DXC.technology

SQL Server Standards

(V1.0)

17 September, 2020

Important Notice

This document has been prepared by MSSQL DBA team which will act as standard for all the account having Microsoft SQL Server instances.

The information contained in all contents of this document and all schedules, annexures and attachments to it (collectively "**Document**") is confidential information of DXC Technology Company or its affiliates (collectively "**DXC**") and is provided for evaluation purposes only. In consideration of receipt of this Document, Customer agrees to maintain the content of this Document in confidence and not to reproduce or otherwise disclose this information to any person or entity outside the Customer group directly responsible for evaluation of its contents, unless otherwise authorized by DXC in writing.

DXC has prepared this Document in good faith based on information made available to it by Customer and DXC reserves the right to make amendments and correct any errors that are identified after submission of it. Neither DXC nor its representatives make any representations or warranties as to the accuracy or completeness of the information provided in here and neither DXC nor its representatives shall have any liability towards Customer or any of its representatives resulting from Customer's use of the information provided.

Contents

1

High level overview

The SQL Server standards are designed to provide a consolidated view of the standards and best practices which should propel your team performance to the top tier. This document is for quick reference to key process and technical setups.

1 Installation Standards

The SQL server installation standard is very important for smooth functioning of SQL Server. Only the required components should be installed while installing SQL Server.

- Always fully document installs so that your SQL Server instances can easily be reproduced in an emergency.
- If possible, install and configure all of your SQL Server instances consistently, following an agreed-upon organization standard.
- Don't install SQL Server services on instances that don’t use them, such as Microsoft Full-Text Indexing, Reporting Services, or Analysis Services.
- For best performance of SQL Server running under Windows, turn off any operating system services that aren't needed.
- For optimum SQL Server performance, you want to dedicate your physical servers to only running a single instance of SQL Server, along with no other applications.
- For best I/O performance, locate the database files (.mdf) and log files (.ldf) on separate spindles to isolate disk access patterns.
- If tempdb will be used heavily, put it on its own separate spindles.
- Do not install SQL Server on a domain controller.
- Be sure that SQL Server is installed on an NTFS partition.
- Don’t use NTFS data file encryption (EFS) and compression on SQL Server database and log files.
- backing up .ini file for future reference
- Disks meant for mdf and ldf files should be formated with 64K cluster size

1.1 Upgrade SQL Server

Whenever we plan to upgrade the SQL Server, there are many things to be considered. Below are some of the pointers that can be used to make sure the upgrade is successful.

- Run the SQL Server Upgrade Advisor before upgrading. Make any necessary changes before performing the upgrade.
- Perform a test upgrade of your test SQL Servers before you upgrade your production servers. And don't forget to test your applications with the new version also.
- Before you upgrade, be sure you have a plan in place to fall back to in case the upgrade is problematic.
- Don't upgrade SQL Server clusters in place. Instead, rebuild them on new hardware.
- If you upgrade from a previous version of SQL Server, you should update all of the statistics in all your databases. This is because statistics are not automatically updated during the upgrade process
- Get VM snapshot taken wherever possible for quick rollback

Policy of customer should supersede these standards

2 Standard maintenance setup

Standard maintenance job should be setup on the system immediately after the installation and make sure this is working as per expectation before releasing the system for smoke testing.

Please use the Ola Hallengren's script for all your DB maintenance.

- Full Backup (retention period based on agreed RPO and RTO)
- Log backup (retention period based on agreed RPO and RTO)
- Differential Backup (optional)
- Integrity Check
- Rebuild Index
- Reorganize Index (can be skipped if rebuild is in place)
- Update Statistics (for Column stats)
- Delete Backup (older than 6 months) and Job History (older than 1 or 2 months)

Some of the basic guidelines for your maintenance jobs are:

- Naming standard for DBA specific jobs/scripts
- Avoid overlapping jobs on the same SQL Server instance. Ideally, each job should run separately at different times.
- When creating jobs, be sure to include error trapping, log job activity, and set up alerts so you know instantly when a job fails.
- Create a special SQL Server login account whose sole purpose is to run jobs, and assign it to all jobs.
- If your jobs include Transact-SQL code, ensure that it is optimized to run efficiently.
- Periodically (daily, weekly, or monthly), rebuild or reorganize the indexes of your databases to remove logical fragmentation and wasted space.
- Periodically, as part of your scheduled database maintenance tasks, run DBCC CHECKDB on all your databases to verify database integrity.
- Avoid running most DBCC commands during busy times of the day. These commands are often I/O intensive and can reduce performance of the SQL Server, negatively affecting users.
- If you rarely restart the SQL Server service, you may find that the current SQL Server log gets very large and takes a long time to load and view. You can close the current error log and create a new one by running sp_cycle_errorlog or DBCC ERRORLOG, so that the log doesn't get overly large. Set this up as a weekly job.
- Script all jobs and store these scripts in a secure area so they can be used if you need to rebuild the servers.

Policy of customer should supersede these standards

3 RTPA

Always take the SQL Server under the support using RTPS automation process.

RTPA and adopt RTPA Automation process.

4 Configuration backup

Configuration is very import for your SQL server and we should always have backup of configuration of your SQL Server. There are some information about the configuration as below.

- SQL Server configuration settings should remain at their default settings. Any changes to these settings should only be made by an experienced DBA who understands the pros and cons of making changes.
- In most cases, Memory settings are important
  - “maximum server memory”, should be set optimally. If the machine has only SQL Server, then 80-20 is best suited with 80% memory should be b allocated to SQL server and 20% should be left for OS
  - “minimum server memory” should be left to their default values. This is because the default values allow SQL Server to dynamically allocate memory in the server for the best overall optimum performance. where you may have to change the “maximum server memory” to an amount that is less than the physical amount of RAM in your server.

Once the system setup, please make a backup copy of the configuration parameters. This can be done by using simple T-SQL command like

```
EXEC sp_configure 'show advanced options',1;
RECONFIGURE WITH OVERRIDE;
EXEC sp_configure;
EXEC sp_configure 'show advanced options',0;
RECONFIGURE WITH OVERRIDE;
```

Policy of customer should supersede these standards

5 Recommended performance setting

Please find attached recommended MSSQL settings for better performance

- Partition the Disk volumes hosting SQL databases (Data and log) with 64 KB allocation unit size
- Instant file initialization
- OS Power Saving setting to High Performance
- Antivirus exclusion on SQL files
- Lock Pages In memory
- Setting appropriate Page file size
- Windows security policy and permissions
- SQL Server Maximum Memory Setting
- Database Default Locations
- Maximum Degree of Parallelism
- Cost Threshold for Parallelism
- Optimize for Ad hoc Workloads
- Auto Update Statistics Asynchronously
- Autoshrink
- TempDB Configuration and Sizing
- Backup Compression
- Remote Dedicated Administrator Connection (DAC)
- Database Autogrowth and files location
- Page Verify Option : PAGE_VERIFY set to CHECKSUM
- Maintenance
- Service Accounts
- SQL Server Port
- Disable SA

For de[illegible] related to these parameters are at attached excel file

Recommended_MSSQL_Settings.xlsx

6 Database backup and restore

As you are aware that data is very important to any business. Data is the Heart of the enterprise; it is crucial for us to protect it. To protect your organization's data, you need to implement a well refined database backup and restore plan. Backing up the database and its other related files can protect against accidental loss of user data, database corruption, hardware failures, and even natural disasters. It's our job as Database admins. Teams to make sure that backups are configured properly and are taken successfully to ensure our data integrity.

Backups have two distinct purposes. The primary purpose is to recover data after a database crash. The secondary purpose of backups is to restore point-in-time in past due to different scenarios.

