Inhalt

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 5 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "ucsencs.def" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Kapitel 1
Datenbank

In diesem Kapitel finden Sie Basisstrategien für die Aufgaben, denen Sie bei Datenbanken regelmäßig gegenüberstehen.

An Datenbanksystemen mangelt es in der Linux-Welt nicht. Die Entscheidung, in diesem Kapitel ausschließlich *MariaDB* zu behandeln, fiel aus zwei Gründen. Zum einen ist MariaDB mittlerweile das Linux-Datenbanksystem mit der weitesten Verbreitung in neuen Distributionen.

Zum anderen ist es als Fork von MySQL spätestens seit der Version 5.0 endgültig den Kinderschuhen entwachsen und enthält Funktionen, die früher nur teuren kommerziellen Systemen vorbehalten waren.

1.1 MariaDB in der Praxis

Alle populären Linux-Distributionen enthalten MariaDB-Pakete, bei vielen wurde MySQL durch MariaDB ersetzt. In CentOS findet sich die Version 10.3.127, in Debian die Version 10.3.27, in Ubuntu die Version 10.3.25 und in openSUSE Leap 10.4.17.

Bitte beachten Sie, dass wir – wenn nicht explizit auf eine andere Konfigurationsdatei verwiesen wird – alle Konfigurationen in der Datei *adminbuch.cnf* in dem Verzeichnis */etc/mysql/mariadb.conf.d* für Debian und Ubuntu bzw. im Verzeichnis */etc/my.cnf.d* unter CentOS und openSUSE Leap vornehmen.

1.1.1 Installation und grundlegende Einrichtung

Unter Debian und Ubuntu installieren Sie MariaDB mit dem folgenden Kommando:

```
apt install mariadb-server
```

Listing 1.1 Installation des Pakets »mariadb-server« unter Debian und Ubuntu

Unter CentOS wird das Paket folgendermaßen installiert:

```
dnf install mariadb-server
```

Listing 1.2 Installation des Pakets »mariadb-server« unter CentOS

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 7 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Unter openSUSE heißt das gewünschte Paket nicht `mariadb-server`, sondern schlicht `mariadb`. So installieren Sie es:

```
zypper install mariadb
```

Listing 1.3 Installation des Pakets »mariadb« unter openSUSE

Abhängig von der verwendeten Linux-Distribution werden Sie möglicherweise bereits während der Installation nach einem Passwort für den MariaDB-Benutzer `root` gefragt. Er ist nicht identisch mit dem Superuser, also dem `root`-Account Ihres Linux-Systems, sondern repräsentiert lediglich den Administrator des MariaDB-Servers.

In allen von uns unterstützten Distributionen starten Sie den Datenbankserver mit `systemctl start mariadb`. Allerdings werden bei allen hier behandelten Distributionen die MySQL-Clients, wie beispielsweise `mysql` und `mysqladmin`, verwendet. Sollte die Installationsroutine Sie nicht nach einem Passwort fragen, vergeben Sie es unmittelbar nach der Installation und dem Start des Servers mit dem folgenden Kommando:

```
/usr/bin/mysqladmin -u root password 'NeuesPasswort'
```

Listing 1.4 Passwort für den »root«-Account des MariaDB-Servers vergeben

Zusätzlich ist es sinnvoll, ein weiteres Administrationskonto einzurichten. Dazu loggen Sie sich zunächst als `root` auf dem MariaDB-Server ein:

```
mysql -u root -p
```

Listing 1.5 Login auf dem MariaDB-Server

Nach der erfolgreichen Eingabe des Passworts ändert sich Ihr Kommandozeilenprompt in `MariaDB [(none)]>`. Jetzt können Sie das neue Benutzerkonto anlegen:

```
GRANT ALL ON *.* TO 'ich'@'localhost' IDENTIFIED BY 'MeinKennwort' WITH GRANT OPTION;
```

Listing 1.6 Ein neues Benutzerkonto mit weitreichenden Rechten anlegen

Achten Sie besonders auf das abschließende Semikolon; es ist unbedingt erforderlich, wird aber häufig vergessen. Mit `exit` verlassen Sie den MariaDB-Kommandoprompt wieder.

[+] Per Default bindet sich MariaDB unter Debian, openSUSE und Ubuntu unmittelbar nach der Installation nur an die IP-Adresse 127.0.0.1 (`localhost`). Wenn Sie diese Einstellung ändern möchten, erstellen Sie bitte unter Debian, openSUSE und Ubuntu eine Datei namens */etc/mysql/mariadb.conf.d/adminbuch.cnf*, in die Sie unterhalb des Eintrags `[mysqld]` einen Eintrag `bind-address = 0.0.0.0` aufnehmen.

[!] Mit der Einstellung `bind-address = 0.0.0.0` bindet sich MariaDB an alle auf dem Server konfigurierten IP-Adressen. Prüfen Sie vor einem solchen Schritt genau, ob diese Konfiguration notwendig und – gerade unter Sicherheitsgesichtspunkten – sinnvoll ist.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 8 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Nach einem Neustart des Datenbankservers sollte Ihnen `lsof` Informationen analog zur folgenden Ausgabe liefern:

```
# lsof -i tcp:3306
COMMAND   PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
mysqld  11735 mysql   14u  IPv4  28487      0t0  TCP *:mysql (LISTEN)
```

Listing 1.7 Feststellen, auf welche IP-Adressen der Datenbankserver hört

Wie Sie sehen können, heißt das Kommando trotz Abspaltung vom MySQL-Projekt immer noch `mysqld` und nicht wie erwartet `mariadb`.

1.1.2 Replikation

Durch Replikationsmechanismen können Sie Datenbanken, die sich auf unterschiedlichen MariaDB-Servern befinden, auf dem gleichen Stand halten.

Das bietet eine Reihe von Vorteilen. So können Sie beispielsweise die Last gezielt auf mehrere Server verteilen:

- Eine Datenbank erhält nur Aktualisierungen, eine andere nur Abfragen.
- Eine Datenbank erhält nur Aktualisierungen, mehrere andere erhalten Abfragen.
- Mehrere Datenbanken erhalten sowohl Aktualisierungen als auch Abfragen.

Mittels Replikation ist es auch möglich, eine Datenbank auf den Notebooks von Außendienstmitarbeitern automatisch mit einer Datenbank in der Firmenzentrale abzugleichen, damit die Mitarbeiter stets über den neuesten Datenbestand verfügen.

Wir werden folgende Replikationsmethoden betrachten:

- **Master-Slave-Replikation**
  Ein Master-Server kann einen oder mehrere Slaves haben. Der Master speichert Updates auch in *Binlogs* (binären Log-Dateien). Wenn ein Slave eine Verbindung zum Master herstellt, teilt er ihm mit, bis zu welcher Position er die Binlogs beim letzten Update gelesen hat. Der Master repliziert daraufhin alle Änderungen, die seit diesem Zeitpunkt aufgelaufen sind, auf den Slave. Danach wartet der Slave, bis er vom Master über weitere Änderungen informiert wird. Während dieser Zeit kann er Anfragen beantworten, Aktualisierungen werden aber immer nur auf dem Master vorgenommen.
- **Master-Master-Replikation (Ring-Replikation)**
  Hier dürfen Schreibvorgänge auf allen beteiligten Servern durchgeführt werden, aber das birgt auch die Gefahr, dass `auto_increment`-Felder auf allen Mastern gleichzeitig inkrementiert werden und so doppelte Werte auftreten. Daher müssen für diesen Fall Vorkehrungen getroffen werden.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 9 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

[!] Die erwähnten binären Logfiles zeichnen von dem Moment an, an dem das Logging gestartet wird, alle Änderungen an den Datenbanken des Master-Servers auf. Wenn Sie einen Replikationsverbund aufbauen, ist es wichtig, dass zum Start der Replikation alle Server den Stand der Master-Datenbanken zu dem Zeitpunkt haben, an dem die binären Logfiles aktiviert wurden. Weichen die Stände voneinander ab, wird es sehr wahrscheinlich zu Fehlern bei der Replikation kommen. Achten Sie auch auf mögliche Versionsunterschiede bei den MariaDB-Servern, die Sie einsetzen. Neuere Slaves können in der Regel mit älteren Mastern zusammenarbeiten, aber nicht umgekehrt.

