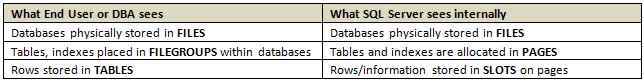
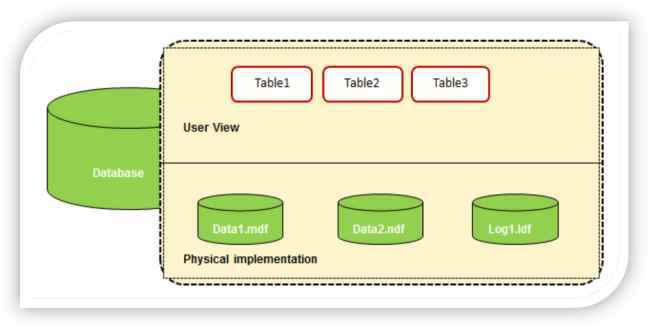
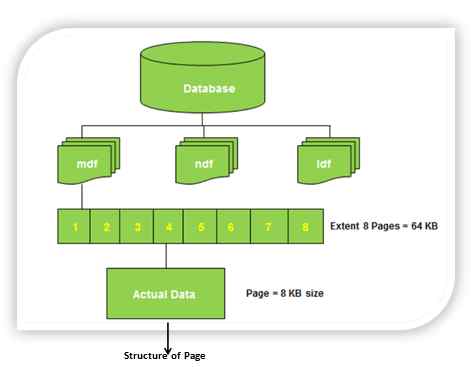
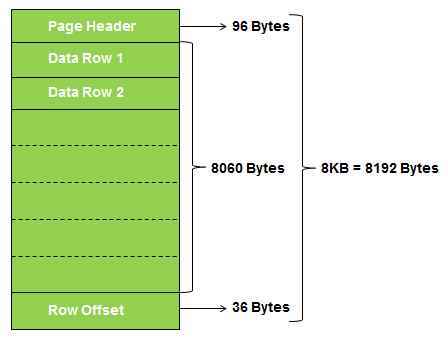
**1. Introduction**  
Like every other DBMS software, SQL Server has also its storage engine which we need to know for better understanding of basics. SQL DBA should know well about these storage basics. In this article we will discuss about SQL Server storage engine. SQL Server storage engine has itself a vast topic, one article will not be sufficient for this because it has many things in it that we should know. After thinking a lot how to start this topic, I decided to discuss these basics in bunch of short articles because if I will try to cover all the things in one article it will get lengthy and we will not easily grasp. So let’s move ahead into SQL Server Storage Basics.  
  
**2. Physical Database Architecture**  
Every SQL DBA knows that the data in SQL Server is stored and organized into the logical components such as tables, views, procedures in databases, which a user can view easily but from the point of database administration we will talk here about physical implementation of data in databases. A database is physically implemented as two or more files on disk, which we called primary data files (.mdf), secondary data files (.ndf) and log files (.ldf). For better understanding we can put the following points:  
  
  
  
  
**Figure 1:**User view & physical implementation of database   
  
**3. Storage Basics Explained**  
**3.1 Primary Data Files:**

**3.1.1 What is Primary Data File?**  
  
Primary data file is a SQL Server database file which contains the startup information and act as a starting point for any database. It holds user data and all objects such as tables, indexes and stored procedures.  
  
  
  
  
**Figure 2:** Illustrating physical files, extents, pages and page structure in SQL Server   
  
**3.1.2 Extension of Primary Data File:**  
It is recommended that primary data file should have .mdf extension but really you are free to give any extension to it like .abc, .ian, .jim, .cathy, .xyz

The following is the example:

* 1. USE [master]
  2. GO
  3. **ALTER** **DATABASE** [tempdb]
  4. **ADD** FILE
  5. (**NAME** = N'tempdev\_Data06',
  6. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data06.ian',
  7. **SIZE** = 100MB, FILEGROWTH = 1024KB)
  8. GO
  10. **ALTER** **DATABASE** [tempdb]
  11. **ADD** FILE
  12. (**NAME** = N'tempdev\_Data07',
  13. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data07.jim',
  14. **SIZE** = 100MB, FILEGROWTH = 1024KB)
  15. GO
  17. **ALTER** **DATABASE** [tempdb]
  18. **ADD** FILE
  19. (**NAME** = N'tempdev\_Data08',
  20. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data08.cathy',
  21. **SIZE** = 100MB, FILEGROWTH = 1024KB)
  22. GO

