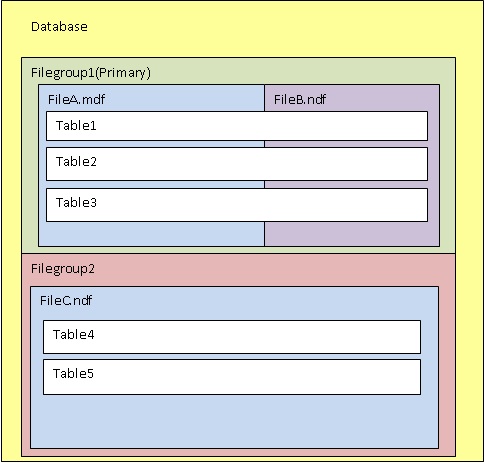
*Shudder.* I like things organized, from the files on my desk to the files in my database. There’s a mechanism in SQL Server to help you separate and organize files: filegroups.

**What is a Filegroup?**

A filegroup is a logical structure to group objects in a database. Don’t confuse filegroups with actual files (.mdf, .ddf, .ndf, .ldf, etc.). You can have multiple filegroups per database. One filegroup will be the primary, and all system tables are stored on it. Then, you add additional filegroups. You can specify one filegroup as the default, and objects not specifically assigned to a filegroup will exist in the default. In a filegroup, you can have multiple files.



Only data files can be assigned to filegroups. Log space is managed separately from data space.

**Why Should I Create Multiple Filegroups?**

There are two primary reasons for creating filegroups: performance and recovery.

Filegroups that contain files created on specific disks can alleviate disk performance issues. For example, you may have one very large table in your database with a lot of read and write activity – an orders table, perhaps. You can create a filegroup, create a file in the filegroup, and then move a table to the filegroup by moving the clustered index. (I’ll cover how to do this later in this post.) If the file is created on a disk separate from other files, you are going to have better performance. This is similar to the logic behind separating data and log files in a database. Performance improves when you spread files across multiple disks because you have multiple heads reading and writing, rather than one doing all the work.

Filegroups can be backed up and restored separately as well. This can enable faster object recovery in the case of a disaster. It can also help the administration of large databases.

**How Do I Create Multiple Filegroups?**

If you are creating a new database, you can specify the filegroups in the CREATE DATABASE statement.

Here, I will create a database named FilegroupTest. It has two filegroups, **PRIMARY**and **FGTestFG2**. There are two data files, **FGTest1\_dat**, assigned to **PRIMARY**; and **FGTest2\_dat**, assigned to **FGTestFG2**.

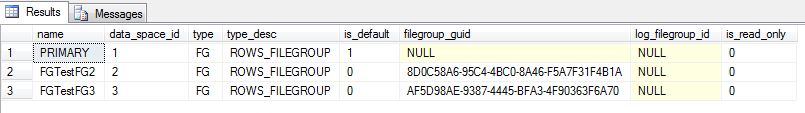
| T-SQL | |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | CREATE DATABASE FilegroupTest  ON PRIMARY  (NAME = FGTest1\_dat,   FILENAME = 'C:Program FilesMicrosoft SQL ServerMSSQL10\_50.MSSQLSERVERMSSQLDATAFGTest1\_dat.mdf'),  FILEGROUP FGTestFG2  (NAME = FGTest2\_dat,   FILENAME = 'C:Program FilesMicrosoft SQL ServerMSSQL10\_50.MSSQLSERVERMSSQLDATAFGTest2\_dat.mdf')  LOG ON  (NAME = FGTest\_log,   FILENAME = 'C:Program FilesMicrosoft SQL ServerMSSQL10\_50.MSSQLSERVERMSSQLDATAFGTest\_log.ldf') |

If you have an existing database, use can use the ALTER DATABASE statement to add a filegroup. I’m going to add**FGTestFG3**to **FilegroupTest**.

| T-SQL | |
| --- | --- |
| 1  2 | ALTER DATABASE FilegroupTest  ADD FILEGROUP FGTestFG3 |

I can view the filegroups in a database using sys.filegroups.

| T-SQL | |
| --- | --- |
| 1  2  3  4 | USE FilegroupTest;  GO  SELECT \*  FROM sys.filegroups |

[](http://blogs.lessthandot.com/wp-content/uploads/users/grrlgeek/sysfilegroups.JPG?mtime=1303779715)

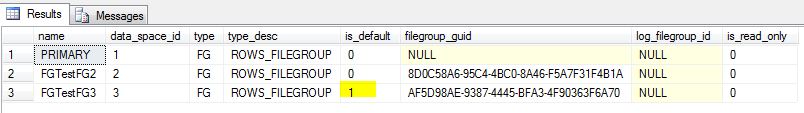
To create a new file, **FGTest3\_dat**, and assign it to **FGTestFG3**, I’ll use ALTER DATABASE again.

| T-SQL | |
| --- | --- |
| 1  2  3  4  5 | ALTER DATABASE FilegroupTest  ADD FILE  (NAME = FGTest3\_dat,   FILENAME = 'C:Program FilesMicrosoft SQL ServerMSSQL10\_50.MSSQLSERVERMSSQLDATAFGTest3\_dat.mdf')  TO FILEGROUP FGTestFG3 |

Right now, my PRIMARY filegroup is the default filegroup. I can change that to **FGTestFG3**using ALTER DATABASE.

| T-SQL | |
| --- | --- |
| 1  2 | ALTER DATABASE FilegroupTest  MODIFY FILEGROUP FGTestFG3 DEFAULT |

When I re-run my sys.filegroups query, is\_default value has changed.