Konfiguration eines Master-Servers

Legen Sie zunächst einen Benutzer für die Replikation an. Der Benutzer für das folgende Beispiel heißt `repl` und bekommt das Passwort `slavepass`, und der User darf sich aus dem angegebenen Netzbereich anmelden:

```
MariaDB [(none)]> GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.*
    -> TO repl@'192.168.1.%' IDENTIFIED BY 'slavepass';

### oder ###

MariaDB [(none)]> GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.*
    -> TO 'repl'@'%.mydomain.com' IDENTIFIED BY 'slavepass';
```

Listing 1.8 Einen Benutzer für die Replikation anlegen

Vergewissern Sie sich nun, dass der Abschnitt `[mysqld]` der Datei *adminbuch.cnf* (siehe auch den Hinweis zu Beginn des Abschnitts 1.1) auf dem Master-Server die Option `log-bin` enthält. Der Abschnitt sollte außerdem eine Option `server-id=master_id` enthalten, wobei `master_id` ein positiver Integer zwischen `1` und `4294967295` (2^{32} - 1) sein muss. Da der Default-Wert `1` ist, kann er auch auf einem schlecht konfigurierten Server noch »versehentlich« gesetzt sein, deshalb sollten Sie einen anderen Wert wählen.

Damit Sie die größtmögliche Dauerhaftigkeit und Konsistenz in einer Replikationskonfiguration für InnoDB[1] mit Transaktionen zu erzielen, sollten Sie in jedem Fall `innodb_flush_log_at_trx_commit=1` und `sync_binlog=1` in der Datei *adminbuch.cnf* auf dem Master angeben. So sieht ein Beispiel für den fertigen Konfigurationsabschnitt aus:

```
[mysqld]
log_bin                 = /var/log/mysql/mariadb-bin.log
max_binlog_size         = 100M
server_id = 100
```

1 InnoDB ist eine freie Storage Engine für MySQL/MariaDB, die sich insbesondere durch Transaktionssicherheit auszeichnet.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 10 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
innodb_flush_log_at_trx_commit=1
sync_binlog=1
```

Listing 1.9 Die Änderungen in der Datei »adminbuch.cnf«

Nachdem Sie diese Änderungen vorgenommen haben, müssen Sie den MariaDB-Daemon neu starten. Falls Sie CentOS benutzen, legen Sie vorher noch mit den folgenden Kommandos das Verzeichnis */var/log/mysql* für den User *mysql* an:

```
mkdir /var/log/mysql
chown mysql:mysql /var/log/mysql
```

Listing 1.10 Das Log-Verzeichnis anlegen und dem User »mysql« zuordnen

Nun muss der Slave erstmalig mit einer Momentaufnahme der Daten des Masters versorgt werden. Synchronisieren Sie alle Tabellen, und unterbinden Sie Schreibanweisungen, indem Sie eine `FLUSH TABLES WITH READ LOCK`-Anweisung auf dem Master ausführen:

```
MariaDB [(none)]> FLUSH TABLES WITH READ LOCK;
```

Listing 1.11 Tabellensynchronisierung und Stopp aller Schreibvorgänge

Wechseln Sie nun in das Datenverzeichnis des Master-Servers (*/var/lib/mysql/*). Dort gibt es für jede Datenbank ein Unterverzeichnis. Kopieren Sie die Dateien aller Datenbanken, die an der Replikation teilnehmen sollen, in das identische Verzeichnis auf dem Slave-Server. Lassen Sie dabei jedoch die Dateien *master.info* und *relay-log.info* aus.

Wenn auf dem Slave-Server andere MariaDB-Benutzerkonten als auf dem Master vorhanden sind, sollten Sie die Datenbank *mysql* besser nicht replizieren.

Während die mit `FLUSH TABLES WITH READ LOCK` erwirkte Lesesperre gültig ist, ermitteln Sie den Namen des aktuellen Binär-Logs und den Versatz auf dem Master:

```
MariaDB [(none)]> SHOW MASTER STATUS;
+-----------------+----------+--------------+------------------+
| File            | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----------------+----------+--------------+------------------+
| mariadb-bin.003 | 73       | test         | manual,mysql     |
+-----------------+----------+--------------+------------------+
mysql >
```

Listing 1.12 Binlog-Name und Versatz auf dem Master ermitteln

Die Spalte `File` zeigt den Namen der Log-Datei an, und `Position` zeigt den Versatz innerhalb der Datei. In diesem Beispiel heißt die Binär-Log-Datei *mariadb-bin.003*, der Versatz ist 73. Notieren Sie diese Werte, denn Sie benötigen sie später beim Konfigurieren des Slaves. Die Werte stellen die Replikationskoordinaten dar, bei denen der Slave die Verarbeitung neuer Updates vom Master startet.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 11 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Wenn der Master zuvor ohne aktiviertes Binär-Logging ausgeführt wurde, sind die von `SHOW MASTER STATUS` oder `mysqldump --master-data` angezeigten Werte für Log-Name und Position leer. In diesem Fall lauten die Werte, die Sie später als Log-Dateinamen und Position auf dem Slave angeben müssen, (Leer-String) und 4.

Nachdem Sie die Momentaufnahme erstellt und Log-Name und Versatz aufgezeichnet haben, können Sie die Schreibaktivitäten auf dem Master neu starten:

```
MariaDB [(none)]> UNLOCK TABLES;
```

Listing 1.13 Schreibaktivitäten wieder starten

Notieren Sie den Log-Namen und Versatz aus der Ausgabe von `SHOW MASTER STATUS` (siehe Listing 1.12) wie zuvor beschrieben. Danach fahren Sie den Server herunter, ohne die Tabellen zu entsperren; hierdurch ist sichergestellt, dass der Server mit einem Status beendet wird, der den Angaben zu Log-Datei und Versatz entspricht:

```
# mysqladmin -u root shutdown
```

Listing 1.14 Den MariaDB-Server stoppen

Die Alternative: SQL-Speicherauszug

Eine Alternative, die sowohl bei MyISAM- als auch bei InnoDB-Tabellen funktioniert, besteht darin, einen SQL-Speicherauszug des Masters anstatt – wie zuvor beschrieben – einer binären Kopie zu erstellen. Hierzu führen Sie `mysqldump --master-data` auf dem Master aus und laden die SQL-Speicherauszugsdatei später in Ihren Slave. Diese Vorgehensweise ist langsamer als die Erstellung einer binären Kopie, hat aber dafür den Vorteil, dass der Slave dabei nicht gestoppt werden muss.

Der Name der binären Log-Datei und der Versatz sind in diesem Fall im Dump zu finden. Die genaue Vorgehensweise finden Sie im folgenden Unterabschnitt »Konfiguration des Slave-Servers«.

Konfiguration des Slave-Servers

Um den Slave-Server zu konfigurieren, nehmen Sie auch auf ihm zunächst die Änderungen an der Datei *adminbuch.cnf* vor. Für den Slave-Betrieb genügt es, den Parameter `server_id` zu setzen:

```
[mysqld]
server_id = 101
```

Listing 1.15 Server-ID des Slaves konfigurieren

Falls Sie die Master-Master-Replikation nutzen möchten, fügen Sie an dieser Stelle in der *adminbuch.cnf* auch die weiteren Parameter hinzu:

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 12 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
log_bin                 = /var/log/mysql/mariadb-bin.log
max_binlog_size         = 100M
innodb_flush_log_at_trx_commit=1
sync_binlog=1
```

Listing 1.16 Weitere Einstellungen für die Master-Master-Replikation

Falls Sie unter CentOS arbeiten, müssen Sie auch auf dem Slave nach dem Muster aus Listing 1.10 das Verzeichnis */var/log/mysql* anlegen und für den User *mysql* beschreibbar machen.

