DBA database

This document is intended to provide guidance about the different functionalities included in the DBA database as complete solution for database maintenance and monitoring.

Database Settings requirements

* Ownership 🡪 [sa] or a member of the fixed server role [sysadmin]
* Trustworthy 🡪 ON

\* These two settings in combination might compromise the security of the instance due to privilege escalation if a malicious user that belongs to the database fixed role [db\_owner] execute scripts using EXECUTE AS [dbo].

DO NOT ADD ANY USER to the db\_owner role to avoid this issue.

Database tables

* DatabaseInformation 🡪 Contains all information for all monitored instances from sys.databases plus specific configurations per database, such as
  + BackupSchedule 🡪 Char(7) to define what kind of backup happen every day of the week (Sun-Mon)
  + BackupRootPath 🡪 Custom path for storing backups
  + BackupNFiles 🡪 Custom number of files the backups will be spread
  + BackupBatchNo 🡪 When there are many databases we can group them in batches so they can run in parallel or different schedules
  + KeepNBackups 🡪 Number of backups kept in the backup path
  + IndexMaintenanceSchedule 🡪 Char(7) to define which day of the week indexes will be maintained (Sun-Mon)
  + DBCCSchedule🡪 Char(7) to define which day of the week DBCC CHECKDB will run (Sun-Mon)
  + StatisticsMaintenanceSchedule🡪 Char(7) to define which day of the week stats will be updated (Sun-Mon)
* *DatabaseInformation\_History 🡪 historical information*
* *DatabaseInformation\_Loading 🡪 staging table*
* DatabaseSizeInformation 🡪 Historical information to see database growth trends
* DaysOfWeekBitWise 🡪 deprecated, schedules used to be based on bit values
* DBCC\_History 🡪 historical information about previous runs of CHECKDB on the current instance
* DisplayToID
* IndexFragmentationHistory 🡪 historical information about index fragmentation on the current instance
* IndexUsageStatsHistory 🡪 historical information about index usage on the current instance
* LastJobExecutionStatus 🡪 last ‘Database Maintenance’ job’s run information for all monitored servers
* LastMirroringStatus 🡪 last status of mirrored databases or that belong to any availability group in the current instance
* PowershellAuditScripts
* ServerAudit 🡪 Output of the server trigger Server Audit in the current instance, currently not in use.
* ServerConfigurations 🡪 Contains all information for all monitored instances from sys.configurations plus specific configurations per instance
  + trace\_flags\_enabled
  + BackupRootPath 🡪 Custom path for storing backups, database name will be added to this path
  + LogFilesRootPath🡪 Custom path for storing job output files, year, month, job name and date folders will be added to this path
  + KeepNBackups
* *ServerConfigurations\_History 🡪 historical information*
* *ServerConfigurations\_Loading 🡪 staging table*
* ServerDisksInformation🡪 Contains information of the different volumes present in every monitored instance.
* *ServerDisksInformation\_History 🡪 historical information*
* *ServerDisksInformation\_Loading🡪 staging table*
* ServerInformation 🡪 Contains information of the hardware and OS settings that might affect SQL Server present in every monitored instance
* *ServerInformation\_History 🡪 historical information*
* *ServerInformation\_Loading 🡪 staging table*
* ServerList 🡪 List of monitored SQL Server instances. It uses column encryption for passwords. Important columns are
  + MonitoringActive 🡪 Only monitored servers will be considered
  + isProduction
  + adminLogin 🡪 Used to connect remotely to servers and run monitoring procedures that require high privileges
  + adminPassword 🡪 Encrypted
  + monitoringLogin 🡪 Used to connect remotely to servers and run monitoring procedures
  + monitoringPassword 🡪 Encrypted
* ServerProperties 🡪 Contains the values of every SERVERPROPERTY() present in every monitored instance
* *ServerProperties\_History 🡪 historical information*
* *ServerProperties\_Loading 🡪 staging table*
* ServerStatus 🡪 Contains information of services running on every monitored instance
* ServerStatus\_Loading 🡪 *staging table*
* SQLServerKeywords 🡪 SQL Server reserved keywords, not in use
* SQLServerProducts 🡪 SQL Server different versions from 7.0 to 2017 with their code names
* SQLServerProductVersions 🡪 SQL Server different builds, not currently maintained (TODO: update the list from <https://sqlserverbuilds.blogspot.com/> preferably)
* StatsMaintenanceHistory 🡪 historical information about stats maintenance on the current instance
* TallyDate 🡪 Dates from 2018 till 2029
* TallyTime 🡪 times for a full day by second
* WindowsSku 🡪 not in use
* WindowsVersions 🡪 not in use

