SQL Server Best Practices



Best Practices for SQL Server in Azure VM

July 11, 2024

This document contains unpublished confidential and proprietary information of Odessa Technologies, Inc. No disclosure or use of any of these materials may be made without the express written consent of Odessa. ©1998-2020 Odessa Technologies, Inc. All Rights Reserved.

## Document revision history.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Description | Version | Author |
| Jan 18, 2023 | Created | 1.0 | Kishore |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Contents

[Document revision history. 2](#_Toc678828574)

[Contents 2](#_Toc232313750)

[1 About the Manual 3](#_Toc550191593)

[1.1 Objective 3](#_Toc1721095538)

[1.2 Intended Audience 3](#_Toc5144166)

[1.3 Pre-Requisites 3](#_Toc2048513167)

[2 SQL Server Best Practices 3](#_Toc1624167103)

[3 Local group policies - database instant file initialization - Add account used by SQL Server services: 4](#_Toc2088654908)

[3.1 Configuration Steps 4](#_Toc1233193931)

[4 E drive host caching (type read-only) where application’s MDF file resides: 5](#_Toc1992063393)

[4.1 Overview 5](#_Toc100458574)

[4.2 Configuration Steps 5](#_Toc1079923052)

[5 Create Temp DB files in D drive: 6](#_Toc291039632)

[5.1 Overview 6](#_Toc272589794)

[5.2 Configuration Steps 6](#_Toc516924078)

[6 Set all Temp DB files size to some%: 6](#_Toc1983568502)

[6.1 Configuration Steps 6](#_Toc1875710796)

[7 Memory allocation ~ 85% of system memory: 7](#_Toc917154936)

[7.1 Overview 7](#_Toc2027375468)

[7.2 Configuration Steps 7](#_Toc1227523797)

[8 Cost for threshold parallelism and Max DOP: 7](#_Toc960350942)

[8.1 Overview 8](#_Toc659335128)

[8.2 Configuration Steps 8](#_Toc1481504155)

[9 Compatibility level for all databases: 9](#_Toc2123314728)

[9.1 Overview 9](#_Toc1742344098)

[9.2 Configuration Steps 9](#_Toc2029580247)

[10 Application DB’s MDF & LDF – Set “Auto growth” and “Initial size”: 10](#_Toc1011661133)

[10.1 DWH 10](#_Toc1029028518)

[10.2 DWH Configuration Steps 10](#_Toc294108148)

[10.3 Core, Partner Portal & Report Builder Config Recommendations 11](#_Toc1615652148)

[11 Place application DB’s MDF & LDF in separate disk: 12](#_Toc1181125373)

[11.1 Configuration Steps 12](#_Toc487896463)

[12 Job to updated statistics daily: 13](#_Toc873824134)

[12.1 Overview 13](#_Toc304770797)

[12.2 Configuration Steps 13](#_Toc1341595158)

[13 Job to rebuild indexes weekly: 14](#_Toc2118626981)

[13.1 Overview 14](#_Toc502573959)

[13.2 Configuration Steps 14](#_Toc301424760)

[14 Post Configuration Actions 15](#_Toc1432188473)

[15 Configurating Windows Updates 15](#_Toc707690665)

[16 Configure DB Owner as “sa” for all Odessa DBs 15](#_Toc1737636868)

[17 References 16](#_Toc1893817416)

# About the Manual

## Objective

The objective of this document is to provide the best practices used in SQL server configuration so that server can provide efficient results. The below listed best practices are recommended by Microsoft. The reference link is provided below under the reference section.

## Intended Audience

The intended audience for this manual is the DBA team.

## Pre-Requisites

To configure the below practices in SQL server and DBA must have following permissions.

* Access to VM
* Local admin access within VM
* SQL Server is preinstalled and agent service in running state
* Index Optimize stored procedure must be present in the application database (say datawarehouse). If not exists, run attached script.



# SQL Server Best Practices

The following are best practices which is recommended to configure within SQL server for best results.

