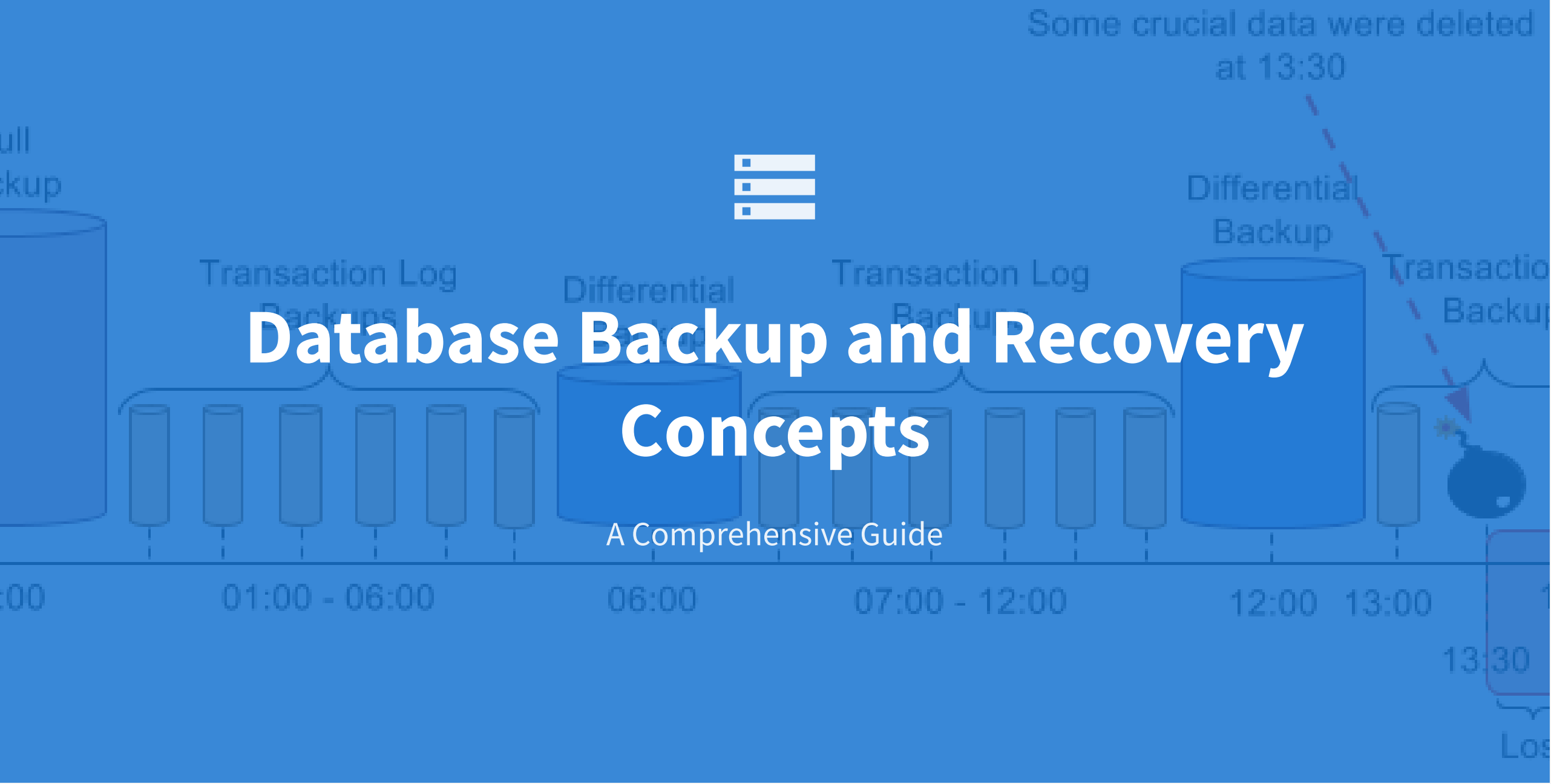
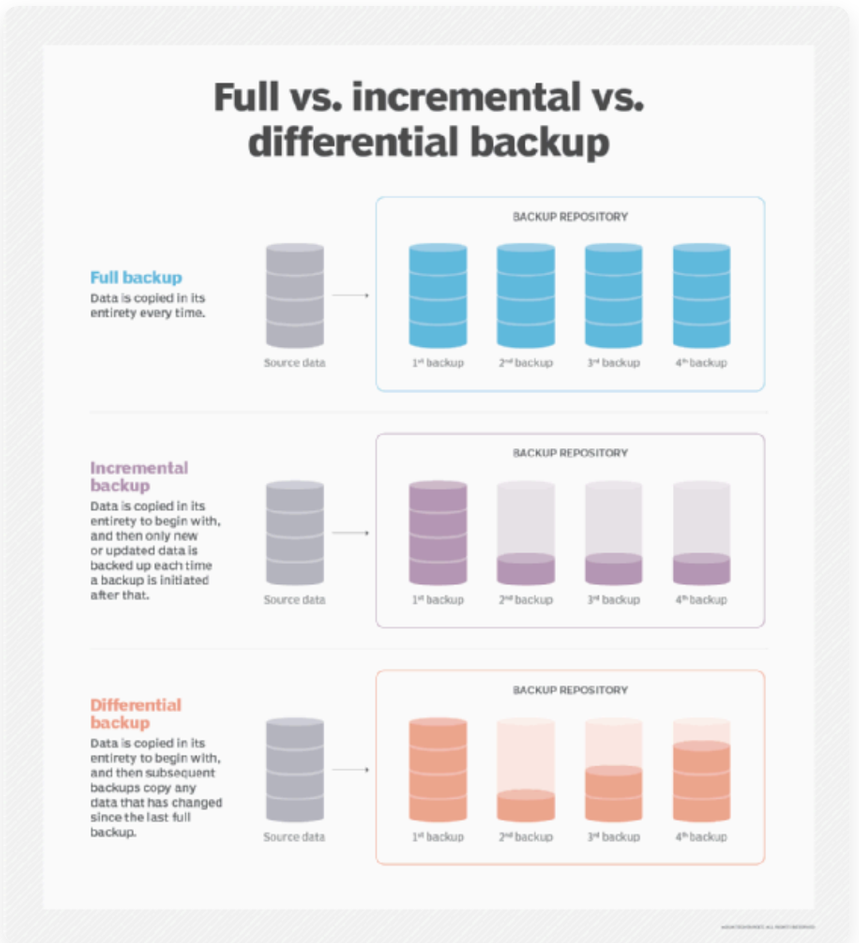


