FERRIS STATE UNIVERSITY

SQL Server Backup, Recovery, and Turnover Management

By. Jack Pfleghaar

Contents

Introduction

In todays data driven organizations, the confidentiality, integrity, availability and overall security of information is essential to business continuity and long-term success. Databases are the foundation for nearly every critical business process. Due to this, a well-designed SQL backup and recovery strategy is essential. This paired by robust change management controls will ensure that all data remains accessible, protected, and reliable under all conditions. This paper will overview the tasks necessary to create a backup (with multiple methods) the tasks to restore a backup, and what is necessary for turnover management.

Backing Up a Database

The schedule, locations, retention periods, and decisions over database environments are all important factors to consider when backing up a database. We will begin by looking at three different schedules for when to back up a database. The first schedule will be considered a “Complete” backup, the second will be considered a “Medium” backup, and the third will be considered a “Simple” backup.

COMPLETE BACKUP

Complete	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
12:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
1:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
2:00 AM	FULL	FULL	FULL	FULL	FULL	FULL	FULL
3:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
4:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
5:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
6:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
7:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
8:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
9:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
10:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
11:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
12:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
1:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
2:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
3:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
4:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
5:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
6:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
7:00 PM	FULL	FULL	FULL	FULL	FULL	FULL	FULL
8:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
9:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
10:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
11:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG

Here you can see an example of a complete backup schedule with full backups daily at 2:00AM and 7:00PM. These both take place outside of work hours while the logs that take place every hour in

between will alert if there are any issues with the server. This would make it easier to roll back into a previous state and have solid knowledge where the issue occurred. This would be considered a backup schedule with good practice.

MEDIUM BACKUP

Medium	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
12:00 AM	FULL	FULL	FULL	FULL	FULL	FULL	FULL
1:00 AM							
2:00 AM							
3:00 AM							
4:00 AM							
5:00 AM							
6:00 AM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
7:00 AM							
8:00 AM							
9:00 AM							
10:00 AM							
11:00 AM							
12:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
1:00 PM							
2:00 PM							
3:00 PM							
4:00 PM							
5:00 PM							
6:00 PM	FULL	FULL	FULL	FULL	FULL	FULL	FULL
7:00 PM							
8:00 PM							
9:00 PM							
10:00 PM							
11:00 PM							

This is the example of a medium backup schedule. This schedule, similarly to the complete schedule, has two full backups a day, this one occurring at midnight and 6PM. In between this, every six hours there are logs taken. This would give this schedule the ability to also key into where issues may take place. This is much less in-depth than the complete backup schedule, yet it still completes two full backups a day.

MINIMUM BACKUP

Minimum	SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
12:00 AM	FULL						
1:00 AM							
2:00 AM							
3:00 AM							
4:00 AM							
5:00 AM							
6:00 AM							
7:00 AM							
8:00 AM							
9:00 AM							
10:00 AM							
11:00 AM							
12:00 PM	LOG	LOG	LOG	LOG	LOG	LOG	LOG
1:00 PM							
2:00 PM							
3:00 PM							
4:00 PM							
5:00 PM							
6:00 PM							
7:00 PM							
8:00 PM							
9:00 PM							
10:00 PM							
11:00 PM							

This is what would be considered a minimum backup schedule. In many ways this would be considered a bad schedule. Here, a backup occurs every Sunday at midnight with a log occurring every day at noon. This is a bad practice for the possibility of a problem occurring late Friday or during the day Saturday where to fix the issue you will need to replace an entire weeks worth of data. This should be considered the bare minimum for what a backup schedule should be.

