SQlite Studio Manual

- Manpreet S. Katari (1/31/2018)

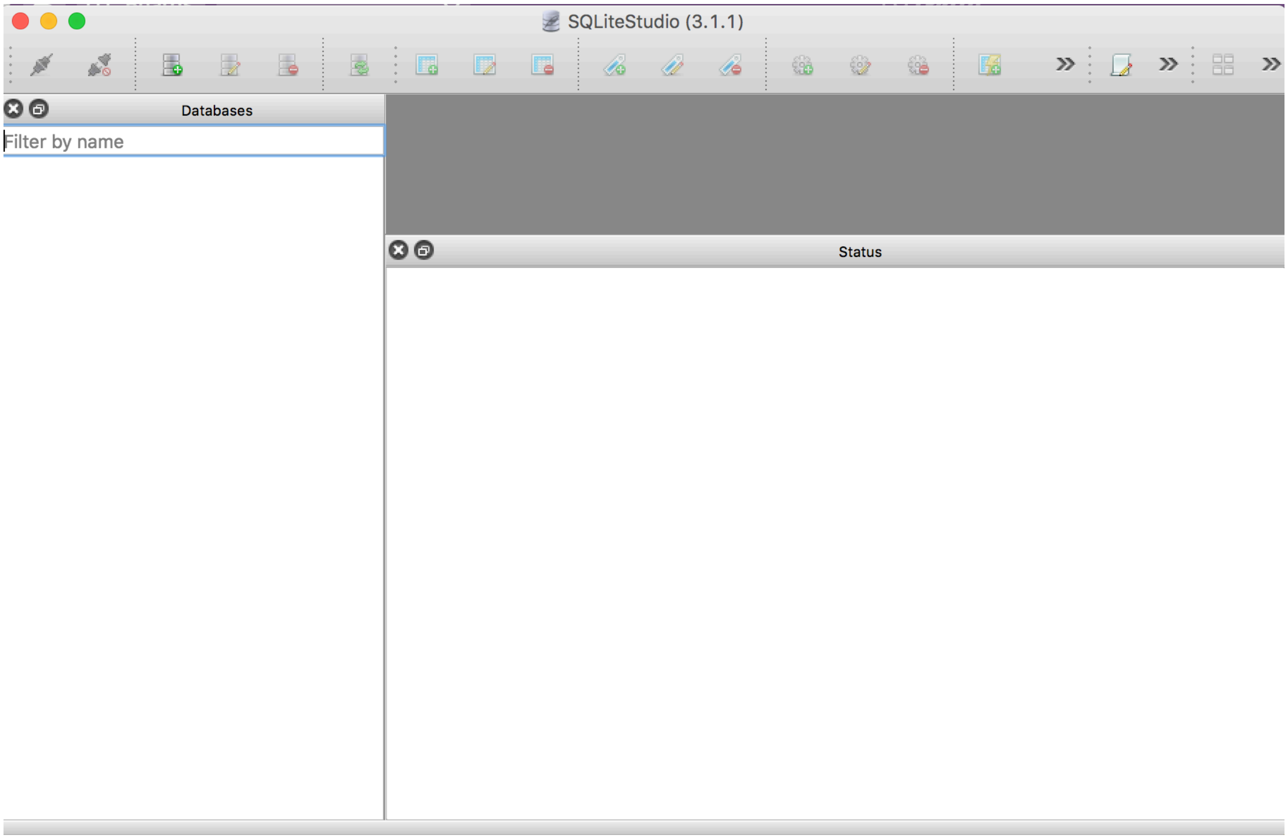
Downloading the Software

Sqlite Studio is free and works across all common platforms (MAC, Windows, and Linux). Go to the following link and download the appropriate installation.