**VM Level Setting**

* Local group policies - database instant file initialization - Add account used by SQL Server services.
* E drive host caching (type read-only) where application’s MDF file resides

**SQL Server Settings I**

* Create Temp DB files in D drive.
* Set all Temp DB files size to some %.
* Memory allocation ~ 85% of system memory
* Cost for threshold parallelism and Max DOP

**Application Database(s) Settings**

* Compatibility level for all databaases
* Application DB’s MDF & LDF – Set auto growth to be set to 2gb.
* Place application DB’s MDF & LDF in separate disk.

**SQL Server Settings II**

* Setup job to update statistics daily.
* Setup job to rebuild indexes weekly.

The detailed configuration for above is list listed under each section.

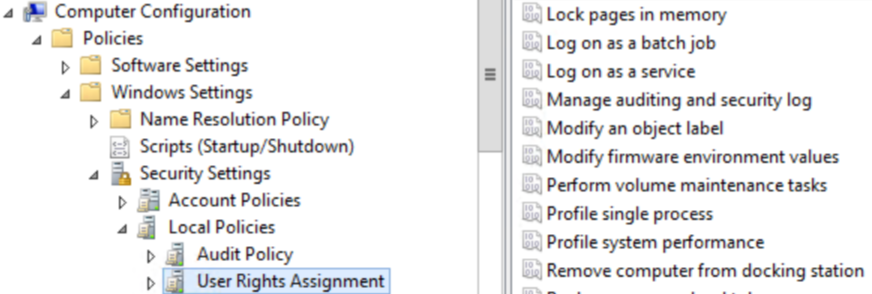
# Local group policies - database instant file initialization - Add account used by SQL Server services:

## Configuration Steps

This option is housed in the local security policies under > local policies > users’ rights assignments. You will add your user to the perform volume maintenances tasks and select apply.

Similar steps can be followed in group policy, however, depending on your structure for your group policy it can get a bit messy.

* Open **Group Policy Management**
* In the left pane, expand your Domain, and right click **Group Policy Objects**
* Create a new Policy with your domain’s naming convention.
* Once created, edit the policy by right clicking and selecting edit
* Navigate to Computer Configuration -> Polices -> Windows Settings -> Security Settings -> Local Policies -> User Rights Assignments
* Scroll down to the **Perform Volume Maintenance Tasks**



 Find the Domain account you use for your SQL Server Service

* In the case of multiple SQL servers with different accounts, create a group for them prior, if you have not already done so.

 Once added, apply the policy.

# E drive host caching (type read-only) where application’s MDF file resides:

## Overview

* Host caching for E drive is done to bring storage closer to the VM that can be written or read too quickly. For the VM hosting Datawarehouse, this must be set to Read-only.
* Drive hosting transaction log file (LDF file) should be set to “None.”

Refer below section.

## Configuration Steps

Login to Azure portal and select the VM & Go to **Disks** section.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LUN / Disk prefix number** | **Drive** | **Volume Name** | **Size** | **Host Caching Setting in Azure VM Disks** |
| 0 | F | Data | 2 TB | ReadOnly |
| 1 | G | Log | 2 TB | None |
| 2 | H | Backup | 2 TB | None |
| 3 | I | Data | 2 TB | ReadOnly |

A screenshot of a computer

Description automatically generated

# Create Temp DB files in D drive:

## Overview

Create multiple Temp db files and place them in the **temporary drive** (D: by default). Temp disk is for short term storage and using local storage on the physical host helps to increase IOPS and lower latency when compared to standard Azure storage.

## Configuration Steps

This can be done at the time of installation. Or can be done post the installation too, by running the below script:

