Common Database Maintenance Tasks

Maintaining a SQL Server database is critical to ensuring that your database environment performs reliably and efficiently. Although advances in the SQL Server platform have made maintenance activities simpler and more straightforward, it is critical that DBAs understand the following:

- Which database maintenance tasks need to be performed and why
- When to perform them
- How to perform them

In this chapter, we cover a series of important database maintenance tasks, how to determine whether and when they need to be performed, and ways to perform those tasks. Subsequent chapters build upon this one by showing you how to use different tools, such as PowerShell, and by showing you how to define maintenance plans to help automate your work.

Backing Up and Restoring

If you talked to 100 DBAs, it’s a pretty safe bet that all of them would list backing up and restoring databases as two of the most important maintenance tasks they perform. Backup and restore are so important that we’ve already covered each topic separately in its own chapter. Chapter 8 shows you how to take backups, and Chapter 9 shows you how to restore your database from those backups.

At the risk of repeating ourselves, we want to stress that, as a DBA, you must devise a backup and restore strategy that will guarantee the integrity of the data in the databases that you manage and that will minimize the amount of downtime that you experience in the case of a catastrophic failure. Such a strategy is referred to as a disaster recovery (DR) plan. You simply must have such a plan. If you don’t have one, make creating it a priority.

---

**Note** For help in creating a disaster recovery plan, we recommend James Leuttkelhoelter’s excellent book *Pro SQL Server Disaster Recovery*, which is also published by Apress. James takes a holistic approach to disaster recovery that encompasses not only the technical side of things but also the human and business sides.
Checking the Database Integrity

Databases should be checked at regular intervals for consistency. We recommend checking the database consistency prior to a database backup. Later, in Chapter 12, when we discuss maintenance plans, you will see how you can set the order of tasks, which allows you to perform the database consistency check first. Consistency issues are typically the result of some form of hardware failure, but they can also occur when the database server is powered off unexpectedly.

You can run four Transact-SQL commands to check various levels of consistency of your database, as described in Table 10-1.

Table 10-1. Commands for Checking Database Integrity

<table>
<thead>
<tr>
<th>T-SQL Command</th>
<th>What It Does</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBCC CHECKALLOC</td>
<td>Checks the consistency of disk space allocation structures for a database</td>
</tr>
<tr>
<td>DBCC CHECKCATALOG</td>
<td>Checks the consistency between the system metadata tables for a specified database</td>
</tr>
<tr>
<td>DBCC CHECKTABLE</td>
<td>Checks all pages and structures in a table or indexed view for consistency</td>
</tr>
<tr>
<td>DBCC CHECKDB</td>
<td>Runs DBCC CHECKALLOC, DBCC CHECKCATALOG, and DBCC CHECKTABLE and validates the following: Indexed view contents</td>
</tr>
<tr>
<td></td>
<td>Any Service Broker data stored in the database</td>
</tr>
<tr>
<td></td>
<td>Consistency between table metadata and file system directories when using FILESTREAM</td>
</tr>
</tbody>
</table>

You should always run the DBCC CHECKDB command to check database integrity prior to backing up, except in cases where the entire database check is too time-consuming. In the event the check is too time-consuming, you can take one of the following approaches:

- Run the three independent DBCC commands that are part of CHECKDB separately to accommodate your schedule.
- Use the PHYSICAL ONLY option for CHECKDB. The PHYSICAL ONLY option can greatly reduce run time, since it checks only the physical structures in the database for consistency.

In either case, you should run CHECKDB with no options periodically for your databases to provide the most comprehensive check of your database consistency. Figure 10-1 provides an example.