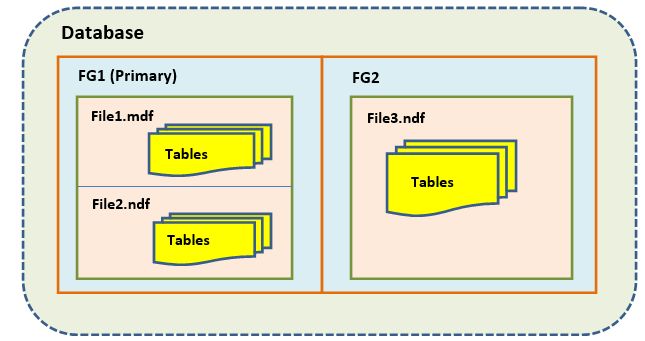
**1. What is Filegroup**  
In SQL Server filegroup is a logical structure which contains objects like data file, tables and indexes. In other words we can say that a filegroup is a logical unit in which all database files are grouped together and simplifies database administration resulting into improved performance by controlling the placement of objects into specific filegroups on specific drive.   
  
  
**Figure 1:** Pictorial representation of filegroups  
  
**2. Why Filegroups**  
Filegroups make administration easier for a DBA. Using multiple filegroups we can gain following benefits;

1. We can separate user data with internal system data using multiple filegroups.
2. We can overcome with maintenance overhead by putting archive (or even read-only) data onto their own filegroups and dedicated set of disks.
3. We can gain performance improvement by putting larger tables/indexes on their own filegroup and/or dedicated set of disks.
4. We can bring some parts of the database online quickly ( piecemeal restore ).

**3. Types of Filegroups**  
There are two types of filegroups:

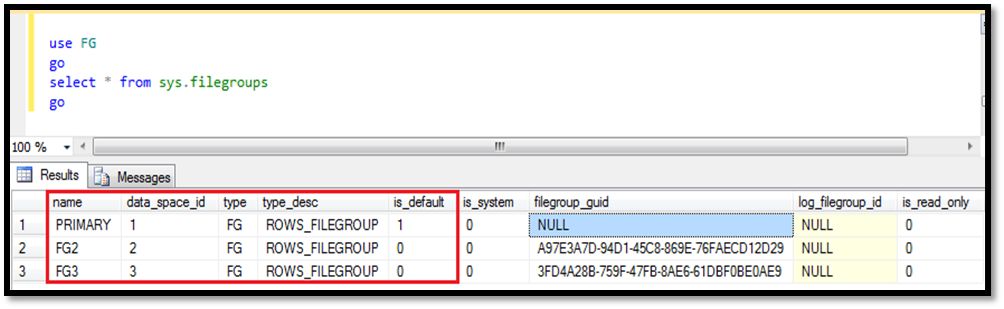
3.1. Primary Filegroup  
3.2. User defined/Secondary Filegroup

**3.1. Primary Filegroup:**  
The filegroup which contains the primary data file and any other files that are not associated to another filegroup is termed as Primary filegroup.   
  
**3.2. User-defined Filegroup:**  
The Filegroups which we create from FILEGROUP keyword using CREATE DATABASE or ALTER DATABASE is termed as user-defined filegroups. This file is created by user or later modifies database by a user.  
  
**4. Filegroup Examples**  
I am using below query to create filegroup with new database:

1. **CREATE** **DATABASE**[FG]
2. **ON** **PRIMARY**
3. (**NAME** = N 'FG\_data',
4. FILENAME = N 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\FG\_data.mdf',
5. **SIZE** = 4096 KB, FILEGROWTH = 1024 KB)
6. LOG **ON**
7. (**NAME** = N 'FG\_log',
8. FILENAME = N 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\FG\_log.ldf',
9. **SIZE** = 1024 KB, FILEGROWTH = 10 % )
10. GO
11. **ALTER** **DATABASE**[FG] **ADD** FILEGROUP[FG2]
12. GO
14. **ALTER** **DATABASE**[FG] **ADD** FILEGROUP[FG3]
15. GO

Using above query we created database name ‘FG’ and primary filegroup which is the default filegroup and FG2 and FG3 two user-defined filegroups.  
  
**5. How to view filegroups**  
By executing below query we can view filegroups in a database:

1. use FG
2. go
3. **select** \* **from** sys.filegroups
4. go

  
**Figure 2:**Viewing filegroups  
  
**6. Creating a file and assigning it to filegroup**  
To add file in a filegroup we execute ALTER DATABASE query.

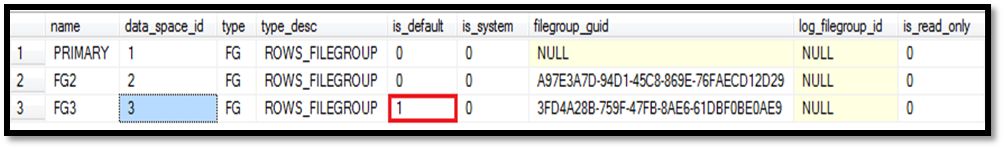
1. **ALTER** **DATABASE** FG
2. **ADD** FILE
3. (**NAME** = FG3\_data,
4. FILENAME = N 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\FG3\_data.ndf')
5. **TO** FILEGROUP FG3

**7. Changing default filegroup**  
  
Again ALTER DATABASE query will execute to change the default filegroup.

