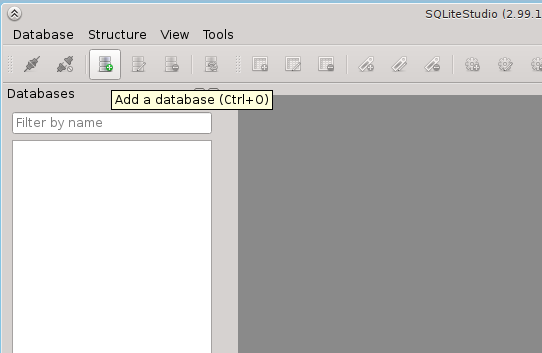
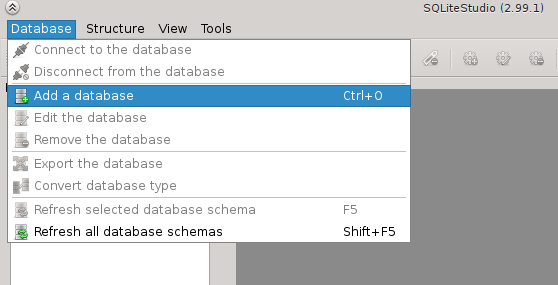
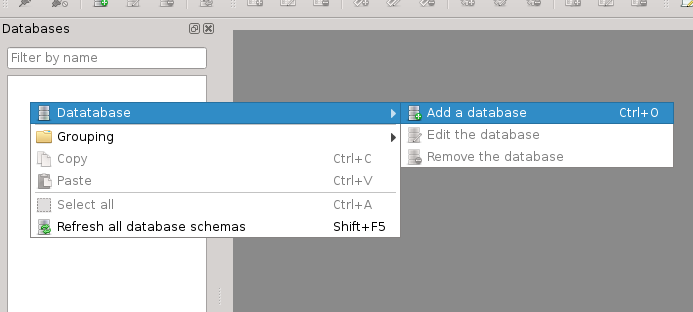
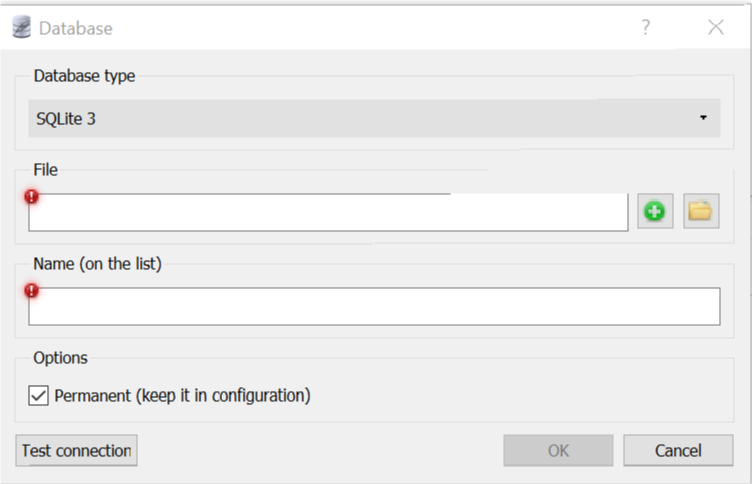
## Start the SQLite GUI application.

## Creating new database

You can create new database in three ways:

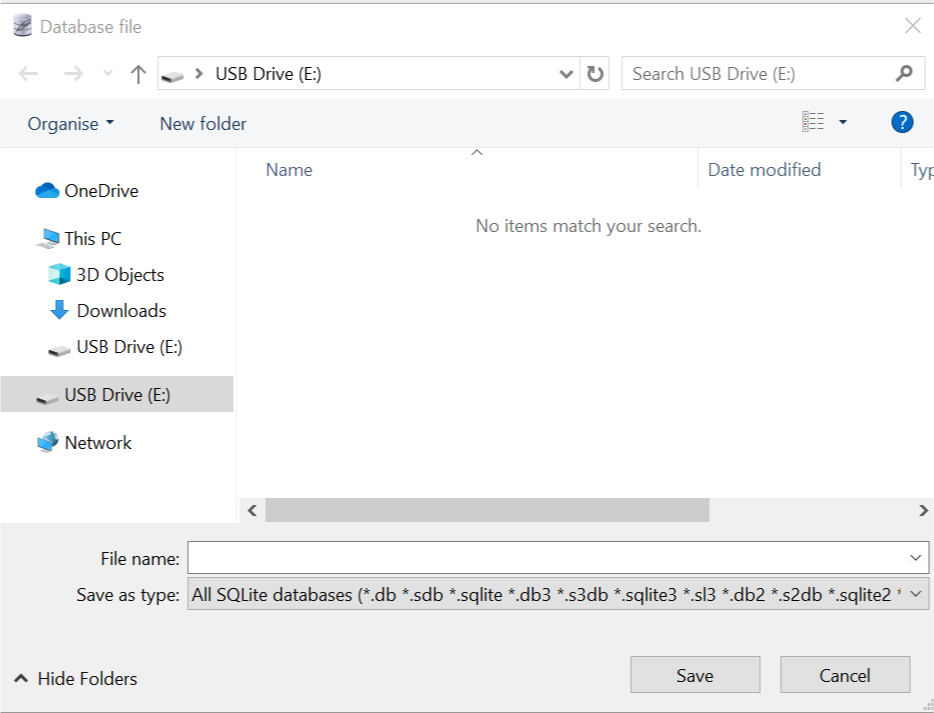
1. Click on toolbar button: 
2. Click on main menu entry: 
3. Right-click on database list and pick entry from context menu: 

Then the database dialog will appear:



Click on the  button:

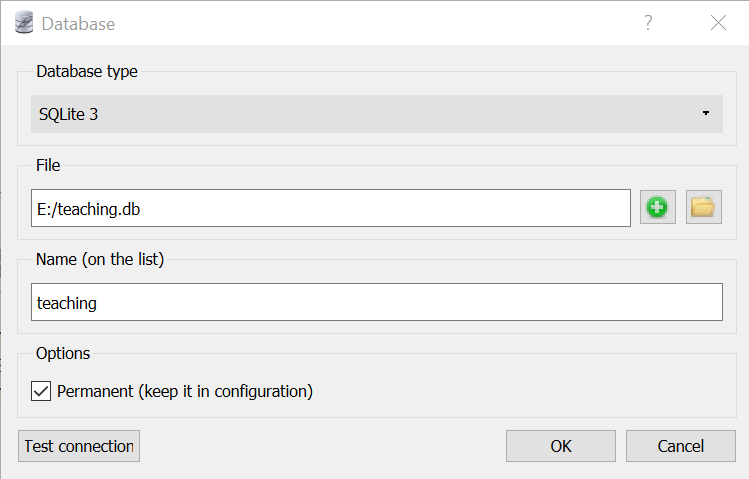
You will see filesystem browsing dialog (this may look different on various operating systems):



In this dialog go to directory where you want to create your database to be created. **Ideally create the database on a memory stick so you can access it wherever you are otherwise create it on the U: drive at University or your C: drive at home.**

Type in the file name for your database (on the image above it's "teaching.db") and click Save.

Back on the Database dialog, the name of the database will be entered based on the database filename you chose in the last step (in this example, it will be teaching). You can leave it as it is or change it to whatever you want. **The database name is just an alias used by SQLiteStudio to present databases in the list. You can pick any name you want as long as it doesn’t already exist as a database in SQLite Studio.**



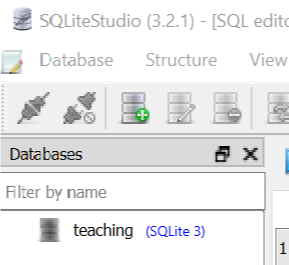
Under Options, ensure ‘Permanent (keep it in configuration)’ is ticked.

When this option is checked, the database is added permanently to SQLiteStudio. This means that when you close SQLiteStudio and start it again, the database will remain on the list. If this option is not checked, then the databases will disappear from the list after SQLiteStudio is restarted.

