SQLite Cheat Sheet

Managing Tables

Create a new table:

```
CREATE TABLE [IF NOT EXISTS] table(
    primary_key INTEGER PRIMARY KEY AUTOINCREMENT,
    column_name type NOT NULL,
    column_name type NULL,
    ...
    datestamp DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

Datatypes in SQLite:

```
- TEXT: a text string
- INTEGER: a whole number
- REAL: a floating point (unlimited decimal places) number
- BLOB: binary data (for instance, a .jpeg or .webp image)
- NULL: a null value
```

The SQL to create this field is:

Rename a table:

```
ALTER TABLE table_name RENAME TO new_name;
```

Add a new column to a table:

```
ALTER TABLE table ADD COLUMN column_definition;
```

Drop an existing column in a table:

```
ALTER TABLE table DROP COLUMN column_name;
```

Drop a table and its data:

```
DROP TABLE [IF EXISTS] table_name;
```

Managing indexes

Creating an index

```
CREATE [UNIQUE] INDEX index_name
ON table_name (c1,c2,...);
```

Delete an index:

```
DROP INDEX index_name;
```

Create an expression index:

```
CREATE INDEX index_name ON table_name(expression);
```

Querying Data

Query all data from a table

```
SELECT * FROM table_name;
```

Query data from the specified column of a table:

```
SELECT c1, c2
FROM table_name;
```

Query unique rows

```
SELECT DISTINCT (c1)
FROM table_name;
```

Query rows that match a condition using a WHERE clause.

```
SELECT *
FROM table_name
WHERE condition;
```

Rename column in the query's output:

```
SELECT c1 AS new_name
FROM table_name;
```

Query data from multiple tables using inner join, left join

```
SELECT *
FROM table_name_1
INNER JOIN table_name_2 ON condition;
```

```
SELECT *
FROM table_name_1
LEFT JOIN table_name_2 ON condition;
```

Count rows returned by a query:

```
SELECT COUNT (*)
FROM table_name;
```

Sort rows using ORDER BY clause:

```
SELECT c1, c2
FROM table_name
ORDER BY c1 ASC [DESC], c2 ASC [DESC],...;
```

Group rows using GROUP BY clause.

```
SELECT *
FROM table_name
GROUP BY c1, c2, ...;
```

Filter group of rows using HAVING clause.

```
SELECT c1, aggregate(c2)
FROM table_name
GROUP BY c1
HAVING condition;
```

Changing Data

Insert a row into a table:

```
INSERT INTO table_name(column1,column2,...)
VALUES(value_1,value_2,...);
```

Insert multiple rows into a table in a single statement:

```
INSERT INTO table_name(column1,column2,...)
VALUES(value_1,value_2,...),
      (value_1,value_2,...),
      (value_1,value_2,...);
```

Update all rows in a table:

```
UPDATE table_name
SET c1 = v1,
    ...
```

Update rows that match with a condition:

```
UPDATE table_name
SET c1 = v1,
    ...
WHERE condition;
```

Delete all rows in a table

```
DELETE FROM table;
```

Delete rows specified by a condition:

```
DELETE FROM table
WHERE condition;
```

Search

Search using LIKE operator:

```
SELECT * FROM table
WHERE column LIKE '%value%';
```

Search using full-text search:

```
SELECT *
FROM table
WHERE table MATCH 'search_query';
```