Start der Master-Slave-Replikation

Zunächst wird auf dem Master eine Datenbank benötigt, die repliziert werden soll. Eine Beispieldatenbank mit nur einer Tabelle und zwei Datensätzen legen Sie wie folgt an:

```
MariaDB [(none)]> create database sonnensystem;
Query OK, 1 row affected (0.01 sec)

MariaDB [(none)]> use sonnensystem;
Database changed

MariaDB [sonnensystem]> create table planeten (id integer, name varchar(20));
Query OK, 0 rows affected (0.03 sec)

MariaDB [sonnensystem]> insert into planeten values (1,'Merkur'),(2,'Venus');
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0
```

Listing 1.17 Eine einfache Beispieldatenbank

Anschließend müssen Sie die Log-Datei und die Position in der Log-Datei ermitteln, um die Synchronisation sauber zu starten. Alle Schreibzugriffe auf die Datenbank müssen unterbunden werden (siehe Listing 1.18). Bitte beachten Sie: Sie dürfen den Client nach dem `flush tables with read lock` nicht verlassen, da sonst der Lock aufgehoben wird.

```
MariaDB [sonnensystem]> flush tables with read lock;
Query OK, 0 rows affected (0.00 sec)

MariaDB [sonnensystem]> show master status;
+--------------------+----------+--------------+------------------+
| File               | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+--------------------+----------+--------------+------------------+
| mariadb-bin.000001 |     1741 |              |                  |
+--------------------+----------+--------------+------------------+
1 row in set (0.00 sec)
```

Listing 1.18 Schreibzugriffe unterbinden und aktuellen Status ermitteln

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 13 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Diese Datenbank wird nun in einer separaten Session in eine Datei exportiert und auf den Slave-Server kopiert:

```
mysqldump sonnensystem --databases --master-data -u root -p > sonnensystem.sql
scp sonnensystem.sql slave:/tmp/
```

Listing 1.19 Die Beispieldatenbank exportieren und auf den Slave-Server kopieren

Wechseln Sie nun auf den Slave-Server, und spielen Sie die Datei in MariaDB ein:

```
mysql -u root -p < /tmp/sonnensystem.sql
```

Listing 1.20 Die Beispieldatenbank auf dem Slave-Server in MariaDB einspielen

Kontrollieren Sie anschließend am MariaDB-Prompt, ob die Datenbank vollständig und korrekt angekommen ist:

```
MariaDB [(none)]> use sonnensystem;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [sonnensystem]> select * from planeten;
+------+--------+
| id   | name   |
+------+--------+
|    1 | Merkur |
|    2 | Venus  |
+------+--------+
2 rows in set (0.00 sec)
```

Listing 1.21 Die Beispieldatenbank auf dem Slave-Server überprüfen

Das hat funktioniert. Fügen Sie jetzt auf dem Master-Server einen weiteren Datensatz zur Tabelle hinzu. Dazu muss natürlich zuerst der Lock aufgehoben werden:

```
MariaDB [sonnensystem]> unlock tables;
Query OK, 0 rows affected (0.00 sec)

MariaDB [sonnensystem]> insert into planeten values (3,'Erde');
Query OK, 1 row affected (0.01 sec)
```

Listing 1.22 Auf dem Master einen weiteren Datensatz zur Beispieldatenbank hinzufügen

Danach starten Sie die Replikation auf dem Slave so wie in Listing 1.23 beschrieben:

```
MariaDB [sonnensystem]> show slave status;
Empty set (0.00 sec)
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 14 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
MariaDB [sonnensystem]> CHANGE MASTER TO MASTER_HOST='masterserver' ,
-> MASTER_USER='repl',
-> MASTER_PASSWORD='slavepass',
-> MASTER_PORT=3306,
-> MASTER_LOG_FILE='mariadb-bin.000001',
-> MASTER_LOG_POS=1741,
-> MASTER_CONNECT_RETRY=10;

MariaDB [sonnensystem]> START SLAVE;
```

Listing 1.23 Die Replikation starten

Der letzte Befehl erzeugt keine Fehlermeldungen oder Ausgaben. Untersuchen Sie jetzt noch einmal `SHOW SLAVE STATUS`:

```
MariaDB [sonnensystem]> SHOW SLAVE STATUS\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: masterserver
                  Master_User: repl
                  Master_Port: 3306
                Connect_Retry: 10
              Master_Log_File: mariadb-bin.000001
          Read_Master_Log_Pos: 1953
               Relay_Log_File: mariadb-relay-bin.000002
                Relay_Log_Pos: 741
        Relay_Master_Log_File: mariadb-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 1953
              Relay_Log_Space: 1037
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 15 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 10
1 row in set (0.00 sec)
```

Listing 1.24 Am MariaDB-Prompt den Slave-Status anzeigen lassen

Der gerade neu hinzugefügte Datensatz (siehe Listing 1.22) sollte nun auf den Slave repliziert worden sein. Kontrollieren Sie dies durch eine einfache `SELECT`-Abfrage auf dem Slave:

```
MariaDB [sonnensystem]> select * from planeten;
+------+--------+
| id   | name   |
+------+--------+
|    1 | Merkur |
|    2 | Venus  |
|    3 | Erde   |
+------+--------+
3 rows in set (0.00 sec)
```

Listing 1.25 Prüfen, ob der neue Datensatz repliziert wurde

Es hat funktioniert: Der Datensatz mit der `ID 3` ist angekommen. Wenn Sie nun einen weiteren Datensatz auf dem Master anlegen, sollte er praktisch ohne Verzögerung auf dem Slave ankommen. Fügen Sie auf dem Master noch einen vierten Datensatz hinzu:

```
MariaDB [sonnensystem]> insert into planeten values (4,'Mars');
Query OK, 1 row affected (0.00 sec)
```

Listing 1.26 Einen weiteren Datensatz auf dem Master hinzufügen

Mit der bekannten `SELECT`-Abfrage sollten Sie den Datensatz auf dem Slave sehen können.

```
MariaDB [sonnensystem]> select * from planeten;
+------+--------+
| id   | name   |
+------+--------+
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 16 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
|    1 | Merkur |
|    2 | Venus  |
|    3 | Erde   |
|    4 | Mars   |
+------+--------+
4 rows in set (0.00 sec)
```

Listing 1.27 Kontrolle auf dem Slave-Server

Damit haben Sie Ihre Master-Slave-Replikation erfolgreich getestet.