Stored procedures

* DBA\_AG\_failover
* DBA\_auditGetDatabaseInformation 🡪 Will return information about the databases present in the current instance
* DBA\_auditGetServerConfigurations🡪 Will return information about the different instance level configurations present in the current instance
* DBA\_auditGetServerDisksInformation 🡪 Will return information about the different drives present in the physical or virtual machine that hosts the current instance
* DBA\_auditGetServerInformation 🡪 Will return information about the hardware and OS present in the physical or virtual machine that hosts the current instance
* DBA\_auditGetServerProperties 🡪 Will return values from the function SERVERPROPERTY for the current instance
* DBA\_auditProcessDatabaseInformation 🡪 Will process the information contained in the staging table dbo.DatabaseInformation\_Loading and compare it to the one in dbo.DatabaseInformation. In case of differences, the current information will update dbo.DatabaseInformation and the obsolete data will generate a new row in dbo.DatabaseInformation\_History
* DBA\_auditProcessJobLastExecution 🡪 Will process the information contained in dbo.LastJobExecutionStatus and will send an email to the Database Administrators Operator
* DBA\_auditProcessServerConfigurations 🡪 Will process the information contained in the staging table dbo. ServerConfigurations\_Loading and compare it to the one in dbo. ServerConfigurations. In case of differences, the current information will update dbo. ServerConfigurations and the obsolete data will generate a new row in dbo. ServerConfigurations\_History
* DBA\_auditProcessServerDisksInformation 🡪 Will process the information contained in the staging table dbo.ServerDisksInformation\_Loading and compare it to the one in dbo. ServerDisksInformation. In case of differences, the current information will update dbo. ServerDisksInformation and the obsolete data will generate a new row in dbo. ServerDisksInformation \_History
* DBA\_auditProcessServerInformation 🡪 Will process the information contained in the staging table dbo. ServerInformation\_Loading and compare it to the one in dbo. ServerInformation. In case of differences, the current information will update dbo. ServerInformation and the obsolete data will generate a new row in dbo. ServerInformation\_History
* DBA\_auditProcessServerProperties 🡪 Will process the information contained in the staging table dbo. ServerProperties\_Loading and compare it to the one in dbo. ServerProperties. In case of differences, the current information will update dbo. ServerProperties and the obsolete data will generate a new row in dbo. ServerProperties\_History
* DBA\_auditProcessServerStatus 🡪 Will process the information contained in the staging table dbo. ServerStatus\_Loading and compare it to the one in dbo. ServerStatus. In case of differences, the current information will update dbo. ServerStatus and the obsolete data will generate a new row in dbo. ServerStatus\_History
* DBA\_backupSizeInfo 🡪 Will display backup information from the current instance that matches the parameters provided
  + @dbname
  + @backupType
  + @numBkp
* DBA\_backupStatusInfo 🡪 TBR
* DBA\_CHECKDBStatusInfo 🡪 Will display information from the current instance’s dbo.DBCC\_History table that match the parameters
  + @dbname, @numdays
* DBA\_cycleERRORLOG 🡪 TBR. Cycles the errorlog file for the current instance if bigger than the threshold
  + @ERRORLOG\_MaxSize
* DBA\_databaseAttach 🡪 Will return the scripts to Attach and Detach databases in the current instance that matches the parameter
  + @dbname
* DBA\_databaseFilesInfo 🡪 Will return information for database file for databases in the current instance that matches the parameters
  + @dbname
  + @fileType
* DBA\_databaseSizeInfo 🡪 Will return information for database size for databases in the current instance that matches the parameters. TBR 🡪 remove first result set as it is the same as DBA\_databaseFilesInfo, show only PIVOT data and remove @includeTotals parameter