1. **ALTER** **DATABASE** FG
2. **MODIFY** FILEGROUP FG3 **DEFAULT**

Now check the default value for filegroup FG3 using sys.filegroups.

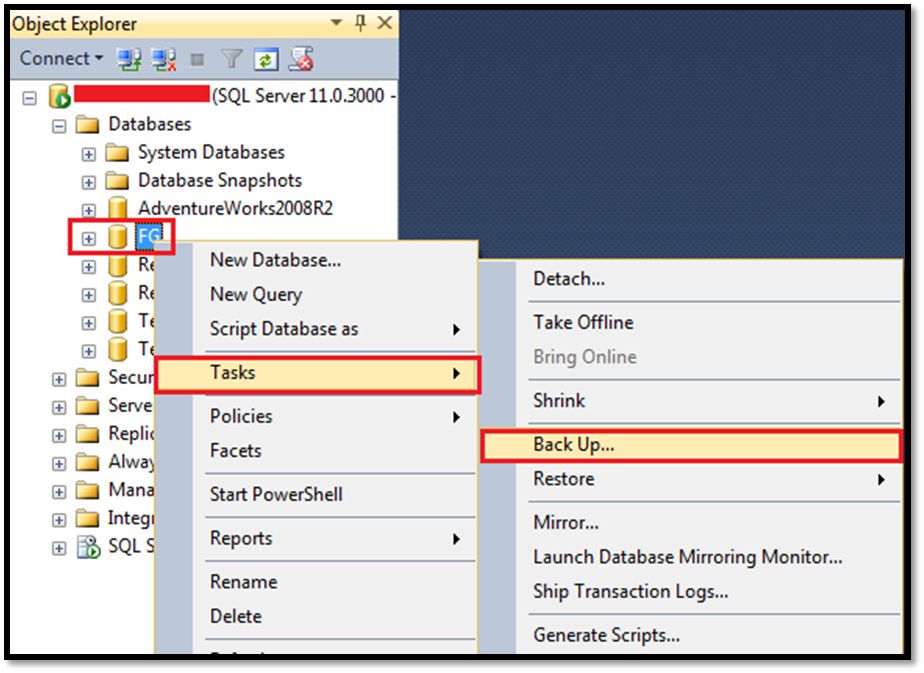
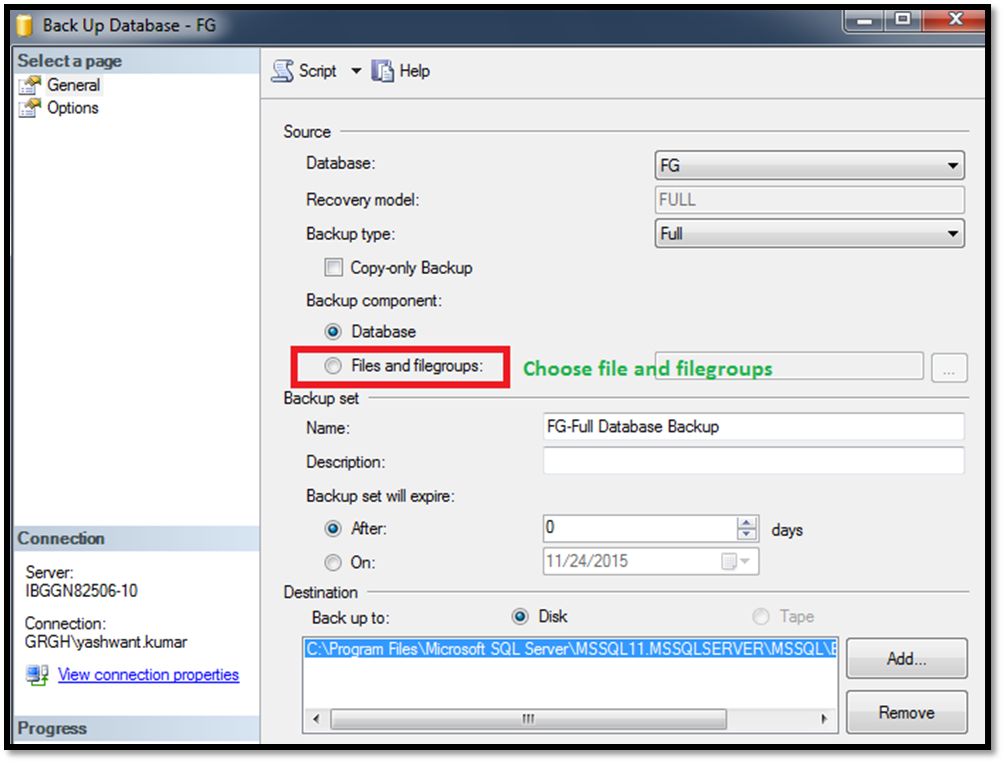
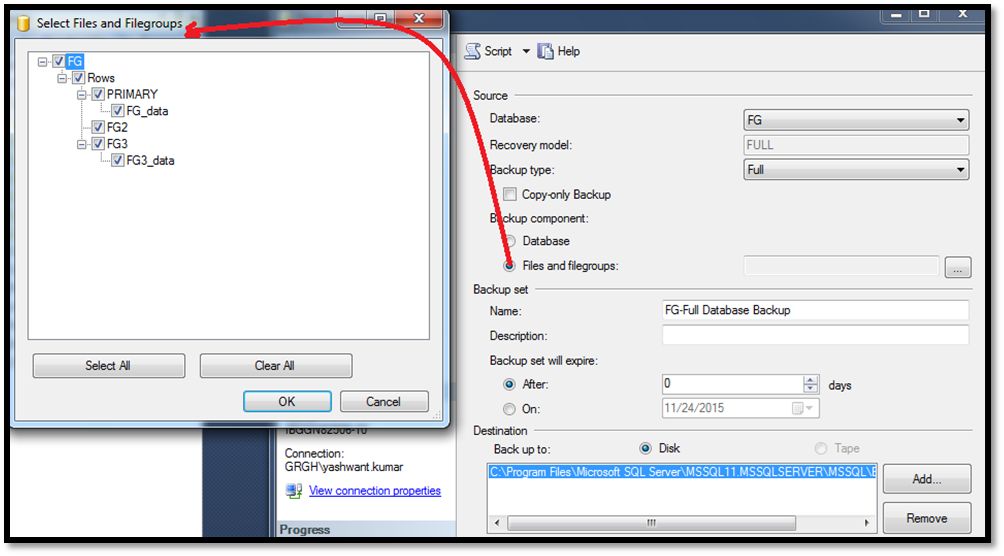
1. use FG
2. go
3. **select** \* **from** sys.filegroups
4. go