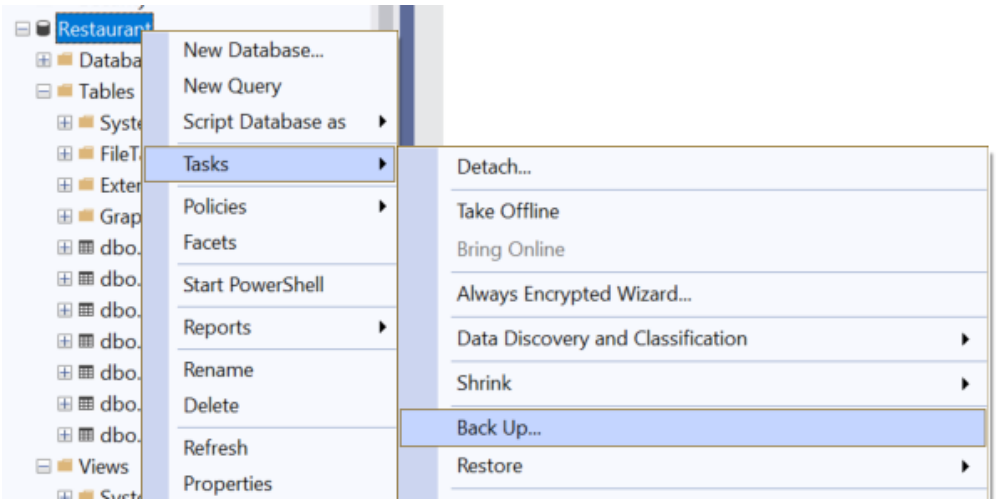
Creating a Backup and Recovering a Database

Creating or recovering a database can be done in one of two ways, by using the GUI or by using SQL queries. This section will cover both.

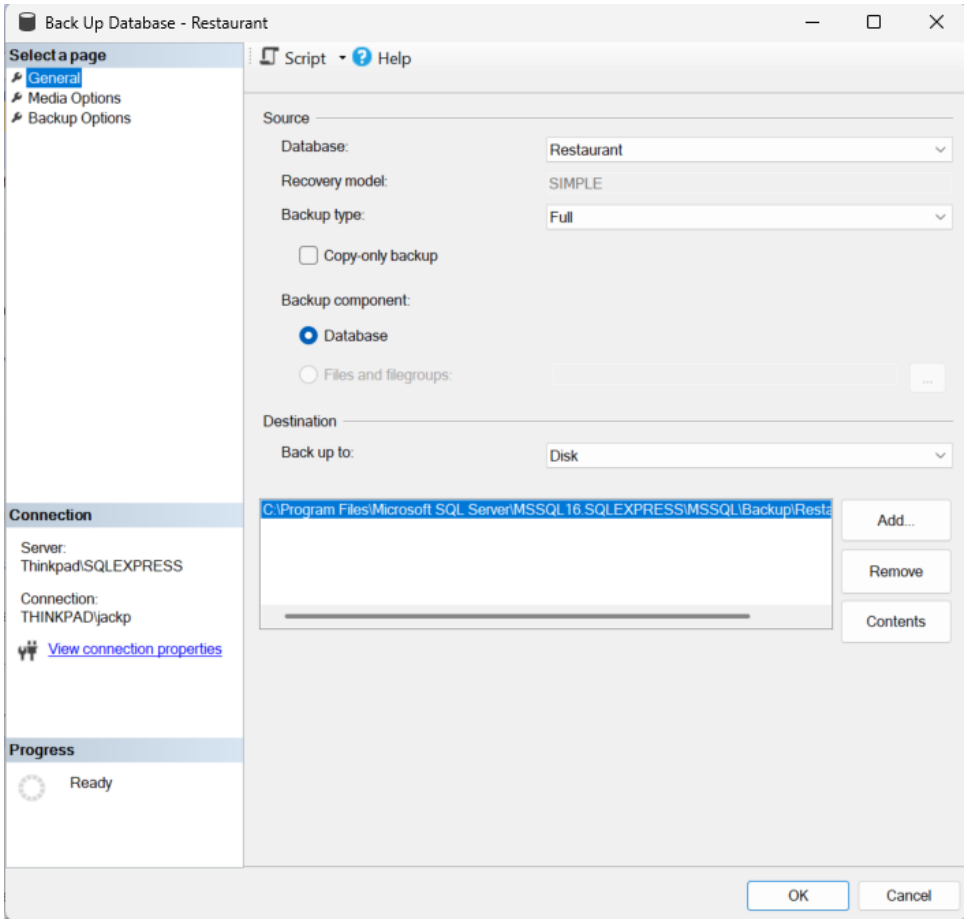
CREATING A BACKUP AND RECOVERING (GUI)

The following pictures will walk you through how to create a backup and recover a database in SSMS using the GUI. The database shown in these photos is a basic example database, but the steps should be the same globally.