Backup schedules:

Full Backup: Every weekend

Differential backup: every day at 10 PM

Transaction log backup: Every two hour (if heavily used database change this to one hour or as less time as possible)

Please use the Ola Hallengren’s script for all your DB backup. This is more widely covering almost al the events.

Important: Periodic restore test for Production DBs

***Policy of customer should supersede these standards**

7 Monitoring

This section covers capability & technology specific best practices regarding Monitoring & Reporting Tools installation and configuration. Monitoring if configured badly or not customized has the ability to generate excessive workload that we consider wasted effort. Mastery of the configuration of Monitoring is a core and essential part of achieving high performance.

You need to thoroughly evaluate why you are monitoring certain events and generating incidents from them. You're monitoring needs to be customized to the role of the service being monitored. Relying solely on the default monitoring template mean you may receive many incident tickets that are not really incidents. An incident is an impact or a degradation to the service that requires timely handling. Do not use the Incident Management process for informational alerts

7.1 Connection to SQL Server

The monitoring tool/script should connect to SQL Server in our inventory sheet and report the failure connect to any one of them

Time to poll: Every 5 min

Error: Critical

7.2 Suspect database count

The monitoring tool/scripts should connect to SQL Server in our control every 15 minutes and report list of all the database(s) in the server or environment.

Time to poll: Every 15 min

Error: Critical

7.3 Connection Check to database

Like connection to SQL server, we will check connection to database and report any connection issue with the database.

Time to poll: Every 5 min

Error: Critical

7.4 Percentage used space for filegroup (data and log)

The monitoring tool should poll the servers and get the alert generated for the percentage of used space for file group related to data files and log files of all the database in SQL Server

Time to poll: Every 5 min

Error: Major

7.5 SQL Agent job failure

The monitoring tool should check the SQL Server agent jobs status and report any failure of job(s).
DBA jobs should be configured for failure alerts to DBA team.
Application specific job monitoring and automatic email to application team rather than routing through DBA

Time to poll: Every 15 min

Error: Warning

7.6 Last Full database backup status

The monitoring tool report out the number of hours since last database backup

Time to poll: Every 60 min

Error: Critical

7.7 Last transaction log backup status

The monitoring tool report the last transaction log backup.

Time to poll: Every 60 min

Error: Critical

7.8 Login failure

The monitoring tool should check the login failure reported 20 times in 2 minutes time gap should be reported as “attack” warning to DBA team.

Time to poll: every 60 mins

Error: Warning.

7.9 Deadlock and blocking

We can include deadlock and or blocking metric based on your account’s requirement This does not fall under mendatory metrics

7.10 High availability monitoring

The monitoring should be enabled for all high availability features like Always On / mirroring. replication / log shipping. If the monitoring facility available in the monitoring tool then respective metrics should be enabled.

7.11 DBSPI MSSQL monitoring

Pleas er DBSPI related information in attached document here

DBMon_DBSPI_Monit
oring_Configuration_N

7.12 List of Events to be suppressed

Although suppression of events can vary from environment to environment. Below suppression of events was considered for one of the project in legacy HPE has yielded very good results. So its support teams wisdom on considering below event suppression.

- A time out occurred while waiting to optimize the query%
- The error is printed in terse mode because there was error during formatting%
- A read operation on a large object failed%
- A fatal error occurred while reading the input stream from the network%
- Time-out occurred while waiting for buffer%
- SQL Server restarted recently%
- Length specified in network packet payload did not match number%
- The pre-login packet used to open the connection is structurally invalid%
- Could not connect because the maximum number of '1' dedicated administrator%
- A system assertion check has failed

8 SQL Server/Database Security

Secure SQL server/database is the need of the hour. We need to make sure the SQL Server is safe and upto date all the time.

