

ستون دیگر **stream_id** نام دارد که از نوع uniqueidentifier ROWGUIDCOL است. همان‌گونه که در یاد دارید، در FileStream 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 پوشه‌ها جابه‌جا نمی‌شدند می‌توانستید از هر دو ستون **path_locator** یا **stream_id** برای شناسایی یک رکورد از جدول بهره ببرید. ولی با جابه‌جایی یک فایل و یا به عبارت دیگر تغییر پدر آن در ساختار سلسله‌مراتبی، مقدار **path_locator** 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 پوشه است، بنابراین به ستونی نیاز داریم که بتوانیم این موضوع را نشان دهیم. بر این پایه از ستون **is_directory** بهره می‌بریم که 1 بودن آن نشان‌دهنده‌ی این است که این ردیف از جدول به یک پوشه ارجاع دارد. نام فایل یا پوشه 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 توسط آن به دست می‌آید. ستون پسین **file_stream** نام دارد که برای پوشه‌ها، 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 برعهده‌ی SQL Server است. افزون بر این 14 ستون، هر FileTable شامل سه ستون محاسباتی به شرح زیر است:

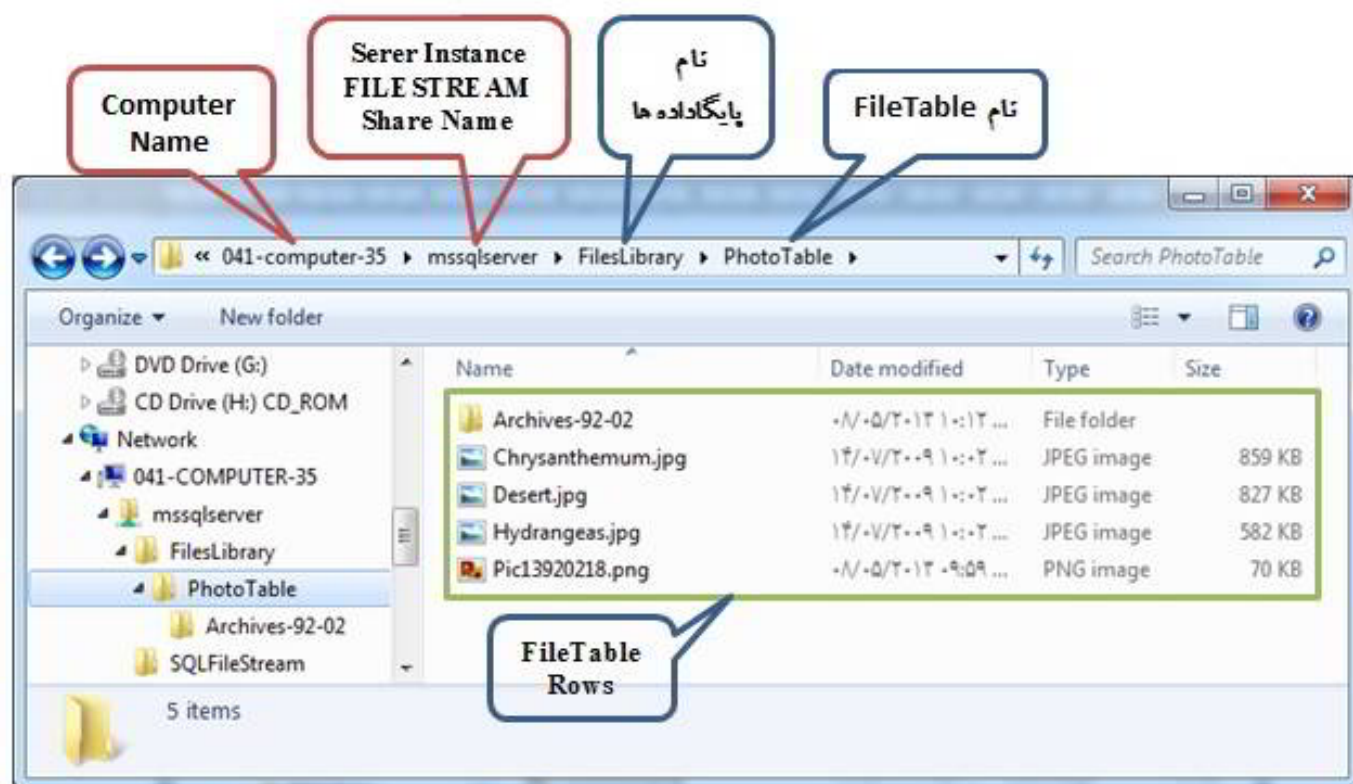
نام ستون	DataType	شرح
parent_path_locator	hierarchyid	مسیر به دست آمده از path_locator
file_type	nvarchar(255)	گونه‌ی فایل به دست آمده از name
cached_file_size	bigint	اندازه‌ی فایل به دست آمده از file_stream

