

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 پیگیری، مدیریت و اداره کردن برنامه‌های SQL Server Express 2008 استفاده برای مدیریت 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
SQL Server Management Studio Basic - یک ابزار مدیریتی ویژوال برای ساخت، ویرایش و مدیریت پایگاه‌های داده

SQL Server 2008 Express with Advanced Services

موتور پایگاه داده SQL Server برای ساخت، ذخیره سازی، بروز رسانی و واکشی داده
SQL Server Management Studio Basic - یک ابزار مدیریتی ویژوال برای ساخت، ویرایش و مدیریت پایگاه‌های داده
Full-Text Search - یک موتور قدرتمند و پرسرعت برای جستجوی داده‌های متن-فشرده

دو حالت برای نصب SQL Server وجود دارد:

نصب یک نمونه جدید

آپگرید

نصب SQL Server Express 2008 از طریق Command Prompt

(1) پس از [دانلود SQL Server Express 2008](#) با پارامتر /X آنرا 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: SQLExpr32_x86_enu.exe /x

نکته: اگر فایل یا سی دی نصب ویژوال استودیو 2010 را دارید می‌توانید فایل نصب SQL Server Express را در مسیر \WCU\SSE پیدا کنید.

(2) همانطور که می‌بینید یک مسیر برای Extract از شما خواسته شده. یک مسیر وارد 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 ایی که باید نصب شود.
 /SQLSYSADMINACCOUNTS (الزامی): مقررات لوگین برای اعضای با نقش "مدیر سیستم".
 /SQLSVCACCOUNT (الزامی): تعیین اکانتی که سرویس SQL Server را در Startup 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 حتما 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 آنها باید در یک خط باشند.
 برای اولین بهتر است از پارامتر /HideConsole استفاده نکنید تا خطای احتمالی رو 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: ۹۰۱ ۱۳۹۲/۰۷/۲۲

ممون از شما. برای کارهای سبک استفاده از [SQL Server CE](#) شاید مناسبتر باشد یا حتی [LocalDB](#).

نویسنده: جوادی
تاریخ: ۰۳ ۱۳۹۲/۰۷/۲۳

خواهش می‌کنم.

بله، بسته به سناریو می‌شود از پایگاه داده‌های سبکتر نیز استفاده کرد.

LocalDB متاسفانه محدودیت‌های زیادی دارد، مثلاً در XP ساپورت نمی‌شود، فقط توسط Sql server 2012 management studio 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 2012 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 های کوچکتر که معمولاً embedded هستند. مثل sqlite و شاید sql ce و بقیه.

نویسنده: جوادی
تاریخ: ۱۹:۵۹ ۱۳۹۲/۰۸/۰۸

من نسخه 2012 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:

۱۵:۲۶ ۱۳۹۲/۱۲/۰۲

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 میشه؟

یک حالتی هست که ویژوال استودیو از من می‌خواد نسخه اکسپرس رو نصب کنم، خوب نسخه ای با حجم و امکانات بالا رو نصب می‌کنم. یک حالتی هم هست که بصورت توکار در یک برنامه قرار داده میشه که کاربر نهایی در اجرای برنامه به مشک برخورد، 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:

۱۸:۵۴ ۱۳۹۲/۱۲/۰۲

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 تصمیم‌گیری می‌کنند. نگارش CE داخل پروسه برنامه اجرا میشه و تا 4 گیگ دیتا داخلش می‌تونید ذخیره کنید. نگارش اکسپرس تا 12 گیگ محدودیت داره. سایر نگارش‌ها محدودیتی ندارند. خلاصه باید به کسب و کار خودتون نگاه کنید. برنامه حسابداری هست که تعداد رکوردهاش تا آخر سال مثلا میشه 10 هزار رکورد؟ نگارش CE براش کافی هست.

