# SQL Server: Maintenance Plans

## Module 4: Common Maintenance Tasks

Jonathan Kehayias
http://www.SQLskills.com/blogs/jonathan/
Jonathan@SQLskills.com

pluralsight
hardcore developer training

# Introduction

- **Database Maintenance Plans provide multiple tasks that can be configured to perform maintenance of SQL Server databases**
- **Understanding the most common tasks, how they work, and how to configure them is important in designing a Maintenance Plan**

- **In this module we'll cover:**
  - Back Up Database Tasks
  - Check Database Integrity Task
  - Rebuild Index Task
  - Reorganize Index Task
  - Update Statistics Task
  - History Cleanup Task
  - Maintenance Cleanup Task

# Maintenance Planning Considerations

- **Proper database maintenance covers multiple subject areas:**
  - Planning backups
  - Checking consistency
  - Index Maintenance
  - Statistics Maintenance
  - Cleaning up after Maintenance
- **As a part of planning your maintenance requirements it is important to understand the following topics:**
  - Recovery Point Objectives (RPO) and Recovery Time Objectives (RTO) (http://bit.ly/16x05Rb)
  - Recovery Models and Backup Types (http://bit.ly/14p9d4K)
  - Planning a Recovery Strategy (http://bit.ly/14p9zIn)

# Back Up Database Task (Full)

- **Performs a full backup of the specified databases**
  - Full backups do not allow the transaction log to clear space for reuse under the FULL recovery model
- **A full backup must be performed before you can perform other types of backups**
- **Backups should ideally be performed to a local disk for performance considerations and then copied to another location for storage**
- **Backups will be made using the .BAK file extension in the file system**
- **Frequency recommendations**
  - Depends on RPO and RTO requirements
  - A common strategy is to perform a daily full backup of the database

# Back Up Database Task (Transaction Log)

- **Backs up all the transaction log records generated since the previous log backup completed**
  - A log backup cannot be performed until a full backup has been performed
  - When a log backup completes, the inactive portion of the log is cleared, allowing the space to be reused, preventing log file growth
- **Performing log backups allows to you continue a restore operation started by restoring a full backup, by restoring the subsequent log backups to roll the database forward in time minimizing data loss**
- **Backups will be made using the .TRN file extension in the file system**
- **Frequency recommendations**
  - One or more times per hour based on RPO requirements
  - A common strategy is to perform a daily full backup of the database and 15 minute log backups to allow at most 15 minutes of data loss

# Back Up Database Task (Differential)

- **Backs up all the data that has changed since the most recent full backup**
- **Differential backups are used to speed up the restore process by replacing a number of log backups taken since the most recent full backup with a single file to be restored**
  - A differential backup is the net effect of all the log backups since the most recent full backup completed
- **Backups will be made using the .BAK file extension in the file system**
- **Frequency recommendations**
  - One or more times per day based on RTO requirements
  - Should be replaced by a new full backup if the size of the differential backup file exceeds 50% of the full database backup file size

# Check Database Integrity Task

- **The Check Database Integrity Task executes DBCC CHECKDB against the selected databases using the WITH NO_INFOMSGS option to check for corruption**
- **A common assumption is that database corruption will be immediately detected by SQL Server**
  - The reality is that SQL Server won't notice the corruption until the corrupt page is accessed from disk by the Database Engine
  - This could be days, weeks, or even months after the actual corruption occurred resulting in data loss for the database(s)
- **It is critical to detect database corruption as soon as it happens to minimize the risk of data loss associated with the corrupted portions of the database**
- **Frequency Recommendation**
  - Run at least once a week based on available maintenance windows

# DBCC CHECKDB Options

- **NOINDEX – skips the intensive checks of non-clustered indexes for user tables to decrease the overall execution time**
  - This option is the only one configurable in the Check Database Integrity Task
  - This is not recommended by Paul Randal, who wrote DBCC CHECKDB
- **DBCC CHECKDB offers many options that are not available within the configuration options of the Check Database Integrity Task:**
  - ALL_ERRORMSGS – shows all reported errors per object
    - This is the default configuration on SQL Server 2008R2 and onwards, but was not a default configuration on SQL Server 2005 or 2008
  - EXTENDED_LOGICAL_CHECKS – performs checks against indexed views, XML indexes, and spatial indexes when present (requires 100+ compatibility)
  - PHYSICAL_ONLY - only checks the integrity of the physical structure of the page and record headers and the allocation consistency of the database
  - DATA_PURITY - checks the database for column values that are not valid or out-of-range (required once post upgrade from SQL Server 2000)