ستون **parent_path_locator** نتیجه‌ی فراخوانی تابع **GetAncestor(1)** در ستون **path_locator** است که جهت به دست آوردن پوشه‌ی پدر یک فایل و پوشه 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 دارید در یک پوشه‌ی 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 مانند کشیدن و رهاکردن با Windows Explorer یا برنامه‌نویسی با System.IO.FileStream و API‌های ویندوز استفاده کنید. این‌چنین:

- ایجاد یک فایل یا پوشه در سیستم فایل -> افزودن یک ردیف به جدول

- افزودن یک ردیف به جدول -> ایجاد یک فایل یا پوشه در سیستم فایل



با کپی فایل‌ها در مسیر بالا، به صورت خودکار رکوردهای زیر در جدول PhotoTable 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	0xFD64A656CE5A91CFDB125242D522...	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 2012 راه کار سوم را پیشنهاد می‌کند. استفاده از File System در این روش FileStream به طور خودکار پر می‌شود.

پیش از ساخت یک FileTable بیان این نکته دارای اهمیت است که با کپی فایل‌ها و پوشه‌ها هیچ چیزی جدیدی به NTFS افزوده نمی‌شود بلکه محتوای فایل به FileStream افزوده می‌شود و SQL Server با بررسی همزمان FileStream و FileTable نمایشی از ردیف‌های FileTable به صورت یک پوشه‌ی اشتراکی نشان می‌دهد. این نکته پاسخی به این پرسش خواهد بود که آیا با استفاده از FileTable حجم پایگاه داده‌ها 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 دستورهایی 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')
```

