

در انتهای قسمت قبل، نحوه‌ی ایجاد یک جدول جدید با فیلدی از نوع فایل استریم بررسی شد، حال اگر جدولی از پیش وجود داشت، نحوه‌ی 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]

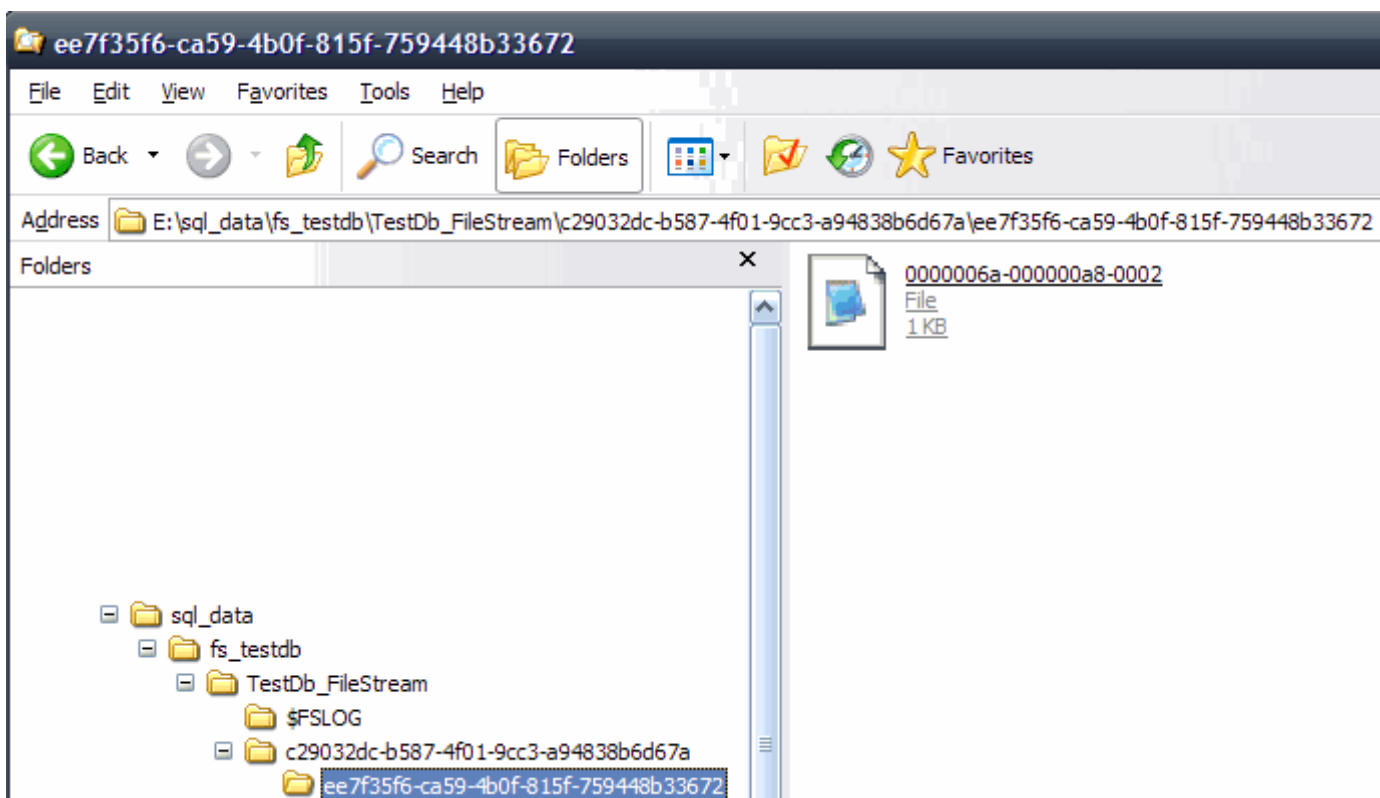
ALTER TABLE [dbo].[tblFiles] ADD DEFAULT (newid()) FOR [FileId]
GO
```

نحوه‌ی 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 نحوه‌ی 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 صرفاً جهت یک مثال ارائه شد. در ادامه، نحوه‌ی ثبت محتویات یک فایل را در فیلدی از نوع فایل استریم و سپس خواندن اطلاعات آن‌را 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 objSqlCommand = new SqlCommand(
                        "INSERT INTO [tblFiles]([Title],[SystemFile]) VALUES(@title , @file)",
```

```

        objSqlCon, objSqlTran))
    {
        objSqlCommand.CommandType = CommandType.Text;

        //تعریف وضعیت پارامترها و مقدار دهی آنها
        objSqlCommand.Parameters.AddWithValue("@title", title);
        objSqlCommand.Parameters.AddWithValue("@file", buffer);

        //اجرای فرامین
        objSqlCommand.ExecuteNonQuery();
    }

    //پایان تراکنش
    objSqlTran.Commit();
}
}

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

    using (SqlConnection objConn = new SqlConnection())
    {
        //کوئری اسی 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 وضعیت و مقدار پارامتر تعریف شده در کوئری
            objCmd.Parameters.AddWithValue("@id", fileId);

            //اجرای فرامین 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 متداول افزودن یک فایل به دیتابیس اس کیوال سرور هیچ تفاوتی ندارد و این جا هم بدون مشکل کار می کند. اطلاعات نهایی به صورت فایل هایی بر روی سیستم که توسط اس کیوال سرور مدیریت خواهند شد و با جدول شما یکپارچه اند، ذخیره می شوند.

در روش دیگری که در اکثر مقالات مرتبط مورد استفاده است، از شیء SqlFileStream کمک گرفته شده و نحوه ی 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 می شود. سپس مسیر سیستمی این فایل را در آرگومان خروجی filepath قرار می دهیم. SystemFile.PathName از اس کیوال سرور 2008 جهت فیلدهای 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. رویه ذخیره شده GetFilePath نیز تنها مسیر سیستمی فایل استریم ذخیره شده را بر می گرداند. به این ترتیب کدهای برنامه به صورت زیر تغییر خواهند 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(
            "لطفا 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())
        {
            //استفاده از رویه ذخیره شده افزودن فایل
            using (SqlCommand objSqlCommand = new SqlCommand(
                "AddFile", objSqlCon, objSqlTran))
            {
                objSqlCommand.CommandType = CommandType.StoredProcedure;

                //مشخص ساختن وضعیت و مقدار پارامتر عنوان
                SqlParameter objSqlParameter1 = new SqlParameter("@Title", SqlDbType.NVarChar, 255);
                objSqlParameter1.Value = title;

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

                //افزودن پارامترها به شیء کامند
                objSqlCommand.Parameters.Add(objSqlParameter1);
                objSqlCommand.Parameters.Add(objSqlParameterOutput);

                //اجرای رویه ذخیره شده
                objSqlCommand.ExecuteNonQuery();

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

                //زمینه تراکنش فایل استریم موجود را دریافت کرده و از آن برای نوشتن محتویات فایل
                //این مورد نیز یکی از تازه‌های اسی 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();
        }
    }
}

/// <summary>
/// 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())
    {
        //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 objSqlParameter1 = new SqlParameter("@ID", SqlDbType.VarChar, 50);
        objSqlParameter1.Value = fileId;
        objSqlCmd.Parameters.Add(objSqlParameter1);

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

        objSqlTran.Commit();
    }
}

return buffer;
}
}
}