  + @dbname
  + @includeTotals
* DBA\_databaseSnapshotCreate 🡪 Will create a database snapshot for the specified database of return the script to create it
  + @dbname
  + @debugging
* DBA\_databaseSnapshotRestore 🡪 Will restore a database snapshot for the specified database of return the script to create it
  + @snapshotName
  + @dbname
  + @deleteOtherSnapshots
  + @debugging
* DBA\_dataTypesFinder 🡪 Will return columns that use any of the specified data types or deprecated data types. TBR use string splitter function rather than using custom rubbish logic.
  + @dbname
  + @dataTypes
  + @onlyDeprecated
* DBA\_deleteOldBackups 🡪 Delete old backups for a given database or all databases within a given instance. This stored procedure is called by DBA\_runDatabaseBackup to clean up after taking a new backup, following the values defined in dbo.DatabaseInformation.
  + @instanceName
  + @dbname
  + @keepNbackups
  + @includeMockFile 🡪 This will display meaningful results in @debugging mode
  + @debugging
* DBA\_ExtendedEventFileReader 🡪 Returns the content of an Extended Event target file in a table format according to the current definition of the session
  + @sessionName
  + @filePath
  + @debugging
* DBA\_FTmonitoring 🡪 Returns information about Full text catalogs and index population. TBR remove sp\_foreachdb and filter the databases beforehand
  + @dbname
* DBA\_getBackupFileInformation 🡪 Returns header and label information from a backup file. This stored procedure is used in [dbo].[DBA\_restoreDBfromBackup] to perform some checks over the files to be restored
  + @FilePath
  + @BackupType
  + @BackupDatabaseName
  + @IsCopyOnly
  + @MediaSetId
  + @FamilyCount
  + @FamilySequenceNumber
* DBA\_getBackupFilesList 🡪 Returns the list of backup files (\*.bak, \*.trn) files from a given list of files
  + @path
  + @order
* DBA\_getDatabasesMaintenanceList 🡪 Returns list of databases for maintenance. It will exclude them from this list if the database is not ONLINE, is READ\_ONLY, is a snapshot, is SINGLE\_USER or is a non-readable replica (only for SQL 2012 onwards). TBR remove is non-readable because that will be determine by the SP that calls this other, as some tasks like backups do not require readability to happen.
* DBA\_getErrorInfo 🡪 Never used, deprecated
* DBA\_getServersToMonitor 🡪 Retuns a list of servers along with login and password to connect
  + @onePerMachine
  + @isAdmin
  + @server\_name
* DBA\_HEAPtablesFinder 🡪 Find all heap tables in user databases or a given database, and script out possible actions to create PK on them (to REVIEW and execute manually)
  + @dbname
* DBA\_help\_revlogin 🡪 TBR superseded by DBA\_SecurityAuditServerLogins
* DBA\_hexadecimal 🡪 TBR deprecated as you can get the same converting binary to string
* DBA\_indexCreateStatement
* DBA\_indexDescription
* DBA\_indexMaintenance
* DBA\_indexMissing
* DBA\_indexUsageStats
* DBA\_indexUsageStatsPersistsHistory
* DBA\_jobsDescription
* DBA\_jobsHistory
* DBA\_ListAllProgrammability
* DBA\_mirroringCurrentStatus
* DBA\_mirroringRoleSwitching
* DBA\_mirroringSendEmailAlert
* DBA\_nullableColumnsStats
* DBA\_parseDelimitedString
* DBA\_parseDelimitedString\_SQL2000
* DBA\_parseRelogOutput
* DBA\_perfmonDatabaseSyncCheck
* DBA\_renameJobLogFiles
* DBA\_restoreAllDBs
* DBA\_restoreDBfromBackup
* DBA\_runCHECKDB
* DBA\_runCHECKDB\_onBackups
* DBA\_runCHECKTABLE
* DBA\_runDatabaseBackup
* DBA\_runLogBackup
* DBA\_runningQueries
* DBA\_securityAuditDatabaseUsers
* DBA\_securityAuditServerLogins
* DBA\_securityCopyDatabaseUsers
* DBA\_selectTop1000rows
* DBA\_shrinkLogFiles
* DBA\_spaceused
* DBA\_statisticsMaintenance
* DBA\_strSQLSearch
* DBA\_syncDBAreplication
* DBA\_tableDescription
* rep\_GetPerfmonData
* repDatabaseSize
* repInstanceDatabases
* repServerDisks
* repServerInstances
* repServers

