

## SQL Server Backup & Restore Guide

SQL Server provides **multiple backup types** and **restore scenarios**, allowing fine-grained recovery depending on your **Recovery Point Objective (RPO)** and **Recovery Time Objective (RTO)**.

### PART 1 — Types of SQL Server Backups

There are **8 major backup types** in SQL Server (2019 and above):

#	Backup Type	Description	Common Use
1	Full Backup	Backs up entire database (all data + part of log)	Base backup for all others
2	Differential Backup	Backs up changes since last full backup	Faster than full, daily backups
3	Transaction Log Backup	Backs up only transaction log records since last log backup	Enables point-in-time recovery
4	File/Filegroup Backup	Backs up individual filegroups	Large DBs with partitioned data
5	Copy-Only Backup	Independent backup that doesn't affect backup chain	Ad-hoc backups
6	Partial Backup	Backs up primary + read-write filegroups only	For large DBs with read-only data
7	Tail-Log Backup	Captures unbacked log at the end before restore	Final backup before restore
8	Snapshot Backup (via VSS)	Taken by third-party tools using Volume Shadow Copy	Fast consistent backups in virtualized environments

#### 1 Full Database Backup

```
BACKUP DATABASE AdventureWorks
TO DISK = 'D:\Backups\AdventureWorks_Full.bak'
WITH INIT, COMPRESSION;
  • Includes all data files + portion of log for consistency.
  • Acts as the foundation for Differential and Log backups.
```

#### 2 Differential Backup

```
BACKUP DATABASE AdventureWorks
TO DISK = 'D:\Backups\AdventureWorks_Diff.bak'
WITH DIFFERENTIAL, COMPRESSION;
  • Contains data changed since last full backup.
  • Smaller & faster than full backup.
```

#### 3 Transaction Log Backup

```
BACKUP LOG AdventureWorks
TO DISK = 'D:\Backups\AdventureWorks_Log.trn'
WITH INIT, COMPRESSION;
  • Used only in Full or Bulk-Logged recovery model.
  • Enables point-in-time restore.
```

#### 4 File/Filegroup Backup

```
BACKUP DATABASE AdventureWorks
FILEGROUP = 'PRIMARY'
TO DISK = 'D:\Backups\AW_Primary_FG.bak'
WITH INIT;
  • Useful for very large databases with multiple filegroups.
```

## 5 Copy-Only Backup

```
BACKUP DATABASE AdventureWorks
TO DISK = 'D:\Backups\AdventureWorks_CopyOnly.bak'
WITH COPY_ONLY, COMPRESSION;
```

- Doesn't interfere with differential or log backup sequence.

## 6 Partial Backup

```
BACKUP DATABASE AdventureWorks
READ_WRITE_FILEGROUPS
TO DISK = 'D:\Backups\AdventureWorks_Partial.bak';

```

- Backs up only **read-write filegroups** and primary filegroup.
- Ideal for large data warehouses.

## 7 Tail-Log Backup

```
BACKUP LOG AdventureWorks
TO DISK = 'D:\Backups\AdventureWorks_TailLog.trn'
WITH NO_TRUNCATE;
```

- Taken before restore** to avoid data loss in an active DB.
- Ensures all committed transactions are preserved.

## 8 Snapshot / VSS Backup

- Triggered by **3rd-party tools** (e.g., Veeam, Commvault, Azure Backup).
- Uses **VSS Writer for SQL Server** to take application-consistent snapshots.

# PART 2 — Restore Scenarios in SQL Server

Here's how you **restore** depending on your **backup strategy** and **failure scenario** 

#	Restore Scenario	Sequence of Actions	Use Case
1	Restore Full Backup	RESTORE DATABASE ... FROM DISK ... WITH RECOVERY;	Basic restore
2	Restore Full + Differential	Restore Full → NORECOVERY → Restore Diff → RECOVERY	Faster restore using diff
3	Restore Full + Log(s)	Restore Full → NORECOVERY → Apply log backups in order → RECOVERY	Point-in-time recovery
4	Point-in-Time Restore	Restore Full → Logs until specific STOPAT time	Recover to exact moment
5	File/Filegroup Restore	Restore specific filegroup(s) → Apply log backups → RECOVERY	Partial restore for large DBs
6	Partial Restore	Restore primary + read-write filegroups	Recover main portion of DB
7	Piecemeal Restore	Restore critical FG first, others later	Large DBs, phased recovery
8	Restore with MOVE	Restore to different path or drive	Migration or DR
9	Standby Restore (Read-Only)	WITH STANDBY='UndoFile'	For log shipping secondary
10	Tail-Log Restore	Restore Tail Log → Apply to latest	When DB is damaged or offline

### ⌚ Example: Full + Log Restore Chain

-- 1. Restore full backup

```
RESTORE DATABASE AdventureWorks
FROM DISK = 'D:\Backups\AdventureWorks_Full.bak'
WITH NORECOVERY;
```

-- 2. Restore log backups in order

```
RESTORE LOG AdventureWorks
FROM DISK = 'D:\Backups\AdventureWorks_Log1.trn' WITH NORECOVERY;
RESTORE LOG AdventureWorks
FROM DISK = 'D:\Backups\AdventureWorks_Log2.trn' WITH RECOVERY;
```