نویسنده:

یاسین

تاریخ:

۱۱:۱۳ ۱۳۹۲/۱۲/۰۳

ممنون، برای کاربران نهایی که از ویندوز XP استفاده می‌کنند، آیا نسخه 2008 مناسب هست یا اینکه تا نسخه 2005 باید تنزل کرد؟

نویسنده:

وحید نصیری

تاریخ:

۱۱:۱۶ ۱۳۹۲/۱۲/۰۳

نگارش 2008 با XP سازگار است. نگارش‌های پس از آن خیر.

نویسنده:

جمال

تاریخ:

۱۸:۱۸ ۱۳۹۲/۱۲/۰۳

میشه لطفا 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:

۲۱:۴۱ ۱۳۹۲/۱۲/۰۳

بهترین کار برای نصب خودکار فایل‌های مورد نیاز جهت اجرای برنامه‌های مبتنی بر دات نت ، استفاده از نرم افزارهای ساخت

Setup می‌باشد. از قبیل

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 خواهید 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 اومد 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 SQL 2008

روش کار : 1- دریافت پارامتر ورودی به صورت رشته

2- درج عناوین اعداد، ارزش مکانی اعداد صحیح و اعشاری هرکدام در یک جدول

3- جدا کردن ارقام صحیح و اعشاری

4- جداکردن سه رقم سه رقم اعداد صحیح و انتقال آنها به جدول مربوطه

5- Join جداول عناوین و ارقام جدا شده

6- ارسال ارقام اعشاری به همین تابع

7- مشخص کردن ارزش مکانی رقم اعشار

8- 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 (0.213) 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));
```

```

INSERT INTO @NumberTitle (val,Title)
VALUES(0, ''), (1, 'چهار', 4), (2, 'سه', 3), (3, 'دو', 2), (4, 'یک', 1),
(5, 'پنج', 6), (6, 'شش', 7), (7, 'هفت', 8), (8, 'هشت', 9),
(9, 'نه', 10), (10, 'ده', 11), (11, 'یازده', 12), (12, 'دوازده', 13),
(13, 'سیزده', 14), (14, 'چهارده', 15), (15, 'پانزده', 16), (16, 'شانزده', 17),
(17, 'هفده', 18), (18, 'هجده', 19), (19, 'نوزده', 20), (20, 'بیست', 30),
(30, 'سی', 40), (40, 'چهل', 50), (50, 'پنجاه', 60), (60, 'شصت', 70),
(70, 'هفتاد', 80), (80, 'هشتاد', 90), (90, 'نود', 100), (100, 'صد', 200),
(200, 'دویست', 300), (300, 'سیصد', 400), (400, 'چهارصد', 500), (500, 'پانصد', 600),
(600, 'ششصد', 700), (700, 'هفتصد', 800), (800, 'هشتصد', 900), (900, 'نهمصد', 1000)

DECLARE @PositionTitle TABLE (id INT, Title NVARCHAR(100));
INSERT INTO @PositionTitle (id, title)
VALUES (1, ''), (2, 'هزار', 3), (3, 'میلیون', 4), (4, 'میلیارد', 5), (5, 'تریلیون', 6),
(6, 'کوادریلیون', 7), (7, 'کوینتیلیون', 8), (8, 'سیکستیلیون', 9), (9, 'سپتیلیون', 10),
(10, 'اکتیلیون', 11), (11, 'نونیلیون', 12), (12, 'دسیلیون', 13),
(13, 'آندسیلیون', 14), (14, 'دودسیلیون', 15), (15, 'تریڈسیلیون', 16),
(16, 'کوآتردسیلیون', 17), (17, 'کوینڈسیلیون', 18), (18, 'سیکسڈسیلیون', 19),
(19, 'سپنڈسیلیون', 20), (20, 'اکتوڈسیلیوم', 21), (21, 'نومڈسیلیون', 22)