Database Backup and Recovery Concepts

A Comprehensive Guide

Backup Types



Full Backup

Complete copy of all data

Pros

Quick restore, easy storage management

Cons

Most storage space, longest backup time

Best for: *Small businesses with small data volumes*

Incremental Backup

Backs up files changed since the last backup

Pros

Efficient storage use, fast backups

Cons

Time-consuming restoration, requires all backup files

Best for: *Businesses with large data volumes*

Differential Backup

Backs up files changed since the last full backup

Pros

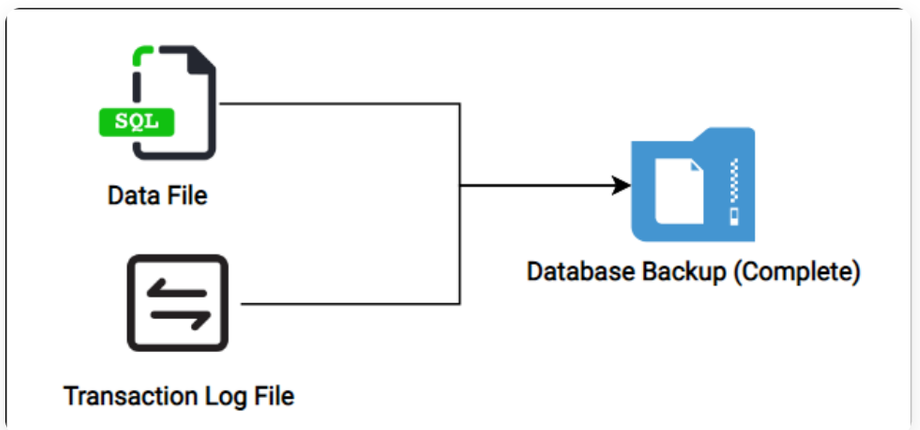
Faster restoration than incremental, less space than full

Cons

More storage than incremental, slower restoration than full

Best for: *Medium-sized organizations*

Recovery Models



Simple Recovery Model

No transaction log backups. Log space reused when checkpoint occurs.

Pros

No administrative overhead

Cons

Risk of data loss, no point-in-time restore

Best for: *Development/test databases, reporting databases*

Full Recovery Model

All transactions fully recorded in transaction log. Log sequence preserved.

Pros

Minimal data loss, supports point-in-time restore

Cons

Transaction log grows infinitely, requires regular log backups

Best for: *Mission-critical applications, high availability*

Bulk-Logged Recovery Model

Similar to Full but bulk operations minimally logged.

