

# Gekonnt reparieren

Auch eine zuverlässige Software wie MySQL kommt nicht ohne Wartung über die Runden. Fragmentierte Tabellen, seltsame Fehlermeldungen oder korrupte Datenbanken treten zwar selten auf, kommt es aber dennoch zu solchen Problemen, hilft sich der Admin mit Bordmitteln einfach selbst. Thomas Wölfer



**MySQL gilt** als Webdatenbank schlechthin. Auf Linux ist sie das am häufigsten eingesetzte Management-Werkzeug – wohl auch, weil kaum Probleme im Betrieb zu erwarten sind. Denn in der Regel summt die Datenbank fröhlich vor sich hin und liefert derweil klaglos alle Daten, die der Benutzer benötigt.

Aber auch MySQL [1] kommt nicht ohne Wartung aus. Kleine Probleme wie fragmentierte Tabellen sind relativ leicht auch für Anfänger in den Griff zu bekommen. Defekte Datenbanken erfordern schon mehr Know-how, denn die Rettungsaktion besteht aus mehreren Schritten. Wie der Administrator Probleme mit einem MySQL-Server löst, erläutert dieser Beitrag.

Voraussetzung für einen vernünftigen Serverbetrieb ist ein durchdachtes Setup. Dabei sind verschiedene Dinge zu beachten. Am wichtigsten jedoch ist, dass die Tabellen möglichst effizient sind. Effizient ist eine Tabelle, bei der die Datentypen aller Spalten nur genau

so groß sind, wie sie tatsächlich sein sollen. So gibt es beispielsweise eine ganze Reihe unterschiedlicher Integer-Datentypen bei MySQL: Sie unterscheiden sich nicht nur in der größten Zahl, die sie aufnehmen können, ihr Speicherbedarf differiert ebenfalls. Je größer eine Zahl werden kann, desto mehr Speicher benötigt MySQL auch für die Daten, die abgelegt werden.

## Datenbanks größter Feind

Die Tabellen mögen noch so effizient angelegt sein, im Laufe der Zeit gerät die Ordnung trotzdem durcheinander. Der Grund: Fragmentierung. Eine Tabelle fragmentiert, wenn Anwender einzelne Teile löschen. Das führt dazu, dass irgendwo in der Datendatei von MySQL ein Loch entsteht. Normalerweise ist dies kein Problem, denn die gelöschten Zeilen protokolliert MySQL in einer verketteten Liste. Kommen wieder Zeilen zur Tabelle hinzu, dann recycelt die Da-

tenbank die freien Positionen und das Datenloch ist wieder gefüllt. Das ist alles gut, schön und effizient.

Nur funktioniert es nicht, wenn die Tabelle Felder mit Datentypen enthält, deren Länge variabel ist, etwa »varchar«, »blob« oder »text«. Es ist also recht wahrscheinlich, dass die einzufügenden Daten mehr Platz benötigen als vorher durch Löschen frei geworden ist. Die neuen Daten passen nicht und MySQL sucht einen anderen Platz. Die Tabelle bleibt somit löchrig wie Schweizer Käse und verschwendet Platz.

Auch die zweite Art der Tabellenfragmentierung entsteht durch Daten mit variabler Länge. Nicht nur das Löschen von Spalten mit solchen Daten führt zur Fragmentierung. Es reicht schon, dass der Anwender eine Zeile ändert. Im Prinzip ist dies das Gleiche, wie eine vorhandene Zeile zu löschen und eine neue mit längeren Daten einzufügen. Die Daten passen nicht mehr gänzlich in das vorherige Feld und liegen fortan zerstückelt in der Datenbankdatei.

## Der Fragmentierung an den Kragen gehen

Gegen fragmentierte Tabellen kann der Admin selbst einiges tun. Es gibt einen MySQL-Befehl und auch ein externes Hilfsprogramm, um die Tabellen zu defragmentieren. Der SQL-Befehl trägt den Namen »optimize table« (**Abbildung 1**). Das Kommando erwartet mindestens den Namen einer Tabelle als Parameter. Möchte der Verwalter mehrere Tabellen bereinigen, trennt er deren Namen einfach per Komma voneinander. Während »optimize table« läuft, ist die entsprechende Tabelle gesperrt. Der

Durchlauf entfernt nicht nur die Fragmentierung, sondern sortiert zusätzlich die Indizes. Das Optimize-Table-Kommando sollte der Admin regelmäßig aufrufen, da es zur Routinewartung gehört. Um die Optimierung der Tabelle zu automatisieren, eignet sich ein Skript, das der Cron-Daemon zum Beispiel in Monatsabständen aufruft.

## Schwerer Fehler: Defekte Tabellen

Im Normalbetrieb sind fragmentierte Tabellen eigentlich das Schlimmste, was passieren kann. Falls beispielsweise ein Stromausfall MySQL davon abhält, einen Record zu schreiben, kann dies zum GAU führen. Defekte Tabellen sind nach einem unterbrochenen Schreibvorgang durchaus die Regel. Ob eine Tabelle korrumpiert ist, prüft der Admin mit dem Befehl »check table« (**Abbildung 2**). Das Kommando erwartet den Namen einer Tabelle oder eine Reihe von Tabellennamen, die durch Kommata getrennt sein müssen. Außerdem nimmt es einen der Parameter »quick«, »fast«, »changed«, »medium« oder »extended« entgegen. Ohne Parameter verwendet der Befehl den Modus »medium«.