DECLARE @DecimalTitle TABLE (id INT, Title NVARCHAR(100));
INSERT INTO @DecimalTitle (id, Title)
VALUES (1, 'دهم', 2), (2, 'صدم', 3), (3, 'هزارم', 4), (4, 'ده-هزارم', 5),
(5, 'صد-هزارم', 6), (6, 'میلیون ام', 7), (7, 'ده-میلیون ام', 8),
(8, 'صد-میلیون ام', 9), (9, 'میلیارد ام', 10), (10, 'ده-میلیارد ام', 11)

```

جداسازی رقم صحیح و اعشاری: عدد ورودی ممکن است حالت‌های مختلفی داشته باشد مثل: 0.000000 , 00002 , 123 , 234.4344000000000

بنابراین براساس ممیز، قسمت صحیح را از اعشاری جدا می‌کنیم. برای ورودی که با ممیز شروع شود، در ابتدا تابع بررسی می‌کنیم و عدد صفر را به رشته 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 قسمت ممیز باشد؛ ولی من ترجیح می‌دهم که از روش دیگری استفاده کنم:

برعکس 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 بعد از 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 کردم. پس به علت اضافه کردن کاراکتر، استارت را از 2 شروع کرده و دیگر نیازی به 1- نخواهیم داشت. با کد زیر قسمت صحیح و اعشاری را بدست می‌آوریم:

```

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 (مثلا 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
    END

    IF LEN(@pNumber) > 2
        SET @pNumber = LEFT(@pNumber, LEN(@pNumber) - 3)
    ELSE
        SET @pNumber = '0'
    END
END

```

سطری که تمام مقادیر آن صفر باشد برای ما بی ارزش محسوب می‌شود، 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 عناوین 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)), ' ', ','), ', ', ' و ')
+ ' ' + pt.title + ' و '
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 Val3, Val2, Val1 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 مثلاً 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 باشد، تابع مقدار NULL بر می‌گرداند (قسمت بررسی رشته ورودی) و هر رشته ای 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(' ممیز '+[dbo].[fnNumberToWord_Persian](@DecimalNumber) + ' '+@PTitle, '')  
RETURN @str
```

مثال: رشته ' 5445789240.54678000000000 ' رشتۀ

پنج میلیارد و چهارصد و چهل و پنج میلیون و هفتصد و هشتاد و نه هزار و دویست و چهل و چهار ممیز پنجاه و چهار هزار و ششصد و هفتاد و هشت صد-هزارم

[دانلود فایل](#)

نظرات خوانندگان

نویسنده: میثم چگینی
تاریخ: ۱۱:۴۱ ۱۳۹۲/۰۸/۰۱

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: ۱۰:۵۳ ۱۳۹۲/۰۸/۰۳

حق با شماست!

من چون از Collation نوع Persian_100_CI_AI 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.

ممنون

نویسنده: مهیار
تاریخ: ۲۰:۳۸ ۱۳۹۲/۱۱/۱۵

سلام،

ممنون 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 توسط تابع Replace سپس با تابع Rtrim تمام کاراکترهای Space آخر رشته را Remove می‌کنیم و مجدد کاراکترهای Space را برمیگردانیم به صفر.

نویسنده: مهیار
تاریخ: ۲۱:۳۷ ۱۳۹۲/۱۱/۱۵

کد مربوط به حلقه While اتان هم کمی ساده‌تر کردم مخصوصا اولین دستور حلقه که مربوط میشه به انتخاب سه رقم آخر رشته. ضمنا نیازی نیست که صراحتا 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
```

```
        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))