1.1.3 Master-Master-Replikation

Sie können die bestehende Master-Slave-Replikation mit nur geringem Aufwand zu einer Master-Master-Replikation erweitern. Dazu legen Sie zunächst auch auf dem jetzigen Slave-Server ein Benutzerkonto für die Replikation an:

```
MariaDB [sonnensystem]> GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.*
    -> TO repl@'192.168.1.%' IDENTIFIED BY 'slavepass';

### oder ###

MariaDB [sonnensystem]> GRANT REPLICATION SLAVE, REPLICATION CLIENT ON *.*
    -> TO 'repl'@'%.mydomain.com' IDENTIFIED BY 'slavepass';
```

Listing 1.28 Einen weiteren Benutzer für die Replikation anlegen

Führen Sie nun – immer noch auf dem bisherigen Slave-Server – am MariaDB-Prompt das Kommando `SHOW MASTER STATUS` aus. Bitte setzen Sie vorher den »read lock«.

```
MariaDB [sonnensystem]> flush tables with read lock;
Query OK, 0 rows affected (0.00 sec)

MariaDB [sonnensystem]> show master status;
+--------------------+----------+--------------+------------------+
| File               | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+--------------------+----------+--------------+------------------+
| mariadb-bin.000002 |       98 |              |                  |
+--------------------+----------+--------------+------------------+
1 row in set (0.00 sec)
```

Listing 1.29 »show master status« auf dem bisherigen Slave-Server

Merken oder notieren Sie sich die Werte für `File` und `Position`. Sie benötigen diese Daten im nächsten Schritt.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 17 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Wechseln Sie nun auf den bisherigen Master-Server, und führen Sie am MariaDB-Prompt die Kommandos aus, die in Listing 1.30 hinter dem Prompt `MariaDB [sonnensystem]>` stehen, nachdem Sie mittels `use sonnesystem;` zur Datenbank *sonnensystem* gewechselt sind:

```
MariaDB [sonnensystem]> show slave status;
Empty set (0.00 sec)
MariaDB [sonnensystem]> change master to master_host='slaveserver',
    -> master_user='repl',
    -> master_password='slavepass',
    -> master_log_file='mariadb-bin.000002',
    -> master_log_pos=98,
    -> master_connect_retry=10;
Query OK, 0 rows affected (0.01 sec)

MariaDB [sonnensystem]> start slave;
Query OK, 0 rows affected (0.01 sec)

MariaDB [sonnensystem]> show slave status;
             Slave_IO_State: Waiting for master to send event
                Master_Host: slaveserver
                Master_User: repl
                Master_Port: 3306
              Connect_Retry: 60
            Master_Log_File: mariadb-bin.000002
        Read_Master_Log_Pos: 98
             Relay_Log_File: mysqld-relay-bin.000002
              Relay_Log_Pos: 235
      Relay_Master_Log_File: mariadb-bin.000002
           Slave_IO_Running: Yes
          Slave_SQL_Running: Yes
            Replicate_Do_DB:
        Replicate_Ignore_DB:
         Replicate_Do_Table:
     Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
                 Last_Errno: 0
                 Last_Error:
               Skip_Counter: 0
        Exec_Master_Log_Pos: 98
            Relay_Log_Space: 235
            Until_Condition: None
             Until_Log_File:
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 18 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
           Until_Log_Pos: 0
        Master_SSL_Allowed: No
        Master_SSL_CA_File:
        Master_SSL_CA_Path:
           Master_SSL_Cert:
         Master_SSL_Cipher:
            Master_SSL_Key:
     Seconds_Behind_Master: 0
1 row in set (0.00 sec)
```

Listing 1.30 Den bisherigen Slave zum zweiten Master-Server befördern

Das letzte `SELECT`-Statement in Listing 1.31 zeigt den Inhalt der Tabelle `Planeten` auf dem bisherigen Master-Server.

```
MariaDB [sonnensystem]> unlock tables;
Query OK, 0 rows affected (0.00 sec)

MariaDB [sonnensystem]> insert into planeten values (5,'Jupiter'),(6,'Saturn');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

MariaDB [sonnensystem]> select * from planeten;
+------+---------+
| id   | name    |
+------+---------+
|    1 | Merkur  |
|    2 | Venus   |
|    3 | Erde    |
|    4 | Mars    |
|    5 | Jupiter |
|    6 | Saturn  |
+------+---------+
6 rows in set (0.00 sec)
```

Listing 1.31 Zwei weitere Planeten anlegen

Wenn Sie nun auf dem ehemaligen Slave-Server einen Datensatz hinzufügen, wird er auf den Master repliziert werden. Wechseln Sie also auf den Ex-Slave-Server, und fügen Sie der Tabelle einen weiteren Planeten hinzu:

```
MariaDB [sonnensystem]> insert into planeten values (7,'Uranus');
Query OK, 1 row affected (0.01 sec)
```

Listing 1.32 Einen Datensatz auf dem ehemaligen Slave-Server hinzufügen

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 19 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Wechseln Sie nun zurück auf den Master-Server, und kontrollieren Sie, ob der Datensatz auch dorthin repliziert wurde (siehe Listing 1.33). Sollte das nicht der Fall sein, gehen Sie bitte die vorhergehenden Schritte noch einmal durch.

```
MariaDB [sonnensystem]> select * from planeten;
+------+---------+
| id   | name    |
+------+---------+
|    1 | Merkur  |
|    2 | Venus   |
|    3 | Erde    |
|    4 | Mars    |
|    5 | Jupiter |
|    6 | Saturn  |
|    7 | Uranus  |
+------+---------+
7 rows in set (0.00 sec)
```

Listing 1.33 Erfolgskontrolle auf dem ehemaligen Master-Server

MariaDB erlaubt explizit die Ring-Replikation (*Multi-Master-Replikation*). Damit können Sie mehr als zwei Server zu einem Ring zusammenfassen. Wenn der Ring unterbrochen ist, ist an der Stelle auch die Replikation für alle beteiligten Server unterbrochen. Hier sind besondere Vorkehrungen zu treffen. Weitere Informationen dazu finden Sie auf den Webseiten, die von der folgenden URL aus verlinkt sind: *https://mariadb.com/kb/en/library/multi-source-replication*

1.2 Tuning

Beim Tuning eines MariaDB-Servers ist es das zentrale Ziel, die Festplattenlast, also die Anzahl der Ein-/Ausgabeoperationen (I/O), zu verringern. Während ein RAM-Zugriff in wenigen Nanosekunden erfolgt, bewegt man sich bei Festplattenzugriffen bereits im Millisekundenbereich, weil das Positionieren der Schreib-/Leseköpfe nun einmal diese Zeit in Anspruch nimmt. Auch die Drehgeschwindigkeit der Festplattenscheiben kann nicht beliebig gesteigert werden. Das vorher Gesagte spielt zwar im Zeitalter von SSDs und NVMes eine untergeordnete Rolle, ist aber trotzdem nicht zu vernachlässigen.

Ferner sollten Sie auf die Latenz und die Geschwindigkeit der Netzwerkanbindung achten, denn wenn der Server seine Anfragen zu langsam erhält oder seine Antworten nicht schnell genug loswird, geht wertvolle Zeit verloren.

Sollten Sie mit dem Gedanken spielen, Ihren Datenbankserver mit besserer Hardware zu bestücken, so sollten Sie zunächst in RAM investieren. Schnellere CPUs bringen nur bei komple-

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 20 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

xen Rechenoperationen einen Vorteil. Der Umstieg von lokalen Festplatten auf SSDS, NVMes oder ein schnelles, gut angebundenes SAN beschleunigt das I/O-Verhalten. In den folgenden Abschnitten geht es aber nicht um Hardware-Tuning.

Stattdessen werden Sie lernen, wie Sie mit den Bordmitteln des Betriebssystems und MariaDB die Speichernutzung und das Laufzeitverhalten Ihres Servers optimieren können.

