سرور بر روی آن تنظیم نشده بود؛ بنابراین ابتدا این replication حذف شد:

```
exec sp_removedbreplication 'dbName', 'both';
```

سپس مجددا دستور زیر جهت مشاهده‌ی وضعیت تراکنش‌های ناتمام صادر شد:

```
DBCC OPENTRAN
```

که این‌بار دیگر هیچ خروجی نداشت.
اکنون با استفاده از روش ذکر شده، لاغ فایل 70 گیگابایتی این دیتابیس به سادگی به چند مگابایت shrink شد.

نظرات خوانندگان

نوبت‌دهنده: www.Abshar-System.ir
تاریخ: ۱۳۹۰/۰۱/۲۲ ۱۱:۰۴:۲۶

آقای نصیری تشكر

جدول زیر تعداد باغهای امنیتی Oracle11g و SQL server 2008 را تا ماه نوامبر 2009 نمایش می‌دهد:

Vulnerabilities	Advisories	Product
0	0	SQL Server 2008
239	7	Oracle11g

و به صورت خلاصه مایکروسافت در 6 سال گذشته تنها 59 باغ امنیتی وصله شده مربوط به نگارش‌های مختلف SQL Server داشته است (از نگارش 2000 به بعد). در طی همین مدت اوراکل (نگارش‌های 8 تا 10) تعداد 233 وصله امنیتی را ارائه داده است. در سال 2006، اس کیوال سرور 2000 با سرویس پک 4، به عنوان امن‌ترین بانک اطلاعاتی موجود در بازار شناخته شد (به همراه PostgreSQL). در همین زمان Oracle10g در قعر این جدول قرار گرفت.

اعداد و آمار از سایت secunia.com استخراج شده است: [+](#) و [-](#)

نظرات خوانندگان

نویسنده: behnam
تاریخ: ۱۳۸۸/۰۹/۲۸

سایت seunia.com تو این زمینه معتبره؟
من تا امروز بر عکس این جریان فکر می کردم!

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۹/۲۹

بله. سایت بی طرفی است و یکی از کارهایش جمع کردن آمارهای اینگونه است.

نویسنده: سار
تاریخ: ۱۳۸۸/۰۹/۲۹

مدیر بخش نرم افزار ما که یک پژوهش هست اگر این رو ببینه سکته می کنه D:

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۰۹/۲۹

کم شدن باگ های امنیتی مایکروسافت در چند سال اخیر سه دلیل عمدۀ داشته:
- استفاده بیشتر از دات نت فریم ورک در محصولات عمدۀ خودش (اس کیوال سرور، SharePoint، exchange server و ...).
برای مثال دات نت فریم ورک به حملات سرریز بافر (اگر کد خالص دات نتی باشد و از API ویندوز به صورت ناشیانه استفاده نکرده باشد) مقاوم است و خیلی مسائل دیگر که به صورت توکار در دات نت لحظه شده.
- وضع کردن یک سری قوانین سفت و سخت در مورد کتابخانه ها و توابع C مورد استفاده در محصولات نوشته شده با سی و CPP
<http://www.dotnettips.info/2009/05/bannedh.html>
- تهیه برنامه هایی که کار آنالیز خودکار فایل های باینری و همچنین سورس کدهای سازمان آن را انجام می دهند. به این صورت قبل از اینکه دیگران از سیستم های آنها باگ های متدائل را درآورند، خودشان نسبت به اینکار اقدام می کنند.
<http://blogs.msdn.com/bharry/archive/2009/10/02/two-free-security-tools-from-microsoft-sdl-team.aspx>
<http://www.dotnettips.info/2009/11/blog-post.html>

نویسنده: Hamidreza
تاریخ: ۱۳۸۸/۱۰/۰۲

یه موضوعی که تو این آمار بهش اشاره نشده تعداد و میزان حرفه ای بودن پروژه هایی هست که دارن از این دیتاپیس ها استفاده میکنن. ممکنه این آمار اینجوری بدبست اوامده باشه که مثلا 10 تا سرور اوراکل تپل رو اوامده مقایسه کرده باشه با 2 تا سرور اس کیو ال معمولی و در این حالت باگ ها رو شمرده باشه. طبیعتاً باگ های امنیتی اونی که در پروژه های بیشتری استفاده شده و سطح پروژه هم اینترپرایزتر هست ملموس تر هست. مثله قضیه ویندوز و لینوکس منظورم هست که اگه فرآگیری ویندوز رو لینوکس هم داشت مطمئناً مورد حمله های بیشتری به نسبت ویندوز قرار می گرفت.

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۱۰/۰۳

- این آمار فقط بر اساس تعداد وصله های ارائه شده است نه بر اساس تعداد پروژه ها یا تعداد سرورها و غیره.
- کسانی که در این زمینه فعالیت می کنند و از محصولات باگ در می آورند عموماً در شرکت های امنیتی کاری به سرورهای شما یا پروژه های شما ندارند. مطالعات و مهندسی معکوس خودشون رو در آزمایشگاه های مجازی تهیه شده انجام می دن.
- علت استقبال از اوراکل در یک سری از شرکت ها داشتن نسخه های لینوکسی آن است.

مايكروسافت مقایسه‌ای رو بین آخرین نگارش‌های اين مخصوصات اينجا انجام داده که فارق از شرکت تهيه کننده آن آمارهای جالبی را ارائه می‌دهد:

<http://www.microsoft.com/sqlserver/2008/en/us/compare-oracle.aspx>

دربافت خطای database is not accessible و نحوهی رفع مشکل

عنوان: دربافت خطای database is not accessible
نوبتندۀ: وحید نصیری
تاریخ: ۱۳۸۸/۱۰/۱۳ ۱۳:۲۷:۰۰
آدرس: www.dotnettips.info
برچسب‌ها: SQL Server

ممکن است هنگام تلاش جهت اتصال به دیتابیس اس کیوال سرور 2005 به بعد از طریق management studio با پیغام خطای زیر مواجه شوید:

The database XYZ is not accessible. (ObjectExplorer)

و یا اگر بر روی همین دیتابیس کلیک راست کرده و به خواص آن مراجعه کنیم، خطای 952 زیر صادر شود:

Database 'XYZ' is in transition. Try the statement later. (Microsoft SQL Server, Error: 952)

اصلاً نگران نباشید؛ هیچ مشکلی نیست!

ابتدا رویه‌ی ذخیره شده‌ی sp_who2 را اجرا کنید. یک لیست از کانکشن‌های باز به دیتابیس‌های موجود را به شما خواهد داد. در این لیست به دنبال کانکشن‌های موجود به دیتابیسی که این خطا را می‌دهد بگردید. Pid این کانکشن‌ها را یافته و سپس با دستور kill آن‌ها را از بین ببرید. مشکل حل خواهد شد.
عموماً نبستن خود management studio سبب این مشکل می‌شود. بنابراین حتی یکبار باز و بسته کردن آن نیز باید این مشکل را برطرف کند (یا تمام management studio های متصل، که البته راه ساده‌تر همان kill کردن pid آن‌ها است).

نظرات خوانندگان

نویسنده: Mohsen
تاریخ: ۱۳۸۸/۱۰/۱۳ ۱۴:۰۵:۳۲

جالب و مفید بود .مرسى

راستی به من هم سر بزن تازه سایتم رو راه انداختم

<http://www.msalehi.net>

نویسنده: وحید نصیری
تاریخ: ۱۳۸۸/۱۰/۱۳ ۱۴:۱۹:۲۴

سلام

موفق باشید. یک مطلب هم در مورد معرفی مطالب وبلاگ‌ها اینجا نوشتدم

http://www.dotnettips.info/2009/09/blog-post_04.html

جهت یافتن و کشف وبلاگ‌های جدید.

مطلوب "[منسخ شده‌ها در نگارش‌های جدید SQL server](#)" را احتمالا به خاطر دارید. جهت تکمیل آن، کوئری زیر را هم می‌توان ذکر کرد:

```
SELECT instance_name,
       cntr_value
  FROM sys.dm_os_performance_counters
 WHERE OBJECT_NAME = 'SQLServer:Deprecated Features'
   AND cntr_value > 0
ORDER BY
       cntr_value DESC
```

توسط این کوئری گزارشی از منسخ شده‌های مورد استفاده در دیتابیس‌های شما ارائه می‌شود. برای مثال چندبار از text و ntext استفاده کرده‌اید، آیا هنوز [compatibility level](#) دیتابیس‌های خود را تغییر نداده‌اید و [مثال‌هایی از این دست](#).

برای مثال جهت یافتن سریع فیلد‌های منسخ شده text و image دیتابیس جاری از کوئری زیر می‌توان کمک گرفت:

```
SELECT O.Name,
       col.name AS ColName,
       systypes.name
  FROM syscolumns col
 INNER JOIN sysobjects o
    ON col.id = o.id
 INNER JOIN systypes
    ON col.xtype = systypes	xtype
 WHERE o.Type = 'U'
   AND OBJECTPROPERTY(o.ID, N'IsMSShipped') = 0
   AND systypes.name IN ('text', 'ntext', 'image')
ORDER BY
       O.Name,
       Col.Name
```

یکی از قابلیت‌های جالب SQL Server در یک شبکه محلی امکان link و اتصال آن‌ها به یکدیگر است. به این صورت امکان کوئری گرفتن (و یا اعمال متداول SQL ای) از دو یا چند سرور مختلف با دستورات T-SQL میسر می‌شود؛ به نحوی که حس یکپارچگی دیتابیس‌های این سرورها را حین کوئری نوشتن خواهیم داشت.
برای مثال فرض کنید دو سرور SQL1 و SQL2 را در شبکه داریم. می‌خواهیم در سرور SQL1 اتصالی را به سرور SQL2 ایجاد کنیم.

```
USE master
```

```
EXEC sp_addlinkedserver
    'SQL2',
    N'SQL Server'

sp_addlinkedsrvlogin @useself='false ', @rmtsrvname = 'SQL2',
@rmtuser = 'sa',
@rmtpassword = 'pass#'
```

دستورات T-SQL فوق کار ثبت یک linked server جدید و اعمال مشخصات کاربری که توسط آن قرار است به سرور SQL2 دسترسی داشت، انجام می‌دهند.

اکنون جهت بررسی این اتصال در سرور SQL1 کوئری زیر را اجرا می‌کنیم:

```
select * from sql2.faxManager.dbo.tblErja
```

که نحوه‌ی فراخوانی جدول مورد نظر باید به صورت Server.DatabaseName.dbo.TableName در آن رعایت شود.
تا اینجا همه چیز خوب است. مشکل از زمانی شروع می‌شود که بخواهیم تراکنش‌ها را نیز دخالت دهیم و اصولی کار کنیم. برای مثال:

```
begin distributed tran
select * from sql2.faxManager.dbo.tblErja
commit tran
```

خطایی که در ویندوز سرور 2003 با آخرین به روز رسانی‌ها ظاهر می‌شود به صورت زیر است:

The operation could not be performed because OLE DB provider for linked server was unable to begin a distributed transaction.
OLE DB provider for linked server returned message "The partner transaction manager has disabled its support for remote/network transactions.".

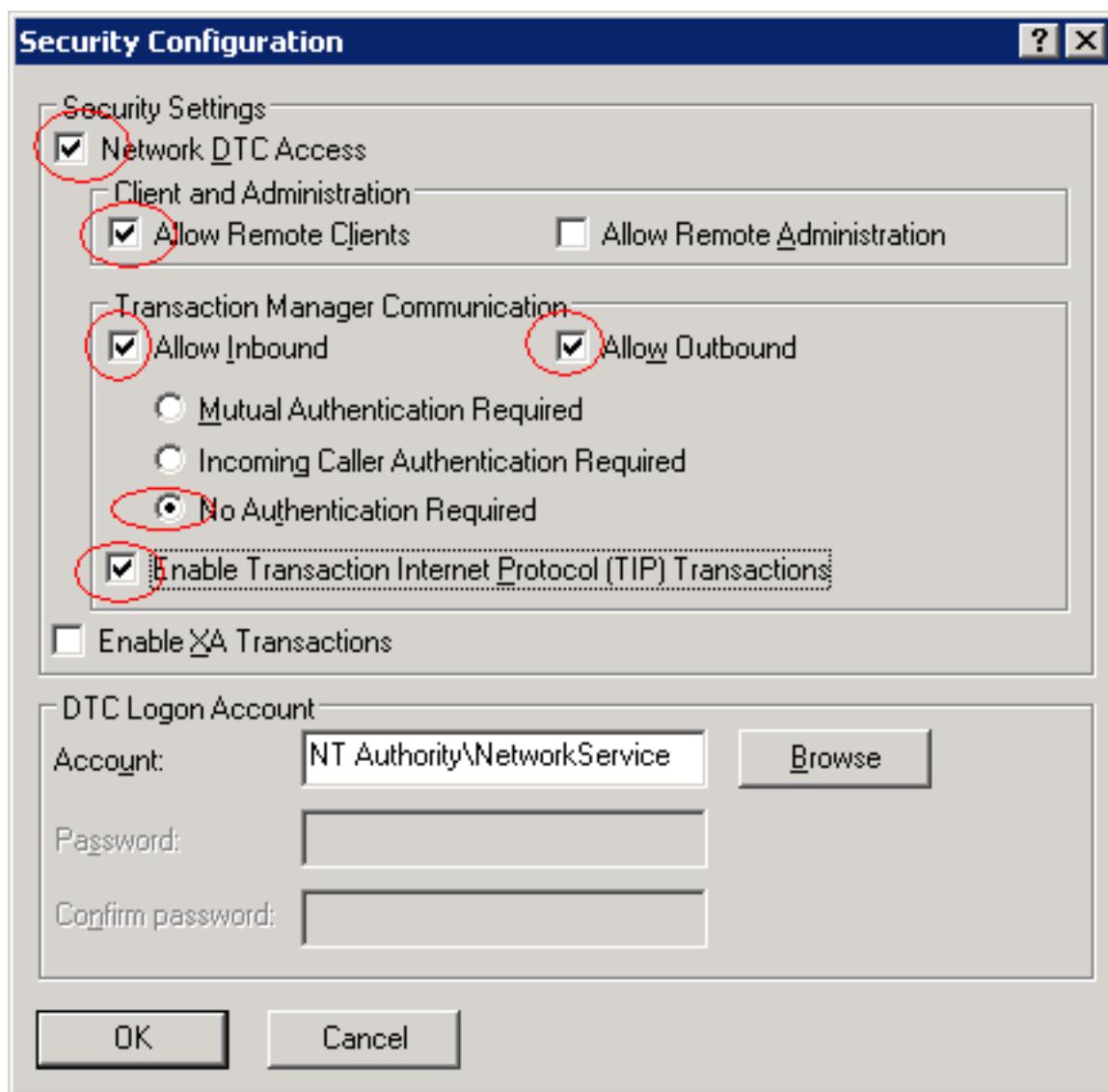
به صورت پیش فرض این نوع تراکنش‌های توزیع شده غیرفعال هستند مگر اینکه فعال شوند و روش حل مشکل نیز به صورت زیر می‌باشد:

قبل از هر کاری به کنسول سرویس‌های ویندوز مراجعه کرده و از در حال اجرا بودن سرویس Distribute Transaction Coordinator اطمینان حاصل کنید.

سپس به قسمت زیر مراجعه نمائید:

Control Panel > Administrative Tools > Component Services

نود مربوط به Component Service را گشوده و سپس بر روی My Computer کلیک راست کرده و گزینه‌ی خواص را انتخاب کنید.
در صفحه‌ی بازه شده به برگه‌ی MSTDC مراجعه کرده و بر روی دکمه‌ی Security Configuration کلیک نمایید.
اکنون تنظیمات آن را مطابق شکل زیر تغییر دهید.



این تنظیم باید بر روی هر دو سرور SQL1 و SQL2 انجام شود.

پس از این تغییرات که شامل راه اندازی مجدد سرویس Distribute Transaction Coordinator نیز خواهد شد، مشکل خطای فوق برطرف شده و امکان استفاده از تراکنش‌ها در linked servers نیز میسر می‌شود.

مشکل دیگری که به آن برخوردم خطای زیر است:

Unable to start a nested transaction for OLE DB provider for linked server . A nested transaction was required because the XACT_ABORT option was set to OFF.
OLE DB provider for linked server returned message "Cannot start more transactions on this session.".

برای حل این مشکل یک سطر زیر را باید به ابتدای کوئری خود اضافه کرد که جزو الزامات تراکنش‌های توزیع شده است و به این صورت از rollback کامل تمامی دستورات موجود فرآخوانی شده T-SQL در صورت بروز کوچکترین خطایی اطمینان حاصل می‌کند:

```
SET XACT_ABORT ON
```

برای مطالعه بیشتر:

[MSDTC Troubleshooting](#)

عموماً هنگام نصب SQL Server ، پیش و پس از آن، بهتر است موارد زیر جهت بالا بردن کیفیت و کارآیی سرور، رعایت شوند:

1- پیش فرض‌های نصب SQL در مورد محل قرارگیری فایل‌های دیتا و لاغ و غیره صحیح نیست. هر کدام باید در یک درایو مجزا مسیر دهی شوند برای مثال:

Data drive D:
Transaction Log drive E:
TempDB drive F:
Backup drive G:

این مورد SharePoint را کسانی که با TempDB کار کرده باشند به خوبی علتش را درک خواهند کرد. پیش فرض نصب افراد تازه کار، نصب SQL Server و تمام مخلفات آن در همان درایو ویندوز است (یعنی همان چندبار کلیک بر روی دکمه Next برای نصب). SharePoint هم به نحو مطلوبی تمام کارهایش مبتنی بر transactions است. یعنی استفاده‌ی کامل از TempDB . نتیجه؟ پس از مراجعه به درایو ویندوز مشاهده خواهید کرد که فقط چند مگ فضای خالی باقی مانده! حالا اینجا است که بدو این مقاله و اون مقاله رو بخون که چطور TempDB را باید از درایو C به جای دیگری منتقل کرد. چیزی که همان زمان نصب اولیه SQL Server باید در مورد آن فکر می‌شد و نه الان که سیستم از کار افتاده.

همچنین وجود این مسیرهای مشخص و پیش فرض و آگاهی از سطوح دسترسی مورد نیاز آن‌ها، از سر دردهای بعدی جلوگیری خواهد کرد. برای مثال : [انتقال فایل‌های دیتابیس اس کیوال سرور 2008](#)

2- پس از رعایت مورد 1 ، نوبت به تنظیمات آنتی ویروس نصب شده روی سرور است. این پوشش‌های ویژه را که جهت فایل‌های دیتا و لاغ و غیره بر روی درایوهای مختلف معرفی کرده‌اید یا خواهید نمود، باید از تنظیمات آنتی ویروس شما Exclude شوند. همچنین در حالت کلی فایل‌هایی با پسوندهای LDF/MDF/NDF باید جزو فایل‌های صرفنظر شونده از دید آنتی ویروس شما معرفی گرددند.

این مورد علاوه بر بالا بردن کارآیی SQL Server ، در حین Boot سیستم نیز تاثیر گذار است. گاهای دیده شده است که آنتی ویروس‌ها این فایل‌های حجمی را در حین راه اندازی اولیه سیستم، پیش از SQL Server ، جهت بررسی گشوده و به علت حجم بالای آن‌ها این قفل‌ها تا مدتی رها نخواهند شد. در نتیجه آغاز سرویس SQL Server را با مشکلات جدی مواجه خواهند کرد که عموماً عیب یابی آن کار ساده‌ای نیست.

3- پیش فرض میزان حافظه‌ی مصرفی SQL Server صحیح نیست. این مورد باید دقیقاً بلافصله پس از پایان عملیات نصب اولیه اصلاح شود. برای مطالعه بیشتر: [تنظیمات پیشنهادی حداقل حافظه‌ی مصرفی اس کیوال سرور](#)

4- آیا مطمئن هستید که از تمام امکانات نگارش جدید SQL Server ای که نصب کرده‌اید در حال استفاده می‌باشید؟
برای مطالعه بیشتر: [تنظیم درجه سازگاری یک دیتابیس اس کیوال سرور](#)

5- بهتر است فشرده سازی خودکار بک آپ‌ها در 2008 SQL Server فعال شوند.
برای مطالعه بیشتر: [+ برای مطالعه بیشتر](#)

6- از paging بیش از حد اطلاعات، از حافظه‌ی فیزیکی سرور به virtual memory و انتقال آن به سخت دیسک سیستم جلوگیری کنید. برای این منظور:
در قسمت Run ویندوز تاپیک کنید : GPEDIT.MSC و پس از اجرای آن با مراجعه به Group policy editor ظاهر شده به مسیر زیر مراجعه کنید:

windows settings -> security settings -> local policies -> user rights assignment -> lock pages in memory

در اینجا به یوزر اکانت سرویس SQL Server دسترسی lock pages in memory را بدهید.
علاوه بر آن در همین قسمت (user rights assignment) گزینه‌ی "Perform Volume Maintenance tasks" را نیز یافته و دسترسی لازم را به یوزر اکانت سرویس SQL Server بدهید.

7- به روز رسانی اطلاعات آماری SQL Server را به حالت غیرهمزمان تنظیم کنید.
اگر مطالب مرتبط با SQL Server این سایت را مرور کرده باشید حتماً با یک سری [DMV](#) که دقیقاً به شما خواهد گفت بر اساس اطلاعات آماری جمع شده برای مثال بهتر است روی چه فیلدهایی Index درست کنید، آشنا شده‌اید. حالت پیشفرض به روز رسانی این اطلاعات آماری，[synchronous](#) است یا همزمان. به این معنا که تا اطلاعات آماری یک کوئری ذخیره نشود، حاصل کوئری به کاربر بازگشت داده نخواهد شد که این امر می‌تواند بر روی کارآیی سیستم تاثیر گذار باشد. اما امکان تنظیم آن به حالت غیر همزمان نیز مطابق کوئری‌های زیر وجود دارد (این مورد از SQL Server 2005 به بعد اضافه شده است):

```
ALTER DATABASE dbName SET AUTO_UPDATE_STATISTICS ON  
ALTER DATABASE dbName SET AUTO_UPDATE_STATISTICS_ASYNC ON
```

8- نصب آخرین سرویس پک موجود فراموش نشود. برای مثال [این سایت](#) آمار تمام به روز رسانی‌ها را نگهداری می‌کند.

9- حتماً رویه‌ای را برای تهیه بک آپ‌های خودکار پیش بینی کنید. برای مثال : [+](#)

10- میزان فضای خالی باقیمانده درایوهای سرور را مونیتور کنید. اطلاعات بیشتر: [+](#)

11- با نصب سرور جدید و تنظیم collation آن به فارسی، به نکات "[یافتن تداخلات SQL Server در Collations](#)" دقت داشته باشید.

نظرات خوانندگان

نویسنده: Mohammad Shams Javi
تاریخ: ۲۰:۲۱:۲۷ ۱۳۸۹/۰۷/۲۳

سلام

خیلی مطلب خوب و مفیدی بود آقای نصیری، خصوصاً موارد ۲ و ۶ و ۷ جالب و آموزنده بودند.

نویسنده: Amir
تاریخ: ۱۴:۵۰:۴۸ ۱۳۸۹/۰۷/۲۸

مطلوب بسیار مفید و کاربردی بود. از اهتمامی که در تهیه این مطالب دارید سپاسگزارم.

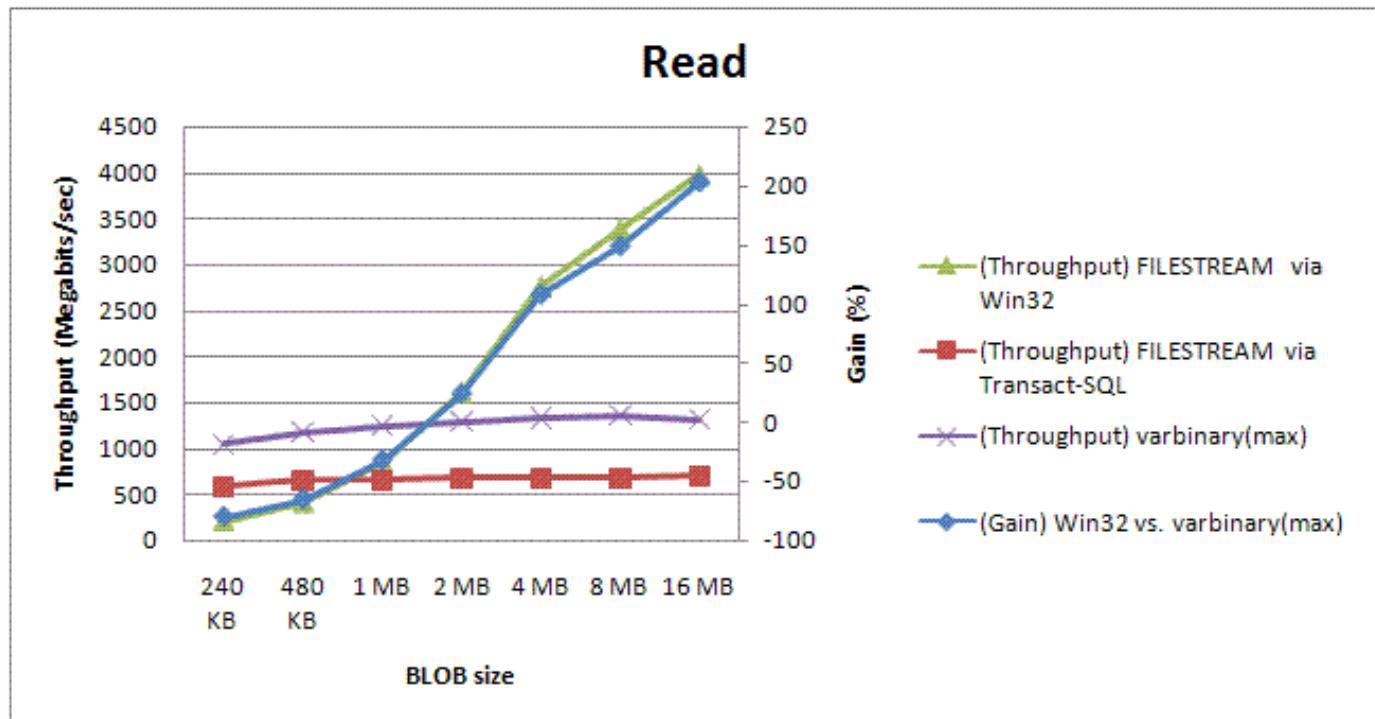
نویسنده: مهدی پایروند
تاریخ: ۱۶:۱۸ ۱۳۹۱/۰۷/۱۰

در اینجا به یوزر اکانت سرویس SQL Server دسترسی `lock pages in memory` را بدهید. دقیقاً منظورتون کدوم یوزر هست.

نویسنده: محمد صاحب
تاریخ: ۱۶:۳۳ ۱۳۹۱/۰۷/۱۰

یوزری که سرویس MSSQLSERVER با اون بالا میاد از قسمت `sql server configuration manager` میتوانی چکش کنی...

سؤال: از نمودار زیر چه چیزی را برداشت می‌کنید؟!



منحنی که بالا رفته یعنی چی؟ یعنی بد، خوبه؟!

منحنی‌های پایین‌تر یعنی چی؟ این‌ها بهترند یا بالای‌های آن‌ها؟

با بالا رفتن حجم فایل‌ها، کدام یک کارآیی بهتری دارد؟ بالای‌ها یا پایین‌ها؟

ماخذ این نمودار: ([+](#)). البته قبل از مراجعه به ماخذ و مطالعه آن، سعی کنید به سوالات فوق پاسخ دهید.

نظرات خوانندگان

نویسنده: alireza

تاریخ: ۱۳۹۰/۰۳/۱۴ ۰۷:۱۸:۳۰

حرف حسابش این هست که استفاده از فایل استریم با استفاده از ویندوز آی پی آی خیلی بهتر از استفاده از این شکل دیتا با استفاده از ترنسیزیکت هست. یعنی اگر میخواهی با دستورات اس کیو ال با این شکل دیتا کار کنی پرفورمنس خوبی نخواهد داشت. و حتی در این حالت استفاده از وارباینری جواب بهتری میدهد . همچنین همانطور که مشخص است این شکل داده ها از 2 مگابایت بالاتر تازه جواب میدهد

یکی از موارد مشکل ساز حین استفاده از T-SQL ، [مقدار دهی اولیه متغیرها به نال](#) است و اگر اسکریپت تهیه شده کمی طولانی باشد، خطایابی مشکلات مرتبط با آن بسیار مشکل می‌شود. برای مثال:

```

Declare
@x int,
@y int

Set @x = 1
If (@x + @y = 1)
BEGIN
    print 'yes!'
End

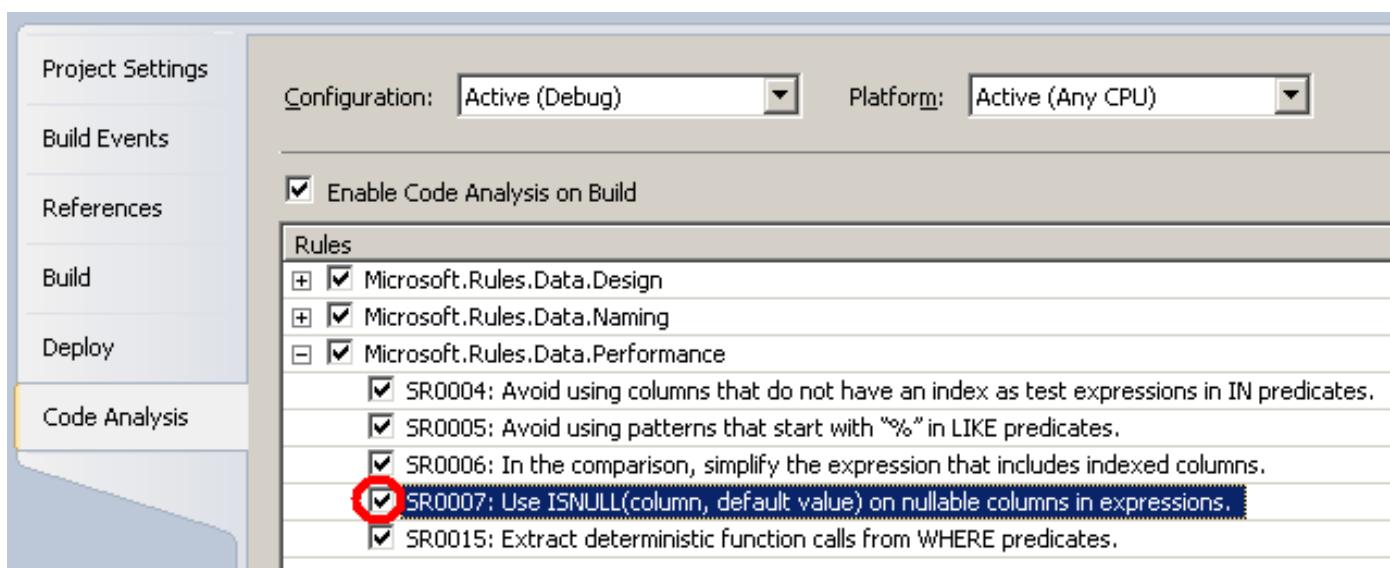
Set @y = (select sum(id) from Account)
If @x + @y = 1
BEGIN
    print 'yes!'
End

```

کد فوق بدون هیچگونه خطایابی اجرا می‌شود و هیچ وقت هم yes را چاپ نمی‌کند. مشکل هم همینجا است. خطایابی قسمت دوم این اسکریپت کمی مشکل‌تر از حالت قبل است. چون در اینجا به نظر متغیر `y` صریحاً مقدار دهی شده است؛ اما در عمل ممکن است برای مثال به دلیل عدم وجود رکوردی در جدول Account ، باز هم null به آن نسبت داده شود.

بنابراین سؤال این است که چگونه این نوع مشکلات را در یک پروژه با تعداد زیادی رویه ذخیره شده، تابع و غیره می‌توان تشخیص داد؟
پاسخ:

در این مورد قبلاً مطلبی در این سایت منتشر شده [+] (البته اگر از نگارش کامل VS 2010 استفاده می‌کنید نیازی به نصب چیزی نخواهید داشت) و نکته‌ی آن بررسی [SR0007](#) است.



نظرات خوانندگان

نوبنده: سپیده
تاریخ: ۱۴۰۱/۰۴/۰۱

خیلی جالب بود، مرسى!

عنوان: اگر نصب سرویس پک اس کیوال سرور Fail شد ...
نویسنده: وحید نصیری
تاریخ: ۱۳۹۰/۰۷/۱۹ ۲۲:۵۱:۰۰
آدرس: www.dotnettips.info
برچسبها: SQL Server

همانطور که مطلع هستید سرویس پک سه SQL Server چند روزی است که منتشر شده . این به روز رسانی بر روی یک سرور بدون مشکل نصب شد؛ در سرور دیگر به علت داشتن یک سری برنامه امنیتی مزاحم (که مثلا دسترسی به رجیستری را مونیتور و سد می کنند) با شکست مواجه و در آخر پیغام Fail نمایش داده شد. مجددا آنرا اجرا کرد، سریع تمام مراحل را تمام کرد باز هم Fail را نمایش داد.

خوب؛ گفتم احتمالا مشکلی نیست. سعی کردم به سرور وصل شوم ... پیغام «این سرور دسترسی از راه دور را نمی پذیرد» و از این حرفهای متداول ظاهر شد. به لاغ موجود در log Event ویندوز که مراجعه کردم پیغام خطای زیر نمایان بود:

```
Script level upgrade for database '
```

master

```
' failed because upgrade step 'sqlagent100_msdb_upgrade.sql' encountered error 5597, state 1, severity 16. This is a serious error condition which might interfere with regular operation and the database will be taken offline. If the error happened during upgrade of the 'master' database, it will prevent the entire SQL Server instance from starting. Examine the previous errorlog entries for errors, take the appropriate corrective actions and re-start the database so that the script upgrade steps run to completion.
```

اوه! اوه! در این **لحظهی عرفانی** ، دیتابیس master نابود شده! نمی شود وصل شد. سروری که داشت تا مدتی قبل بدون هیچ مشکلی کار می کرد، الان دیگر حتی نمی شود به آن وصل شد. به کنسول سرویس های ویندوز مراجعه کردم (services.msc)، سعی کردم سرویس اس کیوال را که از کار افتاده دستی اجرا کنم، پیغام زیر مجددا در log event ظاهر شد:

```
FILESTREAM feature could not be initialized. The Windows Administrator must enable FILESTREAM on the instance .using Configuration Manager before enabling through sp_configure قابلیت FILESTREAM را نمی تواند آغاز کند. پس از مدتی جستجو مشخص شد که این مورد را می شود در رجیستری ویندوز غیرفعال کرد؛ به صورت زیر:
```

- 1) Open up Registry Editor
- 2) Go To HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL10.MSSQLServer\MSSQLSERVER\FileStream
- 3) Edit the value "EnableLevel" and set it to 0
- 4) Restart SQL Server.

پس از انجام اینکار، سرویس اس کیوال استارت شد (از طریق کنسول سرویس های ویندوز). در ادامه، امکان اتصال به آن نیو (حتی با اکانت sa):

```
Login failed for user 'sa'. Reason: Server is in
```

script upgrade mode

. Only administrator can connect at this time. (Microsoft SQL Server, Error: 18401)

باز هم پس از مدتی جستجو [معلوم گردید](#) که «کمی باید صبر کرد». آن پیغام اول کار مبتنی بر تخریب دیتابیس master هم بیمورد است. پس از fail شدن نصب سرویس پک، هنوز برنامه نصب آن در پشت صحنه مشغول به کار است. این مورد به وضوح در task manager ویندوز مشخص است. سرور به مدت 15 دقیقه به حال خود رها شد. پس از آن بدون مشکل اتصال برقرار گردید و همه چیز مجدداً شروع به کار کرد.

بنابراین اگر در حین نصب سرویس پک SQL Server مشکلی پیش آمد، نگران نباشید. باید به نصب آن زمان داد (برنامه mscorsw.exe در پشت صحنه مشغول به کار است). برنامه نصب آن هم هیچ نوع خطای مفهومی را گزارش نمی‌دهد. تمام مراحل، بجای نمایش در برنامه تمام صفحه نصب آن، در log event در یک progress bar ثبت می‌شود. این برنامه تمام صفحه فقط کارش نمایش یک است!

اگر ... هیچکدام از این موارد جواب نداد، امکان بازسازی دیتابیس master نیز وجود دارد: [[^](#), [^](#)]
ولی دست نگه دارید و سریع اقدام نکنید. ابتدا به task manager مراجعه کنید. آیا برنامه mscorsw.exe در حال اجرا است؟ اگر بله، یعنی هنوز کار نصب تمام نشده. حداقل یک ربع باید صبر کنید.

اگر نصب سرویس پک اس کیوال سرور Fail شد ...

نظرات خوانندگان

نوبنده: مجتبی صحرائی
تاریخ: ۱۳۹۰/۰۷/۱۹ ۲۳:۰۶:۵۸

ممنون از به اشتراک گذاشتن این نکته

عنوان: Microsoft® SQL Server® 2012
نویسنده: وحید نصیری
تاریخ: ۰۹:۳۷:۰۰ ۱۳۹۰/۱۲/۱۹
آدرس: www.dotnettips.info
برچسب‌ها: SQL Server

نگارش نهایی 2012 Microsoft® SQL Server® چند روزی هست که [ارائه شده](#). فعلاً نسخه آزمایشی RTM آن در اختیار عموم است.
در ادامه جمع آوری لینک‌های مرتبط به این ارائه را مشاهده خواهید نمود:

[Microsoft® SQL Server® 2012 Express and LocalDB](#)
[Microsoft® SQL Server® 2012 Evaluation](#)
[Microsoft® SQL Server® StreamInsight 2.0](#)
[Microsoft® SQL Server® 2012 Feature Pack](#)
[Microsoft® SQL Server® 2012 Semantic Language Statistics](#)
[Microsoft®SQL Server® 2012 FileStream Driver - Windows Logo Certification](#)
[Microsoft® SQL Server® 2012 Performance Dashboard Reports](#)
[Microsoft® SQL Server® 2012 Report Builder](#)

و یک جدول مقایسه‌ای بین امکانات نگارش‌های رایگان SQL Server 2012 [در اینجا](#)

نظرات خوانندگان

نوبسند: علی قمشلوی
تاریخ: ۲۲:۱۰:۱۷ ۱۳۹۰/۱۲/۱۹

سلام آقای نصیری خیلی داریم استفاده میکنیم لطفا بیشتر بنویسید با تشکر

نوبسند: وحید نصیری
تاریخ: ۲۲:۲۴:۲۷ ۱۳۹۰/۱۲/۱۹

ایمیل‌های خودکار خلاصه اشتراک‌های روزانه من هنوز برقرار است. جهت اطلاع: [\(^ \)](#)

از این آدرس Microsoft® SQL Server® 2012 Evaluation قابل دریافت است. همچنین اگر به سایت‌های وارز مراجعه کنید، به ازای هر نگارش SQL Server 2012، یک بسته دریافتی 4 گیگابایتی را به شما ارائه می‌دهند. یعنی اگر کسی بخواهد نسخه enterprise و نسخه developer را دریافت کند بیش از 8 گیگ را باید دریافت نماید! اما واقعیت این است که نیازی به دریافت هیچکدام نیست. یک فایل ISO مربوط به SQL Server 2012 بیشتر وجود خارجی ندارد. تمام این نگارش‌ها هم فقط براساس Product key است که مشخص می‌شوند. اگر سریال مرتبط با نگارش developer را وارد کنید، این نگارش نصب خواهد شد. اگر سریال نگارش enterprise را وارد کنید، نگارش سازمانی نصب خواهد شد؛ و تمام این‌ها هم فقط با همان یک فایل ISO اصلی ارائه شده توسط مایکروسافت میسر می‌شوند. این فایل ISO اصلی را از اینجا می‌توان دریافت کرد. بدیهی است Product key توکار و پیش فرض آن که در اختیار عموم است، مدت دار می‌باشد. بنابراین حین نصب تنها نیاز به سریال معتبر وجود دارد.

نظرات خوانندگان

نویسنده: Ahmad Moghadas khoo
تاریخ: ۱۷:۲۲:۳۵ ۱۳۹۰/۱۲/۲۲

کلا لنگه نداری.
وحید جان

نویسنده: حسین مرادی نیا
تاریخ: ۰۰:۰۱:۳۵ ۱۳۹۰/۱۲/۲۳

این نگارش نهایی ۲۰۱۲ هست یا باید منتظر نگارش نهایی باشیم؟!

نویسنده: وحید نصیری
تاریخ: ۰۰:۰۷:۴۴ ۱۳۹۰/۱۲/۲۳

[What is RTM](#)

نویسنده: مرتضی
تاریخ: ۱۰:۲۲:۰۹ ۱۳۹۰/۱۲/۲۵

درود بر شما آقای نصیری!

نویسنده: Mohsen
تاریخ: ۰۹:۱۸:۰۷ ۱۳۹۰/۱۲/۲۷

زنده باد مهندس نصیری و سال نو پیشاپیش مبارک!

نویسنده: وحید نصیری
تاریخ: ۲۰:۳۰:۳۱ ۱۳۹۰/۱۲/۲۷

سلامت باشید؛ سال نوی شما هم مبارک.

در [پست قبلی مهندس نصیری](#) با استفاده کتابخانه [ExpressMaint](#) و ویندوز یه سیستم بک آپ گیری را پیاده کرده اند، در این بین با توجه به وجود [SMO](#) میتوان از طریق خط فرمان ویندوز و کمک گیری از کتابخانه [7ZIP](#) و نوشتن فرامین مربوط و ایجاد بچ فایل و افزودن به Schedule ویندوز، نیز میتوان بصورت زیر نیز این پیاده سازی را انجام داد، که بنا بر نیاز اسکریپت بک آپگیری را نوشت:

```
echo off
cls
rem ----- Variables -----
set packName=DbName
set connection=-S Server -U User -P Password
set path=PathForSaveBackup
rem ----- Variables -----

echo create backup
sqlcmd %connection% -Q "BACKUP DATABASE %packName% TO DISK = '%path%\%packName%.bak'

echo compress backup file
..\tools\7za.exe a %path%\%packName%.7z %path%\%packName%.bak

echo delete backup file
del %path%\%packName%.bak
```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۴/۲۴ ۱۳:۷

در SQL Server 2008 اگر سرور رو تنظیم کنید، بکآپ‌های تولیدی آن به صورت خودکار فشرده شده تولید می‌شوند: ([^](#)) و نیازی به ابزار اضافی برای اینکار ندارند.

نویسنده: مهدی پایروند
تاریخ: ۱۳۹۱/۰۵/۰۲ ۹:۵۶

برای استفاده این امکان توکار و بی نیاز شدن از 7Zip باید اسکریپت را به این شکل تغییر دهیم:

```
sqlcmd %connection% -Q "BACKUP DATABASE %packName% TO DISK = '%path%\%packName%.bak', COMPRESSION"
```

عنوان: انتخاب Sub Query درون پرانتزها به کمک [Ctrl+Shift+]
نویسنده: فرهاد فرهمند خواه
تاریخ: ۱۳۹۱/۰۴/۲۶
آدرس: www.dotnettips.info
برچسبها: SQL Server

بوسیله [Ctrl+Shift+] شما می‌توانید از دورن Query‌های تودرتو، Sub Query را انتخاب نمایید، فقط کافیست، کرسر ماوس را ابتدا یا انتهای پرانتز قرار دهید.

امکان فوق از Sql Server 2005 به بعد گذاشته شده است.

منبع:

<http://blog.sqlauthority.com>

گاهی اوقات بد لیل تغییرات در جداول و غیره...، ممکن است لازم شود، از استفاده قرار گرفتن فیلدهای جداول درون Function، Viewها و یا Stored Procedureها، مطلع شوید. در این زمان شما می‌توانید از روش زیر استفاده نماید.

قبل از هر چیز در ابتدا Script زیر را اجرا نمایید. که شامل یک جدول به نام FindField، یک Stored Procedure به نام FindField AS، یک FUNCTION_FindField به نام Function و یک PROCEDURE_FindField به نام PROCEDURE_FindField می‌باشد.

```

Create Table FindField
(
ID int,
Firstname varchar(255),
Lastname varchar(255))
Go
CREATE PROCEDURE PROCEDURE_FindField
AS
    SELECT Firstname FROM FindField;
Go
CREATE FUNCTION FUNCTION_FindField (@id int)
RETURNS int
WITH EXECUTE AS CALLER
AS
begin
declare @id1 int;
    set @id1= (Select Max(Firstname) from dbo.FindField) ;
    RETURN (@id1);
END;
    
```

در ادامه Script زیر را اجرا نمایید:

```

SELECT OBJECT_NAME(object_id) as 'Procedure/Function/View Name',
definition
FROM sys.sql_modules
WHERE definition LIKE '%' + 'Firstname' + '%'
    
```

خروجی بصورت زیر خواهد بود:

	Procedure/Function/View Name	definition
1	PROCEDURE_FindField	CREATE PROCEDURE PROCEDURE_FindField AS ...
2	FUNCTION_FindField	CREATE FUNCTION FUNCTION_FindField (@id int) RE...

همانطور که در شکل مشاهده می‌نمایید، فیلد Firstname در Function و Stored Procedure مورد استفاده واقع شده است.

جهت کسب اطلاعات بیشتر به آدرس زیر مراجعه نمایید:

<http://msdn.microsoft.com/en-us/library/ms175081.aspx>

نظرات خوانندگان

نویسنده: **وحید نصیری**
تاریخ: **۱۳۹۱/۰۵/۰۴ ۱۲:۵۶**

ضمن تشکر، ابزار (افزونه) رایگانی هم برای انجام اینکار وجود دارد:

[SQL Search](#)

نویسنده: **فرهاد یزدان پناه**
تاریخ: **۱۳۹۱/۰۵/۰۴ ۱۹:۱۳**

با تشکر از مطلب جالبتون.

به نظر من یک کار دیگه، استفاده از WITH SCHEMABINDING در زمان ایجاد اشیای وابسته است.

نویسنده: **وحید نصیری**
تاریخ: **۱۳۹۱/۰۵/۰۴ ۱۳:۵۲**

این [روش خوبی](#) است ولی فقط برای viewها و توابع کاربرد دارد و نه برای SPها.

نویسنده: **فرهاد یزدان پناه**
تاریخ: **۱۳۹۱/۰۵/۰۴ ۱۴:۸**

بله همینطوره.

به نظر من ایجاد اشیایی همانند Stored Proc و Function با وجود ORM‌های امروزی، تا حدودی مناسب نیست (خارج از مواردی که مسائل Performance مطرح است).

ایجاد و بازسازی این اشیا در صورت وجود در هنگام Seed پایگاه داده (چه ایجاد و چه تغییر) می‌تواند بسیاری از مشکلات موجود را حل کند.

نویسنده: **مرتضی**
تاریخ: **۱۳۹۱/۰۵/۰۴ ۱۹:۵۶**

سلام

با این کوئری که شما گفتید اگر View, Function , Stored Procedure رمز شده باشد
دیگه نمی‌تونه جواب بد
ولی با این کوئری امکانش هست

```
SELECT TABLE_NAME,
       column_Name
  FROM INFORMATION_SCHEMA.COLUMNS
 WHERE column_Name LIKE '%' + 'Name' + '%'
```

نویسنده: **مرتضی**
تاریخ: **۱۳۹۱/۰۵/۰۴ ۲۰:۲**

اون افزونه رایگانی که مهندس نصیری معرفی کردن از هر دو استفاده می‌کنه یعنی هم متن درون view و Procedure , Function رو جستجو می‌کنه که اگر رمز شده باشد دیگه نتیجه ای نداره و هم عناوین ستون‌های اون شئ رو جستجو می‌کنه

نویسنده: فرهاد فرهمندخواه
تاریخ: ۲۰:۲۴ ۱۳۹۱/۰۵/۰۴

مرسى از افزونه اى که معرفى نموديد،

توابعی که در این بررسی عنوان می‌شود در زمان انتشار نسخه SQL Server 2005 ارائه شده است.

قبل از بررسی توابع، زیر را اجرا می‌نماییم، که شامل جدولی به نام Test و درج چند رکورد درون آن می‌باشد:

```

CREATE TABLE Test (ID INT, Product VARCHAR(100), Price INT, Color VARCHAR(100))
GO
INSERT INTO Test
SELECT 1, 'Toy', 100, 'Black'
UNION ALL
SELECT 2, 'Pen', 100, 'Black'
UNION ALL
SELECT 3, 'Pencil', 100, 'Blue'
UNION ALL
SELECT 4, 'Pencil', 100, 'Red'
UNION ALL
SELECT 5, 'Pencil', 200, 'Yellow'
UNION ALL
SELECT 6, 'Cup', 300, 'Orange'
UNION ALL
SELECT 7, 'Cup', 400, 'Brown'
GO

```

اولین تابعی را که بررسی می‌نماییم، Row_Number می‌باشد و Syntax آن بصورت زیر است:

```
ROW_NUMBER () OVER ([<partition_by_clause>] <order_by_clause>)
```

بوسیله تابع Row_Number می‌توان اعداد توالی (ترتیبی) را به رکوردهای یک جدول نسبت داد. برای روشن‌تر شدن مطلب فوق مثالی را بررسی می‌نماییم.
در ابتدا Query زیر را اجرا نمایید:

```
Select *, ROW_NUMBER() OVER ( ORDER BY Price DESC) AS RN from Test
```

خروجی آن بصورت زیر می‌باشد:

	ID	Product	Price	Color	RN
1	7	Cup	400	Brown	1
2	6	Cup	300	Orange	2
3	5	Pencil	200	Yellow	3
4	1	Toy	100	Black	4
5	2	Pen	100	Black	5
6	3	Pencil	100	Blue	6
7	4	Pencil	100	Red	7

همانطور که در شکل مشاهده می‌نمایید، یک عدد ترتیبی تولید شده و به هریک از رکوردهای جدول نسبت داده شده است، در مطلب بعدی یک مثال کاربردی از Row_Number خواهیم زد.

لازم به یادآوری است که استفاده از Order by در Syntax Row_Number الزامی می‌باشد.
اگر به تابع Row_Number Syntax توجه نماییم، با کلمه Partition مواجه می‌شویم، که جهت گروه بندی استفاده می‌شود، به عبارت دیگر ممکن است شما بخواهید، ابتدا جدول خود را براساس فیلد یا فیلدهایی Group by نمایید و سپس روی آنها Row_Number را اعمال کنید، که در این حالت از Partition استفاده می‌شود.
برای درک بیشتر Query زیر را اجرا نمایید:

```
Select *,ROW_NUMBER() OVER (PARTITION BY Product ORDER BY Price DESC) AS RN from Test
```

خروجی بصورت زیر خواهد بود:

	ID	Product	Price	Color	RN
1	7	Cup	400	Brown	1
2	6	Cup	300	Orange	2
3	2	Pen	100	Black	1
4	5	Pencil	200	Yellow	1
5	3	Pencil	100	Blue	2
6	4	Pencil	100	Red	3
7	1	Toy	100	Black	1

همانطور که در شکل مشاهده می‌نمایید، در ابتدا، جدول براساس فیلد Product، دسته بندی (Group by) شده است و سپس اعداد ترتیبی روی هر by Group بصورت جداگانه اعمال شده است.

تابع RANK()

از تابع فوق در جهت رتبه بندی نمودن فیلدهای یک جدول استفاده می‌شود و Syntax آن بصورت زیر می‌باشد:

```
RANK () OVER ([<partition_by_clause>] <order_by_clause>)
```

برای درک مطلب فوق نیز مثالی می‌زنیم:
ابتدا Query زیر را اجرا نماییم:

```
Select *,RANK() over (ORDER BY Price ) AS RANK from Test
```

خروجی بصورت زیر خواهد بود:

	ID	Product	Price	Color	RANK
1	1	Toy	100	Black	1
2	2	Pen	100	Black	1
3	3	Pencil	100	Blue	1
4	4	Pencil	100	Red	1
5	5	Pencil	200	Yellow	5
6	6	Cup	300	Orange	6
7	7	Cup	400	Brown	7

یادآوری: زمانی که دورن by Order ترتیب صعودی یا نزولی بودن را تعیین نکنیم، by تصویرت پیش فرض صعودی میباشد.

همانطور که در شکل مشاهده می‌نمایید، رتبه بندی انجام شده به ترتیب نمی‌باشد، و برای مقادیر تکراری فیلد Price از Rank یکسانی استفاده شده است. نکته دیگر این که بین اعداد مشاهده شده در فیلد Rank نیز gap ایجاد می‌شود. به عبارت دیگر عمدۀ تفاوت تابع Rank با تابع Row_Number همین مواردی است که بیان شده است.

در Syntax T-SQL تابع Rank نیز کلمه Partition هم وجود دارد، که در جهت Group by فیلد یا فیلد‌های خاصی استفاده می‌شود، و رتبه بندی نیز در این حالت روی Group by انجام می‌گردد.

برای درک بهتر Query زیر را اجرا نمایی:

```
Select *,RANK() over (Partition by Product ORDER BY Price Desc) AS RANK from Test
```

خروجی بصورت زیر خواهد بود:

	ID	Product	Price	Color	RANK
1	7	Cup	400	Brown	1
2	6	Cup	300	Orange	2
3	2	Pen	100	Black	1
4	5	Pencil	200	Yellow	1
5	3	Pencil	100	Blue	2
6	4	Pencil	100	Red	2
7	1	Toy	100	Black	1

همانطور که در شکل مشاهده می‌نمایید، رتبه بندی روی هر Group تصویرت جداگانه اعمال شده است.

تابع Dense_Rank

این تابع نیز همانند تابع Rank عمل می‌کند، با این تفاوت که هیچ gap‌ی بین اعداد آن رخ نمی‌دهد.

با جرای Query زیر خواهیم داشت:

```
Select *,dense_RANK() over (ORDER BY Price ) AS dense_RANK from Test
```

خروجی بصورت زیر خواهد بود:

	ID	Product	Price	Color	dense RANK
1	1	Toy	100	Black	1
2	2	Pen	100	Black	1
3	3	Pencil	100	Blue	1
4	4	Pencil	100	Red	1
5	5	Pencil	200	Yellow	2
6	6	Cup	300	Orange	3
7	7	Cup	400	Brown	4

همانطور که ملاحظه می‌نمایید هیچ gap‌ی بین اعداد Rank ایجاد نشده است.

و برای استفاده از Partition, Dense_Rank در تابع های دیگر میباشد.

تابع NTILE

این تابع نیز مانند توابع بالا در جهت رتبه بندی استفاده می‌شود، و بوسیله تابع فوق شما می‌توانید رکوردهای جدول خود را به تعداد گروههای دلخواه تقسیم نمایید. و آن بصورت زیر میباشد:

```
NTILE (integer_expression) OVER ([<partition_by_clause>] <order_by_clause>)
```

برای درک مطلب فوق مثالی می‌زنیم:

```
Select * ,NTILE(4) over ( ORDER BY Price desc) from Test
```

خروجی بصورت زیر خواهد بود:

	ID	Product	Price	Color	NTILE
1	7	Cup	400	Brown	1
2	6	Cup	300	Orange	1
3	5	Pencil	200	Yellow	2
4	1	Toy	100	Black	2
5	2	Pen	100	Black	3
6	3	Pencil	100	Blue	3
7	4	Pencil	100	Red	4

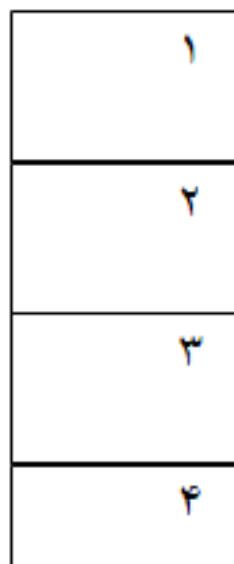
در Syntax تابع فوق اشاره به Integer_Expression شده است. که یک مقدار عددی دریافت می‌کند و بیانگر تعداد گروه بندی دلخواه می‌باشد.

حال سئوال اینجاست که رتبه بندی جدول به چه صورت انجام شده است:

همانطور که مشاهده می‌نمایید، جدول فوق شامل 7 رکورد می‌باشد، و ما در مثال خود، تمایل داشتیم که رکوردهای جدول به چهار گروه تقسیم و سپس رتبه بندی شوند، بنابراین 7 تقسیم بر 4 شده است و باقی مانده آن می‌شود 3

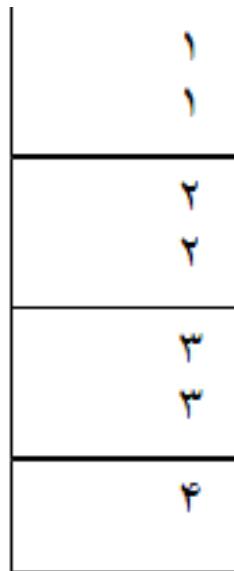
$$4*1+3=7$$

در ابتدا چهار گروه ایجاد می‌شود و در هر خانه یک رکورد قرار می‌گیرد



سپس 3 رکورد باقی می‌ماند که از اولین گروه رو به پایین، برای هر گروه فقط یک رکورد درج می‌شود، یعنی یک رکورد به گروه

یک، یک رکورد به گروه 2 و هم چنین یک رکورد به گروه 3 بنابراین خواهیم داشت:



نکته مهم: اگر تعداد رکورد باقی مانده بعد از تقسیم بیش از یک عدد باشد، در زمان اختصاص دادن به گروه ها، به هر گروه از بالا به پایین فقط یک رکورد اختصاص داده می شود.

مثالی دیگر:

```
Select *,NTILE(3) over ( ORDER BY Price desc) AS NTILE from Test
```

خروجی:

	ID	Product	Price	Color	NTILE
1	7	Cup	400	Brown	1
2	6	Cup	300	Orange	1
3	5	Pencil	200	Yellow	1
4	1	Toy	100	Black	2
5	2	Pen	100	Black	2
6	3	Pencil	100	Blue	3
7	4	Pencil	100	Red	3

در این حالت $3*2+1=7$

امیدوارم مطلب فوق مفید واقع شده باشد.

نظرات خوانندگان

نوبنده: حمیدرضا
تاریخ: ۷:۵۹ ۱۳۹۱/۰۵/۰۹

مطلوب بسیار مفیدی بود. تشکر می‌کنم.

در بعضی مواقع ممکن است که در حین کار و با تراکنش‌های باز، دیتابیس SQL Server دچار مشکل شود و از دسترس خارج شود و این فرض را هم در نظر بگیرید که بک آپ دیتابیس مربوط به زمانی بیش از حد انتظار است. آیا ممکن است که دیتابیس را با وجود از دست دادن فایل لاغ آن بازگردانی کرد؟
جواب: بله ولی بدون عواقب نیست.

بطور معمول در زمانی که تراکنش‌های باز بر روی سرور دیتابیس وجود دارد و بانک کرش میکند، کرش ریکاوری، تراکنش‌های باز را رول بک میکند. این امر مانع از اثرات پراکنده از تراکنش‌های فعلی در پایگاه داده می‌باشد.
اگر لاغ در دسترس نباشد هنگامی که کار سرور SQL شروع می‌شود، پایگاه داده در حالت SUSPECT قرار داده می‌شود. در این موقع تها راه آن لاین نمودن بانک (البته منظور آماده بکار نمودن بانک نیست) استفاده از حالت قابلیت تعمیر اورژانسی است که از نسخه 2005 افزوده شده است که با ساخت یک فایل لاغ جدید و سپس اجرا نمودن DBCC CHECKDB با استفاده از REPAIR_ALLOW_DATA_LOSS

مشکل اینجاست که اگر شما به همین طریق بخواهید ادامه بدھید و از این قابلیت اورژانسی استفاده کنید، مسلماً امکان وجود دسته‌ای از تغییرات در بانک که ممکن است در میانه اعمال بروز رسانی چند رکورد فروشن در یک جدول نیمی از تراکنش‌های آن‌ها روی بانک اعمال شده و بعد از راه اندازی دوباره با لاغ جدید، امکان برگرداندن آن‌ها وجود داشته باشد، مواجه شوید. به این معنی که در بهترین حالت، پایگاه داده بدست آمده با برنامه هماهنگ نیست و یا اینکه به بک آپی که وجود دارد رضایت بدھید.
حال تعمیر اورژانسی، زمانیکه همه راه حل‌های بازگردانی بانک مغلوب شوند آخرین روش مانده است. این حالت، حالت بین دو شر، ریکاور کردن بانک به حالتی نا هماهنگ با نرم افزار و یا برگرداندن به زمانی خیلی عقب‌تر است که در نهایت اقدام به انجام هر دو و هماهنگ‌تر کردن بانک برای کار با برنامه می‌شوید که بسیار وقت گیر و مشکل ساز است.
ولی باز ممکن است این وضعیت پیش بیاد چون شما از اتفاقی که در زمان کرش در بانک افتاده اطلاعی ندارید. برای جلوگیری از این موارد در آینده سازوکار بک آپ گیری از بانک را باید تغییر بدھید و دفعات بک آپ گیری را افزایش بدھید و نیز از روش‌های جدید که قابلیت دسترسی بالا دارند استفاده کنید همانند mirroring و Availability Groups. SQL Server 2012

منبع: [^](#)

لينکهای مرتبه:

1. [^ ?Is the recovery interval guaranteed.](#)
2. [^ Scaling and Sharding Deep Freeze.](#)
3. [^ Why Database Operations Hit the Wall.](#)
4. [^ Avoiding logging for user operations.](#)

5. [^ Resolving the Database Performance Problem Blame Game.](#)

نظرات خوانندگان

نوبسنده: ali تاریخ: ۱۳۹۱/۰۵/۱۰ ۱۱:۹

لطفا کاملتر توضیح بدید.
من دیتابیسی دارم که فایل لاغ آن پاک شده باید چیکار کنم؟

نوبسنده: مهدی پایروند تاریخ: ۱۳۹۱/۰۵/۱۰ ۱۲:۲۱

سوالتون کلی هست لطفا توضیح بیشتری بدید و البته باید توجه داشته باشد که با توجه به اطلاعات بالا اگه میخواهید از تعمیر اورژانسی استفاده کنید نسخه بانک باید از ۲۰۰۵ به بالا باشد

نوبسنده: ali تاریخ: ۱۳۹۱/۰۵/۱۲ ۰:۷

سوال این هست که من یک دیتابیس دارم که فایل LDF آن (فایل log) پاک شده حالا من بدون فایل LDF چطور این دیتابیس را کنم. فایل MDF را واسه Attach انتخاب میکنم Error میده! چون فایل LDF را پیدا نمیکنه.

نوبسنده: وحید نصیری تاریخ: ۱۳۹۱/۰۵/۱۲ ۰:۱۲

سه راه حل اینجا ذکر شده. آزمایش کنید کدام یکی برای وضعیت شما مفید است.

نوبسنده: ali تاریخ: ۱۳۹۱/۰۵/۱۲ ۰:۲۰

آقای نصیری ممنون با همون روش اول حل شد.

```
USE [master]
GO
-- Method 1: I use this method
EXEC sp_attach_single_file_db @dbname='TestDb',
@physname=N'C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA\TestDb.mdf'
GO
```

نوبسنده: پژمان پارسائی تاریخ: ۱۳۹۱/۱۱/۱۱ ۲۲:۵۳

سلام
دیتابیسی هست که به حالت InRecovery رفته. در SQL Server Management Studio در لیست دیتابیس‌ها وجود داره ولی جلوی اسم اون عبارت InRecovery رو نوشته. در این حالت که هست نمیشه که ساختار این دیتابیس اعم از جدول‌ها، داده‌های داخل اون جداول، stored procedureها و ... رو بینیم. از اطلاعات اون هم پشتیبان گیری نشده و به داده‌های داخل اون نیاز هست. شما می‌دونید مشکل چیه و چطور حل میشه ؟
با تشکر

نوبسنده: سعید تاریخ: ۱۳۹۱/۱۱/۱۱ ۲۳:۱

چیز مهمی نیست. احتمالا فضای کم اومده روی هارد یا لاغ پر شده. به علاوه فقط باید بهش زمان بدی تا کارش تموم شه.

نوبسند: پژمان پارسائی
تاریخ: ۱۴:۲ ۱۳۹۱/۱۱/۱۲

این اسکریپت رو پیدا کردم، منتهی اجرаш که میکنم زمان باقی مونده رو اعلام نمیکنه.

```
DECLARE @DBName VARCHAR(64) = 'databasename'

DECLARE @ErrorLog AS TABLE([LogDate] CHAR(24), [ProcessInfo] VARCHAR(64), [TEXT] VARCHAR(MAX))

INSERT INTO @ErrorLog
EXEC sys.xp_readerrorlog 0, 1, 'Recovery of database', @DBName

SELECT TOP 5
[LogDate]
,substring([TEXT], charindex(') is ', [TEXT]) + 4,charindex(' complete (', [TEXT]) - charindex(') is ', [TEXT]) - 4) AS PercentComplete
,cast(substring([TEXT], charindex('approximately', [TEXT]) + 13,charindex(' seconds remain', [TEXT]) - charindex('approximately', [TEXT]) - 13) AS FLOAT)/60.0 AS MinutesRemaining
,cast(substring([TEXT], charindex('approximately', [TEXT]) + 13,charindex(' seconds remain', [TEXT]) - charindex('approximately', [TEXT]) - 13) AS FLOAT)/60.0/60.0 AS HoursRemaining
,[TEXT]

FROM @ErrorLog ORDER BY [LogDate] DESC
```

در ضمن دیتابیس روی درایو D هست و فضای چند ده گیگ خالی دارد. حجم فایل 1df هم تا حدود 400 مگ هست. وقتی هم یک کوئری بخوام روی دیتابیس اجرا کنم خطای زیر رو میده:

```
Database 'Manager' is being recovered. Waiting until recovery is finished.
```

عنوان: LocalDB چیست؟

نویسنده: حسین مرادی نیا

تاریخ: ۱۳۹۱/۰۶/۱۶ ۱:۲۴

آدرس: www.dotnettips.info

گروه‌ها: SQL Server, LocalDB, SQL Server 2012

LocalDB نسخه‌ای جدید از Sql server express است که به توسعه دهندگان این اجازه را می‌دهد تا با نصب آن، از نصب کامل دیگر نسخه‌های Sql server جلوگیری نمایند. LocalDB برای برنامه‌هایی که به صورت Local و بر روی یک سیستم اجرا می‌شوند مورد استفاده قرار می‌گیرد.

مزایای استفاده از این نسخه

فایل نصب با حجم بسیار کم. (28.2MB برای نسخه 32 بیتی و 33.7MB برای نسخه 64 بیتی)

سادگی (بدون نیاز به انجام تنظیمات خاص بر روی سیستم)

اجرا در محیط‌هایی که کاربر جاری دسترسی مدیریتی ندارد.(برای اجرای آن نیاز به Permission های مدیریتی نیست و یک کاربر سطح پایین هم می‌تواند آن را اجرا کند)

سادگی نصب

همانند Sql server Express سازگاری کاملی با T-Sql دارد. همچنین از Stored Procedure ها ، داده‌های جغرافیایی و مکانی (geometry and geography) ، View ها و Triggers پشتیبانی می‌کند. سازگاری با Sql server Provider معمولی

عدم اجرای سرویس خاصی در حافظه برای مدیریت دیتابیس. پروسس‌های LocalDb هر زمان که نیاز باشد اجرا می‌شوند و هر زمان که به آنها نیاز نداشته باشیم به صورت اتوماتیک متوقف می‌شوند.

پشتیبانی از خصوصیت AttachDbFileName در کانکشن استرینگ جهت استفاده از فایل بانک اطلاعات به صورت مستقیم

سرویس پک‌های جدید LocalDB به راحتی بر روی نسخه موجود نصب می‌شوند و نسخه قبلی را به روز رسانی می‌کنند.

نصب یک LocalDB برای همه کاربران یک کامپیوتر

پشتیبانی کامل از Silent Installation

امکان استفاده از آن توسط Asp.net

پشتیبانی از XML (XQuery) و BLOB

پشتیبانی از Ado.net sync framework

پشتیبانی از LINQ

پشتیبانی از Distributed transactions

کانکشن‌های نامحدود (البته به صورت Local)

نیازمندی‌های نصب

نیاز به نصب Sql server 2012 native client . این مورد به همراه LocalDB روی سیستم نصب نمیشود

نیاز به دسترسی مدیریتی جهت نصب

فضای خالی دیسک سخت 140MB

به روز رسانی دات نت فریم ورک 4 به [4.0.2](#) و یا نسخه‌های بالاتر

محدودیت‌ها

عدم پشتیبانی از Windows 2003 و Windows xp , Window server

عدم امکان نصب نسخه 32 بیتی بر روی ویندوز 64 بیتی (حتما باید نسخه 64 بیتی آن را نصب کنید)

فقط می‌توان به صورت Local از آن استفاده کرد. امکان استفاده تحت شبکه وجود ندارد و فقط به کانکشن‌های Local پاسخ می‌دهد.

فقط توسط Management studio SQL Server 2012 management studio را نمی‌توان از طریق LocalDB در دسترس می‌باشد.

قدیمی مدیریت کرد.

عدم پشتیبانی از Visual Studio 2010 از LocalDB

عدم اجرا بر روی موبایل‌های هوشمند

محدودیت سایز بانک اطلاعات : 10GB

عدم پشتیبانی از قابلیت FileStream

محدودیت استفاده از فقط یک CPU

عدم امکان Debuging در هنگام اتصال به LocalDB

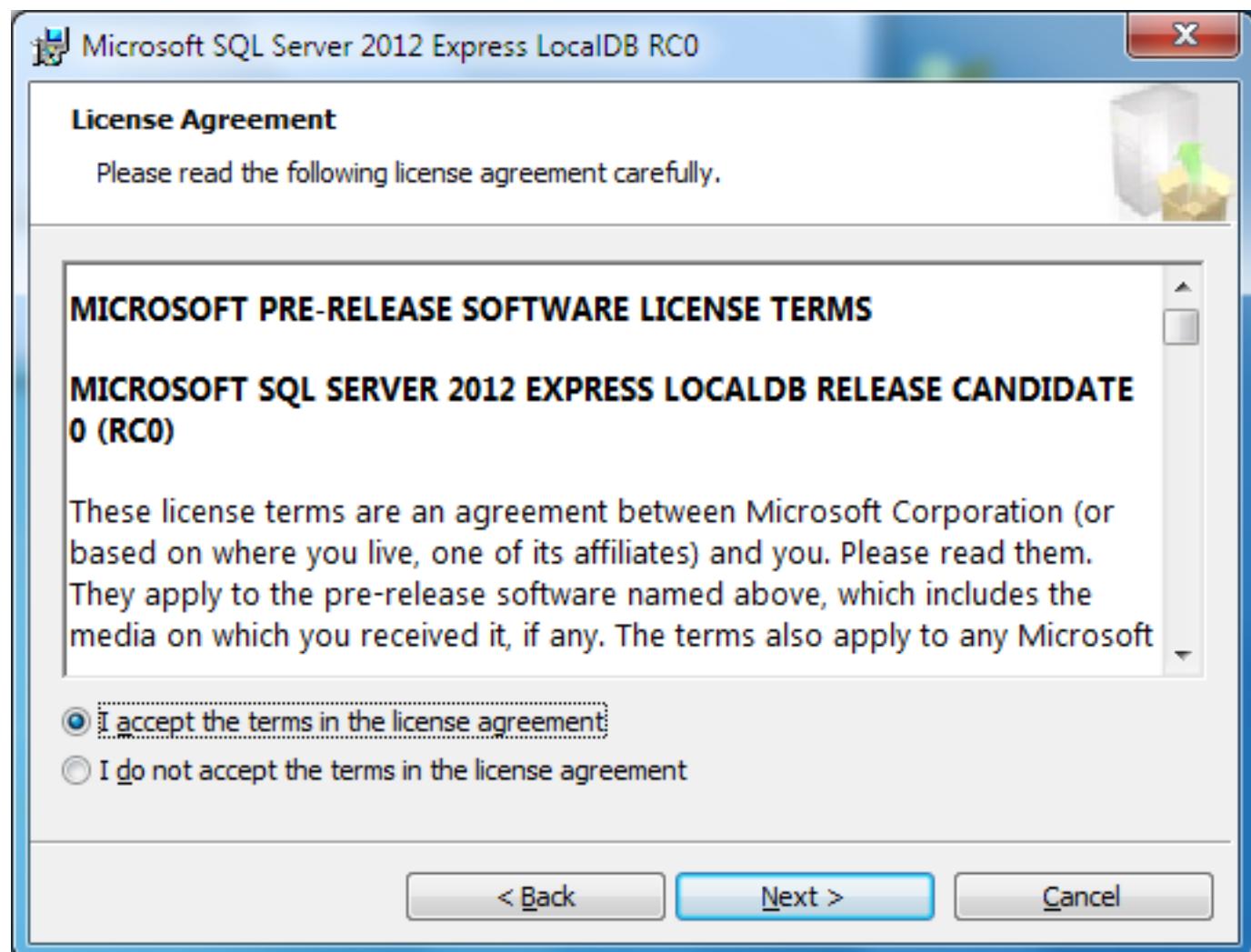
نحوه نصب

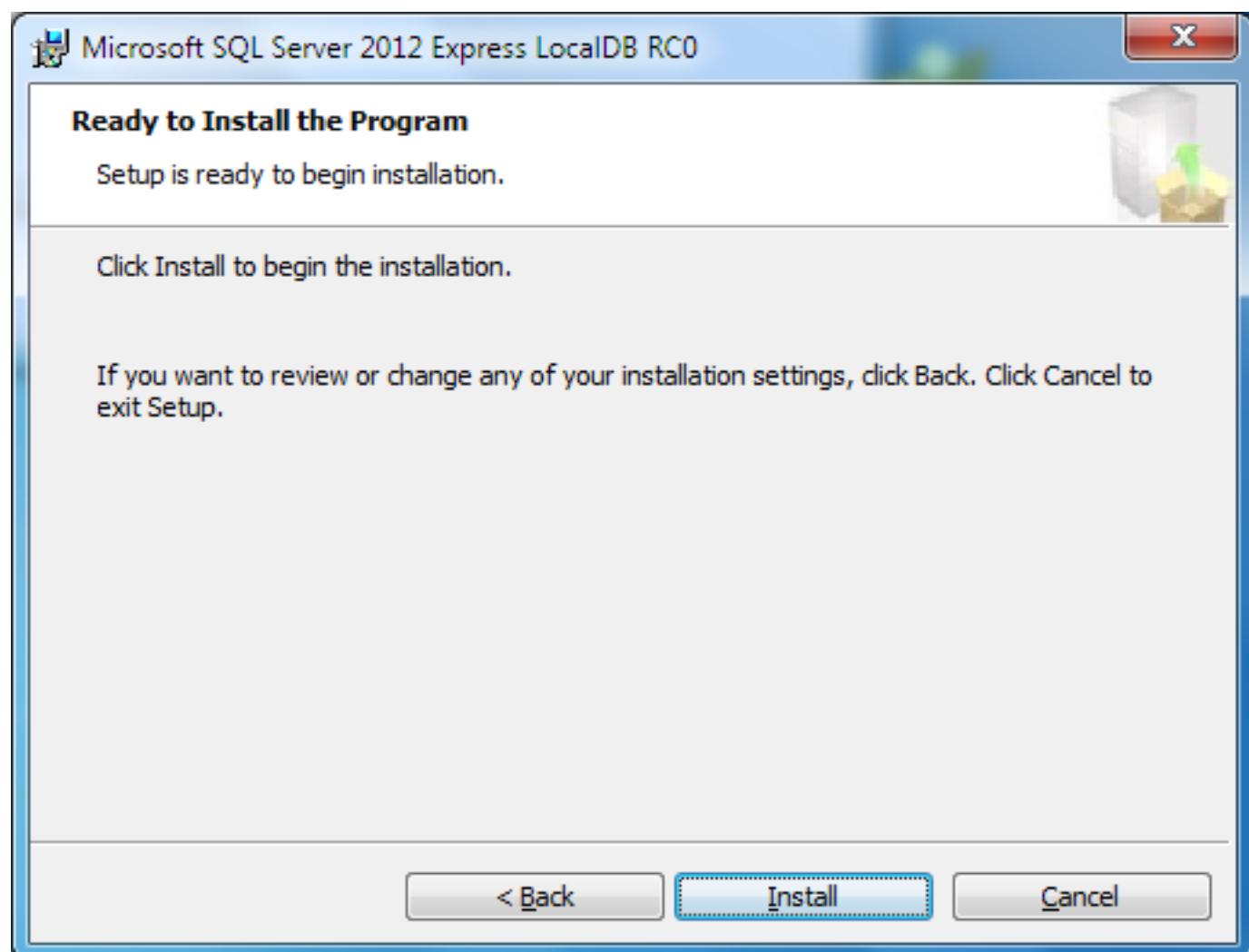
ابتدا SQL Server LocalDB را دانلود نمایید. سپس برای نصب آن بر روی سیستم فقط کافی است که فایل نصب برنامه را اجرا نموده و مربوطه را قبول نمایید. همچنین در صورت نیاز به Silent Installation License کافی است که از دستور زیر در خط فرمان استفاده نمایید:

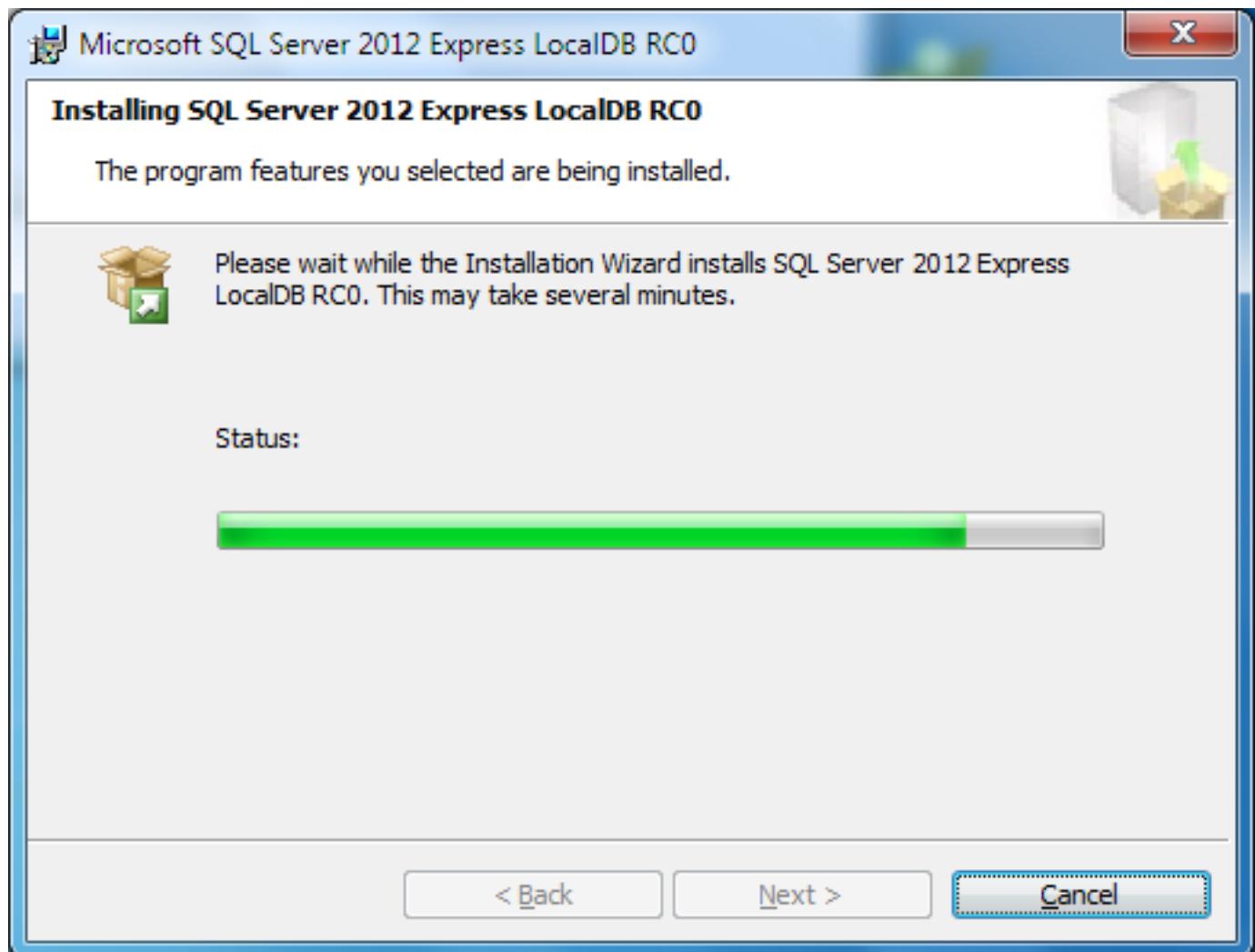
```
msiexec /i SqlLocalDB.msi /qn IACCEPTSQLLOCALDBLICENSETERMS=YES
```

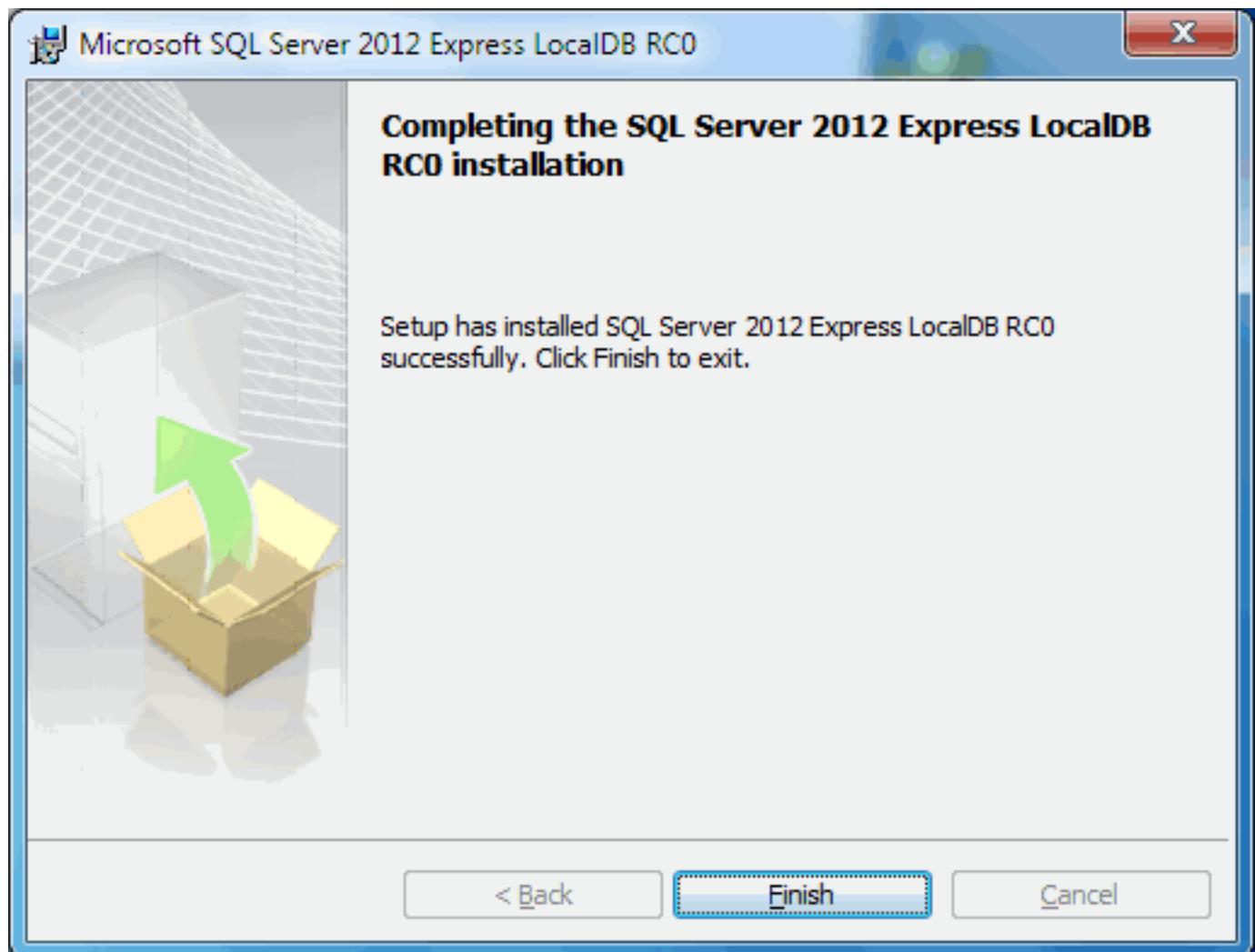
همچنین می‌توانید مراحل نصب را توسط فایل نصب انجام دهید:





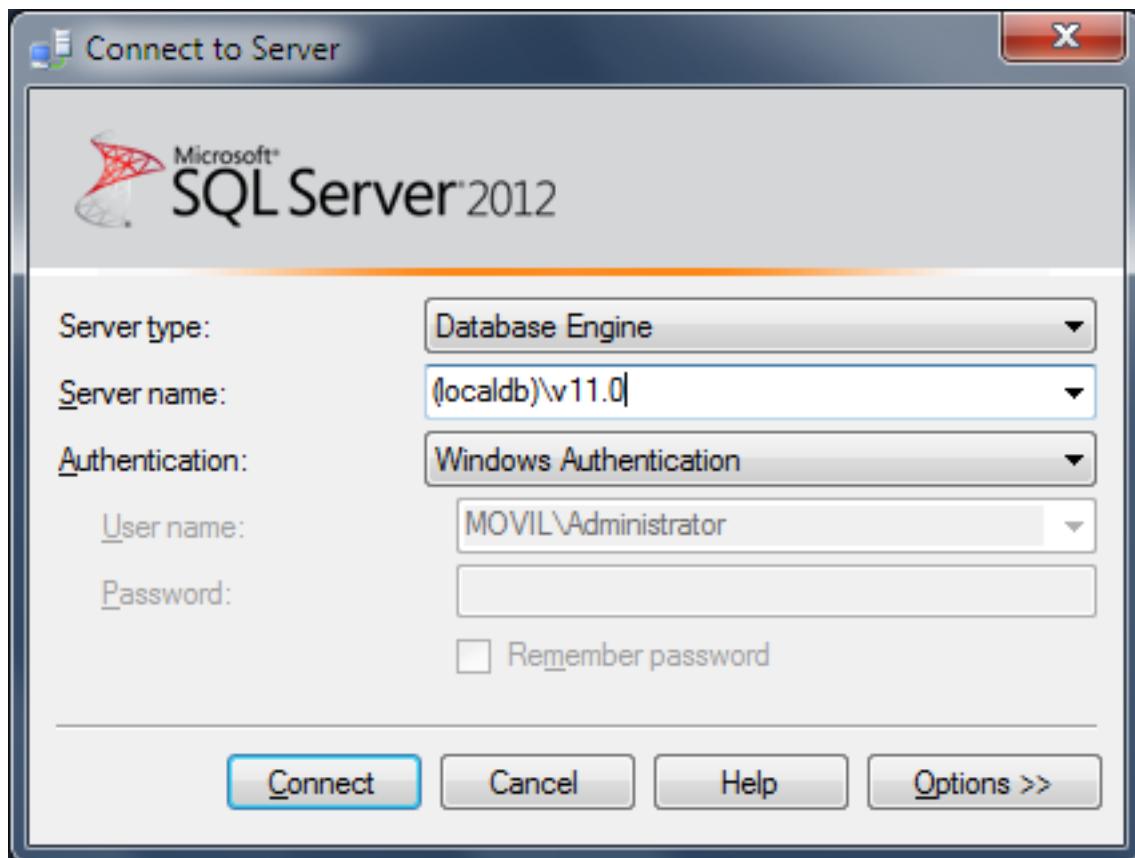






نحوه اتصال به LocalDB توسط Sql Server Management Studio

اگر .net framework را از نسخه 4.0.2 و پایه‌های روز رسانی کرد و باشید می‌توان توسط Sql Server Management Studio به عنوان نام سرور LocalDB 2012 اضافه کرد.



مجددا لازم به ذکر است که امکان اتصال توسط Management Studio های قبلی به بانک LocalDB امکان پذیر نمی باشد.

[برای مطالعه بیشتر](#)

نظرات خوانندگان

نویسنده: ramin_rp
تاریخ: ۹:۳۷ ۱۳۹۱/۰۶/۱۶

سلام
تا جایی که من اطلاع دارم نسخه rtm از 2010 vs ساپورت نمیشه

نویسنده: وحید نصیری
تاریخ: ۱۲:۷ ۱۳۹۱/۰۶/۱۶

هیچ بانک اطلاعاتی خاصی الزاما نیازی به VS.NET ندارد. کانکشن استرینگ را تنظیم می کنید و بعد می تونید [با کد نویسی](#) حتی جدول، فیلد و تمام تنظیمات مورد نظر را اضافه کنید.
ضمن اینکه خیلی از توانایی های آن با VS.NET 2010 هم کار می کند ([^](#)).

نویسنده: علی یزدی
تاریخ: ۲:۲۲ ۱۳۹۱/۰۶/۱۷

واقعاً که عالی بود. ممنون

نویسنده: کیان
تاریخ: ۱۵:۵۹ ۱۳۹۱/۰۶/۲۸

ایا LocalDB جایگزینی برای SQL Server Express خواهد بود در ادامه؟ ([اینجا](#) انتهای پاراگراف اول منظورش چیه دقیقاً؟) (*Express LocalDB should be used in place of the SQL Server Express user instance feature which is deprecated*)

هیچ شکلی نمی توان تحت شبکه کار کرد با این نسخه؟ مثلاً جایی که قراره 3 تا سیستم بصورت کلاینت/서ور کار کنن نمی شه فایل دیتابیس روی یک سیستم باشه و از رو 2 تای دیگه بهش کانکت شد؟ سنارویی که خیلی وقتاً ممکنه استفاده بشه بهترین راه برای سناریوهایی مثل مورد 2 که مثال زدم چیه؟ (یک محیط و یک برنامه معمولی که قرار نیست sql server بصورت full نصب بشه رو سرور) آیا می شه با خیال راحت از **SqlServer Express Edition** استفاده کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۶:۱۸ ۱۳۹۱/۰۶/۲۸

- خیر. نام کامل این محصول «[Microsoft® SQL Server® 2012 Express](#)» است + نسخه «[SQL Server Express LocalDB](#)» از [اینجا](#) قابل دریافت است. بنابراین جایگزین یا حذف نشده.
- هدف اصلی از LocalDb ارائه یک «[embedded database](#)» جدید از طرف مایکروسافت است.
- بنابراین [هدف آن](#) استفاده تحت شبکه نیست. جاهایی استفاده می شود که تک کاربر نهایی دانش آنچنانی در نصب و نگهداری بانک های اطلاعاتی ندارد و برنامه و سیستم بانک اطلاعاتی او یکپارچه به نظر می رسد. از این نمونه بانک های اطلاعاتی embedded باز هم هستند. مانند SQLite, SQL CE, Firebird Embedded و غیره.
- نسخه Express تحت شبکه قابل استفاده است؛ البته [نیاز به تنظیم](#) دارد.

نویسنده: حسین
تاریخ: ۲۱:۱۰ ۱۳۹۱/۰۸/۰۲

من با Management Studio ی SQL Server 2008 همین (localdb)\v11.0 را وارد کردم و متصل شد. مطمئن هستید که فقط با 2012 میشه بهش وصل شد؟

نویسنده: رضا بزرگی
تاریخ: ۲۰:۳۵ ۱۳۹۱/۰۸/۰۶

- میشه لطفا بگین منظور شما از تحت شبکه دقیقا چیه؟
- برنامه MVC4 که با localdb هست را چگونه میشود روی هاستهای اشتراکی استفاده کرد؟ آیا نیاز به تنظیم خاص در کانکشن استرینگ دارد؟

نویسنده: وحید نصیری
تاریخ: ۲۰:۴۹ ۱۳۹۱/۰۸/۰۶

- یعنی حالت سرور داشته باشد و بتوان به آن مستقیما از راه دور متصل شد. مثل SQL Server کامل.
 - هاست اشتراکی باید این بانک اطلاعاتی را پیشتر نصب کرده باشد.
- مانند SQL Server CE نیست که با توزیع چند DLL مشکل آن برطرف شود و نیازی به نصب نداشته باشد.

نویسنده: میثم
تاریخ: ۱۰:۲۴ ۱۳۹۲/۰۱/۱۱

سلام. دیتابیس برنامه من LocalDB هستش و از روش EF Code First هم برای برقراری ارتباط با دیتابیس استفاده میکنم. میخواهم فایل‌های مرتبط با دیتابیسم .mdf و .ldf. توی فolder Bin باشه. ولی وقتی از کانکشن استرینگ زیر استفاده میکنم دیتابیس رو توی C:\Users... قرار میده. در حالی که با SQL Compact چنین مشکلی نداشتم و فایل .sdf. رو در فolder Bin قرار میدار.

```
<entityFramework>
  <defaultConnectionFactory type="System.Data.Entity.Infrastructure.SqlConnectionFactory,
EntityFramework">
    <parameters>
      <parameter value="Data Source=(localdb)\v11.0; Integrated Security=True;
MultipleActiveResultSets=True;" />
    </parameters>
  </defaultConnectionFactory>
</entityFramework>
```

```
<connectionStrings>
  <clear/>
  <add name="MyContext"
    connectionString="Data Source=(localdb)\v11.0; Integrated Security=True;
MultipleActiveResultSets=True; AttachDBFilename=MyDatabase.mdf"
    providerName="System.Data.SqlClient"
  />
</connectionStrings>
```

ممnon میشم راهنماییم کنید.

نویسنده: وحید نصیری
تاریخ: ۱۰:۳۸ ۱۳۹۲/۰۱/۱۱

این یک روش عمومی است و در تمام رشته‌های اتصالی دات نتی کار می‌کند:
از DataDirectory استفاده کنید. مثلا:

```
AttachDBFilename=|DataDirectory|\database.mdf
```

مقدار آن در برنامه‌های ASP.NET به صورت خودکار به پوشش استاندارد App_Data مپ می‌شود. برای سایر حالات می‌توانید اون رو در زمان آغاز برنامه دستی مقدار دهی کنید:

```
AppDomain.CurrentDomain.SetData("DataDirectory", "C:\myDB");
```

ضمناً روش مسیردهی کامل هم همیشه کار می‌کند

```
AttachDbFilename='Full\Path\To.MDF'
```

نویسنده: یزدان
تاریخ: ۱۲:۱۳ ۱۳۹۲/۰ ۱/۱۱

سلام: AppDomain چیه ؟
اینکه میگن توی یک برنامه ممکنه چند AppDomain داشته باشیم چه معنی میده ؟

نویسنده: وحید نصیری
تاریخ: ۱۲:۲۷ ۱۳۹۲/۰ ۱/۱۱

روشی است برای ایزوله سازی اطلاعات برنامه‌ها از یکدیگر. به این ترتیب در یک پروسه هم می‌شود چندین برنامه دیگر را به صورت کاملاً ایزوله از هم بارگذاری و اجرا کرد.

نویسنده: رحیم
تاریخ: ۰:۳۰ ۱۳۹۲/۰ ۱/۱۵

اگر از ویندوز XP رو هم پشتیبانی می‌کرد خیلی خوب می‌شد، بدون این گزینه نمی‌شده رفت سراغش مخصوصاً برای برنامه‌های شرکتی که باید روی اکثر سیستم عامل‌ها اجرا بشه (و فعلاً xp استفاده زیادی داره)

نویسنده: حسین
تاریخ: ۱:۰ ۱۳۹۲/۱۱/۱۸

با تشکر

من 2 تا مشکلی که در هنگام استفاده از local db در ویژوال استدیو 2012 برخورد کردم این بود که 1-هر از گاهی پیغام کم آوردن خطای حافظه رو میداد (مثلاً هنگامی که میخواستم محتویات جداول رو ببینم) و تقریباً توی اکثر اوقات چاره‌ای به جز بستن ویژوال استدیو و باز کردن مجدد آن نداشتم 2-اسکریپت تهیه کردن از جدول و اطلاعات داخل آن واقعاً عذاب آوره و هر جدول رو باید جداگانه اسکریپت تبدیل میکردم خواستم ببینم راهی برای این مشکل وجود دارد؟

نویسنده: محسن درپرسنی
تاریخ: ۲۰:۳۷ ۱۳۹۳/۰ ۱/۰۹

امروز هرچقدر سعی می‌کنم یک دیتابیس Sql server ce رو روی هاست اشتراکی منتقل کنم مدام با خطای زیر مواجه می‌شم . راه حل‌های مختلفی رو هم امتحان کردم مثل تغییر سطح تراست یا تغییر نسخه کامپایل به x86 یا x64 یا [این](#) و [این](#) نهایتاً به جایی نرسیدم .

Unable to load the native components of SQL Server Compact corresponding to the ADO.NET provider of version 8854. Install the correct version of SQL Server Compact. Refer to KB article 974247 for more details.

نویسنده: وحید نصیری
تاریخ: ۲۰:۵۵ ۱۳۹۳/۰ ۱/۰۹

روی این هاست قبلاً یک نسخه‌ی SQL Ce نصب شده. الان شماره درایور دات نت SQL Ce که در برنامه‌ی شما در حال استفاده است، تطابقی با نگارش نصب شده‌ی روی سرور ندارد.

بسته کامل آن را از نیوگت دریافت کنید. [از اینجا](#)

علاوه بر ارائه درایور دات نتی، حاوی دو پوشه bin amd64 و x86 کپی شده در پوشه کمپوننت نیز خواهد بود (فایل‌های native اصل بانک اطلاعاتی). همین پوشه‌ها و فایل‌ها را باید به همراه برنامه نیز ارائه کنید.

نویسنده: علی پناهی
تاریخ: ۱۳۹۳/۰۴/۱۱ ۸:۵۰

در کد زیر اگر کانکشن استرینگ، را کامنت کنم بانک ساخته می‌شود و مشکلی ندارد در غیر این صورت پیغام خطای زیر را می‌دهد:

A file activation error occurred. The physical file name 'tax.mdf' may be incorrect. Diagnose and correct additional errors, and retry the operation.

CREATE DATABASE failed. Some file names listed could not be created. Check related errors.

فایل : app.config

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <configSections>

    <section name="entityFramework"
    type="System.Data.Entity.Internal.ConfigFile.EntityFrameworkSection, EntityFramework, Version=6.0.0.0,
    Culture=neutral, PublicKeyToken=b77a5c561934e089" requirePermission="false" />
    <!-- For more information on Entity Framework configuration, visit
    http://go.microsoft.com/fwlink/?LinkId=237468 --></configSections>

  <connectionStrings>
    <clear/>
    <add name="Context"
      connectionString="Data Source=(localdb)\v11.0; Integrated Security=True;
      MultipleActiveResultSets=True; AttachDBFilename=tax.mdf"
      providerName="System.Data.SqlClient"
      />
  </connectionStrings>

  <entityFramework>
    <defaultConnectionFactory type="System.Data.Entity.Infrastructure.LocalDbConnectionFactory,
    EntityFramework">
      <parameters >
        <parameter value="v11.0"/>
      </parameters>
    </defaultConnectionFactory>
  </entityFramework>
<startup><supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.5" /></startup></configuration>
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۹:۵۰

کمی بالاتر در مورد تنظیمات [data directory](#) بحث شده.

قبل از ایجاد Sequence در 2012 Sql Server، توضیح مختصری را درباره آن می‌دهم.
 در واقع Sequence روشی برای تولید اعداد ترتیبی با قابلیت افزایش یا کاهش عددی دلخواه می‌باشد که توسط کاربر یا برنامه نویس ایجاد می‌شود. بنابراین Sequence User-Defined ها در اینجا ممکن است سئوالی پیش بیاید که اینکار توسط Identity هم قابل انجام است، اما چرا استفاده از Sequence توسط مايكروسافت پیشنهاد می‌شود.
 بدلاًیل زیر استفاده از Sequence بهتر می‌باشد:

ممکن است Application شما قبل از درج رکورد، درون یک جدول نیاز به عدد منحصرفردی داشته باشد.
 عدد تولید شده بوسیله Sequence را می‌توانید بین جداول یا ستونهای مختلف یک جدول به اشتراک بگذارید.
 می‌توانید روند تولید اعداد ترتیبی را Restart نمایید. به عبارت دیگر قابلیت Restart نمودن Sequence وجود دارد.
 می‌توانید Sequence خود را براساس Sort یک یا چند فیلد، تنظیم نمایید.

آن به شرح ذیل می‌باشد:

```
CREATE SEQUENCE [schema_name . ] sequence_name
[ AS [ built_in_integer_type | user-defined_integer_type ] ]
[ START WITH <constant> ]
[ INCREMENT BY <constant> ]
[ { MINVALUE [ <constant> ] } | { NO MINVALUE } ]
[ { MAXVALUE [ <constant> ] } | { NO MAXVALUE } ]
[ CYCLE | { NO CYCLE } ]
[ { CACHE [ <constant> ] } | { NO CACHE } ]
[ ; ]
```

شرح Syntax

در زمان ایجاد Sequence، نوع آن می‌بایست عددی باشد، چنانچه Type آن را مشخص ننمایید، SQL Server، نوع آن را bigint در نظر می‌گیرد.

Start With: بدین مفهوم می‌باشد، که Sequence ایجاد شده از چه عددی آغاز شود.
 INCREMENT BY: مفهومش این است که Sequence به چه مقداری افزایش یا کاهش یابد. به عبارت دیگری عدد تولید شده براساس مقدار Increment by تولید می‌شود.

Minvalue: کمترین مقداری که Sequence می‌تواند ایجاد نماید.

Maxvalue: بیشترین مقداری که Sequence می‌تواند ایجاد نماید.

Cycle: مقداری را که برای Cycle تعیین می‌نماییم، بدین مفهوم است که Sequence پس از چه عددی می‌بایست Restart شود.
 Cache: عددی که برای Cache در نظر می‌گیریم، مفهومش این است که چه تعداد از اعداد تولید شده توسط Sequence، قبل از استفاده، می‌تواند در Cache قرار گیرد.

در ادامه با یک مثال ساده، یک Sequence ایجاد می‌نماییم:

```
CREATE SEQUENCE [dbo].[SequenceTest]
AS [int]
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 30
CYCLE
CACHE
GO
```

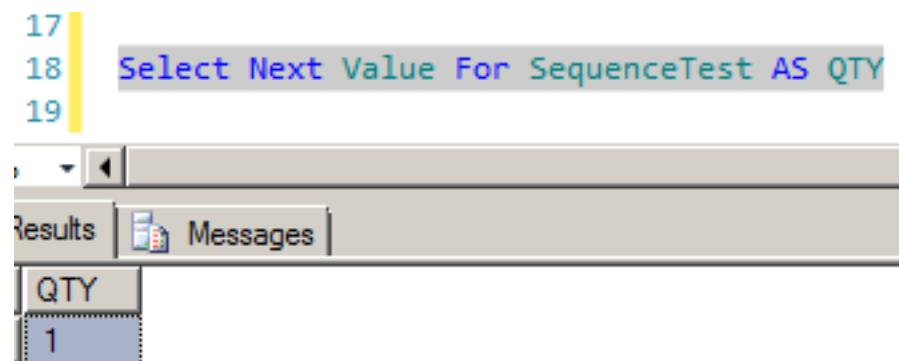
در مثال بالا Start with برابر یک است، یعنی اولین عددی که تولید می‌شود، برابر یک است INCREMENT BY برابر یک است یعنی

در هر بار فراخوانی Sequence یک عدد به عدد تولید شده قبلی افزوده می‌شود. مقدار Minvalue برابر یک است، یعنی کمترین مقداری که Sequence می‌تواند تولید نماید برابر یک است. مقدار Maxvalue برابر 30 است، یعنی بیشترین مقداری که Sequence می‌تواند تولید نماید برابر 30 می‌باشد. و هم برای Sequence فوق در نظر گرفته شده است، بدین معنی، که Sequence به 30 برسد، Restart شده و از مقدار یک شروع به تولید اعداد می‌نماید.

برای اینکه بتوانیم مقدار Sequence را بدست آوریم، کافیست از Syntax زیر استفاده نمایید:

```
NEXT VALUE FOR [ database_name . ] [ schema_name . ] sequence_name
[ OVER (<over_order_by_clause>) ]
```

به عنوان مثال داریم:



اگر Select بالا را تا 30 بار انجام دهید، برای دفعه 31 مقدار آن یک می‌شود، چون در زمان تعریف Sequence، Cycle را انتخاب کردیم. در غیر اینصورت برای دفعه 31 با خطای زیر مواجه می‌شویم.

```
Msg 11728, Level 16, State 1, Line 1
The sequence object 'SequenceTest' has reached its minimum or maximum value. Restart the sequence object to allow new values to be generated.
```

یکی از امکانات جالب Sequence این است که شما می‌توانید Sort شده تنظیم نمایید، برای روش شدن مطلب به مثال زیر توجه نمایید:

در ابتدا مطابق Script زیر جدولی را ایجاد و مقادیری را درون آن درج می‌نماییم:

```
create table Kids
( ID int,
Name varchar(50)
);
Go
insert Kids
values
(1,'Emma')
,(1,'Tabitha')
,(2,'Kendall')
,(3,'DeLaney')
,(4,'Kyle')
,(5,'Jessica')
,(6,'Josh')
,(7,'Kirsten')
,(8,'Amanda')
,(9,'Jimmy')
;
```

سپس یک Schema به نام Sample ایجاد می‌کنیم:

```
CREATE SCHEMA Samples ;
GO
```

```
CREATE SEQUENCE Samples.Test
    AS tinyint
    START WITH 1
    INCREMENT BY 1 ;
GO
```

در ادامه یک Sequence به نام Test ایجاد می‌کنیم:

```
SELECT NEXT VALUE FOR Samples.Test OVER (ORDER BY Name) AS NutID, ID, Name FROM test1.Kids
WHERE Name LIKE '%e%' ;
```

حال زیر را اجرا می‌نماییم:

The screenshot shows the SQL Server Management Studio interface. The query window contains the following code:

```
+1
42 SELECT NEXT VALUE FOR Samples.Test OVER (ORDER BY Name) AS NutID, ID, Name FROM test1.Kids
43 WHERE Name LIKE '%e%' ;
44
```

The Results tab is selected, displaying the following table:

	NutID	ID	Name
1	1	3	Delaney
2	2	1	Emma
3	3	5	Jessica
4	4	2	Kendall
5	5	7	Kirsten
6	6	4	Kyle

امیدوارم مطلب فوق مفید واقع شده باشد.

نظرات خوانندگان

نویسنده: کاربر ۱۳۹۲/۰۴/۲۹
تاریخ: ۱۷:۵۳

سلام مشکلی بر IDENTITY وارد هست بوجود آمدن Gap بین داده هاست مثلا با حذف پی در پی سطرهای جدول بین مقادیر تولید شده Gap بوجود می آید ولی ما این را نمیخواهیم، نیاز داریم که همیشه مقادیر identity پشت سر هم و متوالی باشند یعنی با حذف یک سطر میانی تمام مقادیر شیفت پیدا کنند. و برای رفع این مساله هیچ راهی وجود ندارد. حتی اگر بخواهیم با تابع row_number مقادیر IDENTITY را بعد متوالی کردن بروز رسانی کنیم باز دستور set insert_identity table_name on این اجازه را نمی دهد یعنی این دستور

```
set insert_identity tbl on
update t
set id = new_id
from (select new_id = row_number() over(order by id), * ) t
set insert_identity tbl off
```

در کتابی خواندم که این مشکل با sequence حل میشود.
میخوام بدانم چگونه؟

نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۳۹۲/۰۵/۰۱ ۱۶:۳۰

سلام شما قادر نیستید یک فیلد Identity را بروز رسانی نمایید، دستور set insert_identity Tablename on به شما اجازه به جدول بدون Identity را می دهد، برای اینکه بتوانید Gap مربوط به فیلد Identity را در جدول برطرف کنید، در ابتدا از جدول مورد نظر خود یک کپی تهیه و جدول اصلی را Truncate کنید، سپس یک Sequencer ایجاد و محتویات جدول کپی را بوسیله Sequencer در جدول اصلی کپی نمایید.
فرض کنیم جدول اصلی Table_3 باشد، ابتدا آن را کپی می کنیم در جدولی به نام T

```
Select * into T from table_3
```

سپس دستور Truncate را روی جدول 3 Table_3 اجرا کنید:

```
truncate table dbo.table_3
```

حال یک Sequence ایجاد کنید:

```
CREATE SEQUENCE testEventCounter
AS int
START WITH 1
INCREMENT BY 1 ;
```

در ادامه محتویات جدول کپی را به جدول اصلی منتقل نمایید:

```
SET IDENTITY_INSERT table_3 on
INSERT INTO table_3 (ID, Descritp)
SELECT
    NEXT VALUE FOR testEventCounter AS id
, Descritp
```

FROM T

راه دیگر این است که به جای استفاده از Identity از Sequence در فیلد خود استفاده نمایید، بصورت زیر :

```
CREATE TABLE Table3
(
    ID int PRIMARY KEY CLUSTERED
        DEFAULT (NEXT VALUE FOR SequenceTest),
    De nvarchar(300) NULL
);
GO
```

در هنگام ایجاد جدول Sequence را به فیلد ID ست کردیم.
حال هر زمانی که بخواهید می‌توانید فیلد ID را مطابق Sequence خود بروز رسانی کنید:

```
Update table3 set id=(NEXT VALUE FOR testEventCounter )
```

موفق باشید و امیدوارم مفید واقع شده باشد

نوبسته: کاربر
تاریخ: ۱۳۹۲/۰۵/۰۱ ۲۰:۸

سلام

خیلی ممنون از پاسخی که دادین. مفید واقع شد.
البته فکر کنم برای گارانتی کردن ترتیب sequence می‌توانیم از id می‌توانیم از OVER(ORDER BY) استفاده کنیم.
به این شکل:

```
SET IDENTITY_INSERT table_3 on
INSERT INTO table_3 (ID, Descritp)
SELECT
    NEXT VALUE FOR testEventCounter (OVER ORDER BY ID) AS id
    , Descritp
FROM T
```

به هر شکل با مفهوم جدیدی نیز آشنا شدیم (:)

قبل از مطالعه باید بگویم سطح مقاله مبتدی می‌باشد.

گاهی اوقات در زمان Migration یک دیتابیس شما با جداولی برخورد می‌نمایید که محتویات بعضی از فیلد هایشان ترکیبی از عدد و حروف می‌باشد، و شما برای انجام یکسری از عملیات نیاز دارید. که حروف را از اعداد تمایز نمایید، یا اینکه مکان اولین کاراکتر غیر عددی را بعد از هر عدد، بیابید. برای انجام چنین کاری می‌توان از تابع به نام Patindex استفاده نمود.

تابع PATINDEX به شما امکان می‌دهد که یک یا چند حرف در بین رشته‌های متنه را می‌دهد.

تابع PATINDEX Syntax بصورت زیر میباشد:

```
PATINDEX ( '%pattern%' , expression )
```

تابع PATINDEX شامل دو آرگومان می‌باشد که هر کدام را به اختصار توضیح می‌دهیم:

1- آرگومان اول Pattern نامگذاری شده است، Pattern در واقع یک الگوی اختصاصی میباشد که توسط کاربر، جهت جستجو در یک متن تعیین می‌شود. به بیان ساده‌تر اگر شما دنبال مکان حرف یا کلمه خاصی در یک رشته متنه می‌گردید، می‌بایست آن را در آرگومان Pattern قرار دهید.

لازم است در آرگومان اول حداقل یک % وجود داشته باشد.

حداکثر تعداد کاراکترهایی را که می‌توان در آرگومان اول قرار داد 8000 میباشد.

2- آرگومان دوم یا همان Expression : متنه که عملیات جستجو روی آن اعمال می‌گردد، در این آرگومان قرار می‌گیرد.

اگر تعداد کاراکترهای آرگومان دوم nVarchar(Max) یا Varchar(Max) باشد، در آن صورت Type خروجی تابع PATINDEX از نوع bigint می‌باشد، در غیر این صورت Type خروجی تابع PATINDEX از نوع Int است.

اگر مقدار آرگومان دوم Null باشد، تابع PATINDEX مقدار Null بر می‌گرداند.

یک مثال کاربردی از تابع PATINDEX

یافتن اولین کاراکتر غیر عددی در رکودهای یک جدول

ابتدا در دیتابیس tempdb یک جدول به نام UsingPATINDEX مطابق زیر ایجاد، و چندین رکورد درون آن درج می‌نماییم:

```
USE tempdb
GO
CREATE TABLE UsingPATINDEX (ID INT, Words VARCHAR (100))
GO
INSERT INTO UsingPATINDEX (ID, Words)
SELECT 1, '1one'
UNION ALL
SELECT 2, '11eleven'
UNION ALL
SELECT 3, '2two'
UNION ALL
SELECT 4, '22twentytwo'
UNION ALL
SELECT 5, '111oneeleven'
GO
```

در ادامه QUERY زیر را اجرا نمایید:

```
SELECT PATINDEX('%[a-z]%',Words), 'مکان اولین کاراکتر غیر عددی ' , 'نام اولین کاراکتر غیر عددی بعد از عدد' , 
SUBSTRING(Words,PATINDEX('%[a-z]%',Words),1), 'متن اصلی ' 
FROM UsingPATINDEX
```

خروجی آن به شکل زیر است:

	مکان اولین کاراکتر غیر عددی	نام اولین کاراکتر غیر عددی بعد از عدد	متن اصلی
1	2	o	1one
2	3	e	11eleven
3	2	t	2two
4	3	t	22twentytwo
5	4	o	111oneeleven

توضیح درباره QUERY :

قطعه کد زیر دنبال تمامی حروف a تا z ، درون فیلد Words میگردد و به اولین کاراکتر غیر عددی که میرسد، مکان آن را برمیگرداند.

'مکان اولین کاراکتر غیر عددی' = PATINDEX('%[a-z]%',Words)

قطعه کد زیر با توجه به مکان کاراکتر، خود کاراکتر را برمیگرداند:

'نام اولین کاراکتر غیر عددی بعد از عدد' = SUBSTRING(Words,PATINDEX('%[a-z]%',Words),1)

مثالی دیگر: فرض کنید، دنبال کلمه ای همانند ensure میگردید، بطوریکه دو حرف اول و دو حرف آخر آن را بخاطر میآورید و حرف میانی آن را بخاطر نمیآورید، در آن صورت نیز میتوانید از تابع PATINDEX استفاده نمایید، بدین شکل که به جای حرفی که بخاطر نمیآورید از _ استفاده کنید، همانند QUERY زیر:

```
SELECT PATINDEX('%en_ure%', 'please ensure the door is locked');
```

خروجی عدد 8 میباشد، که مکان حرف e است.

در تابع PATINDEX میتوانید براساس Collation دلخواه عملیات جستجو را انجام دهید، برای روش تر شدن مطلب شکل زیر را مشاهده نمایید:

The screenshot shows the SQL Server Management Studio interface. In the query window, the following T-SQL code is written:

```
Select Patindex( '%ن%', 'استاد' Collate Arabic_CI_AS)
```

The results pane shows a single row with two columns. The first column is labeled '(No column name)' and contains the value '1'. The second column contains the value '3'.

همانطور که در ابتدای مطلبم گفتم در آرگومان اول می‌توان از یک % استفاده نمود، به مثال زیر توجه نمایید:

```
Select PATINDEX('a%', 'abc')
```

خروجی آن مقدار یک است.

مثالی دیگر:

```
Select PATINDEX('%a', 'cba')
```

خروجی آن مقدار 3 می‌باشد.

باید متذکر شویم، با دیدن دو مثال آخر، این تصور ایجاد نشود که تابع PATINDEX شبیه به تابع LIKE می‌باشد، چرا که تابع PATINDEX موقعیت کاراکتر را برابر می‌گرداند، نه خود کاراکتر را، عملکرد تابع PATINDEX شبیه به عملکرد تابع CHARINDEX می‌باشد.

فرض کنید Stored Procedure ی با چند مقدار برگشتی را می‌خواهیم در EF CodeFirst مورد استفاده قرار دهیم. برای مثال فرض کنید زیر را در نظر بگیرید:

```
CREATE PROCEDURE [dbo].[GetAllBlogsAndPosts]
AS
    SELECT * FROM dbo.Blogs
    SELECT * FROM dbo.Posts
```

را Stord Procedure و اکشی کرده و به عنوان خروجی برミگرداند (دو خروجی متفاوت). روش فراخوانی و استفاده از داده‌های این EF CodeFirst به صورت زیر است:

تعریف دو کلاس مدل Blog و Post به ترتیب برای نگهداری اطلاعات و بلاگ‌ها و پست‌ها در زیر آمده است. در ادامه نیز تعریف کلاس BloggingContext را مشاهده می‌کنید.

```
public class Blog
{
    public int BlogId { get; set; }
    public string Name { get; set; }

    public virtual List<Post> Posts { get; set; }
}

public class Post
{
    public int PostId { get; set; }
    public string Title { get; set; }
    public string Content { get; set; }

    public int BlogId { get; set; }
    public virtual Blog Blog { get; set; }
}

public class BloggingContext : DbContext
{
    public DbSet<Blog> Blogs { get; set; }
    public DbSet<Post> Posts { get; set; }
}

using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Data.Entity.Infrastructure;
using System.Data.Objects;

namespace Sproc.Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var db = new BloggingContext())
            {
                db.Database.Initialize(force: false);

                var cmd = db.Database.Connection.CreateCommand();
                cmd.CommandText = "[dbo].[GetAllBlogsAndPosts]";

                try
                {
                    // اجرای پروسیجر
                    db.Database.Connection.Open();
                    var reader = cmd.ExecuteReader();

                    // خواندن رکوردهای blogs
                }
            }
        }
}
```

اجرای EF CodeFirst با چند نوع مقدار برگشتی توسط Stored Procedure

```
var blogs = ((IObjectContextAdapter)db)
    .ObjectContext
    .Translate<Blog>(reader, "Blogs", MergeOption.AppendOnly);

foreach (var item in blogs)
{
    Console.WriteLine(item.Name);
}

// پرسنل نتایج بعدی (همان Posts)
reader.NextResult();
var posts = ((IObjectContextAdapter)db)
    .ObjectContext
    .Translate<Post>(reader, "Posts", MergeOption.AppendOnly);

foreach (var item in posts)
{
    Console.WriteLine(item.Title);
}

finally
{
    db.Database.Connection.Close();
}
}
```

در کدهای بالا ابتدا یک Connection به بانک اطلاعاتی باز می‌شود:

```
db.Database.Connection.Open();
```

و پس از آن نوبت به اجرای Stored Procedure می‌رسد:

```
var reader = cmd.ExecuteReader();
```

در کد بالا پس از اجرای Stored Procedure نتایج بدست آمده در یک reader ذخیره می‌شود. شئ reader از نوع DBDataReader می‌باشد. پس از اجرای Stored Procedure و دریافت نتایج و ذخیره سازی در شئ reader، نوبت به جداسازی رکوردها می‌رسد. همانطور که در تعریف Stored procedure مشخص است این دارای دو نوع خروجی از نوعهای Blog و Post می‌باشد و این دو نوع باید از هم جدا شوند. برای انجام این کار از متod Translate شئ Context استفاده می‌شود. این متod قابلیت کپی کردن نتایج موجود از یک شئ DBDataReader به یک شئ از نوع مدل را دارد. برای مثال:

```
var blogs = ((IObjectContextAdapter)db)
    .ObjectContext
    .Translate<Blog>(reader, "Blogs", MergeOption.AppendOnly);
```

در کدهای بالا تمامی رکوردهایی از نوع Blog از شئ reader خوانده شده و پس از تبدیل به نوع Blog درون شئ Blogs ذخیره می‌شود.

پس از آن توسط حلقه foreach محتویات Blogs پیمایش شده و مقدار موجود در فیلد Name نمایش داده می‌شود.

```
foreach (var item in blogs)
{
    Console.WriteLine(item.Name);
}
```

با توجه به اینکه حاصل اجرای این Stored Procedure دو خروجی متفاوت بوده است، پس از پیمایش رکوردهای Blogs باید به سراغ نتایج بعدی که همان رکوردهای Post می‌باشد برویم. برای اینکار از متod NextResult شئ reader استفاده می‌شود:

```
reader.NextResult();
```

در ادامه برای خواندن رکوردهایی از نوع Post نیز به همان روشی که برای Blog انجام شد عمل می‌شود. [مطالعه بیشتر](#)

نظرات خوانندگان

نویسنده: mashi_ash
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۴:۳۶

لطفا نحوه اجرا کردن پروسیجرها با مقادیر مختلف در ورودی هم کمی توضیح می دادید ممنون می شدم.

با تشکر از توضیحات خوبتون،

نویسنده: محسن_خان
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۴:۵۴

استفاده مستقیم از عبارات SQL در [EF Code first](#) و مثال `void runSp` آن.

نویسنده: بهمن_آبادی
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۶:۱۷

منظور من روش مشابه `parameterAttribute` در mapping کردن ورودی های پروسیجرها در linq می باشد.
نه اینکه صرفا از دستورات sql بصورت command استفاده شود.

مانند استفاده از پروسیجرها با چند ورودی و `multiResult` بودن آن در linq

نویسنده: محسن_خان
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۶:۲۷

در مورد نحوه اجرای رویه های ذخیره شده با ورودی های مختلف پرسیدید، روش اجرаш تا ۵.۰ EF به همین صورتی است که [ملحوظه می کنید](#).

قرار است در [EF 6.0](#) این مساله با Fluent API به نحو دیگری پوشش داده شود.

نویسنده: بهمن_آبادی
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۶:۴۱

دقیقا منظورم همون لینکه EF 6.0 که گفتین هستش.
برای انتشار نسخه جدید EF 6.0 وجود داره که قراره نهایتا تا کی انتشار پیدا بکنه ؟

بازم ممنون.

نویسنده: سید_مهندی_فاتمی
تاریخ: ۱۳۹۲/۰۷/۰۴ ۲۱:۴۷

تشکر دوست عزیز

اما در مدل من که از دیتابیس گرفتم شی رویه ذخیره شده هم به عنوان یک تابع شبیه سازی شده و مثل شما عمل نکردم اما نمی تونم مقادیر رو از این شی بگیرم.

```
var sp1=_db.splogin(user,pas,value1,value2);
```

در توضیح این کد هم بگم که این اس پی 4 پارامتر داره و قراره دو مقدار رو برای من برگردونه .

نویسنده: سانای رحیمی
تاریخ: ۱۳۹۳/۰۴/۲۲ ۱۹:۳۹

سلام

اگر SP ما بدون پارامتر `out` باشد و ما در آخر دستورات آن از دستور `(SCOPE_IDENTITY())` استفاده کرده باشیم چگونه می‌توانیم مقدار آن را به دست بیاوریم؟

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۴/۲۳ ۱۱:۰

در مطلب [استفاده مستقیم از عبارات SQL در EF Code first](#) بحث شده. از متدهای `db.Database.SqlQuery<T>` باید استفاده کنید.

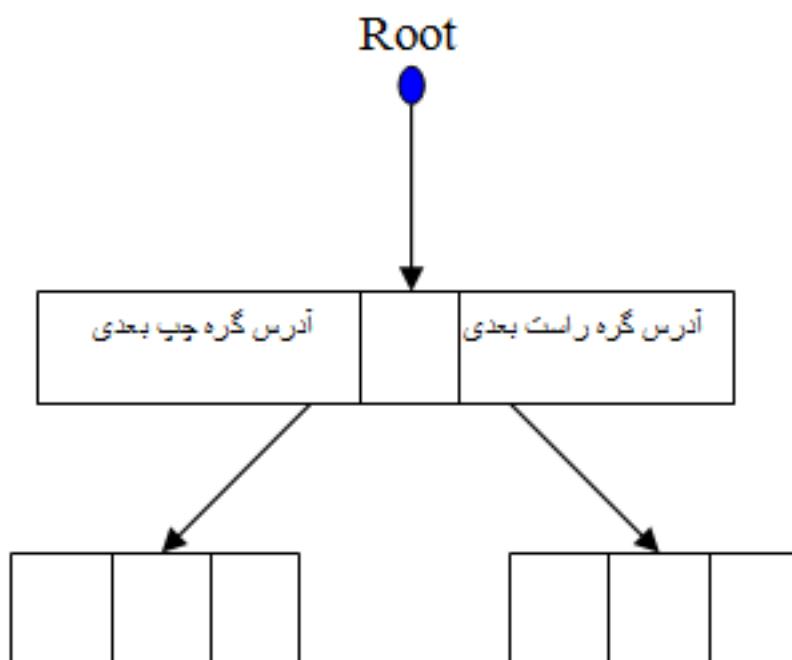
یکی از ویژگی‌های جدید SQL Server 2012 می‌باشد، که کارایی Query های قابل اجرا روى دیتابیس‌های با حجم داده‌ای بسیار بالا را (که اصطلاحاً به آنها Data Warehouse یا انبار داده گویند)، چندین برابر بهبود بخشیده است.

قبل از توضیح در مورد Column Store مختص‌تری در مورد نحوه ذخیره سازی داده‌ها در SQL Server می‌پردازیم. می‌توان گفت در SQL Server دو روش ذخیره سازی وجود دارد، یکی بصورت ردیفی که اصطلاحاً به آن Row Store یا Row-Wise گویند، و دیگری بصورت ستونی که اصطلاحاً به آن Column Store گویند.

در روش ذخیره سازی Row Store، مقادیر ستون‌ها در یک سطر بصورت متوالی ذخیره می‌شوند، در این روش ذخیره سازی از ساختار Heap یا B-Tree استفاده می‌شود.

یادآوری: در ساختار B-Tree، یک گره Root وجود دارد، و گره بعد از Root گره‌ای است که آدرس گره راست بعدی و آدرس گره چپ بعدی را در خود نگه می‌دارد.

شکل زیر نمای یک درخت B-Tree می‌باشد:



جهت کسب اطلاعات بیشتر درمورد ساختار B-Tree

یادآوری: وقتی در یک جدول، ایندکسی از نوع Clustered SQL Server ایجاد نماییم، در ابتدا یک کپی از جدول ایجاد و داده‌های جدول را از نو مرتب می‌نماید، و ساختار صفحه ریشه و دیگر صفحات را ایجاد می‌کند و سپس جدول اصلی را حذف می‌نماید. به جدولی که Clustered Index ندارد، اصطلاحاً Heap گویند.

برخلاف ذخیره سازی Row Store، در ذخیره سازی Column Store، داده‌ها بصورت ستونی ذخیره می‌شوند، در این روش داده‌ها، فشرده سازی می‌شوند و اینکار باعث می‌شود، در زمان درخواست یک Query، نیاز به Disk I/O به حداقل برسد، در نتیجه، زمان و سرعت پاسخگویی به پرس و جوها بسیار افزایش می‌یابد.

شكل زیر نحوه ذخیره سازی داده‌ها، بصورت Row Store را نمایش می‌دهد:

Row Store for B-Tree or Heap

Row1	C1	C2	C3	C4
Row2	C1	C2	C3	C4
Row3	C1	C2	C3	C4

Page 1

Row4	C1	C2	C3	C4
.....	C1	C2	C3	C4
Rown	C1	C2	C3	C4

Page 2

شكل بالا ذخیره سازی داده‌ها، در ساختار B-Tree یا Heap را نمایش می‌دهد، در شکل فوق یک جدول چهار ستونی با N سطر (Row) در نظر گرفته شده است. بطوريکه ستونهای هر Row بطور متواالی در یک صفحه (Page) یکسان ذخیره می‌شوند.

شكل زیر نحوه ذخیره سازی داده‌ها، بصورت Column Store را نمایش می‌دهد:

Column Store Index

Row1	C1	C2	C3	C4
Row2	C1	C2	C3	C4
Row3	C1	C2	C3	C4
Row4	C1	C2	C3	C4
Row5	C1	C2	C3	C4
	Page1	Page2	Page3	Page4

مطابق شکل، ستونهای مربوط به هر Row، همگی در یک صفحه (Page) یکسان ذخیره شده اند. به عنوان مثال ستون C1 که مربوط به سطر اول (Row1) می‌باشد، با ستون C1 که مربوط به سطر دوم (Row2) می‌باشد، در یک ستون و در یک صفحه (Page1) ذخیره شده اند، و الى آخر ...

سؤال: یکبار دیگر به هردو شکل با دقت نگاهی بیاندازید، عمدۀ تفاوت آنها در چیست؟

جواب: درست حدس زدید، تفاوت بارز بین دو روش Row Store و Column Store در نحوه ذخیره سازی داده‌ها می‌باشد. بطور مثال، فرض کنید، در روش ذخیره سازی Row Store، به دنبال مقادیری از ستون C2 می‌باشید، SQL Server می‌بایست کل رکوردهای جدول (منظور همه Row‌ها در همه Page‌ها) را Scan نماید، تا مقادیر مربوط به ستون C2 را بدست آورد. در حالیکه در روش ذخیره سازی Column Store، جهت یافتن مقادیر ستون C2، نیازی به Scan نمودن کل جدول نیست، بلکه SQL Server فقط به ستون C2 Scan نمودن دوست (Page2) یا Page3 می‌نماید. همین امر باعث افزایش چندین برابری، زمان پاسخگویی به هر Query می‌شود.

سؤال: در روش ذخیره سازی Column Store، چگونه مصرف حافظه بهینه می‌شود؟

جواب: واضح است، که در روش SQL Server، Row Store مجبور است، برای بدست آوردن داده‌های مورد نظرتان، کل اطلاعات جدول را وارد حافظه نماید (اطلاعات اضافه‌ای که به هیچ وجه بدرد، نتیجه پرس و جوی شما نمی‌خورد)، و شروع به Scan داده‌های مد نظر شما می‌نماید. بطوریکه در روش SQL Server، Column Store، فقط ستون داده‌های مورد پرس و جو را در حافظه قرار می‌دهد. (در واقع فقط داده‌هایی را در حافظه قرار می‌دهد، که شما به آن نیاز دارید)، بنابراین، طبیعی است که در روش Column Store مقدار حافظه کمتری نسبت به روش Row Store در هنگام اجرای Query استفاده می‌شود. به عبارت دیگر می‌توان گفت که در روش Column Store به دلیل، به حداقل رساندن استفاده از Disk I/o سرعت و زمان پاسخگویی به پرس و جوها چندین برابر می‌شود.

برای درک بیشتر Column Store و Row Store مثالی می‌زنیم:

فرض کنید، قصد بدست آوردن ستونهای C1 و C2 از جدول A را داریم، بنابراین خواهیم داشت:

```
Select C1, C2 from A
```

:Row Store روش

در این روش همه صفحات دیسک (مربوط به جدول A) درون حافظه قرار داده می‌شود، یعنی علاوه بر ستونهای C1 و C2، اطلاعات مربوط به ستونهای C3 و C4 نیز درون حافظه قرار می‌گیرد، بطوریکه مقادیر ستونهای C3 و C4 به هیچ وجه مورد قبول ما نیست، و در خروجی پرس و جوی ما تاثیری ندارد، و فقط بی جهت حافظه اشغال می‌نماید.

:Column Store روش

در این روش فقط صفحات مربوط به ستون C1 و C2 در حافظه قرار می‌گیرد. منظور Page1 و Page2 می‌باشد) بنابراین فقط اطلاعات مورد نیاز در خروجی، در حافظه قرار می‌گیرد. از دیگر مزایای استفاده از روش Column Store، فشرده سازی داده می‌باشد، برای درک بیشتر توضیح می‌دهم:

همانطور که در اوایل مطلب به عرض رساندم، در روش Row Store، داده‌ها در یک سطر و در یک Page ذخیره می‌شوند، بنابراین امکان وجود داده‌های تکراری در یک سطر به حداقل می‌رسد، چرا که، اگر فرض کنیم چهار ستون به نام‌های ID, FirstName, LastName و City، داشته باشیم، در آن صورت بطور حتم، در یک سطر، داده تکراری وجود نخواهد داشت، اما ممکن است در تعداد سطرهای زیاد داده‌های تکراری مانند City Firstname یا غیره بوجود بیاید، این موضوع را بیان کردم، چون می‌خواستم عنوان کنم، بسیاری از الگوریتم‌های فشرده سازی از الگوی تکراری بودن داده، جهت فشرده سازی داده‌ها استفاده می‌کنند، به همین جهت فشرده سازی در روش Row Store به حداقل می‌رسد و فضای اشغال شده در حافظه در این روش بسیار زیاد خواهد بود. اما در روش Column Store، امکان تکراری بودن مقادیر یک ستون بسیار زیاد است، بطور مثال ممکن است تعداد افرادی را که نام شهر آنها "تهران" باشد مثلا 20 بار تکرار شده باشد، و چون در روش Column Store، ستون‌ها در یک Page ذخیره می‌شوند، بنابراین امکان استفاده از الگوریتم‌های فشرده سازی در این روش بسیار بالا می‌باشد، در نتیجه مقدار فضایی را که در حافظه یا دیسک سخت توسط این روش اشغال می‌شود، بسیار کمتر از روش Row Store است. چه موقع می‌توانیم از Column Store استفاده نماییم:

در تعریف Column Store گفته بودم، روش فوق، جهت بپردازید بخشیدن به زمان و سرعت پاسخگویی به Queries را اجرا شده روی دیتابیس‌های با حجم داده ای بسیار بالا (Data Warehouse) می‌باشد، به بیان ساده‌تر Column Store را روی دیتابیس‌های offline یا دیتابیس‌هایی که صرفاً جهت گزارش گیری مورد استفاده قرار می‌گیرند، تنظیم می‌نمایند. در واقع با تنظیم Column Store Index روی Database‌های بزرگ مانند Database‌های بانک‌ها که حجم داده ای میلیونی در جداول آنها وجود دارد، سرعت پاسخگویی Query‌ها، چندین برابر افزایش می‌یابد. در یک جدول می‌توانید، هم یک Row Store Index داشته باشید و هم یک Clustered Index می‌توانید Column Store Index را باشد ()

برای ایجاد Column Store Index Syntax به شرح ذیل می‌باشد:

```
CREATE [ NONCLUSTERED ] COLUMNSTORE INDEX index_name
    ON <object> ( column [ ,...n ] )
    [ WITH ( <column_index_option> [ ,...n ] ) ]
    [ ON {
        { partition_scheme_name ( column_name ) }
        | filegroup_name
        | "default"
    } ]
    [ ; ]
<object> ::=
{
    [ database_name. [ schema_name ] . | schema_name . ]
    table_name
```

```
{
<column_index_option> ::=
{
    DROP_EXISTING = { ON | OFF }
    | MAXDOP = max_degree_of_parallelism
}
```

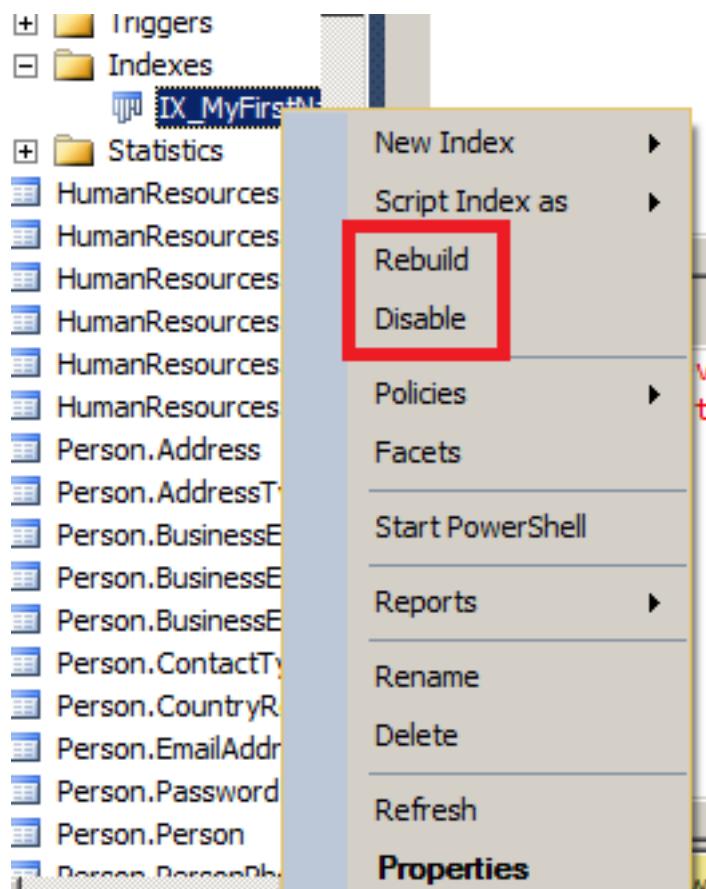
یک Column Store Index می‌بایست از نوع NONCLUSTERED باشد.

مثلاً از یک Column Store Index :

```
CREATE NONCLUSTERED COLUMNSTORE INDEX [IX_MyFirstName_ColumnStore]
ON [Test]
(Firstname)
```

در قطعه کد بالا، یک فیلد Firstname روی نام Column Store Index به نام IX_MyFirstName_ColumnStore از جدول Test ایجاد شده است.

حدودیت‌های استفاده از Column Store Index به اختصار به شرح ذیل می‌باشد:
زمانی که در یک جدول، یک Column Store Index ایجاد نماییم، جدول ما در حالت Read-only قرار می‌گیرد، بطوریکه از آن پس اختیار Insert, Update, Delete و روش فوق را نخواهیم داشت. برای اینکه بتوانید عملیات Insert, Update, Delete را انجام دهید، می‌بایست Column Store Index جدول مربوطه را Disable نمایید، و برای فعال نمودن Column Store Index می‌بایست آن را Rebuild نمایید، با کلیک راست روی ایندکس ایجاد شده در SQL Server 2012 موارد Rebuild و Disable قابل مشاهده می‌باشد.



یا بوسیله Script‌های زیر می‌توانید، عملیات Disable یا Rebuild را روی Column Store Index انجام دهید:

```
ALTER INDEX [IX_MyFirstName_ColumnStore] ON [Test] DISABLE  
ALTER INDEX [IX_MyFirstName_ColumnStore] ON [Test] Rebuild
```

بیشتر از یک Column Store Index نمی‌توانید روی یک جدول ایجاد نمایید.

در صورتی که تمایل داشته باشید بوسیله Alter ، نوع فیلدی (Type)، را که Column Store Index روی آنها اعمال گردیده است، تغییر دهید، در ابتدا می‌بایست Column Store Index خود را Drop یا حذف نمایید، سپس عملیات Alter را اعمال کنید، در غیر اینصورت با خطای SQL Server مواجه می‌شوید.

یک Column Store Index می‌تواند روی 1024 ستون در یک جدول اعمال گردد.

یک Column Store Index نمی‌توانند، Unique باشد و نمی‌توان از آن به عنوان Primary Key یا Foreign Key استفاده نمود.

یاد آوری: با توجه به مزایای استفاده از Column Store Index، باید بگوییم که در حجم‌های داده ای کم استفاده از Row Store Index بهتر می‌باشد. پیشنهاد مایکروسافت برای استفاده از Column Store Index برای دیتابیس‌های با حجم داده ای بسیار بالا می‌باشد. موفق باشید

منابع:

["Columnstore Indexes: A New Feature in SQL Server known as Project ‘Apollo’"](#)

[Columnstore Indexes](#)

[Fundamentals of Columnstore Index](#)

[SQL Server 2012 Column Store Index Example](#)

نظرات خوانندگان

نویسنده: صالح
تاریخ: ۰۵۷ ۱۳۹۱/۰۷/۳۰

دروド بر شما
بسیار بهره جستم از مطلبتون. عالی بود.

نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۱:۱۶ ۱۳۹۱/۰۷/۳۰

ممnon از لطفا شما

نویسنده: علیرضا جهانشاھلو
تاریخ: ۲۲:۵۷ ۱۳۹۱/۰۷/۳۰

بسیار بهره بردم از مطلب خوبتون
با تشکر

نویسنده: libertad
تاریخ: ۱۰:۳۴ ۱۳۹۱/۰۸/۰۳

آیا چنین امکانی در oracle هم وجود دارد؟

نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۲:۳۹ ۱۳۹۱/۰۸/۰۳

بله، چنین امکانی موجود است، برای اطلاعات بیشتر می‌توانید به آدرس زیر مراجعه کنید:
[Compressing Columns](#)

نویسنده: مهران زند
تاریخ: ۱۶:۲۸ ۱۳۹۱/۰۸/۰۵

بسیار عالی و ممنون. نتیجتاً این گونه ایندکس گذاری برای دیتابیس‌های که درد حال رشد هستند یعنی مداوم در حال update .. insert هستند به درد نخواهد خورد.

نویسنده: علی قمشلوی
تاریخ: ۲۰:۵۵ ۱۳۹۱/۰۸/۰۵

با سلام و تشکر به خاطر مطلب عالی و فوق العاده قابل فهمتون
یکی از سئوالاتی که همیشه برای هنگام کوئٹری گرفتن از دیتابیس پیش می‌آمد این بود که :
`select c1 from A1 where c1=x`

`select * from A1 where c1=x`

کدامیک دارای کارایی بیشتری می‌باشند که در این مقاله به طور عالی توضیح داده شده بود.
البته جوابشو نمیدم تا دوستان این مقاله زیبا را از دست ندهند.

نویسنده: محمد صاحب
تاریخ: ۲۲:۵۸ ۱۳۹۱/۰۸/۰۵

البته موردی که در این مقاله عنوان شده یکی از مشکلات Select * هست...

نویسنده: ali

تاریخ: ۲۳:۴۵ ۱۳۹۲/۰۹/۰۵

با سلام یک جایی مطلب شما گنج ماند " یعنی علاوه بر ستونهای C1 و C2، اطلاعات مربوط به ستونهای C3 و C4 نیز درون حافظه قرار می‌گیرد " پس چرا میگن تا آنجا که میشه برای بالا بردن performance در دستور select حتما تعداد فیلدهای آن مشخص باشد با آنچه شما گفتید C3 و C4 درون حافظه لود میشوند اینطور نیست ممنون

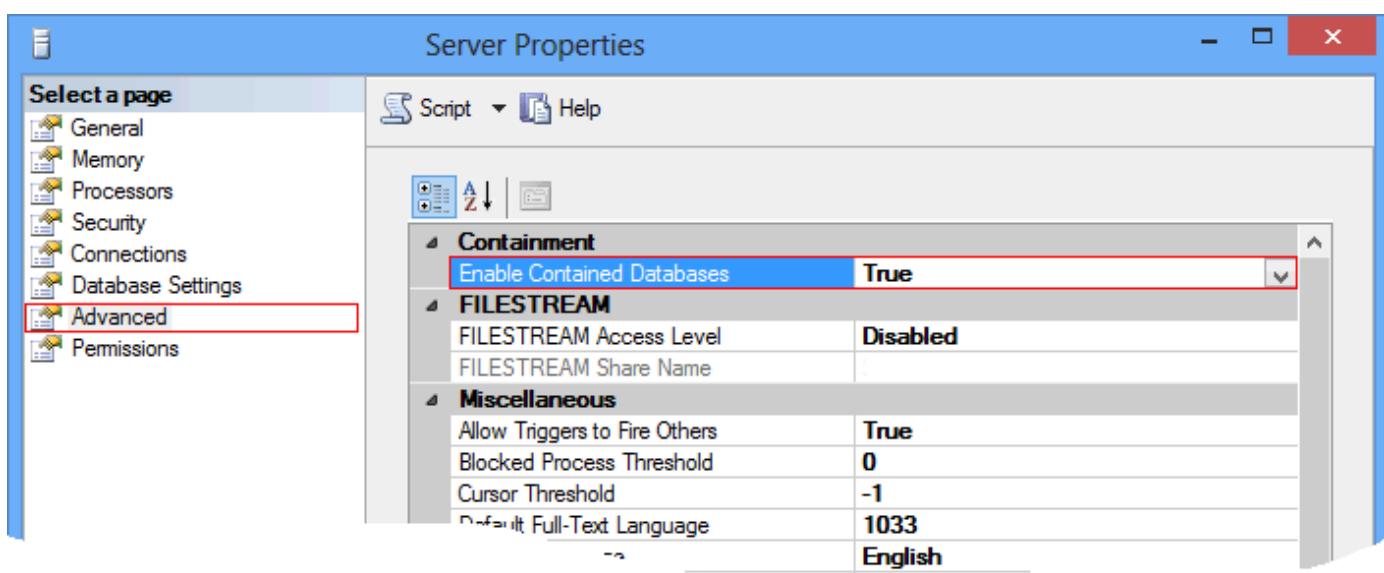
مقدمه: بعضی وقت‌ها به هر دلیل لازم است پایگاه داده برنامه هایمان را به سرور دیگری انتقال دهیم. برخلاف Oracle که انتقال پایگاه داده به سرور دیگر کار مشکل و تخصصی می‌باشد در SQL Server براحتی با یک Attach و Detach مجدد این انتقال انجام خواهد شد. اما اطلاعات Logins در پایگاه داده Master سرور ذخیره شده است و با فایل پایگاه داده انتقال نمی‌باید و به همین خاطر برخی مواقع پس از Attach پایگاه داده با پیغام The database is not accessible یا Cannot open user default database مراجع خواهیم شد که مشکل بخاطر همین نبود کاربران سرور قبلی و Owner متفاوت پایگاه داده می‌باشد که با استفاده از رویه ذخیره شده sp_change_users_login این مشکل قابل حل می‌باشد اما در SQL Server 2012 راحل بهتری فراهم شده است.

Sql Server 2012 در Contained Databases

یکی از امکانات اضافه شده در SQL Server 2012 امکان Contained Databases هست که برای حل همین مشکل و رفع کامل وابستی پایگاه داده با Instance نصب شده و خاصیت قابل حمل کامل database‌ها می‌باشد. در ادامه نحوه استفاده از این امکان را بررسی خواهیم کرد:

فعال کردن خاصیت Contained Databases

قبل از استفاده از Contained Databases می‌بایست این امکان را فعال کرد. برای این کار می‌توانید از SQL Server Management Studio یا T-SQL commands استفاده نمایید. بر روی نام Instance راست کلیک کنید و گزینه Properties را انتخاب نمایید. از گزینه که در شکل زیر مشاهده می‌نمایید خاصیت Enable Contained Databases را بر روی True قرار دهید.



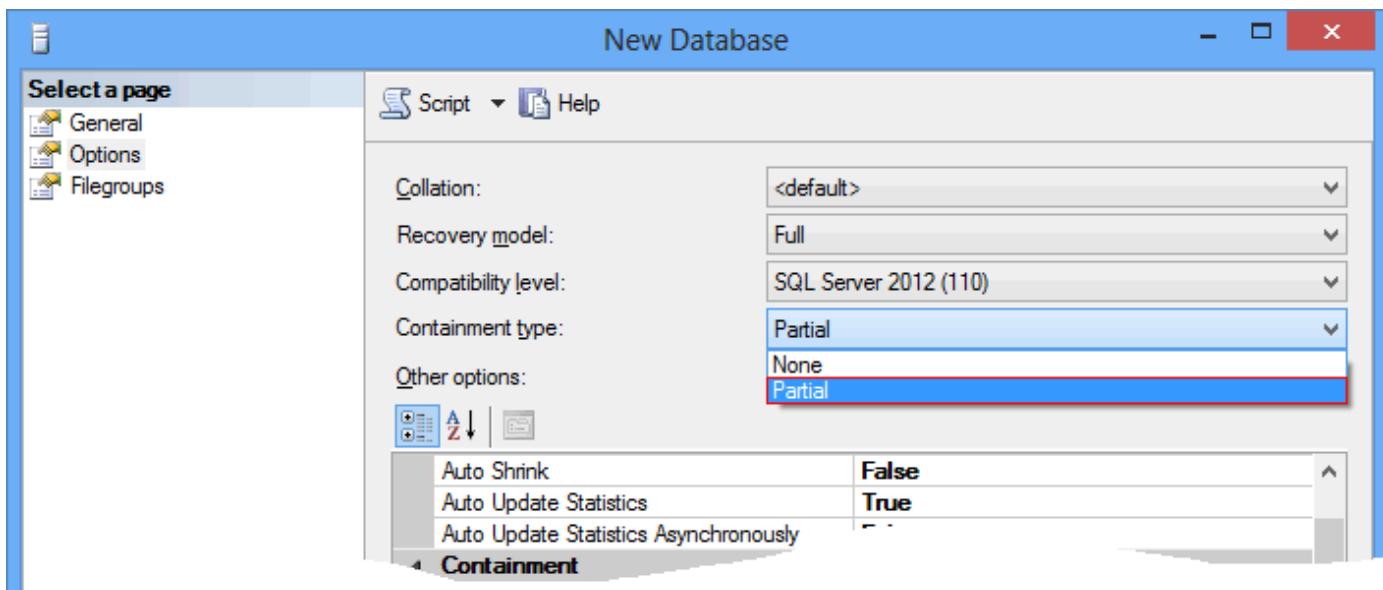
یا می‌توانید از sp_configure این کار را انجام دهید. دستورات زیر این موضوع را نشان می‌دهد.

```
sp_configure 'show advanced options',1
GO
RECONFIGURE WITH OVERRIDE
GO
sp_configure 'contained database authentication',1
GO
RECONFIGURE WITH OVERRIDE
```

GO

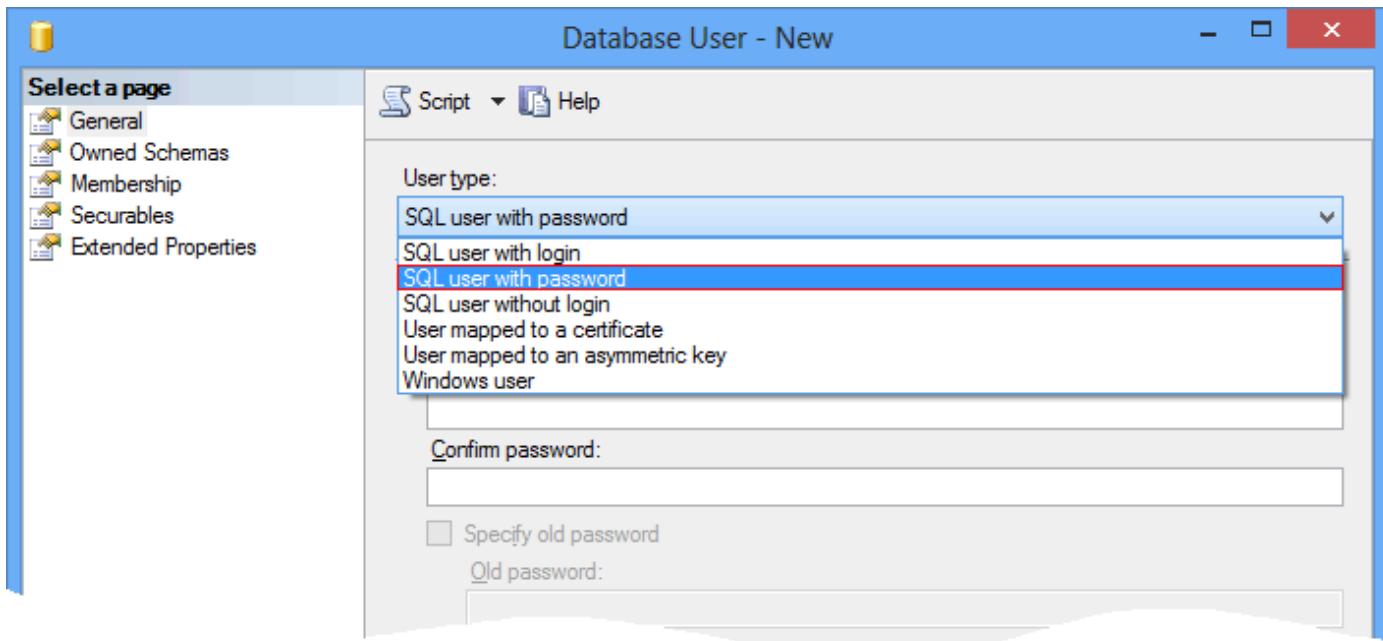
ایجاد یا تغییر یک پایگاه داده از نوع Contained Databases

برای ایجاد یک پایگاه داده با این خاصیت یا تغییر پایگاه داده موجود کافیست مقدار گزینه Containment type را بر روی Partial قرار دهید. برای پایگاه داده موجود از پنجره Properties پایگاه داده صفحه Options را انتخاب کنید.



