MySQL Administration

by Terry Sergeant

Contents

1 Introduction

1.1 Installation

Some OS's automatically enable an anonymous account. This should be disabled. In Windows the my.ini contains initialization files There are example databases provided for experimenting, etc.

1.2 First Steps

When you first install MySQL the root account is accessible without a password and there are test databases, etc. These commands will allow you to install and configure your account:

```
-- From Linux command-line:
sudo install mysql-server mysql
sudo mysql_secure_installation
mysql -u root

-- now we are in the MySQL client:
SELECT user, host, authentication_string, plugin FROM mysql.use
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_pass
FLUSH PRIVILEGES;
\q

-- Back at Linux command-line ... test new password
mysql -u root -p
```

1.3 Some Other Useful Commands

From Linux command-line:

```
systemctl status mysql
systemctl restart mysql
systemctl enable mysql
systemctl is-enabled mysql
```

From MySQL command-line:

```
show databases;
use DBNAME;
show tables;
desc TNAME;
create database DBNAME;
source SQLFILE; -- same as \. SQLFILE
select @@datadir;
```

2 Server Configuration

Here are some common ways to tweak the server's configuration.

- `mysqld --verbose --help` (to see runtime options supported by server)
- typically adjust configuration through startup file (rather than command-line options)
- options files are organized in to [groups]
- connection parameters are typcially given in [client] group
- `/etc/mysql/my.cnf` is global file; `~/.my.cnf` is local

```
[mysqld]
log                             # normal logging
log-bin                         # binary logging
log-slow-queries = /var/log/slow.log # slow query log
default-storage-engine=InnoDB  # set default storage engine
max_connections=20
key_buffersize-128M                    # default is 8MB
```

2.1 SQL Mode

Can change mode to improve SQL standards compliance or other characteristics.

```
[mysqld]
sql-mode=IGNORE_SPACES  # after function name
SELECT @@sql_mode;
```

Common modes are: ANSI-QUOTES,IGNORE-SPACE, ERROR_FOR_DIVISION_BY_ZERO, STRICT_TRANS_TABLES, STRICT_ALL_TABLES,TRADITIONAL,ANSI

2.2 Logging

There are several types of logs that can be kept:

error log for errors

general query log shows all querys that have been performed

slow query log shows querys that take a long time

binary log keeps results of all statements that modify data

Common use cases:

- use binary and slow to debug then disable them
- error log can stay on
- startup data is put in error log
- If replicating servers, MySQL uses log shipping of the binary log.

To delete old log files do one of these:

```
SET GLOBAL expire_logs_days=7;
PURGE BINARY LOGS BEFORE now() - INTERVAL 1 WEEK;
```

To view binary log data:

```
mysqlbinlog logfile
SHOW BINARY LOGS;
```

Some helpful options for use with `mysqlbinlog`:

- `--start-datetime` and `--stop-datetime` can be used to display log data within a specified range; these can be used separately or together
- `--start-position` and `--stop-position` can be used to filter log data by position numbers
- NOTE: results of `mysqlbinlog` can be piped to `more` or to `mysql -u root -p`

3 MySQL Clients

```
mysql      # command-line client
mysqladmin # for managing server
mysqlimport #
mysqldump   # for backups, etc.
```

The above command can be invoked with –help to get information about various options.

Helpful facts about the `mysql` client:

- can be used in interactive mode or batch mode
- use `SELECT VERSION();` for version info
- terminate commands with `;` or `\G` (`\g` for vertical format)
- prompt will change depending on what else is expected
- can do input line editing (use up arrow)
- `source scriptname` is used to execute an SQL file
- commands have long form (word) and short form (`\char`)
- `tee file.txt` causes queries to be logged to a file

4 Data Types

4.1 Numeric

```
INT
FLOAT,DOUBLE
DECIMAL(10,2)
BIT(4) --stores 4 bits
```

4.2 Character

```
CHAR
VARCHAR
TEXT
ENUM
SET
SHOW CHARACTER SET;
```

4.3 Binary

```
BINARY(L)
VARBINARY(L)
for holding images, sound files, etc.
BLOG --binary large object
```

4.4 Temporal

```
TIME -- of day or INTERVAL
YEAR --
DATE -- calendar date
DATETIME -- both date and time
TIMESTAMP -- ditto
...DEFAULT CURRENT_TIMESTAMP...ON UPDATE CURRENT_TIMESTAMP
```

4.5 Other

The value `NULL` means "unknown". It cannot be applied to primary keys or to timestamps.

The type `AUTOINCREMENT` generates successive values on inserts only one column per table must be indexed and not NULL.

The query `SELECT LAST_INSERT_ID()` returns last autoincrement value for the current client.

You can use `DEFAULT` to specify default values for a field. The specified value must be a constant with the allowed exception of `TIMESTAMP` fields which can make use of functions that return time stamps.

5 Obtaining Metadata

Metadata describes structure of data (e.g., names and structure of tables, etc.). There are various commands that can show this information:

```
INFORMATION_SCHEMA database
SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES \
  WHERE TABLE_SCHEMA= 'information_schema' ORDER BY TABLE_NAME;
```

The SHOW command is MySQL-only but provides shortcuts to accessing the `INFORMATION_SCHEMA` database directly.

```
SHOW DATABASES; -- LIKE ...
SHOW TABLES; -- FROM database;
SHOW COLUMNS FROM table; -- WHERE ...
DESCRIBE table;
```

From the command-line you can use `mysqlshow` to provide metadata.