Checks done in the CMS

Job: SQL Server Daily Audit.

There should be a job running in the instance that holds the main central repository that will connect to every monitored server using PowerShell and check the following information using local stored procedures (executed as [dbo])

* Databases, new or changes in database settings. Local SP DBA\_auditGetDatabaseInformation
* Server configurations, changes in server configurations, such as max memory or max degree of parallelism and so on. Local SP DBA\_auditGetServerConfigurations
* Server properties, changes in server properties values, such as default paths, product version and so on. Local SP DBA\_auditGetServerProperties
* Last execution of the job ‘Database Maintenance%’, last execution status to avoid receiving individual emails for each server. Local SP DBA\_jobsHistory

The information collected will be temporarily stored in the DBA tables called xxx\_Loading.

Once all information is collected for each server, another set of stored procedures will run in the central repository instance to process the information, updates the current information, and generate historical data and send emails to @DatabaseAdministrators in case there were changes.

IMPORTANT: New databases are added to dbo.DatabaseInformation during this process with default values for the different maintenance tasks schedules, which are to perform every step (Index, Stats, CheckDB, Full Backup) every day.

If database DO NOT EXIST in dbo.DatabaseInformation, they will be ignored by the different stored procedures that perform the maintenance task leaving those databases unattended.

Job: SQL Server Services Monitoring

To monitor that all the monitored SQL Servers are up and healthy, there should be a job running every so often (Currently 5 minutes) that connects to every server using PowerShell and check the status of every service related to SQL Server, to send an email alert in case a service has changed its status.

This job will perform the following steps

* Collect the data using the PS script ‘SQLServicesCheck.ps1’
* Process the data collected in dbo.ServerStatus\_Loading using the local SP dbo.DBA\_auditProcessServerStatus

Job: Process Perfmon Files

Each monitored server will have a data collector that will output csv files to a shared location. There should be a job running every 1 hour executing a PowerShell script that will do the following for each monitored server

* Create a database for the server if does not exist
* Create a ODBC DSN to connect to SQL Server
* Look in the folder named after the server and process the csv files using relog.exe
* Move the files to a different folder once they are processed.

Job: Perfmon database syncing status check

Each monitored server will have a database that contains the data collected. There should be a job that will check the syncing status of every database that have performance data

Replication.

The DBA database is meant to be used as central repository and as local repository. To do so, the central repository will be the publisher of a two different publications

* DBA\_programmability
* DBA\_tables

DBA\_programmability (Snapshot)

This publication will contain some tables that do not require data, only their structure and stored procedures and functions that are required on each server to make possible the different maintenance tasks to function

