# SQL Server Disk Configuration SAN SSD And Partitioning Best Practices

<https://www.gigxp.com/sql-server-disk-configuration-san-ssd-partitioning-best-practices/>

* **SQL Server Disk configuration** settings are one of the most important aspects of [**SQL Server performance tuning**](https://www.gigxp.com/sql-server-performance-tuning/).
* Typically, disks are one of the slowest parts of the entire SQL subsystem. Without proper disk configuration, SQL Server can slow down, increase locks and waits.
* It can also significantly increase the overall memory and CPU usage.
* In short, disk management and optimization are crucial for [**SQL Server performance**](https://www.gigxp.com/sql-server-performance-tuning-with-dynamic-management-views/).
* In this post, we are going to focus on just that. Read along to know more.

**Standard SQL Server Disk Configuration Settings**

Before going into the specifics, some general requirements are recommended for SQL Server Performance. These are as follows:

* Use separate disk drives for different purposes. (Log File, Data Files, Backup & TempDB)
* Maintain strict Disk latencies. For Database Files (MDF) 15-25 ms. For Log files, between 2-5 ms. It applies to both standard & tempDB data & log files.
* Format all the disk drivers (that will contain Data & Log files) with 64 KB cluster allocation.
* If you are running any corporate anti-virus programs, put \*.mdf, \*.ndf & \*LDF under exemption list.
* TempDB should be planned accordingly based on the optimizations defined by Microsoft. See below:

### **SQL Server Disk Configuration Settings for TempDB**

#### Storage Planning for TempDB

* Set the recovery model of TempDB to SIMPLE. This model automatically reclaims log space to keep space requirements small.
* Disable to default allow settings for TempDB files to grow automatically. It reduces the CPU overhead of managing a dynamic file-growth.
* Create multiple data files for TempDB. How Many?**Total Number of TempDB Primary Data File** = **SQL Processor Count not exceeding eight files**. Which means, if you have 16 cores for SQL Server, create **only eight files**.
* Each data file should be of equal size.
* Try to keep each data files on separate disk drives for achieving high **IO Parallelism**.
* TempDB Data and Log files should be maintained on faster disk drives (Preferably RAID 1 if possible)
* Use RAID-10 or SSD Disks.
* Pre-size TempDB files
* 25% of largest DB size.
* Set Auto Growth to fixed size < 200 MB
* You should have the same number of data files as the number of CPUs up to a maximum of 8.

***Note: The new***[***SQL Server 2016 Installation program***](https://www.gigxp.com/new-sql-server-2016-installation-features-walk-through/)***allows you to create seperate TempDB files during the setup process.***

### **SQL Server Disk Configuration Settings for RAID, Storage Pools, SSD’s and Storage Spaces**

In the last couple of years, storage pools gained a lot of popularity. Instead of complicated and expensive RAID implementations, storage pools are the way to go in the future. Using storage pools, you can add disks & spindles as you need more performance over time. However, even though there is a shared storage pool across workloads, it is important to isolate them at the logical level. It will, in turn, help you to isolate disk bottlenecks of a slow performing query. If you put everything in a single pool, there is no way to tell what is causing the disk performance to degrade. On a high level, isolate the **System DB’s**, **User DB’s**, **TempDB’s** and **Log files** at the logical level.

Modern storage system controllers already do an excellent job of distributing workloads across different disks internally and implementing RAID on top of it does not make a lot of sense. However, it is important to contact your storage vendor for the best practices.

### **Common FAQ’s on SQL Server Disk Configuration Settings**

1. <**Question**>If you add more RAM than the actual dB Size, it stores the dB in memory (RAM) instead of going to disk to retrieve data? <**Answer**>No, typically the SQL Server Buffer Manager decides what to keep in memory and what not to. There is no administrative control over it. However, more RAM is good for SQL Server performance. Especially for repetitive queries.
2. <**Question**>How to measure the “**Disk response**time” in milliseconds? <**Answer**> You can use [**free tools for SQL Server**](https://www.gigxp.com/free-tools-to-monitor-sql-server-performance/) available on the Internet to measure this. For example, In a spindle disk configuration, if data is fragmented and random read/writes patterns are observed, the Disk seek time/response time would be very high.
3. How many TempDB files needs to be created? Should we create eight tempdB files (or 16 to match the core count?).  What size would you suggest for each? <**Answer**> The number of Tempdb files should not exceed a maximum of 8, even if core count is 16.
4. With NVMe based storage pools, Do we still need to look at separate disks for data, log & tempdB devices these days?  Can they not all be stored in a single 20 spindle RAID10 array and separated logically? <**Answer**> We still recommend them to be physically isolated. All of these workloads have very different access patterns. It is, therefore, beneficial to keep these workloads in isolation.
5. <**Question**> Do you still need a Physical Server to get maximum SQL Performance? <**Answer**> If you follow all the best practices while building your SQL Server, you can get similar performance from a SQL Server running in a VM.

### Conclusion:

This information should help you get started with SQL Server Disk Configuration best practices. Hopefully, this was helpful. If you have any comments or suggestions, please feel free to drop us a comment below in the comment section.