Basics

`sqlite filena me.d b`	Will create a new database

Create Tables

```
CREATE TABLE name (
  colname type constr aints,
  colname type constr aints
)
```

This will create a new table. For types and constraints look right.

Insert Data

```
INSERT INTO tablename (column2, column4)
VALUES
(data1, data2),
(data3, data4);
```

You don't have to fill every column of the table. It is often useful to leave out columns with primary keys for default value insertion.

Update Tables

```
UPDATE tablename
SET column5 = value5
WHERE expression
```

This will update the given column with the given value, where the expression evaluates true (usually specific primary key).

Alter Tables

The `ALTER TABLE` function is limited in SQLite. You can either rename the table with:

`ALTER TABLE tablename RENAME TO newtab lename`

Or add a new column with:

`ALTER TABLE tablename ADD COLUMN column def inti on`

It is not possible to change column definitions of existing tables.

Drop Table

```
DROP TABLE tablename
```

Select from Table

```
SELECT table1.co lumn1, table.2 co lumn2,
FROM table1, table2
WHERE expression
ORDER BY table1.co lumn1 order
```

The `ORDER BY` is optional, possible ways to order are `ASC` (ascending) and `DESC` (descending).

Data Types

`NULL`	null value
`INTEGER`	Integer value
`REAL`	8-byte floating point number
`TEXT`	Text with UTF encoding
`BLOB`	Blob-Data (stored like input)
`BOOLEAN`	Will be stored as Integer (0 for false, 1 for true)

Constraints

`PRIMARY KEY`	Sets column as primary key. The constraints `NOT NULL` and `UNIQUE` will be set automatically.
`NOT NULL`	Prevents the column from empty values (`null`).
`UNIQUE`	Each value in the column has to be unique.

`INTEGER PRIMARY KEY`s will automatically become the `ROWID` and therefore will be automatically incremented. This column can be accessed as `ROWID`, `OID`, `_ROWID_` or the column name.