**ALTER DATABASE tempdb MODIFY FILE (NAME = [tempdev], FILENAME = 'D:\Tempdb\tempdev.mdf');**

**ALTER DATABASE tempdb MODIFY FILE (NAME = [templog], FILENAME = 'D:\Tempdb\templog.ldf');**

**ALTER DATABASE tempdb MODIFY FILE (NAME = [temp2], FILENAME = 'D:\Tempdb\temp2.mdf');**

**ALTER DATABASE tempdb MODIFY FILE (NAME = [temp3], FILENAME = 'D:\Tempdb\temp3.mdf');**

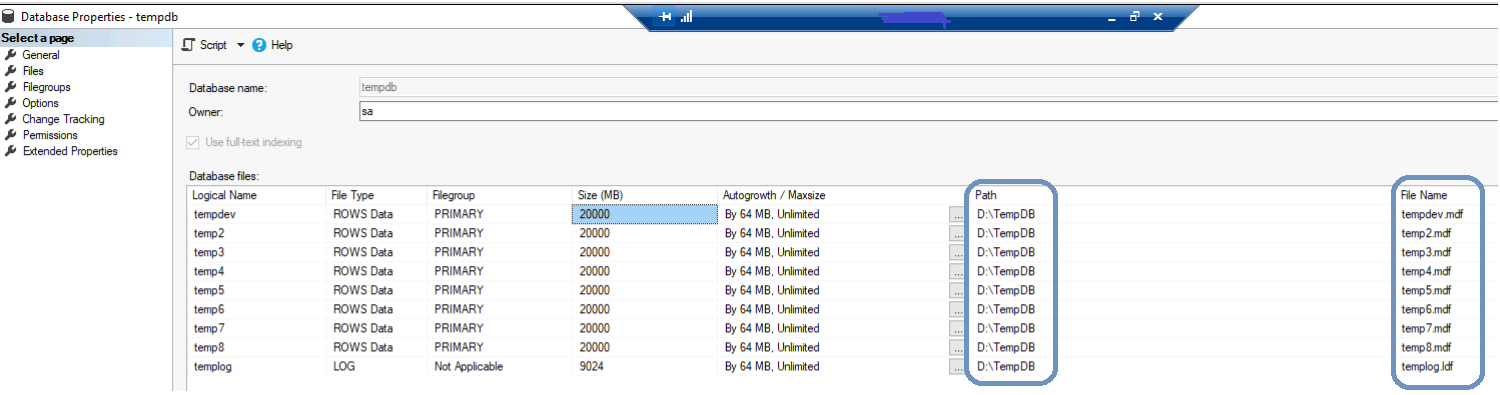
**ALTER DATABASE tempdb MODIFY FILE (NAME = [temp4], FILENAME = 'D:\Tempdb\temp4.mdf');**

**ALTER DATABASE tempdb MODIFY FILE (NAME = [temp5], FILENAME = 'D:\Tempdb\temp5.mdf');**

**ALTER DATABASE tempdb MODIFY FILE (NAME = [temp6], FILENAME = 'D:\Tempdb\temp6.mdf');**

**ALTER DATABASE tempdb MODIFY FILE (NAME = [temp7], FILENAME = 'D:\Tempdb\temp7.mdf');**

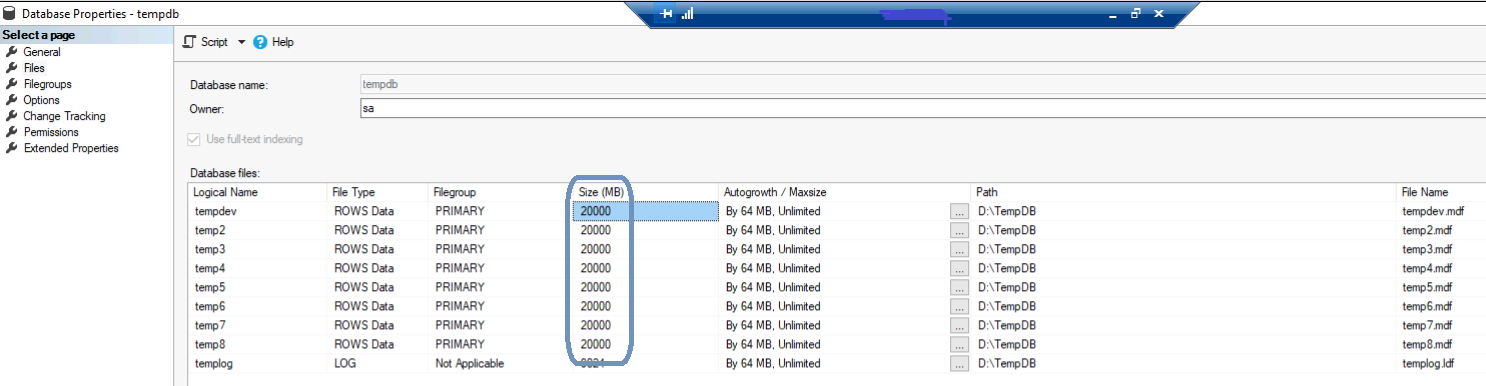
**ALTER DATABASE tempdb MODIFY FILE (NAME = [temp8], FILENAME = 'D:\Tempdb\temp8.mdf');**