Nutzen Sie »mytop«

Das Programm *mytop* zeigt Ihnen eine ständig aktualisierte Übersicht darüber, womit Ihr MariaDB-Server gerade beschäftigt ist. Es liefert auch eine Reihe grundlegender Messwerte, an denen Sie ablesen können, ob Ihre Tuning-Maßnahmen erfolgreich sind.

1.2.1 Tuning des Speichers

Unter der URL *https://launchpad.net/mysql-tuning-primer* erhalten Sie das sehr nützliche *bash*-Skript *mysql-tuning-primer.sh*. Wenn Sie Debian oder Ubuntu nutzen, ändern Sie am Anfang des Skripts `/bin/sh` in `/bin/bash`. Das ist notwendig, weil `/bin/sh` unter Ubuntu ein Link auf `/bin/dash` ist. Beachten Sie bitte auch, dass das Skript das Programm *bc* erfordert. Bitte installieren Sie es, falls es noch nicht vorhanden ist. Führen Sie das Skript auf dem Server aus, den Sie tunen möchten. Beachten Sie dabei die beiden folgenden Punkte:

- Der MariaDB-Server sollte nicht brandneu, sondern schon ein wenig »eingefahren« sein, sonst liefern die Analysen der Speicher- und Cache-Nutzung irreführende Ergebnisse.
- Der Server sollte mindestens 48 Stunden laufen, bevor das Skript gestartet wird, damit es einen guten Überblick über die typische Lastsituation auf dem Server bekommt. Das Skript sollte außerdem zu den normalen Betriebszeiten laufen. Wenn der Server beim Start des Skripts noch keine 48 Stunden Uptime aufweist, erhalten Sie folgende Warnung:

  ```
  Warning: Server has not been running for at least 48hrs.
  It may not be safe to use these recommendations
  ```

  Listing 1.34 Warnung vor zu kurzer Laufzeit des Systems

Auf den folgenden Seiten finden Sie die für Ihre Analysen relevanten Ausgaben des Skripts *mysql-tuning-primer.sh* und Hinweise darauf, wie Sie die Ausgaben interpretieren und als Grundlage für Tuning-Maßnahmen nutzen können.

Slow Queries

```
SLOW QUERIES
The slow query log is NOT enabled.
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 21 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
Current long_query_time = 10.000000 sec.
You have 0 out of 1908 that take longer than 10.000000 sec. to complete
Your long_query_time seems to be fine
```

Listing 1.35 Skriptausgabe: »Slow Queries«

Die Behandlung von *Slow Queries*, also von langsamen Abfragen, wird in Abschnitt 1.2.2, »Tuning von Indizes«, näher betrachtet.

Binary Update Log

```
BINARY UPDATE LOG
The binary update log is NOT enabled.
You will not be able to do point in time recovery
See http://dev.mysql.com/doc/refman/5.1/en/point-in-time-recovery.html
```

Listing 1.36 Skriptausgabe: »Binary Update Log«

Auch das *Binary Update Log* begegnet uns erst an späterer Stelle wieder, und zwar in Abschnitt 1.3, »Backup und Point-In-Time-Recovery«.

Max Connections

```
MAX CONNECTIONS
Current max_connections = 151
Current threads_connected = 1
Historic max_used_connections = 1
The number of used connections is 0% of the configured maximum.
You are using less than 10% of your configured max_connections.
Lowering max_connections could help to avoid an over-allocation of memory
See "MEMORY USAGE" section to make sure you are not over-allocating
```

Listing 1.37 Skriptausgabe: »Max Connections«

Jede Verbindung benötigt 2 MB Speicher. Dazu kommen aber zusätzlich noch die Puffer und Caches, deren Größe in den Variablen `sort_buffer_size`, `read_buffer_size` und `binlog_cache_size` definiert ist. Das Skript hat festgestellt, dass wesentlich weniger Verbindungen benutzt werden, als im Parameter `max_connections` konfiguriert sind. Da für jede potenzielle Verbindung Speicher reserviert wird, ist es sinnvoll, den `max_connections`-Wert abzusenken.

In der MariaDB-Shell geht das wie folgt:

```
MariaDB [(none)]> set global max_connections=10;
Query OK, 0 rows affected (0.00 sec)
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 22 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
MariaDB [(none)]> show variables like 'max_connections';
+-----------------+-------+
| Variable_name   | Value |
+-----------------+-------+
| max_connections | 10    |
+-----------------+-------+
1 row in set (0.00 sec)
```

Listing 1.38 Der Wert »max_connections« wird zunächst reduziert, dann kontrolliert.

Die Einstellung des `max_connections`-Werts wird auf diese Weise nur temporär verändert. Wenn Sie den Wert dauerhaft verändern wollen, gehen Sie wie folgt vor:

- **Debian/Ubuntu:** Legen Sie eine neue Datei im Verzeichnis */etc/mysql/conf.d/* an, deren Name auf *.cnf* enden muss, bespielsweise */etc/mysql/conf.d/max_connections.cnf*. In die Datei schreiben Sie den *Section Header* `[mysqld]` und den gewünschten neuen Wert der Variablen `max_connections`:

  ```
  [mysqld]
  max_connections=10
  ```

  Listing 1.39 Inhalt der Datei »/etc/mysql/conf.d/max_connections.cnf«

- **CentOS/openSUSE:** Speichern Sie die Datei aus Listing 1.39 unter */etc/my.cnf.d* ab.

InnoDB Status

```
INNODB STATUS
Current InnoDB index space = 0 bytes
Current InnoDB data space = 0 bytes
Current InnoDB buffer pool free = 96 %
Current innodb_buffer_pool_size = 8 M
Depending on how much space your innodb indexes take up it may be safe
to increase this value to up to 2 / 3 of total system memory
```

Listing 1.40 Skriptausgabe: »InnoDB Status«

Die verwendete Größe der *InnoDB-Buffer-Pools* können Sie über den Wert der Variablen `innodb_buffer_pool_size` anpassen. Das Vorgehen ist das gleiche wie bei der Variablen `max_connections`. In diesem Beispiel ist keine Änderung notwendig.

Memory Usage

```
MEMORY USAGE
Max Memory Ever Allocated : 44 M
Configured Max Per-thread Buffers : 405 M
Configured Max Global Buffers : 42 M
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 23 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
Configured Max Memory Limit : 447 M
Physical Memory : 244 M
Max memory limit exceeds 90% of physical memory
```

Listing 1.41 Skriptausgabe: »Memory Usage«

[!] Diesen Wert müssen Sie unbedingt im Auge behalten! Sollte das konfigurierte *Memory Limit* über dem tatsächlich verfügbaren Speicher liegen, kann es zum *Swappen* (zur Nutzung von Auslagerungsspeicher) kommen. Der Server würde in diesem Fall sehr langsam werden.

Key Buffer

```
KEY BUFFER
Current MyISAM index space = 301 M
Current key_buffer_size = 64 M
Key cache miss rate is 1 : 156
Key buffer free ratio = 40 %
Your key_buffer_size seems to be too high.
Perhaps you can use these resources elsewhere
```

Listing 1.42 Skriptausgabe: »Key Buffer«

Die Indexblöcke aller *MyISAM*-Tabellen werden in einen Pufferspeicher[2] geschrieben, der von allen Threads gemeinsam genutzt wird. Die Variable `key_buffer_size` enthält die Größe dieses Pufferspeichers. In dem Beispiel aus Listing 1.42 ist er ein wenig zu groß und könnte gefahrlos reduziert werden.