```

استفاده خواهیم کرد

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

[FILESTREAM Storage in SQL Server 2008](#)

نظرات خوانندگان

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

تشکر فراوان از مطلب خوب شما

نویسنده: حمید
تاریخ: ۱۳۹۱/۰۵/۰۶ ۲۰:۱۳

سلام

ممنون از مطلب مفیدتون

یک سؤالی که برام پیش اومده اینه که ، این فایل‌ها که در مسیر مشخص شده ذخیره شدن ، موندگار هستن یا موقتی اند ؟

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

ماندگار هستند.

نویسنده: حمید
تاریخ: ۱۳۹۱/۰۵/۰۶ ۲۱:۰۹

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

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

دیتای اصلی همین فایل‌ها هستند. نحوه دسترسی به آن‌ها اینبار از طریق استریم‌ها است که سربرار کمی دارند نسبت به حالت معمولی که کل فایل، داخل بانک اطلاعاتی ذخیره می‌شود و برای دریافت آن buffer pool 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.

نویسنده: غلامرضا ربال
تاریخ: ۱۳۹۴/۰۶/۰۸ ۱۸:۴۵

با تشکر.

برای مواردی که لازم است چندین فیلد از نوع FileStream داشته باشیم به چه شکل باید عمل کنیم؟ آیا باید به ازای هر کدام ID منحصر به فرد داشته باشیم؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۴/۰۶/۰۸ ۱۹:۴۳

بر اساس [مستندات آن](#) ، نیازی نیست:

A table can have multiple FILESTREAM columns, but the data from all FILESTREAM columns in a table must be stored

in the same

FILESTREAM filegroup. If the FILESTREAM_ON clause is not specified, whichever FILESTREAM filegroup is

set to be the default will be used. This may not be the desired configuration and could lead to performance problems.