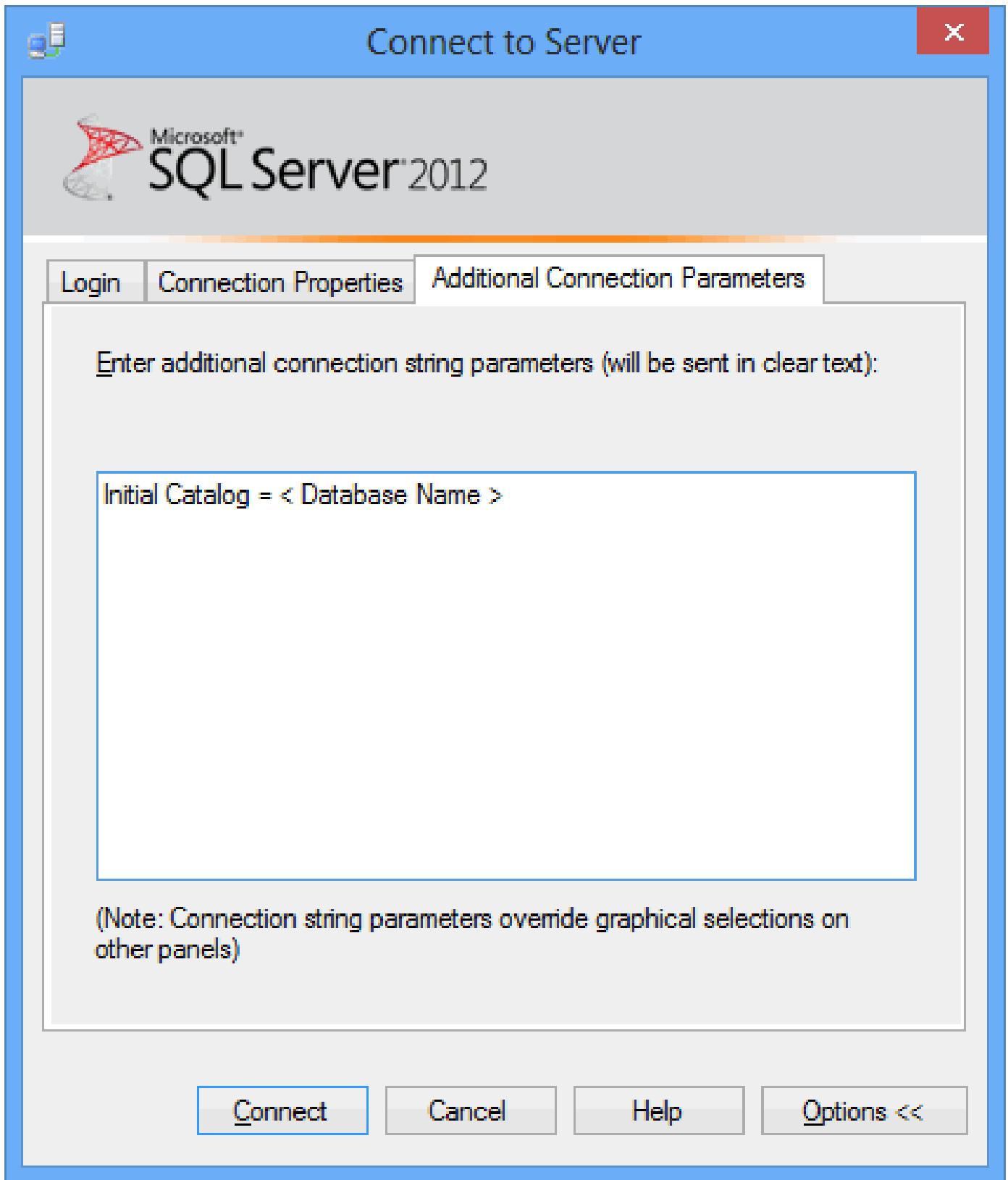
ایجاد یک کاربر برای پایگاه داده Contained Databases

برای تعریف یک کاربر در سطح پایگاه داده پوشه Security پایگاه داده خود را باز کنید بر روی پوشه Users راست کلیک و گزینه New User را انتخاب نمایید از گزینه User type که در شکل زیر نشان داده شده است SQL user with password را انتخاب نمایید و نام کاربر و رمز عبور و تکرار آن را وارد نمایید. کاربر ایجاد شده در سطح پایگاه داده می‌باشد و با انتقال به سرور دیگر نیز قابل دسترسی می‌باشد.



اتصال به پایگاه داده Contained Databases

برای اتصال به پایگاه داده کافیست در حالت SQL Server Authentication نام کاربری و رمز عبور جدید را وارد و گزینه Options را انتخاب و از برگه Additional Connection Parameters نام پایگاه داده مورد نظر را مانند شکل زیر وارد نمایید پس از ورود تنها پایگاه داده خود را مشاهده می‌نمایید. یکی از کاربرهای این قابلیت برای مدیران سرور پایگاه داده می‌باشد که بدون استفاده از مجوز sysadmin به کاربران اجازه دسترسی را می‌دهد.



نظرات خوانندگان

نویسنده: ابراهیم شادا فروغ
تاریخ: ۱۳۹۱/۰۸/۱۱ ۱۱:۵۳

مقاله عالی بود من با این مشکل در یکی از پروژه های کاری ام برخورد کردم و با مطالعه و تمرین و خطا موفق به پیدا کردن راه حل این مشکل شدم
خیلی خوبه که راه کار حل موضوعاتی این چنین در وب فارسی وجود داشته باشه

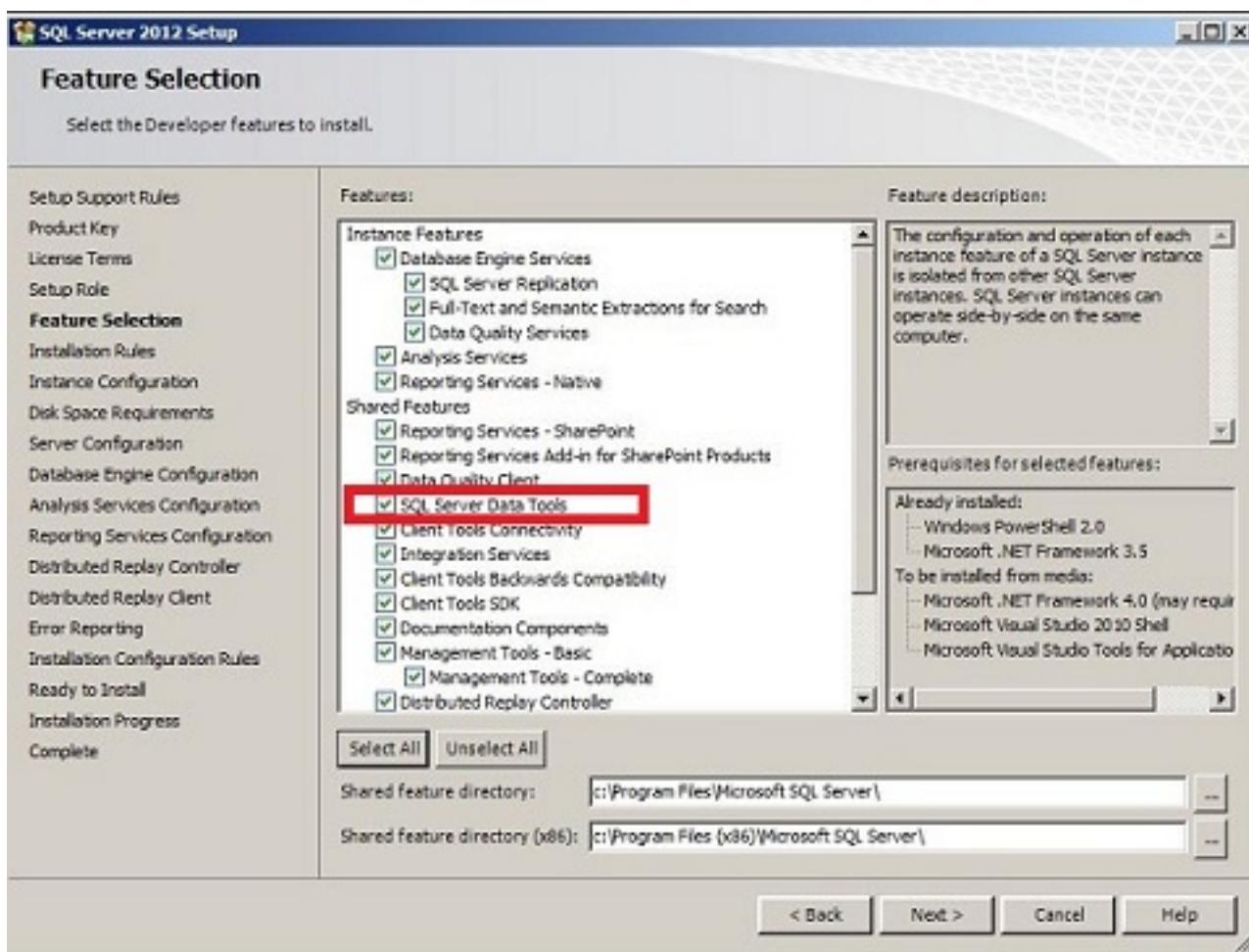
مقدمه

یکی از امکانات اضافه شده در SQL Server 2012 ابزار SQL Server Data Tools می‌باشد. این ابزار در واقع جایگزین ابزار SQL Server 2012 یا به اختصار SSDT می‌باشد. این ابزار در داشته و همچنین قابلیت‌های جدیدی نیز به آن اضافه شد است. اما کاربرد این ابزار ایجاد محیطی یکپارچه برای طراحی و توسعه تمامی نسخه‌های پایگاه داده با استفاده از Visual Studio بر روی امکانات زیر تمرکز دارد:

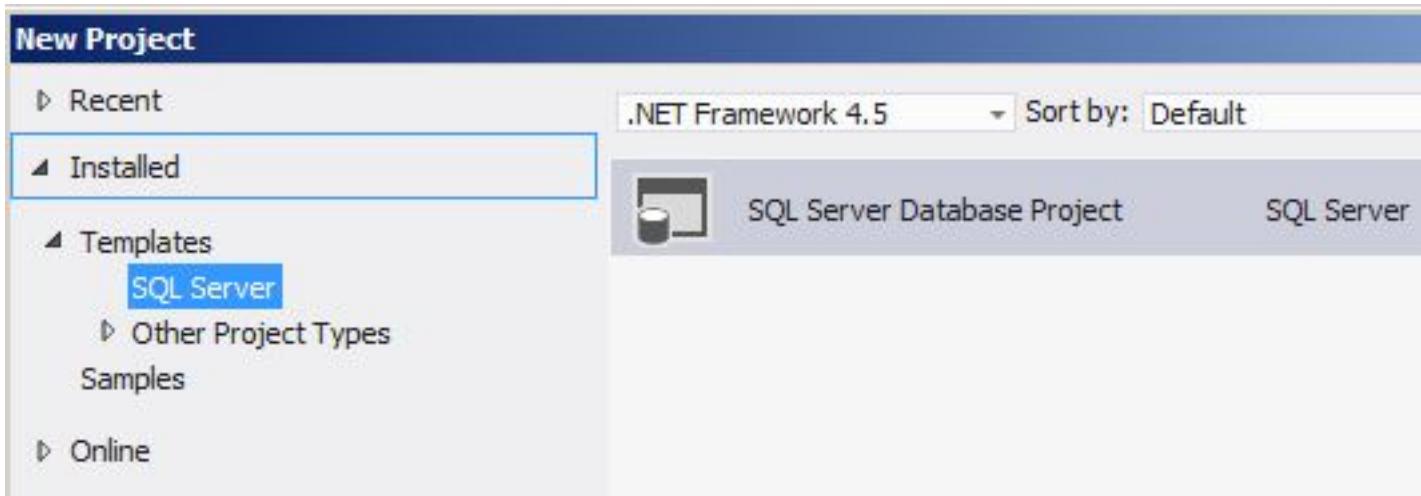
Installation
 Creating a SQL Server Database Project
 Modify Database Schema
 Schema Compare and Update
 Snapshot Project
 Publish

نصب SQL Server Data Tools

آخرین نسخه این ابزار را از این [آدرس](#) دانلود کنید یا زمان نصب SQL Server 2012 گزینه آن را انتخاب کنید



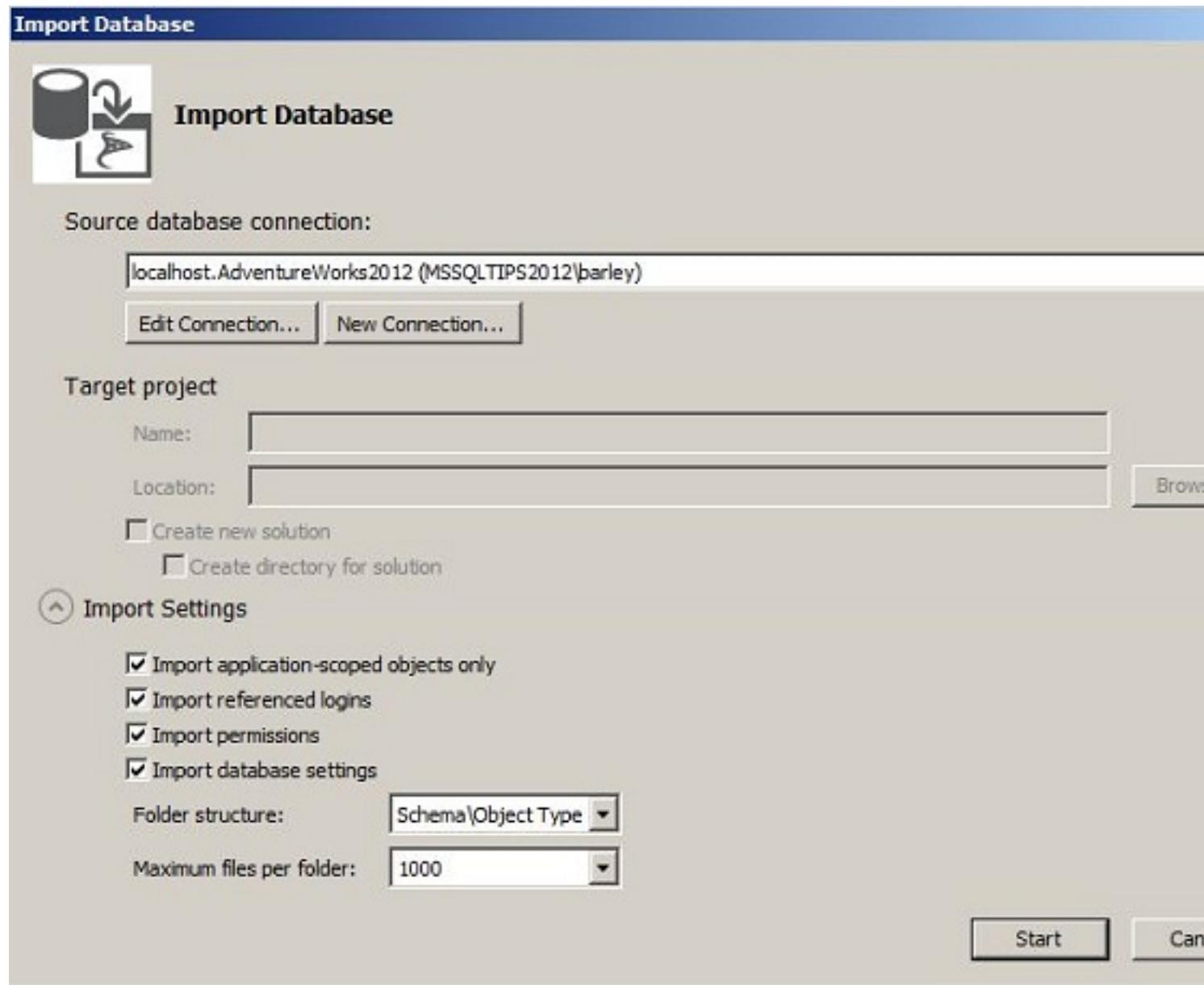
ایجاد یک پروژه جدید از نوع SQL Server Database Project پس از نصب SSDT شما از طریق SqlServer 2012 که همراه Visual Studio 2012 Shell نصب می‌شود یا با 2012 یک پروژه جدید از نوع SQL Server Database Project ایجاد کنید.



Visual Studio به شما امکان توسعه پایگاه داده SQL Server را با استفاده از محیط یکپارچه Project با در اختیار گذاشتن ابزارهای همچون navigation, intellisense, validation, debugging, declarative editing می‌دهد. شی پایگاه داده شما در پروژه و فایل‌های مجزا ذخیره می‌شود مثل آن که شما در حال توسعه برنامه C# یا VB.NET هستید.

ورود پایگاه داده موجود به پروژه

شما می‌توانید شمای پایگاه داده موجود (.dacpac) یا هر T-SQL DAC Package File (SQL Server) یا بر روی پروژه و انتخاب گزینه Import به پروژه خود اضافه و تغییرات لازم را اعمال نماید شکل زیر پنجره Import Database را نشان می‌دهد:

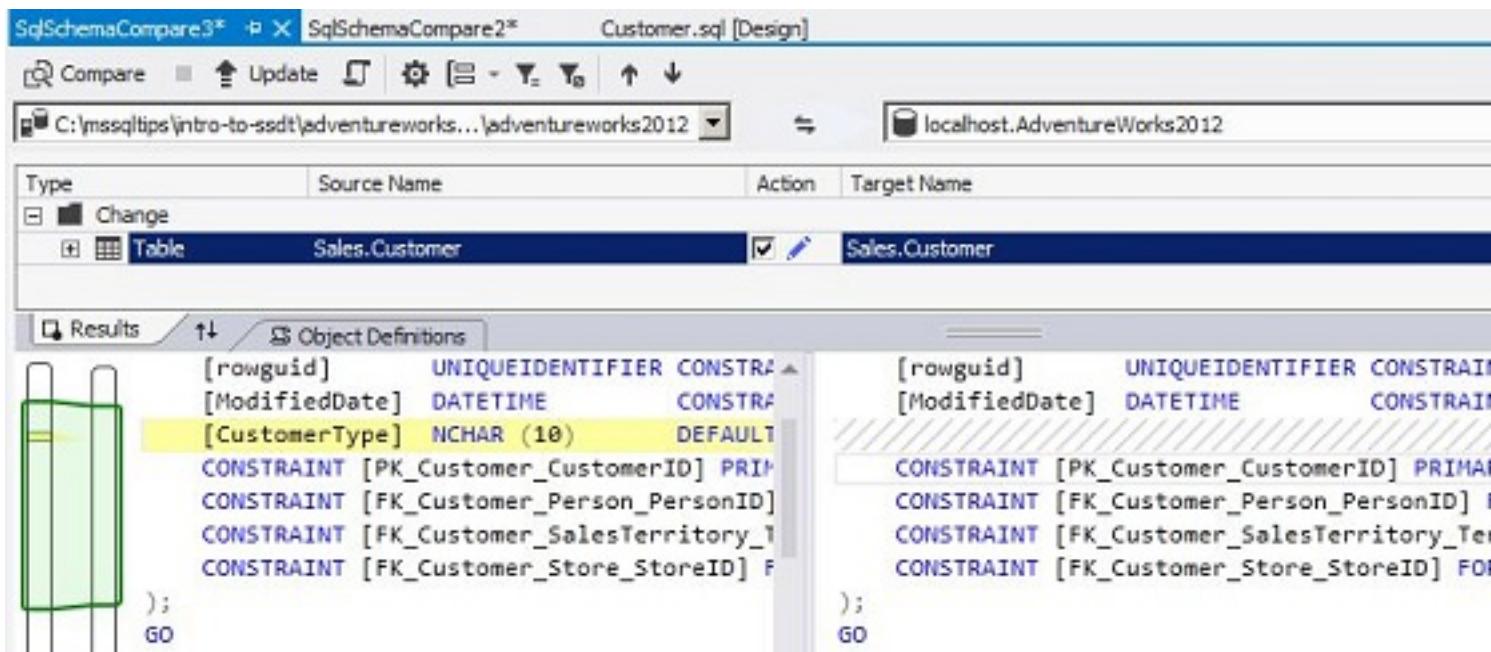


پنجره Solution Explorer امکانات زیر را در اختیار شما قرار می دهد:

- اضافه کردن اشیای همچون Table, View و غیره با راست کیلک کردن بر روی پروژه و انتخاب گزینه Add
- ویرایش اشیا موجود با دبل کیلک کردن بر روی اشیا
- مقایسه شمای پایگاه داده با پایگاه داده دیگر و یا Microsoft SQL Server DAC Package File کافیست بر روی پروژه راست کیلک و گزینه Schema Compare را انتخاب نمایید.
- ایجاد یک Snapshot از شمای پایگاه داده در یک Microsoft SQL Server DAC Package File. یک snapshot برای ایجاد یک پایگاه داده یا ورود در پروژه ای دیگر یا در مقایسه دو پایگاه داده کاربرد دارد.
- انتشار پایگاه داده که امکان ایجاد پایگاه داده بر روی یک سرور SQL Server را فراهم می نماید.

مقایسه شمای دو پایگاه داده با هم

SSDT ابزاری برای مقایسه دو پایگاه داده و بروز رسانی پایگاه داده مقصود از روی شمای منبع از طریق ایجاد یک Script یا به صورت مستقیم را می دهد.



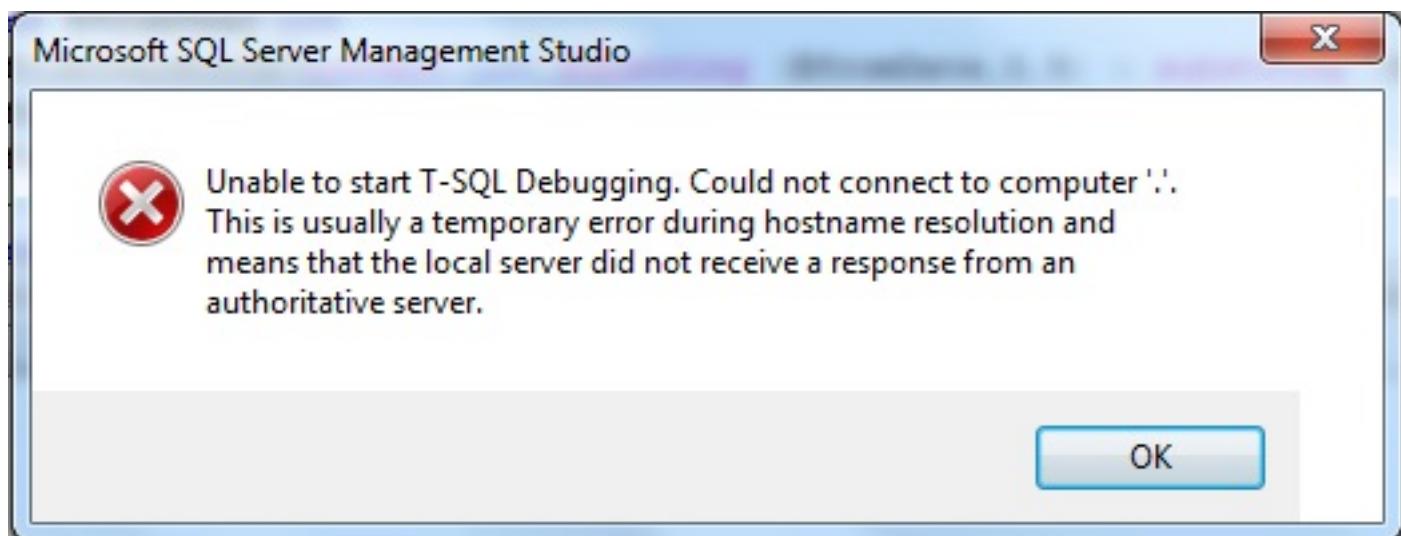
انتشار پروژه با استفاده از SQL Server Data Tools

این ابزار امکان گسترش پروژه شما بر روی SQL Azure instance, SQL Server 2005, 2008, 2008 R2, 2012 را می‌دهد. برای انتشار کافیت از پنجره Solution Explorer بر روی پروژه راست کلیک و گزینه Publish را انتخاب کنید.

پنجره SQL Server Object Explorer

این پنجره با اتصال به سرور SQL Server امکان دسترسی و مدیریت پایگاه داده را به ما می‌دهد. علاوه بر دسترسی به اجزای یک پایگاه داده امکان مدیریت پوشش‌های Security, Server Objects نیز فراهم می‌باشد. همچنین امکان اتصال به Sql Server Express, localdb نیز وجود دارد.

در حین کار با SQL Server نیاز به دیباگ اسکریپتی طولانی و اورژانسی T-SQL بود که در محیط Management Studio با مشکل زیر برخورد کردم:



در این مورد نظرات و پیشنهادات زیادی از جمله ریستارت سرویس SQL Server و تغییض کلمه عبور لاغین و یا پاک کردن کلمات عبور کش شده در سیستم و حتی ریستارت کامپیوتر (: از دوستان همکار در فروم‌های موجود در اینترنت گذاشته شده بود و در گوشه ای هم اشاره به '!'. شده بود که طبق عادت همیشگی برای لاغین به بانک استفاده میکردم و خواسته شده بود که برای لاغین لوکال به SQL Server از نام کامپیوتر بجای '!'. استفاده شود که مفید فایده بود.

مقدمه

مقدار null به معنی پوچ و هیچ می‌باشد اما زمانی که در مقدار دهی جداول از آن استفاده می‌نماییم با توجه به نوع آن ستون فضای متفاوتی اشتغال می‌نماید. شاید در پایگاه داده‌های کوچک زیاد مطرح نباشد اما زمانی که حداقل چند گیگ حجم آن باشد و فرضاً ۲۰ تا ۳۰ درصد آن از مقادیر null پر شده باشد فضای زیای از پوچ گرفته شده است این در حالی است که خیلی از توسعه دهنده‌گان اصلاً به اهمیت استفاده از null توجهی نمی‌کنند و از مقادیری غیر معتبری مثل ۰ یا -۱ در آن ستون به جای null استفاده می‌کنند.

SQL Server Sparse Columns

یا ستون‌های تنک قابلیتی از که از 2008 SQL Server اضافه شده و به ستون‌های عادی امکان استفاده بهینه از فضای ذخیره شده برای مقادیر null را می‌دهد. در واقع sparse column فضای مورد نیاز برای مقادیر null نسبت به مقادیر غیر null را کاهش می‌دهد. با استفاده از sparse column فضای ذخیره شده حداقل ۲۰ تا ۴۰ درصد کمتر خواهد شد.

Sparse Columns ویژگی‌های

ستون‌های SPARSE از کلمه کلیدی SQL Server Database Engine برای تعریف یک ستون که مقادیر آن می‌بایست بهینه شود استفاده می‌نماید.

نمای Catalog جداول با ستون sparse شبیه جداول معمولی می‌باشد.
مقدار برگشته از تابع COLUMN_UPDATED با ستون sparse متفاوت از ستون معمولی است.
در نوع داده‌های زیر امکان استفاده از sparse columns را ندارند:

text	geography
timestamp	geometry
user-defined data types	image
	ntext

فضای بیشتری برای ذخیره داده‌های غیر null نسبت به داده‌های نشانه گذاری نشده با SPARSE لازم دارد و این فضای بیشتر از ستون معمولی است. برآورده فضای ذخیره شده براساس نوع داده با طول ثابت در جدول زیر آورده شده است:

نوع داده	بایت بدون sparse	بایت sparse	بایت sparse	درصد null
bit	0.125	1	5	98%
tinyint	2	1	5	86%
smallint	4	2	6	76%
int	8	4	8	64%
bigint	8	8	12	52%
real	4	4	8	64%
float	8	8	12	52%
smallmoney	4	4	8	64%
money	8	8	12	52%

استفاده از Sparse Columns در SQL Server 2012

نوع داده	بایت بدون sparse	بایت sparse	درصد null
smalldatetime	4	8	64%
datetime	8	12	52%
uniqueidentifier	16	20	43%
date	3	7	69%

نوع داده با دقت - وابسته به طول

نوع داده	بایت بدون sparse	بایت sparse	درصد null
(datetime(2	6	10	57%
(datetime(2	8	12	52%
(time(0	3	7	69%
(time(7	5	9	60%
(datetimetoffset(0	8	12	52%
(datetimetoffset (7	10	14	49%
decimal/numeric(1,s)	5	9	60%
decimal/numeric(38,s)	17	21	42%
vardecimal(p,s)			

نوع داده - داده وابسته به طول

نوع داده	بایت بدون sparse	بایت sparse	درصد null
sql_variant	2*	2*	60%
varchar or char	2*	4*+	60%
nvarchar or nchar	2*	4*	60%
varbinary or binary	2*	4*	60%
xml	2*	4*	60%
hierarchyid	2*	4*	60%

محدودیت‌های استفاده از Sparse columns

sparse column می‌باشد و نمی‌تواند nullable باشد.

sparse column مقدار پیش فرض نمی‌تواند داشته باشد

ستون محاسبه‌ای نمی‌تواند sparse باشد

نمی‌تواند بخشی از clustered index باشد

sparse column نمی‌تواند بخشی از user-defined table باشد

مثالی از کاربرد Sparse columns

```
CREATE TABLE Employees_sparse (
    EMP_ID INT IDENTITY(5001,1) PRIMARY KEY,
    SSN CHAR(9) NOT NULL,
    TITLE CHAR(10) SPARSE NULL,
```

استفاده از Sparse Columns در SQL Server 2012

```
FIRSTNAME VARCHAR(50) NOT NULL,  
MIDDLEINIT CHAR(1) SPARSE NULL,  
LASTNAME VARCHAR(50) NOT NULL,  
EMAIL CHAR(50) SPARSE NULL)  
GO
```

```
CREATE TABLE Employees (  
    EMP_ID INT IDENTITY(5001,1) PRIMARY KEY,  
    SSN CHAR(9) NOT NULL,  
    TITLE CHAR(10) NULL,  
    FIRSTNAME VARCHAR(50) NOT NULL,  
    MIDDLEINIT CHAR(1) NULL,  
    LASTNAME VARCHAR(50) NOT NULL,  
    EMAIL CHAR(50) NULL)  
GO
```

در این دو جدول یکی با سه ستون Sparse و دیگری بدون Sparse ایجاد شده و با 50000 ردیف داده پر شده است حال با رویه ذخیره شده sp_spaceused می‌توان فضای ذخیره شده دو جدول را باهم مقایسه نمود.

```
sp_spaceused 'Employees'  
GO  
sp_spaceused 'Employees_sparse'
```

	name	rows	reserved	data	index_size	unused
1	Employees	50000	5064 KB	5008 KB	32 KB	24 KB

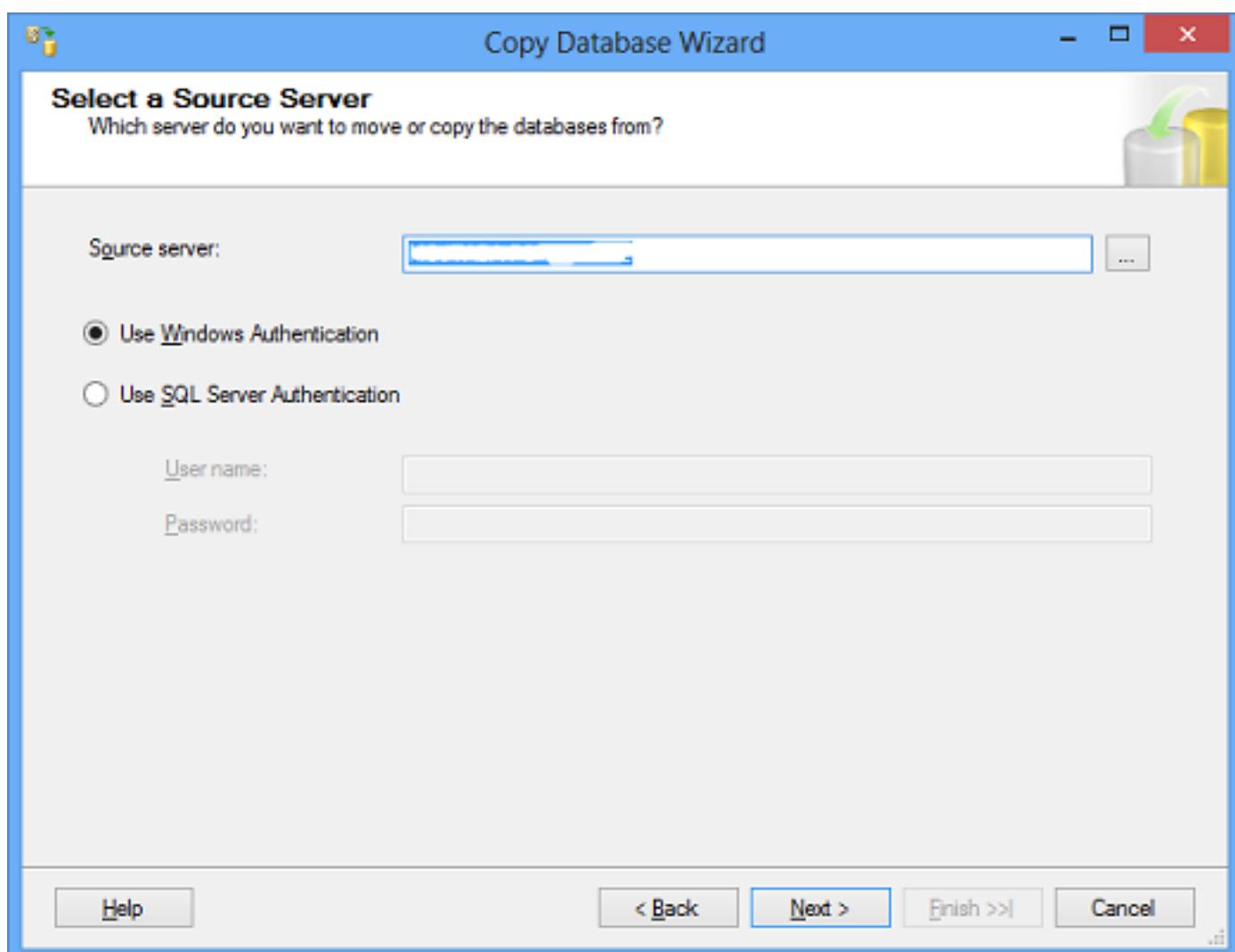
	name	rows	reserved	data	index_size	unused
1	Employees_sparse	50000	3080 KB	3064 KB	16 KB	0 KB

البته ذکر این نکته گفتی است که بهتر است از این تکنیک برای جداولی که تعداد زیادی ستون null دارند استفاده شود.

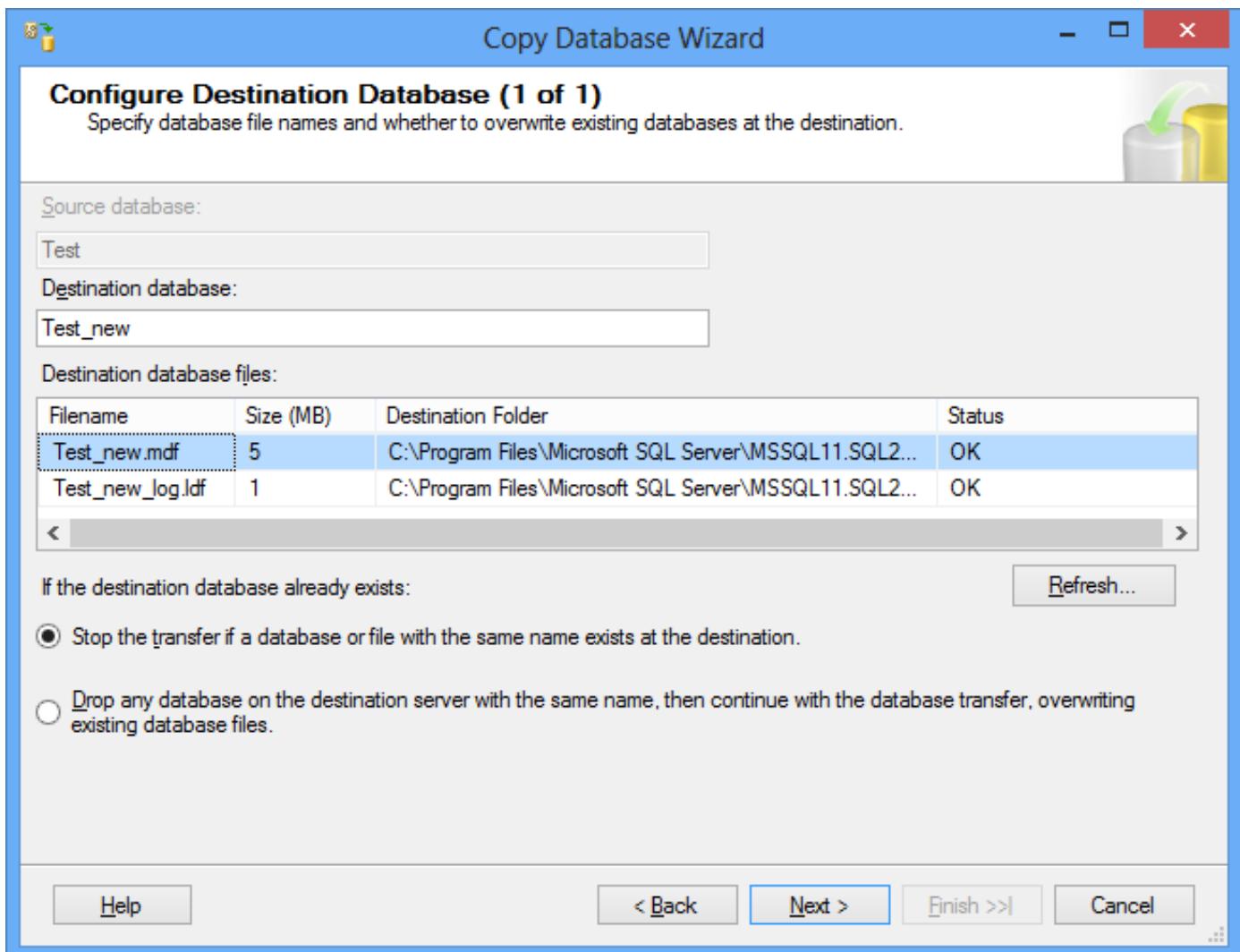
بعضی وقت‌ها به هر علتی لازم است پایگاه داده و فایل هایش را تغییر نام دهیم. اگر در اینترنت جستجو کنیم روش‌های مختلفی برای تغییر نام مثل تغییر با T-SQL Management Studio یا روشن‌های دیگری یافت می‌شود. اما اکثراً در بین انجام به مشکلی غیر قابل پیش‌بینی بر می‌خوریم. پایگاه داده در حالت آفلاین یا Pending قرار گرفته و به خطاهای نا مفهومی بر می‌خوریم. حالا باید دوباره کلی جستجو کنیم تا مشکل بوجود آمده را حل نماییم.

بهترین روش تغییر نام پایگاه داده

بهترین روش استفاده از قابلیت Copy Database خود SQL Server است است که به راحتی این کار را برای ما انجام می‌دهد. بر روی پایگاه داده مورد نظر راست کلیک کرده و از گزینه Tasks گزینه Copy Database را انتخاب کنید. پس از ظاهر شدن پنجره کپی گزینه Next را انتخاب و در مرحله مبدأ و مقصد، سرور جاری را انتخاب کنید و به مرحله بعد بروید. این برای زمانی است که شما می‌خواهید پایگاه داده را در سرور دیگری کپی نماید در پنجره Transfer Method دو روش Detach and Attach و وجود دارد که با همان روش اول به مرحله بعد بروید



در مرحله بعد نام پایگاه داده شما انتخاب شده به مرحله بعد بروید.
مرحله بعد پیکریندی پایگاه داده مقصد می باشد که نام و مسیر پایگاه داده جدید را می توانید مشخص نمایید.



این عملیات با SQL Server Agent صورت می پذیرد به همین خاطر Agent می باشد نصب و Start شده باشد.
با انتخاب گزینه Next مرحله بعد را رد کرده تا عملیات آغاز شود.
در مرحله آخر پایگاه داده قبلی را حذف نمایید.

نظرات خوانندگان

نویسنده: اردلان شاه قلی
تاریخ: ۱۳۹۲/۰۴/۰۶ ۱۱:۵۲

خیلی متشرکم.

البته یک سوال هم دارم. تصور کنید حجم پایگاه داده‌ی من 400 گیگابایت می‌باشد و کل فضای خالی موجود 350 گیگابایت می‌باشد در این صورت امکان استفاده از این روش وجود ندارد؟

Window Function ها برای اولین بار در نسخه SQL Server 2005 ارائه گردیدند، و در ورژن‌های جدیدتر SQL Server، به تعداد این فانکشنها افزوده شده است.

تعريف Window Function

معمولاً از این نوع فانکشنها روی مجموعه‌ای از ROW‌های یک جدول، در جهت اعمال عملیات‌های محاسباتی، ارزیابی داده‌ها، رتبه بندی و غیره... استفاده می‌گردد، به بیان ساده‌تر بواسیله Window Function ها می‌توان، ROW‌های یک جدول را گروه بندی نمود. و روی گروه‌ها از توابع جمعی (Aggregate Functions) استفاده کرد. این نوع فانکشنها از قابلیت و انعطاف پذیری زیادی برخوردار می‌باشند، و بواسیله آنها می‌توان نتایج (خروچی) بسیار مفیدی از Query‌ها، بدست آورد، معمولاً از این نوع فانکشنها در Data Mining (داده کاوی) و گزارشگیری‌ها استفاده می‌گردد. و آگاهی و روش استفاده از Window Function ها برای برنامه نویسان و DBA‌ها، می‌تواند بسیار مفید باشد.

مفهوم Window Function مطابق استاندارد ISO و ANSI می‌باشد، و دیتابیس‌هایی همچون Oracle, DB2, Sybase از آن پشتیبانی می‌نمایند. برای اطلاعات بیشتر می‌توانید به سایت‌های زیر مراجعه کنید: [SQL:2003](#) و [SQL:2008](#)

"Window" در Window Function، به مجموعه ROW‌های اشاره می‌کند، که محاسبات و ارزیابی و غیره... روی آنها اعمال می‌گردد.

Window Function ها برای ارائه قابلیت‌های خود، از Over Clause استفاده می‌کنند. اگر مقاله آشنایی با Row_Number, Rank, Dense_Rank, NTILE را مطالعه کرده باشید، می‌توان هریک از آنها را یک Window Function دانست. برای شروع، به بررسی Over Clause می‌پردازیم، و Syntax آن به شرح ذیل می‌باشد:

```

OVER (
    [ <PARTITION BY clause> ]
    [ <ORDER BY clause> ]
    [ <ROW or RANGE clause> ]
)
<PARTITION BY clause> ::= PARTITION BY value_expression , ... [ n ]
<ORDER BY clause> ::= ORDER BY order_by_expression
    [ COLLATE collation_name ]
    [ ASC | DESC ]
    [ ,...n ]
<ROW or RANGE clause> ::=
{ ROWS | RANGE } <window frame extent>
<window frame extent> ::=
{   <window frame preceding>
    | <window frame between>
}
<window frame between> ::=
BETWEEN <window frame bound> AND <window frame bound>
<window frame bound> ::=
{   <window frame preceding>
    | <window frame following>
}
<window frame preceding> ::=
{
    UNBOUNDED PRECEDING
    | <unsigned_value_specification> PRECEDING
    CURRENT ROW
}

```

```

<window frame following> ::= 
{
    UNBOUNDED FOLLOWING
    | <unsigned_value_specification> FOLLOWING
    CURRENT ROW
}

<unsigned value specification> ::= 
{ <unsigned integer literal> }

```

دارای سه آرگومان اختیاری است که هر کدام را به تفصیل بررسی می‌کنیم:
PARTITION BY clause 1-
بوسیله این پارامتر می‌توانیم Rowهای یک جدول را گروه بندی نماییم. این پارامتر یک Value_expression می‌پذیرد. یک Scalar Subquery ، Scalar Function و value_expression باشد.

ORDER BY clause 2-
از نامش مشخص است و برای Sort استفاده می‌شود، و ویژگی‌های Order By در آن اعمال می‌گردد. به جز .Offset

ROW or RANGE clause 3-
این پارامتر بیشتر برای محدود نمودن Row در یک Partition (گروه) مورد استفاده قرار می‌گیرد، به عنوان مثال نقطه شروع و پایان را می‌توان بوسیله پارامتر فوق تعیین نمود.
Row و Range نسبت به هم یک تفاوت عمده دارند، و آن این است که، اگر از ROW Clause استفاده نمایید، ارتباط Rowهای قبلی یا بعدی، نسبت به Row جاری، بصورت فیزیکی (physical association) سنجیده می‌شود، بطوریکه با استفاده از Range Clause ارتباط سطرهای قبلی و بعدی، نسبت به سطر جاری بصورت منطقی (logical association) در نظر گرفته می‌شود. ممکن است درک این مطلب کمی سخت باشد، در ادامه با مثالهایی که بررسی می‌نماییم، براحتی تفاوت این دو را متوجه می‌شوید.

Range یا Range در قالب‌های متفاوتی مقدار می‌پذیرند، که هر کدام را بررسی می‌کنیم:
UNBOUNDED PRECEDING : بیانگر اولین سطر Partition می‌باشد.
UNBOUNDED FOLLOWING : بیانگر آخرین سطر Partition می‌باشد.

CURRENT ROW : اولین سطر جاری یا آخرین سطر جاری را مشخص می‌نماید.

<unsigned value specification> PRECEDING n یا <unsigned value specification> PRECEDING : تعداد سطرهای قبل از سطر جاری را تعیین می‌کند، n یا <unsigned value specification> FOLLOWING n یا <unsigned value specification> FOLLOWING : تعداد سطرهای قابل توجه آن است که نقطه پایانی نمی‌توان برای Range استفاده نمود.

<unsigned value specification> FOLLOWING n یا <unsigned value specification> FOLLOWING : تعداد سطرهای بعد از سطر جاری را تعیین می‌کند، n یا <unsigned value specification> PRECEDING n یا <unsigned value specification> PRECEDING : تعداد سطرهای قابل توجه آن است که نقطه پایانی نمی‌توان برای Range استفاده نمود.

BETWEEN <window frame bound> AND <window frame bound> : از چارچوب فوق برای Range و Row می‌توان استفاده نمود، و نقطه آغازین و نقطه پایانی توسط قالب فوق تعیین می‌گردد. نکته قابل توجه آن است که نقطه پایانی نمی‌تواند، کوچکتر از نقطه آغازین گردد.

در ادامه برای درک هرچه بیشتر تعاریف بیان شده، چندین مثال می‌زنیم و هر کدام را بررسی می‌نماییم:
در ابتدا Script زیر را اجرا نمایید، که شامل جدولی به نام Revenue (سود، درآمد) و درج چند درکورد در آن:

```

CREATE TABLE REVENUE
(
[DepartmentID] int,
[Revenue] int,
[Year] int
);

insert into REVENUE
values (1,10030,1998),(2,20000,1998),(3,40000,1998),
(1,20000,1999),(2,60000,1999),(3,50000,1999),
(1,40000,2000),(2,40000,2000),(3,60000,2000),
(1,30000,2001),(2,30000,2001),(3,70000,2001)

```

مثال اول : می خواهیم براساس فیلد Revenue جدول DepartmentID را بندی نماییم و از توابع جمعی AVG و SUM روی فیلد درآمد(Revenue) استفاده کنیم.

ابتدا Script زیر را اجرا می کنیم:

```
select *,  
    avg(Revenue) OVER (PARTITION by DepartmentID) as AverageRevenue,  
    sum(Revenue) OVER (PARTITION by DepartmentID) as TotalRevenue  
from REVENUE  
order by departmentID, year;
```

خروجی بصورت زیر خواهد بود:

	DepartmentID	Revenue	Year	AverageRevenue	TotalRevenue
1	1	10030	1998	25007	100030
2	1	20000	1999	25007	100030
3	1	40000	2000	25007	100030
4	1	30000	2001	25007	100030
5	2	20000	1998	37500	150000
6	2	60000	1999	37500	150000
7	2	40000	2000	37500	150000
8	2	30000	2001	37500	150000
9	3	40000	1998	55000	220000
10	3	50000	1999	55000	220000
11	3	60000	2000	55000	220000
12	3	70000	2001	55000	220000

مطابق شکل، جدول براساس فیلد Revenue به سه Partition تقسیم شده است، و عملیات میانگین و جمع روی فیلد Revenue انجام شده است و عملیات Sort روی هرگروه بطور مستقل انجام گرفته است. چنین کاری را نمی توانستیم Group By انجام دهیم.

مثال دوم : نحوه استفاده از ROWS PRECEDING در این مثال قصد داریم عملیات جمع روی فیلد Revenue را روی فیلد Year انجام دهیم. بطوریکه جمع هر مقدار برابر است با سه مقدار قبلی + مقدار جاری: لطفاً رکوردهای زیر را به جدول فوق درج نمایید:

```
insert into REVENUE  
values(1,90000,2002),(2,20000,2002),(3,80000,2002),  
(1,10300,2003),(2,1000,2003), (3,90000,2003),  
(1,10000,2004),(2,10000,2004),(3,10000,2004),  
(1,20000,2005),(2,20000,2005),(3,20000,2005),  
(1,40000,2006),(2,30000,2006),(3,30000,2006),  
(1,70000,2007),(2,40000,2007),(3,40000,2007),  
(1,50000,2008),(2,50000,2008),(3,50000,2008),  
(1,20000,2009),(2,60000,2009),(3,60000,2009),  
(1,30000,2010),(2,70000,2010),(3,70000,2010),  
(1,80000,2011),(2,80000,2011),(3,80000,2011),  
(1,10000,2012),(2,90000,2012),(3,90000,2012)
```

سپس Script زیر را اجرا می نماییم:

```
select Year, DepartmentID, Revenue,
sum(Revenue) OVER (PARTITION by DepartmentID ORDER BY [YEAR]
ROWS BETWEEN 3 PRECEDING AND CURRENT ROW) as Prev3
From REVENUE order by departmentID, year;
```

خروجی :

	Year	DepartmentID	Revenue	Prev3
1	1998	1	10030	10030
2	1999	1	20000	30030
3	2000	1	40000	70030
4	2001	1	30000	100030
5	2002	1	90000	180000
6	2003	1	10300	170300
7	2004	1	10000	140300
8	2005	1	20000	130300
9	2006	1	40000	80300

در Script بالا، جدول را براساس فیلد DepartmentID گروه بندی می‌کنیم، که سه گروه ایجاد می‌شود، هر گروه را بطور مستقل، روی فیلد Year بصورت صعودی مرتب می‌نماییم. حال برای آنکه بتوانیم سیاست جمع، روی فیلد Revenue، را پیاده سازی نماییم، قطعه کد زیر را در Script بالا اضافه کردیم.

```
ROWS BETWEEN 3 PRECEDING AND CURRENT ROW) as Prev3
```

برای شرح چگونگی استفاده از PRECEDING، فقط به شرح گروه اول بسته می‌کنیم. مقدار جمع فیلد Revenue سطر اول، که قبل از آن سطری وجود ندارد، برابر است با مقدار خود، یعنی 10030، مقدار جمع فیلد Revenue سطر دوم برابر است با حاصل جمع مقدار فیلد Revenue سطر اول و دوم، یعنی 30030 . این روند تا سطر چهار ادامه دارد، اما برای بدست آوردن مقدار جمع فیلد Revenue سطر پنجم، مقدار جمع فیلد Revenue سطر دوم، سوم، چهارم و پنجم در نظر گرفته می‌شود، و مقدار فیلد Revenue اول در حاصل جمع در نظر گرفته نمی‌شود، بنابراین مقدار جمع فیلد Revenue سطر پنجم برابر است با 180000 . در صورت مسئله گفته بودیم، مقدار جمع فیلد Revenue هر سطر جاری برابر است با حاصل جمع مقدار سطر جاری و مقادیر سه سطر ماقبل خود.

مثال سوم: نحوه استفاده از ROWS FOLLOWING ، این مثال عکس مثال دوم است، یعنی حاصل جمع مقدار فیلد Revenue هر سطر برابر است با حاصل جمع سطر جاری با سه سطر بعد از خود. بنابراین Script زیر را اجرا نمایید:

```
select Year, DepartmentID, Revenue,
sum(Revenue) OVER (PARTITION by DepartmentID ORDER BY [YEAR]
ROWS BETWEEN CURRENT ROW AND 3 FOLLOWING) as Next3
From REVENUE order by departmentID, year;
```

خروجی :

	Year	DepartmentID	Revenue	Next3
1	1998	1	10030	100030
2	1999	1	20000	180000
3	2000	1	40000	170300
4	2001	1	30000	140300
5	2002	1	90000	130300
6	2003	1	10300	80300
7	2004	1	10000	140000
8	2005	1	20000	180000

مطابق شکل مقدار جمع فیلد اول برابر است با حاصل جمع مقدار سطر جاری و سه سطر بعد از آن.

نکته ای که در مثالهای دوم و سوم، می بایست به آن توجه نمود، این است که در زمان استفاده از Row یا Range یا Order by در الزامی است، در غیر این صورت با خطأ مواجه می شوید.

```

select Year, DepartmentID, Revenue,
       sum(Revenue) OVER (PARTITION by DepartmentID --ORDER BY [YEAR]
                           ROWS BETWEEN CURRENT ROW AND 3 FOLLOWING) as Next3
  From REVENUE order by departmentID, year;

```

Msg 10756, Level 15, State 1, Line 2
Window frame with ROWS or RANGE must have an ORDER BY clause.

نحوه استفاده از UNBOUNDED PRECEDING ، این امکان در T-SQL Server 2012 افزوده شده است.

مثال چهار: در این مثال می خواهیم کمترین سود بدست آمده در چند سال را بدست آوریم:
ابتدا Script زیر را اجرا نمایید:

```

select Year, DepartmentID, Revenue,
       min(Revenue) OVER (PARTITION by DepartmentID ORDER BY [YEAR]
                           ROWS UNBOUNDED PRECEDING) as MinRevenueToDate
  From REVENUE order by departmentID, year;

```

خروجی:

	Year	DepartmentID	Revenue	MinRevenueToDate
1	1998	1	10030	10030
2	1999	1	20000	10030
3	2000	1	40000	10030
4	2001	1	30000	10030
5	2002	1	90000	10030
6	2004	1	10000	10000
7	2005	1	20000	10000
8	2006	1	40000	10000
9	2007	1	70000	10000
10	2008	1	50000	10000
11	2009	1	20000	10000
12	2010	1	30000	10000
13	2011	1	80000	10000
14	2012	1	10000	10000
15	1998	2	20000	20000
16	1999	2	60000	20000

طبق تعریف UNBOUNDED PRECEDING اولین سطر هر Partition را مشخص می‌نماید، و چون از PRECEDING استفاده کرده ایم، بنابراین مقایسه همیشه بین سطر جاری و سطرهای قبل از آن انجام می‌پذیرد. بنابراین خواهیم داشت، کمترین مقدار فیلد Revenue در سطر اول، برابر با مقدار خود می‌باشد، چون هیچ سط्रی ماقبل از آن وجود ندارد. در سطر دوم مقایسه کمترین مقدار، بین 20000 و 10030 انجام می‌گیرد، که برابر است با 10030، در سطر سوم، مقایسه بین مقادیر سطر اول، دوم و سطر سوم صورت می‌گیرد، یعنی کمترین مقدار بین 40000، 20000 و 10030، بنابراین کمترین مقدار سطر سوم برابر است با 10030. به بیان ساده‌تر برای بدست آوردن کمترین مقدار هر سطر، مقدار سطر جاری با مقادیر همه سطرهای ماقبل خود مقایسه می‌گردد.

برای بدست آوردن کمترین مقدار در سطر ششم، مقایسه بین مقادیر سطرهای اول، دوم، سوم، چهارم، پنجم و ششم صورت می‌گیرد که عدد 10000 بدست می‌آید و الى آخر... نکنه: اگر در Order by Over Clause را اعمال نماییم، اما از Row يا Range استفاده نکنیم، SQL Server بصورت پیش فرض از قالب زیر استفاده می‌نماید:

RANGE UNBOUNDED PRECEDING AND CURRENT ROW

برای روشن‌تر شدن مطلب فوق مثالی می‌زنیم:

ابتدا Script زیر را اجرا نمایید، که شامل ایجاد یک جدول و درج چند رکورد در آن می‌باشد:

```

CREATE TABLE Employees (
    EmployeeId INT IDENTITY PRIMARY KEY,
    Name VARCHAR(50),
    HireDate DATE NOT NULL,
    Salary INT NOT NULL
)
GO
INSERT INTO Employees (Name, HireDate, Salary)
VALUES

```

آشنایی با Window Function بخش اول SQL Server ها در

```
('Alice', '2011-01-01', 20000),
('Brent', '2011-01-15', 19000),
('Carlos', '2011-02-01', 22000),
('Donna', '2011-03-01', 25000),
('Evan', '2011-04-01', 18500)
```

GO

سپس زیر را اجرا نمایید:

```
SELECT
    Name,
    Salary,
    AVG(Salary) OVER(ORDER BY HireDate) AS avgSalary
FROM Employees
GO
```

خروجی :

	Name	Salary	avgSalary
1	Alice	20000	20000
2	Brent	19000	19500
3	Carlos	22000	20333
4	Donna	25000	21500
5	Evan	18500	20900

حال اگر Script زیر را نیز اجرا نمایید، خروجی آن مطابق شکل بالا خواهد بود:

```
SELECT
    Name,
    Salary,
    AVG(Salary) OVER(ORDER BY HireDate
                      RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS avgSalary
FROM Employees
GO
```

توضیح درباره Script بالا، در این روش برای بدست آوردن میانگین هر سطر، مقدار سطر جاری با مقادیر سطرهای ماقبل خود جمع و تقسیم بر تعداد سطر می‌شود.

سطر دوم $19000 + 20000 = 19500$ بر دو برابر است با 19500

میانگین سطر پنجم، حاصل جمع فیلد Salary همه مقادیر سطرهای تقسیم بر 5

*** اگر بخواهید بوسیله Over Clause ، میانگین همه سطرهای یکسان باشد می‌توانید از Script زیر استفاده نمایید:

```
SELECT
    Name,
    Salary,
    AVG(Salary) OVER(ORDER BY HireDate
                      RANGE
                      BETWEEN UNBOUNDED PRECEDING
                      AND UNBOUNDED FOLLOWING
                  ) AS avgSalary
FROM Employees
GO
```

خروجی :

	Name	Salary	avgSalary
1	Alice	20000	20900
2	Brent	19000	20900
3	Carlos	22000	20900
4	Donna	25000	20900
5	Evan	18500	20900

منظور از ROWS BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING یعنی در محاسبه میانگین برای هر سطر تمامی مقادیر سطرهای دیگر در نظر گرفته شود.
پایان بخش اول
امیدوارم مفید واقع شده باشد.

نظرات خوانندگان

نویسنده: ناصر
تاریخ: ۱۳۹۱/۰۹/۱۸

بسیار عالی. فقط خواستم تشکری کرده باشم :

نویسنده: reza
تاریخ: ۱۳۹۱/۰۹/۱۸

The screenshot shows a SQL Server Management Studio window. The top tab bar has three tabs: 'SQLQuery2.sql - (I...Er-PC\na3Er (51))', 'SQLQuery1.sql - (I...Er-PC\na3Er (54))', and 'NA3ER-PC.learn - dbo.REVENUE'. The main query window contains the following T-SQL code:

```
select Year, DepartmentID, Revenue,
sum(Revenue) OVER (PARTITION by DepartmentID ORDER BY [YEAR]
ROWS BETWEEN 3 PRECEDING AND CURRENT ROW) as Prev3
From REVENUE order by departmentID, year;
```

In the 'Messages' pane at the bottom, there is an error message:

```
Msg 102, Level 15, State 1, Line 3
Incorrect syntax near 'ROWS'.
```

علت این خطأ چیست؟ آیا این دستور مال 2012 هستش؟ جون من تو 2008 تست میکنم

نویسنده: فرهاد فرهمند خواه
تاریخ: ۱۳۹۱/۰۹/۱۸

سلام

Syntax ارائه شده Over Clause در SQL Server 2008 شامل دو پارامتر است :

```
Ranking Window Functions
<OVER_CLAUSE> ::= =
    OVER ( [ PARTITION BY value_expression , ... [ n ] ]
        <ORDER_BY_Clause> )
```

```
Aggregate Window Functions
<OVER_CLAUSE> ::= =
    OVER ( [ PARTITION BY value_expression , ... [ n ] ] )
```

در SQL Server 2012 پارامتر `<ROW or RANGE clause>` فوق افزوده شد. بنابراین، Query هایی که از Row یا

استفاده کرده اند، در SQL Server 2008 با خطا مواجه می شوند.

نوبسند: معتمدی
تاریخ: ۱۳۹۱/۰۹/۱۸ ۱۲:۵۹

سلام

آیا استفاده از این query ها برای محیط های transactional هم مناسب است؟ یا بیشتر در database های آماری جهت تهیه گزارشات کاربرد دارد؟

از نظر زمان و هزینه ای اجرا و نیز تخصیص منابع سرور به آنها می پرسم.

نوبسند: فرهاد فرهمند خواه
تاریخ: ۱۳۹۱/۰۹/۱۸ ۱۴:۵۷

سلام

سرعت و قابلیت اجرایی Over Clause به مراتب از Group by بهتر است. بطوریکه اگر یک عملیات یکسان را، بطور جداگانه، هم با و هم با Group By انجام دهید. و در Execution Plan مشاهده نمایید، تفاوت را حس خواهید نمود. سایت زیر یک مثال ساده در این رابطه قرار داده است:

در مورد اینکه برای محیط های Transactional مناسب است یا نه، عوامل زیادی در آن دخیل است و بسته به حجم داده ای مورد انتظار شما در خروجی دارد، بطور مثال اگر بخواهید یک گزارش 400 صفحه ای ایجاد نمایید، بطور حتم در چنین محیط هایی هیچ Scriptی مناسب، نیست، اما بطور قاطع می توان گفت که Window Function ها از کارایی بسیار خوبی برخوردار هستند، و انجام عملیات های پیچیده محاسباتی را برای ما آسانتر نموده اند.

نوبسند: فاطمه
تاریخ: ۱۳۹۱/۱۰/۲۳ ۱۷:۲۲

بسیار عالی بود
با تشکر

نوبسند: zarei
تاریخ: ۱۳۹۲/۰۲/۲۱ ۲۲:۱۴

سلام

ممnon از آموزش مفیدتون . سوال من اینه که اگه بخواهیم رکوردهای تکراری حذف بشن باید چیکار کنیم ؟ مثلا من میخام مجموع مبلغ بدھکار برای یک کد کل و در یک سند را در یک ردیف و همین مورد برای مجموع مبلغ بستانکار را نیز در یک رکورد یا ردیف دیگر بدھد . در صورتی که اگر از توابع Sum() استفاده کنیم به ازای هر کد کل درآن سند یک رکورد در خروجی داریم (چه بدھکار و چه بستانکار)

نوبسند: فرهاد فرهمند خواه
تاریخ: ۱۳۹۲/۰۲/۲۲ ۸:۱۹

سلام

اگر سؤال شما را درست متوجه شده باشم، فکر می کنم، می بایست از Pivot استفاده نمایید، مقاله زیر می تواند در درگ Pivot به شما کمک نماید.

موفق باشید.

نویسنده: محمد شهریاری
تاریخ: ۹:۰ ۱۳۹۳/۰۲/۲۱

با سلام
آیا امکان استفاده از `scalar function` ها هم هست ؟

ممnon

نویسنده: فرهاد
تاریخ: ۱۵:۱۶ ۱۳۹۳/۰۲/۲۲

سلام
عموماً از توابع Aggregate Functions می‌توان در Window Function استفاده نمود: [^](#)

فرض کنید که می‌خواهیم خروجی از جدول خود را به صورت XML نمایش یا از طریق وب سرویس در برنامه مان استفاده نماییم. اولین راهی که به ذهنمان می‌رسد خودمان رشته XML را با حلقه‌ای ایجاد نماید یا استفاده از فضای نام System.Xml و کلاس‌های نوشته شده برای این کار. اما خود SQL Server امکانات ویژه‌ای برای کار با ساختار XML مهیا نموده که براحتی می‌توانید خروجی XML از داده‌هایتان ایجاد نمایید.

برای این کار از عبارت [For XML](#) در Select می‌توان استفاده نمود. برای مثال برای بدست آوردن ساختار ساده از Auto استفاده نمایید

```
SELECT BusinessEntityID, PersonType, Title, FirstName, MiddleName, LastName
FROM Person
WHERE BusinessEntityID = 10001
FOR XML AUTO
```

که خروجی بصورت node attribute زیر می‌باشد:

```
<Person.Person BusinessEntityID="10001" PersonType="IN" FirstName="Carolyn" LastName="Alonso" />
```

اما اگر بخواهیم خروجی به صورت Elements باشد کافیست از پارامتر Elements استفاده نمایید

```
SELECT BusinessEntityID, PersonType, Title, FirstName, MiddleName, LastName
FROM Person
WHERE BusinessEntityID = 10001
FOR XML AUTO, ELEMENTS
```

خروجی بصورت زیر می‌باشد:

```
[-] <Person>
  <BusinessEntityID>10001</BusinessEntityID>
  <PersonType>IN</PersonType>
  <FirstName>Carolyn</FirstName>
  <LastName>Alonso</LastName>
</Person>
```

اگر بخواهیم elements و attributes را هم ترکیب کنیم بصورت زیر عمل می‌کنیم:

```
SELECT BusinessEntityID AS '@ID', PersonType, Title, FirstName, MiddleName, LastName
FROM Person
WHERE BusinessEntityID = 10001
FOR XML ELEMENTS
```

خروجی بصورت زیر است:

```
<Person ID="10001">
  <PersonType>IN</PersonType>
  <FirstName>Carolyn</FirstName>
  <LastName>Alonso</LastName>
</Person>
```

حال می‌خواهیم همه node‌ها را یک ریشه قرار دهیم برای این کار از پارامتر ROOT در کنار AUTO به صورت زیر استفاده نمایید:

```
SELECT *
FROM Person
WHERE BusinessEntityID = 15291
FOR XML AUTO , ROOT('Persons')
```

اما اگر بخواهیم نام جدول را با نام دلخواه خود تغییر دهیم از پارامتر PATH به جای AUTO به صورت زیر استفاده نمایید:

```
SELECT *
FROM Person
WHERE BusinessEntityID = 15291
FOR XML PATH('P') , ROOT('Persons')
```

نظرات خوانندگان

نویسنده: احسان میرسعیدی
تاریخ: ۰۲۶ ۱۳۹۱/۰۹/۱۹

بسیار تکنیک کارامدی بود. واقعاً متشرکرم

نویسنده: فرهاد فرهمندخواه
تاریخ: ۷:۲۹ ۱۳۹۱/۰۹/۱۹

سلام
مرسی از مطلب مفید تان

نویسنده: بهزاد
تاریخ: ۱۱:۸ ۱۳۹۱/۰۹/۱۹

بسیار عالی بود ، خیلی برای مفید واقع شد
راه مشابه ای برای خروجی JSON نیست؟
ممnon

نویسنده: مجتبی کاویانی
تاریخ: ۱۱:۵۷ ۱۳۹۱/۰۹/۱۹

ممnon . هنوز به صورت native نه اما از تیم microsoft در این [لينك](#) خواسته شده که For Json را هم اضافه کند.

نویسنده: وحید نصیری
تاریخ: ۱۱:۵۹ ۱۳۹۱/۰۹/۱۹

« [استفاده از JSON در SQL Server](#) »

نویسنده: مهدی ناظم السادات
تاریخ: ۱۸:۵۸ ۱۳۹۲/۰۹/۱۷

حالا دستور Insert نداریم که بشه یه فایل xml رو مثل فایل backup روی دیتابیس restore کنیم؟

نویسنده: محسن خان
تاریخ: ۰:۵۹ ۱۳۹۲/۰۹/۱۸

- می‌تونی با کدنویسی اینکار رو انجام بدی:

```
var reportData = new DataSet();
reportData.ReadXml("yourfile.xml");
var connection = new SqlConnection("DB ConnectionString");
var sbc = new SqlBulkCopy(connection);
sbc.DestinationTableName = "yourXMLTable";
```

- می‌تونی از [export](#) و [import](#) خود SQL Server استفاده کنی.
- و یا از [OPENXML](#) میشه استفاده کرد:

```
INSERT Customers
SELECT *
FROM OPENXML ...
```

نوبنده: ناصرنیازی
تاریخ: ۲۱:۵۱ ۱۳۹۲/۱۱/۲۹

تشکر فراوان از این مطلب فوق العاده کاربردی. یه نکته کوچک به ذهنم رسید. سومین قطعه کدی که نوشتشد در جدول من کار نکرد به نظرم او مد که شاید اینچنین بوده باشه :

کد شما

FOR XML ELEMENTS

کدی که من مد نظرم هست و در جدول من کار می کنه :

for xml auto, ELEMENTS

قبل از مطالعه این بخش لطفا آشنایی با [Window Function ها در SQL Server](#) بخش اول را مطالعه نمایید.

در [بخش اول](#)، در مورد Syntax مربوط به Over Clause صحبت کردیم، و برای درک استفاده از Over Clause، مثالهایی را بررسی نمودیم، در این بخش نیز، به تفاوت Row Clause و Range Clause می پردازیم.

مثال: با ایجاد یک Script، عملیات جمع روی یک فیلد خاص، بوسیله Row Clause و Range Clause انجام می‌دهیم. تفاوت آنها را درک نماییم.

در ادامه Script زیر را اجرا نمایید:

```
DECLARE @Test TABLE
(
RowID INT IDENTITY,
FName VARCHAR(20),
Salary SMALLINT
);
INSERT INTO @Test (FName, Salary)
VALUES ('George', 800),
('Sam', 950),
('Diane', 1100),
('Nicholas', 1250),
('Samuel', 1250),
('Patricia', 1300),
('Brian', 3000),
('Thomas', 1600),
('Fran', 2450),
('Debbie', 2850),
('Mark', 2975),
('James', 3000),
('Cynthia', 3000),
('Christopher', 5000);

SELECT RowID,FName,Salary,
      SumByRows = SUM(Salary) OVER (ORDER BY Salary ROWS UNBOUNDED PRECEDING),
      SumByRange = SUM(Salary) OVER (ORDER BY Salary RANGE UNBOUNDED PRECEDING)
FROM @Test
ORDER BY RowID;
```

خروجی بصورت زیر خواهد بود:

	RowID	FName	Salary	SumByRows	SumByRange
1	1	George	800	800	800
2	2	Sam	950	1750	1750
3	3	Diane	1100	2850	2850
4	4	Nicholas	1250	4100	5350
5	5	Samuel	1250	5350	5350
6	6	Patricia	1300	6650	6650
7	7	Brian	1500	8150	8150
8	8	Thomas	1600	9750	9750
9	9	Fran	2450	12200	12200
10	10	Debbie	2850	15050	15050
11	11	Mark	2975	18025	18025
12	12	James	3000	21025	24025
13	13	Cynthia	3000	24025	24025
14	14	Christopher	5000	29025	29025

با مشاهده شکل بالا، به وضوح می‌توان تفاوت Row و Range را تشخیص داد. در Script بالا از UNBOUNDED PRECEDING استفاده کردیم ، و مفهوم قالب آن به شرح ذیل می‌باشد:

مقدار فیلد Salary سطر جاری = جمع مقادیر فیلد Salary همه سطرهای ماقبل، سطر جاری + مقدار فیلد Salary سطر جاری Row Clause بصورت فیزیکی به سطرهای می‌نگرد و قالب بیان شده در Script را، روی تمامی سطرهای نسبت به جایگاه آنها در جدول، به ترتیب اعمال می‌نماید. و در شکل نیز قابل مشاهده می‌باشد، یعنی به چیدمان سطرهای در خروجی که بصورت فیزیکی نمایش داده شده است، توجه می‌کند، و حاصل جمع هر سطر برابر است با حاصل جمع سطرهای ماقبل + سطر جاری اما Range Clause: به چیدمان فیزیکی سطرهای توجه نمی‌کند، بلکه بصورت منطقی به مقدار فیلد Salary سطرهای توجه می‌نماید، یعنی مقادیری که در یک محدوده (Range) قرار دارند، حاصل جمع آنها، یکی است.

مقدار فیلد Salary سطر چهار و پنج برابر است با 1250 بنابراین حاصل جمع آنها برابر هم می‌باشد. و بصورت زیر محاسبه می‌شود:

$$5350 = 1250 + 1250 + 1100 + 950 + 800$$

روش بیان شده، در مورد سطرهای 12 و 13 نیز صادق است.

امیدوارم با مثالهایی که در بخش اول و بخش دوم بررسی نمودیم، روش استفاده از Over Clause را درک کرده باشیم. Window Function ها را به چهار بخش تقسیم بندی شده اند، که به شرح ذیل می‌باشد:

(توابع رتبه بندی)، که بررسی نمودیم. [Ranking functions](#) 1-

، که در بحث ایجاد Sequence آن را بررسی نمودیم. [NEXT VALUE FOR](#) 2-

(توابع جمعی)، اکثرًا با اینگونه توابع آشنایی هستیم. [Aggregate Functions](#) 3-

(توابع تحلیلی) که در بخش بعدی آن را بررسی می‌نماییم. [Analytic Functions](#) 4-

یکی از منابع بسیار مفید در مورد Window Function ها کتاب [Microsoft SQL Server 2012 High-Performance T-SQL Using Window Functions](#)، می‌باشد، که بطور کامل به Window Function ها اختصاص دارد و تکنیک‌های بسیار مفیدی را بیان می‌کند. مطالعه آن به علاوه مدنظر، پیشنهاد می‌گردد. موفق باشید.

نظرات خوانندگان

نویسنده: محمد صاحب
تاریخ: ۱۳۹۱/۰۹/۱۹ ۱۲:۲۱

دوست عزیز ممنون...

من قسمت Row و Range را که شما توضیح دادی درست متوجه نشدم سرچی که زدم متوجه شدم این قابلیت تقریباً شبیه قسمت WITH TIES است. برای مثال اگه بخواهیم 3 شاگرد برتر کلاس را کوئری بزنیم اگه تو کلاس 3 نفر معدل 18 داشته باشن (با توجه به اینکه یک معدل 20 و 19 داریم) 2 نفر از شاگردها که معدل 18 دارن تو این کوئری نمیان (TOP 3) و ... برداشت منم از Range اینه که بواسطه این 2 مقدار به هم گره خوردن و هنگام محاسبه مقدار یکسانی را تولید میکن.

نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۳۹۱/۰۹/۱۹ ۱۴:۲۱

سلام

اگر سؤال شما رو درست متوجه شده باشم، با یک مثال مفهوم Range رو بررسی می‌کنیم:

```
CREATE TABLE #Transactions
(
    AccountId INTEGER,
    TranDate DATE,
    TranAmt NUMERIC(8, 2)
);
INSERT INTO #Transactions
SELECT *
FROM ( VALUES ( 1, '2011-01-15', 50 ), ( 1, '2011-01-17', 500 ), ( 1, '2011-01-17', 500 ),
        ( 1, '2011-01-16', 500 ), ( 1, '2011-01-24', 75 ), ( 1, '2011-01-26', 125 ),
        ( 1, '2011-02-28', 500 ), ( 2, '2011-01-01', 500 ), ( 2, '2011-01-15', 50 ),
        ( 2, '2011-01-22', 25 ), ( 2, '2011-01-23', 125 ), ( 2, '2011-01-26', 200 ),
        ( 2, '2011-01-29', 250 ), ( 3, '2011-01-01', 500 ), ( 3, '2011-01-15', 50 ),
        ( 3, '2011-01-22', 5000 ), ( 3, '2011-01-25', 550 ), ( 3, '2011-01-27', 95 ),
        ( 3, '2011-01-30', 2500 )
) dt (AccountId, TranDate, TranAmt);
```

روی جدول فوق دو نوع Script اجرا می‌کنیم، مثال اول، براساس AccountID جدول را گروه بندی می‌نماییم. سپس هر گروه را براساس تاریخ Sort می‌کنیم، و در هر گروه مقدار Sum آن را بدست می‌آوریم:

```
SELECT
    AccountId,
    TranDate,
    TranAmt,
    Sum(TranAmt) OVER(partition by AccountId ORDER BY TranDate RANGE UNBOUNDED PRECEDING) AS SumAmt
FROM #Transactions
GO
```

: خروجی

	AccountId	TranDate	TranAmt	SumAmt
1	1	2011-01-15	50.00	50.00
2	1	2011-01-16	500.00	550.00
3	1	2011-01-17	500.00	1550.00
4	1	2011-01-17	500.00	1550.00
5	1	2011-01-24	75.00	1625.00
6	1	2011-01-26	125.00	1750.00
7	1	2011-02-28	500.00	2250.00

مطابق شکل Sort براساس TranDate است، که چهار مقدار 500 در سه بازه تاریخی دیده می‌شود، حال محاسبه جمع هر سطر بصورت زیر است:

سطر دوم با وجود اینکه مقدار آن 500 است و در بازه تاریخی 16-01-2011 قرار دارد: مقدار آن برابر است با $500 + 50 = 550$

سطر سوم و چهارم که در بازه تاریخی 17-01-2011 می‌باشد(به عبارتی در یک محدوده می‌باشند): برابر است با:
 $500 + 500 + 50 = 1550$

در اینجا چیزی حذف نشده، حاصل جمع سطر سوم و چهارم ، چون در یک محدوده (Range) می‌باشد، برابر است با حاصل جمع سطرهای ما قبل یعنی سطر اول و دوم ($500 + 50 = 550$) + حاصل جمع تمامی سطرهای آن محدوده (Range)، یعنی سطر سوم و چهارم ($500 + 500 = 1000$)

مثال دیگر، در این حالت Sort روی فیلد TranAMT انجام می‌شود، و جدول همچنان روی فیلد AccountId گروه بندی می‌شود، بنابراین خواهیم داشت:

```
SELECT
    AccountId,
    TranDate,
    TranAmt,
    Sum(TranAmt) OVER(partition by AccountId ORDER BY TranAmt RANGE UNBOUNDED PRECEDING) AS SumAmt
FROM #Transactions
GO
```

خروجی :

	AccountId	TranDate	TranAmt	SumAmt
1	1	2011-01-15	50.00	50.00
2	1	2011-01-24	75.00	125.00
3	1	2011-01-26	125.00	250.00
4	1	2011-02-28	500.00	2250.00
5	1	2011-01-17	500.00	2250.00
6	1	2011-01-17	500.00	2250.00
7	1	2011-01-16	500.00	2250.00
8	2	2011-01-22	25.00	25.00
9	2	2011-01-15	50.00	75.00
10	2	2011-01-23	125.00	200.00

در شکل، مقدار جمع هیچ سطrix حذف نشده است، و Top ای هم در کار نیست.
حال اگر مثال فوق را روی میانگین در نظر بگیرید، باز هم تمام مقادیر، در محاسبه میانگین تاثیر گذار میباشند.

نویسنده: محمد صاحب
تاریخ: ۱۴:۵۹ ۱۳۹۱/۰۹/۱۹

ممnon...

«سطرهای فیزیکی» و «سطرهای منطقی» که شما گفتید برا من گیج کننده بود من با مثال Top ای که زدم فرق رو متوجه شدم گفتم عنوان کنم دیگران هم استفاده کنن. البته این پست شما خیلی بهتر موضوع رو توضیح داد.

نویسنده: محمد سلم ابادی
تاریخ: ۱۲:۱۱ ۱۳۹۱/۱۰/۳۰

سلام،

ممnon از مطالب مفیدتون.

آیا دو دستور زیر با هم یکسان هستند یا خیر؟

range unbounded preceding
range between unbounded preceding and current row

و کوئری اولتون باید مساله running total باشه، که به سادگی توسط Over Clause حل شده.
کوئری زیر روشی بوده که قبل از نسخه 2012 برای حل اینگونه مسائل مورد استفاده قرار میگرفته

```
SELECT AccountId,
       TranDate,
       TranAmt,
       D.sumAmt AS older_method
    Sum(TranAmt) OVER(partition by AccountId
                      ORDER BY TranDate
                      RANGE UNBOUNDED PRECEDING) AS SumAmt
  FROM #Transactions AS T1
CROSS APPLY (SELECT SUM(T2.TranAmt)
              FROM #Transactions AS T2
             WHERE T2.AccountId = T1.AccountId
               AND T2.TranDate <= T1.TranDate) AS D(SumAmt)
 ORDER BY T1.AccountId, T1.TranDate;
```

نویسنده: فرهاد فرهمندخواه
تاریخ: ۱۳:۲۵ ۱۳۹۱/۱۰/۳۰

سلام

در جواب سؤال شما باید بگوییم هر دو دستور یکی میباشند و هر دو از اولین سطر تا سطر جاری را در نظر میگیرند.

نویسنده: zarei
تاریخ: ۱۹:۳ ۱۳۹۲/۰۲/۲۲

سلام

من یه کوئری توسط () Over() .. نوشتم که تو در تو هست که ترتیب جمع دستور بیرونی برام مهمه .

```
;WITH cteBed ([Counter], id_doc , [Year] ,id_Total , date_duc ,Number_Temp , number_fix , sumbed ,
sumbes , row_no ) AS (
SELECT [Counter], d.id_doc , d.[Year] ,r.id_Total , d.date_duc ,d.Number_Temp ,d.number_fix ,
SUM( r.Mablagh_bed) OVER(PARTITION BY d.[Year] ,r.id_Total , d.Number_Temp) AS sumbed ,
sumbes= 0,
```

```

ROW_NUMBER() OVER (PARTITION BY d.[Year] ,r.id_Total , d.date_duc , d.Number_Temp , d.number_fix ORDER
BY d.date_duc )AS row_no
FROM tbl_Records r JOIN tbl_Documents d ON d.id_doc = r.id_doc ) ,
cteBes ([Counter], id_doc , [Year] ,id_Total , date_duc ,Number_Temp , number_fix , sumbed , sumbes ,
row_no) AS (
SELECT [Counter], d.id_doc , d.[Year] ,r.id_Total , d.date_duc ,d.Number_Temp ,d.number_fix , sumbed
= 0 ,
SUM( r.Mablagh_bes ) OVER(PARTITION BY d.[Year] ,r.id_Total , d.Number_Temp ) AS sumbes,
ROW_NUMBER() OVER (PARTITION BY d.[Year] ,r.id_Total , d.date_duc ,d.Number_Temp , d.number_fix ORDER
BY d.date_duc )AS row_no
FROM tbl_Records r JOIN tbl_Documents d ON d.id_doc = r.id_doc )
SELECT [Counter], id_doc , [Year] ,id_Total , date_duc ,Number_Temp , number_fix , sumbed , sumbes ,
amountBed ,amountBes
,SUM(amountBed)OVER( ORDER BY [Year] ,id_Total , date_duc , number_Temp, number_Fix ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW ) AS bed
,SUM(amountBes)OVER( ORDER BY [Year] ,id_Total , date_duc , number_Temp, number_Fix ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW ) AS bes
FROM (
SELECT [Counter], id_doc , [Year] ,id_Total , date_duc ,Number_Temp , number_fix , sumbed , sumbes ,
amountBed = CASE WHEN id_Total LIKE '1%' OR Id_Total LIKE '2%' OR Id_Total LIKE '7%' OR Id_Total
LIKE '8%' THEN (tt.sumbed-tt.sumbes) ELSE 0 END ,
amountBes=CASE WHEN Id_Total LIKE '3%' OR Id_Total LIKE '4%' OR Id_Total LIKE '5%' OR Id_Total LIKE
'6%' OR Id_Total LIKE '9%' THEN (tt.sumbes-tt.sumbed)ELSE 0 END ,
ROW_NUMBER() OVER (PARTITION BY [Year] ,id_Total , date_duc , Number_Temp , number_fix ORDER BY
date_duc )AS row_no
FROM (
SELECT * FROM cteBed cb WHERE cb.row_no = 1
UNION ALL
SELECT * FROM cteBes cb WHERE cb.row_no = 1
) AS tt ([Counter], id_doc , [Year] ,id_Total , date_duc ,Number_Temp , number_fix , sumbed ,
sumbes, row_no ) WHERE not(sumbed = 0 AND sumbes = 0)
) AS rr

```

اگه تو یه دستور Select از Row_Number استفاده کرده باشم ، اول خروجی رو بدست میاره بعد خروجی رو بر حسب نوع مرتب سازی مربوط به Row_Number مرتب میکنه ؟ و دیگه اینکه خروجی دستور اول که شامل Row_Number هست بعد از مرتب شدن به همون صورت به دست دستور دوم (یا همون Select بیرونی) میرسه یا باز باید روی اون نیز مرتب سازی انجام بدم ؟ اصلا جای ستونی که مربوط به Row_Number هست اول یا آخر فرق میکنه ؟
اینارو به این خاطر پرسیدم ، چون هر بار داده هام جواب متفاوتی میداد و نتونستم تشخیص بدم . ممنون

نویسنده: فرهاد فرهمند خواه
تاریخ: ۰۲/۲۵/۱۳۹۲

سلام

جواب سوال اول: در Syntax Row_Number تابع order by اجباری است، بنابراین عملیات سورت در ابتدا انجام می‌شود و سپس Row_Number (اعداد ترتیبی) روی رکوردها اعمال می‌گردد.
در سایت مایکروسافت به خوبی اشاره شده است که هیچ تضمینی وجود ندارد، خروجی یک Query با استفاده از Row_number در هر بار اجرا، با اجرای قبلی یکی باشد مگر آنکه موارد زیر را رعایت کرده باشید:
1- مقادیر ستونی که برای قسمت Partition در نظر گرفته اید، منحصر بفرد باشد.
2- مقادیری که برای قسمت Order by در نظر گرفته اید منحصر بفرد باشد.
3- ترکیب مقادیر Partition و Order by نیز مقدار منحصر بفردی را ایجاد نماید.

جواب سوال دوم: جای ستون Row_number در زمان نمایش اهمیتی ندارد.
پیشنهاد دوستانه:
1- تا جایی که امکان دارد از OR در Query های خود استفاده ننمایید، باعث افزایش زمان اجرای Query شما می‌شود و هزینه بالایی دارد.
2- از Like نیز در نوشتن Query های خود اجتناب کنید.
برای اطلاعات بیشتر در مورد Row_Number به آدرس زیر مراجعه نمایید: [Row_Number\(\)](#)

موفق باشید.

نویسنده: zarei

تاریخ: ۱۳۹۲/۰۲/۲۷ ۱:۳۸

ممنون از راهنماییتون . ولی برای رسیدن به پاسخ راه دیگه ای به ذهنم نرسید (استفاده از OR و Like) میشه خواهش کنم راه جایگزین رو بهم بگید ؟ اموزشی در خصوص بررسی مفهومی Plan های تولیدی SQL سراغ دارید ؟ دیگه اینکه از کجا میتونم به طور دقیق و مفهومی هزینه استفاده از دستورات SQL را مثل OR و ... رو بخونم ؟ ممنون

نویسنده: فرهاد فرهمندخواه

تاریخ: ۱۳۹۲/۰۲/۲۷ ۲۲:۱۹

سلام

مقاله زیر به خوبی طرز استفاده از Execution Plan را آموزش می‌دهد. [How to read SQL Server graphical query execution plans](#)

دو کتاب زیر، جهت مطالعه و بهینه سازی در ایجاد Query مفید است:

[SQL Server 2012 T-SQL Recipes](#) [SQL Server 2012 Query Performance Tuning](#)

موفق باشید.

در مطلب [قبلی](#) با استفاده از دستور For XML خروجی xml تولید کردیم اما با همین دستور می‌توان تا حدودی خروجی JSON نیر تولید نمود. البته به صورت native هنور در sql server این امکان وجود ندارد که با رای دادن به این [لینک](#) از تیم ماکروسافت بخواهید که این امکان را در نسخه بعدی اضافه کند.
برای این کار یک جدول موقت ایجاد کرده و چند رکورد در آن درج می‌کنیم:

```
declare @t table(id int, name nvarchar(max), active bit)
insert @t values (1, 'Group 1', 1), (2, 'Group 2', 0)
```

حال با استفاده از همان for XML و پارامتر type که نوع خروجی XML را خودمان می‌توانیم تعیین نماییم و پارامتر Path این کار را بصورت زیر انجام می‌دهیم:

```
select '[' + STUFF((
  select
    ',{"id":"' + cast(id as varchar(max))
    + ',"name":"' + name + '''
    + ',"active":"' + cast(active as varchar(max))
    + '}''
  from @t t1
  for xml path(''), type
).value('.','varchar(max)'), 1, 1, '') + ']'
```

توجه کنید در اینجا از پارامتر path بدون نام استفاده شده است و از تابع [STUFF](#) برای در یک رشته در رشته دیگر استفاده شده است. خروجی در زیر آورده شده است:

```
[{"id":1,"name":"Group 1","active":1}, {"id":2,"name":"Group 2","active":0}]
```

حال پیشرفت‌تر آن است که بتوانیم یک join را بصورت فرزندان آن در JSON نمایش دهیم قطعه کد زیر را مشاهده فرمایید:

```
declare @group table(id int, name nvarchar(max), active bit)
insert @group values (1, 'Group 1', 1), (2, 'Group 2', 0)

declare @member table(id int, groupid int, name nvarchar(max))
insert @member values (1, 1, 'Ali'), (2, 1, 'Mojtaba'), (3, 2, 'Hamid')

select '[' + STUFF((
  select
    ',{"id":"' + cast(g.id as varchar(max))
    + ',"name":"' + g.name + '''
    + ',"members": { "children": [ ' +
    (select + STUFF((
      select
        ',{"id":"' + cast(m.id as varchar(max))
        + ',"name":"' + m.name + '"}'
      from @member m
      where m.groupid = g.id
      for xml path(''), type
    ).value('.','varchar(max)'), 1, 1, '')
    + ']}'
    + ',"active":"' + cast(g.active as varchar(max))
    + '}'')
  from @group g
  for xml path(''), type
).value('.','varchar(max)'), 1, 1, '') + ']'
```

خروجی JSON بصورت زیر است:

```
[{"id":1,"name":"Group 1","members":  
  {"children": [{"id":1,"name":"Ali"}, {"id":2,"name":"Mojtaba"}]}]  
,"active":1},  
{"id":2,"name":"Group 2","members":  
  {"children": [{"id":3,"name":"Hamid"}]}}  
, "active":0}]
```

حالت‌های خاص و پیشرفته‌تر را با امکانات t-sql خودتان می‌توانید به همین شکل تولید نمایید.

آشنایی با Window Function ها در SQL Server بخش اول

آشنایی با Window Function ها در SQL Server بخش دوم

در این بخش به دو Function از Analytic Function ها (توابع تحلیلی)، یعنی LAG Function و Lead Function قبل از اینکه به توابع ذکر شده پردازیم، باید عرض کنم، شرح عملکرد اینگونه توابع کمی مشکل می‌باشد، بنابراین با ذکر مثال و توضیح آنها، سعی می‌کنیم، قابلیت هریک را بررسی و درک نماییم.

:Lead Function

این فانکشن در SQL Server 2012 ارائه شده است، و امکان دسترسی، به Data های سطر بعدی نسبت به سطر جاری را در نتیجه یک پرس و جو (Query)، ارائه می‌دهد. بدون آنکه از Self-join استفاده نمایید، تابع فوق بصورت زیر است:

```
LEAD ( scalar_expression [ ,offset ] , [ default ] )
OVER ( [ partition_by_clause ] order_by_clause )
```

شرح Syntax

Scalar_expression: در این Syntax Scalar_expression نام یک فیلد یا ستون درج می‌شود، و مقدار برگشتی فیلد مورد نظر، به مقدار تعیین شده offset نیز بستگی دارد. خروجی Scalar_expression فقط یک مقدار است.
 offset: منظور از Offset در این Syntax همانند عملکرد Over Syntax مربوط به Over می‌باشد. یعنی هر عددی برای offset در نظر گرفته شود، بیانگر نقطه آغازین سطر بعدی یا قبلی نسبت به سطر جاری است. به بیان دیگر، عدد تعیین شده در offset به Sql server می‌فهماند چه تعداد سطر را در محاسبه در نظر نگیرد.
 Default: زمانی که برای Offset مقداری را تعیین می‌نمایید، SQL Server به تعداد تعیین شده در Offset، سطراها را در نظر نمی‌گیرد، بنابراین مقدار خروجی Scalar_expression بطور پیش فرض Null در نظر گرفته می‌شود، چنانچه بخواهید، مقداری غیر از Null درج نمایید، می‌توانید مقدار دلخواه را در قسمت Default وارد کنید.
 OVER ([partition_by_clause] order_by_clause): در [بخش اول](#) بطور کامل توضیح داده شده است.

برای درک بهتر Lead Function چند مثال را بررسی می‌نماییم:
 ابتدا Script زیر را اجرا می‌نماییم، که شامل ایجاد یک جدول و درج 18 رکورد در آن:

```
Create Table TestLead_LAG
(SalesOrderID int not null,
 SalesOrderDetailID int not null ,
 OrderQty smallint not null);
GO
Insert Into TestLead_LAG
    Values (43662,49,1),(43662,50,3),(43662,51,1),
            (43663,52,1),(43664,53,1),(43664,54,1),
            (43667,77,3),(43667,78,1),(43667,79,1),
            (43667,80,1),(43668,81,3),(43669,110,1),
            (43670,111,1),(43670,112,2),(43670,113,2),
            (43670,114,1),(43671,115,1),(43671,116,2)
```

مثال: قصد داریم در هر سطر مقدار بعدی فیلد LeadValue به نام SalesOrderDetailID در فیلد دیگری به نام LeadValue نمایش دهیم، بنابراین Script زیر را ایجاد می‌کنیم:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
LEAD(SalesOrderDetailID) OVER (ORDER BY SalesOrderDetailID) LeadValue
FROM TestLead_LAG s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
```

```
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

خروجی بصورت زیر خواهد بود:

	SalesOrderID	SalesOrderDetailID	OrderQty	LeadValue
1	43663	52	1	77
2	43667	77	3	78
3	43667	78	1	79
4	43667	79	1	80
5	43667	80	1	110
6	43669	110	1	111
7	43670	111	1	112
8	43670	112	2	113
9	43670	113	2	114
10	43670	114	1	NULL

مطابق شکل، براحتی واضح است، که در هر سطر مقدار بعدی فیلد LeadValue در فیلد SalesOrderDetailID در سطر 10، چون مقدار بعدی برای فیلد SalesOrderDetailID وجود ندارد، SQL Server مقدار فیلد LeadValue در نظر می‌گیرد.

در این مثال فقط از آرگومان Scalar_expression، استفاده کردیم، و Offset و Default را مقدار دهی ننمودیم، بنابراین SQL Server بطور پیش فرض هیچ سطربی را حذف نمی‌کند و مقدار Null Default را در نظر می‌گیرد.

مثال دوم: قصد داریم در هر سطر مقدار دو سطر بعدی فیلد LeadValue را در فیلد SalesOrderDetailID نمایش دهیم، و در صورت وجود نداشتن مقدار فیلد SalesOrderDetailID، مقدار پیش فرض صفر، در فیلد LeadValue قرار دهیم، بنابراین آن بصورت زیر خواهد شد:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
LEAD(SalesOrderDetailID,2,0) OVER (ORDER BY SalesOrderDetailID) LeadValue
FROM TestLead_LAG s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

خروجی:

	SalesOrderID	SalesOrderDetailID	OrderQty	LeadValue
1	43663	52		78
2	43667	77	3	79
3	43667	78	1	80
4	43667	79	1	110
5	43667	80	1	111
6	43669	110	1	112
7	43670	111	1	113
8	43670	112	2	114
9	43670	113	2	0
10	43670	114	1	0

در صورت مسئله بیان کرده بودیم، در هر سطر، مقدار فیلد SalesOrderDetailID دو سطر بعدی، را نمایش دهیم، بنابراین مقداری که برای Offset در نظر می‌گیریم، برابر دو خواهد بود، سپس گفته بودیم، چنانچه در هر سطر مقدار فیلد SalesOrderDetailID وجود نداشت، بجای مقدار پیش فرض Null از مقدار صفر استفاده شود، بنابراین به Default مقدار صفر را نسبت دادیم.

LEAD(SalesOrderDetailID,2,0)

در شکل، مطابق صورت مسئله، مقدار فیلد LeadValue سطر اول برابر است با 78، به بیان ساده‌تر برای بدست آوردن مقدار فیلد LeadValue هر سطر، می‌بایست هر سطر را به علاوه 2 (Offset) نماییم، تا سطر بعدی بدست آید، سپس مقدار SalesOrderDetailID را در فیلد LeadValue قرار می‌دهیم.
به سطر 9 و 10 توجه نمایید، که مقدار فیلد LeadValue آنها برابر با صفر است، واضح است، سطر $10 + 2 = 12$ به $2+10=12$ چنین سطربالی در خروجی نداریم، بنابراین بطور پیش فرض مقدار LeadValue در SQL Server Null در نظر گرفته می‌شود، اما نمی‌خواستیم، که این مقدار Null باشد، بنابراین به آرگومان Default مقدار صفر را نسبت دادیم، تا SQL Server، به جای استفاده از Null، مقدار در نظر گرفته شده صفر را استفاده نماید.
اگر چنین فانکشنی وجود نداشت، برای شبیه سازی آن می‌بایست از Join روی خود جدول استفاده می‌نمودیم، و یکسری محاسبات دیگر، که کار را سخت می‌نمود، مثال دوم را با Script زیر می‌توان شبیه سازی نمود:

```
WITH cteLead
AS
(
SELECT SalesOrderID,SalesOrderDetailID,OrderQty,
       ROW_NUMBER() OVER (ORDER BY SalesOrderDetailID) AS sn
FROM TestLead_LAG
WHERE
SalesOrderID IN (43670, 43669, 43667, 43663)
)
SELECT m.SalesOrderID, m.SalesOrderDetailID, m.OrderQty,
       case when sLead.SalesOrderDetailID is null Then 0 Else sLead.SalesOrderDetailID END as leadvalue
FROM cteLead AS m
LEFT OUTER JOIN cteLead AS sLead ON sLead.sn = m.sn+2
ORDER BY m.SalesOrderID, m.SalesOrderDetailID, m.OrderQty
```

جدول موقتی ایجاد نمودیم، که ROW_Number را در آن اضافه کردیم، سپس جدول ایجاد شده را با خود Join کردیم، و گفتیم، که مقدار فیلد LeadValue هر سطر برابر است با مقدار فیلد SalesOrderDetailID دو سطر بعد از آن. و با Case نیز مقدار پیش فرض را صفر در نظر گرفتیم.

:LAG Function

این فانکشن نیز در SQL Server 2012 ارائه شده است، و امکان دسترسی، به Data های سطر قبلی نسبت به سطر جاری را در نتیجه یک پرس و جو (Query)، ارائه می‌دهد. بدون آنکه از Self-join استفاده نمایید، آن شبیه به فانکشن Lead Syntax میباشد و بصورت زیر است:

```
LAG (scalar_expression [,offset] [,default])
OVER ( [ partition_by_clause ] order_by_clause )
```

Syntax مربوط به فانکشن LAG را شرح نمی‌دهم، بدليل آنکه شبیه به فانکشن Lead می‌باشد، فقط تفاوت آن در Offset است، در فانکشن LAG روی سطرهای ماقبل سطر جاری اعمال می‌گردد. مثال دوم را برای حالت LAG Function شبیه سازی می‌نماییم:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
LAG(SalesOrderDetailID,2,0) OVER (ORDER BY SalesOrderDetailID) LAGValue
FROM TestLead_LAG s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
go
```

خروجی :

	SalesOrderID	SalesOrderDetailID	OrderQty	LAGValue
1	43663	52	1	0
2	43667	77	3	0
3	43667	78	1	52
4	43667	79	1	77
5	43667	80	1	78
6	43669	110	1	79
7	43670	111	1	80
8	43670	112	2	110
9	43670	113	2	111
10	43670	114	1	112

همانطور که گفتیم، عکس LAG Function میباشد. یعنی مقدار فیلد LAGValue سطر جاری برابر است با مقدار SalesOrderDetailID دو سطر ما قبل خود. مقدار فیلد LAGValue دو سطر اول و دوم نیز برابر صفر است، چون دو سطر ماقبل آنها وجود ندارد، و مقدار صفر نیز بدليل این است که Default را برابر صفر در نظر گرفته بودیم. مثال: در این مثال از LAG Function و Lead Function همزمان استفاده می‌کنیم، با این تفاوت، که از گروه بندی نیز استفاده شده است:

Zir را اجرا نمایید:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
Lead(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID ORDER BY SalesOrderDetailID) LeadValue,
LAG(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID ORDER BY SalesOrderDetailID) LAGValue
FROM TestLead_LAG s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
go
```

خروجی:

	SalesOrderID	SalesOrderDetailID	OrderQty	LeadValue	LAGValue
1	43663	52	1	NULL	NULL
2	43667	77	3	78	NULL
3	43667	78	1	79	77
4	43667	79	1	80	78
5	43667	80	1	NULL	79
6	43669	110	1	NULL	NULL
7	43670	111	1	112	NULL
8	43670	112	2	113	111
9	43670	113	2	114	112
10	43670	114	1	NULL	113

با بررسی هایی که در مثالهای قبل نمودیم، خروجی زیر را می‌توان براحتی تشخیص داد، و توضیح بیشتری نمی‌دهم.
موفق باشید.

نظرات خوانندگان

نویسنده: محمد
تاریخ: ۱۳۹۱/۱۰/۲۸ ۱۸:۵۳

سلام،

این توابع واقعا کار رو آسون کردن. ما رو از بکارگیری چندین بار self join بی نیاز کردن. بطور نمونه اگه بخواهیم مقدار SalesOrderDetailID سطر قبلی، سطر بعدی و دو سطر بعدی را بدست بیاریم در نسخه 2008 ساده‌ترین و مناسب‌ترین کوئری این هست:

```
WITH cteLead
AS
(
SELECT SalesOrderID, SalesOrderDetailID, OrderQty,
       ROW_NUMBER() OVER (PARTITION BY SalesOrderID
                           ORDER BY SalesOrderDetailID) AS sn
  FROM TestLead_LAG
 WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
)
SELECT m.SalesOrderID, m.SalesOrderDetailID, m.OrderQty,
       COALESCE(sLead1.SalesOrderDetailID, 0) as leadvalue1,
       COALESCE(sLead2.SalesOrderDetailID, 0) as leadvalue2,
       COALESCE(sLag1.SalesOrderDetailID, 0) as lagvalue2,
       COALESCE(sLag2.SalesOrderDetailID, 0) as lagvalue2
  FROM cteLead AS m
 LEFT OUTER JOIN cteLead AS sLead1
    ON m.sn = sLead1.sn - 1
   AND m.SalesOrderID = sLead1.SalesOrderID
 LEFT OUTER JOIN cteLead AS sLead2
    ON m.sn = sLead2.sn - 2
   AND m.SalesOrderID = sLead2.SalesOrderID
 LEFT OUTER JOIN cteLead AS sLag1
    ON m.sn = sLag1.sn + 1
   AND m.SalesOrderID = sLag1.SalesOrderID
 LEFT OUTER JOIN cteLead AS sLag2
    ON m.sn = sLag2.sn + 2
   AND m.SalesOrderID = sLag2.SalesOrderID
 ORDER BY m.SalesOrderID, m.SalesOrderDetailID, m.OrderQty;
```

در حالی که با دو تابعی که شما در اینجا پوشش دادین میشه کوئری فوق العاده ساده‌تر نمود:

```
SELECT s.SalesOrderID, s.SalesOrderDetailID, s.OrderQty,
       LEAD(SalesOrderDetailID, 1, 0) OVER (PARTITION BY SalesOrderID ORDER BY SalesOrderDetailID)
      LeadValue1,
       LAG(SalesOrderDetailID, 1, 0) OVER (PARTITION BY SalesOrderID ORDER BY SalesOrderDetailID)
      LAGValue1,
       LEAD(SalesOrderDetailID, 2, 0) OVER (PARTITION BY SalesOrderID ORDER BY SalesOrderDetailID)
      LeadValue2,
       LAG(SalesOrderDetailID, 2, 0) OVER (PARTITION BY SalesOrderID ORDER BY SalesOrderDetailID)
      LAGValue2,
  FROM TestLead_LAG s
 WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
 ORDER BY s.SalesOrderID, s.SalesOrderDetailID, s.OrderQty
```

نویسنده: حسین
تاریخ: ۱۳۹۱/۱۰/۳۰ ۱۴:۵۸

جالب بود مرسی

برای مطالعه این بخش لازم است، به Syntax مربوط به Over آشنا باشیم، در بخش اول بطور کامل به Syntax مربوط به Over پرداختیم.

در این بخش دو فانکشن دیگر از توابع تحلیلی (Analytic functions) به نامهای First_Value و Last_Value را بررسی می‌نماییم.

First_Value

این فانکشن نیز همانند دیگر فانکشن‌های تحلیلی در نسخه 2012 SQL Server ارائه گردیده است. و اولین مقدار از یک مجموعه مقادیر را بر می‌گرداند. و Syntax آن بصورت ذیل می‌باشد:

```
FIRST_VALUE ( [ scalar_expression ] )
OVER ( [ partition_by_clause ] order_by_clause [ rows_range_clause ] )
```

شرح Syntax:

1- Scalar_expression : مقدار آن می‌تواند نام یک فیلد یا Subquery باشد.

2- Over : در بخش اول بطور مفصل آن را بررسی نمودیم.

قبل از بررسی تابع First_Value Script ابتدا زیر را اجرا نمایید، که شامل یک جدول و درج چند رکورد در آن است.

```
Create Table Test_First_Last_Value
(SalesOrderID int not null,
 SalesOrderDetailID int not null ,
 OrderQty smallint not null);
GO
Insert Into Test_First_Last_Value
    Values (43662,49,1),(43662,50,3),(43662,51,1),
           (43663,52,1),(43664,53,1),(43664,54,1),
           (43667,77,3),(43667,78,1),(43667,79,1),
           (43667,80,1),(43668,81,3),(43669,110,1),
           (43670,111,1),(43670,112,2),(43670,113,2),
           (43670,114,1),(43671,115,1),(43671,116,2)
```

مثال: ابتدا Script ایجاد می‌نماییم، بطوریکه جدول Test_Firts_Last_Value را براساس فیلد SalesOrderID گروه بندی نموده و اولین مقدار فیلد SalesOrderDetailID در هر گروه را مشخص نماید.

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
       FIRST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
                                             ORDER BY SalesOrderDetailID) FstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

خروجی:

	SalesOrderID	SalesOrderDetailID	OrderQty	FstValue
1	43663	52	1	52
2	43667	77	3	77
3	43667	78	1	77
4	43667	79	1	77
5	43667	80	1	77
6	43669	110	1	110
7	43670	111	1	111
8	43670	112	2	111
9	43670	113	2	111
10	43670	114	1	111

طبق Script چهار گروه در خروجی ایجاد شده است و در فیلد FstValue ، اولین مقدار هر گروه نمایش داده می شود. اگر بخش های قبلی Window Function ها را مطالعه کرده باشید، تحلیل اینتابع کار بسیار ساده ای است.

Last_Value

اینتابع نیز در نسخه SQL Server 2012 ارائه گردیده است. و آخرین مقدار از یک مجموعه مقادیر را بر می گرداند، به عبارتی فانکشن Last_Value عکس فانکشن First_Value عمل می نماید و آن به شرح ذیل میباشد:

```
LAST_VALUE ( [scalar_expression] )
OVER ( [ partition_by_clause ] order_by_clause rows_range_clause )
```

شرح Syntax تابع Last_Value شبیه به تابع First_Value می باشد.

مثال: همانند مثال قبل Script ایجاد می نماییم، بطوریکه جدول Test_Firts_Last_Value را براساس فیلد SalesOrderID گروه بندی نموده و آخرین مقدار فیلد SalesOrderDetailID در هر گروه را مشخص نماید.

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
       LAST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
                                             ORDER BY SalesOrderDetailID RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
LstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

خروجی:

	SalesOrderID	SalesOrderDetailID	OrderQty	LstValue
1	43663	52	1	52
2	43667	77	3	80
3	43667	78	1	80
4	43667	79	1	80
5	43667	80	1	80
6	43669	110	1	110
7	43670	111	1	114
8	43670	112	2	114
9	43670	113	2	114
10	43670	114	1	114

خروجی جدول، به چهار گروه تقسیم، و آخرین مقدار هر گروه، در فیلد LstValue نمایش داده شده است. در این مثال نیز تحلیلی نخواهیم داشت، چون فرض بر آن است که بخش‌های قبلی را مطالعه نموده ایم.

موفق باشید.

نظرات خوانندگان

نویسنده: محمد
تاریخ: ۱۳۹۱/۱۰/۲۸ ۱۰:۳۹

سلام،

مطلوب اول: قسمت `order by` در ماده `over` در هر دو کوئری به چه جهت آمده است؟

مطلوب دوم: چه مزیتی نسبت به `min()` داره، منظورم اینه که میشه خروجی کوئری اولتون رو با این کوئری بدست آورد:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,  
       MIN(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID) FstValue  
  FROM Test_First_Last_Value s  
 WHERE SalesOrderID IN (43670, 43669, 43667, 43663)  
 ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

نویسنده: محمد
تاریخ: ۱۳۹۱/۱۰/۲۸ ۱۲:۱۱

سلام،

من 2012 SQL Server ندارم، ولی تا اونجایی که متوجه شدم بر اساس شواهد دو کوئری زیر باید یک نتیجه رو برگرداند. منظورم اینکه که با `last_value` میشه `first_value` استفاده بشه.
اگه من اشتباه میکنم لطفا راهنمایی بفرمایید.

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,  
       LAST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID  
                                              ORDER BY SalesOrderDetailID) LstValue  
  FROM Test_First_Last_Value s  
 WHERE SalesOrderID IN (43670, 43669, 43667, 43663)  
 ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty  
  
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,  
       FIRST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID  
                                              ORDER BY SalesOrderDetailID DESC) FstValue  
  FROM Test_First_Last_Value s  
 WHERE SalesOrderID IN (43670, 43669, 43667, 43663)  
 ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

نویسنده: فرهاد فرهمند خواه
تاریخ: ۱۳۹۱/۱۰/۲۹ ۱۱:۵۳

سلام

جواب سؤال اول: در Syntax تابع `First_value` استفاده از `Order by` اجباری میباشد.

جواب سؤال دوم:

مقاله زدم شما را به اشتباہ انداخت، در زیر با یک مثال، در `First_value` و `Min` را مقایسه میکنیم.
ابتدا یک جدول و چند رکورد، در آن درج میکنیم:

```
CREATE TABLE Employees (  
    EmployeeId INT IDENTITY PRIMARY KEY,  
    Name VARCHAR(50),  
    HireDate DATE NOT NULL,  
    Salary INT NOT NULL  
)  
GO
```

```
INSERT INTO Employees (Name, HireDate, Salary)
VALUES
    ('Alice', '2011-01-01', 20000),
    ('Brent', '2011-01-15', 19000),
    ('Carlos', '2011-02-01', 22000),
    ('Donna', '2011-03-01', 25000),
    ('Evan', '2011-04-01', 18500)
GO
```

در ادامه Script زیر را اجرا می‌کنیم:

```
Select EmployeeId, Name, Salary, HireDate,
    First_Value(HireDate) OVER(ORDER BY Salary RANGE BETWEEN UNBOUNDED PRECEDING
                                AND UNBOUNDED FOLLOWING) AS First,
    Min(HireDate) OVER(ORDER BY Salary
                                RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) AS Min
FROM Employees
ORDER BY EmployeeId
GO
```

خروجی بصورت زیر می‌شود:

	EmployeeId	Name	Salary	HireDate	First	Min
1	1	Alice	20000	2011-01-01	2011-04-01	2011-01-01
2	2	Brent	19000	2011-01-15	2011-04-01	2011-01-01
3	3	Carlos	22000	2011-02-01	2011-04-01	2011-01-01
4	4	Donna	25000	2011-03-01	2011-04-01	2011-01-01
5	5	Evan	18500	2011-04-01	2011-04-01	2011-01-01

در شکل بالا تقاضه First_Value و Min بطور کامل مشخص است، اگر به دقت نمایید، Sort Query براساس Salary انجام شده است، برای حالت First_value، مقدار فیلد HireDate در اولین رکورد، برابر است با 2011-04-01، بنابراین سورت روی نمایش تاثیر گذار است، بطوريکه Sort برای حالت Min، تاثیر گذار نمی‌باشد، وتابع Min، کوچکترین مقدار، از مقادير ستون HireDate را بدست می‌آورد، به بیان ساده‌تر در حالت استفاده از Min، عملیات Sort بیهوده می‌باشد. چون تابع MIN روی کل مقادير يك گروه يا ستون تاثير مي‌گذارد.

نویسنده: فرهاد فرهمند خواه
تاریخ: ۱۳۹۱/۱۰/۲۹ ۱۲:۵۱

سلام

شما می‌توانید، با دستکاری Query‌ها خروجی‌های یکسانی را ایجاد نمایید، دو Query که ایجاد نمودید، خروجی یکسانی ندارند. Query دوم شما با خروجی Last_Value مقاله یکسان است، اما باید بگوییم که مفهوم Last_Value این است که آخرین سطر در یک گروه را بر می‌گرداند. بهتر است [بخش اول](#) را مطالعه نمایید.

علت یکسان نبودن نتیجه دو Query شما در نحوه Sort و مفهوم First_Value و Last_Value می‌باشد: نکنه: اگر در Order by Over Clause را اعمال نماییم، اما از Row استفاده نکنیم، SQL Server بصورت پیش فرض از قالب زیر استفاده می‌نماید:

RANGE UNBOUNDED PRECEDING AND CURRENT ROW

نویسنده: محمد

تاریخ: ۱۳۹۱/۱۰/۲۹ ۱۳:۲۳

ممنون از پاسختون، الان متوجه تفاوتشون شدم.

ستون `first_value` باشد:

```
Select EmployeeId,Name,Salary,HireDate,
First_VALUE(HireDate) OVER(ORDER BY Salary RANGE BETWEEN UNBOUNDED PRECEDING
                           AND UNBOUNDED FOLLOWING) AS First,
D.HireDate AS older_method
FROM Employees
CROSS APPLY (SELECT TOP 1 HireDate
             FROM Employees
             --WHERE E1.EmployeeId = E2.EmployeeId
             ORDER BY Salary ASC) AS D
ORDER BY EmployeeId;
```

نویسنده: محمد

تاریخ: ۱۳۹۱/۱۰/۲۹ ۱۳:۳۱

ممنون از شما، من مطالب بخش اول رو مطالعه کردم.

عبارت `RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING` به معنای تمام سطرهای جدول هست دیگه درسته. یعنی تمام سطرهای جدول از اولین گرفته، جاری گرفته و آخرین رو پوشش میده.

با این توضیحات باید دو کوئیری زیر اینبار حواب یکسانی بدهند:

```
SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
       FIRST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
                                             ORDER BY SalesOrderDetailID RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
LstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
      ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty

SELECT s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty,
       LAST_VALUE(SalesOrderDetailID) OVER (PARTITION BY SalesOrderID
                                             ORDER BY SalesOrderDetailID DESC RANGE BETWEEN UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING)
LstValue
FROM Test_First_Last_Value s
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
      ORDER BY s.SalesOrderID,s.SalesOrderDetailID,s.OrderQty
```

دو کوئیری کاملاً یکسان هستند به غیر از اینکه در کوئیری دوم یک `DESC` اضافه شده و نام تابع از `first` به `last` تغییر کرده است.

به همراه بسته [Features pack](#) اس کیوال سرور 2012، دو بسته SqlDom.msi نیز وجود دارند (نسخه‌های [x86](#) و [x64](#)). این بسته حاوی اسambilی Microsoft.SqlServer.TransactSql.ScriptDom.dll می‌باشد که نهایتاً در آدرس `Program Files\Microsoft SQL Server\110\SDK\Assemblies` کپی خواهد شد.

به کمک آن می‌توان عبارات پیچیده Parse و آنالیز کرد. البته باید در نظر داشت هرچند این بسته جهت SQL Server 2012 ارائه شده اما این اسambilی با نگارش‌های 2005 به بعد اس کیوال سرور کاملاً سازگار است و اساساً نیازی هم به ندارد. در ادامه مروری خواهیم داشت بر نحوه استفاده از آن.

یافتن کوئری‌های * Select در بین انبوهی از اسکریپت‌ها به کمک SQLDom

در مورد [مضرات کوئری‌های *](#) پیشتر مطلبی را در این سایت خوانده‌اید. در ادامه قصد داریم به کمک امکانات اسambilی `select`, `Microsoft.SqlServer.TransactSql.ScriptDom.dll` تعدادی عبارت T-SQL را آنالیز کرده و مشخص کنیم که آیا حاوی * هستند یا خیر. کد کامل آن را در ذیل مشاهده می‌کنید:

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text;
using Microsoft.SqlServer.TransactSql.ScriptDom;

namespace DbCop
{
    // Microsoft® SQL Server® 2012 Transact-SQL ScriptDom
    // SQL Server 2012 managed parser, Supports SQL Server 2005+
    // SQLDom.msi (redist x86/x64)
    // http://www.microsoft.com/en-us/download/details.aspx?id=29065
    // X86: http://go.microsoft.com/fwlink/?LinkID=239634&clcid=0x409
    // X64: http://go.microsoft.com/fwlink/?LinkID=239635&clcid=0x409
    // Program Files\Microsoft SQL
    // Server\110\SDK\Assemblies\Microsoft.SqlServer.TransactSql.ScriptDom.dll

    class Program
    {
        static void Main()
        {
            const string tSql = @"
                -- select * in PROCEDURE
                CREATE PROCEDURE dbo.SelectStarTest
                AS
                SELECT * FROM dbo.tbl1
                go

                -- select * in PROCEDURE with TableVar
                Create Procedure SelectAll
                AS
                Declare @X table(Id integer)
                Select * from @x
                go

                -- select * in PROCEDURE with cte
                CREATE PROCEDURE dbo.SelectAllCte
                AS
                WITH cte
                AS (
                SELECT * FROM sys.objects
                )
                SELECT * FROM cte
                go

                -- normal select *
                select * from tbl1;
                select * from dbo.tbl2;
            ";
        }
    }
}
```

```

    IList<ParseError> errors;
    TSqlScript sqlFragment;
    using (var reader = new StringReader(tSql))
    {
        var parser = new TSql110Parser(initialQuotedIdentifiers: true);
        sqlFragment = (TSqlScript)parser.Parse(reader, out errors);
    }

    if (errors != null && errors.Any())
    {
        var sb = new StringBuilder();
        foreach (var error in errors)
            sb.AppendLine(error.Message);

        throw new InvalidOperationException(sb.ToString());
    }

    var i = 0;
    foreach (var batch in sqlFragment.Batches)
    {
        Console.WriteLine("Batch: {0}, Statement(s): {1}", ++i, batch.Statements.Count);
        foreach (var statement in batch.Statements)
        {
            processStatement(statement);
        }
        Console.WriteLine();
    }

    Console.WriteLine("\nPress a key...");
    Console.Read();
}

private static void processStatement(TSqlStatement statement)
{
    var createProcedureStatement = statement as CreateProcedureStatement;
    if (createProcedureStatement != null)
    {
        var statementList = createProcedureStatement.StatementList;
        foreach (var procedureStatement in statementList.Statements)
        {
            processStatement(procedureStatement);
        }
    }

    var selectStatement = statement as SelectStatement;
    if (selectStatement != null)
    {
        var query = selectStatement.QueryExpression;
        var selectElements = ((QuerySpecification)query).SelectElements;
        foreach (var selectElement in selectElements)
        {
            var expression = selectElement as SelectStarExpression;
            if (expression == null) continue;
            Console.WriteLine(
                "`Select *` detected @StartOffset:{0}, Line:{1}, T-SQL: {2}",
                expression.StartOffset,
                expression.StartLine,
                statementToString(selectStatement));
        }
    }
}

private static string statementToString(TSqlFragment selectStatement)
{
    var text = new StringBuilder();
    for (var i = selectStatement.FirstTokenIndex; i <= selectStatement.LastTokenIndex; i++)
    {
        text.Append(selectStatement.ScriptTokenStream[i].Text);
    }
    return text.ToString();
}
}

```

توضیحات:

پس از نصب SQLDom.msi، ارجاعی را به اسمنلی زیر اضافه نمایید تا بتوانید کد فوق را کامپایل کنید:

Program Files\Microsoft SQL Server\110\SDK\Assemblies\Microsoft.SqlServer.TransactSql.ScriptDom.dll

کار با ایجاد و هله‌ای از TSq1110Parser شروع می‌شود. متده استفاده از Parse آن، آرگومانی از نوع TextReader را قبول می‌کند. برای مثال با استفاده از StringReader می‌توان محتوای یک متغیر رشته‌ای را به آن ارسال کرد و یا توسط StreamReader یک فایل sq1 را پس از فراخوانی متده Parse، بهتر است برسی شود که آیا عبارت T-SQL دریافتی معتبر بوده است یا خیر. اینکار را توسط لیستی از ParseError‌های دریافتی می‌توان انجام داد.

خروجی متده Parse، حاوی یک سری آنالیز شده است. هر عبارت Go در اینجا یک Batch را تشکیل می‌دهد. سپس در داخل هر batch به دنبال batch.Statements خواهیم گشت تا بتوان به عبارات T-SQL آن‌ها دسترسی یافت.

در ادامه کار اصلی توسط متده processStatement صورت می‌گیرد. عبارات دریافتی، در حالت کلی از نوع TSqlStatement هستند اما در اصل می‌توانند یکی از مشتقات آن نیز باشند. در اینجا فقط دو مورد SelectStatement و CreateProcedureStatement بررسی شده‌اند (مطابق رشته tSql ابتدای مثال). هر دو عبارت، از کلاس TSqlStatement مشتق شده‌اند.

در متده processStatement عبارات select معمولی و همچنین آن‌هایی که داخل رویه‌های ذخیره شده تعریف شده‌اند، استخراج شده و در نهایت بررسی می‌شوند که آیا از نوع SelectStarExpression هستند یا خیر (همان * صورت مساله).

خروجی مثال فوق به شرح زیر است:

```

Batch: 1, Statement(s): 1
`Select *` detected @StartOffset:140, Line:5, T-SQL: SELECT * FROM dbo.tbl1

Batch: 2, Statement(s): 1
`Select *` detected @StartOffset:368, Line:12, T-SQL: Select * from @x

Batch: 3, Statement(s): 1
`Select *` detected @StartOffset:659, Line:22, T-SQL: WITH ctex
      AS (
        SELECT * FROM sys.objects
      )
    SELECT * FROM ctex

Batch: 4, Statement(s): 2
`Select *` detected @StartOffset:753, Line:26, T-SQL: select * from tbl1;
`Select *` detected @StartOffset:791, Line:27, T-SQL: select * from dbo.tbl2;

```

نظرات خوانندگان

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۷/۳۰ ۱۸:۱۳

یک نکته‌ی تکمیلی

بسته‌ی ScriptDom را از [نیوگت](#) نیز می‌توانید دریافت کنید:

```
PM> Install-Package Microsoft.SqlServer.TransactSql.ScriptDom
```

فرض کنید میخواهیم سطرهای جدول را 6 تا 6 تا 2 سوا کنیم و به هر کدام یک عددی انتساب دهیم و هر قسم تولید شده را نیز 2 تا 2 تا سوا کنیم و بهش عدد انتساب دهیم.
 به تصویر زیر توجه بفرمایید. ابتدا داده‌ها به دو دسته شش تایی تقسیم شدن (ستون nt1)، سپس هر کدام از این دسته‌ها نیز به سه دسته دوتایی تقسیم شدن (ستون grp) هدف ما تولید دو ستون nt1 و grp توسط query می‌باشد.

	nbr	nt1	grp
1	1	1	1
2	2	1	1
3	3	1	2
4	4	1	2
5	5	1	3
6	6	1	3
7	7	2	1
8	8	2	1
9	9	2	2
10	10	2	2
11	11	2	3
12	12	2	3

برای بدست آوردن مقادیر دو ستون مذکور روش‌های متنوعی وجود دارد که برخی از آنها را در اینجا پوشش میدم.
 قبل از هر چیزی ابتدا جدول را ایجاد و 12 سطر زیر را در آن انتشار دهید:

```
CREATE TABLE T (nbr INT NOT NULL);
INSERT T VALUES (1), (2), (3), (4), (5), (6),
(7), (8), (9), (10), (11), (12);
```

روش اول:

این روش، تعیین پذیری و پویایی ندارد و برای هر سناریویی مناسب نخواهد بود. ولی از آنجایی که دیدم کوئری زیر میتواند یک نمونه از کاربرد Ntile باشه آن را مطرح کردم.

```
SELECT nbr, nt1, NTILE(3) OVER(PARTITION BY nt1 ORDER BY nbr) AS grp
FROM (
    SELECT nbr, NTILE(2) OVER(ORDER BY nbr) nt1
    FROM T
) AS D;
```

تابع ntile داخلی سطرهای جدول را به دو قسم تقسیم می‌کند و برای قسم اول عدد 1 و برای قسم دوم عدد 2 را در نظر می‌گیرد.

تابع ntile بیرونی بر اساس دو عدد 1 و 2 گروه بندی انجام داده و هر گروه را به 3 قسم تقسیم می‌کند. قسمت اول 1، دوم 2 و

سوم 3 خواهد بود.

لازم به ذکر است که باید خارج قسمت تقسیم تعداد سطراها بر عدد `ntile` یک عدد صحیح باشد تا خروجی مناسب داشته باشیم. و همچنین بایستی بدانیم که تعداد سطراهای جدول چنتاست تا آن را به گونه ای تقسیم کنیم که خارج قسمت برابر شود با عدد مورد نظر ما یعنی 6.

روش دوم:

در این روش برخلاف روش قبل که همه چیز توسطتابع بدست می آمد باید خودمان دست به کار شویم و فرمولی را بدست آوریم که نتیجه مورد نظر را تولید کند.

برای حل این مساله ابتدا باید سطراهای جدول را 6 تا 6 تا سوا کنیم و عناصر هر دسته را شماره گذاری کنیم (از 1 تا 6 بر اساس ترتیب مقدار `nbr`) سپس با کمک سایر فرمولها دسته ها را دوتا دوتا شماره گذاری میکنیم.

به تصویر زیر توجه بفرمایید:

	<code>nbr</code>	<code>mk1</code>	<code>mk2</code>	<code>grp2</code>	<code>grp2</code>
1	1	1	1	1	1
2	2	2	2	1	1
3	3	3	3	2	2
4	4	4	4	2	2
5	5	5	5	3	3
6	6	6	6	3	3
7	7	1	1	1	1
8	8	2	2	1	1
9	9	3	3	2	2
10	10	4	4	2	2
11	11	5	5	3	3
12	12	6	6	3	3

در کادر نارنجی رنگ همانطور که اشاره شد ما سطراهای شمارگذاری شده ای داریم که در رنج 1 تا 6 هستند. و در کادر بنفش ستون مورد نظر ما قرار دارد. ستون بنفش با کمک ستون نارنجی بدست آمده است. اگر تقسیم صحیح را `div` و باقیمانده صحیح را `mod` بگیریم فرولهای مورد نظر به این شرح خواهد بود:

$$(nbr - 1) \bmod 6 + 1$$

$$\left(((nbr - 1) \bmod 6 + 1) + 1 \right) \div 2$$

طبق کوئری زیر ستون نارنجی(rnk1/rnk2) را به دو طریق می‌توان ایجاد نمود و ستون بنفس(grp1/grp2) را نیز به دو طریق می‌توان ایجاد نمود.

```
SELECT nbr, rnk1, rnk2,
       (rnk1 + 1) / 2 AS grp2,
       (rnk1 - 1) / 2 + 1 AS grp2
  FROM (
    SELECT nbr,
           ROW_NUMBER() OVER(PARTITION BY (nbr + 5) / 6 ORDER BY nbr) rnk1,
           (nbr - 1) % 6 + 1 AS rnk2
      FROM t
     )d
```

حذف نمودن کاراکتر های ناخواسته توسط Recursive CTE قسمت اول

عنوان:

محمد سلیم ابادی

تاریخ: ۱۵:۳۵ ۱۳۹۱/۱۱/۰۱

آدرس: www.dotnettips.info

برچسبها: SQL Server, T-SQL, recursive cte, replace

شاید برایتان تا حالا پیش آمده باشد که بخواهید یکسری کاراکترهای ناخواسته و اضافه را از یک رشته حذف کنید. بطور مثال تمام کاراکتر هایی غیر عددی را باید از یک رشته حذف نمود تا آن رشته قابلیت تبدیل به نوع integer را بدست بیاورد.

اگر تعداد کاراکترهای ناخواسته محدود و مشخص هستند میتوانید با دستور REPLACE آنها را حذف کنید، مثلا میخواهیم هر سه کاراکتر ~!@ از رشته حذف شوند:

```
DECLARE @s VARCHAR(50) = '~~~~~!@@@@@@@ salam';
SET @s = REPLACE(REPLACE(REPLACE(@s, '~', ''), '!', ''), '@', '');
SELECT @s AS new_string
```

ولی هنگامی که کاراکترها نامحدود بوده امکان نوشتتن تابع REPLACE به کرات بی معنا است در این حالت باید دنبال روشی پویا و تعمیم پذیر بود.

با جستجویی که در اینترنت انجام دادم متوجه شدم تکنیک WHILE یا همون loop یکی از روش های رایج برای انجام اینکار هست، که احتمالا به دلیل سهولت در بکارگیری و سادگی آن بوده که عمومیت پیدا کرده است. مستقل از این صحبتها هدف معرفی یک روش مجموعه گرا (set-based) برای این مساله می باشد.

حذف کاراکترها ناخواسته با تکنیک Recursive CTE راه حل بر اساس جدول زیر است:

```
CREATE TABLE test_string
(id integer not null primary key,
string_value varchar(500) not null);

INSERT INTO test_string
VALUES (1, '@@@ #### salam 12345'),
(2, 'good $$$$$ &&&& bye 00000');
```

حالا فرض کنید می خواهیم هر کاراکتری غیر از حروف الفبای انگلیسی و فاصله (space) از رشته حذف شود. پس دو داده فوق به صورت salam و good bye در انتها در خواهد آمد. برای حذف کاراکترهای ناخواسته فوق query زیر را اجرا کنید.

```
WITH CTE (ID, MyString, Ix) AS
(
    SELECT id,
           string_value,
           PATINDEX('%[^a-z ]%', string_value)
      FROM test_string
UNION ALL
    SELECT id,
           CAST(REPLACE(MyString, SUBSTRING(MyString,Ix , 1), '') AS VARCHAR(500)),
           PATINDEX('%[^a-z ]%', REPLACE(MyString, SUBSTRING(MyString,Ix , 1), '')) 
      FROM CTE
     WHERE Ix > 0
)
SELECT *
  FROM cte
 --WHERE Ix = 0;
 ORDER BY id, CASE WHEN Ix = 0 THEN 1 ELSE 0 END, Ix;
```

:query توضیح

در قسمت anchor اندیس اولین کاراکتر ناخواسته (خارج از رنج حروف الفبا و فاصله) بدست می‌آید. سپس در قسمت recursive هر کاراکتری که برابر باشد با کاراکتر ناخواسته ای که در مرحله قبل بدست آمده از رشته حذف می‌شود این عملیات توسطتابع صورت می‌گیرد و اندیس کاراکتر ناخواسته بعدی بعد از حذف کاراکتر ناخواسته قبلی بدست می‌آید که به مرحله بعد منتقل می‌شود. این مراحل تا آنجایی پیش می‌رود که دیگر کاراکتر ناخواسته ای در رشته وجود نداشته باشد.

به جدول زیر توجه بفرمایید (خروجی query فوق)

ID	MyString	lx
1	@@@## salam 12345	1
2	#### salam 12345	2
3	salam 12345	9
4	salam 2345	9
5	salam 345	9
6	salam 45	9
7	salam 5	9
8	salam	0
9	good \$\$\$\$\$ &&&& bye 00000	6
10	good &&&& bye 00000	7
11	good bye 00000	12
12	good bye	0

نتیجه مطلوب ما آن دو سطری است که در قادر بنفس هستند. که اگر به ستون IX اشاره توجه کنید مقدارش برابر با 0 است.

لطفا به سطر اول جدول توجه بفرمایید مشاهده می‌شود که هر 4 کاراکتر @ یکباره از رشته حذف شدند که بدلیل استفاده از تابع REPLACE میباشد.

نظرات خوانندگان

نویسنده: سید حمزه
تاریخ: ۱۳۹۲/۰۶/۲۷ ۱۶:۵۳

سلام
خیلی جامع بود
 فقط من متوجه نشدم دقیقا کجای کوئریم باید اینو بنویسم.
 و همینطور میخواستم اگر بشه یک فانکشن بسازم و از اون هم توی select استفاده کنم.
 ممنون

:LIKE کامل syntax

```
match_expression [ NOT ] LIKE pattern [ ESCAPE escape_character ]
```

چهار کاراکتر وجود دارد که الگوهای جستجو را توسط آنها تعریف می‌کنند. این کاراکترها wildcard نام دارند. زمانی که کاراکترهای wildcard موجود در الگوی ما به عنوان کاراکترهای عادی در نظر گرفته شده اند نه wildcard آنگاه باید به SQL اطلاع دهیم که آنها را با کاراکترهای wildcard اشتباه نگیرد، برای این منظور چندین راه حل وجود دارد که در ادامه شرح خواهیم داد.

فرض کنید الگوی شما این مقدار باشد:
"نها دو کاراکتر % کاراکترهای wildcard هستند"

```
%[abcde_ ]%
```

روش اول: ESCAPE

```
WHERE string_value LIKE '%@[abcde@_ ]%' ESCAPE '@'
```

دو کاراکتر [و _ علامت گذاری (بوسیله @) و توسط ماده ESCAPE به عنوان کاراکترهای معمولی شناخته شدند.

روش دوم: Bracket

```
WHERE string_value LIKE '%[[[]abcde[_ ] ]%'
```

دو کاراکتر [و _ را داخل [] قرار دادیم.

روش سوم: CHARINDEX

```
WHERE CHARINDEX('[abcde_ ]', string_value) > 0
```

عبارت را بدون هیچ گونه تغییری در تابع استفاده کردیم. این روش تنها زمانی مورد استفاده می‌تواند قرار گیرد که کاراکتر % در ابتدای انتهایی (یا هر دو) آن قرار گرفته باشد.

تمرین: عباراتی را پیدا کنید که الگوی زیر در آن پیدا شود، فقط آخرین کاراکتر wildcard می‌باشد یعنی کاراکتر %

```
[[[%^ab'c'de]]%]
```

برای تست راه حلتان از متغیر جدولی زیر استفاده کنید:

```
declare @t table (s varchar(50))
insert @t values ('[[[%^ab'c'de]]%') ,
('[[[%^ab'cde]]%') , ('[[%^ab'c'de]]%') ,
('[[[%^abc'de]]%');
```


برخی از داده‌ها از ترکیب و ادغام شدن چند داده دیگر بدست می‌آیند. مثلاً شماره دانشجویی از ترکیب چند صفت مختلف بوجود می‌آید (مثل نیمسال و رودی، کددانشگاه، کدرشته تحصیلی...).

برای پیاده سازی اینگونه ستون‌ها SQL Server یک قابلیت به نام [computed column](#) ارائه داده است. برای تعریف این چنین ستون هایی بعد از نام ستون از کلمه AS استفاده می‌کنیم. عبارتی که ستون محاسباتی را تشکیل می‌دهد می‌تواند شامل این موارد باشد: تابع، نام ستون غیر محاسباتی و مقادیر ثابت ولی امکان استفاده از subquery وجود ندارد.

ستون‌های محاسباتی بطور پیشفرض مجازی هستند (بطور فیزیکی بر روی دیسک ذخیره نشده اند). یعنی هر موقع که query اجرا می‌شود آنها نیز مجدد محاسبه شده و نمایش داده می‌شوند.

برای اینکه نوع ذخیره سازی را از مجازی به فیزیکی تبدیل کنیم باید در هنگام ساخت جدول (یا تغییر آن) از کلید واژه PERSISTED استفاده کنیم. وقتی بطور فیزیکی ذخیره شده باشد با هر بار ویرایش یکی از ستون‌های تشكیل دهنده ستون محاسباتی هم ویرایش می‌شود.

ستون محاسباتی بعد از تبدیل شدن از مجازی به فیزیکی می‌تواند به عنوان کلید اولیه و ایندکس در نظر گرفته شود.

به مثال زیر توجه کنید:

جدولی داریم با دو ستون، قرار هست بر اساس ترکیب مقادیر دو ستون جستجویی انجام دهیم. ضمن اینکه ترکیب دو ستون باید منحصر بفرد باشد. برای این منظور یک unique index روی دو ستون لحاظ می‌کنیم.

```
create table t1
(
col1 char(1),
col2 char(1)
)

create unique nonclustered index ix_uq on t1 (col1 , col2);

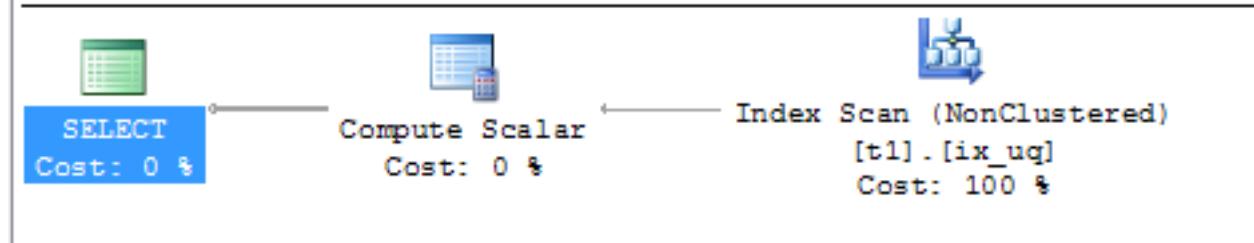
insert t1
values('A', 'B'), ('B', 'C'), ('C', 'D'), ('D', 'E'), ('E', 'F'),
('G', 'H'), ('I', 'J'), ('K', 'L'), ('M', 'N'), ('O', 'P');
```

اکنون به دنبال سطربی می‌گردیم که ترکیب مقادیر دو ستون آن برابر با OP باشد. پس query زیر را اجرا می‌کنیم

```
select col1 + col2
from t1
where col1 + col2 = 'OP'
```

اما همانطور که در تصویر زیر مشاهده می‌شود عمل Index Seek صورت نگرفته است. زمانی که از ستون به عنوان یک عبارت استفاده شود Index Seek نخواهیم داشت. منظور عبارت، الحق مقداری با ستون، قرار گرفتن ستون در یک تابع و ... می‌باشد.

```
Query 1: Query cost (relative to the batch) : 100%
SELECT [col1]+[col2] FROM [t1] WHERE ([col1]+[col2])=@1
```

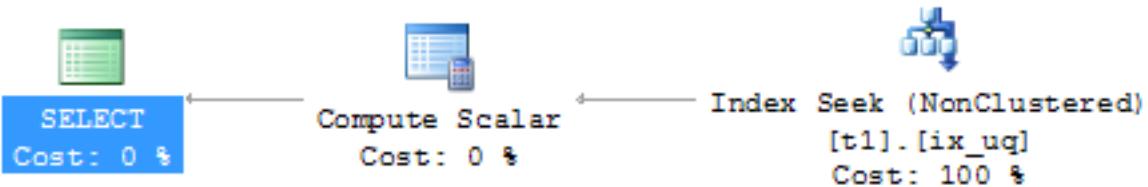


ستون محاسباتی (computed column)

برای اینکه Index Seek داشته باشیم بایستی مقادیر را جداگانه مقایسه کنیم (ستون‌ها به صورت عبارت محاسباتی نباشند)

```
select col1 + col2  
from t1  
where col1 = '0' and col2 = 'P'
```

Query 1: Query cost (relative to the batch) : 100%
SELECT [col1]+[col2] FROM [t1] WHERE [col1]=@1 AND [col2]=@2



ولی ما می‌خواهیم شرط بر اساس ترکیب دو ستون باشد. خب اینجا هست که Computed Columns مطرح می‌شوند.

```
alter table t1 add col3 as col1 + col2 persisted  
create clustered index ix1 on t1 (col3)
```

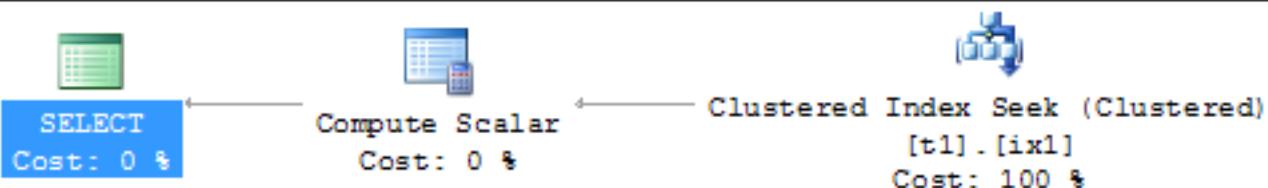
با دستور اول یک ستون محاسباتی از نوع persisted به جدول اضافه نمودیم. و با دستور دوم یک Index روی ستون محاسباتی ایجاد نمودیم.

حال مجدداً عمل جستجو را انجام میدهیم ولی به کمک ستون محاسباتی که اخیراً ایجاد نمودیم:

```
select *  
from t1  
where col3 = 'OP';
```

حال مشاهده می‌شود که شاخص ix1 اسکن نشده است. و از آنجایی که شاخص از نوع Clustered است مشکل هم نخواهیم داشت.

Query 1: Query cost (relative to the batch) : 100%
SELECT * FROM [t1] WHERE [col3]=@1



متاسفانه کاربران زیادی وجود دارند که هنوز درک صحیحی از جامعیت داده‌های ارجاعی (referential Integrity) ندارند. نمی‌دانند که relationship چیزی جز قید کلید خارجی (foreign key) نیست. در ادامه مفاهیم زیر را در حد آشنایی توضیح خواهیم داد:

کلید خارجی ترکیبی (composite foreign key)
 خود ارجاعی (self referencing)
 اعمال تغییرات به صورت آبشاری (cascade)

چندین مسیر برای اعمال (multiple cascading path)
 جدول اتصال (junction table)- ارتباط یک به یک

توسط دستور `create table` به دو شکل می‌توانیم بر روی ستون‌ها قید (کلید اولیه، check، کلید خارجی، کلید یونیک...) تعریف نمود:

قید ستونی
 قید جدولی

syntax مربوط به قید کلید خارجی در مدل ستونی به صورت زیر است:

```

<column_constraint> ::=  

[ CONSTRAINT constraint_name ]  

{ ...  

| [ FOREIGN KEY ]  

  REFERENCES [ schema_name . ] referenced_table_name [ ( ref_column ) ]  

  [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]  

  [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]  

  [ NOT FOR REPLICATION ]  

}  

...
  
```

نکته: بطور پیش فرض برای کلید خارجی اعمال update و delete روی وضعیت no action تنظیم شده است. به این معنا که اگر سعی کنیم کلید اولیه جدول مرجع را بروز رسانی یا حذف کنیم ممانعت به عمل خواهد آمد. برای رفع این مشکل هم می‌توانید از طریق design اقدام کنید و هم در هنگام ساخت جدول توسط DDL (همانطور که در دستورات فوق مشاهده می‌شود).

کلید خارجی ترکیبی

زمانی که در جدول والد (parent) کلید اولیه ترکیبی باشد، هر جدولی که بخواهد به کلید جدول والد ارجاعی داشته باشد باید از ترکیب دو ستون برای ساخت کلید خارجی استفاده کند.

فرض کنید جدول parent به این صورت است (ترکیب دو ستون col1 و col2 کلید اولیه است)

```

create table parent  

(  

col1 int not null,  

col2 int not null,  

col3 char(1) null,  

-- Composite Primary Key  

primary key(col1, col2)  

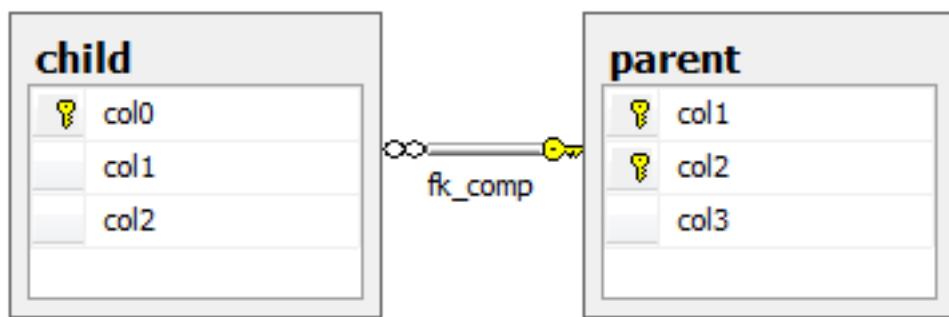
);
  
```

و جدول child که دارای قید کلید خارجی ترکیبی به نام fk_comp است و به جدول parent ارجاع داده است:

```
create table child
(
col0 int primary key,
col1 int null,
col2 int null,
-- Composite Foreign Key Constraint
constraint fk_comp
foreign key (col1, col2)
references parent(col1, col2)
);
```

در این DDL هم از قید جدولی برای تعریف کلید خارجی ترکیبی استفاده شده است.

نمودار این دو جدول:



پس به عنوان نتیجه گیری، هرگاه جدول اصلی دارای کلید ترکیبی بود در جداول child نیز باید از کلید خارجی ترکیبی برای ایجاد relationship استفاده نمود.

اما این دو جدول را به یک شیوه دیگر نیز می‌توان طراحی نمود. در جدول parent ترکیب دو ستون col1 و col2 را منحصر بفرد (unique) گرفته و ستونی دیگر (مثلاً از نوع identity) را به عنوان کلید اولیه در نظر گرفت (یا یک ستون از نوع محاسباتی تعريف کرده و آن را کلید قرار داد)

```
create table parent
(
col0 int not null primary key identity,
col1 int not null,
col2 int not null,
col3 char(1) null,
-- Composite Unique Key
unique(col1, col2)
);

create table child
(
col0 int primary key,
col1 int null references parent
);
```

خود ارجاعی و *multiple cascading path*

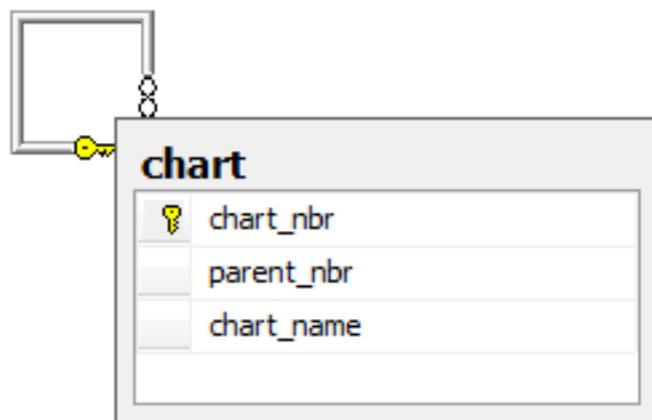
فرض کنید بخش‌های مختلف یک سازمان که بصورت چارت است را توسط جدول پیاده سازی کردیم. ستون‌های جدول به این شرح هستند:

کد بخش
نام بخش
کد بخش بالایی

ستون "کد بخش بالایی" نیز خود یک بخش است. برای پیاده سازی این چنین ساختارهایی از جدول زیر کمک گرفته می شود:

```
create table chart
(
chart_nbr int not null primary key,
parent_nbr int null references chart,
chart_name varchar(5) null
);
```

تصویر نمودار جدول chart



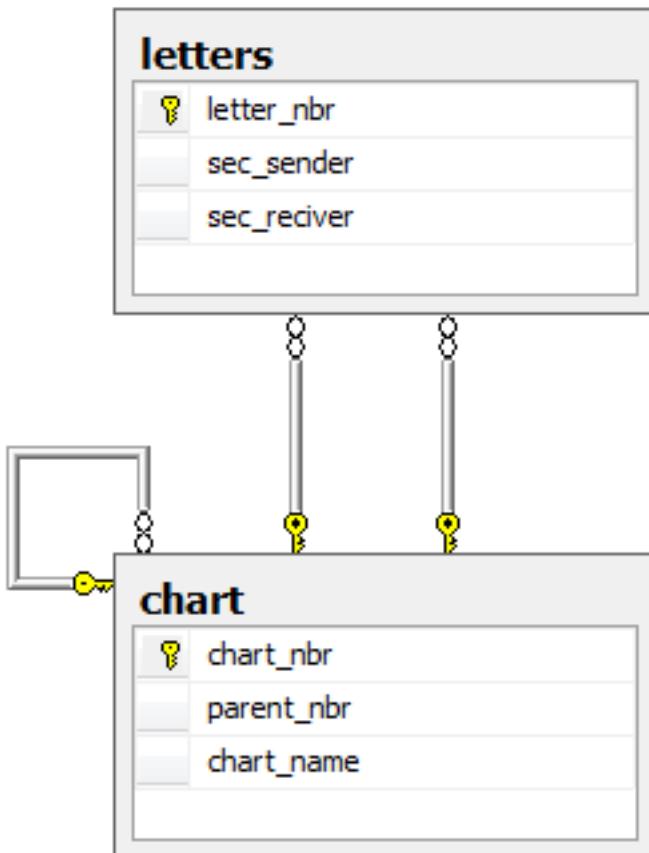
حالا فرض کنید میخواهیم اطلاعات نامه هایی که بین بخش ها رد و بدل میشود را در یک جدول ذخیره کنیم. جدول دارای ستون های زیر خواهد بود:

شماره نامه
کد بخش فرستنده
کد بخش گیرنده

ستون شماره نامه کلید اولیه و دو ستون دیگه کلیدهای خارجی هستند که به جدول chart مراجعه می کنند:

```
create table letters
(
letter_nbr int primary key,
sec_sender int not null references chart,
sec_reciever int not null references chart
);
```

نمودار جدول نامه ها و چارت:



نکته ای که در اینجا وجود دارد این است که اگر کلید جدول chart بروز شود آنگاه SQL Server از دو راه می‌تواند جدول letters را بروز رسانی کند، به این علت پیغام خطایی با عنوان multiple cascading paths صادر می‌شود. برای رفع این مشکل باید از trigger کمک گرفت.

جدول اتصال (junction table)

برای پیاده سازی رابطه N-N از جدول واسطه کمک گرفته می‌شود. برای این منظور رابطه N-N را باید به دو رابطه 1-N تجزیه کرد. فرض کنید یک جدول مربوط به خلبانان و جدول دیگر مربوط به مسیرهای پروازی (مثل مسیر ایران-ترکیه، ایران-عربستان...) است. یک خلبان ممکن است در چند مسیر پروازی هواپیما را هدایت کرده باشد و یا بالعکس یک مسیر پروازی ممکن است توسط N خلبان طی شده باشد.

برای پیاده سازی اینگونه سیستم هایی باید یک جدول ایجاد نمود که دارای دو کلید خارجی باشد یکی آنها به جدول خلبانان و دیگری به مسیرهای پروازی مرتبط است.

می‌توان ترکیب دو کلید خارجی جدول واسطه را کلید اولیه در نظر گرفت.
پس خواهیم داشت:

```

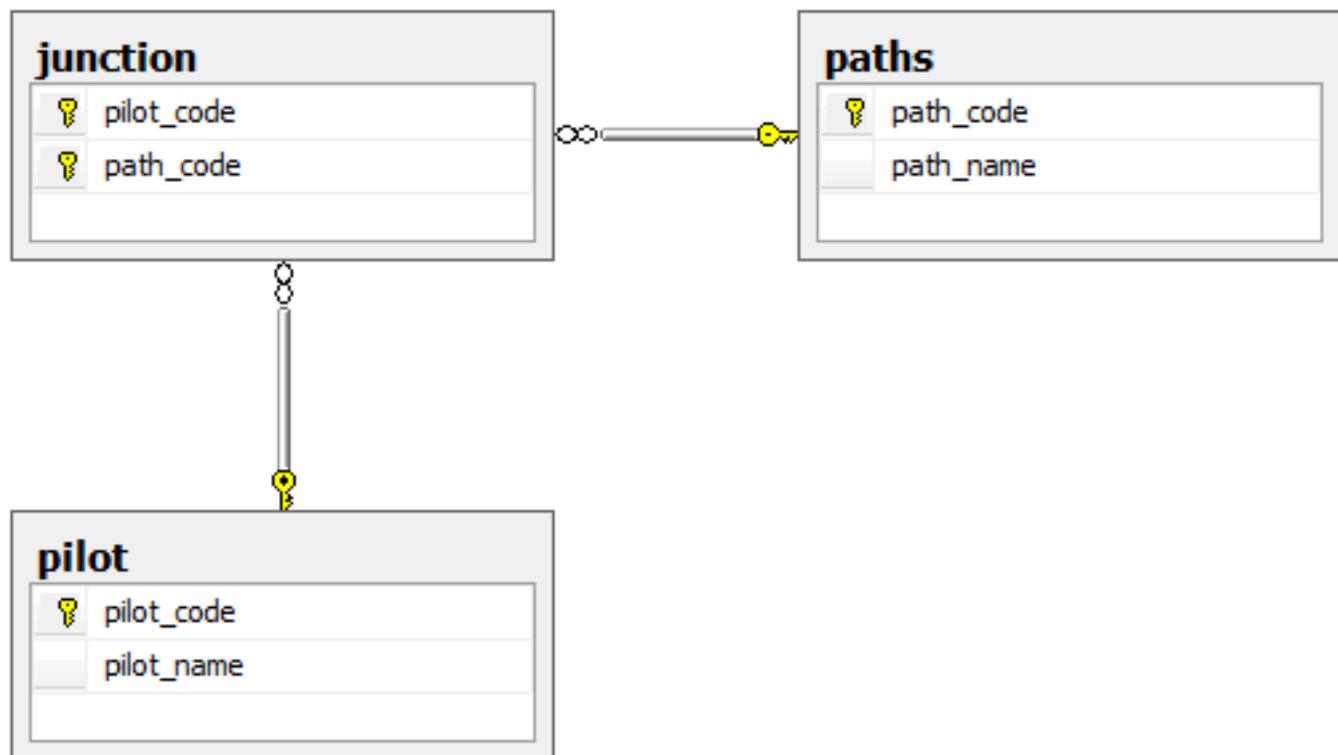
create table pilot
(
pilot_code int primary key,
pilot_name varchar(20)
);

create table paths
(
path_code int primary key,
path_name varchar(20)
);

```

```
create table junction
(
pilot_code int references pilot,
path_code int references paths,
primary key (pilot_code, path_code)
);
```

و نمودار آن:



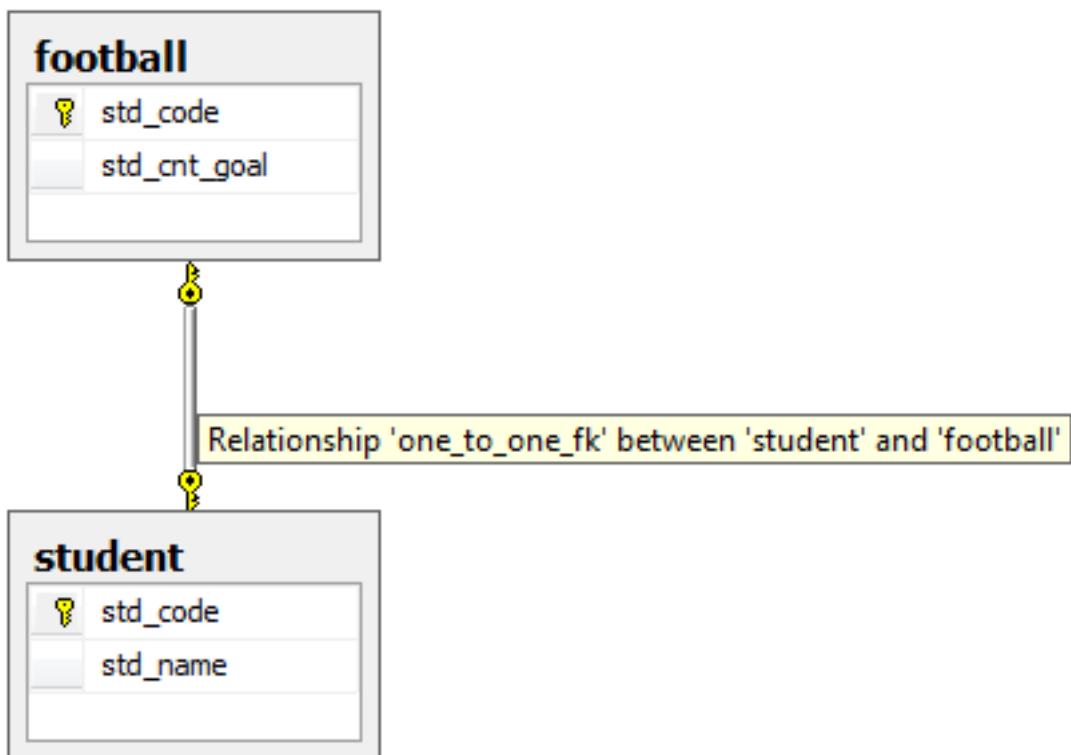
رابطه یک به یک

زمانی که نمونه‌های محدودی از یک موجودیت دارای مقدار برای یکسری خصیصه هستند بهتر است جدول به دو جدول تجزیه شود تا فضای اضافی صرف جدول نشود. مثلا در مدرسه تنها 10 درصد دانش آموزان جزء تیم فوتبال هستند حال اگر بخواهیم اطلاعات مربوط به تیم فوتبال مثل تعداد گل زده، تعداد بازی ... در جدول اصلی ذخیره کنیم برای 90 درصد دانش آموزان مقداری نخواهیم داشت. برای حل این مساله ارتباط یک به یک پیشنهاد می‌شود.

