Backup and retention strategies should scale with database size, criticality, and business RPO/RTO requirements.

Let's break this down for small, high, and very large databases with detailed recommendations, examples, and best practices.

\$\tilde{\mathbb{X}}\$ 1. Key Considerations Before Designing the Strategy

• RPO (Recovery Point Objective):

How much data loss is acceptable? (Minutes, hours, or a day)

• RTO (Recovery Time Objective):

How quickly must the database be restored?

Database Size:

Small (<100 GB), Large (100 GB-1 TB), Very Large (>1 TB)

Workload Type:

OLTP (frequent transactions) vs OLAP (read-heavy)

• Storage & Network:

Disk space, IOPS, throughput available for backups/restores

• Compliance Requirements:

Retention period (30 days, 1 year, 7 years for audit?)

2. Backup Strategy by Database Size

DB Size	Backup Strategy	Why This Works
- Full Backup: Nightly (or weekly if size allows) - Differential Backup: Every 4–6 hrs - Log Backup: Every 15–30 mins (if FULL recovery model)		Small DBs have shorter backup windows and fast restores.
Medium/High (100 GB – 1 TB)		Saves space and time — differentials reduce I/O overhead of frequent fulls.
тв)	Backup: Daily - Log Backup: Every 5–10 mins - File/Filegroup Backup: Optional for very large partitioned DBs - Backup	Full backups can take hours — striping + compression optimizes backup/restore. Filegroup backups speed up recovery for large data warehouses.

Example SQL Server Backup Command (Striped & Compressed)

BACKUP DATABASE [SalesDB]

TO DISK = 'E:\SQLBackups\SalesDB_1.bak',

DISK = 'F:\SQLBackups\SalesDB_2.bak'

WITH INIT, COMPRESSION, CHECKSUM, STATS = 10;

- Striping spreads I/O load across multiple drives.
- **✓ Compression** reduces backup size by 40–70%.
- Checksum ensures backup integrity.

Praveen Madupu

3. Retention Strategy by Database Size

DB Size	Retention Recommendation	Storage Best Practices
Small DB	ll- Keep 7 daily fulls . 4 weekly. 12 monthly - Keep logs for last 14 days	Can afford to store many copies due to smaller size.
Medium DB	backup to cheaper storage (NAS/Cloud)	Rotate backups to avoid consuming expensive SAN space.
Very Large DB	- Retain only last 1 full backup + differentials/logs on local storage - Move older full backups to Azure Blob/Glacier/cheap cold storage - For compliance, keep yearly snapshots offline	Saves expensive storage, but still meets audit/compliance needs.

Key Retention Rules:

- Use maintenance cleanup jobs or Ola Hallengren's @CleanupTime parameter.
- Move old backups to **offsite storage** (cloud, tape, or external disk).
- Periodically **test restores** retention policy is meaningless if backups are corrupt or restore takes too long.

4. Performance Optimization for Large Backups

- Backup Compression: Reduces size & improves speed.
- **Backup Encryption:** Use AES_256 for security.
- **Striping:** Use multiple backup files on different drives for faster writes.
- **Backup to URL:** Directly write to Azure Blob Storage for long-term archival.
- MaxTransferSize & BufferCount tuning: For VLDBs, tune parameters for optimal throughput.

5. Restore Strategy

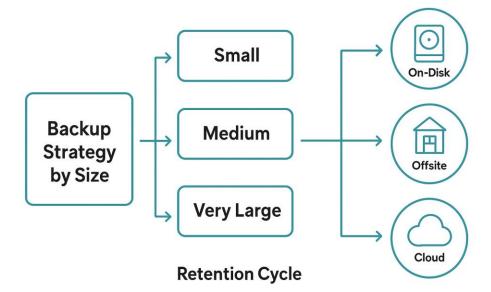
Your backup plan is only as good as your restore plan. Test:

- Point-in-time restores using log backups.
- Partial restores (file/filegroup for VLDB).
- Disaster recovery simulations at least quarterly.

III Example Strategy Summary Table

Size	Full Backup	Differential	Log Backup	Retention
Small (<100 GB)	lDailv	Every 4–6 hrs	15–30 mins	30 days on-disk, monthly archived
Medium (100 GB–1 TB)	Weekly	Daily	10–15 mins	14–30 days on-disk, monthly archived
Very Large (>1 TB)	Weekly (Striped + Compressed)	Daily	5–10 mins	7–14 days on-disk, monthly/yearly archived to cheaper storage

- **Best Practices**
- Automate backups using SQL Agent or Ola Hallengren's DatabaseBackup job
- ✓ Use **CHECKSUM + VERIFYONLY** after backups
- ✓ Monitor backup job failures with alerts (email/Teams/Slack)
- ✓ Document RPO/RTO expectations with business owners
- Periodically test **full restore time** to ensure it meets RTO



amps.com