

SQL Datenbanken reparieren mit DBCC CHECKDB

In diesem kurzen Beitrag wird das **Database Console Command (DBCC) CheckDB** Kommando für SQL Server umfassend erklärt und dessen Funktion an einem kleinen praktischen Beispiel verdeutlicht.

Zunächst stellt sich die Frage: um was handelt es sich bei diesem Kommando genau? Das DBCC CHECKDB Kommando dient dazu, Dinge wie die logische Integrität von Datenbankenobjekten, Indexbeziehungen und viele andere Strukturprüfungen durchzuführen und zu evaluieren. Schlägt eine dieser Überprüfungen fehl, wird dies auf der Datenbankkonsole ausgegeben.

Der beste Weg, um nun einen von DBCC CHECKDB gemeldeten Fehler zu beheben, besteht darin, die letzte bekannte gute Sicherung auszuführen. Ist diese jedoch nicht vorhanden oder gar beschädigt, kann auf die Datenbank im sogenannten **Notfallstatus (Emergency State)** zugegriffen werden. Dieser Status ermöglicht den Zugriff auf eine Datenbank, die sich im **Verdächtigen Modus (Suspect Mode)** befindet. Weiterhin können hier DBCC CHECKDB – Reparaturoptionen ausgeführt werden, um so die beschädigte Datenbank zu reparieren.

Die richtige Verwendung von DBCC CHECKDB

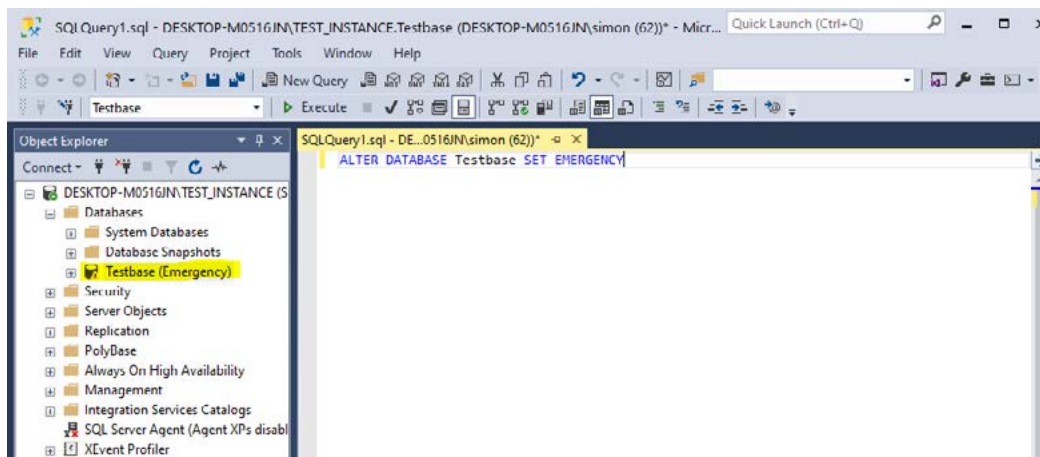
Anhand dieses Beispiels möchten wir erklären, wie wir durch die richtige Verwendung von DBCC CHECKDB eine Datenbank reparieren können.

In diesem Beispiel nutzen wir eine SQL Server 2019 Instanz mit dem Namen TEST_INSTANCE und die Datenbank Testbase.

Als Erstes verbinden wir uns mit unserer Beispiel-Datenbank via SQL Server Management Studio (SSMS), das sie [hier](#) kostenlos downloaden können.

Anschließend wird die Datenbank in den emergency mode gesetzt. Hierfür führen wir das folgende Kommando aus:

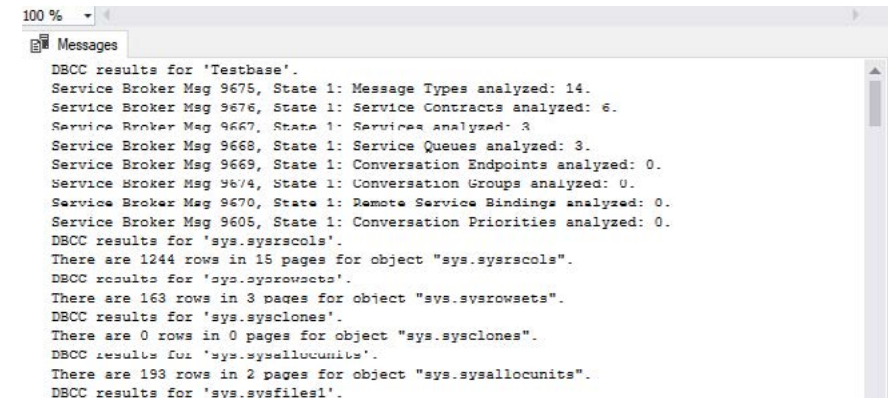
```
ALTER DATABASE Testbase SET EMERGENCY
```