FileTable به FileStream متکی است؛ بر این پایه پیش از ایجاد یک FileTable باید FileStream را روی پایگاه داده‌ها 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 FileTable روی یک پایگاه داده‌ی ساخته شده بند SET FILESTREAM را در دستور ALTER DATABASE بنویسید.) در این بند، از DIRECTORY_NAME برای نام‌گذاری یک پوشه برای پایگاه داده‌ها استفاده می‌کنیم. این پوشه در یک پوشه ریشه به نام SQL Server instance 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.

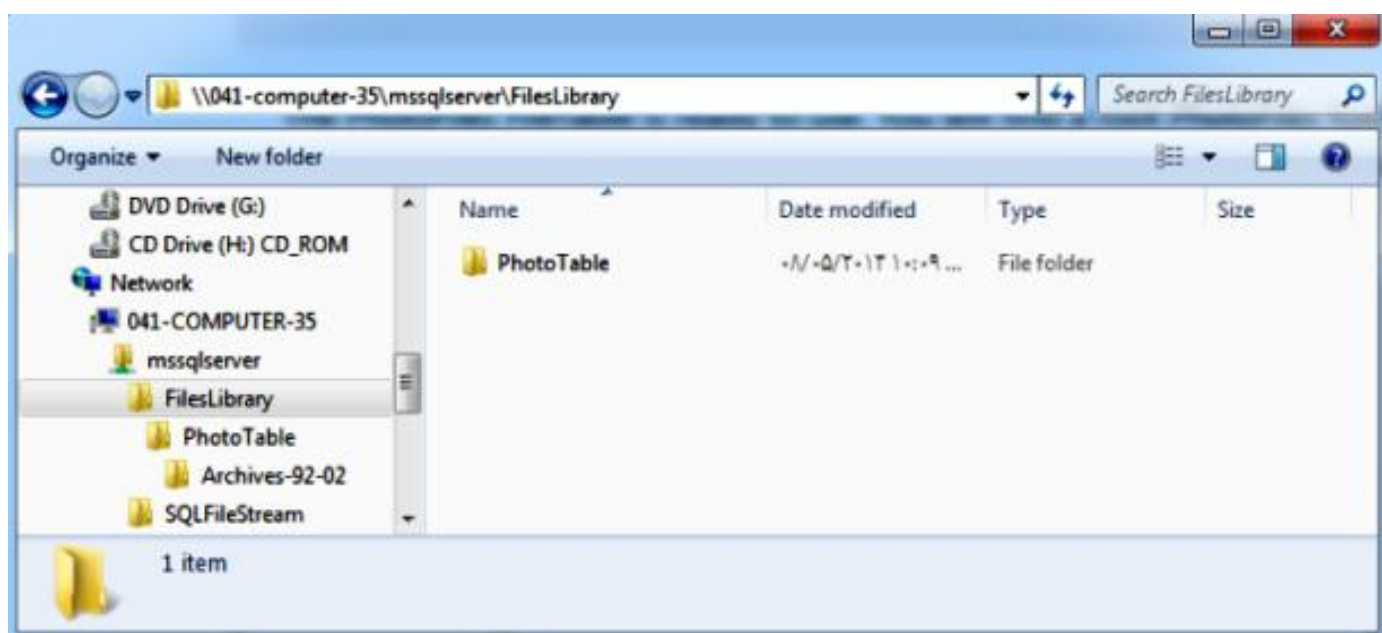
با توجه به آنچه گفته شد برای ایجاد یک پایگاه داده با امکان ساخت FileTable دستورهایی زیر را اجرا کنید:

```
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 پایگاه داده‌ها 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 نوشتن دستور امکان پذیر است و مانند یک Table عادی از محیط کاربری SQL Server نمی‌توان بهره برد. در Object Explorer از گرهی Tables، گرهی FileTables را باز کنید و روی جدولی که هم‌اکنون ساختیم راست کلیک کنید. با انتخاب گزینه‌ی Explore FileTable Directory پنجره‌ی زیر باز می‌شود:



دنباله دارد ...

نظرات خوانندگان

نویسنده: ali.rezayee
تاریخ: ۱۶:۳۶ ۱۳۹۲/۰۳/۰۴

با سلام
موضوع بسیار جالب و مهمی است، لطفاً سریع‌تر ادامه دهید.

نویسنده: علی رضایی
تاریخ: ۱۹:۵۸ ۱۳۹۳/۰۴/۲۷

[بهروش‌های استفاده از FILESTREAM](#)

نویسنده: M.Shakeri
تاریخ: ۱۰:۱۴ ۱۳۹۳/۰۵/۰۸

فکر می‌کنم برای افزودن این قابلیت به دیتابیس دسترسی به دیتابیس master نیاز باشه که هاست‌ها این امکان رو نمیدن.

نویسنده: محسن خان
تاریخ: ۱۲:۸ ۱۳۹۳/۰۵/۰۸

خیلی از قابلیت‌ها مخصوص سازمان‌ها طراحی می‌شن.

نویسنده: Hamidbp
تاریخ: ۱۴:۴۰ ۱۳۹۳/۰۷/۱۴

می‌خواستم بدونم چطور میشه یک فایل رو که در Filetable ذخیره شده، به اصطلاح Fetch کنم و توسط برنامه مربوطه (مثلا adobe acrobat یا word یا ...) نشونش بدم؟