  
**Figure 3:** Showing that the data files have different extensions i.e .ian, .jim and .cathy   
  
**3.1.3 How many Primary Data Files, a database can have:**  
Every database can only have ONE primary data file (mdf), but do you really agree with this point. Let’s dig it more with some experiments:  
  
**Step 1:** I changed the name of primary data file and transaction log file with the following query to make it easy to understand:

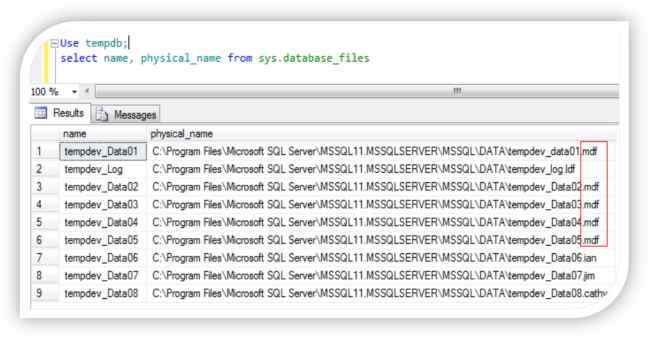
* 1. USE [master]
  2. GO
  3. **ALTER** **DATABASE** [tempdb] **MODIFY** FILE (**NAME**=N'templog', NEWNAME= N'tempdev\_Log')
  4. GO
  5. **ALTER** **DATABASE** [tempdb] **MODIFY** FILE (**NAME**=N'tempdev', NEWNAME=N'tempdev\_Data01')
  6. GO

**Step 2:**Now I added four more data files: '*tempdev\_Data02', 'tempdev\_Data03', 'tempdev\_Data04'and 'tempdev\_Data05'*with the following script:

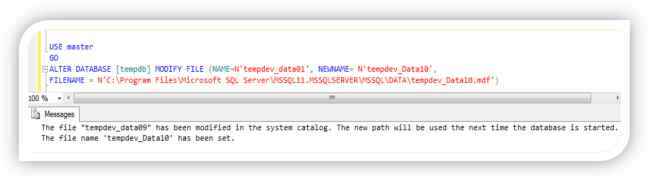
* 1. USE [master]
  2. GO
  3. **ALTER** **DATABASE** [tempdb]
  4. **ADD** FILE
  5. ( **NAME** = N'tempdev\_Data02',
  6. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data02.mdf' ,
  7. **SIZE** = 100MB , FILEGROWTH = 1024KB )
  8. GO
  10. **ALTER** **DATABASE** [tempdb]
  11. **ADD** FILE
  12. ( **NAME** = N'tempdev\_Data03',
  13. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data03.mdf' ,
  14. **SIZE** = 100MB , FILEGROWTH = 1024KB )
  15. GO
  17. **ALTER** **DATABASE** [tempdb]
  18. **ADD** FILE
  19. ( **NAME** = N'tempdev\_Data04',
  20. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data04.mdf' ,
  21. **SIZE** = 100MB , FILEGROWTH = 1024KB )
  22. GO
  24. **ALTER** **DATABASE** [tempdb]
  25. **ADD** FILE
  26. ( **NAME** = N'tempdev\_Data05',
  27. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data05.mdf' ,
  28. **SIZE** = 100MB , FILEGROWTH = 1024KB )
  29. GO

**Step 3:**Let’s see all the files with the following query:

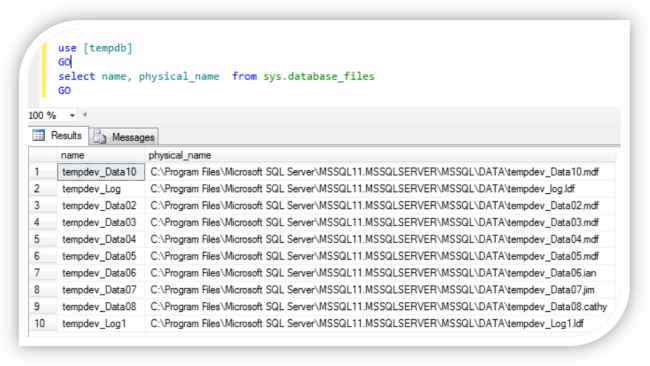
* 1. Use tempdb;
  2. **select** **name**, physical\_name **from** sys.database\_files