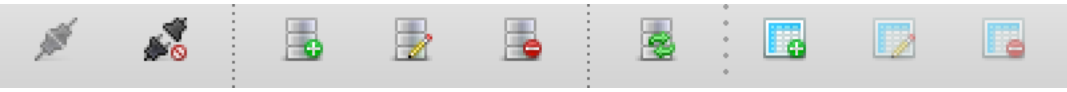
- https://sqlitestudio.pl/

Creating databases

When you first start Sqlite Studio, the application looks something like this:

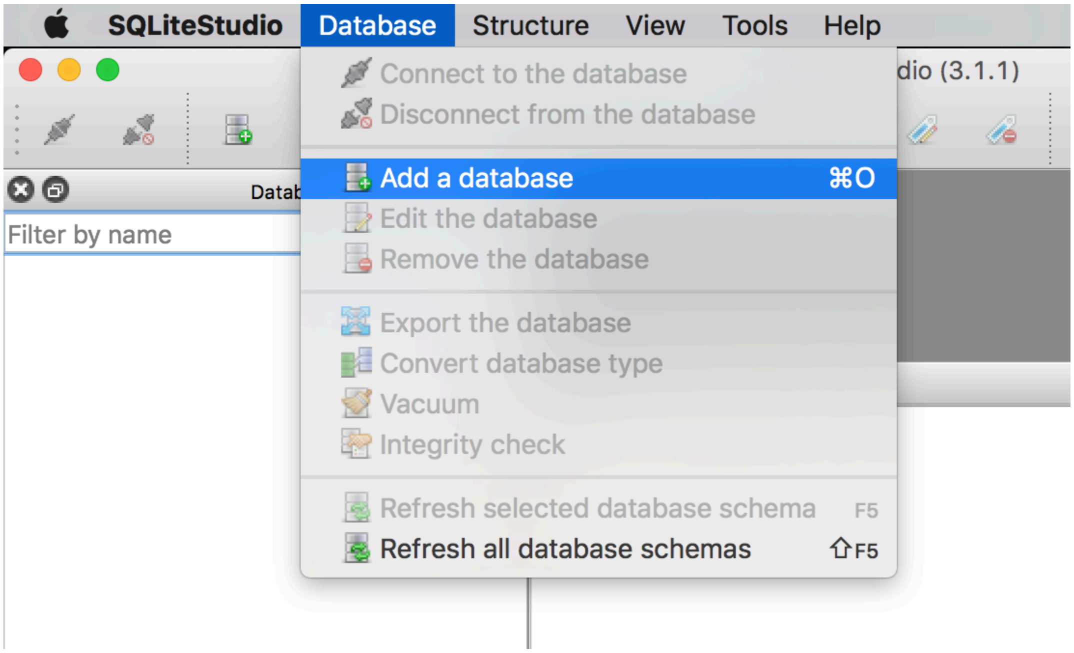


For creating a database, we need to focus on only a set of icons.

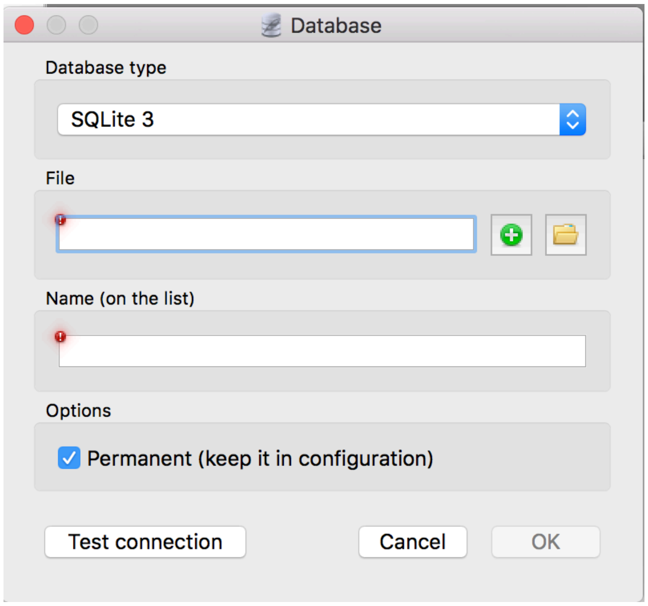


- The first one is to connect to the database that is selected
- The second is to disconnect from the database that is selected
- Followed by creating a new database
- Editing the database
- Removing the database
- Creating a table
- Editing the table
- Removing the table

The first thing we need to do is create a database to connect to. From the menu bar, select **Add a database**



A new window will popup that looks like this



Database

Database type: SQLite 3

File:

Name (on the list):