[](http://blogs.lessthandot.com/wp-content/uploads/users/grrlgeek/sysfilegroups2.JPG?mtime=1303779975)

**How Do I Move An Object to a Different Filegroup?**

You can move a table from one filegroup to another, provided the table has a clustered index on it.

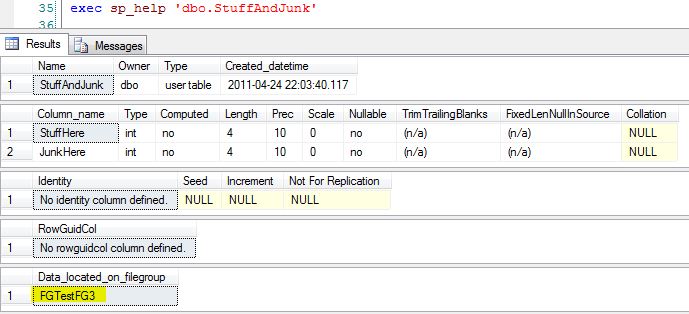
Note: You can move a heap (a table with no clustered index). To do so, you would create an index, move it, and drop the index.

First, I create a table with no indexes.

| T-SQL | |
| --- | --- |
| 1  2  3 | CREATE TABLE StuffAndJunk  (StuffHere INT NOT NULL,   JunkHere INT NOT NULL) |

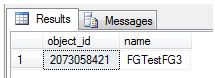
I can use sp\_help to see which filegroup this was created on. It was created on the default, **FGTestFG3**.

| T-SQL | |
| --- | --- |
| 1 | exec sp\_help 'dbo.StuffAndJunk' |

[](http://blogs.lessthandot.com/wp-content/uploads/users/grrlgeek/sphelp.JPG?mtime=1303780300)

I can also return this information using the sys.filegroups, sys.allocation\_units and sys.partitions tables.

| T-SQL | |
| --- | --- |
| 1  2  3  4  5  6 | SELECT PA.object\_id, FG.name  FROM sys.filegroups FG      INNER JOIN sys.allocation\_units AU ON AU.data\_space\_id = FG.data\_space\_id      INNER JOIN sys.partitions PA ON PA.partition\_id = AU.container\_id  WHERE PA.object\_id =      (SELECT object\_id(N'FilegroupTest.dbo.StuffAndJunk')) |

[](http://blogs.lessthandot.com/wp-content/uploads/users/grrlgeek/objectid1.JPG?mtime=1303780416)

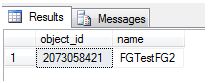
I cannot move the table only. That is simply not part of the ALTER TABLE syntax. According to [BOL](http://msdn.microsoft.com/en-us/library/ms190273.aspx), MOVE TO “Specifies a location to move the data rows currently in the leaf level of the clustered index.”

Note: It is possible to create a table in a secondary filegroup, move the data from the first filegroup to the second, and then drop the table from the primary. Be aware that these types of operations cause a high level of transaction log entries. Ensure that the transaction log is properly sized to prevent a large amount of growth in the logs or inadvertently affect things that rely on the transaction logs such as log shipping or mirroring.

I’m going to add a clustered index to the table. When I do this, I specify which filegroup I want it created on. I create**StuffJunk**on **FGTestFG2**.

| T-SQL | |
| --- | --- |
| 1  2  3 | CREATE CLUSTERED INDEX StuffJunk      ON StuffAndJunk (StuffHere, JunkHere)      ON FGTestFG2 |

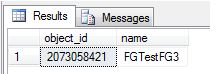
If I run my sys.filegroups query again, I can see I have the same object\_id, but it has moved to a different filegroup.

[](http://blogs.lessthandot.com/wp-content/uploads/users/grrlgeek/objectid2.JPG?mtime=1303780550)

How would I move a table with an existing clustered index? Let’s move **StuffAndJunk** back to **FGTestFG3**. I would issue a create clustered index command with the option to drop existing, like this.

| T-SQL | |
| --- | --- |
| 1  2  3  4 | CREATE CLUSTERED INDEX StuffJunk      ON StuffAndJunk (StuffHere, JunkHere)      WITH (DROP\_EXISTING = ON)      ON FGTestFG3 |

Re-running my sys.filegroups query shows that the index, and thus the data and table, are on **FGTestFG3**.

[](http://blogs.lessthandot.com/wp-content/uploads/users/grrlgeek/objectid3.JPG?mtime=1303780648)

**Organize, Organize, Organize!**

Help your databases look like this:

[](http://blogs.lessthandot.com/wp-content/uploads/users/grrlgeek/organized-filing-cabinet.jpg?mtime=1303781231)

Filegroups are a great way to organize your data, increasing performance and providing additional disaster recovery. If you can do this in the planning stages, it’s great, but be aware that you can add filegroups in later, too.