  
**Figure 4:** Showing that the data files tempdev\_data01, tempdev\_data02, tempdev\_data03, tempdev\_data04, tempdev\_data05 having the same extension .mdf  
  
So still the statement “*Every database can only have ONE primary data file ( mdf )*” is true or something wrong with this statement or the statement is not correct.  
  
The answer lies in the statement which we made in **3.1.2 section:** “It is recommended that primary data file should have .mdf extension but really you are free to give any extension to it such as .abc, .ian, .jim, .cathy, .xyz “ what we discussed earlier and the statement “Every database can only have ONE primary data file ( mdf )” is also true because a primary data file is the data file associated with any database when database is created for the very first time, and YES this can only be ONE. Same is true for transaction log file ( .ldf ) and secondary data file ( .ndf )  
  
Here, I want to specify one more thing, suppose we have several mdf files for a particular database & we want to know that which is the main; to uncover it let’s take the above example which is in figure 4, here we have FIVE data files tempdev\_data01, tempdev\_data02, tempdev\_data03, tempdev\_data04, tempdev\_data05. Now suppose someone changed the name then how we will identify the only ONE primary data file. Let’s take an example:  
  
**Step 1:**In figure 4 I showed you FIVE mdf files for tempdb database and three data files which have .ian, .jim and .cathy extension. Now I am going to change the name of tempdev\_data01.mdf to tempdev\_data10.mdf with the following query:

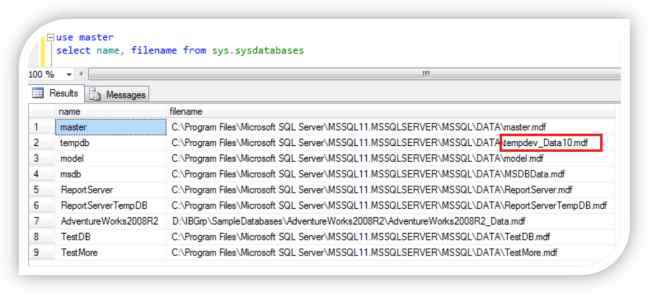
* 1. USE master
  2. GO
  3. **ALTER** **DATABASE** [tempdb] **MODIFY** FILE (**NAME**=N'tempdev\_data01', NEWNAME= N'tempdev\_Data10',
  4. FILENAME = N'C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\tempdev\_Data10.mdf')

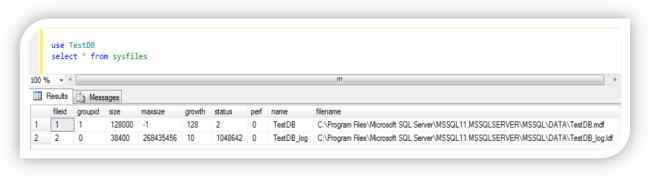
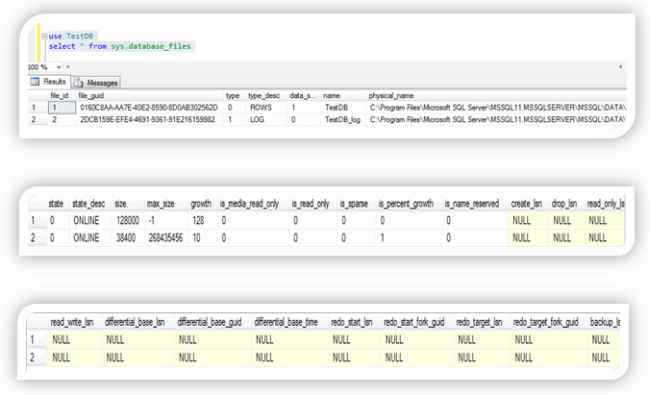
  
**Figure 5:** Modifying data file  
  
**Step 2:**Check the name and physical\_name for files in tempdb database with the following query:

* 1. use [tempdb]
  2. GO
  3. **select** **name**, physical\_name **from** sys.database\_files
  4. GO

  
**Figure 6:** Showing multiple data files with .df extension and some different extension  
  
Now there are again FIVE data files with .mdf extension and THREE data files with other extensions: .jim, .ian and .cathy. So many data files how we find that the only ONE primary data file which we are discussing, because till this discussion we know that the primary data file should be one and only ONE. So let’s find out the only ONE primary data file.  
  
