

SQLite

I like to use [sqlite](#) in most of my projects that require simple/small databases. It has a [C/C++ interface library](#), very simple to use (because no server required), contained in a single file and even [available on Android](#)! So, here are my notes on using them... mainly for my own personal reference.

Main documentation can be found [here](#).

Basic Commands

Most SQL commands in SQLite are similar to other SQL found in major database software (e.g. [MariaDB](#), [PostgreSQL](#), [MySQL](#)). So, if you have done SQL before, this should be very familiar.

To create a new table:

```
CREATE TABLE IF NOT EXISTS tablename (col-name col-TYPE [,col-name col-TYPE]);
```

Notice that IF NOT EXISTS is optional.

to be continued...

Modify rows with decimal points

Assuming we want to modify rows (with column *col* that contains decimal points) in table *tablename*:

```
UPDATE tablename SET col=substr(col,1,LENGTH(col)-3) WHERE col LIKE '%.00';
```

Common DB Search

Selecting specific column (operations) within specific time range:

```
SELECT col1,(col2*2.0),datetime(col3,'8 hours') FROM table1  
WHERE id=1 AND datetime(col3,'8 hours')<datetime('2016-10-15T23:59:59');
```

Multiple DB Files

One thing to note is that there is no database concept - a file IS a database. But we can attach another database for simultaneous access.

First open a DB file:

```
sqlite3 'file1.sqlite'
```

Within sqlite, this file will be known as the main database. There is another system database called temp (so you should not use this name as an alias for the other database).

So, to access another database:

```
ATTACH 'file2.sqlite' AS NEXT;
```

You can now access any table like:

```
SELECT * FROM NEXT.tablename;
```

To copy from/to the same table (exactly same schema) between both database:

```
INSERT INTO main.tablename SELECT * FROM NEXT.tablename;
```

Or to copy selective columns:

```
INSERT INTO main.tablename(column1, column2) SELECT (column3, column4) FROM  
NEXT.tablename;
```

MariaDB/MySQL

Initial Setup

start installation

```
# mysql_install_db --user=mysql
```

if forgot to supply user, make sure permission for /var/lib/mysql is given to user mysql

```
# chown -R mysql:mysql /var/lib/mysql
```

start daemon

```
# sh /etc/rc.d/rc.mysqld start
```

as the name implies, run

```
# /usr/bin/mysql_secure_installation
```

Common Commands

create database

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

```
mysql> CREATE DATABASE app_db;  
mysql> GRANT ALL privileges ON app_db.* TO 'user_app'@'localhost' IDENTIFIED  
BY 'pass_app';  
mysql> SET password FOR 'user_app'@'%' = password('pass_app');  
mysql> FLUSH privileges;
```

note not sure if i need that set password line... will test that

recover root password

```
$ /etc/rc.d/rc.mysql stop  
$ mysqld_safe --skip-grant-tables &  
$ mysql -u root
```

```
mysql> USE mysql;  
mysql> UPDATE USER SET password=PASSWORD('<newpass>') WHERE USER='root';  
mysql> FLUSH privileges;  
mysql> exit
```

show permission / grants

```
$ mysql -u root -p -e "show grants for 'user_app'@'localhost';"
```

delete database

```
$ mysql -u root -p -e "drop database app_db;"
```

dumping this here for now.... not sure why i have this in my note



```
select db,user,select_priv from mysql.db where db='mylvmsdb' \g
```

From:

<https://www.azman.my1matrix.cc/> - Azman @MY1

Permanent link:

<https://www.azman.my1matrix.cc/doku.php?id=notes:dbase>

Last update: **2026/02/08 03:46**