6 Storage Engines

Storage engines are independent of SQL commands. Storage engine determines storage medium, transactional capabilities, locking, backup and recovery, and optimization

Use `SHOW ENGINES;` to see what engines are supported by your installation. Add `ENGINE=engine` to a `CREATE TABLE` statement if you want something other than the default engine.

Use `SHOW TABLE STATUS tablename;` to see engine for a given table.

6.1 MyISAM (default)

The MyISAM engine is the default. Some characteristics of MyISAM tables:

- each table has 3 files (.frm, .MYD, .MYI)
- flexibile `auto_increment`
- can be compressed and read-only
- table-level locking
- portable storage format
- tables take up litte space
- fixed row format / dynamic row format / compressed format
- to compress a table use myisampack
- concurrent behavior: writer preference
- NO SUPPORT FOR TRANSACTIONS!
- NO SUPPORT FOR FOREIGN KEY CONSTRAINTS!

6.2 InnoDB

The InnoDB engine provides support for transactions and is, therefore, the common choice for concurrent web applications.

- represented by .frm file and other files growing as needed
- supports transactions (ACID compliant)
- auto-recovery after crash
- row-level locking
- foreign key constraints are maintained if requested
- uses graph detection for deadlock

6.3 InnoBase

InnoBase is a plugin that provides advantages over InnoDB.

6.4 MEMORY

Used for tables stored in memory. The result is that access to the tables is very fast. Also the tables are temporary.

6.5 Other Engines

ARCHIVE for compression ... but slower

CSV just what it sounds like! stored in `table_name.csv`

BLACKHOLE does't store data, but can hold structure

MySQL Cluster group of nodes working together to store/retrieve data

Falcon developed by Sun. special release. for high-traffic, transactional apps

7 Security and User Management

MySQL restricts access via ACLs (Access Control Lists).

7.1 Risks

Some risks to be aware of:

- network security risks
- operating system security risks
- filesystem security risks

7.2 Some Commands

Grant tables store access control info:

- `user` table stores privileges
- `db` db-level privileges

An account contains both a host name and a user name

```
FLUSH PRIVILEGES; -- forces reload of modifications
GRANT SELECT world.* TO 'kari'@'localhost' \
    IDENTIFIED BY 'password';
GRANT whichprivs whichdbs TO user@host \
    IDENTIFIED BY 'password';
SHOW GRANTS; -- FOR CURRENT_USER();
SHOW GRANTS FOR 'kari'@'myhost.example.com';
SELECT user, host, password FROM mysql.user; -- WHERE ...
CREATE USER 'jim'@'localhost' IDENTIFIED BY 'password';
DROP USER 'jim'@'localhost';
REVOKE whichprivs ON whichdbs FROM user;
```

8 Backup and Recovery

8.1 Backup Types

There are three categories of backups:

HOT happen while server is actively being used

COLD happen when user's are not allowed to access server

WARM allows data to be read, but not modified during backup

There are several ways to backup a MySQL database:

disk : Copy actual files on disk. This is a COLD procedure and provides for a fast restore. Can only restore to same engine. Example steps for disk backup on InnoDB tables:

1. Stop Server
2. Confirm proper shutdown
3.
```
cp /var/lib/mysql/data/ib_logfile* /var/backup
cp /var/lib/mysql/data/ibdata* /var/backup
cp /var/lib/mysql/data/*.frm /var/backup
cp /etc/my.cnf /var/backup/
```
4. Restart server

binary log : Records modifications to data; is HOT; provides sequential/incremental (not total) backup. Use `SET SQL_LOG_BIN = ON` to turn on binary logging.

logical/textual : Use the `mysqldump` command to dump database to a file. This is a WARM technique:

```
mysqldump --opt mydbname -u root \
        -p > mydb_backup.sql
mysqldump --opt mydbname --table=tablename \
        -u root -p > mytable_backup.sql
mysqldump --opt --all-databases -u root -p > \
        alldb_backup.sql
```

Some common options for use with `mysqldump`:

- `--opt` shorthand for a variety of commonly needed options
- `--flush-logs` will start a new set of binary logs after the backup is made
- `--master-data` will write the binary log file name and position to the dump file; NOTE: `--master-data=2` will also name binary log that that was started after the backup
- `--all-databases` will copy everything
- `--lock-all-tables` will make all tables across all databases read-only during the backup
- `--single-transaction` in an database using InnoDB engine, this provides a consistent snapshot while allowing some concurrent write activity; if not using InnoDB then use `--lock-all-tables` instead; do not use this option if there is a possibility of table structure being modified during the backup
- `--routines` include stored procedures/functions in the backup

MySQL replication : Multiple database hosts keep current copy of the database in a master/slave arrangment that automatically backs up master to slave. Queries can be distributed among the hosts and backups are implicit.

8.2 Recovery

Recovery methods, of course, depend on the type of backup used:

disk : Simply move the copied files back to the original location from which they were copied.

binary log : Re-execute changes in binary log since last backup. Use the command `mysqlbinlog` to extract portions of a file. e.g., `mysqlbinlog bin.000050 bin000051 | mysql`

logical/textual : These are some ways to dump/restore files. Suppose we have a database named `world` and a table named `Country` saved in a file named `dump.sql`:

```
mysql -u root -p world < dump.sql
```

NOTE: if `--tab` was given then import via `mysqlimport`

NOTE: We can use `mysqldump` and `mysql` and the pipe to copy table(s) from one database to another:

```
mysqldump world Country | mysql test
mysqldump world Country | mysql -h other.host.com world
```

MySQL replication : Database is always replicated.