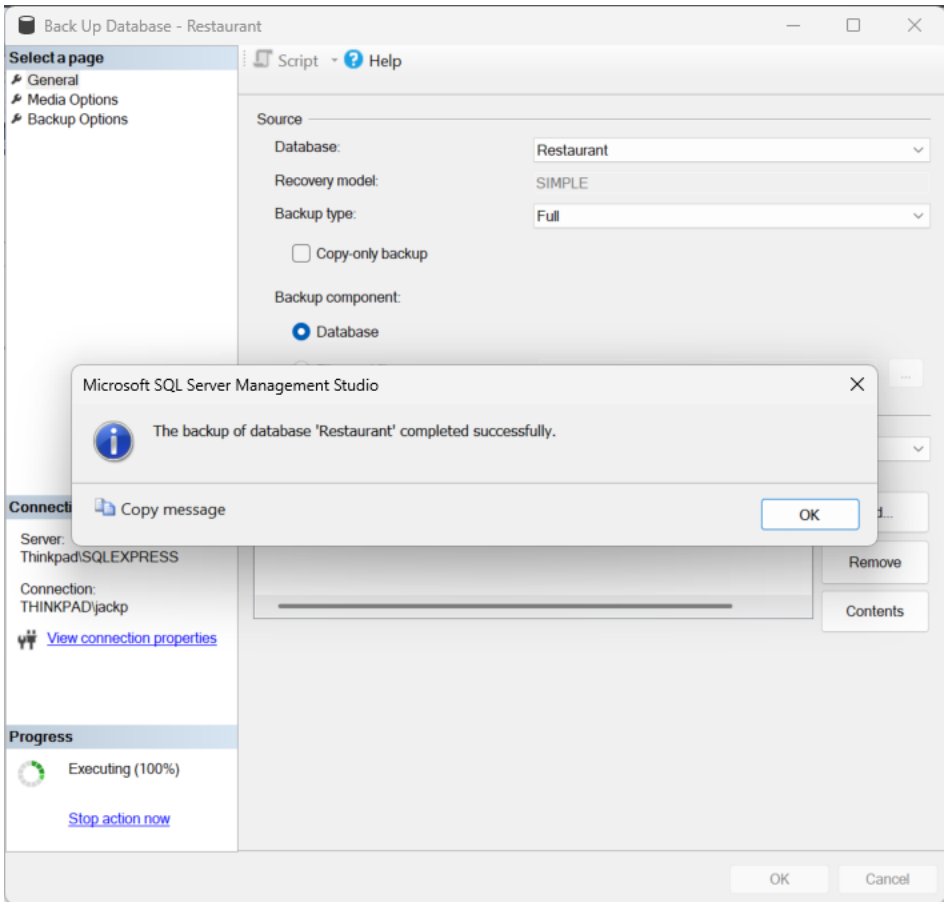
1. Begin by right clicking the database you want to back up and selecting “Tasks” > “Back Up”