# Set all Temp DB files size to some%:

## Configuration Steps

Right click on Temp DB and select properties in the object explorer of SSMS. Go to Files section and set the size to one third(approx.) and divide by 8 of temp disk of size - say 500GB disk, it must be 20GB (8\*20 = 160)



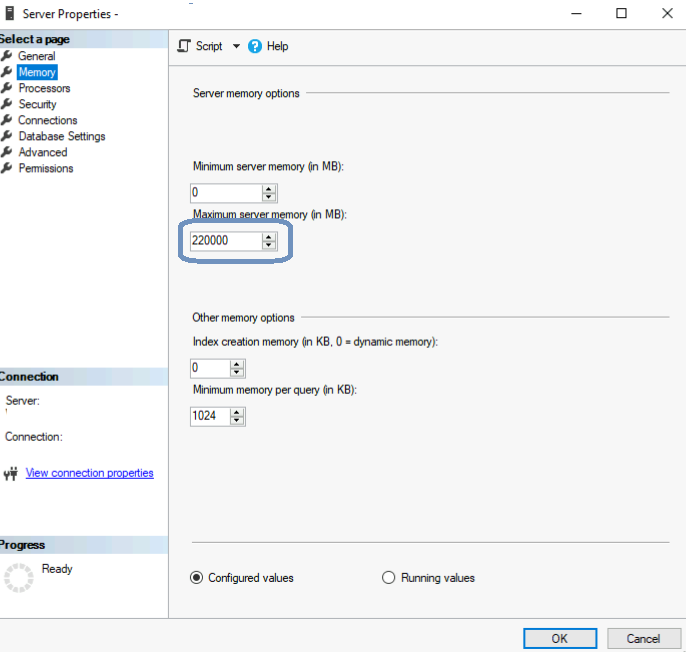
# Memory allocation ~ 85% of system memory:

## Overview

Rightly reserve the memory required for SQL server to perform efficiently and leave the remaining for application(s) and system tasks.

## Configuration Steps

Right click on SQL server instance and select properties in the object explorer of SSMS. Go to Memory section and set the max server memory to 85%(approx.) of server’s memory, say 220GB out of 256GB.



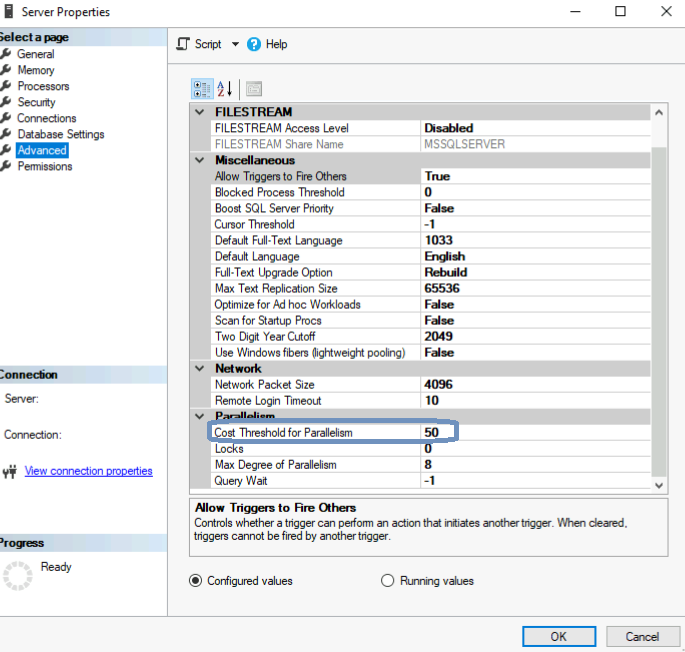
# Cost for threshold parallelism and Max DOP:

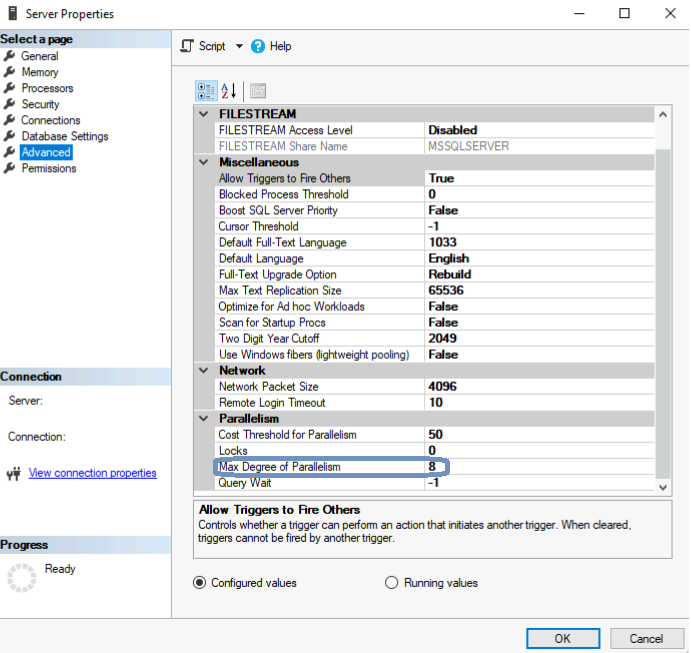
## Overview

The SQL Server query optimizer uses the Max degree of parallelism and cost threshold of Parallelism for determining whether the query will execute in serial or parallel mode. If the query cost is above the cost threshold of Parallelism, SQL Server runs the query parallelly.

## Configuration Steps

Right click on SQL server instance and select properties in the object explorer of SSMS. Go to Advanced section and set the **Cost for threshold parallelism to 50 and Max DOP to 8**.





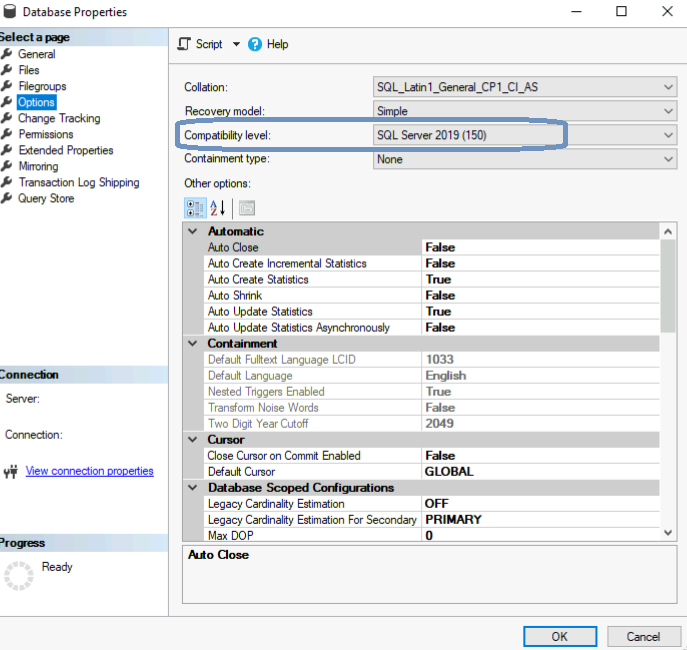
# Compatibility level for all databases:

## Overview

Compatibility level is an option associated with each database and it controls some database behaviors to be compatible with a particular version of SQL Server.