# Dealing with Data Corruption

- **For the best information about how to appropriately deal with database corruptions reported by the Check Database Integrity Task, see Paul's course on SQL Server: Detecting and Correcting Database Corruption**
- **The Books Online content for DBCC CHECKDB shows a number of REPAIR options associated with the command**
  - These should only be used if you fully understand the potential outcome
  - Before running any REPAIR option, it is recommended that a FULL database backup be taken to provide a recovery point for the database
  - If you do not fully understand the implications of a specific corruption, you should consult an expert before running a REPAIR operation
  - Usually it is more practical to restore from backups rather than to achieve data loss

# Rebuild Index Task

- **Rebuilds the indexes specified in the task configuration:**
  - When multiple databases are selected in a single task, all indexes in the selected databases will be rebuilt
  - When a single database is selected, individual objects can be selected in that database to rebuild all of the indexes on the selected objects
- **As a side-effect all index based statistics are updated with fullscan**
  - Since all data is being read to rebuild the indexes computing statistics with fullscan has minimal overhead
- **Limitations**
  - Indexes are rebuilt regardless of the fragmentation level
  - Does not support partition level rebuilds for partitioned tables or indexes

# Rebuild Index Task (2)

- **Free Space options – specifies the free space to leave in the leaf level of the index pages during the rebuild**
  - Default free space per page – maintains the currently configured fill factor value for the indexes
  - Change free space per page to – specifies that the free space will be changed to the configured value as a part of the rebuild
    - Note: The SharePoint whitepaper for SQL Server maintenance incorrectly specifies 80% free space instead of 20%
- **Sort in tempdb option – specifies to store intermediate sort runs in the tempdb database rather than in the database the index is being rebuilt in**
- **Offers the ability to rebuild indexes online if using Enterprise Edition**
  - For indexes that can not be rebuilt online, this provides the option to rebuild offline or do nothing at all

# Reorganize Index Task

- **Should not be used with the same schedule as the Rebuild Index Task**
- **This task reorganizes indexes but does not include a update of the index statistics like the Rebuild Index Task does**
  - As a result, when the Reorganize Index Task is used a subsequent Update Statistics Task should also be included to update the column statistics of the databases
- **Reorganizing indexes is a fully online operation and can often be a good interim step between regular executions of the Rebuild Index Task for routine maintenance**

# Update Statistics Task

- **Updates the statistics specified in the task configuration:**
  - When multiple databases are selected in a single task, all statistics matching the Update specification are affected
  - When a single database is selected, individual objects can be selected in that database to update all statistics matching the Update specification on the selected objects
- **Update configuration allows:**
  - All existing statistics – updates all statistics on selected objects
  - Column statistics only – only updates column statistics
  - Index statistics only – only updates statistics associated with indexes
- **Scan type – determines the method used for updating the statistics**
  - Full scan- scans all of the rows in the table to update the statistics
  - Sample by – samples the table data based on the specified percentage reducing the amount of I/O required for updating the statistics

# History Cleanup Task

- **The msdb database stores historical information for troubleshooting and administrative tracking of:**
  - Backup and restore activity
  - SQL Server Agent job execution
  - Maintenance plan execution
- **This task deletes historical information that is older than the specified retention period, maintaining the size of the msdb database**
- **Frequency recommendation**
  - This task is normally scheduled once a week, but can be executed more/less frequently based on the server activity

# Maintenance Cleanup Task

- **As a part of ongoing maintenance tasks, backup files, maintenance plan reports, and other files may be created in the file system**
- **This task allows files of a specific extension to be deleted based on a specified retention period for the task**
    - □ Only one extension can be specified per Maintenance Cleanup Task, requiring multiple implementations of this task in a plan to clean up all of the necessary file extensions
- **Multiple Maintenance Cleanup Tasks can be configured using:**
    - □ The Maintenance Plan Designer, covered in Module 6
    - □ A SSIS Package, covered in Module 6
    - □ Multiple Maintenance Plans

# Summary

- **The tasks covered in this module are the most common tasks for routine maintenance requirements of SQL Server databases**
- **The Backup Database Task is one of the most important tasks you can implement, and should be implemented based on the RPO and RTO requirements for the databases**
  - Meeting your specific requirements may require a combination of Full, Transaction Log, and Differential backup tasks in the configuration
- **Routinely running the Check Database Integrity Task to detect corruption as soon as possible is the best way to minimize data loss**

- **The next module will look at:**
  - Other Tasks in Maintenance Plans