

1 What is Accelerated Database Recovery (ADR)?

Accelerated Database Recovery is a **SQL Server engine feature** introduced in **SQL Server 2019** (and Azure SQL Database) that **redesigns the database recovery process** to make it:

- **Instant rollback** of transactions (instead of waiting for long undo phases)
- **Faster crash recovery**
- **Consistent performance** for long-running transactions
- **Improved availability** after unexpected shutdowns

2 Why ADR Was Introduced

In traditional SQL Server recovery:

- Recovery consists of **Analysis, Redo, and Undo** phases.
- Long-running transactions → long undo phase → long rollback times.
- Even cancelling a query with a huge transaction (e.g., large delete) could take minutes/hours.

With ADR:

- Uses a **persisted version store** inside the database.
- Undo work happens instantly by reverting to previous row versions, without scanning the entire transaction log.

3 How ADR Works Internally

Key components:

1. **Persisted Version Store (PVS)**
 - Unlike tempdb version store (used for snapshot isolation), ADR keeps row versions **inside the database**.
 - Old row versions are stored in PVS and cleaned up asynchronously.
2. **Logical Revert**
 - Instantly marks changes from aborted transactions as “invisible” without physically undoing them right away.
3. **Synchronous Transaction Log Truncation**
 - Faster log space reuse, preventing log file bloat from long transactions.
4. **Asynchronous Cleanup**
 - Physically removes obsolete row versions in the background.

4 Enabling Accelerated Database Recovery

You can enable ADR at **database level**:

```
ALTER DATABASE [YourDatabaseName]  
  
SET ACCELERATED_DATABASE_RECOVERY = ON;
```

To disable:

```
ALTER DATABASE [YourDatabaseName]  
  
SET ACCELERATED_DATABASE_RECOVERY = OFF;
```

5 Checking ADR Status

```
SELECT name, is_accelerated_database_recovery_on  
  
FROM sys.databases;
```

6 Example: ADR in Action

Without ADR:

You run a large transaction:

```
BEGIN TRAN;  
  
DELETE FROM Sales WHERE OrderDate < '2010-01-01';
```

1. (This deletes millions of rows.)

Midway, you realize it's a mistake and cancel:

```
ROLLBACK TRAN;
```

2. ❌ Rollback takes minutes or hours depending on size.


With ADR Enabled:

Same large delete:

```
BEGIN TRAN;  
  
DELETE FROM Sales WHERE OrderDate < '2010-01-01';
```

1. Cancel transaction:


```
ROLLBACK TRAN;
```

2.  Rollback is **instant** – SQL Server marks the rows as reverted using PVS and cleans up later.


7 Benefits of ADR

Feature	Without ADR	With ADR
Rollback Time for Large Transactions	Proportional to transaction size	Instant
Recovery After Crash	Can take long for big undo phase	Much faster
Log File Growth During Long Transactions	Can grow large	Reduced
Impact on Availability	Higher downtime	Reduced downtime

8 When to Use

 Ideal for:

- OLTP systems with frequent large transactions.
- Environments requiring **minimal downtime** after a crash or failover.
- Databases with long-running reporting/ETL transactions.

 Considerations:

- **Slightly higher storage usage** due to PVS.
- Version store cleanup might introduce small background I/O load.
- Not supported in **READ-ONLY** databases.

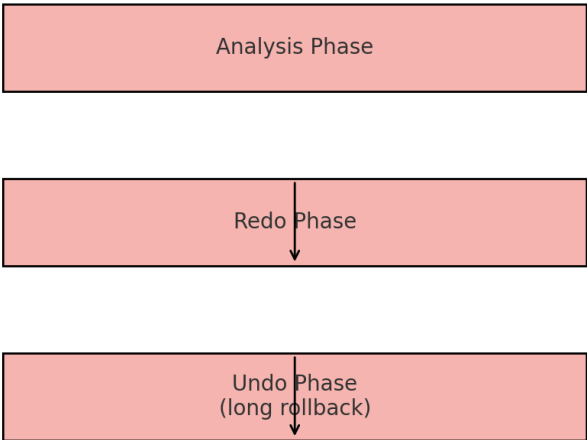
9 Observing ADR Effects

You can monitor ADR-related activity:

```
SELECT *  
FROM sys.dm_tran_persistent_version_store_stats;
```

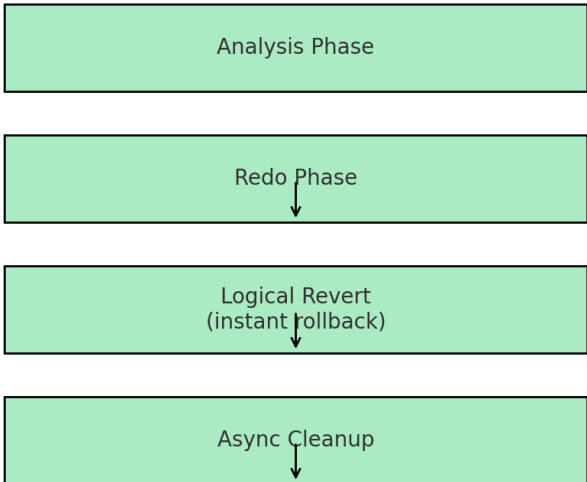
Shows space usage and cleanup activity for PVS.

Traditional Recovery Process



Rollback time depends on transaction size
(can be very long)

Accelerated Database Recovery (ADR)



Rollback is instant; cleanup happens in background