Query Cache

```
QUERY CACHE
Query cache is enabled
Current query_cache_size = 128 M
Current query_cache_used = 74 M
Current query_cache_limit = 4 M
Current Query cache Memory fill ratio = 57.94 %
Current query_cache_min_res_unit = 4 K
MariaDB won't cache query results that are larger than query_cache_limit in size
```

Listing 1.43 Skriptausgabe: »Query Cache«

Die Variable `query_cache_size` definiert die Größe des Caches, der zum Zwischenspeichern von Abfrageergebnissen insgesamt zur Verfügung steht. Der Standardwert ist `0` (Null), das bedeutet, dass der Cache deaktiviert ist. Im Beispiel wurden dem Cache 128 MB zur Verfügung gestellt.

2 Dieser Pufferspeicher für die *MyISAM*-Indexblöcke wird auch *Schlüssel-Cache* genannt.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 24 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Es werden nur die Ergebnisse zwischengespeichert, die nicht größer sind als der Wert in `query_cache_limit`. In diesem Beispiel liegt das Limit bei 4 MB, der Default-Wert ist 1 MB.

Sort Operations

```
SORT OPERATIONS
Current sort_buffer_size = 8 M
Current read_rnd_buffer_size = 256 K
Sort buffer seems to be fine
```

Listing 1.44 Skriptausgabe: »Sort Operations«

Der Sortierungspuffer (*Sort Buffer*) wird beispielsweise für Abfragen benötigt, die ein `ORDER BY`-Statement enthalten. Je nach Anzahl der Datensätze, die regelmäßig sortiert werden müssen, kann dieser Wert größer oder kleiner gewählt werden.

Joins

```
JOINS
Current join_buffer_size = 2.00 M
You have had 1138756 queries where a join could not use an index properly
You should enable "log-queries-not-using-indexes"
Then look for non indexed joins in the slow query log.

If you are unable to optimize your queries you may want to increase your
join_buffer_size to accommodate larger joins in one pass.
Note! This script will still suggest raising the join_buffer_size when
ANY joins not using indexes are found.
```

Listing 1.45 Skriptausgabe: »Joins«

Ein Beispiel dieses Werts wird in Abschnitt 1.2.2, »Tuning von Indizes«, näher beleuchtet.

Open Files Limit

```
OPEN FILES LIMIT
Current open_files_limit = 4096 files

The open_files_limit should typically be set to at least 2x-3x
that of table_cache if you have heavy MyISAM usage.
Your open_files_limit value seems to be fine
```

Listing 1.46 Skriptausgabe: »Open Files Limit«

Dies ist kein MariaDB-Parameter: Für die Limitierung der gleichzeitig geöffneten Dateien ist das Betriebssystem zuständig. Um diesen Wert auf 4096 festzulegen, geben Sie vor dem Start von MariaDB als *root* das Kommando `ulimit -n 4096` auf der Kommandozeile ein.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 25 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Table Cache

```
Current table_cache value = 1993 tables
You have a total of 780 tables

You have 1582 open tables.
The table_cache value seems to be fine
```

Listing 1.47 Skriptausgabe: »Table Cache«

Dies ist die Anzahl der geöffneten Tabellen für alle Threads. Wenn Sie nun den Wert für `table_cache` erhöhen, benötigt MariaDB mehr Dateideskriptoren. Durch die Eingabe des Befehls `flush tables` in der MariaDB-Shell werden alle offenen Tabellen geschlossen und nur diejenigen wieder geöffnet, die tatsächlich benötigt werden.

Temp Tables

```
TEMP TABLES
Current max_heap_table_size = 32 M
Current tmp_table_size = 32 M

Of 744743 temp tables, 15% were created on disk
Created disk tmp tables ratio seems fine
```

Listing 1.48 Skriptausgabe: »Temp Tables«

Wenn Sie viele Abfragen mit `GROUP BY`-Statements haben, ist es vorteilhaft, diese Werte (im Beispiel 32 MB) zu erhöhen, sofern Sie ausreichend RAM zur Verfügung haben.

Wächst die Größe der temporären Tabelle im Arbeitsspeicher über diese Beschränkung hinaus, wird MariaDB sie automatisch in eine *MyISAM*-Tabelle auf der Festplatte umwandeln, was sich natürlich nachteilig auf die Zugriffsgeschwindigkeit auswirken wird.

Table Scans

```
TABLE SCANS
Current read_buffer_size = 508 K
Current table scan ratio = 2313 : 1
read_buffer_size seems to be fine
```

Listing 1.49 Skriptausgabe: »Table Scans«

Ein *Table Scan* ist unvermeidlich, wenn eine Tabelle keinen Index besitzt. Manchmal führt MariaDB trotz des vorhandenen Index einen *Table Scan* aus. Das passiert, wenn der eingebaute Optimierer der Ansicht ist, es sei schneller, die Tabelle komplett zu lesen, als nur einzelne Datensätze herauszusuchen.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 26 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Table Locking

```
TABLE LOCKING
Current Lock Wait ratio = 1 : 1198

You may benefit from selective use of InnoDB.
If you have long running SELECT's against MyISAM tables and perform
frequent updates consider setting 'low_priority_updates=1'
If you have a high concurrency of inserts on Dynamic row-length tables
consider setting 'concurrent_insert=2'.
```

Listing 1.50 Skriptausgabe: »Table Locking«

Wenn Sie *MyISAM*-Tabellen benutzen, werden Sie sehr wahrscheinlich davon profitieren, den Parameter `concurrent_insert=2` zu aktivieren. MariaDB erlaubt damit die gleichzeitige Ausführung von `SELECT`- und `INSERT`-Statements. Nehmen Sie sich die Zeit, die für Sie beste Kombination dieser Parameter herauszufinden. Da es kein Patentrezept gibt, werden Sie um eigene Tests nicht herumkommen. Die Mühe lohnt sich, denn mit den richtigen Einstellungen holen Sie das Maximum aus Ihrem Datenbankserver heraus.

1.2.2 Tuning von Indizes

Im vorigen Abschnitt wurden *Slow Queries* bereits ebenso erwähnt wie Abfragen, die keine Indizes benutzen. MariaDB kann alle Abfragen, die länger als eine bestimmte Zeit benötigen, in eine darauf spezialisierte Log-Datei schreiben, in das *Slow Query Log*. Aktiviert wird es durch den folgenden Eintrag in der MariaDB-Konfigurationsdatei:

```
log_slow_queries = /var/log/mysql/mysql-slow.log
long_query_time = 2
log-queries-not-using-indexes
```

Listing 1.51 Aktivierung des »Slow Query Log«

In diesem Beispiel wandern alle Abfragen, die länger als zwei Sekunden brauchen, in die Log-Datei */var/log/mysql/mysql-slow.log*. Abfragen, die keinen Index benutzen, werden dort ebenfalls protokolliert (`log-queries-not-using-indexes`).

Passen Sie »long_query_time« an

Ob der Schwellenwert von zwei Sekunden für Ihren Datenbankserver vernünftig gewählt ist, ist keineswegs sicher. Wenn Sie ein *Data Warehouse* betreiben, sind lange Abfragezeiten normal. Wenn Sie dagegen eine Webseite oder einen Online-Shop betreiben, kommt es eher auf schnelle Antwortzeiten an. Passen Sie den Wert der Variablen `long_query_time` also an Ihre Bedürfnisse an!

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 27 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Wenn Sie einen hohen Prozentsatz von langsamen Abfragen im *Slow Query Log* finden, lohnt es sich, diesen mit dem SQL-Kommando `EXPLAIN` auf den Grund zu gehen. Das Erzeugen zusätzlicher oder besserer Indizes kann die `SELECT`-Performance drastisch erhöhen, allerdings dauern damit auch `INSERT`- und `UPDATE`-Statements deutlich länger.