```
create table student
(
std_code int primary key,
std_name varchar(25) not null
);

create table football
(
std_code int primary key
constraint one_to_one_fk
references student,
std_cnt_goal int not null
default (0)
);
```

توجه داشته باشید که ستون std_code هم کلید اولیه هست و هم کلید خارجی که به جدول student ارجاع داده شده است.

**نتیجه گیری**

یک ستون همزمان می‌تواند کلید اولیه باشد و هم کلید خارجی (مثلا در ارتباط یک به یک) همانطور که کلید اولیه ترکیبی داریم به همان شکل هم کلید خارجی ترکیبی داریم. یک جدول می‌تواند به خودش ارجاع دهد که به آن اصطلاحا self-referencing relationship چیزی جز کلید خارجی نیست و کلید خارجی نیز چیزی جز یک قید برای جامعیت داده‌ها نیست جامعیت داده ارجاعی را می‌توان توسط trigger پیاده سازی کرد اگر SQL Server بیش از یک مسیر برای تغییر جدول child داشته باشد با مشکل مواجه خواهد شد

نظرات خوانندگان

نویسنده: Mohsen
تاریخ: ۱۳۹۱/۱۱/۰۷ ۸:۱۵

بسیار عالی بود. مفاهیم پایگاه داده برای جامعه نرم افزار ایران بسیار حیاتی هستند.

نویسنده: mohammad
تاریخ: ۱۳۹۱/۱۱/۰۸ ۲۰:۴۰

با سلام. ممنون از مطلبتون. لطفا توضیحی راجع به اینکه چطور میتوان با دادن یک `id` تمام ریشه‌های اون رو در حالت خودارجاعی درآورد بدید.

نویسنده: mohammad
تاریخ: ۱۳۹۱/۱۱/۰۸ ۲۰:۵۵

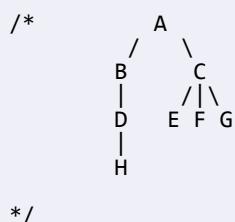
ببخشید منظورم برگ‌ها بود که اشتباه نوشتم ریشه. با تشکر

نویسنده: محمد سلم ابادی
تاریخ: ۱۳۹۱/۱۱/۰۸ ۲۱:۵۶

سلام،

برای این منظور باید از recursive cte کمک گرفت.

فرض کنید درختی به شکل زیر داریم:



و هدف بdst آوردن تمام زیر شاخته‌های گره A است.

ابتدا باید تمام گره‌هایی که مقدار گره پدرشان برابر با A است را بdst بیاریم یعنی گره‌های B و C

حالا باید تمام گره‌هایی که گره پدرشان B و یا C است را بdst بیاریم یعنی گره‌های D E F G

و در مرحله بعد باید تمام گره‌هایی را بdst بیاریم که گره پدرشان برابر با یکی از مقادیر بdst آمده در مرحله قبل (یعنی D E F) یعنی G

این الگوریتم را توسط Recursive CTE پیاده میکنیم:

```

declare @t table
(
  id char(1) primary key not null,
  pid char(1) null --references @t
);
insert @t
values ('A', null), ('B', 'A'), ('C', 'A'),
('D', 'B'), ('H', 'D'), ('E', 'C'), ('F', 'C'), ('G', 'C');

with cte as
(
  select id
  from @t
  where pid = 'A'
)

```

گذری بر مفاهیم relationship

```
union all  
select t.id  
from cte c  
join @t t  
on t.pid = c.id  
)  
select * from cte
```

موفق باشید

نوبسنده: mohammad
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۰:۵۳

ممnon خیلی هم خوب.
اما یک سوال؟ آیا این روش فشار زیادی روی سرور ایجاد نمیکنه. مثلا در یک برنامه تحت وب با تعداد زیاد کاربر؟

نوبسنده: محمد سلم ابادی
تاریخ: ۱۳۹۱/۱۱/۰۹ ۱۲:۱۵

این یک روش معمولی و رایج هست که همگان دارن ازش استفاده میکنند. بعید میدانم که در عملکرد مشکلی داشته باشد.
اگر فرضا سرعت اجرای query ا atan پایین بود یک اندیس (index) روی ستون "کد والد" تعریف کنید
به این شکل

```
create nonclustered index ix_parent on table_name (parent_id)
```

مقدمه

در [قسمت پیشین](#) نشان داده شد که چگونه کاراکترهای خارج از رنج حروف الفبا ای انگلیسی از عبارات موجود در یک جدول حذف شدند.

اکنون شرایط کمی تغییر کرده است کاراکترهای ناخواسته در قالب یک مجموعه (جدول) به ما ارائه داده می‌شوند. ما بایستی تمام کاراکترهای داده شده را از عبارات (موجود در جدول) در صورت تطابق حذف کنیم.

جدول کاراکترهای ناخواسته Unwanted و جدول داده‌ها Data نامگذاری شده اند.

```
CREATE TABLE Data
(
id INTEGER NOT NULL PRIMARY KEY IDENTITY,
data VARCHAR(50) NOT NULL
);

INSERT INTO Data
VALUES ('~!hasan @#$%^&*(reza)[ali]^^^^^^^^^'),
('Ja[][][]va~!@#$*d-mohammad'),
('Mohammad'), ('Maryam');

CREATE TABLE Unwanted
(
id INT NOT NULL PRIMARY KEY IDENTITY,
chars CHAR(1) NOT NULL UNIQUE
);

INSERT Unwanted
VALUES ('~'), ('!'), ('@'), ('#'), ('$'), ('%'),
('['), (']'), ('^'), ('&'), ('*');
```

الگوریتم

قبل از هر چیزی فرض می‌گیریم که داده‌ها ستون Id جدول Unwanted کاملاً متوالی نیستند و gap بین مقادیر id وجود دارد. پس برای داشتن مقادیر متوالی از تابع row_number استفاده شده است.

بعد از آن باید عباراتی که حداقل با یک کاراکتر ناخواسته تطابق پیدا می‌کنند انتخاب شوند. و عملیات پاکسازی روی این عبارات به تعداد کاراکترهای موجود در جدول unwanted انجام می‌شود. یعنی در مرحله اول شمارنده برابر است با تعداد کاراکترهای ناخواسته سپس در هر فراخوانی یک مقدار از این شمارنده کم شده تا اینکه به 0 برسد در اینجا کار به اتمام می‌رسد.

بعد از پایان یافتن فراخوانی‌های query بازگشتی باید سطرهایی که شمارنده آنها برابر با 0 است انتخاب شده و عباراتی که در مرحله اول به دلیل عدم تطابق انتخاب نشدن بایستی به نتیجه اضافه شوند. این کار را توسط Union all انجام شده است.

```
WITH pre AS
(
SELECT ROW_NUMBER() OVER(ORDER BY id) AS id, chars
    FROM Unwanted
),
cte AS
(
SELECT data.id,
       nbr,
       CAST(Data AS VARCHAR(50)) AS Data
  FROM Data
 CROSS JOIN (SELECT COUNT(*) AS nbr FROM Unwanted)t
 WHERE EXISTS
       (SELECT *
          FROM Unwanted
         WHERE Data LIKE '%#' + chars + '%' ESCAPE '#')

UNION ALL

SELECT C.id,
       nbr - 1,
       CAST(REPLACE(C.data, U.chars, '') AS VARCHAR(50))
  FROM cte AS C
         JOIN Unwanted U ON C.id = U.id;
```

```
    INNER JOIN pre U
    ON C.nbr = U.id
    WHERE nbr > 0
)
SELECT data
  FROM cte
 WHERE nbr = 0

UNION ALL

SELECT data
  FROM Data
 WHERE NOT EXISTS
 (SELECT id
   FROM cte
  WHERE cte.id = data.id);
```

مربوط به SQL Server 2008 از کار افتاده بود و راه اندازی نمی‌شد. خطای مرتبط با آن در لاغ‌های ویندوز به نحو زیر بود:

SQLServerAgent could not be started (reason: Unable to connect to server '(local)'; SQLServerAgent cannot start).

پس از مدتی جستجو، عنوان شده بود که مسیر رجیستری زیر را یافته:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL10.MSSQLSERVER\SQLServerAgent

و در آن ServerHost را به نام سرور ویرایش کنید. سپس سرور را ری استارت نمایید. این تغییر انجام شد اما باز هم SQL Server راه اندازی نمی‌شد.

لاغ‌های آن در مسیر ذیل ثبت می‌شوند:

C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\Log\SQLAGENT.OUT

در اینجا خطاهای زیر ثبت شده بودند:

```
2013-01-28 20:02:34 - ! [298] SQLServer Error: 780, SQL Server Network Interfaces: The logon attempt failed [SQLSTATE HY000]
2013-01-28 20:02:34 - ! [298] SQLServer Error: 780, Cannot generate SSPI context [SQLSTATE HY000]
2013-01-28 20:02:34 - ! [000] Unable to connect to server ; SQLServerAgent cannot start
2013-01-28 20:02:34 - ! [298] SQLServer Error: 780, SQL Server Network Interfaces: The logon attempt failed [SQLSTATE HY000]
2013-01-28 20:02:34 - ! [298] SQLServer Error: 780, Cannot generate SSPI context [SQLSTATE HY000]
2013-01-28 20:02:34 - ! [382] Logon to server failed (DisableAgentXPs)
2013-01-28 20:02:34 - ? [098] SQLServerAgent terminated (normally)
```

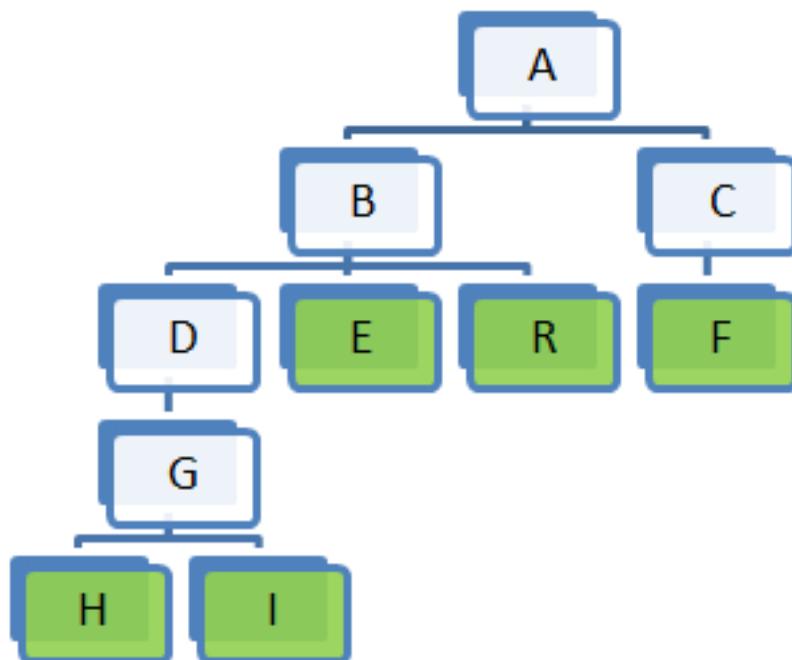
همانطور که مشخص است مشکل اصلی در عدم توانایی لاگین اکانت SQL Server Agent ذکر شده است.
برای تغییر آن مسیر زیر را طی کنید:

SQL Configuration manager -> SQL Server Agent -> Logon User -> NT/Local Service

به عبارتی در قسمت Local Service، تنظیمات SQL Server Agent را یافته و نوع اکانت آن را به تغییر دهید.
پس از آن سرویس بدون مشکل استارت شد.

امروز در یک تالار سوالی مطرح شد با این عنوان "چگونه می‌توانم گره‌های برگ یک شاخه را بدست بیاورم". خب راه حلی که فوراً به ذهنم رسید استفاده از یک query بازگشتی (recursive) بود.

به ساختار سلسله مراتبی زیر توجه بفرمایید:



گره‌هایی که با رنگ سبز علامت گذاری شده اند را گره‌های برگ می‌نامیم چون که آن گره‌ها بدون زیر شاخه هستند. فرض کنید از ما خواسته شده است با داشتن گره A تمام گره‌های این شاخه را بدست بیاوریم.

دو مرحله را باید طی کنیم ابتدا تمام گره‌هایی که زیر شاخه گره A هستند را باید بدست آورد سپس توسط یک گزاره گره‌های برگ را فیلتر کنیم.

در واقع گره‌هایی برگ هستند که پدر هیچ گره‌ی دیگری نباشند.

```

declare @t table
(id char(1) primary key,
parent char(1));

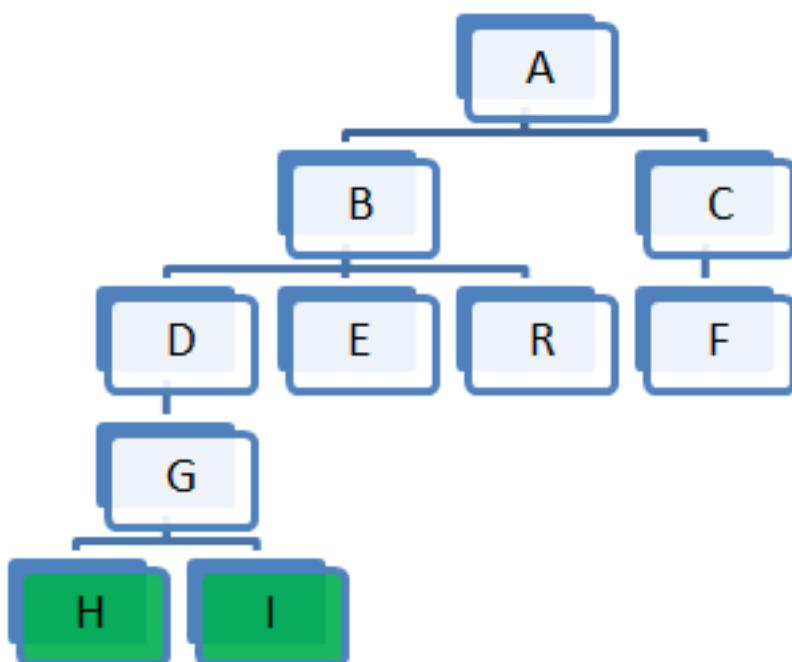
insert @t values
('A',null),                                --Level 1
('B', 'A'), ('C', 'A'),                      --Level 2
('D', 'B'), ('E', 'B'), ('R', 'B'),          --Level 3
('G', 'D'), ('H', 'G'), ('I', 'G');          --Level 4
                                         --Level 5
  
```

```

;with cte as
(
select id, rnk=0 from @t
where parent = 'A'
  
```

```
union all
select t.id, rnk+1
from cte join @t t
on cte.id = t.parent
)
select *
from cte
where not exists
(select *
from @t
where parent = cte.id);
```

و حالا به درخت زیر توجه بفرمایید:



هدف پیدا کردن برگ هایی از شاخه مورد نظر است که در پایینترین سطح قرار گرفته باشند. برای این منظور از همان query بازگشتی استفاده کرده و با کمک تابع `dense_rank()` گرههای مورد نظر را بدست میاوریم.

```
;with cte as
(
select id, rnk=0 from @t
where parent = 'A'

union all

select t.id, rnk+1
from cte join @t t
on cte.id = t.parent
)
select *
from
(
  select *, dense_rank() over(order by rnk desc) rk
  from cte
)t
where rk = 1
```


نظرات خوانندگان

نویسنده: بهمن خلفی
تاریخ: ۱۴:۴۱ ۱۳۹۱/۱۱/۱۷

ممنون از مطلب مفیدتون

اما یک سوال : چگونه میتوان به نتیجه زیر دست پیدا کرد

a , ab , ac , abd , abe , abr , abdg , abdgh ,

نویسنده: محمد سلم ابادی
تاریخ: ۱۵:۳۰ ۱۳۹۱/۱۱/۱۷

من دقیقا متوجه نشدم نتیجه مورد نظر شما چیست.

آیا نتیجه مورد نظر شما به صورت الحاق یافته (concatenated) هست یا نه؟
در هر صورت باید یکی از دو query زیر نتیجه مورد نظر شما را تولید کند.

```
declare @t table
(id char(1) primary key,
parent char(1));

insert @t values
('A',null),                                --Level 1
('B', 'A'), ('C', 'A'),                      --Level 2
('D', 'B'), ('E', 'B'), ('R','B'), ('F', 'C'), --Level 3
('G', 'D'),                                --Level 4
('H', 'G'), ('I', 'G');                     --Level 5

;with cte as
(
select id, rnk=0,
       concats = cast(id as varchar(10))
from @t
where parent is null

union all

select t.id, rnk+1,
       cast(cte.concats + t.id as varchar(10))
from cte join @t t
on cte.id = t.parent
)
select * from cte
/*
id    rnk      concats
----- -----
A     0        A
B     1        AB
C     1        AC
F     2        ACF
D     2        ABD
E     2        ABE
R     2        ABR
G     3        ABDG
H     4        ABDGH
I     4        ABDGI
*/
;with cte as
(
select id, rnk=0,
       concats = cast(id as varchar(10))
from @t
where parent is null

union all

select t.id, rnk+1,
       cast(cte.concats + t.id as varchar(10))
from cte join @t t
on cte.id = t.parent
)
```

بدست آوردن برگهای یک درخت توسط Recursive CTE

```
select stuff(d.list,1,1,'') as concats
from (select ','+concats
      from cte
     for xml path(''))d(list)
/*
concats
-----
A,AB,AC,ACF,ABD,ABE,ABR,ABDG,ABDGH,ABDGI
*/
```

موفق باشید

مقدمه (شرح مساله)

چندی پیش در تالار T-SQL سوالی مطرح شد راجع به مساله ای که معروف است به `.top N per group` که مساله ای است که بمعنی مجموعه ای از نتایج است که در گروه های مخصوص محدودیتی بود که کاربر مورد نظر داشت؛ که آن محدودیت چیزی نبود جز query بایستی در نسخه 2000 جوابگو باشد.

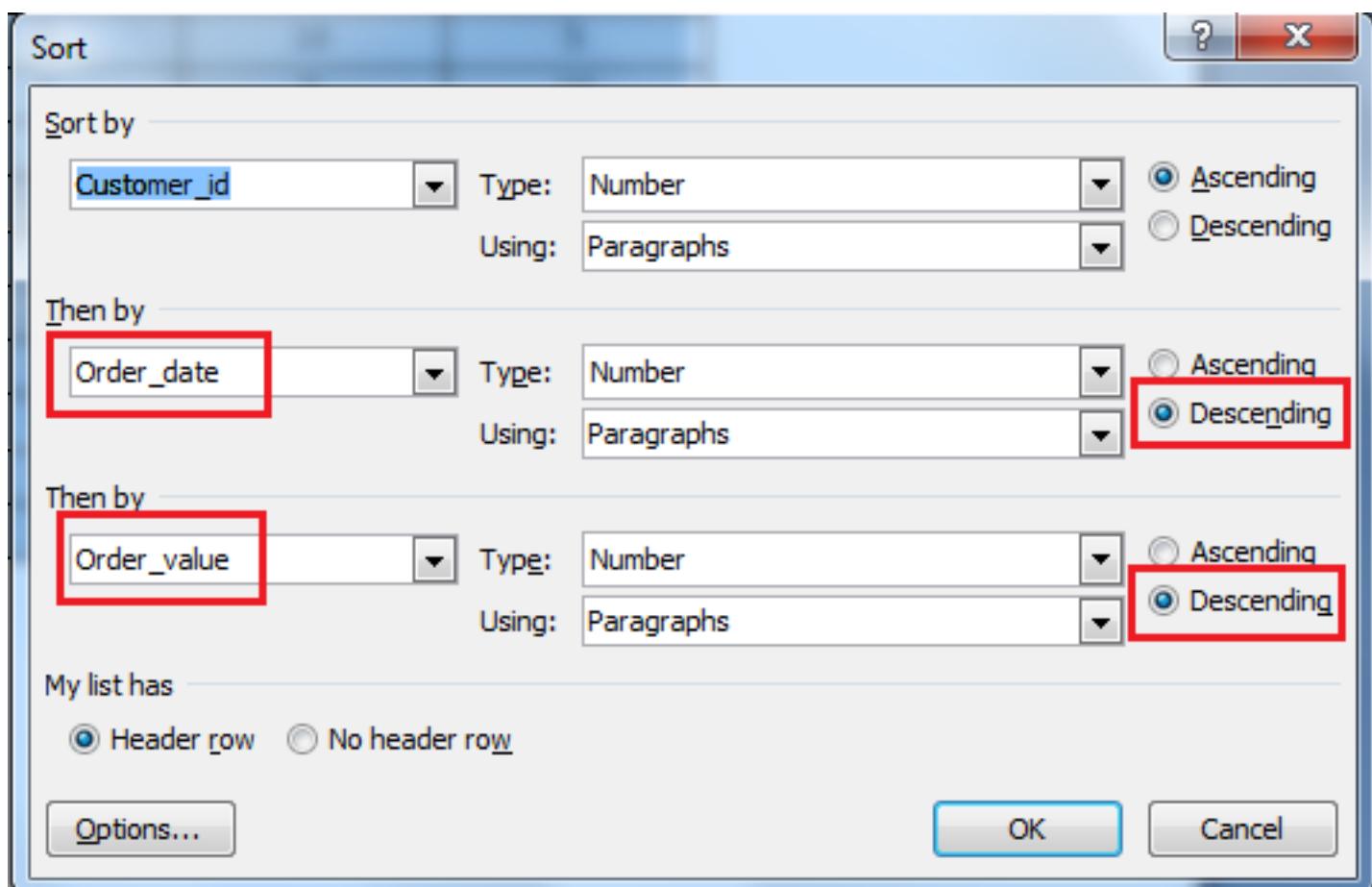
قطعاً شده است که بخواهید مثلاً به ازای هر مشتری آخرین سفارش آن را انتخاب کنید. این مساله Top N نامیده می‌شود.

فرض کنید جدولی داریم که حاوی سفارشات مشتریان می‌باشد. هر مشتری می‌تواند چندین سفارش داشته باشد؛ هر سفارش دارای حداقل دو مقدار "تاریخ سفارش" و "مبلغ سفارش" است. هدف پیدا کردن آخرین سفارشات هر مشتری می‌باشد. نکته: اگر چند تاریخ برای آخرین سفارش مشتری وجود داشت آنگاه بایستی بر اساس مبلغ سفارش مرتب سازی نزولی صورت بگیرد. یا به عبارت دیگر ابتدا باید مرتب سازی نزولی بر اساس ستون تاریخ سفارش انجام شود و سپس مرتب سازی نزولی بر اساس ستون مبلغ سفارش.

فرض می‌گیریم داده‌های جدول ما چیزیست شبیه به این:

Row_id	Customer_id	Order_date	Order_value
6	1	10	15
9	1	10	5
4	1	9	10
3	1	8	20
10	2	12	15
7	2	10	15
11	3	15	5
5	3	10	20
8	3	10	10
1	3	5	15
2	3	5	15

سطرهایی از جدول که رنگی شده اند سطرهای مورد نظر ما هستند که باید در خروجی ظاهر شوند. داده‌های جدول با کمک قابلیت Sort نرم افزار word مرتب سازی شده اند، این تصویر را به این خاطر در اینجا قرار دادم چون که دیدم می‌تواند در شفاف سازی مساله به من کمک کند. ابتدا مرتب سازی نزولی بر اساس ستون `order_date` و `order_value` در پایان اولین سطر مربوط به هر مشتری به عنوان خروجی مورد نظر انتخاب می‌شوند.



راه حل ها

خب پر واضح است که در نسخه 2005 و بعد از آن ساده ترین و بهینه ترین راه حل استفاده از تابع row_number می باشد.

```
SELECT row_id, customer_id, order_date, order_value
  FROM (SELECT *,
               ROW_NUMBER() OVER(PARTITION BY customer_id
                                 ORDER BY order_date DESC, order_value DESC) AS rnk
         FROM table_name
        )t
 WHERE rnk = 1;
```

اما با محدودیتی که در نسخه 2000 وجود دارد راه حلی بهتر از این پیدا نخواهیم کرد:

```
SELECT *
  FROM table_name t
 WHERE row_id = (SELECT TOP 1 row_id
                  FROM table_name
                 WHERE customer_id = t.customer_id
                 ORDER BY order_date DESC, order_value DESC);
```

حالا چه می شود راه حلی بخواهیم مستقل از هر یک از نسخه های SQL Server

```
SELECT MIN(row_id) AS row_id, customer_id, order_date, order_value
  FROM table_name t
 WHERE order_date =
```

```
(SELECT MAX(order_date)
   FROM table_name
  WHERE customer_id = t.customer_id)
AND order_value =
(SELECT MAX(order_value)
   FROM table_name
  WHERE customer_id = t.customer_id
    AND order_date =
(SELECT MAX(order_date)
   FROM table_name
  WHERE customer_id = t.customer_id))
 GROUP BY customer_id, order_date, order_value;
```

SQL Server قابلیت فعال(enable) و غیر فعال(disable) کردن دو قید کلید خارجی و check را برای ما مهیا کرده است. ما می‌توانیم بعد از ساخت جدول و انتشار مقداری داده در آن قیدهایی را ایجاد کنیم. بطور پیشفرض اگر شرط قید ما برقرار بود قید به طور صحیح ساخته می‌شود و اگر شرط قید ما برقرار نباشد قید با خطای conflict مواجه خواهد شد. بطور کلی غیر فعال کردن قیدها کار درستی نیست. ولی در برخی مواقع برای تسريع در اجرای کد می‌توانیم قید را غیر فعال کنیم. بطور مثال اگر یک میلیون داده قرار است در جدول درج شود و مطمئن هستیم که این داده‌ها جامعیت داده‌ها را حفظ می‌کنند آنگاه می‌توانیم قید را برای تسريع در عمل درج بطور موفق غیر فعال کنیم.

فعال و غیر فعال کردن از طریق DDL

با غیر فعال کردن قیود داده‌ها را در وضعیت نامناسبی قرار می‌دهیم ولی همان طور که اشاره شد بطور موفق اشکالی پیش نخواهد آمد.

در ادامه ابتدا طریقه غیرفعال کردن و مجدداً فعال کردن قیود را توسط دستور alter table نشان خواهیم داد سپس به سراغ امکانات ویزاردی می‌رویم. ابتدا یک جدول تک ستونه ایجاد می‌کنیم:

```
CREATE TABLE testTable  
(column1 integer not null);
```

الان هیچ قیدی روی جدول لحاظ نشده است. پس هر داده که در رنج domain ستون باشد را می‌توانیم درج کنیم. پس بطور نمونه این داده‌ها را درج می‌کنیم:

```
INSERT INTO testTable  
VALUES (-10), (0), (10), (20), (30), (40)
```

حالا تصمیم داریم قیدی روی ستون column1 بگذاریم که توسط آن تنها اعداد مثبت در جدول درج شوند. پس داریم:

```
ALTER TABLE testTable  
WITH CHECK  
ADD CONSTRAINT NoNegative  
CHECK (column1 > 0);
```

ولی چون داده‌هایی در جدول از قبل وجود داشته اند که قید ما را نقض کنند این قید ساخته نخواهد شد و با پیغام زیر مواجه خواهیم شد:

```
The ALTER TABLE statement conflicted with the CHECK constraint "NoNegative".
```

برای ساخت این قید روی این داده‌ها تنها راه استفاده از کلید واژه‌های WITH NOCHECK است یعنی:

```
ALTER TABLE testTable  
WITH NOCHECK  
ADD CONSTRAINT NoNegative  
CHECK (column1 > 0);
```

و اکنون سعی می‌کنیم یک مقدار منفی در جدول درج کنیم:

فعال و غیر فعال کردن قیود

```
INSERT INTO testTable VALUES (-5)
/*
The INSERT statement conflicted with the CHECK constraint "NoNegative".
*/
```

اما قیدی که ساخته بودیم در جدول در حال اعمال شدن است. برای درج مقدار منفی باید غیر را غیر فعال کنیم.

```
ALTER TABLE TestTable
NOCHECK CONSTRAINT NoNegative
```

و حالا مقدار منفی را درج میکنیم. و برای برگرداندن وضعیت CHECK به وضعیت NOCHECK باید از کلید واژه های استفاده کنیم. چرا که داده هایی در جدول درج شده اند که قید مورد نظر ما را نقض می کنند.

```
ALTER TABLE TestTable
WITH NOCHECK
CHECK CONSTRAINT NoNegative
```

فعال و غیر فعال کردن از طریق design

در قسمت object explorer قید مورد نظر را پیدا کرده و روی آن راست کلیک کرده و گزینه Modify را انتخاب کنید. سپس در پنجره باز شده در قسمت Table Designer تغییرات مورد نظر خود را اعمال کنید.

اس کیوال سرور، از سال 2005 به بعد، به صورت توکار امکان تعریف و ذخیره سازی اطلاعات schema free و یا schema less را به کمک فیلدهایی از نوع XML ارائه داده است؛ به همراه یکپارچگی آن با زبان XQuery برای تهیه کوئری‌های سریع سمت سرور. در فیلدهای XML می‌توان اطلاعات انواع و اقسام اشیاء را بدون اینکه نیازی به تعریف تک تک فیلدهای مورد نیاز، در بانک اطلاعاتی وجود داشته باشد، ذخیره کرد. یک نمونه از کاربرد چنین امکانی، نوشتن برنامه‌های «فرم ساز» است. برنامه‌هایی که کاربران آن می‌توانند فیلد اضافه و کم کرده و نهایتاً اطلاعات را ذخیره و از آن‌ها کوئری بگیرند.

خوب، این فیلد کمتر بحث شده XML، فقط در اس کیوال سرور و نگارش‌های اخیر آن وجود دارد. اگر نیاز به کار با بانک‌های اطلاعاتی سبکتری وجود داشت چطور؟ یک راه حل عمومی برای این مساله مراجعه به روش‌های NoSQL است. یعنی بطور کلی بانک‌های اطلاعاتی رابطه‌ای کنار گذاشته شده و به یک سکوی کاری دیگر سوچیج کرد. در این بین، [SisoDb](#) راه حل میانه‌ای را ارائه داده است. با کمک SisoDb می‌توان اطلاعات را به صورت schema less و بدون نیاز به تعریف فیلدهای متناظر آن‌ها، در انواع و اقسام بانک‌های اطلاعاتی JSON ذخیره و بازیابی کرد. این انواع و اقسام، شامل SQL Server CE نیز می‌شود.

دریافت و نصب SisoDb

دریافت و نصب [SisoDb](#) بسیار ساده است. به کمک package manager NuGet و امکانات Sisodb، کلمه manager را جستجو کنید. در بین مداخل ظاهر شده، پرواایدر مورد علاقه خود را انتخاب و نصب نمائید. برای مثال اگر قصد دارید با SQL Server CE کار کنید، SisoDb.SqlCe4 را انتخاب و یا اگر SQL Server 2008 مدنظر شما است، SisoDb.Sql12008 را انتخاب و نصب نمائید.

ثبت و بازیابی اطلاعات به کمک SisoDb

کار با SisoDb بسیار روان است. نیازی به تعاریف نگاشت‌ها و ORM خاصی نیست. یک مثال مقدماتی آن را در ادامه ملاحظه می‌کنید:

```

using SisoDb.Sql2008;

namespace SisoDbTests
{
  public class Customer
  {
    public int Id { get; set; }
    public int CustomerNo { get; set; }
    public string Name { get; set; }
  }

  class Program
  {
    static void Main(string[] args)
    {
      /*var cnInfo = new SqlCe4ConnectionStringInfo(@"Data source=sisodb2013.sdf;");
      var db = new SqlCe4DbFactory().CreateDatabase(cnInfo);
      db.EnsureNewDatabase();*/

      var cnInfo = new Sql2008ConnectionStringInfo(@"Data Source=(local);Initial Catalog=sisodb2013;Integrated Security = true");
      var db = new Sql2008DbFactory().CreateDatabase(cnInfo);
      db.EnsureNewDatabase();

      var customer = new Customer
      {
        CustomerNo = 20,
        Name = "Vahid"
      };
      db.UseOnceTo().Insert(customer);

      using (var session = db.BeginSession())
      {
        var info = session.Query<Customer>().Where(c => c.CustomerNo == 20).FirstOrDefault();
        var info2 = session.Query<Customer>().Where(c => c.CustomerNo == 20 &&
c.Name=="Vahid").FirstOrDefault();
      }
    }
  }
}
  
```

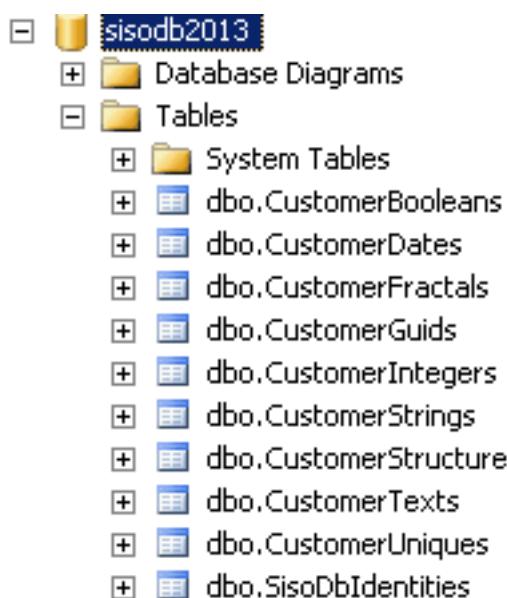
```

    }
}
}
}
```

در این مثال، ابتدا اتصال به بانک اطلاعاتی برقرار شده و سپس بانک اطلاعاتی جدید تهیه می‌شود. سپس یک وله از شیء مشتری ایجاد و ذخیره می‌گردد. در ادامه دو کوئری بر روی بانک اطلاعاتی انجام شده است.

SisoDb داخلی ساختار

SisoDb به ازای هر کلاس، حداقل 9 جدول را ایجاد می‌کند. در ادامه نحوه ذخیره سازی شیء مشتری ایجاد شده و مقادیر خواص آنرا نیز مشاهده می‌نمایید:



```

1 SELECT TOP 1000 [RowId]
2      , [StructureId]
3      , [MemberPath]
4      , [Value]
5      , [StringValue]
6   FROM [sisodb2013].[dbo].[CustomerIntegers]
```

	RowId	StructureId	MemberPath	Value	StringValue
1	1	1	Id	1	1
2	2	1	CustomerNo	20	20

ذخیره سازی جداگانه خواص عددی

```

1 | SELECT TOP 1000 [RowId]
2 |     , [StructureId]
3 |     , [MemberPath]
4 |     , [Value]
5 | FROM [sisodb2013].[dbo].[CustomerStrings]

```

	RowId	StructureId	MemberPath	Value
1	1	1	Name	Vahid

ذخیره سازی جداگانه خواص رشته‌ای

```

1 | SELECT TOP 1000 [RowId]
2 |     , [StructureId]
3 |     , [Json]
4 | FROM [sisodb2013].[dbo].[CustomerStructure]

```

	RowId	StructureId	Json
1	1	1	{"Id":1,"CustomerNo":20,"Name":"Vahid"}

ذخیره سازی کلی شیء مشتری با فرمت JSON به صورت یک رشته

```

1 SELECT TOP 1000 [EntityName]
2      , [CurrentId]
3  FROM [sisodbd2013].[dbo].[SisoDbIdentities]

```

	EntityName	CurrentId
1	Customer	1

همانطور که ملاحظه می‌کنید، یک جدول کلی SisoDbIdentities ایجاد شده است که اطلاعات نام اشیاء را در خود نگهداری می‌کند. سپس اطلاعات خواص اشیاء یکبار به صورت JSON ذخیره می‌شوند؛ با تمام اطلاعات تو در توی ذخیره شده در آن‌ها و همچنین یکبار هم هر خاصیت را به صورت یک رکورد جداگانه، بر اساس نوع کلی آن‌ها، در جداول رشته‌ای، عددی و امثال آن ذخیره می‌کند.

شاید بپرسید که چرا به همان فیلد JSON اکتفاء نشده است؟ از این جهت که پردازشگر سمت بانک اطلاعاتی آن همانند فیلد‌های XML در SQL Server و نگارش‌های مختلف آن وجود ندارد (برای مثال به کمک زبان T-SQL می‌توان از زبان XQuery در خود بانک اطلاعاتی، بدون نیاز به واکنش کل اطلاعات در سمت کلاینت، به صورت یکپارچه استفاده کرد). به همین جهت برای کوئری گرفتن و یا تهیه ایندکس، نیاز است این موارد جداگانه ذخیره شوند.

به این ترتیب زمانیکه کوئری تهیه می‌شود، برای مثال:

```
var info = session.Query<Customer>().Where(c => c.CustomerNo == 20).FirstOrDefault();
```

به کوئری زیر ترجمه می‌گردد:

```

SELECT DISTINCT TOP(1) (s.[StructureId]),
               s.[Json]
  FROM  [CustomerStructure] s
  LEFT JOIN [CustomerIntegers] mem0
    ON mem0.[StructureId] = s.[StructureId]
    AND mem0.[MemberPath] = 'CustomerNo'
 WHERE  (mem0.[Value] = 20);

```

و یا کوئری ذیل:

```
var info2 = session.Query<Customer>().Where(c => c.CustomerNo == 20 &&
c.Name=="Vahid").FirstOrDefault();
```

معادل زیر را خواهد داشت:

```

SELECT DISTINCT TOP(1) (s.[StructureId]),
               s.[Json]
  FROM  [CustomerStructure] s
  LEFT JOIN [CustomerIntegers] mem0
    ON mem0.[StructureId] = s.[StructureId]
    AND mem0.[MemberPath] = 'CustomerNo'
  LEFT JOIN [CustomerStrings] mem1
    ON mem1.[StructureId] = s.[StructureId]
    AND mem1.[MemberPath] = 'Name'
 WHERE  ((mem0.[Value] = 20) AND (mem1.[Value] = 'Vahid'));

```

در هر دو حالت از جداول کمکی تعریف شده برای تهیه کوئیری استفاده کرده و نهایتاً فیلد JSON اصلی را برای نگاشتنهایی به اشیاء تعریف شده در برنامه بازگشت می‌دهد.

در کل این هم یک روش تفکر و طراحی Schema less است که با بسیاری از بانک‌های اطلاعاتی موجود سازگاری دارد. برای مشاهده اطلاعات بیشتری در مورد جزئیات این روش می‌توان به [Wiki](#) آن مراجعه کرد.

سلام :

سال نو مبارک ! امیدوارم سال بسیار خوبی در پیش داشته باشید :)

از زمانی که استفاده از Code FirstORM های رایج شده ، اجرای اسکریپت های طولانی جهت ایجاد دیتابیس خیلی استفاده ندارد، اما حالت خاص همیشه پیش می آید. مثلاً قصد داریم پیش از آغاز برنامه پس از ایجاد دیتابیس توسط Entity Framework به یک سری جداول فیلدی با نوع داده‌ی Geometry اضافه کنیم. یا باید به دیتابیس یک سری View و Stored Procedure اضافه کرد. ADO.NET اجرای Script که حاوی Go باشد را پشتیبانی Generate شده توسط SQL Server دستور Go هستند.

اما روش‌ها و ترفند‌های زیادی برای اجرای یک فایل Script طولانی حاوی دستور Go روی دیتابیس وجود دارد :

```

private static string GetScript()
{
    string path = AppDomain.CurrentDomain.BaseDirectory +
        @"Scripts\script.sql";
    var file = new FileInfo(path);

    string script = file.OpenText().ReadToEnd();
    return script;
}

private static void ExecuteScript()
{
    string script = GetScript();

    //split the script on "GO" commands
    var splitter = new[] {"\r\nGO\r\n"};
    string[] commandTexts = script.Split(splitter,
                                         StringSplitOptions.RemoveEmptyEntries);
    foreach (string commandText in commandTexts)
    {
        using (var ctx = new DbContext())
        {
            if (!string.IsNullOrEmpty(commandText))
            {
                ctx.Database.ExecuteSqlCommand(commandText);
            }
        }
    }
}

```

در اینجا به جای آنکه تلاش کنیم یک فایل را روی دیتابیس یک جا اجرا کنیم دستورات را جدا کرده و تک به تک اجرا می‌کنیم.
بروزرسانی: [روش بهتر](#)

نظرات خوانندگان

نویسنده: محسن
تاریخ: ۱۷:۱۷ ۱۳۹۲/۰ ۱/۰۳

ممنون. میشه برای بهبود سرعت insert های زیاد، از [SQL Bulk Copy](#) هم استفاده کرد.

نویسنده: شاهین کیاست
تاریخ: ۱۹:۲۲ ۱۳۹۲/۰ ۱/۰۳

بله درسته،

در سناریویی که برای خودم پیش آمده بود کوئری‌ها فقط Insert نبودند و همچنین فقط قرار بود یک بار در ابتدای برنامه اجرا شوند برای همین روی سرعت حساسیت نداشتم.

نویسنده: مهدی موسوی
تاریخ: ۱۹:۵۳ ۱۳۹۲/۰ ۱/۰۳

سلام: سال نو مبارک (:)

آقای کیاست، بهترین روش برای اجرای Script هایی که حاوی کلمه کلیدی GO هستند، استفاده از SMO است. به کد زیر دقت کنید:

```
Server server = new Server(new ServerConnection() { ConnectionString = cnnStr });
server.ConnectionContext.ExecuteNonQuery(sql);
```

اینجا همه چیز توسط SMO کنترل میشه و دیگه نیازی به آنالیز Script اصلی بر اساس عبارت GO نیست.

موفق باشید.

نویسنده: شاهین کیاست
تاریخ: ۱۴ ۱۳۹۲/۰ ۱/۰۴

سلام قربان :
خیلی ممنون.

در استفاده از SMO در NET 4. مشکل داشتم در نتیجه از این روش که نیازمند به Assembly‌های SMO نیست استفاده کردم.
کامنت شما را در انتهای مطلب قرار دادم.

نویسنده: میثم خوشبخت
تاریخ: ۱۷:۱۸ ۱۳۹۲/۰ ۱/۰۵

شاهین جان
ممنون از مطلب مفیدت
من این مورد رو قبلا نوشتتم. به همین روشی که شما نوشتید. منتها تو یه Query با مشکل مواجه شد و اجرا نشد. واسه همین از روش زیر استفاده کردم که همه جا جواب میده:

```
var commandTexts=Regex.Split(script,@"\s*GO\s+");
foreach(var commandText in commandTexts)
{
    // Execution Code
}
```

نویسنده: مهدی موسوی

سلام.

خیر، همه جا جواب نمیده! عرض کردم، نیازی به جداسازی Script بر اساس GOهای موجود در اون ندارید. Script زیر رو در نظر بگیرید (عنوان مثالی نقض) تا متوجه بشید چرا که شما نیز ایراد داره:

```
USE [MyDB]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GO AND EAT](
[Id] [int] IDENTITY(1,1) NOT NULL,
[MyVal] [decimal](18, 0) NOT NULL,
) ON [PRIMARY]
GO
```

لطفاً اینو در نظر بگیرید که منظورم این نیست که با تصحیح Regex Pattern می‌توانید این مشکل رو نیز برطرف کنید، منظورم اینه که با استفاده از SMO می‌توانید از نوشتن کدهای Error-Prone دوری کنید. فقط همین. (:)

موفق باشید.

نویسنده: وحید نصیری
تاریخ: ۱۷:۱۸ ۱۳۹۲/۰۱/۰۶

پشت صحنه [SMO](#) برای علاقمندان در مورد نحوه جدا سازی labatchها از یکدیگر در عمل.

نویسنده: علی
تاریخ: ۱۵:۴۷ ۱۳۹۲/۰۲/۲۴

برای اجرای چنین دستوراتی نیازی به تکه تکه کردن نداریم فقط کافیه go رو از اسکریپت حذف کنیم خودش همرو اجرا می‌کنه قبل از خودم چنین کاری رو انجام دادم

نویسنده: محسن خان
تاریخ: ۱۶:۵۸ ۱۳۹۲/۰۲/۲۴

کل این بحث در مورد نحوه یافتن صحیح این GO بدون تخریب اطلاعات جانبی بود. حالا می‌خوای حذفش کن یا تکه تکه اش کن!

در این بخش فانکشن دیگری از توابع تحلیلی به نام CUME_DIST را بررسی می‌نماییم.

:CUME_DIST

بوسیله تابع CUME_DIST می‌توان ارزیابی نمود، در یک گروه، چه درصد از مقادیر، مساوی یا کوچکتر از مقدار سطر جاری می‌باشد، به این تابع cumulative distribution نیز گفته می‌شود.

تابع CUME_DIST به صورت زیر است:

```
CUME_DIST( )
OVER ( [ partition_by_clause ] order_by_clause )
```

شرح :Syntax

Partition By Clause 1- بوسیله پارامتر فوق می‌توانید، نتیجه پرس جو (Query)، خود را دسته بندی نمایید.

order by clause 2- همانطور که از نامش مشخص است، جهت مرتب نمودن خروجی Query می‌باشد.

عموماً شرح عملکرد تابع تحلیلی، کمی مشکل است. بنابراین برای درک، عملکرد تابع CUME_DIST چند مثال را بررسی می‌کنیم.
در ابتدا بوسیله Script زیر یک جدول ایجاد و 10 رکورد در آن درج می‌کنیم:

```
Create Table TestCUME_DIST
(SalesOrderID int not null,
OrderQty smallint not null,
ProductID int not null
);
GO
Insert Into TestCUME_DIST
Values (43663,1,760),(43667,3,710),(43667,1,773),
(43667,1,775),(43667,1,778),(43669,1,747),
(43670,1,709),(43670,2,710),(43670,2,773),(43670,1,776)
```

مثال اول: Script زیر را اجرا می‌کنیم، سپس خروجی آن را بررسی می‌نماییم:

```
SELECT SalesOrderID, OrderQty,
CUME_DIST() OVER(ORDER BY SalesOrderID) AS [CUME_DIST]
FROM TestCUME_DIST ORDER BY [CUME_DIST] DESC
```

پس از اجرا خروجی بصورت زیر خواهد بود:

	SalesOrderID	OrderQty	CUME_DIST
1	43670	1	1
2	43670	2	1
3	43670	2	1
4	43670	1	1
5	43669	1	0.6
6	43667	3	0.5
7	43667	1	0.5
8	43667	1	0.5
9	43667	1	0.5
10	43663	1	0.1

در ادامه احازه دهید، مقادیری که در فیلد CUME_DIST بدست آمده است را بصورت تصویری بررسی کنیم.
مقادیر سطر اول تا چهارم:

	SalesOrderID	OrderQty	CUME_DIST
1	43670	1	1
2	43670	2	1
3	43670	2	1
4	43670	1	1
5	43669	1	0.6
6	43667	3	0.5
7	43667	1	0.5
8	43667	1	0.5
9	43667	1	0.5
10	43663	1	0.1

سوال اول: چند سطر، SalesOrderID آن برابر ۴۳۶۷۰ می باشد؟
C1=4 جواب: چهار سطر (Row)، فرض می کنیم

سوال دوم: چند سطر (Row)، زیر SalesOrderID=43670 قرار دارد؟
C2=6 جواب: شش سطر (Row)، فرض می کنیم

سوال سوم: تعداد کل سطرهای (Rows)، چند عدد می باشد؟
C3=10 جواب: ۱۰ سطر (Row)، فرض می کنیم

برای بدست آوردن CUME_DIST چهار سطر اول از فرمول زیر پیروی می نماییم:
 $Rows = (C1+C2)/C3 = (4+6)/10 = 1$

* برای بدست آوردن CUME_DIST سطر پنجم نیز خواهیم داشت:

$Rows = (1+5)/10 = 6/0$ بنابراین خواهیم داشت: $Rows = (c1+c2)/c3$

مثال دوم: ابتدا Script زیر را اجرا نمایید:

```
SELECT SalesOrderID, OrderQty, ProductID,
       CUME_DIST() OVER(PARTITION BY SalesOrderID ORDER BY ProductID ) AS [CUME_DIST]
FROM TestCUME_DIST
WHERE SalesOrderID IN (43670, 43669, 43667, 43663)
ORDER BY SalesOrderID DESC, [CUME_DIST] DESC
```

خروجی :

	SalesOrderID	OrderQty	ProductID	CUME_DIST
1	43670	1	776	1
2	43670	2	773	0.75
3	43670	2	710	0.5
4	43670	1	709	0.25
5	43669	1	747	1
6	43667	1	778	1
7	43667	1	775	0.75
8	43667	1	773	0.5
9	43667	3	710	0.25
10	43663	1	760	1

همانگونه که ملاحظه می‌کنید، در این مثال، خروجی، براساس SalesOrderID به چهار گروه تقسیم می‌شود و عملیات مرتب سازی روی فیلد ProductID انجام می‌گیرد، بنابراین CUME_DIST، روی هر گروه بر روی فیلد ProductID محاسبه می‌شود.

گروه اول : نحوه محاسبه Cume_DIST سطر اول:

سوال: چه تعداد از مقادیر ProductID آن برابر 776 میباشد؟

جواب: فقط مقدار سطر اول، بنابراین خواهیم داشت $c1=1$

سوال: چه تعداد از مقادیر کوچکتر از ProductID=776 میباشد؟

جواب: مقدار سه سطر، در واقع مقادیر سطر دوم، سوم و چهارم کوچکتر از مقدار سطر اول می‌باشند، $c2=3$

سوال: تعداد کل سطرهای گروه اول چه مقدار می‌باشد؟

جواب: 4 سطر

بنابراین برای بدست آوردن CUME_DIST سطر اول خواهیم داشت:

$$\text{Rows} = (1+3)/4 = 1$$

محاسبه سطر دوم از گروه اول بدون شرح:

$$\text{Rows} = (1+2)/4 = 0/75$$

امیدوارم مفید واقع شده باشد.

افزودن یک DataType جدید برای نگهداری تاریخ خورشیدی - ۱

عنوان:

حامد قنادی

نویسنده:

۲۳:۳۰ ۱۳۹۲/۰۲/۰۹

تاریخ:

www.dotnettips.info

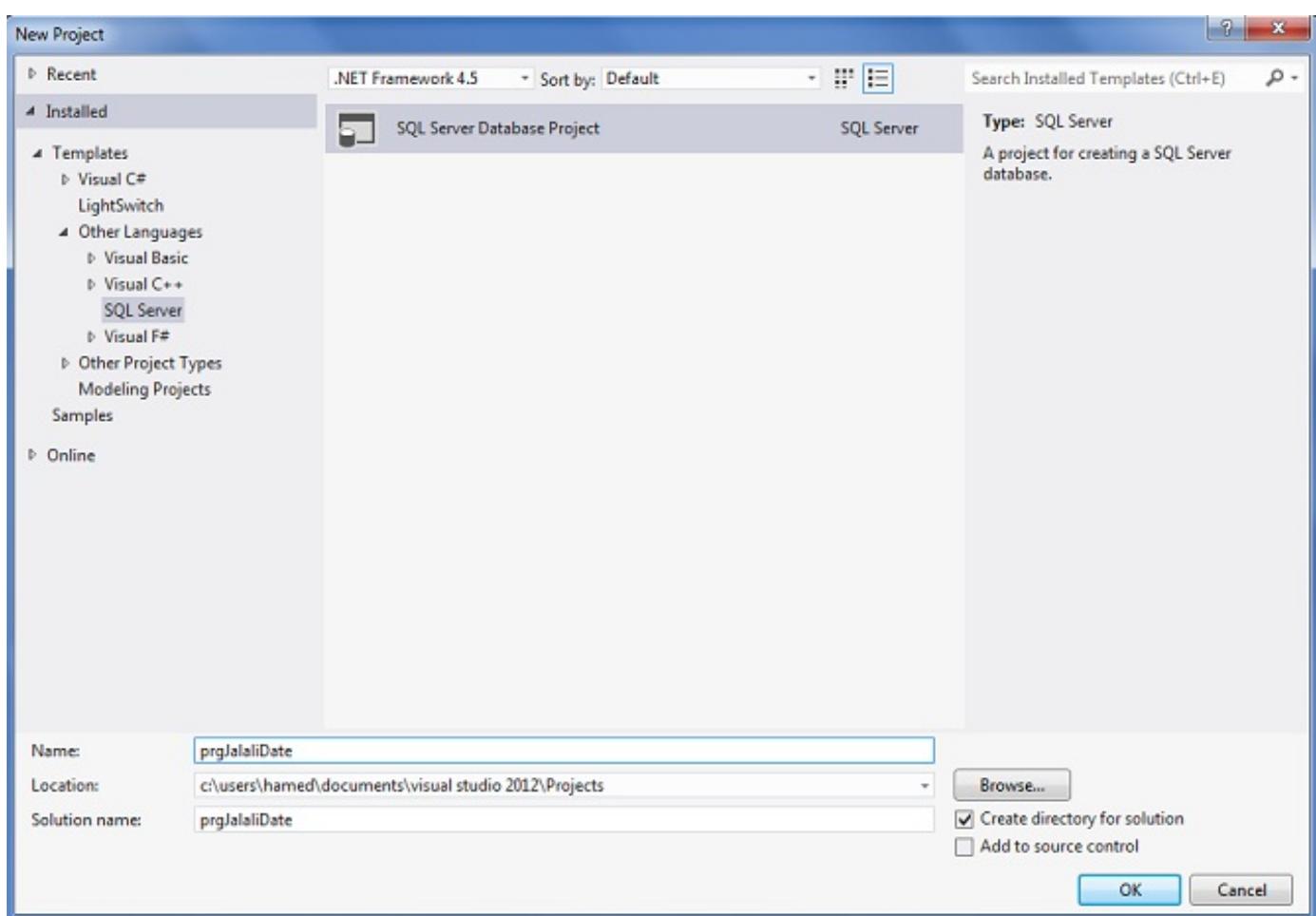
آدرس:

SQL Server, SQL Server 2012, C#.NET, Persian, CLR

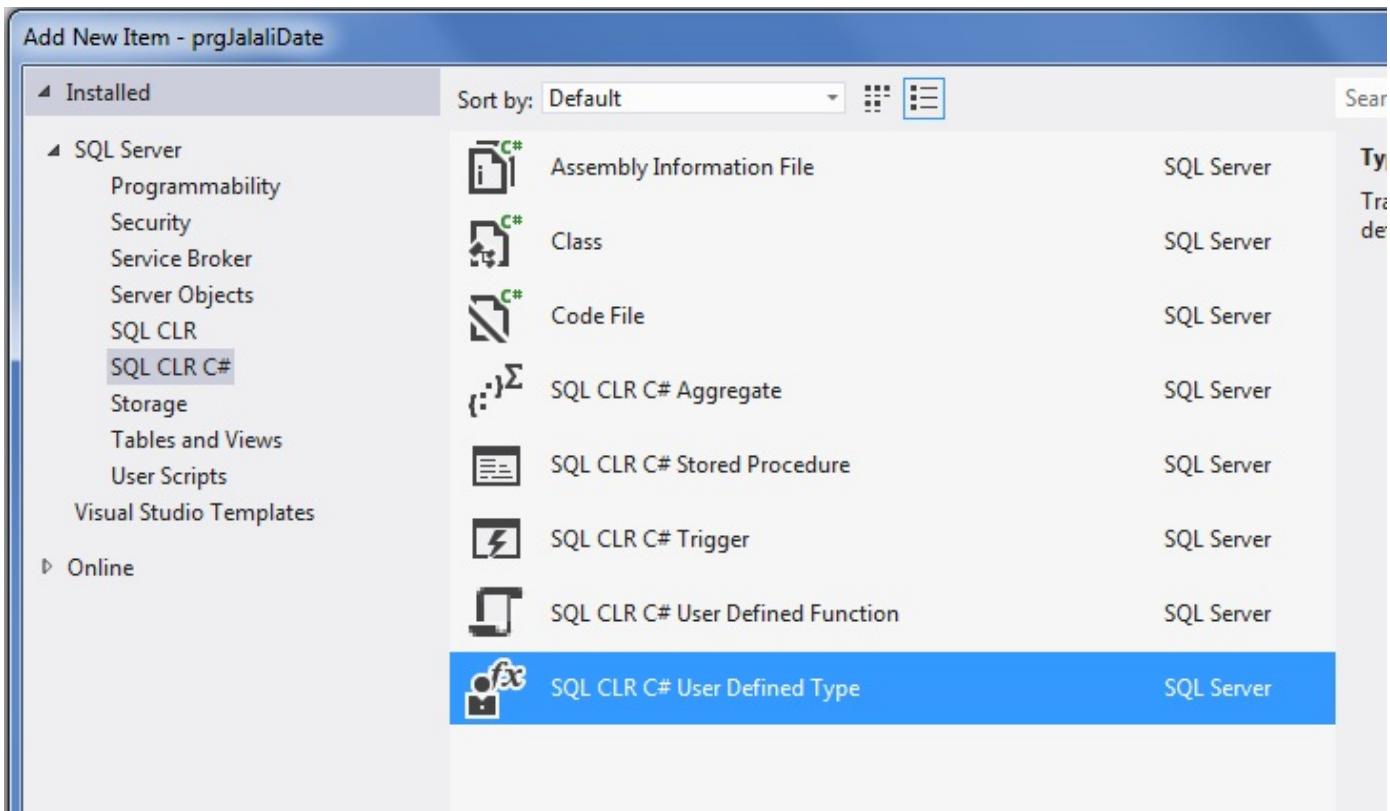
گروهها:

ثبت و نگهداری تاریخ خورشیدی در SQL Server از دیرباز یکی از نگرانی‌های برنامه‌نویسان و طراحان پایگاه داده‌ها بوده است. در این نوشتار، راه‌کار تعریف یک SQL Server 2012 DataType به روش CLR آموزش داده خواهد شد.

در ویژوال استودیو یک پروژه‌ی جدید از نوع SQL Server Database Project به شکل زیر ایجاد کنید:



نام پروژه را به یاد تقویم خیام، prgJalaliDate روی نام پروژه راست‌کلیک کرده، سپس روی کلیک کنید. در پنجره‌ی بازشده مطابق شکل SQL CLR C# User Defined Type را برگزینید؛ سپس نام Add New Item JalaliDateType را برای آن انتخاب کنید.



متن موجود در صفحه‌ی بازشده را کاملاً حذف کرده و با کد زیر جای‌گزین کنید.
 (در کد زیر همه‌ی توابع لازم برای مقداردهی به سال، ماه، روز، ساعت، دقیقه و ثانیه و البته گرفتن مقدار از آن‌ها، تبدیل تاریخ خورشیدی به میلادی، گرفتن تاریخ به تنها ی، افزایش یا کاهش زمان برپایه‌ی یکی از متغیرهای زمان و بررسی و اعتبارسنجی انواع بخش‌های زمان گنجانده شده است. در صورت پرسش یا پیشنهاد روی هر کدام در قسمت نظرات، پیام خود را بنویسید).

```
using System;
using System.Data.SqlTypes;
using Microsoft.SqlServer.Server;

[Serializable()]
[SqlUserDefinedType(Format.Native)]
public struct JalaliDate : INullable
{
    private Int16 m_Year;
    private byte m_Month;
    private byte m_Day;
    private byte m_Hour;
    private byte m_Minute;
    private byte m_Second;
    private bool is_Null;

    public Int16 Year
    {
        get
        {
            return (this.m_Year);
        }
        set
        {
            m_Year = value;
        }
    }

    public byte Month
    {
        get
        {
            return (this.m_Month);
        }
    }

    public byte Day
    {
        get
        {
            return (this.m_Day);
        }
    }

    public byte Hour
    {
        get
        {
            return (this.m_Hour);
        }
    }

    public byte Minute
    {
        get
        {
            return (this.m_Minute);
        }
    }

    public byte Second
    {
        get
        {
            return (this.m_Second);
        }
    }
}
```

```
        }
        set
        {
            m_Month = value;
        }
    }

    public byte Day
    {
        get
        {
            return (this.m_Day);
        }
        set
        {
            m_Day = value;
        }
    }

    public byte Hour
    {
        get
        {
            return (this.m_Hour);
        }
        set
        {
            m_Hour = value;
        }
    }

    public byte Minute
    {
        get
        {
            return (this.m_Minute);
        }
        set
        {
            m_Minute = value;
        }
    }

    public byte Second
    {
        get
        {
            return (this.m_Second);
        }
        set
        {
            m_Second = value;
        }
    }

    public bool IsNull
    {
        get
        {
            return is_Null;
        }
    }

    public static JalaliDate Null
    {
        get
        {
            JalaliDate jl = new JalaliDate();
            jl.is_Null = true;
            return (jl);
        }
    }

    public override string ToString()
    {
        if (this.IsNull)
        {
            return "NULL";
        }
        else
```

۱- افزودن یک جدید برای نگهداری تاریخ خورشیدی - DataType

```
    {
        return this.m_Year.ToString("D4") + "/" + this.m_Month.ToString("D2") + "/" +
this.m_Day.ToString("D2") + " " + this.Hour.ToString("D2") + ":" + this.Minute.ToString("D2") + ":" +
this.Second.ToString("D2");
    }
}

public static JalaliDate Parse(SqlString s)
{
    if (s.IsNull)
    {
        return Null;
    }

    System.Globalization.PersianCalendar pers = new System.Globalization.PersianCalendar();
    string str = Convert.ToString(s);
    string[] JDate = str.Split(' ')[0].Split('/');

    JalaliDate jl = new JalaliDate();

    jl.Year = Convert.ToInt16(JDate[0]);
    byte MonthsInYear = (byte)pers.GetMonthsInYear(jl.Year);
    jl.Month = (byte.Parse(JDate[1]) <= MonthsInYear ? (byte.Parse(JDate[1]) > 0 ?
byte.Parse(JDate[1]) : (byte)1) : MonthsInYear);
    byte DaysInMonth = (byte)pers.GetDaysInMonth(jl.Year, jl.Month); ;
    jl.Day = (byte.Parse(JDate[2]) <= DaysInMonth ? (byte.Parse(JDate[2]) > 0 ?
byte.Parse(JDate[2]) : (byte)1) : DaysInMonth);
    if (str.Split(' ').Length > 1)
    {
        string[] JTime = str.Split(' ')[1].Split(':');
        jl.Hour = (JTime.Length >= 1 ? (byte.Parse(JTime[0]) < 23 && byte.Parse(JTime[0]) >=
(byte)0 ? byte.Parse(JTime[0]) : (byte)0) : (byte)0);
        jl.Minute = (JTime.Length >= 2 ? (byte.Parse(JTime[1]) < 59 && byte.Parse(JTime[1]) >=
(byte)0 ? byte.Parse(JTime[1]) : (byte)0) : (byte)0);
        jl.Second = (JTime.Length >= 3 ? (byte.Parse(JTime[2]) < 59 && byte.Parse(JTime[2]) >=
(byte)0 ? byte.Parse(JTime[2]) : (byte)0) : (byte)0);
    }
    else { jl.Hour = 0; jl.Minute = 0; jl.Second = 0; }

    return (jl);
}

public SqlString GetDate()
{
    return this.m_Year.ToString("D4") + "/" + this.m_Month.ToString("D2") + "/" +
this.m_Day.ToString("D2");
}

public SqlString GetTime()
{
    return this.Hour.ToString("D2") + ":" + this.Minute.ToString("D2") + ":" +
this.Second.ToString("D2");
}

public SqlDateTime ToGregorianTime()
{
    System.Globalization.PersianCalendar pers = new System.Globalization.PersianCalendar();
    return SqlDateTime.Parse(pers.ToDateTime(this.Year, this.Month, this.Day, this.Hour,
this.Minute, this.Second, 0).ToString());
}

public SqlString JalaliDateAdd(SqlString interval, int increment)
{
    System.Globalization.PersianCalendar pers = new System.Globalization.PersianCalendar();
    DateTime dt = pers.ToDateTime(this.Year, this.Month, this.Day, this.Hour, this.Minute,
this.Second, 0);
    string CInterval = interval.ToString();
    bool isConvert = true;
    switch (CInterval)
    {
        case "Year":
            dt = pers.AddYears(dt, increment);
            break;
        case "Month":
            dt = pers.AddMonths(dt, increment);
            break;
        case "Day":
            dt = pers.AddDays(dt, increment);
            break;
        case "Hour":
```

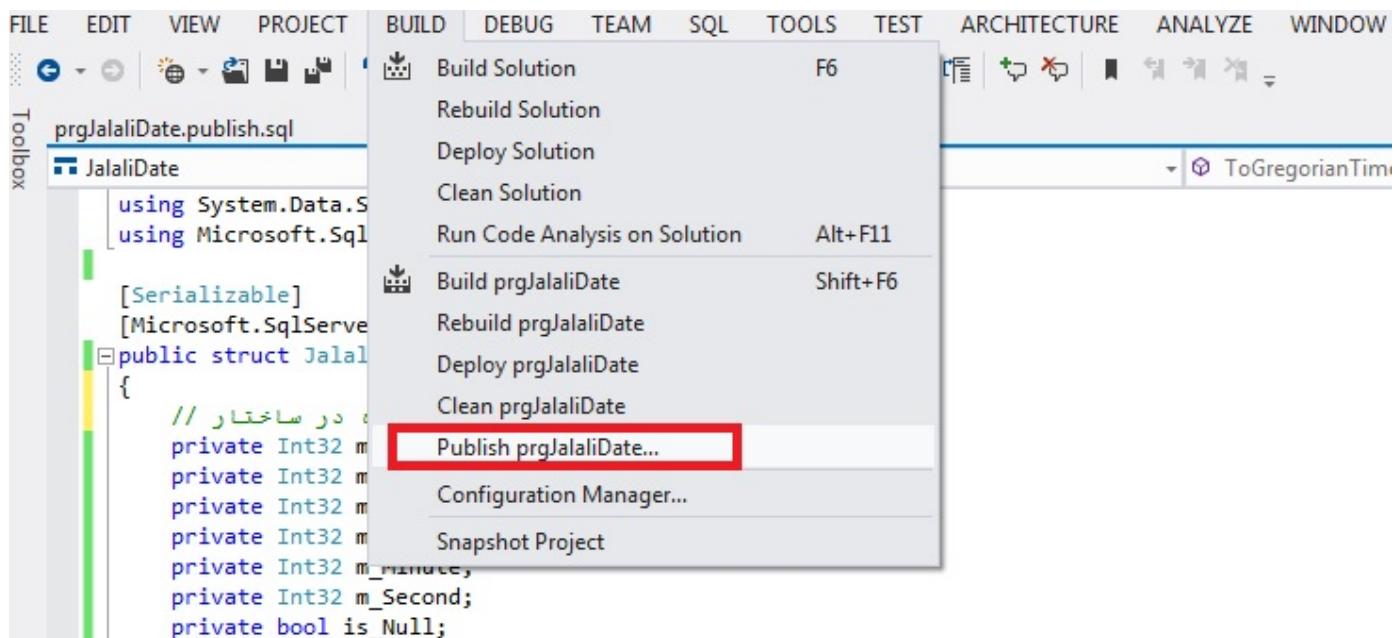
۱- افزودن یک جدید برای نگهداری تاریخ خورشیدی - DataType

```
        dt = pers.AddHours(dt, increment);
        break;
    case "Minute":
        dt = pers.AddMinutes(dt, increment);
        break;
    case "Second":
        dt = pers.AddSeconds(dt, increment);
        break;
    default:
        isConvert = false;
        break;
    }

    if (isConvert == true)
    {
        this.Year = (Int16)pers.GetYear(dt);
        this.Month = (byte)pers.GetMonth(dt);
        this.Day = (byte)pers.GetDayOfMonth(dt);
        this.Hour = (byte)pers.GetHour(dt);
        this.Minute = (byte)pers.GetMinute(dt);
        this.Second = (byte)pers.GetSecond(dt);
    }

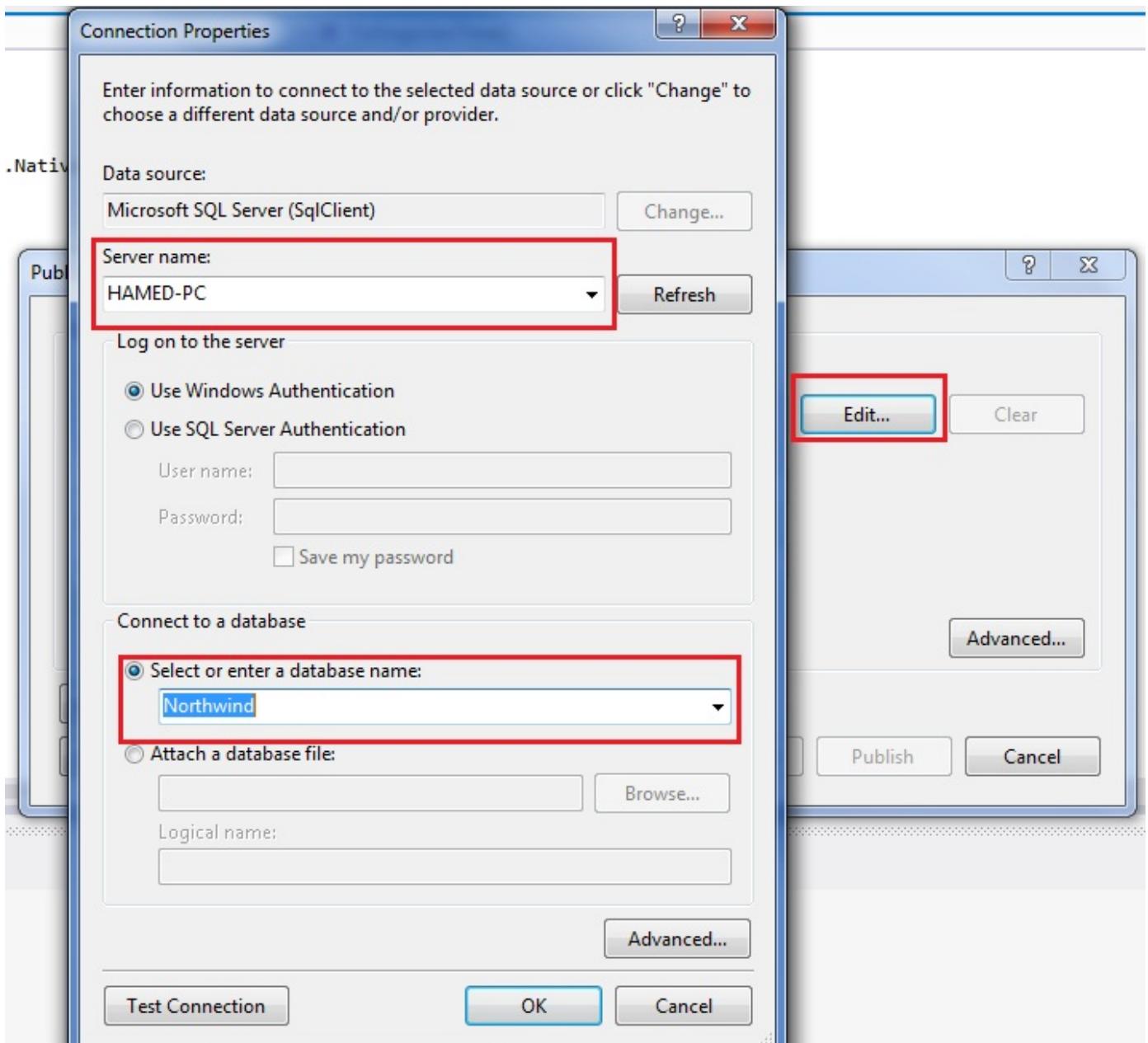
    return this.m_Year.ToString("D4") + "/" + this.m_Month.ToString("D2") + "/" +
this.m_Day.ToString("D2") + " " + this.Hour.ToString("D2") + ":" + this.Minute.ToString("D2") + ":" +
this.Second.ToString("D2");
}
}
```

از منوهای بالا روی منوی Build و سپس گزینه‌ی Publish prgJalaliDate کلیک کنید:



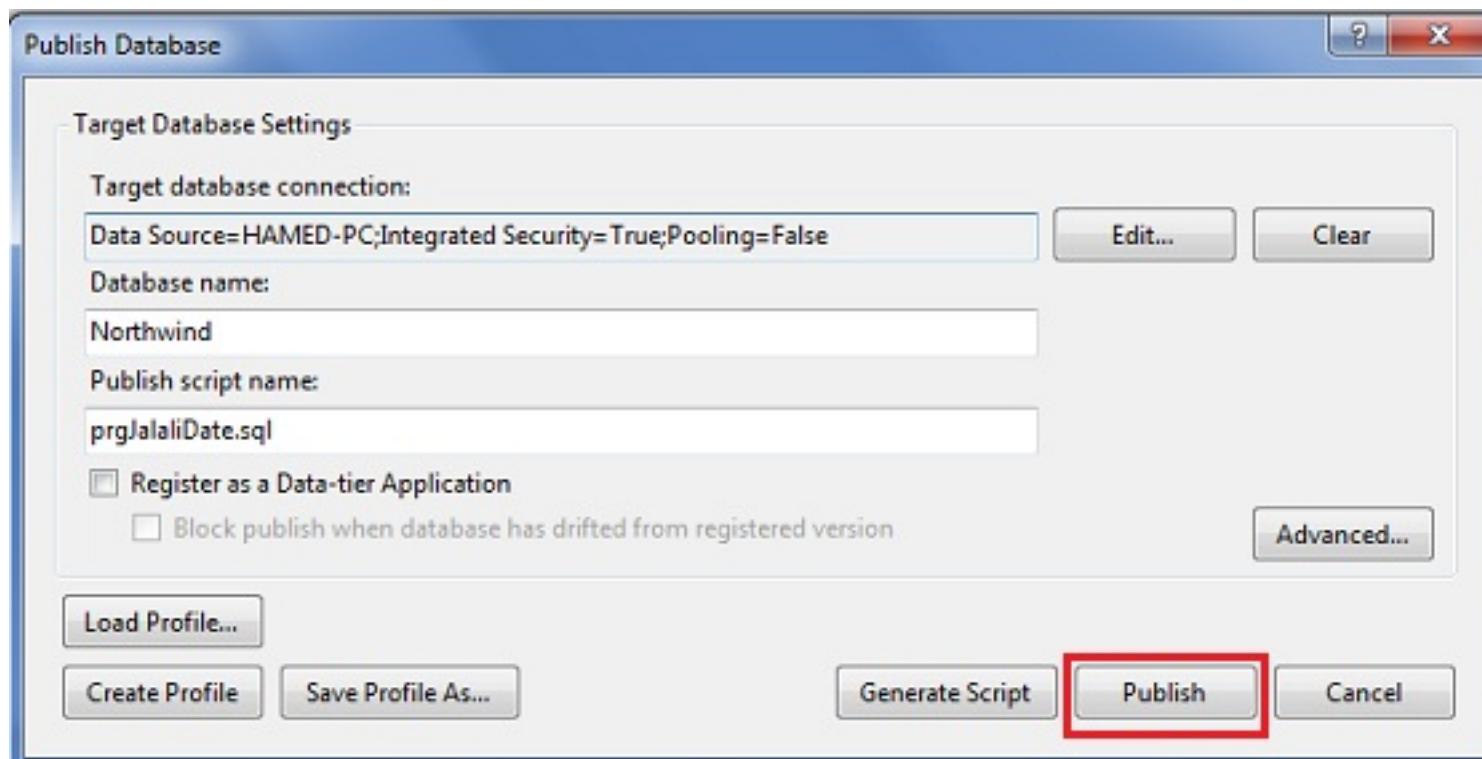
در پنجره‌ی بازشده روی دکمه‌ی Edit کلیک کنید سپس تنظیمات مربوط به اتصال به پایگاه داده را انجام دهید.

1- افزودن یک جدید برای نگهداری تاریخ خورشیدی - DataType



روی دکمه OK کلیک کنید و سپس در پنجره اولیه، روی دکمه Publish کلیک کنید:

افزودن یک DataType جدید برای نگهداری تاریخ خورشیدی - 1



به همین سادگی، DataType مربوطه در SQL Server 2012 ساخته می‌شود. خبر خوش این‌که شما می‌توانید با راست‌کلیک روی نام پروژه و انتخاب گزینه‌ی Project Setting در قسمت Properties تنظیمات مربوط به نگارش SQL Server را انجام دهید. (از نگارش 2005 به بعد در VS 2012 پشتیبانی می‌شود).

اکنون زمان آن رسیده است که dataType ایجاد شده را در SQL Server 2012 بیازماییم. SQL Server را باز کنید و دستور زیر را در آن اجرا کنید.

```
USE Northwind
GO
CREATE TABLE dbo.TestTable
(
    Id int NOT NULL IDENTITY (1, 1),
    TestDate dbo.JalaliDate NULL
) ON [PRIMARY]
GO
```

همین‌طور که مشاهده می‌کنید؛ امکان به کارگیری DataType تعریف شده وجود دارد.
اکنون چند رکورد درون این جدول درج می‌کنیم:

```
Insert into TestTable (TestDate) Values ('1392/02/09'), ('1392/02/09 22:40'), ('1392/12/30 22:40')
```

پس از اجرای این دستور خطای زیر در پایین صفحه‌ی SQL Server نمایان می‌شود:

```
Msg 6263, Level 16, State 1, Line 1
Execution of user code in the .NET Framework is disabled. Enable "clr enabled" configuration option.
```

این خطا به این خاطر است که CLR را در SQL Server فعال نکرده ایم. جهت فعال کردن CLR دستور زیر را اجرا کنید:

```
sp_configure 'clr enabled', 1
Reconfigure
```

بار دیگر دستور درج را اجرا می‌کنیم:

```
Insert into TestTable (TestDate) Values ('1392/02/09'), ('1392/02/09 22:40'), ('1392/12/30 22:40')
```

	Id	TestDate
▶	1	1392/02/09 00:00:00
	2	1392/02/09 22:40:00
	3	1392/12/29 22:40:00
*	NULL	NULL

ملاحظه می‌کنید که داده‌ها در جدول مربوطه ذخیره شده است. در رکورد نخست چون ساعت، دقیقه و ثانیه تعریف نشده است؛ به طور هوشمند صفر درج شده است. در رکورد دوم، ساعت و دقیقه مقدار دارد ولی ثانیه صفر ثبت شده است. و در رکورد سوم چون سال 1392 کبیسه نیست؛ به صورت هوشمند آخرین روز ماه به جای روز ثبت شده است. هر چند می‌توان با دست‌کاری در توابع سی‌شارپ، این قوانین را عوض کرد.

اکنون زمان آن رسیده است که توسط یک پرس‌وجو، همه‌ی توابعی که در سی‌شارپ برای این نوع داده نوشته‌یم، بیازماییم.
پرس‌وجوی زیر را اجرا کنید:

```
Select TestDate.ToString() as JalaliDateTime,
       TestDate.GetDate() as JalaliDate, TestDate.GetTime() as JalaliTime,
       TestDate.ToGregorianTime() as GregorianTime,
       TestDate.JalaliDateAdd('Day',1) JalaliTomorrow,
       TestDate.Month as JalaliMonth from TestTable
```

خروجی این پرس‌وجو به شکل زیر خواهد بود:

	JalaliDateTime	JalaliDate	JalaliTime	GregorianTime	JalaliTomorrow	JalaliMonth
1	1392/02/09 00:00:00	1392/02/09	00:00:00	2013-04-29 00:00:00.000	1392/02/10 00:00:00	2
2	1392/02/09 22:40:00	1392/02/09	22:40:00	2013-04-29 22:40:00.000	1392/02/10 22:40:00	2
3	1392/12/29 22:40:00	1392/12/29	22:40:00	2014-03-20 22:40:00.000	1393/01/01 22:40:00	12

البته درباره‌ی ستون پنجم و ششم شما می‌توانید روی همه‌ی اجزای تاریخ افزایش و کاهش داشته باشید و هم‌چنین می‌توانید با تابع مربوطه هر کدام از اجزای زمان را جداگانه به دست بیاورید که در این مثال عدد ماه نشان داده شده است.

نیازی به گفتن نیست که می‌توانید به سادگی از توابع مربوط به DateTime در SQL Server بهره ببرید. برای مثال برای به دست

۱ - افزودن یک DataType جدید برای نگهداری تاریخ خورشیدی

آوردن فاصله‌ی میان دو روز از پرس‌وجوی زیر استفاده کنید:

```
Declare @a JalaliDate = '1392/02/07 00:00:00'  
Declare @b JalaliDate = '1392/02/05 00:00:00'  
SELECT DATEDIFF("DAY",@b.ToGregorianTime(),@a.ToGregorianTime()) AS DiffDate
```

شاد و پیروز باشید.

نظرات خوانندگان

نویسنده: ali
تاریخ: ۱۳۹۲/۰۲/۱۰ ۰:۵

Nice article
Thanks

نویسنده: سام ناصری
تاریخ: ۱۳۹۲/۰۲/۱۰ ۴:۲۲

بسیار خوب.

فقط من تو مقاله شما دلیلی برای اینکه چرا زمان و تاریخ را میخواهی به این صورت ذخیره کنی متوجه نشدم؟ چرا به همان شکل استانداردش ذخیره نکنیم؟
 دیگر اینکه نوع داده جدید به چه شکل در دیتابیس ذخیره میشود. Sql Server از کجا میداند که باید چگونه لیترالها را پارس کند(چگونه متود Parse برگزیده میشود)؟ عملگرهای مقایسه چگونه کار خواهند کرد؟

نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۱۰ ۷:۸

با درود
به پرسش‌های شما در بخش دوم پاسخ خواهم داد.

نویسنده: قاسم
تاریخ: ۱۳۹۲/۰۲/۱۰ ۹:۴۲

سلام، میشه مراحل انجام کار توی VS2010 هم بنویسید، فقط مراحل کار. خیلی ممنون میشم

نویسنده: فرشید علی اکبری
تاریخ: ۱۳۹۲/۰۲/۱۰ ۹:۵۵

سلام
کدهای شمارو درست همونطوریکه گفتین کپی کردم ولی موقعی که Publish رو میزنم پیغام زیر رو میده :
Publish cannot begin until your project(s) build successfully
و پنجره publish ظاهر نمی‌شه و هیچ جایی هم برای تعریف کانکشن وجود نداره... مشکل از کجاست؟
دوم اینکه c1r رو هربار که خواستیم روی یک دیتابیس جدید این نوع رو تعریف کنیم باید فعال کنیم... مثلا توی ایجاد سال مالی جدید که سیستم یک دیتابیس خام ایجاد میکنه تا مانده حسابها رو بهش انتقال بدیم قبلش باید دستور فعال سازی c1r رو هم مجددا بدیم یا فقط توی زمان طراحی کفایت میکنه؟
سوم اینکه : در مورد استفاده اون توی EF Code First هم اگه نکته‌ی خاصی وجود داره محبت کنین ممنون میشم.
شاد و پیروز باشید.

نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۱:۴

با درود

New Project -> Database -> SQL Server -> Visual C# SQL CLR Database Project
- تنظیمات اتصال به پایگاه داده ها
- انتخاب دکمه Yes

Add New Item -> User Defined Type -
- کپی کدها

- استفاده از Deploy در منوی Build یا استفاده از روشی که در بخش دوم نوشتار آموزش داده خواهد شد.

نویسنده: حامد قنادی
تاریخ: ۱۱:۲۶ ۱۳۹۲/۰۲/۱۰

با درود

1- لطفاً پروژه را پیش از Publish یک بار Rebuild کنید. احتمالاً به خاطر یک ارور خاص Publish نمی‌شود. اگر به راحتی Rebuild شده ولی باز هم Publish کار نمی‌کند؛ می‌توانیم با روش دیگری که در بخش دوم آموزش می‌دهم DLL را به SQL معرفی کنیم.

2- کافی است یک بار پس ساخت Database این پرس‌وجو را اجرا کنید.

3- تست نکردم ولی به نظر می‌رسد این نوع داده از سمت EF شبیه به Hierarchy باشد.

نویسنده: ش.د.
تاریخ: ۱۴:۳۵ ۱۳۹۲/۰۲/۱۰

آیا فقط در sq12012 قابل اجرا می‌باشد؟

نویسنده: محسن خان
تاریخ: ۱۴:۵۴ ۱۳۹۲/۰۲/۱۰

در متن نوشتمن از نگارش 2005 به بعد اس کیوال سرور این قابلیت استفاده از افزونه‌های CLR اضافه شدن.

نویسنده: سید امیر سجادی
تاریخ: ۱۸:۳۸ ۱۳۹۲/۰۲/۱۰

با تشکر از مطلب مفیدتون.

چند تا سوال برام پیش اومده.

اول اینکه آیا به صورت یک DLL به بانک اضافه می‌شه؟

دوم اینکه اگه از بانک بک آپ بگیرید و جایی دیگه خواستیم اون رو رویستور کنیم چی می‌شه؟

آپلود بانک روی هاست (بک آپ یا اتج) ؟

نویسنده: امیر بختیاری
تاریخ: ۹:۴۳ ۱۳۹۲/۰۲/۱۱

به صورت اسembly به بانک اضافه می‌شه

وقتی بک آپ بگیرید و رویستور کنید همراه بک آپ این اسembly هم انقال داده می‌شود

در اتج هم به همین شکل

فقط زمانی که از این اسembly در توابع و پروسیجرها استفاده می‌کنید نمی‌توانید حذف کنید و در صورت تغییرات باید اسembly را به روز کنید.

یک مورد دیگه که من زیاد تو هاست‌های شیر شده بهش برخوردم اینه که این امکان توشون فعال نیست و ادمین سرورها هم به سختی این امکان را فعال می‌کنند پس اگر خواستید از این امکان استفاده کنید ابتدا از فعال بودن آن مطمئن شوید

نویسنده: rahim
تاریخ: ۱۲:۱۴ ۱۳۹۳/۰۵/۱۱

با تشکر از مطلب مفیدتون

من زمانی که از این روش استفاده می‌کنم در هنگام درج رکورد جدید با پیغام خطای زیر مواجه می‌شم.

: "A .NET Framework error occurred during execution of user-defined routine or aggregate "JalaliDate

افزودن یک DataType جدید برای نگهداری تاریخ خورشیدی - ۱

.System.OverflowException: Value was either too large or too small for an unsigned byte

نویسنده: محمد
تاریخ: ۰:۲۳ ۱۳۹۳/۰۸/۲۲

با سلام؛ من زمانی که این کد را در SQL 2012 میکنم یک Error در خصوص ToGregorianCalendar() نمایش داده میشه. لطفا راهنمایی بفرمایید. با تشکر

```
Msg 6522, Level 16, State 2, Line 1
A .NET Framework error occurred during execution of user-defined routine or aggregate
"SpatialDateTime":
System.FormatException: String was not recognized as a valid DateTime.
System.FormatException:
at System.DateTimeParse.ParseExactMultiple(String s, String[] formats, DateTimeFormatInfo dtfi,
DateTimeStyles style)
at System.DateTime.ParseExact(String s, String[] formats, IFormatProvider provider, DateTimeStyles
style)
at System.Data.SqlTypes.SqlDateTime.Parse(String s)
at SpatialDateTime.ToGregorianTime().
```

نویسنده: محسن خان
تاریخ: ۰:۵۵ ۱۳۹۳/۰۸/۲۲

یک try/catch بذار، تا بتونی تاریخ مشکل دار رو پیدا کنی:

```
var pers = new PersianCalendar();
var date = pers.ToDateTime(this.Year, this.Month, this.Day, this.Hour, this.Minute, this.Second,
0).ToString();
try
{
    return SqlDateTime.Parse(date);
}
catch(Exception ex)
{
    throw new InvalidOperationException("Can't parse " + date);
}
```

بیش از هر چیز به شما پیشنهاد می‌کنم؛ بار دیگر کد سی‌شارپ درس نخست را در پروژه‌ی خود کپی کنید و سپس Publish را بزنید. پس از ارسال آن مطلب، تغییراتی در جهت بهینه‌سازی کد دادم که به نظرم بهتر است شما نیز در پروژه‌ی خود به کار ببرید.

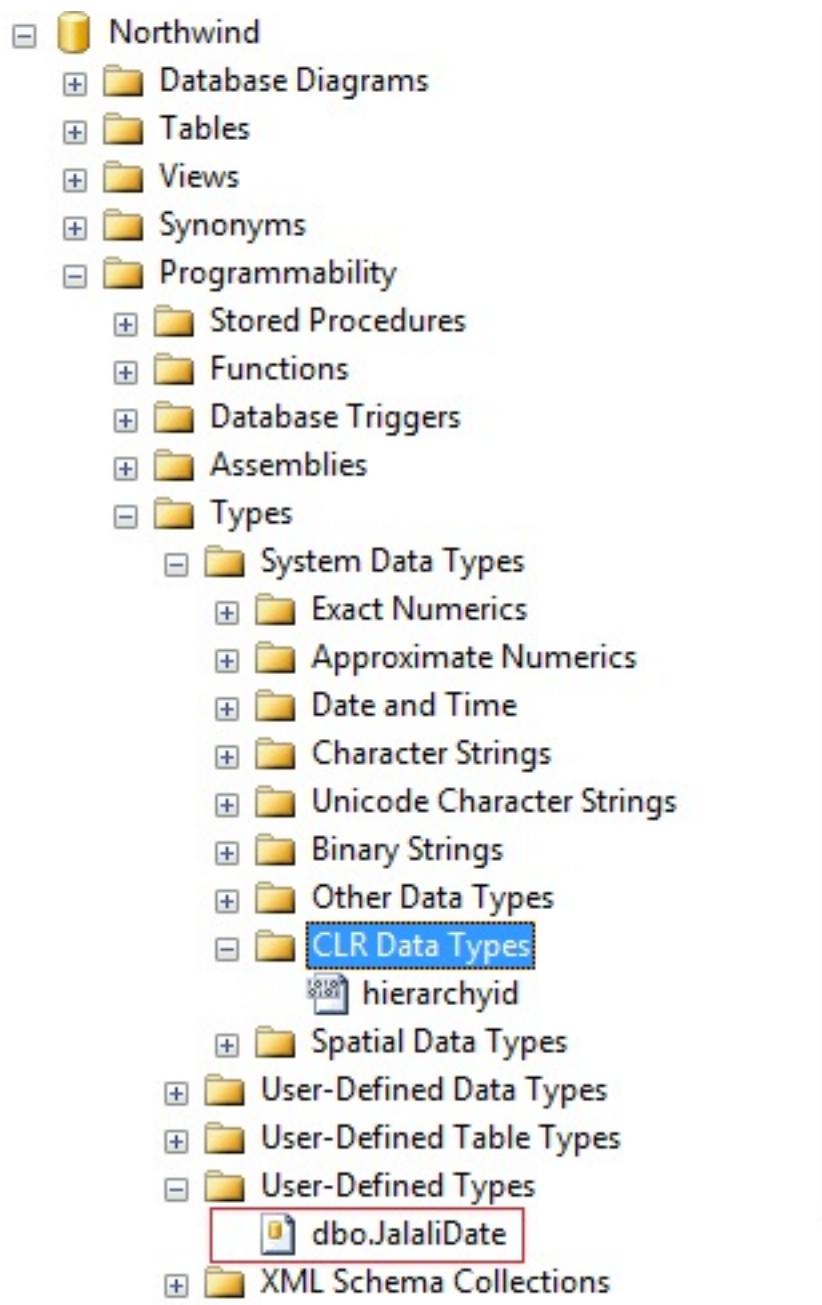
چرا از این نوع داده استفاده کنیم؟

نخستین پرسشی که ممکن است برای شما پیش بیاید این است که چرا بهتر است از این نوع داده استفاده کنیم. برای پاسخ به این پرسش باید راه کارهای گذشته را بررسی کنیم. معمولاً طراحان پایگاه داده‌ها برای استفاده از تاریخ خورشیدی، زمان را به صورت میلادی ثبت می‌کنند؛ سپس با یک scalar-valued function زمان درج شده را به خورشیدی تبدیل می‌کنند. در این صورت می‌توان با یک تابع کوچک دیگر بخش مربوط به ساعت را نیز از همان ستون به دست آورد. در این صورت می‌توانیم از کلیه‌ی متدهای مربوط به DateTime در SQL از جمله افزایش و کاهش و تقاضل دو تاریخ بهره برد. برخی دیگر از طراحان، ستون از نوع char(10) در نظر می‌گیرند و تاریخ خورشیدی را به صورت ده‌کاراکتری در آن ذخیره می‌کنند. این روش هرچند نیاز به تبدیل به خورشیدی زمان باید یک فیلد دیگر از نوع کاراکتری و یا در نگارش‌های نوین‌تر از نوع time تعریف کنیم. برخی دیگر از هر دو را در کنار هم استفاده می‌کنند و در واقع جهت سرعت بالاتر نمایش و بررسی داده‌ها از طریق محیط SQL Server از فیلد کاراکتری تاریخ خورشیدی و برای مقایسه و بدست آوردن ساعت از فیلد نوع DateTime استفاده می‌کنند.

از نظر فضای اشغال شده نوع DateTime، هشت بایت، smalldatetime (در صورت استفاده) 4 بایت و فیلد 10 کاراکتری تاریخ با 8 بایت فضا اشغال می‌کند در صورتی که نوع JalaliDate با درنظر گرفتن همه‌ی مزایای انواع داده‌ی استفاده شده برای تاریخ، فقط مختلف زمان از سال تا ثانیه را با یک متد به دست می‌آورید. می‌توانید به راحتی به تاریخ خود زمانی را بیفزایید یا بکاهید و در گزارش‌ها بدون نگرانی از تبدیل درست استفاده کنید. چون کد باز است می‌توانید با کمی حوصله امکانات دیگر مد نظر خود را به آن بیفزایید و از آن در SQL بهره ببرید.

چگونه این نوع داده را حذف کنم؟

شما می‌توانید به سادگی نوع داده‌ی ایجاد شده توسط CLR را در مسیر زیر بیابید و اقدام به حذف آن نمایید:



همان‌طور که مشاهده می‌شود؛ حتی نوع داده‌ی سیستمی hierarchyid که جهت ساختار سلسله‌مراتبی مانند چارت سازمانی یا درخت تجهیزات استفاده می‌شود؛ نیز یک نوع داده‌ی CLR است.

آیا راه دیگری نیز برای افزودن این نوع داده به جز SQL Publish به کردن وجود دارد؟
مانند بسیاری دیگر از گونه‌های پروژه، در اینجا نیز شما یک فایل DLL خواهید داشت. این فایل برپایه‌ی تنظیماتی که شما در قسمت Properties پروژه‌ی خود انجام می‌دهید ساخته می‌شود. پس از تغییر مسیر فایل DLL در دستور زیر توسط یک New Query در دستور زیر توسط یک Database خود، آن را اجرا کنید:

```
CREATE ASSEMBLY JalaliDate
FROM 'F:\prgJalaliDate.dll'
WITH PERMISSION_SET = SAFE;
```

همچنین در صورت ویرایش‌های دوباره پروژه از دستور زیر استفاده کنید:

2 افزودن یک DataType جدید برای نگهداری تاریخ خورشیدی -

```
ALTER ASSEMBLY JalaliDate  
FROM 'F:\prgJalaliDate.dll'
```

با استفاده از دستورهای زیر می‌توانید از چگونگی درج فایل‌های افزوده شده آگاه شوید:

```
select * from sys.assemblies  
select * from sys.assembly_files
```

	name	principal_id	assembly_id	clr_name	permission...	permission_set_desc	is_visible	create_date	modfy_date	is_user_defined
1	Microsoft.SqlServer.Types	4	1	microsoft.sqlservertypes.ver...	3	UNSAFE_ACCESS	1	2012-02-10 20:...	2012-02-10 ...	0
2	prgJalaliDate	1	65536	prgJalaliDate, version=0.0.0.0...	1	SAFE_ACCESS	1	2013-04-29 22:...	2013-04-29 ...	1
	assembly_id	name	file_id	content						
1	1	microsoft.sqlservertypes.dll	1	0x4D5A90000300000004000000FFFF0000B800000000000000...						
2	65536	prgJalaliDate	1	0x4D5A90000300000004000000FFFF0000B800000000000000...						
3	65536	prgJalaliDate.pdb	2	0x4D6963726F736F667420432F432B2B204D534620372E30...						

تا اینجا SQL Server، دیالال مربوط به پروژه را شناخته است. برای تعریف نوع داده از دستور زیر بهره ببرید:

```
CREATE TYPE dbo.JalaliDate  
EXTERNAL NAME JalaliDate.[JalaliDate];
```

این کار همانند استفاده از گزینه‌ی Publish در Visual Studio است.

همچنین چنان‌چه در SQL Server 2012 از منوی راست‌کلیک پایگاه داده‌ها روی گزینه Tasks و سپس Generate Scripts انتخاب کنیم، از مشاهده‌ی سند ساخته شده، درخواهیم یافت که حتی دستورهای مربوط به ساخت اسمبلی CLR با تبدیل فایل به کد در Scripts وجود دارد و با اجرای آن در سروری دیگر، انتقال می‌یابد.

```
GO  
***** Object: SqlAssembly [prgJalaliDate] Script Date: 2013/04/30 08:27:00 *****  
CREATE ASSEMBLY [prgJalaliDate]  
FROM 0x4D5A90000300000004000000FFFF0000B800000000000000.....  
WITH PERMISSION_SET = SAFE  
  
GO  
***** Object: UserDefinedType [dbo].[JalaliDate] Script Date: 2013/04/30 08:27:00 *****  
CREATE TYPE [dbo].[JalaliDate]  
EXTERNAL NAME [prgJalaliDate].[JalaliDate]
```

GO

دنباله دارد ...

نظرات خوانندگان

نویسنده: محمد عادل
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱:۴۳

در این حالت ، چطور میتوانیم در EF Code First از این DataType استفاده کنیم ؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۱ ۸:۳۴

از برنامه [Entity Framework Power Tools](#) برای مهندسی معکوس ساختار موجود استفاده کنید و بعد مشاهده کنید که چه کدی رو تولید میکنند. کار شما در این حالت code first نیست.

نویسنده: حامد حسین نژاد
تاریخ: ۱۳۹۲/۰۲/۱۱ ۹:۲۰

البته باید این را هم در نظر گرفت که کلا CLR Integration باعث کندی دیتابیس میشود و انواع داده CLR بسیار کنتر از بقیه اجرا میشوند. این موضوع ممکن است که در دیتابیس های کوچک چندان مهم بنظر نیاید ولی در دیتابیس های بزرگ باعث بروز مشکل خواهد شد. علاوه بر این استفاده از انواع داده CLR و یا توابع CLR در دیتابیس باعث میشود که امکان پارتبیشن بندی جداول آن دیتابیس وجود نداشته باشد.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۱:۳۴

[مقاله‌ای هست اینجا](#) در مورد کارآیی CLR در SQL Server به نظر میرسه سریعتر است حدود ۱۱ درصد نسبت به T-SQL معمولی. برای پارتبیشن بندی میتوانید اینکار را انجام بدهید فقط این نوع خاص قابل انتخاب نیست. مابقی فیلد ها [رو میتوانید](#) انتخاب کنید.

نویسنده: حامد حسین نژاد
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۸:۳۱

البته کارآیی CLR، بسته به مورد استفاده، متفاوت است ([این لینک](#)). در مواردی مثل همین مثال، اگه تعداد سطور جدول زیاد باشه، کارآیی رو به شدت کاهش میده. مخصوصاً اگه بخواین از Data Warehousing هم استفاده کنید.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۱ ۱۹:۲۸

این به شدت رو میتوانید در موردن عدده و رقم ارائه بددید؟ در همون لینکی که دادید عنوان شده پیاده سازی RegEx روی سطور بالا خیلی سریع تر هست با CLR. در هر حال بهتره با عدد و رقم و محاسبات بحث کرد.

نویسنده: حامد حسین نژاد
تاریخ: ۱۳۹۲/۰۲/۱۲ ۱۴:۲۸

مثلاً فرض کنید جدولی دارید که اطلاعات Task های یک شرکت رو نگه میداره که شامل تاریخ شروع و تاریخ پایان هر Task هم هست. اگه این جدول یک میلیون سطر داشته باشه و بخوایم Task هایی که مدت زمان انجام اونها کمتر از ۵ روز بوده رو انتخاب کنیم تفاوت فاصله با `datetime` خواهیم داشت.

البته، همونطور که گفتم بسته به نوع استفاده داره و ممکنه از دیتابیسی به دیتابیس دیگه فرق بکنه.

در دو درس گذشته با dataType ای که با هم نوشتیم؛ می‌توانستیم این پرس‌وجوها را داشته باشیم:

```
Declare @MyDate JalaliDate = '1392/02/11 21:38:24'
Select @MyDate.ToString() as MyDateTime
      , @MyDate.GetDate() as MyDate
      , @MyDate.GetTime() as MyTime
      , @MyDate.Year as MyYear
      , @MyDate.Month as MyMonth
      , @MyDate.Day as MyDay
      , @MyDate.Hour as MyHour
      , @MyDate.Minute as MyMinute
      , @MyDate.Second as MySecond
      , @MyDate.JalaliDateAdd('Year',1) as NextYear
      , @MyDate.JalaliDateAdd('Month',1) as NextMonth
      , @MyDate.JalaliDateAdd('Day',1) as NextDay
      , @MyDate.JalaliDateAdd('Hour',1) as NextHour
      , @MyDate.JalaliDateAdd('Minute',1) as NextMinute
      , @MyDate.JalaliDateAdd('Second',1) as NextSecond
```

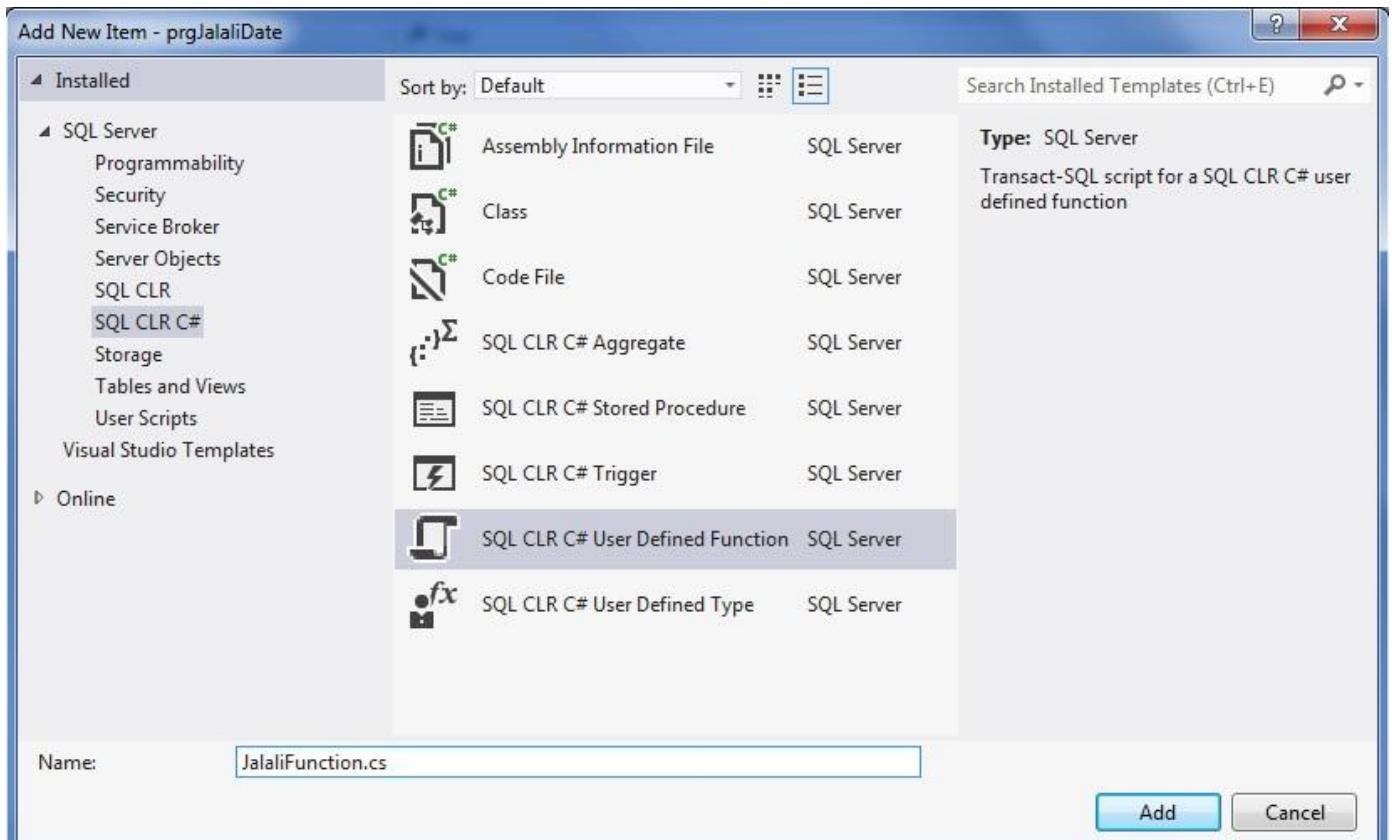
که در نهایت خروجی کار به شکل زیر خواهد بود:

	Results	Messages								
	MyDateTime	MyDate	MyTime	MyYear	MyMonth	MyDay	MyHour	MyMinute	MySecond	I
1	1392/02/11 21:38:24	1392/02/11	21:38:24	1392	2	11	21	38	24	
<hr/>										
NextYear	NextMonth	NextDay	NextHour	NextMinute	NextSecond					
1393/02/11 21:38:24	1392/03/11 21:38:24	1392/02/12 21:38:24	1392/02/11 22:38:24	1392/02/11 21:39:24	1392/02/11 21:38:25					

با این همه، همان‌سان که در پرس‌وجوی بالا هم می‌بینید؛ ناگزیر شدم تاریخ و زمان را خودم وارد کنم و هر آینه در واقعیت شما جهت پرس‌وجو روی زمان کنونی، ناگزیر به استفاده از یک تابع برای تبدیل تقویم میلادی به خورشیدی هستیم. به نظر شما بهتر نیست دست به کار شویم و تابعی برای تبدیل تاریخ میلادی به خورشیدی بنویسیم؟

برای این کار پروژه‌ای را که در دو درس 1 ساختیم باز کنید و سپس روی نام پروژه در Solution Explorer راست‌کلیک کرده و Add New Item را انتخاب کنید.

3- جدید برای نگهداری تاریخ خورشیدی - افزودن یک DataType



محتويات فایل بازشده را حذف کنید و دستورهای زیر را جایگزین کنید:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Data.SqlTypes;
using Microsoft.SqlServer.Server;

public partial class UserDefinedFunctions
{
    [Microsoft.SqlServer.Server.SqlFunction]
    public static JalaliDate GetCurrentDateTime()
    {
        System.Globalization.PersianCalendar pers = new System.Globalization.PersianCalendar();
        DateTime CurrentDate = DateTime.Now;
        JalaliDate jl;
        jl.Year = (Int16)pers.GetYear(CurrentDate);
        jl.Month = (byte)pers.GetMonth(CurrentDate);
        jl.Day = (byte)pers.GetDayOfMonth(CurrentDate);
        jl.Hour = (byte)pers.GetHour(CurrentDate);
        jl.Minute = (byte)pers.GetMinute(CurrentDate);
        jl.Second = (byte)pers.GetSecond(CurrentDate);
        return jl;
    }
}
```

بار دیگر پروژه را به روشنی که در درس یک آموزش داده شد: Publish کنید. اکنون میتوانیم تاریخ و زمان کنونی را در متغیری از نوع JalaliDate وارد کنیم.

```
Declare @MyDate JalaliDate = dbo.GetCurrentDateTime()
```

یا به این شکل تاریخ و زمان جاری را مشاهده کنیم:

```
Select dbo.GetCurrentDateTime().ToString()
```

Results	Messages
(No column name)	
1	1392/02/13 13:12:31

دنیاله دارد...

نظرات خوانندگان

نویسنده: فرشید علی اکبری
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۰:۳۴

از دوستان اگه کسی نحوه استفاده این تکنیک رو توی EF Code First میدونه لطفا مطلب رو Share کنه تا امکان استفاده این دسته از عزیزان هم از زحمتی که آقای قنادی کشیدن باشه با تشکر.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۰:۴۲

```
public DateTime AddDate { set; get; }
```

فیلد محاسباتی سمت برنامه که در بانک اطلاعاتی قرار نخواهد گرفت //
`[NotMapped]`
`public string PersianDate`
`{`
 `get { return MyDateConverter(AddDate); }`
`}`

در `code first` از ویژگی `NotMapped` استفاده کنید تا بتونید با استفاده از تابع کمکی تبدیل تاریخ خودتون راحت این تبدیلات رو انجام بدید. در بانک اطلاعاتی `DateTime` استاندارد رو ذخیره کنید، در سمت کدها برای مسایل نمایشی از خاصیت `PersianDate` که در `NotMapped` تعریف شده، میشه استفاده کرد. به علاوه روی `DateTime` استاندارد راحت میشه کوئری‌های LINQ رو اجرا کرد بدون محدودیت. زمانیکه مثلا یک `List` تهیه شد، قسمت بعدی، کار نمایشی است که از خاصیت `NotMapped` میشه کمک گرفت. این روش با تمام بانک‌های اطلاعاتی کار می‌کنه.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۲/۱۴ ۱۱:۴۱

```
[Column(TypeName = "xml")]
public string XmlValue { get; set; }

[NotMapped]
public XElement XmlValueWrapper
{
    get { return XElement.Parse(XmlValue); }
    set { XmlValue = value.ToString(); }
}
```

روش عمومی کار با نوع‌های خاصی که در EF تعریف نشدن، استفاده از ویژگی `Column` و مشخص کردن `Type` آن است؛ مانند مثالی که در بالا ملاحظه می‌کنید. البته این نوع خاص، در سمت کدها باید به صورت رشته تعریف شود. مثلا از سال 2005 به این طرف فیلد XML به SQL Server اضافه شده. اما نمی‌شود ازش در EF به همون شکل XML استفاده کرد. باید تبدیلش کنی به `String` تا قابل استفاده بشه. یک نمونه دیگر ش نوع خاص `Spatial` هست که در نگارش‌های اخیر SQL Server اضافه شده (`geography` و `geometry`). این مورد فقط از EF 5.0 به بعد پشتیبانی توکاری ازش ارائه شده. یا برای `hierarchyID` در EF معادل وجود نداره. برای تعریف این مورد نیز در یک مدل باید از `string` استفاده کرد.

بعد اگر این نوع خاص (که الان به صورت رشته دریافت شده) قابل نگاشت به نوعی مشخص در سمت کدهای برنامه بود (یعنی صرفا یک رشته ساده نبود) مثلا می‌شود از ویژگی `NotMapped` برای تبدیل آن و تعریف آن به شکل یک فیلد محاسباتی استفاده کرد.

با سلام

پروژه ای که شرح دادین رو ایجاد و در SQL server 2012 , Publish زمانی که جدول را در Object Browser با Mode ویرایش باز می کنم هیچ مشکلی وجود ندارد و داده ها درست نمایش داده می شوند اما زمانی که با دستورات T-SQL کار می کنم مقادیر را به صورت یک رشته از کاراکترها نمایش می دهد که نامفهوم می باشد . تصویر اجرای کوئری ها را می دارم لطفا راهنمایی کنید .

	Id	TestDate
▶	۱	۱۳۹۲/۰۲/۰۹ ۰۰:۰۰:۰۰
	۲	۱۳۹۲/۰۲/۰۹ ۲۲:۴۰:۰۰
	۳	۱۳۹۲/۱۲/۲۹ ۲۲:۴۰:۰۰
*	NULL	NULL

```
select *
from TestTable
```

Results Messages

	Id	TestDate
	۱	۰۰:۰۰:۰۰ ۱۳۹۲-۰۲-۰۹
	۲	۰۰:۰۰:۰۰ ۱۳۹۲-۰۲-۰۹ ۲۲:۴۰:۰۰
	۳	۰۰:۰۰:۰۰ ۱۳۹۲-۱۲-۲۹ ۲۲:۴۰:۰۰

```
select dbo.GetCurrentDateTime()
```

Results Messages

	(No column name)
۱	۰۰:۰۰:۰۰ ۱۳۹۲-۰۲-۰۹ ۲۲:۴۰:۰۰

به صورت پیش فرض با فرمت serialized binary value بازگشت داده می‌شوند.

```
SELECT id, TestDate.ToString() FROM TestTable;
```

در ابزارهای کوئیری گرفتن ad-hoc مثل SSMS باید یک متده استوون اضافه کنید تا مقدار نمایشی واضحی حاصل شود.

بیش از آشنایی با FileTable نیاز است که پیشینه‌ای از شیوه‌های ذخیره‌سازی فایل و یا بهتر بگوییم BLOB در SQL Server را داشته باشیم. نخستین شیوه‌ی نگهداری فایل استفاده از Image است که در 2000 SQL Server کاربرد داشت و هم‌اکنون استفاده از آن به دلیل کاهش بسیار کارآیی منسخه شده است. به دلایل مشکلات بسیار فراوان Image هم‌زمان بسیاری از طراحان پایگاه داده‌ها، جهت کاهش حجم جدول‌ها و پیروی آن حجم پایگاه داده‌ها، فایل را در سیستم‌فایل نگهداری می‌کردند و تنها مسیر آن را در فیلدی از نوع کارکتری در پایگاه‌داده‌ها ذخیره می‌کردند. این روش هرچند از حجم پایگاه داده‌ها می‌کاست ولی به دلیل عدم دخالت SQL Server در مدیریت فایل‌ها مشکلات دیگری را به وجود آورد.

از SQL Server 2005 نوع داده‌ی varbinary(max) معرفی شد که برخی از چالش‌های به کاربری Image را کاست و درباره‌ی بسیاری از موارد مانند ذخیره‌ی عکس پرسنلی هنوز هم کاربرد دارد؛ ولی توجه داشته باشید که استفاده از این فیلد فقط برای فایل‌های کم‌تر از 256 کیلوبایت سفارش شده است و برای بالاتر از آن، کارآیی کاهش فراوانی خواهد یافت.

در SQL Server 2008 نوع داده‌ی جدیدی به نام FileStream به وجود آمد به این شکل که یک DataGroup از نوع FileStream به پایگاه‌داده افزوده می‌شود و در واقع با یک پوشش در سیستم‌فایل در پیوند است. از این پس هنگام ساخت یک جدول به جای استفاده از نوع داده‌ی varbinary از نوع FileStream استفاده می‌کنیم با مد نظر داشتن این نکته که حتماً باید یک فیلد از نوع Uniqueidentifier هم در آن جدول تعریف شده باشد. شیوه‌ی کار نیز به این صورت خواهد بود که خود رکورد در جدول ذخیره می‌شود و فقط محتوای فایل در آن مسیری از NTFS ذخیره می‌شود. برخلاف روش درج مسیر فایل در جدول که پس از حذف رکورد، فایل هم‌چنان در سیستم فایل می‌ماند؛ این بار با حذف رکورد فایل مربوطه نیز حذف خواهد شد. افزون بر این مدیریت پشتیبانی از فایل‌ها نیز بر عهده‌ی پایگاه داده‌ها خواهد بود. اندازه‌ی فایل‌ها در FileStream محدودیت‌های پیشین را نخواهد داشت و شما به اندازه‌ی حجم درایو هاردیسک می‌توانید فایل در آن ذخیره کنید. نکته‌ی دیگر درباره‌ی فایل‌های با حجم سنگین که می‌توانید Stream مربوط به یک فایل را به صورت بخش‌بخش در سمت مشتری بارگذاری کنید و به او نشان دهید. در FileStream امنیت و تراکنش فایل‌ها بر عهده‌ی SQL Server است و از این دیدگاه بسیار ساده‌تر و کارآتر از FileSystem است. (برای آشنایی بیشتر با FileStream از نوشتار [این نوشتار](#) از مهندس وحید نصیری را مطالعه کنید.)

گونه‌ی FileTable از ویژگی‌های نوین SQL Server 2012 است که تکمیل‌کننده‌ی FileStream است. FileTable آمیزشی از hierarchyid و سیستم‌فایل ویندوز برای ارائه‌ی توانایی‌های نوین مدیریت BLOB در SQL Server است. همان‌گونه که از دو واژه‌ی تشکیل‌دهنده‌اش پیداست؛ هم‌زمان یک جدول و یک سیستم‌فایل معمولی است. FileStream به هر روی یک جدول از پایگاه‌داده‌های SQL Server است با یک تفاوت که ساختار آن از پیش تعریف شده است. ستون‌های FileStream و نوع داده‌ی آن از پیش توسط SQL Server مشخص شده است. ستون‌های تشکیل‌دهنده‌ی در برگیرنده‌ی جدول زیر است:

نام ستون	DataType	شرح
stream_id	uniqueidentifier ROWGUIDCOL	شناسه‌ی یکتای ردیف
file_stream	varbinary(max) FILESTREAM	محتوای فایل، اگر پوشه باشد؛ Null است.
name	nvarchar(255)	نام فایل یا پوشه
path_locator	hierarchyid	جای فایل یا پوشه در سیستم فایل
creation_time	datetimeoffset(7)	زمان ایجاد
last_write_time	datetimeoffset(7)	زمان و اپسین ویرایش
last_access_time	datetimeoffset(7)	زمان و اپسین دسترسی
is_directory	bit	۰ = فایل / ۱ = پوشه
is_offline	bit	
is_hidden	bit	
is_READONLY	bit	
is_ARCHIVE	bit	
is_SYSTEM	bit	
is_TEMPORARY	bit	

هر ردیف از FileTable نماینده‌ی یک فایل یا پوشه در File System است. ستون path_locator که از نوع hierarchyid است نشان‌دهنده‌ی مسیر یک فایل یا پوشه است. که از 2008 SQL Server معرفی شده است؛ بهترین نوع داده برای نگهداری ارتباط ساختار سلسله‌مراتبی مانند چارت سازمانی، درخت تجهیزات یک کارخانه و یا در همین نمونه درخت فایل‌ها و پوشه‌ها است. پس می‌توانیم از همه‌ی امکانات hierarchyid در FileTable نیز برخوردار شویم. این که این فایل به ترتیب در چه پوشه‌هایی قرار گرفته است یا این که این پوشه شامل چه فایل‌ها یا پوشه‌هایی خواهد بود. این که پوشه‌هایی هم‌فرزنده‌ی جاری کدام است و یا یا توابع مربوط به جابه‌جایی فایل‌ها و پوشه‌ها.

دنباله دارد...

نظرات خوانندگان

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۶:۲۶

با سلام و خسته نباشید
مطلوب خیلی خوبی بود فقط اگر امکان دارد در یک بخش درباری hierarchyid نیز کامل به همراه مثال توضیح دهید
با تشکر

نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۷:۲۷

درود بر شما
پیشنهاد خوبی است. پیش از این نوشتاری در این باره نوشته بودم (هرچند منتشر نکرده ام). ولی با یادداشت شما به این اندیشه افتادم که مروری بر این بحث در SQL Server 2012 داشته باشم و با ویرایشی نوین، در این تارنمای منتشر کنم.

نویسنده: ali.rezayee
تاریخ: ۱۳۹۲/۰۲/۱۷ ۱۹:۳۱

با سلام و تشکر از این مطلب خوب.
آیا نوع فیلدها قابل تغییر است؟
برای مثال در فیلد Name چون varchar است نمیتوان نام فارسی برای فایل ثبت کرد، آیا امکان تغییر آن به Nvarchar ممnon از شما.

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۷ ۲۲:۵۸

در اصل nvarchar هست که به نظر در جدول فوق باید اصلاح بشه.

نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۱۸ ۶:۵۳

بله. اشتباه از من بود. این فیلدها غیرقابل تغییر هستند ولی هیچ مشکلی هم با زبان پارسی ندارند.

نویسنده: سید امیر سجادی
تاریخ: ۱۳۹۲/۰۹/۱۴ ۲۰:۱۹

سلام. ممنون از مطلب مفیدتون. فقط یک مشکلی هست و اون عدم پشتیبانی توسط Linq2Sql هست. فکر کنم فقط با DataSet مشکلی نداشته باش و اون هم بخاطر این هست که نوع hierarchyid به نوع(867) nvachar تبدیل میکنه.
لطفاً اگه برای این مشکل راه حلی هست بنویسید که دوستان هنگام استفاده به مشکل برنخورند

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۴ ۲۰:۳۱

Linq2Sql که خیلی وقت هست با EF جایگزین شده. EF هم از نوعهای خاص دیتابیسها عموماً پشتیبانی نمیکنه. بحث EF هم نیست. اکثر ORMهای موجود همینطور هستند.

نویسنده: سید امیر سجادی

یعنی فقط باید از DataSet استفاده کرد !

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۱۵ ۱:۱

ADO.NET می‌توانه خروجی از نوع DataSet هم بده؛ ولی اسمش DataSet تمام ها هم برقرار از ADO.NET کار می‌کنند. بنابراین اگر تصور کنیم که ابداً نمی‌شود با این‌ها کار کرد، خیر. اینطور نیست. شما با این‌ها هم می‌توانی مستقیماً SQL بنویسی-[ef](#) تا به قابلیت‌هایی دسترسی پیدا کنی که در ORM پیش بینی نشده. [code-first-13](#)

نویسنده: سعید
تاریخ: ۱۳۹۲/۱۰/۱۴ ۲۳:۵۱

سلام
چطور امنیت فایل‌ها رو برقرار کنیم؟ منظورم اینه که کسی نتوونه اونا رو پاک یا مشاهده و یا اضافه کنه، مگه اینکه ما بهش اجازه بدیم راهی برای این کار هست؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۱۵ ۰:۴۵

مگه کاربران شما مستقیماً با فایل سرور کار می‌کنند؟ روال عادی آن کار از طریق یک برنامه مثلاً وب هست. امنیت رو در اونجا پیاده سازی کنید.

نویسنده: سعید
تاریخ: ۱۳۹۲/۱۰/۲۲ ۱۱:۱۱

من دارم از اپلیکیشن استفاده می‌کنم. از داخل برنامه درست هست، یعنی همونی که شما می‌گی. ولی اگه کسی از بیرون برنامه بخواهد این کارو کنه چی؟
مستقیم بره تو پوشه و فایل‌ها رو تغییر بده. من می‌خواهم بدون `sql` می‌توونه جلوی این کارو هم بگیره؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۲۲ ۱۲:۱۲

- به چه نحوی از SQL Server استفاده می‌کنید؟ آیا سرور و برنامه دسکتابپ شما روی یک کامپیوتر هستند؟ برای اینکار بهتر است از SQLite یا SQL CE استفاده کنید؛ یا حتی LocalDB. هدف از SQL Server نصب آن روی یک سرور و خدمات دهی به چندین و چند کامپیوتر تحت شبکه است. برای استفاده روی یک کامپیوتر یعنی کسب و کار کوچک و عملانیازی به SQL Server 2012 ندارد اینکار. زندگی مصرف کننده را سخت نکنید. نصب و نگهداری یک سرور کار هر شخصی نیست و برای سازمان‌ها طراحی شده و نه مصارف کوچک تک کاربره دسکتابپ.

- با این توضیح اگر کسی به سرور شما دسترسی دارد، آیا نمی‌تواند مثلاً اگر فایل‌ها در دیتابیس ذخیره می‌شوند، اون‌ها رو دستی با یک کوئری حذف کند؟ امنیت کار با سرور این‌جا است که مطرح می‌شود و همچنین اطمینان به ادمین‌ها.

- در مورد امنیت [file table](#) مراجعه کنید به مستندات مایکروسافت. مثلاً: [FileTables are secured by SQL Server security](#) only

ستون دیگر **stream_id** نام دارد که از نوع **uniqueidentifier ROWGUIDCOL** نیز ناگزیر به تعریف چنین ستونی بودیم. بنابراین **FileTable** استثناء نیست و در اینجا نیست چنین فیلدی توسط SQL Server تعریف می‌شود. اگر فایل‌ها و پوشش‌ها جایه‌جا نمی‌شند می‌توانستید از هر دو ستون **stream_id** یا **path_locator** برای شناسایی یک رکورد از جدول بهره ببرید. ولی با جایه‌جایی یک فایل و یا به عبارت دیگر تغییر پدر آن در ساختار سلسله‌مراتبی، مقدار نیز تغییر می‌کند، پس ناگزیر به استفاده از این ستون برای ارجاع به یک ردیف در جدول هستیم. هر ردیف از جدول نمایان‌گر یک فایل یا پوشش است، بنابراین به ستونی نیاز داریم که بتوانیم این موضوع را نشان دهیم. بر این پایه از ستون **is_directory** می‌بریم که 1 بودن آن نشان‌دهنده این است که این ردیف از جدول به یک پوشش ارجاع دارد. نام فایل یا پوشش در ستونی به نام **name** نگهداری می‌شود که رشته‌ای از نوع **nvarchar(255)** است. افزون بر این ستون، ستون‌های دیگری نیز در این جدول وجود دارد که ویژگی‌های یک فایل مانند پنهان‌بودن، فقطخواندنی و ... توسط آن به دست می‌آید. ستون پسین **file_stream** نام دارد که برای پوشش‌ها، محتوای آن **Null** است. علت آن این است که محتوای واقعی فایل در این ستون نگهداری می‌شود. در واقع یک **varbinary(max)** با ویژگی‌های **fileStream** است که محتوای باینری آن در سیستم فایل NTFS ذخیره می‌شود. مدیریت پشت صحنه‌ی این ستون بر عهده‌ی SQL Server است. افزون بر این 14 ستون، هر **FileTable** شامل سه ستون محاسباتی به شرح زیر است:

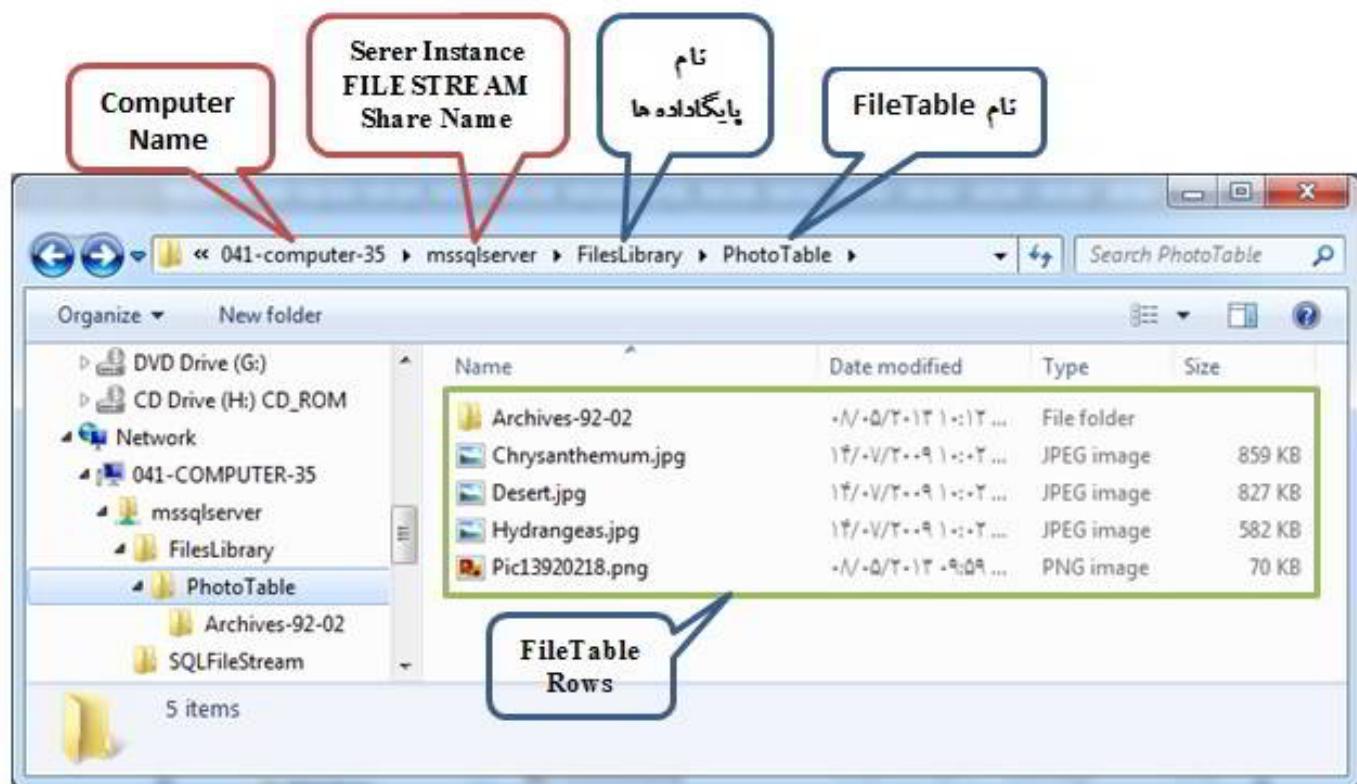
نام ستون	DataType	شرح
parent_path_locator	hierarchyid	مسیر به دست آمده از path_locator
file_type	nvarchar(255)	گونه‌ی فایل به دست آمده از name
cached_file_size	bigint	اندازه‌ی فایل به دست آمده از file_stream

ستون **parent_path_locator** نتیجه‌ی فراخوانی تابع **GetAncestor(1)** در ستون **path_locator** است که جهت به دست آوردن پوششی پدر یک فایل و پوشش استفاده می‌شود. ستون **file_type** که از مقدار رشته‌ای ستون **name** تجزیه شده است، پسوند فایل را برمی‌گرداند. و ستون **cached_file_size** اندازه‌ی بایت ذخیره‌شده ستون **file_stream** را برمی‌گرداند. با این ساختار ثابت در اینجا، هر **FileTable** هر آن‌چه از **File System** نیاز دارد در یک پوششی اشتراکی به شما می‌دهد.

این یعنی نمایش بی‌واسطه **FileTable** به هر کاربر یا برنامه. به طوری که برای نمایش یا بهروزرسانی جدول می‌توانید از روش استاندارد **I/O API** مانند **Kafka** و **RHadoop** با **Windows Explorer** یا برنامه‌نویسی با **System.IO.FileStream** و **System.IO API** های ویندوز استفاده کنید. این چنین:

- ایجاد یک فایل یا پوشش در سیستم فایل -> افزودن یک ردیف به جدول

- افزودن یک ردیف به جدول -> ایجاد یک فایل یا پوشش در سیستم فایل



با کپی فایل‌ها در مسیر بالا، به صورت خودکار رکوردهای زیر در جدول PhotoTable در پایگاهداده‌ها افزوده می‌شود:

	stream_id	file_stream	name	path_locator	parent_pa
1	FABBA0DA-A1B7-E211...	0x89504E470D0A1A0A0000000D49484452...	Pic13920218.png	0xFF1E127B0A0A9B2FDD353FFA0403...	NULL
2	A0CB07E7-A1B7-E211...	0xFFD8FFE000104A4649460001020100600...	Hydrangeas.jpg	0xFECCB4A9F110A38FD391534845521...	NULL
3	A2CB07E7-A1B7-E211...	0xFFD8FFE000104A4649460001020100600...	Chrysanthemum.jpg	0xFD64A656CE5A91CFDB12524D522...	NULL
4	A4CB07E7-A1B7-E211...	0xFFD8FFE000104A4649460001020100600...	Desert.jpg	0xFF6A01070D9AB3EFC61352DE20FB1...	NULL
5	3DCCC316-A2B7-E211...	NULL	Archives-92-02	0xFD00916BA6D2930FCFD2351E24281...	NULL
6	63B6DF2A-A2B7-E211...	0xFFD8FFE000104A4649460001020100600...	Tulips.jpg	0xFD00916BA6D2930FCFD2351E24281...	0xFD009

به طور خلاصه پیش از این برای افزودن به FileStream دو راه کار پیش رو داشتید. یکی استفاده از T-SQL و دیگر sqlFileStream اکنون SQL Server راه کار سوم را پیشنهاد می‌کند. استفاده از FileStream در این روش به طور خودکار پر می‌شود.

پیش از ساخت یک FileTable بیان این نکته دارای اهمیت است که با کپی فایل‌ها و پوشش‌ها هیچ چیز جدیدی به NTFS افزوده نمی‌شود بلکه محتوای فایل به FileTable افزوده می‌شود و SQL Server با بررسی همزمان FileTable و FileStream نمایشی از ردیف‌های FileTable به صورت یک پوششی اشتراکی نشان می‌دهد. این نکته پاسخی به این پرسش خواهد بود که آیا با استفاده از FileTable حجم پایگاهداده‌ها دو برابر خواهد شد و در نتیجه دشواری‌ها و چالش‌های نگهداری و پشتیبانی را پیش رو خواهیم داشت!؟ که پاسخ "خیر" خواهد بود.

ایجاد یک FileTable

پیش از این در همین تارنما، روش فعال کردن FileStream در SQL Server را آموزش دیده اید. اگر درست به خاطر داشته باشید، چیزی شبیه به دستورهای زیر بود:

```
CREATE DATABASE MyFileArchive
ON PRIMARY
(NAME = MyFileArchive_data,
FILENAME = 'C:\Demo\MyFileArchive_data.mdf'),
FILEGROUP FileStreamGroup CONTAINS FILESTREAM
(NAME = PhotoFileLibrary_blobs,
FILENAME = 'C:\Demo\MyFiles')
LOG ON
(NAME = PhotoFileLibrary_log,
FILENAME = 'C:\Demo\MyFileArchive_log.ldf')
```

متکی است؛ بر این پایه پیش از ایجاد یک FileStream باشد FileTable را روی پایگاهدادهها فعال کنیم. این کار با یک تعریف درست توسط بند FILEGROUP...CONTAINS FILESTREAM انجام می‌شود.

برای ایجاد FileTable تنها کافی است که بند WITH FILESTREAM را به دستور CREATE DATABASE بیفزایید. (یا برای فعال کردن روی یک پایگاهداده ساخته شده بند ALTER DATABASE SET FILESTREAM را در دستور بنویسید). در این بند، از SQL Server برای نام‌گذاری یک پوشه برای پایگاهداده‌ها استفاده می‌کنیم. این پوشه در یک پوشه ریشه به نام DIRECTORY_NAME نمایش داده خواهد شد. بخش دوم بند NON_TRANSACTED_ACCESS=FULL است که دسترسی غیرتراکنشی را فعال می‌کند. با این کار برای هر FileTable در پایگاه داده یک زیرپوشه درون پوشه‌ای که به نام DIRECTORY_NAME نام‌گذاری شده است؛ ساخته می‌شود.

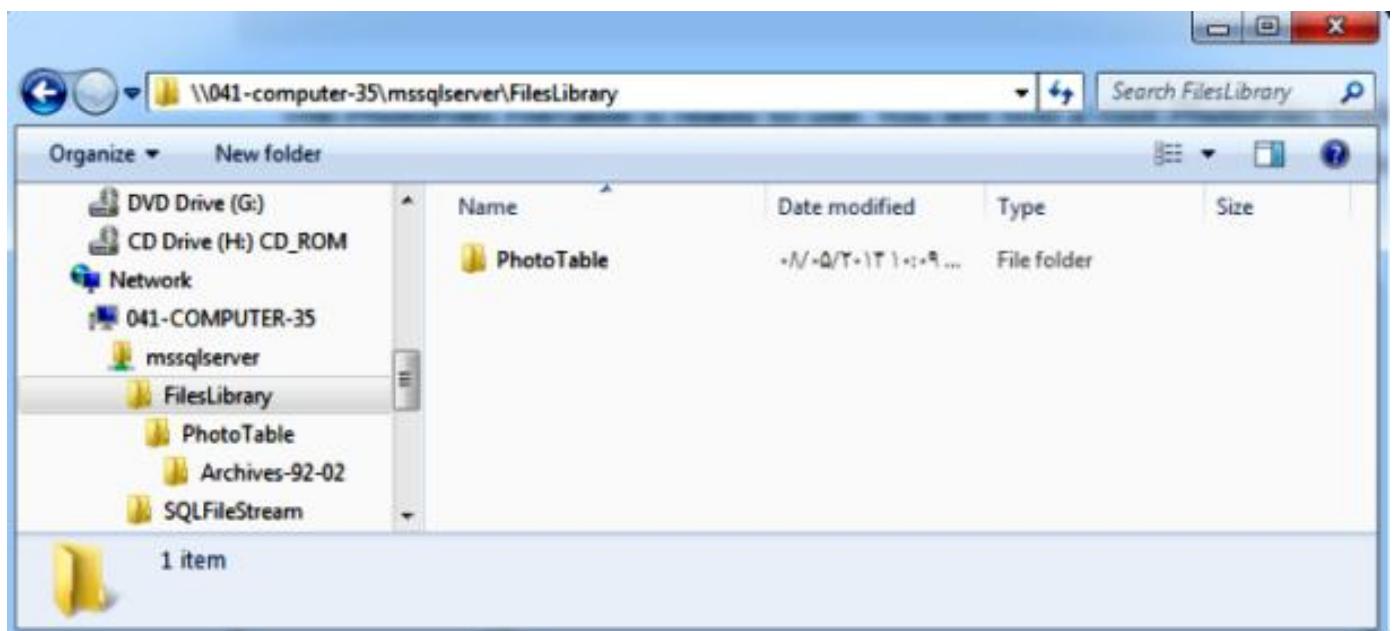
با توجه به آنچه گفته شد برای ایجاد یک پایگاهداده با امکان ساخت FileTable دستورهای زیر را اجرا کنید:

```
CREATE DATABASE MyFileArchive
ON PRIMARY
(NAME = MyFileArchive_data,
FILENAME = 'C:\Demo\MyFileArchive_data.mdf'),
FILEGROUP FileStreamGroup CONTAINS FILESTREAM
(NAME = PhotoFileLibrary_blobs,
FILENAME = 'C:\Demo\MyFiles')
LOG ON
(NAME = PhotoFileLibrary_log,
FILENAME = 'C:\Demo\MyFileArchive_log.ldf')
WITH FILESTREAM
(DIRECTORY_NAME='FilesLibrary',
NON_TRANSACTED_ACCESS=FULL)
```

اکنون برای ساخت یک FileTable درون این پایگاهداده‌ها از دستور زیر استفاده کنید:

```
USE MyFileArchive
GO
CREATE TABLE PhotoTable AS FileTable
GO
```

توجه داشته باشید که چون ستون‌های FileTable از پیش تعریف شده است؛ ایجاد آن فقط با نوشتتن دستور امکان پذیر است و مانند یک Table عادی از محیط کاربری SQL Server نمی‌توان بهره برد. در Object Explorer از گره‌های Tables، FileTables را باز کنید و روی جدولی که همان‌کنون ساختیم راست‌کلیک کنید. با انتخاب گزینه‌ی Explore FileTable Directory پنجره‌ی زیر باز می‌شود:



دنباله دارد ...

نظرات خوانندگان

نوبنده: ali.rezayee | تاریخ: ۱۳۹۲/۰۳/۰۴ ۱۶:۳۶

با سلام
موضوع بسیار جالب و مهمی است، لطفاً سریع‌تر ادامه دهید.

نوبنده: علی رضایی | تاریخ: ۱۳۹۳/۰۴/۲۷ ۱۹:۵۸

بهروش‌های استفاده از FILESTREAM

نوبنده: M.Shakeri | تاریخ: ۱۳۹۳/۰۵/۰۸ ۱۰:۱۴

فکر می‌کنم برای افزودن این قابلیت به دیتابیس دسترسی به دیتابیس master نیاز باشه که هاست‌ها این امکان رو نمیدن.

نوبنده: محسن خان | تاریخ: ۱۳۹۳/۰۵/۰۸ ۱۲:۸

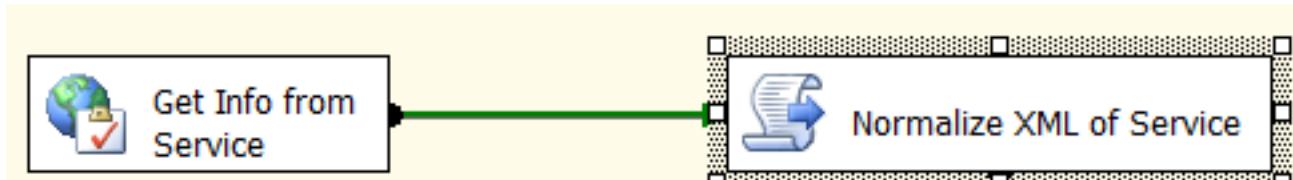
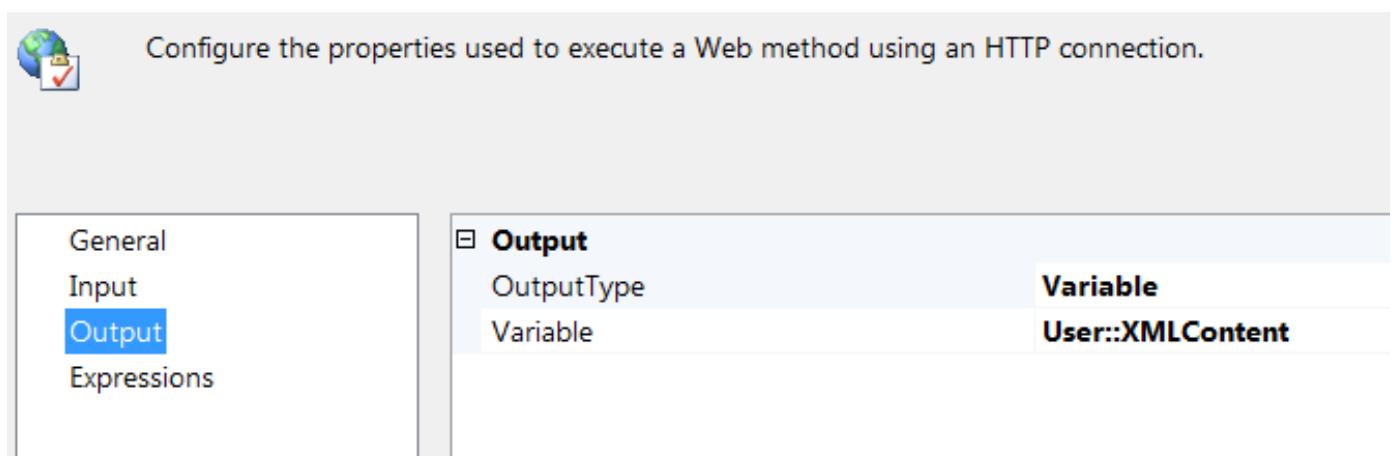
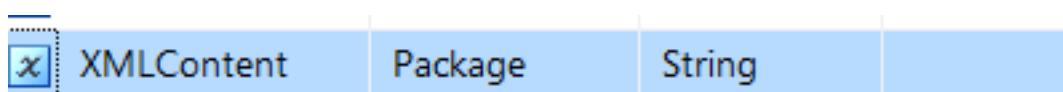
خیلی از قابلیت‌ها مخصوص سازمان‌ها طراحی می‌شون.

نوبنده: Hamidbp | تاریخ: ۱۳۹۳/۰۷/۱۴ ۱۴:۴۰

می‌خواستم بدونم چطور می‌شه یک فایل رو که در Filetable ذخیره شده، به اصطلاح Fetch کنم و توسط برنامه مربوطه (مثلا word ya ...) نشونش بدم؟

ممکن است در موقعی نیاز به اطلاعات استخراج شده از وب سرویس داشته باشید که در همان مقطع زمانی به آن دسترسی ندارید . مسلما برای این منظور باید آن اطلاعات را ذخیره کرده تا در صورت نیاز بتوان به آنها رجوع کرد . یکی از راه حل‌ها ذخیره آن در پایگاه داده (در اینجا Sql Server) است که در این پست به کمک امکانات BIDS در پکیج‌های SSIS و کوئری‌های SQL این مشکل را ببرطرف میکنیم. برای مشاهده نحوه استخراج اطلاعات از وب سرویس [به اینجا](#) مراجعه کنید .

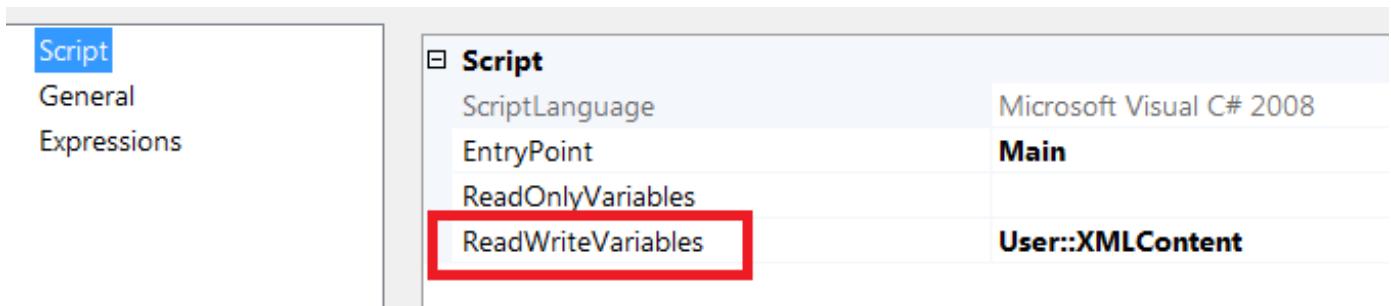
تنها تفاوتی که در این پست در کار با سرویس با پست اشاره شده در بالا وجود دارد ذخیره اطلاعات استخراج شده است که در آن پست در یک فایل xml ذخیره شدند ولی در اینجا ما نیاز داریم تا اطلاعات را در یک متغیر با حوزه کاری Package ذخیره کنیم (به این معنی که مختص به همان flow نباشد و در تمام پکیج دیده شود)



به دلیل اینکه هدر xml خروجی از سرویس دارای چندین namespace هست هنگام کار با آنها به مشکل خواهیم خورد. (هم هنگام کار با xml task ها و هم هنگام کار با xml در sql)

```
<ArrayOfUserInfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
  <User>
```

به همین دلیل باید این قسمت از محتوا را حذف کرد . برای همین پس از گرفتن اطلاعات از سرویس آن را به کمک یک task حذف می کنیم

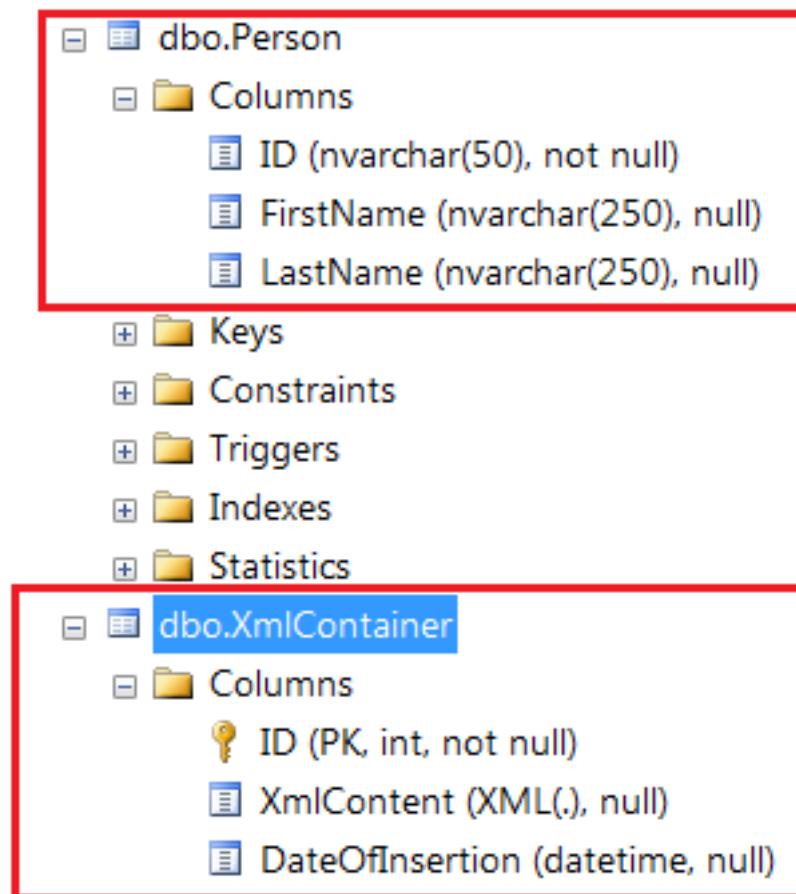


```
public void Main()
{
    //xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://tempuri.org/"
    object o = "null";
    byte[] emptyBytes = new byte[0];
    try
    {
        Dts.VariableDispenser.LockForRead("User::XMLContent");
        Variable xmclc = Dts.Variables["User::XMLContent"];
        o = xmclc.Value;
        if (o != null)
        {
            string s = o.ToString();
            s = s.Replace("xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"", "");
            s = s.Replace("xmlns:xsd=\"http://www.w3.org/2001/XMLSchema\"", "");
            s = s.Replace("xmlns=\"http://tempuri.org\"", "");
            Dts.VariableDispenser.Reset();
            Dts.VariableDispenser.LockForWrite("User::XMLContent");
            Dts.Variables["User::XMLContent"].Value = s;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Error SSIS : " + Environment.NewLine + Environment.NewLine + ex.Message, "Error on Package...");
        Dts.TaskResult = (int)ScriptResults.Failure;
    }
    finally
    {
        Dts.VariableDispenser.Reset();
    }

    Dts.Log("nameSpace Removed Successfully...", 0, emptyBytes);
    Dts.TaskResult = (int)ScriptResults.Success;
}
```

در این مرحله اطلاعات استخراج شده را باید در SQL درج کنیم . برای همین ساختاری که باید اطلاعات را در SQL نگه دارد را در دیتابیس ایجاد می کنیم :

جدول person برای نگهداری اطلاعات سرویس XMLContainer برای نگهداری XMLهای سرویس .(برای داشتن History



برای درج هم از SP استفاده می‌کنیم :

```
ALTER PROCEDURE [dbo].[Add_step1]

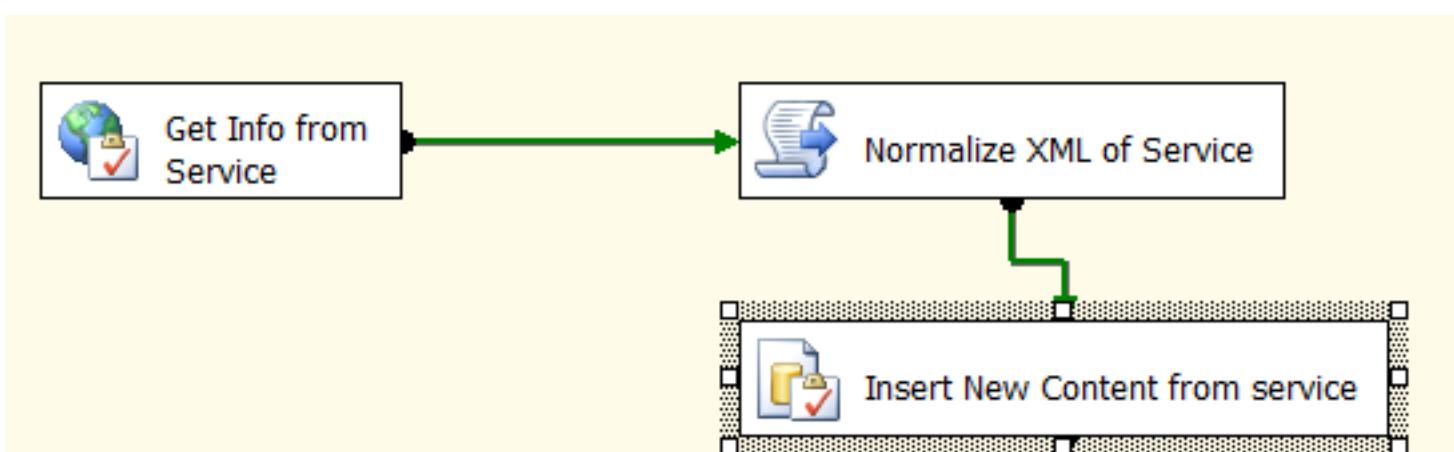
    @xml XML
AS
BEGIN
    SET NOCOUNT ON;

    INSERT INTO dbo.XmlContainer
    values( @xml, GETDATE() )

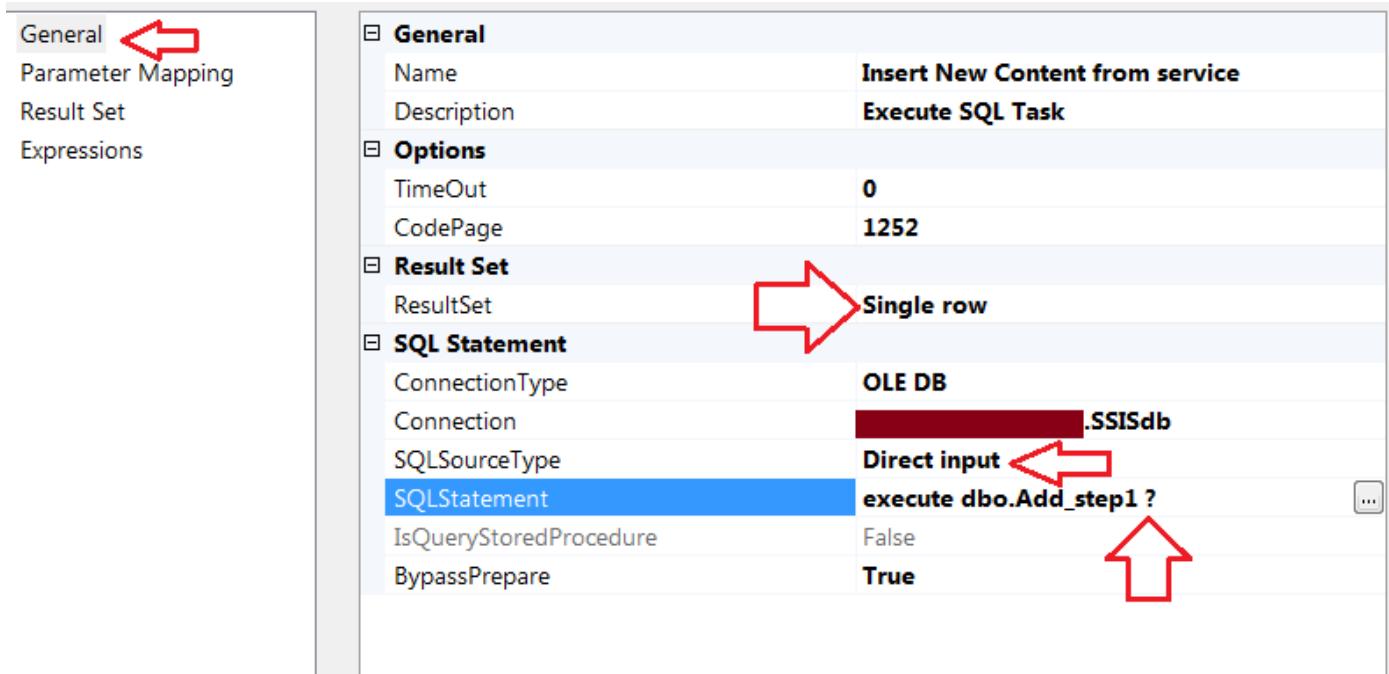
    DECLARE @id INT
    SET @id = SCOPE_IDENTITY();

    SELECT @id AS Result

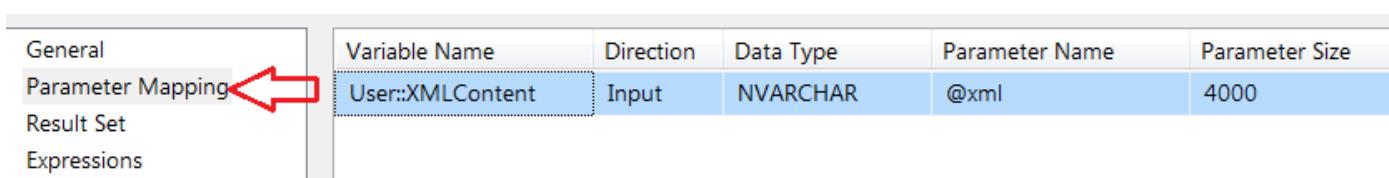
END
```



: SQL Task و پیکربندی



نکته اول ایجاد کانکشن به پایگاه داده است که در اینجا از نمایش جزئیات آن صرف نظر شده است . نکته دیگری پارامتر SP است که چون یک پارامتر دارد یک علامت سوال قرار میگیرد . اگر چند پارامتر بود به صورت علامت‌های سوال با ویرگول از هم جدا می‌شوند .
سپس در بخش parameter mapping به ترتیب مقدار دهی می‌شوند :



و مقدار خروجی SP که شناسه آیتم درج شده است را در یک متغیر نگهداری می‌کنیم :

The screenshot shows the 'Result Set' tab of the SSIS Result Set Editor. On the left, there is a sidebar with tabs: General, Parameter Mapping, Result Set (which is selected and highlighted in blue), and Expressions. The main area contains a table with two columns: 'Result Name' and 'Variable Name'. In the 'Result Name' column, the value 'Result' is listed. In the 'Variable Name' column, the value 'User::SqlResultId' is listed.