**Step 3:**To find out one and only ONE primary data file for a database execute the following query:

* 1. Use tempdb;
  2. **select** **name**, filename **from** sys.sysdatabases;

  
**Figure 7:** Fetching the name and location of only ONE primary data file  
  
Now it is clearly visible that the only ONE primary data file is **tempdev\_Data10.mdf.**Someone here can ask one question why we use **sysdatabases** only for query not other query, the answer is simple because the information about the only ONE primary data file is stored in the filename column of the sysdatabases in the master database.

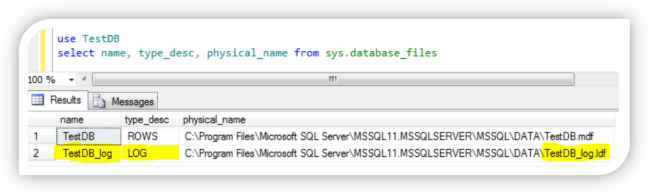
**3.1.4 Location of Primary Data Files:**  
The location of the primary data file is stored in the **filename** column of the **sysdatabases** in the master database.  
  
When SQL Server starts up database it looks for this file because in this file the information for all databases exists.  
  
Primary data file contains **sys.database\_files** system table which stores information about all other files in a specific database. Here someone can ask you why sys.database\_files not sysfiles [sysfiles also give you the information about all files in the database]. So let’s move ahead to know why I said sys.database\_files.  
 **sysfiles vs sys.database\_files**  
The compatibility view sysfiles returns one row for every file that is associated with a database.   
  
  
**Figure 8:** Showing output from sysfiles  
  
The catalog view sys.database\_files is the replacement for sysfiles. Like sysfiles, sys.database\_files also returns one row for every file that is associated to a database. sys.database\_files gives more information than the sysfiles and have more readable output. It gives information about state i.e. ONLINE/OFFLINE, lsn, growth, media etc. Below is the output of sys.database\_files split in 3 parts because of space.  
  
  
**Figure 9:** Showing output from sys.database\_files  
  
So from the above discussion we can conclude also that the primary data file have pointer to all other files in the specific database.   
  
**3.2 Secondary Data Files**

* A database can have any number of secondary data files, these files are optional and user-defined. We can use secondary files to spread data across multiple disks.
* Data files other than the primary data file make secondary data files.
* It is not necessary to have secondary data files as these are optional. We can create it depending on the requirement & database environment.
* We can put secondary data files in default filegroup or any other filegroup defined by user.
* .ndf is the recommended extension for secondary data files.

**3.3. Transaction Log Files**

**3.3.1 What is Transaction Log File?**  
Transaction log file is a sequential record of transactions which holds the log information that is used to help in disaster recovery scenarios to recover the database.   
  
**3.3.2 Extension of Transaction Log File:**  
Recommended extension for log file is .ldf, again it is not mandatory you are free to give any extension to it.  
  
**3.3.3 How many Transaction Log Files, a database can have:**  
We can create multiple log files for a database, but there can be only one log file will be active at a time because transactions are written to first file until it will get full. Once it will get full transactions will get written in second file and so on.  
  
**3.3.4 Location of Transaction Log Files:**  
Location of transaction log file is stored in sys.database\_files, we can use the following query to get the location of transaction log file for a specific database:

* 1. use TestDB
  2. **select** **name**, type\_desc, physical\_name **from** sys.database\_files

  
**Figure 10:** Getting name and location of transaction log files

**4. Storage Limitation/Capacity of SQL Server**  
SQL Server database engine has the = following storage capacity for different SQL Server objects:  
  
