MySQL client MySQL shell and dump

START HERE

`mysql -u <user> -p <db>`
Connect and pick a database

`mysqldump <db> > <file.sql>`
Back up a database

`SHOW DATABASES;`
List databases

`\q`
Quit

`USE <db>;`
Switch database

`SHOW TABLES;`
List tables

`DESCRIBE <table>;`
Show columns

`\G`
Print a result vertically

CONCEPTS

A statement runs only at its terminator: ; or \G. Backslash commands like \q and \s take effect at once and need no semicolon.

Server host:3306
Database USE <db>
Table rows, columns, indexes

Account names are 'user'@'host': the same user name from another host is a different account with its own password and grants.

Passing -p<password> leaks it to ps and shell history. Use bare -p to be prompted, or a login path from mysql_config_editor.

CONNECT

`mysql -u <user> -p`
Connect to localhost, prompt for password

`mysql -u <user> -p -h <host> -P <port> <db>`
Connect to a remote server and database

`mysql -e "<sql>" <db>`
Run a statement and exit

`mysql -S <socket>`
Connect through a Unix socket

CONNECT (cont.)

`mysql --protocol=TCP -h 127.0.0.1`
Force TCP instead of the socket
With host localhost the client uses the socket and ignores -P; use 127.0.0.1 or --protocol=TCP.

`mysql_config_editor set --login-path=<name> -u <user> -h <host> -p`
Store credentials in ~/.mylogin.cnf

`mysql --login-path=<name>`
Connect with a stored login path

`mysql --safe-updates`
Refuse UPDATE/DELETE without key or LIMIT

SHELL COMMANDS

`\s`
Show connection and server status

`\u <db>`
Switch database

`\G`
Run and print vertically

`\c`
Cancel the statement being typed

`\q`
Quit

`\. <file.sql>`
Run SQL from a file

`\e`
Edit the statement in $EDITOR

SHELL COMMANDS (cont.)

`\P <pager>`
Pipe output through a pager

`DELIMITER //`
Change the statement terminator
Needed when defining stored routines whose body contains semicolons; reset with DELIMITER ;

INSPECT

`SHOW DATABASES;`
List databases

`SHOW TABLES;`
List tables in the current database

`DESCRIBE <table>;`
Show columns, types and keys

`SHOW CREATE TABLE <table>\G`
Show the full DDL

`SHOW INDEX FROM <table>;`
List indexes

`SHOW TABLE STATUS\G`
Engine, row estimate and size per table

`SHOW FULL PROCESSLIST;`
See running queries

`SHOW VARIABLES LIKE '<pattern>';`
Look up server variables

`SELECT VERSION(), DATABASE(), CURRENT_USER();`
Server version, schema, account

QUERY

`SELECT * FROM <table> LIMIT 10;`
Peek at rows

`SELECT COUNT(*) FROM <table>;`
Count rows

`EXPLAIN SELECT ...;`
Show the query plan

`EXPLAIN ANALYZE SELECT ...;`
Run and time each plan step
Runs the query for real; available since MySQL 8.0.18 and MariaDB uses ANALYZE SELECT instead.

`EXPLAIN FORMAT=TREE SELECT ...;`
Plan as an indented tree

`START TRANSACTION;`
Begin a transaction

`ROLLBACK;`
Undo the open transaction

`COMMIT;`
Make the transaction permanent

`CREATE INDEX <idx> ON <table> (<col>);`
Add an index

`ANALYZE TABLE <table>;`
Refresh index statistics

EXPLAIN type column, best first

type	Meaning
`const`	one row via key
`ref`	index lookup
`range`	index range scan
`index`	full index scan
`ALL`	full table scan

USERS & GRANTS

`CREATE USER '<user>'@'<host>' IDENTIFIED BY '<pw>';`
Create an account

`GRANT ALL ON <db>.* TO '<user>'@'<host>';`
Grant on one database

`GRANT SELECT, INSERT ON <db>.<table> TO '<user>'@'<host>';`
Grant specific privileges

`SHOW GRANTS FOR '<user>'@'<host>';`
List an account's privileges

`REVOKE <priv> ON <db>.* FROM '<user>'@'<host>';`
Remove privileges

`ALTER USER '<user>'@'<host>' IDENTIFIED BY '<pw>';`
Change a password

`DROP USER '<user>'@'<host>';`
Delete an account

`SELECT user, host FROM mysql.user;`
List accounts

`FLUSH PRIVILEGES;`
Reload grant tables
Not needed after GRANT, REVOKE or CREATE USER; only after editing the mysql. tables directly.*

OUTPUT

`mysql -B -N -e "<sql>"`
Tab-separated rows, no header

OUTPUT (cont.)

`mysql -E -e "<sql>"`
Vertical output

DUMP & RESTORE

`mysqldump <db> > <file.sql>`
Dump one database

`mysqldump --single-transaction <db> > <file.sql>`
Consistent InnoDB dump without locks

`mysqldump --databases <db1> <db2>`
Dump several, with CREATE DATABASE

`mysqldump --all-databases`
Dump everything

`mysqldump <db> <table1> <table2>`
Dump chosen tables

`mysqldump --no-data <db>`
Schema only

`mysqldump --routines --triggers --events <db>`
Include stored code
Triggers are on by default; routines and events are not.

`mysql <db> < <file.sql>`
Restore a dump

`mysqldump <db> | gzip > <file.sql.gz>`
Compressed dump

`gunzip -c <file.sql.gz> | mysql <db>`
Restore a compressed dump