### ⌚ Example: Point-in-Time Recovery

```
RESTORE DATABASE AdventureWorks
FROM DISK = 'D:\Backups\AdventureWorks_Full.bak' WITH NORECOVERY;
```

RESTORE LOG AdventureWorks

```
FROM DISK = 'D:\Backups\AdventureWorks_Log.trn'
WITH STOPAT = '2025-10-18T16:30:00', RECOVERY;
```

## 🔴 PART 3 — Common Real-Time Scenarios

Scenario	Required Backups	Typical Action
DB corruption	Full + Log + Tail Log	Restore sequence
Accidental delete	Full + Diff + Log	Point-in-time recovery
Drive failure	Filegroup backups	File-level restore
Log file full	Backup log to free space	Shrink or fix
Ransomware	Offline backup copy	Full restore from clean source
Disaster recovery	Copy backups offsite	Restore with MOVE to new server

## 🏁 Summary — Backup & Restore Matrix

Backup Type	Used For	Restorable As	Affects Backup Chain
Full	Baseline	Yes	Yes
Differential	Changes since full	Yes	Yes
Log	Log entries since last log backup	Yes	Yes
Copy-Only	Ad-hoc full	Yes	No
Partial	Read-write groups only	Yes	Yes
Tail-Log	Final log capture	Yes	No
File/Filegroup	Individual files	Yes	Yes
Snapshot	Instant image	Yes (via VSS)	No

There are indeed some **specialized or advanced backup methods** that aren't part of the standard Full/Diff/Log/File/Filegroup list but are used in certain scenarios. Here's a detailed breakdown:

## ⌚ Other Types of SQL Server Backups / Methods

### 1 Encrypted Backups

- SQL Server allows you to encrypt backups with **AES 128/192/256** or **Triple DES**.
- Ensures that backup files are secure and cannot be read without the certificate or key.

BACKUP DATABASE AdventureWorks

TO DISK = 'D:\Backups\AdventureWorks\_Encrypted.bak'

WITH ENCRYPTION

(

ALGORITHM = AES\_256,

SERVER CERTIFICATE = MyBackupCert

);

- **Use case:** Regulatory compliance (HIPAA, GDPR).

### 2 Compressed Backups

- Already mentioned optionally in standard backups, but technically a **backup optimization** method.
- Reduces disk space usage and can speed up network transfers.

BACKUP DATABASE AdventureWorks

TO DISK = 'D:\Backups\AdventureWorks\_Compressed.bak'

WITH COMPRESSION;

### 3 Mirror Backups

- SQL Server can create **backup mirrors**, writing a single backup operation to multiple destinations simultaneously.

BACKUP DATABASE AdventureWorks

TO DISK = 'D:\Backups\AW1.bak',

DISK = 'E:\Backups\AW2.bak'

WITH INIT;

- **Use case:** Redundant storage across disks or servers.

### 4 Partial & Piecemeal Backups

- I briefly mentioned this earlier, but **piecemeal** is slightly different:
  - You can restore **critical filegroups first** (primary + read-write) to make the database partially online while other filegroups restore in the background.
- **Use case:** Very large OLTP or DW databases (>1 TB).

### 5 Third-Party Snapshot / Hardware-Based Backups

- Tools like **Veeam, Commvault, NetApp snapshots** integrate with SQL Server via **VSS Writer**.
- Can create **instant, application-consistent backups** without SQL-native backup overhead.
- Often used in **SAN environments or cloud VMs**.

### 6 Differential File Backups

- Specialized variation where you take a **differential backup of a specific file or filegroup** instead of the entire DB.
- **Use case:** Large DBs with huge read-only filegroups.

**7 Backup to URL / Azure / Cloud**

- SQL Server supports backing up directly to **Azure Blob Storage** or **Azure Managed Disks**.

BACKUP DATABASE AdventureWorks

TO URL = 'https://myaccount.blob.core.windows.net/backups/AdventureWorks.bak'

WITH CREDENTIAL = MyAzureCredential;

- **Use case:** Cloud-native disaster recovery, offsite storage.

**8 Log Shipping / Continuous Copy**

- Not a backup in the classic sense, but **continuous transaction log shipping** to a secondary server acts as a live backup.
- Provides **near real-time DR**.

**9 File-Snapshot Backups for Large DBs**

- You can take **snapshots of individual files** via VSS or SAN-level snapshots.
- Very fast, minimal downtime.
- **Note:** Requires careful planning; not native SQL Server backup syntax.

**◆ Summary Table**

Type	Description	Use Case
Encrypted Backup	Backup secured with certificate/key	Compliance, secure storage
Compressed Backup	Backup compressed to save space	Disk/network optimization
Mirror Backup	Backup written to multiple destinations	Redundancy
Piecemeal/Partial	Restore critical FG first	Large DB recovery
Third-Party Snapshot	VSS or SAN snapshot	Fast, zero-downtime backup
Differential File Backup	Only changed files	Large DBs with huge files
Backup to URL/Cloud	Direct backup to Azure/Cloud	Offsite DR
Log Shipping	Continuous log copy	Near real-time DR
File-Snapshot Backup	Snapshot of individual files	Fast recovery, minimal downtime