

SQL Server: Shrinking a Database

SQL Server Documentation: Shrinking a Database

What is Shrinking a Database?

Shrinking a database in SQL Server is the process of reducing the physical file size of the data (.mdf) or log (.ldf) files by removing unused space.

This is done using the DBCC SHRINKDATABASE or DBCC SHRINKFILE commands.

Advantages of Shrinking a Database:

- Reclaims disk space: Frees up unused storage space, especially after massive data deletions or archiving.
- Temporary relief on storage: Useful in environments with limited disk space for short-term fixes.
- Reduces backup size: Smaller physical size may reduce full backup size and backup time.

Disadvantages of Shrinking a Database:

- Fragmentation: Causes index fragmentation, which leads to performance degradation.
- Temporary benefit: The space will likely be re-used over time.
- CPU and IO overhead: Shrinking consumes server resources and can impact performance.
- Doesn't always shrink log files: Log files won't shrink unless log backups or truncation are done first.

When to Shrink a Database (Best Practices):

Do It:

- After massive data deletion, archiving, or purging.
- When disk space is critically low.
- During non-production hours.

- As part of a one-time cleanup.

Don't Do It:

- As a regular maintenance task.
- On highly transactional databases.
- Without checking for fragmentation and reindexing.

How to Shrink a Database:

Method 1: Using SQL Server Management Studio (SSMS)

1. Right-click the database > Tasks > Shrink > Database.
2. Review the space info.
3. Click OK.

Method 2: Using T-SQL

-- Shrink entire database

```
DBCC SHRINKDATABASE (YourDatabaseName);
```

-- Shrink data file only

```
DBCC SHRINKFILE (YourDataFileName, TARGET_SIZE_MB);
```

-- Shrink log file

```
DBCC SHRINKFILE (YourLogFileName, TARGET_SIZE_MB);
```

To find your file names:

```
USE YourDatabaseName;
```

GO

```
SELECT file_id, name, type_desc, physical_name FROM sys.database_files;
```

Example Scenario:

"We deleted 10 GB worth of historical parking data from the POS table. After a full backup and index rebuild, we shrunk the data file to reclaim disk space before migrating the database."

Recommendation:

Shrink only when necessary, rebuild indexes after, and monitor performance. Document every shrink operation in your logs.

Documentation by: Jose Antonio "Tony" Acebuche