SQLite3 Cheat Sheet

SQLite is a public domain C-language library implementing a small, fast, self-contained, reliabile, and full-featured, SQL database engine.

Manipulating data

Create database	`> .open example.db;`
Create table and define fields	`> CREATE TABLE IF NOT EXISTS mytable (` `→ foo TEXT NOT NULL);`
View tables in database	`> .tables`
Insert data into a table	`> INSERT INTO mytable (foo)` `→ VALUES ('aaa'), ('bbb'),('ccc');`
View table schema	`> .schema mytable`
Add a new column to **mytable**	`> ALTER TABLE mytable ADD bar INTEGER;`
Update data in a table	`> UPDATE mytable SET bar=123` `→ WHERE foo='aaa';`

Joins

Display an inner join	`> SELECT * FROM mytable`
	`→ INNER JOIN othertable`
	`→ ON mytable.rowid=othertable.foo;`

Display a left join	`> SELECT * FROM mytable LEFT JOIN` `→ ON mytable.id=othertable.foo;`
Display a cross join	`> SELECT * FROM mytable` `→ CROSS JOIN othertable;`

Data types

TEXT	Text data
INTEGER	Whole number
REAL	Fleating point number
BLOB	Binary data
NULL	Null value

Some SQLite functions

`abs()`	Absolute value
`max() min()`	Maximum and minimum values
`upper() lower()`	Convert case of string
`length()`	Length of string
`random()`	(Pseudo) random integer

SQLite3 Cheat Sheet

Select

Display all data	`> SELECT * FROM mytable;`
Display data of the third row	`> SELECT * FROM mytable` `→ WHERE rowid=3;`
Display **foo** and **bar** columns	`> SELECT foo,bar FROM mytable;`
Display first 10 results	`> SELECT * FROM mytable LIMIT 10;`
Sort by column **foo**	`> SELECT * FROM mytable ORDER BY foo;`

Views

A view is a virtual table providing a template for displaying the results of a specific query.

Create a new view	`> CREATE VIEW myview AS`
	`→ SELECT foo FROM mytable`
	`→ WHERE example > 10;`

Show existing views	`> .tables`
Display data with a view	`> SELECT * FROM myview;`
Delete (*drop*) a view	`> DROP VIEW myview;`

Column constraints

Set default text for a field	`DEFAULT 'default text'`
Enforce unique value	`UNIQUE`

Designate a column as a unique identifier	`PRIMARY KEY`
	`> CREATE TABLE mytable`
	`→ (Id INTEGER PRIMARY KEY);`

Pointer to a primary key of a different table	`FOREIGN KEY`

Impose a condition for validation	`CHECK`
	`> CREATE TABLE mytable`
	`→ (CHECK(condition>0), bar TEXT);`

Prevent NULL values	`NOT NULL`