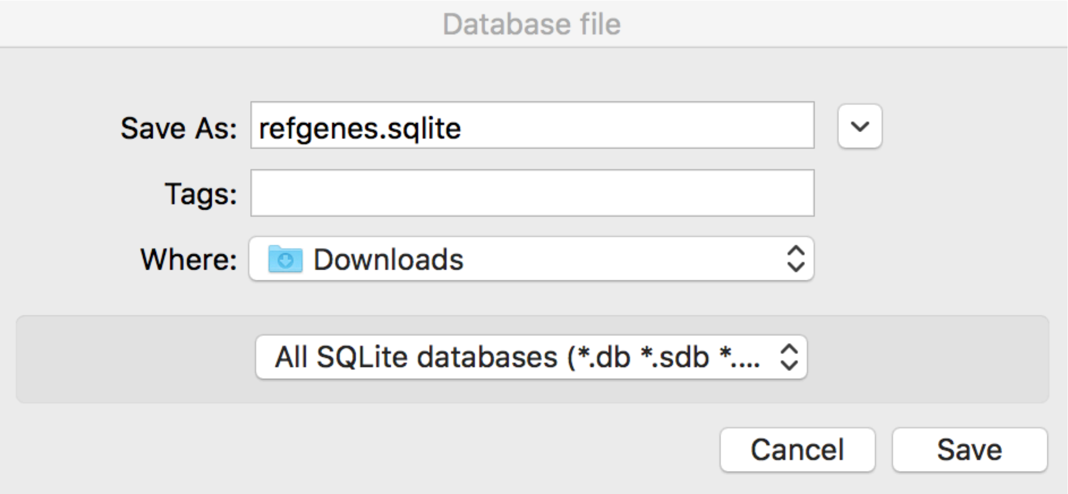
Options

☑ Permanent (keep it in configuration)

This will give you the option to either **Create a new database file** by using the *green +* sign, or **Browse for existing database file** using the *folder* icon.

If an **.sqlite** or **.db** file was provided to you then you need to simply browse to that database file and it will automatically open it.

For this exercise we are going to create a new database by importing a tab delimited file. So go ahead and click on the *green +* sign and you will prompted to provide the name like this:



Database file

Save As: refgenes.sqlite

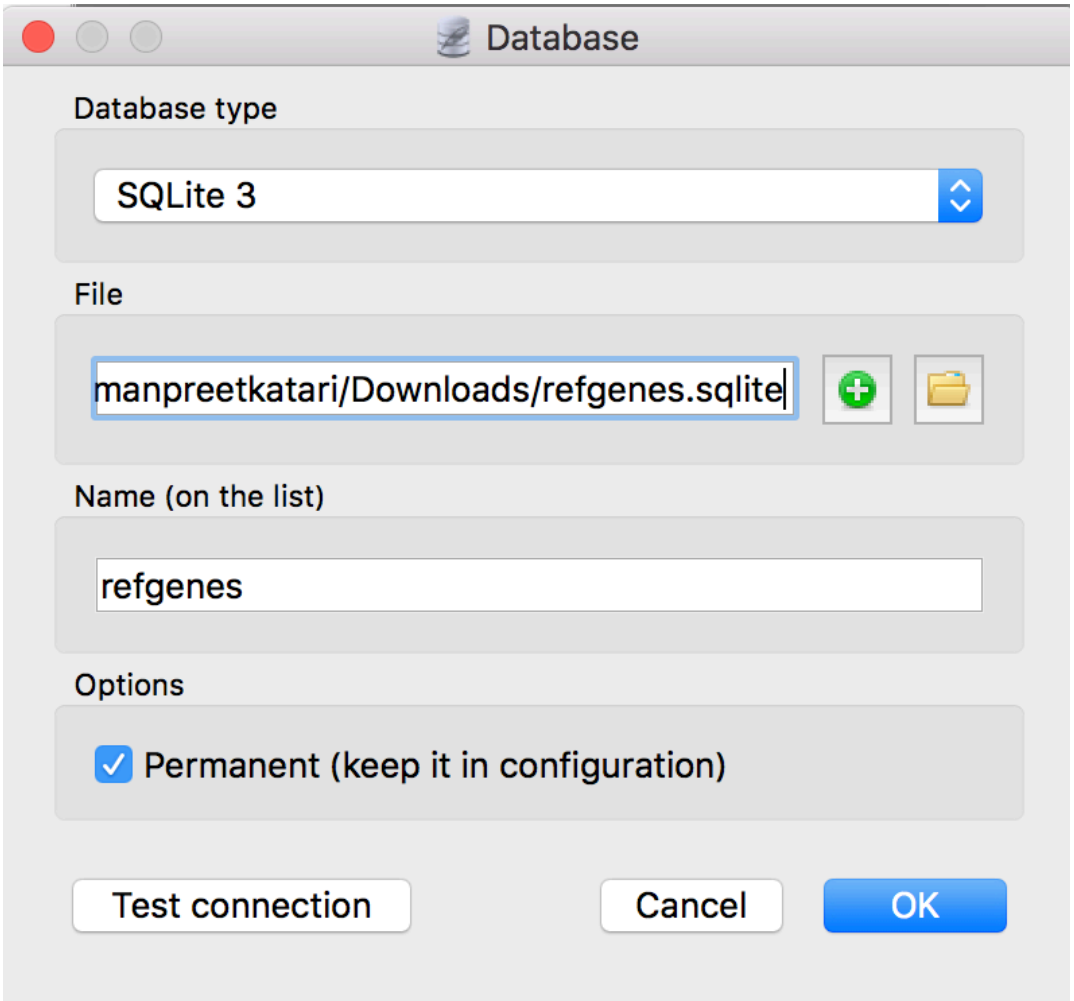
Tags:

Where: Downloads

All SQLite databases (*.db *.sdb *....

Let's call it **refgenes.sqlite** and click on *Save*.

The next window should look like this as a confirmation and then go ahead and click on *OK*



Database type

SQLite 3

File

manpreetkatari/Downloads/refgenes.sqlite

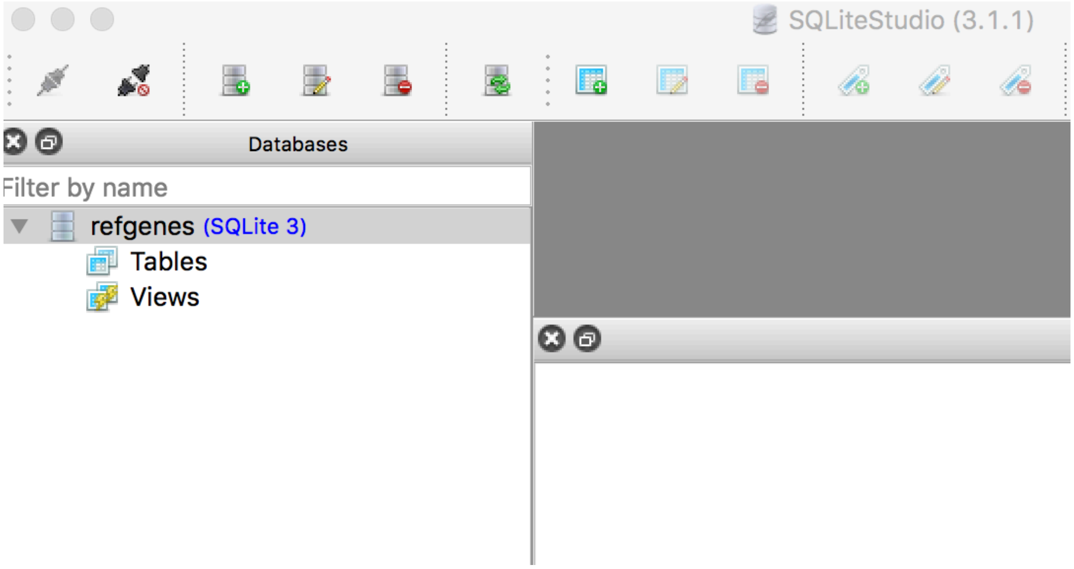
Name (on the list)

refgenes

Options

☑ Permanent (keep it in configuration)

Now you should be back to the original window with your database **refgenes** listed under databases. You are not actually connected to the database until you either **double click** on the database name or click on the **connect to the database** icon (this is the first one on the menu bar). Your database is connected when you can see **Tables** and **Views** under your database like this.



Loading the data

There are two different options to creating and loading tables into the database:

1) Create new tables from scratch

2) Import a file that already contains the data in the structure you need for your table.

Importing data file as a table

The easiest and simplest way to create and load a table is to simply load a file (preferrably a *csv* or *tab-delimited* file) .