2. This will bring you to a Back Up Database Wizard that looks like this:

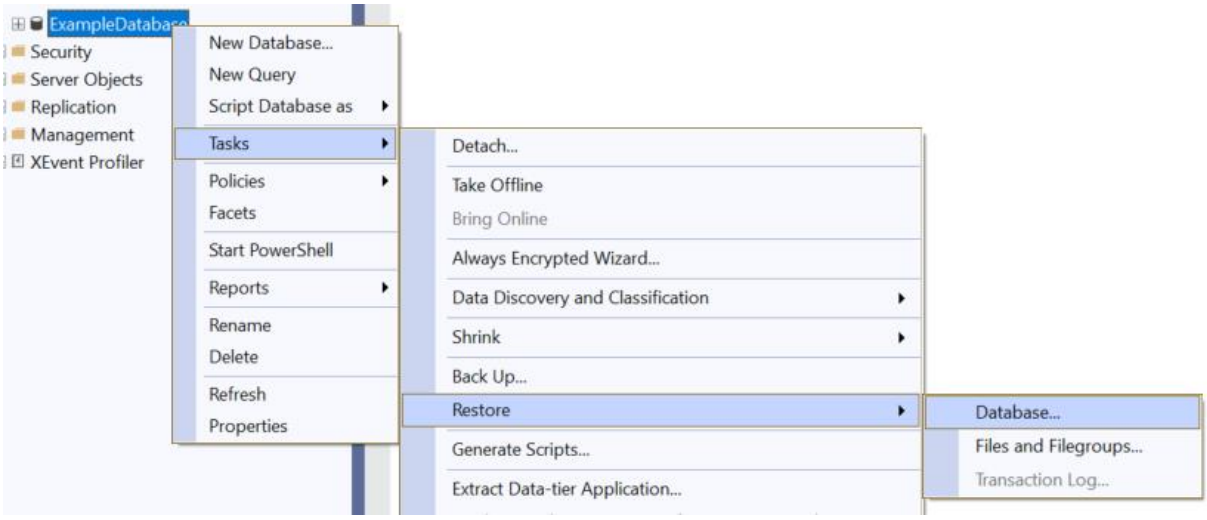


3. For my example I am going to use the defaults, however, this is where you would have the option to pick a back up destination, backup type, compression, retention period, and more.
4. Once prepared, click “OK” to create the backup. You should receive a screen like this:

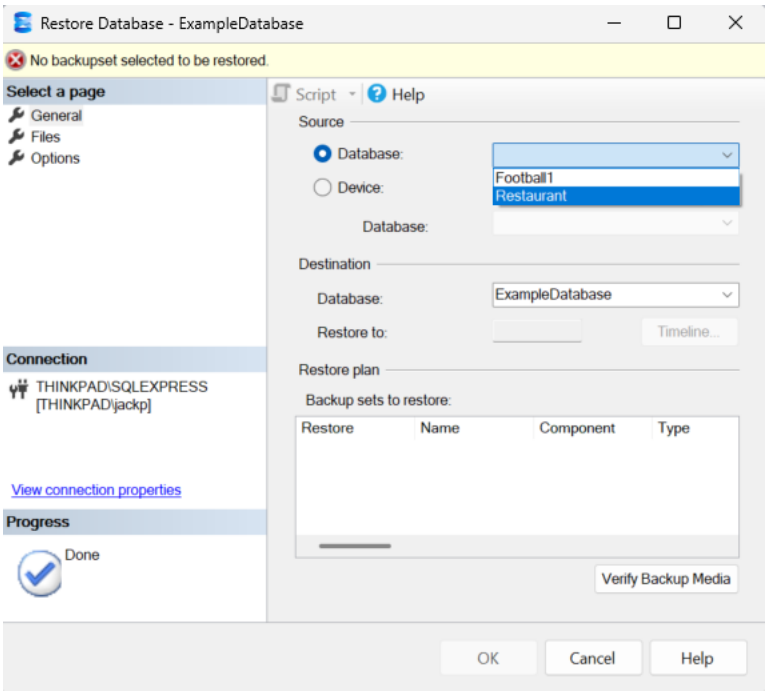


Restoring a database is just as simple. The process of doing this is as follows:

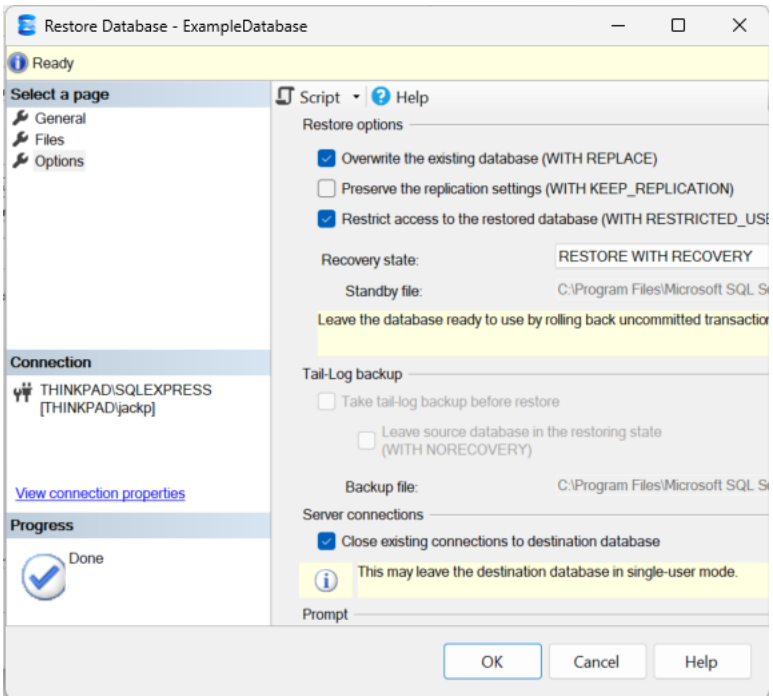
1. Begin by right clicking the database you want to retore and selecting “Tasks” > “Restore” > “Database”