Result Name	Variable Name
Result	User::SqlResultId

برای قدم بعد می خواهیم اطلاعات موجود در XML استخراج شده در پایگاه داده را در جدول مربوطه ذخیره کنیم . برای این کار این SP را می نویسیم :

```
ALTER PROCEDURE [dbo].[transform_Step2]
@id int
AS
BEGIN

    SET NOCOUNT ON;
    DECLARE @trans CHAR(50), @xml XML
    SET @trans = 'MyTrans'
    BEGIN TRY
        BEGIN TRANSACTION @trans

        DELETE FROM Person -- empty table
        SELECT @xml = xc.XmlContent FROM XmlContainer xc
        WHERE xc.ID = @id
        DECLARE @dt DATETIME , @hoc int;
        EXEC sp_xml_preparedocument @hoc OUTPUT, @xml
        INSERT INTO Person (ID,FirstName,LastName)
        SELECT ID, FirstName, LastName
        FROM OPENXML(@hoc, 'ArrayOfUserInfo/UserInfo')
        WITH
        (
            ID [nvarchar](50) 'ID',
            FirstName [nvarchar](250) 'FirstName',
            LastName [nvarchar](250) 'LastName'
        )
        EXEC sp_xml_removedocument @hoc -- empty cache
        IF (@@ERROR <> 0 )
        BEGIN
            ROLLBACK TRANSACTION @trans
        END
        ELSE
        BEGIN
            COMMIT TRANSACTION @trans
            SELECT 'OK' AS 'ERROR'
        END

        END TRY
        begin CATCH
            ROLLBACK TRANSACTION @trans
            SELECT ERROR_MESSAGE() AS 'ERROR'
        END CATCH
    END
```

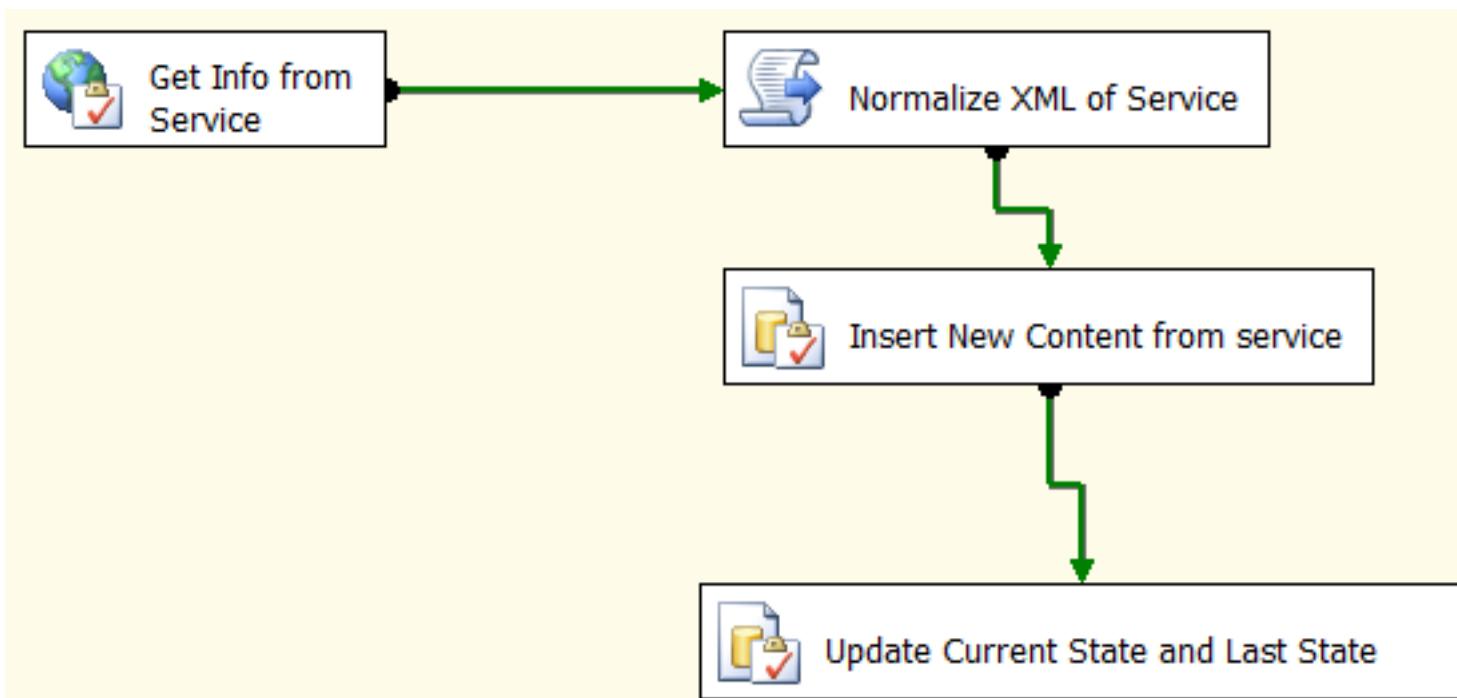
نکات مهم موارد زیر هستند :

1 - استفاده از xml parse برای کردن 1

2 - استفاده از sp سیستمی sp_xml_removedocument و sp_xml_preparedocument

3- اطلاعات شناسه و نام و نام خانوادگی نودهای inner text xml هستند . اگر این موارد به صورت attribute باشد باید قبل از نام آنها علامت @ قرار بگیرند.

پس از ایجاد این Sp باید آن را در package فراخوانی کنیم :



و پیکربندی این SQL Task :

نحوه انتقال اطلاعات استخراج شده از وب سرویس به کمک SSIS

General Parameter Mapping Result Set Expressions

General		Update Current State and Last State
Name		Execute SQL Task
Description		
Options		
TimeOut		0
CodePage		1252
Result Set		
ResultSet		Single row
SQL Statement		
ConnectionType		OLE DB
Connection		.SSISdb
SQLSourceType		Direct input
SQLStatement		execute transform_Step2 ?
IsQueryStoredProcedure		False
BypassPrepare		True

: SP این پارامترهای و

General Parameter Mapping Result Set Expressions

Variable Name	Direction	Data Type	Paramet...	Paramet...
User::SqlResultId	Input	LARGE_INTEGER	@id	0

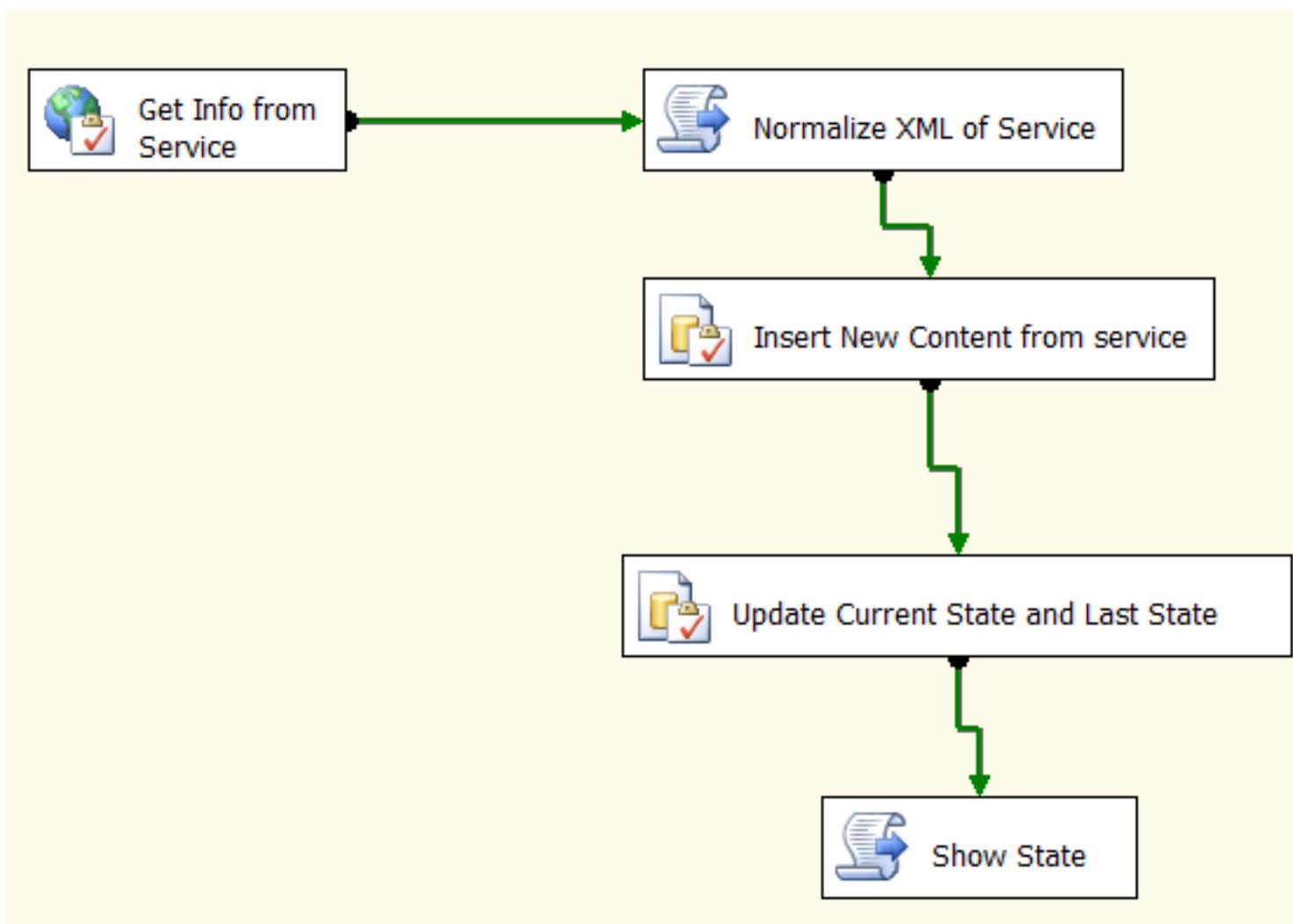
: ذخیره نتیجه تراکنش در متغیری در پکیج و

Name	Scope	Data Type	Value
FinalState	Package	String	

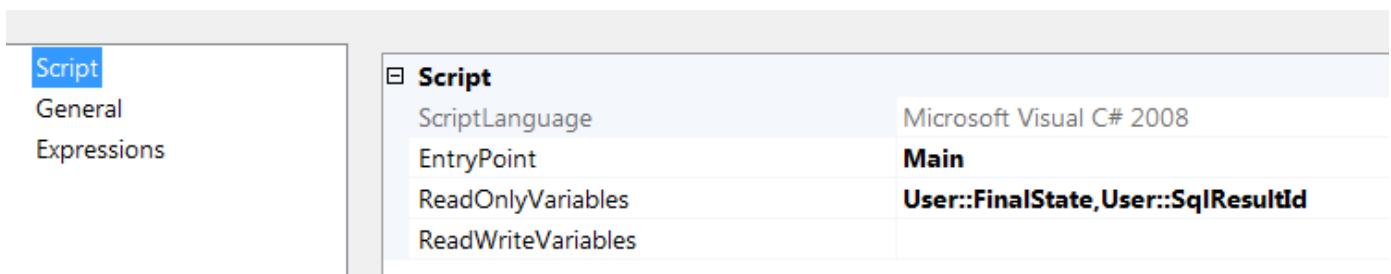
General Parameter Mapping Result Set Expressions

Result Name	Variable Name
ERROR	User::FinalState

و اکنون پکیج ما ظاهری شبیه به این مورد خواهد داشت :



: (Script Task می توانیم وضعیت تراکنش را به کاربر نمایش دهیم (به کمک facility



```
public void Main()
{
    const string _id = "User::SqlResultId";
    const string _err = "User::FinalState";

    Dts.VariableDispenser.LockForRead(_id);
    Dts.VariableDispenser.LockForRead(_err);

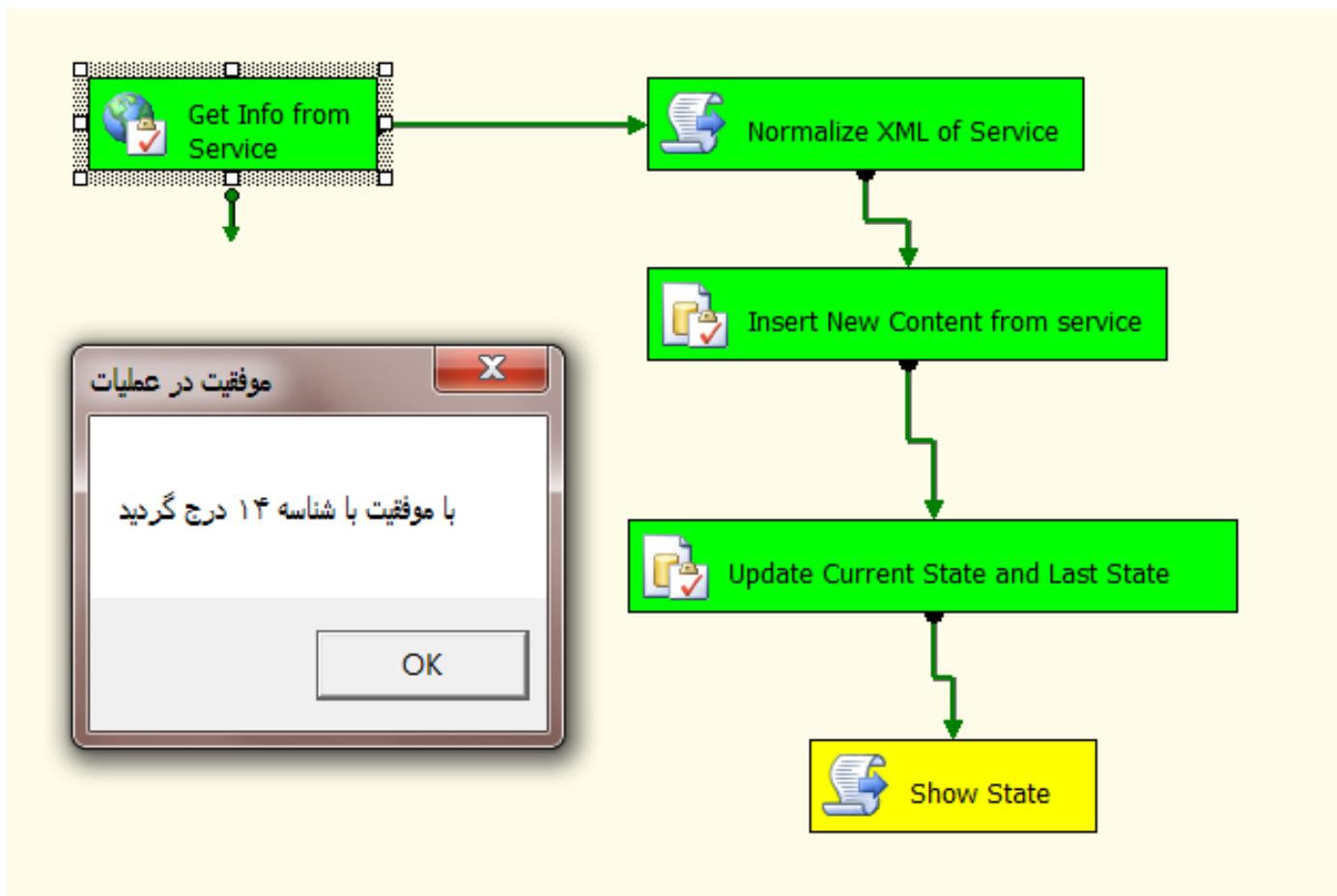
    string id = Convert.ToString(Dts.Variables[_id].Value);
    string error = Convert.ToString(Dts.Variables[_err].Value);

    string msg = "درج گردید " + id + " با موفقیت با شناسه " + id;
    Dts.VariableDispenser.Reset();

    if (error.Equals("OK"))
    {
        MessageBox.Show(msg, "موفقیت در عملیات");
        Dts.TaskResult = (int)ScriptResults.Success;
    }
    else
    {
        MessageBox.Show("Error : " + Environment.NewLine + error, "خطا در انجام عملیات");
        Dts.TaskResult = (int)ScriptResults.Failure;
    }

    // TODO: Add your code here
}
```

و تمام ...



Screenshot of a SQL Server Management Studio (SSMS) Results grid showing the output of the process:

ID	XmlContent	DateOfInsertion
1	<ArrayOfUserInfo><UserInfo><ID>bd36d158-f2bd-40d...	2013-05-16 21:36:44.84
2	<ArrayOfUserInfo><UserInfo><ID>5848557b-6482-4d8...	2013-05-16 21:44:21.49
3	<ArrayOfUserInfo><UserInfo><ID>5f465898-bce8-4153...	2013-05-16 21:45:02.61
4	<ArrayOfUserInfo><UserInfo><ID>0836a55b-3d8f-4ce...	2013-05-16 21:53:19.83
5	<ArrayOfUserInfo><UserInfo><ID>73670cd5-1d09-4a9...	2013-05-16 22:35:52.70
6	<ArrayOfUserInfo><UserInfo><ID>316e06a0-4063-479...	2013-05-16 22:36:30.89
7	<ArrayOfUserInfo><UserInfo><ID>c56ab672-9415-418...	2013-05-16 22:38:56.86
8	<ArrayOfUserInfo><UserInfo><ID>c991b4ee-a389-427...	2013-05-17 10:55:12.27

[موفق باشید](#)

نظرات خوانندگان

نویسنده: Amir
تاریخ: ۱۳۹۲/۱۰/۱۸ ۰:۴۴

با سلام و سپاس از مطالب مفیدتون

سوالی دارم ممنون میشم اگه جواب بدید

برای انتقال داده‌های هشت سرور که از نوع پارادوکس به 2012 SQL Server هستند آیا به غیر از روش SSIS راه دیگری وجود دارد؟ ناگفته نماند که داده‌ها حجم قابل توجهی دارند و با روش SSIS سرعت شبکه را کاهش می‌دهد.

نکته دیگر، داده‌ها به صورت کامل به SQL Server منتقل نمی‌شوند و بعضی جداول منتقل شده یا خالی هستند یا جداول تکراری داریم در SQL Server که در نتیجه باعث قابل اطمینان نبودن Data Warehouse می‌شود.

با سپاس

نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۳۹۲/۱۰/۱۸ ۱۲:۲۹

سلام

اجازه بدید اول به این نکته اشاره کنم که SSIS روش نیست بلکه یک ابزاره. روش مد نظر برای Migration داده، ETL نام دارد .SSIS (Extract , Transform , Load)

حالا برای انجام عمل ETL ابزارهای دیگری هم می‌شه پیدا کرد مثل DB2 Infosphere informatica یا اون‌های رو [اینجا](#) ببینید در مورد مشکلاتی که در باره داده‌ها یا شبکه می‌فرمایید هم باید یک plan مناسب برای این مهاجرت داده ای ایجاد کنید . جهت اطلاع عرض کنم که این پروسه ممکنه گاها تا یک ماه طول بکشه بسته به پلی که طراحی شده و شرایط و محدودیت هایی که موجوده. [اطلاعات بیشتر رو می‌تونید از اینجا کسب کنید](#) موفق باشید

نویسنده: Amiir
تاریخ: ۱۳۹۲/۱۰/۱۸ ۱۳:۴۵

ممنون از راهنماییتون

عنوان:	Full Text Search و Rank فیلدهای بازیابی شده
نویسنده:	دل محسن
تاریخ:	۱۰:۵۵ ۱۳۹۲/۰۳/۱۰
آدرس:	www.dotnettips.info
گروهها:	SQL Server, Full Text Search

حالی را در نظر بگیرید که بخواهید تعداد زیادی رکورد را که حجم هر رکورد هم قابل ملاحظه هست، نگهداری کنید(مثلا چندین هزار مقاله) و همچنین قابلیت جستجو را در این رکوردها لحاظ کنید به صورتی که بر اساس رکوردهایی که بیشترین تعداد تکرار کلمات مدنظر را دارند مرتب شوند.

شاید اولین راه حل، مطلب آقای [سلیمان آبادی](#) در [اینجا](#) باشد، که تعداد تکرار یک کلمه را در فیلدی در جدول بیان کردند و درست هم هست اما با 2 شرط:

- 1) رکوردهای ما حجم کمی داشته باشند چرا که دستور LIKE پاسخ سریعی را با حجم بالای اطلاعات به ما نمی‌دهد.
- 2) رکوردهای ما از خانواده‌ی char باشند. برای مثال اگر مقالات ما به صورت pdf باشند این کد جواب نمی‌دهد.

اما راه دوم استفاده از [Full Text Search](#) و دستور CONTAINSTABLE هست که 2 شرط لازم را برای راه حل اول احتیاج نداره. در اینجا فقط نحوه‌ی استفاده از CONTAINSTABLE رو مطرح می‌کنیم.

جدولی از موارد یافت شده را بر اساس معیارهایی که ما به اون معرفی می‌کنیم، ایجاد می‌کند. این جدول حاوی دو فیلد KEY(کلید فیلد مورد نظر) و RANK(مقداری بین 0 تا 1000) است که میزان همسانی رکورد با معیار ما را مشخص می‌کند و ما با استفاده از این فیلد می‌توانیم رکوردهایمان را مرتب کنیم.

به این کد توجه کنید:

```
SELECT t.Title, p.[RANK]
  FROM Articles AS t
    INNER JOIN CONTAINSTABLE(Articles, Data, 'management' ) AS p
      ON t.Id = p.[KEY]
 ORDER BY p.RANK
```

در اینجا کار جستجو انجام شده و بر حسب میزان نزدیکی محتویات رکورد با معیار ما مرتب شده است.
نکته: هیچ فرقی نمی‌کند که محتویات فیلد مورد نظر شما یک متن ساده، یک فایل word یا حتی pdf باشد. فقط باید تنظیمات Full Text Search درست انجام شود.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۱۰ ۲۲:۴۹

برای PDF که عنوان کردید، آیا جستجوی فارسی کار می‌کند؟ از چه IFilter ای برای اینکار استفاده کردید؟

نویسنده: دل محسن
تاریخ: ۱۳۹۲/۰۳/۱۱ ۱۴:۵۰

بله کار میکنه. من با Adobe IFilter (رایگان) و Foxit IFilter (تجاری هست) کار کردم.

نویسنده: علی
تاریخ: ۱۳۹۲/۰۶/۰۶ ۱۲:۲۲

سلام.

من با rank کار می‌کنم توى پروژه هام.

منتها خیلی وقت‌ها رکوردهایی رنک بالایی می‌گیرند در حالی که رکوردهای مشابه دیگر رنک پایینی می‌گیرند. البته کمتر از ۵ درصد اتفاق می‌افته

در مورد خود رنک مطلب خوبی پیدا نکردم که اصلاً خود مفهوم رنک رو برام توضیح بده

نویسنده: علی
تاریخ: ۱۳۹۲/۱۰/۲۴ ۱۹:۲۷

منم توى این رنک بدجور گیر کردم. به جایجایی کامات حساسه . مثلًا «خواجه علی» و «علی خواجه».

قبل از اینکه این موضوع را بررسی کنیم باید با دستور Delete و Truncate آشنا شویم.

بررسی دستور Delete :

همانگونه که می‌دانیم از این دستور برای حذف رکوردها استفاده می‌کنند. با اجرای دستور Delete به راحتی می‌توانید تعدادی از رکوردهای یک جدول را حذف کنید. ساده‌ترین شکل استفاده از دستور Delete به صورت زیر می‌باشد.

```
DELETE FROM table_name
WHERE some_column=some_value
```

برای مثال در صورتیکه بخواهیم مشتریانی را حذف کنیم که کشور (Country) آنها USA است باید از دستور زیر استفاده کنیم.

```
DELETE FROM Customers
WHERE Country='USA'
GO
```

با اجرای این دستور هر رکوردی که در شرط مربوطه صدق کند حذف خواهد شد. (خوب این رو که همه می‌دانند) اما نکته مهمی که دستور Delete دارد این است کار این دستور به شکل Transactional می‌باشد. یعنی یا کلیه رکوردهایی که آنها USA است حذف می‌شود و یا هیچکدام از آنها. پس اگر شما 200000 رکورد داشته باشید که در این شرط صدق کند اگر وسط کار Delete (البته اگر عملیات حذف طولانی باشد) منصرف شوید می‌توانید با Cancel کردن این دستور عملیات Rollback Transaction را به خودکار توسط SQL Server انجام دهید. در صورتیکه عملیات Cancel را انجام دهید SQL Server Log File برای بازگرداندن مقادیر حذف شده استفاده خواهد کرد.

اما نکته دیگری که دستور Delete دارد این است که این دستور Log کلیه رکوردهایی را که قرار است حذف کند در Log File می‌نویسد. این Log شامل اصل رکورد، تاریخ و زمان حذف، نام کاربر و... می‌باشد. شاید الان متوجه شوید که دستور Delete چرا در برخی از مواقع که قرار است حجم زیادی از اطلاعات را حذف نماید به گندی این کار را انجام می‌دهد. (چون باید Log رکوردهای حذف شده در Log File نوشته شود).



بررسی دستور Truncate Table :

Truncate در لغت به معنی بریدن و کوتاه کردن می‌باشد. با استفاده دستور Truncate Table می‌توانید محتوای کلیه رکوردهای موجود در یک جدول را در کسری از ثانیه حذف کنید. نکته مهمی که باید درباره دستور Truncate Table بدانید این است که تاثیر استفاده از این دستور بر روی کلیه رکوردها بوده و به

هیچ عنوان نمی‌توان برای این دستور شرط (Where Clause) اعمال نمود.

شكل کلی استفاده از این دستور به صورت زیر می‌باشد.

```
TRUNCATE TABLE table_name
GO
```

برای مثال اگر بخواهیم کلیه رکوردهای موجود در جدول Customers را حذف نمایید کافی است با استفاده از این دستور اینکار را انجام دهیم.

```
TRUNCATE TABLE Customers
GO
```

با اجرای این دستور در کسری از ثانیه کلیه رکوردهای جدول Customers حذف خواهد شد. (بهتر است از این دستور زمانی استفاده کنید که بخواهید ساختار جدول شما باقی بماند)

- اما در مورد دستور Delete و Truncate Table باید به نکات زیر توجه کنید.
 - 1- دستور Truncate Table فاقد قسمت شرط (Where Clause) می‌باشد در صورتیکه دستور Delete دارای قسمت شرط (Where Clause) است
 - 2- دستور Truncate Table در Log File آدرس Page و مقدار فضای آزاد شده (کمترین میزان Log) را می‌نویسد اما در صورتیکه دستور Delete در Log File ثبت می‌نماید.
 - 3- دستور Truncate Table باعث می‌شود که Page های متعلق به جدول deallocate شوند. این معنی را می‌دهد که رکوردهای موجود در جدول واقعاً حذف نشوند بلکه Extent های مربوط به آن Page ها علامت Empty خورده تا دفعات بعد مورد استفاده قرار گیرند اما دستور Delete به طور فیزیکی محتوای Page ها مربوط به جدول را خالی می‌کند.
 - نکته : پس از Truncate شدن رکوردها امکان بازگشت آنها وجود ندارد.
 - 4- در صورتیکه جدول شما دارای ایندکس باشد. دستور Truncate Table آزاد کردن فضای مربوط به ایندکس را در یک مرحله انجام می‌دهد (مطابق بند 3) همچنین Log مربوط به این حذف به شکل حداقل (مطابق بند 2) در Log File ثبت می‌شود. اما دستور Delete هر رکوردي را که از ایندکس حذف می‌کند در Log File ثبت می‌کند.
 - 5- دستور Delete به دستور Truncate Table مربوط به هیچ عنوان هنگام اجرای دستور Truncate Table فعال نمی‌شود. در صورتیکه با اجرای Trigger 6- در صورتیکه جدول شما دارای Relation Reference (Relational Reference) باشد امکان استفاده از دستور Truncate Table وجود ندارد. لازم به ذکر است حتی اگر Reference را غیر فعال کنید باز هم امکان استفاده از دستور Truncate Table وجود نخواهد داشت و تلاش برای اجرای دستور Truncate Table باعث نمایش خطای زیر خواهد شد.

```
TRUNCATE TABLE CUSTOMERS
GO
```

100 %

Messages

Msg 4712, Level 16, State 1, Line 1
Cannot truncate table 'Customers' because it is being referenced by a FOREIGN KEY constraint.

در صورتیکه در دستور Delete امکان حذف رکوردها به ازای جداولی که دارای Relation هستند وجود دارد. فقط باید به این نکته توجه کنید که ترتیب حذف رکوردها از جداول Master و Detail را رعایت کنید.

7- دستور Truncate Table مقدار Identity را Reset کرده و آن را به Seed (هسته/مقدار اولیه) بر می‌گرداند. در صورتیکه دستور Delete تاثیری بر روی مقدار Identity ندارد

8- دستور Truncate Table تنها توسط کاربرانی قابل اجرا است که نقش SysAdmin و یا DB_Owner را داشته باشند در صورتیکه دستور Delete توسط هر کاربری که مجوز Delete بر روی جدول را داشته باشد قابل اجرا می‌باشد.

9- پس از اجرای دستور Truncate Table تعداد رکوردهای حذف شده نمایش داده نمی‌شود. در صورتیکه هنگام اجرای دستور Delete تعداد رکوردهای حذف شده نمایش داده می‌شود.

The image shows two side-by-side SSMS windows. The left window displays the command `TRUNCATE TABLE Customers2 GO` and the message `Command(s) completed successfully.`. The right window displays the command `DELETE FROM Customers2 GO` and the message `(91 row(s) affected)`.

نظرات خوانندگان

نوبنده: محمد مهدی ملائیان
تاریخ: ۱۳۹۲/۰۴/۰۲ ۱۵:۳۰

مطلوب مفیدی بود. منتظر مطالب شما هستم.

یکی از Identity هایی که در SQL Server به ازای Column های عددی می‌توان در نظر گرفت. به طور خیلی ساده هنگامی که این Attribute به ازای یک فیلد عددی تنظیم گردد. چنانچه رکوردی در جدول مربوط به Identity درج شود فیلد Identity مقداری را به طور اتوماتیک دریافت خواهد نمود.

نحوه دریافت مقدار به ازای فیلد Identity با توجه به آخرین مقدار آن و گام افزایش است که در هنگام ایجاد identity تعریف می‌گردد.

برای ایجاد یک فیلد از نوع Identity می‌توانید زمانیکه جدول خود را ایجاد می‌کنید این Attribute را به فیلد مورد نظر خود تخصیص دهید.

مثال ۱: این مثال نحوه ایجاد یک فیلد از نوع Identity را نمایش می‌دهد.

```
USE tempdb
GO
CREATE TABLE Customers1
(
    ID INT IDENTITY,-- ID INT IDENTITY(1,1)
    Name NVARCHAR(100),
    [Address] NVARCHAR(200)
)
GO
```

همانطور که در مثال ۱ مشاهده می‌کنید فیلد ID از نوع Identity تعریف شده است. در این حالت (ID int IDENTITY) مقدار شروع و گام افزایش به ازای این فیلد ۱ در نظر گرفته خواهد شد. در این صورت اگر چند رکورد زیر را به ازای این جدول درج کنید. مقدار Identity به صورت زیر خواهد بود.

```
INSERT INTO Customers1 (Name,[Address]) VALUES
(N'مسعود','میانه'),
(N'فرید','میانه'),
(N'احمد','میانه')
GO
SELECT * FROM Customers1
```

	ID	Name	Address
1	1	مسعود	میانه
2	2	فرید	میانه
3	3	احمد	میانه

مثال ۲: این مثال نحوه ایجاد یک فیلد از نوع Identity به همراه مقدار شروع و گام افزایش را مشخص می‌کند.

```
USE tempdb
GO
CREATE TABLE Customers2
(
    ID INT IDENTITY(100,2),
    Name NVARCHAR(100),
```

Identity و مباحث مربوط به آن (قسمت اول) - آشنایی با Identity

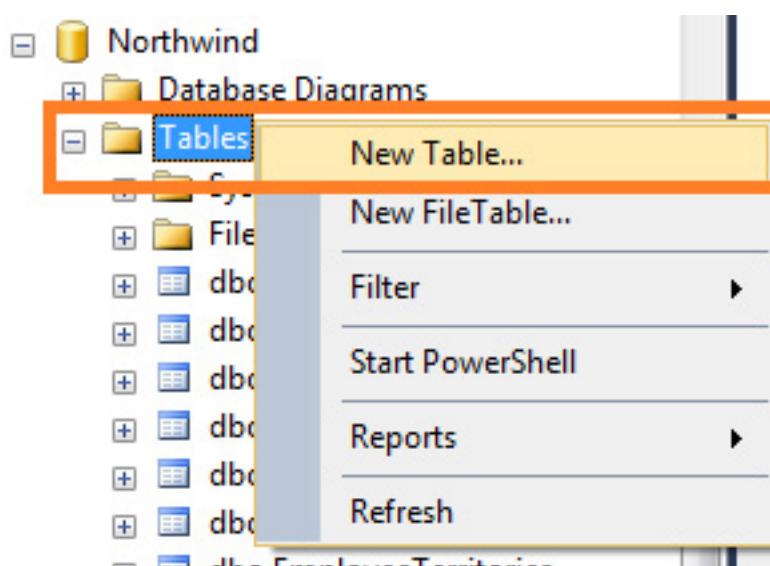
```
[Address] NVARCHAR(200)
)
GO
```

همانطور که در مثال 2 مشاهده می‌کنید فیلد ID از نوع Identity تعریف شده است و مقدار شروع آن از 100 و همچنین گام افزایش 2 در نظر گرفته شده است. در این صورت اگر چند رکورد زیر را به ازای این جدول درج کنید، مقدار Identity به صورت زیر خواهد بود.

```
INSERT INTO Customers2 (Name,[Address]) VALUES
(N'میانه',N'مسعود'),
(N'میانه',N'فرید'),
(N'میانه',N'احمد')
GO
SELECT * FROM Customers2
```

ID	Name	Address
1	100	مسعود
2	102	فرید
3	104	احمد

مثال 3: این مثال نحوه تنظیم یک فیلد به صورت Identity را در محیط SQL Server Management Studio (SSMS) آموزش می‌دهد.
1- برای شروع کار همانند تصویر زیر بر روی قسمت Table کلیک راست کنید و گزینه New Table... را انتخاب کنید.



2- پس از نمایش پنجره زیر فیلدی را که می‌خواهید از نوع Identity باشد را انتخاب کرده و در قسمت Column Properties

خاصیته Is Identity Yes را برابر قرار دهید تا فیلد مورد نظر شما از نوع Identity در نظر گرفته شود. لازم به ذکر است که مقدار شروع Identity Increment گام افزایش را مشخص می‌نماید.

TAHERI-PC.Northwind - dbo.Table_1* X Identity.sql - (local)\Administrator (51)

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Name	nvarchar(100)	<input checked="" type="checkbox"/>
Address	nvarchar(100)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Column Properties

Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۳/۲۱ ۱۷:۱

آیا فیلد Identity به خودی خود زمانیکه primary key و unique نباشه (مانند مثال‌های بالا) ارزش و کاربردی دارد؟

نویسنده: فرید طاهری
تاریخ: ۱۳۹۲/۰۳/۲۲ ۹:۲۳

این موضوع بستگی به سناریو شما دارد
اما معمولا در بیشتر مواقع Identity را به شکل Unique در نظر می‌گیرند ذکر این نکته هم ضروری است که
در SQL Server معمولا Primary Key بوسیله یک Unique Clustered Index هندل می‌شود
1- (هر چند می‌شود اون رو به صورت یک Unique Non Clustered Index در نظر گرفت)
2- Clustered Index ترتیب و چینش فیزیکی رکوردها را مشخص می‌کند یعنی اگر Identity به عنوان کلاستر ایندکس باشد
چینش و ترتیب فیزیکی رکوردها بر اساس Identity خواهد بود (سطح leaf Level مربوط به ایندکس که در کلاستر ایندکس
همان Data Level است)

همانگونه که می‌دانید مقدار Identity پس از درج به آن تخصیص می‌یابد چنانچه بخواهید به این مقدار دسترسی پیدا کنید چندین روش به ازای اینکار وجود دارد که ما در این مقاله سه روش معمول را بررسی خواهیم نمود.

1- استفاده از متغیر سیستمی @@Identity

2- استفاده از تابع () Scope_Identity

3- استفاده از تابع Ident_Current

هر سه این توابع مقدار Identity ایجاد شده برای جداول را نمایش می‌دهند. اما تفاوت هایی باهم دارند که در ادامه مقاله این تفاوت‌ها بررسی شده است.

1- متغیر سیستمی @@Identity : این متغیر سیستمی حاوی آخرین Identity ایجاد شده به ازای Session جاری شما است. لازم به ذکر است اگر به واسطه Insert شما، Identity دیگری در یک حوزه دیگر (مانند یک Trigger) ایجاد شود مقدار موجود در این متغیر حاوی آخرین Identity ایجاد شده است. (یعنی Identity ایجاد شده توسط آن تریگر و نه خود جدول). لازم به ذکر است این موضوع به طور کامل در ادامه مقاله شرح داده شده است.

2- استفاده از تابع () Scope_Identity : با استفاده از این تابع می‌توانیم آخرین Identity ایجا دشده به ازای Session جاری را بدست آوریم. لازم به ذکر است مقادیر Identity ایجاد شده توسط سایر حوزه‌ها تاثیر در مقدار بازگشتی توسط این تابع ندارد. در ادامه مقاله این موضوع به طور کامل بررسی شده است.

3- استفاده از تابع ident_Current : این تابع آخرین مقدار Identity موجود در یک جدول را نمایش می‌دهد. ذکر این نکته ضروری است که Identity ایجاد شده توسط سایر Session‌ها هم روی خروجی این تابع تاثیرگذار است. چون این تابع آخرین Identity موجود در جدول را به شما نمایش می‌دهد و نه Identity ایجاد شده به ازای یک Session را.

برای بدست آوردن یک Identity کافی است که پس از درج رکورد در جدول مورد نظر متغیر سیستمی @@Identity و یا توابع Scope_Identity و یا Ident_Current را همانند مثال زیر Select کنید.

```
USE TEMPDB
GO
IF OBJECT_ID(N'Employees', N'U') IS NOT NULL
    DROP TABLE Employees1;
GO
CREATE TABLE Employees
(
ID int IDENTITY,
FirstName NVARCHAR(50),
LastName NVARCHAR(50)
)
GO
INSERT INTO Employees (FirstName,LastName) VALUES ('طاهری', 'مسعود')
GO
SELECT @@IDENTITY AS [@@IDENTITY]
SELECT SCOPE_IDENTITY() AS [SCOPE_IDENTITY()]
SELECT IDENT_CURRENT('Employees1') AS [IDENT_CURRENT('Employees1')]
GO
```

خروجی دستورات بالا پس از درج رکورد مورد نظر به صورت زیر است.

	Results	Messages
@@IDENTITY		
1	1	
SCOPE_IDENTITY()		
1	1	
IDENT_CURRENT('Employees')		
1	1	

اما ممکن است از خودتان این سوال را بپرسید که آیا این توابع در سطح شبکه آخرین مقدار Identity درج شده توسط سایر Session‌ها را نمایش می‌دهند و یا Session جاری را؟ (منظور identity موجود در Session را نموده است).

برای دریافت پاسخ این سوال مطابق مراحل اسکریپت‌های زیر را اجرا نمایید.

1- ایجاد جدول Employees1

```
USE TEMPDB GO
IF OBJECT_ID(N'Employees1', N'U') IS NOT NULL
    DROP TABLE Employees1;
GO
CREATE TABLE Employees1
(
ID int IDENTITY(1,1),
FirstName NVARCHAR(50),
LastName NVARCHAR(50)
)
GO
```

همانطور که مشاهده می‌کنید مقدار شروع برای Identity برابر 1 و گام افزایش هم برابر 1 در نظر گرفته شده است. ((Identity(1,1)) .

2- در Session جدید دستورات زیر را اجرا نمایید. (درج رکورد جدید در جدول Employees1 و واکشی مقدار USE tempdb

```
GO
INSERT INTO Employees1(FirstName, LastName) VALUES ('طاهری', 'فرید')
GO
SELECT @@IDENTITY AS [@@IDENTITY]
SELECT SCOPE_IDENTITY() AS [SCOPE_IDENTITY()]
SELECT IDENT_CURRENT('Employees1') AS [IDENT_CURRENT('Employees1')]
GO
```

مریوط به SPID Connection جاری

```

USE tempdb
GO
INSERT INTO Employees1(FirstName, LastName) VALUES (N'فرید', N'طاهری')
GO
SELECT @@IDENTITY AS [@@IDENTITY]
SELECT SCOPE_IDENTITY() AS [SCOPE_IDENTITY()]
SELECT IDENT_CURRENT('Employees1') AS [IDENT_CURRENT('Employees1')]
GO
  
```

	Results	Messages
@@IDENTITY	1	
SCOPE_IDENTITY()	1	
IDENT_CURRENT(Employees1)	1	

همانگونه که ملاحظه می‌کنید Identity , Scope_Identity@() و Ident_Current (عدد 1) ایجاد شده بوسیله دستور Insert را به شما نمایش می‌دهند.

1- و در انتها در یک Session دیگر دستورات زیر را اجرا نمایید. (واکنشی مقدار Identity)

```

USE tempdb
GO
SELECT @@IDENTITY AS [@@IDENTITY]
SELECT SCOPE_IDENTITY() AS [SCOPE_IDENTITY()]
SELECT IDENT_CURRENT('Employees1') AS [IDENT_CURRENT('Employees1')]
GO
  
```

مریبوط به
جاری Connection

```

USE tempdb
GO
SELECT @@IDENTITY AS [@@IDENTITY]
SELECT SCOPE_IDENTITY() AS [SCOPE_IDENTITY()]
SELECT IDENT_CURRENT('Employees1') AS [IDENT_CURRENT('Employees1')]
GO
  
```

	Results	Messages
@@IDENTITY	1 NULL	
SCOPE_IDENTITY()	1 NULL	
IDENT_CURRENT('Employees1')	1 1	

همانطور که مشاهده می‌کنید در این SQL خواسته‌ایم آخرین مقدار Identity را به ما نشان داده شود. باید به این نکته توجه کنید با توجه به اینکه در این Session عملیات درجی هنوز انجام نگرفته است که ما Identity ایجاد شده را مشاهده نماییم. بنابراین صرفاً تابع Identity_Current موجود در جدول را به ما نمایش می‌دهد.

پس می‌توان به این نکته رسید که

ایجاد به ازای Scope_Identity : Identity و Identity@@@ های دیگر دسترسی ندارد.

آخرین Identity موجود در جدول را به شما نمایش می‌دهد. بنابراین باید این نکته را در نظر داشته باشید که Identity ها ایجاد شده توسط سایر Session ها روی مقدار بازگشتی این تابع تاثیرگذار است.

اما یکی دیگر از مباحث مهم درباره Identity Scope تاثیر Identity است (یعنی چه!). برای اینکه با مفهوم این موضوع آشنا شوید اسکریپت‌های مربوط به مثال زیر را بدقت اجرا کنید.

1- جدول Employees1 ایجاد

```

USE TEMPDB
GO
IF OBJECT_ID(N'Employees1', N'U') IS NOT NULL
    DROP TABLE Employees1;
GO
CREATE TABLE Employees1
(
ID int IDENTITY(1,1),
FirstName NVARCHAR(50),
LastName NVARCHAR(50)
)
GO
  
```

همانطور که مشاهده می‌کنید مقدار شروع برای Identity برابر 1 و گام افزایش هم برابر 1 در نظر گرفته شده است.) . (Identity(1,1

2- ایجاد جدول Employees2

```
USE TEMPDB
GO
IF OBJECT_ID('N'Employees2', N'U') IS NOT NULL
    DROP TABLE Employees2;
GO
CREATE TABLE Employees2
(
    ID int IDENTITY(100,1),
    FirstName NVARCHAR(50),
    LastName NVARCHAR(50)
)
GO
```

همانطور که مشاهده می‌کنید مقدار شروع برای Identity برابر 100 و گام افزایش هم برابر 1 در نظر گرفته شده است.) . (Identity(100,1

3- ایجاد یک Trigger به ازای جدول Employees1

```
USE tempdb
GO
CREATE TRIGGER Employees1_Insert ON Employees1 FOR INSERT
AS
BEGIN
    INSERT Employees2(FirstName,LastName)
    SELECT FirstName,LastName FROM INSERTED
END;
GO
```

ایجاد شده به ازای جدول Employees1 به ازای عملیات Insert اجرا می‌شود. همچنین مقادیر درج شده در جدول Employees1 بوسیله جدول Inserted در دسترس است. لازم به ذکر است جدول Inserted یک جدول موقت بوده که توسط Trigger ایجاد شده و داخل خود آن معتبر است.

هدف ما از ایجاد این Trigger یک کپی از رکوردهایی که در جدول Employees1 درج می‌شوند است. این کپی قرار است با استفاده از دستور Insert...Select در جدول Employees2 ایجاد گردد.

4- درج یک رکورد در جدول Employees1 و واکشی مقدار Identity

```
USE tempdb
GO
INSERT INTO Employees1(FirstName,LastName) VALUES ('طاهری','N','مسعود')
GO
SELECT @@IDENTITY AS [@@IDENTITY]
SELECT SCOPE_IDENTITY() AS [SCOPE_IDENTITY()]
SELECT IDENT_CURRENT('Employees1') AS [IDENT_CURRENT('Employees1')]
SELECT IDENT_CURRENT('Employees2') AS [IDENT_CURRENT('Employees2')]
GO
```

	Results	Messages
@@IDENTITY		
1	100	
SCOPE_IDENTITY()		
1	1	
IDENT_CURRENT('Employees1')		
1	1	
IDENT_CURRENT('Employees2')		
1	100	

مقادیر استخراج شده به ازای Identity به شرح زیر است

1- **Identity@@1**: پس از درج رکورد در جدول Employees1 متغیر سیستمی Identity@@@ مقدار 100 را نمایش داده است . دلیل این موضوع بر میگردد به Trigger موجود در جدول Employees1 خواهد شد.

با توجه به اینکه جدول Employees1 دارای یک فیلد Identity بوده است هنگام درج رکورد در جدول مقدار @@@=1@1@ بوده است که این Trigger خود با جدولی دیگری درگیر است که دارای Identity است . مقدار متغیر identity=100@@@ خواهد شد.

2- **Scope_Identity()**: مقدار نمایش داده شده توسط تابع Scope_Identity() برابر با مقدار Identity تخصیص (عدد 1) داده شده به ازای رکورد شما میباشد که این موضوع در اغلب موارد مد نظر برنامهنویسان میباشد.

3- **Ident_Current('Employees1')**: مقدار نمایش شده توسط تابع Ident_Current مقدار Identity (عدد 1) موجود در جدول Employees1 است.

4- **Ident_Current('Employees2')**: مقدار نمایش شده توسط تابع Ident_Current مقدار Identity (عدد 100) موجود در جدول Employees2 است.

چند نکته مهم

1- مقدار بازگردانده شده توسط تابع Ident_Current مقدار Identity آخرين نظر شما بوده است و عملیات درج سایر کاربران در این مقدار تاثیر گذار است.

2- برای بدست آوردن مقدار Identity درست بهتر است از تابع Scope_Identity استفاده نماییم. معمولاً در بیشتر مواقع مقدار بازگردانده شده توسط این تابع مد نظر برنامه نویسان است.

3- هم برای بدست آوردن Identity از تابع Scope_Identity استفاده میکند .

نظرات خوانندگان

نوبسند: کاربر
تاریخ: ۱۳۹۲/۰۴/۳۱ ۴:۵۰

بین دوست من مطلبتون رو خوندم هم اینو و هم قبلی رو، ازش خوشم اوmd اما چیزی راجب درج صریح یا بروز رسانی مقادیر Identity ننوشته بودین. یا اینکه نمیشه در یک جدول دو identity property داشت. من بلدم با set identity_insert table_name on/off کاری کنم که خودم دستی مقداری را برای خصیصه identity لحاظ کنم. ولی متأسفانه نتوانستم مقدار یک ستون با خصیصه Identity را بروز رسانی (با همون update) کنم. لطفا بهم بگید که اصلاً این کار ممکنه یا من بلد نیستم. البته براساس query زیر بمن SQL Server گفته که نمیشه این ستون را update کرد که ظاهرا هم همین طور(ستون id همانطور که در پیام آمده از نوع identity هست)

```
update t
set id = new_id
from (select id, row_number() over(order by id) new_id from #temp)t
--Cannot update identity column 'id'.
```

اصلاً اجازه بدين يه جور ديگه سوال رو مطرح کنم من نياز دارم تمام مقادير identity رو بروز رسانی کنم تا کاملاً پشت سر هم و متوالى بشن اين کار را ميتونم با يكتابع row_number و يك derived table انجام بدم (اگر بذارن!) همانطور که قبلانشان دادم، يا با روش زير اين کار را بکنم که البته اجرا نميشه به اين دليل که در یک جدول نميشه دو identity property داشت. با فرض اجرا شدن دستور select into باز هم در دستور update با مشکل بر می خورديم (چون نميشه ستون id را بروز رسانی کرد)

```
select id, identity(int, 1,1) new_id
into #temptable
from #temp
order by id asc

/*
cannot add identity column, using the SELECT INTO statement, to table '#temptable',
which already has column 'id' that inherits the identity property.
*/
update t
set id = new_id
from #temp t
join #temptable d
on t.id = d.id;
```

البته يك راهي برای حل اين مساله هست اونم اينه که ابتدا بياييم تمام دادهها جدول را در جدول ديگه اي درج کنيم سپس تمام دادههای جدول را حذف کنيم سپس دادههای حذف شده را با id جديid و مرتب شده در جدول اول درج کنيم. به اين شكل

```
declare @t table(id int)

insert into @t
select id from #temp

delete from #temp

set identity_insert #temp on
insert #temp(id)
select row_number() over(order by id) from @t
set identity_insert #temp off
```

اما مشکلي که وجود داره اينه که اگر جدول ما parent باشه با مشکل واجه ميشيم تمام سطرهای child جداول child یتیم میشن.

من قصد ندارم صورت مساله نقد و بررسی بشه و اصولی بودن یا صحیح بودنش مورد ارزیابی قرار بگیره فقط برام این یک سوال شده.

مساله عمومی که راجب این ستون وجود داره استفاده کردن از Gap‌های حاصل شده در این ستون برای درجه‌های بعدی است. که آن نیز بسیار ساده و در دسترس است.
query آیا شما میدانید که چگونه این مشکل با sequence ای که در نسخه 2012 معرفی شده است حل می‌شود؟

نویسنده: وحید نصیری
تاریخ: ۱۴:۵۵ ۱۳۹۲/۰۴/۳۱

- خیر. چندین نوع استراتژی برای تعیین PK وجود دارند که یکی از آن‌ها فیلد‌های Identity است و این تنها روش و الزاماً بهترین روش نیست.
- مثلاً زمانی‌که با ORM‌ها کار می‌کنید استفاده از فیلد‌های Identity در حین ثبت تعداد بالای از رکوردها مشکل ساز می‌شوند. چون این فیلد‌ها تحت کنترل دیتابیس هستند و نه برنامه، ORM نیاز دارد پس از هر بار Insert یکبار آخرین Id را از بانک اطلاعاتی واکشی کند. همین مساله یعنی افت سرعت در تعداد بالای Insert‌ها (چون یکبار کوئری Select باید ارسال شود و یکبار هم یک اضافی دوم برای دریافت Id تولیدی توسط دیتابیس).
- روش دوم تعیین PK استفاده از نوع Guid است. در این حالت، هم مشکل حذف رکوردها و خالی شدن یک شماره را در این بین ندارید و هم چون عموماً تحت کنترل برنامه است، سرعت کار کردن با آن بالاتر است. فقط تنها مشکل آن زیبا نبودنش است در مقایسه با یک عدد ساده فیلد‌های Identity.

در مورد فیلد‌های Identity Id به صلاح نیست چون:
الف) همانطور که عنوان کردید روابط بین جداول را به هم خواهد ریخت.
ب) در یک وب سایت و یا هر برنامه‌ای، کلاً آدرس‌ها و ارجاعات قدیمی را از بین می‌برد. مثلاً فرض کنید شماره این مطلب 1381 است و شما آن را یادداشت کرده‌اید. در روزی بعد، برنامه نویس شماره Id‌ها را کلاً ریست کرده. در نتیجه یک هفته بعد شما به شماره 1381 ای خواهید رسید که تطابقی با مطلب مدنظر شما ندارد (حالا فرض کنید که این عدد شماره پرونده یک شخص بوده یا شماره کاربری او و نتایج و خسارات حاصل را درنظر نگیرید).
ج) این خوب است که در بین اطلاعات یک ردیف خالی وجود دارد. چون بر این اساس می‌توان بررسی کرد که آیا واقعاً رکوردي حذف شده یا خیر. گاهی از اوقات کاربران ادعا می‌کنند که اطلاعات ارسالی آن‌ها نیست در حالیکه نبود این رکوردها به دلیل حذف بوده و نه عدم ثبت آن‌ها. با بررسی این Id‌ها می‌شود با کاربران در این مورد بحث کرد و پاسخ مناسبی را ارائه داد.

و اگر شماره‌ای که به کاربر نمایش می‌دهید فقط یک شماره ردیف است (و از این لحاظ می‌خواهید که حتماً پشت سرهم باشد)،
بهتر است یک View جدید ایجاد کنید تا این Id خود افزاینده را تولید کند (بدون استفاده از pk جدول).

پ.ن.

هدف من از این توضیحات صرفاً عنوان این بود که به PK به شکل یک فیلد read only نگاه کنید. این دقیقاً برخوردي است که Entity framework با این مفهوم دارد و صحیح است و اصولی. اگر در یک کشور هر روزه عده‌ای به رحمت ایزدی می‌روند به این معنا نیست که سازمان ثبت احوال باید شماره شناسنامه‌ها را هر ماه ریست کند!

نویسنده: فرید طاهری
تاریخ: ۲۰:۰ ۱۳۹۲/۰۴/۳۱

با تشکر از آقای نصیری و پاسخ مناسبی که ارائه کرده اند
در مورد استفاده از GUID به جای identity باید به یک نکته هم اشاره کنم که در بیشتر مواقع اگر مقدار GUID که به ازای یک فیلد UNIQUEIDENTIFIER تنظیم می‌کنید به صورت SEQUENTIAL نباشد باعث Fragmentation شدن ایندکس خواهد شد.
برای مقایسه بهتر بین Fragmentation ایندکس مربوط به Identity و GUID به مثال زیر دقت کنید. هر دو مثال فیلد ID خود را به شکل Clustered Index دارند بعد از درج تعدادی رکورد مساوی در دو جدول Fragmentation مربوط به جدولی که دارای GUID است به شدت بالا است که این موضوع باعث کاهش کارایی خواهد شد

```
USE TEMPDB
GO
```

```
IF OBJECT_ID('TABLE_GUID')>0
DROP TABLE TABLE_GUID
GO
CREATE TABLE TABLE_GUID
(
ID UNIQUEIDENTIFIER PRIMARY KEY,
FirstName NVARCHAR(1000),
LastName NVARCHAR(1000)
)
GO
IF OBJECT_ID('TABLE_IDENTITY')>0
DROP TABLE TABLE_IDENTITY
GO
CREATE TABLE TABLE_IDENTITY
(
ID INT IDENTITY PRIMARY KEY,
FirstName NVARCHAR(1000),
LastName NVARCHAR(1000)
)
GO
INSERT INTO TABLE_GUID(ID,FirstName,LastName) VALUES
(NEWID(),REPLICATE('FARID*',100),REPLICATE('Taheri*',100))
GO 10000

INSERT INTO TABLE_IDENTITY(FirstName,LastName) VALUES
(REPLICATE('FARID*',100),REPLICATE('Taheri*',100))
GO 10000

--Fragmentation بررسی وضعیت
SELECT * FROM sys.dm_db_index_physical_stats(DB_ID(),OBJECT_ID('TABLE_GUID'),NULL,NULL,'DETAILED')
DBCC SHOWCONTIG(TABLE_GUID)
GO
SELECT * FROM sys.dm_db_index_physical_stats(DB_ID(),OBJECT_ID('TABLE_IDENTITY'),NULL,NULL,'DETAILED')
DBCC SHOWCONTIG(TABLE_IDENTITY)
GO
```

خوب برای اینکه Fragmentation این نوع جداول را رفع کنید چند راه داریم

1- تولید GUID به صورت Sequential (لازم می‌دانم اشاره کنم این قابلیت در SQL Server وجود دارد ولی مقدار تولید شده باید به شکل یک Default Constraint باشد که این موضوع نیازمند این است که شما اگر در سورس به این GUID نیاز پیدا کنید مجبور به زدن Select ... شوید. اگر بخواهید در سورس این کار را انجام دهید باید از Extention هایی که برای اینکار وجود دارند استفاده کنید فکر کنم Hibernate رو پشتیبانی کنه در مورد EF دقیقاً اطلاع ندارم باید اهل فن نظر بدن)

2- تنظیم مقدار Fillfactor به ازای ایندکس
3- Reorganize دوره ای ایندکس

نویسنده: برنامه نویس
تاریخ: ۱۹:۱ ۱۳۹۳/۰۷/۰۴

باسلام: اگر همزمان 10 کاربر در شبکه بخوان درج کنن و هر بار identity درست بعد از درج نشان داده شود تابع Scope_Identity درست عمل میکند؟ تداخل به وجود نمی‌آید؟

نویسنده: محسن خان
تاریخ: ۱۳:۳۲ ۱۳۹۳/۰۷/۰۵

در مطلب [بررسی Locks و Transactions در SQL Server](#) پاسخ خود را خواهید یافت.

عنوان:	ویدئوی آموزش مقدمات CodeFirst در قالب یک کلاس آموزشی به همراه مثال
نویسنده:	سید مجتبی حسینی
تاریخ:	۱۳۹۲/۰۴/۰۳ ۱:۲۰
آدرس:	www.dotnettips.info
گروه‌ها:	Entity framework, SQL Server, EF Code First

این ویدئو

به مدت حدود ۱ یکساعت و حجمی حدود ۵۰ مگابایت، مربوط به یک کلاس آموزشی است که در ضمن آن به بررسی مقدماتی چگونگی بکاربردن روش **CodeFirst** برای تولید دیتابیس و جداول آن، پرداخته‌ام. پیش‌اپیش از نواقص و نارسایی‌های احتمالی آن، پژوهش می‌طلبم.

سرفصل مطالبی که بطور مختصر مطرح شده‌اند، عبارتند از:

معرفی EF CodeFirst و کاربرد آن

استفاده از EntityFramework برای افزودن Nuget Package Manager

ایجاد کلاس نمونه **User** و معرفی کلاس **User** به عنوان جدولی از دیتابیس

ایجاد **ConnectionString** و نکات مربوط به آن برای ایجاد صحیح جداول در SQL Server

چگونگی ایجاد فیلد کلیدی

روش ذخیره سازی اطلاعات در جدول

روش ایجاد رابطه یک به چند با ایجاد دو جدول کمکی **Log** و **Work** و مرتبط با جدول **User**

روش جستجو در جداول بدون استفاده مستقیم از SQL Query

نظرات خوانندگان

نویسنده: صادق
تاریخ: ۱۳۹۲/۰۴/۰۶ ۱۷:۵۶

با سلام و عرض ادب
می خواستم بدونم ادامه این ویدئو را چطوری می تونم تهیه کنم . با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۴/۰۶ ۱۸:۳۱

روی لینک «این ویدیو» در ابتدای مطلب کلیک کنید.

نویسنده: sorosh
تاریخ: ۱۳۹۲/۰۴/۰۶ ۱۸:۴۹

با سلام؛ می خواستم بدونم ویدئو دیگه‌ای آیا این مجموعه داره یا همینه فقط ؟ مرسی

نویسنده: سید مجتبی حسینی
تاریخ: ۱۳۹۲/۰۴/۰۷ ۱۳:۰

سلام و درود.
در بینایین جلسات آموزش سی شارپ پیشرفته به جهت تبیین یکی از کاربردهای نوین کدنویسی شیءگرا و همچنین تقویت انگیزه یادگیرندگان در یادگیری همه مطالب ، چه انتزاعی و چه کاربردی، به طرح این بحث فقط در یک جلسه اکتفا کردم.
باتشکر

نویسنده: آگاه
تاریخ: ۱۳۹۲/۱۱/۲۱ ۱۵:۳۶

نمیشه آموزش ویدیویی این بحث رو ادامه بدین؟

خیلی شیوا کارا مد و مفید بود.. مخصوصا اینکه کلی جزئیات هم بینش گفته میشد.. :)

مدتی است که حالت READ_COMMITTED_SNAPSHOT بسیار مورد توجه واقع شده:

- در سایت Stack overflow از آن استفاده می‌شود ([^](#)).
- در SQL Server Azure حالت پیش فرض ایجاد دیتابیس‌ها و تراکنش‌های جدید است ([^](#)).
- در Entity framework حالت پیش فرض تراکنش‌های ایجاد شده، قرار گرفته است ([^](#)).

و ... در Oracle، تنها حالت مدیریت مسایل همزمانی است! (البته به نام [MVCC](#)، اما با همین عملکرد)

اما SQL Server در READ_COMMITTED_SNAPSHOT چیست و کاربرد آن کجا است؟

اگر استفاده گسترده و سنگینی از SQL Server داشته باشید، حتماً به پیغام‌های خطای [deadlock](#) آن برخورده‌اید:

Transaction (Process ID 54) was deadlocked on lock resources with another process and has been chosen as the deadlock victim.
Rerun the transaction.

روش پیش فرض مدیریت مسایل همزمانی در READ COMMITTED SQL Server، حالت READ COMMITTED است. به این معنا که اگر در طی یک تراکنش مشغول به تغییر اطلاعاتی باشیم، سایر کاربران از خواندن نتیجه آن (اصطلاحاً Dirty read گفته می‌شود) منع خواهند شد؛ تا زمانیکه این تراکنش با موقیت به پایان برسد. هرچند در این حالت سایر تراکنش‌ها امکان ویرایش یا حذف اطلاعات را خواهند داشت، به علاوه اگر در طی این تراکنش، اطلاعاتی خوانده شوند، سایر تراکنش‌ها تا پایان تراکنش جاری، قادر به تغییر این اطلاعات خوانده شده خواهند بود (منشاء بروز خطاهای deadlock یاد شده در سیستم‌های پرترافیک).

در SQL Server 2005 برای بهبود مقیاس پذیری READ_COMMITTED_SNAPSHOT و کاهش خطاهای deadlock، مکانیزم معرفی گشت.

به صورت خلاصه زمانیکه که تراکنش مورد نظر تحت حالت READ COMMITTED SNAPSHOT انجام می‌شود، optimistic reads and pessimistic writes خواهیم داشت (خواندن‌های خوشبینانه و نوشتن‌های بدینانه). در این حالت تضمین می‌شود که خواندن اطلاعات داخل یک تراکنش، شامل اطلاعات تغییر یافته توسط سایر تراکنش‌های همزمان خواهد بود. همچنین زمانیکه در این بین، اطلاعاتی خوانده می‌شود، بر روی این اطلاعات برخلاف حالت READ COMMITTED قفل قرار داده نمی‌شود. بنابراین تراکنش‌هایی که در حال خواندن اطلاعات هستند، تراکنش‌های همزمانی را که در حال نوشتمن اطلاعات می‌باشند، قفل نخواهد کرد و بر عکس.

نحوه فعال سازی READ_COMMITTED_SNAPSHOT

فعال سازی READ_COMMITTED_SNAPSHOT باید ابتدا در سطح یک بانک اطلاعاتی SQL Server انجام شود:

```
ALTER DATABASE testDatabase SET ALLOW_SNAPSHOT_ISOLATION ON;
ALTER DATABASE testDatabase SET READ_COMMITTED_SNAPSHOT ON;
```

کاری که در اینجا انجام خواهد شد، ایجاد یک snapshot یا یک کپی فقط خواندنی، از بانک اطلاعاتی کاری شما می‌باشد. بنابراین در این حالت، زمانیکه یک عبارت Select را فراخوانی می‌کنید، این خواندن، از بانک اطلاعاتی فقط خواندنی تشکیل شده، صورت خواهد گرفت. اما تغییرات بر روی دیتابیس اصلی کاری درج شده و سپس این snapshot به روز می‌شود.

حالت [READ_COMMITTED_SNAPSHOT](#) خصوصاً برای برنامه‌های وبی که تعداد بالای Read در مقابل تعداد کمی Write دارد، به شدت بر روی کارآیی و بالا رفتن سرعت و مقیاس پذیری آن‌ها تاثیر خواهد داشت؛ به همراه حداقل تعداد deadlock‌های حاصل شده.

در Entity framework وضعیت به چه صورتی است؟

از حالت پیش فرض مدیریت مسایل همزمانی در SQL Server یا همان حالت READ COMMITTED در زمان فراخوانی متده استفاده می کند. SaveChanges در 6 EF این حالت پیش فرض به [READ_COMMITTED_SNAPSHOT](#) تغییر کرده است. البته همانطور که عنوان شد، پیشتر باید با نک اطلاعاتی را نیز جهت پذیرش این نوع تراکنشها آماده ساخت. اگر از نگارش‌های پایین‌تر از 6 EF استفاده می کنید، برای استفاده از حالت READ_COMMITTED_SNAPSHOT باید صراحتاً از [IsolationLevel](#) را مشخص ساخت:

```
using (var transactionScope =
    new TransactionScope(TransactionScopeOption.Required,
    new TransactionOptions { IsolationLevel = IsolationLevel.Snapshot })) {
    // update some tables using entity framework
    context.SaveChanges();
    transactionScope.Complete();
}
```

نظرات خوانندگان

نویسنده: ali.rezayee | تاریخ: ۱۴/۰۴/۹۲۱۳:۴۱

با سلام و تشکر بخاطر این مطلب عالی.
امکان دارد در خصوص بخش «SQL Server در READ_COMMITTED_SNAPSHOT چیست و کاربرد آن کجا است؟» یک مثال عملی بزنید، مطلب کمی گنگ است.
منون

نویسنده: وحید نصیری | تاریخ: ۱۴/۰۴/۹۲۱۳:۱۸

فرض کنید یک جدول نظرات دارید با این تعریف

```
CREATE TABLE [BlogComments](
[Id] [int] IDENTITY(1,1) NOT NULL,
[Body] [nvarchar](max) NULL,
[Date1] [datetime] NOT NULL,
CONSTRAINT [PK_BlogComments] PRIMARY KEY CLUSTERED
(
[Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

بعد در management studio دو پنجره اجرای کوئری جدید را ایجاد کنید. در پنجره اول بنویسید:

```
-- در پنجره اول
BEGIN TRAN
UPDATE [BlogComments] SET Body='Test' WHERE id=1
```

در پنجره دوم بنویسید

```
-- در پنجره دوم
SELECT TOP 1000 [Id] ,[Body] ,[Date1] FROM [BlogComments]
```

- ابتدا عبارت پنجره اول را اجرا کنید. این پنجره حاوی یک تراکنش نا تمام است. شروع دارد اما به عمد پایان آن را ذکر نکردیم.
- الان پنجره دوم را اجرا کنید.
مشاهده خواهید کرد که ... به جواب نمی‌رسید. کوئری اجرا نمی‌شود و سیستم قفل شده چون تراکنش اول commit نشده (مثلاً یک تراکنش طولانی را اینجا شبیه سازی کردیم؛ یا حتی یک اشتباه در تعاریف T-SQL انجام شده).

در ادامه، عملیات این پنجره‌ها را دستی متوقف کنید. بعد مطابق دستوراتی که پیشتر ذکر شد، READ_COMMITTED_SNAPSHOT را روی دیتابیس فعال کنید.
مجدداً دو مرحله قبل را اجرا کنید. در این حالت کوئری دوم اجرا خواهد شد، چون اطلاعات را از کپی فقط خواندنی بانک اطلاعاتی شما دریافت می‌کند؛ بر اساس آخرین اطلاعات commit شده در سیستم.

نویسنده: مرادی | تاریخ: ۱۵/۰۴/۹۲۱۳:۵۳:۱۰

با سلام، در قسمتی از مطلبتان، آورده اید "یک snapshot یا یک کپی فقط خواندنی،"

که این پیش فرض اشتباه را در ذهن خواننده ایجاد می‌کند که از دیتابیس یک کپی گرفته می‌شود، در حالی که از دیتابیس کپی

گرفته نمی‌شود، بلکه Snapshot حاوی تغییرات دیتابیس از لحظه ایجاد Snapshot تا به حال است، مثلاً می‌گوید در جدول People، سه رکورد درج شده است، از تفاضل دیتابیس و این Snapshot می‌توان به وضعیت دیتابیس قبل از ایجاد Snapshot پی‌برد، به همین دلیل است که ایجاد یک Snapshot ولو روی یک دیتابیس چند گیگابایتی نیز در کسری از ثانیه انجام می‌پذیرد. علاوه بر این امکان استفاده از این امکان در SQL 2000 نیست، ولو با ADO.NET، که البته چیزی را از ارزش‌های این روش کم نمی‌کند، با سپاس از مطلب خوبtan

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۵ ۱۱:۴۸

بحث اصلی هم همین نحوه و محل ذخیره سازی snapshot است.
- Sanpshot مطابق واژه نامه مایکروسافت معنای «نگارش» را می‌دهد. در این حالت کلیه کوئری‌های داخل یک تراکنش، یک نگارش یا snapshot از دیتابیس را مشاهده خواهد کرد. این نگارش یا tempdb Row version یا tempdb نگه داری می‌شود. با فعال سازی **SNAPSHOT isolation**، هر زمانیکه یک ردیف به روز رسانی می‌شود، موتور SQL Server یک نسخه از اطلاعات اولیه این ردیف را در tempdb ذخیره می‌کند (اینجا بود که عنوان شد با یک کپی فقط خواندنی از اطلاعات در حین واکشی اطلاعات سر و کار خواهد داشت).

خلاصه الگوریتم کاری آن:

الف) با آغاز یک تراکنش، یک عدد متوالی منحصر‌فرد تراکنش (شماره نگارش) ایجاد شده و به آن نسبت داده می‌شود.
ب) در حین این تراکنش، موتور SQL Server، به tempdb مراجعه کرده و شماره نگارشی نزدیک و کمتر از شماره نگارش تراکنش جاری را پیدا می‌کند. همچنین SQL Server بررسی می‌کند که این شماره یافته شده حتماً جزو تراکنش‌های پایان یافته سیستم باشد.

ج) بر اساس این شماره یافته شده، نگارش معتبری از اطلاعات از tempdb استخراج می‌شود.
به این ترتیب یک تراکنش، کلیه اطلاعات موجود در ابتدای کار خود را بدون قرار دادن قفلی بر روی جداول مرتبط، دریافت خواهد کرد.

اطلاعات بیشتر

- در متن ذکر گردید که از SQL Server 2005 به بعد قابلیت فوق اضافه شده.
- همچنین SQL Server 2000 دیگر پشتیبانی رسمی ندارد و استفاده از آن حداقل از لحاظ امنیتی معقول نیست.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۲۳ ۱۲:۵

خبری در اینباره از تیم SQL Server بعدی

The Hekaton team also found that multi-version concurrency control (MVCC) proved robust in scenarios with higher workloads and higher contention

اطلاعات بیشتر

نویسنده: amir ranjbarian
تاریخ: ۱۳۹۲/۰۵/۰۸ ۱۸:۳۷

با سلام؛ در صورتی که بخواهیم این مورد را در دیتابیسی که از ALTER DATABASE استفاده می‌کنیم با این خطا failed because the READ_COMMITTED_SNAPSHOT and the ALLOW_SNAPSHOT_ISOLATION options cannot be set to ON when a database has FILESTREAM filegroups. To set READ_COMMITTED_SNAPSHOT or ALLOW_SNAPSHOT_ISOLATION to ON, you must remove the FILESTREAM filegroups from the database.

مواجه می‌شویم من در دیتابیس از filestremme استفاده کردم برای ذخیره فایلهای مورد نیاز در نرم افزار خودم، می‌خواستم بدونم آیا استفاده از این روش (filestremme) اصولاً خوب هست یا نه؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۵/۰۸ ۱۸:۴۵

این محدودیت از نگارش **R2** اس کیوال سرور 2008 به بعد برطرف شده.

مقدمه

تکنولوژی CTE از نسخه SQL Server 2005 رسمیت یافته است و شامل یک result set موقتی^[1] است که دارای نام مشخص بوده و می-توان از آن در دستورات SELECT, INSERT, UPDATE, DELETE استفاده کرد. همچنین از CTE می‌توان در دستور CREATE VIEW و دستور SELECT مربوط به آن استفاده کرد. در نسخه SQL Server 2008 نیز امکان استفاده از CTE در دستور MERGE فراهم شده است.

در SQL Server از دو نوع CTE بازگشتی^[2] و غیر بازگشتی^[3] پشتیبانی می-شود. در این مقاله سعی شده است نحوه تعریف و استفاده از هر دو نوع آن آموزش داده شود.

أنواع روش-های ایجاد جداول موقت برای استفاده از جداول موقت در سرور اسکیوال، سه راه زیر وجود دارد.

روش اول: استفاده از دستوری مانند زیر است که سبب ایجاد جدول موقتی در بانک سیستمی tempdb می-شود. زمانی-که شما ارتباط خود را با سرور SQL قطع می-کنید به صورت اتوماتیک جداول موقت شما از بانک tempdb حذف می-شوند. این روش در برنامه نویسی پیشنهاد نمی-شود و فقط در کارهای موقتی و آزمایشی مناسب است.

```
SELECT * INTO #temptable FROM [Northwind].[dbo].[Products]
UPDATE #temptable SET [UnitPrice] = [UnitPrice] + 10
```

روش دوم: استفاده از متغیر نوع Table است، که نمونه آن در مثال زیر دیده می-شود. زمانیکه از محدوده^[4] جاری کد^[5] خودتان خارج شوید آن متغیر نیز از حافظه پاک میشود. از این روش، عموماً در کدهای UserDefined Procedureها و Functionها استفاده می-شود.

```
DECLARE @tempTable TABLE
(
    [ProductID] [int] NOT NULL,
    [ProductName] [nvarchar](40) NOT NULL,
    [UnitPrice] [money] NULL
)

INSERT INTO @tempTable
SELECT
    [ProductID],
    [ProductName],
    [UnitPrice]
FROM [Northwind].[dbo].[Products]

UPDATE @temptable SET [UnitPrice] = [UnitPrice] + 10
```

روش سوم: استفاده از CTE است که مزیت-هایی نسبت به دو روش قبلی دارد و در بخش بعدی به نحوه تعریف و استفاده از آن خواهیم پرداخت.

کار با CTE

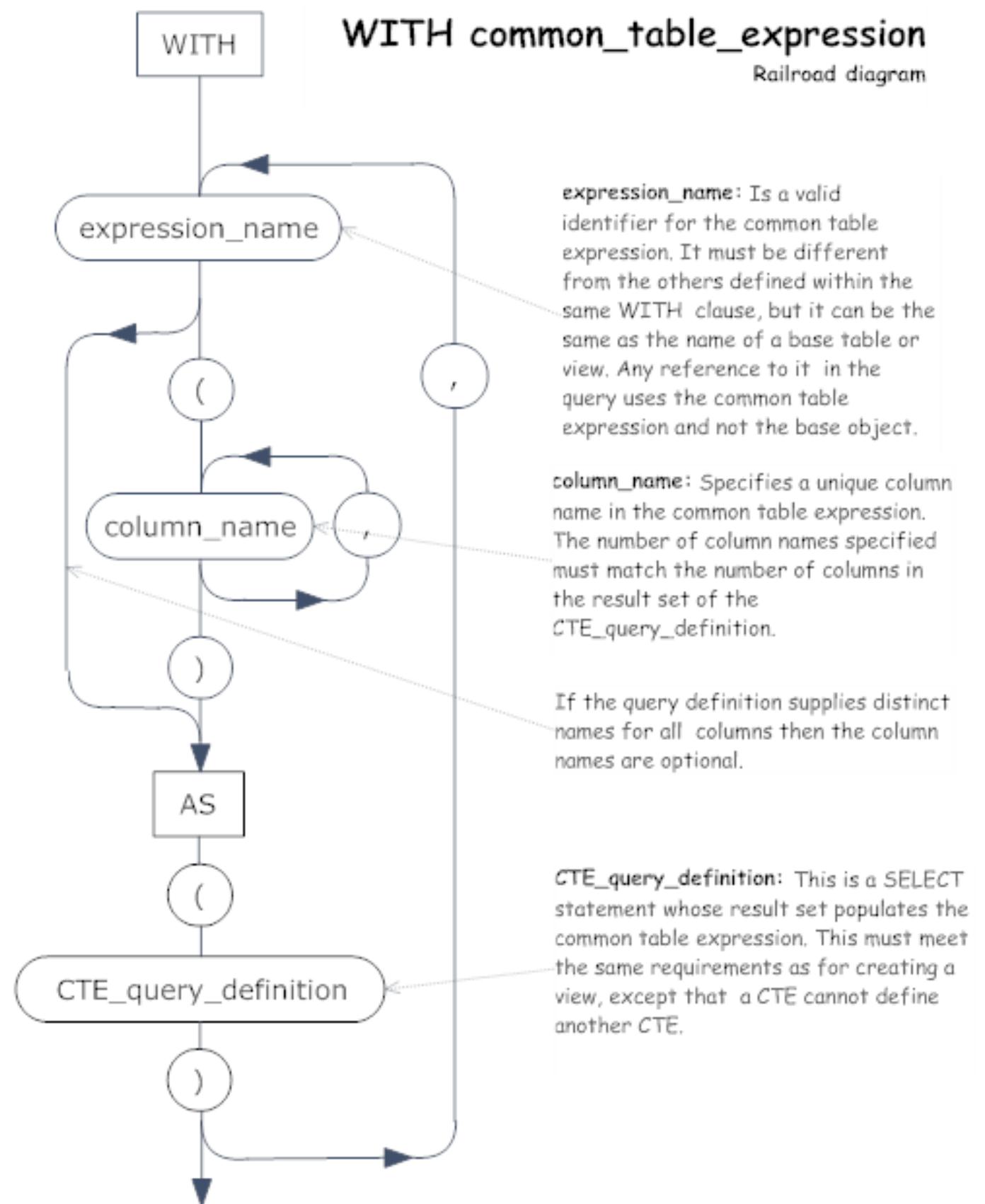
ساده-ترین شکل تعریف یک CTE به صورت زیر است:

```
WITH yourName [(Column1, Column2, ...)]
AS
()
```

```
) your query
```

با کلمه WITH شروع شده و یک نام اختیاری به آن داده می-شود. سپس فهرست فیلدهای جدول موقت را درون زوج پرانتز، مشخص می-کنید. تعریف این فیلدها اختیاری است و اگر حذف شود، فیلدهای جدول موقت، مانند فیلدهای کوئری مربوطه خواهد بود.

result set می-شود. قواعد تعریف این کوئری مشابه قواعد تعریف کوئری است که در دستور CREATE VIEW کاربرد دارد.



همانطور که از این تصویر مشخص است می-توان چندین بلوک از این ساختار را به دنبال هم تعریف نمود که با کاما از هم جدا می-شوند. در واقع یکی از کاربردهای CTE ایجاد قطعات کوچکی است که امکان استفاده مجدد را به شما داده و می-تواند سبب خواناتر شدن کدهای پیچیده شود.

یکی دیگر از کاربردهای آنچهای CTE می‌باشد که شما نمی‌خواهید یک شی Viewی عمومی تعریف کنید و در عین حال می‌خواهید از مزایای Viewها بهره‌مند شوید. و همچنین از کاربردهای دیگر CTE تعریف جدول موقت و استفاده از آن جدول به صورت همزمان در یک دستور است.

بعد از آنکه CTE یا CTE‌های خودتان را تعریف کردید آنگاه می-توانید مانند جداول معمولی از آنها استفاده کنید. استفاده از این جداول توسط دستوری خواهد بود که دقیقاً بعد از تعریف CTE نوشته می‌شود.

ایجاد یک CTE غیر بازگشتی [6]

مثال اول، یک CTE غیر بازگشتی ساده را نشان میدهد.

```
WITH temp
AS
(
    SELECT
        [ProductName],
        [UnitPrice]
    FROM [Northwind].[dbo].[Products]
)
SELECT * FROM temp
ORDER BY [UnitPrice] DESC
```

مثال دوم نمونه‌ای دیگر از یک CTE غیر بازگشتی است.

```
WITH orderSales (OrderID, Total)
AS
(
    SELECT
        [OrderID],
        SUM([UnitPrice]*[Quantity]) AS Total
    FROM [Northwind].[dbo].[Order Details]
    GROUP BY [OrderID]
)
SELECT
    O.[ShipCountry],
    SUM(OS.[Total]) AS TotalSales
FROM [Northwind].[dbo].[Orders] AS O INNER JOIN [orderSales] AS OS
ON O.[OrderID] = OS.[OrderID]
GROUP BY O.[ShipCountry]
ORDER BY TotalSales DESC
```

هدف این کوئری، محاسبه کل میزان فروش کالاهای، به ازای هر کشور می‌باشد. ابتدا از جدول Order Details مجموع فروش هر سفارش محاسبه شده و نتیجه آن در یک CTE به نام orderSales قرار می‌گیرد و از JOIN این جدول موقت با جدول Orders محاسبه نهایی انجام شده و نتیجه‌ای مانند این تصویر حاصل می‌شود.

	ShipCountry	TotalSales
1	USA	263566.98
2	Germany	244640.63
3	Austria	139496.63
4	Brazil	114968.48
5	France	85498.76
6	Venezuela	60814.89
7	UK	60616.51
8	Sweden	59523.70
9	Ireland	57317.39
10	Canada	55334.10
11	Belgium	35134.98
12	Denmark	34782.25

نتیجه خروجی

مثال سوم استفاده از دو CTE را به صورت همزمان نشان می-دهد:

```
WITH customerList
AS
(
    SELECT
        [CustomerID],
        [ContactName]
    FROM [Northwind].[dbo].[Customers]
    WHERE [Country] = 'UK'
)
,orderList
AS
(
    SELECT
        [CustomerID],
        [OrderDate]
    FROM [Northwind].[dbo].[Orders]
    WHERE YEAR([OrderDate]) < 2000
)
SELECT
    cl.[ContactName],
    YEAR(ol.[OrderDate]) AS SalesYear
FROM customerList AS cl JOIN orderList AS ol
ON cl.[CustomerID] = ol.[CustomerID]
```

مثال چهارم استفاده مجدد از یک CTE را نشان می-دهد. فرض کنید جدولی به نام digits داریم که فقط یک فیلد digit دارد و دارای 10 رکورد با مقادیر 0 تا 9 است. مانند تصویر زیر

	digit
	0
	1
	2
	3
	4
	5
	6
▶	7
	8
	9
*	NULL

نتیجه خروجی

حال می-خواهیم از طریق CROSS JOIN اعداد 1 تا 100 را با استفاده از مقادیر این جدول تولید کنیم. کد زیر آنرا نشان می-دهد:

```
WITH digitList
AS
(
    SELECT [digit] from [digits]
)
SELECT
    a.[digit] * 10 + b.[digit] + 1 AS [Digit]
FROM [digitList] AS a CROSSJOIN [digitList] AS b
```

در این کد یک CTE تعریف شده و دو بار مورد استفاده قرار گرفته است. مثلا اگر بخواهید اعداد 1 تا 1000 را تولید کنید می-توانید سه بار از آن استفاده کنید. حاصل این دستور result set مانند زیر است.

Results

Digit

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13

نتیجه

Results

87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

نتیجه

حتی می-توان از یک CTE در کوئری CTE بعدی مانند کد زیر استفاده کرد.

```
WITH CTE_1 AS
(
    ....
),
CTE_2 AS
(
    SELECT ... FROM CTE_1 JOIN ...
)
SELECT *
FROM FOO
LEFTJOIN CTE_1
LEFTJOIN CTE_2
```

ایجاد یک CTE بازگشتی[7]

از CTE بازگشتی برای پیمایش جداولی استفاده می-شود که رکوردهای آن دارای رابطه سلسله مراتبی یا درختی است. نمونه این جداول، جدول کارمندان است که مدیر هر کارمند نیز مشخص شده است یا جدولی که ساختار سازمانی را نشان می-دهد یا جدولی که موضوعات درختی را در خود ذخیره کرده است. یکی از مزایای استفاده از CTE بازگشتی، سرعت کار آن در مقایسه با روش-های پردازشی دیگر است.

ساختار کلی یک دستور CTE بازگشتی به صورت زیر است.

```
WITH cteName AS
(
    query1
    UNION ALL
    query2
)
```

در بدنه CTE حداقل دو عضو[8] (کوئری) وجود دارد که بایستی با یکی از عبارت-های زیر به هم متصل شوند.

UNION

UNION ALL

INTERSECT

EXCEPT

شامل دستوری است که اولین سری از رکوردهای result set نهایی را تولید می-کند. اصطلاحاً به این کوئری anchor query1 member می-گویند.

بعد از دستور query1، حتماً بایستی از UNION ALL و امثال آنها استفاده شود.

سپس query2 ذکر می-شود. اصطلاحاً به این کوئری recursive member گفته می-شود. این کوئری شامل دستوری است که سطوح بعدی درخت را تولید خواهد کرد. این کوئری دارای شرایط زیر است.

حتماً بایستی به CTE که همان cteName است اشاره کرده و در جایی از آن استفاده شده باشد. به عبارت دیگر از رکوردهای موجود در جدول موقت استفاده کند تا بتواند رکوردهای بعدی را تشخیص دهد.

حتماً بایستی مطمئن شوید که شرایط کافی برای پایان حلقه پیمایش رکوردها را داشته باشد در غیر این صورت سبب تولید حلقه بی پایان[9] خواهد شد.

بدنه CTE می-تواند حاوی چندین recursive member و چندین anchor member باشد ولی فقط recursive member هستند که به اشاره می-کنند.

برای آنکه نکات فوق روشن شود به مثال-های زیر توجه کنید.

فرض کنید جدولی از کارمندان و مدیران آنها داریم که به صورت زیر تعریف و مقداردهی اولیه شده است.

```

IFOBJECT_ID('Employees','U')ISNOTNULL
DROPTABLE dbo.Employees
GO

CREATE TABLE dbo.Employees
(
    EmployeeID int NOT NULL PRIMARY KEY,
    FirstName varchar(50) NOT NULL,
    LastName varchar(50) NOT NULL,
    ManagerID int NULL
)
GO

INSERT INTO Employees VALUES (101, 'Alireza', 'Nematollahi', NULL)
INSERT INTO Employees VALUES (102, 'Ahmad', 'Mofarrahzadeh', 101)
INSERT INTO Employees VALUES (103, 'Mohammad', 'BozorgGhommi', 102)
INSERT INTO Employees VALUES (104, 'Masoud', 'Narimanji', 103)
INSERT INTO Employees VALUES (105, 'Mohsen', 'Hashemi', 103)
INSERT INTO Employees VALUES (106, 'Aref', 'Partovi', 102)
INSERT INTO Employees VALUES (107, 'Hosain', 'Mahmoudi', 106)
INSERT INTO Employees VALUES (108, 'Naser', 'Pourali', 106)
INSERT INTO Employees VALUES (109, 'Reza', 'Bagheri', 102)
INSERT INTO Employees VALUES (110, 'Abbas', 'Najafian', 102)

```

مثال اول: می-خواهیم فهرست کارمندان را به همراه نام مدیر آنها و شماره سطح درخت نمایش دهیم. کوئری زیر نمونه‌ای از یک کوئری بر اساس CTE بازگشتی می-باشد.

```

WITH cteReports(EmpID, FirstName, LastName, MgrID, EmpLevel)
AS
(
SELECT EmployeeID, FirstName, LastName, ManagerID, 1
FROM Employees
WHERE ManagerID ISNULL
UNION ALL
SELECT e.EmployeeID, e.FirstName, e.LastName, e.ManagerID, r.EmpLevel + 1
FROM Employees e INNER JOIN cteReports r
ON e.ManagerID = r.EmpID
)
SELECT
FirstName + ' ' + LastName AS FullName,
EmpLevel,
(SELECT FirstName + ' ' + LastName FROM Employees
WHERE EmployeeID = cteReports.MgrID) AS Manager
FROM cteReports
ORDER BY EmpLevel, MgrID

```

کوئری اول در بدن CTE رکورد مدیری را می-دهد که ریشه درخت بوده و بالاسری ندارد و شماره سطح درخت این رکورد را 1 در نظر می-گیرد.

کوئری دوم در بدن CTE از یک JOIN بین Employees و cteReports استفاده کرده و کارمندان زیر دست هر کارمند قبلی (فرزندان) را بدست آورده و مقدار شماره سطح آنرا به صورت Level+1 تنظیم می-کند.

در نهایت با استفاده از CTE و یک subquery جهت بدست آوردن نام مدیر هر کارمند، نتیجه نهایی تولید می-شود.

مثال دوم: می-خواهیم شناسه یک کارمند را بدهیم و نام او و نام مدیران وی را به عنوان جواب در خروجی بگیریم.

```

WITH cteReports(EmpID, FirstName, LastName, MgrID, EmpLevel)
AS
(
SELECT EmployeeID, FirstName, LastName, ManagerID, 1
FROM Employees
WHERE EmployeeID = 110
UNION ALL
SELECT e.EmployeeID, e.FirstName, e.LastName, e.ManagerID, r.EmpLevel + 1
FROM Employees e INNER JOIN cteReports r
ON e.EmployeeID = r.MgrID

```

```
)
SELECT
FirstName +' '+ LastName AS FullName,
EmpLevel
FROMcteReports
ORDERBY EmpLevel
```

اگر دقیق کنید اولین تفاوت در خط اول مشاهده می-شود. در اینجا مشخص می-کند که اولین سری از رکوردها چگونه انتخاب شود. مثلاً کارمندی را می-خواهیم که شناسه آن 110 باشد.

دومین تفاوت اصلی این کوئری با مثال قبلی، در قسمت دوم دیده می-شود. شما می-خواهید مدیر (پدر) کارمندی که در آخرین پردازش در جدول موقع قرار گرفته است را استخراج کنید.

a temporary named result set [1]

recursive [2]

nonrecursive [3]

Scope [4]

مثلاً محدوده کدهای یک روال یا یک تابع [5]

nonrecursive [6]

recursive [7]

member [8]

Infinite loop [9]

نظرات خوانندگان

نوبت‌دهنده: همراه
تاریخ: ۱۳۹۳/۰۱/۲۶ ۱۱:۷

ضمن تشکر از پست مفید جناب عمران یکی از استفاده‌های CTE افزودن شماره ردیف به ساختار خروجی و محدود کردن نتیجه با توجه به شماره ردیف است.
مثلاً ردیف 20 تا 30 ... که البته با پارامتر پاس می‌شوند.

```
WITH RCTE AS
(
SELECT TOP (100)
ROW_NUMBER() OVER (ORDER BY Invoice.InsertDate ASC) AS RowNumber,
Invoice.ID,
Invoice.PreInvoiceNo,
Invoice.InvoiceNo,
Invoice.IssueDate,
Invoice.CustomerID, ...
FROM
Invoice
WHERE
Invoice.HistorySequence = 1
)
SELECT DISTINCT
RCTE.ID,
RCTE.PreInvoiceNo,
RCTE.InvoiceNo,
(dbo.fnc_Calendar_Gregorian_to_Persian(RCTE.IssueDate) + 'T' + CONVERT(CHAR(8), RCTE.IssueDate, 14)) AS IssueDate,
RCTE.CustomerID,
Customer.NameEn AS CustomerNameEn,
Customer.NameFa AS CustomerNameFa,
FROM
RCTE
INNER JOIN Customer ON RCTE.CustomerID = Customer.ID
WHERE
RowNumber BETWEEN @StartFrom AND (@RowsCount + @StartFrom - 1)
```

روشی را که مایکروسافت برای پرداختن به مقوله NoSQL تاکنون انتخاب کرده است، قرار دادن ویژگی‌هایی خاصی از دنیای NoSQL مانند امکان تعریف اسکیمی متفاوت، داخل مهمترین بانک اطلاعاتی رابطه‌ای آن، یعنی SQL Server است، که در ادامه به آن خواهیم پرداخت. همچنین در سمت محصولات پردازش ابری آن نیز امکان دسترسی به محصولات NoSQL کاملی وجود دارد.

Azure table storage (1)

Azure table storage در حقیقت یک Key-value store ابری است و برای کار با آن از اینترفیس پروتکل استاندارد OData استفاده می‌شود. علت استفاده و طراحی یک سیستم Key-value store در اینجا، مناسب بودن اینگونه سیستم‌ها جهت مقاصد عمومی است و به این ترتیب می‌توان به بازه بیشتری از مصرف کنندگان، خدمات ارائه داد. پیش از ارائه Azure table storage، مایکروسافت سرویس خاصی را به نام SQL Server Data Services که به آن Azure SQL نیز گفته می‌شود، معرفی کرد. این سرویس نیز یک Key-Value store است؛ هر چند از SQL Server به عنوان مخزن نگهداری اطلاعات آن استفاده می‌کند.

SQL Azure XML Columns (2)

فیلهای XML از سال 2005 به امکانات توکار SQL Server اضافه شدند و این نوع فیلهای XML، بسیاری از مزایای دنیای NoSQL را درون SQL Server رابطه‌ای مهیا می‌سازند. برای مثال با تعریف یک فیلد به صورت XML، می‌توان از هر ردیف به ردیف دیگر، اطلاعات متفاوتی را ذخیره کرد؛ به این ترتیب امکان کار با یک فیلد که می‌تواند اطلاعات یک شیء را قبول کند و در حقیقت امکان تعریف اسکیمی پویا و متغیر را در کنار امکانات یک بانک اطلاعاتی رابطه‌ای که از اسکیمی ثابت پشتیبانی می‌کند، میسر می‌شود. در این حالت در هر ردیف می‌توان تعدادی ستون ثابت را با یک ستون XML با اسکیمی کاملاً پویا ترکیب کرد. همچنین SQL Server در این حالت قابلیتی را ارائه می‌دهد که در بسیاری از بانک‌های اطلاعاتی NoSQL میسر نیست. در اینجا در صورت نیاز و لزوم می‌توان اسکیمی کاملاً مشخصی را به یک فیلد XML نیز انتساب داد؛ هر چند این مورد اختیاری است و می‌توان یک XML un typed را نیز بکار برد. به علاوه امکانات کوئری گرفتن توکار از این اطلاعات را به کمک XPath ترکیب شده با T-SQL، نیز فراموش نکنید.

بنابراین اگر یکی از اهداف اصلی گرایش شما به سمت دنیای NoSQL، استفاده از امکان تعریف اطلاعاتی با اسکیمی متفاوت و پویا است، فیلهای نوع XML اس کیوال سرور را مدنظر داشته باشید.

یک مثال عملی: فناوری Azure Dev Fabric's Table Storage (نسخه Windows ویندوز Azure) که روی ویندوز‌های معمولی اجرا می‌شود؛ یک شبیه ساز خانگی) به کمک SQL Server و فیلهای XML آن طراحی شده است.

SQL Azure Federations (3)

در اینجا منظور از Federations در حقیقت همان پیاده سازی قابلیت Sharding بانک‌های اطلاعاتی SQL Azure NoSQL است که برای توزیع اطلاعات بر روی سرورهای مختلف طراحی شده است. به این ترتیب دو قابلیت Partitioning و همچنین Replication به صورت خودکار در دسترس خواهند بود. هر Partition در اینجا، یک SQL Azure کامل است. بنابراین چندین بانک اطلاعاتی فیزیکی، یک بانک اطلاعاتی کلی را تشکیل خواهند داد.

هر چند در اینجا Sharding (که به آن Federation member گفته می‌شود) و در پی آن مفهوم «عقابت یک دست شدن اطلاعات» وجود دارد، اما درون یک Shard یا یک Federation member مفهوم ACID پیاده سازی شده است. از این جهت که هر Shard واقعاً یک بانک اطلاعاتی رابطه‌ای است. اینجا است که مفهوم برنامه‌های Multi-tenancy را برای درک آن باید درنظر داشت. برای نمونه یک برنامه وب را درنظر بگیرید که قسمت اصلی اطلاعات کاربران آن بر روی یک Shard قرار دارد و سایر اطلاعات بر روی سایر Shards پراکنده شده‌اند. در این حالت است که یک برنامه وب با وجود مفهوم ACID در یک Shard می‌تواند سریع پاسخ دهد که آیا کاربری پیشتر در سایت ثبت نام کرده است یا خیر و از ثبت نام‌های غیرمجاز جلوگیری به عمل آورد.

در اینجا تنها موردی که پشتیبانی نشده‌است، کوئری‌های Fan-out می‌باشد که پیشتر در مورد آن بحث شد. از این جهت که با نحوه خاصی که Sharding آن طراحی شده است، نیازی به تهیه کوئری‌هایی که به صورت موازی بر روی کلیه Shards برای جمع آوری اطلاعات اجرا می‌شوند، نیست. هر چند از هر shard با استفاده از برنامه‌های دات نت، می‌توان به صورت جداگانه نیز کوئری

OData (4)

اگر به CouchDB و امکان دسترسی به امکانات آن از طریق وب دقت کنید، در محصولات مایکروسافت نیز این دسترسی REST API پیاده سازی شده‌اند.

یک RESTful API است برای دسترسی به اطلاعاتی که به شکل XML یا JSON بازگشت داده می‌شوند. انواع و اقسام کلاینت‌هایی برای آن از جاوا اسکریپت گرفته تا سیستم‌های موبایل، دات نت و جاوا، وجود دارند. از این API نه فقط برای خواندن اطلاعات، بلکه برای ثبت و به روز رسانی داده‌ها نیز استفاده می‌شود. در سیستم‌های جاری مایکروسافت، بسیاری از فناوری‌ها، اطلاعات خود را به صورت OData در اختیار مصرف کنندگان قرار می‌دهند مانند Azure table storage، کار با SQL Azure از طریق WCF Data Services (جایی که OData از آن نشات گرفته شده)، Azure Data Market (برای ارائه فیدهایی از اطلاعات خصوصاً رایگان)، ابزارهای گزارشگیری مانند SQL Server reporting services، لیست‌های شیرپوینت و غیره. به این ترتیب به بسیاری از قابلیت‌های دنیای NoSQL مانند کار با اطلاعات JSON بدون ترک دنیای رابطه‌ای می‌توان دسترسی داشت.

5) امکان اجرای MongoDB و امثال آن روی سکوی کاری Azure

امکان توزیع MongoDB بر روی یک Worker role سکوی کاری Azure وجود دارد. در این حالت بانک‌های اطلاعاتی این سیستم‌ها بر روی Azure Blob Storage قرار می‌گیرند که به آن‌ها Azure drive نیز گفته می‌شود. همین روش برای سایر بانک‌های اطلاعاتی NOSQL نیز قابل اجرا است.

به علاوه امکان اجرای Hadoop نیز بر روی Azure وجود دارد. مایکروسافت به کمک شرکتی به نام Hortonworks نسخه ویندوزی HortonWorks را افرادی تشکیل داده‌اند که پیشتر در شرکت یاهو بر روی پروژه Hadoop کار را توسعه داده‌اند. HortonWorks می‌کرده‌اند.

6) قابلیت‌های فرا رابطه‌ای SQL Server

الف) فیلدهای XML (که در ابتدای این مطلب به آن پرداخته شد). به این ترتیب می‌توان به یک اسکیمی انعطاف پذیر، بدون از دست دادن ضمانت ACID رسید.

ب) فیلد HierarchyId برای ذخیره سازی اطلاعات چند سطحی. برای مثال در بانک‌های اطلاعاتی NoSQL سندگرا، یک سند می‌تواند سند دیگری را در خود ذخیره کند و ال آخر.

ج) Sparse columns: ستون‌های اسپارس تقریباً شبیه به Key-value stores عمل می‌کنند و یا حتی Wide column stores این نیست که واقعاً قابل مقایسه است. در اینجا هنوز اسکیمی وجود دارد، اما برای نمونه علت استفاده از Wide column stores نمی‌دانید ساختار داده‌های مورد استفاده چیست، بلکه در این حالت می‌دانیم که در هر ردیف تنها از تعداد محدودی از فیلدها استفاده خواهیم کرد. به همین جهت در هر ردیف تمام فیلدها قرار نمی‌گیرند، چون در اینصورت تعدادی از آن‌ها همواره خالی باقی می‌مانند. مایکروسافت این مشکل را با ستون‌های اسپارس حل کرده است؛ در اینجا هر چند ساختار کلی مشخص است، اما مواردی که هر بار استفاده می‌شوند، تعداد محدودی می‌باشند. به این صورت SQL Server تنها برای ستون‌های دارای مقدار، فضای را اختصاص می‌دهد. به این ترتیب از لحاظ فیزیکی و ذخیره سازی نهایی، به همان مزیت Wide column stores خواهیم رسید.

د) در اس کیوال سرور بسیار شبیه به پیوست‌های سندگاهی بانک‌های اطلاعاتی NoSQL سندگرا هستند. در اینجا نیز اطلاعات در فایل سیستم ذخیره می‌شوند اما ارجاعی به آن‌ها در جداول مرتبط وجود خواهد داشت.

SQL Server Parallel Data Warehouse Edition (7)

نگارش خاصی از SQL Server است که در آن یک شبکه از SQL Serverها به صورت یک وله منطقی در اختیار برنامه نویس‌ها قرار می‌گیرد.

این نگارش، از فناوری خاصی به نام MPP یا massively parallel processing برای پردازش کوئری‌ها استفاده می‌کند. در اینجا همانند بانک‌های اطلاعاتی NoSQL، یک کوئری به نود اصلی ارسال شده و به صورت موازی بر روی تمام نودها پردازش گردیده (همان مفهوم Map Reduce که پیشتر در مورد آن بحث شد) و نتیجه در اختیار مصرف کننده قرار خواهد گرفت. نکته مهم آن نیز در عدم نیاز به نوشتن کدی جهت رخدادن این عملیات از طرف برنامه نویس‌ها است و موتور پردازشی آن جزوی از سیستم اصلی است. تنها کافی است یک کوئری SQL صادر گردد تا نتیجه نهایی از تمام سرورها جمع آوری و بازگردانده شود. این نگارش ویژه تنها به صورت یک Appliance به فروش می‌رسد (به صورت سخت افزار و نرم افزار باهم) که در آن CPU‌ها،

فضاهای ذخیره سازی اطلاعات و جزئیات شبکه به دقت از پیش تنظیم شده‌اند.

یکی از محصولات پرکاربرد و حرفه ای مایکروسافت در زمینه تولید گزارش SQL Server Reporting Services یا به اختصار SSRS میباشد . در این پست نحوه ایجاد یک گزارش ساده به صورت والد و فرزندی ، انتشار گزارش روی وب سرور و مدیریت نمایش ستون ها با استفاده از Expression ها در محیط BIDS بیان می کنم

برای انجام این پروسه ، از ابزار BIDS استفاده خواهیم کرد . همچنین برای اطلاعات و داده ها از دیتابیس آزمایشی Adventure works استفاده می کنیم ([دانلود](#))

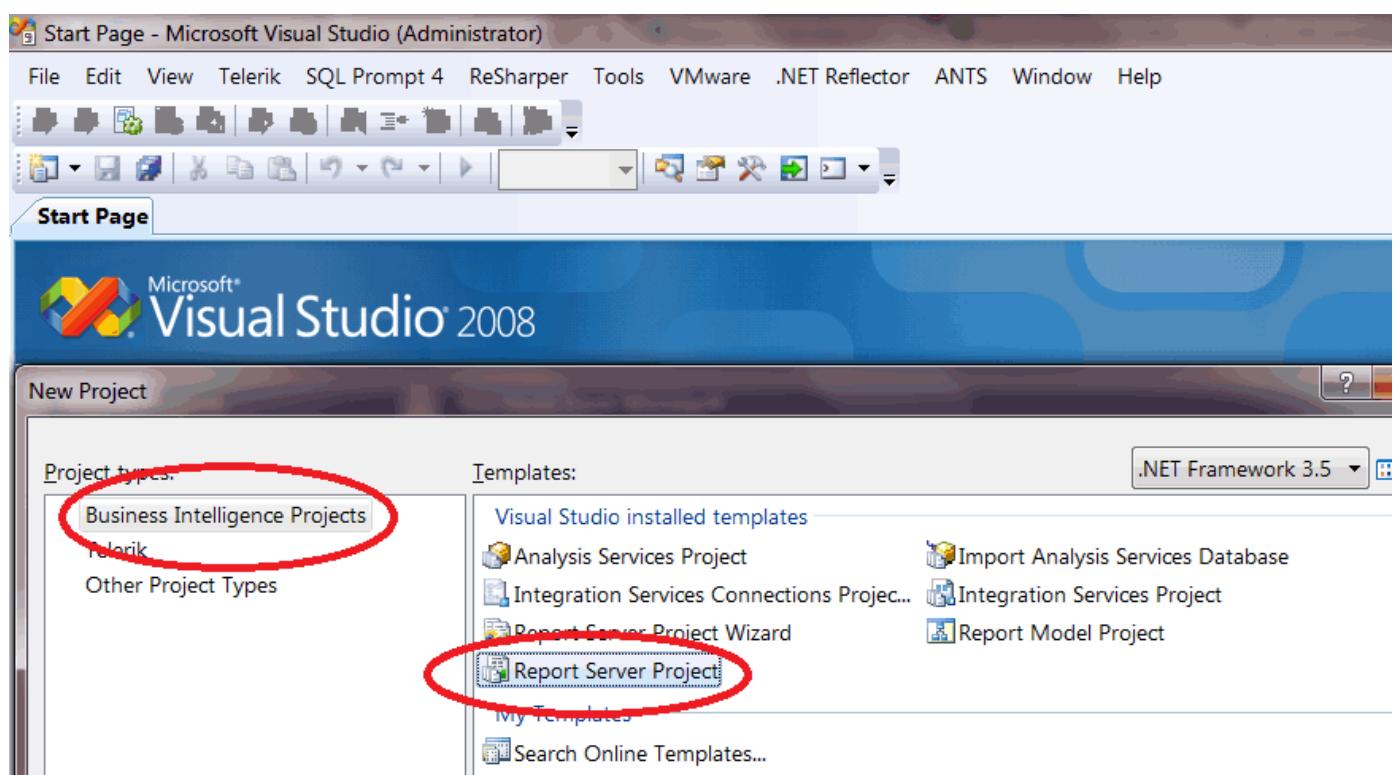


- + AdventureWorks
- + AdventureWorksDW
- + AdventureWorksDW2008R2
- + AdventureWorksLT
- + AdventureWorksLT2008R2

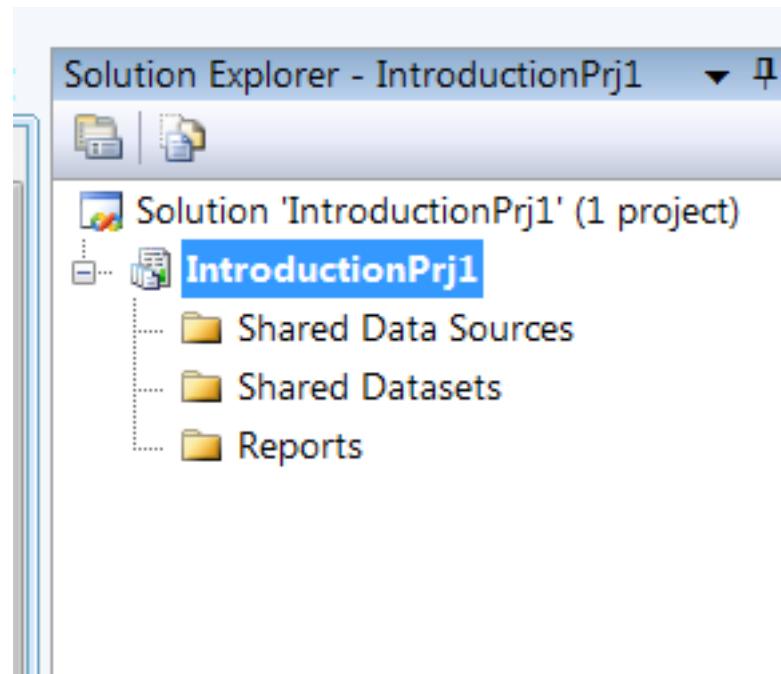
صورت مسئله : گزارش از دپارتمان‌های کاری و تعداد کارمندان مرد و زن هر دپارتمان . که با کلیک روی هر سطر گزارش (بسته به جنسیت) بتوان لیست تمام افراد آن دپارتمان را دید .

بخش اول :

برای شروع BIDS را باز کرده و یک پروژه از نوع Report Server Project استارت بزنید .

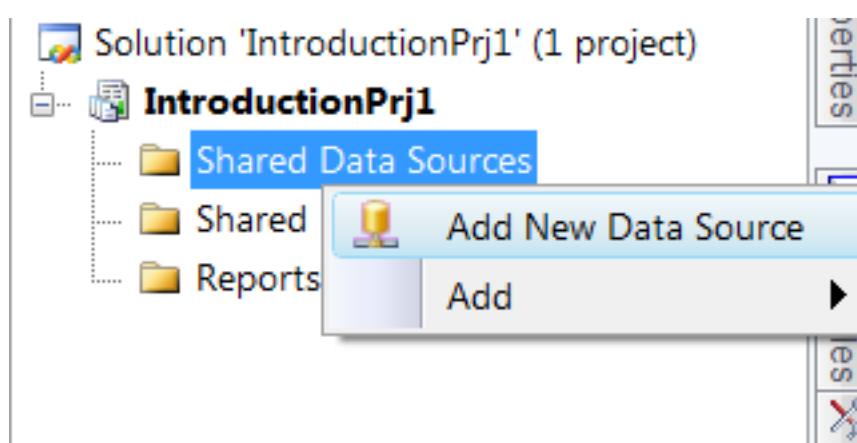


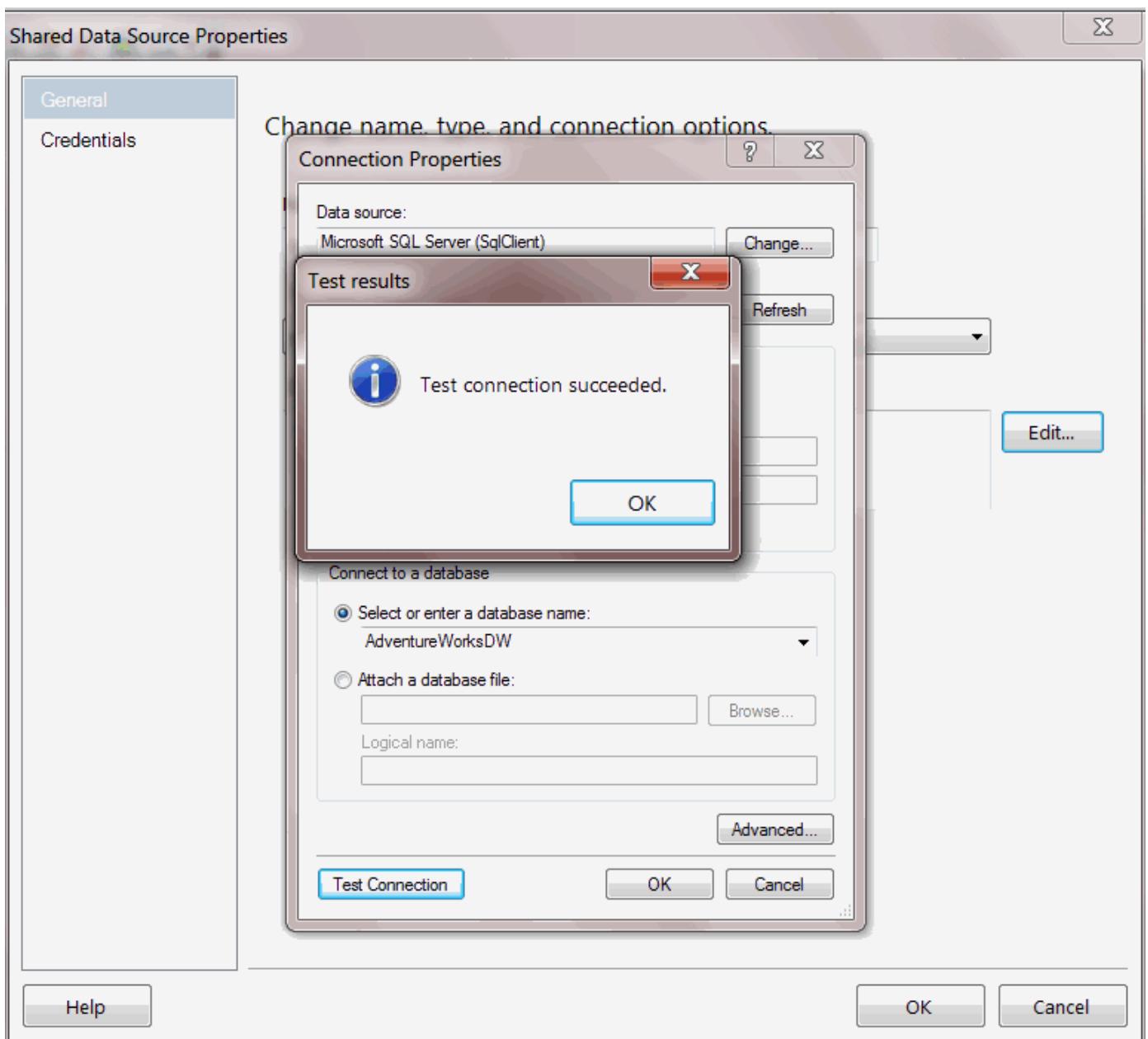
نام این پروژه را در اینجا introductionPrj1 قرار دادم .



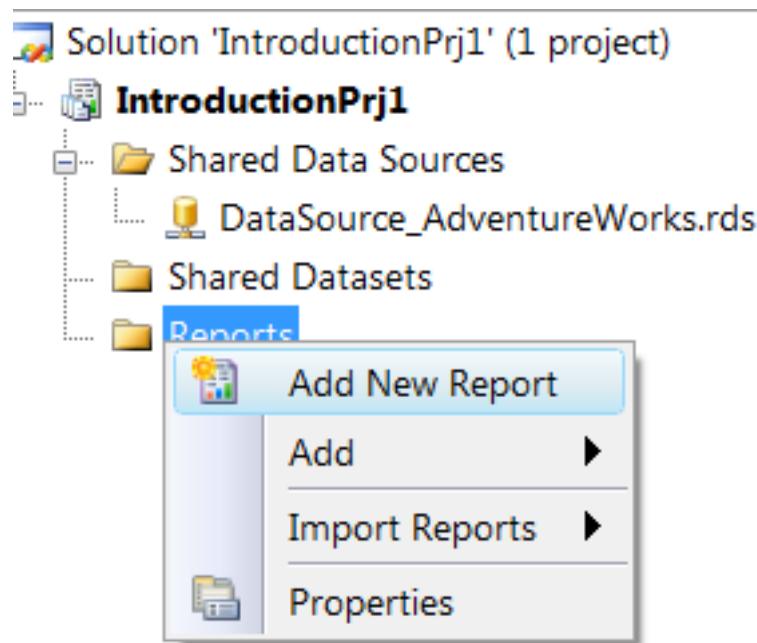
همانطور که ملاحظه می‌کنید، سه پوشه در Solution explorer قرار دارد که برای تعریف پایگاه داده با پوشه Shared Data Sources و برای تعریف گزارشات از پوشه Reports استفاده خواهیم کرد.

برای این منظور ابتدا یک data source تعریف می‌کنیم:





و سپس شروع به ساختن یک گزارش می‌کنیم :



مراحل را تا رسیدن به مرحله تعریف Query پی میگیریم .



:انتخاب دیتا سورس

Select the Data Source

Select a data source from which to obtain data for this report or create a new one.

Shared data source

DataSource_AdventureWorks

New data source

Name:

DataSource1

Type:

Microsoft SQL Server

Connection string:

در اینجا می‌خواهیم قسمت اول گزارش یعنی فهرست کردن تعداد کارمندان هر دپارتمان را به تفکیک جنسیت مشاهده کنیم :

```
SELECT DEPARTMENTNAME, GENDER, COUNT(1) AS COUNT FROM DBO.DIMEMPLOYEE GROUP BY DEPARTMENTNAME,GENDER  
ORDER BY DEPARTMENTNAME,GENDER
```

The screenshot shows the 'Report Wizard' interface with the title 'Design the Query'. Below it, a sub-instruction reads 'Specify a query to execute to get the data for the report.' A note below suggests using a query builder. A blue button labeled 'Query Builder...' is visible. The 'Query string:' section contains the following SQL code:

```
SELECT DepartmentName, Gender, COUNT(1) AS COUNT  
FROM DimEmployee  
GROUP BY DepartmentName, Gender  
ORDER BY DepartmentName, Gender
```

برای مشاهده صحت دستور می‌توانید از Query Builder کمک بگیرید :

نحوه تهیه گزارش در SSRS و انتشار آن روی وب سرور

Query Designer

Edit as Text Import... | SQL

DimEmployee

* (All Columns)
EmployeeKey
ParentEmployeeKey
EmployeeNationalIDAlternateKey
ParentEmployeeNationalIDAlternateKey

Column	Alias	Table	Outp...	Sort Type	Sort Order	Group By	Filter	Or...
DepartmentN...		DimEmplo...	<input checked="" type="checkbox"/>	Ascending	1	Group By		
Gender		DimEmplo...	<input checked="" type="checkbox"/>	Ascending	2	Group By		
1	COUNT		<input checked="" type="checkbox"/>			Count		
			<input checked="" type="checkbox"/>					
			<input checked="" type="checkbox"/>					

```
SELECT DepartmentName, Gender, COUNT(1) AS COUNT
FROM DimEmployee
GROUP BY DepartmentName, Gender
ORDER BY DepartmentName, Gender
```

DepartmentName	Gender	COUNT
Document Control	F	1
Document Control	M	4
Engineering	F	3
Engineering	M	3
Executive	C	2

1 of 30 | Cell is Read Only.

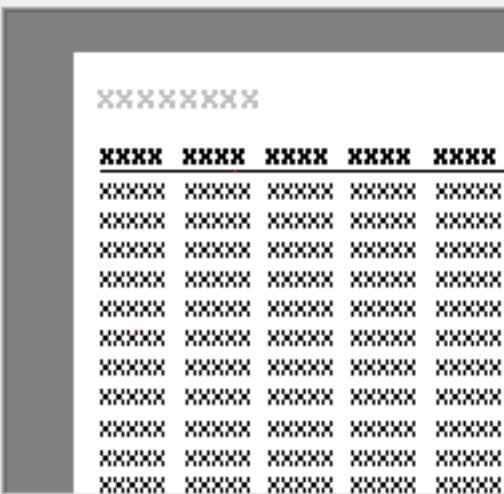
Help OK Cancel

ادامه تنظیمات را مانند تصویر پی بگیرید (تعریف Tabular بودن گزارش و طراحی جدول و theme و نام گذاری گزارش)

Select the Report Type

Select the type of report that you want to create.

- Tabular
 - Matrix



Design the Table

Choose how to group the data in the table.

Available fields:

Page>

Group>

Details>

Displayed fields:

DepartmentName
Gender
COUNT

XXXX
XXXX
XXXX
XXXXX
XXX XXX XXX
XXX XXX XXX
XXXXX
XXX XXX XXX
XXX XXX XXX



Choose the Table Style

Choose a style for the table.

Slate
Forest
Corporate
Bold
Ocean
Generic

Completing the Wizard

Provide a name and click Finish to create the new report.

Report name:

MyParentReport

Report summary:

Data source: DataSource_AdventureWorks

Connection string:

Report type: Table

Layout type: Stepped

Style: Ocean

Details: DepartmentName, Gender, COUNT

Query:

```
SELECT DepartmentName, Gender, COUNT(1) AS COUNT
FROM DimEmployee
GROUP BY DepartmentName, Gender
ORDER BY DepartmentName, Gender
```

به این ترتیب بخش اول گزارش ایجاد شد . حال باید زیر گزارش مربوطه را ایجاد کنیم :

مجددا مراحل را برای ساخت یک گزارش جدید پیگیری کنید و برای دستور کوئری از دستور زیر استفاده کنید :

```
SELECT EMPLOYEEKEY, FIRSTNAME, LASTNAME, MIDDLENAME, TITLE, HIREDATE, BIRTHDATE, EMAILADDRESS, PHONE, GENDER
FROM DBO.DIMEMPLOYEE WHERE DEPARTMENTNAME=@DEPARTMENTNAME AND GENDER=@GENDER
```

Design the Query

Specify a query to execute to get the data for the report.

Use a query builder to design your query.

Query Builder...

Query string:

```
SELECT EmployeeKey, FirstName, LastName, MiddleName, Title, HireDate, BirthDate, EmailAddress, Phone,  
Gender  
FROM DimEmployee  
WHERE (DepartmentName = @DEPARTMENTNAME) AND (Gender = @GENDER)
```

و تست گزارش :

The screenshot shows the Microsoft SQL Server Management Studio (Query Designer) interface. On the left, the 'DimEmployee' table is selected in the 'Query Parameters' dialog, with the 'EmployeeKey' column checked. Below it, a preview grid shows columns: EmployeeKey, FirstName, LastName, MiddleName, Title, HireDate, BirthDate, EmailAddress, and Phone. The first five rows of data are displayed. At the bottom of the preview grid, it says 'Cell is Read Only.' On the right, the 'Query Parameters' dialog is open, prompting for values for '@DEPARTMENTNAME' (Finance) and '@GENDER' (F). The 'OK' button is highlighted.

Column	Alias	Table	Output Type	Sort Type
EmployeeKey		DimEmployee	<input checked="" type="checkbox"/>	
FirstName		DimEmployee	<input checked="" type="checkbox"/>	
LastName		DimEmployee	<input checked="" type="checkbox"/>	

```

SELECT EmployeeKey, FirstName, LastName, MiddleName, Title, HireDate
FROM DimEmployee
WHERE (DepartmentName = @DEPARTMENTNAME) AND (Gender = @GENDER)
    
```

EmployeeKey	FirstName	LastName	MiddleName	Title	HireDate	BirthDate	EmailAddress	Phone
61	Deborah	Poe	E	Accounts Rec...	01/19/1999 ...	04/07/1966 ...	deborah0@a...	602-5
73	Wendy	Kahn	Beth	Finance Man...	01/26/1999 ...	11/12/1974 ...	wendy0@ad...	248-5
96	Candy	Spoon	L	Accounts Rec...	02/07/1999 ...	03/26/1966 ...	candy0@adv...	920-5
182	Barbara	Morland	C	Accountant	03/22/1999	02/01/1966	barbara1@...	822-5

Cell is Read Only.

Help OK Cancel

و بقیه قسمت‌ها مانند قبل :

Report Wizard

Design the Table

Choose how to group the data in the table.

Available fields:

- Page>
- Group>
- Details>
- < Remove

Displayed fields:

- Gender
- EmployeeKey
- FirstName
- LastName
- MiddleName
- Title
- HireDate
- BirthDate
- EmailAddress
- Phone

XXXX
XXXX
XXXX
XXXXX
XXX XXX XXX
XXX XXX XXX
XXXXX
XXX XXX XXX
XXX XXX XXX

↑
↓

Report Wizard

Completing the Wizard

Provide a name and click Finish to create the new report.

Report name:

MyChildReport

Report summary:

Data source: DataSource_AdventureWorks

Connection string:

Report type: Table

Layout type: Stepped

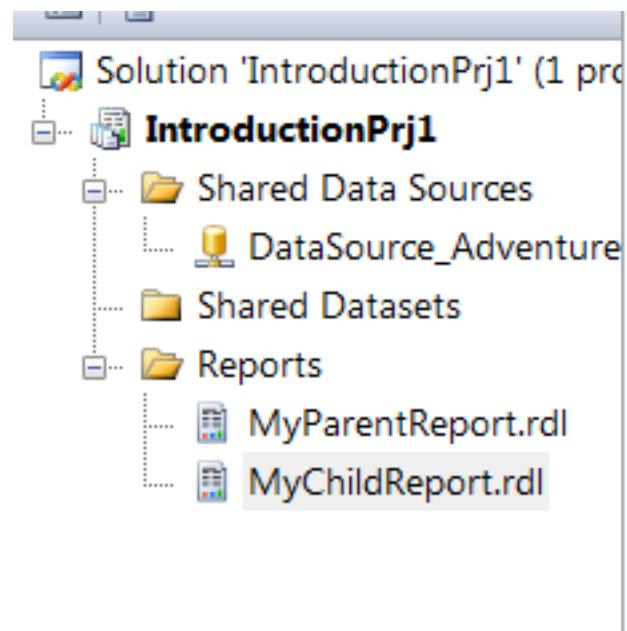
Style: Bold

Page: Gender

Details: EmployeeKey, FirstName, LastName, MiddleName, Title, HireDate, BirthDate, EmailAddress, Phone

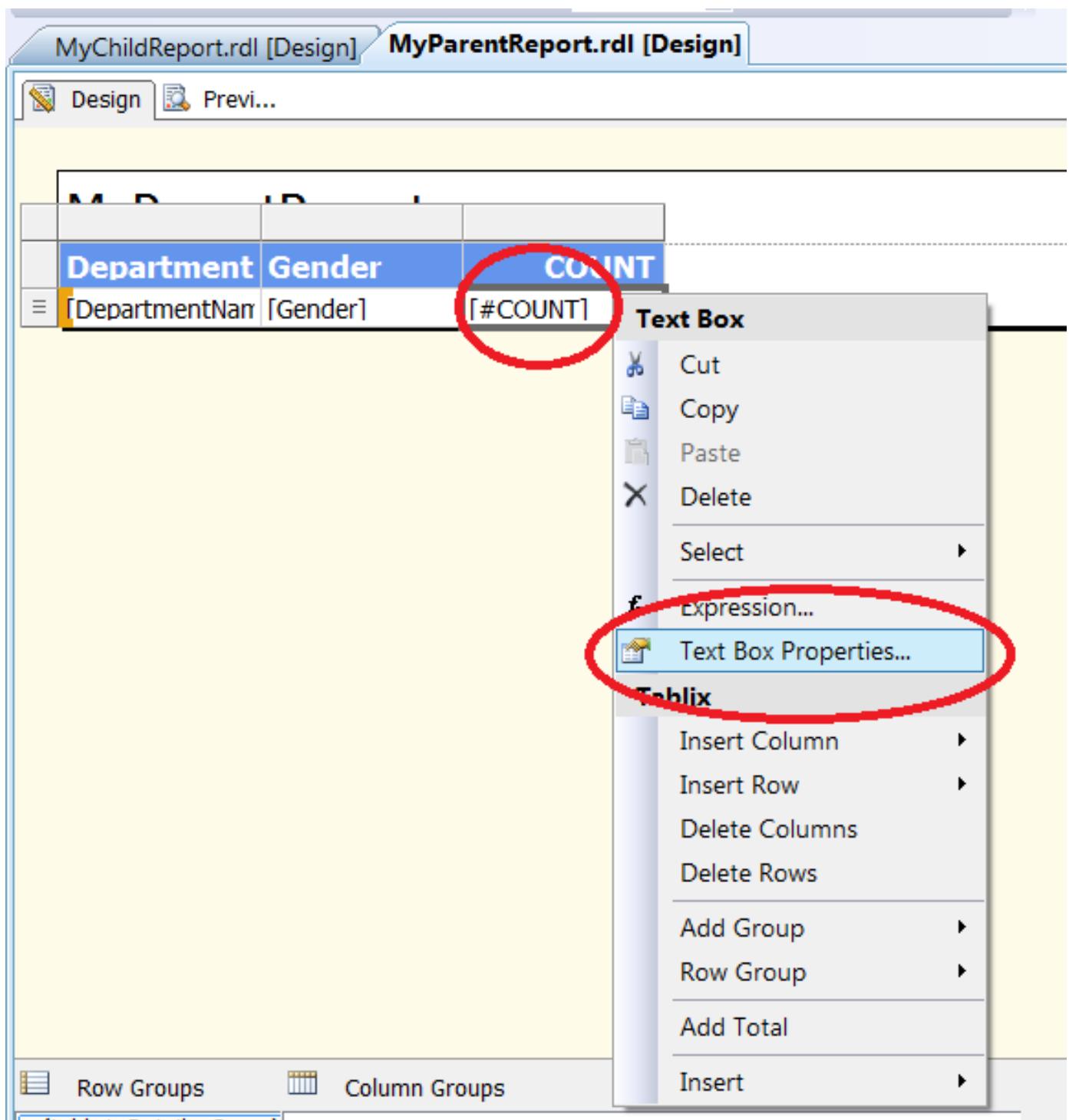
Query: SELECT EmployeeKey, FirstName, LastName, MiddleName, Title, HireDate, BirthDate, EmailAddress, Phone, Gender
FROM DimEmployee
WHERE (DepartmentName = @DEPARTMENTNAME) AND (Gender = @GENDER)

تا به این مرحله data source و گزارش‌ها ایجاد شدند :

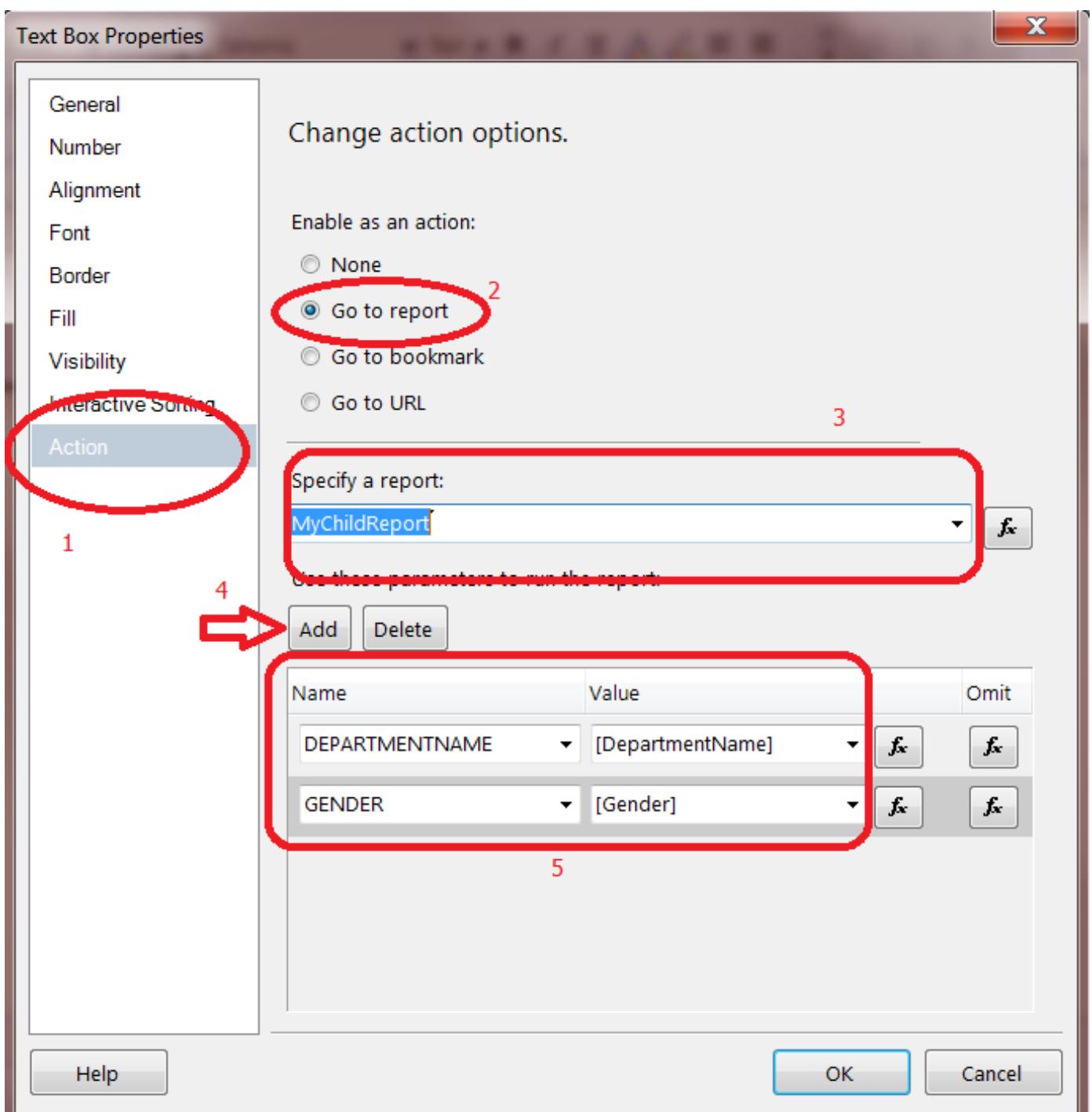


اکنون باید ارتباط بین دو گزارش را برقرار کنیم :

گزارش والد را باز کرده و روی ستون COUNT کلیک راست نموده و گزینه Properties را انتخاب نمایید :



سپس در تب action گزینه Go to Report را انتخاب نموده و گزارش فرزند را انتخاب نمایید.



در انتها هم باید پارامترها را تعریف کنید . خروجی مانند زیر خواهد بود :

Department	Gender	COUNT
Document	F	1
Document	M	4

از تب General قابل اعمال است.

و در نهایت با کلیک روی این سطر می‌توانید گزارش مرتبط را مشاهده کنید:

Employee Key	First Name	Last Name	Middle Name	Title	Hire Date	Birth Date	Email Address	Phone
56	Tengiz	Kharatishvili	N	Control Specialist	1/17/1999 12:00:00 AM	5/29/1980 12:00:00 AM	tengiz0@adventure-works.com	910-555-0116
92	Zainal	Arifin	T	Document Control Manager	2/5/1999 12:00:00 AM	3/2/1966 12:00:00 AM	zainal0@adventure-works.com	204-555-0115
130	Sean	Chai	N	Document Control Assistant	2/23/1999 12:00:00 AM	4/12/1977 12:00:00 AM	sean1@adventure-works.com	205-555-0132
216	Chris	Norred	K	Control Specialist	4/7/1999 12:00:00 AM	6/26/1977 12:00:00 AM	chris1@adventure-works.com	575-555-0126

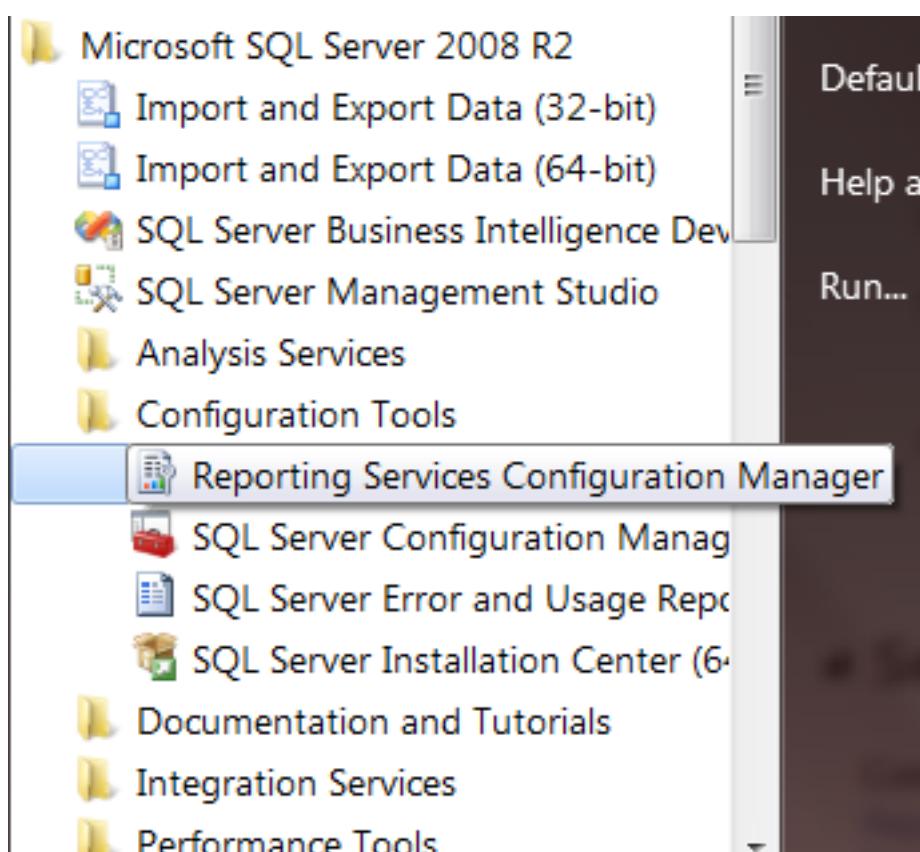
تا به این مرحله گزارش تکمیل شد که البته برای ظاهر آن هم باید فکری کرد که در این پست اشاره ای نمی‌شود.

بخش دوم :

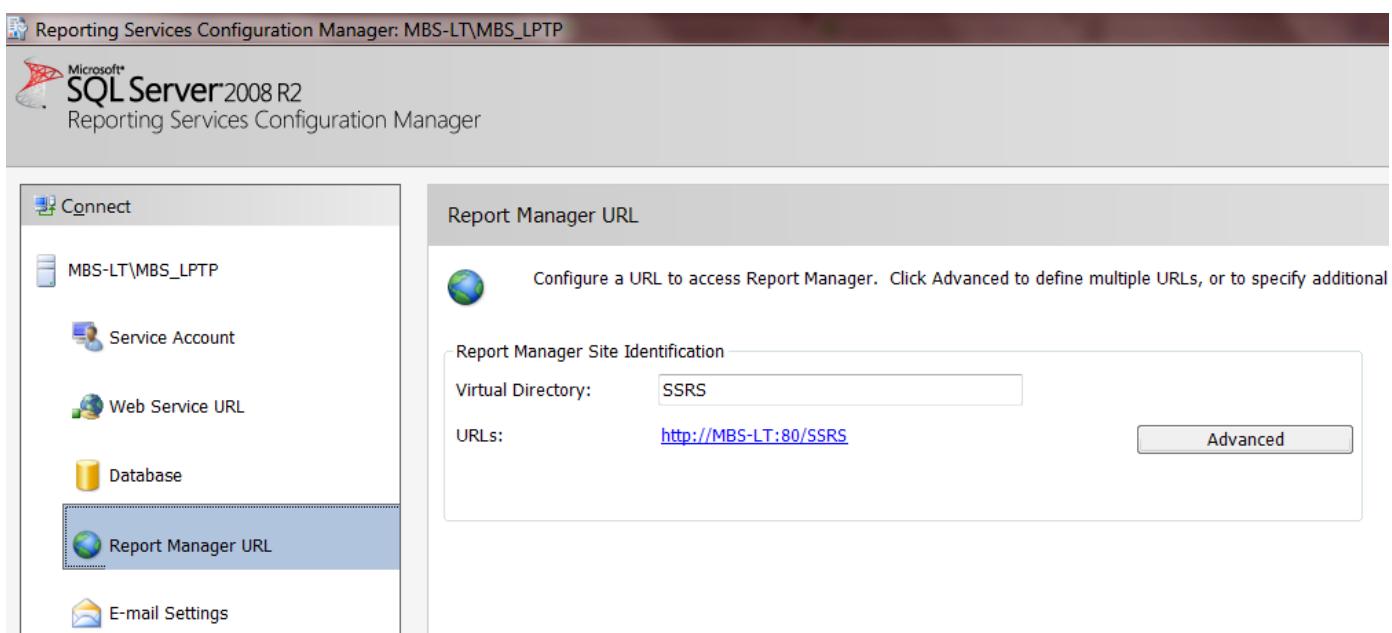
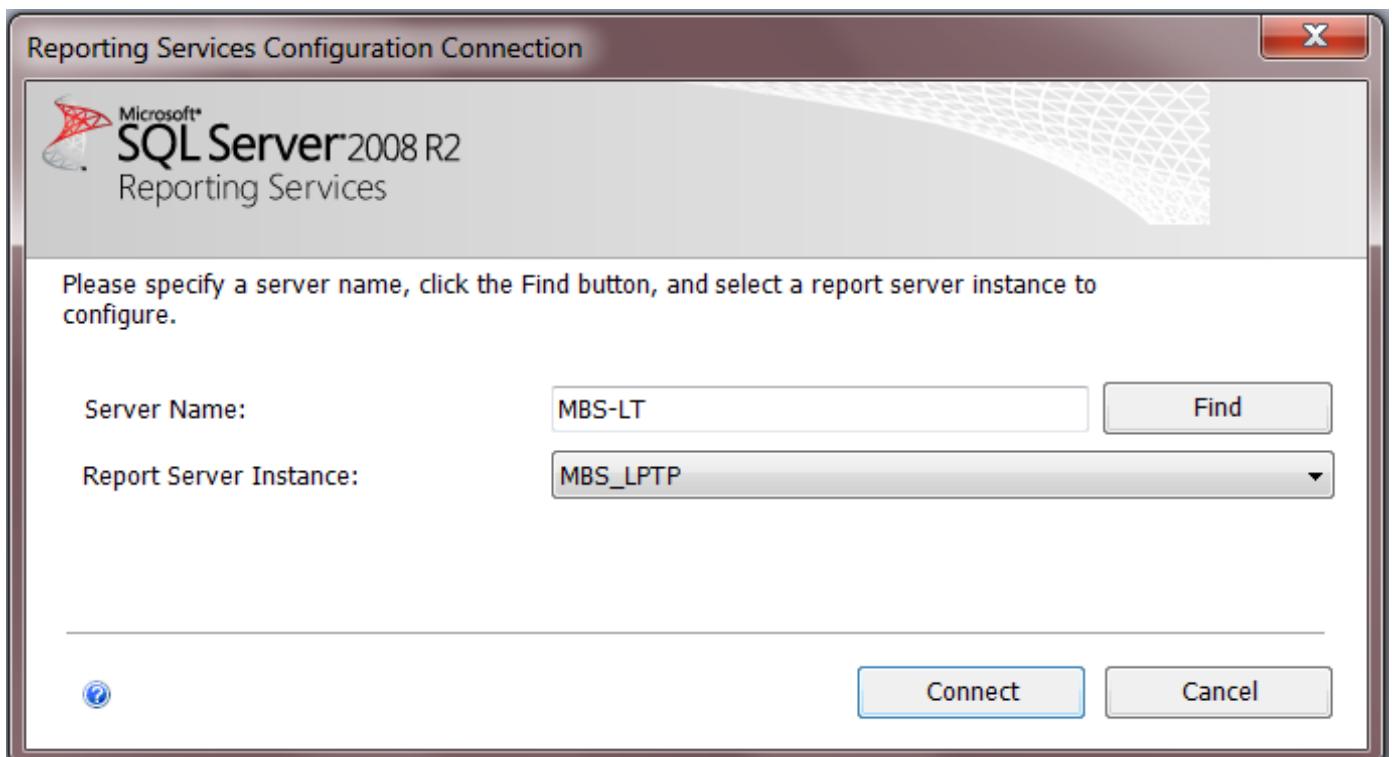
گزارش جاری فقط قابل استفاده از طریق BIDS است و با توجه به محدودیت دسترسی باید آن را در جایی قرار داد تا کاربران بتوانند از آن استفاده کنند. برای این منظور باید تنظیمات SSRS Web Application انجام شود تا بتواند روی سرور عملیاتی قرار گیرد.

در صورتی که تنظیمات SSRS برای قرار گرفتن روی وب سرور انجام نشده باشد و ما بخواهیم گزارش را Deploy کنیم خطاب دریافت خواهیم کرد.

پس در ادامه نحوه تنظیم وب سرور را بیان می‌کنم و پس از آن گزارش را روی وب سرور قرار می‌دهیم: برای این منظور باید برنامه SQL Server Reporting Services Configuration Manager که در مسیر نصب Reporting Services Configuration Manager است برویم

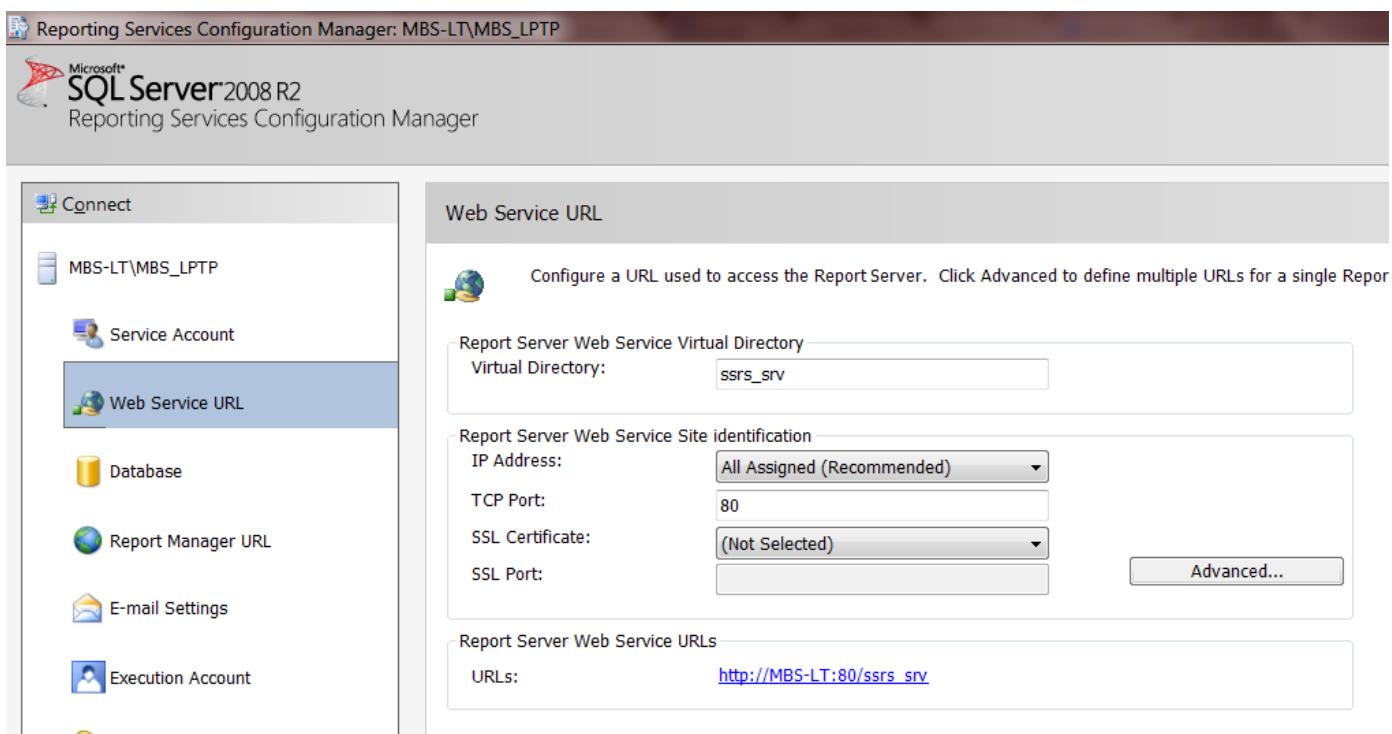


پس از اتصال به سرور به تب Report Manager Url بروید:



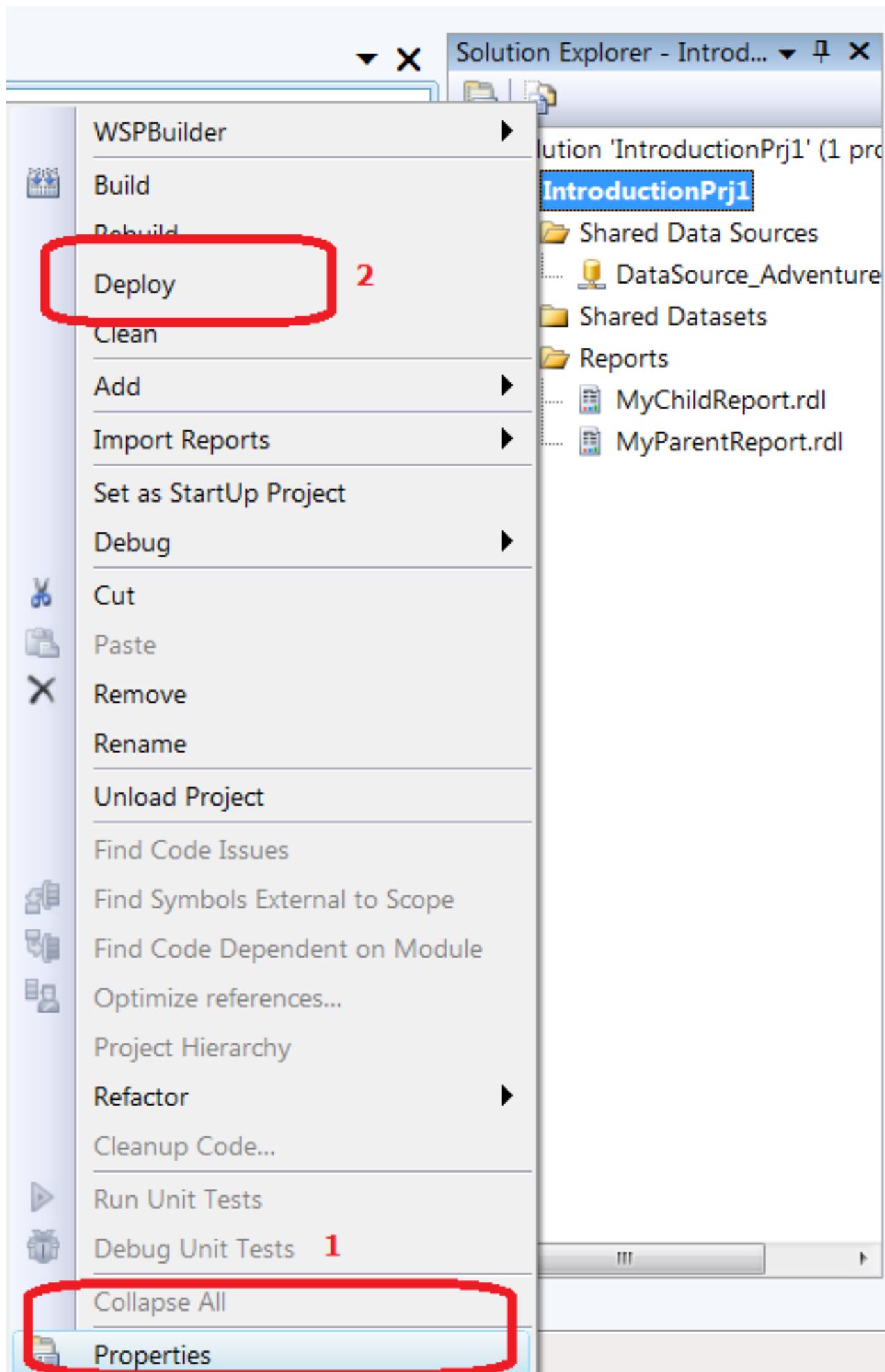
در این مرحله باید سرور را تنظیم کنید تا بتوانیم پروژه را روی آن Deploy کنیم . از باز بودن پورت اطمینان حاصل کنید . سپس وب سرویس را تنظیم کنید که هر دو فقط شامل نام Credential و Virtual Directory آن می شود . (مگر اینکه تنظیمات خاصی داشته باشید).

نحوه تهیه گزارش در SSRS و انتشار آن روی وب سرور

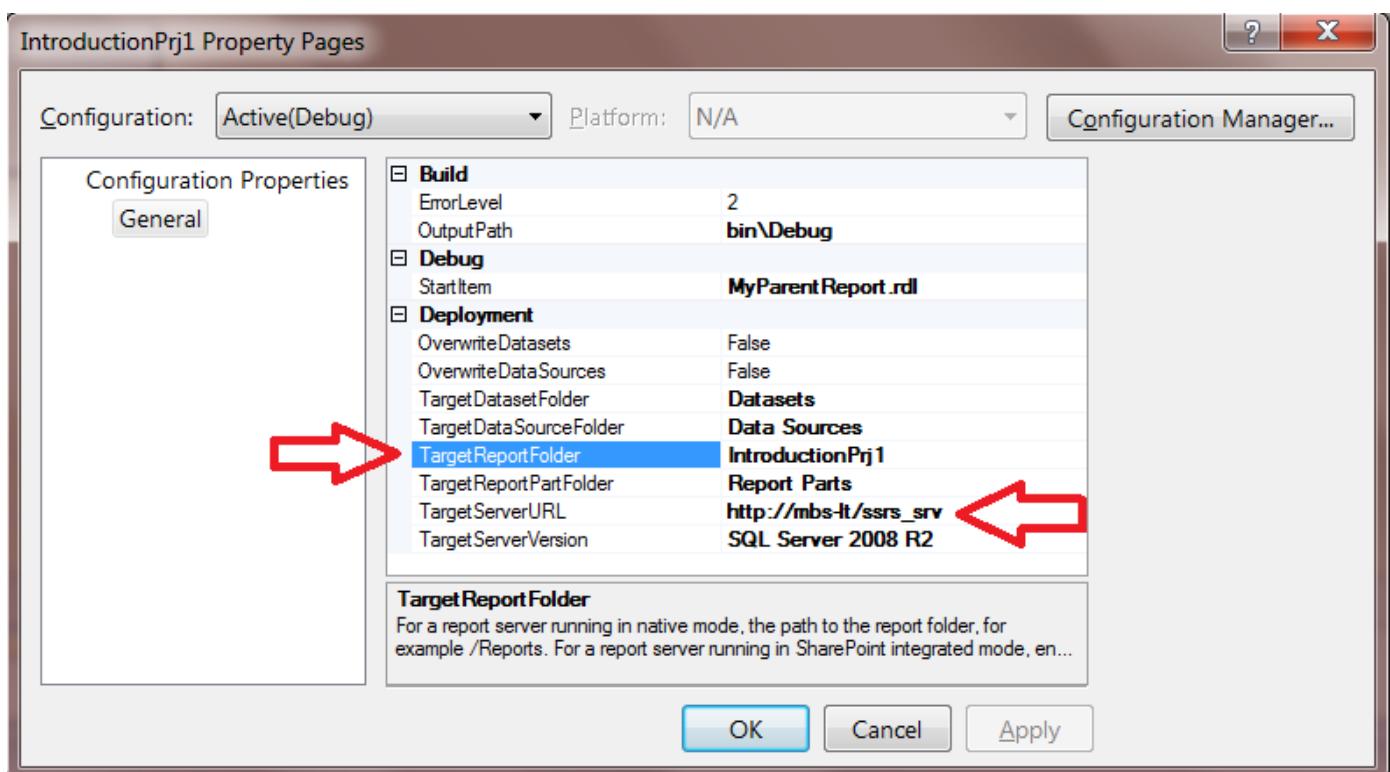


در صورت اجرا کردن مسیر URL باید بتوانید صفحه خانگی آن را مشاهده کنید :

که البته هنوز هیچ گزارشی روی آن قرار نگرفته است. سپس به گزارش خود باز می‌گردیم تا تنظیمات سرور را روی BIDS تکمیل کنیم :
برای این منظور روی پروژه کلیک راست کنید و ابتدا روی Properties کلیک کنید.



تنظمات مهم شامل مسیر سرور که برابر با همان سرویسی است که تنظیم کرده اید و مسیر پوشه ای که گزارش‌ها روی سرور در آن قرار خواهند گرفت، می‌باشد. وقت کنید که باید حتماً پوشه موجود باشد. اگر به حالت پیشفرش رها کنید، گزارشات در قرار خواهند گرفت.



روی OK کلیک کنید و پروژه را Deploy کنید.

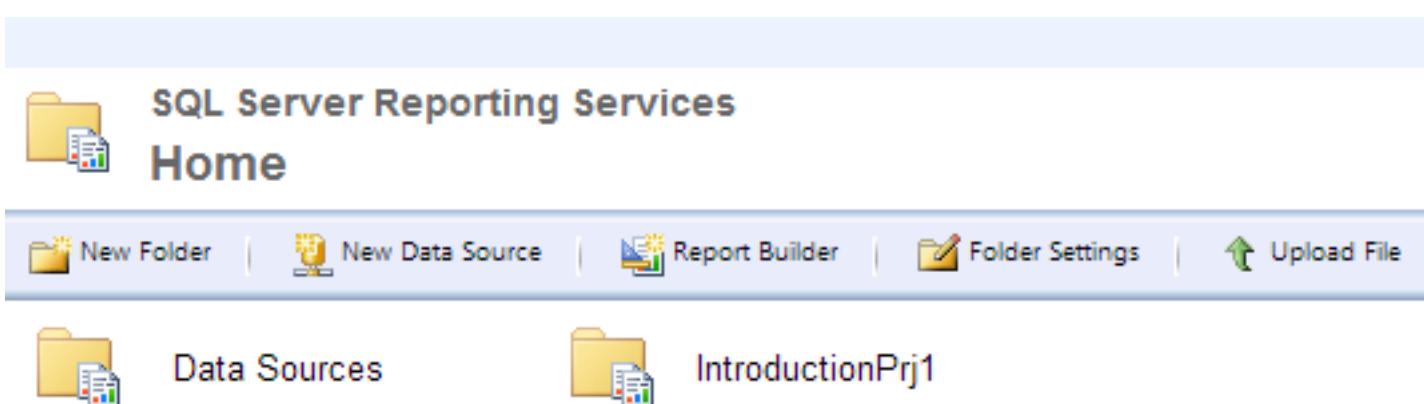
Output

Show output from: Build