First click on the **import** icon, which is four blue arrows pointing in.

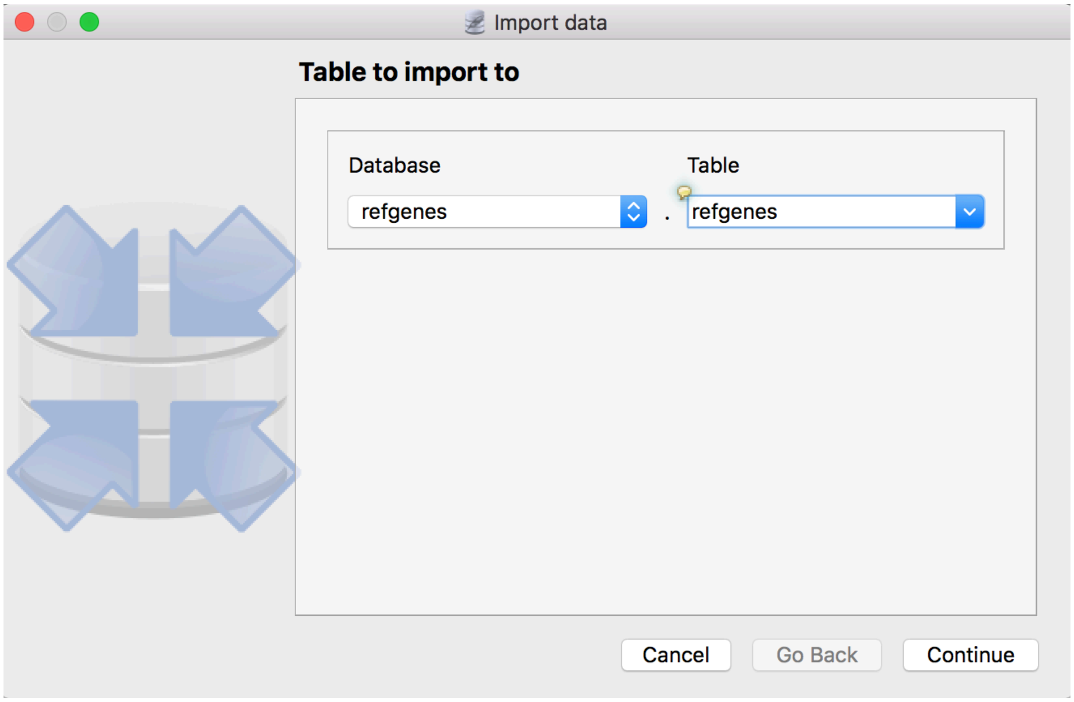
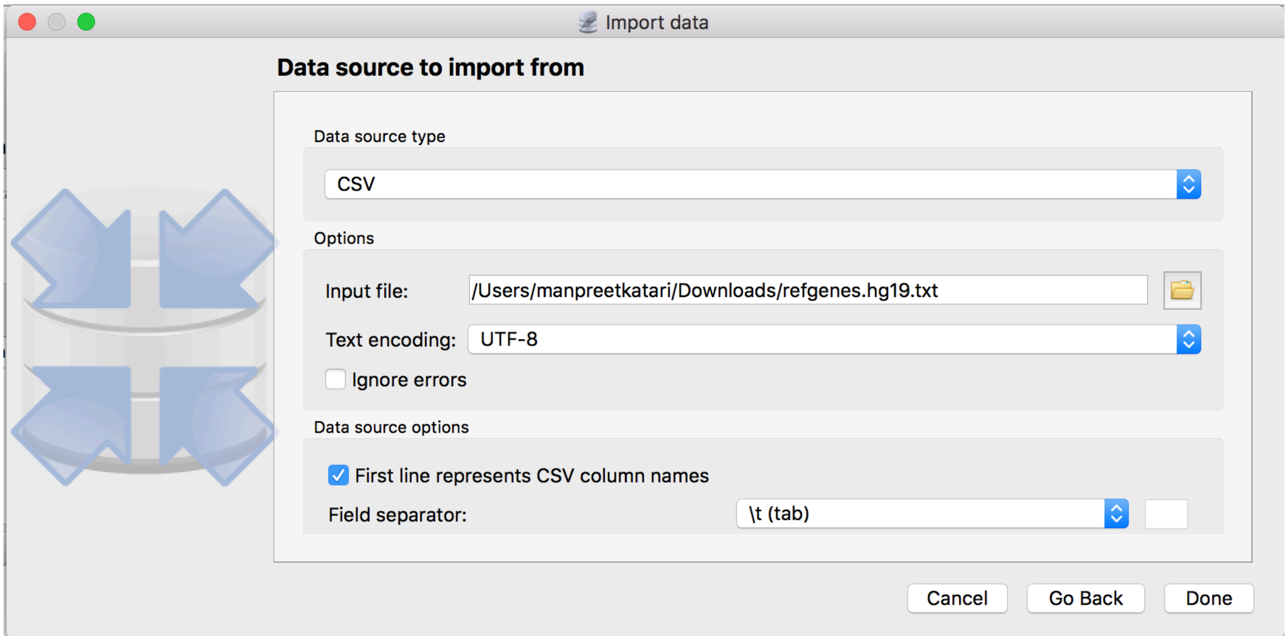