Die »quick«- und »fast«-Optionen führen im Wesentlichen nur einen Minimaltest durch. Der Modus »changed« prüft nur jene Tabellen, die sich nach dem letzten Durchlauf geändert haben. Der »medium«-Test ist dagegen schon recht ausführlich. Aber nur mit der »extended«-Option stellt der Admin wirklich sicher, dass die Tabelle vollständig konsistent ist. Der Befehl ist interessant vor einem Backup oder falls seltsame Fehler auftreten, MySQL aber keine passende Meldung ausgibt.

In der Regel ist MySQL allerdings sehr gesprächig, falls es einen Defekt entdeckt. Es liefert entsprechende Fehlermeldungen:

- *TabelleName.Frm* is locked against change
- Can't find *TabelleName.myi*
- Unexpected end of file
- Record file is crashed
- Got error *NummerDesFehlers* from table handler

Die letzte Meldung gibt eine Fehlernummer an. Mit ihr und dem Peek-Error-Kommando (»perror«) kommt der Admin an weitere Informationen zum Fehler. Am häufigsten treten folgenden Meldungen auf:

- Index file is crashed
- Record file is crashed
- Old database file
- Record was already deleted
- No more room in record file

Die meisten Fehler kann der Administrator noch leicht reparieren. Doch gibt es zwei Ausnahmen: die Fehler 135 (»No more room in record file«) und 136 (»No more room in index file«). Bei diesen Meldungen verwendet der Administrator das Alter-Table-Kommando, damit die Tabelle wieder wachsen kann:

```
ALTER TABLE TabelleName
MAX_ROWS=NeuerWert
AVG_ROW_LENGTH=NeuerWert;
```

## Tabelle korrumpiert und kein Backup zur Hand

Ist eine Tabelle korrumpiert, spielt der Admin einfach das Backup ein. Falls auch die Datensicherung defekt ist, gibt es immer noch eine große Chance, wieder an die Daten zu gelangen: Dann tritt das Kommando »repair table« auf den Plan. Dessen Durchlauf produziert recht viel Text auf der Konsole. Als letzte Anzeige erscheint der Zustand der Tabelle, hier sollte »OK« stehen.

Der Repair-Table-Befehl ist nicht ganz ungefährlich. Während er läuft, sollte der Verwalter ein wachsames Auge auf den Mysql-Daemon haben. Stirbt dieser

noch während »repair table« sich die Tabelle vornimmt, dann ist nach dem nächsten Start des »mysqld« Folgendes zu tun: Vor allem anderen muss das Repair-Table-

```
mysql> optimize table users;
+-----+-----+-----+-----+
| Table | Op      | Msg_type | Msg_text |
+-----+-----+-----+-----+
| test.users | optimize | status   | OK       |
+-----+-----+-----+-----+
1 row in set (38.12 sec)
mysql>
```

**Abbildung 1:** Der Optimize-Table-Befehl benötigt – abhängig von der Größe der Tabelle – oft mehrere Minuten für den Durchlauf.

Kommando nochmals durchlaufen. Vergisst der Admin dies, überschreibt der MySQL-Daemon eventuell die Indexdatei beim nächsten Kommando.

Falls der Reparaturbefehl kein »OK« zurückliefert, konnte er den Schaden nicht beheben. In solchen Fällen sollte der Admin zu »myisamchk« [2] greifen. Dieses Programm repariert ebenfalls defekte Tabellen und bringt dafür auch mehr Optionen mit:

```
myisamchk --safe-recover
```

## Falls die Reparatur versagt

Wenn das schnelle und elegante Repair-Kommando nicht weiterhilft, ist trotzdem noch längst nicht alles verloren. Mit den folgenden Schritten sollte es gelingen, die Daten wiederherzustellen. Zu beachten ist: Während der Verwalter eine MySQL-Datenbank mit den Kommandozeilentools repariert, darf der MySQL-Daemon nicht aktiv sein. Wie der Daemon zum Stillstand kommt, ist im Prinzip egal. Ein »mysqldadmin shutdown« reicht völlig aus.

Doch Achtung: Der »mysqld« läuft noch so lange weiter, bis der Prozess endgültig endet. Wie lange das dauert, ist in erster Linie von der Belastung des Servers abhängig. Das Programm »top« zeigt laufend alle aktiven Prozesse an, sodass niemand raten muss, wann sich »mysqld« beendet hat.

Sobald der MySQL-Daemon gestoppt ist, geht es daran, die Rechte für die Datendateien zu prüfen. Damit die Reparatur erfolgreich verläuft, sollte mindestens das Leserecht gesetzt sein. Die Schreibrechte sind nur in den Fällen erforderlich, wenn die Dateien zur Reparatur beschrieben werden müssen. Zu beachten ist außerdem, dass jede Tabelle aus zwei wichtigen Dateien besteht: der Datendatei mit der Endung »MYD« und der Indexdatei »\*.MYI«.

