

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 کنترل های 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 ۱۰×۵ رکورد داده را بارگزاری کنیم، ۱۰۰۰۰۰×۵ رکورد یعنی ۵۰۰۰۰۰ رکورد را برای به دست آوردن ۵۰ 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 می باشد 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 میرود و جزء لاینفکی 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 می‌ریزیم.

برای استفاده از این کد فقط کافیست که پارامترهای ورودی را مقداردهی نمایید. مثلاً 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:

۱۹:۵۹ ۱۳۹۳/۰۷/۲۲

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:

۲۰:۱۸ ۱۳۹۳/۰۷/۲۲

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 نمی کند
متشکرم

نویسنده:

امید

تاریخ:

۲۱:۲۰ ۱۳۹۳/۰۷/۲۲

در [sql 2012](#) به بعد جهت صفحه بندی دستورات offset و fetch اضافه شده که از لحاظ Performance بهینه تر از باقی روش های می باشد . [مقایسه صفحه بندی های مختلف](#)

نویسنده:

محمد حسین عزتی

تاریخ:

۲۱:۵۴ ۱۳۹۳/۰۷/۲۲

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 ورژن های بانک اطلاعاتی بعد از 2012 قابل استفاده است. هنوز بسیاری از نرم افزارها و سازمان های ما هنوز با ورژن های قدیمی تر کار می کنند.

متشکرم

نویسنده:

حمیدرضا کاظم نادی

تاریخ:

۱۱:۲۹ ۱۳۹۳/۰۷/۲۳

ممنون از مطلب خوبتون

به نظرم اگه جوری Sp را مینوشتید که یک ورودی متنی Query یا یک جدول موقت می گرفت و عمل Paging را روی اون انجام میداد مطلبتون بسیار کامل تر بود. مثلا ورودی Sp به این صورت بود که ('select * from Tb1_1',1,10) باز هم ممنون

نویسنده:

محمد حسین عزتی

تاریخ:

۱۲:۵۴ ۱۳۹۳/۰۷/۲۳

سلام

ممنونم از نظرتون

دوتا پارامتر داره از ورودی دریافت می کنه و هدف نحوه انجام صفحه بندی بوده

متشکرم

نویسنده:

رحمت اله رضایی

تاریخ:

۱۳:۳۶ ۱۳۹۳/۰۷/۲۳

روشی بسیار قدیمی است و این روزها آنچنان کاربرد ندارد.

برای صفحه بندی :

- در SQL Server 2008 از [ROW_NUMBER](#) استفاده می کنند.

- در SQL Server 2012 به بعد از [OFFSET FETCH](#) استفاده می کنند .

نویسنده:

محسن خان

تاریخ:

۱۳۹۳/۰۷/۲۳ ۱۴:۱۰

تاریخچه ای از روش های مختلف صفحه بندی اطلاعات در SQL Server در این مقاله بحث شده به همراه بررسی کارایی آن ها:

[Comparing performance for different SQL Server paging methods](#)