Seien Sie in jedem Fall nicht allzu pedantisch. Man muss nicht jede einzelne Abfrage optimieren. Solange es nicht überhandnimmt, dürfen einzelne Abfragen auch einmal etwas länger dauern – in einer Datenbank, in der es sehr häufig Aktualisierungen gibt, ist es vielleicht besser, auf Abfragen zu warten, als die Aktualisierungen warten zu lassen. Was generell besser ist, hängt von Ihrer speziellen Anwendung ab.

Ein Beispiel aus der Praxis

Das *Slow Query Log* können Sie mit dem folgenden Befehl zusammenfassen lassen:

```
mysqldumpslow /var/log/mysql/mysql-slow.log
```

Listing 1.52 Eine Zusammenfassung des »Slow Query Log« erzeugen

Wenn Sie als Parameter noch ein `-r` eingeben, können Sie sich die am längsten laufenden Abfragen zum Schluss ausgeben lassen. Achtung: Damit finden Sie nur den Typ der Abfrage, Werte werden durch den Platzhalter »N« ersetzt.

Wenn Sie nun im *slow-query.log* nach einer Abfrage dieses Typs suchen, können Sie diese mit dem Befehl `EXPLAIN` erklären lassen. So lässt sich beispielsweise in der Datenbank eines Mailservers folgende Abfrage definieren:

```
MariaDB [mailserver]> select * from virtual_aliases where \
                      destination='mail@example.com';
```

Listing 1.53 Eine einfache Datenbankabfrage

Mit `EXPLAIN` können Sie sehen, wie der MariaDB-Server die Abfrage bearbeitet:

```
MariaDB [mailserver]> explain select * from virtual_aliases where \
                      destination='mail@example.com';

+----+-------------+-----------------+------+---------------+-------+---------
| id | select_type | table           | type | possible_keys | key   | key_len
+----+-------------+-----------------+------+---------------+-------+---------
|  1 | SIMPLE      | virtual_aliases | ALL  | NULL          | NULL  | NULL
+----+-------------+-----------------+------+---------------+-------+---------

-+------+------+-------------+
 | ref  | rows | Extra       |
-+------+------+-------------+
 | NULL |  127 | Using where |
```


Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 28 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
-+------+------+-------------+

1 row in set (0.00 sec)
```

Listing 1.54 Anwendung des »explain«-Statements

Schauen wir uns nun die einzelnen Elemente der Ausgabe an:

- `id`
  Die `id` gibt die Sequenznummer der `SELECT`-Anweisung innerhalb der Abfrage an.
- `select_type`
  Tabelle 1.1 zeigt die möglichen `SELECT`-Typen:

Typ	Bedeutung
`PRIMARY`	äußerste `SELECT`-Anweisung
`SIMPLE`	einfache `SELECT`-Anweisung (ohne `UNION` oder Unterabfragen)
`DERIVED`	abgeleitete Tabellen-`SELECT`-Anweisung (Unterabfrage in `FROM`-Klausel)
`SUBQUERY`	erste `SELECT`-Anweisung in einer Unterabfrage
`DEPENDENT SUBQUERY`	erste `SELECT`-Anweisung in einer Unterabfrage, abhängig von der äußeren Abfrage
`UNCACHABLE SUBQUERY`	erste `SELECT`-Anweisung in einer Unterabfrage, die nicht gecacht werden kann
`UNION`	zweite oder spätere `SELECT`-Anweisung in einer `UNION`
`UNION RESULT`	Ergebnis einer `UNION`
`DEPENDENT UNION`	zweite oder spätere `SELECT`-Anweisung in einer `UNION`, abhängig von der äußeren Abfrage
`UNCACHABLE UNION`	nicht cachbare `UNION`

Tabelle 1.1 »SELECT«-Typen

- `table`
  Dies ist der Name der Tabelle, auf die sich die Abfrage bezieht.
- `type`
  Hier wird der `Join`-Typ angegeben. `ALL` bedeutet, dass ein kompletter Tablescan durchgeführt wird. Das kann bei großen Tabellen sehr lange dauern, es besteht Handlungsbedarf. `INDEX` hätte angegeben, dass der Indexbaum gescannt worden ist. Weitere Möglichkei-

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 29 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

ten wären (von der schlechtesten zur besten): `range`, `index_subquery`, `unique_subquery`, `index_merge`, `ref_or_null`, `ref`, `eq_ref`, `const`, `system`. Die Erläuterungen hierzu entnehmen Sie bitte der Webseite zu `EXPLAIN` unter der Adresse *http://dev.mysql.com/doc/refman/5.1/en/explain.html*.

- `possible_keys`
  Dieser Wert gibt an, welche Keys (Indizes) möglich wären; in unserem Fall keiner.
- `key`
  Das ist der Index, der letztendlich benutzt wurde.
- `key_len`
  Der Wert zeigt die Länge des Schlüssels an.
- `rows`
  Dies ist die Anzahl der Zeilen, die MariaDB in seine Untersuchung mit einbezieht.
- `extra`
  Hier zeigt MariaDB an, wie es die Abfrage auflöst.

Für unser Beispiel ist die Erkenntnis wichtig, dass es keinen passenden Index gibt. Mit dem folgenden Kommando am MariaDB-Prompt erzeugen wir ihn:

```
MariaDB [(none)]> create index idx_destination on virtual_aliases(destination);
Query OK, 127 rows affected (0.12 sec)
Records: 127  Duplicates: 0  Warnings: 0
```

Listing 1.55 Nachträglich einen Index erzeugen

Mit einem weiteren `EXPLAIN`-Statement kontrollieren wir den Erfolg:

```
MariaDB [(none)]> explain select * from virtual_aliases where \
                  destination='mail@example.com';

+----+-------------+-----------------+------+-----------------+-----------------
| id | select_type | table           | type | possible_keys   | key
+----+-------------+-----------------+------+-----------------+-----------------
|  1 | SIMPLE      | virtual_aliases | ref  | idx_destination | idx_destination
+----+-------------+-----------------+------+-----------------+-----------------

-+---------+-------+------+-------------+
 | key_len | ref   | rows | Extra       |
-+---------+-------+------+-------------+
 | 82      | const |    1 | Using where |
-+---------+-------+------+-------------+
```

Listing 1.56 Erfolgskontrolle der Indexerstellung

Dieser Index würde Abfragen auf die Datenbank in Zukunft deutlich beschleunigen.

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 30 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Zusammenfassung der Indexerstellung

1. Schalten Sie das *Slow Query Log* ein.
2. Finden Sie mit `mysqldumpslow` die Abfrageklasse, die das System am stärksten beeinflusst.
3. Suchen Sie aus dem *Slow Query Log* ein typisches Beispiel dafür.
4. Untersuchen Sie dieses Beispiel mithilfe des `EXPLAIN`-Kommandos.
5. Erzeugen Sie einen (zusätzlichen) Index.

1.3 Backup und Point-In-Time-Recovery

Sicherlich legen Sie in bestimmten Abständen ein vollständiges Backup Ihrer Datenbanken an. Sollte es zu einem Datenverlust kommen, können Sie das Backup hoffentlich wieder einspielen (*Restore*). Änderungen, die seit dem Anlegen des Backups vorgenommen wurden, sind aber leider verloren – es sei denn, Sie nutzen *Point-In-Time-Recovery*. *Point-In-Time-Recovery* bedeutet, dass Sie die Änderungen am Datenbestand, die seit einem bestimmten Zeitpunkt in der Vergangenheit vorgenommen worden sind, ebenfalls restaurieren können. Das Mittel, das Ihnen diesen Kunstgriff ermöglicht, ist das *binäre Logging*. Dabei werden die Änderungen inkrementell in Binärdateien geschrieben, aus denen sie im Bedarfsfall wiederhergestellt werden können.

Damit Sie *Point-In-Time-Recovery* nutzen können, müssen Sie zunächst dafür sorgen, dass Ihr Server die binären Logfiles schreibt. Fügen Sie der MariaDB-Konfiguration dazu folgende Zeilen hinzu (bitte denken Sie unter CentOS daran, das Verzeichnis zu erstellen und zu berechtigen; sehen Sie sich dazu auch Listing 1.10 an):