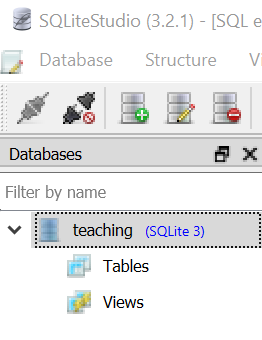
You can press "Test connection" to make sure that your database file can be created in specified localisation (in regards of directory permissions, free space, etc). If everything is okay, than you will see green icon next to it:

If there was any problem, the red icon will appear.

You can now press "Ok" to create the database. You will see it in the databases list on the left:



Double-click to open it:



You should see Tables and Views but these will be empty as it’s a newly-created database.

## Installing the Extension Functions

By default, SQLite only provides some basic mathematical and aggregate functions so to add these to your database, do as follows:

From the Tools menu, choose the ‘Open extensions manager’ option.

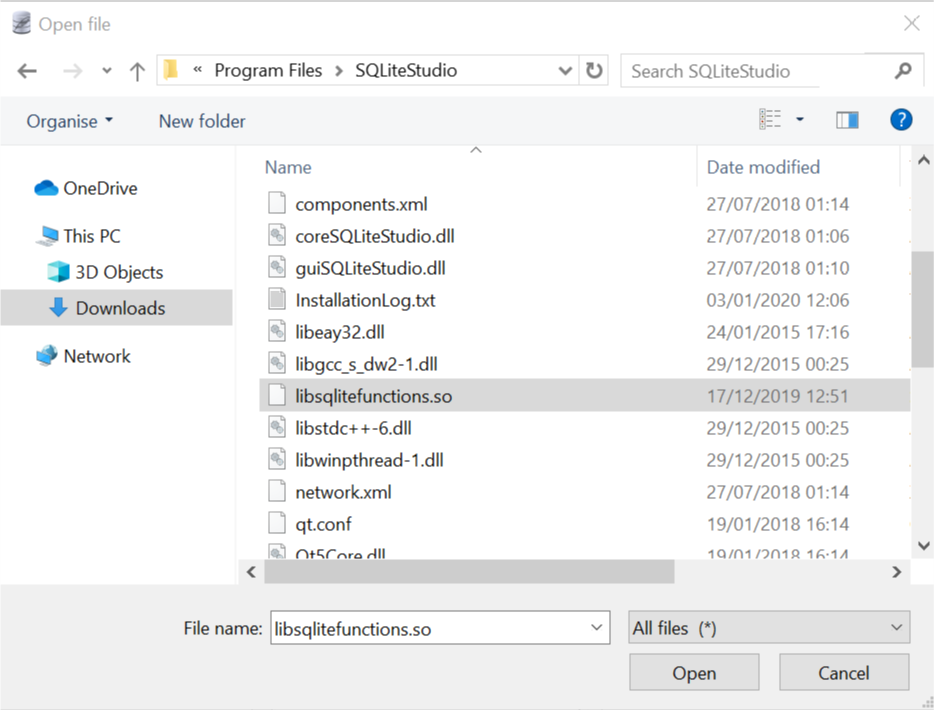
Then click on the ‘Add new extension’ icon.



Then click on the Open file button to the right of the Extension File text box



In the Open File dialog, type C:\ and then navigate to the directory where SQLite is installed (Program Files, SQLiteStudio), select ‘All files’ from the drop-down in the bottom right of the window and you should see the file libsqlitefunctions.so.



## Double-click on libsqlitefunctions.so file to return it to the previous window.

## Click on ‘Register in following databases’ and tick your own database

## 

## Click on the ‘Commit all extension changes’ icon

## 

## Finally test that the library has loaded correctly as follows:

## From the Tools menu, choose the ‘Open SQL Editor’ option.

## Type the following command:

## select cos(radians(45));

## If the library has loaded correctly, this query should return 0.707106781186548

## Opening an existing database file

## To open a existing database file that isn’t showing in the list of databases on the left-side of the SQLite Studio window, click Database and ‘Add a database’. Then, on the Database dialog, click the icon. The database file dialog box will open

## 

## Navigate to where the SQLite database file is located and click the Open button.

## Back on the Database dialog box, click ‘OK’ and the database will be added to the list of database showing in SQLite Studio. Double-click on the database name in the list and the database should become available to use.