The lists is as follows

* Tables (filtered by 1=0 to not replicate any data)
  + DBCC\_History
  + IndexFragmentationHistory
  + IndexUsageStatsHistory
  + LastMirroringStatus
  + ServerAudit
  + StatsMaintenanceHistory
* Procedures
  + DBA\_AG\_failover
  + DBA\_auditGetDatabaseInformation
  + DBA\_auditGetServerConfigurations
  + DBA\_auditGetServerDisksInformation
  + DBA\_auditGetServerInformation
  + DBA\_auditGetServerProperties
  + DBA\_backupSizeInfo
  + DBA\_backupStatusInfo
  + DBA\_CHECKDBStatusInfo
  + DBA\_cycleERRORLOG
  + DBA\_databaseAttach
  + DBA\_databaseFilesInfo
  + DBA\_databaseSizeInfo
  + DBA\_databaseSnapshotCreate
  + DBA\_databaseSnapshotRestore
  + DBA\_dataTypesFinder
  + DBA\_deleteOldBackups
  + DBA\_ExtendedEventFileReader
  + DBA\_FTmonitoring
  + DBA\_getBackupFileInformation
  + DBA\_getBackupFilesList
  + DBA\_getDatabasesMaintenanceList
  + DBA\_getErrorInfo
  + DBA\_HEAPtablesFinder
  + DBA\_help\_revlogin
  + DBA\_hexadecimal
  + DBA\_indexCreateStatement
  + DBA\_indexDescription
  + DBA\_indexMaintenance
  + DBA\_indexMissing
  + DBA\_indexUsageStats
  + DBA\_indexUsageStatsPersistsHistory
  + DBA\_jobsDescription
  + DBA\_jobsHistory
  + DBA\_ListAllProgrammability
  + DBA\_mirroringCurrentStatus
  + DBA\_mirroringRoleSwitching
  + DBA\_mirroringSendEmailAlert
  + DBA\_nullableColumnsStats
  + DBA\_parseDelimitedString
  + DBA\_parseDelimitedString\_SQL2000
  + DBA\_renameJobLogFiles
  + DBA\_restoreAllDBs
  + DBA\_restoreDBfromBackup
  + DBA\_runCHECKDB
  + DBA\_runCHECKDB\_onBackups
  + DBA\_runCHECKTABLE
  + DBA\_runDatabaseBackup
  + DBA\_runLogBackup
  + DBA\_runningQueries
  + DBA\_securityAuditDatabaseUsers
  + DBA\_securityAuditServerLogins
  + DBA\_securityCopyDatabaseUsers
  + DBA\_selectTop1000rows
  + DBA\_shrinkLogFiles
  + DBA\_spaceused
  + DBA\_statisticsMaintenance
  + DBA\_strSQLSearch
  + DBA\_syncDBAreplication
  + DBA\_tableDescription
  + repDatabaseSize
  + repInstanceDatabases
  + repServerDisks
  + repServerInstances
  + repServers
* Functions
  + fn\_SQLVersion
  + formatMStimeToHR
  + formatSecondsToHR
  + formatTimeToText
  + getBackupRootPath
  + getDayNamesList
  + getDriveFromFullPath
  + getFileExtensionFromFilename
  + getFileNameFromPath
  + getFilePathFromFullPath
  + getInstanceDefaultPaths
  + getNumericSQLVersion
  + getSpaceUsed
  + GMTdateTime
  + hexadecimal
  + ImplyingPermissions
  + UNQUOTENAME
* Views
  + ThisServer\_DatabaseInformation
  + ThisServer\_DatabaseSizeInformation
  + ThisServer\_LastMirroringStatus
  + ThisServer\_ServerDisksInformation
  + ThisServer\_ServerInformation
  + ThisServer\_ServerProperties

DBA\_tables (Transactional)

This publication will replicate a subset of the tables that are required on each server to make possible the different maintenance tasks to function.

Changes that might affect the behaviour of maintenance tasks, like the columns BackupSchedule, BackupRootPath, BackupNFiles, BackupBatchNo, KeepNBackups, IndexMaintenanceSchedule, DBCCSchedule, StatisticsMaintenanceSchedule in the table DatatabaseInformation must be changed in the central repository in order to be populated to the rest of the servers.

The tables that are replicated are

* DatabaseInformation
  + All columns
* DatabaseSizeInformation
  + All columns
* DaysOfWeekBitWise
  + All columns
* ServerConfigurations
  + All columns
* ServerDisksInformation
  + All columns
* ServerList
  + ID
  + Server\_name
  + ~~Server\_ip\_address~~
  + isProduction
  + ~~adminLogin~~
  + ~~adminPassword~~
  + ~~monitoringLogin~~
  + ~~monitoringPassword~~
* ServerProperties
  + All columns