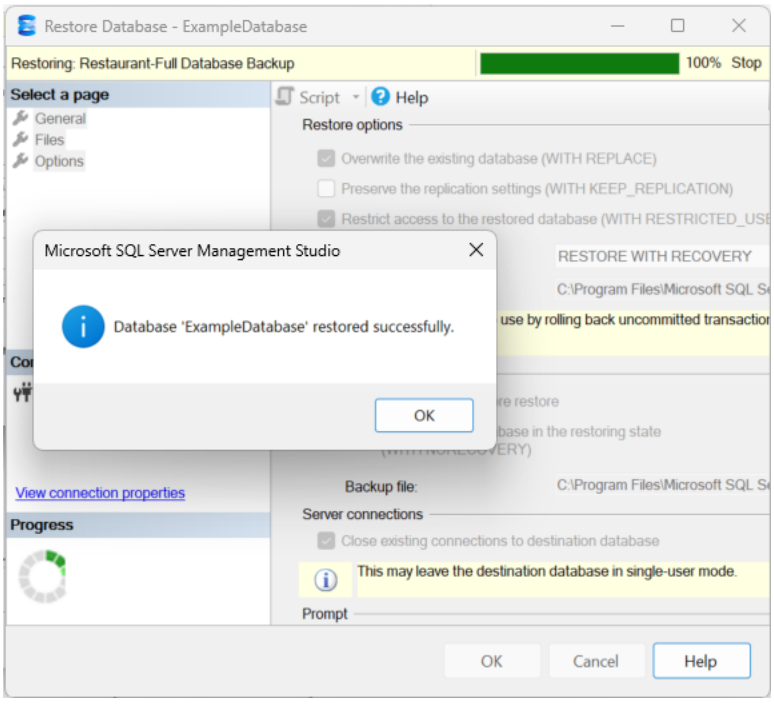
2. This will bring up a similar wizard to before. Now you need to select which database you want to restore. In my case this is the Restaurant database.



3. We are now going to chose any options that we may find important. I have selected “Overwrite the existing database (WITH REPLACE)”, “Restrict access to the restored database (WITH RESTRICTED_USER)”, and “Close existing connections to destination database”. These options will ensure the data is restores, that the database is not being altered as we are restoring, and that nobody has unrestricted access to the database during the restoration. The wizard with these selections can be seen here:



4. Click “OK” once all selections are completed to begin the restoration of the database.



CREATING A BACKUP AND RECOVERING (QUERY)

There are times when using SQL Queries may be more useful than using the GUI. I have included the following skeleton of a code that can be adjusted for individual use below.

The SQL code to create a backup or restore a database as a query would go as follows:

```
use Master
go
Backup Database [Database Name]
to Disk = 'C:\[Location]' –File path where the backup will be saved
With Format,
        INIT,
        Name = 'Full Backup of [Database Name] Database',
        Stats = 10; --Gives progress updates on the backup being taken

Restore Database [Database Name]
From Disk = 'C:\[Location]' –File path where the backup is
With Replace, -- Replace forces overwriting of the database if it exists
Stats = 10; --Gives progress updates on the database being restored
```

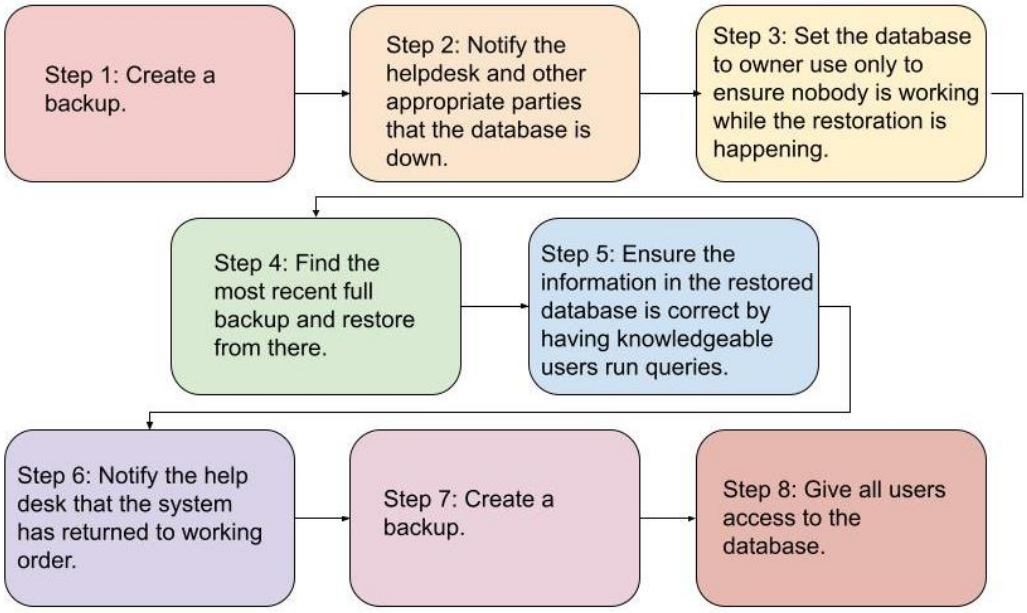
In the same way as before, we may want to remove people's connections to a database while working on restoration. The code for doing this is as follows:

```
sp_who --check all connections.
        --When you find a connection you want to kill (example: user 56) run     --the following code.

kill 56 --command to kill any connections to a db
```

