

پیش 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 SQL Server 2000 کاربرد داشت و هم‌اکنون استفاده از آن به دلیل کاهش بسیار کارایی منسوخ 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 کاراکتری 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 به‌کاربری 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، کارایی کاهش فراوانی خواهد یافت.

در SQL Server 2008 نوع داده‌ی جدیدی به نام FileStream به وجود آمد به این شکل که یک FileGroup از نوع Data در FileStream به پایگاه‌داده افزوده می‌شود و در واقع با یک پوشه در سیستم فایل 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 برعهده‌ی پایگاه داده‌ها خواهد بود. اندازه‌ی فایل‌ها در FileStream محدودیت‌های پیشین را نخواهد داشت و شما به اندازه‌ی حجم درایو هارددیسک 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 برعهده‌ی SQL Server است و از این دیدگاه بسیار ساده‌تر و کارآتر از FileSystem است. (برای آشنایی بیشتر با FileStream، [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 FileStream با hierarchyid و سیستم فایل ویندوز برای ارائه‌ی توانایی‌های نوین مدیریت BLOB در SQL Server است. FileTable همان‌گونه که از دو واژه‌ی تشکیل‌دهنده‌اش پیداست؛ هم‌زمان یک جدول و یک سیستم فایل معمولی است. FileTable به هر روی یک جدول از پایگاه‌داده‌های SQL Server است با یک تفاوت که ساختار آن از پیش تعریف شده است. ستون‌های FileTable و نوع داده‌ی آن از پیش توسط SQL Server مشخص شده است. ستون‌های تشکیل‌دهنده‌ی FileTable دربرگیرنده‌ی جدول زیر است:

نام ستون	Data Type	شرح
stream_id	Uniqueidentifier ROWGUIDCOL	شناسه 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 واپسین ویرایش
ast_access_time	datetimeoffset(7)	زمان واپسین دسترسی
is_directory	bit	۰ = فایل / ۱ = پوشه
is_offline	bit	
is_hidden		
is_readonly		
is_archive		
is_system		
is_temporary		

هر ردیف از FileTable نماینده ی یک فایل یا پوشه در File System است. ستون path_locator که از نوع hierarchyid است نشان دهنده ی مسیر یک فایل یا پوشه است. hierarchyid که از SQL Server 2008 معرفی شده است؛ بهترین نوع داده برای نگهداری ارتباط ساختار سلسله مراتبی مانند چارت سازمانی، درخت تجهیزات یک کارخانه و یا در همین نمونه درخت فایل ها و پوشه ها است. پس می توانیم از همه ی امکانات hierarchyid در FileTable نیز برخوردار شویم. این که این فایل به ترتیب در چه پوشه هایی قرار گرفته است یا این که این پوشه شامل چه فایل ها یا پوشه هایی خواهد بود. این که پوشه های هم فرزند پوشه ی جاری کدام است و یا توابع مربوط به جابه جایی فایل ها و پوشه ها.

دنباله دارد...

نظرات خوانندگان

نویسنده: امیر بختیاری
تاریخ: ۱۶:۲۶ ۱۳۹۲/۰۲/۱۷

با سلام و خسته نباشید
مطلب 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: ۱۷:۲۷ ۱۳۹۲/۰۲/۱۷

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 تارنما منتشر کنم.

نویسنده: ali.rezayee
تاریخ: ۱۹:۳۱ ۱۳۹۲/۰۲/۱۷

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 هست؟
ممنون از شما.

نویسنده: محسن خان
تاریخ: ۲۲:۵۸ ۱۳۹۲/۰۲/۱۷

در اصل [nvarchar هست](#) که به نظر در جدول فوق باید اصلاح بشه.

نویسنده: حامد قنادی
تاریخ: ۶:۵۳ ۱۳۹۲/۰۲/۱۸

بله. اشتباه از من بود. این فیلدها غیرقابل تغییر هستند ولی هیچ مشکلی هم با زبان پارسی ندارند.