  
**Figure 3:** Changing default filegroups  
  
Now you can clearly see that the is\_default value is 1 for FG3 filegroup. It means the default filegroup is FG3.  
  
**8. Filegroups Backup**  
We can take filegroups backup with two ways:

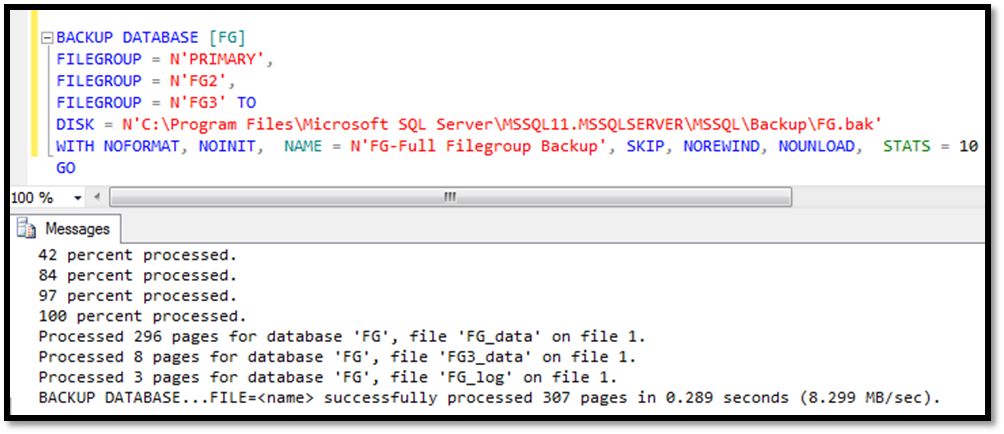
8.1. With SQL Server Management Studio (SSMS)  
8.2. With T-SQL

**8.1. With SQL Server Management Studio (SSMS):**  
To take backup of filegroups with ssms follow below steps:

1. Select database>>do right click.
2. Go to task >> click on backup,
3. Backup database window will appear, here choose option file and filegroups under backup component.
4. When you select file and filegroups another window will open which will show you all the filegroups for that database.
5. Click on check boxes to take backup of one or more filegroups according to your requirement.

**Step 1 and step 2:**  
  
  
**Figure 4:** Showing how to reach backup option  
  
**Step3:**  
  
  
**Figure 5:** Choosing files and filegroups backup  
  
**Step 4 and step 5:**  
**Figure 6:** Selecting filegroups for backup   
  
**8.2. Filegroups backup with T-SQL:**  
You can execute below query to take backup of filegroups. In below example my database name is ‘**FG**’ and I am taking backup of all 3 filegroups i.e.

1. **PRIMARY**, FG2, FG3
3. BACKUP **DATABASE**[FG]
4. FILEGROUP = N 'PRIMARY',
5. FILEGROUP = N 'FG2',
6. FILEGROUP = N 'FG3'
7. **TO**
8. DISK = N 'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\Backup\FG.bak'
9. **WITH** NOFORMAT, NOINIT, **NAME** = N 'FG-Full Filegroup Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10
10. GO

  
**Figure 7:** Taking filegroups backup with t-sql  
  
**Points to remember:**

* All pages for the system tables are allocated in the primary filegroup.
* Log files ( .ldf) does not use filegroups.
* No file can be a member of more than one filegroup.