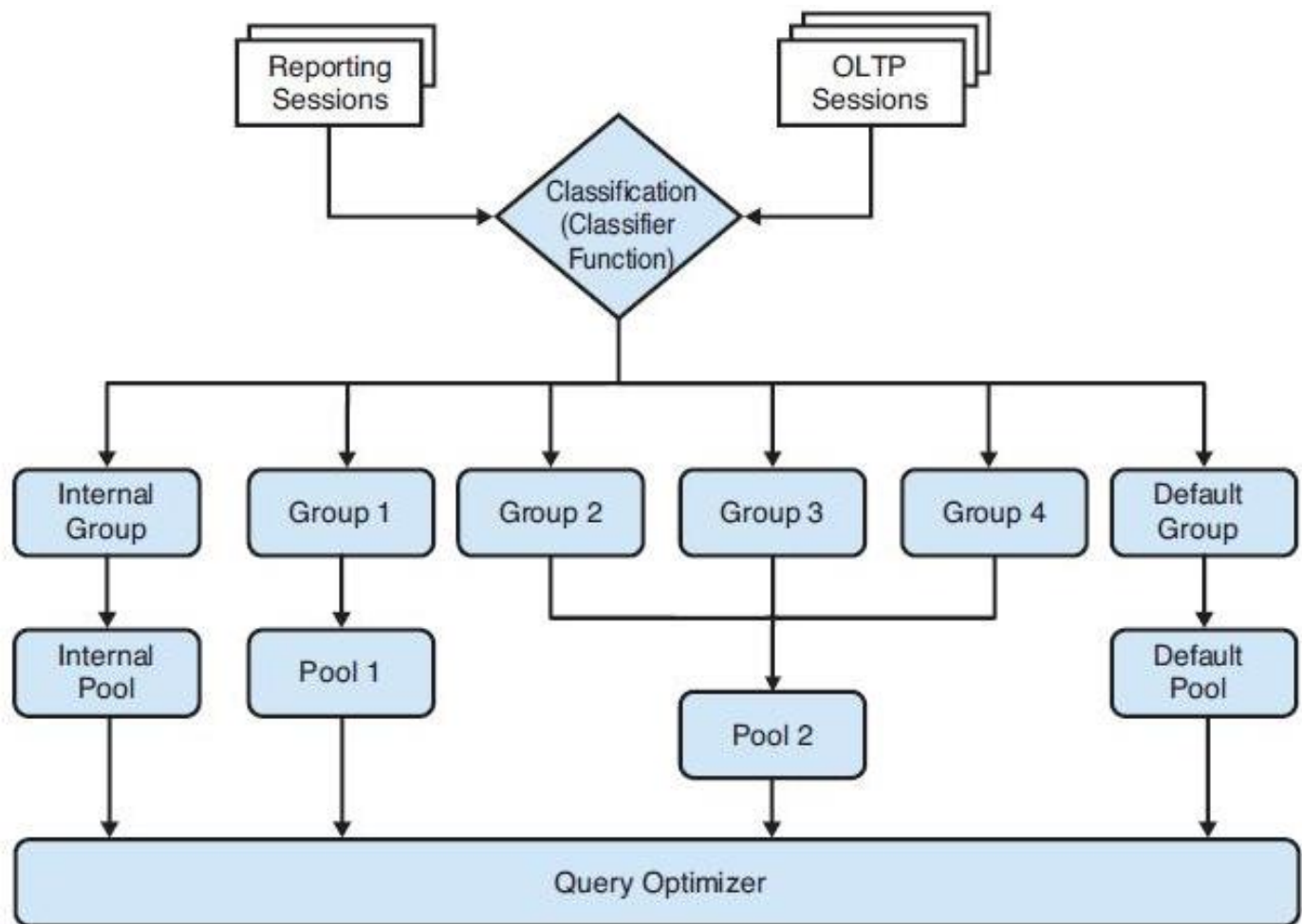
SET @pNumber= @IntegerNumber
```

مقدمه

Resource Governor، 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 پیش بینی فراهم می‌کند تا توسط فرآیندهای غیر قابل پیش بینی، تحت تاثیر منفی قرار نگیرد. با استفاده از Resource Governor، قادر خواهید 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 گزارش ها، دیگر تحت تاثیر منفی قرار نمی‌گیرد.

1- بررسی اجمالی Resource Governor:

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) ارزیابی می‌کند، به منظور تشخیص اینکه چگونه آنها را دسته بندی کند. پس از 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 اختصاص داده 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 Optimizer به Query برای اجرا داده می‌شود.



2- اجزای Resource Governor:

Resource Governor، از سه قسمت اصلی تشکیل شده است: Classification، Workload Groups و Resource Pools. درک این سه قسمت و چگونگی تعامل آنها به درک و استفاده از Resource Governor کمک می‌کند.

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 می‌شود. تابع نام یک Workload Group را برمی‌گرداند که Resource Governor از آن برای مسیر دهی Session به Workload Group مناسب استفاده می‌کند. هنگامی که Resource Governor پیکربندی می‌شود فرآیند ورود به سیستم برای یک Session شامل گام‌های زیر است:

- Login authentication
- LOGON trigger execution
- Classification

2-2- Workload Groups:

Workload Groups، ظروفي برای اتصالات مشابه هستند که با توجه به معیارهای طبقه بندی برای هر اتصال گروه بندی می‌شوند. Workload Groups همچنین مکانیسمی برای تجمیع نظارت بر روی منابع مصرفی فراهم می‌کند. Resource Governor 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 می‌شود. معیارهای طبقه بندی را برای گروه‌های داخلی نمی‌توانید تغییر دهید و همچنین هیچ یک از درخواست‌های کاربران را برای انتقال به گروه داخلی نمی‌توانید رده بندی کنید، با این حال بر گروه داخلی می‌توانید 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.
- کوششی برای رده بندی درخواستی به گروهی که وجود ندارد.
- خرابی کلی Classification

Resource Governor، در مجموع 20 عدد Workload Group را پشتیبانی می‌کند. از آنجائی که دو عدد از آنها برای Workload Groupهای داخلی و پیش فرض ذخیره شده اند در مجموع 18 عدد Workload Group تعریف شده توسط کاربر (user-defined) می‌توان تعریف نمود.

Resource pools 2-3:

Resource Pool (مخزن منابع)، نشان دهنده تخصیص منابع فیزیکی به SQL Server است. یک 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 حداکثر 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 حداکثر تخصیص CPU و حداقل و حداکثر تخصیص حافظه تنظیم می‌گردد. با تنظیم حداقل، در دسترس بودن منبع از مخزن تضمین می‌شود. از آنجائی که در هر رزرو حداقل منابع تداخلی نمی‌تواند وجود داشته باشد، مجموع مقادیر حداقل در تمام مخازن از 100% کل منابع Server نمی‌تواند تجاوز کند.