Recovery Steps

I have attached a figure to represent the steps taken to restore a database and recover information. These steps go as follows:



Once backups are created, it is equally as important to determine where and how they are stored for long-term protection.

Database Environments

Once a backup is saved, you must consider things such as what environment you want to keep the database stored in, backup security, storage locations, support protocols, and more. Each subheading in this section includes details on all of these and more.

OVERVIEW OF DATABASE TYPES

Our organization operates across multiple database environments, each serving unique business functions and requiring tailored backup and recovery strategies. Due to this, we need to investigate a few types of databases. Transactional databases (such as ones used for order processing) require frequent log backups to ensure there are tight recovery points. Analytical databases and data warehouses are larger, read-intensive, and are backed up much less frequently with the trade off of being retained longer. Legacy systems (such as ones used for accounting) are updated infrequently and follow balanced schedules. For external hosted systems, such as the cloud-based HR database, backup and recovery depend on the vendor and their specific service level agreements (SLAs). All of these are used for different things and the differences must be considered when choosing which one is best for your institution.

BACKUP SECURITY

Backup security is integral to maintain data integrity and compliance measures. SSMS gives the option to encrypt using a multitude of encryption methods. Access to these backup files are then restricted through role-based access control, multi-factor authentication, or more. There should be weekly reviews to verify backup integrity, encryption key health, and access permissions.

STORAGE LOCATIONS

Backups should be stored in multiple locations to achieve redundancy. This ensures accessibility and disaster resilience. This gives us the ability to have primary on-premises copies for quick recovery and a secondary copy allocated to cloud storage. The entire backup should not be made fully cloud based or fully in-person though. By relying on an outdoor company to handle your data you may set yourself up in a situation where it takes a long time to restore and you need to ask another company for your rightful data. On the other hand, if all of the backups are in person and a disaster impacts the location, all your data will now be destroyed, and you will need to start from scratch.

ROTATION AND RETENTION

Daily backups are to be kept for seven days, weekly backups for up to four weeks, and monthly backups for up to one year. Any backups stored in a data warehouse are to be stored for seven years. There should be quarterly verification that all retained backups are recoverable and intact.

VERIFICATION AND TESTING

Routine testing validates both the reliability and performance of our recovery procedures. Quarterly test restorations should be performed on randomly selected databases to ensure the data can be fully recovered without corruption. Automated reporting tools will monitor daily backup jobs and generate alerts of any failures or anomalies. These processes will confirm that our data protection measure remain reliable across all environments.

Change Management and Turnover Control

There are two main forms of turnover control, simple and complex. Simple creates a low probability of resulting in a negative impact to an environment while complex could have a high impact if the change results in an issue. There are better and worse times to implement each one of these. For example, it may be better to do a simple change during the day and a complex change during the night. Due to this, there should be deadlines for submitting changes. Simple changes which may occur during the day or not long after the workday has completed should have the form completed and approved by noon the day of. For a complex change, the paperwork would need to be submitted and approved the day before. I have provided an example of a change request form.

The link to download this form can be accessed here: https://images.sampletemplates.com/wp-content/uploads/2016/09/30154238/Change-Request-Template-for-Software-Changes.zip

Change Request Template for Software Changes.pdf