## Configuration Steps

In SQL Server Management Studio (SSMS), right-click on the database name, select Properties, select the Options section, click on the drop-down next to Compatibility level and select the **SQL Server 2019 (150).**



# Application DB’s MDF & LDF – Set “Auto growth” and “Initial size”:

## DWH

1. An auto-growth event is the process by which the SQL Server engine expands the size of a database file when it runs out of space.
2. Initial size Configuration – (One time configuration)
   * Before Day 0 ETL run – (refer Size (MB) column in below screenshot)
     1. The initial size for Data File 1 should be set to 100 GB/102400 MB.
     2. The initial size for Data File 2 should be set to 20 GB/20480 MB.
     3. The initial size for Log File should be set to 20 GB/20480 MB.

## DWH Configuration Steps

In SQL Server Management Studio (SSMS), right-click on the database name, select Properties, select the Files section, and set the **Autogrowth** to **500 MB** for LDF files, and **1024 MB** for MDF Files

A screenshot of a computer

Description automatically generated

## Core, Partner Portal & Report Builder Config Recommendations

|  |  |  |  |
| --- | --- | --- | --- |
| **PRODUCTION SQL MI (Core, PP & RB)** | | | |
| Database | Size Type | File Type | Recommended Size |
| Core | Initial Size | Data File | 270 GB |
| Core | Initial Size | Log File | 140 GB (143,360 MB) |
| Core | Auto Growth | Data File | 512 MB |
| Core | Auto Growth | Log File | 256 MB |
| Partner Portal | Initial Size | Data File | 30 GB |
| Partner Portal | Initial Size | Log File | 10 GB |
| Partner Portal | Auto Growth | Data File | 512 MB |
| Partner Portal | Auto Growth | Log File | 256 MB |
| Report Builder | Initial Size | Data File | 60 GB |
| Report Builder | Initial Size | Log File | 10 GB |
| Report Builder | Auto Growth | Data File | 512 MB |
| Report Builder | Auto Growth | Log File | 256 MB |

# Place application DB’s MDF & LDF in separate disk:

## Configuration Steps

MDF and LDF files of an application database can be placed in different disks (say E and F) at the time of creation. Or can be moved with the help of below scripts and steps to be followed.

1)Run this command on master DB

**ALTER DATABASE DATABASENAME MODIFY FILE (name='DATAFILENAME', filename=’Location')**

2)Run this command to bring the db offline

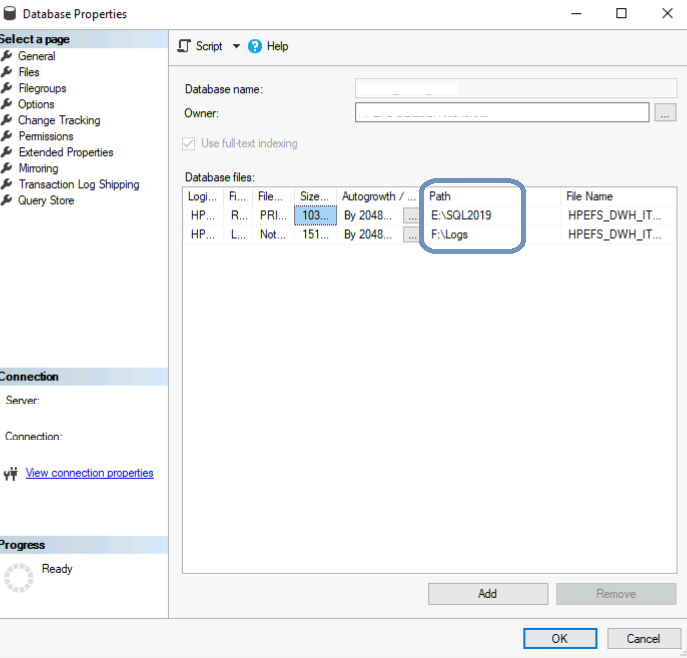
**ALTER DATABASE DATABASENAME SET OFFLINE WITH ROLLBACK IMMEDIATE;**