- مقدار حداکثر در محدوده بین حداقل و شامل 100% مقدار می‌تواند تنظیم گردد. تنظیم حداکثر نشان دهنده مقدار حداکثری است که یک Session می‌تواند مصرف کند، مادامی که منابع در دسترس باشند و توسط مخزن دیگر که با حداقل مقدار غیر صفر پیکربندی شده، استفاده نشود. هنگامی که یک مخزن با حداقل مقدار غیر صفر تعریف شده، مقدار حداکثر موثر از مخزن‌های دیگر دوباره تنظیم می‌شوند، در صورت لزوم حداکثر 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 Poo11 با مقدار حداقل 20% و مقدار حداکثر 100% تعریف شده، مخزن دیگری Poo12 با مقدار حداقل 50% و مقدار حداکثر 70% تعریف شده است. حداکثر مقدار موثر برای Poo11 برابر 50% است (100% منهای مقدار حداقل 50% مخزن Poo12) و حداکثر مقدار موثر برای Poo12، 70% است زیرا حداکثر مقداری است که پیکربندی شده است، گر چه 80% باقی می‌ماند.
- بخش مشترکی از مخزن (مقدارش بین مقدار حداقل و مقدار حداکثر موثر است) که برای تعیین مقدار منابع مورد استفاده است، توسط مخزن می‌تواند مصرف شود اگر منابعی موجود باشد و توسط مخازن دیگر مصرف نشده باشد. هنگامی که منابعی توسط 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:

Poo1A با حداقل مقدار 10% و حداکثر مقدار 100% تعریف می‌شود.

Poo1B با حداقل مقدار 35% و حداکثر مقدار 90% تعریف می‌شود.

Poo1C با حداقل مقدار 30% و حداکثر مقدار 80% تعریف می‌شود.

مقدار موثر Poo1A و مجموع در صد منابع به اشتراک گذاشته Poo1A به شرح زیر محاسبه خواهد شد:

(حداکثر مقدار Poo1A) - (حداقل مقدار Poo1B) - (حداقل مقدار Poo1C) = (حداکثر مقدار موثر Poo1A)

(حداکثر مقدار موثر Poo1A) - (حداقل مقدار Poo1A) = (اشتراک Poo1A)

جدول زیر مقدار حداکثر 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 حداکثر 100% را دارد و مصرف منابع توسط مخزن داخلی، از طریق تنظیمات در هر مخزن دیگر محدود یا کاسته نمی‌شود.

به عبارت دیگر حداکثر 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 Pool نمی‌تواند ایجاد یا حذف شود اما می‌تواند تغییر کند. Default Pool علاوه بر Default group می‌تواند شامل گروه‌های تعریف شده توسط کاربر (user-defined) نیز باشد.

3- پیکر بندی 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 Workload Groups و نسبت دادن آن به مخازن
- ایجاد Classification function
- ثبت Classification function به Resource Governor

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 workgroup 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:

3-3-1- Importance :

اهمیت نسبی (کم، متوسط یا بالا) Workload Group درون Resource Pool را تعیین می‌کند. اگر چندین Workload Group را در یک Resource Pool تعریف کنید این تنظیمات تعیین می‌کند که درخواست‌ها در عرض یک Workload Group در اولویت بالاتر یا پایین‌تری 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 workgroup‌هایی با اهمیت بالا، سه برابر بیشتر از workgroup‌های با اهمیت متوسط و نه برابر بیشتر از

workgroupهای کم اهمیت، مبادرت خواهد کرد.

3-3-2- Maximum Request :

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.

3-3-3- CPU Time :

حداکثر مقدار زمان پردازنده در ثانیه را مشخص می‌کند که یک درخواست درون Workload Group می‌تواند استفاده کند. تنظیم پیش فرض، صفر، به معنی نامحدود است.

3-3-4- % Memory Grant :