- Ensure the physical security of each SQL Server, preventing any unauthorized users from physically access your servers.
- Only install required network libraries and network protocols on your SQL Server instances.
- Minimize the number of sysadmins allowed to access SQL Server.
- As a DBA, log on with sysadmin privileges only when needed. Create separate accounts for DBAs to access SQL Server when sysadmin privileges are not needed.
- Assign the SA account a very obscure password, and never use it to log onto SQL Server. Instead, use a Windows Authentication account to access SQL Server as a sysadmin.
- Give users the least amount of permissions they need to perform their job.
- Use stored procedures or views to allow users to access data instead of letting them directly access tables.
- When possible, use Windows Authentication logins instead of SQL Server logins.
- Use strong passwords for all SQL Server login accounts.
- Don’t grant permissions to the public database role.
- Remove user login IDs who no longer need access to SQL Server.
- Disable the guest user account from each user database by using REVOKE CONNECT FROM GUEST.
- Don’t use cross database ownership chaining if not required. Never grant permission to the xp_cmdshell to non-sysadmins.
- Remove sample databases from all production SQL Server instances.
- Use Windows Global Groups, or SQL Server Roles to manage groups of users that need similar permissions.
- Avoid creating network shares on any SQL Server.
- Configure login auditing so you can see who has succeeded, and failed, to login.
- Don't use the SA account, or login IDs who are members of the sysadmin group, as accounts used to access SQL Server from applications.
- Ensure that your SQL Servers are behind a firewall and are not exposed directly to the Internet.
- In SQL Server 2005 and earlier, remove the BUILTIN/Administrators group to prevent local server administrators from being able to access SQL Server. In SQL Server 2008, the BUILTIN/Administrators group does not exist by default.
- Run each separate SQL Server service under a different Windows domain account.
- Only give SQL Server service accounts the minimum rights and permissions needed to run the service. In most cases, local administrator rights are not required, and domain administrator rights are never needed.
- When using distributed queries, use linked servers instead of remote servers. Remote servers only exist for backward compatibility.
- Do not browse the web from a production SQL Server instance.

- Instead of installing virus/antispyware protection on a SQL Server, perform scans from a remote server during a part of the day when user activity is less.
- Add operating system and SQL Server service packs and hot fixes soon after they are released and tested, as they often include security enhancements.
- Encrypt all SQL Server backups with a third-party backup tool, such as Red Gate SQL Backup Pro.
- Only enable C2 auditing or Common Criteria compliance if required, as they add significant performance overhead.
- Consider running a SQL Server security scanner against your SQL servers to identify security holes.
- Consider enabling SSL or IPSEC for connections between SQL Server and its clients.
- Rename or disable SA account
- Set service account to SQL Service only thru config manager so as to ensure that service account will have only required permissions rather adding it to local admins group

9 Capacity & Performance Management

This section covers the capability specific best practices around managing capacity proactively, via housekeeping scripts and regularly reviewing the capacity and performance of the systems. The primary goal here is to ensure the customers systems have the resources to maintain an available service

9.1 Proactive capacity management

Proactive capacity management for filesystems is one of most important tasks to improve service availability and reduce cost. There’s no need for urgent or emergency changes when proactive capacity management is handled properly. It will also minimize number of false threshold alarms, configuration issues due to insufficient time for planning and resourcing.

10 Capacity management

This section will cover specific best practices around changes & recommended changes that should be converted into standard changes that reduce Change Administration. Also covered are any automation scripts that can be used to perform changes.

10.1 Normal change

Each normal change should have clear Install plan, backout plan, communication plan and impact description. Example below shows what is important in change planning:

Database Patching Change – needs to have full visibility by both Applications and OS teams, as well as considering the business impact on failure

10.2 Standard Change

Standard changes will increase speed and customer satisfaction and also reduce cost. Therefore it’s important that each normal change is evaluated after implementation if it should be converted to standard change.

A Change should be converted to standard change if:

- change does not cause downtime impact for end users
- change repeats often enough (for e.g. once per month)
- change plan does not change

Contact account change manager to get successfully implemented normal change converted to standard change. Change manager is responsible to get customer approval for standard change and get it added to change management tool.