```
----- Build started: Project: IntroductionPrj1, Configuration: Debug -----
Skipping 'DataSource_AdventureWorks.rds'. Item is up to date.
Skipping 'MyChildReport.rdl'. Item is up to date.
Skipping 'MyParentReport.rdl'. Item is up to date.
Build complete -- 0 errors, 0 warnings
----- Deploy started: Project: IntroductionPrj1, Configuration: Debug -----
Deploying to http://mbs-lt/ssrs\_srv
Deploying data source '/Data Sources/DataSource_AdventureWorks'.
Deploying report '/IntroductionPrj1/MyChildReport'.
Deploying report '/IntroductionPrj1/MyParentReport'.
Deploy complete -- 0 errors, 0 warnings
===== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped =====
===== Deploy: 1 succeeded, 0 failed, 0 skipped =====
```



اگر به سایت برگردید ، گزارشات را می‌توانید مشاهده کنید



Home > IntroductionPrj1 > MyParentReport

1 of 1 100% Find

MyParentReport

Department Name	Gender	COUNT
Document Control	F	1
Document Control	M	4
Engineering	F	3
Engineering	M	مشاهده جزئیات
Executive	F	2
Executive	M	1
Facilities and Maintenance	F	2
Facilities and	M	5

Home > IntroductionPrj1 > MyChildReport

1 of 1 100% Find | Next

MyChildReport

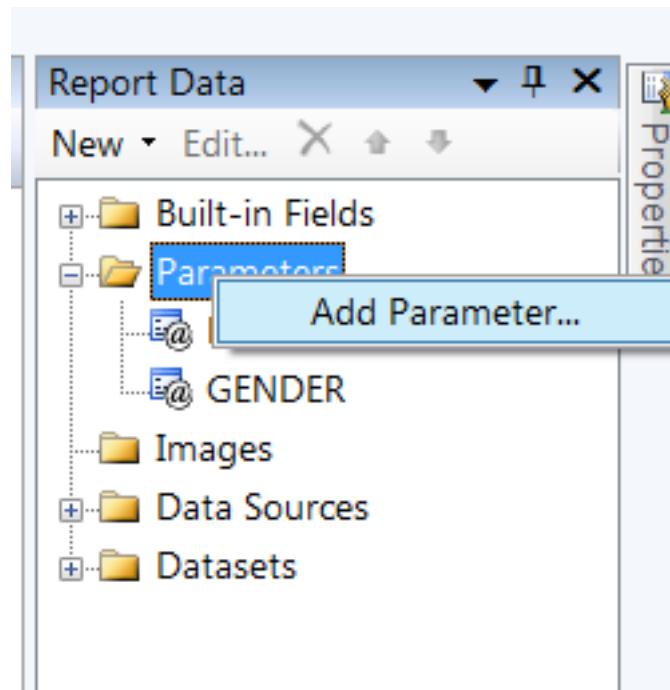
M

Employee Key	First Name	Last Name	Middle Name	Title	Hire Date	Birth Date	Email Address
56	Tengiz	Kharatishvili	N	Control Specialist	1/17/1999 12:00:00 AM	5/29/1980 12:00:00 AM	tengiz0@nature-works.com
92	Zainal	Arifin	T	Document Control Manager	2/5/1999 12:00:00 AM	3/2/1966 12:00:00 AM	zainal0@nature-works.com
130	Sean	Chai	N	Document Control Assistant	2/23/1999 12:00:00 AM	4/12/1977 12:00:00 AM	sean1@nature-works.com
216	Chris	Norred	K	Control Specialist	4/7/1999 12:00:00 AM	6/26/1977 12:00:00 AM	chris1@nature-works.com

بخش سوم :

در این مرحله می خواهیم یکی از ویژگی هایی که در گزارش گیری کاربرد زیادی دارد ، یعنی نمایش ستون های دلخواه در گزارش را به کمک SSRS و BIDS به کار ببریم.

برای این منظور به پنجره Report Data مراجعه کرده و روی Parameters کلی راست کرده و گزینه Add Parameter را انتخاب کنید



فیلد ها را مانند زیر پر کنید : (در اینجا می خواهیم یک dropdown به کاربر نشان داده شود تا انتخاب کند که ستون نمایش داده شود یا خیر . همچنین مقدار پیشفرض بله است)

Report Parameter Properties

General

Available Values

Default Values

Advanced

Change name, data type, and ot

Name:

ShowBirthDate

Prompt:

آیا ستون تاریخ تولد تعایش داده شود؟

Data type:

Text

Allow blank value ("")

Allow null value

Allow multiple values

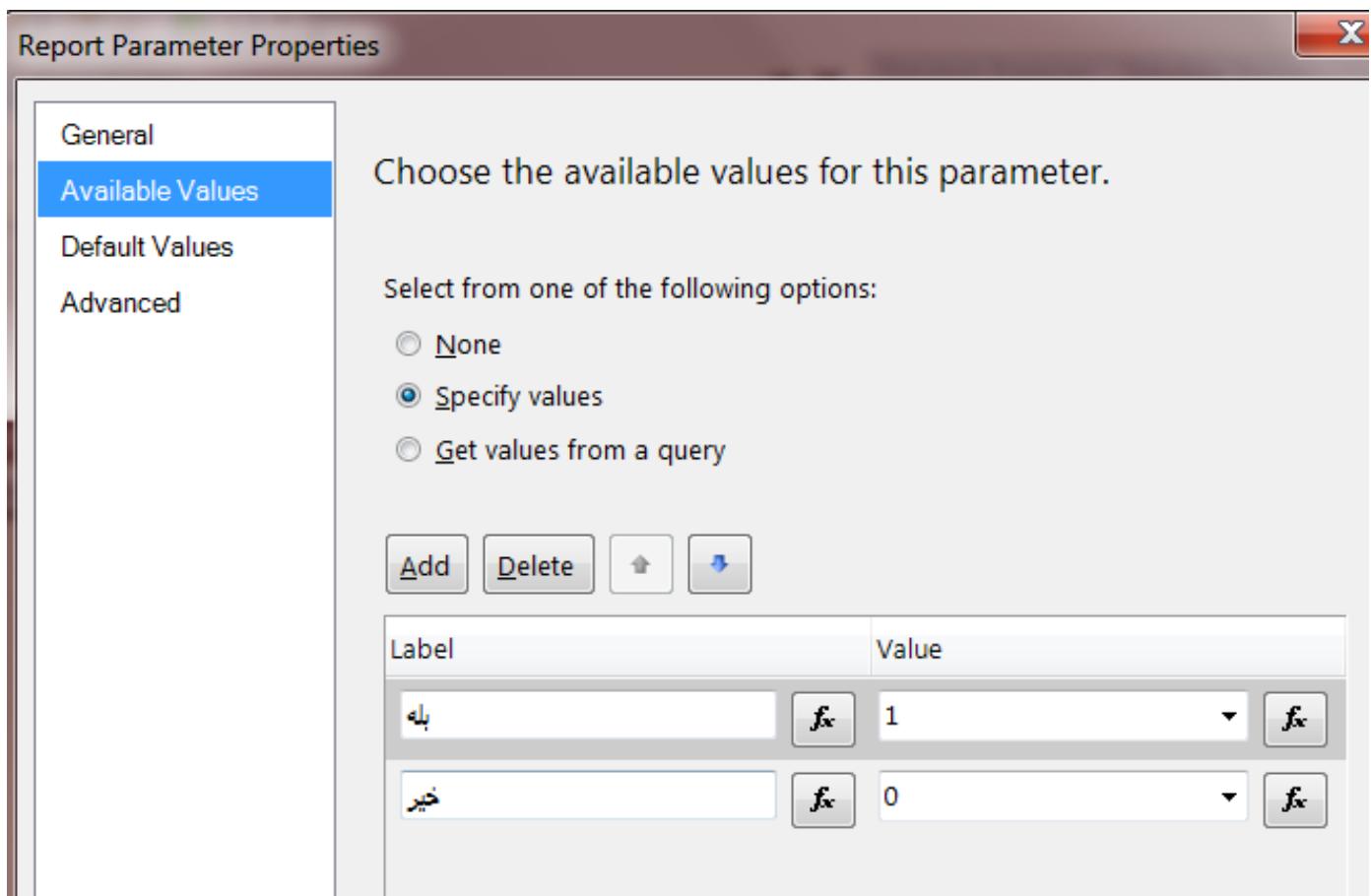
Select parameter visibility:

Visible

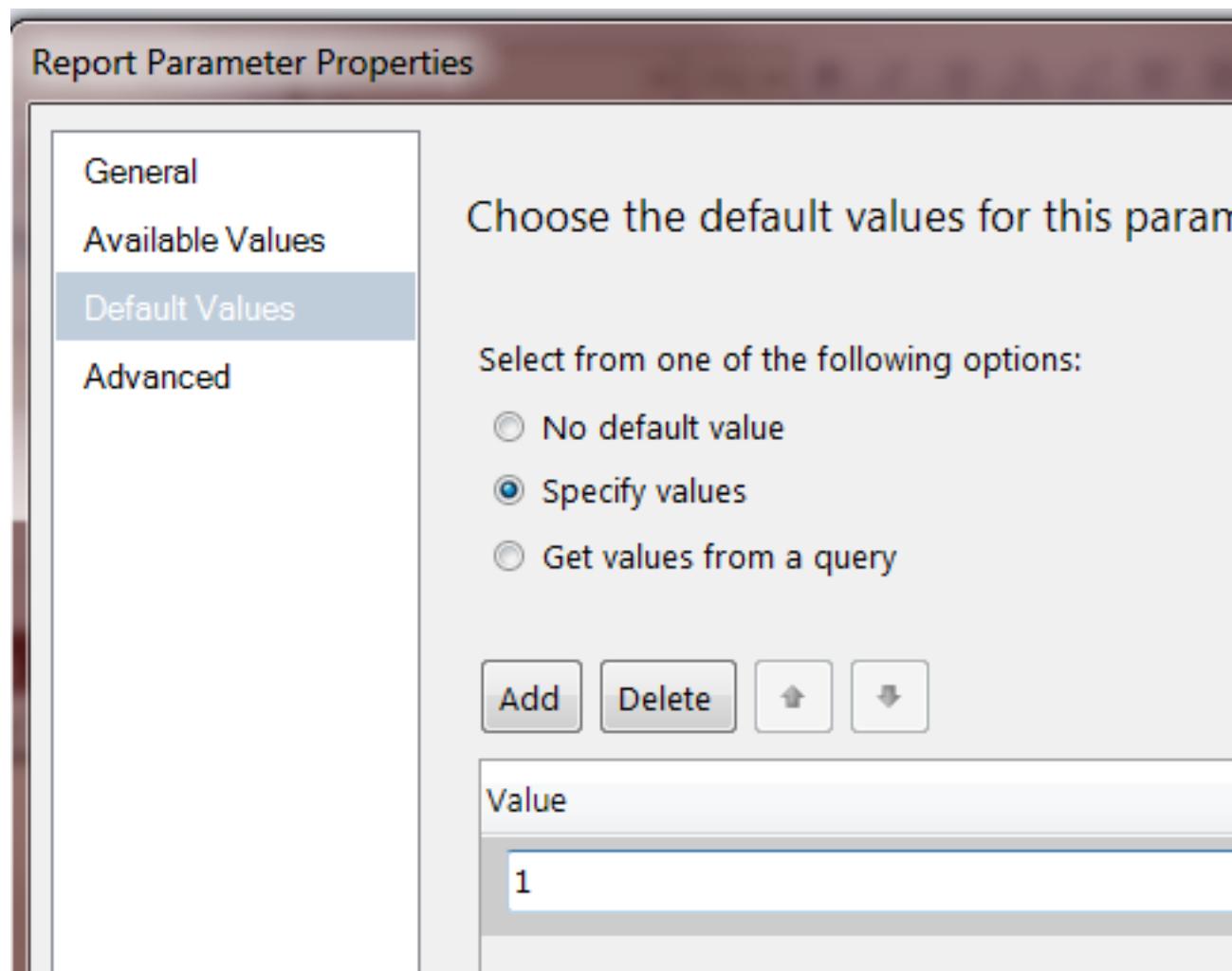
Hidden

Internal

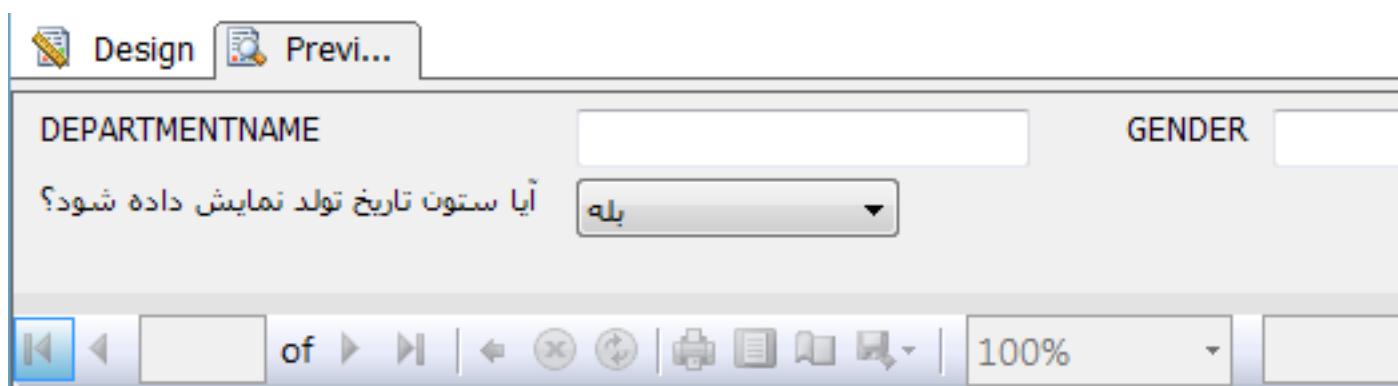
گزینه‌های مورد نظر را انتخاب نمایید



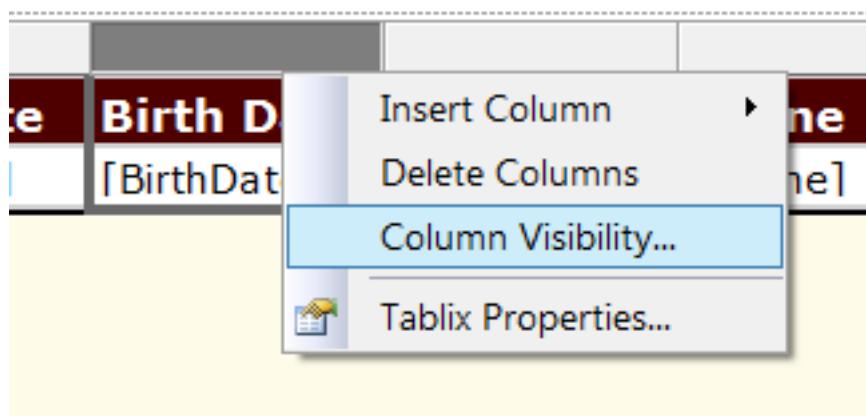
و تعیین مقدار پیش فرض



: نتیجه در BIDS



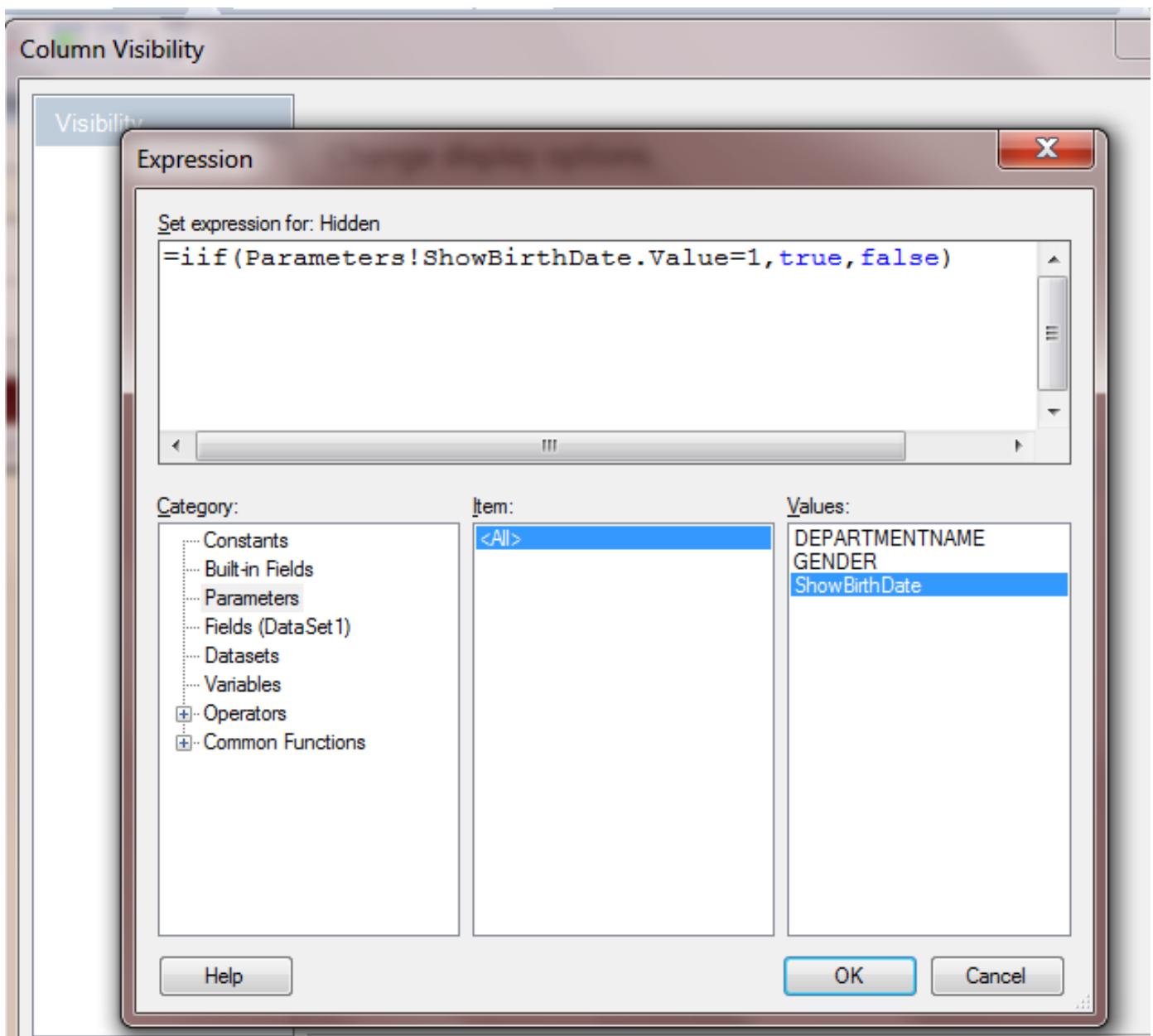
اکنون باید تغییر مقدار این ستون را بر روی گزارش اعمال کنیم :
برای همین منظور روی ستون BirthDate در حالت Design کلیک راست کرده و گزینه Column Visibility را انتخاب کنید :



سپس باید تنظیم کنیم که در نمایش و عدم نمایش این ستون باید بر اساس یک عبارت یا expression باشد .

A screenshot of the 'Column Visibility' dialog box. The title bar says 'Column Visibility'. The left sidebar has a 'Visibility' tab selected. The main area says 'Change display options.' and 'When the report is initially run:'. There are three radio buttons: 'Show' (unchecked), 'Hide' (unchecked), and 'Show or hide based on an expression' (checked). Below this is an input field for the expression, and to its right is a small 'fx' button. A red rectangular box highlights the 'fx' button.

عبارت زیر را وارد کنید (به این معنی که اگر مقدار 1 بود نمایش داده شود در غیر این صورت نمایش داده نشود) برای اطلاعات بیشتر در مورد دستورات Expression به [اینجا](#) و [اینجا](#) و [اینجا](#) مراجعه کنید
`(iif(Parameters!ShowBirthDate.Value=1,true,false=`



اکنون میتوانید گزارش را deploy کنید و تنظیمات سطوح دسترسی کاربران را انجام دهید

[موفق باشید](#)

نظرات خوانندگان

نویسنده: محمد دلیری
تاریخ: ۱۴۰۶/۰۶/۱۵

سلام ممنونم از مطلبتون جالب بود ! . چجور میشه این نمونه گزارش گیری رو در Windows Application استفاده کرد؟

نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۳۹۲/۰۶/۱۶

با استفاده از SOAP API های SSRS می‌توانید (+ و + و +) و در بخش Reporting موجود در VS Toolbox کنترل Report Viewer بهتون کمک می‌کنه. موفق باشید

نویسنده: bahman
تاریخ: ۱۳۹۲/۰۶/۱۹

سلام؛ مقاله خوبی بود. فقط من یه سوال دارم. چطوری از پروسیجر ای استفاده کنیم که پرامتر میگیره. ممنون

نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۳۹۲/۰۶/۱۹

سلام
باید از Dataset استفاده کنید (+ و + و +)

نویسنده: arefeh
تاریخ: ۱۳۹۳/۰۱/۱۸

سلام
ممنون از آموزش شما .
مراحل گفته شده را اجرا کردم . زمانی که پروژه را deploy می‌کنم خطای زیر را می‌دهد
could not connect to the report server .verify that the target server url is valid and that you have the correct
permissions to connect to the report server

نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۳۹۳/۰۱/۱۸

سلام
از صحت آدرس‌های تعریف شده اطمینان پیدا کنید.

در قسمت Reporting Services Configuration Manager در زیر مجموعه‌های URL و Report Manager URL و همچنین

در قسمت TargetServerRUL قسمت Properties احتمال دوم، عدم وجود دسترسی کافی برای تنظیمات فوق است با کاربر مدیر ارشد این کار را انجام دهید

نویسنده: علیرضا
تاریخ: ۲۱:۴۳ ۱۳۹۳/۰ ۱/۲۰

مهمترین مشکل ابزارهای قدرتمند گزارشگیری اینه که اکثرا فقط با دستورات SQL کار می‌کنند. برای اینکه SSRS با مجموعه‌ای از موجودیتهای دامنه Domain Entities رو به Entity Framework میشه متصل کرد؟

نویسنده: محمد باقر سیف اللهی
تاریخ: ۲۲:۴۰ ۱۳۹۳/۰ ۱/۲۰

به هیچ وجه ، انتباق خوب یک ابزار با یک پایگاه داده از طریق SQL یک اشکال محسوب نمیشه ! منظورتون از برقراری ارتباط بین EF و SSRS اینه که data source انتخابی برای SSRS را EF انتخاب کنیم ؟ اگر بله ؛ دلیله این کار چیه که از یک سو EF به SQL وصل بشه و از طرفی هم به ...SSRS... مستقیم این ارتباط رو برقرار می‌کنیم... (یعنی SSRS به SQL و با همان دستورات بدون نیاز به وجود واسطه ای به نام EF) هر ابزاری را با هدف خاصی پیاده سازی می‌کنند و اصلا دلیلی نداره که انتظار برقراری از طریق EF با SSRS وجود داشته باشه ! EF برای یک مقیاس و کاربرد استفاده میشه و SSRS برای یک مقیاس و کاربرد دیگر .

و اگر نه ، ممکنه بیشتر توضیح بدید؟

نویسنده: علیرضا
تاریخ: ۲۱:۰ ۱۳۹۳/۰ ۱/۲۳

راهی وجود داره که گزارش تهیه کنه؟ Domain Object از SSRS ها

نویسنده: وحید نصیری
تاریخ: ۲۱:۴۲ ۱۳۹۳/۰ ۱/۲۳

EF -> WCF Data Service -> SSRS via XML -
[Data Processing Extension](#) -
[Using SSRS In ASP.NET MVC Application](#) -
[Microsoft.Reporting.WebForms.ReportDataSource](#) -

در یک برنامه فروشگاه، جداول مشتری و خریدهای او را در نظر بگیرید. خرید ۳ سال قبل مشتری خاصی به آدرس قبلی او ارسال شده است. خرید امروز او به آدرس جدید او ارسال خواهد شد. سؤال: آیا با وارد کردن و به روز رسانی آدرس جدید مشتری، باید سابقه اطلاعاتی قبلی او حذف شود؟ اجناس ارسالی پیشین او، واقعاً به آدرس دیگری ارسال شده‌اند و نه به آدرس جدید او. چگونه باید اینگونه اطلاعاتی را که در طول زمان تغییر می‌کنند، در بانک‌های اطلاعاتی رابطه‌ای نرمال شده مدیریت کرد؟ از این نمونه‌ها در دنیای کاری واقعی بسیارند. برای مثال قیمت اجناس نیز چنین وضعی را دارند. یک بسته مگنوم، سال قبل ۳۰۰ تومان بود؛ امسال شده است ۱۵۰۰ تومان. یک سطل ماست ۲۵۰۰ تومان بود؛ امروز همان سطل ماست ۶۵۰۰ تومان است. چطور باید سابقه فروش این اجناس را نگهداری کرد؟

منابع مطالعاتی مرتبط

[این موضوع](#) اینقدر مهم است که تابحال چندین کتاب در مورد آن تالیف شده است:

- [Temporal Data & the Relational Model](#)
- [Trees and Hierarchies in SQL](#)
- [Developing Time-Oriented Database Applications in SQL](#)
- [Temporal Data: Time and Relational Databases](#)
- [Temporal Database Entries for the Springer Encyclopedia of Database Systems](#)
- [Temporal Database Management](#)

نکته مهمی که در این مأخذ وجود دارند، واژه کلیدی «[Temporal data](#)» است که می‌تواند در جستجوهای اینترنتی بسیار مفید واقع شود.

بررسی ابعاد زمان

فرض کنید کارمندی را استخدام کرده‌اید که ساعتی ۲۰۰۰ تومان از ابتدای فروردین ماه حقوق دریافت می‌کند. حقوق این شخص از ابتدای مهرماه قرار است به ساعتی ۲۴۰۰ تومان افزایش یابد. اگر مامور مالیات در بهمن ماه در مورد حقوق این شخص سؤال پرسید، ما چه پاسخی را باید ارائه دهیم؟ قطعاً در بهمن ماه عنوان می‌کنیم که حقوقش ساعتی ۲۴۰۰ تومان است؛ اما واقعیت این است که این عدد از ابتدای استخدام او ثابت نبوده است و باید تاریخچه تغییرات آن، در نحوه محاسبه مالیات سال جاری لحاظ شود.

بنابراین در مدل سازی این سیستم به دو زمان نیاز داریم:
 الف) actual time یا زمان رخ دادن واقعه‌ای. برای مثال حقوق شخصی در تاریخ ابتدای مهر ماه تغییر کرده است. به این تاریخ در منابع مختلف Valid time نیز گفته می‌شود.
 ب) record time یا زمان ثبت یک واقعه؛ مثلاً زمان پرداخت حقوق. به آن Transaction time هم گفته شده است.
 یک مثال:

record date	actual date	حقوق دریافتی
۱۳۹۲/۰۱/۰۱	۱۳۹۲/۰۱/۰۱	۲۰۰۰ روز/۰۱
۱۳۹۲/۰۲/۰۱	۱۳۹۲/۰۱/۰۱	۲۰۰۰ روز/۰۱
...		
۱۳۹۲/۰۷/۰۱	۱۳۹۲/۰۷/۰۱	۲۴۰۰ روز/۰۱
...		
۱۳۹۲/۱۷/۰۱	۱۳۹۲/۰۷/۰۱	۲۴۰۰ روز/۰۱

در این لیست، ریز حقوق پرداختی به یک شخص را ملاحظه می‌کنید. actual date، زمان‌هایی هستند که حقوق پایه شخص در

آن‌ها تغییر کرده و record date زمان‌هایی هستند که به شخص حقوق داده شده است. به ترکیب Transaction Time و Valid Time بitemporal data اصطلاحاً می‌گویند.

مشکلات طراحی‌های متداول اطلاعات وابسته به زمان

در طراحی‌های متداول، عموماً یک جدول کارمندان وجود دارد و یک جدول لیست حقوق‌های پرداختی. رکوردهای لیست حقوق‌های پرداختی نیز توسط یک کلید خارجی به اطلاعات هر کارمند متصل است؛ از این جهت که نمی‌خواهیم اطلاعاتی تکراری را در جدول لیست حقوقی ثبت کنیم و طراحی نرم‌افزاری شده‌ای مدنظر می‌باشد.

خوب؛ اول مهرماه حقوق شخصی تغییر کرده است. بنابراین کارمند بخش مالی اطلاعات شخص را به روز می‌کند. با این کار، کل سابقه حقوق‌های پرداختی شخص نیز از بین خواهد رفت. چون وجود این کلید خارجی به معنای استفاده از آخرین اطلاعات به روز شده یک کارمند در جدول لیست حقوقی است. الان اگر از جدول لیست حقوقی گزارش بگیریم، کارمندان همواره از آخرین حقوق به روز شده خودشان استفاده خواهند کرد.

راه حل‌های متفاوت مدل سازی اطلاعات وابسته به زمان
برای رفع این مشکل مهم، راه حل‌های متفاوتی وجود دارند که در ادامه آن‌ها را بررسی خواهیم کرد.

الف) نگهداری اطلاعات وابسته به زمان در جداول نهایی مرتب

اگر حقوق پایه شخص در زمان‌های مختلف تغییر می‌کند، بهتر است عدد نهایی این حقوق پرداختی نیز در یک فیلد مشخص، در همان جدول لیست حقوقی ثبت شود. این مورد به معنای داشتن «داده‌ای تکراری» نیست. از این جهت که داده‌ای تکراری است که اطلاعات آن در تمام زمان‌ها، دارای یک مقدار و مفهوم باشد و اطلاعات حقوق یک شخص اینچنین نیست.

ب) نگهداری اطلاعات تغییرات حقوقی در یک جداول جداگانه

یک جدول ثانویه حقوق جاری کارمندان مرتب با جدول اصلی کارمندان باید ایجاد شود. در این جدول هر رکورد آن باید دارای بازه زمانی (valid_start_time و valid_end_time) مشخصی باشد. مثلاً از تاریخ X تا Tاریخ Z، حقوق کارمند شماره 11، 2000 تومان در ساعت بوده است. از Tاریخ H تا Tاریخ I اطلاعات دیگری ثبت خواهند شد. به این ترتیب با گزارشگیری از جدول لیست حقوق‌های پرداخت شده، سابقه گذشته اشخاص محو نشده و هر رکورد بر اساس قرارگیری در یک بازه زمانی ثبت شده در جدول ثانویه حقوق جاری کارمندان تفسیر می‌شود.

در این حالت باید دقت داشت که بازه‌های زمانی تعریف شده، با هم تداخل نکنند و برنامه ثبت کننده اطلاعات باید این مساله را به ازای هر کارمند کنترل کند و یا با ثبت record_date، اجازه ثبت بازه‌های تکراری را نیز بدهد (توضیحات در قسمت بعد). به این جدول، یک Temporal table نیز گفته می‌شود. نمونه دیگر آن، نگهداری قیمت یک کالا است از یک تاریخ تا تاریخی مشخص. به این ترتیب می‌توان کوئری گرفت که بستنی مگنوم فروخته شده در ماه آبان سال قبل، بر مبنای قیمت آن زمان، دقیقاً چقدر فروش کرده است و نه اینکه صرفاً بر اساس آخرین قیمت روز این کالا گزارشگیری کنیم که در این حالت اطلاعات نهایی استخراج شده صحیح نیستند.

حال اگر به این طراحی در جدولی دیگر Transaction time یا زمان ثبت یک رکورد یا زمان ثبت یک فروش را هم اضافه کنیم، به جداول حاصل Bitemporal Tables می‌گویند.

مدیریت به روز رسانی‌ها در جداول Temporal

در جداول Temporal، حذف فیزیکی اطلاعات مطلقاً ممنوع است؛ چون سابقه سیستم را تخریب می‌کند. اگر اطلاعاتی در این جداول دیگر معتبر نیست باید تنها تاریخ پایان دوره آن به روز شوند یا یک رکورد جدید بر اساس بازه‌های جدید ثبت گردد. همچنین به روز رسانی‌ها در این جداول نیز معادل هستند با یک Insert جدید به همراه فیلد record_date و نه به روز رسانی واقعی یک رکورد قبلی (نشیبه به سیستم‌های حسابداری باید عمل کرد).

یک مثال:

فرض کنید حقوق کارمندی که مثال زده شد، در مهرماه به ساعتی 2400 تومان افزایش یافته است و حقوق نهایی نیز پرداخته شده است. بعد از یک ماه مشخص می‌شود که مدیر عامل سیستم گفته بوده است که ساعتی 2500 تومان و نه ساعتی 2400 تومان! (از این نوع مسائل در دنیای واقعی زیاد رخ می‌دهند!) خوب؛ اکنون چه باید کرد؟ آیا باید رفت و رکورد ساعتی 2400 تومان را به روز کرد؟ خیر. چون سابقه پرداخت واقعی صورت گرفته را تخریب می‌کند. به روز رسانی شما ابداً به این معنا نخواهد بود که دریافتی

واقعی شخص در آن تاریخ خاص، ساعتی 2500 بوده است.

بنابراین در جداول Temporal، تنها «تغییرات افزودنی» مجاز هستند و این تغییرات همواره به عنوان آخرین رکورد جدول ثبت می‌شوند. به این ترتیب می‌توان اصطلاحاً «مابه التفاوت» حقوق پرداخت نشده را به شخص خاصی، محاسبه و پرداخت کرد (می‌دانیم در یک بازه زمانی خاص به او چند حقوق داده‌ایم. همچنین می‌دانیم که این بازه در یک record_date دیگر لغو و با عددی دیگر، جایگزین شده‌است).

برای مطالعه بیشتر

[Bitemporal Database Table Design - The Basics](#)

[Temporal Data Techniques in SQL](#)

[Database Design: A Point in Time Architecture](#)

[Temporal database](#)

[Temporal Patterns](#)

راه حلی دیگر؛ استفاده از بانک‌های اطلاعاتی NoSQL

بانک‌های اطلاعاتی NoSQL برخلاف بانک‌های اطلاعاتی رابطه‌ای برای اعمال Read بهینه سازی می‌شوند و نه برای Write. در چند دهه قبل که بانک‌های اطلاعاتی رابطه‌ای پدیدار شدند، یک سخت دیسک 10 مگابایتی حدود 4000 دلار قیمت داشته است. به همین جهت مباحثت نرم‌افزاری اطلاعات و ذخیره نکردن اطلاعات تکراری تا این حد در این نوع بانک‌های اطلاعاتی مهم بوده است. اما در بانک‌های اطلاعاتی NoSQL، اگر قرار است فیش حقوقی شخصی ثبت شود، می‌توان کل اطلاعات جاری او را یکجا داخل یک سند ثبت کرد (از اطلاعات شخص در آن تاریخ تا اطلاعات تمام اجزای فیش حقوقی در قالب یک شیء تو در توی JSON). به همین جهت بسیار سریع هستند برای اعمال Read و گزارشگیری. همچنین این نوع سیستم‌ها برای نگهداری نگارش‌های مختلف یک سند بهینه سازی شده‌اند و جزو ساختار توکار آن‌ها است. بنابراین در این نوع سیستم‌ها اگر نیاز است از یک سند خاصی گزارش بگیریم، دقیقاً اطلاعات همان تاریخ خاص را دارا است و اگر اطلاعات پایه سیستم را به روز کنیم، از امروز به بعد در سندهای جدید ثبت خواهد شد. این نوع سیستم‌ها رابطه‌ای نیستند و بسیاری از مباحثت نرم‌افزاری اطلاعات در آن‌ها ضرورتی ندارد. قرار است یک فیش حقوقی شخص را نمایش دهیم؟ خوب، چرا تمام اطلاعات مورد نیاز او را در قالب یک شیء JSON تو در توی حاضر و آماده نداشته باشیم؟

نظرات خوانندگان

نویسنده: ناصر
تاریخ: ۱۳۹۲/۰۷/۱۷ ۰:۵۴

من هم قبل از این روش برای قیمت‌های جدید و قدیم یک کالا در یک سیستم فروشگاهی استفاده کردم. با نوشتن یک تریگر که به محض تغییر روی قیمت کالا، بلاfaciale داخل یک جدول دیگه این تغییرات درج می‌شود. و موقع نمایش، قیمت جدید و قدیم هر دو با هم به مشتری نمایش داده می‌شوند.

نویسنده: سیروس
تاریخ: ۱۳۹۲/۰۷/۱۸ ۱۱:۱۱

به نظر من در نگهداری به این روش نیازی به تاریخ پایان نیست، مثلا هنگام تغییر قیمت کالا، رکوردی با تاریخ روز در جدول **temporal** ثبت می‌کنیم و در تغییر دوباره رکورد جدید دیگری ثبت می‌شود. کارکردن به این روش آسانتر به نظر می‌رسد و یک فیلد کمتر داریم و نیازی هم به چک کردن درست بودن بازه‌ی تاریخی نیست.

Date	Price	ProductID
1392/01/01	1000	1
1392/03/05	1500	1
1392/06/27	1780	1

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۳۹۳/۰۴/۰۲ ۱۱:۴۵

البته روش شما برای حالتی مناسب است که بازه‌های تاریخی به هم متصل باشند.

مقدمه

نصب Microsoft Sql Server یکی از عملیات مشکل برای کاربر نهایی می‌باشد. برای رفع این مشکل، باید آنرا بصورت اتوماتیک و بدون درگیر کردن کاربر آن را نصب کنیم. برای اینکار دو روش موجود است:

- (1) استفاده از فایل **Configure.ini**
- (2) ارسال پارامتر به فایل **Setup.exe** از طریق **Command Prompt**

قابلیت‌های نسخه‌های مختلف SQL Server Express 2008

Reporting Services	Full-Text Search	Management Studio Basic	Database Engine	نسخه / قابلیت
		X		Management Studio Basic
			X	Runtime Only
		X	X	with Tools
X	X	X	X	with Advanced Services

SQL Server 2008 Management Studio Express (SSMSE)

ابزارهای مدیریتی گرافیکی رایگان برای پیکربندی، مدیریت و اداره کردن برنامه‌های 2008 SQL Server Express چندین نمونه از موتور پایگاه داده SQL Server که توسط نسخه‌های مختلف SQL Server 2008 ساخته شده اند.

SQL Server 2008 Express (Runtime Only)

موتور پایگاه داده SQL Server برای ساخت، ذخیره سازی، بروز رسانی و واکشی داده

SQL Server 2008 Express with Tools

موتور پایگاه داده SQL Server برای ساخت، ذخیره سازی، بروز رسانی و واکشی داده - یک ابزار مدیریتی ویژوال برای ساخت، ویرایش و مدیریت پایگاه‌های داده SQL Server Management Studio Basic

SQL Server 2008 Express with Advanced Services

موتور پایگاه داده SQL Server برای ساخت، ذخیره سازی، بروز رسانی و واکشی داده - یک ابزار مدیریتی ویژوال برای ساخت، ویرایش و مدیریت پایگاه‌های داده SQL Server Management Studio Basic - یک موتور قدرتمند و پرسرعت برای جستجوی داده‌های متن-فشرده

دو حالت برای نصب SQL Server وجود دارد:

نصب یک نمونه جدید

آپگرید

نصب 2008 SQL Server Express از طریق Command Prompt

- 1) پس از [دانلود 2008 SQL Server Express](#) با پارامتر X/ انرا از حالت فشرده خارج کنید.

برای این کار Command Prompt (یا همان cmd) را باز کنید و با دستور cd به مسیری که فایل SQL Server Express هست بروید. حالا نام فایل نصب را همراه با پارامتر X/ در cmd تایپ کنید. مثلا: /x: SQLExpr32_x86_enu.exe

نکته: اگر فایل یا سی دی نصب ویژوال استودیو 2010 را دارید می‌توانید فایل نصب SQL Server Express را در مسیر \WCUSSE\WCU\SSE پیدا کنید.

2) همانطور که می‌بینید یک مسیر برای Extract از شما خواسته شده. یک مسیر وراد کنید.

3) در مسیر ساخته شده، یک فایل با پسوند bat. بسازید و آنرا یا یک ویرایشگر متنی باز کنید و دستورات زیر را در آن تایپ

کنید:

```
Setup.exe /q /Action=Install /Hideconsole /Features=SQL,Tools
/InstanceName=SQLEXpress /SQLSYSADMINACCOUNTS="Builtin\Administrators"
/SQLSVCACCOUNT=<DomainName\UserName> /SQLSVCPASSWORD=<StrongPassword>
```

توضیح پارامترهای فوق بشرح زیر است:

/q (اختیاری): اینکه Setup به حالت خاموش (quiet mode) و بدون رابط کاربری اجرا شود.

/Action (الزامی): عملیاتی که باید انجام شود. این پارامتر مقدار install و upgrade را قبول می‌کند.

/Features (الزامی): اینکه کدام قابلیت‌های SQL Server باید نصب شوند.

/HideConsole (اختیاری): اگر از این پارامتر استفاده شود، پنجره کنسول نمایش داده نخواهد شد.

/InstanceName (الزامی): نام نمونه ای که باید نصب شود.

/SQLSYSADMINACCOUNTS (الزامی): مقررات لوگین برای اعضای با نقش "مدیر سیستم".

/Startup (الزامی): تعیین اکانتی که سرویس SQL Server را در Windows ویندوز اجرا کند.

/SQLSVCACCOUNT (اگر از یک اکانت لوکال یا تحت Domain استفاده کنید الزامی است): تعیین پسورد اکانت پارامتر SQLSVCACCOUNT

باتوجه به سیاست‌ها نصب می‌توانید از پارامترها دیگری نیز استفاده کنید. بعنوان مثال پارامترهای زیر برای نصب روی سیستمی که نام کاربری و پسورد انرا نداریم مناسب است:

```
setup.exe /q /Action=Install /Features=SQL /InstanceName=SQLEXpress /SECURITYMODE=SQL /SAPWD="1234567"
/SQLSYSADMINACCOUNTS="Builtin\Administrators" /SQLSVCACCOUNT="NT AUTHORITY\SYSTEM"
/SQLSVCSTARTUPTYPE="Automatic" /TCPENABLED=1
```

نکته: اگر مقدار پارامتر SECURITYMODE برابر SQL باشد حتماً باید پارامتر SAPWD مقداردهی شود.

نکته: اگر مقدار TCPENABLED برابر 1 باشد پروتکل TCP/IP فعال می‌شود. اگر هیچ نمونه‌ی دیگری روی سیستم نصب نباشد مقدار TCP Port برابر 1433 است، در غیر اینصورت یک مقدار تصادفی تولید می‌شود.

نکته: برای خوانایی بیشتر، پارامترهای فوق در چند خط نوشته شده اند. برای اجرای صحیح در یک فایل bat، همه انها باید در یک خط باشند.

برای اولین بهتر است از پارامتر HideConsole استفاده نکنید تا خطای احتمالی را مشاهده کنید. برای آپگرید کردن نیز می‌توانید از دستور زیر استفاده کنید:

```
Setup.exe /q /Hideconsole /ACTION=upgrade /INSTANCENAME=SQLEXpress
```

برای مشاهده دیگر پارامترها به مستندات MSDN مراجعه کنید. همچنین می‌توان [نصب از طریق فایل Configuration](#) را نیز انجام داد.

امیدوارم مفید واقع شده باشد.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۹:۱ ۱۳۹۲/۰۷/۲۲

ممnon از شما. برای کارهای سبک استفاده از [SQL Server CE](#) شاید مناسب‌تر باشد یا حتی [LocalDB](#).

نویسنده: جوادی
تاریخ: ۰:۳ ۱۳۹۲/۰۷/۲۳

خواهش می‌کنم.

بله، بسته به سناریو می‌شود از پایگاه داده‌های سبک‌تر نیز استفاده کرد.

Sql server 2012 management studio متناسفانه محدودیت‌های زیادی دارد، مثلا در XP ساپورت نمی‌شود، فقط توسط LocalDB در دسترس هست که برای محصولات عمومی مناسب نیست. البته محل استفاده خاص خودش را دارد. باز هم همه چیز به نحوه و محل استفاده بستگی دارد.

نویسنده: افشار محبی
تاریخ: ۱۵:۴۳ ۱۳۹۲/۰۸/۰۸

شما که زحمت نوشتن این مقاله را کشیده اید کاش از نسخه ۲۰۱۲ استفاده می‌کردید.

نویسنده: محسن خان
تاریخ: ۱۷:۳ ۱۳۹۲/۰۸/۰۸

به نظر ایشون بیشتر نگارش‌هایی مدنظرشون هست که روی XP هم قابل اجرا باشند که [نگارش ۲۰۱۲ جزو آن‌ها نیست](#).

نویسنده: افشار محبی
تاریخ: ۱۹:۲ ۱۳۹۲/۰۸/۰۸

بله درسته. من اصلاً حواسم به پشتیبانی از xp و کامپیوترهای قدیمی نبود. ویندوز xp هم معضلی شده برای خودش. از یک طرف نمی‌توان پشتیبانی را قطع کرد، چون تعداد زیادی کامپیوتر در داخل کشور هنوز در همین حد هستند از طرف دیگر پشتیبانی از فناوری به این حد قدیمی چندان عاقلانه نیست.

البته یک راه دیگر هم هست. استفاده از db engine embeded معمولاً sqlite و شاید sql ce و بقیه.

نویسنده: جوادی
تاریخ: ۱۹:۵۹ ۱۳۹۲/۰۸/۰۸

من نسخه ۲۰۱۲ رو تست نکردم اما گمان نکنم تفاوتی داشته باشه؛ یعنی بعد از اکسترکت کردن با [استفاده از Command Prompt و ارسال پارامتر به اون](#) میشه یک Instance نصب کرد.

نویسنده: محسن خان
تاریخ: ۱۸:۵۵ ۱۳۹۲/۰۸/۱۳

یک مقایسه خوب هم در اینجا هست که می‌تونه به انتخاب نگارش مناسب کمک کنه:

[SQL Express v LocalDB v SQL Compact Edition](#)

نویسنده: یاسین
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۵:۲۶

سلام، ببینید من ورژن‌های متعددی که برای دانلود نسخه 2012 قرار داده شده رو به همراه حجمشون پایین نوشتم. به عنوان مبتدی می‌خواستم بدونم چه وقت از کدام یک از اینها استفاده می‌شده؟
یک حالتی هست که ویژوال استودیو از من می‌خواهد نسخه اکسپرس رو نصب کنم، خوب نسخه‌ای با حجم و امکانات بالا رو نصب می‌کنم. یک حالتی هم هست که بصورت توکار در یک برنامه قرار داده می‌شده که کاربر نهایی در اجرای برنامه به مشک برخوره، حالا چطور متوجه بشیم که کدام نسخه برای کاربر نهایی مناسبه.

تشکر

SQL Server Express with Tools (with LocalDB, Includes the database engine and SQL Server Management Studio Express) 1.4G

SQL Server Management Studio (Tools only) 939MB

SQL Server Express LocalDB (MSI installer) 24MB

SQL Server Express with Advanced Services (contains the database engine, Express Tools, Reporting Services, and Full Text Search) 1.7G

SQL Server Express (Containing only the database engine) 133MB

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۸:۵۴

نگارش‌های اکسپرس و local db رایگان هستند. نگارش CE یا Compact edition نیز رایگان است. سایر نگارش‌ها خیر. بنابراین خیلی‌ها در سراسر دنیا بر این اساس تصمیم گیری می‌کنند. نگارش CE داخل پروسه برنامه اجرا می‌شود و تا 4 گیگ دیتا داخلش می‌تواند ذخیره کنید. نگارش اکسپرس تا 12 گیگ محدودیت دارد. سایر نگارش‌ها محدودیتی ندارند. خلاصه باید به کسب و کار خودتون نگاه کنید. برنامه حسابداری هست که تعداد رکوردهاش تا آخر سال مثلاً می‌شود 10 هزار رکورد؟ نگارش CE برآش کافی هست.

نویسنده: یاسین
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۱:۱۳

ممnon، برای کاربران نهایی که از ویندوز XP استفاده می‌کنند، آیا نسخه 2008 مناسب هست یا اینکه تا نسخه 2005 باید تنزيل کرد؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۱:۱۶

نگارش 2008 با XP سازگار است. نگارش‌های پس از آن خیر.

نویسنده: جمال
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۸:۱۸

می‌شده لطفاً روشی را برای نصب خودکار بانک اطلاعاتی بهنگام نصب برنامه‌ی اصلی معرفی نمایید.
چون در روال عادی برنامه‌های دات نت این کار باید به صورت دستی انجام بشه.

نویسنده: شهریار
تاریخ: ۱۳۹۲/۱۲/۰۳ ۲۱:۴۱

بهترین کار برای نصب خودکار فایلهای مورد نیاز جهت اجرای برنامه‌های مبتنی بر دات نت، استفاده از نرم افزارهای ساخت

میباشد. از قبیل Setup

InstallShield 1-

InstallAware 2-

Advanced Installer 3-

مورد اول ، از قدیمی‌ترین نرم افزارها و حرفه ای میباشد. ولی متأسفانه در تهیه نسخه به روز آن کمی دچار مشکل خواهد شد.

مورد دوم ادعای مقایسه با اینستال شیلد را دارد. از عیوبی که من در استفاده از این نرم افزار دیدم میتوان به حجم بالای نرم افزار اصلی اشاره کرد که بیش از دو گیگابایت است و هر دفعه نسخه جدید اوmd شما باید مجدداً این حجم را دانلود کنید.

مورد سوم که بهترین گزینه نیز میباشد ، بسیار خوش دست و سبک میباشد. و به راحتی تمام موارد مورد نیاز جهت اجرای برنامه‌های شما را نصب مینماید.

استفاده شده از 2008 SQL

- روش کار : ۱- دریافت پارامتر ورودی به صورت رشته
- ۲- درج عناوین اعداد، ارزش مکانی اعداد صحیح و اعشاری هر کدام در یک جدول
- ۳- جدا کردن ارقام صحیح و اعشاری
- ۴- جدا کردن سه رقم سه رقم اعداد صحیح و انتقال آنها به جدول مربوطه
- ۵- Join جداول عناوین و ارقام جدا شده
- ۶- ارسال ارقام اعشاری به همین تابع
- ۷- مشخص کردن ارزش مکانی رقم اعشار
- ۸- اتصال رشته حروف صحیح و اعشاری

در آخر این مطلب کد این تابع را به صورت کامل، برای دانلود قرار داده ام.

بررسی قسمت‌های مختلف کد

برای اینکه محدودیتی در تعداد ارقام صحیح و اعشاری نداشته باشیم، پارامتر ورودی را از نوع VARCHAR می‌گیریم. پس باید ورودی را بررسی کنیم تا رشته عددی باشد.

بررسی رشته ورودی:

```
-- @pNumber پارامتر ورودی
IF LEN(ISNULL(@pNumber, '')) = 0 RETURN NULL
IF (PATINDEX('%[^0-9.-]%', @pNumber) > 0)
    OR (LEN(@pNumber) -LEN(REPLACE(@pNumber, '.', '')) > 1)
    OR (LEN(@pNumber) -LEN(REPLACE(@pNumber, '.', '')) > 1)
    OR (CHARINDEX('.', @pNumber) > 1)
RETURN 'خطا'
IF PATINDEX('%[^0]%', @pNumber) = 0 RETURN 'صفر'
IF (CHARINDEX('.', @pNumber) = 1) SET @pNumber='0'+@pNumber
DECLARE @Negative AS VARCHAR(5) = '';
IF LEFT(@pNumber, 1) = '-'
BEGIN
    SET @pNumber = SUBSTRING(@pNumber, 2, 100)
    SET @Negative = 'منفی'
END
```

- بررسی NULL، خالی بودن و یا داشتن فاصله در رشته، با دانستن اینکه تابع LEN فاصله‌های آخر یک رشته را در نظر نمی‌گیرد.
- بررسی رشته ورودی برای پیدا کردن کاراکتر غیر عددی، نقطه و منفی. بررسی تعداد علامت منفی و نقطه که بیشتر از یک مورد نباشند، و در نهایت بررسی اینکه علامت منفی در ابتدای رشته ورودی باشد.
- بررسی صفر بودن ورودی(0)، مقدار ورودی شروع شونده با ممیز(30.0213) و مقدار عددی منفی(-21210.0021).
- چیز دیگری به ذهنم نرسید!

درج عناوین در جداول مربوطه:

فکر کنم اینجا به علت وجود کاراکترهای فارسی و انگلیسی کد کمی بهم ریخته نمایش داده می‌شود.

```
DECLARE @NumberTitle TABLE (val INT, Title NVARCHAR(100));
```

بنابراین براساس ممیز، قسمت صحیح را از اعشاری جدا می‌کنیم. برای ورودی که با ممیز شروع شود، در ابتدا تابع بررسی مم، کنیم و عدد صفر را به رشتہ اضافه می‌کنیم.

بعد از ممیز و اعداد بزرگتر از یک، با صفرهای بی ارزش چه کنیم؟ شاید اولین چیزی که به ذهن برسد استفاده از حلقه (WHILE) برای حذف صفرهای بی ارزش باشد؛ ولی من ترجیح می‌دهم که از روش دیگری استفاده کنم:

بر عکس کردن رشته قسمت اعشاری، پیدا کردن مکان اولین عدد غیر صفر منهای یک، و کم کردن عدد بدست آمده از طول رشته اعشاری، قسمت مورد نظر ما را برخواهد گرداند:

```
SUBSTRING(@DecimalNumber,1, len(@DecimalNumber )-PATINDEX('%[^0]%', REVERSE (@DecimalNumber))-1)
```

اما اگر عدد ورودی **20.0** باشد همچنان صفر بی ارزش بعداز ممیز را خواهیم داشت. برای رفع این مشکل کافی است که کاراکتری غیر از صفر را به اول رشته اعشاری اضافه کنیم. من از علامت **'?' استفاده کرم. پس به علت اضافه کردن کاراکتر، استارت را از 2 شروع کرده و دیگر نیازی به 1- خواهیم داشت. با کد زیر قسمت صحیح و اعشاری را بدست می آوریم:**

```
DECLARE @IntegerNumber NVARCHAR(100),
@DecimalNumber NVARCHAR(100),
@PointPosition INT =case CHARINDEX('. ', @pNumber) WHEN 0 THEN LEN(@pNumber)+1 ELSE CHARINDEX('. ',
@pNumber) END

SET @IntegerNumber= LEFT(@pNumber, @PointPosition - 1)
SET @DecimalNumber= '?' + SUBSTRING(@pNumber, @PointPosition + 1, LEN(@pNumber))
SET @DecimalNumber= SUBSTRING(@DecimalNumber,2, len(@DecimalNumber )-PATINDEX('%[^0]%', REVERSE
(@DecimalNumber)))

SET @pNumber= @IntegerNumber
```

جداد کردن سه رقم سه رقم: - بدست آوردن یکان، دهگان و صدگان

- برای قسمت دهگان، اگر عددی بین 10 تا 19 باشد به صورت کامل (مثلاً 15) و در غیر این صورت فقط رقم دهگان. برای بدست آوردن یکان اگر دو رقم آخر بین 10 و 19 بود صفر و در غیر این صورت یکان برگردانده می‌شود و در جدول MyNumbers درج می‌گردد.

```

DECLARE @Number AS INT
DECLARE @MyNumbers TABLE (id INT IDENTITY(1, 1), Val1 INT, Val2 INT, Val3 INT)

WHILE (@pNumber) <> '0'
BEGIN
    SET @number = CAST(SUBSTRING(@pNumber, LEN(@pNumber) -2, 3)AS INT)

    INSERT INTO @MyNumbers
    SELECT (@Number % 1000) -(@Number % 100),
    CASE
    WHEN @Number % 100 BETWEEN 10 AND 19 THEN @Number % 100
    ELSE (@Number % 100) -(@Number % 10)
    END,
    CASE
    WHEN @Number % 100 BETWEEN 10 AND 19 THEN 0
    ELSE @Number % 10
    END

    IF LEN(@pNumber) > 2
        SET @pNumber = LEFT(@pNumber, LEN(@pNumber) -3)
    ELSE
        SET @pNumber = '0'
END

```

سطری که تمام مقادیر آن صفر باشد برای ما بی ارزش محسوب می شود، مانند سطر یک در عکس زیر (جدول MyNumbers) برای عدد **1200955000** :

	id	Val1	Val2	Val3
1	1	0	0	0
2	2	900	50	5
3	3	200	0	0
4	4	0	0	1

استفاده از JOIN : JOIN کردن جدول اعداد با عنوانی عددی براساس ارزش آنها و JOIN جدول اعداد با جدول ارزش مکانی براساس ID به صورت نزولی(شماره سطر).

```

DECLARE @Str AS NVARCHAR(2000) = '';
SELECT @Str += REPLACE(REPLACE(LTRIM(RTRIM(nt1.Title + ' ' + nt2.Title + ' ' + nt3.title)), ' ', ','), ',',
' ،')
FROM @MyNumbers AS mn
INNER JOIN @PositionTitle pt
    ON pt.id = mn.id
INNER JOIN @NumberTitle nt1
    ON nt1.val = mn.Val1
INNER JOIN @NumberTitle nt2
    ON nt2.val = mn.Val2
INNER JOIN @NumberTitle nt3
    ON nt3.val = mn.Val3
WHERE (nt1.val + nt2.val + nt3.val > 0)
ORDER BY pt.id DESC

```

Replace داخلی: جایگزین کردن "دو فاصله‌ی خالی" با "یک فاصله‌ی خالی"
Replace بیرونی: جایگزینی فاصله‌های خالی با ' و '

همانطور که در بالا اشاره کردم سطرهایی که val1, val2, val3 آن صفر باشد برای ما بی ارزش هستند، پس آنها را با شرط نوشته شده حذف می‌کنیم.

بدست آوردن مقدار اعشاری: خوب! حالا نوبت به عدد اعشاری می‌رسد. برای بدست آوردن حروف، مقدار اعشاری بدست آمده

را به همین تابع ارسال می‌کنیم و برای بدست آوردن عنوان ارزش مکانی، براساس طول اعشار (ID) آن را در جدول مربوطه پیدا می‌کنیم.

اگر عدد ورودی مثلا 0.355 باشد، تابع باید صفر اول را شناسایی و قسمت عناوین اعشاری را به آن اضافه کند، که این کار با شرط ذیل انجام می‌شود.

اگر رشته اعشار بدون مقدار NULL تابع مقدار NULL بر می‌گرداند (قسمت بررسی رشته ورودی) و هر رشته‌ای که با NULL جمع شود برابر با NULL خواهد بود. در این صورت با توجه به کد زیر مقداری به رشته Str به عنوان قسمت اعشاری، اضافه نمی‌گردد.

```

IF @IntegerNumber='0'
SET @Str=CASE WHEN PATINDEX('%[^0]%', @DecimalNumber) > 0 THEN @Negative ELSE '' END + 'صفر'
ELSE
SET @Str = @Negative + LEFT (@Str, LEN(@Str) -2)

DECLARE @PTitle NVARCHAR(100)=ISNULL((SELECT Title FROM @DecimalTitle WHERE id=LEN(@DecimalNumber)), '')
SET @Str += ISNULL('مميز '+[dbo].[fnNumberToWord_Persian](@DecimalNumber) +' '+@PTitle,'')
RETURN @str

```

مثال: رشتہ 5445789240.5467800000000

پنج میلیارد و چهارصد و چهل و پنج میلیون و هفتاد و هشتاد و نه هزار و دویست و چهل ممیز پنجاه و چهار هزار و ششصد و هفتاد و هشت صد هزارم

[دانلود فایل](#)

نظرات خوانندگان

نویسنده: میثم چگینی
تاریخ: ۱۳۹۲/۰۸/۰۱ ۱۱:۴۶

با تشکر

یک مورد کوچک داره که برای نمایش درست نیاز به اضافه کردن N قبل از تمام کلمات فارسی می‌باشد
' صفر'N

نویسنده: حمیدرضا عابدینی
تاریخ: ۱۳۹۲/۰۸/۰۳ ۱۰:۵۳

حق با شماست!

من چون از Persian_100_CI_AI Collation استفاده کردم همه‌ی N‌ها را حذف کردم. ولی برای دیگر موارد برای اینکه به صورت علامت ؟ ظاهر نشود، همانطور که شما گفتید باید از N استفاده کرد.

ممnon

نویسنده: مهیار
تاریخ: ۱۳۹۲/۱۱/۱۵ ۲۰:۳۸

سلام،

ممnon از راه حلی که ارائه دادید.

برای حذف صفرهای غیرضروری موجود در اعشار پیشنهاد من کد زیر است که ساده‌تر و خواناتر است:

```
declare @value varchar(50) ='0.1010000000'  
select replace(rtrim(replace(@value,'0',' ')), ' ','0');
```

توضیح کد: ابتدا تمام صفرهای موجود در رشته را تبدیل به کاراکتر Space می‌کنیم Replace با تابع Rtrim تمام کاراکترهای Space آخر رشته را Remove می‌کنیم و مجدد کاراکترهای Space را بر می‌گیریم به صفر.

نویسنده: مهیار
تاریخ: ۱۳۹۲/۱۱/۱۵ ۲۱:۳۷

کد مربوط به حلقه While اtan هم کمی ساده‌تر کردم مخصوصاً اولین دستور حلقه که مربوط می‌شود به انتخاب سه رقم آخر رشته. ضمناً نیازی نیست که صراحتاً متغیر را به integer تبدیل کنید. با صفر جمع یا با یک ضرب کنید تا بصورت Implicit تبدیل صورت بگیره:

```
WHILE (@pNumber) <> '0'  
BEGIN  
    SET @number = RIGHT(@pNumber, 3) + 0  
    INSERT INTO @MyNumbers  
    SELECT  
        @Number / 100 * 100,  
    CASE  
        WHEN nbr BETWEEN 10 AND 19 THEN nbr  
        ELSE nbr / 10 * 10  
    END,  
    CASE  
        WHEN nbr BETWEEN 10 AND 19 THEN 0  
        ELSE nbr % 10  
    END  
    FROM (SELECT @Number % 100)S(nbr);  
  
    IF LEN(@pNumber) > 2
```

تبدیل اعداد صحیح و اعشاری به حروف در T-SQL با استفاده از Join

```
SET @pNumber = LEFT(@pNumber, LEN(@pNumber) -3)
ELSE
    SET @pNumber = '0'
END
```

نویسنده: حمیدرضا عابدینی
تاریخ: ۲۳:۳۲ ۱۳۹۲/۱۱/۲۰

سلام مهیار جان!
کد بسیار خوب و خوانایی گذاشتید، ممنونم!
در نتیجه:

```
DECLARE @IntegerNumber NVARCHAR(100),
@DecimalNumber NVARCHAR(100),
@PointPosition INT =case CHARINDEX('.', @pNumber) WHEN 0 THEN LEN(@pNumber)+1 ELSE CHARINDEX('. ', @pNumber) END

SET @pNumber=replace(rtrim(replace(@pNumber,'0',' ')), ' ','0');
SET @IntegerNumber= LEFT(@pNumber, @PointPosition - 1)
SET @DecimalNumber= SUBSTRING(@pNumber, @PointPosition+1 , LEN(@pNumber))

SET @pNumber= @IntegerNumber
```

عنوان:	SQL Instance
نوبتندہ:	۳ منفرد
تاریخ:	۱۹:۲۵ ۱۳۹۲/۱۰/۱۲
آدرس:	www.dotnettips.info
گروهها:	C#, SQL Server, Registry

ممکن است کاربر بر روی سیستم خود نسخه‌های مختلفی از SQL Express، SQL Server را نصب کرده باشد. برای مثال SQL 2008، SQL 2005، SQL 2012 و یا نسخه‌ای خاص (مثلا 2012) را چند بار روی سیستم خود نصب کرده باشد. SQL برای تفکیک این نسخه‌ها و نصب‌ها از مفهومی با عنوان Instance استفاده می‌کند. یعنی به هر نسخه نصب شده نامی یکتا می‌دهد تا بتوان به تفکیک به آنها دسترسی داشت.

برای اتصال به این نسخه‌ها باید در بخش آدرس سرور، از ترکیب نام سیستم و نام Instance به این شکل استفاده کرد:
SystemName\Instance

بعضی مواقع لازم است که لیست Instance‌های نصب شده روی سیستم کاربر را به دست آوریم. ADO.NET کلاسی به همین منظور تعبیه کرده که شبکه را جستجو کرده و SQL Instance‌های مختلف را که قابل دسترسی هستند را برای شما لیست می‌کند. استفاده از این کلاس بسیار ساده است:

```
using System.Data.SqlClient;
class Program
{
    static void Main()
    {
        // Retrieve the enumerator instance and then the data.
        SqlDataReaderEnumerator instance =
            SqlDataReaderEnumerator.Instance;
        System.Data.DataTable table = instance.GetDataSources();

        // Display the contents of the table.
        DisplayData(table);

        Console.WriteLine("Press any key to continue.");
        Console.ReadKey();
    }

    private static void DisplayData(System.Data.DataTable table)
    {
        foreach (System.Data DataRow row in table.Rows)
        {
            foreach (System.Data DataColumn col in table.Columns)
            {
                Console.WriteLine("{0} = {1}", col.ColumnName, row[col]);
            }
            Console.WriteLine("=====");
        }
    }
}
```

البته با توجه به اینکه شبکه را جستجو می‌کند در نرم افزار شما وقفه خواهد انداخت. خوب اگه بخواهیم Instance‌های نصب شده روی سیستم کاربر را پیدا کنیم چی؟ ساده‌ترین و سریعترین راه استفاده از رجیستری سیستم است. نام Instance‌ها در رجیستری ویندوز در آدرس زیر قابل دسترسی است:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names

برای استفاده از این کلید در C# می‌توان از کد زیر کمک بگیرید:

```
var key = Registry.LocalMachine.OpenSubKey(@"SOFTWARE\Microsoft\Microsoft SQL Server\Instance Names");

foreach (string sk in key.GetSubKeyNames())
{
    var rkey = key.OpenSubKey(sk);
    foreach (string s in rkey.GetValueNames())
    {
        MessageBox.Show("Sql instance name:" + s);
    }
}
```

فقط دو نکته قابل توجه است. برنامه باید در CPU Any کامپایل شود تا در سیستم‌های 64 بیتی بتوانید به محل درست رجیستری دسترسی پیدا کنید. چون نرم افزارهای 32 بیت در ویندوز 64 بیت در سیستم wow64 اجرا می‌شود که دسترسی به رجیستری آن در آدرس wow64 هر قسمت رجیستری است. بنابراین کد فوق در حالت CPU Any و غیر فعال بودن Prefer 32-bit قسمت در آدرس Properties برنامه به درستی اجرا می‌شود.

نکته: در SQL Default Instance مقدار MSSQLSERVER می‌باشد.

خواندن سریع اطلاعات فایل اکسل و ذخیره در بانک SQL

عنوان: مجتبی فخاری
نوبتندہ: ۱۳۹۲/۱۱/۱۷
تاریخ: ۱۸:۳۵
آدرس: www.dotnettips.info
گروه‌ها: C#, SQL Server, Excel

یکی از زمانبرترین عملیات‌ها در نرم افزارهای اتوماسیون، خواندن اطلاعات از فایل‌های اکسل با حجم بالا است. در صورتی که این کار را می‌توان با استفاده از کلاس `SqlBulkCopy` به سرعت انجام داد. در ادامه نحوه استفاده از این کلاس، همراه نمونه کدها آورده شده است.

توضیحات به صورت [Comment](#) است.

```
try
{
    انتخاب فایل اکسل//
    OpenFileDialog ofd = new OpenFileDialog();
    ofd.Title = "انتخاب فایل حاوی اطلاعات";
    ofd.Filter = "2003 (*.xls)|*.xls|2007 (*.xlsx)|*.xlsx";
    if (ofd.ShowDialog() == DialogResult.OK)
    {
        string excelConnectionString = "";
        string SourceFilePath = ofd.FileName;
        ایجاد کانکشن استرینگ برای خواندن کل اطلاعات از فایل اکسل و ریختن آنها در یک دیتابیل به نام//
        if (System.IO.Path.GetExtension(ofd.FileName) == ".xlsx")
            تشخیص نوع فایل اکسل برای ایجاد کانکشن استرینگ برای نسخه‌های مختلف اکسل//
            excelConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" +
SourceFilePath + ";Extended Properties=Excel 12.0";
        else if (System.IO.Path.GetExtension(ofd.FileName) == ".xls")
            excelConnectionString = "Provider=Microsoft.Jet.Oledb.4.0;Data Source=" +
SourceFilePath + ";Extended Properties=Excel 8.0";
        DataTable dt = new DataTable("tblinfos");

        using (System.Data.OleDb.OleDbConnection connection = new
System.Data.OleDb.OleDbConnection(excelConnectionString))
        {
            connection.Open();
            System.Data.OleDb.OleDbDataAdapter da = new
System.Data.OleDb.OleDbDataAdapter("SELECT * FROM [List$]", connection); //List
名 شیت در فایل اکسل است
            da.Fill(dt);
            connection.Close();
        }

        using (System.Data.OleDb.OleDbConnection connection = new
System.Data.OleDb.OleDbConnection(excelConnectionString))
        {
            this.Cursor = Cursors.WaitCursor;
            connection.Open();
            ایجاد ارتباط با بانک اس کیو ال//
            string sqlConnectionString = @"Data
Source=.SQLEXPRESS;AttachDbFilename=|DataDirectory|\BankDPR.mdf;Max Pool Size=6000; Connection
Timeout=50;Integrated Security=True;User Instance=True";

            Dataaccess db = new Dataaccess();
            DataTable dtr = db.select("Select Top(1) * From tblinfos"); //Fast Copy
            سтолنهای جدول مورد نظر برای تطبیق با ستونهای فایل اکسل
            *****
            using (System.Data.SqlClient.SqlBulkCopy bulkCopy = new
System.Data.SqlClient.SqlBulkCopy(sqlConnectionString,
System.Data.SqlClient.SqlBulkCopyOptions.KeepIdentity))
            {
                for (int i = 1; i < dtr.Columns.Count; i++)
                {
                    bulkCopy.ColumnMappings.Add(dt.Columns[i - 1].Caption,
dtr.Columns[i].Caption); //астفاده از خاصیت تطبیق داده می‌شود sql مورد نظر بانک
                }
            }
        }
    }
}
```

بدست آورن نام//
نام ستونهای فایل اکسل با نام ستونهای جدول ColumnMappings با استفاده از خاصیت Fast Copy
SqlBulkCopy یک شی از کلاس//
SqlBulkCopy تعريف یک شی از کلاس//
SqlBulkCopy Options KeepIdentity
DestinationTableName = "tblinfos";

خواندن سریع اطلاعات فایل اکسل و ذخیره در بانک SQL

```
است اطلاعات درون ان کپی گردد
به dt انجام عملیات کپی اطلاعات از دیتاتیبل با نام//;(dt)
بانک          bulkCopy.WriteToServer(dt);

        }

        this.Cursor = Cursors.Arrow;
        MessageBox.Show("اطلاعات از فایل شما خوانده شد");

    }

}

catch (Exception ex)
{
    this.Cursor = Cursors.Arrow;
    MessageBox.Show(ex.Message, "error");
}
```

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۱۷ ۲۱:۲۲

احتمالاً این قطعه کد مستقیماً از داخل سورس یکی از پروژه‌های شما بیرون آمده (مثال نیست؛ واقعی هست). می‌شد کمی اون را [refactor](#) کرد مثلاً یک متاد از داخلش بیرون آورد که این متاد داخلش مسیج باکس نباشه یا باز کردن یک صفحه دیالوگ و تغییر کرس. هم نداشته باشه چون باید در یک سطح بالاتر `try catch` بشه مشکلاتش. اون `select`ها مثلاً می‌شندن چند پارامتر، برای اینکه این کد قابلیت استفاده مجدد بهتری پیدا کنه. یا مثلاً اون `sqlConnectionString` از داخل کدها بیرون می‌آمد و می‌شد یک پارامتر جدید نمایش کرس هم داخل این متاد قرار نمی‌گرفت. نام جدول نهایی هم مثلاً یک پارامتر دیگر می‌شد برای سهولت استفاده مجدد و همچنین تست بهتر یک قطعه کوچک از کار. خود متاد اصلی هم می‌شد دو متاد کوچک‌تر؛ یکی کار `load` رو انجام می‌داد و دیگری کار `insert` سریع.

نویسنده: پالادین
تاریخ: ۱۳۹۲/۱۱/۱۹ ۱۳:۳۸

من هم با دوستمون محسن خان موافقم. اتفاق‌ها زیادی در مورد این کد باید بیافته تا به یک کد خوب تبدیل بشه. شما کاملاً به صورت Smart UI کد زدید. امیدوارم وقت بزارید و این کد رو بهینه کنید. توی همین سایت مثال‌های خوبی واسه‌ی یادگیری هست..

نویسنده: مجتبی فخاری
تاریخ: ۱۳۹۲/۱۱/۲۰ ۱۱:۳۶

بله این کد رو من دقیقاً از یکی از برنامه‌های آوردم و مال چندین سال پیشه.
اما خوبیش اینه که بدون خطاب جواب میده و فقط کافیه اسم جداول رو عوض کنید.

نویسنده: محمدی
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۲۹

با سلام و تشکر. این برنامه پیغام میدهد `List` وجود ندارد.

```
SELECT * FROM [List$]
```

لطفاً راهنمایی بفرمایید من چطور میتوانم نام شیت‌ها را بدست بیاورم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۲ ۱۰:۴۷

در کد فوق اکثر قسمت‌ها بر اساس یک سری پیش فرض مشخص تهیه شده‌اند و آن‌ها را تبدیل به پارامتر متغیر نکرده‌اند. نام شیت، همان نام برگه جاری است:



نویسنده: احمد زواری
تاریخ: ۱۳۹۳/۰۴/۱۱ ۲۳:۴۰

سلام. من از این روش برای خواندن فایل موجود در آدرس <http://www.site.com/market/export.aspx?type=dtod&date=20120405> استفاده کدم، ولی متاسفانه فیلد های عددی را NULL مقدار میگیرد. مشکل از چیست و چه جوری برطرف میشود؟

نوبنده: وحید نصیری
تاریخ: ۰:۲۴ ۱۳۹۳/۰۴/۱۲

این فایل استاندارد اکسل نیست. خروجی آن را با نوت پد باز کنید؛ یک فایل HTML معمولی است.

نوبنده: احمد زواری
تاریخ: ۳:۳۰ ۱۳۹۳/۰۴/۱۲

الان من نیاز شدید دارم این فایل رو تبدیل به SQL کنم، و چون هر روز این فایل تغییر میکنه نمیتونم توی فایل تغییری بدم، چه راه حلی هست برای این کار؟

نوبنده: وحید نصیری
تاریخ: ۹:۱۹ ۱۳۹۳/۰۴/۱۲

فایل های HTML را با استفاده از کتابخانه [HTML Agility Pack](#) پردازش می کنند.

فرض کنید

- تمام اکانت‌های مدیریتی توکار SQL Server را حذف کرده‌اید (یا برایتان حذف کرده‌اند).
- بجز کاربر SA، تمام کاربران را از نقش SYSADMIN حذف کرده‌اید؛ شامل تمام اکانت‌های ویندوزی و همچنین خود SQL Server.
- پسورد SA را هم فراموش کرده‌اید یا ندارید.

خوب، الان چکار می‌خواهید بکنید؟!

احتمالاً نصب مجدد سرور را پیشنهاد دهید یا بانک اطلاعاتی Master را بازسازی کنید که در هر دو حالت تمام تنظیمات سرور را از دست خواهید داد. روش دیگری هم بدون از دست دادن تنظیمات سرور وجود دارد که در ادامه آنرا بررسی خواهیم کرد.

افزودن یک اکانت مدیریتی جدید به SQL Server

اگر دسترسی کامل مدیریتی خود را به SQL Server از دست داده‌اید باید ابتدا به آن سرور لاگین کنید؛ با این فرض که کاربر وارد شده به سیستم، جزو local administrators group است.

```
C:\>sqlcmd -S .
1> create login [MachineName\TestUser] from windows;
2> go
1> exec sp_addsrvrolemember 'MachineName\TestUser', 'sysadmin'
2> go
1> select is_srvrolemember('sysadmin', 'MachineName\TestUser')
2> go
-----
1
(1 rows affected)
1>
```

سپس خلاصه مواردی را که ملاحظه می‌کنید، اجرای سه دستور است:

الف) اجرای sqlcmd با پارامتر S و مشخص سازی وله‌ی مورد نظر

البته حالت کامل آن در صورتیکه از وله‌ی پیش فرض استفاده نمی‌کنید SQLCMD -S Server_Name\Instance_Name است. S نیز در اینجا باید با حروف بزرگ نوشته شود.

ب) اجرای create login را بر روی یکی از اکانت‌های محلی سیستم اجرا کنید. مثلا MachineName\administrator یا هر اکانت موجود دیگری که لازم است. نام آن نیز باید حتماً به شکل server_name\user_name باشد.

در حین استفاده از sqlcmd، هر فرمان زمانی اجرا می‌شود که در سطر پس از آن، یک go نوشته شده و enter شود.

ج) سپس توسط sp_addsrvrolemember به این اکانت اضافه شده، دسترسی sysadmin بدھید.

برای آزمایش آن فقط کافی است از متدهای is_srvrolemember برای کوئری گرفتن استفاده کنید و یا سعی کنید توسط اکانت اضافه شده، به SQL Server لاگین کنید. این اکانت اکنون در قسمت Security/logsins قابل مشاهده است.

اگر نمی‌خواهید از اکانت‌های ویندوزی استفاده کنید، create login را به نحو ذیل مقدار دهی کنید:

```
C:\>sqlcmd -S .
1> use master
2> go
Changed database context to 'master'.
1> create login test_user with password='123#123'
2> go
1> exec sp_addsrvrolemember 'test_user', 'sysadmin'
2> go
1>
```

سپس به این کاربر اضافه شده با کلمه‌ی عبور مشخص، توسط `sp_addsrvrolemember` دسترسی `sysadmin` بدهید.

نظرات خوانندگان

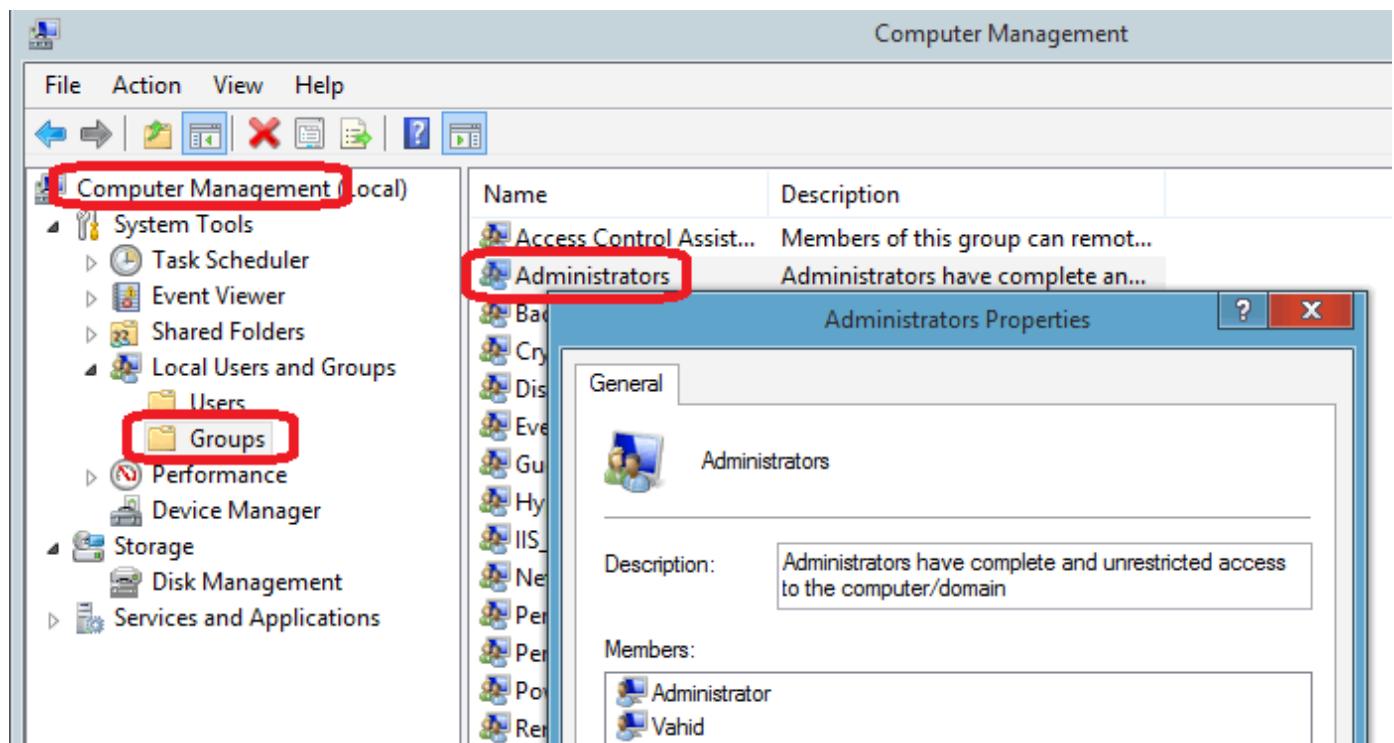
نویسنده: وحید
تاریخ: ۱۳۹۲/۱۲/۱۱ ۱۳:۱۹

با سلام؛ دستور خط اول را که می‌نویسم ارور زیر را می‌دهد

```
Sqlcmd: Error: Microsoft SQL Server Native Client 11.0 : Login failed for user 'NARM-2-PC\Narm-2'..
```

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۲/۱۱ ۱۴:۸

در متن قید شده «... با این فرض که کاربر وارد شده به سیستم، جزو local administrators group است ...». یعنی:



یکی از وظایف اصلی مدیر و یا توسعه دهنده یک بانک اطلاعاتی، نوشتتن کدهای T-SQL و اندازه گیری عملکرد آنها می باشد. ابزارهای مختلفی برای انجام این کار وجود دارد، چه آنهاست که در خود SQL Server بصورت محلی وجود دارند و چه آنهاست که توسط شرکت های ثالث ارائه می شوند. اما مسئله مهمی که باید در نظر بگیرید چگونگی نوشتتن یک پرس و جو (Query) و اندازه گیری کارآئی آن می باشد و اینکه باید روی چه مواردی متمرکز شد. در اکثر مواقع گرفتن زمان اجرای یک پرس و جو تا اندازه ای خوب می باشد. یکی از مواردی که باید روی آن متمرکز شد منابع استفاده شده توسط سرور می باشد، در حالیکه زمان اجرای پرس و جو به پارامترهای دیگری همچون بار سرور نیز بستگی دارد. علاوه بر استفاده از پروفایلر و نقشه اجرای کوئری (Execution Plan)، می توانید از SET STATISTICS TIME نیز استفاده نمایید.

SET STATISTICS TIME تنظیمی است که برای اندازه گیری منابع مورد نیاز اجرای یک پرس و جو به شما کمک می کند. SET STATISTICS TIME آمار مربوط به زمان تجزیه و تحلیل (Parse)، کامپایل و اجرای هر دستور در یک پرس و جو را نمایش می دهد. راه های مختلف اندکی برای مقایسه آماری دو پرس و جو و انتخاب بهترین آنها برای استفاده وجود دارند. برای روش نکردن این تنظیم دو راه وجود دارد. ابتدا اینکه از دستور Set برای روشن و خاموش کردن استفاده نمایید و یا اینکه از طریق Query Analyzer اقدام به انجام این کار نمایید.

SET STATISTICS TIME ON

برای اینکه بتوانید آمارهای این تنظیمات را مشاهده کنید می بایست قبل از اجرای پرس و جو تنظیم مورد نظر را روشن نمایید. در نظر داشته باشید که با روشن کردن این تنظیم، برای تمامی پرس و جوهای مربوط به آن جلسه (Session) روشن خواهد ماند تا زمانیکه آنرا خاموش نمایید.

```
SELECT ProductID, StartDate, EndDate, StandardCost
FROM Production.ProductCostHistory
WHERE StandardCost < 500.00;
```

با اجرای دستورات بالا خروجی آن بصورت زیر می باشد:

```
SQL Server parse and compile time:
CPU time = 0 ms, elapsed time = 1 ms.

(269 row(s) affected)
```

```
SQL Server Execution Times:
CPU time = 1 ms, elapsed time = 2 ms.
```

زمان صرف شده برای اجرای یک کوئری به دو بخش تقسیم می شود:

زمان کامپایل و تجزیه و تحلیل (parse and compile time) : زمانیکه یک کوئری را برای اجرا به SQL Server ارائه می دهید، آنرا از نظر خطای نحوی بررسی می نماید و بهینه ساز یک نقشه بهینه را برای اجرا تولید می نماید. اگر به خروجی نگاه Server کنید، زمان پردازش (CPU time) و زمان سپری شده (elapsed time) را نشان می دهد. منظور از زمان پردازش زمان واقعی صرف شده روی پردازندۀ می باشد و زمان سپری شده اشاره به زمان تکمیل شدن عملیات کامپایل و تجزیه و تحلیل می باشد. تفاوت بین زمان پردازش و زمان سپری شده ممکن است زمان انتظار در صفحه برای گرفتن پردازندۀ و یا انتظار برای تکمیل عملیات IO باشد.

زمان اجرا (Execution Times) : این زمان اشاره به زمان سپری شده برای تکمیل اجرای نقشه کامپایل شده دارد. زمان پردازش اشاره به زمان واقعی صرف شده روی پردازندۀ دارد و زمان سپری شده نیز مجموع زمان صرف شده تا تکمیل اجرای دستور که شامل زمان انتظار برای تکمیل عملیات IO و زمان صرف شده برای انتقال خروجی به کلاینت می باشد، دارد. زمان پردازش می تواند به عنوان مبنای برای اندازه گیری کارآئی مورد استفاده قرار بگیرد. این مقدار در اجراهای مختلف تقاضت چندانی با هم ندارند جز اینکه کوئری و یا داده ها تغییر نمایند. توجه نمایید که زمان براساس میلی ثانیه می باشد. زمان سپری شده نیز به فاکتورهایی مانند

بارگذاری روی سرور، بارگذاری IO، و پهنای باند بین سرور و کلاینت وابسته می باشد. بنابراین همیشه زمان پردازش به عنوان مبنای برای اندازه گیری کارایی استفاده می شود.

در این بخش به بررسی **SET STATISTICS IO** در SQL Server **SET STATISTICS TIME** در پرداختیم. در بخش بعدی به بررسی **IO** اندازه گیری کارایی پرس و جوها از نظر میزان استفاده IO خواهیم پرداخت.

چندی پیش لازم بود که از طریق ویندوز، [یک یوزر ادمین برای sql](#) بسازم که خدا را شکر میسر شد. اما خوب است بدانید با SQL Server نیز می‌توانید یک کاربر جدید ویندوز بسازید. این مورد را در ادامه‌ی بحث بررسی خواهیم کرد.

البته باید دقت داشت که در حقیقت این امکان را توسط خط فرمان فعال شده‌ی ویندوز، به نام `xp_cmdshell` در اختیار خواهیم داشت. توسط این رویه ذخیره شده، هر کاری را که در ویندوز توسط خط فرمان استاندارد آن می‌توانید انجام دهید، اینجا نیز قادر به انجام آن خواهید بود. برای مثال یک کاربر را ادمین کنید و امثال آن. برای اینکار ابتدا باید این امکان را جهت کاربر `sa`، یا هر کاربر دیگری که در SQL Server نقش `sysadmin` دارد، فعال کنید. برای این منظور کوئری‌های ذیل را اجرا کنید:

```
-- To allow advanced options to be changed.
EXEC sp_configure 'show advanced options', 1
GO
-- To update the currently configured value for advanced options.
RECONFIGURE
GO
-- To enable the feature.
EXEC sp_configure 'xp_cmdshell', 1
GO
-- To update the currently configured value for this feature.
RECONFIGURE
GO
```

بعد برای اینکه کاربری را اضافه کنید، کد زیر را کپی و اجرا کنید:

```
xp_cmdshell 'net user nadeema /Add'
```

و در آخر جهت تغییر گروه این کاربر و افزودن آن به مجموعه‌ی ادمین‌های محلی، می‌توانید کد زیر را نوشته و اجرا کنید:

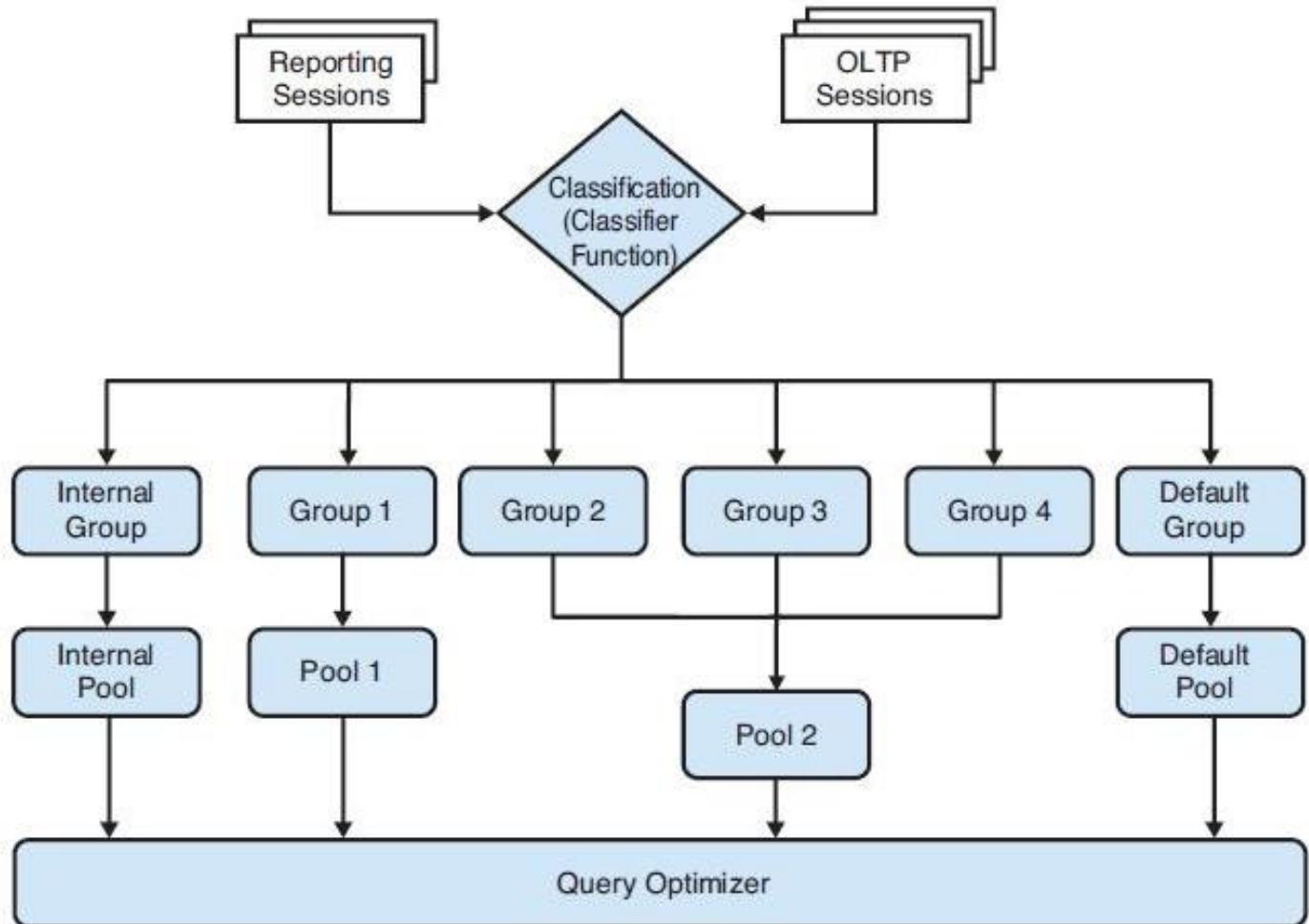
```
xp_cmdshell 'net localgroup Administrator nadeema /add'
```

اجازه می‌دهد تا انواع مختلف Session را بر روی Server طبقه بندی کنید که به نوبه خود چگونگی کنترل تخصیص منابع سرور به فعالیت داده شده را به شما اعطا می‌کند. این قابلیت کمک می‌کند که ادامه فرآیندهای OLTP تضمین شود و یک عملکرد قابل پیش‌بینی فراهم می‌کند تا توسط فرآیندهای غیر قابل پیش‌بینی، تحت تاثیر منفی قرار نگیرد. با استفاده از Resource Governor، قادر خواهد بود نحوه دستیابی به Session را به منظور محدود کردن منابع خاص برای SQL Server مشخص کنید. به عنوان مثال می‌توانید مشخص کنید که بیش از 20 درصد از پردازنده یا منابع حافظه به گزارش‌های در حال اجرا اختصاص داده نشود. هنگامیکه این ویژگی فعال باشد، مهم نیست چه تعداد گزارش در حال اجرا است، آنها هرگز نمی‌توانند از تخصیص منابع تعیین شده تجاوز کنند. البته این موضوع عملکرد گزارش گیری را کاهش می‌دهد ولی عملکرد فرآیندهای OLTP حداقل توسط گزارش‌ها، دیگر تحت تاثیر منفی قرار نمی‌گیرد.

1- بررسی اجمالی Resource Governor

با کنترل تخصیص منابع بر حسب Workload کار می‌کند. هنگامی که یک درخواست اتصال به موتور بانک اطلاعاتی ارسال می‌شود درخواست براساس یک تابع رده بندی (Classification function) طبقه بندی می‌شود. تابع رده بندی یک تابع اسکالر است که از طریق T-SQL تعریف می‌شود. تابع رده بندی، اطلاعات را درباره یک اتصال (برای مثال login ID، application name, hostname, server role دسته بندی درخواست اتصال، آنها به گروه‌های حجم کاری (Workload Group) که برای رده بندی تعریف شده اند، شکسته می‌شوند. هر Workload Group مرتبط با یک مخزن منابع (Resource Pool) است.

یک Resource Pool، منابع فیزیکی SQL Server را نمایش می‌دهد (در حال حاضر در SQL Server 2008)، تنها منابع فیزیکی موجود برای پیکربندی پردازنده و حافظه است) و مقدار حداکثر پردازنده و یا منابع حافظه را که به نوع خاصی از Workload می‌شود، تعیین می‌کند. هنگامی که یک اتصال طبقه بندی شده و در Workload Group صحیح خود قرار می‌گیرد به این اتصال، پردازنده و منابع حافظه به اندازه نسبت داده شده به آن تخصیص داده می‌شود و سپس Query به Query Optimizer برای اجرا داده می‌شود.



2- اجزای Resource Governor :

Resource Governor، از سه قسمت اصلی تشکیل شده است: Classification، Workload Groups و Resource Pools. در ک این سه قسمت و چگونگی تعامل آنها به در ک و استفاده از Resource Governor کمک می کند.

2-1- Classification :

Classification، فرآیند ارزیابی اتصالات ورودی کاربر و اختصاص آن به یک Workload Group است که توسط منطق موجود در یک تابع تعریف شده توسط کاربر (user-defined function) انجام می شود. تابع نام یک Workload Group را برمی گرداند که از آن برای مسیر دهی Session به Workload Group مناسب استفاده می کند. هنگامی که Resource Governor پیکربندی می شود فرآیند ورود به سیستم برای یک Session شامل گام های زیر است:

- Login authentication
- LOGON trigger execution
- Classification
- Workload Groups

Workload Groups، ظرفی برای اتصالات مشابه هستند که با توجه به معیارهای طبقه بندی برای هر اتصال گروه بندی می شوند. همچنین مکانیسمی برای تجمع نظارت بر روی منابع مصرفی فراهم می کند. Workload Groups دو Workload Group از پیش تعریف شده دارد: یک گروه داخلی (internal group) و یک گروه پیش فرض (default group).

Internal Workload Group، تنها توسط فرآیندهای داخلی موتور بانک اطلاعاتی استفاده می شود. معیارهای طبقه بندی را برای گروه های داخلی نمی توانید تغییر دهید و همچنین هیچ یک از درخواست های کاربران را برای انتقال به گروه داخلی نمی توانید رده بندی کنید، با این حال بر گروه داخلی می توانید نظارت کنید.

درخواست‌های اتصال به طور خودکار هنگامی که شرایط زیر وجود دارد، به *Default Workload Group* رده بندی می‌شوند:

- معیاری برای طبقه بندی درخواست وجود ندارد.
- کوچشی برای رده بندی درخواستی به گروهی که وجود ندارد.
- خرابی کلی Classification

در مجموع 20 عدد *Workload Group* را پشتیبانی می‌کند. از آنجائی که دو عدد از آنها برای *Resource Governor* (user-defined) *Workload Group* تعریف شده توسط کاربر (user-defined) می‌توان تعریف نمود.

:Resource pools 2-3-

(مخزن منابع)، نشان دهنده تخصیص منابع فیزیکی به *Resource Pool* از دو بخش تشکیل شده است:

- در بخش نخستین حداقل رزرو منابع را مشخص می‌کنیم، این بخش از مخزن منابع با مخازن دیگر همپوشانی نمی‌کند.
 - در بخش دیگر حداقل ممکن رزرو منابع را برای مخزن مشخص می‌کنیم، تخصیص منابع با مخازن دیگر مشترک است.
- در SQL Server 2008 مخزن منابع با تعیین حداقل و حداقل تخصیص CPU و حداقل و حداقل تخصیص حافظه تنظیم می‌گردد. با تنظیم حداقل، در دسترس بودن منبع از مخزن تضمین می‌شود. از آنجائی که در هر رزرو حداقل منابع داخلی نمی‌تواند وجود داشته باشد، مجموع مقادیر حداقل در تمام مخازن از 100% کل منابع Server نمی‌تواند تجاوز کند.
- مقدار حداقل در محدوده بین حداقل و شامل 100% مقدار می‌تواند تنظیم گردد. تنظیم حداقل نشان دهنده مقدار حداقل است که بک Session می‌تواند مصرف کند، مادامی که منابع در دسترس باشد و توسعه مخزن دیگر که با حداقل مقدار غیر صفر پیکربندی شده، استفاده نشود. هنگامی که یک مخزن با حداقل مقدار غیر صفر تعریف شده، مقدار حداقل موثر از مخزن‌های دیگر دوباره تنظیم می‌شوند، در صورت لزوم حداقل مقدار موجود از جمع کل حداقل منابع مخازن دیگر کسر می‌گردد.
- برای مثال، دو مخزن تعریف شده توسط کاربر (user-defined) را در نظر بگیرید. مخزن اول P001 با مقدار حداقل 20% و مقدار حداقل 100% تعریف شده، مخزن دیگری P0012 با مقدار حداقل 50% و مقدار حداقل 70% تعریف شده است. حداقل مقدار موثر برای P0011 50% است (100% منهای 50% مخزن 12) و حداقل مقدار موثر برای P0012، 70% است زیرا حداقل مقداری است که پیکربندی شده است، گرچه 80% باقی می‌ماند.
- بخش مشترکی از مخزن (مقدارش بین مقدار حداقل و مقدار حداقل موثر است) که برای تعیین مقدار منابع مورد استفاده است، توسعه مخزن می‌تواند مصرف شود اگر منابعی موجود باشد و توسعه مخازن دیگر مصرف نشده باشد. هنگامی که منابعی توسعه یک مخزن مصرف می‌شوند، آنها به یک مخزن مشخص نسبت داده می‌شوند، به بیان دیگر اشتراکی نیستند تا زمانی که فرآیند در آن مخزن به اتمام برسد.
- برای توضیح بیشتر یک سناریو که در آن سه مخزن تعریف شده توسط کاربر (user-defined) وجود دارد، را در نظر بگیرید:
- P001A با حداقل مقدار 10% و حداقل مقدار 100% تعریف می‌شود.
 - P001B با حداقل مقدار 35% و حداقل مقدار 90% تعریف می‌شود.
 - P001C با حداقل مقدار 30% و حداقل مقدار 80% تعریف می‌شود.
- مقدار موثر P001A و مجموع در صد منابع به اشتراک گذاشته P001A به شرح زیر محاسبه خواهد شد:
- $$\text{حداکثر مقدار P001A} - (\text{حداکثر مقدار P001B} + \text{حداکثر مقدار P001C}) = \text{حداکثر مقدار موثر P001A}$$
- $$(\text{حداکثر مقدار موثر P001A} - (\text{حداکثر مقدار P001A} + \text{حداکثر مقدار P001B})) = \text{اشتراک P001A}$$
- جدول زیر مقدار حداقل موثر و اشتراکی را برای هر مخزن در این پیکربندی نمایش می‌دهد:

Resource Pool	MIN %	MAX %	Effective MAX %	Shared %
Internal	0	100	100	100
Default	0	100	25	25
PoolA	10	100	35	25
PoolB	35	90	50	15
PoolC	30	80	35	5

Internal Pool، منابع مصرف شده توسط فرآیندهای داخلی موتور بانک اطلاعاتی را نشان می‌دهد. این مخزن تنها شامل گروههای داخلی است و به هیچ وجه قابل تغییر نیست. مخزن داخلی مقدار ثابت حداقل صفر و حداقل 100% را دارد و مصرف منابع توسط مخزن داخلی، از طریق تنظیمات در هر مخزن دیگر محدود یا کاسته نمی‌شود.

به عبارت دیگر حداقل مقدار موثر مخزن داخلی همیشه 100% است. هر workloads در مخزن داخلی برای عملکرد Server حیاتی در نظر گرفته می‌شود و Resource Governor در صورت لزوم اجازه می‌دهد تا مخازن داخلی 100% منابع موجود را مصرف کند حتی اگر به معنی نقض نیازمندیهای منابع از سایر مخازن باشد.

Default Pool، اولین مخزن تعریف شده کاربر است. قبل از هرگونه پیکربندی، Default Pool تنها حاوی Default group است. Default group علاوه بر Default Pool نمی‌تواند ایجاد یا حذف شود اما می‌تواند تغییر کند. Default group می‌تواند شامل گروههای تعریف شده توسط کاربر (user-defined) نیز باشد.

3- پیکربندی Resource Governor : پیکربندی Resource Governor شامل مراحل زیر است:

- فعال کردن Resource Governor

- ایجاد مخازن منابع (Resource Pools) تعریف شده توسط کاربر (user-defined)

- تعریف دادن آن به مخازن

Classification function

- ثبت Resource Governor به Classification function

3-1- فعال کردن Resource Governor

پیش از اینکه بتوانید یک Resource Pool را ایجاد کنید، نیاز است تا نخست Resource Governor را فعال کنید.

3-2- تعریف Resource Pool :

ویژگی‌های موجود برای یک Resource Pool عبارتند از:

%Name, Minimum CPU %, Maximum CPU%, Min Memory%, Max Memory

3-3- تعریف Workload Group :

پس از اینکه Resource Pool را تعریف کردید، گام بعدی ایجاد یک Workload Group و اختصاص آن به Resource Pool مناسب است. چندین Workload Group را می‌توان به مخزن (Pool) یکسان نسبت داد اما یک Workgroup را به چندین Resource Pool نمی‌توان نسبت داد. خواص انتخابی موجود برای Workload Groups به شما اجازه می‌دهد سطح بهتری از کنترل را روی اجرای دستورات یک Workload Group تنظیم کنید. انتخاب‌های موجود عبارتند از:

Importance

اهمیت نسبی (کم، متوسط یا بالا) Workload Group درون Resource Pool را تعیین می‌کند. اگر چندین Workload Group را در یک Resource Pool تعیین کنید این تنظیمات تعیین می‌کند که درخواست‌ها در عرض یک Workload Group در اولویت بالاتر یا پائین‌تری از Workload Group‌های دیگر درون همان Resource Pool اجرا شوند، مقدار متوسط تنظیم پیش فرض است. در حال حاضر فاکتورهای وزنی برای هر تنظیم کم برابر 1، متوسط برابر 3 و زیاد برابر 9 است. به این معنی که زمانبند به اجرای Session‌های درون workgroup های با اهمیت بالا، سه برابر بیشتر از workgroup‌های با اهمیت متوسط و نه برابر بیشتر از

workgroup کم اهمیت، مبادرت خواهد کرد.

: Maximum Request 3-3-2-

حداکثر تعداد درخواست‌های همزمان که اجازه دارند در یک Workload Group اجرا شوند را مشخص می‌کند. تنظیم پیش فرض، صفر، تعداد نامحدود دستور را اجازه می‌دهد.

: CPU Time 3-3-3-

حداکثر مقدار زمان پردازندۀ در ثانیه را مشخص می‌کند که یک درخواست درون Workload Group می‌تواند استفاده کند. تنظیم پیش فرض، صفر، به معنی نامحدود است.

: % Memory Grant 3-3-4-

به صورت در صد، حداکثر مقدار اعطای حافظه برای اجرا (Execution grant memory)، که یک تک دستور از Resource Pool می‌تواند اخذ کند را مشخص می‌کند. این درصد نسبی است از مقدار حافظه ای که به به Resource Pool نسبت داده می‌شود. محدوده مجاز مقادیر از 0 تا 100 است. تنظیم پیش فرض 25 است.

Execution grant memory، مقدار حافظه ای است که برای اجرای query استفاده می‌شود (نه برای Buffer کردن یا cache کردن) که می‌تواند صرفه نظر از Resource Pool یا Session توسط تعدادی از Workload Group‌ها به اشتراک گذاشته شود. توجه شود که تنظیم این مقدار به صفر از اجرای عملیات Join Hash و دستورات مرتب سازی در Workload Group‌های تعریف شده توسط کاربر (user-defined) جلوگیری می‌کند. همچنین این مقدار توصیه نمی‌شود بیشتر از 70 باشد زیرا ممکن است Server قادر نباشد، اگر های همزمان در حال اجرا باشند، حافظه آزاد کافی اختصاص دهد.

: Grant Time-out 3-3-5-

حداکثر زمان، به ثانیه، که یک منبع منتظر می‌ماند تا در دسترس شود را مشخص می‌کند. اگر منبع در دسترس نباشد، فرآیند ممکن است با یک خطای time-out مواجه شود. تنظیم پیش فرض، صفر، به معنی این است که سرور را با استفاده از محاسبات داخلی بر مبنای هزینه پرس و جو (query cost) با تعیین حداکثر زمان برآورد می‌کند.

: Degree of Parallelism 3-3-6-

حداکثر درجه موازی سازی (DOP) را برای پرس و جوهای موازی تعیین می‌کند. محدوده مجاز مقادیر از 0 تا 64 است. تنظیم پیش فرض، صفر، به معنی این است که فرآیندها از تنظیمات عمومی استفاده می‌کنند.

3-4- ایجاد یک Classification function

پس از تعریف Resource Pool و Workload Group نیاز است که شامل منطق ارزیابی اتصالات و نسبت دادن آنها به Workload Group مناسب است. SQL Server Session برای هر اتصال Classification function جدید به بکار می‌رود. هر Workload Group در Session نسبت داده شده به آن باقی می‌ماند تا زمانی که به پایان برسد، مگر اینکه صراحتاً یک گروه متفاوت دوباره نسبت داده شود. فقط یک Classification function فعال در هر زمان می‌تواند وجود داشته باشد. در صورت عدم تعریف شدن یا عدم فعال بودن Workload Group Default همه اتصالات به Classification function نسبت داده می‌شوند. Classification function یک نام مستعار برای دیتا تایپ sysname که نوع آن workgroup است (که یک نام مستعار برای Default 128 است). بر می‌گرداند. اگر تابع تعریف شده مقدار 'Default', NULL یا نام گروهی که وجود ندارد را برگرداند، Session به Workload Group Default نسبت داده می‌شود. همچنین اگر به هر دلیلی تابع با موفقیت خاتمه نیابد Session به Default نسبت داده می‌شود.

منطق Classification function معمولاً مبتنی بر ویژگی‌های اتصال است و اغلب از طریق مقدار بازگشتی توابع سیستمی از قبیل:

Workload Group APP_NAME()، SUSER_NAME()، IS_MEMBER()، IS_SERVERROLEMEMBER()، HOST_NAME() اتصال مشخص می‌شود. علاوه بر این توابع می‌توانید از ویژگی‌های توابع دیگر برای ساخت منطق رده بندی استفاده کنید. تابع LOGINPROPERTY() شامل دو ویژگی (DefaultLanguage و DefaultDatabase) می‌باشد که می‌تواند برای ConnectionPROPERTY() استفاده شود. بعلاوه تابع (TCP Port و Local IP address) یا Client's IP Address را برای استفاده اتصالات فراهم می‌کند. برای مثال می‌توانید برای یک اتصال، یک Workload Group نسبت دهید، مبتنی بر اینکه subnet یک اتصال از کجا می‌آید. نکته: اگر قصد دارید از هر یک از توابع (HOST_NAME و APP_NAME) در تابع RDE بندی TAN استفاده کنید، توجه داشته باشید این امکان وجود دارد مقادیر بازگردنده شده توسط کاربران تغییر داده شوند، گرچه به طور کلی گرایش به استفاده از تابع (APP_NAME) برای RDE بندی اتصالات بیشتر است.

4- بررسی نمونه ای از پیکربندی Resource Governor

برای سادگی، در این قسمت مثالی ارائه می شود که از تابع (SUSER_NAME) استفاده می کند: در گام نخست، دو ایجاد می شود (OLTPPool و ReportPool)

```

CREATE RESOURCE POOL [ReportPool] WITH(
    min_cpu_percent=0,
        max_cpu_percent=20,
    min_memory_percent=0,
        max_memory_percent=30)
GO
CREATE RESOURCE POOL [OLTPPool] WITH(
    min_cpu_percent=80,
        max_cpu_percent=100,
    min_memory_percent=75,
        max_memory_percent=100)
GO

```

در گام بعدی، دو Workload Group ایجاد می شود (OLTPWG1 و ReportWG1)

```

CREATE WORKLOAD GROUP [ReportWG1] WITH(
    group_max_requests=0,
        importance=Medium,
    request_max_cpu_time_sec=0,
    request_max_memory_grant_percent=25,
    request_memory_grant_timeout_sec=0,
        max_dop=0) USING [ReportPool]
GO
CREATE WORKLOAD GROUP [OLTPWG1] WITH(
    group_max_requests=0,
        importance=High,
    request_max_cpu_time_sec=0,
    request_max_memory_grant_percent=25,
    request_memory_grant_timeout_sec=0,
        max_dop=0) USING [OLTPPool]
GO

```

سپس دو Login ایجاد می شود (oltp_user و report_user) که در تابع ردہ بندی استفاده خواهند شد برای مشخص کردن این که اتصالات به کدام Workload Group نسبت داده شوند. پس از اضافه کردن Login ها به عنوان User ها به Database Sesions مورد نظر ما، در بانک اطلاعاتی Master تابع ردہ بندی (Classification function) را ایجاد می کنیم:

```

use master
go
CREATE FUNCTION dbo.WorkgroupClassifier ()
RETURNS SYSNAME WITH SCHEMABINDING
AS
BEGIN
DECLARE @WorkloadGroup SYSNAME = N'Unidentified';
SET @WorkloadGroup = CASE suser_name()
WHEN N'report_user' THEN N'ReportWG1'
WHEN N'oltp_user' THEN N'OLTPWG1'
ELSE N'Unidentified'
END;
RETURN @WorkloadGroup;
END;
Go
GRANT EXECUTE on dbo.WorkgroupClassifier to public
Go

```

می توان تابع () WorkgroupClassifier با اجرای دستور زیر برای Login های متفاوت تست نمود:

```
select dbo.WorkgroupClassifier()
```

در ادامه دستور زیر برای پیکربندی تابع رده بندی به Resource Governor استفاده می شود:

```

ALTER RESOURCE GOVERNOR
WITH (CLASSIFIER_FUNCTION = dbo.WorkgroupClassifier);
ALTER RESOURCE GOVERNOR RECONFIGURE;
```

5- اصلاح پیکربندی Resource Governor

می توانید در محیط SSMS تنظیمات Workload Group و Resource Pool را تغییر دهید (برای مثال حداکثر استفاده CPU برای یک Workload Group یا درجه اهمیت یک Resource Pool). متناظراً می توان از دستورات T-SQL استفاده نمود.

نکته: پس از اجرای دستورات ALTER WORKLOAD GROUP یا ALTER RESOURCE POOL، برای اعمال کردن تغییرات اجرای دستور ALTER RESOURCE GOVERNOR RECONFIGURE نیاز می باشد.

5-1- حذف Workload Group

یک Workload Group را اگر هر نوع Session فعال نسبت داده شده به آن وجود داشته باشد، نمی توان حذف نمود. اگر یک Session شامل Workload Group های فعال باشد، حذف Workload Group یا جابجایی آن به یک Resource Pool متفاوت، هنگامی که دستور ALTER RESOURCE GOVERNOR RECONFIGURE برای اعمال نمودن تغییرات فراخوانی می شود، با خطأ مواجه خواهد شد.

5-2- حذف Resource Pools

یک Resource Pool را اگر هر نوع Workload Group نسبت داده شده به آن وجود داشته باشد، نمی توان حذف نمود. نخست نیاز

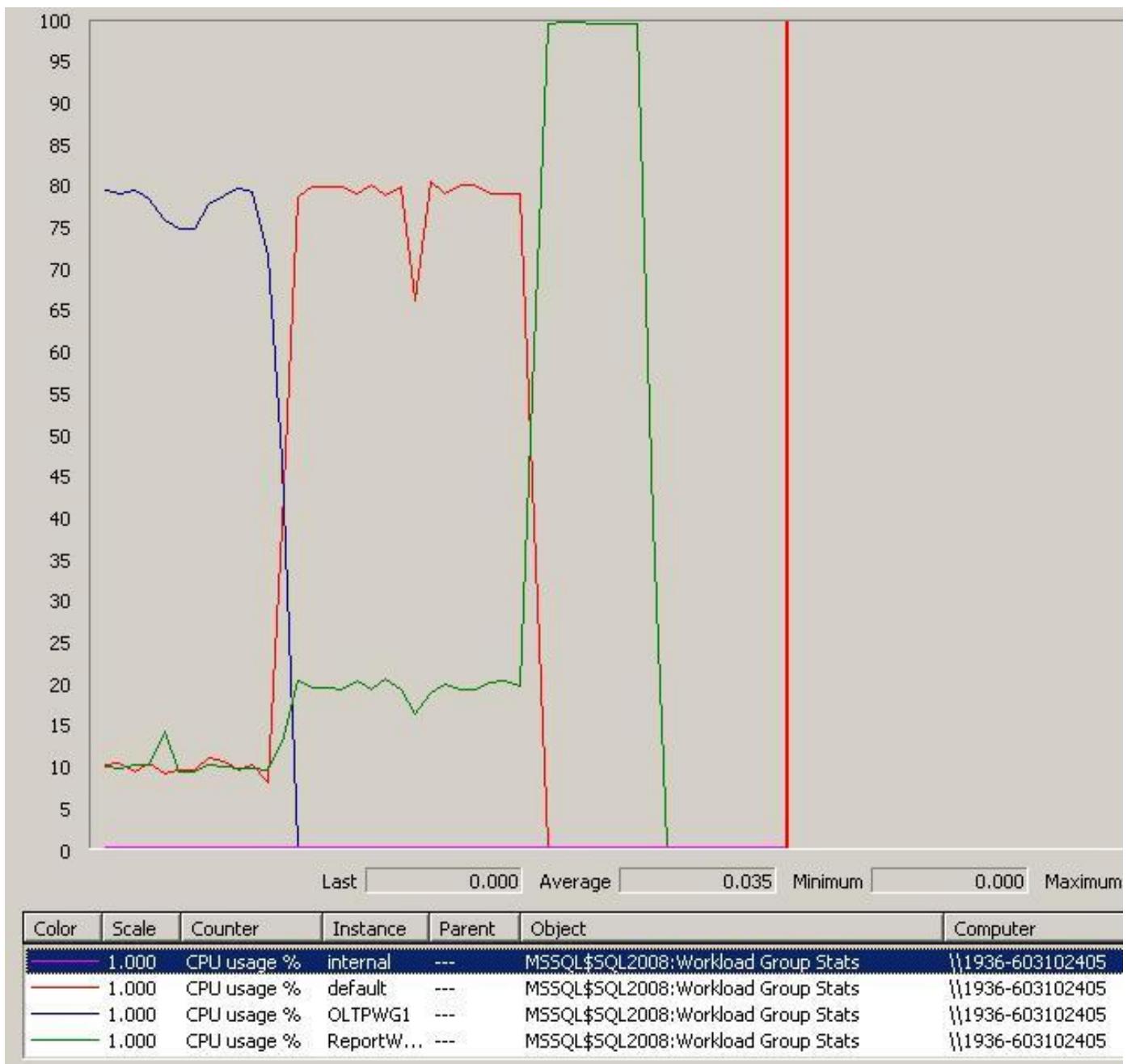
دارید Workload Group حذف شود و یا به Resource Pool دیگری جابجا گردد.
:Classification function 5-3- اصلاح

اگر نیاز دارید تغییراتی در تابع رده بندی ایجاد نماید، مهم است توجه داشته باشید که تابع رده بندی تا زمانی که مشخص شده (marked) برای Resource Governor است، نمی‌توان آنرا حذف و یا تغییر داد. پیش از اینکه بتوان تابع رده بندی را اصلاح و یا حذف نمود نخست نیاز دارید Resource Governor را غیر فعال نماید. متناسبًا می‌توان تابع رده بندی را جایگزین کرد با اجرای دستور CLASSIFIER_FUNCTION (passing) یک اسم متفاوت برای ALTER RESOURCE GOVERNOR و فرستادن همچنین می‌توان با اجرای دستور زیر تابع رده بندی جاری را غیر فعال نمود:

```
ALTER RESOURCE GOVERNOR
WITH (CLASSIFIER_FUNCTION = NULL);
ALTER RESOURCE GOVERNOR RECONFIGURE;
```

تابع رده بندی می‌توان تعریف کرد که نام Workload Group را از جداول یک بانک اطلاعاتی جستجو کند به جای اینکه نام Workload Group به صورت hard-coding و مطابق با ضوابط درون تابع باشد. عملکرد، در موقع دسترسی به جدول برای جستجو کردن نام Workload Group، نباید تا حد زیادی تحت تاثیر قرار گیرد.

6- نظارت بر Resource Governor
با استفاده از Resource Pool و Workload Group Dynamic Management View (DMV) و Performance Monitor, events SQL Server:Workload Group Stats و SQL Server:Performance Monitor (Monitor) موجود است: Server:Resource Pool Stats شکل زیر مربوط به پیکر بندی مثال مورد نظرمان می‌باشد:



7- نتیجه گیری

چندین مزیت بالقوه ارائه می‌دهد، در درجه اول قابلیت اولویت بندی منابع Server برای کاربران و برنامه‌های کاربردی (applications) بحرانی، جلوگیری از "runaway" یا درخواست‌های غیرمنتظره‌ای که به شدت و بطور قابل توجهی روی کارائی Server تاثیر منفی می‌گذارند.

ضمناً Resource Governor چندین مشکل بالقوه نیز عرضه می‌کند، برای مثال پیکربندی اشتباہ Resource Governor تنها به عملکرد کلی Server آسیب نمی‌رساند بلکه به طور بالقوه روی سرور قفل (Lock) می‌تواند ایجاد کند و نیاز به استفاده از اتصال اختصاصی Administrator برای متصل شدن به SQL Server به منظور اشکال یابی و رفع مشکل می‌باشد. بنابراین توصیه شده است که تنها در صورتی که DBA با تجربه‌ای هستید و درک خوب و آشنائی خوبی با Workload هایی که روی بانک اطلاعاتی اجرا می‌شوند دارید، Resource Governor را بکار ببرید. حتی در این صورت، ضروری است که پیکربندی تان را روی یک Server تستی پیش از اینکه روی محیط تولیدی بگسترانید، تست نمائید.

Resource Governor به عنوان یک ویژگی با نام تجاری جدید در SQL Server 2008، با تعدادی محدودیت همراه است که احتمالاً در نسخه‌های بعدی SQL Server حذف خواهد شد، از محدودیت‌های بارز :

- محدودیت منابع (Resource), که به CPU و حافظه محدود می‌شوند. Disk I/O و منابع شبکه را در SQL Server 2008 نمی‌توان محدود کرد.
- استفاده از منابع برای Integration Service, Analysis Service و Reporting Service را نمی‌توان محدود کرد . در این نسخه محدودیت‌های منابع تنها روی هسته موتور بانک اطلاعاتی بکار برده می‌شود.
- محدودیت‌های SQL Instance روی یک Resource Governor محدود نمی‌شود.

نظرات خوانندگان

نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۸/۲۱

در SQL Server 2012 به منظور تضمین عملکرد تعداد پشتیبانی از مخازن منابع از 20 عدد به 64 عدد افزایش یافته است. همچنین در SQL Server 2014 پشتیبانی از O/I نیز اضافه گردید. (تا پیش از ارائه نسخه 2014 محدودیت روی منابع تنها به CPU و حافظه خلاصه می‌شد)

مقدمه SQL Server، با هر تقاضا به عنوان یک واحد مستقل رفتار می‌کند. در وضعیت‌های پیچیده‌ای که فعالیت‌ها توسط مجموعه‌ای از دستورات SQL انجام می‌شود، به طوری که یا همه باید اجرا شوند یا هیچکدام اجرا نشوند، این روش مناسب نیست. در چنین وضعیت‌هایی، نه تنها تقاضاهای موجود در یک دنباله به یکدیگر بستگی دارند، بلکه شکست یکی از تقاضاهای موجود در دنباله، به معنای این است که کل تقاضاهای موجود در دنباله باید لغو شوند، و تغییرات حاصل از تقاضاهای اجرانشده در آن دنباله خنثی شوند تا بانک اطلاعاتی به حالت قبلی برگردد.

1- تراکنش چیست؟ تراکنش شامل مجموعه‌ای از یک یا چند دستور SQL است که به عنوان یک واحد عمل می‌کنند. اگر یک دستور SQL در این واحد با موفقیت اجرا نشود، کل آن واحد خنثی می‌شود و داده‌هایی که در اجرای آن واحد تغییر کرده اند، به حالت اول برگردانده می‌شود. بنابراین تراکنش وقتی موفق است که هر یک از دستورات آن با موفقیت اجرا شوند. برای درک مفهوم تراکنش مثال زیر را در نظر بگیرید: سهامدار A در معامله‌ای 400 سهم از شرکتی را به سهامدار B می‌فروشد. در این سیستم، معامله وقتی کامل می‌شود که حساب سهامدار A به اندازه 400 بدهکار و حساب سهامدار B همزمان به اندازه 400 بستانکار شود. اگر هر کدام از این مراحل با شکست مواجه شود، معامله انجام نمی‌شود.

2- خواص تراکنش هر تراکنش دارای چهار خاصیت است (معروف به ACID) که به شرح زیر می‌باشند:

2-1 خاصیت یکپارچگی (Atomicity) یکپارچگی به معنای این است که تراکنش باید به عنوان یک واحد منسجم (غیر قابل تفکیک) در نظر گرفته شود. در مثال مربوط به مبادله سهام، یکپارچگی به معنای این است که فروش سهام توسط سهامدار A و خرید آن سهام توسط سهامدار B، مستقل از هم قابل انجام نیستند و برای این که تراکنش کامل شود، هر دو عمل باید با موفقیت انجام شوند.

اجرای یکپارچه، یک عمل "همه یا هیچ" است. در عملیات یکپارچه، اگر هر کدام از دستورات موجود در تراکنش با شکست مواجه شوند، اجرای تمام دستورات قبلی خنثی می‌شود تا به جامعیت بانک اطلاعاتی آسیب نرسد.

2-2 خاصیت سازگاری (Consistency) سازگاری زمانی وجود دارد که هر تراکنش، سیستم را در یک حالت سازگار قرار دهد (چه تراکنش به طور کامل انجام شود و چه در اثر وجود خطای خنثی گردد). در مثال مبادله سهام، سازگاری به معنای آن است که هر بدهکاری مربوط به حساب فروشنده، موجب همان میزان بستانکاری در حساب خریدار می‌شود. در SQL Server، سازگاری با راهکار ثبت فایل سابقه انجام می‌گیرد که تمام تغییرات را در بانک اطلاعاتی ذخیره می‌کند و جزئیات را برای ترمیم تراکنش ثبت می‌نماید. اگر سیستم در اثنای اجرای تراکنش خراب شود، فرآیند ترمیم SQL Server با استفاده از این اطلاعات، تعیین می‌کند که آیا تراکنش با موفقیت انجام شده است یا خیر، و در صورت عدم موفقیت آن را خنثی می‌کند. خاصیت سازگاری تضمین می‌کند که بانک اطلاعاتی هیچگاه تراکنش‌های ناقص را نشان نمی‌دهد.

2-3 خاصیت تفکیک (Isolation) تفکیک موجب می‌شود هر تراکنش در فضای خودش و جدا از سایر تراکنش‌های دیگری که در سیستم انجام می‌گیرد، اجرا شود و نتایج هر تراکنش فقط در صورت کامل شدن آن قابل مشاهده است. اگر چندین تراکنش همزمان در سیستم در حال اجرا باشند، اصل تفکیک تضمین می‌کند که اثرات یک تراکنش تا کامل شدن آن، قابل مشاهده نیست. در مثال مربوط به مبادله سهام، اصل تفکیک به معنای این است که تراکنش بین دو سهامدار، مستقل از تمام تراکنش‌های دیگری است که در سیستم به مبادله سهام می‌پردازنند و اثر آن وقتی برای افراد قابل مشاهده است که آن تراکنش کامل شده باشد. این اصل در مواردی که سیستم همزمان از چندین کاربر پشتیبانی می‌کند، مفید است.

2-4 پایداری (Durability) پایداری به معنای این است که تغییرات حاصل از نهایی شدن تراکنش، حتی در صورت خرابی سیستم نیز پایدار می‌ماند. اغلب سیستم‌های مدیریت بانک اطلاعاتی رابطه‌ای، از طریق ثبت تمام فعالیت‌های تغییر دهنده‌ی داده‌ها در بانک اطلاعاتی، پایداری را تضمین می‌کنند. در صورت خرابی سیستم یا رسانه ذخیره سازی داده‌ها، سیستم قادر است آخرین

بهنگام سازی موفق را هنگام راه اندازی مجدد، بازیابی کند. در مثال مربوط به مبادله سهام، پایداری به معنای این است که وقتی منتقال سهام از سهامدار A به B با موفقیت انجام گردید، حتی اگر سیستم بعداً خراب شد، باید نتیجه‌ی آن را منعکس سازد.

3- مشکلات همزمانی (Concurrency Effects):

3-1- Dirty Read: زمانی روی می‌دهد که تراکنشی رکوردی را می‌خواند، که بخشی از تراکنشی است که هنوز تکمیل نشده است، اگر آن تراکنش Rollback شود اطلاعاتی از بانک اطلاعاتی دارید که هرگز روی نداده است. اگر سطح جداسازی تراکنش (پیش فرض) Read Committed باشد، این مشکل بوجود نمی‌آید.

3-2- Non-Repeatable Read: زمانی ایجاد می‌شود که رکوردی را دو بار در یک تراکنش می‌خوانید و در این اثنا یک تراکنش مجزای دیگر داده‌ها را تغییر می‌دهد. برای پیشگیری از این مسئله باید سطح جداسازی تراکنش برابر با Repeatable Read یا Serializable باشد.

3-3- Phantoms: با رکوردهای مرموزی سروکار داریم که گویی تحت تاثیر عبارات Update و Delete صادر شده قرار نگرفته اند. به طور خلاصه شخصی عبارت Insert را درست در زمانی که Update مان در حال اجرا بوده انجام داده است، و با توجه به اینکه ردیف جدیدی بوده و قفلی وجود نداشته، به خوبی انجام شده است. تنها چاره این مشکل تنظیم سطح Serializable است و در این صورت بهنگام رسانی‌های جداول نباید درون بخش Where قرار گیرد، در غیر این صورت Lock خواهد شد.

3-4- Lost Update: زمانی روی می‌دهد که یک Update به طور موفقیت آمیزی در بانک اطلاعاتی نوشته می‌شود، اما به طور اتفاقی توسط تراکنش دیگری بازنویسی می‌شود. راه حل این مشکل بستگی به کد شما دارد و بایست به نحوی تشخیص دهید، بین زمانی که داده‌ها را می‌خوانید و زمانی که می‌خواهید آنرا بهنگام کنید، اتصال دیگری رکورد شما را بهنگام کرده است.

4- منابع قابل قفل شدن 6 منبع قابل قفل شدن برای SQL Server وجود دارد و آن‌ها سلسله مراتبی را تشکیل می‌دهند. هر چه سطح قفل بالاتر باشد، Granularity کمتری دارد. در ترتیب آبشاری Granularity عبارتند از:

- **Database:** کل بانک اطلاعاتی قفل شده است، معمولاً طی تغییرات Schema بانک اطلاعاتی روی می‌دهد.
- **Table:** کل جدول قفل شده است، شامل همه اشیای مرتبط با جدول.
- **Extent:** کل Extent (متشکل از هشت Page) قفل شده است.
- **Page:** همه داده‌ها یا کلیدهای Index در آن Page قفل شده اند.
- **Key:** قفلی در کلید مشخصی یا مجموعه کلیدهایی Index وجود دارد. ممکن است سایر کلیدهای در همان Index Page تحت تاثیر قرار نگیرند.
- **Row or Row Identifier (RID):** هر چند قفل از لحاظ فنی در Row Identifier قرار می‌گیرد ولی اساساً کل ردیف را قفل می‌کند.

5- تسريع قفل (Lock Escalation) و تاثیرات قفل روی عملکرد اگر تعداد آیتم‌های قفل شده کم باشد نگهداری سطح بهتری از Granularity (مثلاً RID به جای Page) معنی دارد. هر چند با افزایش تعداد آیتم‌های قفل شده، سریار مرتبط با نگهداری آن قفل‌ها در واقع باعث کاهش عملکرد می‌شود، و می‌تواند باعث شود قفل به مدت طولانی‌تری در محل باشد (هر چه قفل به مدت طولانی‌تری در محل باشد، احتمال این که شخصی آن رکورد خاص را بخواهد بیشتر است). هنگامی که تعداد قفل نگهداری شده به آستانه خاصی برسد آن گاه قفل به بالاترین سطح بعدی افزایش می‌یابد و قفل‌های سطح پایین‌تر نباید به شدت مدیریت شوند (آزاد کردن منابع و کمک به سرعت در مجادله). توجه شود که تسريع مبنی بر تعداد قفل هاست و نه تعداد کاربران.

6- حالات قفل (Lock Modes): همانطور که دامنه وسیعی از منابع برای قفل شدن وجود دارد، دامنه‌ای از حالات قفل نیز وجود دارد.

6-1- Shared Locks (S): زمانی استفاده می‌شود، که فقط باید داده‌ها را بخوانید، یعنی هیچ تغییری ایجاد نخواهد کرد. Shared Lock با سایر Lock‌های دیگر سازگار است، البته قفل‌های Shared Lock دیگری هستند که با Shared Lock سازگار نیستند. یکی از کارهای Shared Lock انجام می‌دهد، ممانعت از انجام Dirty Read از طرف کاربران است.

6-2- Exclusive Locks (X): این قفل‌ها با هیچ قفل دیگری سازگار نیستند. اگر قفل دیگری وجود داشته باشد، نمی‌توان به Exclusive Lock دست یافت و همچنین در حالی که Exclusive Lock فعال باشد، به هر قفل جدیدی از هر شکل اجازه ایجاد شدن در منبع را نمی‌دهند. این قفل از اینکه دو نفر همزمان به حذف کردن، بهنگام رسانی و یا هر کار دیگری مبادرت ورزند، پیشگیری می‌کند.

6-3- Update Locks (U): این قفل‌ها نوعی پیوند میان Shared Locks و Exclusive Locks هستند. برای انجام Update باید بخش Where (در صورت وجود) تایید اعتبار کنید، تا دریابید فقط چه ردیف‌هایی را می‌خواهید بهنگام رسانی کنید. این بدان معنی است که فقط به Shared Lock نیاز دارید، تا زمانی که واقعاً بهنگام رسانی فیزیکی را انجام دهید. در زمان بهنگام سازی فیزیکی نیاز به Exclusive Lock دارید. Update Lock نشان دهنده این واقعیت است که دو مرحله مجزا در بهنگام رسانی وجود دارد، Shared Lock ای دارید که در حال تبدیل شدن به Exclusive Lock است. Update Lock تمامی Update Lock‌های دیگر را از تولید شدن باز می‌دارند، و همچنین فقط با Intent Shared Lock و Shared Lock ها سازگار هستند.

6-4- Intent Locks: با سلسله مراتب شی سر و کار دارد. بدون Intent Lock، اشیای سطح بالاتر نمی‌دانند چه قفلی را در سطح پایین‌تر داشته اید. این قفل‌ها کارایی را افزایش می‌دهند و 3 نوع هستند:

6-4-1- Intent Shared Lock (IS: Shared Lock): در نقطه پایین‌تری در سلسله مراتب، تولید شده یا در شرف تولید است. این نوع قفل تنها به Page و Table اعمال می‌شود.

6-4-2- Intent Exclusive Lock (IX): همانند Intent Shared Lock است اما در شرف قرار گرفتن در آیتم سطح پایین‌تر است.

6-4-3- Shared With Intent Exclusive (SIX: Shared Lock): در پایین سلسله مراتب شی تولید شده یا در شرف تولید است اما قصد اصلاح داده‌ها را دارد بنابراین در نقطه مشخصی تبدیل به Intent Exclusive Lock می‌شود.

6-5- Schema Locks: به دو شکل هستند:

6-5-1- Schema Modification Lock (Sch-M): تغییر Schema به شی اعمال شده است. هیچ پرس و جویی یا سایر عبارت‌های نمی‌توانند در مورد این شی در مدت قفل Sch-M اجرا شوند. با همه حالات قفل ناسازگار است.

6-5-2- Schema Stability Lock (Sch-S): بسیار شبیه به Shared Lock است، هدف اصلی این قفل پیشگیری از Sch-M است وقتی که قبل از قفل هایی برای سایر پرس و جو-ها (یا عبارت‌های Create، Alter و Drop) در شی فعال شده اند. این قفل با تمامی انواع دیگر قفل سازگار است به جزء Sch-M.

6-6- Bulk Update Locks (BU): این قفل‌ها بارگذاری موازی داده‌ها را امکان پذیر می‌کنند، یعنی جدول در مورد هر فعالیت نرمال (عبارات T-SQL) قفل می‌شود، اما چندین عمل bcp یا Bulk Insert را می‌توان در همان زمان انجام داد. این قفل فقط با Sch-S و سایر قفل‌های BU سازگار است.

7- سطوح جداسازی (Isolation Level):

7-1- Read Committed (وضعيت پیش فرض): با Shared Lock همه Read Committed های ایجاد شده، به محض اینکه عبارت ایجاد کننده آنها تکمیل شود، به طور خودکار آزاد می‌شوند. به طور خلاصه قفل‌های مرتبط با عبارت Select به محض تکمیل عبارت آزاد می‌شوند و SQL Server منتظر پایان تراکنش نمی‌ماند. اگر تراکنش پرس و جویی را انجام می‌دهد که داده‌ها را اصلاح می‌کند (Insert، Update و Delete) قفل‌ها برای مدت تراکنش نگه داشته می‌شوند.

با این سطح پیش فرض، می‌توانید مطمئن شوید جامعیت کافی برای پیشگیری از Dirty Read دارید، اما همچنان Non-Repeatable Read می‌تواند روی دهد.

Read Uncommitted 7-2-: خطرناک‌ترین گزینه از میان تمامی گزینه‌ها است، اما بالاترین عملکرد را به لحاظ سرعت دارد. در واقع با این تنظیم سطح تجربه همه مسائل متعدد هم زمانی مانند Dirty Read امکان پذیر است. در واقع با تنظیم این سطح به اعلام می‌کنیم هیچ قفلی را تنظیم نکرده و به هیچ قفلی اعتنا نکند، بنابراین هیچ تراکنش دیگری را مسدود نمی‌کنیم. می‌توانید همین اثر Read Uncommitted را با اضافه کردن نکته بهینه ساز NOLOCK در پرس و جوها بدست آورید.

Repeatable Read 7-3-: سطح جداسازی را تا حدودی افزایش می‌دهد و سطح اضافی محافظت همزمانی را با پیشگیری از Dirty Read و همچنین Non-Repeatable Read فراهم می‌کند. پیشگیری از Non-Repeatable Read بسیار مفید است اما حتی نگه داشتن Shared Lock تا زمان پایان تراکنش می‌تواند دسترسی کاربران به اشیا را مسدود کند، بنابراین به بهره وری لطمه وارد می‌کند. نکته بهینه ساز برای این سطح REPEATABLEREAD است.

Serializable 7-4-: این سطح از تمام مسائل هم زمانی پیشگیری می‌کند به جز برای Lost Update. این تنظیم سطح به واقع بالاترین سطح آنچه را که سازگاری نامیده می‌شود، برای پایگاه داده فراهم می‌کند. در واقع فرآیند بهنگام رسانی برای کاربران مختلف به طور یکسان عمل می‌کند به گونه‌ای که اگر همه کاربران یک تراکنش را در یک زمان اجرا می‌کردند، این گونه می‌شد «پردازش امور به طور سریالی». با استفاده از نکته بهینه ساز HOLDLOCK یا SERIALIZABLE در پرس و جو شیوه سازی می‌شود.

Snapshot 7-5-: جدترین سطح جداسازی است که در نسخه 2005 اضافه شد، که شبیه ترکیبی از Read و Read Committed Uncommitted است. به طور پیش فرض در دسترس نیست، در صورتی در دسترس است که گزینه ALLOW_SNAPSHOT_ISOLATION برای بانک اطلاعاتی فعال شده باشد. (برای هر بانک اطلاعاتی موجود در تراکنش Snapshot مشابه Read Uncommitted هیچ قفلی ایجاد نمی‌کند. تفاوت اصلی آن‌ها در این است که تغییرات صورت گرفته در بانک اطلاعاتی را در زمان‌های متفاوت تشخیص می‌دهند. هر تغییر در بانک اطلاعاتی بدون توجه به زمان یا Commit شدن آن، توسط پرس و جو هایی که سطح جداسازی Read Uncommitted را اجرا می‌کنند، دیده می‌شود. با Snapshot فقط تغییراتی که قبل از شروع تراکنش، Commit شده اند، مشاهده می‌شود. از شروع تراکنش Commit، تمامی داده‌ها دقیقاً مشاهده می‌شوند، زیرا در شروع تراکنش Commit شده اند. نکته: در حالی که Snapshot توجهی به قفل‌ها و تنظیمات آنها ندارد، یک حالت خاص وجود دارد. چنانچه هنگام انجام Snapshot یک عمل Rollback (بازیافت) بانک اطلاعاتی در جریان باشد، تراکنش Snapshot قفل‌های خاصی را برای عمل کردن به عنوان یک مکان نگهدار و سپس انتظار برای تکمیل Rollback تنظیم می‌کند. به محض تکمیل Rollback، قفل حذف شده و Snapshot به طور طبیعی به جلو حرکت خواهد کرد).

Isolation level	Dirty read	Non-Repeatable read	Phantom
Read uncommitted	Yes	Yes	Yes
Read committed	No	Yes	Yes
Repeatable read	No	No	Yes
Snapshot	No	No	No
Serializable	No	No	No

نظرات خوانندگان

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۲/۰۵ ۱۲:۳۷

سلام ، خسته نباشید. ممنون از زحماتی که برای این مقاله کشیدید.
من یه سوال داشتم درباره تراکنش (Transaction) : من توی پروژه ام از isolation و sqlTransaction سریالایز استفاده کدم
در این بلاک، ۹ جدول من مورد عملیات درج و یا ویرایش قرار می‌گیره (و البته دو وب سرویس نیز صدا زده می‌شه) و معمولاً ۲ یا
۳ ثانیه برای ثبت این تراکنش زمان صرف می‌شه . تا اینجا مشکلی نیست ولی اگر چند کاربر مثل ۵ نفر این عملیات انجام بدن در
بعضی از مواقع (به ندرت) به ترتیب عملیات برای هر نفر انجام می‌شه (همین توقع از سریالایز می‌رمه) . ولی در اکثر مواقع خطأ
deadlock می‌ده و یا خطأ timeout و همچنین جاهای مختلف برنامه که به یکی از این ۹ جدول select می‌شه به انها هم همین
خطاهای رومی ده . لطفا در صورت امکان راه حلی برای مشکل من بگین . ممنون

نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۲/۰۵ ۱۵:۴۵

با سلام و سپاس از محبت تان، چند نکته به ذهن من میرسد، که شاید راهگشا باشد:
- تعداد دستورات درون بلاک Transaction تا جایی که میسر است، کمینه باشد.
- عملیات مربوط به وب سرویس، خارج از Transaction انجام گیرد. (به عنوان یک نکته در کار با Transaction باید عملیات
Input و Output خارج از Transaction انجام گیرد).
- طبعتا هر چه درجه Isolation بالاتر باشد، Lock سختگیرانه‌تر برخورد می‌کند.
- برای مشکل بن بست، اگر می‌توانید ترتیب در اختیار گرفتن جداول را (پس از شناسائی محل بن بست) تغییر دهید و یا از جداول
Temp استفاده کنید.
- چنانچه در قسمت هایی از برنامه همانطور که اشاره کردید فقط نیاز به خواندن مقادیر جداول دارید، از بهینه ساز پرس و جوی
nolock استفاده کنید. برای مثال:

```
select * from tbl with (NOLOCK)
```

استفاده از DDL Trigger امکان ایجاد Trigger برای عملیات DDL (Data Definition Language) در SQL Server 2005 از Trigger گردید. عملیاتی مانند ایجاد یک جدول جدید در بانک اطلاعاتی، اضافه شدن یک Login جدید و یا ایجاد یک بانک اطلاعاتی جدید را به وسیله این نوع Trigger می‌توان کنترل نمود. در حقیقت Trigger DDL به شما اجازه می‌دهد که از تاثیر تعدادی از دستورات DDL جلوگیری کنید. بدین ترتیب که تقریباً هر دستور DDL به طور خودکار، تراکنشی (Transactional) اجرا می‌شود. می‌توان با دستور ROLLBACK TRANSACTION اجرای دستور DDL را لغو نمود. توجه شود همه دستورات DDL به صورت تراکنشی اجرا نمی‌شوند، به عنوان مثال دستور ALTER DATABASE ممکن است Database را تغییر دهد. در این صورت ساختار فایلی Database را تغییر می‌دهد، از آنجائی که سیستم عامل ویندوز به صورت تراکنشی عمل نمی‌کند بنابراین شما نمی‌توانید این عمل فایل سیستمی را لغو نمایید. به هر حال شما می‌توانید Trigger را با ALTER DATABASE فعال (fire) کنید برای عملیات Auditing، ولی نمی‌توان از انجام عمل ALTER DATABASE جلوگیری کرد.

برای نمونه می‌خواهیم از حذف و یا تغییر جداول یک بانک اطلاعاتی که به صورت عملیاتی در حال سرویس دهی است جلوگیری کنیم، برای اینکار از دستورهای زیر استفاده می‌کنیم:

```
create trigger Prevent_AlterDrop
on database
for drop_table, alter_table
as
  print 'table can not be dropped or altered'
  rollback transaction
```

از عبارت ON برای مشخص کردن محدوده Trigger در سطح SQL Instance (در این صورت ON All SERVER نوشته می‌شود) و یا در سطح Database (در این حالت ON DATABASE نوشته می‌شود) استفاده می‌شود و از عبارت FOR برای مشخص کردن رویداد یا گروه رویدادی که سبب فراخوانی Trigger می‌شود، استفاده خواهد شد.

1- معرفی تابع EVENTDATA() این تابع، یک تابع سیستمی مهم است که در DDL Trigger استفاده می‌شود. در حالیکه در هر سطحی فعال (fire) شود تابع سیستمی (raise) EVENTDATA() فراخوانی (raise) می‌شود. خروجی تابع [در قالب XML است](#). می‌توان اطلاعات را از تابع EVENTDATA دریافت کرد و آنها را در یک جدول با فیلڈی از جنس XML و یا با استفاده از XPath Query ثبت کرد (Logging). عناصر کلیدی (Key Elements) تابع EVENTDATA به شرح زیر است:

- **EventType**: نوع رویدادی که باعث فراخوانی Trigger شده است.
- **PostTime**: زمانی که رویداد رخ می‌دهد.
- **SPIID**: کاربری که باعث ایجاد رویداد شده است.
- **ServerName**: نام SQL Instance که رویداد در آن رخداده است.
- **LoginName**: نام Login که عمل مربوط به وقوع رویداد را اجرا می‌کند.
- **UserName**: نام User که عمل مربوط به وقوع رویداد را اجرا می‌کند.
- **DatabaseName**: نام Database که رویداد در آن رخداده است.
- **ObjectType**: نوع Object که اصلاح، حذف و یا ایجاد شده است.
- **ObjectName**: نام Object که اصلاح، حذف و یا ایجاد شده است.
- **TSQLCommand**: دستور T-SQL که اجرا شده و باعث اجرا شدن Trigger شده است.

2- بررسی یک سناریو نمونه
برای نمونه در دستورات زیر جدولی با نام ddl_log

```
CREATE TABLE ddl_log
(
  EventType nvarchar(100),
  PostTime datetime,
```

معرفی و استفاده از DDL Triggers در SQL Server

```
SPID nvarchar(100),
ServerName nvarchar(100),
LoginName nvarchar(100),
UserName nvarchar(100),
DatabaseName nvarchar(100),
ObjectName nvarchar(100),
ObjectType nvarchar(100),
DefaultSchema nvarchar(100),
[SID] nvarchar(100),
TSQLCommand nvarchar(2000));
```

و یک Trigger با نام log برای رویدادهایی که در سطح Database رخ می‌دهد، ایجاد می‌کنیم.