به صورت در صد، حداکثر مقدار اعطا 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 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 Workload Group توسط تعدادی از Sessionها به اشتراک گذاشته شود. توجه شود که تنظیم این مقدار به صفر از اجرای عملیات Hash Join 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 Queryهای همزمان در حال اجرا باشند، حافظه آزاد کافی اختصاص دهد.

3-3-5- Grant Time-out :

حداکثر زمان، به ثانیه، که یک query برای 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 time-out را با استفاده از محاسبات داخلی بر مبنای هزینه پرس و جو (query cost) با تعیین حداکثر زمان برآورد می‌کند.

3-3-6- Degree of Parallelism :

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، به یک Classification function نیاز است که شامل منطق ارزیابی اتصالات و نسبت دادن آنها به Workload Group مناسب است. Classification function برای هر اتصال Session جدید به SQL Server بکار می‌رود. هر Session در Workload Group نسبت داده شده به آن باقی می‌ماند تا زمانی که به پایان برسد، مگر اینکه صراحتاً به 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 Classification function همه اتصالات به Workload Group Default نسبت داده می‌شوند. Classification function یک نام workgroup که نوع آن SYSNAME است (که یک نام مستعار برای دیتا تایپ nvarchar 128 است). برمی گرداند. 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 Workload Group Default نسبت داده می‌شود.

منطق Classification function معمولاً مبتنی بر ویژگی‌های اتصال است و اغلب از طریق مقدار بازگشتی توابع سیستمی از قبیل:

HOST_NAME()، IS_SERVERROLEMEMBER()، IS_MEMBER()، SUSER_SNAME()، SUSER_NAME() و یا APP_NAME() نام Workload Group اتصال مشخص می‌شود. علاوه بر این توابع می‌توانید از ویژگی‌های توابع دیگر برای ساخت منطق رده بندی استفاده کنید. تابع LOGINPROPERTY() شامل دو ویژگی (DefaultDatabase و DefaultLanguage) می‌باشد که می‌تواند برای Classification function استفاده شود. علاوه تابع CONNECTIONPROPERTY() پروتکل‌ها و دسترسی به نقل و انتقالات در شبکه، همچنین جزئیات طرح احراز هویت، Local IP address و TCP Port و 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 یا APP_NAME()) در تابع رده بندی تان استفاده کنید، توجه داشته باشید این امکان وجود دارد مقادیر بازگردانده شده توسط این توابع توسط کاربران تغییر داده شوند، گر چه به طور کلی گرایش به استفاده از تابع APP_NAME() برای رده بندی اتصالات بیشتر است.

4- بررسی نمونه ای از پیکربندی Resource Governor

برای سادگی، در این قسمت مثالی ارائه می‌شود که از تابع SUSER_NAME() استفاده می‌کند: در گام نخست، دو Resource Pool ایجاد می‌شود (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 می‌شود (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 می‌شود (report_user و oltp_user) که در تابع رده بندی استفاده خواهند شد برای مشخص کردن این که اتصالات Seesion به کدام Workload Group نسبت داده شوند. پس از اضافه کردن Login ها به عنوان User ها به Database مورد نظر مان، در بانک اطلاعاتی Master تابع رده بندی (Classification function) را ایجاد می‌کنیم:

```

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() را در محیط SSMS 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 Resource Pool و Workload Group را تغییر دهید (برای مثال حداکثر استفاده CPU برای یک Resource Pool و یا درجه اهمیت یک Workload Group). متناوباً می توان از دستورات T-SQL استفاده نمود. نکته: پس از اجرای دستورات ALTER RESOURCE POOL یا ALTER WORKLOAD GROUP، 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 Workload Group شامل Session های فعال باشد، حذف Workload Group و یا جابجائی آن به یک Resource Pool متفاوت، هنگامی که دستور ALTER RESOURCE GOVERNOR RECONFIGURE برای اعمال نمودن تغییرات فراخوانی می شود، با خطا 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.

5-3- اصلاح Classification function:

اگر نیاز دارید تغییراتی در تابع رده بندی ایجاد نمائید، 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 نمائید. متناوباً می‌توان تابع رده بندی را جایگزین کرد با اجرای دستور ALTER RESOURCE GOVERNOR و فرستادن (passing) یک اسم متفاوت برای CLASSIFIER_FUNCTION، 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 Performance Monitor، events و Dynamic Management View (DMV) می‌توان Workload Group و Resource Pool را نظارت (Monitor) کرد. دو شی Performance برای این کار موجود است: SQL Server:Workload Group Stats و SQL Server:Resource Pool Stats. شکل زیر مربوط به پیکر بندی مثال مورد نظرمان می‌باشد:



7- نتیجه گیری

Resource Governor 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 غیر منتظره ای که به شدت و بطور قابل توجهی روی کارایی Server تاثیر منفی می‌گذارند.

ضمناً Resource Governor چندین مشکل بالقوه نیز عرضه می‌کند، برای مثال پیکربندی اشتباه 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 تجربه ای هستید و درک خوب و آشنائی خوبی با Workload هایی که روی بانک اطلاعاتی اجرا می‌شوند دارید، Resource Governor را بکار برید. 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 محدودیت های بارز :

- محدودیت منابع (Resource)، که به CPU و حافظه محدود می‌شوند. I/O Disk و منابع شبکه را در SQL Server 2008 نمی‌توان محدود کرد.
- استفاده از منابع برای Reporting Service، Analysis Service و Integration Service را نمی‌توان محدود کرد. در این نسخه محدودیت‌های منابع تنها روی هسته موتور بانک اطلاعاتی بکار برده می‌شود.
- محدودیت‌های Resource Governor روی یک SQL Instance تعریف و بکار برده می‌شود.

نظرات خوانندگان

نویسنده:

محمد رجبی

تاریخ:

۱۳:۳۷ 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 I/O نیز اضافه گردید. (تا پیش از ارائه نسخه 2014 محدودیت روی منابع تنها به CPU و حافظه خلاصه می شد)