Table to import to

Database: refgenes . Table: refgenes

Under Table, give your table a nice name. In this case let's call it **refgenes** again. If a table with that name doesn't already exist, it create a new one. Select **Continue** to provide the name of the file.



Data source to import from

Data source type

CSV

Options

Input file: /Users/manpreetkatari/Downloads/refgenes.hg19.txt

Text encoding: UTF-8

☐ Ignore errors

Data source options

☑ First line represents CSV column names

Field separator: \t (tab)

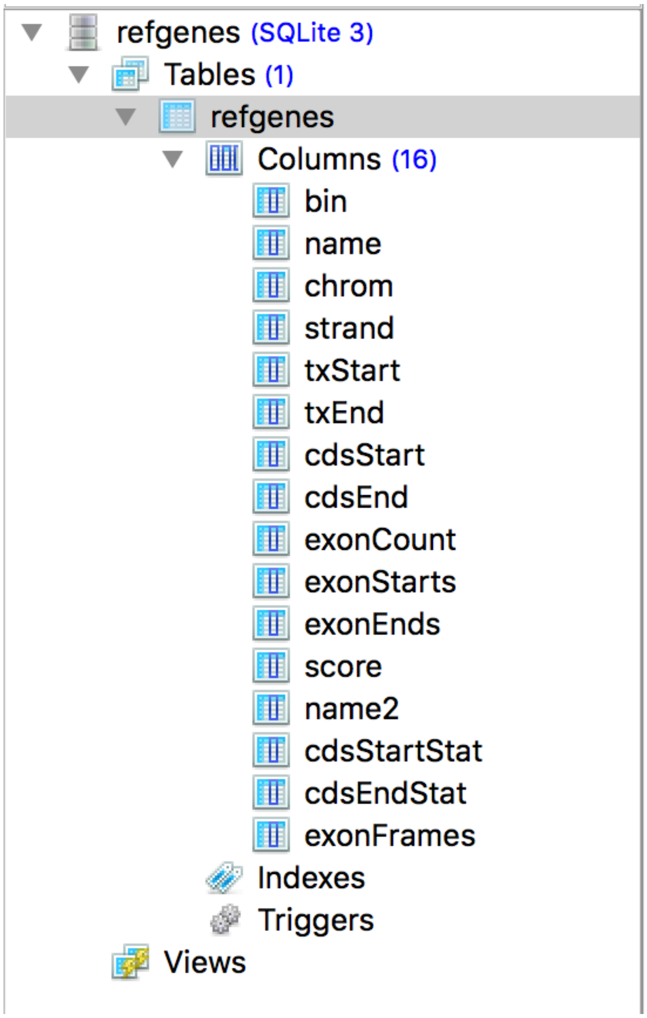
In the window there are three things you have to do:

1) Click on the Folder Icon and find the file that you want to load. If the extention of the file is not **CSV** make sure you ask it list **All files**

2) If your file has the name of the fields in the first row, make sure that the **First line represents CSV column names** box.

3) If your file is not comma separated but rather tab delimited, then make sure you pick the correct **Field separator**.

It will take some time for the data to load but once it's done you should see a **(1)** near Tables. Here you should see **refgenes** and if you open up the columns tab under refgenes, you will see all the fields.