3)Move/Copy the DB file(s) to new location manually

4)Run this command to bring the db online

**ALTER DATABASE DATABASENAME SET ONLINE WITH ROLLBACK IMMEDIATE;**

**DATABASENAME – To be replaced with actual.**



# Job to be updated statistics daily:

## Overview

Updating statistics ensures that queries compile with up to date statistics

## Configuration Steps

Create a SQL Agent job and schedule it to run daily. Add below SQL query to execute as T-SQL script under step,

EXECUTE [dbo].[IndexOptimize]

@Databases = ' **DATABASENAME** ',

@FragmentationLow = NULL ,

@FragmentationMedium = NULL ,

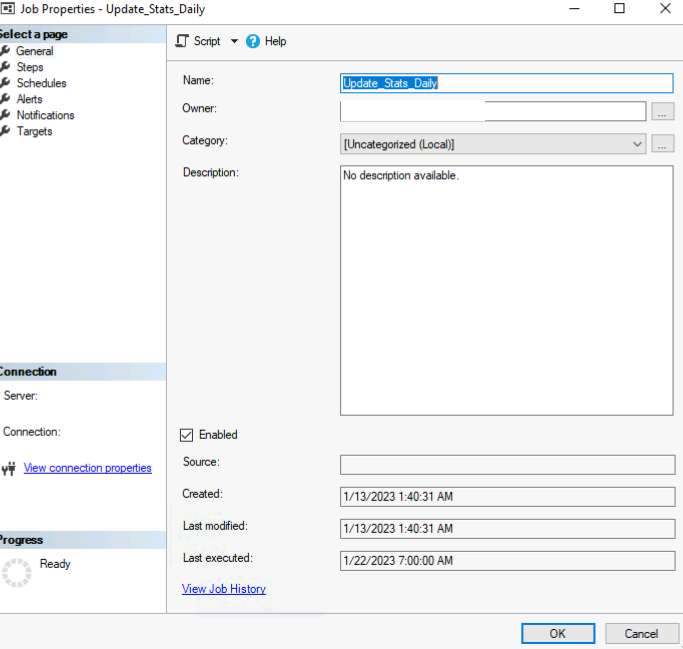
@FragmentationHigh = NULL ,

@UpdateStatistics = 'ALL' ,

@OnlyModifiedStatistics = N'Y' ,

@LogToTable = N'Y';

**DATABASENAME – To be replaced with actual.**



# Job to rebuild indexes weekly:

## Overview

Rebuilding an index drops and re-creates the index. This removes fragmentation, reclaims disk space by compacting the pages based on the specified or existing fill factor setting, and reorders the index rows in contiguous pages.

## Configuration Steps

Create a SQL Agent job and schedule it to run once in a week. As it is heavy performing activity, schedule it to run on Sunday.

Add below SQL query to execute as T-SQL script under step,

EXECUTE dbo.IndexOptimize

@Databases = ‘**DATABASENAME’**,

@FragmentationLow = NULL,

@FragmentationMedium = 'INDEX\_REORGANIZE,INDEX\_REBUILD\_ONLINE,INDEX\_REBUILD\_OFFLINE',

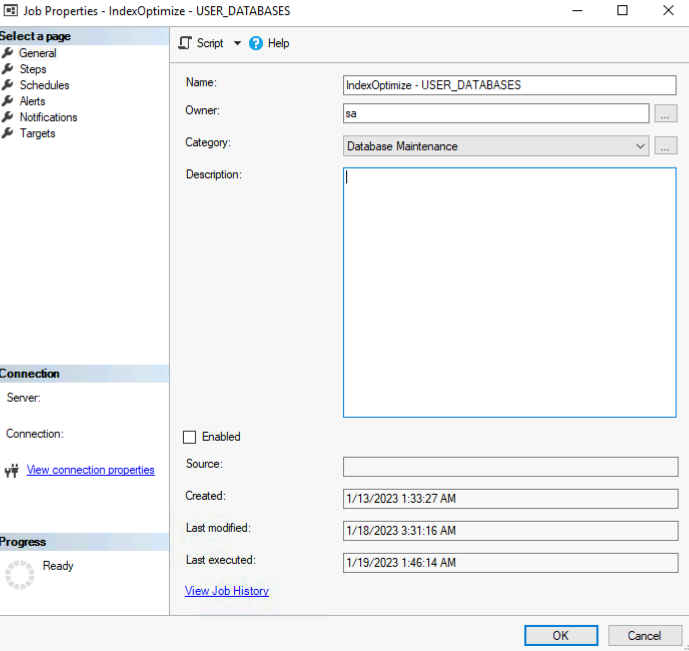
@FragmentationHigh = 'INDEX\_REBUILD\_ONLINE,INDEX\_REBUILD\_OFFLINE',