```
CREATE TRIGGER [Log] ON DATABASE
FOR DDL_DATABASE_LEVEL_EVENTS
AS
DECLARE @data XML
SET @data = EVENTDATA()
INSERT INTO ddl_log
VALUES (
    @data.value('(/EVENT_INSTANCE/EventType)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/PostTime)[1]', 'datetime'),
    @data.value('(/EVENT_INSTANCE/SPID)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/ServerName)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/LoginName)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/UserName)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/DatabaseName)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/ObjectName)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/ObjectType)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/DefaultSchema)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/SID)[1]', 'nvarchar(100)'),
    @data.value('(/EVENT_INSTANCE/TSQLCommand)[1]', 'nvarchar(2000)'));
```

نمونه ای از مقادیر ذخیره شده در جدول ddl_log به شکل زیر خواهد بود:

EventType	PostTime	SPID	ServerName	LoginName	UserName	DatabaseName	ObjectName	
CREATE_TABLE	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	Tmp_TestTable	
ALTER_TABLE	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	Tmp_TestTable	
DROP_TABLE	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	TestTable	
RENAME	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	Tmp_TestTable	
DROP_TABLE	11/7/2010 10:38:26 AM	52	1936-603102405\SQL2008	sa	dbo	Test	TestTable	
DefaultSchema	SID	TSQLCComm and						
NULL	NULL	CREATE TABLE dbo.Tmp_TestTable (a nvarchar(50) NULL) ON [PRIMARY]						
NULL	NULL	ALTER TABLE dbo.Tmp_TestTable SET (LOCK_ESCALATION = TABLE)						
NULL	NULL	DROP TABLE dbo.TestTable						
NULL	NULL	EXECUTE sp_rename N'dbo.Tmp_TestTable', N'TestTable', 'OBJECT'						
NULL	NULL	DROP TABLE TestTable ;						

3- ملاحظات

در صورت فعال شدن Trigger می‌توان برخی موارد محدودیت زمانی، کاربر اجرا کننده و ... را اضافه نمود. برای مثال در دستور زیر اجازه تغییرات در این زمان (بین 7:00 A.M تا 8:00 P.M) امکان پذیر نیست و در صورت اقدام پیغام خطای دریافت می‌کنید و دستورات Create لغو خواهند شد و اگر خارج از زمان فوق دستورات DDL را اجرا کنید دستورات به طور موفقیت آمیز اجرا می‌شود و البته تغییرات نیز Log می‌شوند.

```
IF DATEPART(hh,GETDATE()) > 7 AND DATEPART(hh,GETDATE()) < 20
BEGIN
    RAISERROR ('You can only perform this change between 8PM and 7AM. Please try
this change again or contact Production support for an override.', 16, 1)
ROLLBACK
END
```

این Trigger تاثیرات کمی بر روی کارایی دارد به این دلیل که معمولاً رویدادهای DDL به ندرت رخ می‌دهد. می‌توانید هنگامی که قصد دارید دستورات DDL را اجرا کنید موقتاً Trigger را با دستورات زیر غیر فعال نمایید:

```
DISABLE TRIGGER ALL ON DATABASE
DISABLE TRIGGER ALL ON ALL SERVER
```

پس از Overriding کردن می‌توانید مجدداً Trigger را فعال کنید:

```
ENABLE TRIGGER ALL ON DATABASE
ENABLE TRIGGER ALL ON ALL SERVER
```

4- معرفی DDL Event Groups

برای مشاهده جزئیات بیشتر می‌توانید به این [لينک](#) مراجعه کنید.

Server Level

DDL_SERVER_LEVEL_EVENTS

DDL_LINKED_SERVER_EVENTS

DDL_LINKED_SERVER_LOGIN_EVENTS

DDL_REMOTE_SERVER_EVENTS

DDL_EXTENDED_PROCEDURE_EVENTS

DDL_MESSAGE_EVENTS

DDL_ENDPOINT_EVENTS

DDL_SERVER_SECURITY_EVENTS



DDL_PARTITION_SCHEME_EVENTS

DDL_SSB_EVENTS

DDL_MESSAGE_TYPE_EVENTS

DDL_CONTRACT_EVENTS

DDL_QUEUE_EVENTS

DDL_SERVER_EVENTS

DDL_ROUTE_EVENTS

DDL_REMOTE_SERVICE_BINDING_EVENTS

DDL_XML_SCHEMA_COLECTION_EVENTS

DDL_FULLTEXT_CATALOG_EVENTS

DDL_DEFAULT_EVENTS

DDL_EXTENDED_PROPERTY_EVENTS

DDL_PLAN_GUIDE_EVENTS

DDL_RULE_EVENTS

DDL_SYNONYM_EVENTS

DDL_EVENT_NOTIFICATION_EVENTS

DDL_ASSEMBLY_EVENTS

DDL_TYPE_EVENTS

نظرات خوانندگان

نویسنده: محمد ۱۳۹۳/۰۶/۲۵
تاریخ: ۲۰:۴۱

با سلام

اول از همه تشکر میکنم از شما دوست عزیز بابت زحماتی که کشیدید.

سئوال بنده اینه اگر بخواهیم خودمون یک نوع TRIGGER جدید به Database اضافه کنیم به چه صورت امکان پذیره؟
به طور مثال من میخوام یک تریگر فقط برای کلیه فعالیت‌ها در حوزه‌ی بکاپ و ریستور بنویسم.
مثل تریگر Drop_table یا Alter_table که ماهیتشون از قبل تعریف شده است.

با تشکر

نویسنده: محمد رجب ۱۳۹۳/۰۷/۰۲
تاریخ: ۱۰:۳۴

با سلام و احترام؛ همانطور که در متن به عرض رسانده شده:

"از عبارت ON برای مشخص کردن محدوده Trigger در سطح SQL Instance (در این صورت ON ALL SERVER) در این حالت Database (در این حالت ON DATABASE) استفاده می‌شود و از عبارت FOR برای مشخص کردن رویداد یا گروه رویدادی که سبب فراخوانی Trigger می‌شود، استفاده خواهد شد."

در خصوص مثالی که اشاره کردید، به نظرم می‌رسد از Trigger برای این منظور استفاده نمی‌شود (در حوزه‌ی بکاپ و ریستور)، شاید اگر قصدتان به منظور ثبت log و ... باشد استفاده از Auditing کنید. به این منظور در Audit با توجه به جدول زیر می‌توان اقدام به ثبت موارد نمود:

Server-Level Audit Action Groups

Description	Event Class	Action group name
یک دستور Restore یا Backup صادر شود	Audit Backup/Restore	BACKUP_RESTORE_GROUP

به طور مختصر Auditing به شرح زیر است:

SQL Server Audit

بازبینی (Auditing) شامل پیگیری و ثبت رویدادهای است که در سطح Database و SQL Instance یا های روی یک سیستم اتفاق می‌افتد. چندین سطح برای Auditing در SQL Server وجود دارد که به صلاحیت و نیازمندی‌های نصب شما وابسته است. شما می‌توانید گروه اقدامات بازبینی سرویس دهنده (server audit action groups) را به ازای هر SQL Instance و گروه اقدامات بازبینی بانک اطلاعاتی (database audit action groups) را به ازای هر بانک اطلاعاتی ثبت کنید. رویداد Audit هر زمان عملی که مورد رسیدگی قرار گرفته اتفاق افتاد، رخ می‌دهد.

تا پیش از SQL SERVER 2008، شما باید از خصیصه‌های متعددی برای انجام یک مجموعه کامل بازبینی (Auditing) برای نمونه DDL Trigger، SQL Trace و Trigger، DML Trigger، SQL Instance، بر روی یک Audit specification استفاده می‌کردید.

SQL SERVER 2008، همه قابلیت‌های Auditing را روی یک Audit Specification ترکیب می‌کند. Audit Specification با تعريف یک شی بازبینی (audit object) در سطح سرویس دهنده برای ثبت (logging) یک دنباله بازبینی (audit trial) آغاز می‌شود. توجه شود که بازبینی (audit object) در سطح سرویس دهنده برای ثبت (logging) یک دنباله بازبینی (audit trial) آغاز می‌شود. Database Audit Specification و Server Audit Specification ایجاد کنید.

Server Audit Specification، گروه اقدامات در سطح سرویس دهنده را جمع آوری می‌کند که با رویدادهای وسیعی فعال می‌شوند، این گروه اقدامات تحت عنوان Server-Level Audit Action Groups تشریح شده اند. شما می‌توانید یک Server Audit Specification را به ازای هر Audit ایجاد کنید چرا که هر دو در محدوده یک SQL Instance ایجاد می‌شوند.

Database Audit Specification، گروه اقدامات در سطح بانک اطلاعاتی را جمع آوری می‌کند که با رویدادهای وسیعی فعال می‌شود. این گروه اقدامات تحت عنوان های Database-Level Audit Actions و Database-Level Audit Action Groups تشریح

شده اند. می‌توانید یک Audit را به ازای هر Database در بانک اطلاعاتی SQL Server ایجاد کنید. همچنین می‌توانید هر گروه اقدامات بازبینی (audit action groups) یا رویدادهای بازبینی (audit events) را به یک Database Audit Specification اضافه کنید. گروه اقدامات بازبینی، گروه اقدامات از پیش تعریف شده ای هستند و رویدادهای بازبینی اقدامات تجزیه ناپذیری هستند که توسط موتور بانک اطلاعاتی مورد رسیدگی قرار می‌گیرند، هر دو در محدوده بانک اطلاعاتی (Database) هستند. این اقدامات برای Audit فرستاده می‌شوند تا در Target Windows Security Log (Database) باشد. Windows Application Log سرویس دهنده شما به Local System، Local Policy، Generate security audits و Network Service دریافت خواهد کرد.

(گروه اقدامات بازبینی در سطح سرویس دهنده) Server-Level Audit Action Groups

این گروه اقدامات به گروه رویداد Security Audit شبیه هستند. به طور خلاصه این گروه اقدامات، اقداماتی را که در یک SQL Instance شامل می‌شوند، در بر می‌گیرد. برای مثال اگر گروه اقدام مناسب با Schema اضافه شده باشد هر شیء در هر Schema که مورد دستیابی قرار می‌گیرد، ثبت می‌شود. اقدامات در سطح سرویس دهنده به شما اجازه نمی‌دهد که جزئیات اقدامات در سطح بانک اطلاعاتی را فیلتر کنید. یک بازبینی در سطح بانک اطلاعاتی برای انجام به جزئیات دقیق فیلتر کردن نیاز دارد، برای مثال اجرای دستور Select برای login Customers یا Employee در گروه Audit هستند.

(گروه اقدامات بازبینی در سطح بانک اطلاعاتی) Database-Level Audit Action Groups

این گروه اعمال به کلاس‌های رویداد Security Audit شبیه هستند.

Database-Level Audit Actions

اقدامات در سطح بانک اطلاعاتی، اقدامات بازبینی خاصی را به طور مستقیم روی Database، Schema و اشیاء (از قبیل View، رویه‌های ذخیره شده، توابع و ...) فراهم می‌کند. این اقدامات برای فیلد (Columns) صدق نمی‌کنند.

Audit-Level Audit Action Groups

شما می‌توانید اقداماتی را که در فرآیند Auditing هستند، بازبینی کنید که می‌تواند در محدوده سرویس دهنده یا بانک اطلاعاتی باشد. در محدوده بانک اطلاعاتی تنها برای database audit specification رخ می‌دهد. جهت بررسی بیشتر به این [لينك](#) مراجعه شود.

نویسنده: MF

تاریخ: ۱۷:۷ ۱۳۹۳/۰۷/۰۲

INSTEAD OF TRIGGERS در تریگر استفاده نمود ؟

نویسنده: محمد رجبی

تاریخ: ۱۲:۱ ۱۳۹۳/۰۷/۰۳

مواردی که اشاره کردید مربوط به DML Triggers می‌باشند.

برای اطلاعات بیشتر به این لینک [Understanding DDL Triggers vs. DML Triggers](#) مراجعه شود.

نویسنده: مجید فاضلی

تاریخ: ۱۲:۴۶ ۱۳۹۳/۰۷/۰۶

باید از از حالت INSTEAD OF استفاده کنیم در DML Trigger ای که قراره نوشته بشه.

می‌توانیم در یک جدول از دیتابیس مان بر اساس یک شرط خاص، عملیات Insert, Delete, Update، عملیات Insert, Delete, Update را مدیریت کنیم. بعنوان مثال در قطعه کد زیر ما قبل از عملیات Insert در جدول tb1Test چک می‌کنیم که اگر مقدار ستون FirstName برابر با null بود عملیات Insert آن رکورد در دیتابیس لغو شود.

```
ALTER TRIGGER [dbo].[Prevent_Befor_Insert_Null]
ON [dbTest].[dbo].[tblTest]
INSTEAD OF INSERT
AS
BEGIN
```

```
SET NOCOUNT ON
IF OBJECT_ID(N'dbTest.dbo.tblTest.FirstName') IS NULL
BEGIN
DECLARE @Id INT
SET @Id = (SELECT Id FROM inserted)
RAISERROR ('Name must be at least 16 characters long.', 16, 1)
ROLLBACK
END
END
```

از دو طریق می‌توان به مقادیر فیلدهای رکورد جاری دسترس داشت:

1- استفاده از OBJECT_ID و ذکر نام فیلد مورد نظر

2- گرفتن فیلد مورد نظر از جدول INSERTED یا DELETED

Trigger DML ها دارای دو جدول خاص بنامهای INSERTED و DELETED هستند که توسط خود SQL Server مدیریت می‌شوند. در حقیقت در پشت صحنه، ما با این دو جدول در هنگام تغییر مقادیر داده‌های جداول دیتابیس کار می‌کنیم و نمی‌توانیم بصورت مستقیم داده‌های جداول موجود در دیتا بیس مان را تغییر دهیم.

جدول DELETED و INSERTED حاوی رکورد جاری است که تحت تاثیر عمل درج، ویرایش و حذف در دیتابیس قرار گرفته است.

اطلاعات بیشتر در [اینجا](#) و [اینجا](#)

نویسنده: محمد

تاریخ: ۱۵ شهریور ۱۳۹۳

سلام؛ تشکر از توضیحات شما. اجازه بدهید من طور دیگری سؤالم رو مطرح کنم. به طور مثال ما برای کار با تاریخ شمسی در SQL چندین روش پیش رو داریم که وارد جزئیات آن نمی‌شوم ولی یکی از این روش‌ها که به خوبی جواب میدهد استفاده از CLR است که ما با توسط این قابلیت می‌توانیم یک نوع دیتا تایپ جدید، با ماهیت جدید در اس کیو ال اضافه کنیم. حالا منظور بندۀ این است که آیا برای تریگرها هم می‌شود این کار را انجام داد یا خیر؟ مثلاً توسط CLR یا هر روش دیگری که وجود دارد، ما بیاییم و یک نوع تریگر کاملاً جدید و Customize شده برای خودمان درست کنیم. به طور مثال: زمانی که کاربر از دیتابیس بخواهد بکاپ تهیه کند یا آن را ریستور کند، یکسری فعالیتها به آن فعالیت اضافه شود یا در راستای آن انجام شود. با تشکر

1- اندازه گیری تعداد Transaction‌ها در واحد زمان روی یک Database خاص در SQL Server جهت بدست آوردن تعداد Transaction‌ها در واحد زمان (Transactions Per Second) روی یک Database خاص در یک سیستم عملیاتی، جهت ارتقاء سخت افزاری، تست فشار و ... می‌توانید از یک DMV با نام [sys.dm_os_performance_counters](#) به طریق زیر استفاده نمایید:

```

declare @cntr_value bigint
Select @cntr_value=cntr_value
from sys.dm_os_performance_counters
where instance_name='AdventureWorks' and
counter_name='Write Transactions/sec'
/* ایجاد یک تاخیر مثلثی یک ثانیه */
waitfor delay '00:00:01'

Select cntr_value -@cntr_value
from sys.dm_os_performance_counters
where instance_name='AdventureWorks' and
counter_name='Write Transactions/sec'
  
```

View معرفی شده تمامی شمارندهای عملکردی را برای یک Instance شامل می‌شود، ستون instance_name برابر نام بانک اطلاعاتی مورد نظر می‌باشد.

sys.sp_MSforeachtable 2-

از رویه‌های ذخیره شده UnDocumented در SQL Server می‌باشد و این قابلیت را دارا است که برای هر یک از جداول موجود در یک بانک اطلاعاتی، یک رویه‌ای را اجرا کند. برای مثال با استفاده از دستور زیر، می‌توانید تعداد سطرها، اندازه‌ی داده‌ها و ایندکس‌های یک جدول را بدست آورید

```
EXEC sys.sp_MSforeachtable 'sp_spaceused ''?''';
```

به عنوان یک مثال کاربردی، با اجرای دستور زیر می‌توان جداول بانک اطلاعاتی مورد نظرتان را از لحاظ معیارهایی که پیشتر ذکر آن رفت، مورد بررسی قرار دهید.

```

USE [AdventureWorksDW2008R2]
GO

CREATE TABLE #TableSpaceUsed(
[name] [nvarchar](120) NULL,
[rows] [nvarchar](120) NULL,
[reserved] [nvarchar](120) NULL,
[data] [nvarchar](120) NULL,
[index_size] [nvarchar](120) NULL,
[unused] [nvarchar](120) NULL
) ON [PRIMARY]

Insert Into #TableSpaceUsed
EXEC sys.sp_MSforeachtable 'sp_spaceused ''?'''';

Select * from #TableSpaceUsed
Order by CAST([rows] as int) desc

Drop table #TableSpaceUsed
  
```

خروجی مثال فوق به شکل زیر است.

	name	rows	reserved	data	index_size	unused
1	FactInternetSalesReason	64515	2208 KB	2048 KB	48 KB	112 KB
2	FactResellerSales	60855	25736 KB	11864 KB	13008 KB	864 KB
3	FactInternetSales	60398	20464 KB	8200 KB	11312 KB	952 KB
4	FactFinance	39409	2184 KB	2152 KB	8 KB	24 KB
5	DimCustomer	18484	8832 KB	7800 KB	904 KB	128 KB
6	FactAdditionalInternationalProductDescription	15168	4216 KB	4008 KB	64 KB	144 KB
7	FactCurrencyRate	14264	584 KB	472 KB	16 KB	96 KB
8	FactSurveyResponse	2727	352 KB	192 KB	160 KB	0 KB
9	ProspectiveBuyer	2059	728 KB	632 KB	88 KB	8 KB
10	DimDate	1188	232 KB	152 KB	48 KB	32 KB
11	DimReseller	701	280 KB	192 KB	88 KB	0 KB
12	DimGeography	655	136 KB	104 KB	16 KB	16 KB
13	DimProduct	606	6120 KB	5936 KB	88 KB	96 KB
14	DimEmployee	296	176 KB	120 KB	48 KB	8 KB
15	FactSalesQuota	163	48 KB	8 KB	40 KB	0 KB
16	FactCallCenter	120	48 KB	16 KB	32 KB	0 KB
17	DatabaseLog	115	408 KB	384 KB	8 KB	16 KB
18	DimCurrency	105	32 KB	8 KB	24 KB	0 KB
19	DimAccount	99	48 KB	16 KB	32 KB	0 KB
20	DimProductSubcategory	37	32 KB	8 KB	24 KB	0 KB
21	DimPromotion	16	32 KB	8 KB	24 KB	0 KB
22	DimOrganization	14	16 KB	8 KB	8 KB	0 KB
23	DimSalesTerritory	11	32 KB	8 KB	24 KB	0 KB
24	DimSalesReason	10	16 KB	8 KB	8 KB	0 KB
25	DimDepartmentGroup	7	16 KB	8 KB	8 KB	0 KB
26	DimProductCategory	4	32 KB	8 KB	24 KB	0 KB
27	DimScenario	3	16 KB	8 KB	8 KB	0 KB
28	AdventureWorksDWBuildVersion	1	72 KB	8 KB	8 KB	56 KB

نظرات خوانندگان

نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۶/۱۸ ۱۲:۵۴

یک مورد استفاده کاربردی، از رویه **sp_MSforeachtable** به هنگام تجمعی بانک‌های اطلاعاتی است (برای مثال تجمعی دو DB کوچک در یک DB بزرگتر). برای این منظور می‌توان در ابتدا تمامی Constraint‌های جداول را موقتاً غیر فعال کرد، عملیات Load داده در جداول را انجام داد و مجدد آنها را فعال نمود.

چنانچه از SSIS Package برای اینکار استفاده شود، با فرض اینکه در مرحله قبل Schema تمامی اشیاء بانک منتقل شده است، Control Flow این Package شامل Step‌های زیر می‌باشد:

گام اول - غیر فعال کردن تمامی Constraint‌های جداول

برای این منظور از Execute SQL Task موسوم به این صورت استفاده می‌شود که در بخش SQL Statement دستور زیر نوشته شود:

```
exec sp_MSforeachtable 'ALTER TABLE ? NOCHECK CONSTRAINT ALL'
```

گام دوم - انتقال داده‌ها به شکل Data Only

برای این منظور از Transfer SQL Server Objects Task به صورت زیر استفاده می‌شود:
در بخش Object تغییرات زیر اعمال شود:

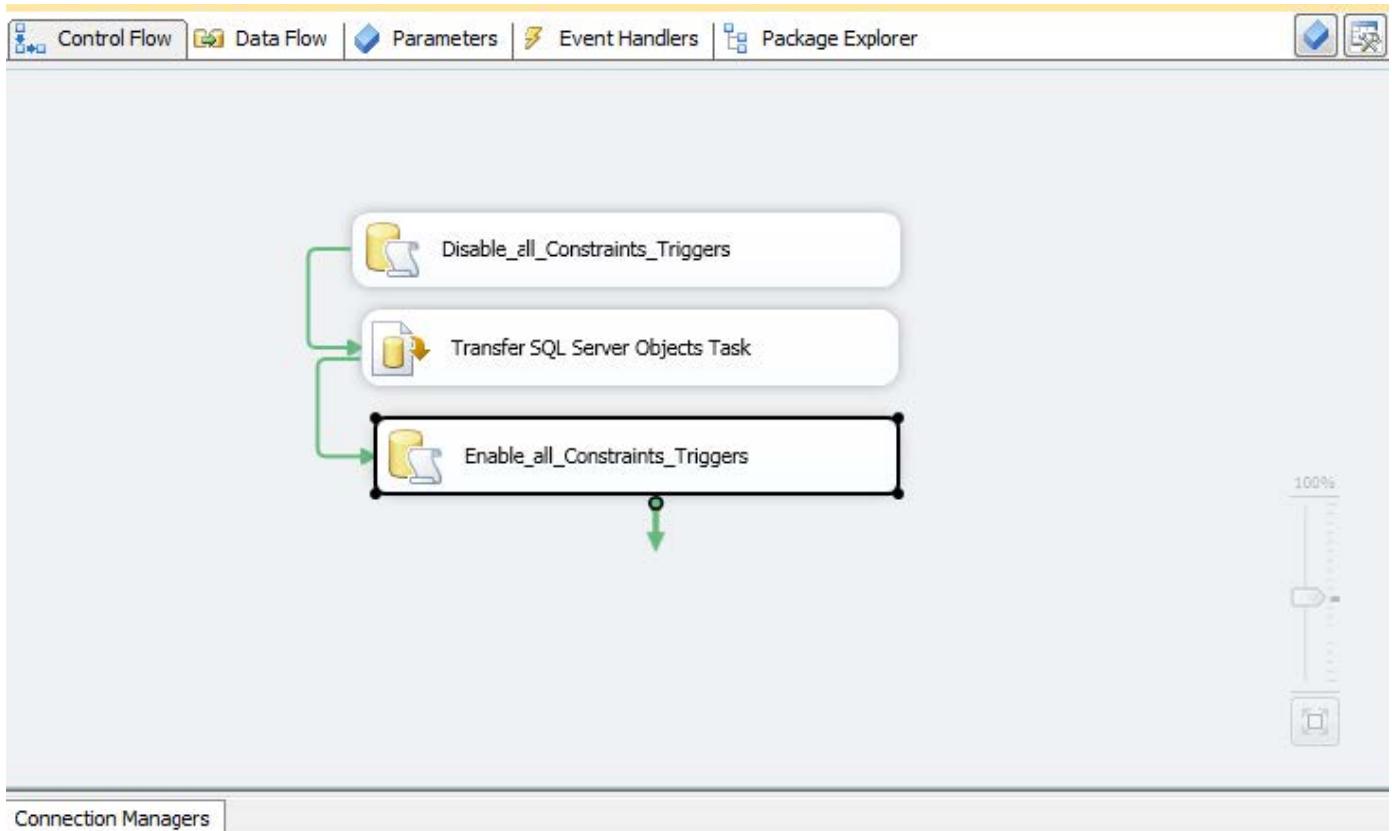
- در قسمت Destination ، پروپریتی CopyData با مقدار True و ExistingData با مقدار Append تنظیم شود.
- در قسمت CopyAllTables ، پروپریتی Copy & Option با مقدار True در بخش Destination تنظیم شود.

گام سوم - فعال کردن مجدد Constraint‌های جداول

برای این منظور از Execute SQL Task موسوم به این صورت زیر استفاده می‌شود:
در بخش SQL Statement دستور زیر نوشته شود:

```
exec sp_MSforeachtable 'ALTER TABLE ? CHECK CONSTRAINT ALL'
```

نکته: در صورت وجود Trigger برای جداول بانک اطلاعاتی نیز می‌توان به همین شکل عمل نمود. (ابتدا آنها را غیر فعال و مجدد فعال کرد)



برای مثال شکل SSIS Package فوق به صورت تصویر فوق است.

مقدمه Profiler یک ابزار گرافیکی برای ردیابی و نظارت بر کارآئی SQL Server است. امکان ردیابی اطلاعاتی در خصوص رویدادهای مختلف و ثبت این داده‌ها در یک فایل (با پسوند trc) یا جدول برای تحلیل‌های آتی نیز وجود دارد. برای اجرای این ابزار مراحل زیر را انجام دهید:

Start > Programs > Microsoft SQL Server > Performance Tools > SQL Server Profiler

و یا در محیط Management Studio از منوی Tools گزینه SQL Server Profiler را انتخاب نمائید.

1- اصطلاحات

1-1- رویداد (Event): یک رویداد، کاری است که توسط موتور بانک اطلاعاتی (Database Engine) انجام می‌شود. برای مثال هر یک از موارد زیر یک رویداد هستند.

- متصل شدن کاربران (login connections) قطع شدن ارتباط یک login
- اجرای دستورات T-SQL، شروع و پایان اجرای یک رویه، شروع و پایان یک دستور در طول اجرای یک رویه، اجرای رویه‌های دور Remote Procedure Call
- باز شدن یک Cursor
- بررسی و کنترل مجوزهای امنیتی

1-2- کلاس رویداد (Event Class): برای بکارگیری رویدادها در Profiler، از یک Event Class استفاده می‌کنیم. یک Audit Login با استفاده از کلاس Audit Login قابل پیاده سازی است. هر یک از موارد زیر یک Event Class هستند.

SQL:BatchCompleted -
Audit Login -
Audit Logout -
Lock: Acquired -
Lock: Released -

1-3- گروه رویداد (Event Category): یک گروه رویداد شامل رویدادهایی است که به صورت مفهومی دسته بندی شده اند. برای مثال، کلیه رویدادهای مربوط به قفل‌ها از جمله Lock: Acquired (بدست آوردن قفل) و Lock: Released (رها کردن قفل) در گروه Locks قرار دارند.

1-4- ستون داده‌ای (Data Column): یک ستون داده‌ای، خصوصیت و جزئیات یک رویداد را شامل می‌شود. برای مثال در یک Trace که رویدادهای Lock: Acquired را نظارت می‌کند، ستون Binary Data شامل شناسه (ID) یک صفحه و یا یک سطر قفل شده است و یا اینکه ستون Duration مدت زمان اجرای یک رویه را نمایش می‌دهد.

1-5- الگو (Template): یک الگو، مشخص کننده تنظیمات شامل رویدادهایی است که نیاز دارید بر آنها نظارت داشته باشید. هنگامیکه یک Trace براساس یک الگو اجرا شود، رویدادهای مشخص شده، نظارت می‌شوند و نتیجه به صورت یک فایل یا جدول قابل مشاهده خواهد بود.

1-6- ردیاب Trace: یک داده‌ها را براساس رویدادهای انتخاب شده، جمع آوری می‌کند. امکان اجرای بلافصله یک Trace برای جمع آوری اطلاعات با توجه به رویدادهای انتخاب شده و ذخیره کردن آن برای اجرای آتی وجود دارد.

1-7- فیلتر Filter: هنگامی که یک Trace یا الگو ایجاد می‌شود، امکان تعریف شرایطی برای فیلتر کردن داده‌های جمع آوری شده نیز وجود دارد. این کار باعث کاهش حجم داده‌های گزارش شده می‌شود. برای مثال اطلاعات مربوط به یک کاربر خاص جمع آوری می‌شود و یا اینکه رشد یک بانک اطلاعاتی مشخص بررسی می‌شود.

2- انتخاب الگو Profiler Trace Templates (از آنجائیکه اصولاً انتخاب Event‌های مناسب، کار سخت و تخصصی می‌باشد برای راحتی کار تعدادی Template‌های آماده وجود دارد، برای مثال TSQL_Duration تأکیدش روی مدت انجام کار است و یا SP_Counts در مواردی که بخواهیم رویه‌های ذخیره شده را بهینه کنیم استفاده می‌شود در جدول زیر به شرح هر یک پرداخته شده است:

الگو	هدف
Blank	ایجاد یک Trace کلی
SP_Counts	ثبت اجرای هر رویه ذخیره شده برای تشخیص اینکه هر رویه چند بار اجرا شده است
Standard	ثبت آمارهای کارائی برای هر رویه ذخیره شده و Query‌های Login که اجرا می‌شوند و عملیات ورود و خروج هر (پیش فرض)
TSQL	ثبت یک لیست از همه رویه‌های ذخیره شده و Query‌های عادی SQL که اجرا می‌شوند ولی آمارهای کارائی را شامل نمی‌شود
TSQL_Duration	ثبت مدت زمان اجرای هر رویه ذخیره شده و هر Query عادی SQL
TSQL_Grouped	ثبت تمام login‌ها و logout‌ها در طول اجرای رویه‌های ذخیره شده و هر Query عادی SQL، شامل اطلاعاتی برای شناسائی برنامه و کاربری که درخواست را اجرا می‌کند
TSQL_Locks	ثبت اطلاعات انسداد (deadlock) و بن بست (blocking) از قبیل blocked processes, deadlock chains, deadlock ...، graphs . این الگو همچنین درخواست‌های تمام رویه‌های ذخیره شده و تمامی دستورات هر رویه و هر Query عادی SQL را دریافت می‌کند
TSQL_Replay	ثبت اجرای رویه‌های ذخیره شده و Query‌های SQL در یک Instance و مهیا کردن امکان اجرای دوباره عملیات در سیستمی دیگر
TSQL_SPs	ثبت کارائی برای Query‌های SQL، رویه‌های ذخیره شده و تمامی دستورات درون یک رویه ذخیره شده و نیز عملیات ورود و خروج هر Login
Tuning	ثبت اطلاعات کارائی برای Query‌های عادی SQL و رویه‌های ذخیره شده و یا تمامی دستورات درون یک رویه ذخیره شده

3- انتخاب رویداد SQL Trace Event Groups (رویدادها در 21 گروه رویداد دسته بندی می‌شوند که در جدول زیر لیست شده اند:

هدف	گروه رویداد
13 رویداد برای واسطه سرویس (Service Broker)	Broker
1 رویداد برای بارگذاری اسمبلی‌های CLR (Common Language Runtime)	CLR
7 رویداد برای ایجاد، دستیابی و در اختیار گرفتن Cursor	Cursors
6 رویداد برای رشد/کاهش (grow/shrink) فایل‌های Database/Mirroring	Database
2 رویداد برای آگاه کردن وضعیت نابسامان درون یک Instance	Deprecation
16 رویداد برای خطاهای هشدارها و پیغام‌های اطلاعاتی که ثبت شده است	Errors and Warnings
3 رویداد برای پیگیری یک شاخص متنی کامل	Full Text
9 رویداد برای بدست آوردن، رها کردن قفل و بن بست (Deadlock)	Locks
5 رویداد برای درخواست‌های توزیع شده و RPC (اجرای رویه‌های دور)	OLEDB
3 رویداد برای وقتی که یک شی ایجاد، تغییر یا حذف می‌شود	Objects
14 رویداد برای ثبت نقشه درخواست‌ها (Query Plan) برای استفاده نقشه راهنمایی (Plan Guide) به منظور بهینه سازی کارائی درخواست‌ها، همچنین این گروه رویداد در خواست‌های متنی کامل (full text) را ثبت می‌کند	Performance
10 رویداد برای ایجاد Online Index	Progress Report
4 رویداد برای سرویس اطلاع رسان (Notification Service)	Query Notifications
2 رویداد برای وقتی که یک جدول یا شاخص، پویش می‌شود	Scans
44 رویداد برای وقتی که مجوزی استفاده شود، جابجایی هویتی SQL Instance، تنظیمات امنیتی اشیائی تغییر کند، یک Database شروع و متوقف شود و یک Database جایگزین شود یا از آن پشتیبان گرفته شود	Security Audit
3 رویداد برای Mount Tape، تغییر کردن حافظه سرور و بستن یک فایل Trace	Server
3 رویداد برای وقتی که Connection‌ها موجود هستند و یک فعال می‌شود، همچنین یک Trigger و یک تابع دسته بندی (classification functions) مربوط به مدیریت منابع (resource governor) رخ دهد	Sessions
12 رویداد برای اجرای یک رویه ذخیره شده و دستورات درون آن، کامپایل مجدد و استفاده از حافظه نهانی (Cache)	Stored Procedures
13 رویداد برای شروع، ذخیره، تأیید و لغو یک تراکنش	Transactions
9 رویداد برای اجرای SQL‌های Query و جستجوهای XQUERY (در داده‌های XML)	TSQL

هدف	گروه رویداد
10 رویداد که شما می‌توانید پیکربندی کنید	User Configurable

به طور معمول بیشتر از گروه رویدادهای Locks, Performance, Security Audit, Stored Procedures و TSQL استفاده می‌شود.

4- انتخاب ستون‌های داده ای (Data Columns) اگرچه می‌توان همه‌ی 64 ستون داده ای ممکن را برای ردیابی انتخاب کرد ولیکن داده‌های Trace شما زمانی مفید خواهند بود که اطلاعات ضروری را ثبت کرده باشید. برای مثال شماره تراکنش‌ها را، برای یک رویداد RPC:Completed می‌توانید برگردانید، اما همه رویداهای ذخیره شده مقادیر را تغییر نمی‌دهند بنابراین شماره ترتیب تراکنش‌ها فضای بیهوده ای را مصرف می‌کند. بعلاوه همه ستون‌های داده ای برای تمامی رویدادهای Trace معتبر نیستند. برای مثال SQL:BatchStarting و RPC:Starting برای رویدادهای Duration و Read ، Write ، ApplicationName، NTUserName، LoginName، ClientProcessID، SPID، HostName، LoginSID، NTDomainName SessionLoginName و مشخص می‌کند چه کسی و از چه منشاء دستور را اجرا کرده است.

ستون SessionLoginName معمولاً نام Login ای که از آن برای متصل شدن به SQL Instance استفاده شده است را نشان می‌دهد. در حالیکه ستون LoginName نام کاربری را که دستور را اجرا می‌کند نشان می‌دهد (EXECUTE AS). ستون ApplicationName خالی است مگر اینکه در ConnectionString برنامه کاربردیمان این خصوصیت (Property) مقداردهی شده باشد. ستون StartTime و EndTime زمان سرحدی برای هر رویداد را ثبت می‌کند این ستون‌ها بویژه در هنگامی که به عملیات Correlate نیاز دارید مفید هستند.

5- بررسی چند سناریو نمونه

- **یافتن درخواست‌های (Queries)** که بدترین کارایی را دارا هستند. برای ردیابی درخواست‌های ناکارا، از رویداد RPC:Completed از دسته SQL:BatchCompleted و رویداد Stored Procedure از دسته SQL:BatchStarting استفاده می‌شود.

- **ناظارت بر کارایی رویه‌ها** برای ردیابی کارائی رویه‌ها، از رویدادهای SP:Starting، SP:Completed، SP:StmtCompleted و SP:StmtStarting از کلاس SP:StmtStarting و رویدادهای Stored Procedure از کلاس SQL:BatchCompleted استفاده می‌شود.

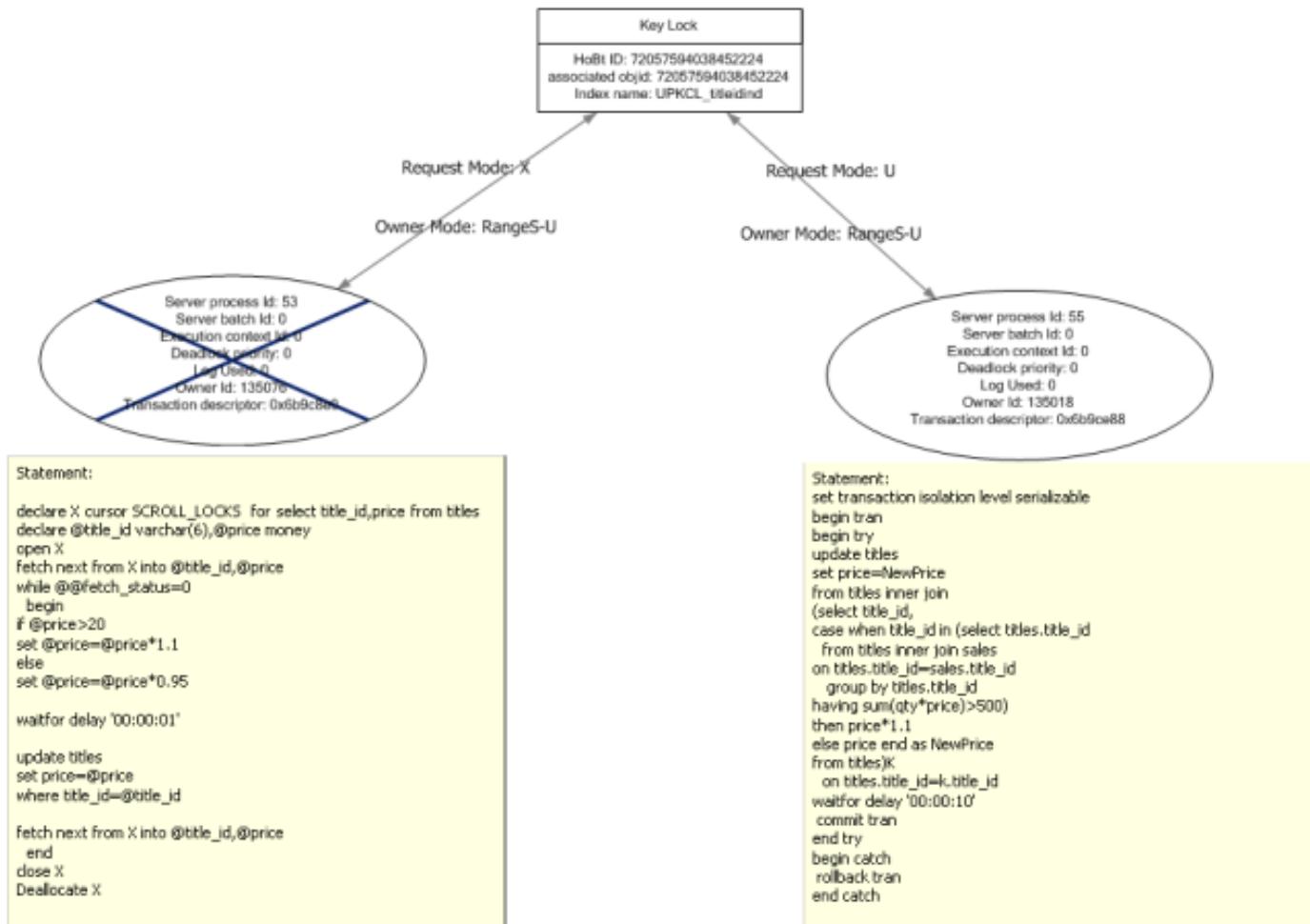
- **ناظارت بر اجرای دستورات T-SQL** توسط هر کاربر برای ردیابی دستوراتی که توسط یک کاربر خاص اجرا می‌شود، نیاز به ایجاد یک ناظارت بر رویدادهای کلاس‌های Sessions و ExistingConnection و TSQL داریم همچنین لازم است نام کاربر در قسمت فیلتر و با استفاده از DBUserName مشخص شود.

- **اجرا دوباره ردیاب (Trace Replay)** این الگو معمولاً برای debugging استفاده می‌شود برای این منظور از الگوی Replay استفاده می‌شود. در ضمن امکان اجرای دوباره عملیات در سیستمی دیگر با استفاده از این الگو مهیا می‌شود.

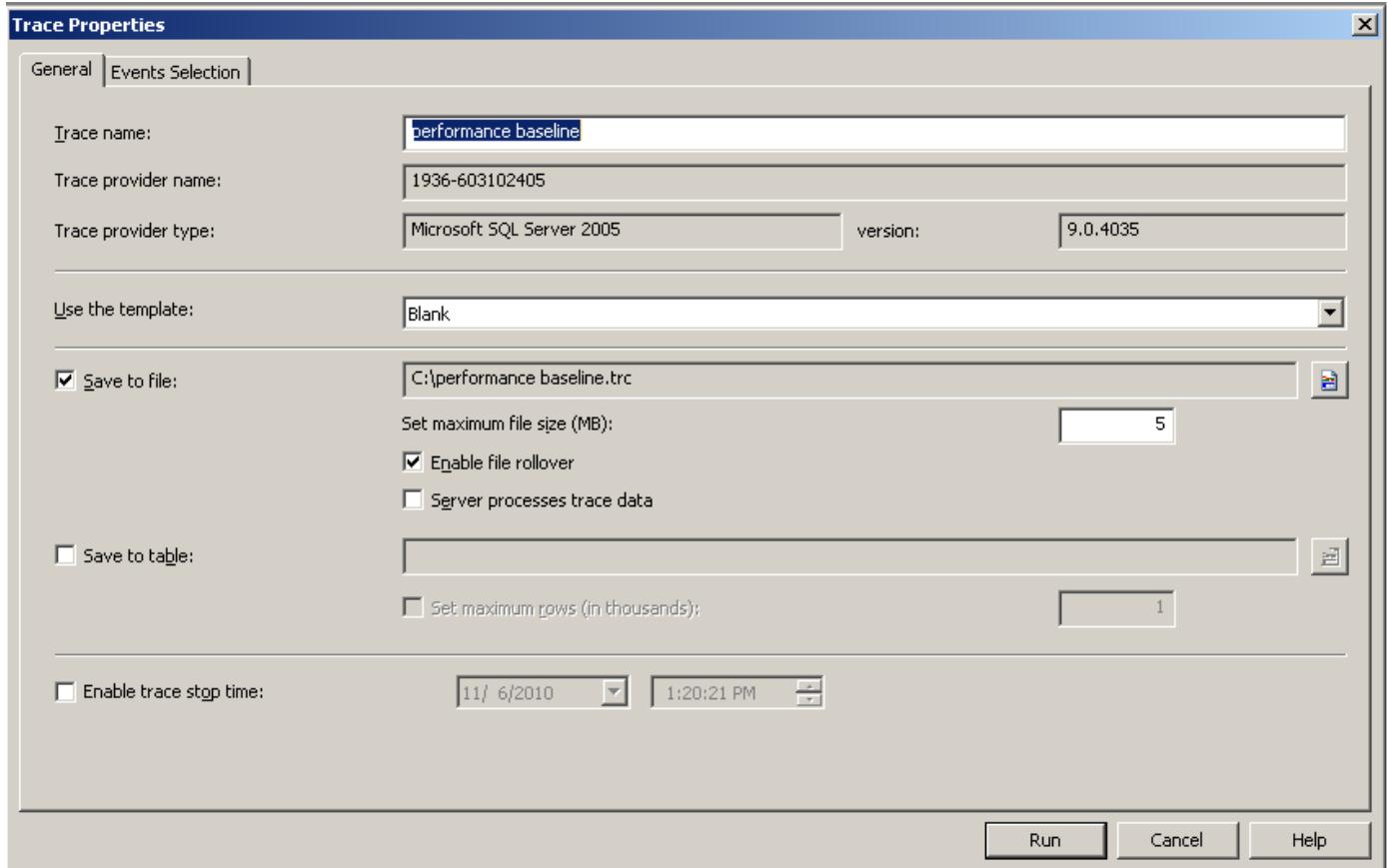
- **ابزار Tuning Advisor (راهنمای تنظیم کارائی)** این ابزاری برای تحلیل کارائی یک یا چند بانک اطلاعاتی و تاثیر عملکرد آنها بر بار کاری (Workload) سرویس دهنده است. یک بار کاری مجموعه ای از دستورات T-SQL است که روی بانک اطلاعاتی اجرا می‌شود. بعد از تحلیل تاثیر بارکاری بر بانک اطلاعاتی، Tuning Advisor توصیه هایی برای اضافه کردن، حذف و یا تغییر طراحی فیزیکی ساختار بانک اطلاعاتی ارائه می‌دهد این تغییرات شامل پیشنهاد برای تغییر ساختاری موارد Clustered Indexes، Nonclustered Indexes، Indexed View است. برای ایجاد بارکاری می‌توان از یک ردیاب تهیه شده در SQL Profiler استفاده کرد برای این منظور از الگوی Tuning استفاده می‌شود و یا رویدادهای RPC:Completed، SQL:BatchCompleted و SP:StmtCompleted را ردیابی نمائید.

- **ترکیب ابزارهای ناظارتی (Correlating Performance and Monitoring Data)** یک Trace برای ثبت اطلاعاتی که در یک SQL Instance رخ می‌دهد، استفاده می‌شود. System Monitor برای ثبت شمارنده‌های کارائی (performance counters) استفاده می‌شود و همچنین از منابع سخت افزاری و اجزای دیگر که روی سرور اجرا می‌شوند، تصاویری فراهم می‌کند. توجه شود که در مورد یک فایل ردیاب (trace file) و یک Log (abزار Performance Counter Log) استفاده می‌شود که در آن داده ای StartTime و EndTime باشد انتخاب شود، برای این کار از منوی File گزینه Import Performance Data انتخاب می‌شود.

- جستجوی علت رخدادن یک بن بست برای ردیابی علت رخدادن یک بن بست، از رویدادهای RPC:Starting و رویدادهای Lock:Deadlock Chain، Lock:Deadlock graph، Stored Procedure از SQLBatchStarting دسته Locks استفاده می‌شود. (در صورتی که نیاز به یک ارائه گرافیکی دارید graph از Deadlock استفاده نماید، خروجی مطابق تصویر زیر می‌شود).

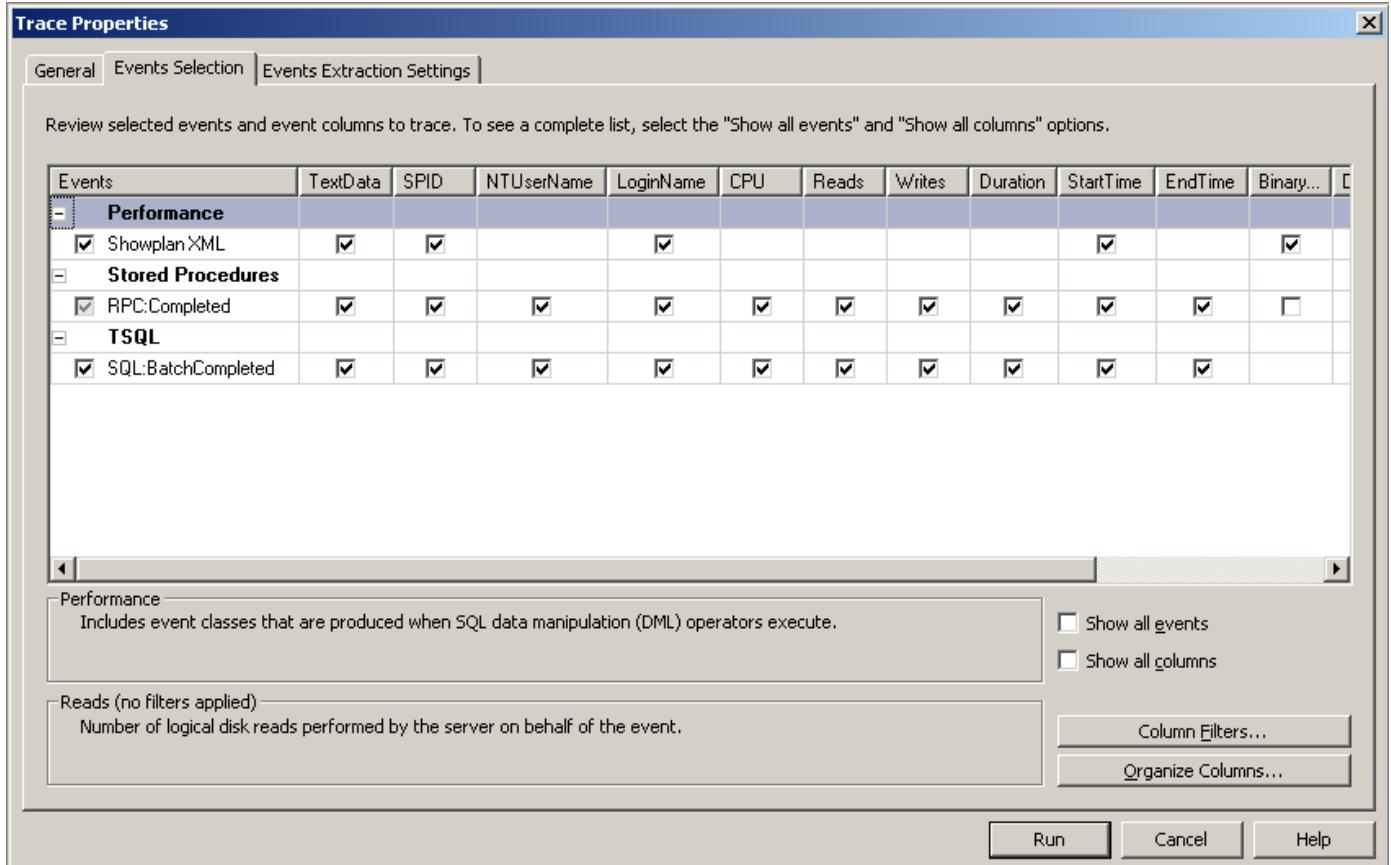


- 1- ایجاد یک Trace 1- Profiler را اجرا کنید از منوی File گزینه New Trace مورد نظرتان متصل شوید.
- 2- مطابق تصویر زیر برای Trace یک نام و الگو و تنظیمات ذخیره سازی فایل را مشخص کنید.

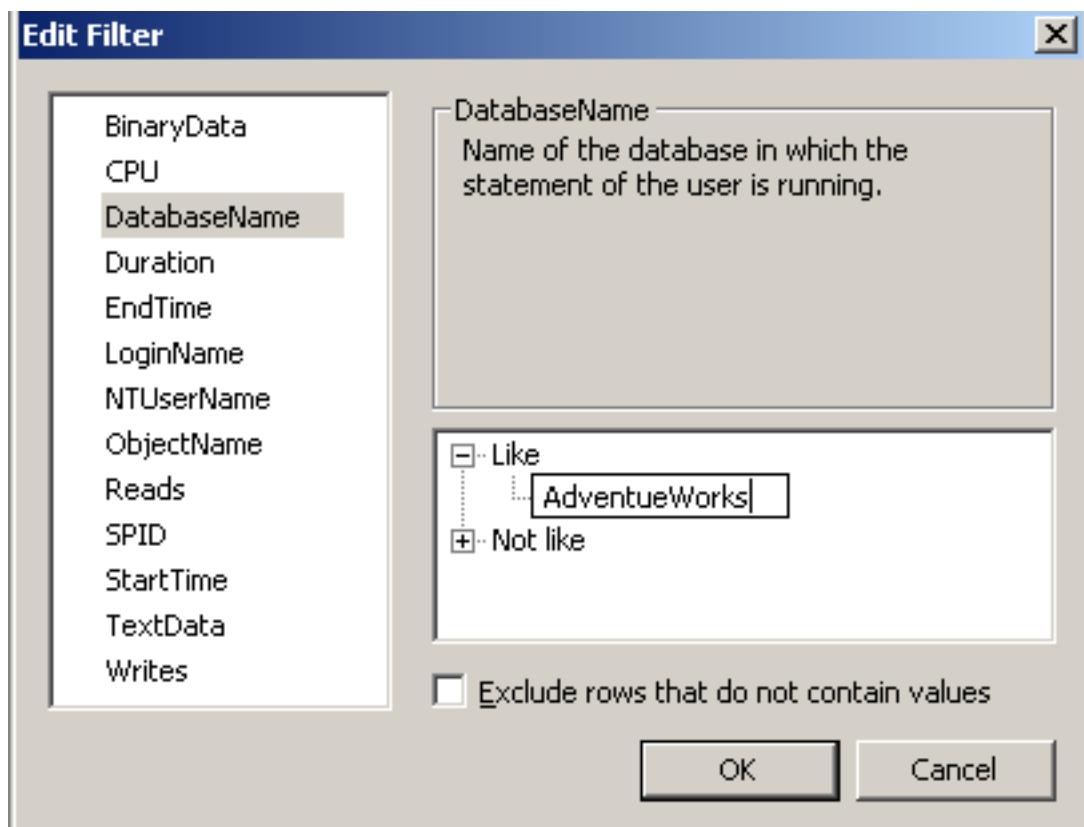


3- بر روی قسمت Events Selection کلیک نمایید.

4- مطابق تصویر زیر رویدادها و کلاس رویدادها را انتخاب کنید، ستون های TextData, NTUserName, LoginName, ستون های CPU.Reads, Writes, Duration, SPID, StartTime, EndTime, BinaryData, DataBaseName, ServerName کنید.



5- روی Column Filters کلیک کنید و مطابق تصویر زیر برای DatabaseName فیلتری تنظیم کنید.



6- روی Run کلیک کنید. تعدادی Query و رویه ذخیره شده مرتبط با پایگاه داده AdventureWorks اجرا کنید.

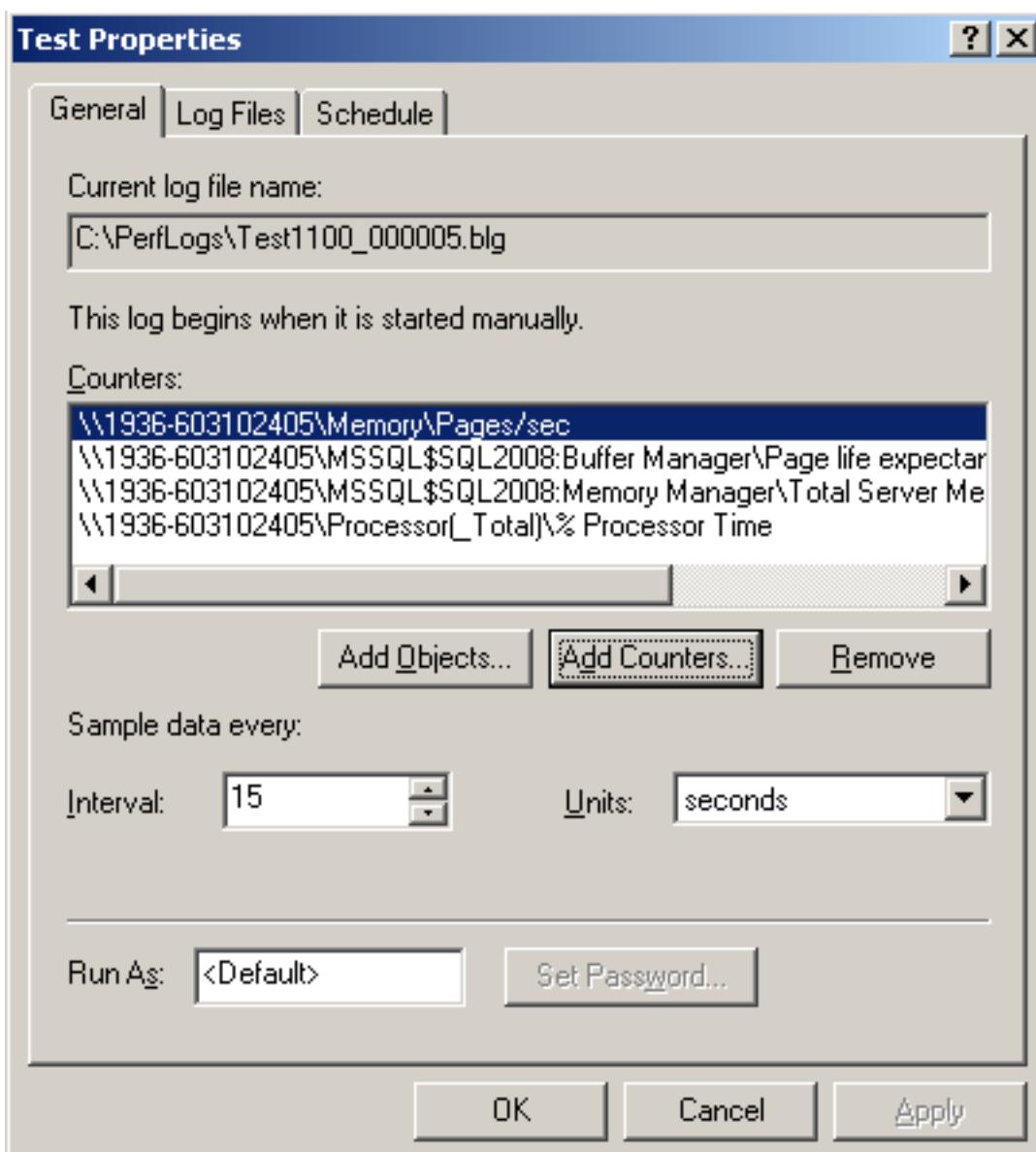
5- ایجاد یک Counter Log برای ایجاد یک Counter Log مراحل زیر را انجام دهید:

1- ابزار Performance را اجرا کنید (برای این کار عبارت PerfMon را در قسمت Run بنویسید).

2- در قسمت Counter Logs یک log ایجاد کنید.

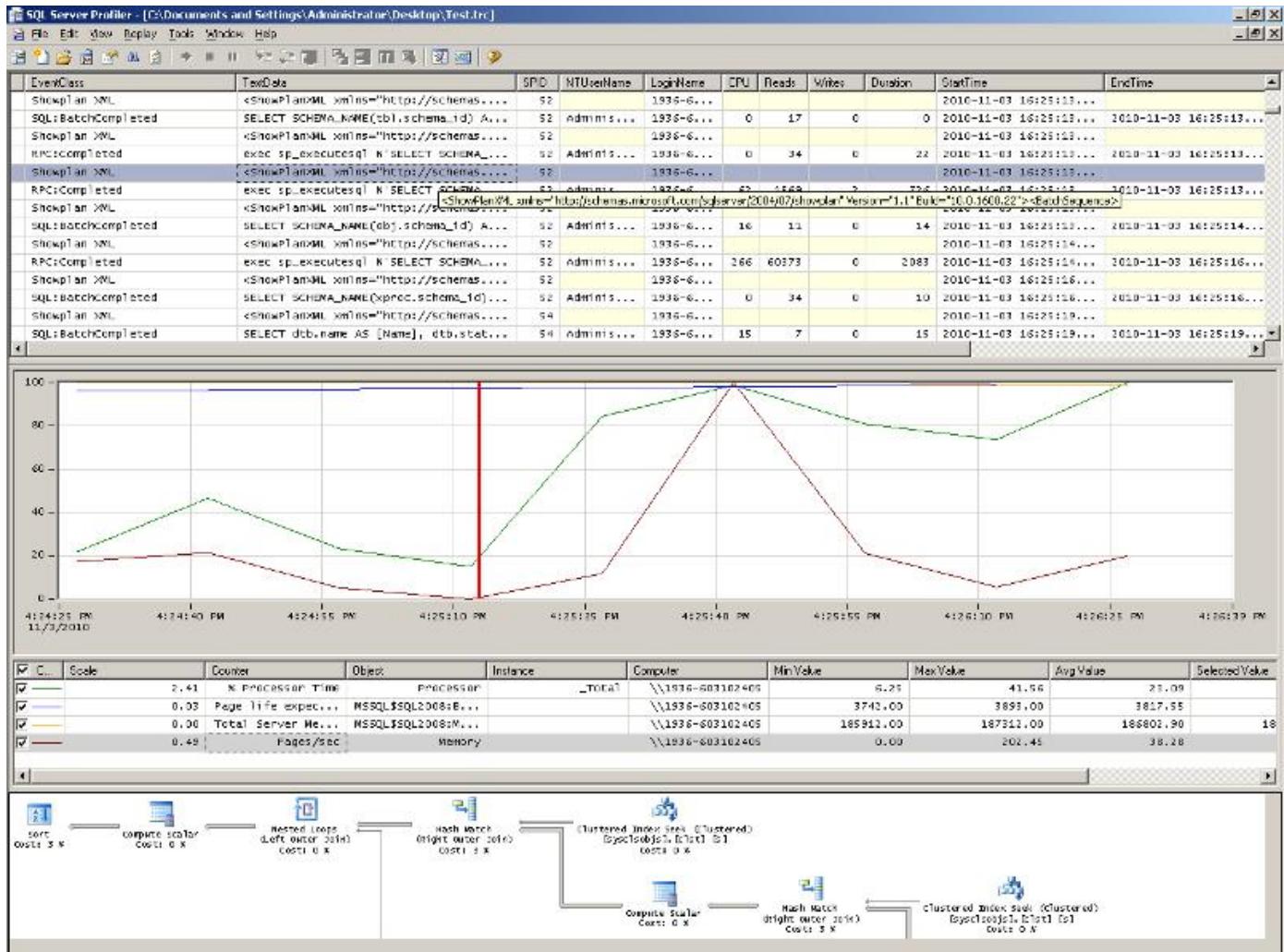
3- روی Add Counters کلیک کرده و مطابق تصویر موارد زیر را انتخاب کنید.

Performance Object	Select counters from list
Network Interface	Output Queue Length
Processor	% Processor Time
System	Processor Queue Length
SQLServer	Buffer Manager:Page life expectancy



4- روی Ok کلیک کنید تا Counter Log ذخیره شود سپس روی آن راست کلیک کرده و آنرا Start کنید.

- 5- ترکیب ابزارهای نظارتی (Correlating SQL Trace and System Monitor Data) 1- Profiler را اجرا کنید از منوی File گزینه Open و سپس File را انتخاب کنید فایل trc را که در گام اول ایجاد کردید، باز نمایید.
- 2- از منوی Import Performance Data گزینه File را انتخاب کنید و فایل counter log را که در مرحله قبل ایجاد کردید انتخاب کنید.



نکته: اطلاعات فایل trc را می‌توان درون یک جدول وارد کرد، بدین ترتیب می‌توان آنالیز بیشتری داشت به عنوان مثال دستورات زیر این عمل را انجام می‌دهند.

```
SELECT * INTO dbo.BaselineTrace
FROM fn_trace_gettable(' c:\performance baseline.trc ', default);
```

با اجرای دستور زیر جدولی با نام BaselineTrace ایجاد و محتویات Trace مان (performance baseline.trc) در آن درج می‌گردد.

مقدمه

در اکثر موارد در یک Landscape عملیاتی، چنانچه به تجمیع و انتقال داده‌ها از بانک‌های اطلاعاتی مختلف نیاز باشد، از SSIS Package اختصار SQL Server Integration Service استفاده می‌شود و معمولاً با تعریف یک Job در سطح SQL Server به اجرای Package در زمانهای مشخص می‌پردازند. چنانچه در موقعیتی لازم باشد که از طریق برنامه کاربردی توسعه یافته، به اجرای Package مبادرت ورزیده شود و البته نخواهیم Job تعریف شده را از طریق کد برنامه، اجرا کنیم و در واقع این امکان را داشته باشیم که همانند یک رویه ذخیره شده تعریف شده در سطح بانک اطلاعاتی به اجرای عمل فوق پردازیم، یک راه حل می‌تواند تعریف یک CLR Stored Procedures باشد. در این مقاله به بررسی این موضوع پرداخته می‌شود، در ابتدا لازم است به بیان تئوری موضوع پرداخته شود (قسمت‌های ۱ الی ۵) در ادامه به ذکر پیاده سازی روش پیشنهادی پرداخته می‌شود.

1- اجرای Integration Service Package

- ابزار خط فرمان command-line dtexec.exe
- ابزار اجرائی پکیج dtexecui.exe
- استفاده از job SQL Server Agent

توجه: همچنین یک Package را در زمان طراحی در Business Intelligence Development Studio (BIDS) می‌توان اجرا نمود.

2- استفاده از dtexec جهت اجرای Package

با استفاده از ابزار dtexec می‌توان Package‌های ذخیره شده در فایل سیستم، یک Integration Service را اجرا نمود. یا SQL Instance

توجه: در سیستم عامل‌های 64 بیتی، ابزار dtexec موجود در Integration Service با نسخه 64 بیتی نصب می‌شود. چنانچه با استفاده از 32 بیتی اجرا کنید، لازم است ابزار dtexec نسخه 32 بیتی نصب شود. ابزار dtexec دستیابی به تمامی ویژگی‌های پیکربندی و اجرای Package از قبیل اتصالات، مشخصات (Properties)، متغیرها، logging و شاخص‌های پردازشی را فراهم می‌کند.

توجه: زمانی که از نسخه ابزار dtexec که با SQL Server 2008 ارائه شده استفاده می‌کنید برای اجرای یک SSIS Package نسخه 2005 Integration Service به صورت موقت Package را به نسخه 2008 ارتقا می‌دهد، اما نمی‌توان از ابزار dtexec برای ذخیره این تغییرات استفاده کرد.

2-1- ملاحظات نصب dtexec روی سیستم‌های 64 بیتی

به صورت پیش فرض، یک سیستم عامل 64 بیتی که هر دو نسخه 64 بیتی و 32 بیتی ابزار خط فرمان Integration Service را دارد، نسخه 32 بیتی نصب شده را در خط فرمان اجرا خواهد کرد. نسخه 32 بیتی بدین دلیل اجرا می‌شود که در متغیر محیطی (Path environment variable) Path directory مسیر Path نسخه 32 بیتی قرار گرفته است. به طور معمول:

```
<drive>:\Program Files(x86)\Microsoft SQL Server\100\DTSP\Binn)
```

توجه: اگر از SQL Server Agent برای اجرای Package استفاده می‌کنید، SQL Server Agent به طور خودکار از ابزار نسخه 64 بیتی استفاده می‌کند. SQL Server Agent از Registry و نه از متغیر محیطی Path استفاده می‌کند. برای اطمینان از اینکه نسخه 64 بیتی این ابزار را در خط فرمان اجرا می‌کنید، directory را به این قرار گرفته است: <drive>:\Program Files\Microsoft SQL Server\100\DTSP\Binn> و ابزار را از این مسیر اجرا کنید و یا برای همیشه مسیر قرار گرفته در متغیر محیطی path را با مسیری که نسخه 64 بیتی قرار دارد، جایگزین کنید.

2-2- تفسیر کدهای خروجی هنگامی که یک کد خروجی (Return Code) بر می‌گرداند:

توصیف	مقدار
Package با موفقیت اجرا شده است.	0
Package با خطا مواجه شده است.	1
در حال اجرا توسط کاربر لغو شده است.	3
Package پیدا نشده است.	4
Package بارگذاری نشده است.	5
ابزار با یک خطای نحوی یا خطای معنایی در خط فرمان برخورد کرده است.	6

2-3-2 قوانین نحوی dtexec تمامی گزینه‌ها (Options) باید با یک علامت Slash (/) و یا Minus (-) شروع شوند. یک آرگومان باید در یک quotation mark محصور شود چنانچه شامل یک فاصله خالی باشد. گزینه‌ها و آرگومان‌ها بجز رمزعبور حساس به حروف کوچک و بزرگ نیستند.

Syntax 2-3-1-

```
dtexec /option [value] [/option [value]]...
```

2-3-2 Parameters نکته: در Integration Service DTS run که برای Data Transformation Service) ابزار خط فرمان dtsrun استفاده می‌شد، با ابزار خط فرمان dtexec جایگزین شده است. نسخه SQL Server 2000

- تعدادی از گزینه‌های خط فرمان dtsrun به طور مستقیم در dtexec معادل دارند برای مثال نام Server و نام Package.
- تعدادی از گزینه‌های dtsrun به طور مستقیم در dtexec معادل ندارند.
- تعدادی گزینه‌های خط فرمان جدید dtexec وجود دارد که در ویژگی‌های جدید Integration Service پشتیبانی می‌شود.

2-3-3 مثال 1) به منظور اجرای یک SSIS Package که در SQL Server ذخیره شده است، با استفاده از Windows Authentication :

```
dtexec /sq <Package Name> /ser <Server Name>
```

2) به منظور اجرای یک SSIS Package که در پوشه SSIS Package Store در File System ذخیره شده است :

```
dtexec /dts "\File System\<Package File Name>"
```

3) به منظور اجرای یک SSIS Package که در سیستم فایل ذخیره شده است و مشخص کردن گزینه logging :

```
dtexec /f "c:\<Package File Name>" /l "DTS.LogProviderTextFile; <Log File Name>"
```

4) به منظور اجرای یک SSIS Package که در SQL Server ذخیره شده با استفاده از SQL Server Authentication نموده (user:ssis;pwd:ssis@ssis) و رمز (123) :

```
dtexec /server "<Server Name>" /sql "<Package Name>" / user "ssis" /Password "ssis@ssis" /De "123"
```

3- تنظیمات سطح حفاظتی یک Package به منظور حفاظت از داده‌ها در Integration Service می‌توانید یک سطح

حفظاتی (protection level) را تنظیم کنید که به حفاظت از داده‌های صرفاً حساس یا تمامی داده‌های یک Package کمک نماید. به علاوه می‌توانید این داده‌ها را با یک User Key یا یک Password رمزگذاری نمائید یا به رمزگذاری داده‌ها در بانک اطلاعاتی اعتماد کنید. همچنین سطح حفاظتی که برای یک Package استفاده می‌کنید، الزاماً ایستا (static) نیست و در طول چرخه حیات یک Package می‌تواند تغییر کند. اغلب سطح حفاظتی در طول توسعه یا به محض (deploy) استقرار Package تنظیم می‌شود.

توجه: علاوه بر سطوح حفاظتی که توصیف شد، Package‌ها در بانک اطلاعاتی msdb ذخیره می‌شوند که همچنین می‌توانند توسط نقش‌های ثابت در سطح بانک اطلاعاتی (fixed database-level roles) حفاظت شوند. Integration Service شامل 3 نقش ثابت بانک اطلاعاتی برای نسبت دادن مجوزها به Package است که عبارتند از db_ssisoperator و db_ssiscadmin و db_ssisltduser.

1-3 درک سطوح حفاظتی در یک Package اطلاعات زیر به عنوان حساس تعریف می‌شوند:

- بخش password در یک connection string در نظر گرفته می‌شود.
- گرچه، اگر گزینه ای را که همه چیز را رمزگذاری کند، انتخاب کنید تمامی حساس در گزینه connection string می‌شود.
- گره‌های task-generated XML که بر جسب (tagged) هایی حساس هستند.
- هر متغیری که به عنوان حساس نشان گذاری شود.

3-1-1 Do not save sensitive هنگامی که Package ذخیره می‌شود از ذخیره مقادیر ویژگی‌های حساس در جلوگیری می‌کند. این سطح حفاظتی رمزگذاری نمی‌کند اما در عوض از ذخیره شدن ویژگی‌هایی که حساس نشان گذاری شده اند به همراه Package جلوگیری می‌کند.

3-1-2 Encrypt all with password به منظور رمزگذاری تمامی Package از یک Password استفاده می‌شود. این سطح حفاظتی در SSIS Designer یا اجرای Package با توجه به این کاربری می‌شود که کاربر هنگامی که Export را ایجاد یا Package می‌کند، ارائه می‌دهد. به منظور باز کردن Package در dtsexec فرمان با استفاده از Package کاربر بایست رمز Package را ارائه نماید. بدون رمز کاربر قادر به دستیابی و اجرای Package نیست.

3-1-3 Encrypt all with user key به منظور رمزگذاری تمامی Package از یک کلید که مبتنی بر Profile کاربر جاری می‌باشد، استفاده می‌شود. تنها کاربری که Export را ایجاد یا Package می‌کند، می‌تواند Package را در SSIS Designer باز کند و یا را توسط ابزار خط فرمان dtsexec اجرا کند.

3-1-4 Encrypt sensitive with password به منظور رمزگذاری تنها مقادیر ویژگی‌های حساس در Package از یک Password استفاده می‌شود. برای رمزگذاری از DPAPI استفاده می‌شود. داده‌های حساس به عنوان بخشی از Package ذخیره می‌شوند اما آن داده‌ها با استفاده از Password رمزگذاری می‌شوند. به منظور باز نمودن Package در SSIS Designer کاربر باید رمز Package را ارائه دهد. اگر رمز ارائه نشود، Package بدون داده‌های حساس باز می‌شود و کاربر باید مقادیر جدیدی برای داده‌های حساس فراهم کند. اگر کاربر سعی نماید Package را بدون ارائه رمز اجرا کند، اجرای Package با خطأ مواجه می‌شود.

3-1-5 Encrypt sensitive with user key به منظور رمزگذاری تنها مقادیر ویژگی‌های حساس در Package از یک کلید که مبتنی بر Profile کاربر جاری می‌باشد، استفاده می‌شود. تنها کاربری که از همان Profile استفاده می‌کند، Package را می‌تواند باز کردد (load). اگر کاربر متفاوتی Package را باز نماید، اطلاعات حساس با مقادیر پوچی جایگزین می‌شود و کاربر باید مقادیر جدیدی برای داده‌های حساس فراهم کند. اگر کاربر سعی نماید Package را بدون ارائه رمز اجرا کند، اجرای Package با خطأ مواجه می‌شود. برای رمزگذاری از DPAPI استفاده می‌شود.

3-1-6 Rely on server storage for encryption (ServerStorage) با استفاده از نقش‌های بانک اطلاعاتی، SQL Server تمامی Package را حفاظت می‌کند. این گزینه تنها زمانی پشتیبانی می‌شود که Package در بانک اطلاعاتی msdb ذخیره شده است.

4- استفاده از نقش‌های Integration Service برای کنترل کردن دستیابی به SSIS Package، SSIS شامل 3 نقش ثابت در سطح بانک اطلاعاتی است. نقش‌ها می‌توانند تنها روی Package هایی که در بانک اطلاعاتی msdb ذخیره شده اند، بکار روند. با استفاده از SSMS می‌توانید نقش‌ها را به Package ها نسبت دهید، این انتساب نقش‌ها در بانک اطلاعاتی msdb ذخیره می‌شود.

Role	Read action	Write action
db_ssisadmin or sysadmin	Enumerate own packages	
	Enumerate all packages	Import packages
	View own packages	Delete own packages
	View all packages	Delete all packages
	Execute own packages	Change own package roles
	Execute all packages	Change all package roles
	Export own packages	
	Export all packages	* به نکته رجوع شود
	Execute all packages in SQL Server Agent	
db_ssisltduser	Enumerate own packages	Import packages
	Enumerate all packages	Delete own packages
	View own packages	Change own package roles
	Execute own packages	
	Export own packages	
db_ssisoperator	Enumerate all packages	
	View all packages	
	Execute all packages	
	Export all packages	
	Execute all packages in SQL Server Agent	None
Windows administrators	View execution details of all running packages	Stop all currently running packages

* نکته: اعضای نقش‌های dc_admin و db_ssisadmin ممکن است قادر باشند مجوزهای خودشان را تا سطح sysadmin ارتقا دهند. براساس این ترفیع مجوز امکان اصلاح و اجرای Package‌ها از طریق SQL Server Agent میسر می‌شود. برای محافظت در برابر این ارتقا، با استفاده از یک account (حساب) با دسترسی محدود، Job هایی که این Package‌ها را اجرا می‌کنند، پیکربندی شوند یا تنها اعضای نقش sysadmin به نقش‌های dc_admin و db_ssisadmin افزوده شوند.

همچنین جدول sysssispackages در بانک اطلاعاتی msdb شامل Package هایی است که در SQL Server ذخیره می‌شوند. این جدول شامل ستون هایی که اطلاعاتی درباره نقش هایی که به Package‌ها نسبت داده شده است، می‌باشد. به صورت پیش فرض، مجوزهای نقش‌های ثابت بانک اطلاعاتی db_ssisoperator و db_ssisadmin و شناسه منحصر به فرد کاربری (unique security identifier) که Package را ایجاد کرده برای خواندن Package بکار می‌رود، و مجوزهای نقش db_ssisadmin و شناسه منحصر به فرد کاربری که Package را ایجاد کرده برای نوشتگی Package به کار می‌رود. یک User باید عضو نقش db_ssisoperator و db_ssisltduser یا db_ssisadmin باشد. یک User باید عضو نقش db_ssisadmin برای داشتن دسترسی نوشتگی Package باشد.

5- اتصال به صورت Remote به Integration Service زمانی که یک کاربر بدون داشتن دسترسی کافی تلاش کند به یک Integration Service به صورت Remote متصل شود، با پیغام خطای "Access is denied" مواجه می‌شود. برای اجتناب از این پیغام خطای می‌توان تضمین کرد که کاربر مجوز مورد نیاز DCOM را دارد. به منظور پیکربندی کردن دسترسی کاربر به صورت Remote به سرویس Integration مراحل زیر را دنبال کنید:

Component Service - dcomcnfg را باز نمایید (در Run عبارت dcomcnfg را تایپ کنید).

اجرای SSIS Package از طریق برنامه کاربردی

- گره Component Service را باز کنید، گره Computer و سپس My Computer را باز نمایید و روی DCOM Config کلیک نمایید.
 - گره DCOM Config را باز کنید و از لیست برنامه هایی که می توانند پیکربندی شوند MsDtsServer را انتخاب کنید.
 - روی Properties برنامه MsDtsServer رفته و قسمت Security را انتخاب کنید.
 - در قسمت Lunch and Activation Permissions، مورد Edit را انتخاب و سپس روی Customize کلیک نمایید تا پنجره Lunch باز شود.
 - در پنجره Lunch Permission، کاربران را اضافه و یا حذف کنید و مجوزهای مناسب را به کاربران یا گروههای مناسب نسبت دهید. مجوزهای موجود عبارتند از Local Activation، Remote Activation و Remote Lunch.
 - در قسمت Access Permission مراحل فوق را به منظور نسبت دادن مجوزهای مناسب به کاربران یا گروههای مناسب انجام دهید.
 - سرویس Integration را Restart کنید.
- مجوز دسترسی Lunch به منظور شروع و خاتمه سرویس، اعطای را رد می شود و مجوز دسترسی Activation به منظور متصل شدن به سرویس، اعطای (grant) یا رد (deny) می شود.

6- پیاده سازی در ابتدا به ایجاد یک CLR Stored Procedures پرداخته می شود نام اسمبلی ساخته شده به این نام RunningPackage.dll می باشد و حاوی کد زیر است:

```
Partial Public Class StoredProcedures
    '-----
    'exec dbo.Spc_NtDtexec 'Package','ssis','ssis@ssis','1234512345'
    '-----

<Microsoft.SqlServer.Server.SqlProcedure()
Public Shared Sub Spc_NtDtexec(ByVal PackageName As String, _
    ByVal UserName As String, _
    ByVal Password As String, _
    ByVal Decrypt As String)
    Dim p As New System.Diagnostics.Process()
    p.StartInfo.FileName = "C:\Program Files\Microsoft SQL Server\100\DTExec.exe"
    p.StartInfo.RedirectStandardOutput = True
    p.StartInfo.Arguments = "/sql " & PackageName & " /User " & UserName & " /Password " & Password
    & " /De " & Decrypt
    p.StartInfo.UseShellExecute = False
    p.Start()
    p.WaitForExit()
    Dim output As String
    output = p.StandardOutput.ReadToEnd()
    Microsoft.SqlServer.Server.SqlContext.Pipe.Send(output)
End Sub
End Class
```

در حقیقت توسط این روش به اجرای برنامه dtexec.exe و ارسال پارامترهای مورد نیاز جهت اجرا پرداخته می شود. با توجه به توضیحات تئوری بیان شده، سطح حفاظتی Package ایجاد شده Encrypt all with password توصیه می شود که رمز مذکور در قالب یکی از پارامتر ارسالی به رویه ساخته شده موسوم به Spc_NtDtexec ارسال می گردد.

در قدم بعدی نیاز به Register کردن dll ساخته شده در سطح بانک اطلاعاتی SQL Server است، این گامها پس از اتصال به Server Management Studio به شرح زیر است:

1- فعال کردن CLR در سرویس SQL Server

```
SP_CONFIGURE 'clr enabled',1
GO
RECONFIGURE
```

2- فعال کردن ویژگی TRUSTWORTHY در بانک اطلاعاتی مورد نظر

```
ALTER DATABASE <Database Name> SET TRUSTWORTHY ON
GO
RECONFIGURE
```

3- ایجاد Stored Procedure و Assembly در بانک اطلاعاتی مورد نظر

در ریشه C: ساخته شده با نام RunningPacakge.dll Assembly در وجود آن لزومی به وجود آن نمیباشد.

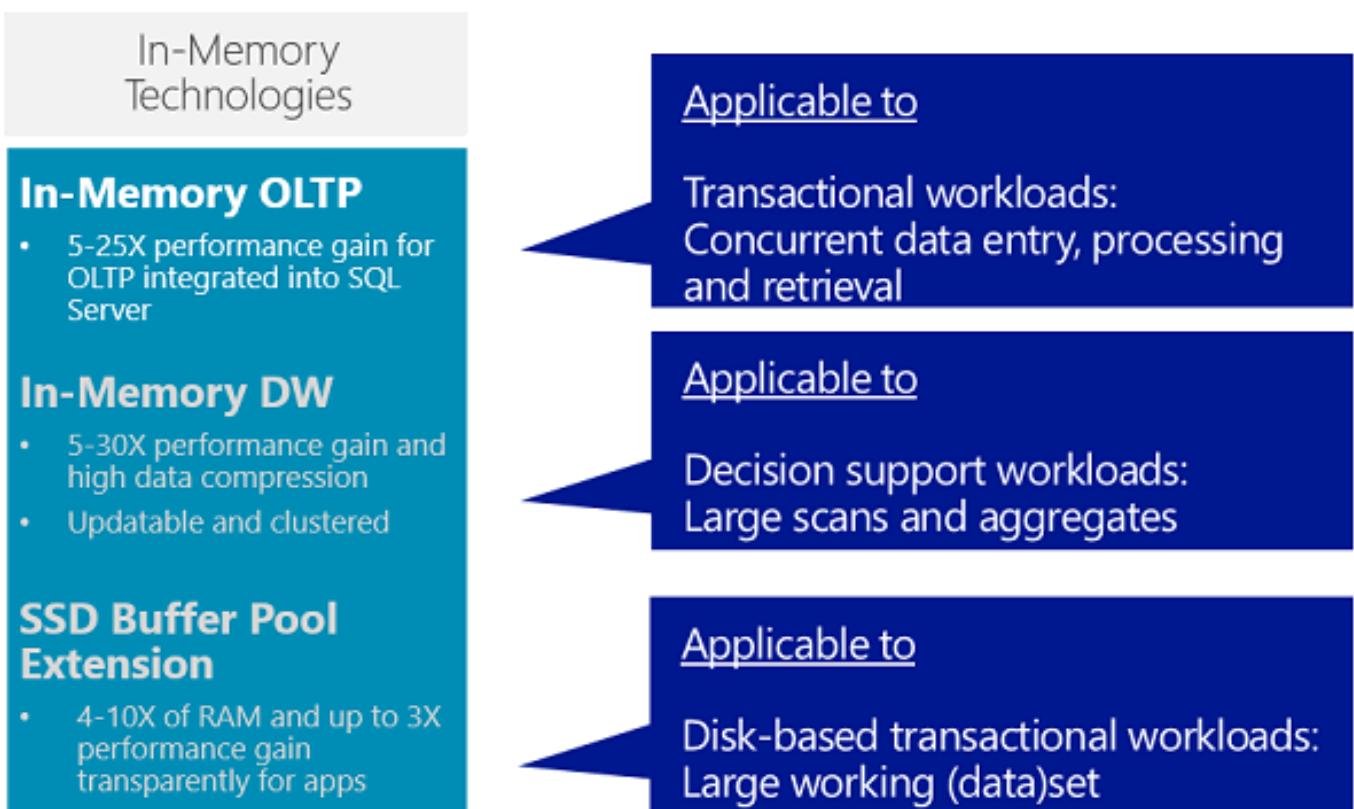
```
USE <Database Name>
GO
CREATE ASSEMBLY [RunningPackage]
AUTHORIZATION [dbo]
FROM 'C:\RunningPackage.dll'
WITH PERMISSION_SET = UNSAFE
Go
CREATE PROCEDURE [dbo].[Spc_NtDtexec]
@PackageName [nvarchar](50),
@UserName [nvarchar](50),
@Password [nvarchar](50),
@Decrypt [nvarchar](50)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME [RunningPackage].[RunningPackage.StoredProcedures].[Spc_NtDtexec]
GO
```

توجه: برنامه بایست دسترسی اجرای رویه ذخیره شده Spc_NtDtexec را در بانک اطلاعاتی مورد نظر داشته باشد همچنین بایست عضو نقش db_ssoperator در بانک اطلاعاتی msdb باشد. (منظور از Application User، لایگین است که در برنامه قرار داده اید). Connection string در بردازید.

در برنامه کاربردی تان کافی است متده شکل زیر ایجاد و با توجه به نیازتان در برنامه به فراخوانی آن و اجرای Package پردازید.

```
Private Sub ExecutePackage()
    Dim oSqlConnection As SqlClient.SqlConnection
    Dim oSqlCommand As SqlClient.SqlCommand
    Dim strCnt As String = String.Empty
    strCnt = "Data Source=" & txtServer.Text & ";User ID=" & txtUsername.Text & ";Password=" &
    txtPassword.Text & ";Initial Catalog=" & cmbDatabaseName.SelectedValue.ToString() & ";"
    Try
        oSqlConnection = New SqlClient.SqlConnection(strCnt)
        oSqlCommand = New SqlClient.SqlCommand
        With oSqlCommand
            .Connection = oSqlConnection
            .CommandType = System.Data.CommandType.StoredProcedure
            .CommandText = "dbo.Spc_NtDtexec"
            .Parameters.Clear()
            .Parameters.Add("@PackageName", System.Data.SqlDbType.VarChar, 50)
            .Parameters.Add("@UserName", System.Data.SqlDbType.VarChar, 50)
            .Parameters.Add("@Password", System.Data.SqlDbType.VarChar, 50)
            .Parameters.Add("@Decrypt", System.Data.SqlDbType.VarChar, 50)
            .Parameters("@PackageName").Value = txtPackageName.Text.Trim()
            .Parameters("@UserName").Value = txtUsername.Text.Trim()
            .Parameters("@Password").Value = txtPassword.Text.Trim()
            .Parameters("@Decrypt").Value = txtDecrypt.Text.Trim()
        End With
        If (oSqlCommand.Connection.State <> System.Data.ConnectionState.Open) Then
            oSqlCommand.Connection.Open()
            oSqlCommand.ExecuteNonQuery()
            System.Windows.MessageBox.Show("Success")
        End If
        If (oSqlCommand.Connection.State = System.Data.ConnectionState.Open) Then
            oSqlCommand.Connection.Close()
        End If
        Catch ex As Exception
            MessageBox.Show(ex.Message, "Error")
        End Try
    End Sub 'ExecutePackage
```

درون حافظه‌ای، مهم‌ترین ویژگی جدید SQL Server 2014 است. موتور بانک اطلاعاتی disk based اس کیوال سرور، حدود ۱۵ تا ۲۰ سال قبل تهیه شده‌است و موتور جدید درون حافظه‌ای OLTP آن، بزرگترین بازنویسی این سیستم از زمان ارائه‌ی آن می‌باشد و شروع این پروژه به ۵ سال قبل بر می‌گردد. علت تهیه‌ی آن نیز به نیازهای بالای پردازش‌های همزمان مصرف کنندگان این محصول در سال‌های اخیر، نسبت به ۱۵ سال قبل مرتبط است. با استفاده از امکانات OLTP درون حافظه‌ای، امکان داشتن جداول معمولی disk based و جداول جدید memory optimized با هم در یک بانک اطلاعاتی میسر است؛ به همراه مهیا بودن تمام زیرساخت‌هایی مانند تهیه بک آپ، بازیابی آن‌ها، امنیت و غیره برای آن‌ها.



آیا جداول بهینه سازی شده‌ی برای حافظه، همان DBCC PINTABLE منسوخ شده هستند؟

در نگارش‌های قدیمی‌تر اس کیوال سرور، دستوری وجود داشت به نام [DBCC PINTABLE](#) که سبب ثابت نگه داشتن صفحات جداول مبتنی بر دیسک یک دیتابیس، در حافظه می‌شد. به این ترتیب تمام خواندن‌های مرتبط با آن جدول، از حافظه صورت می‌گرفت. مشکل این روش که سبب منسوخ شدن آن گردید، اثرات جانبی آن بود؛ مانند خوانده شدن صفحات جدیدتر (با توجه به اینکه ساختار پردازشی و موتور بانک اطلاعاتی تغییری نکرده بود) و نیاز به حافظه بیشتر تا حدی که کل کش بافر سیستم را پر می‌کرد و امکان انجام سایر امور آن مختل می‌شدند. همچنین اولین ارجاعی به یک جدول، سبب قرار گرفتن کل آن در حافظه می‌گشت. به علاوه ساختار این سیستم نیز همانند روش مبتنی بر دیسک، بر اساس همان روش‌های قفل گذاری، ذخیره سازی اطلاعات و تهیه ایندکس‌های متداول بود.

اما جداول بهینه سازی شده‌ی برای حافظه، از یک موتور کاملاً جدید استفاده می‌کنند؛ با ساختار جدیدی برای ذخیره سازی اطلاعات و تهیه ایندکس‌ها، دسترسی به اطلاعات آن‌ها شامل قفل گذاری‌های متداول نیست و در آن حداقل زمان دسترسی به

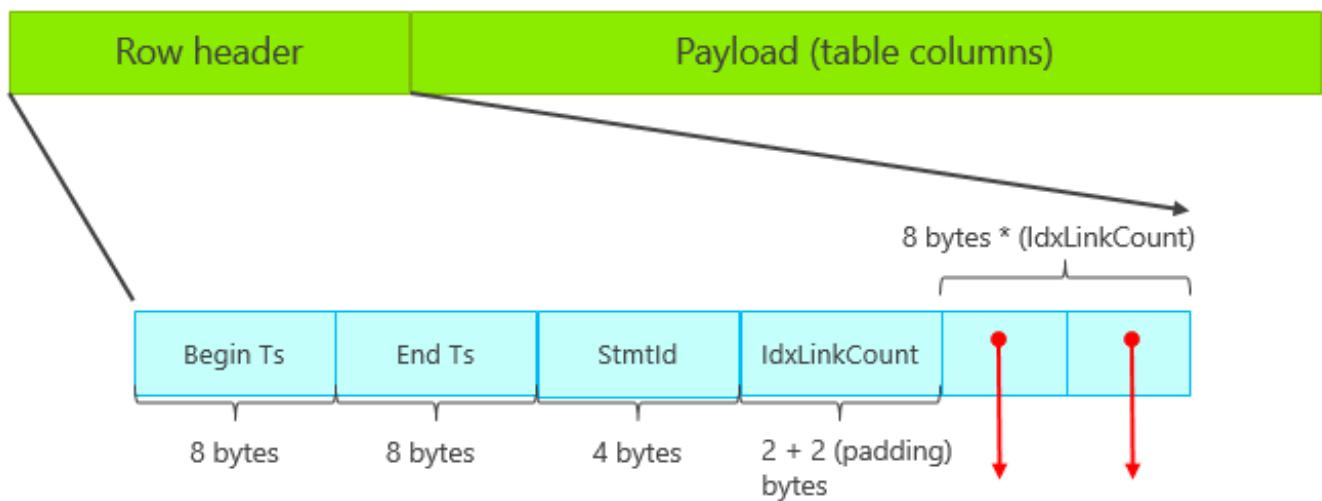
اطلاعات درنظر گرفته شده است. همچنین در آن‌ها data pages یا index pages و کش بافر نیز وجود ندارد.

نحوه ذخیره سازی و مدیریت اطلاعات جداول بهینه سازی شده برای حافظه

جداول بهینه سازی شده برای حافظه، فرمت ردیف‌های کاملاً جدیدی را نیز به همراه دارند و جهت قرارگرفتن در حافظه و دسترسی سریع به آن‌ها بهینه سازی شده‌اند. برخلاف جداول مبتنی بر دیسک سخت که اطلاعات آن‌ها در یک سری صفحات خاص به نام‌های data or index pages ذخیره می‌شوند، اینگونه جداول، دارای ظروف مبتنی بر صفحه نیستند و از مفهوم چند نگارشی برای ذخیره سازی اطلاعات استفاده می‌کنند؛ به این معنا که ردیف‌ها به ازای هر تغییری، دارای یک نگارش جدید خواهند بود و بلاfaciale در همان نگارش اصلی به روز رسانی نمی‌شوند.

در اینجا هر ردیف دارای یک timestamp شروع بیانگر تراکنشی است که ردیف را ثبت کرده و timestamp پایان برای مشخص سازی تراکنشی بکار می‌رود که ردیف را حذف کرده است. اگر timestamp پایان، دارای مقدار بی‌نهایت باشد، به این معنا است که ردیف متناظر با آن هنوز حذف نشده است. به روز رسانی یک ردیف در اینجا، ترکیبی است از حذف یک ردیف موجود و ثبت ردیفی جدید. برای یک عملیات فقط خواندن، تنها نگارش‌هایی که timestamp معتبری داشته باشند، قابل مشاهده خواهند بود و از مابقی صرفنظر می‌گردد.

در درون حافظه‌ای که از روش چندنگارشی همزمانی استفاده می‌کند، برای یک ردیف مشخص، ممکن است چندین نگارش وجود داشته باشند؛ بسته به تعداد باری که یک رکورد به روز رسانی شده است. در اینجا یک سیستم garbage collection همیشه فعال، نگارش‌هایی را که توسط هیچ تراکنشی مورد استفاده قرار نمی‌گیرند، به صورت خودکار حذف می‌کند؛ تا مشکل کمبود حافظه رخ ندهد.



آیا می‌توان به کارآیی جداول بهینه سازی شده برای حافظه با همان روش متداول مبتنی بر دیسک اما با بکارگیری حافظه بیشتر و استفاده از یک SSD RAID رسید؟

خیر! حتی اگر کل بانک اطلاعاتی مبتنی بر دیسک را در حافظه قرار دهید به کارآیی روش جداول بهینه سازی شده‌ی برای حافظه نخواهید رسید. زیرا در آن هنوز مفاهیمی مانند index pages و data pages به همراه یک buffer pool پیچیده وجود دارند. در روش‌های مبتنی بر دیسک، ردیف‌ها از طریق page id و row offset آن‌ها قابل دسترسی می‌شوند. اما در جداول بهینه سازی شده‌ی برای حافظه، ردیف‌های جداول با یک B-tree خاص به نام Bw-Tree در دسترس هستند.

میزان حافظه‌ی مورد نیاز برای جداول بهینه سازی شده‌ی برای حافظه

باید درنظر داشت که تمام جداول بهینه سازی شده‌ی برای حافظه، به صورت کامل در حافظه ذخیره خواهند شد. بنابراین بدیهی

است که نیاز به مقدار کافی حافظه در اینجا ضروری است. توصیه صورت گرفته، داشتن حافظه‌ای به میزان دو برابر اندازه اطلاعات است. البته در اینجا چون با یک سیستم هیرید سر و کار داریم، حافظه‌ی کافی جهت کار buffer pool مختص به جداول مبتنی بر دیسک را نیز باید درنظر داشت.

همچنین اگر به اندازه‌ی کافی حافظه در سیستم تعیه نشود، شاهد شکست مدام تراکنش‌ها خواهد بود. به علاوه امکان بازیابی و restore جداول را نیز از دست خواهد داد.

البته لازم به ذکر است که اگر کل بانک اطلاعاتی شما چند تراپایت است، نیازی نیست به همین اندازه یا بیشتر حافظه تهیه کنید. فقط باید به اندازه‌ی جداولی که قرار است جهت قرار گرفتن در حافظه بهینه سازی شوند، حافظه تهیه کنید که حداقل آن 256 گیگابایت است.

چه برنامه‌هایی بهتر است از امکانات OLTP درون حافظه‌ای SQL Server 2014 استفاده کنند؟

- برنامه‌هایی که در آن‌ها تعداد زیادی تراکنش کوتاه مدت وجود دارد به همراه درجه‌ی بالای از تراکنش‌های همزمان توسط تعداد زیادی کاربر.

- اطلاعاتی که توسط برنامه زیاد مورد استفاده قرار می‌گیرند را نیز می‌توان در جداول بهینه سازی شده جهت حافظه قرار داد.

- زمانیکه نیاز به اعمال دارای write بسیار سریع و با تعداد زیاد است. چون در جداول بهینه سازی شده برای حافظه، صفات داده‌ها و ایندکس‌ها وجود ندارند، نسبت به حالت مبتنی بر دیسک، بسیار سریعتر هستند. در روش‌های متداول، برای نوشتن اطلاعات در یک صفحه، مباحث همزمانی و قفل‌گذاری آنرا باید در نظر داشت. در صورتیکه در روش بهینه سازی شده برای حافظه، به صورت پیش فرض از حالتی همانند [snapshot isolation](#) و همزمانی مبتنی بر نگارش‌های مختلف رکورد استفاده می‌شود.

- تنظیم و بهینه سازی جداولی با تعداد Read بالا. برای مثال، جداول پایه سیستم که اطلاعات تعاریف محصولات در آن قرار دارند. این نوع جداول عموماً با تعداد Read‌های بالا و تعداد Write کم شناخته می‌شوند. چون طراحی جداول مبتنی بر حافظه از hash و اشاره‌گرهایی برای دسترسی به رکوردهای موجود استفاده می‌کند، اعمال Read آن نیز بسیار سریعتر از حالت معمول هستند.

- مناسب جهت کارهای data warehouse و ETL Staging Table. در جداول مبتنی بر حافظه امکان عدم ذخیره سازی اطلاعات بر روی دیسک سخت نیز پیش بینی شده است. در این حالت فقط اطلاعات ساختار جدول، ذخیره‌ی نهایی می‌گردد و اگر سرور نیز ری استارت گردد، مجدداً می‌تواند اطلاعات خود را از منابع اصلی data warehouse تامین کند.

محدودیت‌های جداول بهینه سازی شده برای حافظه در SQL Server 2014

- تغییر اسکیما و ساختار جداول بهینه سازی شده برای حافظه مجاز نیست. به بیان دیگر دستور ALTER TABLE برای اینگونه جداول کاربردی ندارد. این مورد جهت ایندکس‌ها نیز صادق است. همان زمانیکه جدول ایجاد می‌شود، باید ایندکس آن نیز تعریف گردد و پس از آن این امکان وجود ندارد.

نهایاً راه تغییر اسکیما اینگونه جداول، Drop و سپس ایجاد مجدد آن‌ها است.

البته باید درنظر داشت که SQL Server 2014 اولین نگارش این فناوری را ارائه داده است و در نگارش‌های بعدی آن، بسیاری از این محدودیت‌ها قرار است که برطرف شوند.

- جداول بهینه سازی شده برای حافظه حتماً باید دارای یک ایندکس باشند. البته اگر یک primary key را برای آن‌ها تعریف نمائید، کفایت می‌کند.

- از unique index پشتیبانی نمی‌کند، مگر اینکه از نوع primary key باشد.

- حداقل 8 ایندکس را می‌توان بر روی اینگونه جداول تعریف کرد.

- امکان تعریف ستون identity در آن وجود ندارد. اما می‌توان از قابلیت [sequence](#) برای رسیدن به آن استفاده کرد.

- DML triggers را پشتیبانی نمی‌کند.

- کلیدهای خارجی و قیود را پشتیبانی نمی‌کند.

- حداقل اندازه‌ی یک ردیف آن 8060 بایت است. بنابراین از نوع‌های داده‌ای max دار و XML پشتیبانی نمی‌کند.

این مورد در حین ایجاد جدول بررسی شده و اگر اندازه‌ی ردیف محاسبه‌ی شده‌ی آن توسط SQL Server 2014 بیش از 8060 بایت باشد، جدول را ایجاد نخواهد کرد.

اگر سرور را ری استارت کنیم، چه اتفاقی برای اطلاعات جداول بهینه سازی شده‌ی برای حافظه رخ می‌دهد؟

حالت DURABILITY انتخاب شده‌ی در حین ایجاد جدول بهینه سازی شده‌ی برای حافظه، تعیین کننده‌ای این مساله است. اگر انتخاب شده باشد، کل اطلاعات شما با ری استارت سرور از دست خواهد رفت؛ البته اطلاعات ساختار جدول حفظ خواهد گردید. اگر حالت SCHEMA_ONLY انتخاب شود، اطلاعات شما پس از ری استارت سرور نیز در دسترس خواهد بود. این اطلاعات به صورت خودکار از لاغ تراکنش‌ها بازیابی شده و مجدداً در حافظه قرار می‌گیرند.

حالت SCHEMA_ONLY برای مصارف برنامه‌های data warehouse بیشتر کاربرد دارد. جایی که اطلاعات قرار است از منابع داده‌ی مختلفی تامین شوند.

برای مطالعه بیشتر

[SQL Server 2014: NoSQL Speeds with Relational Capabilities](#)

[SQL Server 2014 In-Memory OLTP Architecture and Data Storage](#)

[Overview of Applications, Indexes and Limitations for SQL Server 2014 In-Memory OLTP Tables](#)

[Microsoft SQL Server 2014: In-Memory OLTP Overview](#)

[SQL Server in Memory OLTP for Database Developers](#)

[Exploring In-memory OLTP Engine \(Hekaton\) in SQL Server 2014 CTP1](#)

نظرات خوانندگان

نویسنده: فرید طاهری
تاریخ: ۱۳۹۳/۰۳/۱۱ ۱۰:۳

جناب نصیری با تشکر از مقاله مفیدتون لازم می‌دونم در جهت تکمیل مباحثت به چند نکته اشاره کنم

1- «اگر حالت SCHEMA_AND_DATA انتخاب شود، اطلاعات شما پس از ریاستارت سرور نیز در دسترس خواهد بود. این اطلاعات به صورت خودکار از لاغ تراکنش‌ها بازیابی شده و مجدداً در حافظه قرار می‌گیرند.».

بازیابی اطلاعات مربوط به تراکنش‌هایی که به ازای In Memory OLTP است بواسیله Log File و Data File + Delta File در صورتیکه Schema_AND_Data را به ازای این نوع جداول فعال کنید داده‌های شما در Data File و داده‌های حذف شده در Delta File ثبت می‌گردد. مکانیزم Log File برای In Memory OLTP همچنان مانند جداول Disk base وجود دارد اما با بهینه سازی مناسب مانند ثبت Log Record کمتر به ازای عملیات کاربران و ...

2- در جایی دیگر در متن اشاره شده که اجازه استفاده از Identity In Memory OLTP را به کاربر نمی‌دهد باید اشاره کنم که این موضوع برای نسخه CTP بوده است در نسخه RTM این قابلیت وجود دارد. لازم می‌دانم اشاره کنم که در Books Online شده که امکان استفاده وجود ندارد و در جایی هم گفته شده وجود دارد.

به مثال زیر دقت کنید

```
CREATE TABLE test(
    [ID] BIGINT IDENTITY(1,1) NOT NULL PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT=10000),
    N1 NVARCHAR(100),
    N2 NVARCHAR(100),
    N3 NVARCHAR(100)
) WITH (MEMORY_OPTIMIZED=ON,DURABILITY = SCHEMA_AND_DATA)
GO
```

این مثال در SQL Server 2014 RTM Edition قابل اجرا است اما یکسری محدودیت داریم. مثلاً مقدار شروع و گام افزایش باید 1 باشد. باز هم از مطالب خوب شما متشرکم مقاله مفیدی بود.

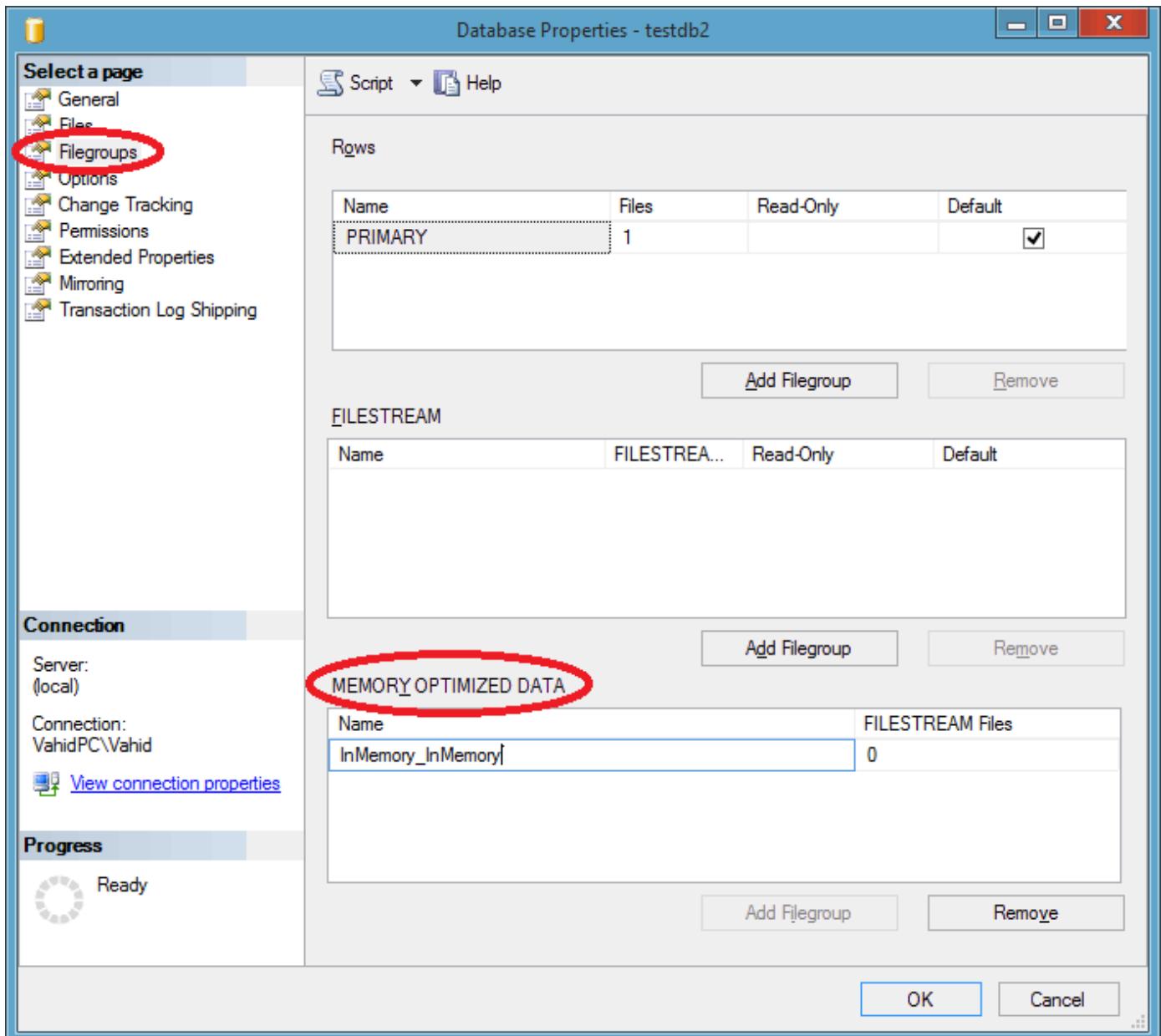
عنوان:	ایجاد جداول بهینه سازی شده برای حافظه در SQL Server 2014
نوبت‌نده:	وحید نصیری
تاریخ:	۲۱:۵ ۱۳۹۳/۰۳/۱۱
آدرس:	www.dotnettips.info
گروه‌ها:	SQL Server, In-Memory OLTP

پس از [نگاهی به مفاهیم مقدماتی OLTP درون حافظه‌ای در 2014 SQL Server](#)، در ادامه به نحوه انجام تنظیمات خاص جداول بهینه سازی شده برای حافظه خواهیم پرداخت.

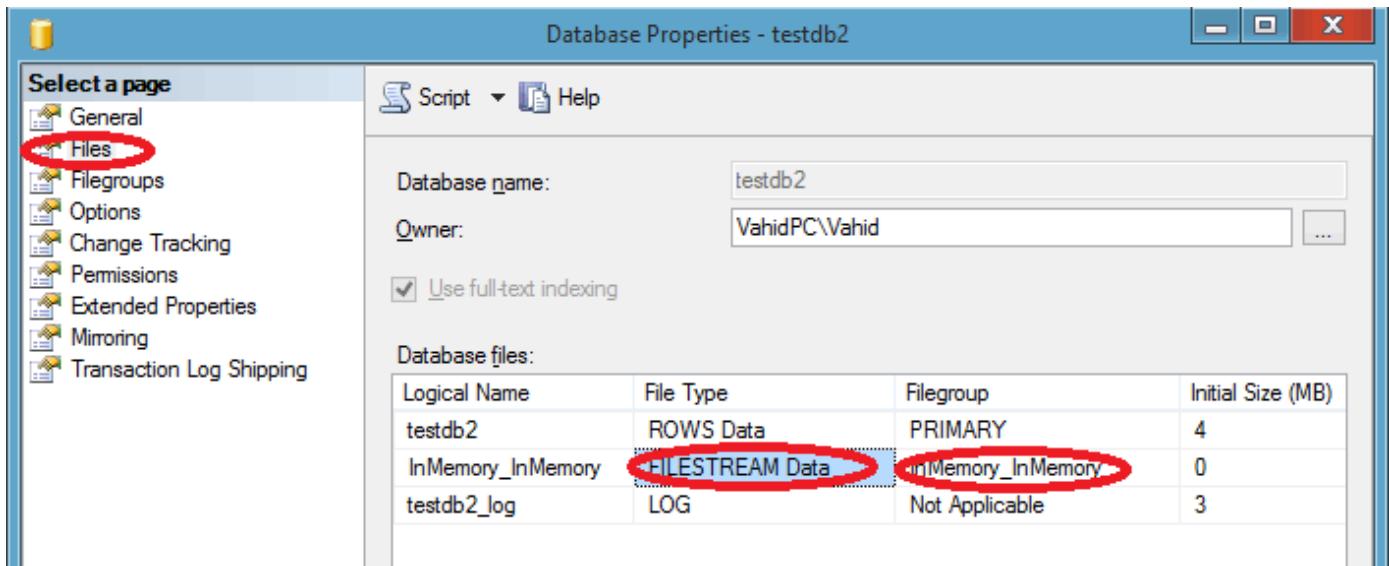
ایجاد یک بانک اطلاعاتی با پشتیبانی از جداول بهینه سازی شده برای حافظه

برای ایجاد جداول بهینه سازی شده برای حافظه، ابتدا نیاز است تا تنظیمات خاصی را به بانک اطلاعاتی آن اعمال کنیم. برای اینکار می‌توان یک بانک اطلاعاتی جدید را به همراه یک filestream filegroup ایجاد کرد که جهت جداول بهینه سازی شده برای حافظه، ضروری است؛ یا اینکه با تغییر یک بانک اطلاعاتی موجود و افزودن filegroup یاد شده نیز می‌توان به این مقصود رسید. در اینگونه جداول خاص، اطلاعات در حافظه‌ی سیستم ذخیره می‌شوند و برخلاف جداول مبتنی بر دیسک سخت، صفحات اطلاعات وجود نداشته و نیازی نیست تا به کش بافر وارد شوند. برای مقاصد ذخیره سازی نهایی اطلاعات جداول بهینه سازی شده برای حافظه، موتور OLTP درون حافظه‌ای آن، فایل‌های خاصی را به نام checkpoint در یک filestream filegroup ایجاد می‌کند که از آن‌ها جهت ردیابی اطلاعات استفاده خواهد کرد و نحوی ذخیره سازی اطلاعات در آن‌ها از شیوه‌ی با کارآیی بالایی به نام append only mode با توجه به متفاوت بودن نحوه ذخیره سازی ذخیره سازی اینگونه اطلاعات اینگونه جداول و دسترسی به آن‌ها از طریق استریم‌ها، توصیه شده‌است که filestream filegroup‌های تهیه شده را در یک Solid State Drive قرار دهید.

پس از اینکه بانک اطلاعاتی خود را به روش‌های معمول ایجاد کردید، به برگه‌ی خواص آن در management studio مراجعه کنید. سپس صفحه‌ی file groups را در آن انتخاب کرده و در پایین برگه‌ی آن، در قسمت جدید memory optimized data، بر روی دکمه‌ی Add کلیک کنید. سپس نام دلخواهی را وارد نمائید.



پس از ایجاد یک گروه فایل جدید، به صفحه‌ی **files** خواص بانک اطلاعاتی مراجعه کرده و بر روی دکمه‌ی **Add** کلیک کنید. سپس **File type** این ردیف اضافه شده را از نوع **file stream data** و آنرا همان گروه فایلی که پیشتر ایجاد کردیم، تنظیم کنید. در ادامه **logical name** دلخواهی را وارد کرده و در آخر بر روی دکمه‌ی **Ok** کلیک کنید تا تنظیمات مورد نیاز جهت تعاریف جدول بهینه سازی شده برای حافظه به پایان برسد.



این مراحل را توسط دو دستور T-SQL ذیل نیز می‌توان سریعتر انجام داد:

```
USE [master]
GO
ALTER DATABASE [testdb2]
    ADD FILEGROUP [InMemory_InMemory] CONTAINS MEMORY_OPTIMIZED_DATA
GO
ALTER DATABASE [testdb2]
    ADD FILE ( NAME = N'InMemory_InMemory', FILENAME =
N'D:\SQL_Data\MSSQL11.MSSQLSERVER\DATA\InMemory_InMemory' )
        TO FILEGROUP [InMemory_InMemory]
GO
```

ساختار گروه فایل بهینه سازی شده برای حافظه

گروه فایل بهینه سازی شده برای حافظه، دارای چندین دربرگیرنده است که هر کدام چندین فایل را در خود جای خواهند داد:

- Root File که در برگیرنده‌ی متادیتای اطلاعات است.
- Data File که شامل ردیف‌های اطلاعات ثبت شده در جداول بهینه سازی شده‌ی برای حافظه هستند. این ردیف‌ها همواره به انتهای data file می‌شوند و دسترسی به آن‌ها ترتیبی است. کارآیی IO این روش نسبت به روش دسترسی انفاقی به مراتب بالاتر است. حداکثر اندازه این فایل 128 مگابایت است و پس از آن یک فایل جدید ساخته می‌شود.
- Delta File شامل ردیف‌هایی است که حذف شده‌اند. به ازای هر ردیف، حداقل اطلاعاتی از آن را در خود ذخیره خواهد کرد؛ شامل ID ردیف حذف شده و شماره تراکنش آن. همانطور که پیشتر نیز ذکر شد، این موتور جدید درون حافظه‌ای، برای یافتن راه چاره‌ای جهت به حداقل رسانی قفل گذاری بر روی اطلاعات، چندین نگارش از ردیف‌ها را به همراه timestamp آن‌ها در خود ذخیره می‌کند. به این ترتیب، هر به روز رسانی به همراه یک حذف و سپس ثبت جدید است. به این ترتیب دیگر بانک اطلاعاتی نیازی نخواهد داشت تا به دنبال رکورد موجود برگردد و سپس اطلاعات آن را به روز نماید. این موتور جدید فقط اطلاعات به روز شده را در انتهاهای رکوردهای موجود با فرمت خود ثبت می‌کند.

ایجاد جداول بهینه سازی شده برای حافظه

پس از آماده سازی بانک اطلاعاتی خود و افزودن گروه فایل استریم جدیدی به آن برای ذخیره سازی اطلاعات جداول بهینه سازی شده برای حافظه، اکنون می‌توانیم اینگونه جداول خاص را در کنار سایر جداول متداول موجود، تعریف و استفاده نمائیم:

```
-- It is not a Memory Optimized
CREATE TABLE tblNormal
(
    [CustomerID] int NOT NULL PRIMARY KEY NONCLUSTERED,
```

```
[Name] nvarchar(250) NOT NULL,
CustomerSince DATETIME not NULL
    INDEX [ICustomerSince] NONCLUSTERED
)
-- DURABILITY = SCHEMA_AND_DATA
CREATE TABLE tblMemoryOptimized_Schema_And_Data
(
    [CustomerID] INT NOT NULL
PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT = 1000000),
    [Name] NVARCHAR(250) NOT NULL,
    [CustomerSince] DATETIME NOT NULL
INDEX [ICustomerSince] NONCLUSTERED
) WITH (MEMORY_OPTIMIZED = ON, DURABILITY = SCHEMA_AND_DATA)

-- DURABILITY = SCHEMA_ONLY
CREATE TABLE tblMemoryOptimized_Schema_Only
(
    [CustomerID] INT NOT NULL
PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT = 1000000),
    [Name] NVARCHAR(250) NOT NULL,
    [CustomerSince] DATETIME NOT NULL
INDEX [ICustomerSince] NONCLUSTERED
) WITH (MEMORY_OPTIMIZED = ON, DURABILITY = SCHEMA_ONLY)
```

در اینجا سه جدول را مشاهده می‌کنید که در بانک اطلاعاتی آماده شده در مرحله‌ی قبل، ایجاد خواهند شد. مورد اول یک جدول معمولی است که از آن برای مقایسه سرعت ثبت اطلاعات با سایر جداول ایجاد شده، استفاده خواهد شد.

همانطور که مشخص است، دو جدول بهینه سازی شده برای حافظه، همان سه ستون جدول معمولی مبتنی بر دیسک سخت را دارا هستند؛ اما با این تفاوت‌ها:

- دارای ویژگی **MEMORY_OPTIMIZED = ON** می‌باشند. به این ترتیب اینگونه جداول نسبت به جداول متدائل مبتنی به دیسک سخت متمایز خواهند شد.
- دارای ویژگی **DURABILITY** بوده و توسط مقدار **SCHEMA_AND_DATA** آن مشخص می‌کنیم که آیا قرار است اطلاعات و ساختار جدول، ذخیره شوند یا تنها قرار است ساختار جدول ذخیره گردد (حالت **SCHEMA_ONLY**).
- بر روی ستون **Id** آن‌ها یک **hash index** ایجاد شده‌است که وجود آن ضروری است و در کل بیش از 8 ایندکس را نمی‌توان تعریف کرد.

برخلاف ایندکس‌های **B-tree** جداول مبتنی بر سخت دیسک، ایندکس‌های جداول بهینه سازی شده برای حافظه، اطلاعات را تکرار نمی‌کنند. این‌ها صرفا اشاره‌گرهایی هستند به ردیف‌های اصلی اطلاعات. به این معنا که این ایندکس‌ها لاغ نشده و همچنین بر روی سخت دیسک ذخیره نمی‌شوند. کار بازیابی بانک اطلاعاتی و آغاز آن به صورت خودکار انجام می‌شود. به همین جهت مباحثی مانند **index fragmentation** و **نگهداری ایندکس‌ها** دیگر در اینجا معرفی نمی‌کنند.

دو نوع ایندکس را در اینجا می‌توان تعریف کرد. اولین آن‌ها **hash index** است و دومین آن‌ها **range index**. هش ایندکس‌ها برای حالاتی که در کوئری‌ها از عملگر تساوی استفاده می‌شود بسیار مناسب هستند. برای عملگرهای مقایسه‌ای از ایندکس‌های بازه‌ای استفاده می‌شود.

همچنین باید دقت داشت که پس از ایجاد ایندکس‌ها، دیگر امکان تغییر آن‌ها و یا تغییر ساختار جدول ایجاد شده نیست.

همچنین ایندکس‌های تعریف شده در جداول بهینه سازی شده برای حافظه، تنها بر روی ستون‌هایی غیرنال پذیر از نوع **BIN2** مانند **int** و **datetime** قابل تعریف هستند. برای مثال اگر سعی کنیم بر روی ستون **Name** ایندکسی را تعریف کنیم، به این خطأ خواهیم رسید:

```
Indexes on character columns that do not use a *_BIN2 collation are not supported with indexes on memory optimized tables.
```

- در حین تعریف هش ایندکس‌ها، مقدار **BUCKET_COUNT** نیز باید تنظیم شود. هر **bucket** توسط مقداری که حاصل هش کردن یک ستون است مشخص می‌شود. کلیدهای منحصر‌بفرد دارای هش‌های یکسان در **bucket** یکسانی ذخیره می‌شوند. به همین جهت توصیه شده‌است که حداقل مقدار **bucket** تعیین شده در اینجا مساوی یا بیشتر از مقدار تعداد کلیدهای منحصر‌بفرد یک جدول باشد؛ مقدار پیش فرض 2 برابر توسط مایکروسافت توصیه شده است.
- نوع‌های قابل تعریف ستون‌ها نیز در اینجا به موارد ذیل محدود هستند و جمع طول آن‌ها از 8060 باید بیشتر شود:

```
bit, tinyint, smallint, int, bigint, money, smallmoney, float, real, datetime, smalldatetime,
datetime2,
date, time, numeric, decimal, char(n), varchar(n) ,nchar(n), nvarchar(n), sysname, binary(n),
varbinary(n), and Uniqueidentifier
```

Create Table DDL

```
CREATE TABLE [Customer](
    [CustomerID] INT NOT NULL
        PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT = 1000000),
    [Name] NVARCHAR(250) NOT NULL,
    [CustomerSince] DATETIME NULL
        INDEX [ICustomerSince] NONCLUSTERED
)
WITH (MEMORY_OPTIMIZED = ON, DURABILITY = SCHEMA_AND_DATA);
```

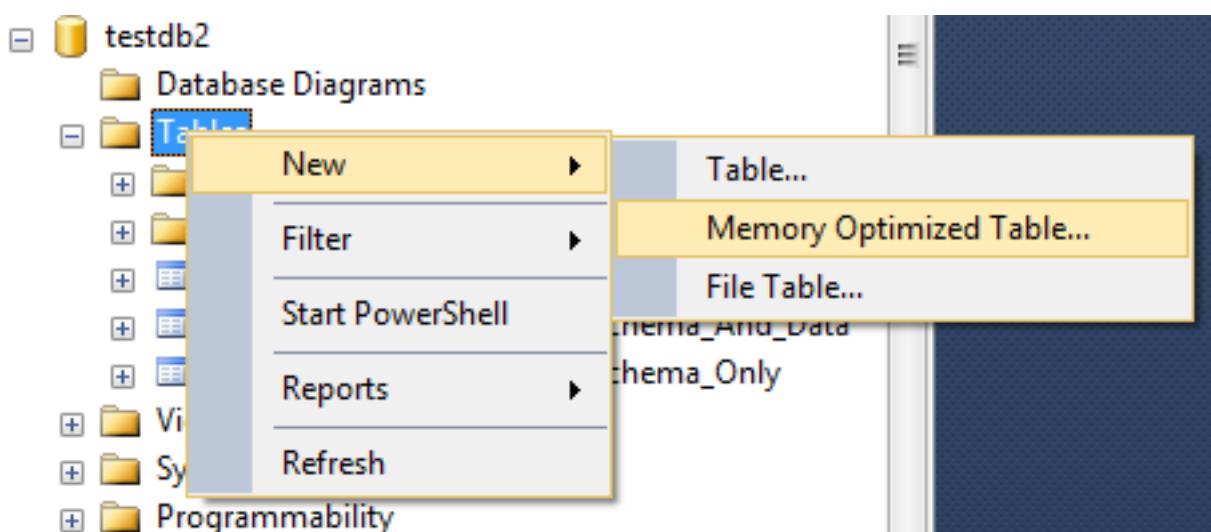
Hash Index
BUCKET_COUNT 1-2X nr of unique index key values

Indexes are specified inline
NONCLUSTERED indexes are supported

This table is memory optimized

This table is durable
Non-durable tables:
DURABILITY=SCHEMA_ONLY

همچنین در management studio، گزینه‌ی جدید new -> memory optimized table ای برای تهیه این نوع جداول، به صورت خودکار تولید گردد.



البته این گزینه تنها برای بانک‌های اطلاعاتی که دارای گروه فایل استریم مخصوص جداول بهینه سازی شده برای حافظه هستند،

فعال می باشد.

ثبت اطلاعات در جداول معمولی و بهینه سازی شده برای حافظه و مقایسه کارآیی آنها

در مثال زیر، 100 هزار رکورد را در سه جدولی که پیشتر ایجاد کردیم، ثبت کرده و سپس مدت زمان اجرای هر کدام از مجموعه عملیات را بر حسب میلی ثانیه بررسی می کنیم:

```

set statistics time off
SET STATISTICS IO Off
set nocount on
go
-----

Print 'insert into tblNormal'

DECLARE @start datetime = getdate()
declare @insertCount int = 100000
declare @startId int = 1
declare @customerID int = @startId

while @customerID < @startId + @insertCount
begin
    insert into tblNormal values (@customerID, 'Test', '2013-01-01T00:00:00')
    set @customerID +=1
end
Print DATEDIFF(ms,@start,getdate());
go
-----


Print 'insert into tblMemoryOptimized_Schema_And_Data'

DECLARE @start datetime = getdate()
declare @insertCount int = 100000
declare @startId int = 1
declare @customerID int = @startId

while @customerID < @startId + @insertCount
begin
    insert into tblMemoryOptimized_Schema_And_Data values (@customerID, 'Test', '2013-01-01T00:00:00')
    set @customerID +=1
end
Print DATEDIFF(ms,@start,getdate());
Go
-----


Print 'insert into tblMemoryOptimized_Schema_Only'

DECLARE @start datetime = getdate()
declare @insertCount int = 100000
declare @startId int = 1
declare @customerID int = @startId

while @customerID < @startId + @insertCount
begin
    insert into tblMemoryOptimized_Schema_Only values (@customerID, 'Test', '2013-01-01T00:00:00')
    set @customerID +=1
end
Print DATEDIFF(ms,@start,getdate());
Go

```

با این خروجی تقریبی که بر اساس توانمندی های سخت افزاری سیستم می تواند متفاوت باشد:

```

insert into tblNormal
36423

insert into tblMemoryOptimized_Schema_And_Data
30516

insert into tblMemoryOptimized_Schema_Only
3176

```

و برای حالت select خواهیم داشت:

```
set nocount on
print 'tblNormal'
set statistics time on
select count(CustomerID) from tblNormal
set statistics time off
go
print 'tblMemoryOptimized_Schema_And_Data'
set statistics time on
select count(CustomerID) from tblMemoryOptimized_Schema_And_Data
set statistics time off
go
print 'tblMemoryOptimized_Schema_Only'
set statistics time on
select count(CustomerID) from tblMemoryOptimized_Schema_Only
set statistics time off
go
```

با این خروجی

```
tblNormal
SQL Server Execution Times:
CPU time = 46 ms, elapsed time = 52 ms.

tblMemoryOptimized_Schema_And_Data
SQL Server Execution Times:
CPU time = 32 ms, elapsed time = 33 ms.

tblMemoryOptimized_Schema_Only
SQL Server Execution Times:
CPU time = 31 ms, elapsed time = 30 ms.
```

تأثیر جداول بهینه سازی شده برای حافظه را در 350K inserts بخوبی میتوان با نمونه های متداول مبتنی بر دیسک مقایسه کرد.

برای مطالعه بیشتر

[Getting started with SQL Server 2014 In-Memory OLTP](#)

[Introduction to SQL Server 2014 CTP1 Memory-Optimized Tables](#)

[Overcoming storage speed limitations with Memory-Optimized Tables for SQL Server](#)

[Memory-optimized Table – Day 1 Test](#)

[Memory-Optimized Tables – Insert Test](#)

[Memory Optimized Table – Insert Test ...Again](#)

نظرات خوانندگان

نویسنده: علی رضایی
تاریخ: ۱۳۹۳/۰۳/۱۲ ۱۹:۲۲

با سلام و تشکر فراوان جهت این آموزش.

میشه لطفاً بررسی کنید چرا نتیجه نهایی برای من متفاوت شده، طوری که زمان جستجو برای جدول نرمال کمتره!
راستی زمان ورود اطلاعات (100 رکورد) شبیه به زمان‌های مثال شما است و من از یک هارد SSD Corsair استفاده میکنم.

```

set nocount on
print 'tblNormal'
set statistics time on
select count(CustomerID) from tblNormal
set statistics time off
go
print 'tblMemoryOptimized_Schema_And_Data'
set statistics time on
select count(CustomerID) from tblMemoryOptimized_Schema_And_Data
set statistics time off
go
print 'tblMemoryOptimized_Schema_Only'
set statistics time on
select count(CustomerID) from tblMemoryOptimized_Schema_Only
set statistics time off
go
```

Table	CPU time	elapsed time
tblNormal	16 ms	6 ms
tblMemoryOptimized_Schema_And_Data	15 ms	18 ms
tblMemoryOptimized_Schema_Only	16 ms	17 ms

ویرایش:

تست جدید با بیش از 11 میلیون رکورد و خاموش کردن فایروال کومودو و خاموش کردن windows defender

```

set nocount on
print 'tblNormal'
set statistics time on
select count(CustomerID) from tblNormal
set statistics time off
go
print 'tblMemoryOptimized_Schema_And_Data'
set statistics time on
select count(CustomerID) from tblMemoryOptimized_Schema_And_Data
set statistics time off
go
print 'tblMemoryOptimized_Schema_Only'
set statistics time on
select count(CustomerID) from tblMemoryOptimized_Schema_Only
set statistics time off
go

```

100 %

Results Messages

	(No column name)
1	11072240

	(No column name)
1	11000000

	(No column name)
1	11000000

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۳/۳۷

- بستگی دارد. چقدر سیستم شما RAM دارد. مشخصات CPU آن چیست و خیلی از مسایل جانبی دیگر (مانند تحت نظر بودن پوشش‌های فایل استریم‌ها توسط آنتی ویروس یا خیر).
- هدف اصلی از این تحولات، کارهای همزمان و بررسی تفاوت بهبود در کارهای چند ریسمانی و چند کاربری است. مثال فوق یک مثال تک ریسمانی است.
- یک آزمایش را باید چندبار تکرار کرد و بعد میانگین گرفت. برای تکرار هم نیاز است کش‌های سیستم را هربار حذف کنید:

```
set nocount on  
CHECKPOINT  
DBCC FREEPROCCACHE()  
DBCC DROPCLEANBUFFERS
```

- بعد از پایان Insert تعداد ردیف‌های زیاد در یک جدول درون حافظه‌ای باید اطلاعات آماری آن را دستی به روز کرد ([^](#)).

```
UPDATE STATISTICS tblNormal WITH FULLSCAN, NORECOMPUTE;  
UPDATE STATISTICS tblMemoryOptimized_Schema_And_Data WITH FULLSCAN, NORECOMPUTE;  
UPDATE STATISTICS tblMemoryOptimized_Schema_Only WITH FULLSCAN, NORECOMPUTE;
```

این دستورات باید قبل از اجرای سه کوئری آخر قرار گیرند.

- تعداد bucket count هم مهم است. در [اینجا](#) فرمولی برای محاسبه آن ارائه شده:

```
Select POWER(  
    2,  
    CEILING( LOG( COUNT( 0)) / LOG( 2)))  
    AS 'BUCKET_COUNT'  
FROM  
    (SELECT DISTINCT CustomerId  
        FROM tblMemoryOptimized_Schema_And_Data) T
```

نویسنده: وحید نصیری
تاریخ: ۰۵۹ ۱۳۹۳/۰۳/۱۴

ارزش واقعی جداول درون حافظه‌ای را باید با اعمال تراکنش‌های همزمان و بررسی میزان پاسخگویی سیستم بررسی کرد و نه صرفا با یک آزمایش ساده تک رسمنانی. برای این منظور برنامه‌ای به نام ostress.exe توسط مایکروسافت تهیه شده‌است که امکان انجام یک چنین آزمایشاتی را میسر می‌کند. برای دریافت آن به آدرس‌های ذیل مراجعه کنید:

[RML Utilities X64](#)

[RML Utilities X86](#)

که نهایتا در این مسیر C:\Program Files\Microsoft Corporation\RMLUtils نصب خواهد شد.

سپس در خط فرمان این سه دستور را امتحان کنید:

```
-- Insert 10000 records using 20 threads, Repeat Execution 3 times  
  
-- disk-based  
"C:\Program Files\Microsoft Corporation\RMLUtils\ostress.exe" -n20 -r3 -S. -E -dTestdb2 -q -Q"set  
statistics time off; SET STATISTICS IO Off; set nocount on; DECLARE @start datetime = getdate();  
declare @insertCount int = 10000; declare @startId int = 1; while @startId < @insertCount begin insert  
into tblNormal values ('Test', '2013-01-01T00:00:00') set @startId +=1 end; Print  
DATEDIFF(ms,@start,getdate());" -oc:\temp\output  
  
-- memory-optimized, tblMemoryOptimized_Schema_And_Data  
"C:\Program Files\Microsoft Corporation\RMLUtils\ostress.exe" -n20 -r3 -S. -E -dTestdb2 -q -Q"set  
statistics time off; SET STATISTICS IO Off; set nocount on; DECLARE @start datetime = getdate();  
declare @insertCount int = 10000; declare @startId int = 1; while @startId < @insertCount begin insert  
into tblMemoryOptimized_Schema_And_Data values ('Test', '2013-01-01T00:00:00') set @startId +=1 end;  
Print DATEDIFF(ms,@start,getdate());" -oc:\temp\output  
  
-- memory-optimized, tblMemoryOptimized_Schema_Only  
"C:\Program Files\Microsoft Corporation\RMLUtils\ostress.exe" -n20 -r3 -S. -E -dTestdb2 -q -Q"set  
statistics time off; SET STATISTICS IO Off; set nocount on; DECLARE @start datetime = getdate();  
declare @insertCount int = 10000; declare @startId int = 1; while @startId < @insertCount begin insert  
into tblMemoryOptimized_Schema_Only values ('Test', '2013-01-01T00:00:00') set @startId +=1 end;  
Print DATEDIFF(ms,@start,getdate());" -oc:\temp\output
```

زمانیکه را که در پایان کار نمایش می‌دهد، مبنای واقعی مقایسه است.

البته برای اجرای این دستورات نیاز است فیلد CustomerID identity را تعریف کنید (در هر سه جدول مطلب جاری).