Unfortunately the default option for SQLite Studio is not to guess the *data type* of each field. So to edit the data types of each columns, first select the table **refgenes** and then click on the **Edit Table** icon.

You will see several tabs, but by default the **Structure** tab will be selected showing you the fields. Here you have add the *data type* for each column by **double clicking** the empty cell and selecting the valid option. In case you are not sure what type of data is in which field, select the **Data** tab to show you the values.

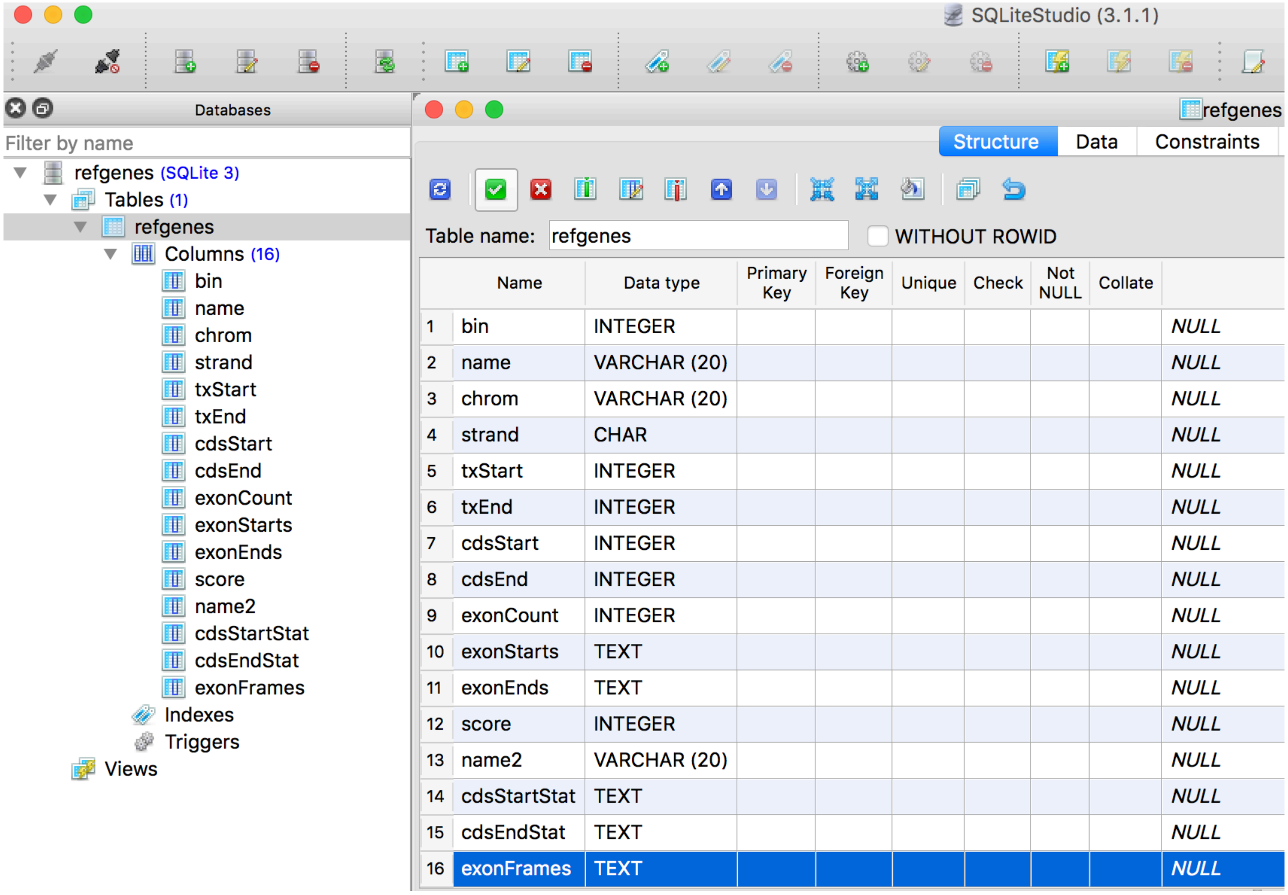
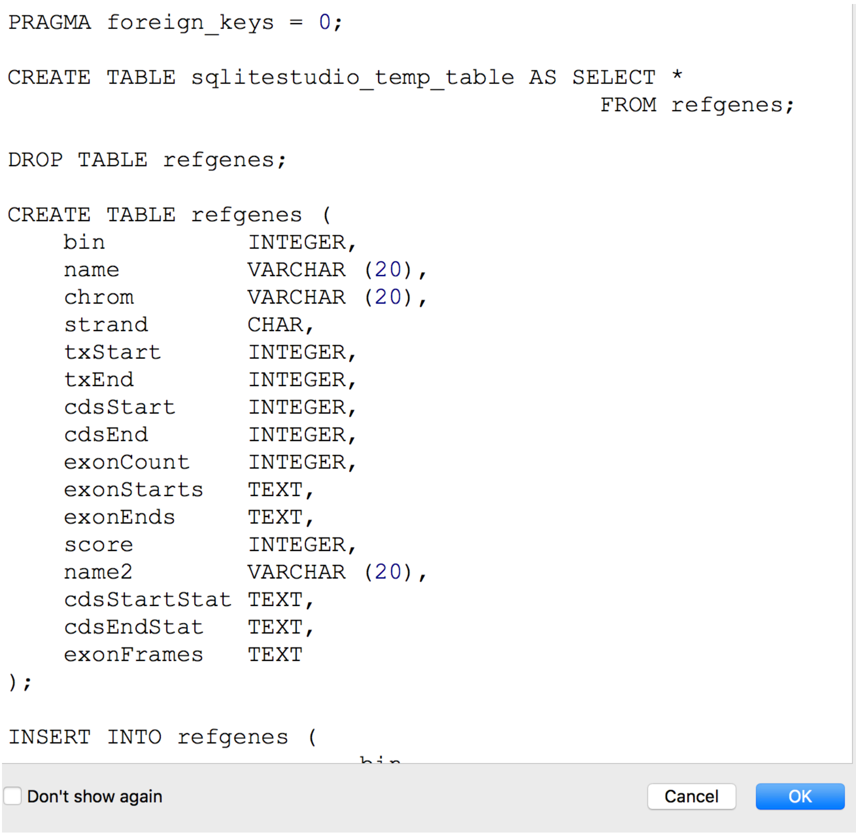