Pros

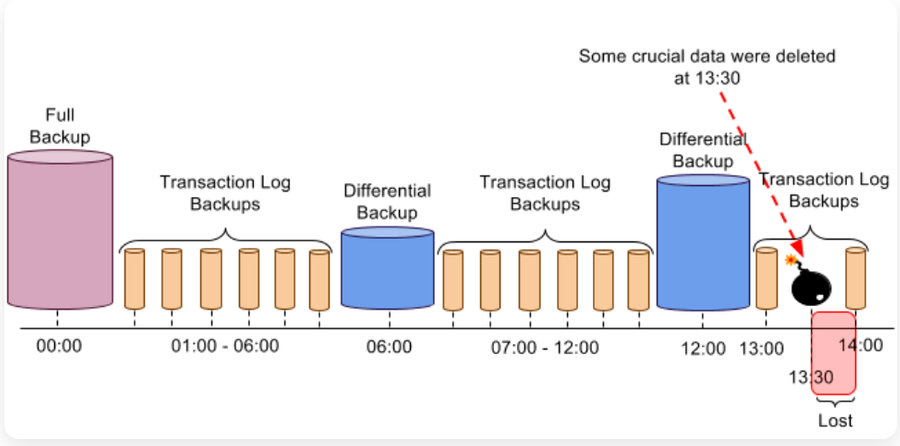
Better performance for bulk operations, reduces log space

Cons

Cannot restore to specific point-in-time during bulk operations

Best for: *Databases with periodic bulk operations*

Point-in-Time Recovery (PITR)



What is PITR?

Allows restoring a database to a specific point in time, within seconds of precision

How it Works

Creates a full backup first, then constantly backs up transaction logs. Recovery involves accessing the full backup and replaying transaction logs to the specified time.

Benefits

- Minimizes data loss
- Recovery to moments before failure
- Supports precise recovery objectives

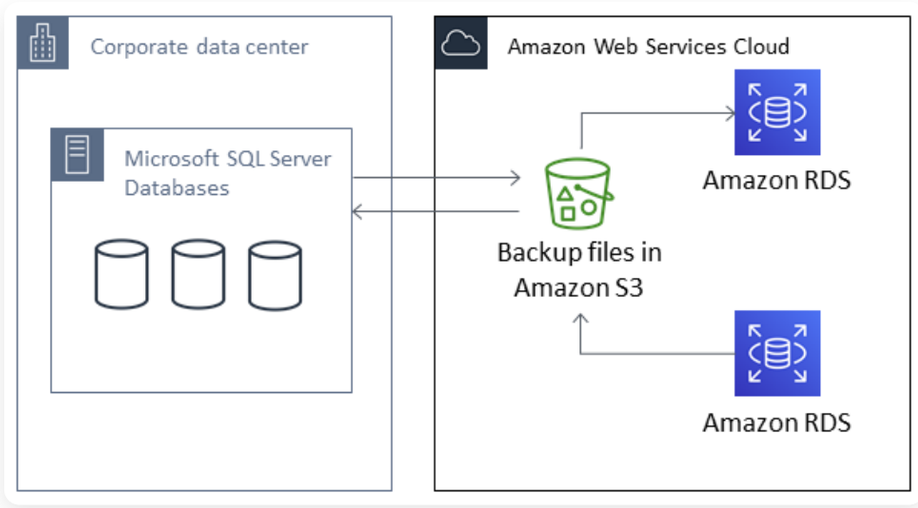
Limitations

- Max retention period (e.g., 35 days)
- Recent activity limitations
- Requires continuous backup config

Use Cases

- Critical databases with minimal data loss tolerance
- Compliance requirements
- Recovering from human errors

Backup Automation and Validation



Importance of Automation

- Reduces human error
- Ensures consistency
- Saves time
- Enables reliable testing

How to Automate

- Use RESTful APIs
- Orchestrate functionality
- CI/CD integration
- PowerShell scripting

Validation Techniques

- Regular restore testing
- Backup integrity checks
- RTO validation
- RPO validation

Best Practices

- Schedule regular automated backups
- Implement monitoring and alerting
- Maintain documentation
- Test recovery procedures regularly

Summary and Best Practices

Key Takeaways

- Choose backup types based on **data volume** and **recovery requirements**
- Select recovery models based on **data criticality** and **acceptable data loss**
- Implement PITR for databases requiring **minimal data loss**
- Automate backup processes to ensure **consistency** and **reliability**

Best Practices

- Follow the **3-2-1 rule**: 3 copies, 2 different media, 1 off-site
- **Regularly test** backup restores to ensure recoverability
- **Monitor** backup jobs and set up alerts for failures
- **Document** backup and recovery procedures
- **Review and update** backup strategy as business needs change

Recommendations

- **Combine different backup types** for optimal protection
- Use **full recovery model** for critical databases
- Implement **automated validation** of backup integrity
- Establish clear **RTO** and **RPO** objectives