```
-- It is not Memory Optimized
CREATE TABLE tblNormal
(
    [CustomerID] int identity NOT NULL PRIMARY KEY NONCLUSTERED,
    [Name] nvarchar(250) NOT NULL,
    CustomerSince DATETIME not NULL
    INDEX [ICustomerSince] NONCLUSTERED
)

-- DURABILITY = SCHEMA_AND_DATA
CREATE TABLE tblMemoryOptimized_Schema_And_Data
(
    [CustomerID] INT identity NOT NULL
PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT = 131072),
    [Name] NVARCHAR(250) NOT NULL,
    [CustomerSince] DATETIME NOT NULL
INDEX [ICustomerSince] NONCLUSTERED
) WITH (MEMORY_OPTIMIZED = ON, DURABILITY = SCHEMA_AND_DATA)

-- DURABILITY = SCHEMA_ONLY
CREATE TABLE tblMemoryOptimized_Schema_Only
(
    [CustomerID] INT identity NOT NULL
PRIMARY KEY NONCLUSTERED HASH WITH (BUCKET_COUNT = 131072),
    [Name] NVARCHAR(250) NOT NULL,
    [CustomerSince] DATETIME NOT NULL
INDEX [ICustomerSince] NONCLUSTERED
) WITH (MEMORY_OPTIMIZED = ON, DURABILITY = SCHEMA_ONLY)
```

نویسنده: حمید رضا نظری
تاریخ: ۱۳۹۳/۰۹/۱۰ ۱۰:۴۰

موقع اجرای دستور ALTER DATABASE SQL2014_Demo

ALTER DATABASE SQL2014_Demo ADD FILEGROUP MemFG CONTAINS MEMORY_OPTIMIZED_DATA

GO

این خطأ رو میده لطفا راهنمایم کنید

Msg 534, Level 15, State 72, Line 37

FILEGROUP ... CONTAINS MEMORY_OPTIMIZED_DATA' failed because it is not supported in the edition of this SQL Server instance 'TFS-SERVER'. See books online for more details on feature support in different SQL Server editions

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۱۰ ۱۰:۵۰

- پیام «[it is not supported](#)» به معنای عدم قابلیت در نگارش SQL Server ای است که از آن استفاده می‌کنید.
- فقط در نگارش‌های SQL Server 2014 Enterprise , Developer , and Evaluation editions پشتیبانی می‌شود. برای مثال نگارش [Standard](#) این قابلیت را ندارد.
- هنگام نصب هم باید گزینه‌ی «Database Engine Services -> install support for In-Memory OLTP engine» انتخاب شده باشد.

نویسنده: حمید رضا نظری
تاریخ: ۱۳۹۳/۰۹/۱۰ ۱۳:۲۴

من نگارش 64 را نصب کردم ولی باز خطأ میده

نوبنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۹/۱۰ ۱۳:۵۵

مباحث جداول بهینه سازی شده برای حافظه، روی این سیستم تهیه شدند:

```
select @@VERSION
```

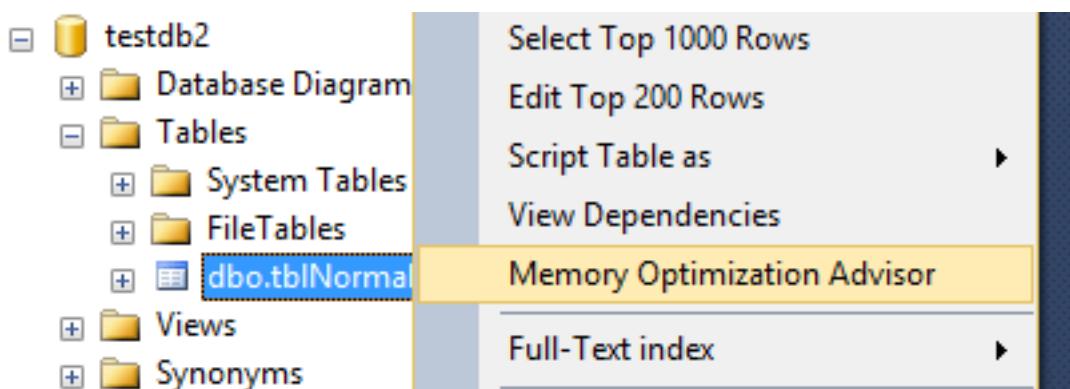
با این خروجی (که مشخصات instance 'TFS-SERVER' جاری را بر می‌گرداند؛ در مثال شما باید بررسی شود که چه نگارشی دارد)

```
Microsoft SQL Server 2014 - 12.0.2254.0 (X64)
Jul 25 2014 18:52:51
Copyright (c) Microsoft Corporation
Developer Edition (64-bit) on Windows NT 6.3 <X64> (Build 9600: )
```

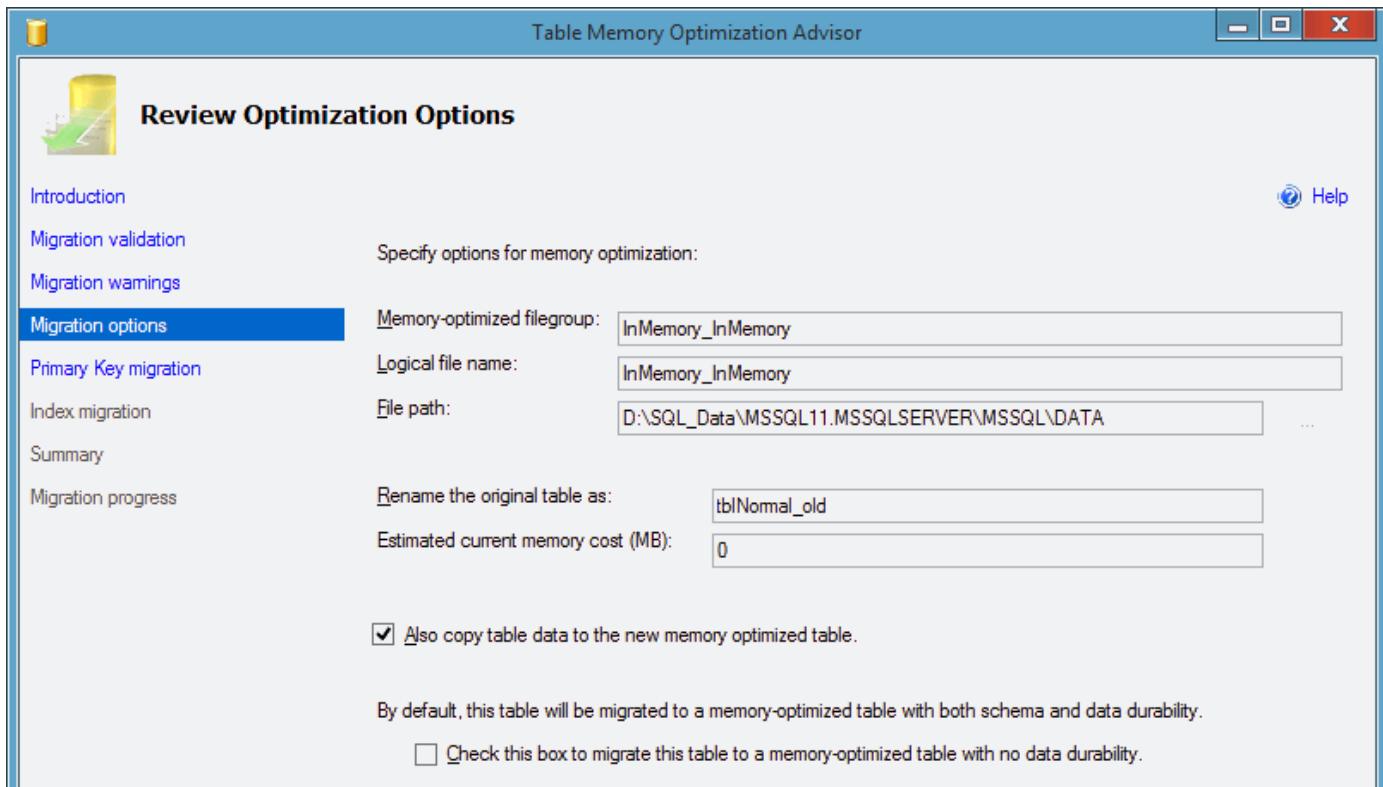
در 2014 SQL Server، به آن ابزارهای جدیدی اضافه شده‌اند تا کار تبدیل و مهاجرت جداول معمولی، به جداول بهینه سازی شده‌ی برای حافظه را ساده‌تر کنند. برای مثال امکان جدیدی به نام Transaction performance collector به نام چهت بررسی کارآیی تراکنش‌های جداول و یا رویه‌های ذخیره شده در محیط کاری جاری، طراحی شده‌است. پس از آن، این اطلاعات را آنالیز کرده و بر اساس میزان استفاده از آن‌ها، توصیه‌هایی را در مورد مهاجرت یا عدم نیاز به مهاجرت به سیستم جدید OLTP درون حافظه‌ای ارائه می‌دهد. در ادامه این ابزارهای جدید را بررسی خواهیم کرد.

ابزار Memory Optimization Advisor

یک Wizard Memory Optimization Advisor مانند است که از آن برای گرفتن مشاوره در مورد تبدیل جداول موجود مبتنی بر دیسک ساخت، به نمونه‌های بهینه سازی شده برای حافظه می‌توان استفاده کرد. کار آن بررسی ساختار جداولی است که قصد مهاجرت آن‌ها را دارد. برای مثال همانطور که [بیشتر نیز عنوان شد](#)، جداول بهینه سازی شده برای حافظه محدودیت‌هایی دارند؛ مثلاً نباید کلید خارجی داشته باشند. این Wizard یک چنین مواردی را آنالیز کرده و گزارشی را ارائه می‌دهد. پس از اینکه مراحل آن را به پایان رساندید و مشکلاتی را که گزارش می‌دهد، برطرف نمودید، کد تبدیل جدول را نیز به صورت خودکار تولید می‌کند. برای دسترسی به آن، فقط کافی است بر روی نام جدول خود کلیک راست کرده و گزینه‌ی memory optimization advisor را انتخاب کنید.



در دو قسمت اول این Wizard، کار بررسی ساختار جدول در حال مهاجرت صورت می‌گیرد. اگر نوع داده‌ای در آن پشتیبانی نشود یا قیود ویژه‌ای در آن تعریف شده باشند، گزارشی را جهت رفع، دریافت خواهید کرد. پس از رفع آن، به صفحه‌ی گزینه‌های مهاجرت می‌رسیم:



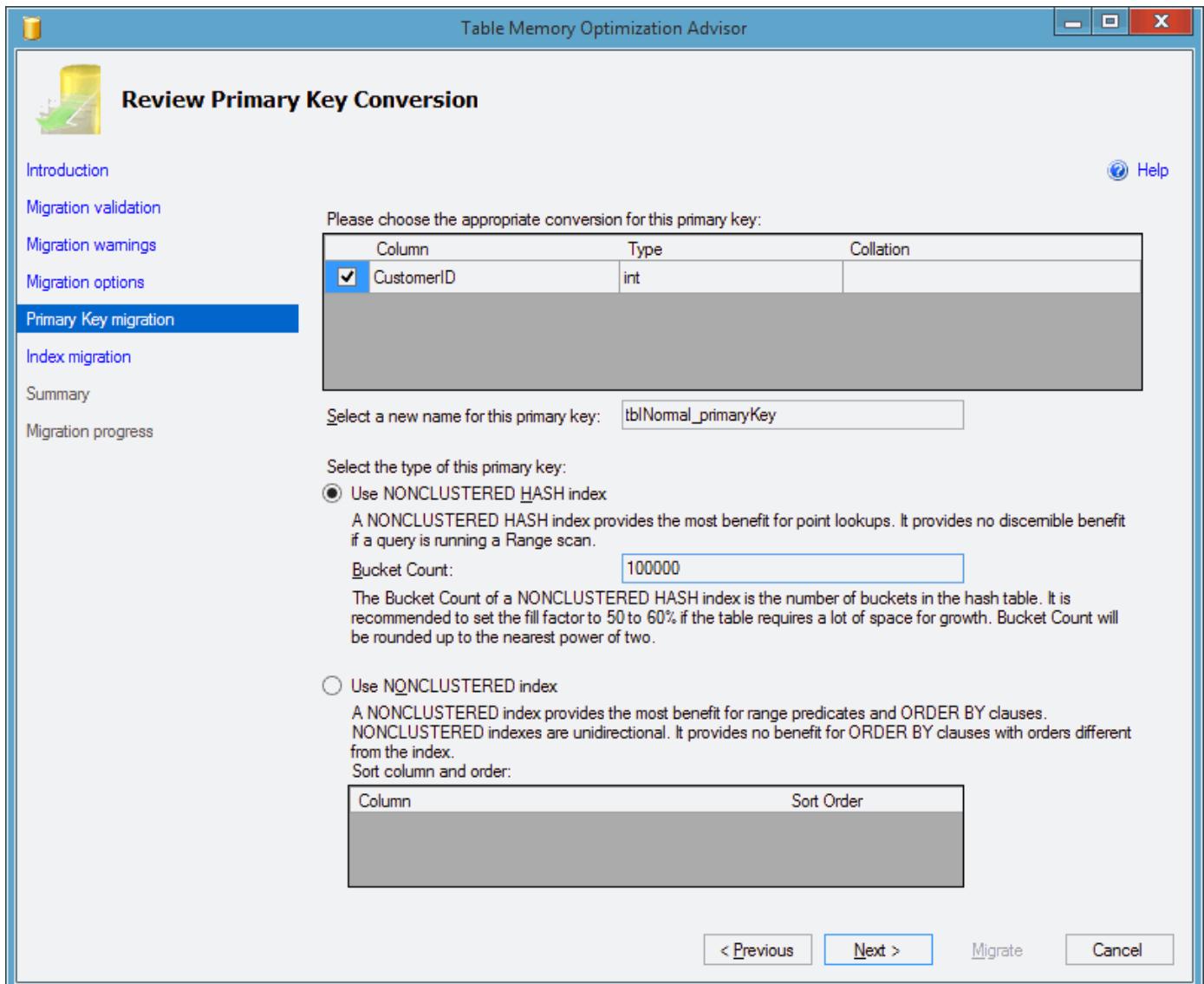
همانطور که ملاحظه می‌کنید، گروه فایل ایجاد شده در قسمت قبل، به صورت خودکار انتخاب شده است.

در ادامه می‌توان نام دیگری را برای جدول مبتنی بر دیسک وارد کرد. در اینجا به صورت خودکار کلمه‌ی old به آخر نام جدول اضافه شده است. در حین تولید جدول جدید بهینه سازی شده‌ی بر اساس ساختار جدول فعلی، این جدول قدیمی به صورت خودکار تغییر نام خواهد یافت و کلیه اطلاعات آن حفظ می‌شود.

همچنین تخمینی را نیز از مقدار حافظه‌ی مورد نیاز برای نگهداری این جدول جدید درون حافظه‌ای نیز ارائه می‌دهد. در این مثال چون رکوردهای در جدول انتخابی وجود نداشته است، تخمین آن صفر است. عدد ارائه شده توسط آن بسیار مهم است و باید به همین میزان برای سیستم خود حافظه تهیه نمائید و یا از حافظه‌ی موجود استفاده کنید.

در پایین صفحه می‌توان انتخاب کرد که آیا داده‌های جدول فعلی، به جدول درون حافظه‌ای انتقال یابند یا خیر. به علاوه نوع ماندگاری اطلاعات آن نیز قابل تنظیم است. اگر گزینه‌ی آخر را انتخاب کنید به معنای حالت SCHEMA_ONLY است. حالت پیش فرض آن SCHEMA_AND_DATA می‌باشد که در قسمت‌های قبل بیشتر در مورد آن بحث شد.

در دو صفحه‌ی بعد، کار انتخاب range index و hash index انجام می‌شود:



در اینجا hash index بر روی فیلد ID تولید شده است، به همراه تعیین bucket count آن و در صفحه‌ی بعدی range index بر روی فیلد تاریخ تعریف گردیده است:

The screenshot shows the 'Table Memory Optimization Advisor' application window. The title bar reads 'Table Memory Optimization Advisor'. The main area is titled 'Review Index Conversion' with a small icon of a yellow cylinder with a green checkmark.

The left sidebar contains navigation links: Introduction, Migration validation, Migration warnings, Migration options, Primary Key migration, **Index migration** (which is selected and highlighted in blue), and Summary. Below these are links for Migration progress and Help.

The central panel has a heading 'Please choose the appropriate conversion for this index:' followed by a table:

Column	Type	Collation
<input checked="" type="checkbox"/> CustomerSince	datetime	

Below this, a section titled 'Select the type of this index:' contains two radio button options:

- Use NONCLUSTERED HASH index

A NONCLUSTERED HASH index provides the most benefit for point lookups. It provides no discernible benefit if a query is running a Range scan.

Bucket Count:
The Bucket Count of a NONCLUSTERED HASH index is the number of buckets in the hash table. It is recommended to set the fill factor to 50 to 60% if the table requires a lot of space for growth. Bucket Count will be rounded up to the nearest power of two.
- Use NONCLUSTERED index

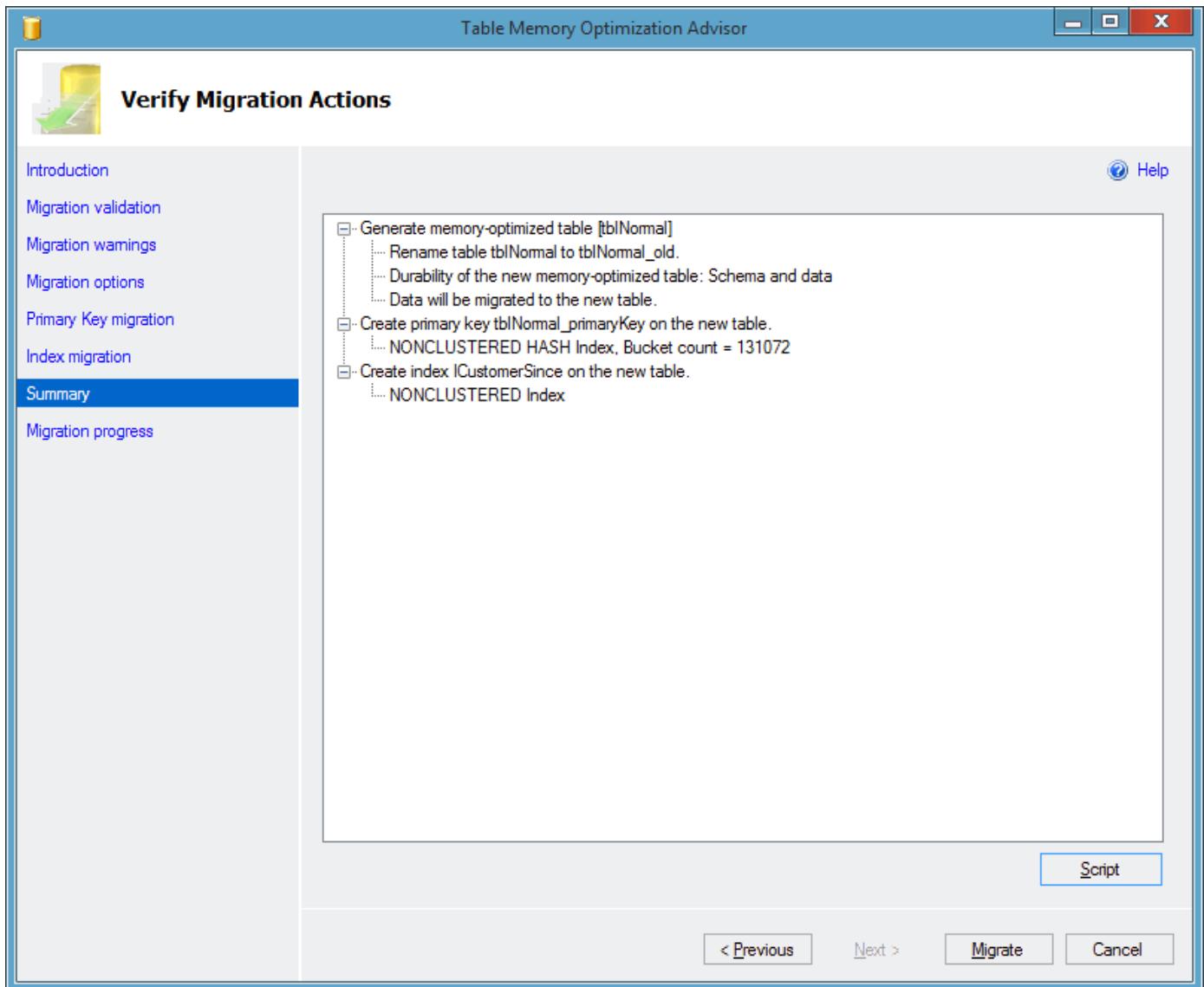
A NONCLUSTERED index provides the most benefit for range predicates and ORDER BY clauses. NONCLUSTERED indexes are unidirectional. It provides no benefit for ORDER BY clauses with orders different from the index.

Sort column and order:

Column	Sort Order
CustomerSince	ASC

At the bottom right are buttons for < Previous, Next >, Migrate, and Cancel.

در آخر می‌توان با کلیک بر روی دکمه Script، صرفا دستورات T-SQL تغییر ساختار جدول را دریافت کرد و یا با کلیک بر روی دکمه migrate به صورت خودکار کلیه موارد تنظیم شده را اجرا نمود.



خلاصه‌ی این مراحل که توسط دکمه‌ی Script آن تولید می‌شود، به صورت زیر است:

```

USE [testdb2]
GO

EXEC dbo.sp_rename @objname = N'[dbo].[tblNormal]', @newname = N'tblNormal_old', @objtype = N'OBJECT'
GO

USE [testdb2]
GO

SET ANSI_NULLS ON
GO

CREATE TABLE [dbo].[tblNormal]
(
[CustomerID] [int] NOT NULL,
[Name] [nvarchar](250) COLLATE Persian_100_CI_AI NOT NULL,
[CustomerSince] [datetime] NOT NULL,
INDEX [ICustomerSince] NONCLUSTERED
(
[CustomerSince] ASC
),
CONSTRAINT [tblNormal_primaryKey] PRIMARY KEY NONCLUSTERED HASH
(
[CustomerID]
)WITH ( BUCKET_COUNT = 131072)

```

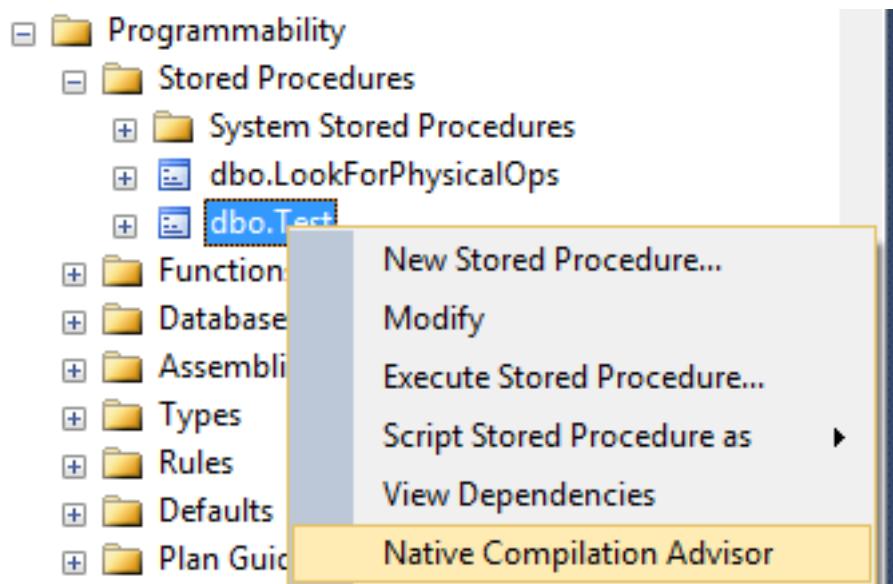
```

)WITH ( MEMORY_OPTIMIZED = ON , DURABILITY = SCHEMA_AND_DATA )
GO
INSERT INTO [testdb2].[dbo].[tblNormal] ([CustomerID], [Name], [CustomerSince]) SELECT [CustomerID],
[Name], [CustomerSince] FROM [testdb2].[dbo].[tblNormal_old]
GO

```

که در آن ابتدا کار تغییر نام جدول قبلی صورت می‌گیرد. سپس یک جدول جدید با ویژگی `MEMORY_OPTIMIZED = ON` را ایجاد می‌کند. در ساختار این جدول، `hash index` و `range index` تعریف شده، قابل مشاهده هستند. در آخر نیز کلیه اطلاعات جدول قدیمی را به جدول جدید منتقل می‌کند.

علاوه بر `memory optimization advisor` مخصوص جداول، ابزار دیگری نیز به نام `Native compilation advisor` برای آنالیز رویه‌های ذخیره شده تهیه شده است:

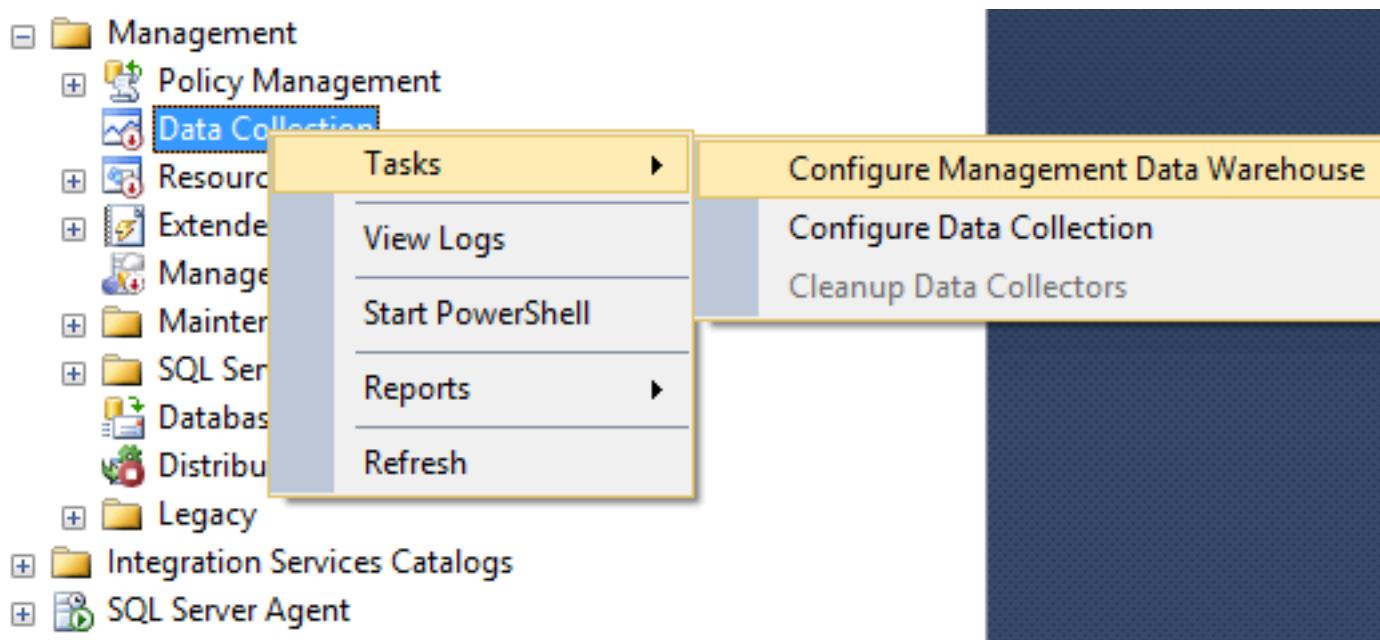


آیا سیستم فعلی ما واقعاً نیازی به ارتقاء به جداول درون حافظه‌ای دارد؟

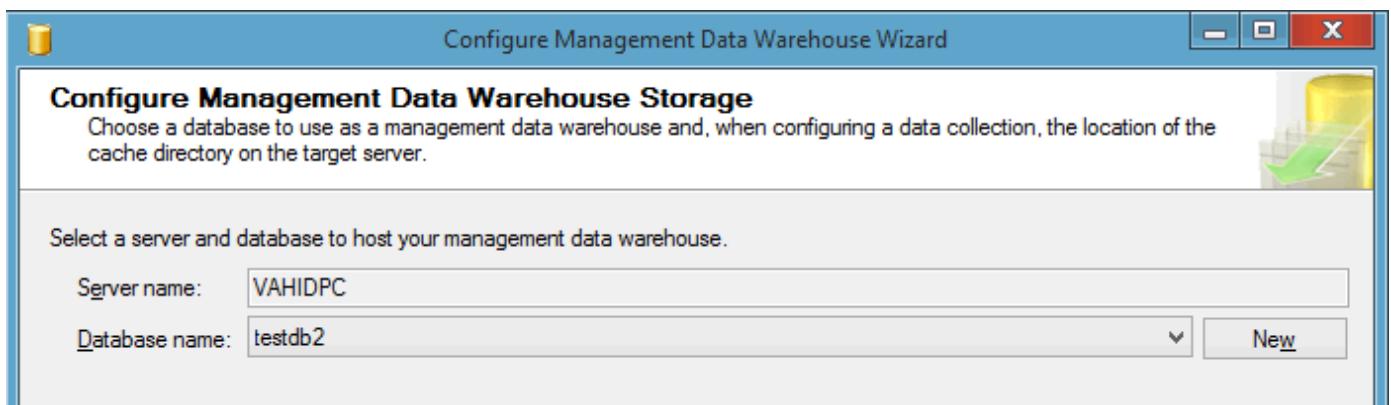
تا اینجا در مورد نحوه ایجاد جداول موجود را به ساختار جدید بررسی کردیم. ولی آیا واقعاً یک چنین تغییراتی برای ما سودمند هستند؟ برای پاسخ دادن به این سؤال ابزاری به نام AMR به 2014 management studio اضافه شده است (Analyze, Migrate, Report). کار آن تحت نظر قرار دادن جداول و رویه‌های ذخیره شدهی بانک اطلاعاتی است و سپس بر اساس بار سیستم، تعداد درخواست‌های همزمان و میزان استفاده از جداول و تراکنش‌های مرتبط با آن‌ها، گزارشی را ارائه می‌دهد. بر این اساس بهتر می‌توان تصمیم گرفت که کدام جداول بهتر است به جداول درون حافظه‌ای تبدیل شوند.

برای تنظیم آن باید مراحل ذیل طی شوند:

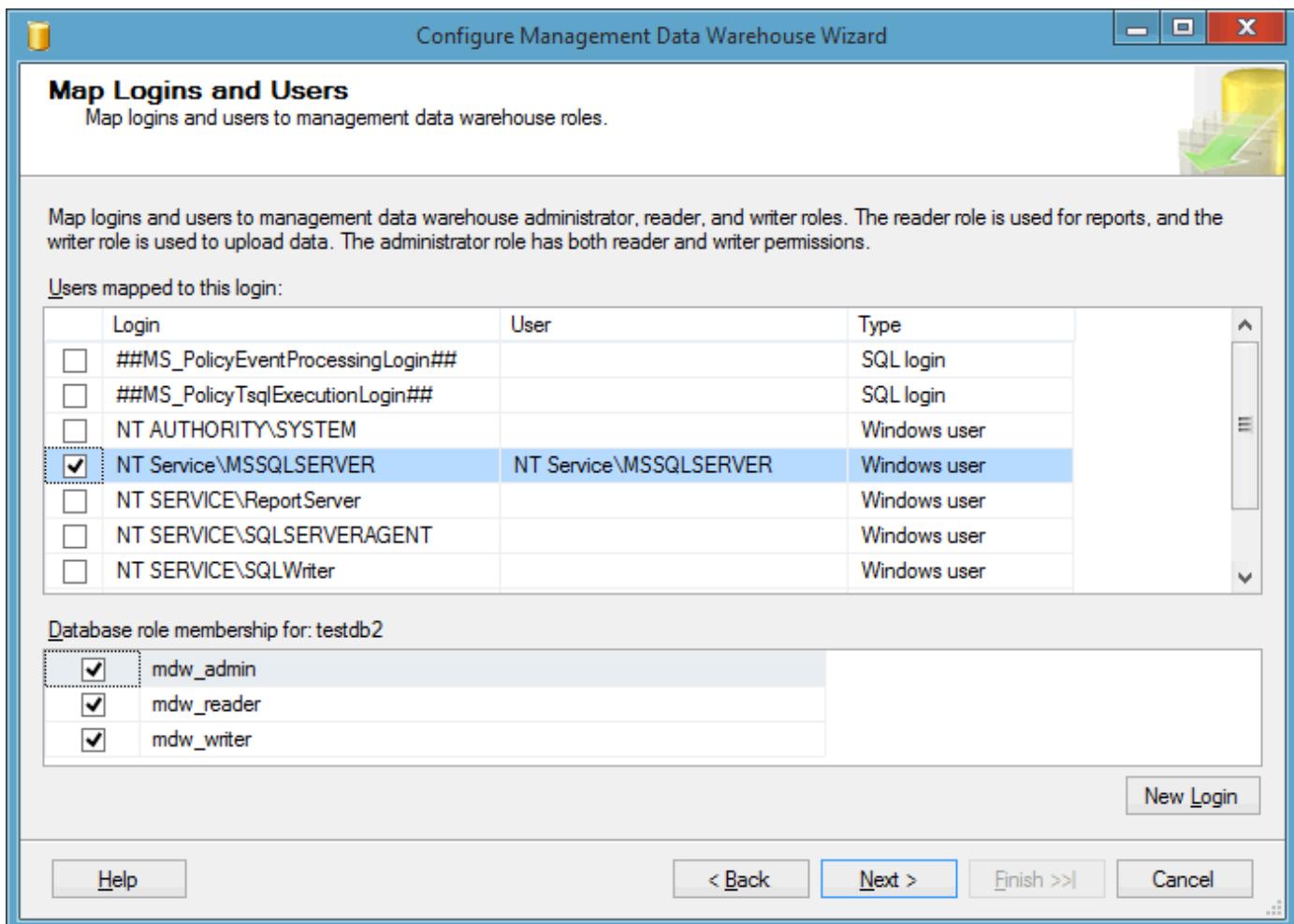
در `Management Studio`, به برگه‌ی `Object Explorer` آن مراجعه کنید. سپس پوشه‌ی `Management` آن را یافته و بر روی گزینه‌ی `Data Collection` کلیک راست نمایید:



در اینجا گزینه **Configure Management Data Warehouse** را انتخاب نمایید. در صفحه‌ی باز شده، ابتدا بانک اطلاعاتی مدنظر را انتخاب نمایید. همچنین بهتر است بر روی دکمه new کلیک کرده و یک بانک اطلاعاتی جدید را برای آن ایجاد نمایید، تا دچار تداخل اطلاعاتی و ساختاری نگردد:

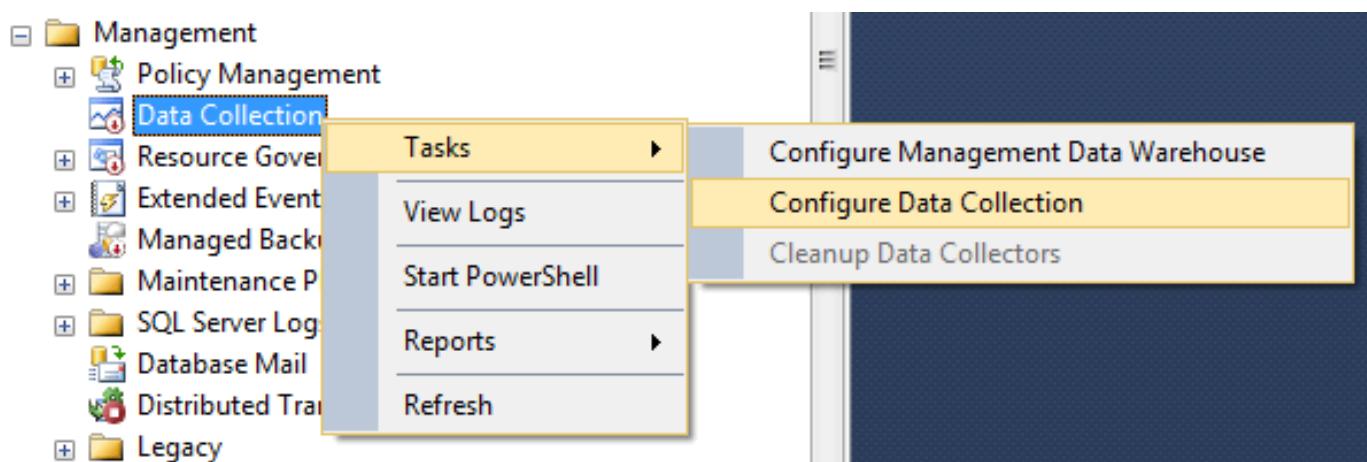


در ادامه نام کاربری را که قرار است کار مدیریت ثبت و جمع آوری اطلاعات را انجام دهد، به همراه نقش‌های آن انتخاب نمایید:



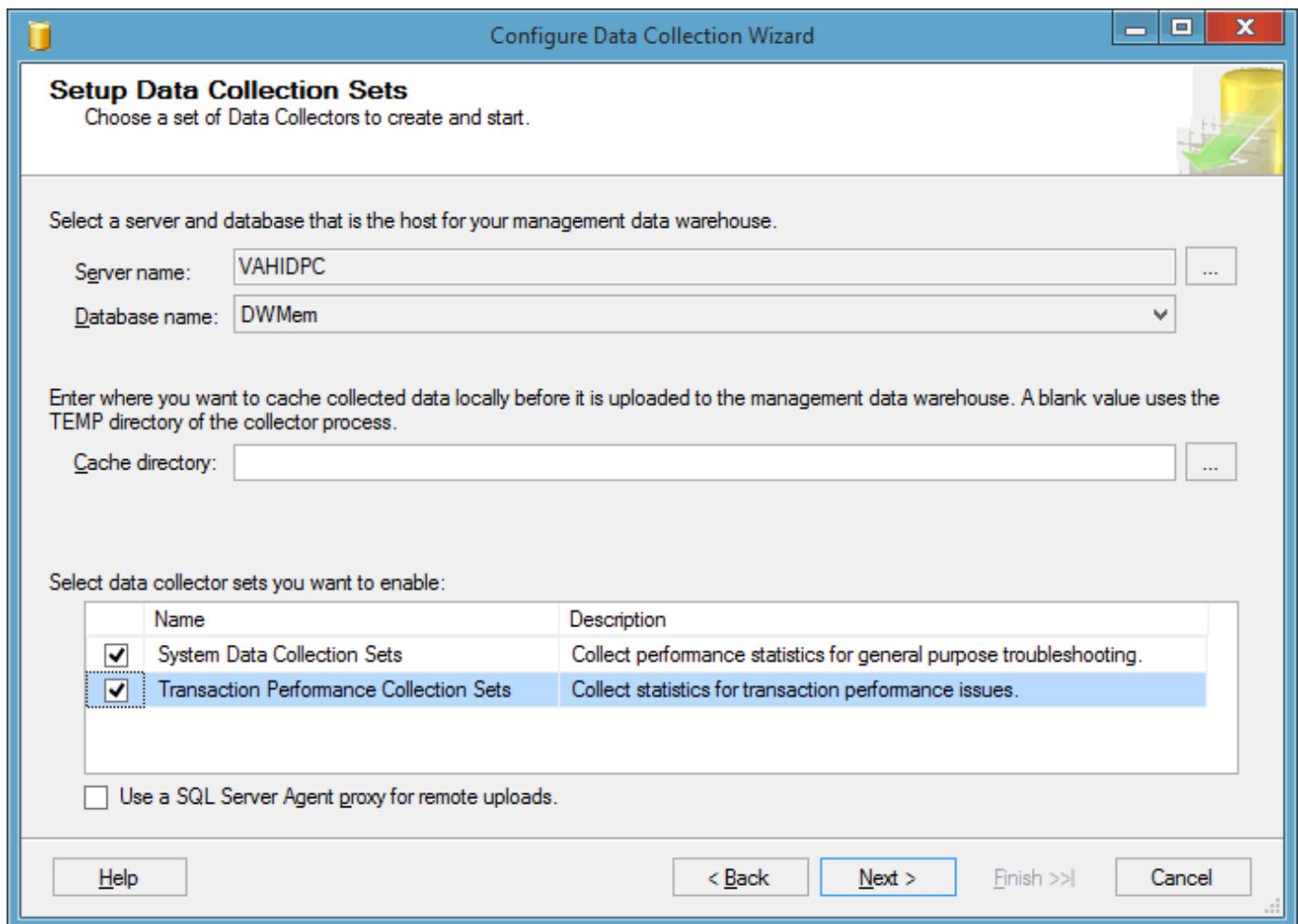
و در آخر در صفحه‌ی بعدی بر روی دکمه‌ی **Finish** کلیک کنید.

پس از ایجاد و انتخاب بانک اطلاعاتی **Management Data Warehouse**, نوبت به تنظیم گزینه‌های جمع آوری اطلاعات است:



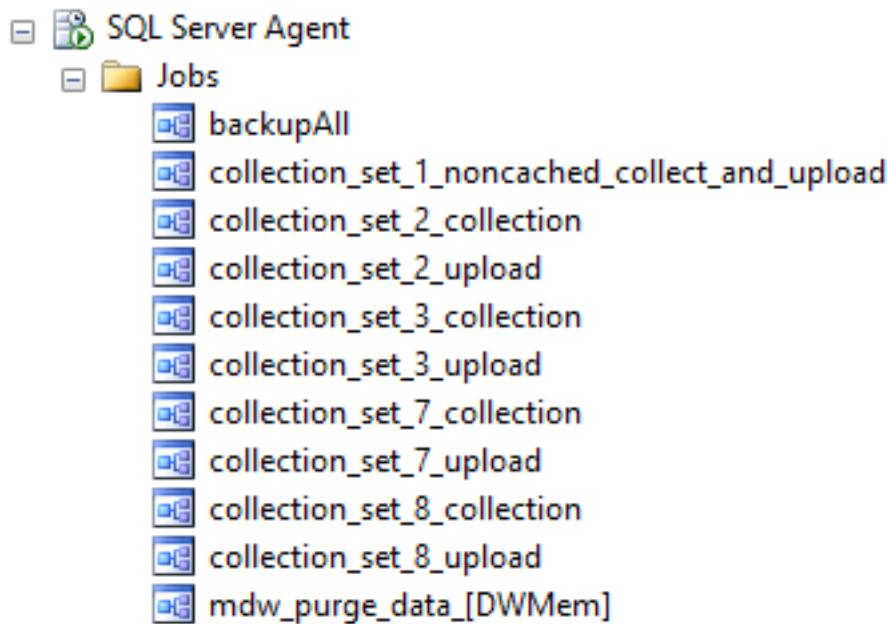
در اینجا ابتدا سرور جاری را انتخاب کنید. پس از آن به صورت خودکار در لیست بانک‌های اطلاعاتی قابل انتخاب، تنها همان بانک

اطلاعاتی جدیدی را که برای مرحله‌ی قبل ایجاد کردیم، می‌توان مشاهده کرد.



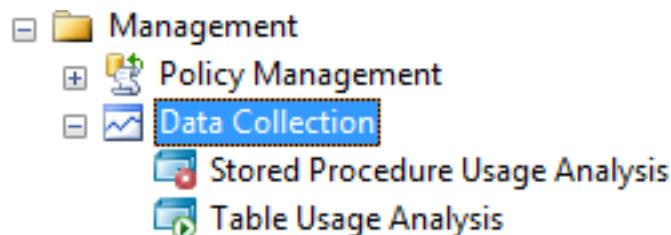
در صفحه‌ی بعد، گزینه‌ی «Transaction Performance Collection Sets» را انتخاب نمائید که دقیقاً گزینه‌ی مدنظر ما جهت یافتن آماری از وضعیت تراکنش‌های سیستم است.
در ادامه بر روی گزینه‌های next و finish کلیک کنید تا کار تنظیمات به پایان برسد.

اکنون اگر به لیست وظایف تعریف شده در SQL Server agent مراجعه کنید، می‌توانید، وظایف مرتبط با جمع آوری داده‌ها را نیز مشاهده نمائید:

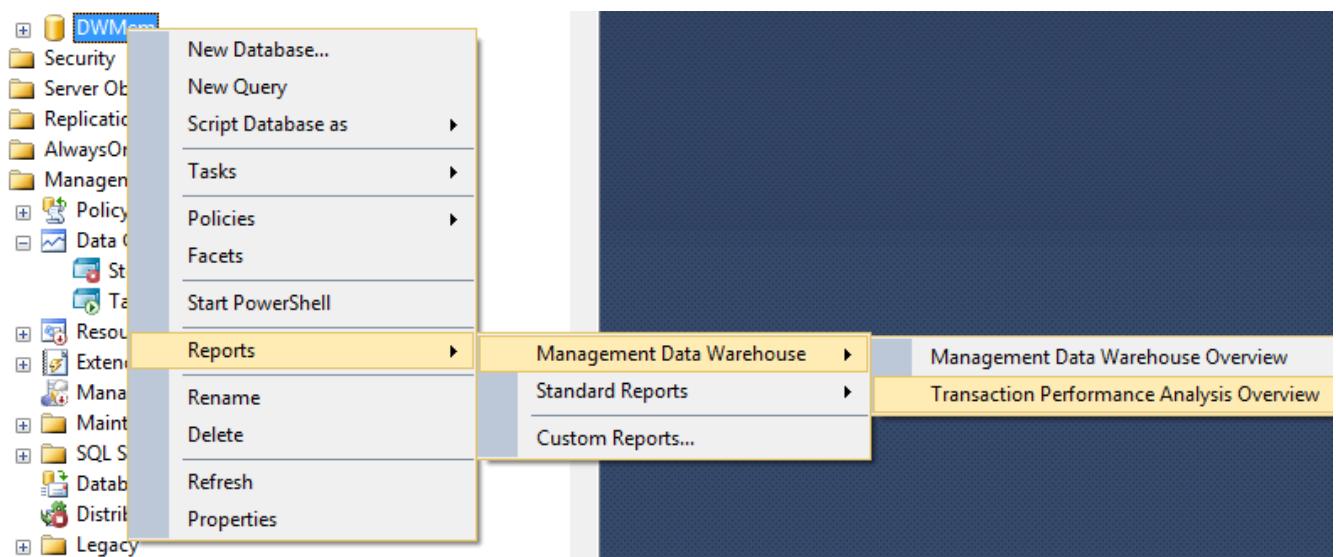


وظایف Stored Procedure Usage Analysis هر نیم ساعت یکبار و وظایف Table Usage Analysis هر 15 دقیقه یکبار اجرا می‌شوند. البته امکان اجرای دستی این وظایف نیز مانند سایر وظایف SQL Server وجود دارد.

همچنین در پوشه‌ی management، گزینه‌ی Data collection نیز دو زیر شاخه اضافه شده‌اند که نمایانگر آنالیز میزان مصرف جداول و رویه‌های ذخیره شده می‌باشند:



پس از این کارها باید مدتی صبر کنید (مثلاً یک ساعت) تا سیستم به صورت معمول کارهای متداول خودش را انجام دهد. پس از آن می‌توان به گزارشات AMR مراجعه کرد.



برای اینکار بر روی بانک اطلاعاتی Management Data Warehouse که در ابتدای عملیات ایجاد شد، کلیک راست نمایید و سپس مراحل ذیل را طی کنید:

Reports > Management Data Warehouse > Transaction Performance Analysis Overview

Transaction Performance Analysis Overview

on VAHIDPC at 02/06/2014 02:45:27 ب.

Microsoft SQL Server 2014

Welcome to the AMR tool for in-memory OLTP.

This report helps you identify bottlenecks in your database and provide assistance to migrate them to in-memory OLTP. To begin, click on the last snapshot upload time hyperlink of one of these options to see the report.



Tables Analysis

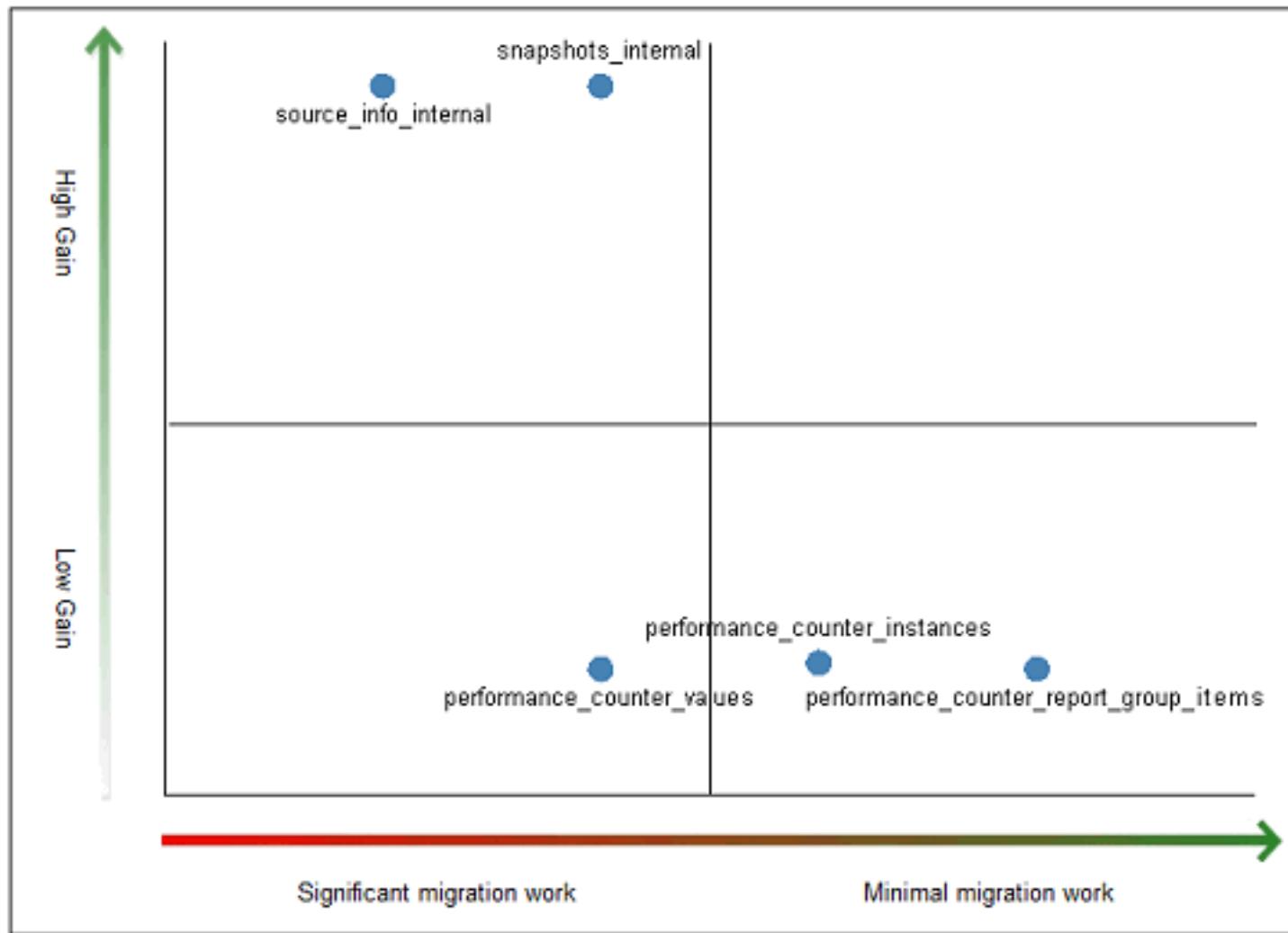


Stored Procedure Analysis

Instance Name	Usage Analysis	Contention Analysis	Usage Analysis
VAHIDPC	02/06/2014 02:45:04 ب.	02/06/2014 02:45:04 ب.	(No Data)

در گزارش ایجاد شده، ذیل گزینه‌ی usage analysis لینک‌های وجود دارند که با مراجعه به آن‌ها، چارت‌هایی از میزان مصرف بانک‌های اطلاعاتی مختلف سیستم ارائه می‌شود. اگر پیام No data available را مشاهده کردید، یعنی هنوز باید مقداری صبر کنید تا کار جمع آوری اطلاعات به پایان برسد.

در این چارت‌ها بانک‌های اطلاعاتی که در سمت راست، بالای تصویر قرار می‌گیرند، انتخاب مناسبی برای تبدیل به بانک‌های اطلاعاتی درون حافظه‌ای هستند. محور افقی آن از چپ به راست بیانگر میزان کاهش سختی انتقال یک جدول به جدول درون حافظه‌ای است (با درنظر گرفتن تمام مسایلی که باید تغییر کنند یا نوع‌های داده‌ای که باید اصلاح شوند) و محور عمودی آن نمایانگر میزان بالا رفتن پاسخ دهی سیستم در جهت انجام کار بیشتر است.



هر زمان هم که کار تصمیم‌گیری شما به پایان رسید، می‌توانید بر روی گزینه‌ی Data collection کلیک راست کرده و آن را غیرفعال نمائید.

برای مطالعه بیشتر

- [SQL Server 2014 Field Benchmarking In-Memory OLTP and Buffer Pool Extension Features](#)
- [New AMR Tool: Simplifying the Migration to In-Memory OLTP](#)
- [A Tour of the Hekaton AMR Tool](#)
- [SQL Server 2014 Memory Optimization Advisor](#)
- [Getting started with the AMR tool for migration to SQL Server In-memory OLTP Tables](#)
- [How to Use Microsoft's AMR Tool](#)
- [SQL Server 2014's Analysis, Migrate, and Report Tool](#)

به صورت پیش فرض SQL Server از روش write-ahead log - WAL استفاده می‌کند. به این معنا که کلیه تغییرات، پیش از commit بهایی باید در لگ فایل آن نوشته شوند. این مساله با تعداد بالای تراکنش‌ها تا حدودی بر روی سرعت سیستم می‌تواند تاثیرگذار باشد. برای بهبود این وضعیت، در SQL Server 2014 قابلیتی به نام delayed_durability اضافه شده است که با فعال سازی آن، کلیه اعمال مرتبه با لگ‌های تراکنش‌ها به صورت غیرهمزمان انجام می‌شوند. به این ترتیب تراکنش‌ها زودتر از معمول به پایان خواهد رسید؛ با این فرض که نوشته شدن تغییرات در لگ فایل‌ها، در آینده‌ای محتمل انجام خواهند شد. این مساله به معنای فدا کردن D در ACID است ([Atomicity, Consistency, Isolation, Durability](#)). البته باید دقت داشت که رسیدن به هزینه‌بر است و شاید خیلی از اوقات تمام اجزای آن نیازی نباشند یا حتی بتوان با اندکی تخفیف آن‌ها را اعمال کرد؛ مانند D به تاخیر افتاده.

برای اینکار SQL Server از یک بافر 60 کیلوبایتی برای ذخیره سازی اطلاعات لگ‌هایی که قرار است به صورت غیرهمزمان با تراکنش‌ها نوشته شوند، استفاده می‌کند. هر زمان که این 60KB پر شد، آن را flush کرده و ثبت خواهد نمود. به این ترتیب به دو مزیت خواهیم رسید:

- پردازش تراکنش‌ها بدون منتظر شدن جهت commit نهایی در دیسک سخت ادامه خواهند یافت. صبر کمتر به معنای امکان پردازش تراکنش‌های بیشتری در یک سیستم پر ترافیک است.
- با توجه به بافری که از آن صحبت شد، اینبار اعمال Write به صورت یک سری batch اعمال می‌شوند که کارآیی و سرعت بیشتری نسبت به حالت تکی دارند.

اندکی تاریخچه

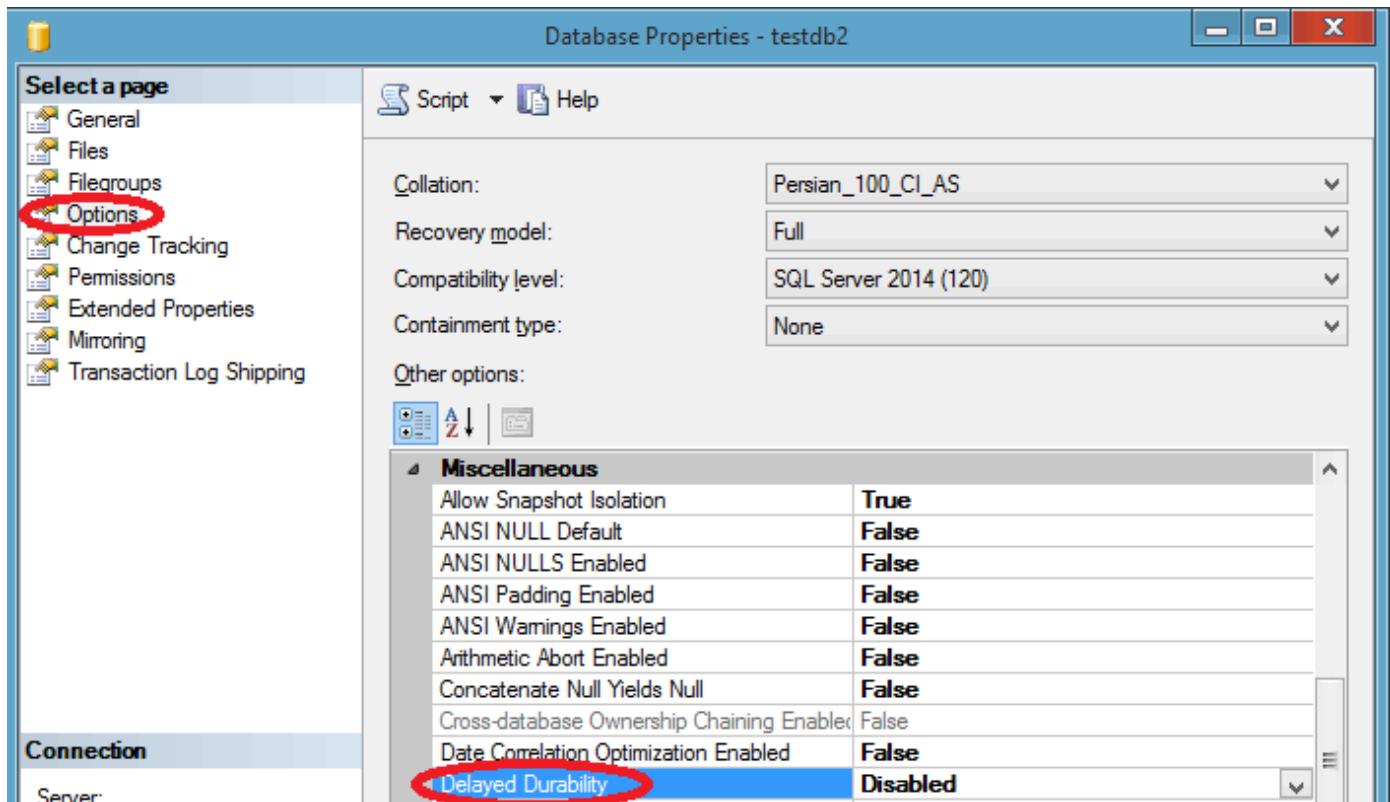
ایده یک چنین عملی 28 سال قبل توسط [Hal Berenson](#) ارائه شده است؛ اوراکل آن را در سال 2006 تحت عنوان Asynchronous Commit پیاده سازی کرد و مایکروسافت در سال 2014 آن را ارائه داده است.

فعال سازی ماندگاری غیرهمزمان در SQL Server

فعال سازی این قابلیت در سطح بانک اطلاعاتی، در سطح یک تراکنش مشخص و یا در سطح رویه‌های ذخیره شده کامپایل شده مخصوص OLTP درون حافظه‌ای، میسر است.

برای فعال سازی ماندگاری با تاخیر در سطح یک دیتابیس، خواهیم داشت:

```
ALTER DATABASE dbname SET DELAYED_DURABILITY = DISABLED | ALLOWED | FORCED;
```



در اینجا اگر ALLOWED را انتخاب کنید، به این معنا است که لاغ کلیه تراکنش‌های مرتبط با این بانک اطلاعاتی به صورت غیرهمزمان نوشته می‌شوند. حالت FORCED نیز دقیقاً به همین معنا است با این تفاوت که اگر حالت ALLOWED انتخاب شود، تراکنش‌های ماندگار (آن‌هایی که به صورت دستی DELAYED_DURABILITY را غیرفعال کرده‌اند)، سبب flush کلیه تراکنش‌هایی با ماندگاری به تاخیر افتاده خواهند شد و سپس اجرا می‌شوند. در حالت Forced تنظیم دسترسی OFF در سطح تراکنش‌ها تاثیری نخواهد داشت؛ اما در حالت ALLOWED این مساله به صورت دستی در سطح یک تراکنش قابل لغو است. البته باید توجه داشت، صرفنظر از این تنظیمات، یک سری از تراکنش‌ها همیشه ماندگار هستند و بدون تاخیر؛ مانند تراکنش‌های سیستمی، تراکنش‌های بین دو یا چند بانک اطلاعاتی و کلیه تراکنش‌هایی که با Change FileTable، Change Data Capture و Tracking سر و کار دارند.

در سطح تراکنش‌های می‌توان نوشت:

```
COMMIT TRANSACTION WITH (DELAYED_DURABILITY = ON);
```

و یا در رویه‌های ذخیره شده کامپایل شده مخصوص OLTP درون حافظه‌ای خواهیم داشت:

```
BEGIN ATOMIC WITH (DELAYED_DURABILITY = ON, ...)
```

سؤال: آیا فعال سازی DELAYED_DURABILITY بر روی مباحث isolation levels و locking تاثیر دارد؟
پاسخ: خیر. کلیه تنظیمات قفل گذاری‌ها همانند commit و بر اساس isolation levels تعیین شده، رخ خواهند داد. تنها تفاوت در اینجا است که با فعال سازی DELAYED_DURABILITY، کار commit بدون صبر کردن برای پایان نوشته شدن اطلاعات در لاغ سیستم صورت می‌گیرد. به این ترتیب قفل‌های انجام شده زودتر آزاد خواهند شد.

سؤال: میزان از دست دادن اطلاعات احتمالی در این روش چقدر است؟

در صورتیکه سرور کرش کند یا ری‌استارت شود، حداقل به اندازه‌ی 60KB اطلاعات را از دست خواهید داد (اندازه‌ی بافری که برای اینکار درنظر گرفته شده‌است). البته عنوان شده‌است که اگر ری‌استارت یا خاموشی سرور، از پیش تعیین شده باشد، ابتدا

کلیه لاغهای flush نشده، ذخیره شده و سپس ادامهی کار صورت خواهد گرفت؛ ولی زیاد به آن اطمینان نکنید. اما همواره با فراخوانی sys.sp_flush_log می‌توان به صورت دستی بافر لاغهای سیستم را flush کرد.

یک آزمایش

در ادامه قصد داریم یک جدول جدید را در بانک اطلاعاتی آزمایشی testdb2 ایجاد کنیم. سپس یکبار تنظیم DELAYED_DURABILITY = FORCED را انجام داده و 10 هزار رکورد را ثبت می‌کنیم و بار دیگر DELAYED_DURABILITY = DISABLED را تنظیم کرده و همین عملیات را تکرار خواهیم کرد:

```

CREATE TABLE tblData(
    ID INT IDENTITY(1, 1),
    Data1 VARCHAR(50),
    Data2 INT
);
CREATE CLUSTERED INDEX PK_tblData ON tblData(ID);
CREATE NONCLUSTERED INDEX IX_tblData_Data2 ON tblData(Data2);

-----
alter database testdb2 SET DELAYED_DURABILITY = FORCED;

-----
SET NOCOUNT ON
Print 'DELAYED_DURABILITY = FORCED'
DECLARE @counter AS INT = 0
DECLARE @start datetime = getdate()
WHILE (@counter < 10000)
BEGIN
    INSERT INTO tblData (Data1, Data2) VALUES('My Data', @counter)
    SET @counter += 1
END
Print DATEDIFF(ms,@start,getdate());
GO

-----
alter database testdb2 SET DELAYED_DURABILITY = DISABLED;
truncate table tblData;

-----
SET NOCOUNT ON
Print 'DELAYED_DURABILITY = DISABLED'
DECLARE @counter AS INT = 0
DECLARE @start datetime = getdate()
WHILE (@counter < 10000)
BEGIN
    INSERT INTO tblData (Data1, Data2) VALUES('My Data', @counter)
    SET @counter += 1
END
Print DATEDIFF(ms,@start,getdate());
GO
-----
```

با این خروجی:

```

DELAYED_DURABILITY = FORCED
666
DELAYED_DURABILITY = DISABLED
2883
```

در این آزمایش، سرعت insert‌ها در حالت DELAYED_DURABILITY = FORCED حدود 4 برابر است نسبت به حالت معمولی.

برای مطالعه بیشتر

[Control Transaction Durability](#)

[SQL Server 2014 Delayed Durability/Lazy Commit](#)

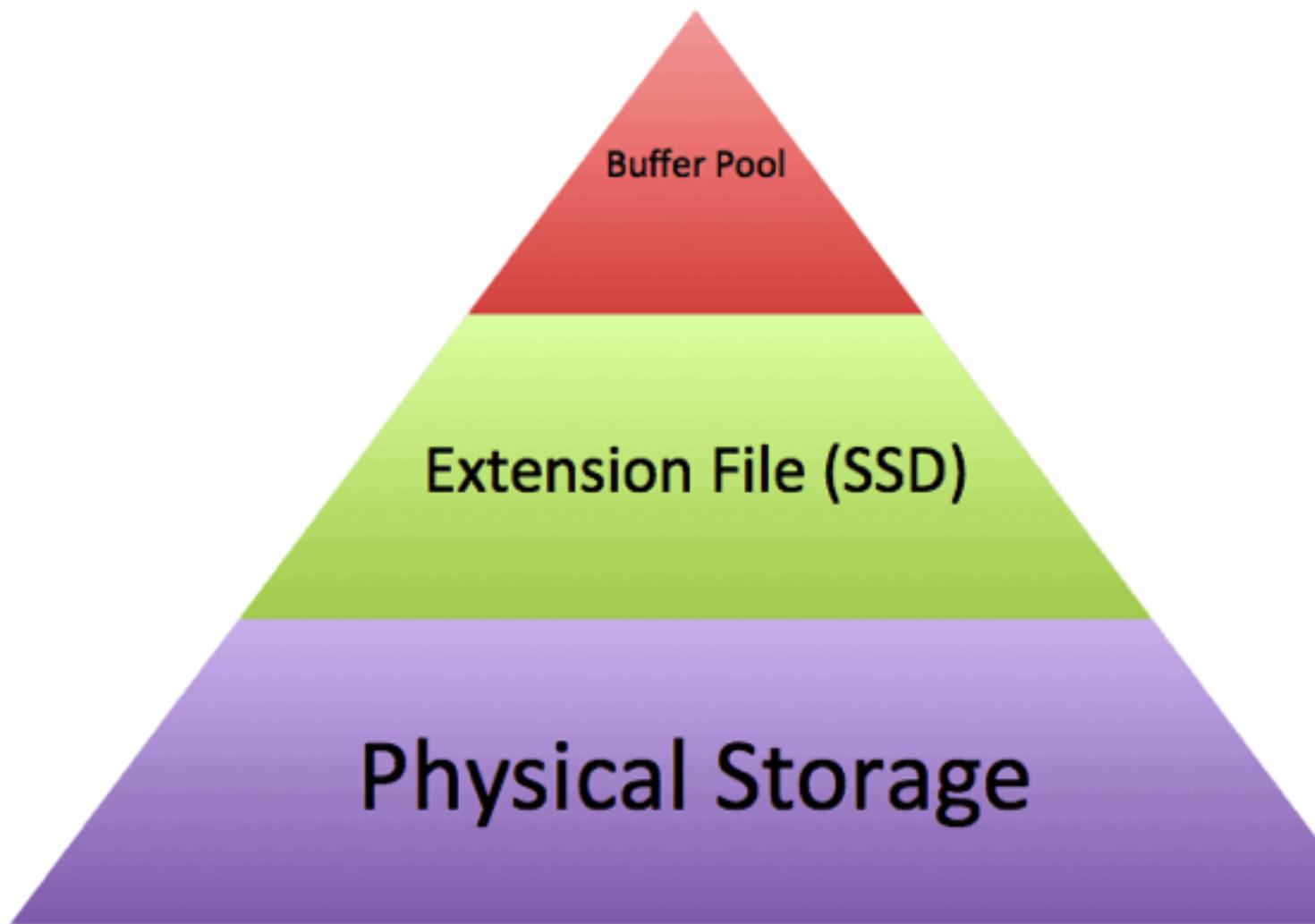
[Delayed Durability in SQL Server 2014 – Part 1](#)

[Is In-Memory OLTP Always a silver bullet for achieving better transactional speed](#)

[Delayed Durability in SQL Server 2014](#)

یکی از مصرف کنندگان اصلی حافظه در SQL Server است. برای مثال زمانیکه اطلاعاتی را از بانک اطلاعاتی دریافت می‌کنید، این داده‌ها در Buffer Pool کش می‌شوند. همچنین SQL Server اطلاعات کلیه Execution Plans را نیز در جزئی از Buffer Pool است، برای استفاده‌ی مجدد نگهداری می‌کند. هر چقدر حافظه‌ی فیزیکی سرور شما بیشتر باشد، مقدار نیز به همین میزان افزایش خواهد یافت که البته حد اکثر آن را می‌توان در تنظیمات حافظه‌ی سرور محدود کرد (Max Buffer Pool).

در دنیای واقعی میزان حافظه‌ی فیزیکی سرورها محدود است. در 2014 SQL Server راه حلی برای این مشکل تحت عنوان Buffer Pool Extensions ارائه شده است که محل قرارگیری آن را در تصویر ذیل مشاهده می‌کنید:



از یک فایل ساده که به آن Extension File نیز گفته می‌شود، تشکیل شده است و امکان ذخیره سازی آن بر روی هاردهای سریعی مانند SSD Drive می‌سرد. این فایل، ساختاری را همانند page file در سیستم عامل ویندوز دارد. در این حالت بجای اضافه کردن RAM بیشتر به سرور، یک Extension File را می‌توان بکار گرفت. هر زمان که Buffer Pool اصلی تحت فشار قرار گیرد (به میزان حافظه‌ای بیش از حافظه‌ی فیزیکی سرور نیاز باشد)، از این افزونه‌ی فایلی استفاده خواهد

شد.

اطلاعات جزئیات Buffer Pool را توسط کوئری ذیل می‌توان بدست آورد:

```
Select * from sys.dm_os_buffer_descriptors
```

نحوه‌ی فعال سازی و تنظیم Buffer Pool Extensions

قبل از هر کاری بهتر است وضعیت افزونه‌ی Buffer pool را بررسی کرد:

```
select * from sys.dm_os_buffer_pool_extension_configuration
```

SQLQuery1.sql - (l...VahidPC\Vahid (60))*

```
select * from sys.dm_os_buffer_pool_extension_configuration
```

100 %

Results Messages

	path	file_id	state	state_description	current_size_in_kb
1	NULL	-1	0	BUFFER POOL EXTENSION DISABLED	NULL

همانطور که ملاحظه می‌کنید، در حالت پیش فرض غیرفعال است.

سپس یک فایل یک گیگابایتی را به عنوان افزونه‌ی Buffer pool ایجاد می‌کنیم.

```
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION ON
(FILENAME = 'd:\BufferPoolExt.BPE', SIZE = 1GB);
```

تصویه شده‌است که این فایل را در یک درایور پر سرعت SSD قرار دهید؛ ولی محدودیتی از لحاظ محل قرارگیری ندارد (هر چند [به نظر فقط](#) در حالتیکه از SSD Drive استفاده شود واقعاً کار می‌کند).

اینبار اگر کوئری اول را اجرا کنیم، چنین خروجی قابل مشاهده است:

SQLffQuery1.sql - (l...VahidPC\Vahid (60))*

```
select * from sys.dm_os_buffer_pool_extension_configuration
```

100 %

Results Messages

	path	file_id	state	state_description	current_size_in_kb
1	d:\BufferPoolExt.BPE	0	5	BUFFER POOL EXTENSION CLEAN PAGE CACHING ENABLED	1048576

این فایل به صورت خودکار در حین ریاستارت یا خاموش شدن سرور، حذف شده و با راه اندازی مجدد آن، باز تولید خواهد شد.

تغییر اندازه‌ی افزونه‌ی Buffer pool

اگر سعی کنیم، یک گیگابایت را مثلاً به 10 گیگابایت افزایش دهیم:

```
ALTER SERVER CONFIGURATION  
SET BUFFER POOL EXTENSION ON  
(FILENAME = 'd:\BufferPoolExt.BPE', SIZE = 10GB);
```

با خطای ذیل مواجه خواهیم شد:

```
Could not change the value of the 'BPoolExtensionPath' property
```

برای رفع این مشکل، ابتدا باید افزونه‌ی Buffer pool را غیرفعال کرد:

```
ALTER SERVER CONFIGURATION  
SET BUFFER POOL EXTENSION OFF
```

سپس می‌توان مجدداً اندازه و یا مسیر دیگری را مشخص کرد. بهتر است اندازه‌ی این فایل را حدود 16 برابر حداکثر میزان حافظه‌ی سرور (Max Server Memory) تعیین کنید.

همچنین توصیه شده‌است که پس از غیرفعال کردن این افزونه، بهتر است یکبار instance را ری‌استارت کنید.

چه زمانی بهتر است از افزونه‌ی Buffer pool استفاده شود؟

در محیط‌های OLTP، استفاده از یک چنین افزونه‌ای می‌تواند میزان کارآئی و پاسخگویی سیستم را به شدت افزایش دهد (تا [50 درصد](#)).

سؤال: آیا غیرفعال کردن افزونه‌ی Buffer pool سبب از دست رفتن اطلاعات می‌شود؟
خیر، تنها pages clean را در خود ذخیره می‌کند؛ یعنی تنها اطلاعاتی که Commit شده‌اند در آن حضور خواهند داشت و در این حالت حذف آن یا ری‌استارت کردن سرور، سبب از دست رفتن اطلاعات نخواهند شد.

برای مطالعه بیشتر

[Buffer Pool Extension](#)

[SQL Server 2014 Buffer Pool Extensions](#)

[Do you require a SSD to use the Buffer Pool Extension feature in SQL Server 2014](#)

[Buffer Pool Extensions in SQL Server 2014](#)

[SQL Server 2014 – Buffer Pool Extension](#)

پشتیبانی از Spatial data در SQL Server

از SQL Server 2008 به بعد، نوع داده جدیدی به نام geography به نوع‌های قابل تعریف ستون‌ها اضافه شده است. در این نوع ستون‌ها می‌توان طول و عرض جغرافیایی یک نقطه را ذخیره کرد و سپس به کمک توابع توکاری از آن‌ها کوئری گرفت.

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Location	geography	<input checked="" type="checkbox"/>
Name	geography	<input checked="" type="checkbox"/>
Type	geometry	<input checked="" type="checkbox"/>
	hierarchyid	<input checked="" type="checkbox"/>
	image	<input checked="" type="checkbox"/>
	int	<input checked="" type="checkbox"/>
	money	<input checked="" type="checkbox"/>
	nchar(10)	<input checked="" type="checkbox"/>
	ntext	<input type="checkbox"/>

در اینجا نمونه‌ای از نحوه‌ی تعریف و همچنین مقدار دهی این نوع ستون‌ها را مشاهده می‌کنید:

```

CREATE TABLE [Geo](
[id] [int] IDENTITY(1,1) NOT NULL,
[Location] [geography] NULL
)

insert into Geo( Location , long, lat ) values
( geography::STGeomFromText ('POINT(-121.527200 45.712113)', 4326))
  
```

متد Geography::STGeoFromText یک SQL CLR function است. این متد در مثال فوق، مختصات یک نقطه را دریافت کرده است. همچنین نیاز دارد بداند که این نقطه توسط چه نوع سیستم مختصاتی ارائه می‌شود. عدد 4326 در اینجا یک SRID یا Spatial Reference System Identifier استاندارد است. برای نمونه اطلاعات ارائه شده توسط Bing یا Google از این استاندارد ارائه می‌شوند. در اینجا متدهای توکار دیگری مانند geography::STDistance برای یافتن فاصله مستقیم بین نقاط نیز ارائه شده‌اند. خروجی آن بر حسب متر است.

Entity framework در Spatial Data پشتیبانی

پشتیبانی از نوع مخصوص geography، در EF 5 توسط نوع داده‌ای DbGeography ارائه شد. این نوع داده‌ای immutable است. به این معنا که پس از نمونه سازی، دیگر مقدار آن قابل تغییر نیست.

در اینجا برای نمونه مدلی را مشاهده می‌کنید که از نوع داده‌ای DbGeography استفاده می‌کند:

```
using System.Data.Entity.Spatial;
namespace EFGeoTests.Models
{
    public class GeoLocation
    {
        public int Id { get; set; }
        public DbGeography Location { get; set; }
        public string Name { get; set; }
        public string Type { get; set; }

        public override string ToString()
        {
            return string.Format("Name:{0}, Location:{1}", Name, Location);
        }
    }
}
```

به همراه یک Context، تا کلاس GeoLocation در معرض دید EF قرار گیرد:

```
using System;
using System.Data.Entity;
using EFGeoTests.Models;

namespace EFGeoTests.Config
{
    public class MyContext : DbContext
    {
        public DbSet<GeoLocation> GeoLocations { get; set; }

        public MyContext()
            : base("Connection1")
        {
            this.Database.Log = sql => Console.WriteLine(sql);
        }
    }
}
```

برای مقدار دهی خاصیت Location از نوع DbGeography می‌توان از متدهای زیر استفاده کرد که بسیار شبیه به متدهای geography::STGeoFromText عمل می‌کنند:

```
private static DbGeography createPoint(double longitude, double latitude, int coordinateSystemId = 4326)
{
    var text = string.Format(CultureInfo.InvariantCulture.NumberFormat, "POINT({0} {1})", longitude, latitude);
    return DbGeography.PointFromText(text, coordinateSystemId);
}
```

تهیه منبع داده‌ی جغرافیایی

برای تدارک یک مثال واقعی جغرافیایی، نیاز به اطلاعاتی دقیق داریم. این نوع اطلاعات عموماً توسط یک سری فایل مخصوص به نام [Shapefiles](#) که حاوی اطلاعات برداری جغرافیایی هستند ارائه می‌شوند. برای نمونه اطلاعات جغرافیایی به روز ایران را از آدرس زیر می‌توانید دریافت کنید:

<http://download.geofabrik.de/asia/iran.html>

<http://download.geofabrik.de/asia/iran-latest.shp.zip>

پس از دریافت این فایل، به تعدادی فایل با پسوندهای shp، shx و dbf خواهیم رسید. فایل‌های shx یک سری ایندکس بوده و فایل‌های dbf از نوع بانک اطلاعاتی dBase IV می‌باشند.

همچنین اگر فایل‌های prj را باز کنید، یک چنین اطلاعاتی در آن موجودند:

```
GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137,298.257223563]],PRIMEM["Greenwich",0],UNIT["Degree",0.017453292519943295]]
```

نکته‌ی مهمی که در اینجا باید مد نظر داشت، استاندارد معادل است با استاندارد [EPSG GCS_WGS_1984](#) آن است. این استاندارد coordinateSystemId SQL Server حائز اهمیت است (پارامتر createPoint) و ممکن است از هر فایلی به فایل دیگر متفاوت باشد.

خواندن فایل‌های shp در دات نت

پس از دریافت فایل‌های shp و بانک‌های اطلاعاتی مرتبط با اطلاعات جغرافیایی ایران، اکنون نوبت به پردازش این فایل‌های مخصوص با فرمت بانک اطلاعاتی فاکس پرو مانند، رسیده است. برای این منظور می‌توان از پروژه‌ی سورس باز ذیل استفاده کرد:

[C# Esri Shapefile Reader](#)

این پروژه در خواندن فایل‌های shp بدون نقص عمل می‌کند اما توانایی خواندن نام‌های فارسی وارد شده در این نوع بانک‌های اطلاعاتی را ندارد. برای رفع این مشکل، سورس آن را از [Codeplex](#) دریافت کنید. سپس فایل Shapefile.cs را گشوده و ابتدای خاصیت Current آنرا به نحو ذیل تغییر دهید:

```
/// <summary>
/// Gets the current shape in the collection
/// </summary>
public Shape Current
{
    get
    {
        if (_disposed) throw new ObjectDisposedException("Shapefile");
        if (! _opened) throw new InvalidOperationException("Shapefile not open.");
        // get the metadata
        StringDictionary metadata = null;
        if (! RawMetadataOnly)
        {
            metadata = new StringDictionary();
            for (int i = 0; i < _dbReader.FieldCount; i++)
            {
                string value = _dbReader.GetValue(i).ToString();
                if (_dbReader.GetDataTypeName(i) == "DBTYPE_NVARCHAR")
                {
                    برای نمایش متون فارسی نیاز است //
                    value = Encoding.UTF8.GetString(Encoding.GetEncoding(720).GetBytes(value));
                }
                metadata.Add(_dbReader.GetName(i),
                            value);
            }
        }
    }
}
```

در اینجا فقط سطر استفاده از Encoding خاصی با شماره 720 و تبدیل آن به UTF8 اضافه شده است. پس از آن بدون مشکل می‌توان برچسب‌های فارسی را از فایل‌های dBase IV این نوع بانک‌های اطلاعاتی استخراج کرد (اصلاح شده‌ی آن در فایل پیوست مطلب موجود است).

```
using System.Collections.Generic;
using System.Linq;
using Catfood.Shapefile;

namespace EFGeoTests
{
    public class MapPoint
    {
        public Dictionary<string, string> Metadata { set; get; }
        public double X { set; get; }
        public double Y { set; get; }
    }
}
```

```

public static class ShapeReader
{
    public static IList<MapPoint> ReadShapeFile(string path)
    {
        var results = new List<MapPoint>();

        using (var shapefile = new Shapefile(path))
        {
            foreach (var shape in shapefile)
            {
                if (shape.Type != ShapeType.Point)
                    continue;

                var shapePoint = shape as ShapePoint;
                if (shapePoint == null)
                    continue;

                var metadataNames = shape.GetMetadataNames();
                if (!metadataNames.Any())
                    continue;

                var metadata = new Dictionary<string, string>();
                foreach (var metadataName in metadataNames)
                {
                    metadata.Add(metadataName, shape.GetMetadata(metadataName));
                }

                results.Add(new MapPoint
                {
                    Metadata = metadata,
                    X = shapePoint.Point.X,
                    Y = shapePoint.Point.Y
                });
            }
        }

        return results;
    }
}

```

در کدهای فوق به کمک کتابخانه‌ی Esri Shapefile Reader C#، اطلاعات نقاط بانک اطلاعاتی shape files را خوانده و به صورت لیست‌هایی از MapPoint بازگشت می‌دهیم. نکته‌ی مهم آن، Metadata است که از هر فایلی به فایل دیگر می‌توان متفاوت باشد. به همین جهت این اطلاعات را به شکل ویژگی‌های key/value در این نوع بانک‌های اطلاعاتی ذخیره می‌کنند.

افزودن اطلاعات جغرافیایی به بانک اطلاعاتی SQL Server به کمک Entity framework

فایل places.shp را در مجموعه فایل‌هایی که در ابتدای بحث عنوان شدند، می‌توانید مشاهده کنید. قصد داریم اطلاعات نقاط آن را به مدل GeoLocation انتساب داده و سپس ذخیره کنیم:

```

var points = ShapeReader.ReadShapeFile("IranShapeFiles\\places.shp");
using (var context = new MyContext())
{
    context.Configuration.AutoDetectChangesEnabled = false;
    context.Configuration.ProxyCreationEnabled = false;
    context.Configuration.ValidateOnSaveEnabled = false;

    if (context.GeoLocations.Any())
        return;

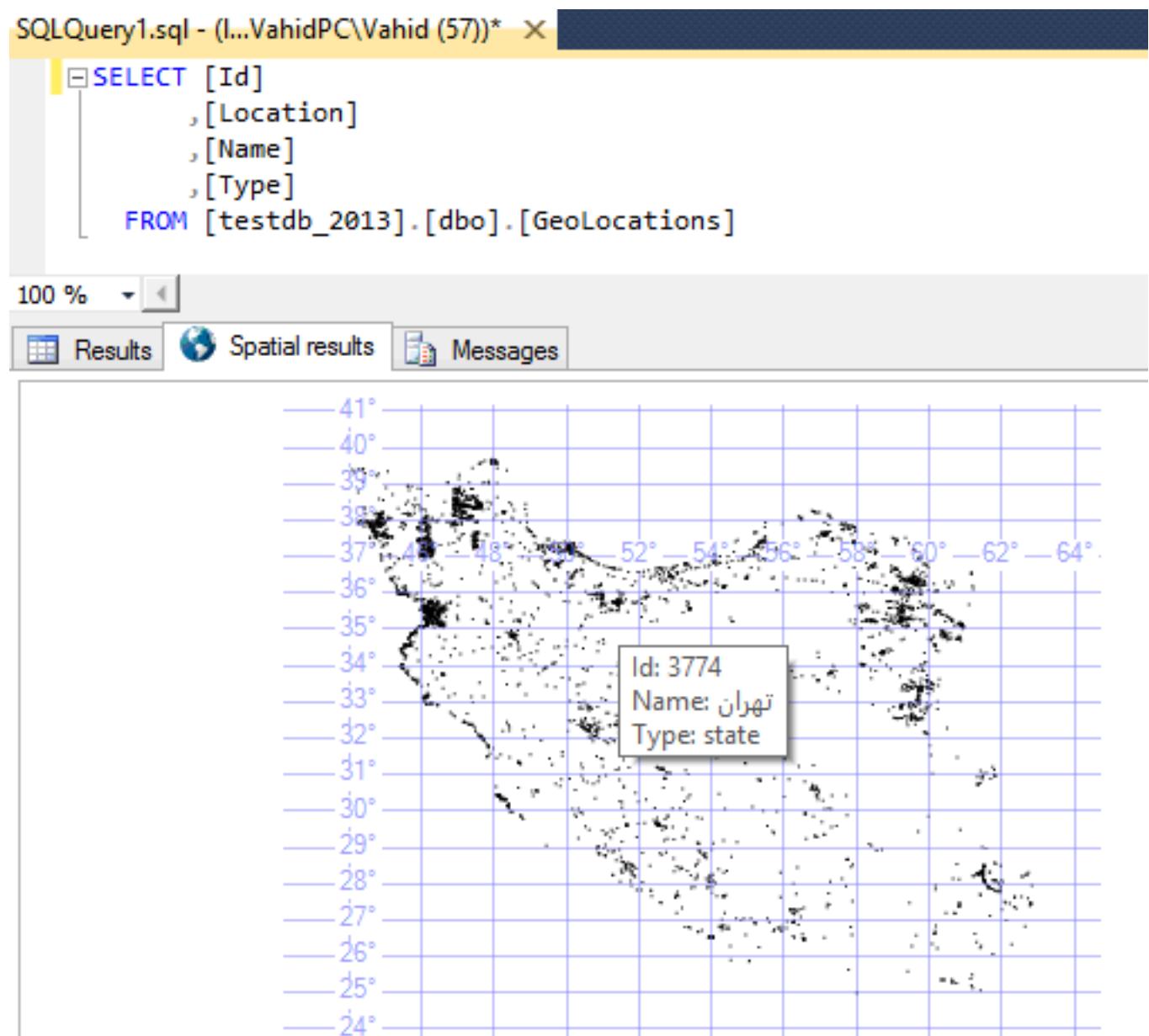
    foreach (var point in points)
    {
        context.GeoLocations.Add(new GeoLocation
        {
            Name = point.Metadata["name"],
            Type = point.Metadata["type"],
            Location = createPoint(point.X, point.Y)
        });
    }

    context.SaveChanges();
}

```

تعریف متد createPoint را که بر اساس X و Y نقاط، معادل قابل پذیرش آنرا جهت SQL Server تهیه می‌کند، در ابتدای بحث مشاهده کردید.

در فایل‌های مرتبط با places.shp، متادیتا name، معادل نام شهرهای ایران است و type آن بیانگر شهر، روستا و امثال آن می‌باشد. پس از اینکه اطلاعات مکان‌های ایران، در SQL Server ذخیره شدند، نمایش بصری آن‌ها را در management studio نیز می‌توان مشاهده کرد:



کوئری گرفتن از اطلاعات جغرافیایی

فرض کنید می‌خواهیم مکان‌هایی را با فاصله کمتر از 5 کیلومتر از تهران پیدا کنیم:

```
var tehran = createPoint(51.4179604, 35.6884243);
using (var context = new MyContext())
{
    // find any locations within 5 kilometers ordered by distance
```

```
var locations = context.GeoLocations
    .Where(loc => loc.Location.Distance(tehran) < 5000)
    .OrderBy(loc => loc.Location.Distance(tehran))
    .ToList();

foreach (var location in locations)
{
    Console.WriteLine(location.Name);
}
```

همانطور که پیشتر نیز عنوان شد، متد `Distance` بر اساس متر کار می‌کند. به همین جهت برای تعریف 5 کیلومتر به نحو فوق عمل شده است. همچنین نحوه مرتب سازی اطلاعات نیز بر اساس فاصله از یک مکان مشخص صورت گرفته است. و یا اگر بخواهیم دقیقاً بر اساس مختصات یک نقطه، مکانی را بیابیم، می‌توان از متد `SpatialEquals` استفاده کرد:

```
var tehran = createPoint(51.4179604, 35.6884243);
using (var context = new MyContext())
{
    // find any locations within 5 kilometers ordered by distance
    var tehranLocation = context.GeoLocations.FirstOrDefault(loc =>
        loc.Location.SpatialEquals(tehran));
    if (tehranLocation != null)
    {
        Console.WriteLine(tehranLocation.Type);
    }
}
```

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید:

[EFGeoTests.zip](#)

نظرات خوانندگان

نویسنده: محمد باقر سیف اللهی
تاریخ: ۱۰:۴۱ ۱۳۹۳/۰۳/۳۱

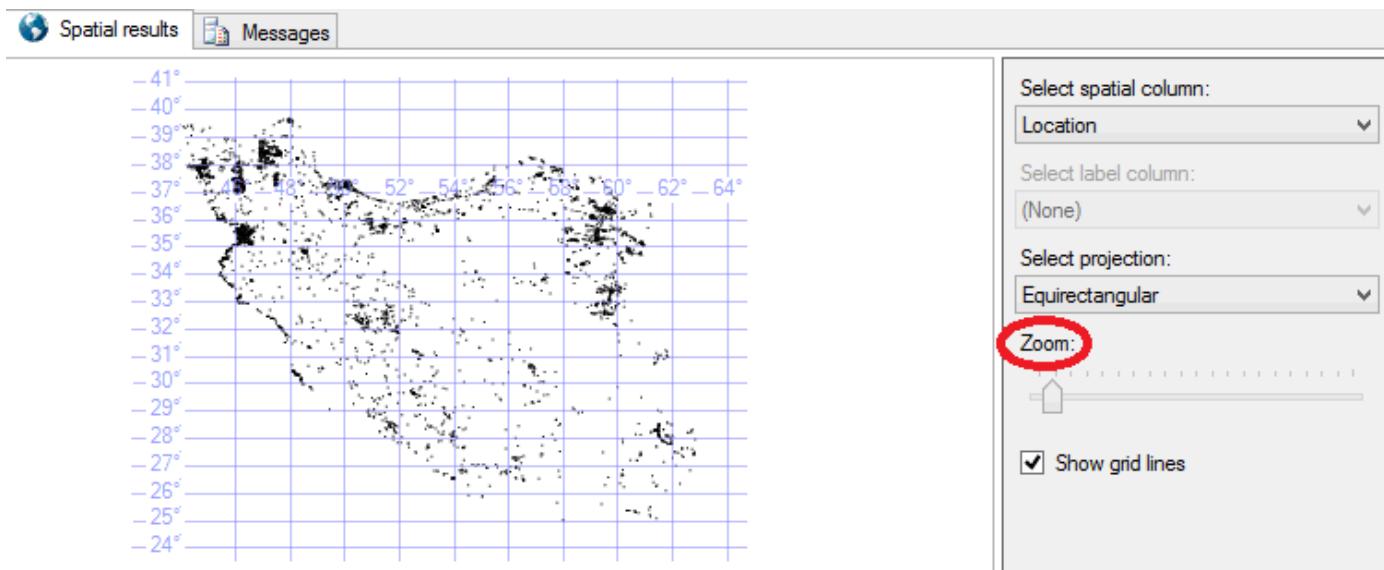
سلام مشکلی که قبل تر در حین کار با فایل‌های shp برخوردم، تغییر مقیاس نقشه و نگاشت مختصات خاص و توزیع شده ای بود که باید روی نقشه نمایش داده می‌شد. (مثلاً نمایش مراکز استان‌ها روی نقشه و تغییر scale نقشه و به تبع اون، تغییر مکان مختصات) برای این کار چه راهکاری هست؟

- * دسترسی به فایل‌های dbf و prj و sbx و sbn و shx وجود دارد.

ممnon

نویسنده: وحید نصیری
تاریخ: ۱۱:۱۰ ۱۳۹۳/۰۳/۳۱

این‌ها بیشتر مسایل نمایشی است و توانمندی ابزار نمایش دهنده‌ی اطلاعات نقشه. نیازی نیست در اصل دیتابیس و اطلاعات، تغییری حاصل شود؛ چون اندازه‌ی نمایشی حتی اگر 10 برابر هم شود، در فاصله‌ی بین تهران و شیراز نهایتاً تغییری حاصل نخواهد شد و طول و عرض جغرافیایی مکان‌ها ثابت خواهند ماند. برای نمونه اگر مثال پیوست شده را اجرا کنید، خود management studio امکان تغییر اندازه‌ی نمایشی را دارد:



در برنامه‌های دات نت هم برای مثال از [SharpMap](#) می‌شود برای نمایش این نوع اطلاعات به همراه [تغییر اندازه و ابعاد خودکار](#) نقشه استفاده کرد. برای برنامه‌های وب هم [VectorMap.js](#) چنین قابلیت‌هایی را دارد.

نویسنده: بهزاد ایزدی
تاریخ: ۱۵:۰۰ ۱۳۹۳/۰۴/۰۲

سلام
ممnon برای این آموزش خوب بعضی مناطق فایل shp رو ندارند و فقط دو نوع OSM برashون موجود هست راهی برای خواندن اونها وجود داره ؟

نوبسنده: وحید نصیری
تاریخ: ۱۵:۱۲ ۱۳۹۳/۰۴/۰۲

- برنامه‌ی سورس باز [ArcGIS Editor for OpenStreetMap](#) یک چنین قابلیتی را دارد.
- همچنین یک سری 4 قسمتی را در [اینجا](#) می‌توانید در مورد تبدیل open street maps به داده‌های SQL Server مطالعه کنید.

نوبسنده: شاهین کیاست
تاریخ: ۱۷:۶ ۱۳۹۳/۰۷/۱۳

آیا ممکن هست به جای نوع داده‌ی `DbGeography` از نوع داده‌ی `SQLGeometry` استفاده کرد؟
- ویرایش
- تنها کافی است نوع `Property` مورد نظر را `DbGeometry` تعیین کرد.

1- مقدمه پارتیشن بندی در بانک اطلاعاتی SQL Server، از ویژگی‌هایی است که از نسخه 2005، به این محصول اضافه شده است. بکارگیری این قابلیت که با Split کردن، محتوای یک جدول و قرار دادن آنها در چندین فایل، برای جداول حجمی، به ویژه جداولی که داده‌های آن حاوی مقادیر تاریخچه‌ای است، بسیار سودمند است.

садگی در مدیریت داده‌ها و شاخص‌های موجود یک جدول (از قبیل اندازه فضای ذخیره سازی و استراتژی Back up گیری)، اجرای سریعتر کوئری‌هایی که روی یک محدوده از داده‌ها کار می‌کنند و سهولت در آرشیو داده‌های قدیمی یک جدول، از قابلیت‌هایی است که استفاده از این ویژگی بوجود می‌آورد.

Scalability استفاده از این ویژگی روی یک بانک اطلاعاتی و در یک Instance است. بنابراین مباحث مرتبه با معماری را پوشش نمی‌دهد و صرفاً Solution ای است که در یک Instance بانک اطلاعاتی استفاده می‌شود.

Filegroup و Data File 2- هر بانک اطلاعاتی در حالت پیش فرض، شامل یک فایل داده‌ای (MDF) و یک فایل ثبت تراکنشی (LDF). می‌باشد. می‌توان جهت ذخیره سطرهای داده‌ای از فایل‌های بیشتری تحت نام فایل‌های ثانویه (NDF). استفاده نمود. به همان طریق که در فایل سیستم، فایل‌ها به پوشش‌ها تخصیص داده می‌شوند، می‌توان Data File را به Filegroup تخصیص داد. چنانچه چندین Data File به یک Filegroup تخصیص داده شوند، داده‌ها در تمامی Data File‌ها به طریق Round-Robin توزیع می‌شوند.

Partition Function 3- مطابق با مقادیر تعريف شده در بدنه دستور، محدوده داده‌ای (پارتیشن‌ها) با استفاده از Function ایجاد می‌شود. با در نظر گرفتن ستونی که به عنوان Partition Key انتخاب شده، این تابع یک Data Type را به عنوان ورودی دریافت می‌کند. در هنگام تعريف محدوده برای پارتیشن‌ها، به منظور مشخص کردن محدوده هر پارتیشن از Left و Right استفاده می‌شود.

نمایش دهنده‌ی حد بالای هر محدوده است و به طور مشابه، Right برای مشخص کردن حد پائین آن محدوده استفاده می‌شود. به منظور درک بهتر، به شکل زیر توجه نمایید:

```
CREATE PARTITION FUNCTION myRangePF1
(int) AS RANGE LEFT FOR VALUES
(1,100,1000)
```

Partition	1	2	3	4
Values	C1 <=1	C1 > 1 AND C1<=100	C1>100 AND C1<=1000	C1>1000

```
CREATE PARTITION FUNCTION myRangePF1
(int) AS RANGE RIGHT FOR VALUES
(1,100,1000)
```

Partition	1	2	3	4
Values	C1 <1	C1 >= 1 AND C1<100	C1>=100 AND C1<1000	C1>=1000

همانطور که مشاهده می‌شود، همواره نیاز به یک Filegroup اضافه‌تری از آنچه مورد نظرتان در تعریف تابع است، می‌باشد. بنابراین اگر Function دارای n مقدار باشد، به $n+1$ مقدار برای Filegroup نیاز است. همچنین هیچ محدودیتی برای اولین و آخرین بازه در نظر گرفته نمی‌شود. بنابراین جهت محدود کردن مقادیری که در این بازه‌ها قرار می‌گیرند، می‌توان از Check Constraint استفاده نمود.

Right or Left 3-1-

یک سوال متداول اینکه از کدام مورد استفاده شود؟ در پاسخ باید گفت، به چگونگی تعریف پارسیشن هایتان وابسته است. مطابق شکل، تنها تفاوت این دو، در نقاط مرزی هر یک از پارسیشن‌ها می‌باشد. در بیشتر اوقات هنگام کار با داده‌های عددی می‌توان از Left استفاده نمود و بطور مشابه هنگامیکه نوع داده‌ها از جنس زمان است، می‌توان از Right استفاده کرد.

Partition Schema 4-

گام بعدی پس از ایجاد Partition Function، تعریف Partition Schema است، که به منظور قرار گرفتن هر یک از پارسیشن‌های تعریف شده توسط Filegroup در Function های مناسب آن استفاده می‌شود.

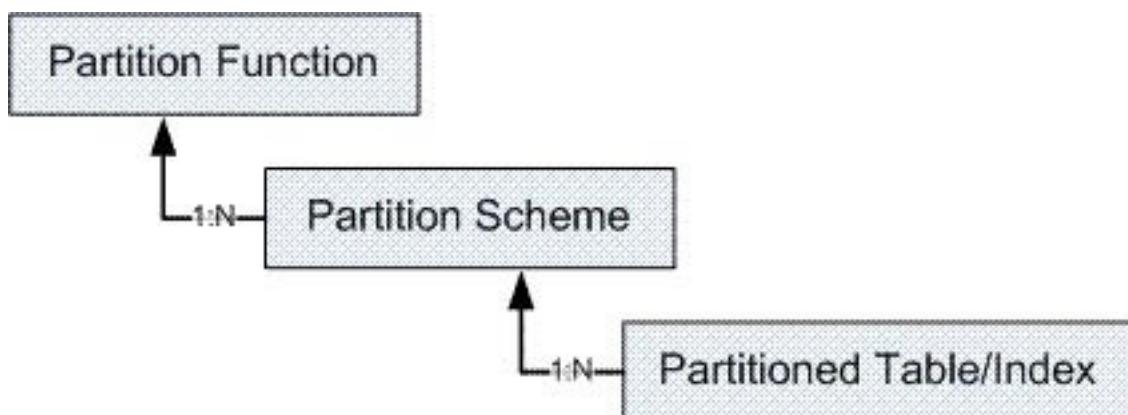
Partition Table 5-

گام پایانی ایجاد یک جدول، استفاده از Partition Schema است، که داده‌ها را با توجه به رویه درون Partition Function مورد استفاده، ذخیره می‌کند. همانطور که می‌دانید هنگام ایجاد یک جدول، می‌توان مکان ذخیره شدن آنرا مشخص نمود.

```
Create Table <name> (...) ON ...
```

دستور بعد از بخش ON، مشخص کننده مکان ذخیره جدول می‌باشد.

در هنگام ایجاد یک جدول، معمولاً جدول در Filegroup پیش فرض که PRIMARY است، قرار می‌گیرد. می‌توان با نوشتن نام Partition Schema و همچنین Partition Key که پیشتر ذکر آن رفت، بعد از بخش ON، برای جدول مشخص نمائیم که داده‌های آن به چه ترتیبی ذخیره شوند. ارتباط این سه به شرح زیر است:



توجه شود زمانیکه یک جدول اضافه می‌شود، یک Unique Clustered Index به آن ساخته می‌شود. چنانچه Primary Key شامل یک Clustered Index باشد، جدول با استفاده از این ستون (ستون‌های) شاخص ذخیره خواهد شد، در حالیکه اگر Non Clustered Index شامل یک Primary Key باشد، یک ساختار ذخیره-سازی اضافی ایجاد خواهد شد که داده‌های جدول در آن قرار خواهند گرفت.

Index & Data Alignment 6-

به عنوان یک Best Practice هنگام ایجاد یک Partition Table به منظور پارسیشن بندی، از ساختار Aligned Index استفاده شود. بدین ترتیب که تعریف Index، شامل Partition Key (ستونی که معیاری برای پارسیشن بندی است) باشد. چنانچه این عمل انجام شود، داده‌های ذخیره شده مرتبط با هر پارسیشن متناظر با همان شاخص، در فایل داده‌ای (NDF). ذخیره خواهد شد. از این رو چنانچه کوئری درخواست شده از جدول روی یک محدوده باشد

Where [OrderDate] Between ...

تنها از شاخص متناظر با این داده استفاده می‌شود. بدین ترتیب بکارگیری آن برای Execution Plan بسیار سودمند خواهد بود. همچنین می‌توان استراتژی بازیافت سودمندی با Back up گیری از Filegroup ایجاد کرد. هنگامی که Index‌ها به صورت Aligned هستند می‌توان در کسری از ثانیه، محتوای یک Partition را به یک جدول دیگر منتقل نمود (تنها با تغییر در Meta Data آن).

بدین ترتیب برای بهرمندی از این مزایا، استفاده از Aligned Index توصیه شده است.

Operations 7-

از نیازمندی‌های متداول در پارسیشنینگ می‌توان به افزودن، حذف پارسیشن‌ها و جابجایی محتوای یک پارسیشن که برای عملیات آرشیو استفاده می‌شود، اشاره کرد.

Split Partition 7-1-

به منظور ایجاد یک محدوده جدید به پارسیشن‌ها استفاده می‌شود. یک نکته مهم مادامی که عملیات انتقال داده‌ها به پارسیشن جدید انجام می‌گیرد، روی جدول یک قفل انحصاری قرار می‌گیرد و بدین ترتیب عملیات ممکن است زمانی باشد.

به عنوان یک Best Practice همواره یک Split Partition خالی را نمایید و پس از آن اقدام به بارگذاری داده در آن نمائید. به یاد داشته باشید پیش از انجام عملیات splitting روی Partition Function با تغییر در Partition Schema (و بکارگیری Next Used) مشخص نمایید چه محدوده‌ای در این Filegroup جدید قرار خواهد گرفت.

Merge Partition 7-2-

به منظور ادغام پارسیشن‌ها استفاده می‌شود، چنانچه پارسیشن خالی نیست، برای عملیات ادغام مسائل Performance به علت اینکه در طول عملیات از Lock (قفل انحصاری) استفاده می‌شود، در نظر گرفته شود.

Switch Partition 7-3-

چنانچه جدول و شاخص‌های آن به صورت Aligned هستند، می‌توانید از Switch in و Switch out استفاده نمایید. عملیات بدین ترتیب انجام می‌شود که بلافضله محتوای یک پارسیشن یا جدول (Source) در یک پارسیشن خالی جدولی دیگر و یا یک جدول خالی (Target) قرار می‌گیرد. عملیات تنها روی Meta Data انجام می‌گیرد و هیچ داده‌ای منتقل نمی‌شود.

محدودیت‌های بکارگیری به شرح زیر است:

- جدول یا پارسیشن Target باید حتماً خالی باشد.
- جداول Target و Source حتماً باید در یک Filegroup یکسان قرار داشته باشند.
- جدول Source باید حاوی Aligned Index‌های مورد نیاز Target و همچنین مطابقت در Filegroup را دارا باشد.
- چنانچه Target به عنوان یک پارسیشن است، اگر Source جدول است بایست دارای یک Check Constraint باشد در غیر این صورت چنانچه یک پارسیشن است باید محدوده آن در محدوده Target قرار گیرد.

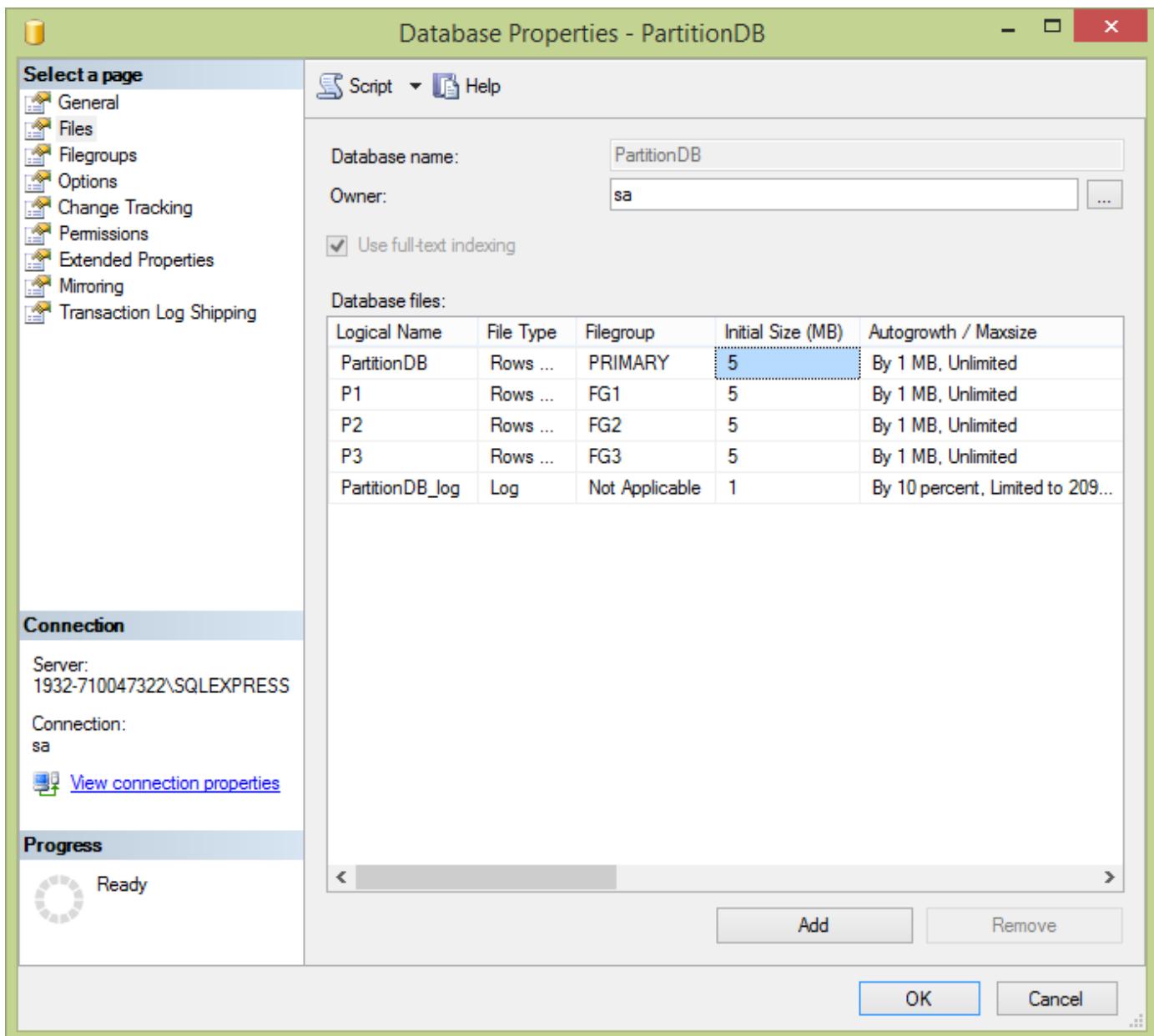
8- بررسی یک سناریوی نمونه

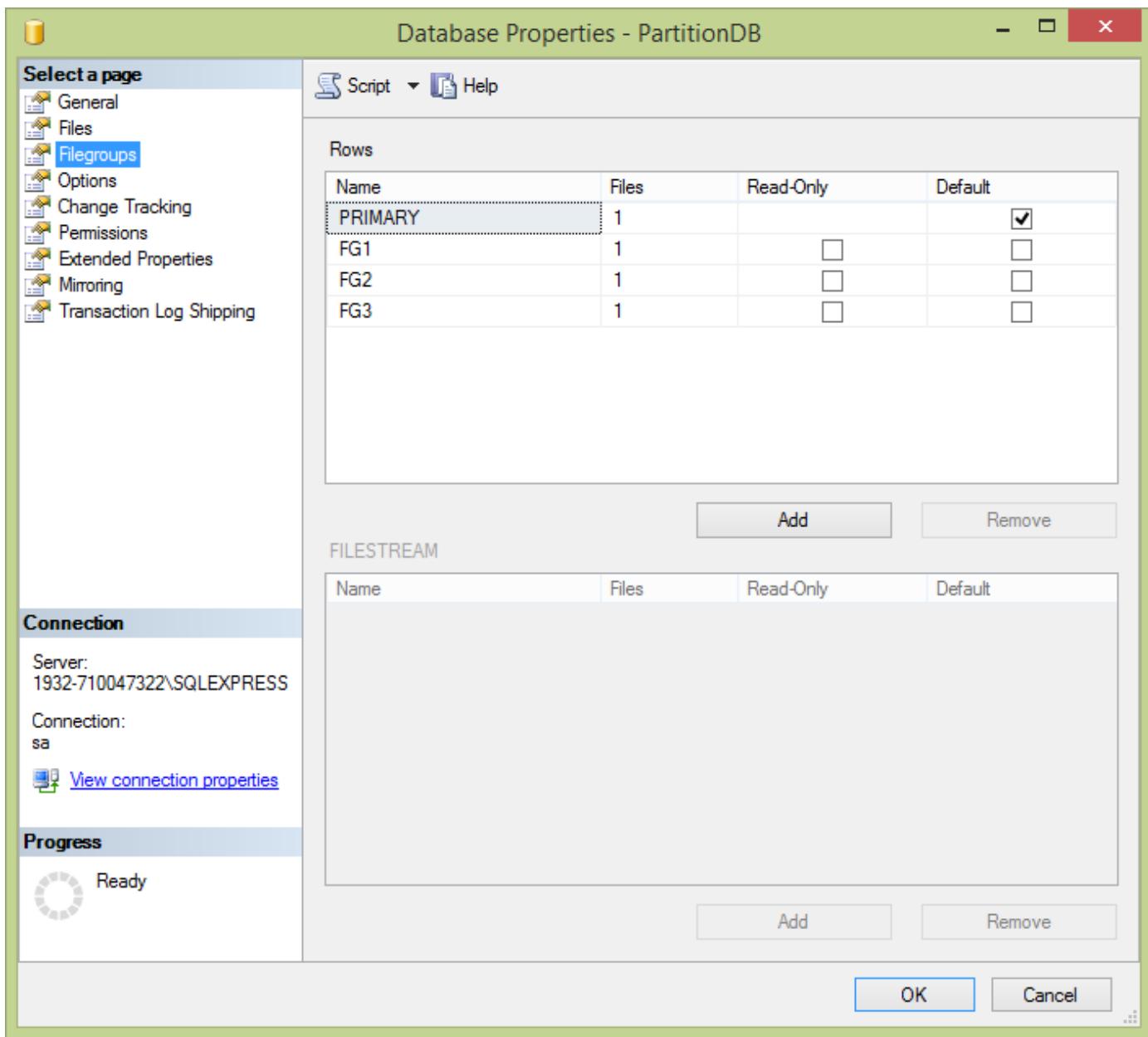
در ابتدا یک بانک اطلاعاتی را به طریق زیر ایجاد می‌کنیم:

این بانک مطابق تصویر، شامل 3 عدد فایل گروپ (FG1، FG2 و FG3) و 3 عدد دیتا فایل (P1، P2 و P3) می‌باشد. Filegroup P1 پیش فرض است، که چنانچه در تعریف جداول به نام Primary Partition Key و Partition Schema مرتبط باشد، به طور پیش فرض در Filegroup موسوم به Primary قرار می‌گیرد. چنانچه چک باکس Default انتخاب شود، همانطور که قابل حدس زدن است، آن Filegroup در صورت مشخص نکردن نام در تعریف جدول، به عنوان مکان ذخیره سازی انتخاب می‌شود. چک باکس Read Only نیز همانطور که از نامش پیداست، چنانچه روی یک Filegroup تنظیم گردد، عملیات مربوط به Write روی

استفاده از قابلیت پارسیشن بندی در آرشیو جداول بانک‌های اطلاعاتی SQL Server

داده‌های آن قابل انجام نیست و برای Filegroup هایی که جنبه نگهداری آرشیو را دارند، قابل استفاده است. چنانچه Filegroup ای را از حالت Read Only دوباره خارج کنیم، می‌توان عملیات Write را دوباره برای آن انجام داد.





پس از ایجاد بانک اطلاعاتی، گام بعدی ایجاد یک Partition Schema و پس از آن یک Partition Function می‌کنید در Filegroup از سه مقدار استفاده شده، بنابراین در Partition Schema باید از چهار Partition Function استفاده شود، که در مثال ما از Filegroup پیش فرض که Primary است، استفاده شده است.

```
USE [PartitionDB]
GO
CREATE PARTITION FUNCTION pfOrderDateRange(DATETIME)
AS
RANGE LEFT FOR VALUES ('2010/12/31','2011/12/31','2012/12/31')
GO
CREATE PARTITION SCHEME psOrderDateRange
AS
PARTITION pfOrderDateRange TO (FG1,FG2,FG3,[PRIMARY])
GO
```

پس از طی گام‌های قبل، به ایجاد یک جدول به صورت Aligned Index مبادرت ورزیده می‌شود.

```
CREATE TABLE Orders
```

استفاده از قابلیت پارسیشن بندی در آرشیو جداول بانک‌های اطلاعاتی SQL Server

```
(  
    OrderID INT IDENTITY(1,1) NOT NULL,  
    OrderDate DATETIME NOT NULL,  
    OrderFreight MONEY NULL,  
    ProductID INT NULL,  
    CONSTRAINT PK_Orders PRIMARY KEY CLUSTERED (OrderID ASC, OrderDate ASC)  
    ON psOrderDateRange (OrderDate)  
 ) ON psOrderDateRange (OrderDate)  
GO
```

در ادامه برای بررسی درج اطلاعات در پارسیشن با توجه به محدوده آنها اقدام به افزودن رکوردهایی در جدول ساخته شده می‌نماییم.

```
SET NOCOUNT ON  
DECLARE @OrderDate DATETIME  
DECLARE @X INT  
SET @OrderDate = '2010/01/01'  
SET @X = 0  
WHILE @X < 300  
BEGIN  
    INSERT dbo.Orders ( OrderDate, OrderFreight, ProductID)  
    VALUES( @OrderDate + @X, @X + 10, @X)  
    SET @X = @X + 1  
END  
GO  
SET NOCOUNT ON  
DECLARE @OrderDate DATETIME  
DECLARE @X INT  
SET @OrderDate = '2011/01/01'  
SET @X = 0  
WHILE @X < 300  
BEGIN  
    INSERT dbo.Orders ( OrderDate, OrderFreight, ProductID)  
    VALUES( @OrderDate + @X, @X + 10, @X)  
    SET @X = @X + 1  
END  
GO  
SET NOCOUNT ON  
DECLARE @OrderDate DATETIME  
DECLARE @X INT  
SET @OrderDate = '2012/01/01'  
SET @X = 0  
WHILE @X < 300  
BEGIN  
    INSERT dbo.Orders ( OrderDate, OrderFreight, ProductID)  
    VALUES( @OrderDate + @X, @X + 10, @X)  
    SET @X = @X + 1  
END  
GO
```

از طریق دستور Select زیر می‌توان نحوه توزیع داده‌ها را در جدول مشاهده کرد.

```
USE [PartitionDB]  
GO  
SELECT OBJECT_NAME(i.object_id) AS OBJECT_NAME,  
p.partition_number, fg.NAME AS FILEGROUP_NAME, ROWS, au.total_pages,  
CASE boundary_value_on_right  
WHEN 1 THEN 'Less than'  
ELSE 'Less or equal than' END AS 'Comparition', VALUE  
FROM sys.partitions p JOIN sys.indexes i  
ON p.object_id = i.object_id AND p.index_id = i.index_id  
JOIN sys.partition_schemes ps ON ps.data_space_id = i.data_space_id  
JOIN sys.partition_functions f ON f.function_id = ps.function_id  
LEFT JOIN sys.partition_range_values rv  
ON f.function_id = rv.function_id  
AND p.partition_number = rv.boundary_id  
JOIN sys.destination_data_spaces dds  
ON dds.partition_scheme_id = ps.data_space_id  
AND dds.destination_id = p.partition_number  
JOIN sys.filegroups fg  
ON dds.data_space_id = fg.data_space_id  
JOIN (SELECT container_id, SUM(total_pages) AS total_pages  
FROM sys.allocation_units  
GROUP BY container_id) AS au  
ON au.container_id = p.partition_id WHERE i.index_id < 2
```

خروجی دستور فوق به شرح زیر است:

	OBJECT_NAME	partition_number	FILEGROUP_NAME	ROWS	total_pages	Comparison	VALUE
1	Orders	1	FG1	300	4	Less or equal than	2010-12-31 00:00:00.000
2	Orders	2	FG2	300	4	Less or equal than	2011-12-31 00:00:00.000
3	Orders	3	FG3	300	4	Less or equal than	2012-12-31 00:00:00.000
4	Orders	4	PRIMARY	0	0	Less or equal than	NULL

در ادامه به ایجاد یک Filegroup جدید می‌پردازیم.

```
/* Query 2-3- Split a partition*/
-- Add FG4:
ALTER DATABASE PartitionDB ADD FILEGROUP FG4
Go
ALTER PARTITION SCHEME [psOrderDateRange] NEXT USED FG4
GO
ALTER PARTITION FUNCTION [pfOrderDateRange]() SPLIT RANGE('2013/12/31')
GO
-- Add Partition 4 (P4) to FG4:
GO
ALTER DATABASE PartitionDB ADD FILE
(
NAME = P4,
FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL10_50.SQLEXPRESS\MSSQL\DATA\P4.NDF'
, SIZE = 1024KB , MAXSIZE = UNLIMITED, FILEGROWTH = 10%)
TO FILEGROUP [FG4]
--
GO
```

و در ادامه به درج اطلاعاتی برای بررسی نحوه توزیع داده‌ها در Filegroup هایمان می‌پردازیم.

```
SET NOCOUNT ON
DECLARE @OrderDate DATETIME
DECLARE @X INT
SET @OrderDate = '2013/01/01'
SET @X = 0
WHILE @X < 300
BEGIN
INSERT dbo.Orders ( OrderDate, OrderFreight, ProductID)
VALUES( @OrderDate + @X, @X + 10, @X)
SET @X = @X + 1
END
GO
SET NOCOUNT ON
DECLARE @OrderDate DATETIME
DECLARE @X INT
SET @OrderDate = '2012/01/01'
SET @X = 0
WHILE @X < 300
BEGIN
INSERT dbo.Orders ( OrderDate, OrderFreight, ProductID)
VALUES( @OrderDate + @X, @X + 10, @X)
SET @X = @X + 1
END
GO
```

خروجی کار تا این مرحله به شکل زیر است:

استفاده از قابلیت پارتیشن بندی در آرشیو جداول بانک‌های اطلاعاتی SQL Server

	OBJECT_NAME	partition_number	FILEGROUP_NAME	ROWS	total_pages	Compartition	VALUE
1	Orders	1	FG1	300	4	Less or equal than	2010-12-31 00:00:00.000
2	Orders	2	FG2	300	4	Less or equal than	2011-12-31 00:00:00.000
3	Orders	3	FG3	600	5	Less or equal than	2012-12-31 00:00:00.000
4	Orders	5	PRIMARY	0	0	Less or equal than	NULL
5	Orders	4	FG4	300	4	Less or equal than	2013-12-31 00:00:00.000

جهت ادغام پارتیشن‌ها به طریق زیر عمل می‌شود:

```
/* Query 2-4- Merge Partitions */
ALTER PARTITION FUNCTION [pfOrderDateRange]() MERGE RANGE('2010/12/31')
Go
```

پس از اجرای دستور فوق خروجی به شکل زیر خواهد بود:

	OBJECT_NAME	partition_number	FILEGROUP_NAME	ROWS	total_pages	Compartition	VALUE
1	Orders	1	FG2	600	7	Less or equal than	2011-12-31 00:00:00.000
2	Orders	2	FG3	600	5	Less or equal than	2012-12-31 00:00:00.000
3	Orders	4	PRIMARY	0	0	Less or equal than	NULL
4	Orders	3	FG4	300	4	Less or equal than	2013-12-31 00:00:00.000

به منظور آرشیو نمودن اطلاعات به طریق زیر از Switch استفاده می‌کنیم. ابتدا یک جدول موقتی برای ذخیره رکوردهایی که قصد آرشیو آنها را داریم، ایجاد می‌کنیم. همانگونه که در تعریف جدول مشاهده می‌کنید، نام Filegroup ای که برای ساخت این جدول استفاده می‌شود، با Filegroup ای که قصد آرشیو اطلاعات آنرا داریم، یکسان است.

در ادامه می‌توان مثلاً با ایجاد یک Temporary Table به انتقال این اطلاعات بدون توجه به Filegroup آنها پرداخت.

```
/* Query 2-5- Switch Partitions */
USE [PartitionDB]
GO
CREATE TABLE [dbo].[Orders_Temp](
[OrderID] [int] IDENTITY(1,1) NOT NULL,
[OrderDate] [datetime] NOT NULL,
[OrderFreight] [money] NULL,
[ProductID] [int] NULL,
CONSTRAINT [PK_OrdersTemp] PRIMARY KEY CLUSTERED ([OrderID] ASC,[OrderDate] ASC)ON FG2
) ON FG2
GO
USE [tempdb]
GO
CREATE TABLE [dbo].[Orders_Hist](
[OrderID] [int] NOT NULL,
[OrderDate] [datetime] NOT NULL,
[OrderFreight] [money] NULL,
[ProductID] [int] NULL,
CONSTRAINT [PK_OrdersTemp] PRIMARY KEY CLUSTERED ([OrderID] ASC,[OrderDate] ASC)
)
GO
USE [PartitionDB]
GO
ALTER TABLE [dbo].[Orders] SWITCH PARTITION 1 TO [dbo].[Orders_Temp]
GO
INSERT INTO [tempdb].[dbo].[Orders_Hist]
SELECT * FROM [dbo].[Orders_Temp]
GO
DROP TABLE [dbo].[Orders_Temp]
GO
```

استفاده از قابلیت پارتیشن بندی در آرشیو جداول بانک‌های اطلاعاتی SQL Server

```
SELECT * FROM [tempdb].[dbo].[Orders_Hist]
```

پس از اجرای کامل این دستورات، توزیع داده در بانک اطلاعاتی مثال مورد بررسی به شکل زیر است.

	OBJECT_NAME	partition_number	FILEGROUP_NAME	ROWS	total_pages	Compartition	VALUE
1	Orders	2	FG3	600	5	Less or equal than	2012-12-31 00:00:00.000
2	Orders	4	PRIMARY	0	0	Less or equal than	NULL
3	Orders	3	FG4	300	4	Less or equal than	2013-12-31 00:00:00.000
4	Orders	1	FG2	0	0	Less or equal than	2011-12-31 00:00:00.000

نظرات خوانندگان

نویسنده: محمد رضا موسائی
تاریخ: ۱۸:۲۳ ۱۳۹۳/۰۷/۲۵

بسیار امکان زیبایی است و بهتر اینکه اطلاعات قابل دسته بندی در فایل‌های `ndf` بوده و از همه مهمتر اینکه می‌توان از هر پارتبیشن به صورت جداگانه `backup` تهیه کرد.

نویسنده: مسعود خان
تاریخ: ۱۸:۵۲ ۱۳۹۳/۰۸/۱۳

سلام جناب رجبی مطلب بسیار خوبی بود مخصوصاً که من واقعاً بهش نیاز داشتم اما سوالاتی برام پیش اومده من دیتابیس بزرگی دارم که سه جدولش از همه بزرگته و یکیش ۱۹۰ میلیون و دو تا جدول ۴۳ میلیون رکوردی دارم و بقیه جداول زیر ۵ میلیون هست چون در شروع کار تجربه خوبی نداشتیم و از حجم اطلاعات مطمئن نبودم روی دیتابیس `pk` نذاشتیم حالا که می‌خواه `pk` و ایندکس گذاریش کنم برای هر جدول یک `filegroup` و در هر فایل گروپ یک فایل برای `pk` و یک فایل برای `index`ها گذاشتیم و تقریباً هر کدام از اون جداول بزرگ را به ۳۰ تا جدول تقسیم کردم تا سرعت پرس و جوهای زیادم بسیار کمتر بشه که به نظرم خوب نمی‌آمد (کار پرس و جو و ایجاد کوئری را دشوارتر می‌کنه) تا با راه حل شما اشنا شدم اما سوال مان اینجاست اگر داده هام رو بر اساس تاریخ در `datafile`ها پارتبیشن بندی کنم زمانی که بخواه مثلاً اطلاعات تا قبل از سال ۸۵ را آرشیو کنم تاثیری در زمان پرس و جوی من دارد و کلیدها و ایندکس‌ها وضعیتشون به چه شکل خواهد بود اگر از راه حل خودم استفاده نکنم چون همینجوریش حجم دیتابیس ۳۵ گیگ و با گذاشتی کلید و ایندکس چیزی بین ۱۵ تا ۲۰ گیگ هم اضافه می‌شه که خیلی بد هست. الان که یه خوردده رو کدها کار کردم یه سوال برام پیش اومد مثالی می‌زنم من جدولی دارم که تاریخ را بنام شرکت می‌گیره و یک کد گزارش میده حالا من این کد گزارش را با مثلاً ۱۰ هزار رکورد در جدول دوم ذخیره می‌کنم و باز با همون کد گزارش ۵۰۰۰ رکورد را در جدول سوم ذخیره می‌کنم و به همین ترتیب برای تمام روزها و برای کل شرکت‌ها این کد گزارش تولید و با حجم انبوهی از رکوردها در جداول ذخیره می‌شوند حالا سوال اینه که چطور بر اساس تاریخ که در جدول فقط اول هست جداول دیگر را پارتبیشن بندی کنم؟

نویسنده: محمد رجبی
تاریخ: ۱۳:۴۰ ۱۳۹۳/۰۸/۱۴

با سلام و احترام

پیشنهاد می‌کنم مطلب " [Download "Partitioned Table and Index Strategies Using SQL Server 2008" white paper](#)" را مطالعه فرمائید. به منظور پیاده سازی این قابلیت در بانک اطلاعاتی تان یک راه حل می‌تواند اینگونه باشد که یک DB جدید ایجاد نمایید که در آن تمامی زیرساخت‌ها ایجاد شود (ایندکس، `Partition Function` و ...) در ادامه به انتقال داده‌های از بانک عملیاتی به DB جدید پیروزی داشته باشید. برای مشاهده جزئیات به این [مطلب](#) مراجعه نمایید.

نویسنده: مسعود خان
تاریخ: ۲۰:۴۷ ۱۳۹۳/۰۸/۱۴

جناب رجبی ممنون بابت پاسختون من فایل را دارم می‌خونم که بخش هایش رو خودتون توضیح دادید اما با منطقی که جداولم دارم به مشکل خوردم. من کمی گشتم و با فشرده سازی به صورت `row` و `page` اشنا شدم که بسیار جالب بودو من ان را برای بزرگترین جدولم تست کردم و نتایج قابل توجهی داشتم برای `row` حجم جدول از ۱۷.۶ به ۹.۸ کاهش پیدا کرد برای `page` حجم جدول از ۱۷.۶ به ۴.۸ کاهش پیدا کرد که خیلی خوبه اما سوالی پیش می‌دارم این فشرده سازی چقدر در پرس و جوها و کوئری‌های پیچیده تاثیر داره اصلاً این تاثیر مثبت هست یا منفی؟ من هر روز از این جداول پرس و جو می‌کنم بنابراین می‌شود گفت خواندن از این جداول در حد متوسطی است نوشتن ندارد و ایا فشرده سازی بر روی پریمری کدها و ایندکس‌ها هم تاثیر دارد و اگر دارد مثبت است یا خیر؟ و تاثیرش بر روی پرس و جوها چقدر است؟

استفاده از قابلیت پارسیشن بندی در آرشیو جداول بانک‌های اطلاعاتی SQL Server

و سوال اخر به نظرتون با این پارسیشن بندی من pk و ایندکس‌ها را در فایل گروپی جدا ذخیره کنم یا در فایل primary به خصوص که بخواهم اطلاعات قبل از سال 85 را ارشیو کنم و هر ساله این ارشیو یک سال بیشتر می‌شود با تشکر

نویسنده: محمد رجبی
تاریخ: ۲۱:۴۴ ۱۳۹۳/۰۸/۱۵

با سلام، ابتدا از سوال آخر شروع می‌کنم، چنانچه تمایل دارید از امکان پارسیشنینگ در آرشیو بانک اطلاعاتی تان استفاده کنید همانگونه که در متن اشاره شده، باید جداول به شکل Aligned Index تعریف شوند. (پس نیاز به Filegroup می‌باشد) در مورد فشرده سازی به هیچ وجه هدف افزایش سرعت در پرس و جوها نیست بلکه کاهش حجم، هدف غائی است. همینطور هدف از ایجاد ایندکس در این است که هر رکورد در مکان صحیح خود به هنگام درج قرار گیرد و بدین ترتیب چنانچه مجموعه مرتب شده باشد، مرتبه اجرائی از $O(n \log n)$ به $O(1)$ کاهش می‌یابد و ... به همین خاطر است که در سیستم‌های OLTP توصیه شده Index هوشمندانه استفاده شود (بدلیل اینکه عملیات درج، بروزرسانی و حذف به کندی صورت نگیرد) و در سیستم‌های DSS بر عکس به دلیل ماهیت غیر عملیاتی بودن آنها استفاده فراوان از Index توصیه شده است.

بطور کلی ساختار ایندکس پایه در SQL Server عبارتند از: ایندکس‌های Clustered، ایندکس‌های Nonclustered در Heap و ایندکس‌های Nonclustered در یک ایندکس‌های Clustered شده؛ روش ذخیره فیزیکی داده بین این ایندکس‌ها متفاوت است و همچنین روشهای SQL Server برای پیمایش ایندکس در B-Tree استفاده می‌کند بسته به این سه نوع متفاوت خواهد بود.

نویسنده: مسعود خان
تاریخ: ۱۰:۷ ۱۳۹۳/۰۸/۱۶

ممnon از پاسختون من دیشب بالاخره تونستم راهی برای پیاده سازی این روشهای دیتابیس پیدا و اجرا کنم و نتیجه بسیار رضایت بخش بود تقریبا برای پرس و جو هایی که شامل aligned index میشند تقریبا 70 بار سریعتر بود اما حجم فایل‌های .ndf روی هم رفته تقریبا 4 گیگ میشه در صورتی که خود جدول 1.48 گیگ هست

فعلا دارم روی فشرده سازی و تعریف ایندکس روی بعضی فیلد‌ها کار می‌کنم اما نمی‌دونم برای ستون int مثلا برای تعداد کدام ایندکس را پیاده سازی کنم بهتر است و تعریف این ایندکس‌ها مثل تعریف pkها که در فایل‌ها پخش میشون ب چه صورت هست فعلای دارم روشن کار می‌کنم اما ممنون میشم در این زمینه هم راهنماییم کنید با تشکر

نویسنده: مسعود خان
تاریخ: ۱۲:۵۵ ۱۳۹۳/۰۸/۱۶

جناب رجبی یه سوال در رابطه با pk دارم اونم این هست که پریمری کد را روی دو فیلد گذاشتید که کلاستر هست حالا سوال این هست که گذاشتن دو فیلد به عنوان یک پریمری کد چقدر در حجم و کارایی تاثیر دارد ؟ اگه سه تا بشه چطور ؟

CONSTRAINT PK_Orders PRIMARY KEY CLUSTERED (OrderID ASC, OrderDate ASC)

و همینطور میشه مثالی بزنید که بشه ایندکس را هم به صورت پریمری به جدول اضافه کرد که ایندکس‌ها در فایل‌ها تقسیم بشن دقیقا مثل pk بالا؟

نویسنده: محمد رجبی
تاریخ: ۸:۴۷ ۱۳۹۳/۰۸/۱۷

در مثال مذکور از Primary Key Constraint در زمان تعریف یک Partition Key در منظور داشتن ساختار Aligned Index استفاده نموده ایم.

در هنگام ایجاد یک Primary Key Constraint بطور خودکار یک Unique Clustered Index نیز روی ستون (های) شرکت یافته در Table براساس این فیلد (ها) به شکل Sort شده نگهداری می‌شود، ضمن اینکه Primary Key تعريف می‌شود و بدین ترتیب

هر Table می‌تواند، شامل 1 عدد Clustered Index و 249 عدد Nonclustered Index باشد.

نویسنده: پروانه
تاریخ: ۱۶:۲۷ ۱۳۹۳/۰۸/۱۸

با سلام ،

پایگاه داده‌های سیستم‌های VOIP که اطلاعات تماس در لحظه در آن ذخیره می‌شوند، پیوسته در حال رشد است. نیاز است که یک سری اطلاعات آن آرشیو شود.

به غیر از پارتبیشن بندی، چه روش‌های ساده‌تری وجود دارد برای اینکار. به نظر می‌آید پارتبیشن بندی در زمان‌هایی که می‌خواهیم یک پارتبیشن به پارتبیشن دیگر ادغام شود کند است. سوال دیگر هم این است که در صورت استفاده از پارتبیشن بندی، آیا می‌توان ماه به ماه اطلاعات در پارتبیشن دیگر ادغام شود یا این کار سالانه باید انجام گیرد؟ ما لازم داریم اطلاعات فقط یک سال در یک پارتبیشن قرار گیرد،

نویسنده: محمد رجبی
تاریخ: ۱۲:۴۵ ۱۳۹۳/۰۸/۱۹

با سلام - همانطور که می‌دانیم؛ هدف فرآیند آرشیو حذف گروهی از داده‌ها از بانک اطلاعاتی می‌باشد. این داده‌ها دیگر مورد نیاز در سیستم عملیاتی نمی‌باشند ولی می‌خواهیم آنها را حفظ نمائیم. تکنیک‌های متفاوتی برای این منظور وجود داشته است از قبیل: - Application partitioning: به طور کلی جهت دستیابی به اطلاعات (از محیط عملیاتی و یا محیط‌های گزارشی گیری با ماهیت آرشیو) از درون خود برنامه کاربردی با ماشین مورد نظر ارتباط برقرار می‌گردد.

- Partition Views (DPVs): جداول یکسانی ایجاد می‌گردید، که هر یک حاوی قسمتی از اطلاعات بودند و بدین ترتیب برای دستیابی به تمامی داده‌های این جداول با ایجاد یک View (پیوند میان این جداول) کلیه اطلاعات قابل دسترسی بود. در تکنیک پارتبیشنینگ کلیه محتوای جدول بصورت یکجا قابل دستیابی است، همچنین جهت آرشیو با استفاده از عملگر Switch این امکان را داریم که در کسری از ثانیه (با تنها انتقال data meta) به آرشیو اطلاعات بپردازیم. تا پیش از انتشار نسخه 2012 توان پشتیبانی از 1.000 پارتبیشن وجود داشت این امکان در نسخه 2014 به 15.000 پارتبیشن رسیده است.

همانگونه که ذکر گردید جهت آرشیو از عملگر Merge (و نه عملگر Switch) همچنین در خصوص نحوه انجام پارتبیشن بندی از آنجا که تابع یک Data type می‌گیرد الزامی به نوع Date نیست و مطابق شکل می‌تواند نوع آن Int نیز باشد، ولی عملیات آرشیو در ذات خود به اطلاعات تاریخچه ای اشاره می‌کند. جهت اطلاعات بیشتر به لینک [Partitioned Tables and Indexes](#) مراجعه شود.

نویسنده: مسعود خان
تاریخ: ۱۹:۳۶ ۱۳۹۳/۰۸/۲۱

جناب رجبی یک سوال سیستم ارشیو کردن در پارتبیشن بندی به چه شکل هست و اینکه اگر بخشی از پارتبیشن‌ها مثلاً قبل از سال 2012 را به ارشیو ببریم در سرعت پرس و جوها تاثیر دارد و اینکه بعد از ارشیو کردن میشه از ارشیو در اورد او پارتبیشن‌ها را یا نه؟

نویسنده: محمد رجبی
تاریخ: ۱۲:۳۰ ۱۳۹۳/۰۸/۲۵

از آنجایی که هر Data File شامل یک مسیر فیزیکی است (در یک Filegroup قرار می‌گیرد)، بنابراین این امکان وجود دارد که محتوای یک جدول حجیم را به چندین قسمت در سطح کنترل تقسیم نمود، اگر سوالتان در خصوص اینکه «پارتبیشن‌ها» در متن ذکر اینکه آرشیو شدند، می‌توان از وضعیت آرشیو خارج نمود» و منظور Read Only نمودن پارتبیشن‌ها است همانگونه که در متن ذکر شده "چک باکس Read Only ... چنانچه روی یک Filegroup تنظیم گردد، عملیات مربوط به Write روی داده‌های آن قابل انجام

استفاده از قابلیت پارتبیشن بندی در آرشیو جداول بانک‌های اطلاعاتی SQL Server

نیست و برای Filegroup هایی که جنبه نگهداری آرشیو را دارند، قابل استفاده است. و چنانچه Filegroup ای را از حالت Read Only دوباره خارج کنیم، می‌توان عملیات Write را دوباره برای آن انجام داد. "در این حالت به منظور بالا بردن Performance می‌توانید از RAID 0 یا RAID 1 جهت بهره مندی از Fast read استفاده کنید.

نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۸/۲۵ ۱۲:۳۹

می‌توانید برای جداولی که حاوی داده می‌باشند نیز از امکان Partitioning استفاده نمایید. برای جدول Orders سناریوی مورد بررسی، بدون استفاده از PartitionSchema به شکل زیر ایجاد می‌شود:

```
CREATE TABLE Orders
(
    OrderID INT IDENTITY(1,1) NOT NULL,
    OrderDate DATETIME NOT NULL,
    OrderFreight MONEY NULL,
    ProductID INT NULL,
    CONSTRAINT PK_Orders PRIMARY KEY CLUSTERED (OrderID ASC, OrderDate ASC)
    ON [PRIMARY]
) ON [PRIMARY]
GO
```

می‌توانید به شکل زیر محتوای جدول که در فایل گروپ PRIMARY قرار گرفته را در سایر پارتبیشن‌ها قرار داد.

