hofmannsven / README.md

Last active 5 hours ago • Report gist

My simply MySQL Command Line Cheatsheet

README.md

MySQL

Getting started:

- http://www.sqlteaching.com/
- https://www.codecademy.com/courses/learn-sql

Related tutorials:

- MySQL-CLI
- Analyzing Business Metrics
- SQL joins infografic

Tools:

- DataGrip
- Sequel Pro

Commands

Access monitor: `mysql -u [username] -p;` (will prompt for password)

Show all databases: `show databases;`

Access database: `mysql -u [username] -p [database]` (will prompt for password)

Create new database: `create database [database];`

Select database: `use [database];`

Determine what database is in use: `select database();`

Show all tables: `show tables;`

Show table structure: `describe [table];`

List all indexes on a table: `show index from [table];`

Create new table with columns: `CREATE TABLE [table] ([column] VARCHAR(120), [another-column] DATETIME);`

Adding a column: `ALTER TABLE [table] ADD COLUMN [column] VARCHAR(120);`

Adding a column with an unique, auto-incrementing ID: `ALTER TABLE [table] ADD COLUMN [column] int NOT NULL AUTO_INCREMENT PRIMARY KEY;`

Inserting a record: `INSERT INTO [table] ([column], [column]) VALUES ('[value]', [value]');`

MySQL function for datetime input: `NOW()`

Selecting records: `SELECT * FROM [table];`

Explain records: `EXPLAIN SELECT * FROM [table];`

Selecting parts of records: `SELECT [column], [another-column] FROM [table];`

Counting records: `SELECT COUNT([column]) FROM [table];`

Counting and selecting grouped records: `SELECT *, (SELECT COUNT([column]) FROM [table]) AS count FROM [table] GROUP BY [column];`

Selecting specific records: `SELECT * FROM [table] WHERE [column] = [value];` (Selectors: `<`, `>`, `!=`; combine multiple selectors with `AND`, `OR`)

Select records containing `[value]`: `SELECT * FROM [table] WHERE [column] LIKE '%[value]%';`

Select records starting with `[value]`: `SELECT * FROM [table] WHERE [column] LIKE '[value]%';`

Select records starting with `val` and ending with `ue`: `SELECT * FROM [table] WHERE [column] LIKE '[val_ue]';`

Select a range: `SELECT * FROM [table] WHERE [column] BETWEEN [value1] and [value2];`

Select with custom order and only limit: `SELECT * FROM [table] WHERE [column] ORDER BY [column] ASC LIMIT [value];` (Order: `DESC`, `ASC`)

Updating records: `UPDATE [table] SET [column] = '[updated-value]' WHERE [column] = [value];`

Deleting records: `DELETE FROM [table] WHERE [column] = [value];`

Delete *all records* from a table (without dropping the table itself): `DELETE FROM [table];` (This also resets the incrementing counter for auto generated columns like an id column.)

Delete all records in a table: `truncate table [table];`

Removing table columns: `ALTER TABLE [table] DROP COLUMN [column];`

Deleting tables: `DROP TABLE [table];`

Deleting databases: `DROP DATABASE [database];`

Custom column output names: `SELECT [column] AS [custom-column] FROM [table];`

Export a database dump (more info here): `mysqldump -u [username] -p [database] > db_backup.sql`

Use `--lock-tables=false` option for locked tables (more info here).

Import a database dump (more info here): `mysql -u [username] -p -h localhost [database] < db_backup.sql`

Logout: `exit;`

Aggregate functions

Select but without duplicates: `SELECT distinct name, email, acception FROM owners WHERE acception = 1 AND date >= 2015-01-01 00:00:00`

Calculate total number of records: `SELECT SUM([column]) FROM [table];`

Count total number of `[column]` and group by `[category-column]`: `SELECT [category-column], SUM([column]) FROM [table] GROUP BY [category-column];`

Get largest value in `[column]`: `SELECT MAX([column]) FROM [table];`

Get smallest value: `SELECT MIN([column]) FROM [table];`

Get average value: `SELECT AVG([column]) FROM [table];`

Get rounded average value and group by `[category-column]` : `SELECT [category-column], ROUND(AVG([column]), 2) FROM [table] GROUP BY [category-column];`

Multiple tables

Select from multiple tables: `SELECT [table1].[column], [table1].[another-column], [table2].[column] FROM [table1], [table2];`

Combine rows from different tables: `SELECT * FROM [table1] INNER JOIN [table2] ON [table1].[column] = [table2].[column];`

Combine rows from different tables but do not require the join condition: `SELECT * FROM [table1] LEFT OUTER JOIN [table2] ON [table1].[column] = [table2].[column];` (The left table is the first table that appears in the statement.)

Rename column or table using an *alias*: `SELECT [table1].[column] AS '[value]', [table2].[column] AS '[value]' FROM [table1], [table2];`

Users functions

List all users: `SELECT User,Host FROM mysql.user;`

Create new user: `CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';`

Grant `ALL` access to user for `*` tables: `GRANT ALL ON database.* TO 'user'@'localhost';`

Find out the IP Address of the Mysql Host

`SHOW VARIABLES WHERE Variable_name = 'hostname';` (source)

bash_profile

```
1 alias mysql=/Applications/MAMP/Library/bin/mysql
```

my.cnf

```
1 [mysqld]
2 max_allowed_packet=64M
```

franz-josef-kaiser commented on 8 Nov 2014

Delete *all* records from a table (without dropping the table itself): `delete from [tablename]` . This also resets the incrementing counter for auto generated columns like an `id` column.

hofmannsven commented on 9 Nov 2014 Owner

Thanks! I've added it above and also shared it on WPSE :)

anjanb commented on 24 Jul 2015

Grunt ALL
Should that not be "GRANT ALL" ?