SQLite Cheat Sheet

SQLite cheat sheet lists the most common SQLite statements that help you work with SQLite more quickly and effectively.

Managing databases

Attach another database to the current database connection:

```
ATTACH DATABASE file_name AS database_name;
```

Optimize the database:

```
VACUUM
```

Managing Tables

Create a new table:

```
CREATE TABLE [IF NOT EXISTS] table(
   primary_key INTEGER PRIMARY KEY,
   column_name type NOT NULL,
   column_name type NULL,
```

```
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

```
2 FROM table_name;
```

Query unique rows

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

Query rows that match a condition using a WHERE clause.

```
1 SELECT *
2 FROM table_name
3 WHERE condition;
```

Rename column in the query's output:

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

Query data from multiple tables using inner join, left join

```
1 SELECT *
2 FROM table_name_1
3 INNER JOIN table_name_2 ON condition;
```

```
1 SELECT *
2 FROM table_name_1
3 LEFT JOIN table_name_2 ON condition;
```

Count rows returned by a query:

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

Sort rows using ORDER 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,
```

```
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';
```