```
ALTER TABLE Orders DROP CONSTRAINT PK_Orders
WITH (MOVE TO psOrderDateRange (OrderDate))
Go
```

نویسنده: مهدی آفازاده
تاریخ: ۱۳۹۳/۱۲/۰۳ ۱۱:۵

سلام

آیا فقط تنها در نسخه Sql Server Enterprise Edition می‌توان از پارتبیشن استفاده کرد ؟

نویسنده: وحید فرهمندیان
تاریخ: ۱۳۹۳/۱۲/۰۴ ۱۷:۵۷

جهت مشاهده امکانات موجود در نسخ مختلف SQL Server به [اینجا](#) مراجعه کنید.

مقدمه در لینکی که چندی پیش به اشتراک گذاشته بودم؛ به مطلبی تحت این عنوان اشاره شده بود: ["آبا از KPI باید به انباره داده و هوش تجاری رسید؟"](#) (بر گرفته از وبلاگ آقای جام سحر) که در آن به موانع پیش روی انجام پروژه‌های BI در ایران پرداخته شده است.

این مقاله بر گرفته از فصل سوم یکی از White Paper های ماکروسافت با عنوان [Microsoft EDW Architecture, Guidance and Deployment Best Practices](#) می‌باشد. که به شرح عملیات Loading در فاز ETL می‌پردازد. از آنجا که به منظور پیاده سازی این نوع پروژه‌ها معمولاً در ایران برونو سپاری صورت می‌گیرد و مدیران شرکت‌ها بیشتر درگیر سیستم‌های OLTP هستند و مجری پروژه (شرکت پیمانکار) معمولاً کوتاه‌ترین مسیر انتخاب می‌کند (و امروزه نیک میدانیم که "انتخاب مسیرهای کوتاه در زمان کم می‌تواند به پیچیدگی‌های بسیار جدی در دراز مدت منجر شود!") و همچنین از آنجا که متاسفانه به دلیل عدم ثبات مدیریت در ایران معمولاً مدیریت برای تحويل پروژه تحت فشار است و نه برای مسائل پشتیبانی " و مسائل دیگری از این دست؛ چنانچه در تحويل گیری محصول به درستی تست نرم افزار صورت نگیرد، در نظر گرفتن موارد زیر:

Verification: Are we building the product right? ~ Software correctly implements a specific function

Validation: Are we building the right product? ~ Software is traceable to customer requirements

پروژه با شکست مواجه می‌شود و انتظارات مدیران بهره بردار را برآورده نمی‌کند. به هر روی در این مقاله به ترجمه مطالب زیر پرداخته می‌شود، توصیه می‌کنم در صورتی که با خواندن متن انگلیسی مشکلی ندارید، اصل مقاله مذکور خوانده شود.

Full Load vs Incremental Load 1-

Detecting Net Changes 2-

Pulling Net Changes – Last Change Column 2-1-

Pulling Net Changes – No Last Change Column 2-2-

Pushing Net Changes 2-3-

ETL Patterns 3-

Destination load Patterns 3-1-

Versioned Insert Pattern 3-2-

Update Pattern 3-3-

Versioned Insert: Net Changes 3-4-

Data Integration Best Practices 4-

Basic Data Flow Patterns 4-1-

Update Pattern 4-1-1-

Update Pattern – ETL Framework 4-1-2-

Versioned Insert Pattern 4-1-3-

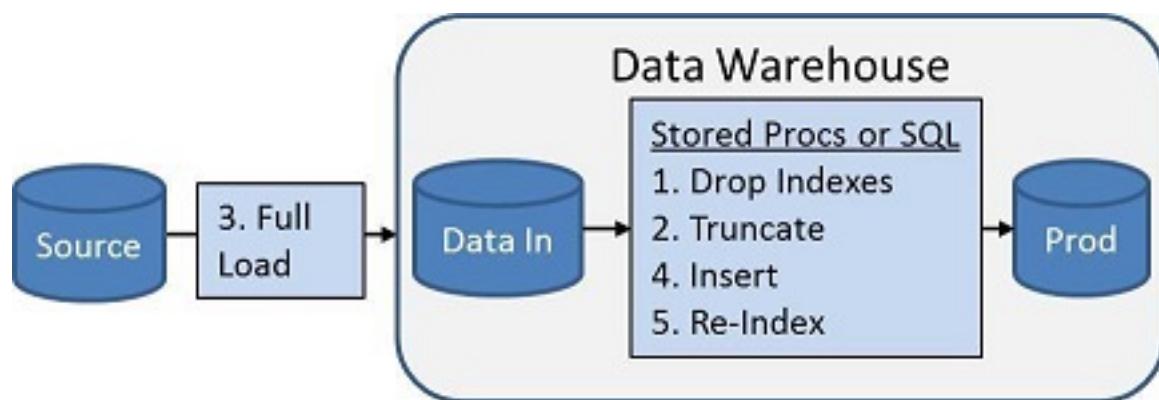
Update vs. Versioned Insert 4-1-4-

Dimension Patterns 4-2-

Fact Table Patterns 4-3-

Managing Inferred Members 4-3-1-

این طریق که هر بار عملیات بارگذاری صورت می‌گرفت، DW از نو دوباره ساخته می‌شد. شکل زیر مراحل مختلف انجام شده در این روش را نمایش می‌دهد:



پروسه Full Load شامل مراحل زیر بود:

از آنجا که Index ها زمان بارگذاری را افزایش می دادند، این عمل صورت می پذیرفت.

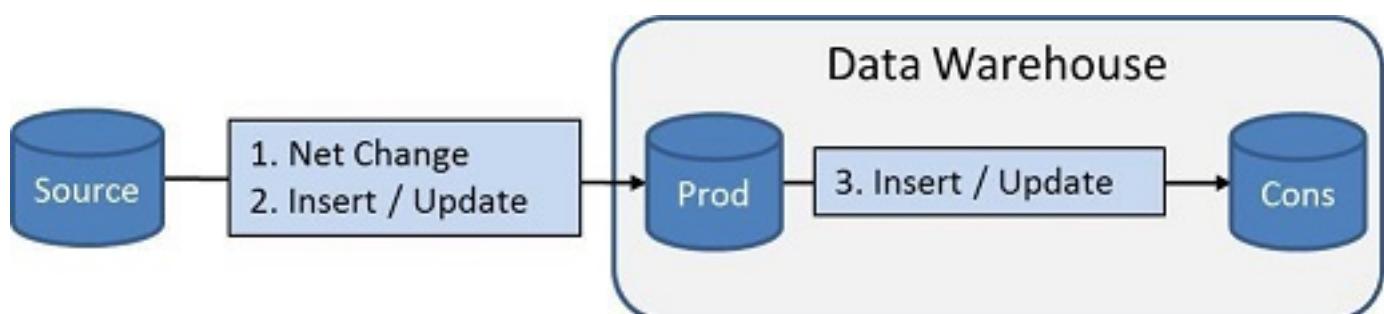
تمامی رکوردهای موجود در جداول حذف می شدند.

Bulk Copy

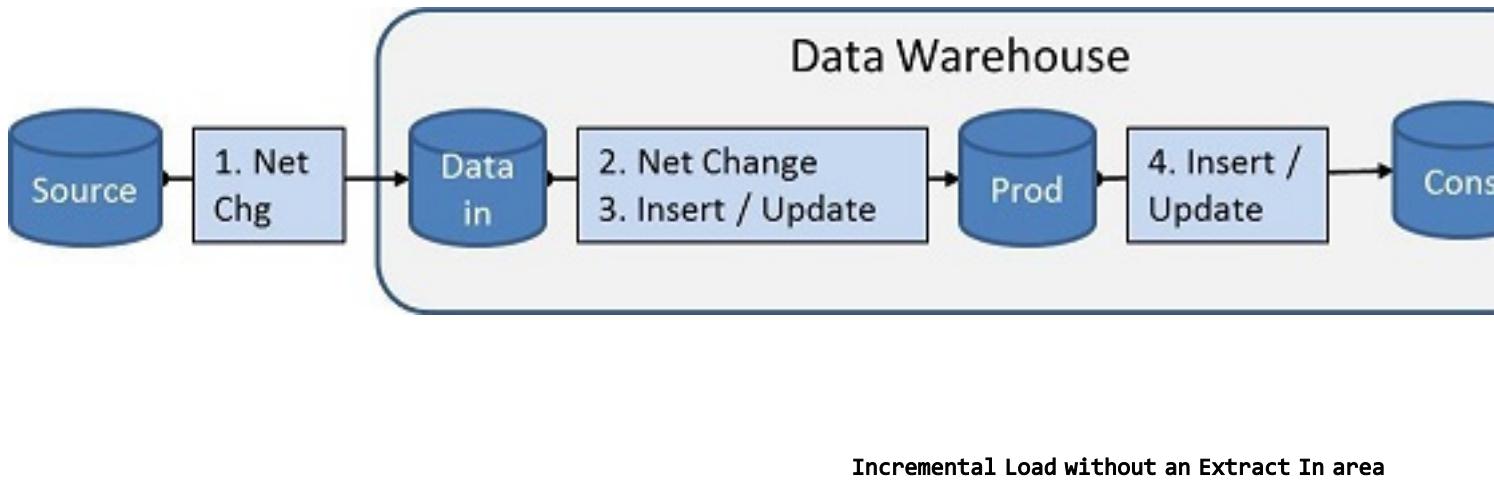
Load Data

شامل عملیاتی نظیر شاخص گذاری روی داده هایی است که اخیراً بارگذاری شده اند و....

روی هم رفته Full Load مسئله ای مشکل ساز بود، زیرا نیاز به زمانی برای بارگذاری مجدد داده ها داشت و مسئله ای نداشت امنیت دستیابی به گزارشاتی تاریخچه ای با ماهیت زمان برای مشتریان کسب و کار بود. به این دلیل که همواره یک کپی از آخرین داده های موجود در سیستم عملیاتی درون DW قرار می گرفت؛ که با بکارگیری Full Load اغلب قادر به ارائه ای این نوع از گزارشات نبودیم، بدین ترتیب سازمان ها به نسل دوم روی آورند که در این دیدگاه از مفهوم Incremental Load استفاده می شود. اشکال زیر مراحلی که در این روش انجام می شود را نمایان می سازد:



Incremental Load with an Extract In area



مراحل Incremental Load شامل:

بارگذاری تغییرات نسبت به آخرین فرآیند بارگذاری انجام شده

درج / بروزرسانی تغییرات درون Production area

درج / بروزرسانی تغییرات درون Consumption area نسبت به

تفاوت های اصلی میان Incremental Load و Full Load در این است که در Incremental Load

نیازی به پردازش های اضافی جهت حذف شاخص ها، پاک کردن تمامی رکوردهای جداول و ساخت مجدد شاخص ها نیست.

البته نیاز به رویه ای جهت شناسایی تغییرات می باشد.

و همچنین نیاز به بروزرسانی بعلاوه درج رکوردهای جدید نیز می باشد.

ترکیب این عوامل برای ساخت Incremental Load کارآمد تر، منجر به پیچیده تر شدن پیاده سازی و نگهداری آن نیز می شود.

Detecting Net Changes 2-

فرآیند لود افزایشی ETL، بایست قادر به شناسائی رکوردهای تغییر یافته در مبداء باشد، که این عمل با استفاده از هر یک از تکنیک های Push یا Pull انجام می شود.

در تکنیک Pull، فرآیند ETL رکوردهای تغییر یافته در مبداء را انتخاب می کند:

ایده آآل وجود داشتن یک ستون Last Changed در سیستم مبداء است؛ که از آن می توان جهت انتخاب رکوردهای تغییر یافته استفاده نمود.

چنانچه ستون Last Changed وجود نداشته باشد، تمامی رکوردهای مبداء باید با رکوردهای مقصد مقایسه شود.

در تکنیک Push، مبداء تغییرات را شناسائی می کند و آنها را به سمت مقصد Push می کند؛ این درخواست می تواند توسط فرآیند ETL انجام شود.

از آنجایی که پردازش ETL معمولاً در زمان هایی که Peak کاری وجود ندارد، اجرا می شود، استفاده از مکانیسم Pull برای شناسایی تغییرات نسبت به مکانیزم Push ارجحیت دارد.

Pulling Net Changes – Last Change Column 2-1- بیشتر جداول در سیستم های مبداء حاوی ستون هایی هستند که زمان ایجاد و یا اصلاح رکوردها را ثبت می کنند. در نوع دیگری از سیستم های مبداء ستونی با مقدار عددی وجود دارد، که هر زمان رکوردي تغییر یافت به آن ستون مقداری اضافه می شود. هر دوی این تکنیک ها به فرآیند ETL اجازه می دهند، بطور کارآمدی رکوردهای تغییر یافته را انتخاب کند. (با مقایسه، بیشترین مقدار قرار گرفته در آن ستون؛ که در طول آخرین اجرای فرآیند ETL بدست آمده است).

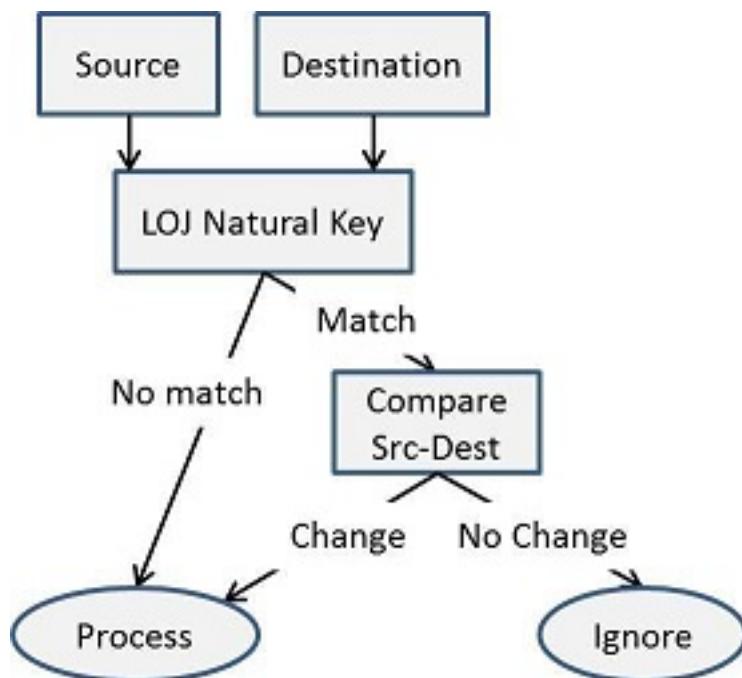
نمونه ای از جداول سیستم مبداء که دارای تغییرات زمانی است در شکل زیر نمایش داده می شود.

Source	WHERE ISNULL(ModifiedOn, CreatedOn) > LastMaxDate					
Natural Key	Attributes	Dates	Numbers	CreatedOn	ModifiedOn	

همچنین شکل زیر نشان می دهد، چگونه یک مقدار عددی می تواند به منظور انتخاب رکوردهای تغییریافته استفاده شود.

Staging	WHERE Lineageld > LastMaxLineageld					
Surrogate Key	Natural Key	Attributes	Dates	Numbers	Lineage Id	

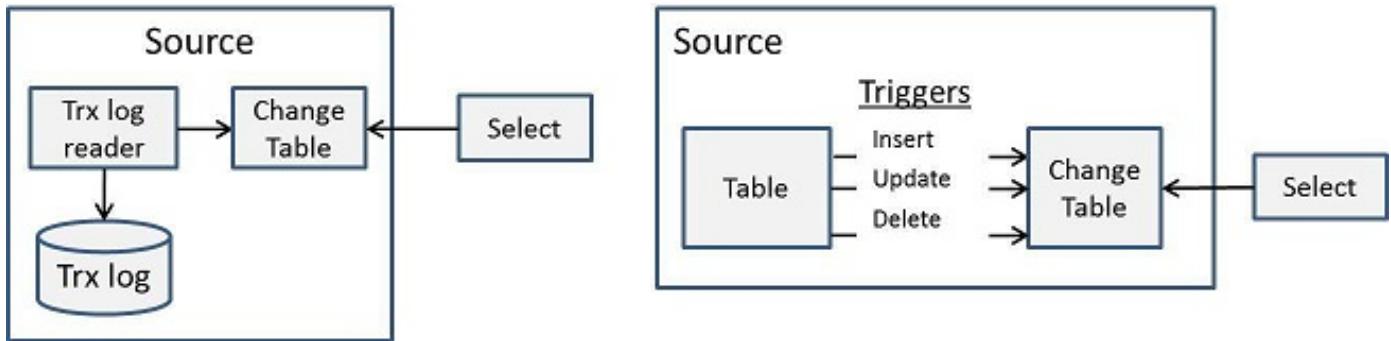
شکل زیر گردش فرآیند را هنگامی که ستون Last Change وجود ندارد: **Pulling Net Changes – No Last Change Column 2-2-** نمایش می دهد.



این گردش فرآیند شامل:

Join میان مبداء و مقصد با استفاده از یک دستور Left Outer Join است. تمامی رکوردهای مبداء که در مقصد وجود ندارند، پردازش می شوند. زمانی که رکوردی در مقصد وجود داشته باشد مقادیر داده های مبداء و مقصد مقایسه می شوند. تمامی رکوردهای مبداء که تغییر یافته اند پردازش می شوند. از آنجایی که تمامی رکوردها پردازش می شوند، این روش بویژه برای جداول حجمی: روش کارآمدی نیست.

دو متد متداول Push وجود دارد که در تصویر زیر نمایش داده شده است.



تفاوت این دو روش به شرح زیر است:

در سناریو اول (شکل سمت چپ): بانک اطلاعاتی رابطه ای سیستم مبداء Transaction Log را مرتب مانیتور می کند تا تغییرات را شناسائی کرده و در ادامه تمامی این تغییرات را در جدولی در مقصد درج می کند.

در سناریو دوم: توسعه دهندهای Trigger هایی ایجاد می کنند تا هر زمان که رکوردی تغییر یافتد، تغییرات در جدولی که در مقصد وجود دارد درج گردد.

مسئله ای که در هر دو مورد وجود دارد Load Performance اضافه ای است؛ که روی سیستم مبداء وجود دارد و می تواند سیستم های OLTP را تحت تاثیر قرار دهد. به هر روی سناریو نخست معمولاً کاراتر از سناریویی است که از Trigger استفاده می کند.

ETL Patterns 3-

پس از شناسائی رکوردهایی که در مبداء تغییر یافته اند، نیاز داریم تا این تغییرات در مقصد اعمال شود. در این قسمت به معرفی الگوهایی که برای اعمال این تغییرات وجود دارد می پردازیم.

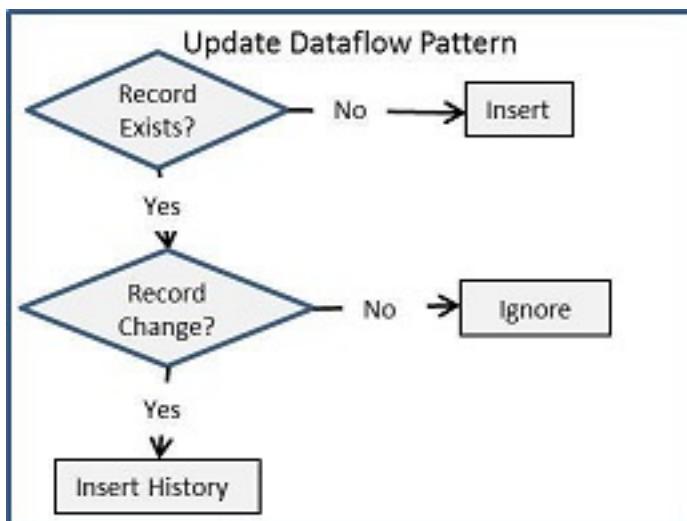
Destination load Patterns 3-1-

تشخیص چگونگی اضافه نمودن تغییرات در مقصد تابع دو عامل زیر است:

آیا رکورد هم اینک در مقصد وجود دارد؟

الگوی استفاده شده برای جدول مقصد به کدام شکل است؟ (Versioned Insert Update) یا

فلوچارت زیر نشان می دهد، به چه شکل جداول مقصد متاثر از چگونگی پردازش رکوردهای مبداء قرار دارند. توجه داشته باشید که عمل بررسی بطور جداگانه و در یک لحظه صورت می گیرد.



Versioned Insert Pattern 3-2-

نمونه ای از الگوی Versioned Insert Kimball Type II Slowly Changing Dimension موجودیت دارای ورژن های متعددی است. مطابق تصویر زیر؛ این الگو به ستون های اضافه ای نیاز دارد که وضعیت نمونه ای از یک رکورد را نمایش دهد.

Surrogate Key	Natural Key	Data...	Start Date	End Date	Record Status	Version #
---------------	-------------	---------	------------	----------	---------------	-----------

این ستون ها به شرح زیر هستند:

Start Date: زمانی که وضعیت آن نمونه از رکورد فعال می شود.

End Date: زمانی که وضعیت آن نمونه از رکورد غیر فعال می شود.

Record Status: وضعیت های یک رکورد را نشان می دهد، که حداقل به شکل Active یا Inactive است.

Version #: این ستون که اختیاری می باشد، ورژن آن نمونه از رکورد را ثبت می کند.

برای مثال شکل زیر؛ بیانگر وضعیت اولیه رکوردي در این الگو است:

Surrogate Key	Natural Key	Data...	Start Date	End Date	Status	Ver #
100	AB549		2001-02-05	NULL	A	1

فرض کنید که این رکورد در تاریخ 2 March 2010 در سیستم مبداء تغییر می کند. فرآیند ETL این تغییر را شناسائی می کند و همانند تصویر زیر؛ به شکل نمونه ای ثانویه از این رکورد، اقدام به درج آن می کند.

Surrogate Key	Natural Key	Data...	Start Date	End Date	Status	Ver #
100	AB549		2001-02-05	2010-03-02	I	1
537	AB549		2010-03-02	NULL	A	2

توجه داشته باشید زمانی که رکورد دوم در جدول درج می شود، به منظور بازتاب این تغییر؛ رکورد اول به شکل زیر بروزرسانی می گردد:

End Date: تا این زمان وضعیت این رکورد فعال بوده است.
Record Status: که به Active تغییر پیدا می کند.

در برخی از پیاده سازی های DW عمده از الگوی Versioned Insert استفاده می شود و هرگز از الگوی Update استفاده نمی شود. مزیت این استراتژی در این است که تمامی تاریخچه تغییرات ردیابی و ثبت می شود. به هر روی غالباً هزینه ثبت کردن این تغییرات منجر به ایجاد نسخه های زیادی از تغییرات می شود. تیم DW برای مواردی که تغییرات متأثر از گزارشات تاریخچه ای نیستند، می توانند الگوی Update را در نظر گیرند.

Update Pattern 3-3-

الگوی Update روی رکورد موجود، تغییرات سیستم مبداء را بروزرسانی می کند. مزیت این روش در این است که همواره یک رکورد وجود دارد و در نتیجه باعث ایجاد Query های کارآمدتر می شود. تصویر زیر بیانگر ستون هایی است که برای پشتیبانی از الگوی Update بایست ایجاد کرد.

Surrogate Key	Natural Key	Data...	Record Status	Version #
---------------	-------------	---------	---------------	-----------

این ستون ها به شرح زیر هستند:

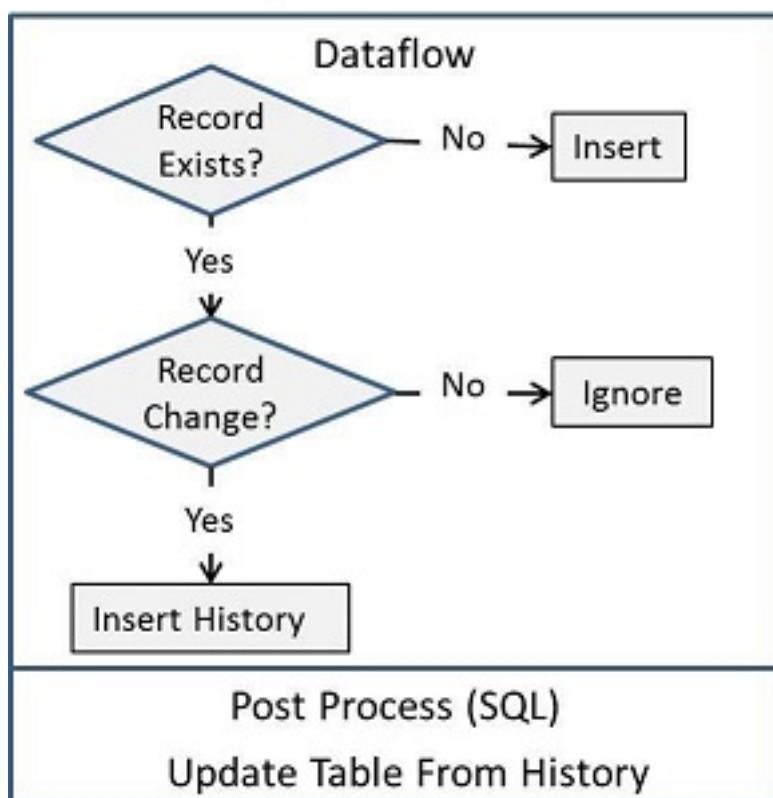
Record Status: وضعیت های یک رکورد را نشان می دهد که حداقل به شکل Active یا Inactive است.
Version #: این ستون که اختیاری می باشد، ورژن آن نمونه از رکورد را ثبت می کند.

موارد اصلی الگوی Update عبارتند از:

تاریخ ثبت نمی شود. ابزاری ارزشمند برای نظارت بر داده ها، تغییرات تاریخی است و زمانی که ممیزی داده رخ می دهد؛ می تواند مفید واقع شود.
بروزرسانی ها یک الگوی مبتنی بر مجموعه هستند. استفاده از بروزرسانی هر بار یک رکورد در ابزار ETL خیلی کارآمد (موجه) نیست.

یک روش دیگر برای در نظر گرفتن موارد فوق؛ اضافه کردن یک جدول برای درج ورژن ها به الگوی Update است که در شکل زیر نشان داده شده است.

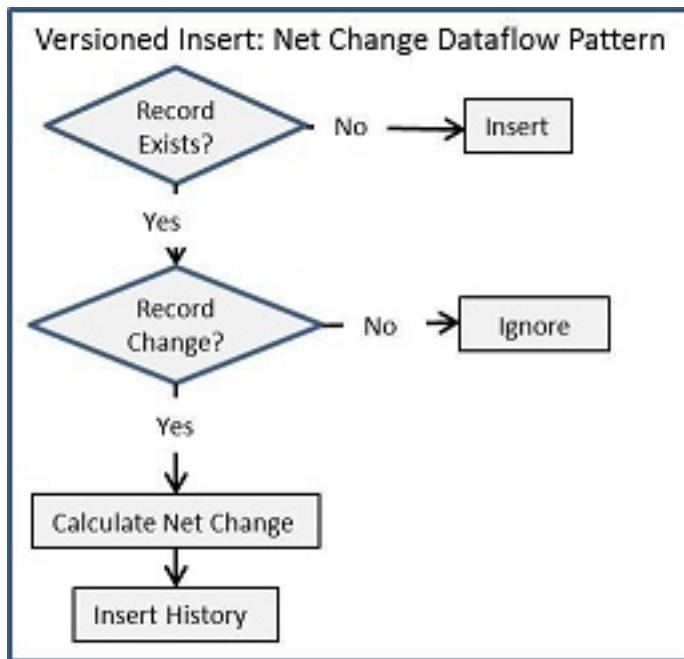
Update Pattern



اضافه نمودن یک جدول تاریخچه، که تمامی تغییرات سیستم مبداء را ثبت می‌کند؛ نظارت و ممیزی داده‌ها را نیز فراهم می‌کند و همچنین بروزرسانی‌های کارآمد مبتنی بر مجموعه را برای جداول DW به ارمنان می‌آورد.

Versioned Insert: Net Changes 3-4-

این الگو غالباً در جداول حجیم Fact که بروزرسانی آنها پر هزینه است استفاده می‌شود. شکل زیر منطق استفاده شده در این الگو را نشان می‌دهد.



توجه داشته باشید در این الگو:

مقادیر مالی و عددی محاسبه شده؛ به عنوان یک Net Change از نمونه قبلی رکورد در جدول Fact ذخیره می‌شود. هیچ گونه فعالیت Post Processing صورت نمی‌گیرد (از قبیل بروزرسانی جداول Fact پس از کامل شدن Data Flow). هدف استفاده از این الگو اجتناب از بروزرسانی روی جداول بسیار حجمی می‌باشد. عدم بروزرسانی و همچنین اندازه جدول Fact زمینه ای را فراهم می‌کند که منطق شناسائی رکوردهای تغییریافته پیچیده تر می‌شود. این پیچیدگی از آنجا ناشی می‌شود که نیاز به مقایسه رکوردهای جدول Fact آتی با جدول Fact موجود می‌باشد.

Data Integration Best Practices 4-

هم اکنون پس از آشنایی با مفاهیم و الگوهای توزیع دادهها به ارائه تعدادی نمونه می‌پردازیم؛ که بتوان این ایده‌ها و الگوها را در عمل پوشش داد.

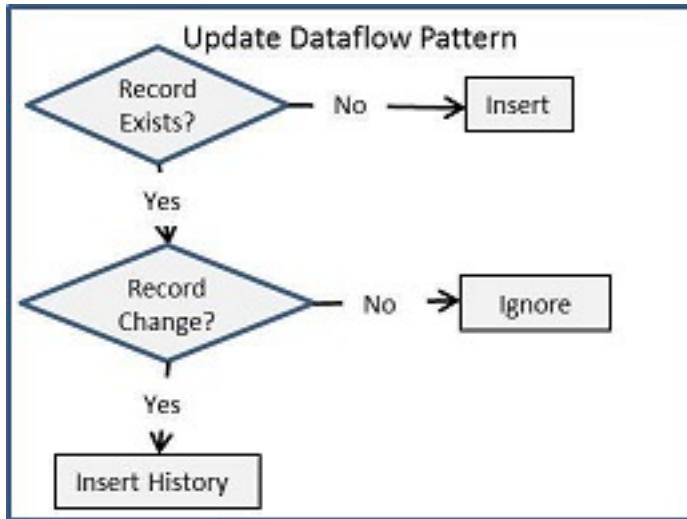
Basic Data Flow Patterns 4-1-

هر یک از الگوهای Update Pattern و Versioned Insert Pattern می‌توانند برای انواعی از جداول بکار روند که معروفترین آن‌ها توسط Kimball ساخته شده اند.

Update Pattern: از Slowly Changing Dimension Type I (SCD I) استفاده می‌کند.
 Versioned Insert Pattern: از Slowly Changing Dimension Type II (SCD II) استفاده می‌کند.
 Fact Table: نوع الگویی که استفاده می‌کند به نوع جدول Fact ای که Load خواهد شد بستگی دارد.

Update Pattern 4-1-1-

مطابق تصویر زیر جدولی که تنها حاوی ورژن فعلی رکورد هاست؛ از Update Dataflow Pattern استفاده می‌کند.



مواردی که در مورد این گردش کاری باید در نظر داشت به شرح زیر است:

این Data Flow فقط سطرهایی را به یک مقصد اضافه خواهد کرد. SSIS دارای گزینه "Table or view fast load" می‌باشد که بارگذاری‌های انبو و سریع را پشتیبانی می‌کند.

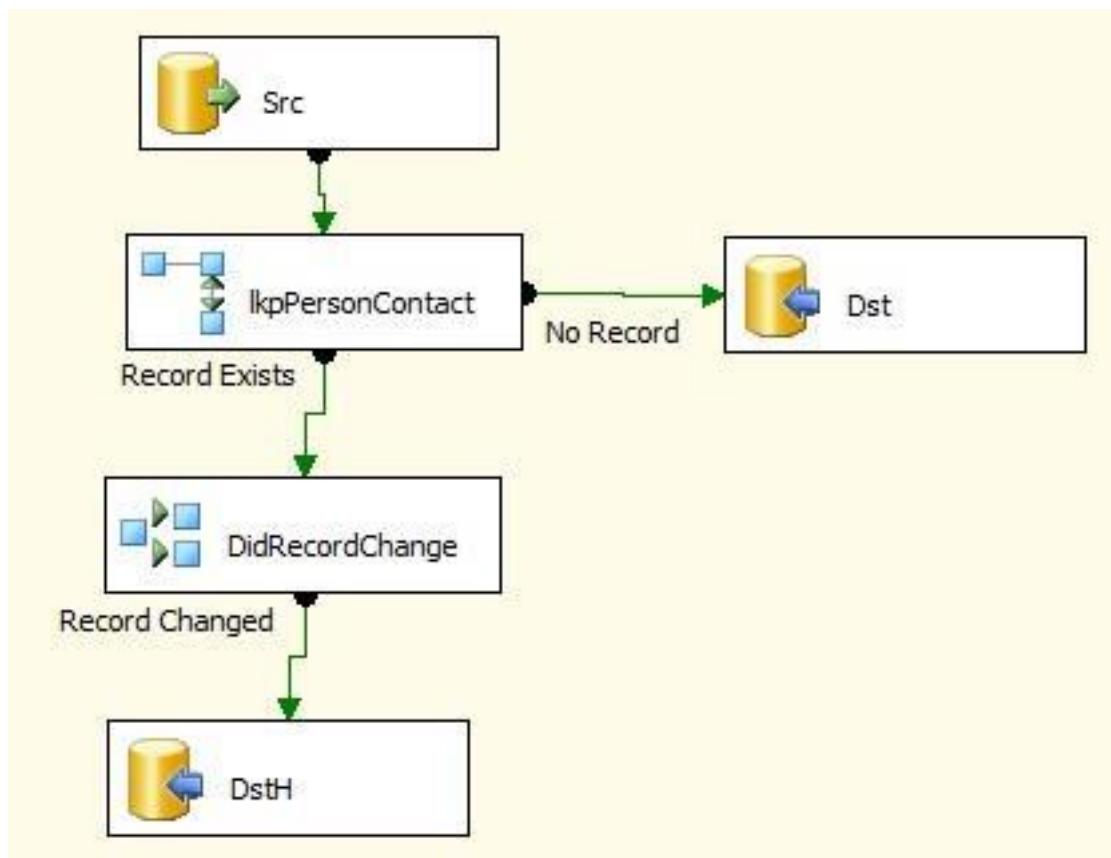
درین یک Data Flow بروزرسانی رکوردها را می‌توان با استفاده از تبدیل OLE DB Command انجام داد. توجه داشته باشید خروجی‌های این تبدیل در یک دستور Update به ازای هر رکورد بکار می‌رود؛ مفهوم بروزرسانی انبو در این Data Flow وجود ندارد. بدین ترتیب الگوی فعلی ارائه شده؛ تنها رکوردها را درج می‌کند و هرگز در این Data Flow رکوردها Update نمی‌شوند.

هر جدول دارای یک جدول تاریخچه است که برای ذخیره همه فعالیت‌های مرتبط با آن بکار می‌رود. یک رکورد در جدول تاریخچه زمانی درج خواهد شد؛ که رکورد مبداء در مقصد وجود داشته باشد ولی دارای مقداری متفاوت باشد.

راه دیگر فرستادن تغییرات رکوردها به یک جدول کاری است که پس از پایان یافتن فرآیند Update ، خالی (Truncate) می‌شود. مزیت نگهداری تمامی رکوردها در یک جدول تاریخچه؛ ایجاد یک دنباله ممیزی است که می‌تواند برای نظارت بر داده‌ها به منظور نمایان ساختن موارد مطرح شده توسط مصرف کننده‌های کسب و کار استفاده شود.

گزینه‌های متفاوتی برای تشخیص تغییرات رکوردها وجود دارد که در ادامه به شرح آنها می‌پردازیم.

شكل زیر نمایش دهنده چگونگی پیاده سازی Update Dataflow Pattern در یک SSIS می‌باشد:



این SSIS شامل عناصر زیر است:

:Destination table lookup

به منظور تشخیص اینکه رکورد در جدول مقصد وجود دارد از "LkpPersonContact" استفاده می کنیم.

:Change detection logic

با استفاده از "DidRecordChange" مبداء و مقصد مقایسه می شوند. اگر تفاوتی بین مبداء و مقصد وجود نداشت؛ رکورد نادیده گرفته می شود. چنانچه بین مبداء و مقصد تفاوت وجود داشت؛ رکورد در جدول تاریخچه درج خواهد شد.

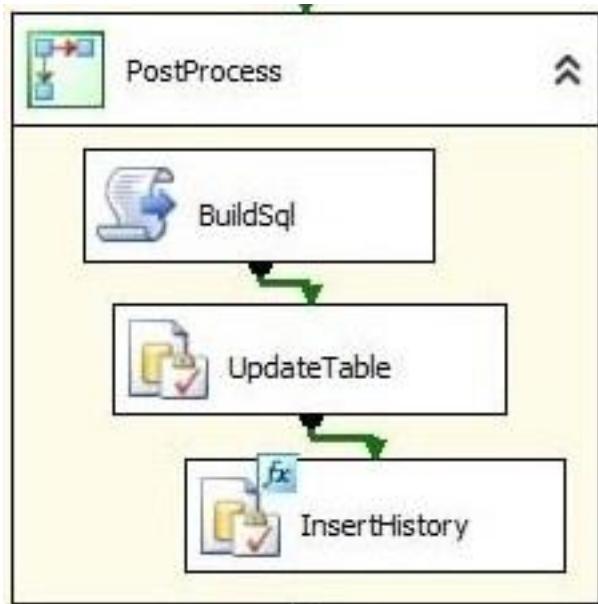
:Detection Inserts

رکوردها در جدول مقصد درج خواهند شد در صورتیکه در آن وجود نداشته باشند.

:Destination History Inserts

رکوردها در جدول تاریخچه مقصد درج خواهند شد، در صورتیکه (در مقصد) وجود داشته باشند.

پس از اتمام Data Flow یک روال Post-processing مسئولیت بروزرسانی رکوردهای جدول اصلی و رکوردهای ذخیره شده در جدول تاریخچه را بر عهده دارد که می تواند مطابق تصویر زیر با استفاده از یک Execute Process Task پیاده سازی شود.



ETL Framework Pattern: Set based post processing for Update and Versioned Insert patterns. Variables:

- Type: 1 for Update, 2 for Versioned Insert pattern
- xfrUpdateKeyColumns: Natural key(s), comma separated
- xfrUpdateColumns: update column list, comma separated

BuildSql: Generate the Insert and Update SQL

UpdateTable: Update statement for both patterns

InsertFirstUpdateHistory: Insert initial record into history

مسئولیت اجرای تمامی فعالیت های زیر را در این الگو بر عهده دارد که شامل:

بروزرسانی رکوردهای جداول با استفاده از رکوردهای درج شده در جدول تاریخچه.

درج تمامی رکوردهای جدید (نسخه اولیه و در درون جدول تاریخچه). کلید اصلی جداولی که ستون آنها IDENTITY است مقدار نامشخصی دارد؛ تا زمانی که درج صورت گیرد، این به معنای آن است که پیش از انتقال آنها به جدول تاریخچه نیاز است منتظر درج شدن آنها باشیم.

Update Pattern – ETL Framework 4-1-2-

تصویر زیر بیانگر انجام این عملیات با استفاده از ابزارهای ETL است.

در نگاه نخستین ممکن است Data Flow از نوع اصلی خود پیچیده تر به نظر آید؛ که در واقع این گونه نیز هست، زیرا در فاز توسعه بیشتر Framework ها جهت پیاده سازی به یک زمان اضافه تری نیاز دارند. به هر روی این زمان جهت اجتناب از هزینه روزانه تطبیق داده ها گرفته خواهد شد.

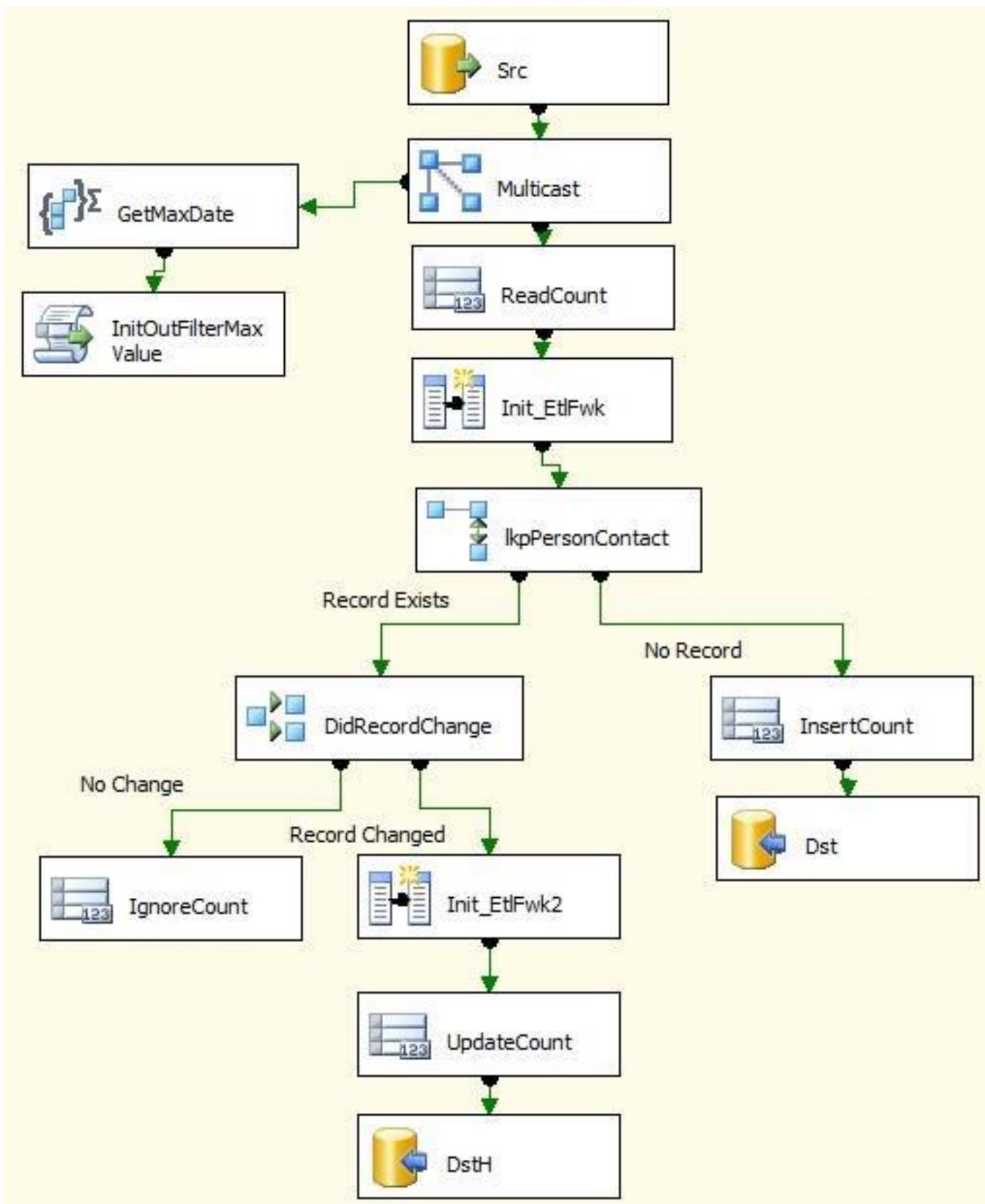
مزایای حاصل شده از افزودن این منطق اضافی عبارت است از:

پشتیبانی از ستون هایی که کارهای ممیزی و نظارت بر داده ها را آسان تر می کنند.

تعداد سطرهای شاخص مناسبی است که می تواند بهبود آن Data Flow خاص را فراهم کند. ناظر اطلاعات با استفاده از تعداد رکوردها می تواند ناهنجاری ها را شناسائی کند.

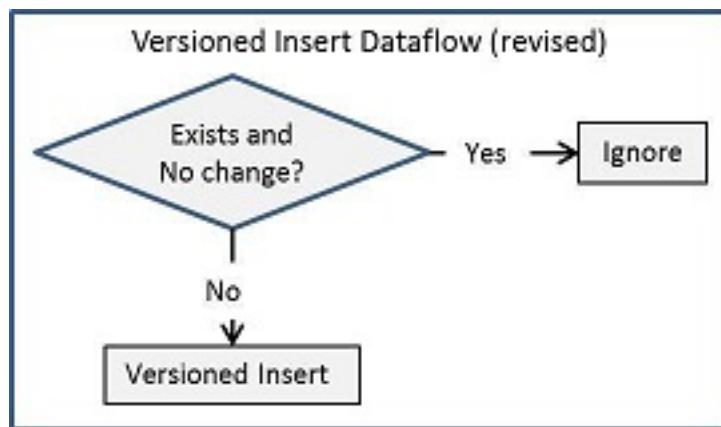
بهره برداران ETL و ناظران اطلاعات می توانند با استفاده از خلاصه تعداد رکوردها در کم بیشتری درباره فعالیت های آن کسب کنند.

پس از آنکه تعداد رکوردها، مشکوک به نظر آمد؛ تحقیقات بیشتری می تواند اتفاق افتد. (با عمیق تر شدن در جزئیات گزارشات)



Versioned Insert Pattern 4-1-3-

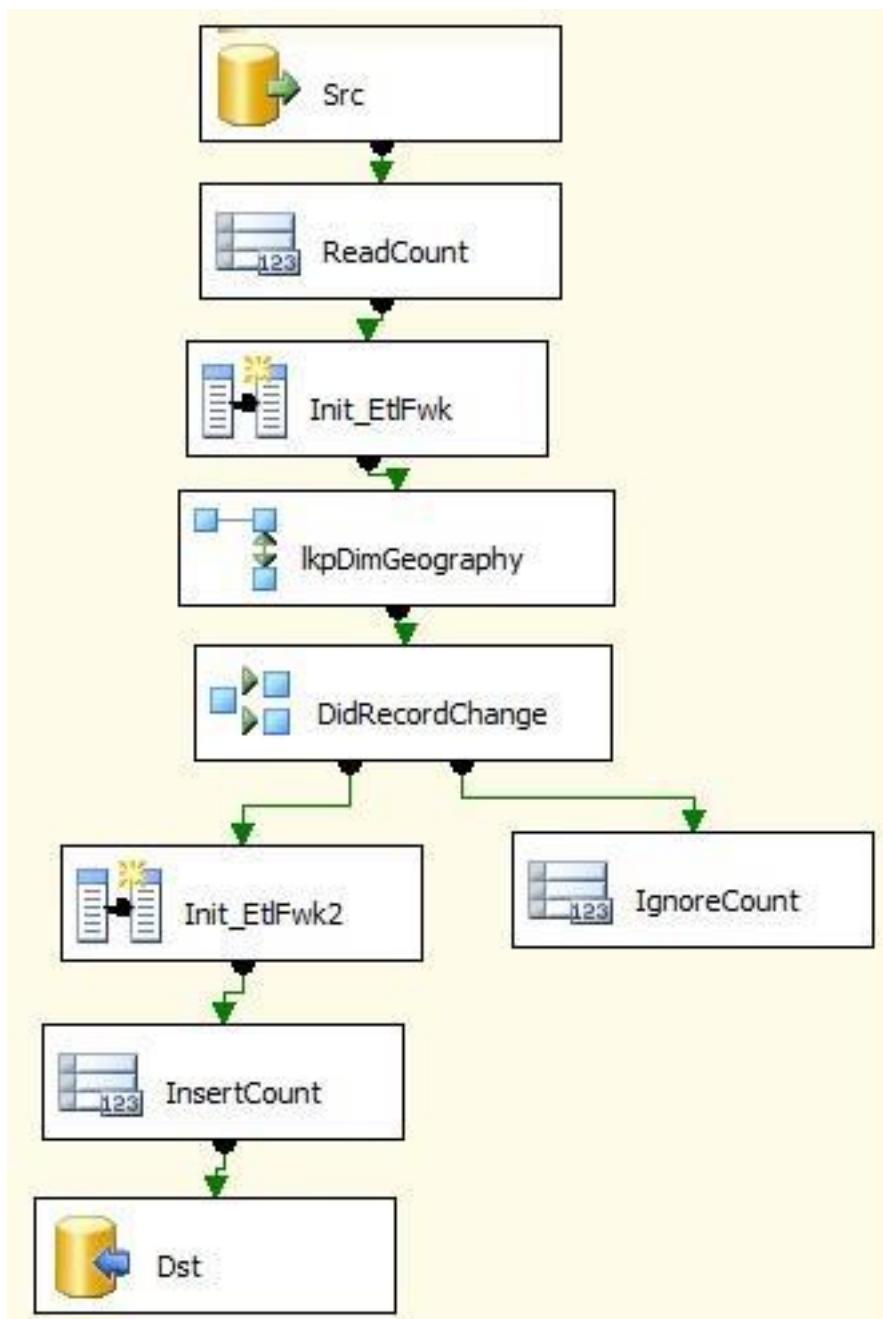
جدولی که به صورت Versioned Insert Dataflow Pattern پر شده است می تواند از استفاده کند. همانند شکل زیر که گردش کار در آن برای کارآئی بیشتر بازنگری شده است.



توجه داشته باشید Data Flow در این روش شامل:

تمامی رکوردهای جدید و تغییر یافته در جدول Versioned Insert قرار می‌گیرند.
این روش دارای Data Flow ساده‌تری نسبت به الگوی Update می‌باشد.

شكل زیر SSIS versioned insert data flow pattern را نشان می‌دهد:



تعدادی نکته در Data Flow فوق وجود دارد که عبارتند از:

در شیء "lkpDimGeography" گزینه "Ignore Failures" با مقدار "Redirect rows to no match output" تنظیم شده است.
شیء "DidRecordChange" بررسی می‌کند چنانچه ستون‌های مبداء و مقصد یکسان باشند، آیا کلید اصلی جدول مقصد Null است. اگر این عبارت True ارزیابی شود، رکورد نادیده گرفته می‌شود.
منطق شناسائی تغییرات دربردارنده تغییرات ستون داده ای در مبداء نمی‌باشد.
ستون و تعداد رکوردها مشابه با Data Flow قبلی (ETL Framework) می‌باشد.

Update vs. Versioned Insert 4-1-4-

الگوی Update نسبت الگوی Versioned Insert دارای پیاده سازی ساده‌تر و فعالیت‌های I/O کمتری است. از منظر دیگر، جدولی که از الگوی Update استفاده می‌کند، دارای تعداد رکوردهای کمتری است که می‌تواند به معنای Performance بهتر نیز تعبیر شود. ممکن است سوالی مطرح شود، اینکه چرا برای انجام کار به جدول تاریخچه نیاز است؛ این جدول را که نمی‌توان Truncate نمود،

پس چرا به منظور بروزرسانی از جدول اصلی استفاده می شود؟ پاسخ این پرسش در این است که جدول تاریخچه، ناظر اطلاعات و ممیزین داده را قادر می سازد، تغییرات در طول زمان را پیگیری نمایند.

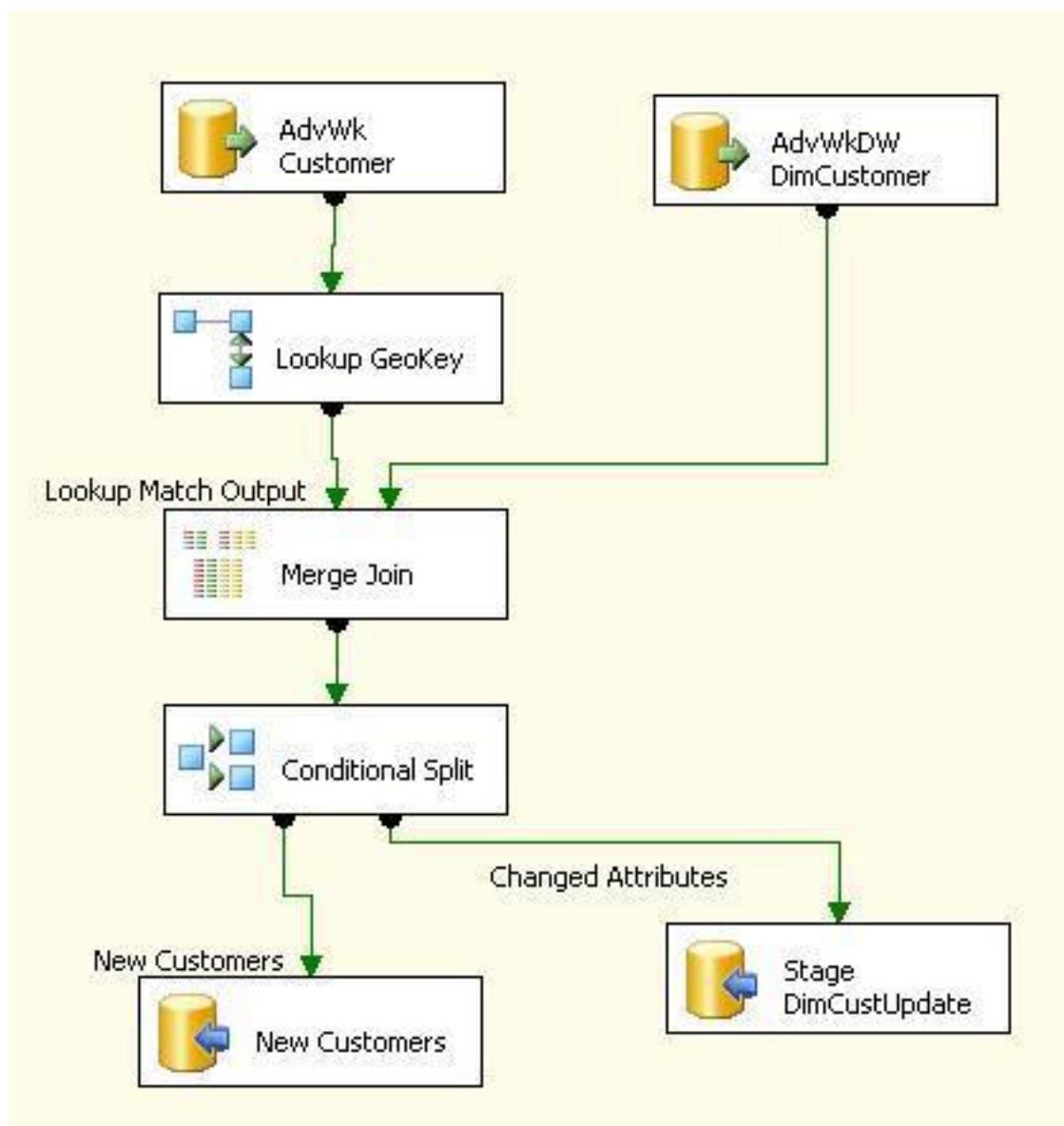
Dimension Patterns 4-2- بروزرسانی Dimension موارد زیر را شامل می شود:

پیگیری تاریخچه
انجام بروزرسانی
تشخیص رکوردهای جدید
surrogate keys مدیریت

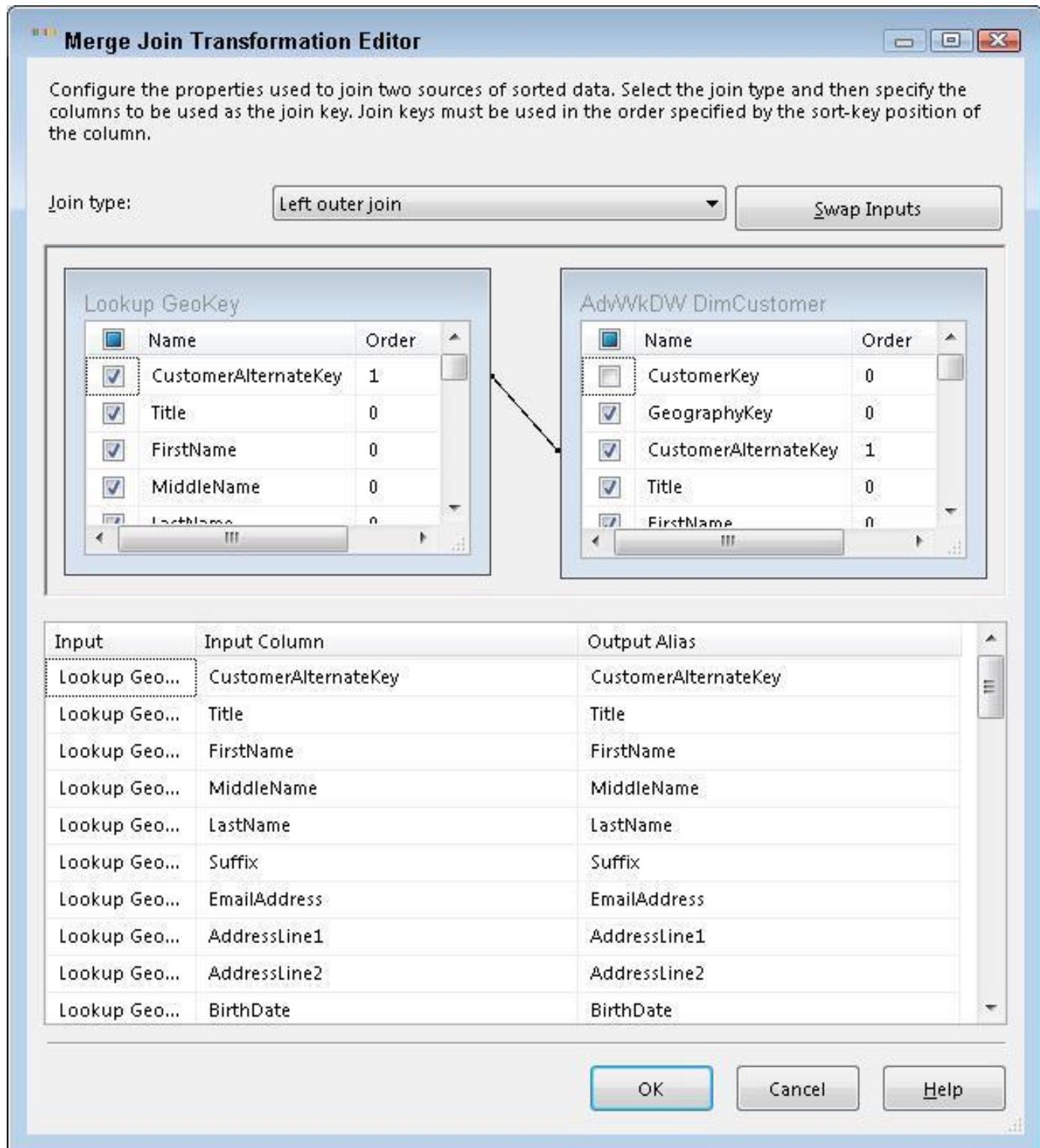
چنانچه با یک Dimension کوچک مواجه هستید (با مقدار هزاران رکورد یا کمتر، که با صدها هزار رکورد یا بیشتر ضدیت دارد)، می توانید از تبدیل "Slowly Changing Dimension" که بصورت Built-in SSIS موجود است، استفاده نمائید. به هر روی با آنکه این تبدیل چندین ویژگی محدود کننده Performance دارد، اغلب کارآمدتر از پروسسه هایی که توسط خودتان ایجاد می شود. در واقع فرآیند بارگذاری در جداول Dimension با مقایسه داده ها بین مبداء و مقصد انجام می شود. به طور معمول مقایسه روی یک ورژن جدید و یا مجموعه ای از سطرهای جدید یک جدول با مجموعه داده های موجود در جدول متناظرش صورت می گیرد. پس از تشخیص چگونگی تغییر در داده ها، یک سری عملیات درج و بروزرسانی انجام می شود. شکل زیر نمونه ای از پردازش سریع در Dimension را نمایش می دهد؛ که شامل مراحل اساسی زیر است:

منبع فوقانی سمت چپ، رکوردها را در یک SSIS از یک سیستم مبداء (یا یک سیستم میانی) به شکل Pull دریافت می کند. منبع فوقانی سمت راست، داده ها را از خود جدول Dimension به شکل Pull دریافت می کند. با استفاده از Join Merge Key رکوردها از طریق Source شان مقایسه می شوند. (در شکل بعدی جزئیات این مقایسه نمایش داده شده است).

با استفاده از یک Conditional Spilt داده ها ارزیابی می شوند؛ سطرها یا مستقیماً در جدول Dimension درج می شوند (منبع تحتانی سمت چپ) و یا در یک جدول عملیاتی (منبع تحتانی سمت راست) جهت انجام بروزرسانی درج می شوند. در گام پایانی (که نمایش داده نشده) مجموعه ای از بروزرسانی بین جدول عملیاتی و جدول Dimension صورت می گیرد.

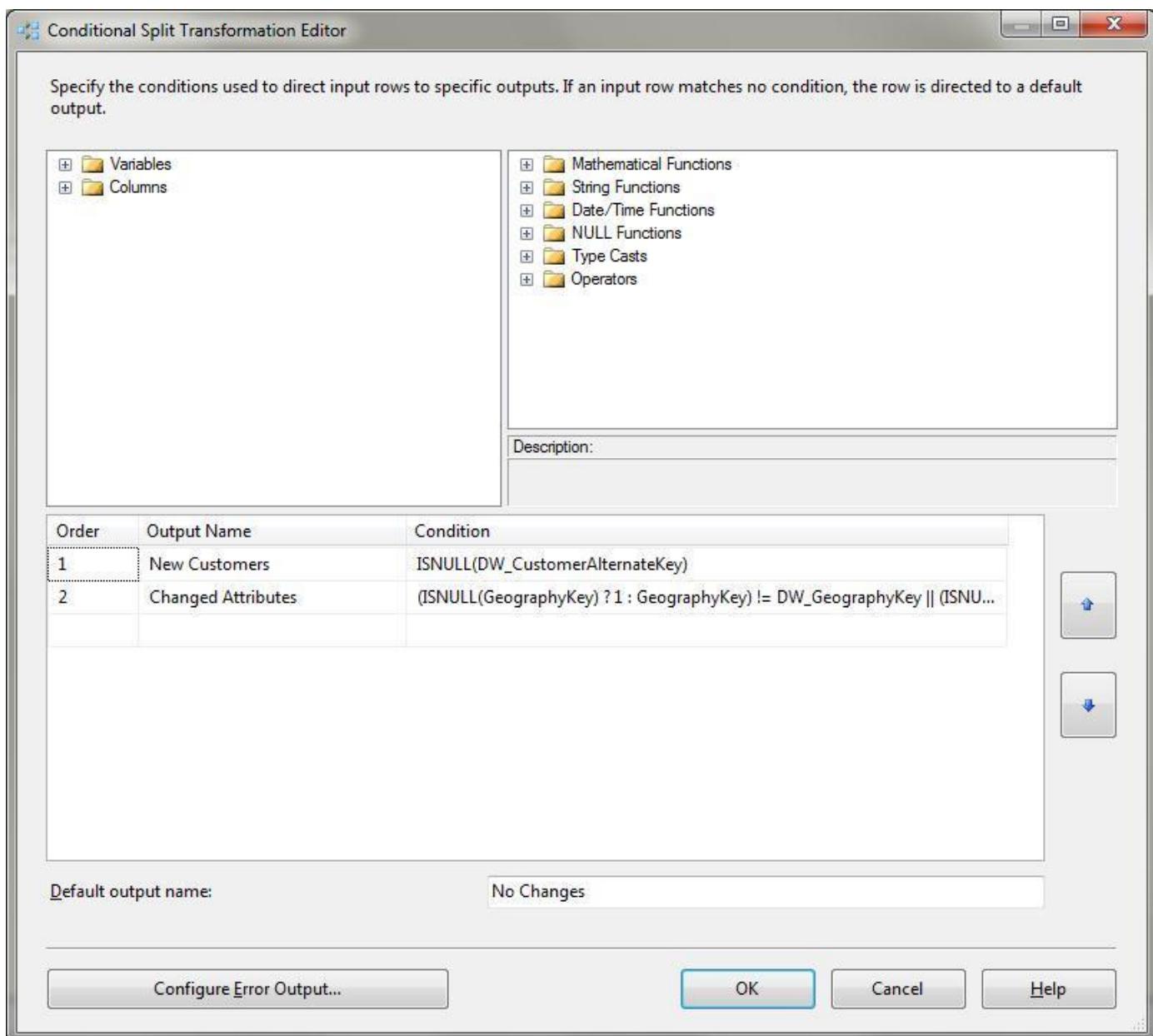


با Merge Join ارتباطی بین رکوردهای مبداء و رکوردهای مقصد برقرار می‌شود. (در این مثال "CustomerAlternateKey"). هنگامی که از این دیدگاه استفاده می‌کنید، خاطر جمع شوید که نوع Join با مقدار "Left outer join" تنظیم شده است؛ بدین ترتیب قادر هستید تا رکوردهای جدید را از مبداء تشخیص دهید؛ از آنجا که هنوز در جدول Dimension قرار نگرفته اند.



گام پایانی به منظور تشخیص اینکه آیا رکورد، جدید یا تغییر یافته است (یا بلاتکلیف است)، مقایسه داده هاست. شکل زیر نمایش می دهد چگونه این ارزیابی با استفاده از تبدیل "Conditional Spilt" صورت می گیرد.

بررسی بارگذاری داده ها در انبار های داده و معرفی الگوهای بکار رفته در آن



Conditional Split مستقیماً با استفاده از یک Adapter تعریف شده روی مقصد یا یک جدول کاری بروزرسانی که از یک Adapter تعریف شده روی مقصد استفاده می‌کند؛ توسط مجموعه دستور Update زیر، رکوردها را در جدول Dimension می‌دهد. دستور Update زیر مستقیماً با استفاده از روش Join روی جدول کاری، مجموعه ای را بصورت ابوهه بروزرسانی می‌کند.

```
UPDATE AdventureWorksDW2008R2.dbo.DimCustomer
SET AddressLine1 = stgDimCustomerUpdates.AddressLine1
, AddressLine2 = stgDimCustomerUpdates.AddressLine2
, BirthDate = stgDimCustomerUpdates.BirthDate
, CommuteDistance = stgDimCustomerUpdates.CommuteDistance
, DateFirstPurchase = stgDimCustomerUpdates.DateFirstPurchase
, EmailAddress = stgDimCustomerUpdates.EmailAddress
, EnglishEducation = stgDimCustomerUpdates.EnglishEducation
, EnglishOccupation = stgDimCustomerUpdates.EnglishOccupation
, FirstName = stgDimCustomerUpdates.FirstName
, Gender = stgDimCustomerUpdates.Gender
, GeographyKey = stgDimCustomerUpdates.GeographyKey
, HouseOwnerFlag = stgDimCustomerUpdates.HouseOwnerFlag
, LastName = stgDimCustomerUpdates.LastName
, MaritalStatus = stgDimCustomerUpdates.MaritalStatus
, MiddleName = stgDimCustomerUpdates.MiddleName
```

```
, NumberCarsOwned = stgDimCustomerUpdates.NumberCarsOwned
, NumberChildrenAtHome = stgDimCustomerUpdates.NumberChildrenAtHome
, Phone = stgDimCustomerUpdates.Phone
, Suffix = stgDimCustomerUpdates.Suffix
, Title = stgDimCustomerUpdates.Title
, TotalChildren = stgDimCustomerUpdates.TotalChildren
FROM AdventureWorksDW2008.dbo.DimCustomer DimCustomer
    INNER JOIN dbo.stgDimCustomerUpdates ON
DimCustomer.CustomerAlternateKey = stgDimCustomerUpdates.CustomerAlternateKey
```

Fact Table Patterns 4-3-

جداوی Fact به پردازش های منحصر به فردی نیازمند هستند، نخست به کلید های Dimension Surrogate جدول نیاز دارند تا های محاسبه شدنی را بدست آورند. این اعمال از طریق تبدیلات Derived Column و Lookup، Merge Join صورت می گیرد. با بروزرسانی ها، تفاضل رکوردها و یا Snapshot بیشتر این فرآیندهای دشوار انجام می شوند.

Inserts 4-3-1-

روی اغلب جداول Fact عمل درج صورت می گیرد؛ که کار متداولی در جدول Fact می باشد. شاید ساده ترین کار که در فرآیند ساخت ETL صورت می گیرد، عملیات درج روی تنها تعدادی از جدول Fact می باشد. درج کردن در صورت لزوم بارگذاری انبوی داده ها، مدیریت شاخص ها و مدیریت پارتیشن ها را شامل می شود.

Updates 4-3-2-

بروزرسانی روی جداول Fact معمولاً به یکی از سه طریق زیر انجام می گیرد:

از طریق یک تغییر یا بروزرسانی رکورد
از طریق یک دستور Insert خنثی کننده (Via an Insert of a compensating transaction)
با استفاده از یک SQL MERGE

در موردی که تغییرات با فرکانس کمی روی جدول Fact صورت می گیرد و یا فرآیند بروزرسانی قابل مدیریت است؛ ساده ترین روش انجام یک دستور Update روی جدول Fact می باشد. نکته مهمی که هنگام انجام بروزرسانی باید به خاطر داشته باشید، استفاده از روش بروزرسانی مبتنی بر مجموعه است؛ به همان طریق که در قسمت الگوهای Dimension ذکر آن رفت. در طریقی دیگر (درج compensating) می توان اقدام به درج رکورد تغییر یافته نمود، تا ترجیحاً بروزرسانی روی آن صورت گیرد. این استراتژی به سادگی داده های جدول Fact میان سیستم مبداء و مقصد را که تغییر یافته اند، به صورت یک رکورد جدید درج خواهد کرد. تصویر زیر مثالی از اجرای موارد فوق را نمایش می دهد.

Source ID	Measure Value
12345	80

Source data

Source ID	Measure Value
12345	100

Current fact table data

Source ID	Current Measure Value
12345	100
12345	- 20

New fact table data

در آخرین روش از یک دستور SQL MERGE استفاده می شود که در آن با استفاده از ادغام و مقایسه، تمامی داده های جدید و تغییر یافته جدول Fact، درج و یا بروزرسانی می شوند. نمونه ای از استفاده دستور Merge به شرح زیر است:

```
MERGE dbo.FactSalesQuota AS T
USING SSIS_PDS.dbo.stgFactSalesQuota AS S
ON T.EmployeeKey = S.EmployeeKey
AND T.DateKey = S.DateKey
WHEN MATCHED AND BY target
THEN INSERT(EmployeeKey, DateKey, CalendarYear, CalendarQuarter, SalesAmountQuota)
VALUES(S.EmployeeKey, S.DateKey, S.CalendarYear, S.CalendarQuarter, S.SalesAmountQuota)
WHEN MATCHED AND T.SalesAmountQuota != S.SalesAmountQuota
THEN UPDATE SET T.SalesAmountQuota = S.SalesAmountQuota
;
```

اشکال این روش Performance است؛ گرچه این دستور به سادگی عملیات درج و بروزرسانی را انجام می دهد ولی به صورت سطر به سطر عملیات انجام می شود (در هر زمان یک سطر). در موقعیت هایی که با مقدار زیادی داده مواجه هستید، اغلب بهتر است به صورت انبوه عملیات درج و به صورت مجموعه عملیات بروزرسانی انجام گیرد.

Managing Inferred Members 4-3-3-

زمانیکه یک ارجاع در جدول Fact به یک عضو Dimension که هنوز بارگذاری نشده است بوجود آید؛ یک Inferred Member تعبیر می شود. به سه طریق می توان این Inferred Member ها را مدیریت نمود:

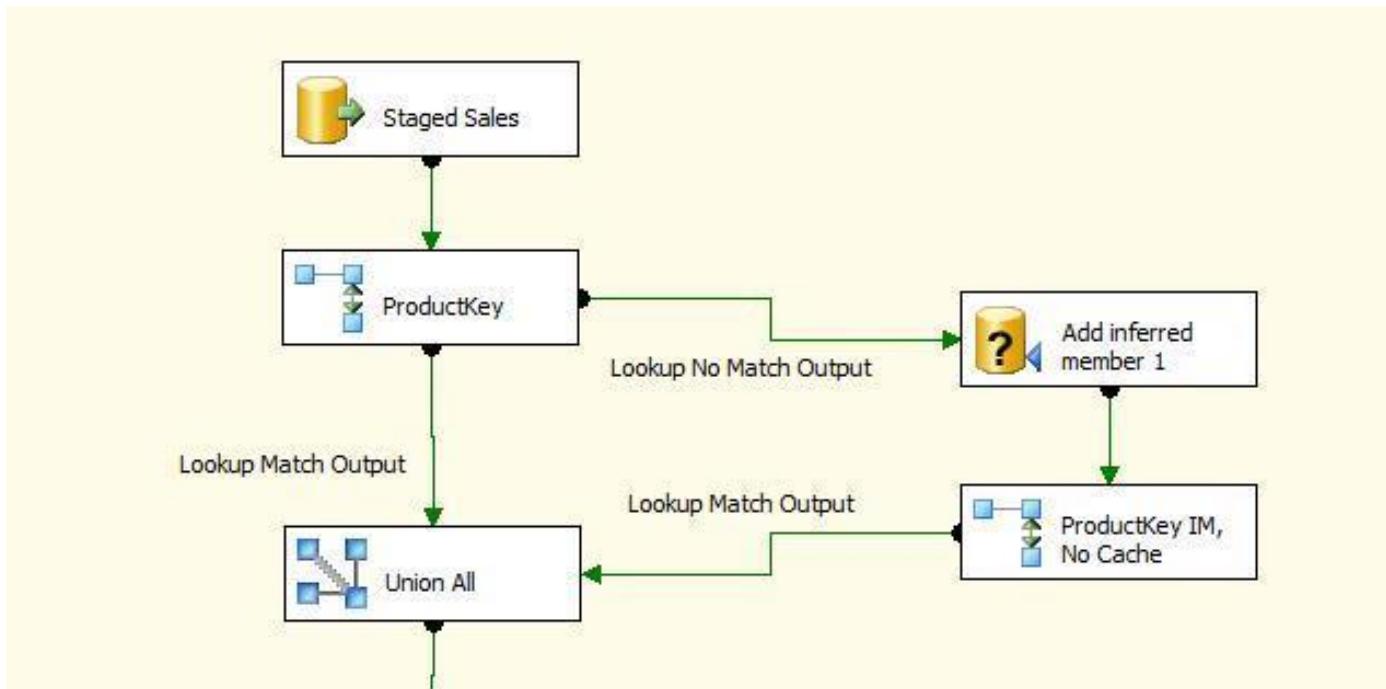
ركوردهای جدول Fact پیش از درج اسکن شوند؛ ایجاد هر Dimension در Inferred Member و سپس بارگذاری رکوردها در جدول Fact

در طول عملیات بارگذاری روی Fact؛ هر رکورد مفقوده شده به یک جدول موقتی ارسال شود، رکوردهای مفقوده شده به Dimension اضافه شود، در ادامه مجدد آن رکوردهای Fact در جدول بارگذاری شوند.

در یک Data Flow زمانی که یک رکورد مفقود شده، بلاتکلیف تعبیر می شود؛ آن زمان یک رکورد به Dimension اضافه شود و

بدست آمده را برگردانیم: سپس Dimension بارگذاری شود.

شکل زیر این موارد را نمایش می دهد:



بر اساس [رفتار پیش فرض](#) در دیتابیس SQL Server، در زمان انجام دادن یک دستور که منجر به ایجاد تغییرات در اطلاعات موجود در جدول می‌شود (برای مثال دستور Update)، جدول مربوطه به صورت کامل Lock می‌شود، ولو آن دستور Update، فقط با یکی از رکوردهای آن جدول کار داشته باشد.

در سیستم‌های با تعداد تراکنش بالا و دارای تعداد زیاد کلاینت، این رفتار پیش فرض موجب ایجاد صفحی از تراکنش‌های در حال انتظار بر روی جداولی می‌شود که ویرایش‌های زیادی بر روی آنها رخ می‌دهد. اگرچه که بنظر این مشکل [راه حل‌های زیادی دارد](#)، لکن آن راه حلی که همیشه موثر عمل می‌کند استفاده از SQL Server Table Hints است.

SQL به تمامی آن دستوراتی گفته می‌شود که هنگام اجرای دستور اصلی (برای مثال Select و یا Update) Rفتار پیش فرض SQL Server را بر اساس Hint ارائه شده تغییر می‌دهند. لیست کامل این Hint‌ها را می‌توانید در [اینجا مشاهده کنید](#).

Hint ای که در اینجا برای ما مفید است، آن است که به SQL Server بگوییم هنگام اجرای دستور Update، به جای Lock کردن کل جدول، فقط رکورد در حال ویرایش را Lock کند، و این باعث می‌شود تا باقی تراکنش‌ها، که ای بسا با سایر رکوردهای آن جدول کار داشته باشند متوقف نشوند، که البته این مسئله کمی به افزایش مصرف حافظه می‌انجامد، لکن مقدار افزایش بسیار ناچیز است.

این rowlock نام دارد در تراکنش‌های با Isolation Level تنظیم شده بر روی Snapshot باید با یک Table Hint دیگر با نام updlock ترکیب شود.

توضیحات مفصل‌تر این دو Hint در لینک مربوطه آمده است.
بنابر این، بجای دستور

```
update products
set Name = "Test"
Where Id = 1
```

داریم

```
update products with (nolock,updlock)
set Name = "Test"
where Id = 1
```

تا اینجا مشکل خاصی وجود ندارد، آنچه که از اینجا به بعد اهمیت دارد این است که در هنگام کار با Entity Framework، اساساً ما نویسنده دستورات Update نیستیم که به آنها Hint اضافه کنیم یا نه، بلکه دستورات SQL بوسیله Entity Framework ایجاد می‌شوند.

در Entity Framework، مکانیزمی تعییه شده است با نام Db Command Interceptor که به شما اجازه می‌دهد دستورات SQL ساخته شده را [Log کنید](#) و یا قبل از اجرا [تغییر دهید](#)، که برای اضافه نمودن Table Hint‌ها ما از این روش استفاده می‌کنیم، برای انجام این کار داریم: (توضیحات در ادامه)

```
public class UpdateRowLockHintDbCommandInterceptor : IDbCommandInterceptor
{
    public void NonQueryExecuting(DbCommand command, DbCommandInterceptionContext<Int32>
interceptionContext)
    {
        if (command.CommandType != CommandType.Text) return; // (1)
        if (!(command is SqlCommand)) return; // (2)
        SqlCommand sqlCommand = (SqlCommand)command;
        String commandText = sqlCommand.CommandText;
        String updateCommandRegularExpression = "(update) ";
    }
}
```

```
Boolean isUpdateCommand = Regex.IsMatch(commandText, updateCommandRegularExpression,
RegexOptions.IgnoreCase | RegexOptions.Multiline); // You may use better regular expression pattern here.

if (isUpdateCommand)
{
    Boolean isSnapshotIsolationTransaction = sqlCommand.Transaction != null &&
sqlCommand.Transaction.IsolationLevel == IsolationLevel.Snapshot;
    String tableHintToAdd = isSnapshotIsolationTransaction ? " with (rowlock , updlock) set
" : " with (rowlock) set ";
    commandText = Regex.Replace(commandText, "^(" + tableHintToAdd + ") ", (match) =>
{
    return tableHintToAdd;
}, RegexOptions.IgnoreCase | RegexOptions.Multiline);
    sqlCommand.CommandText = commandText;
}
```

این کد در قسمت (1) ابتدا تشخیص می‌دهد که آیا این یک Command Text است یا خیر، برای مثال اگر فراخوانی یک Stored Procedure است، ما با آن کاری نداریم.

در قسمت دوم تشخیص می‌دهیم که آیا با SQL Server در حال تعامل هستیم، یا برای مثال با Oracle و ...، که ما برای Table ها فقط با SQL Server Hint کار داریم.

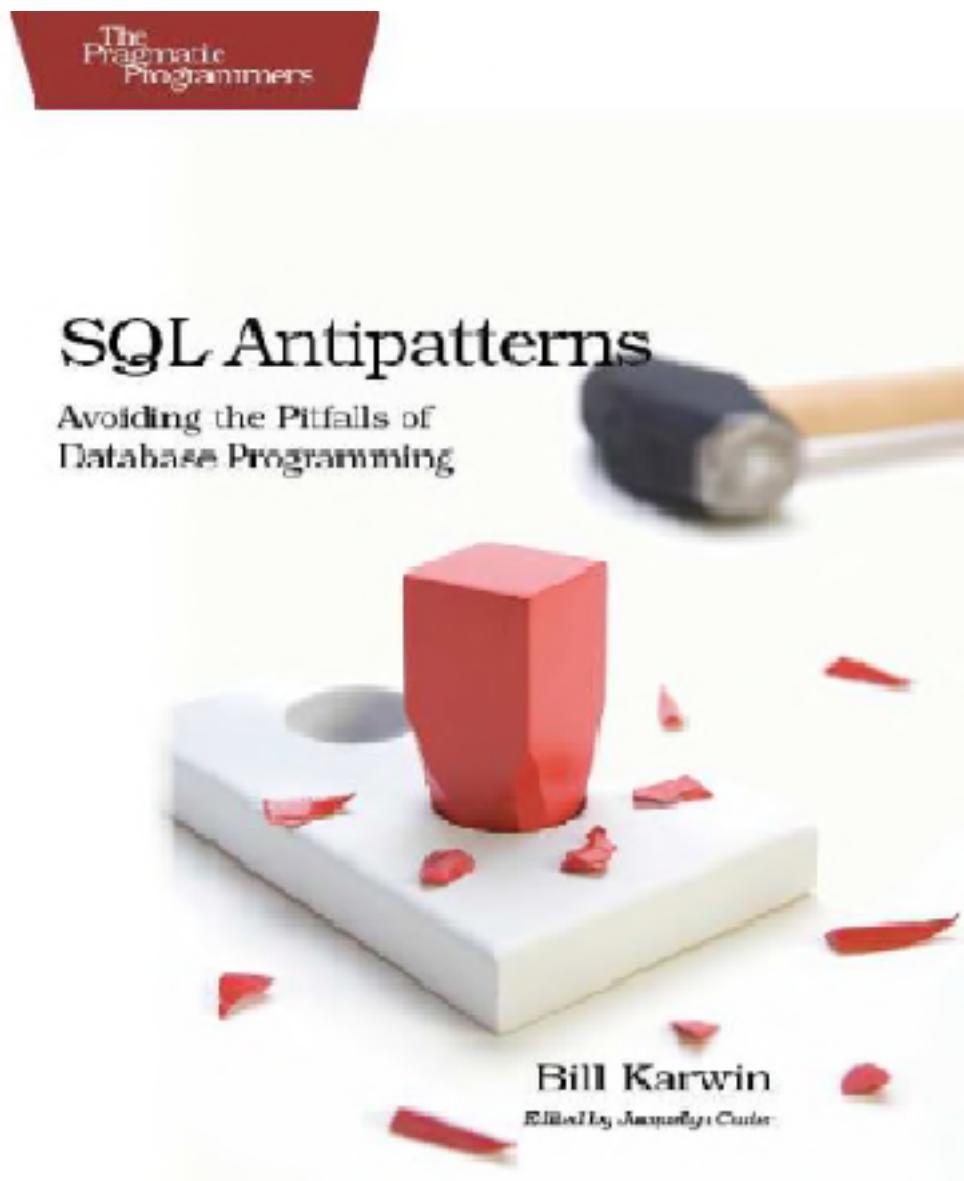
سپس باید تشخیص دهیم که آیا این یک دستور update است یا خیر؟ برای این منظور از Regular Expression ها استفاده کرده ایم، که خیلی به بحث آموزش این پست مربوط نیست، به صورت کلی از Regular Expression ها برای یافتن و بررسی و جایگزینی عبارات با قاعده در هنگام کار با رشته‌ها استفاده می‌شود.

ممکن است Regular Expression ای که شما می‌نویسید بسیار بهتر از این نمونه باشد، که در این صورت خوشحال می‌شوم در قسمت نظرات آنرا قرار دهید.

در نهایت با بررسی Transaction Isolation Level Snapshot مربوطه که درج یک یا هر دو Table Hint مربوطه اقدام می‌نماییم.

عنوان: SQL Antipattern #1
نویسنده: سید مرتضی حسینی
تاریخ: ۱۳۹۳/۰۴/۲۹
آدرس: www.dotnettips.info
گروه‌ها: SQL Server, SQL, Antipatterns

در این سلسله نوشتار قصد دارم ترجمه و خلاصه چندین فصل از کتاب ارزشمند ([SQL Antipatterns: Avoiding the Pitfalls](#)) که حاصل تلاش گروه IT موسسه هدایت فرهیختگان جوان می‌باشد، را [of Database Programming \(Pragmatic Programmers\)](#) ارائه نمایم.



بخش اول : Jaywalking

در این بخش در حال توسعه ویژگی نرم افزاری هستیم که در آن هر کاربر به عنوان یک کاربر اصلی برای یک محصول تخصیص داده می‌شود. در طراحی اصلی ما فقط اجازه می‌دهیم یک کاربر متعلق به هر محصول باشد، اما امکان چنین تقاضایی وجود دارد

که چند کاربر نیز به یک محصول اختصاص داده شوند.

در این صورت، اگر پایگاه داده را به نحوی تغییر دهیم که شناسه‌ی حساب کاربران را در لیستی که با کاما از یکدیگر جدا شده‌اند ذخیره نماییم، خیلی ساده‌تر به نظر می‌رسد نسبت به اینکه بصورت جداگانه آنها را ثبت نماییم.

برنامه نویسان معمولاً برای جلوگیری از ایجاد جدول واسطه [1] که رابطه‌های چند به چند زیادی دارد از یک لیست که با کاما داده‌هایش از هم جدا شده‌اند، استفاده می‌کند. بدین جهت اسم این بخش `jaywalking`, `antipatten`, `zj` می‌باشد، زیرا نیز عملیاتی است که از تقاطع جلوگیری می‌کند.

1.1 هدف: ذخیره کردن چندین صفت

طراحی کردن جدولی که ستون آن فقط یک مقدار دارد، بسیار ساده و آسان می‌باشد. شما می‌توانید نوع داده‌ایی که متعلق به ستون می‌باشد را انتخاب نمایید. مثلاً از نوع (...int,date)

چگونه می‌توانیم مجموعه‌ایی از مقادیری که به یکدیگر مرتبط هستند را در یک ستون ذخیره نماییم؟

در مثال ردیابی خطای پایگاه داده، ما یک محصول را به یک کاربر، با استفاده از ستونی که نوع آن `integer` است، مرتبط می‌نماییم. هر کاربر ممکن است محصولاتی داشته باشد و هر محصول به یک `contact` اشاره کند. بنابراین یک ارتباط چند به یک بین محصولات و کاربر برقرار است. بر عکس این موضوع نیز صادق است؛ یعنی امکان دارد هر محصول متعلق به چندین کاربر باشد و یک ارتباط یک به چند ایجاد شود. در زیر جدول محصولات را به صورت عادی آورده شده است:

```
CREATE TABLE Products (
product_id SERIAL PRIMARY KEY,
product_name VARCHAR(1000),
account_id BIGINT UNSIGNED,
-- ...
FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
);

INSERT INTO Products (product_id, product_name, account_id)
VALUES (DEFAULT, 'Visual TurboBuilder' , 12);
```

Antipattern 2.1 : قالب بندی لیست هایی که با کاما از یکدیگر جدا شده اند

برای اینکه تغییرات در پایگاه داده را به حداقل برسانید، می‌توانید نوع شناسه‌ی کاربر را از نوع `varchar` قرار دهید. در این صورت می‌توانید چندین شناسه‌ی کاربر که با کاما از یکدیگر جدا شده‌اند را در یک ستون ذخیره نمایید. پس در جدول محصولات، شناسه‌ی کاربران را از نوع `varchar` قرار می‌دهیم.

```
CREATE TABLE Products (
product_id SERIAL PRIMARY KEY,
product_name VARCHAR(1000),
account_id VARCHAR(100), -- comma-separated list
-- ...
);

INSERT INTO Products (product_id, product_name, account_id)
VALUES (DEFAULT, 'Visual TurboBuilder' , '12,34' );
```

اکنون مشکلات کارایی و جامعیت داده‌ها را در این راه حل پیشنهادی بررسی می‌نماییم.

بدست آوردن محصولاتی برای یک کاربر خاص

بدلیل اینکه تمامی شناسه‌ی کاربران که بصورت کلید خارجی جدول `Products` می‌باشند به صورت رشته در یک فیلد ذخیره

شده‌اند و حالت ایندکس بودن آنها از دست رفته است، بدست آوردن مخصوصاتی برای یک کاربر خاص سخت می‌باشد. به عنوان مثال بدست آوردن مخصوصاتی که کاربری با شناسه‌ی 12 خریداری نموده به صورت زیر می‌باشد:

```
SELECT * FROM Products WHERE account_id REGEXP '[[[:<:]]12[[[:>:]]]' ;
```

بدست آوردن کاربرانی که یک محصول خاص خریداری نموده اند بدست `Accounts` table `account_id` با `Join` کردن `Products` یعنی جدول مرجع نامناسب و غیر معمول می‌باشد. مساله مهمی که وجود دارد این است که زمانیکه بدین صورت شناسه‌ی کاربران را ذخیره می‌نماییم، ایندکس بودن آنها از بین می‌رود. کد زیر کاربرانی که مخصوصی با شناسه‌ی 123 خریداری کرده‌اند را محاسبه می‌کند.

```
SELECT * FROM Products AS p JOIN Accounts AS a
ON p.account_id REGEXP '[[[:<:]]' || a.account_id || '[[[:>:]]]'
WHERE p.product_id = 123;
```

ایجاد توابع تجمعی [2]

مبانی چنین توابعی از قبیل `sum()`, `count()`, `avg()` بدين صورت می‌باشد که بر روی گروهی از سطراها اعمال می‌شوند. با این حال با کمک ترفندهایی می‌توان برخی از این توابع را تولید نماییم هر چند برای همه آن‌ها این مساله صادق نمی‌باشد. اگر بخواهیم تعداد کاربران برای هر محصول را بدست بیاوریم ابتدا باید طول لیست شناسه‌ی کاربران را محاسبه کنیم، سپس کاما را از این لیست حذف کرده و طول جدید را از طول قبلی کم کرده و با یک جمع کنیم. نمونه کد آن در زیر آورده شده است:

```
SELECT product_id, LENGTH(account_id) - LENGTH(REPLACE(account_id, ',', '' )) + 1
AS contacts_per_product
FROM Products;
```

ویرایش کاربرانی که یک محصول خاص خریداری نموده‌اند

ما به راحتی می‌توانیم یک شناسه‌ی جدید را به انتهای این رشته اضافه نماییم؛ فقط مرتب بودن آن را بهم میریزیم. در نمونه اسکریپت زیر گفته شده که در جدول محصولات برای محصولی با شناسه‌ی 123 بعد از کاما یک کاربر با شناسه‌ی 56 درج شود:

```
UPDATE Products
SET account_id = account_id || ',' || 56
WHERE product_id = 123;
```

برای حذف یک کاربر از لیست کافیست دو دستور sql را اجرا کرد: اولی برای `fetch` کردن شناسه‌ی کاربر از لیست و بعدی برای ذخیره کردن لیست ویرایش شده. بطور مثال تمامی افرادی که محصولی با شناسه‌ی 123 را خریداری کرده‌اند، از جدول محصولات انتخاب می‌کنیم. برای حذف کاربری با شناسه‌ی 34 از لیست کاربران، بر حسب کاما مقادیر لیست را در آرایه می‌ریزیم، شناسه‌ی مورد نظر را جستجو می‌کنیم و آن را حذف می‌کنیم. دوباره مقادیر درون آرایه را بصورت رشته درمی‌اوریم و جدول محصولات را با لیست جدید کاربران برای محصولی با شناسه‌ی 123 بروزرسانی می‌کنیم.

```
<?php
$stmt = $pdo->query(
"SELECT account_id FROM Products WHERE product_id = 123");
$row = $stmt->fetch();
$contact_list = $row['account_id'];

// change list in PHP code
$value_to_remove = "34";
$contact_list = split(", ", $contact_list);
$key_to_remove = array_search($value_to_remove, $contact_list);
unset($contact_list[$key_to_remove]);
$contact_list = join(", ", $contact_list);
$stmt = $pdo->prepare(
"UPDATE Products SET account_id = ?
WHERE product_id = 123");
$stmt->execute(array($contact_list));
```

به دلیل آنکه نوع فیلد account_id varchar می‌باشد احتمال این وجود دارد داده‌ای نامعتبر وارد نماییم چون پایگاه داده‌ها هیچ عکس العملی یا خطای را نشان نمی‌دهد، فقط از لحاظ معنایی دچار مشکل می‌شویم. در نمونه زیر banana یک داده‌ی غیر معتبر می‌باشد و ارتباطی با شناسه‌ی کاربران ندارد.

```
INSERT INTO Products (product_id, product_name, account_id)
VALUES (DEFAULT, 'Visual TurboBuilder' , '12,34,banana');
```

انتخاب کردن کاراکتر جداگانه

نکته قابل توجه این است که کاراکتری که عنوان جداگانه در نظر می‌گیریم باید در هیچ‌کدام از داده‌های ورودی ما امکان بودنش وجود نداشته باشد.

محدودیت طول لیست

در زمان طراحی جدول، برای هر یک از فیلدها باید حداقل طولی را قرار دهیم. به عنوان نمونه ما اگر (30) varchar در نظر بگیریم بسته به تعداد کاراکترهایی که داده‌های ورودی ما دارند می‌توانیم جدول را پر نماییم. اسکرپت زیر شناسه‌ی کاربرانی که محصولی با شناسه‌ی 123 را خریده‌اند، ویرایش می‌کند. با این تفاوت که با توجه به محدودیت لیست، در نمونه‌ی اولی شناسه‌ی کاربران دو کاراکتری بوده و 10 داده بعنوان ورودی پذیرفته است در حالیکه در نمونه‌ی دوم بخاطر اینکه شناسه‌ی کاربران شش کاراکتری می‌باشد فقط 4 شناسه می‌توانیم وارد نماییم.

```
UPDATE Products SET account_id = '10,14,18,22,26,30,34,38,42,46'
WHERE product_id = 123;
```

```
UPDATE Products SET account_id = '101418,222630,343842,467790'
WHERE product_id = 123;
```

3.1 موارد تشخیص این :Antipattern

سؤالات زیر نشان می‌دهند که Jaywalking antipattern مورد استفاده قرار گرفته است:

- حداقل پذیرش این لیست برای داده ورودی چه میزان است؟
- چه کاراکتری هرگز در داده‌های ورودی این لیست ظاهر نمی‌شود؟ **4.1 مواردی که استفاده از این Antipattern مجاز است:** دی نرمال کردن کارایی را افزایش می‌دهد. ذخیره کردن شناسه‌ها در یک لیست که با کاما از یکدیگر جدا شده‌اند نمونه بارزی از دی نرمال کردن می‌باشد. ما زمانی می‌توانیم از این مدل استفاده نماییم که به جدا کردن شناسه‌ها از هم نیاز نداشته باشیم. به همین ترتیب، اگر در برنامه، شما یک لیست را از یک منبع دیگر با این فرمت دریافت نمایید، به سادگی آن را در پایگاه داده خود به همان فرمت بصورت کامل می‌توانید ذخیره و بازیابی نمایید و احتیاجی به مقادیر جداگانه ندارید. درهنگام استفاده از پایگاه داده‌های دی نرمال دقت کنید. برای شروع از پایگاه داده نرمال استفاده کنید زیرا به کدهای برنامه شما امکان انعطاف بیشتری می‌دهد و کمک می‌کند تا پایگاه داده‌ها تمامیت داده(Data integrity) را حفظ کند. **5.1 راه حل:** استفاده کردن از جدول واسطه در این روش شناسه‌ی کاربران را در جدول جداگانه‌ای که هر کدام از آنها یک سطر را به خود اختصاص داده اند، ذخیره می‌نماییم.

```
CREATE TABLE Contacts (
    product_id BIGINT UNSIGNED NOT NULL,
    account_id BIGINT UNSIGNED NOT NULL,
    PRIMARY KEY (product_id, account_id),
    FOREIGN KEY (product_id) REFERENCES Products(product_id),
    FOREIGN KEY (account_id) REFERENCES Accounts(account_id)
);
```



جدول Contacts یک جدول رابطه ای بین جداول Products, Accounts

بدست آوردن محصولات برای کاربران و موارد مربوط به آن

ما بر احتی می‌توانیم تمامی محصولاتی که مختص به یک کاربر هستند را بدست آوریم. در این شیوه خاصیت ایندکس بودن شناسه کاربران حفظ می‌شود به همین دلیل query‌های آن برای خواندن و بهینه کردن راحت‌تر می‌باشند. در این روش به کاراکتری برای جدا کردن ورودی‌ها از یکدیگر نیاز نداریم چون هر کدام از آنها در یک سطر جداگانه ثبت می‌شوند. برای ویرایش کردن کاربرانی که یک محصول را خریداری نموده اند، کافیست یک سطر از جدول واسطه را اضافه یا حذف نماییم. در نمونه کد زیر، ابتدا در جدول Contacts کاربری با شناسه 34 که محصولی با شناسه 456 را خریداری کرده، درج شده است و در خط بعد عملیات حذف با شرط آنکه شناسه کاربر و محصول به ترتیب 34,456 باشد روی جدول Contacts اعمال شده است.

```
INSERT INTO Contacts (product_id, account_id) VALUES (456, 34);
DELETE FROM Contacts WHERE product_id = 456 AND account_id = 34;
```

ایجاد توابع تجمیعی

به عنوان نمونه در مثال زیر بر احتی ما می‌توانیم تعداد محصولات در هر حساب کاربری را بدست آوریم:

```
SELECT account_id, COUNT(*) AS products_per_account
FROM Contacts
GROUP BY account_id;
```

اعتبارسنجی شناسه محصولات

از آنجاییکه مقادیری که در جدول قرار دارند کلید خارجی می‌باشند می‌توان صحت اعتبار آنها را بررسی نمود. عنوان مثال Account.account_id اشاره می‌کند. در ضمن برای هر فیلد نوع آن را می‌توان مشخص کرد تا فقط همان نوع داده را بپذیرد.

محدودیت طول لیست

نسبت به روش قبلی تقریبا در این حالت محدودیتی برای تعداد کاراکترهای ورودی نداریم.

مزیت‌های دیگر استفاده از جدول واسط

کارایی روش دوم بهتر از حالت قبلی می‌باشد چون ایندکس بودن شناسه‌ها حفظ شده است. همچنین بر احتی می‌توانیم فیلدی را به این جدول اضافه نماییم مثل (...time, date)

Intersection Table [1]

Aggregate [2]

نظرات خوانندگان

نویسنده: م منفرد
تاریخ: ۰:۴۸ ۱۳۹۳/۰۴/۳۰

سلام
منظورتان از گروه `it` موسسه هدایت فرهیختگان چه بود؟ ایشان کتاب را ترجمه کردند؟ چگونه می‌توان آن را تهیه کرد؟
با تشکر

نویسنده: فرید
تاریخ: ۲۰:۱۸ ۱۳۹۳/۰۶/۲۶

سلام؛ به نظر میرسه این خصیصه مربوط به `mysql` است. من نتوانستم آنرا در `sqlserver` اجرا کنم.

بخش دوم : Naive Trees

فرض کنید یک وب سایت حرفه‌ای خبری یا علمی-پژوهشی داریم که قابلیت دریافت نظرات کاربران را در مورد هر مطلب مندرج در سایت یا نظرات داده شده در مورد آن مطالب را دارا می‌باشد. یعنی هر کاربر علاوه بر توانایی اظهار نظر در مورد یک خبر یا مطلب باید بتواند پاسخ نظرات کاربران دیگر را نیز بدهد. اولین راه حلی که برای طراحی این مطلب در پایگاه داده به ذهن ما می‌رسد، ایجاد یک زنجیره با استفاده از کد SQL زیر می‌باشد:

```
CREATE TABLE Comments (
comment_id SERIAL PRIMARY KEY,
parent_id BIGINT UNSIGNED,
comment TEXT NOT NULL,
FOREIGN KEY (parent_id) REFERENCES Comments(comment_id)
);
```

البته همان طور که پیداست بازیابی زنجیره‌ای از پاسخ‌ها در یک پرس‌وچوی SQL کار سختی است. این نخ‌ها عموماً عمق نامحدودی دارند و برای به دست آوردن تمام نخ‌های یک زنجیره باید پرس‌وچوهای زیادی را اجرا نمود.

ایده‌ی دیگر می‌تواند بازیابی تمام نظرها و ذخیره‌ی آن‌ها در حافظه‌ی برنامه به صورت درخت باشد. ولی این روش برای ذخیره هزاران نظری که ممکن است در سایت ثبت شود و علاوه بر آن مقالات جدیدی که اضافه می‌شوند، تقریباً غیرعملی است.

1.2 هدف: ذخیره و ایجاد پرس‌وچو در سلسله‌مراتب

وجود سلسله‌مراتب بین داده‌ها امری عادی محسوب می‌گردد. در ساختار داده‌ای درختی هر ورودی یک گره محسوب می‌گردد. یک گره ممکن است تعدادی فرزند و یک پدر داشته باشد. گره اول پدر ندارد، ریشه و گره فرزند که فرزند ندارد، برگ و گره‌ای دیگر، گره‌های غیربرگ نامیده می‌شوند.

مثال‌هایی که از ساختار درختی داده‌ها وجود دارد شامل موارد زیر است:

در این ساختار برای مثال در ارتباط بین کارمندان و مدیر، هر کارمند یک مدیر دارد که نشان‌دهنده‌ی پدر یک کارمند در ساختار درختی است. هر مدیر هم یک کارمند محسوب می‌گردد.

در این ساختار برای مثال در سیستم نظردهی و پاسخ‌ها، ممکن است زنجیره‌ای از نظرات در پاسخ به نظرات دیگر استفاده گردد. در درخت، فرزندان یک گرهی نظر، پاسخ‌های آن گره هستند.

در این فصل ما از مثال ساختار دوم برای نشان دادن Antipattern و راه حل آن بهره می‌گیریم.

Antipattern 2.2 : همیشه مبتنی بر یکی از والدین

راه حل ابتدایی و ناکارآمد

اضافه نمودن ستون parent_id . این ستون، به ستون نظر در همان جدول ارجاع داده می‌شود و شما می‌توانید برای اجرای این رابطه از قید کلید خارجی استفاده نمایید. پرس‌وچویی که برای ساخت مثالی که ما در این بحث از آن استفاده می‌کنیم در ادامه آمده است:

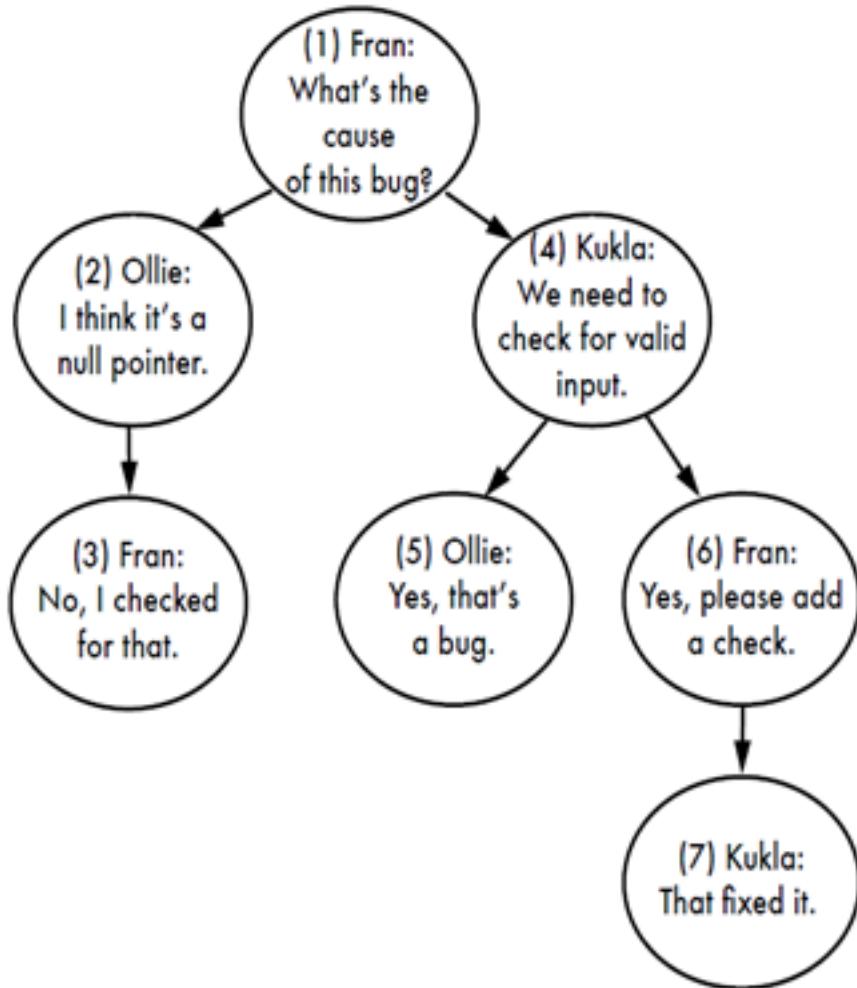
```
CREATE TABLE Comments (
    comment_id SERIAL PRIMARY KEY,
    parent_id BIGINT UNSIGNED,
    bug_id BIGINT UNSIGNED NOT NULL,
    author BIGINT UNSIGNED NOT NULL,
    comment_date DATETIME NOT NULL,
    comment TEXT NOT NULL,
    FOREIGN KEY (parent_id) REFERENCES Comments(comment_id),
    FOREIGN KEY (bug_id) REFERENCES Bugs(bug_id),
    FOREIGN KEY (author) REFERENCES Accounts(account_id)
);
```

مثالی از پرس‌وجوی فوق را می‌توانید در زیر ببینید:

comment_id	parent_id	author	comment
1	NULL	Fran	What's the cause of this bug?
2	1	Ollie	I think it's a null pointer.
3	2	Fran	No, I checked for that.
4	1	Kukla	We need to check for invalid input.
5	4	Ollie	Yes, that's a bug.
6	4	Fran	Yes, please add a check.
7	6	Kukla	That fixed it.

لیست مجاورت :

لیست مجاورت خود می‌تواند به عنوان یک antipattern در نظر گرفته شود. البته این مطلب از آنجایی نشأت می‌گیرد که این روش توسط بسیاری از توسعه‌دهنگان مورد استفاده قرار می‌گیرد ولی نتوانسته است به عنوان راه حل برای معمول‌ترین وظیفه‌ی خود، یعنی ایجاد پرس‌وجو بر روی کلیه فرزندان، باشد.



- با استفاده از پرس‌وجوی زیر می‌توان فرزند بلافاصله‌ی یک "نظر" را برگرداند:

```
SELECT c1.*, c2.*  
FROM Comments c1 LEFT OUTER JOIN Comments c2  
ON c2.parent_id = c1.comment_id;
```

ضعف پرس‌وجوی فوق این است که فقط می‌تواند دو سطح از درخت را برای شما برگرداند. در حالیکه خاصیت درخت این است که شما را قادر می‌سازد بدون هیچ گونه محدودیتی فرزندان و نوه‌های متعدد (سطوح بیشمار) برای درخت خود تعریف کنید. بنابراین به ازای هر سطح اضافه باید یک `join` به پرس‌جوی خود اضافه نمایید. برای مثال اگر پرس‌وجوی زیر می‌تواند درختی با چهار سطح برای شما برگرداند ولی نه بیش از آن:

```
SELECT c1.*, c2.*, c3.*, c4.*  
FROM Comments c1 -- 1st level  
LEFT OUTER JOIN Comments c2  
ON c2.parent_id = c1.comment_id -- 2nd level  
LEFT OUTER JOIN Comments c3  
ON c3.parent_id = c2.comment_id -- 3rd level  
LEFT OUTER JOIN Comments c4  
ON c4.parent_id = c3.comment_id; -- 4th level
```

این پرس‌جو به این دلیل که با اضافه شدن ستون‌های دیگر، نوه‌ها را سطوح عمیق‌تری برمی‌گرداند، پرس‌وجوی مناسبی نیست.

در واقع استفاده از توابع تجمیعی ، مانند COUNT() مشکل می‌شود.

راه دیگر برای به دست آوردن ساختار یک زیردرخت از لیست مجاورت برای یک برنامه، این است که سطرهای مورد نظر خود را از مجموعه بازیابی نموده و سلسه‌مراتب مورد نظر را در حافظه بازیابی نماییم و از آن به عنوان درخت استفاده نماییم:

```
SELECT * FROM Comments WHERE bug_id = 1234;
```

نگهداری کردن یک درخت با استفاده از لیست مجاورت

البته برخی از عملکردها با لیست مجاورت به خوبی انجام می‌گیرد. برای مثال اضافه نمودن یک گره (نظر)، مکان‌یابی مجدد برای یک گره یا یک زیردرخت.

```
INSERT INTO Comments (bug_id, parent_id, author, comment)
VALUES (1234, 7, 'Kukla', 'Thanks!');
```

بازیابی دوباره مکان یک نود یا یک زیردرخت نیز آسان است:

```
UPDATE Comments SET parent_id = 3 WHERE comment_id = 6;
```

با این حال حذف یک گره از یک درخت در این روش پیچیده است. اگر بخواهیم یک زیردرخت را حذف کنید باید چندین پرس‌وحو برای پیدا کردن تمام نوه‌ها بنویسیم و سپس حذف نوه‌ها را از پایین‌ترین سطح شروع کرد و تا جایی که قید کلید خارجی برقرار شود ادامه دهیم. البته می‌توان از کلید خارجی با تنظیم ON DELETE CASCADE استفاده کرد تا این کارها به طور خودکار انجام گیرد. حال اگر بخواهیم یک نود غیر برگ را حذف کرده یا فرزندان آن را در درخت جایجا کنیم، ابتدا باید parent_id فرزندان آن نود را تغییر داده و سپس نود مورد نظر را حذف می‌کنیم:

```
SELECT parent_id FROM Comments WHERE comment_id = 6; -- returns 4
UPDATE Comments SET parent_id = 4 WHERE parent_id = 6;
DELETE FROM Comments WHERE comment_id = 6;
```

3.2 موارد تشخیص این Antipattern

سؤالات زیر نشان می‌دهند که Naive Trees antipattern مورد استفاده قرار گرفته است:
چه تعداد سطح برای پشتیبانی در درخت نیاز خواهیم داشت؟

من همیشه از کار با کدی که ساختار داده‌ی درختی را مدیریت می‌کند، می‌ترسم

من باید اسکریپتی را به طور دوره‌ای اجرا نمایم تا سطرهای یتیم موجود در درخت را حذف کند.

4.2 مواردی که استفاده از این Antipattern مجاز است:

قدرت لیست مجاورت در بازیابی پدر یا فرزند مستقیم یک نود می‌باشد. قرار دادن یک سطر هم در لیست مجاورت کار ساده‌ای است. اگر این عملیات، تمام آن چیزی است که برای انجام کارتان مورد نیاز شما است، بنابراین استفاده از لیست مجاورت می‌تواند مناسب باشد.

برخی از برندهای RDBMS از افزونه‌هایی پشتیبانی می‌کنند که قابلیت ذخیره‌ی سلسله مراتب را در لیست مجاورت ممکن می‌سازد. مثلاً SQL-99، پرس‌وحوی بازگشتنی را تعریف می‌کند که مثال آن در ادامه آمده است:

```
WITH CommentTree (comment_id, bug_id, parent_id, author, comment, depth)
AS (
SELECT *, 0 AS depth FROM Comments
WHERE parent_id IS NULL
UNION ALL
SELECT c.*, ct.depth+1 AS depth FROM CommentTree ct
JOIN Comments c ON (ct.comment_id = c.parent_id)
)
SELECT * FROM CommentTree WHERE bug_id = 1234;
```

Oracle 9i، IBM DB2 و Microsoft SQL Server 2005، Oracle 11g، PostgreSQL 8.4 نیز از پرس‌وجوی بازگشتی پشتیبانی می‌کنند. 10g از عبارت WITH استفاده می‌کند، ولی نه برای پرس‌وجوهای بازگشتی. در عوض می‌توانید از پرس‌وجوی زیر برای ایجاد پرس‌وجوی بازگشتی استفاده نمایید:

```
SELECT * FROM Comments
START WITH comment_id = 9876
CONNECT BY PRIOR parent_id = comment_id;
```

5.2 راه حل: استفاده از مدل‌های درختی دیگر

جایگزین‌های دیگری برای ذخیره‌سازی داده‌های سلسله مراتبی وجود دارد. البته برخی از این راه حل‌ها ممکن است در لحظه‌ی اول پیچیدتر از لیست مجاورت به نظر آیند، ولی برخی از عملیات درخت که در لیست مجاورت بسیار سخت یا ناکارآمد است، را آسان‌تر می‌کنند. **شمارش مسیر**:

مشکل پرهزینه بودن بازیابی نیاکان یک گره که در روش لیست مجاورت وجود داشت در روش شمارش مسیر به این ترتیب حل شده است: اضافه نمودن یک صفت به هر گره که رشته‌ای از نیکان آن صفت در آن ذخیره شده است. رشته‌ای در جدول Comments به جای استفاده از id، parent_id، یک ستون به نام path که نوع آن varchar است تعریف شده است. رشته‌ای که در این ستون تعریف شده است، ترتیبی از فرزندان این سطر از بالا به پایین درخت است. مانند مسیری که در سیستم عامل UNIX، برای نشان دادن مسیر در سیستم فایل استفاده شده است. شما می‌توانید از / به عنوان کاراکتر جداکننده استفاده نمایید. دقت کنید برای درست کار کردن پرس‌وجوها حتماً در آخر مسیر هم این کاراکتر را قرار دهید. پرس‌وجوی تشکیل چنین درختی به شکل زیر است:

```
CREATE TABLE Comments (
    comment_id SERIAL PRIMARY KEY,
    path VARCHAR(1000),
    bug_id BIGINT UNSIGNED NOT NULL,
    author BIGINT UNSIGNED NOT NULL,
    comment_date DATETIME NOT NULL,
    comment TEXT NOT NULL,
    FOREIGN KEY (bug_id) REFERENCES Bugs(bug_id),
    FOREIGN KEY (author) REFERENCES Accounts(account_id)
```

comment_id	path	author	comment
1	1/	Fran	What's the cause of this bug?
2	1/2/	Ollie	I think it's a null pointer.
3	1/2/3/	Fran	No, I checked for that.
4	1/4/	Kukla	We need to check for invalid input.
5	1/4/5/	Ollie	Yes, that's a bug.
6	1/4/6/	Fran	Yes, please add a check.
7	1/4/6/7/	Kukla	That fixed it.

در این روش، هر گره مسیری دارد که شماره خود آن گره هم در آنتهای آن مسیر قرار دارد. این به دلیل درست جواب دادن پرس‌وجوهای ایجاد شده است. می‌توان نیاکان را با مقایسه‌ی مسیر سطر کنونی با مسیر سطر دیگر به دست آورد. برای مثال برای یافتن نیاکان گره (نظر) شماره‌ی 7 که مسیر آن 7/6/4/1/ می‌باشد، می‌توان چنین نوشت:

```
SELECT * FROM Comments AS c
WHERE '1/4/6/7/' LIKE c.path || '%' ;
```

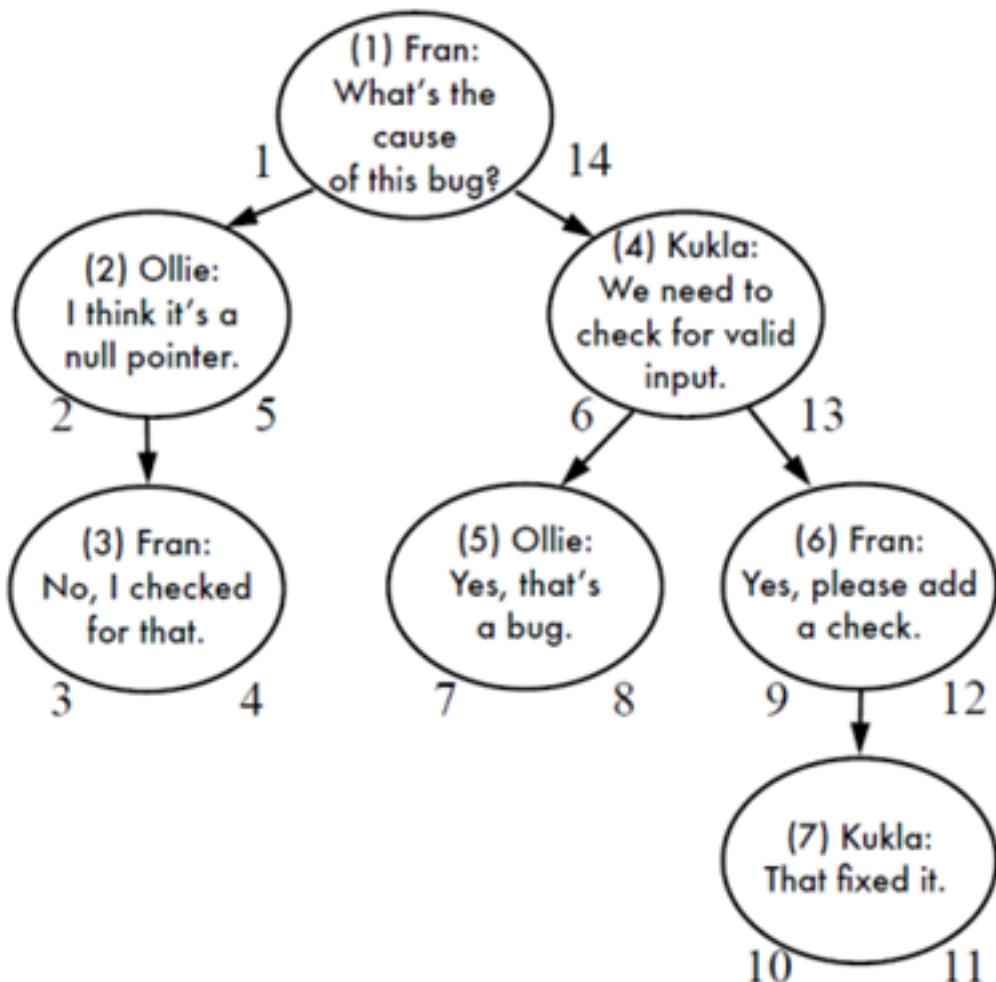
این پرس‌وجو الگوهای را می‌یابد که از مسیرهای 1/%/4/1, %/6/4/1 و 1/% نشأت می‌گیرد.
همچنین فرزندان (نوههای) یک گره، مثلاً گرهی 4 را که مسیرش 1/4/ است را می‌توان با پرس‌وجوی زیر یافت:

```
SELECT * FROM Comments AS c
WHERE c.path LIKE '1/4/' || '%' ;
```

الگوی 1/%/4/ با مسیرهای 1/, 5/4/1, 6/4/1 و 7/6/4/1 تطابق می‌یابد.
همچنین می‌توان پرس‌وجوهای دیگری را نیز در این مسیر به سادگی انجام داد؛ مانند محاسبه‌ی مجموع هزینه‌ی گره‌ها در یک زیردرخت یا شمارش تعداد گره‌ها.
اضافه نمودن یک گره هم مانند ساختن خود مدل است. می‌توان یک گرهی غیر برگ را بدون نیاز به اصلاح هیچ سطری اضافه نمود. برای این کار مسیر را از گرهی پدر کپی کرده و در انتهای شماره‌ی خود گره را به آن اضافه می‌کنیم.
از مشکلات این روش می‌توان به عدم توانایی پیگاه داده‌ها در تحمیل این نکته که مسیر یک گره درست ایجاد شده است و یا تضمین وجود گره‌ای در مسیری خاص، اشاره نمود. همچنین نگهداری رشته‌ی مسیر یک گره مبتنی بر کد برنامه است و اعتبارسنجی آن کاری هزینه‌بر است. این رشته اندازه‌ای محدود دارد و درخت‌هایی با عمق نامحدود را پشتیبانی نمی‌کند.

مجموعه‌های تودرتو :

مجموعه‌های تودرتو، اطلاعات را با هر گره‌ای که مربوط به مجموعه‌ای از نوهایش است، به جای این که تنها مربوط به یک فرزند بلافصلش باشد، ذخیره می‌کنند.



این اطلاعات می‌توانند به وسیله‌ی هر گره‌ای که در درخت با دو شماره‌ی `nsleft` و `nsright` ذخیره شده، نمایش داده شوند:

```
CREATE TABLE Comments ( comment_id SERIAL PRIMARY KEY,
nsleft INTEGER NOT NULL,
nsright INTEGER NOT NULL,
bug_id BIGINT UNSIGNED NOT NULL,
author BIGINT UNSIGNED NOT NULL,
comment_date DATETIME NOT NULL,
comment TEXT NOT NULL,
FOREIGN KEY (bug_id) REFERENCES Bugs (bug_id),
FOREIGN KEY (author) REFERENCES Accounts(account_id)
);
```

comment_id	nsleft	nsright	author	comment
1	1	14	Fran	What's the cause of this bug?
2	2	5	Ollie	I think it's a null pointer.
3	3	4	Fran	No, I checked for that.
4	6	13	Kukla	We need to check for invalid input.
5	7	8	Ollie	Yes, that's a bug.
6	9	12	Fran	Yes, please add a check.
7	10	11	Kukla	That fixed it.

شماره‌ی سمت چپ یک گره از تمام شماره‌های سمت چپ فرزندان آن گره کوچک‌تر و شماره‌ی سمت راست آن گره از تمام شماره‌های سمت راست آن گره بزرگ‌تر است. این شماره‌ها هیچ ارتباطی به `comment_id` مربوط به آن گره ندارند.

یک راه حل ساده برای تخصیص این شماره‌ها به گره‌ها این است که از سمت چپ یک گره آغاز می‌کنیم و اولین شماره را اختصاص می‌دهیم و به همین به گره‌ای سمت چپ فرزندان می‌آییم و شماره‌ها را به صورت افزایشی به سمت چپ آن‌ها نیز اختصاص می‌دهیم. سپس در ادامه به سمت راست آخرین نود رفته و از آن جا به سمت بالا می‌آییم و به همین ترتیب به صورت بازگشتی تخصیص شماره‌ها را ادامه می‌دهیم.

با اختصاص شماره‌ها به هر گره، می‌توان از آن‌ها برای یافتن نیاکان و فرزندان آن گره بهره جست. برای مثال برای بازیابی گره‌ی 4 و فرزندان (نوه‌های) آن باید دنبال گره‌هایی باشیم که شماره‌های آن گره‌ها بین `nsleft` و `nsright` شماره 4 باشد:

```
SELECT c2.* FROM Comments AS c1
JOIN Comments as c2
ON c2.nsleft BETWEEN c1.nsleft AND c1.nsright
WHERE c1.comment_id = 4;
```

همچنین می‌توان گره‌ی شماره‌ی 6 و نیاکان آن را با دنبال نمودن گره‌هایی به دست آورد که شماره‌های آن‌ها در محدوده‌ی شماره‌ی گره‌ی 6 باشد:

```
SELECT c2.* 
FROM Comments AS c1
JOIN Comment AS c2
ON c1.nsleft BETWEEN c2.nsleft AND c2.nsright
WHERE c1.comment_id = 6;
```

یک مزیت مهم روش مجموعه‌ای تودرتو، این است که هنگامی که یک گره را حذف می‌کنیم، نووهای آن به طور مستقیم به عنوان فرزندان پدر گرهی حذف شده تلقی می‌شوند.

برخی از پرس‌وجوهایی که در روش لیست مجاورت ساده بودند، مانند بازیابی فرزند یا پدر بلافصل، در روش مجموعه‌های تودرتو پیچیده‌تر می‌باشند. برای مثال برای یافتن پدر بلافصل گرهی شماره‌ی 6 باید چنین نوشت:

```
SELECT parent.* FROM Comment AS c
JOIN Comment AS parent
ON c.nsleft BETWEEN parent.nsleft AND parent.nsright
LEFT OUTER JOIN Comment AS in_between
ON c.nsleft BETWEEN in_between.nsleft AND in_between.nsright
AND in_between.nsright BETWEEN parent.nsleft AND parent.nsright
WHERE c.comment_id = 6
AND in_between.comment_id IS NULL;
```

دستکاری درخت، اضافه، حذف و جابجا نمودن گره‌ها در آن درروش مجموعه‌های تودرتو از مدل‌های دیگر پیچیده‌تر است. هنگامی که یک گرهی جدید را اضافه می‌کنیم، باید تمام مقادیر چپ و راست بزرگ‌تر از مقدار سمت چپ گرهی جدید را مجدداً محاسبه کنیم؛ که این شامل برادر سمت راست گرهی جدید، نیاکان آن و برادر سمت راست نیاکان آن می‌باشد. همچنین اگر گرهی جدید به عنوان گرهی غیربرگ اضافه شده باشد، شامل فرزندان آن هم می‌شود. برای مثال اگر بخواهیم گرهی جدیدی به گرهی 5 اضافه نماییم، باید چنین بنویسیم:

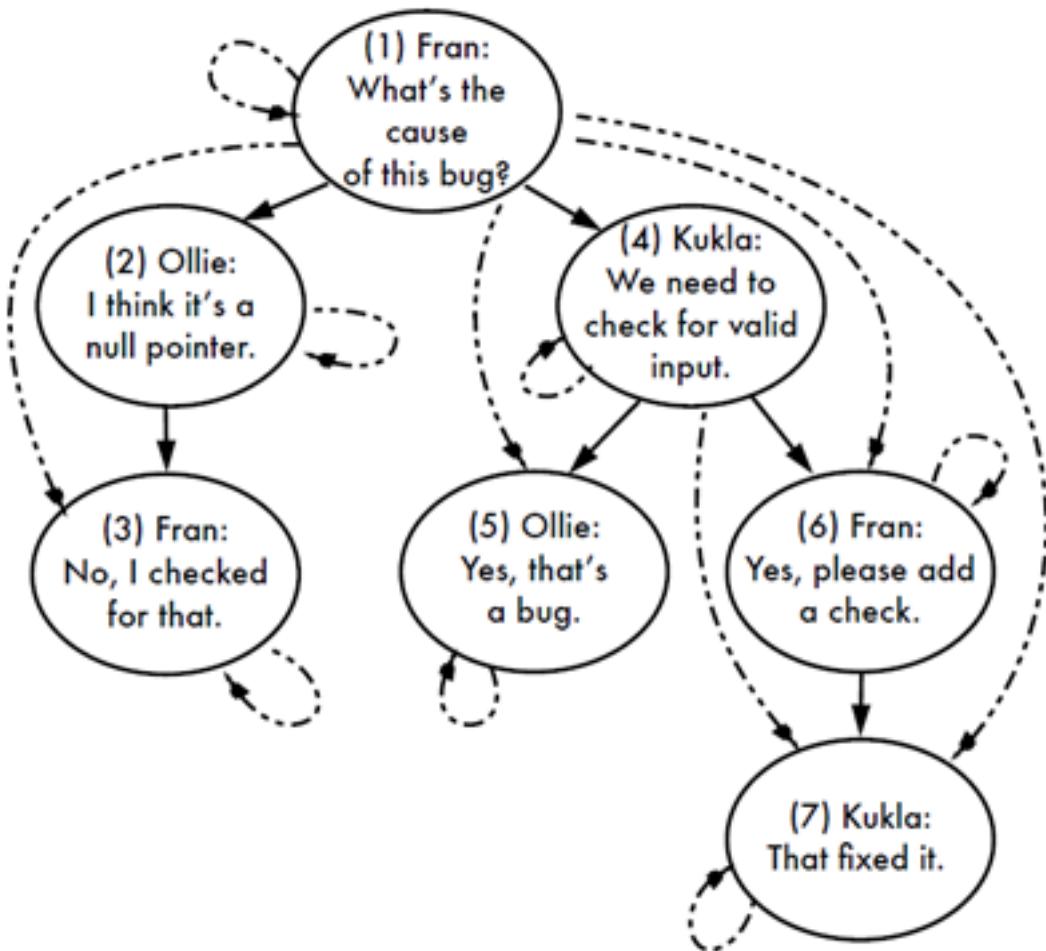
```
-- make space for NS values 8 and 9
UPDATE Comment
SET nsleft = CASE WHEN nsleft >= 8 THEN nsleft+2 ELSE nsleft END,
nsright = nsright+2
WHERE nsright >= 7;

-- create new child of comment #5, occupying NS values 8 and 9
INSERT INTO Comment (nsleft, nsright, author, comment)
VALUES (8, 9, 'Fran', 'Me too!');
```

تنها مزیت این روش نسبت به روش‌های قبلی ساده‌تر و سریع‌تر شدن ایجاد پرس‌وجوهای برای پیدا کردن فرزندان یا پدران یک درخت است. اگر هدف استفاده از درخت شامل اضافه نمودن متعدد گره‌ها است، مجموعه‌های تودرتو انتخاب خوبی نیست.

Closure Table

راه حل closure table روشی دیگر برای ذخیره سلسله‌مراتبی است. این روش علاوه بر ارتباطات مستقیم پدر-فرزنده، تمام مسیرهای موجود در درخت را ذخیره می‌کند.



این روش علاوه بر داشتن یک جدول نظرها، یک جدول دیگر به نام TreePaths با دو ستون دارد که هر کدام از این ستون‌ها یک کلید خارجی به جدول Comment هستند:

```

CREATE TABLE Comments (
    comment_id SERIAL PRIMARY KEY,
    bug_id BIGINT UNSIGNED NOT NULL,
    author BIGINT UNSIGNED NOT NULL,
    comment_date DATETIME NOT NULL,
    comment TEXT NOT NULL,
    FOREIGN KEY (bug_id) REFERENCES Bugs(bug_id),
    FOREIGN KEY (author) REFERENCES Accounts(account_id)
);
CREATE TABLE TreePaths (
    ancestor BIGINT UNSIGNED NOT NULL,
    descendant BIGINT UNSIGNED NOT NULL,
    PRIMARY KEY(ancestor, descendant),
    FOREIGN KEY (ancestor) REFERENCES Comments(comment_id),
    FOREIGN KEY (descendant) REFERENCES Comments(comment_id)
);
  
```

ancestor	descendant	ancestor	descendant	ancestor	descendant
1	1	1	7	4	6
1	2	2	2	4	7
1	3	2	3	5	5
1	4	3	3	6	6
1	5	4	4	6	7
1	6	4	5	7	7

به جای استفاده از جدول Comments برای ذخیره اطلاعات مربوط به یک درخت از جدول TreePath استفاده می‌کنیم. به ازای هر یک گره در این درخت یک سطر در جدول ذخیره می‌شود که ارتباط پدر فرزندی را نمایش می‌دهد و الزاماً نباید این دو پدر فرزند بلافصل باشد. همچنین یک سطر هم به ازای ارتباط هر گره با خودش به جدول اضافه می‌گردد.

پرس‌وجوهای بازیابی نیاکان و فرزندان (گره‌ها) از طریق جدول TreePaths ساده‌تر از روش مجموعه‌های تودرتو است. مثلاً برای بازیابی فرزندان (نوه‌های) گرهی شماره‌ی 4، سطرهایی که نیاکان آن‌ها 4 است را به دست می‌آوریم:

```
SELECT c.* FROM Comments AS c
JOIN TreePaths AS t ON c.comment_id = t.descendant
WHERE t.ancestor = 4;
```

برای به دست آوردن نیاکان گرهی شماره‌ی 6، سطرهایی از جدول TreePaths را به دست می‌آوریم که فرزندان آن‌ها 6 باشد:

```
SELECT c.*
FROM Comments AS c
JOIN TreePaths AS t ON c.comment_id = t.ancestor
WHERE t.descendant = 6;
```

برای اضافه کردن گرهی جدید، برای مثال به عنوان فرزند گرهی شماره‌ی 5، ابتدا سطری که به خود آن گره برمی‌گردد را اضافه می‌کنیم، سپس یک کپی از سطوری که در جدول TreePaths، به عنوان فرزندان (نوه‌های) گرهی شماره‌ی 5 هستند (که شامل سطری که به خود گرهی 5 به عنوان فرزند اشاره می‌کند) به جدول اضافه نموده و فیلد descendant آن را با شماره‌ی گرهی جدید جایگزین می‌کنیم:

```
INSERT INTO TreePaths (ancestor, descendant) SELECT t.ancestor, 8
FROM TreePaths AS t
WHERE t.descendant = 5
UNION ALL
SELECT 8, 8;
```

در اینجا می‌توان به اهمیت ارجاع یک گره به خودش به عنوان پدر (یا فرزند) پی برد. برای حذف یک گره، مثلاً گرهی شماره‌ی 7، تمام سطوری که فیلد descendant آن‌ها در جدول TreePaths برابر با 7 است حذف می‌کنیم:

```
DELETE FROM TreePaths WHERE descendant = 7;
```

برای حذف یک زیردرخت کامل، برای مثال گرهی شماره‌ی 4 و فرزندان (نوه‌های) آن، تمام سطوری که در جدول TreePaths دارای فیلد descendant با مقدار 4 هستند، حذف می‌کنیم. علاوه بر این باید نودهایی که به عنوان descendant به فیلد گرهی 4، ارجاع داده می‌شوند نیز باید حذف گردد:

```
DELETE FROM TreePaths
WHERE descendant IN (SELECT descendant
FROM TreePaths
WHERE ancestor = 4);
```

دقت کنید وقتی گره‌ای را حذف می‌کنیم، بدان معنی نیست که خود گره (نظر) را حذف می‌کنیم. البته این برای مثال نظر و پاسخ آن مقداری عجیب است ولی در مثال کارمندان در چارت سازمانی امری معمول است. هنگامی که ارتباطات یک کاربر را تغییر می‌دهیم، از حذف در جدول TreePaths استفاده می‌کنیم و این قضیه که ارتباطات کارمندان در جدول جداگانه‌ای ذخیره شده است به ما انعطاف‌پذیری بیشتری می‌دهد.

برای جابجایی یک زیردرخت از مکانی به مکان دیگری در درخت، سطرهایی که ancestor گرهی بالایی زیردرخت را برمی‌گردانند و فرزندان آن گره را حذف می‌کنیم. برای مثال برای جابجایی گرهی شماره‌ی 6 به عنوان فرزند گرهی شماره‌ی 4 و قرار دادن آن به عنوان فرزند گرهی شماره‌ی 3، این چنین عمل می‌کنیم. فقط باید حواسمن جمع باشد سطري که گرهی شماره‌ی 6 به خودش ارجاع داده است را حذف نکنیم:

```
DELETE FROM TreePaths
WHERE descendant IN (SELECT descendant
FROM TreePaths
WHERE ancestor = 6)
AND ancestor IN (SELECT ancestor
FROM TreePaths
WHERE descendant = 6
AND ancestor != descendant);
```

آن‌گاه این زیردرخت جدا شده را با اضافه کردن سطرهایی که با ancestor مکان جدید و descendant زیردرخت، منطبق هستند، به جدول اضافه می‌کنیم:

```
INSERT INTO TreePaths (ancestor, descendant)
SELECT supertree.ancestor, subtree.descendant
FROM TreePaths AS supertree
CROSS JOIN TreePaths AS subtree
WHERE supertree.descendant = 3
AND subtree.ancestor = 6;
```

روش Closure Table آسان‌تر از روش مجموعه‌های تودرتو است. هر دوی آن‌ها روش‌های سریع و آسانی برای ایجاد پرس‌وجو برای نیاکان و نوه‌ها دارند. ولی Closure Table برای نگهداری اطلاعات سلسله مراتب آسان‌تر است. در هر دو طراحی ایجاد پرس‌وجو در فرزندان و پدر بلافصل سرراست‌تر از روش‌ای لیست مجاورت و شمارش مسیر می‌باشد. می‌توان عملکرد Closure Table را برای ایجاد پرس‌وجو روی فرزندان و پدر بلافصل را آسان‌تر نیز نمود. اگر فیلد path_length را به جدول TreePaths اضافه نماییم این کار انجام می‌شود. path_length گره‌ای که به خودش ارجاع می‌شود، صفر است. فرزند بلافصل هر گره 1، path_length نوی آن 2 می‌باشد و به همین ترتیب path_length را در هر سطر مقداردهی می‌کنیم. اکنون یا فتن فرزند گرهی شماره‌ی 4 آسان‌تر است:

```
SELECT *
FROM TreePaths
WHERE ancestor = 4 AND path_length = 1;
```

از کدام طراحی استفاده نماییم؟

در اینجا این سؤال مطرح است که ما باید از کدام طراحی استفاده نماییم. در پاسخ به این سؤال باید گفت که هر کدام از این روش‌ها نقاط قوت و ضعفی دارند که ما باید نسبت به عملیاتی که می‌خواهیم انجام دهیم از این طراحی‌ها استفاده کنیم. جدولی که در ادامه آمده است، مقایسه‌ای است میان میزان سهولت اجرای این طراحی‌ها در استفاده از پرس‌وجوهای متفاوت.

Design	Tables	Query Child	Query Tree	Insert	Delete	Referential Integration
Adjacency List	1	Easy	Hard	Easy	Easy	Yes
Recursive Query	1	Easy	Easy	Easy	Easy	Yes
Path Enumeration	1	Easy	Easy	Easy	Easy	No
Nested Sets	1	Hard	Easy	Hard	Hard	No
Closure Table	2	Easy	Easy	Easy	Easy	Yes

لازم به ذکر است در اینجا ستون سوم (Query Child) به معنای پرس‌وجوهایی است که با فرزندان کار می‌کند و ستون چهارم (Query Tree) به معنای پرس‌وجوهایی است که با کل درخت کار می‌کند، می‌باشد.

نظرات خوانندگان

نویسنده: علیرضا صبوری
تاریخ: ۱۴:۴۷ ۱۳۹۳/۰۵/۰۵

فکر میکنم عموما پرسش جوی بازگشتی اگر سایپورت بشه توسط دیتابیس بهترین روش همان لیست مجاورت هستش که مدیریت درخت رو برامون ساده میکنه و دیتابیس کنترل بستری رو هر نود ما داره. البته به غیر از مواردی خاص... ممنون از مطلب مفیدتون ولی سوالی که دارم اینه از نظر Performance مقایسه ای انجام شده که آیا استفاده از لیست بازگشتی چقدر از نظر سرعت در بازیابی اطلاعات با سایر روش‌ها تفاوت داره ؟ مبنعی اگر سراغ دارید ممنون میشم معرفی کنین.

در خیلی مواقع ملاحظه میشود که برای نمایش تعدادی از رکوردهای یک جدول در پایگاه داده، کل مقادیر موجود در آن توسط یک دستور select به دست می آید و صفحه بندی خروجی، به کنترلهای موجود سپرده میشود. اگر پایگاه داده ما دارای تعداد زیادی رکورد باشد، آن موقع است که دچار مشکل می شویم. فرض کنید به طور همزمان ۵ نفر (که تعداد زیادی نیستند) از برنامه ما که شامل ۱۰۰۰۰۰ سطر داده میباشد استفاده کنند و در هر صفحه، ۱۰ رکورد نمایش داده شود و صفحه بندی ما از نوع معقولی نباشد. در این صورت به جای اینکه با 5×10 رکورد داده را بارگزاری کنیم، 5×5 رکورد یعنی ۲۵ رکورد را برای به دست آوردن ۵ رکورد بارگزاری میکنیم. در زیر روشی شرح داده میشود که توسط آن، این سربار اضافه از روی برنامه و سرورهای مربوطه حذف شود. به stored procedure و توضیحات مربوط به آن توجه فرمایید :

```

CREATE PROCEDURE sp_PagedItems
(
    @Page int,
    @RecsPerPage int
)
AS
-- We don't want to return the # of rows inserted
-- into our temporary table, so turn NOCOUNT ON
SET NOCOUNT ON

--Create a temporary table
CREATE TABLE #TempItems
(
    ID int IDENTITY,
    Name varchar(50),
    Price currency
)

-- Insert the rows from tblItems into the temp. table
INSERT INTO #TempItems (Name, Price)
SELECT Name, Price FROM tblItem ORDER BY Price

-- Find out the first and last record we want
DECLARE @FirstRec int, @LastRec int
SELECT @FirstRec = (@Page - 1) * @RecsPerPage
SELECT @LastRec = (@Page * @RecsPerPage + 1)

-- Now, return the set of paged records, plus, an indication of we
-- have more records or not!
SELECT *, MoreRecords =
(
    SELECT COUNT(*)
    FROM #TempItems TI
    WHERE TI.ID >= @LastRec
)
FROM #TempItems
WHERE ID > @FirstRec AND ID < @LastRec

-- Turn NOCOUNT back OFF
SET NOCOUNT OFF

```

در این کد دو پارامتر از نوع integer تعریف میکنیم. اول پارامتر @Page که مربوط به شماره صفحه ای میباشد که قصد دارید آن را بارگزاری نمایید. دومین پارامتر با نام @RecsPerPage تعداد رکوردهایی است که هر بار میخواهید بارگزاری شوند. مثلاً اگر میخواهید هر بار ۱۵ عدد از رکوردها را نمایش دهید، این مقدار را باید برابر ۱۵ قرار دهیم. در مرحله بعد یک جدول موقت با نام TempItems# ساخته شده است که به طور موقت مقادیری را در حافظه نگه میدارد. نکته کلیدی که جلوتر از آن استفاده شده، ستون با نام ID است که از نوع auto-increment بوده و روی جدول موقت تعریف شده است. این ستون شناسه هر سطر را در خود نگه میدارد که به صورت اتوماتیک بالا میروند و جزو لاینکی از این نوع paging میباشد. پس از آن جدول موقت را توسط

رکوردهای جدول واقعی با نام `tblItem` توسط دستور `select` پر میکنیم.

در مرحله بعد شماره اولین و آخرین سطر مورد نظر را بر اساس پارامترهای ورودی محاسبه کرده و در متغیرهای `FirstRec`@ و `LastRec`@

برای استفاده از این کد فقط کافیست که پارامترهای ورودی را مقداردهی نمایید. مثلا اگر میخواهید در یک کنترل `Grid` از آن استفاده کنید باید ابتدا یک کوئری داشته باشید که تعداد کل سطرها را به شما بدهد و بر اساس این مقدار تعداد صفحات مورد نظر را به دست آورید. پس از آن با کلیک روی هر کدام از شماره صفحات آن را به عنوان مقدار به پارامتر مورد نظر بفرستید و از آن لذت ببرید.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۷/۲۲ ۱۹:۵۹

ضمن تشکر از شما. یک اصلاح کوچک: جدول موقتی ایجاد شده در پایان کار رویه ذخیره شده باید drop بشه.

نویسنده: محمد حسین عزتی
تاریخ: ۱۳۹۳/۰۷/۲۲ ۲۰:۱۸

از دقت شما به این نکته ظریف ممنونم
این موضوع در راستای آموزش عنوان مطلبش بود اما نکته شما جهت بالا بردن کیفیت و بهینه کردن کد مورد استفاده قرار میگیرد
و عدم drop مشکلی در رسیدن به هدف مورد نظر ایجاد نمی‌کند
متشرکم

نویسنده: امید
تاریخ: ۱۳۹۳/۰۷/۲۲ ۲۱:۲۰

در 2012 sql به بعد جهت صفحه بندی دستورات offset و fetch اضافه شده که از لحاظ Performance بهینه‌تر از باقی روش‌های میباشد. مقایسه صفحه بندی‌های مختلف

نویسنده: محمد حسین عزتی
تاریخ: ۱۳۹۳/۰۷/۲۲ ۲۱:۵۴

ممنونم بخاطر لینک مفیدی که در قسمت نظر ارسال نمودید
تنها نقطه ضعف این مقاله همینطور که خود شما هم متذکر شده اید این است که برای ورزش‌های بانک اطلاعاتی بعد از 2012 قابل استفاده است. هنوز بسیاری از نرم افزارها و سازمانهای ما هنوز با ورزش‌های قدیمی‌تر کار می‌کنند.

متشرکم

نویسنده: حمیدرضا کاظم نادی
تاریخ: ۱۳۹۳/۰۷/۲۳ ۱۱:۲۹

ممنون از مطلب خوبتون
به نظرم اگه جوری Sp را مینوشتید که یک ورودی متنی Query یا یک جدول موقت میگرفت و عمل Paging را روی اون انجام میداد
مطلوبتون بسیار کامل‌تر بود. مثلاً ورودی Sp به این صورت بود که ('select * from Tbl_1',1,10) بازهم ممنون

نویسنده: محمد حسین عزتی
تاریخ: ۱۳۹۳/۰۷/۲۳ ۱۲:۵۴

سلام
ممنونم از نظرتون
دوتا پارامتر داره از ورودی دریافت می‌کنه و هدف نحوه انجام صفحه بندی بوده

متشرکم

نویسنده: رحمت الله رضابی
تاریخ: ۱۳۹۳/۰۷/۲۳ ۱۳:۳۶

روشی بسیار قدیمی است و این روزها آنچنان کاربرد ندارد.

برای صفحه بندی :

- در 2008 از SQL Server استفاده می کنند . ROW_NUMBER
- در 2012 به بعد از OFFSET FETCH استفاده می کنند .

نویسنده: محسن خان
تاریخ: ۱۴:۱۰ ۱۳۹۳/۰۷/۲۳

تاریخچه ای از روش های مختلف صفحه بندی اطلاعات در SQL Server در این مقاله بحث شده به همراه بررسی کار آیی آن ها:

[Comparing performance for different SQL Server paging methods](#)

یکی از قابلیتهای SQL SERVER است. این قابلیت در واقع این امکان را برای ما فراهم می‌کند تا بر روی یک همزمان چندین کوئری را به صورت موازی ارسال کنیم. در این حالت برای هر کوئری یک سشن مجزا در نظر گرفته می‌شود.

مدل:

```
namespace EnablingMARS.Models
{
    public class Product
    {
        public int Id { get; set; }
        public string Title { get; set; }
        public string Desc { get; set; }
        public float Price { get; set; }
        public Category Category { get; set; }
    }

    public enum Category
    {
        Cate1,
        Cate2,
        Cate3
    }
}
```

:Context کلاس

```
namespace EnablingMARS.Models
{
    public class ProductDbContext : DbContext
    {
        public ProductDbContext() : base("EnablingMARS") {}
        public DbSet<Product> Products { get; set; }
    }
}
```

ابتدا یک سطر جدید را توسط کد زیر به دیتابیس اضافه می‌کنیم:

```
MyContext.Products.Add(new Product()
{
    Title = "title1",
    Desc = "desc",
    Price = 4500f,
    Category = Category.Cate1
});
MyContext.SaveChanges();
```

اکنون می‌خواهیم قیمت محصولاتی را که در دسته‌بندی Cate1 قرار دارند، تغییر دهیم:

```
foreach (var product in _dvContext.Products.Where(category => category.Category == Category.Cate1))
{
    product.Price = 50000;
    MyContext.SaveChanges();
}
```

خوب؛ اکنون اگر برنامه را اجرا کنیم با خطای زیر مواجه می‌شویم:

There is already an open DataReader associated with this Command which must be closed first.

این استثناء زمانی اتفاق می‌افتد که بر روی نتایج حاصل از یک کوئری دیگر را ارسال کنیم. البته استثنای صادر شده بستگی به کوئری دوم شما دارد ولی در حالت کلی و با مشاهده Stack Trace، پیام فوق نمایش داده می‌شود. همانطور که در کد بالا ملاحظه می‌کنید درون حلقه‌ی `forach` ما به پراپرتی `Price` دسترسی پیدا کرده‌ایم، در حالیکه کوئری اصلی ما هنوز فعال (Active) است. MARS در اینجا به ما کمک می‌کند که بر روی یک `Connection`، بیشتر از یک کوئری فعال داشته باشیم. در حالت عادی Entity Framework Code First این ویژگی را به صورت پیش‌فرض برای ما فعال نمی‌کند. اما اگر خودمان کانکشن استرینگ را اصلاح کنیم، این ویژگی SQL SERVER 5 فعال می‌گردد. برای حل این مشکل کافی است به کانکشن استرینگ، `MultipleActiveResultSets=true` را اضافه کنیم:

```
"Data Source=(LocalDB)\v11.0;Initial Catalog=EnablingMARS; MultipleActiveResultSets=true"
```

لازم به ذکر است که این قابلیت از نسخه SQL SERVER 2005 به بالا در دسترس می‌باشد. همچنین در هنگام استفاده از این قابلیت می‌بایستی موارد زیر را در نظر داشته باشید: وقتی کانکشنی در حالت MARS برقرار می‌شود، یک سشن نیز همراه با یکسری اطلاعات اضافی برای آن ایجاد شده که باعث ایجاد Overhead خواهد شد.

دستورات مارس [thread-safe](#) نیستند.

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۱۰:۳۲ ۱۳۹۳/۰۷/۲۴

با تشکر. یک سؤال: اگر بجای `dvContext.Products.Where` بنویسیم `dvContext.Products.Include(x=>x.Category)` باز هم این مشکل هست؟

نویسنده: سیروان عفیفی
تاریخ: ۱۰:۴۱ ۱۳۹۳/۰۷/۲۴

بله می شود به صورت eager loading این مشکل را نیز حل کرد. ولی مسئله performance را نیز باید در نظر داشته باشد.
البته می بایست به این صورت نوشته شود:

```
foreach (var product in _dvContext.Products.Where(x => x.Category.CatName == "Cat1").Include(x =>
x.Category).ToList())
{
    product.Price = 50000;
    _dvContext.SaveChanges();
}
```

حالت فوق در صورتی است که یک Category با نام Navigation property داشته باشد.

مدتی قبل مطلبی را در مورد کتابخانه‌ی ویژه SQL Server که یک T-SQL Parser است، در [این سایت مطالعه کردید](#). در این قسمت، همان مطلب را به نحو بهتر و ساده‌تری بازنویسی خواهیم کرد. مشکلی که در دراز مدت با SQLDom وجود خواهد داشت، مواردی مانند `CreateProcedureStatement` و `SelectStarExpression` و امثال آن هستند. این‌ها را از کجا باید تشخیص داد؟ همچنین مراحل بررسی این اجزاء، نسبتاً طولانی هستند و نیاز به یک راه حل عمومی‌تر در این زمینه وجود دارد.

راه حلی برای این مشکل در مطلب «[XML 'Visualizer' for the TransactSql.ScriptDom parse tree](#)» ارائه شده‌است. در اینجا تمام اجزای `TSqlFragment` مورد بررسی و استخراج قرار گرفته و نهایتاً یک فایل XML از آن حاصل می‌شود. اگر نکات ذکر شده در این مقاله را تبدیل به یک برنامه با استفاده مجدد کنیم، به چنین شکلی خواهیم رسید:

```

1 <?xml version="1.0" encoding="utf-16"?>
2 <TSqlScript memberName="root">
3   <TSqlBatch memberName="Batches">
4     <SelectStatement memberName="Statements">
5       <QuerySpecification memberName="QueryExpression">
6         <UniqueRowFilter.NotSpecified />
7         <SelectStarExpression memberName="SelectElements">
8           <StartOffset>7</StartOffset>
9           <FragmentLength>1</FragmentLength>
10          <StartLine>1</StartLine>
11          <StartColumn>8</StartColumn>
12          <FirstTokenIndex>2</FirstTokenIndex>

```

این برنامه را از اینجا می‌توانید دریافت کنید:

[DomToXml.zip](#)

همانطور که در تصویر مشاهده می‌کنید، اینبار به سادگی، `SelectStarExpression` قابل تشخیص است و تنها کافی است در T-SQL پردازش شده، به دنبال `SelectStarExpression`ها بود. برای اینکار جهت ساده شدن آنالیز می‌توان با ارث بری از کلاس پایه `TSqlFragmentVisitor` شروع کرد:

```

using System;
using System.Linq;
using Microsoft.SqlServer.TransactSql.ScriptDom;

namespace DbCop
{
    public class SelectStarExpressionVisitor : TSqlFragmentVisitor
    {
        public override void ExplicitVisit(SelectStarExpression node)

```

```
        {
            Console.WriteLine(
                "`Select *` detected @StartOffset:{0}, Line:{1}, T-SQL: {2}",
                node.StartOffset,
                node.StartLine,
                string.Join(string.Empty, node.ScriptTokenStream.Select(x => x.Text)).Trim());
            base.ExplicitVisit(node);
        }
    }
```

در کلاس پایه `TSqlFragmentVisitor` به ازای تمام اشیاء شناخته شده‌ی `ScriptDom`، یک متد `ExplicitVisit` قابل بازنویسی در نظر گرفته شده‌است. در اینجا برای مثال نمونه‌ی `SelectStarExpression` آن را بازنویسی کرده‌ایم.

```
مرحله‌ی بعد، اجرای این کلاس Visitor است:
```

```
public static class GenericVisitor
{
    public static void Start(string tSql, TSqlFragmentVisitor visitor)
    {
        IList<ParseError> errors;
        TSqlScript sqlFragment;
        using (var reader = new StringReader(tSql))
        {
            var parser = new TSql120Parser(initialQuotedIdentifiers: true);
            sqlFragment = (TSqlScript)parser.Parse(reader, out errors);
        }

        if (errors != null && errors.Any())
        {
            var sb = new StringBuilder();
            foreach (var error in errors)
                sb.AppendLine(error.Message);

            throw new InvalidOperationException(sb.ToString());
        }
        sqlFragment.Accept(visitor);
    }
}
```

در اینجا متد Accept کلاس TSq1120Parser، امکان پذیرش یک Visitor را دارد. به این معنا که Parser در حال کار، هر زمانیکه در حال آنالیز قسمتی از T-SQL دریافتی بود، نتیجه را به اطلاع یکی از متدهای کلاس پایه TSq1FragmentVisitor نیز خواهد رساند. بنابراین دیگر نیازی به نوشتتن حلقه و بررسی تک تک اجزای خروجی TSq1120Parser نیست. اگر نیاز به بررسی SelectStarExpression داریم، فقط کافی است Visitor آن را طراحی کنیم.

مثالی از نحوهی استفاده از کلاس GenericVisitor فوق را در اینجا ملاحظه می‌کنید:

```
var tsq1 = @"
WITH ctex AS (
SELECT * FROM sys.objects
)
SELECT * FROM ctex";
GenericVisitor.Start(tsq1, new SelectStarExpressionVisitor());
```

عنوان:	اتصال به SQL از راه دور (Remote) و یا به یک سرور در شبکه
نویسنده:	میثم خوشبخت
تاریخ:	۱۴:۱۵ ۱۳۹۳/۰۹/۱۱
آدرس:	www.dotnettips.info
گروه‌ها:	SQL Server

در نرم افزارهای بزرگ و چند کاربره، اتصال به بانک اطلاعاتی کامپیوتر سرور، یکی از نیازهای اساسی برنامه نویسان محسوب می‌گردد. در این بخش با دو اصطلاح بسیار مهم سروکار داریم.

1. کلاینت (Client) : منظور از کلاینت کامپیوتری است که میخواهد به سرور متصل گردد و از SQL کامپیوتر سرور خدماتی را دریافت نماید.

2. سرور (Server) : کامپیوتری است که میخواهیم به آن متصل شویم و داده‌ها را بصورت متمرکز بر روی آن ذخیره و بازیابی نماییم.

به دو روش می‌توان به سرور متصل شد:

Windows Authentication .1

در این روش جهت اتصال به بانک اطلاعاتی، کامپیوتر مبدأ یا Client باید عضو شبکه‌ای باشد که Server در آن وجود دارد. در واقع برای شبکه‌هایی استفاده می‌شوند که دارای Domain می‌باشند و Client به عنوان یک کاربر شناخته شده در سرور تعریف شده است.

SQL Authentication .2

در این روش کلاینت به عنوان یک کاربر یا Login در SQL تعریف شده است و دارای نام کاربری و رمز عبور می‌باشد.

جهت اتصال از راه دور به یک سرور دارای SQL، باید تنظیمات زیر را برای کامپیوتر سرور انجام دهیم:

1. به SQL Server کامپیوتر سرور متصل شوید.

2. در پنجره Object Explorer بر روی نام سرور (اولین آیتم موجود در لیست) کلیک راست کنید و گزینه Properties را انتخاب نمایید.

3. در پنجره ظاهر شده (Server Properties) و در قسمت Select a page (سمت چپ پنجره) بر روی Security کلیک کنید.

4. در سمت راست پنجره گزینه SQL Server and Windows Authentication mode را انتخاب کنید.

5. دکمه OK را انتخاب کنید. پنجره پیغامی مبنی بر Restart سرور نمایش داده می‌شود. این پنجره را تایید کنید.

6. مجدداً بر روی نام سرور کلیک راست کنید و گزینه Restart را انتخاب نموده و در پیغام ظاهر شده Yes را انتخاب نمایید.

تا به اینجا سرور آماده پذیرش اتصال از راه دور بصورت SQL Authentication می‌باشد. حال نوبت به تعریف یک Login می‌باشد تا توسط این Login بتوانید به سرور از راه دور متصل شوید. مراحل زیر را برای تعریف Login دنبال کنید:

1. در پنجره Object Explorer به مسیر Security > Logins بروید.

2. بر روی پوشه Logins کلیک راست نموده و گزینه New Login... را انتخاب نمایید.

3. در پنجره ظاهر شده در بخش Login name نامی را به کاربر اختصاص دهید. (به عنوان مثال user1)

4. گزینه SQL Server authentication را انتخاب نموده و در بخش Password و Confirm password رمز عبوری را به این کاربر اختصاص دهید. (به عنوان مثال abc123)

5. گزینه Enforce password policy را از حالت انتخاب خارج کنید تا رمز عبور را از قید سیاستهای رمزگذاری ویندوز خارج کنید.

6. در قسمت Select a page (سمت چپ پنجره) بر روی Server Roles کلیک کنید.

7. در سمت راست پنجره گزینه sysadmin یا هر نوع دسترسی دیگری را که مایل هستید انتخاب نمایید. توجه: با انتخاب sysadmin کاربر ایجاد شده به کل سرور و بانک‌های اطلاعاتی دسترسی کامل يا Admin دارد. اگر نمیخواهید کاربر چنین دسترسی داشته باشد، در بخش فوق فقط گزینه public انتخاب شده باشد.

8. در قسمت Select a page (سمت چپ پنجره) بر روی User Mapping کلیک کنید. در این بخش نحوه دسترسی کاربر را به بانک‌های اطلاعاتی موجود، مشخص می‌کنیم.

9. در سمت راست پنجره و در بخش Users mapped to this login یک یا چند بانک اطلاعاتی را که میخواهید توسط این Login قابل دسترسی باشند را انتخاب نمایید.

10. پس از انتخاب هر بانک اطلاعاتی، در قسمت پایین (Database role membership for:) نوع دسترسی کاربر به آن را انتخاب کنید. در اینجا من db_owner را انتخاب می‌کنم تا کاربر دسترسی کامل به بانک اطلاعاتی انتخاب شده را داشته باشد.

11. دکمه OK را انتخاب کنید تا Login مورد نظر ساخته شود.

حالا می‌توانید از راه دور و حتی از روی خود سرور با کاربر ایجاد شده به سرور متصل شوید. برای این منظور SQL را Disconnect نمایید و یا یکبار (SSMS) را ببندید و دوباره اجرا نمایید. در پنجره Connect to Server اطلاعات زیر را وارد نمایید:

Server name: نام یا IP سرور (به عنوان مثال 192.168.0.1)

Authentication: انتخاب گزینه SQL Server Authentication

Login: طبق مثال user1

Password: طبق مثال abc123

پس از ورود با مشخصات فوق فقط می‌توانید به بانک اطلاعاتی دسترسی باشید که در قسمت User Mapping انتخاب کرده بودید. اگر sysadmin را انتخاب کرده باشید به تمامی بانک‌های اطلاعاتی موجود دسترسی دارید.

برخی مشکلات اتصال از راه دور

ممکن است در زمان اتصال از راه دور با مشکل عدم امکان اتصال به سرور مواجه شوید. برای این منظور و اطمینان از صحت تنظیمات سرور، موارد زیر را در سرور بررسی نمایید تا بدرستی تنظیم شده باشند:

1. به مسیر Start > All Programs > Microsoft SQL Server 2008/2005 > Configuration Tools > SQL Server

مراجعه کنید و موارد زیر را بررسی نمایید:

1. بر روی SQL Server Services کلیک کنید و در سمت راست پنجره بررسی کنید که ستون State مربوط به SQL Server و Browser در وضعیت Running باشد.
2. بر روی آیتم‌های زیر مجموعه SQL Server Network Configuration کلیک کنید و در سمت راست پنجره بررسی کنید که آیتم‌های TCP/IP و Shared Memory ، Named Pipes Enabled باشند.
3. بر روی آیتم SQL Native Client Configuration > Client Protocols کلیک کنید و در سمت راست پنجره بررسی کنید که آیتم‌های TCP/IP و Shared Memory ، Named Pipes Enabled باشند.
2. بررسی کنید که فایروال سیستم سرور غیر فعال باشد و یا SQL Server به برنامه‌های Trust فایروال اضافه شده باشد.

نظرات خوانندگان

نویسنده: علی یگانه مقدم
تاریخ: ۲۳:۳۹ ۱۳۹۳/۰۹/۱۱

این مقاله رو دیدم یاد موضوعی افتادم ، یادم هست موقعی که از ۲۰۰۵ sql server استفاده می کردیم موقع انتقال جداول از این بانک به بانک راه دور ، جداول انتقال می یافت ولی کلیدها خیر حالا یا من اشتباهی می کردم یا مشکل از طرف sql server بوده ، البته بعضی ها هم می گفتن همین مشکل رو دارند

من بیشتر از generate scripts برای انتقال استفاده می کنم که نسبت به اون روش انتقال جداول سریعتر عمل می کنه و برای همین دیگه توی نسخه های جدیدتر این موضوع رو تست نکردم

نویسنده: محمد طاهری
تاریخ: ۱۰:۴۵ ۱۳۹۳/۰۹/۱۵

سلام آقای مقدم

شما برای این کار زمانی که می خواهید Script از Database خود تهیه کنید با پس از مشخص کردن Object های مورد نظر از قبیل (Table,SP,...) باید گزینه Advance رو کلیک کنید و در پنجره پیش رو کلیه ترتیباتی که در خصوص Object ها وجود دارد را می توانید انجام دهید .

و در آخر موضوعی که شما مد نظرتان است را می توانید با True کردن گزینه مربوط به Index می توانید کلیدها را هم انتقال بدینید.

انشالله تو این هفته سعی می کنم مقاله ای کامل درباره Script در بانک اطلاعاتی در سایت قرار دهم.
سئوالی بود در خدمتم

نویسنده: علی یگانه مقدم
تاریخ: ۱۲:۰ ۱۳۹۳/۰۹/۱۵

ممnon آقای طاهری بابت پاسختون

منظور من انتقال جداول در حالت ویزادی بود ، یه امکان انتقال در حالت ویزادی برای جداول و تمامی آجکت ها هست و در اون این مشکل بود

برای همین من دیگه از اسکریپت استفاده می کنم که انتقالش به مراتب راحت تر و تنظیمات زیادی هم داره ، و گرنه در حالت اسکریپت مشکلی ندارم

مشکل از طریق ویزارد مربوطه هست که مثلا جداول رو انتخاب می کردم و بعد هم سرور مقصد و بعد جداول رو انتقال میداد ولی وقتی چک می کردی می دیدی کلیدها نیست

نحوه تعریف Linked Server و دریافت اطلاعات از سروری دیگر

عنوان: نحوه تعریف Linked Server و دریافت اطلاعات از سروری دیگر
نویسنده: میثم خوشبخت
تاریخ: ۱۴:۱۵ ۱۳۹۳/۰۹/۱۱
آدرس: www.dotnettips.info
گروهها: SQL Server

برخی مواقع شما نیاز دارید تا یک Query را بر روی یک سرور اجرا نمایید و این Query برخی اطلاعات خود را از سرور دیگر دریافت می‌نماید. در این صورت باید یک پل ارتباطی بین سرور جاری و سرور دیگر وجود داشته باشد تا بتوانید در یک Query به سرور دیگر متصل شوید و اطلاعاتی را دریافت نمایید. در حالت عادی یک Query فقط می‌تواند بر روی سرور جاری اجرا شده و اطلاعاتی را بازیابی نماید. اما اگر همین Query بخواهد به سرور دیگر متصل شود، آن سرور باید در سرور جاری بصورت Linked Server تعریف شده باشد.

به عنوان مثال:

من سروری با آدرس 192.168.0.1 دارم که دارای پایگاه داده‌ای با نام Salary می‌باشد. نام این سرور را A می‌گذارم. همچنین من سرور دیگری با آدرس 192.168.1.100 دارم که دارای پایگاه داده‌ای با نام Accounting است. نام این سرور را B می‌گذارم.

حالا می‌خواهم در سرور A یک Query بنویسم که جدول Payment را با اتصال به سرور B به جدول Document متصل نموده و نتیجه JOIN این دو جدول را نمایش دهد. به عنوان مثال:

```
SELECT * FROM Payment AS pay JOIN Document AS doc
ON pay.DocumentId = doc.Id
```

این Query به هیچ عنوان اجرا نخواهد شد. زیرا نمی‌تواند جدول Document را پیدا کند. برای این منظور باید سرور B را به سرور A معرفی کنیم که این کار از طریق Linked Server انجام خواهد شد.

نحوه ایجاد یک Linked Server

بر روی سیستم من دو نسخه از SQL نصب شده است. یکی Standard Edition و دیگری Express Edition . من می‌خواهم در نسخه Express به نسخه Standard Linked Server یک ایجاد کنم. بنابراین با اتصال به نسخه Standard مراحل زیر را طی می‌کنم:

1. یک New query ایجاد می‌کنم.

2. دستورات زیر را در Query ایجاد شده می‌نویسم:

```
sp_addlinkedserver 'MyServer', '', 'SQLNCLI', '.\sqlexpress'
```

توضیحات:

sp_addlinkedserver نام رویه‌ای است که یک Linked Server را ایجاد می‌نماید. پارامتر اول نام Linked Server را مشخص می‌نماید که جهت دسترسی به سرور دیگر مورد استفاده قرار می‌گیرد. پارامتر دوم Product Name می‌باشد که من خالی گذاشتیم. پارامتر سوم Provider Name یا نام فراهم کننده داده‌ای است. چون من می‌خواهم به یک سرور SQL متصل شوم (SQL Provider Native Client) را انتخاب کرم. اگر به منبع داده‌ای دیگری مثل MySQL ، Access ، Oracle و ... متصل می‌شوید باید

پارامتر چهارم نام یا IP سروری است که می‌خواهیم به آن لینک شویم.
دیگری را منتخب کنید.

3. با فشردن F5 یا منوی Execute این Query را اجرا کنید.

با اجرای موفقیت آمیز مراحل فوق باید عنوان MyServer را در مسیر Server Objects > Linked Server مشاهده کنید. در نسخه Express پایگاه داده‌ای با نام test دارم که شامل جدولی به نام tbl می‌باشد. با نوشتن زیر می‌توانم محتويات این جدول را مشاهده کنم:

```
SELECT * FROM MyServer.test.dbo.tbl
```

ممکن است جهت اتصال به سرور لینک شده نیاز به نام کاربری و رمز عبور داشته باشد. جهت تعریف نام کاربری و رمز عبور برای سرور لینک شده از دستورات زیر استفاده کنید:

```
sp_addlinkedsvrlogin 'MyServer',@rmtuser='user1', @rmtpassword='abc123'
```

توضیحات:

پارامتر اول نام Linked Server را به یک sp_addlinkedsvrlogin نام رویه‌ای است که نام کاربری و رمز عبور را اضافه می‌کند.

پارامتر دوم نام کاربری جهت اتصال به سرور لینک شده می‌باشد.

پارامتر سوم رمز عبور جهت اتصال به سرور لینک شده می‌باشد.

نظرات خوانندگان

نویسنده: بابک جهانگیری
تاریخ: ۱۶:۵۳ ۱۳۹۳/۰۹/۱۱

آیا هنگام ارسال رمز عبور از طریق کانکشن استرینگ برای اتصال به یک سرور اس کیو ال ، چه به صورت لینک سرور و چه به صورت اتصال از طریق یک آپلیکیشن ویندوزی یا وبی داده‌های رمز عبور به صورت کد شده و ایمن ارسال می‌شوند ؟

نویسنده: محمد رجب
تاریخ: ۹:۱ ۱۳۹۳/۰۹/۱۲

با سلام و احترام؛ ضمن سپاس از نوشتن این مطلب، نکات تکمیلی زیر برای خوانندگان ارائه می‌گردد:
پس از ایجاد Linked Server به طریقی که در مطلب فوق بدان اشاره گردید، به دو صورت می‌توان به اجرای دستورات در Server دیگر پرداخت. به شکل Distributed Query و یا استفاده از تابع () OPENQUERY .
برای مثال دو SQL Instance SQL1 و SQL2 موجود است که قصد داریم عملیاتی را از SQL2 با استفاده از امکان OpenQuery و Distributed Query در SQL1 انجام دهیم، برای مثال:

Distributed Query:

```
SELECT * FROM [SQL2].[test].[dbo].[emp]
```

OPENQUERY:

```
SELECT * from OPENQUERY([SQL2], 'SELECT * FROM [test].[dbo].[emp]')
```

در استفاده از OpenQuery باید جانب احتیاط را رعایت نمود، چنانچه قصد اجرای دستورات ساده‌ای را در Remote Server دارید، توصیه می‌شود از Distributed Query استفاده شود و اگر دستورات پیچیده و محاسباتی هستند استفاده از OPENQUERY هزینه کمتری دارد. همچنین در نظر داشته باشید چنانچه در دستورات مربوط به جدولی ستونی با دیتا تایپ XML وجود داشت قادر به اجرای دستورات به هر دو شکل نخواهد بود، در واقع کوئری‌های توزیع شده این مورد را پشتیبانی نمی‌کنند. برای مثال پس از اجرای دستورات زیر با پیغام خطاب زیر مواجه می‌شوید: «Xml data type is not supported in distributed queries»

```
SELECT * FROM [نام Linked Server].[AdventureWorks].[Person].[Person]
GO
SELECT * FROM OPENQUERY([نام Linked Server ],'SELECT * FROM [AdventureWorks].[Person].[Person]')
```

همچنین در نظر داشته باشید چنانچه در دستورات مربوط به جدولی ستونی با دیتا تایپ CLR وجود داشت قادر به اجرای دستورات به شکل OpenQuery خواهد بود ولی در صورت استفاده از Distributed Query با خطای مواجه می‌شوید با این پیغام «Objects exposing columns with CLR types are not allowed in distributed queries»

```
SELECT * FROM [نام Linked Server] .[AdventureWorks].[Person].[Address]
GO
SELECT * FROM OPENQUERY([نام Linked Server ] , 'SELECT * FROM [AdventureWorks].[Person].[Address]')
```

برای کسب اطلاعات بیشتر به لینک زیر مراجعه فرمائید:

[Distributed query \(Four-part\) or OPENQUERY when executing linked server queries in SQL Server](#)

نویسنده: محمد طاهری
تاریخ: ۱۰:۳۸ ۱۳۹۳/۰۹/۱۵

سلام آقای جهانگیری

- شما برای این کار یکسری تنظیمات را باید در Enterprise manager انجام بدهید
- 1- نصب Certificate در ویندوز خودتان (از طریق MMC)
 - 2- فعال کردن Certificate Manager در قسمت Enterprise Manager
 - 3- پارامتر Encrypt را هم در قسمت Connection String خود لحاظ کنید.
-

اگر سئوالی بود بنده در خدمتم.

در این مقاله سعی شده است به تفاوت‌های Function و Stored Procedure ها در SQL Server به صورت مختصر و مفید بپردازیم: Function ها حتما باید مقدار بازگشتی داشته باشند ولی مقدار بازگشتی در SP* ها به صورت اختیاری است. یک SP می‌تواند از صفر تا n خروجی داشته باشد. Function ها می‌توانند پارامترهای ورودی داشته باشند ولی SP ها علاوه بر پارامترهای ورودی، می‌توانند پارامتر خروجی نیز داشته باشند.

Function ها می‌توانند از داخل SP ها صدا زده شوند ولی SP ها قابل فراخوانی از داخل Function ها نیستند. Select, Delete, Insert Function ها تنها می‌توانند دستورات Select را اجرا کنند ولی SP ها می‌توانند تمامی دستورات را از قبیل ... اجرا نمایند.

Function ها قابل فراخوانی از طریق دستورات Select هستند ولی SP ها این قابلیت را ندارند. Function ها قابلیت استفاده به صورت مستقیم را در دستوراتی مانند Having، دارند ولی SP ها این گونه نیستند. Function ها در هر بار استفاده کامپایل می‌شوند لیکن SP ها می‌توانند یکبار کامپایل شده، آنها برای استفاده آتی نگهداری شوند. این مهمترین تفاوت این دو می‌باشد. استثناء و خطاهای در SP ها قابل دریافت توسط Try...Catch می‌باشند ولی در Function ها این قابلیت وجود ندارد. در SP ها می‌توان از Transaction ها استفاده کرده و آنها را مدیریت نمایید؛ لیکن Function ها این امکاناتی را ندارند. در کل به خاطر داشته باشیم اگر تفاوت امکانات متفاوت یک محصول را بهتر درک کنیم، می‌توانیم از آنها بهتر استفاده نماییم.

SP: Stored Procedure*

نظرات خوانندگان

نویسنده: سانتا
تاریخ: ۲۳:۵۸ ۱۳۹۳/۱۲/۱۸

ممکنه برای مورد 7 یک لینک معتبر ارایه کنید

نویسنده: م منفرد
تاریخ: ۰:۳۸ ۱۳۹۳/۱۲/۱۹

به نظر موارد زیر میتواند به شما کمک کند:
<http://stackoverflow.com/questions/1048572/how-to-precompile-stored-procedures-in-sql-server>
<http://www.sql-server-performance.com/2003/optimizing-sp-recompiles>
<http://blog.sqlauthority.com/2009/11/08/sql-server-stored-procedure-are-compiled-on-first-run-sp-taking-longer-to-run-first-time/>

نویسنده: محمد خورستدي
تاریخ: ۲۳:۲۱ ۱۳۹۳/۱۲/۱۹

در Function ها فقط میتوان از table variable استفاده کرد ولی در SP ها هم میتوان از Temp table ها و هم از table ها استفاده کرد.
variable function ها را میتوان در Join استفاده کرد ولی این قابلیت را ندارند(شاید مورد ۴)

نوع داده‌ی HierarchyID به همراه SQL Server 2008 برای کار با داده‌هایی با ساختار درختی ارائه شد. در حال حاضر هیچکدام ازORM‌های موجود، پشتیبانی رسمی را از این نوع داده به عمل نمی‌آورند؛ اما با توجه به سورس باز بودن Entity framework، یک Fork مستقل از آن تهیه شده است و این نوع داده‌ی جدید به همراه متدهای مرتبط با آن، به این Fork اضافه شده‌اند.

- اصل Fork [در اینجا](#)
- تاریخچه‌ی این Fork غیر رسمی [در اینجا](#)
- بسته‌ی نیوگت آن [در اینجا](#)

چون تیم EF در نگارش فعلی این کتابخانه حاضر به افزودن این نوع جدید نشده است، بنابراین بجای بسته‌ی اصلی Entity framework نیاز است بسته‌ی `EntityFrameworkWithHierarchyId` نصب کنید.

```
PM> install-package EntityFrameworkWithHierarchyId
```

یک تذکر مهم:

چون امضای دیجیتال این بسته، با امضای دیجیتال بسته‌ی اصلی EF یکی نیست، اگر پروژه‌ی شما صرفاً از EF استفاده می‌کند، مشکلی نخواهد داشت. اما اگر برای مثال از ASP.NET Identity کامپایل شده‌ی برای کار با EF اصلی استفاده کنید، پیام یافتن نشدن DLL مرتبط را دریافت خواهید کرد.

تعریف مدلی با خاصیتی از نوع جدید `HierarchyId`

```
public class Employee
{
  public int Id { get; set; }

  [Required, MaxLength(100)]
  public string Name { get; set; }

  [Required]
  public HierarchyId Node { get; set; } // نوع داده جدید
}
```

در اینجا مدلی را ملاحظه می‌کنید که از نوع داده‌ی جدید `HierarchyId` استفاده می‌کند. همانطور که عنوان شد این نوع در بسته‌ی [EntityFrameworkWithHierarchyId](#) موجود است.

تعریف `Context` و مقدار دهی اولیه‌ی آن

در این حالت `Context` برنامه به همراه تنظیمات اولیه‌ی `Migrations` آن یک چنین شکلی را پیدا خواهد کرد:

```
public class MyContext : DbContext
{
  public DbSet<Employee> Employees { get; set; }

  public MyContext()
    : base("Connection1")
  {
    this.Database.Log = log => Console.WriteLine(log);
  }
}

public class Configuration : DbMigrationsConfiguration<MyContext>
{
  public Configuration()
}
```

```

{
    AutomaticMigrationsEnabled = true;
    AutomaticMigrationDataLossAllowed = true;
}

protected override void Seed(MyContext context)
{
    if (context.Employees.Any())
        return;

    context.Database.ExecuteSqlCommand(
        "ALTER TABLE [dbo].[Employees] ADD NodePath as Node.ToString() persisted");
    context.Database.ExecuteSqlCommand(
        "ALTER TABLE [dbo].[Employees] ADD Level AS Node.GetLevel() persisted");
    context.Database.ExecuteSqlCommand(
        "ALTER TABLE [dbo].[Employees] ADD ManagerNode as Node.GetAncestor(1) persisted");
    context.Database.ExecuteSqlCommand(
        "ALTER TABLE [dbo].[Employees] ADD ManagerNodePath as Node.GetAncestor(1).ToString()
persisted");

    context.Database.ExecuteSqlCommand(
        "ALTER TABLE [dbo].[Employees] ADD CONSTRAINT [UK_EmployeeNode] UNIQUE NONCLUSTERED
(Node)");
    context.Database.ExecuteSqlCommand(
        "ALTER TABLE [dbo].[Employees] WITH CHECK ADD CONSTRAINT [EmployeeManagerNodeNodeFK] " +
        "FOREIGN KEY([ManagerNode]) REFERENCES [dbo].[Employees] ([Node])");

    context.Employees.Add(new Employee { Name = "Root", Node = new HierarchyId("/") });
    context.Employees.Add(new Employee { Name = "Emp1", Node = new HierarchyId("/1/") });
    context.Employees.Add(new Employee { Name = "Emp2", Node = new HierarchyId("/2/") });
    context.Employees.Add(new Employee { Name = "Emp3", Node = new HierarchyId("/1/1") });
    context.Employees.Add(new Employee { Name = "Emp4", Node = new HierarchyId("/1/1/1") });
    context.Employees.Add(new Employee { Name = "Emp5", Node = new HierarchyId("/2/1") });
    context.Employees.Add(new Employee { Name = "Emp6", Node = new HierarchyId("/1/2/") });

    base.Seed(context);
}
}

```

در اینجا نحوه‌ی تعریف رکوردهای جدید مبتنی بر HierarchyID را مشاهده می‌کنید که توسط آن‌ها تعدادی کارمند، در یک سازمان فرضی ثبت شده‌اند.

همچنین چند فیلد محاسباتی نیز بر اساس امکانات توکار SQL Server اضافه شده‌اند. متدهایی مانند ToString, GetLevel, GetAncestor و امثال آن جزئی از پیاده سازی توکار SQL Server هستند. همچنین این متدها توسط کتابخانه EntityFrameworkWithHierarchyId

کوئری نویسی

مرتب سازی رکوردها بر اساس HierarchyId آن‌ها

```

using (var context = new MyContext())
{
    Console.WriteLine("\ngetItems OrderByDescending(employee => employee.Node)");

    var employees = context.Employees.OrderByDescending(employee => employee.Node).ToList();
    foreach (var employee in employees)
    {
        Console.WriteLine("{0} {1}", employee.Id, employee.Node);
    }
}

```

با این خروجی

```

SELECT
    [Extent1].[Id] AS [Id],
    [Extent1].[Name] AS [Name],
    [Extent1].[Node] AS [Node]
    FROM [dbo].[Employees] AS [Extent1]

```

کار با نوع داده‌ی HierarchyID توسط Entity framework

```
ORDER BY [Extent1].[Node] DESC  
  
6 /2/1/  
3 /2/  
7 /1/2/  
5 /1/1/1/  
4 /1/1/  
2 /1/  
1 /
```

یافتن یک HierarchyId خاص و سپس یافتن کلیه‌ی فرزندان آن در یک سطح پایین‌تر

```
using (var context = new MyContext())  
{  
    Console.WriteLine("\nGetAncestor(1) of /1/");  
  
    var firstItem = context.Employees.Single(employee => employee.Node == new HierarchyId("/1/"));  
    foreach (var item in context.Employees.Where(employee => firstItem.Node ==  
employee.Node.GetAncestor(1)))  
    {  
        Console.WriteLine("{0} {1}", item.Id, item.Name);  
    }  
}
```

این کوئری را به این شکل نیز می‌توان عنوان کرد: یافتن یک HierarchyId و سپس یافتن کلیه نودهایی که والدشان (این HierarchyId) است. عدد یک در اینجا مشخص کننده‌ی Level یا سطح است.
با این خروجی:

```
SELECT TOP (2)  
    [Extent1].[Id] AS [Id],  
    [Extent1].[Name] AS [Name],  
    [Extent1].[Node] AS [Node]  
    FROM [dbo].[Employees] AS [Extent1]  
    WHERE cast('/1/' as hierarchyid) = [Extent1].[Node]  
  
SELECT  
    [Extent1].[Id] AS [Id],  
    [Extent1].[Name] AS [Name],  
    [Extent1].[Node] AS [Node]  
    FROM [dbo].[Employees] AS [Extent1]  
    WHERE (@p_linq_0 = ([Extent1].[Node].GetAncestor(1))) OR ((@p_linq_0 IS  
NULL) AND ([Extent1].[Node].GetAncestor(1) IS NULL))  
-- p_linq_0: '/1/' (Type = Object)  
  
4 Emp3  
7 Emp6
```

کوئری‌های فوق را می‌توان بجای استفاده از متدهای GetAncestor و IsDescendantOf استفاده از متدهای IsDescendantOf و GetLevel که زیر نیز نوشته‌اند:

```
var list = context.Employees.Where(  
    employee => employee.Node.IsDescendantOf(new HierarchyId("/1/")) &&  
    employee.Node.GetLevel() == 2).ToList();
```

با این خروجی SQL (یک کوئری بجای دو کوئری):

```
SELECT  
    [Extent1].[Id] AS [Id],  
    [Extent1].[Name] AS [Name],  
    [Extent1].[Node] AS [Node]  
    FROM [dbo].[Employees] AS [Extent1]  
    WHERE (([Extent1].[Node].IsDescendantOf(cast('/1/' as hierarchyid))) = 1)  
    AND (2 = ([Extent1].[Node].GetLevel()))
```

جابجا کردن نودها توسط متod GetReparentedValue

در کوئری ذیل، تمامی فرزندان ریشه‌ی /1/ یافت شده و سپس والد آن‌ها به صورت پویا تغییر داده می‌شود:

```
var items = context.Employees.Where(employee => employee.Node.IsDescendantOf(new HierarchyId("/1/")))
    .Select(employee => new
    {
        Id = employee.Id,
        OrigPath = employee.Node,
        ReparentedValue = employee.Node.GetReparentedValue(new HierarchyId("/1/"),
HierarchyId.GetRoot()),
        Level = employee.Node.GetLevel()
    }).ToList();

foreach (var item in items)
{
    Console.WriteLine("Id:{0}; OrigPath:{1}; ReparentedValue:{2}; Level:{3}", item.Id, item.OrigPath,
item.ReparentedValue, item.Level);
}
```

با این خروجی

```
SELECT
    [Extent1].[Id] AS [Id],
    [Extent1].[Node] AS [Node],
    [Extent1].[Node].GetReparentedValue(cast('/1/' as hierarchyid), hierarchyid::GetRoot()) AS [C1],
    [Extent1].[Node].GetLevel() AS [C2]
FROM [dbo].[Employees] AS [Extent1]
WHERE ([Extent1].[Node].IsDescendantOf(cast('/1/' as hierarchyid))) = 1

Id:2; OrigPath:/1/; ReparentedValue:/; Level:1
Id:4; OrigPath:/1/1/; ReparentedValue:/1/; Level:2
Id:5; OrigPath:/1/1/1/; ReparentedValue:/1/1/; Level:3
Id:7; OrigPath:/1/2/; ReparentedValue:/2/; Level:2
```

کدهای کامل این مثال را از اینجا می‌توانید دریافت کنید

[HierarchyIdTests.zip](#)