```
log_bin                 = /var/log/mysql/mariadb-bin.log
max_binlog_size         = 100M
innodb_flush_log_at_trx_commit=1
sync_binlog=1
```

Listing 1.57 Binäres Logging aktivieren

Nach einem Neustart des Datenbankservers werden die gesamten binären Log-Dateien unter */var/log/mysql/* angelegt. Für die folgenden Beispiele nutzen wir eine Datenbank mit dem Namen *sonnensystem*.

1.3.1 Restore zum letztmöglichen Zeitpunkt

Erstellen Sie ein vollständiges Backup der Datenbank mit diesem Kommando:

```
mysqldump --flush-logs --single-transaction sonnensystem -u root -p > backup.sql
```

Listing 1.58 Vollständiges Backup einer Datenbank

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 31 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Wenn nun zu einem späteren Zeitpunkt – das heißt, nachdem seit dem Backup weitere Daten in der Datenbank geändert wurden – eine Katastrophe passiert und ein Restore notwendig wird, spielen Sie zunächst das Backup wieder ein:

```
mysql -u root -p < backup.sql
```

Listing 1.59 Restore der gesicherten Datenbank

Nun müssen noch die Änderungen restauriert werden, die seit dem Erstellen des Backups vorgenommen wurden. Diese Änderungen finden sich in den binären Logfiles.

Finden Sie zunächst heraus, wie viele binäre Logfiles seit dem Erstellen des Backups angelegt wurden. Sollten es mehrere sein, müssen Sie zum Wiederherstellen alle Logfiles angeben. Listing 1.60 zeigt ein Restore aus zwei binären Logfiles (es handelt sich um eine einzige Zeile, die nur aufgrund der Seitenbreite umbrochen wurde):

```
mysqlbinlog --database sonnensystem /var/log/mysql/mariadb-bin.000001 \
/var/log/mysql/mariadb-bin.000002  | mysql -u root -p
```

Listing 1.60 Restore aus den binären Logfiles

1.3.2 Restore zu einem bestimmten Zeitpunkt

Wenn Sie einen falschen Befehl eingegeben haben, wie beispielsweise das versehentliche Löschen einer Tabelle, können Sie per *Point-In-Time-Recovery* auch ein Restore genau bis zu dem Zeitpunkt durchführen, bevor Sie die Tabelle gelöscht haben.

Nehmen wir zum Beispiel an, die Tabelle *planeten* in der Datenbank *sonnensystem* ist mit dem Befehl `drop table planeten` gelöscht worden. Der Fehler ist Ihnen glücklicherweise sofort aufgefallen. Suchen Sie im Verzeichnis */var/log/mysql/* das aktuellste der dort abgelegten binären Logfiles (Sie benötigen das aktuellste, weil der Fehler ja gerade erst passiert ist). In unserer Musterdatenbank hat das Verzeichnis beispielsweise den in Listing 1.61 gezeigten Inhalt:

```
# ls -l /var/log/mysql
total 60
-rw-rw---- 1 mysql mysql   353 28. Aug 17:17 mariadb-bin.000001
-rw-rw---- 1 mysql mysql   353 28. Aug 17:20 mariadb-bin.000002
-rw-rw---- 1 mysql mysql   353  6. Okt 09:29 mariadb-bin.000003
-rw-rw---- 1 mysql mysql  1855  6. Okt 11:48 mariadb-bin.000004
-rw-rw---- 1 mysql mysql   136  6. Okt 10:22 mariadb-bin.index
-rw-rw---- 1 mysql mysql 33085  6. Okt 11:44 mysqld.log
```

Listing 1.61 Inhalt des Verzeichnisses »/var/log/mysql/«

Das aktuellste binäre Logfile ist hier die Datei *mariadb-bin.000004*. Führen Sie auf der Kommandozeile den folgenden Befehl aus:

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 32 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank/datenbank" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

```
mysqlbinlog ---database sonnensystem /var/log/mysql/mariadb-bin.000004
```

Listing 1.62 Den Inhalt des binären Logfiles anschauen

Sie erhalten unter anderem die Ausgabe aus Listing 1.63. Dort sehen Sie, dass an Position 1897 das Löschen der Tabelle gestartet wurde (`at 1897`) und dass es an Position 2019 beendet wurde (`end_log_pos 2019`):

```
# at 1855
#181006 12:06:40 server id 105  end_log_pos 1897 CRC32 0xac3f2172       \
GTID 0-105-11 ddl
/*!100001 SET @@session.gtid_seq_no=11*//*!*/;
# at 1897
#181006 12:06:40 server id 105  end_log_pos 2019 CRC32 0x60355708       \
Query   thread_id=30      exec_time=0     error_code=0
SET TIMESTAMP=1538820400/*!*/;
DROP TABLE `planeten` /* generated by server */
/*!*/;
DELIMITER ;
# End of log file
ROLLBACK /* added by mysqlbinlog */;
/*!50003 SET COMPLETION_TYPE=@OLD_COMPLETION_TYPE*/;
/*!50530 SET @@SESSION.PSEUDO_SLAVE_MODE=0*/;
```

Listing 1.63 Ergebnis der Suche im binären Logfile (Auszug)

Nun spielen Sie das letzte vollständige Backup wieder ein:

```
mysql -u root -p < backup.sql
```

Listing 1.64 Restore der gesicherten Datenbank

Als Nächstes restaurieren Sie alle Änderungen bis zum fatalen `DROP TABLE`:

```
mysqlbinlog --database=sonnensystem /var/log/mysql/mariadb-bin.000004 \
--stop-position=1897 | mysql -u root -p
```

Listing 1.65 Restore der Daten bis zum »Drop Table«

Falls Sie nach dem `DROP TABLE` noch weitergearbeitet haben, können Sie auch diese Änderungen noch zurückholen, indem Sie auch die Daten nach dem Zeitstempel `1897` zurückspielen:

```
mysqlbinlog --database=sonnensystem /var/log/mysql/mariadb-bin.000007 \
--start-position=2019 | mysql -u root -p
```

Listing 1.66 Restore der Daten ab dem Zeitstempel »2019«

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 33 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank.ind" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x

Index

Deimeke/Heinlein/Kania/Miesen/van Soest – Das Linux-Administrationshandbuch – Sechste Auflage – Frühjahr 2021 Rheinwerk Verlag, ISBN: 978-3-8362-8088-3
Layout: gp.cls, Version 3.6.008 (9th May, 2021), (C) Dr. Daniel Lauer, Bonn Mit TEX bearbeitet am 24. September 2021, 10:02 Uhr Umlaut-Check: ÄÖÜäöüß. Ligaturen: fl fi
S. 35 Einstellungen: ohne Marginalien – ohne Seitenrand – mit Taben – mit Kapitel-Intro – extra breit – Zoom auf 100.0%
Aktuelle Quelldatei: "datenbank.ind" , Aktueller Job: datenbank, Aktuelles Encoding: utf8x