Vor der Reparatur ist ein Test fällig. Dafür kommt das »myisamchk«-Kommando zum Einsatz:

```
myisamchk *.MYI
```

Als Reaktion können drei Dinge passieren: Myisamchk findet für keine Tabelle Fehler oder zeigt für einige Tabellen Fehler an. Außerdem kann das Programm noch abstürzen und dabei sehr merkwürdige Meldungen ausgeben. In diese Gattung fällt zum Beispiel der Hinweis »out of memory«.

Meldet Myisamchk Fehler in einer Tabelle, hat der Verwalter eine relativ einfache Reparaturprozedur vor sich (siehe nächster Abschnitt). Stürzt das Programm ab, ist auch noch nichts verloren. Selbst in diesem Fall ist eine vollständig konsistente Datenbank möglich. Wie man dabei vorgeht, behandelt der Abschnitt „Die aufwändige Reparatur“ weiter unten.

## Die einfache Reparatur

Für eine einfache Reparatur sind zwei Dinge auszuprobieren. Bringen beide nicht den gewünschten Erfolg, helfen nur noch die Tipps der aufwändigen Reparatur weiter unten. Der erste Versuch ist ein Aufruf von »myisamchk«:

```
myisamchk -r -q TabellenName
```

Falls sich der Fehler im Indexfile eingestrichelt hatte, repariert dies die Tabelle bereits. Die Option »-r« steht für Recover und ist die Standardoption für den Reparaturfall. »-q« verlangt eine schnelle (quick) Reparatur, bei der die Datendatei nicht geändert wird.

Führt das Kommando nicht zum gewünschten Erfolg, ist zuerst ein Backup der Datendatei anzulegen. Danach sorgt ein erneuter Aufruf von Myisamchk ohne die Option »-q« dafür, dass defekte und gelöschte Records aus der Tabelle entfernt werden. Zudem baut das Kom-

mando eine neue Indexdatei. Allerdings kann auch dies fehlschlagen. Bevor der aufwändige Reparaturversuch ansteht, könnte noch folgender Aufruf helfen: »myisamchk --safe-recover«. Dabei handelt es sich um eine etwas ältere Methode, sie beherrscht einige Tricks, die der Schalter »-r« nicht kann.

## Die aufwändige Reparatur

Ist die Indexdatei völlig zerstört oder fehlt sie sogar ganz, muss der Verwalter einen mühsameren Weg bei der Reparatur beschreiten. Zunächst verschiebt er die Datendatei an eine sichere Stelle. Ist das geschehen, erzeugt er eine leere Tabelle und ein neues Indexfile. Dazu eignet sich die Tabellenbeschreibung aus der Datenbank:

```
mysql> SET AUTOCOMMIT=1
mysql> TRUNCATE TABLE TabellenName
mysql> quit
```

Danach ist das soeben erzeugte neue Datenfile mit der alten Datendatei zu überschreiben. Das Ergebnis besteht aus einem richtigen (aber leeren) Index und allen alten Daten. Nun geht es weiter, wie im Abschnitt „Die einfache Reparatur“ beschrieben:

```
myisamchk -r -q TabellenName
```

Jetzt bleibt nur noch eine einzige Datei übrig, die beschädigt werden könnte: die Tabellendefinition. Allerdings ändert sich diese Datei nicht, nachdem die Tabelle einmal angelegt ist. Ein Hardwarefehler kann aber trotzdem dazu führen, dass die Datei unbrauchbar wird. In diesem Fall gibt es nicht mehr viel, was der Administrator tun kann: Entweder steht ein Backup der Tabellendefinition zur Verfügung oder er weiß noch auswendig, wie die Tabelle genau ausgesehen hat, und bastelt sie neu. (jre) ■

### Infos

[1] MySQL: [<http://www.mysql.com>]

[2] Myisamchk: [[http://www.mysql.de/doc/de/myisamchk\\_repair\\_options.html](http://www.mysql.de/doc/de/myisamchk_repair_options.html)]

### Der Autor

Thomas Wölfer ist seit 20 Jahren Software-Entwickler. Zurzeit verwaltet er die Mail-, Web- und Datenbankserver des Portals Nickles.de.

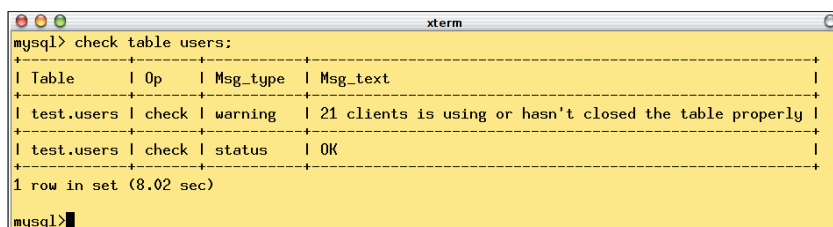


Abbildung 2: Status »OK« – so sollte die Meldung nach erfolgreichem »check table«-Durchlauf aussehen.