Table name: refgenes ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	
1	bin	INTEGER							*NULL*
2	name	VARCHAR (20)							*NULL*
3	chrom	VARCHAR (20)							*NULL*
4	strand	CHAR							*NULL*
5	txStart	INTEGER							*NULL*
6	txEnd	INTEGER							*NULL*
7	cdsStart	INTEGER							*NULL*
8	cdsEnd	INTEGER							*NULL*
9	exonCount	INTEGER							*NULL*
10	exonStarts	TEXT							*NULL*
11	exonEnds	TEXT							*NULL*
12	score	INTEGER							*NULL*
13	name2	VARCHAR (20)							*NULL*
14	cdsStartStat	TEXT							*NULL*
15	cdsEndStat	TEXT							*NULL*
16	exonFrames	TEXT							*NULL*

Once you have entered all the values for data-type, make sure to **committ** the changes by click on the *green arrow*. This will show you a window which includes all the SQL code that will be executed to make everything work. Notice that it first creates a temporary table (a copy o f refgenes), then it **Drops** refgenes and recreates it with the proper data types for each field, and then it copies all the data back into refgenes. Click **OK** to confirm the changes.



```
PRAGMA foreign_keys = 0;

CREATE TABLE sqlitestudio_temp_table AS SELECT *
                                          FROM refgenes;

DROP TABLE refgenes;

CREATE TABLE refgenes (
    bin          INTEGER,
    name         VARCHAR (20),
    chrom        VARCHAR (20),
    strand       CHAR,
    txStart      INTEGER,
    txEnd        INTEGER,
    cdsStart     INTEGER,
    cdsEnd       INTEGER,
    exonCount    INTEGER,
    exonStarts   TEXT,
    exonEnds     TEXT,
    score        INTEGER,
    name2        VARCHAR (20),
    cdsStartStat TEXT,
    cdsEndStat   TEXT,
    exonFrames   TEXT
);

INSERT INTO refgenes (
                         bin
```

Now you are ready to query the database.