نویسنده: سید امیر سجادی
تاریخ: ۲۰:۱۹ ۱۳۹۲/۰۹/۱۴

سلام. ممنون از مطلب مفیدتون. فقط یک مشکلی هست و اون عدم پشتیبانی توسط Linq2Sql هست. فکر کنم فقط با DataSet مشکلی نداشته باشه و اون هم بخاطر این هست که نوع hierarchyid به نوع nvachar(867) تبدیل میکنه.
لطفا اگه برای این مشکل راه حلی هست بنویسید که دوستان هنگام استفاده به مشکل برنخورند

نویسنده: محسن خان
تاریخ: ۲۰:۳۱ ۱۳۹۲/۰۹/۱۴

Linq2Sql که خیلی وقت هست با EF جایگزین شده. EF هم از نوعهای خاص دیتابیسها عموما پشتیبانی نمیکنه. بحث EF هم نیست. اکثر ORMهای موجود همینطور هستند.

نویسنده: سید امیر سجادی

تاریخ: ۲۳:۸ ۱۳۹۲/۰۹/۱۴

یعنی فقط باید از DataSet استفاده کرد !؟

نویسنده: محسن خان

تاریخ: ۱:۱ ۱۳۹۲/۰۹/۱۵

ADO.NET می‌تونه خروجی از نوع DataSet هم بده؛ ولی اسمش DataSet نیست. تمام ORM ها هم برفراز ADO.NET کار می‌کنند. بنابراین اگر تصور کنیم که ابداء نمی‌شود با این‌ها کار کرد، خیر. اینطور نیست. شما با ORM ها هم می‌تونید مستقیماً SQL بنویسید -[ef](#) [code-first-13](#) تا به قابلیت‌هایی دسترسی پیدا کنی که در ORM پیش بینی نشده.

نویسنده: سعید

تاریخ: ۲۳:۵۱ ۱۳۹۲/۱۰/۱۴

سلام

چطور امنیت فایل‌ها رو برقرار کنیم؟ منظورم اینه که کسی نتونه اونا رو پاک یا مشاهده و یا اضافه کنه، مگه اینکه ما بهش اجازه بدیم راهی برای این کار هست؟

نویسنده: محسن خان

تاریخ: ۰:۴۵ ۱۳۹۲/۱۰/۱۵

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: ۱۱:۱۱ ۱۳۹۲/۱۰/۲۲

من دارم از اپلیکیشن استفاده میکنم. از داخل برنامه درست هست، یعنی همونی که شما میگی. ولی اگه کسی از بیرون برنامه بخواد این کارو کنه چی؟ مستقیم بره تو پوشه و فایل‌ها رو تغییر بده. من میخوام بدونم sql میتونه جلوی این کارو هم بگیره؟

نویسنده: محسن خان

تاریخ: ۱۲:۱۲ ۱۳۹۲/۱۰/۲۲

- به چه نحوی از SQL Server استفاده می‌کنید؟ آیا سرور و برنامه دسکتاپ شما روی یک کامپیوتر هستند؟ برای اینکار بهتر است از SQL CE یا SQLite 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 عملاً نیازی 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 دسکتاپ.

- با این توضیح اگر کسی به سرور شما دسترسی دارد، آیا نمی‌تواند مثلاً اگر فایل‌ها در دیتابیس ذخیره می‌شدند، 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](#)

نویسنده: حسین حاجی بابایی

تاریخ: ۱۰:۸ ۱۳۹۴/۰۴/۰۳

استفاده از FileTable در EF <http://www.eidias.com/blog/2013/12/4/using-sql-filetable-in-entity-framework>

نویسنده: محسن خان
تاریخ: ۱۳۹۴/۰۴/۰۳ ۱۰:۲۶

SP نوشته. بله. همیشه مستقیماً SQL بنویسی منتها این اسمش کار با ORM نیست.