@FragmentationLevel1 = 20,

@FragmentationLevel2 = 30,

@UpdateStatistics = 'ALL',

@OnlyModifiedStatistics = 'Y'

**DATABASENAME – To be replaced with actual.**

# Database Integrity Check Job – Monthly

## Overview

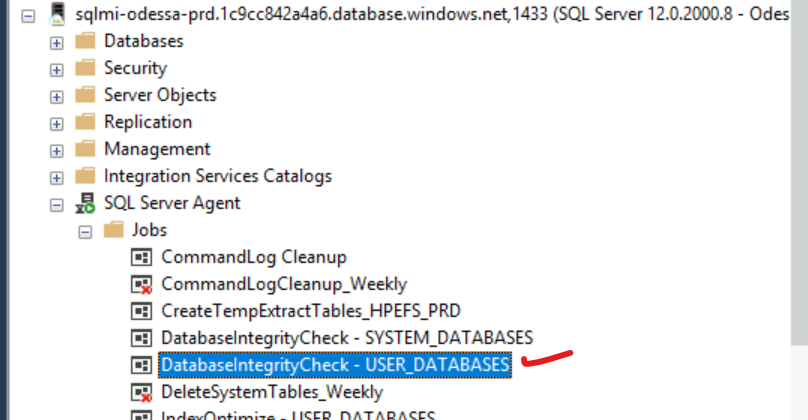
Database Integrity Check job (DatabaseIntegrityCheck - USER\_DATABASES) needs to be scheduled as per the HPE organization standards.

Please schedule it once in a **month or more frequently for Datamart database**.

This job is needed to check Database Integrity. Also, schedule alerts for this specific job to get alerted for any failures.

The same job is to be configured for both SQL MI databases and DWH DB.

**SQL MI:**



**DWH VM1:**

A screenshot of a computer

Description automatically generated

**DWH VM2:**

A screenshot of a computer

Description automatically generated

## Configuration Steps

# Post Configuration Actions

Restart SQL Server services

# Configurating Windows Updates

During VM creation, you should opt for manual patching instead of automatic to ensure the windows updates are scheduled when no business processes are running, and prior communication is sent to all stakeholders. For the existing VMs, if auto windows upgrade is enabled, then you can switch to manual by changing AUoption registry setting to 3. Refer to the link for details on the required setting:

[https://learn.microsoft.com/en-us/windows/deployment/update/waas-wu-settings](https://ind01.safelinks.protection.outlook.com/?url=https%3A%2F%2Flearn.microsoft.com%2Fen-us%2Fwindows%2Fdeployment%2Fupdate%2Fwaas-wu-settings&data=05%7C01%7CParesh.Joshi%40odessainc.com%7C42a1768d2d2a4b13d82508dba3b380bb%7Ce38a428600e849b39e527cb9a949eca7%7C0%7C0%7C638283763606280428%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&sdata=a1T6ee58nRGJ%2FcsfOrTA%2FkQ5OKiq13EGMxsLO%2BHf4%2BE%3D&reserved=0)

# Configure DB Owner as “sa” for all Odessa DBs

A screenshot of a computer

Description automatically generated

# References

The above best practices are recommended by Microsoft and those can be referred to in below link.

* <https://learn.microsoft.com/en-us/azure/azure-sql/virtual-machines/windows/performance-guidelines-best-practices-checklist?view=azuresql>