There are still accounts where standard changes are not used effectively and standard change catalog does not exist. Contact Account Service Manager and Account Delivery Lead if standard change catalog does not exist.

Example: Add new datafile to tablespace realizing more disk space is needed to Database filesystem/drive.

11 Availability Management

As a Delivery Lead you are expected to focus on ensuring the Availability Assurance Team members work together to drive the plans that ultimately mean that the agreed Service Level Agreements are met for your customer. An understanding of how availability is calculated and what is means for your service is important. If you have not been received the NGDM Availability training please contact your Center Coach or manager.

All Database Clusters includes Active/Passive,

Alwayson and other cluster solutions that are in production must be tested at least once per year through the execution of a Failover Test Case or successful failover in normal operation.

For systems that are in production, the Cluster Failover Tests should be run during planned maintenance breaks and must be performed using the mentioned general process and technical procedures, customized as required for the client specific implementation.

Cluster previous testing results (or live failover) need to be documented re-verifying those outcomes are still operating as expected.

Regular and consistent testing of the cluster failover is a necessary procedure for system administrators and application support to have confidence that the appropriate automated action will be taken and that the service level for availability will be met. The Base Failover Test will reveal the most common causes of errors in cluster configuration and functionality and will reduce the possibilities of RtOPs, which many times arise as a contributor factor in the loss of service.

Required documentation for testing of high availability solutions are in the process of development

12 Service Level management

Typically the SLA we have agreed in our contract is the SLA we should have configured in our systems, this is how our Availability SLA's are reported out to our clients. Please ensure that Incident and Service Call SLA's are properly configured in the work flow systems and that System and Instance Availability Targets are accurate in the CMDB.

13 Risk Management

Risk Management are coordinated activities to direct and control an organization with regard to risk.

Risk is the effect of uncertainty on objectives. An effect is a deviation from the expected — positive and/or negative. Objectives can have different aspects (such as financial, health and safety, and environmental goals) and can apply at different levels (such as strategic, organization-wide, project, product and process). Risk is often characterized by reference to potential events and consequences, or a combination of these. Risk is often expressed in terms of a combination of the consequences of an event (including changes in circumstances) and the associated likelihood of occurrence. Uncertainty is the state, even partial, of deficiency of information related to, understanding or knowledge of an event, its consequence, or likelihood

14 HA and DR

Having an effective HA & DR policy helps companies to avoid business downtime and achieve maximum productivity. There are some solution to have High Availability and Disaster Recovery in SQL Server

14.1 Replication

Replication is a good solution for Disaster Recovery. But in this process, we do not copy the entire database. Rather, we copy some certain components like tables, views and other objects. It means that the secondary SQL database is not a carbon copy of the primary database. Since the replicated database provides delayed response, it is not considered as a solution for High Availability.

14.2 Clustering

Clustering is a nice way to ensure High Availability. It covers us during Server failure but does not work in case of SAN failure. When Storage area network (SAN) does not work, cluster also goes down. For this reason, clustering cannot be used during Disaster Recovery.

14.3 LogShipping

Just like the earlier two methods, log shipping also involves both primary and secondary databases. With this method enables, log backups of primary database will take place in the secondary database in a scheduled interval(e. g. after every 5 minutes.) However, the problem arises if the Server closes down within this 5 minutes. Then you lose your data. Despite this disadvantage, it can be used as a measure for Disaster Recovery.

14.4 Availability group / Mirroring

AG / Database Mirroring can be a part of HA & DR policy. We can have only one copy of the primary database. If we have more than that, we are unable to read from secondary server. The database in secondary server will be in recovery mode.

15 Migration Approaches

In typical Migration Projects, data housed in Database Server A (provided by Vendor A) is extracted, transformed, and loaded into Database Server B (produced by Vendor B). Each of these servers offers a wide array of functions to query data. Though the function names and syntaxes may vary, all the DB servers in the market provide maximum querying power and flexibility.

We will be providing the details in next version of this document.