Nun kann das DBCC CHECKDB Kommando ausgeführt werden, um bestehende Korruptionsfehler genau zu analysieren. Hierfür führen Sie bitte das folgende Kommando aus:

```
DBCC CHECKDB (Testbase)
```

Anschließend erscheint diese Analyse auf der Datenbankenkonsole:



Werden hier durch DBCC CHECKDB etwaige Fehler innerhalb der Datenbank ermittelt, werden für diese direkte Reparaturmöglichkeiten angeboten.

Bevor nun aber Reparaturmaßnahmen an der Datenbank durchgeführt werden, sollte diese in den **single user mode** versetzt werden. So kann verhindert werden, dass andere Benutzer während des Reparaturprozesses Veränderungen an der Datenbank vornehmen können.

Dafür dieses Kommando ausführen:

```
ALTER DATABASE Testbase SET SINGLE_USER
```

Nun kann mit der eigentlichen Reparatur der Datenbank begonnen werden. Hierfür führen Sie das folgende Kommando aus:

```
DBCC CHECKDB (N'Testbase', REPAIR_ALLOW_DATA_LOSS) WITH ALL_ERRORMSG, NO_INFOMSGS  
GO
```

Damit führen wir den Reparatur-Modus REPAIR_ALLOW_DATA_LOSS aus. Dieser ist, auch wenn er bei der Reparatur aller gemeldeten Fehler in der SQL Server Datenbank hilft, nicht immer die beste Option. Denn, wie der Name schon verrät, verursacht dieser Modus einen Datenverlust. Er sollte daher als eine Art "letzter Ausweg" gesehen werden, falls eine Datenbank aus ihrer Sicherung nicht mehr wiederhergestellt werden kann.

Wenn also keine Sicherung der Datenbank verfügbar ist und kein Datenverlust riskiert werden kann, sollte stattdessen eine spezielle MS SQL-Reparatursoftware verwendet werden.

Hierbei ist wichtig zu erwähnen, dass DBCC neben REPAIR_ALLOW_DATA_LOSS auch noch zwei andere Reparatur Modi bereitstellt:

- × **REPAIR_FAST**: Diese Option hilft dabei, die Syntax für die Abwärtskompatibilität beizubehalten. Andere Reparaturoptionen werden nicht durchgeführt.
- × **REPAIR_REBUILD**: Bei dieser Option kann eine Datenbank ohne Datenverlust repariert werden. Es kann verwendet werden, um fehlende Zeilen in nicht-gruppierten Indizes zu reparieren und einen Index neu zu erstellen.

Nachdem nun die Datenbank erfolgreich repariert wurde, wird diese zurück in den multi user mode versetzt. Hierfür, ähnlich wie beim Versetzen in den **single user mode**, führen Sie einfach das folgende Kommando aus:

```
ALTER DATABASE Testbase SET MULTI_USER
```

Fazit

DBCC CHECKDB ist sehr hilfreich, um Konsistenzfehler und Beschädigungen innerhalb einer Datenbank zu überprüfen und zu erkennen. Beschädigte Datenbanken sind einer der Hauptverursacher von Ausfallzeiten und Datenverlust, weswegen eine regelmäßige Überprüfung wichtig ist.

Der richtige Weg, mit solchen Fällen umzugehen, ist die Wiederherstellung einer früheren Sicherung der Datenbank. Wenn keine Sicherung vorhanden, oder diese selbst beschädigt ist, kann die Datenbank mit der Reparaturoption **REPAIR_ALLOW_DATA_LOSS** von DBCC CHECKDB repariert werden. Jedoch verstärkt sich mit dieser Option auch die Gefahr eines Datenverlustes.