## Managing database list

### Editing database details

You can edit database details at any moment. Just right-click on it and pick "Edit database" from context menu. The standard database dialog will appear.

Changing database's name is allowed as far, as the new name is unique across all databases being currently on the list.

**The database name is just symbolic name, an alias used by SQLiteStudio to present database in the list.**

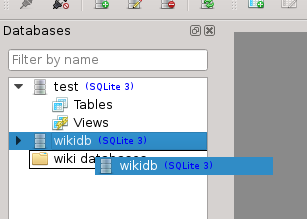
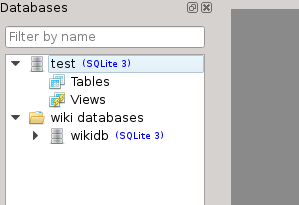
When changing database file path, it will mean that under the same name you will access the new database file. **It will not copy nor move the old database file to new location.**

When editing database details it's not possible to change it's type (SQLite2, SQLite3, or any other), because this is allowed only when creating new database. If you want to convert one SQLite version into another, use [database conversion dialog](https://github.com/pawelsalawa/sqlitestudio/wiki/User_Manual#converting-sqlite-database-version).

### Organizing database list

You can move databases up and down with mouse "drag & drop". You can create virtual "directories" where you can group databases. It helps maintaining big number of databases on the list.

To create a group, right-click on the databases group and pick "Grouping" → "Create group". Then you can drag & drop databases into that group:

Groups can be nested.

If you right-click on the database and pick "Create group", then the clicked database will be automatically moved into the group after the group was created.

### Deleting database from the list

To delete database from the list, select the database and either:

* right-click on it and pick "Remove database",
* or press "Delete" from the keyboard.

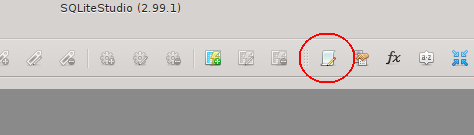
You will be asked for confirmation, before the database is actually removed from the list.

**Deleting database from the list does not delete database file! It only removes it from the list of databases in SQLiteStudio.** There's no way to delete actual database file using SQLiteStudio. This is for safety reason - if you really need to delete the database file, do it using your operating system utils.

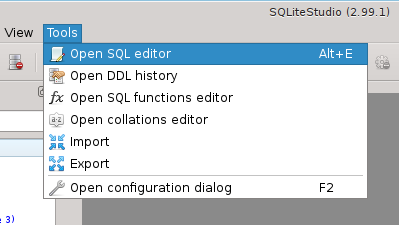
## Executing SQL queries

### SQL Editor window

To execute any SQL queries on SQLite database you have to open SQL Editor window:

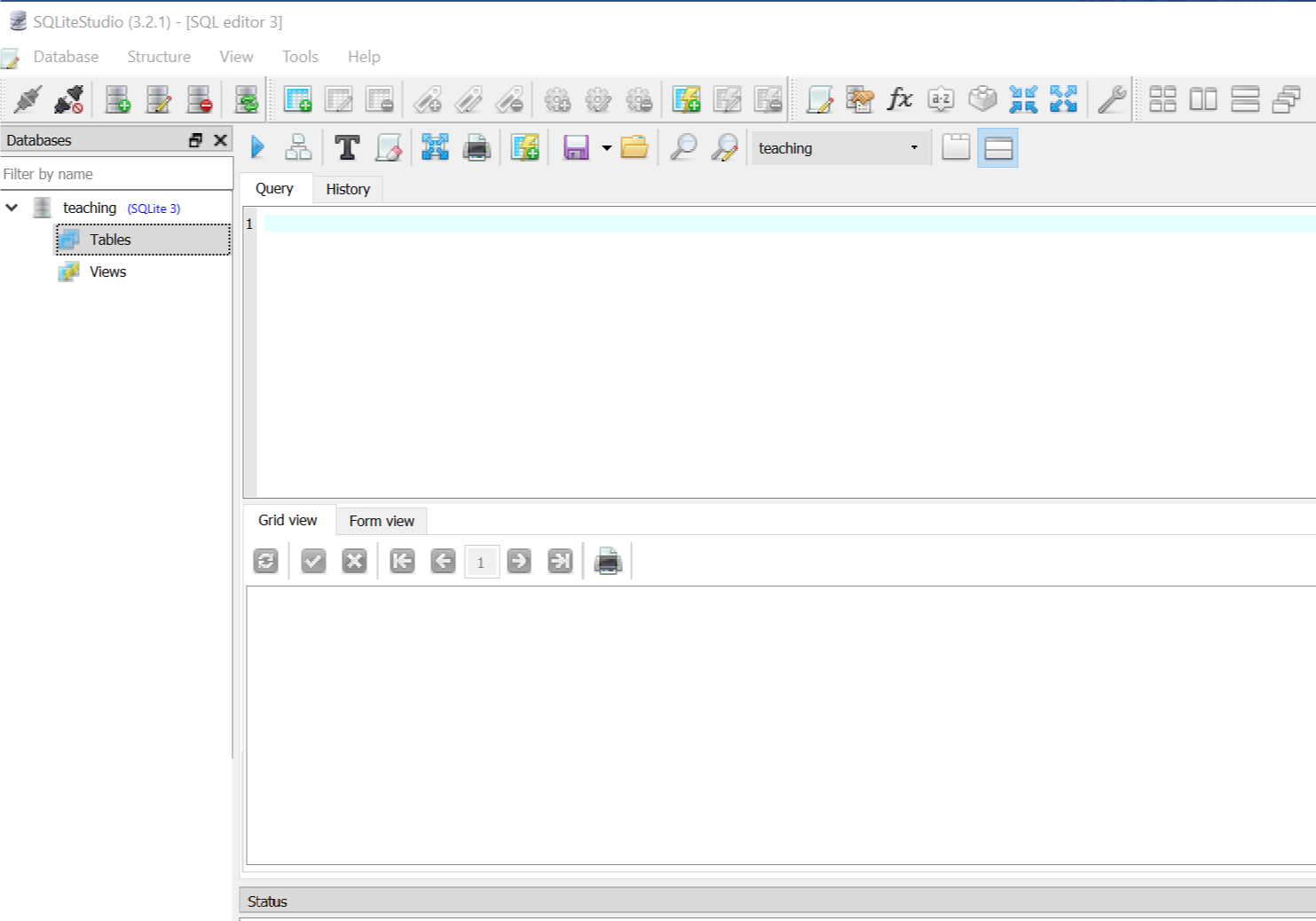


You can find it also in the "Tools" menu:

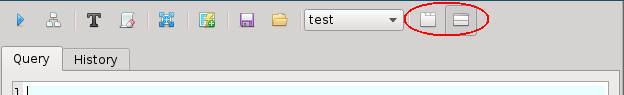


It's also available under keyboard shortcut "Alt+e" (by default, can be changed).

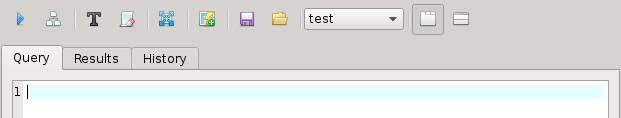
The SQL editor has a upper text edit for entering SQL queries and lower view for displaying results of the query:



If you prefer to have query edit field on one page and results on another page, you can switch it very quickly from toolbar:



And in the result you get 3 tabs in the editor, instead of 2:



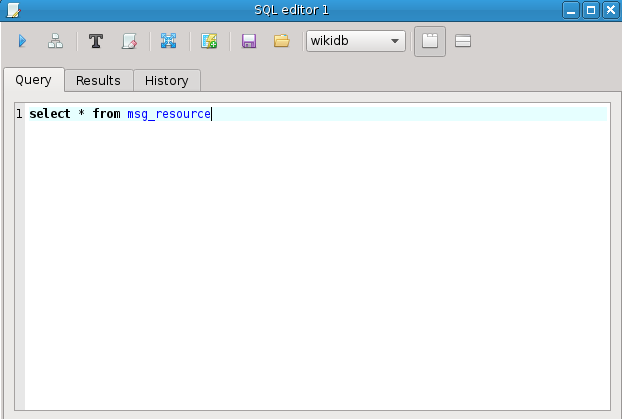
### Buttons on toolbar

* 1st button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Exec_query.png) executes query the you typed in the query text field below.
* 2nd button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Explain_query.png) executes **EXPLAIN** statement for the query below.
* 3rd button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Format_sql.png) uses currently configured [SqlFormatterPlugin](https://github.com/pawelsalawa/sqlitestudio/wiki/SqlFormatterPlugin) to format queries typed below.
* 4th button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Clear_history.png) clears query execution history (available in the last tab of the SQL Editor window).
* 5th button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Table_export.png) opens [Export\_dialog](https://github.com/pawelsalawa/sqlitestudio/wiki/User_Manual" \l "exporting-schema-and-data) for exporting results from query typed below.
* 6th button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Createviewfromquery.png) creates view from the **SELECT** query typed below.
* 7th button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Save_sql_file.png saves contents of the query text field below into the file.
* 8th button (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Open_sql_file.png) loads contents of selected file into the query edit field below.
* 9th position is a combo box, where you can pick current working database for the SQL Editor window. All queries are executed on database selected in this combobox.
* 10th and 11th buttons (https://github.com/pawelsalawa/sqlitestudio/wiki/images/Results_in_tab.png https://github.com/pawelsalawa/sqlitestudio/wiki/images/Results_below.png) are a shortcut to configure SQL Editor results presentation mode - in separate tab, or below the query field.

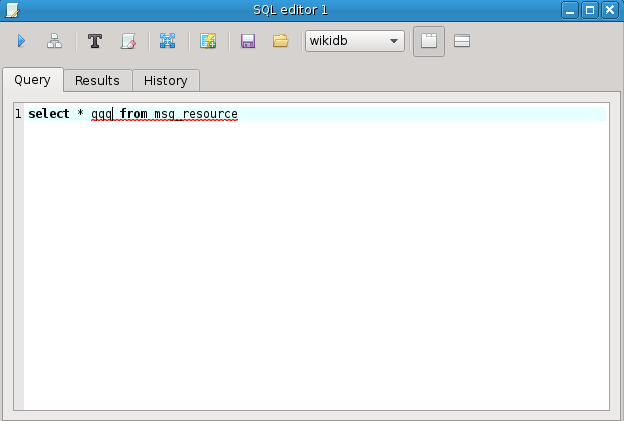
The list may differ if any plugin adds its own buttons to the toolbar.

### Typing SQL queries

When you type SQL query in the editor, it will highlight SQL syntax, but also if you type the correct name of any object (that is existing table, index, trigger, view, database or column name), the name gets highlighted with blue color (by default, it's configurable):



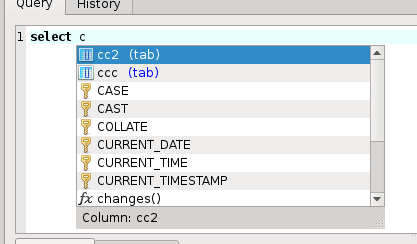
If you use incorrect SQL syntax, the problematic part will be underlined with wavy line:



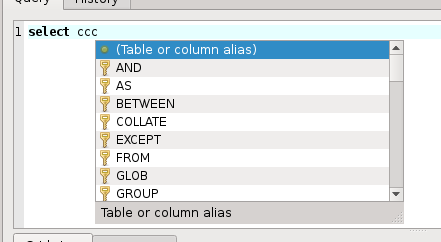
To execute the query, use toolbar button https://github.com/pawelsalawa/sqlitestudio/wiki/images/Exec_query.png, or keyboard shortcut "F9" (the shortcut is configurable).

### Code completion

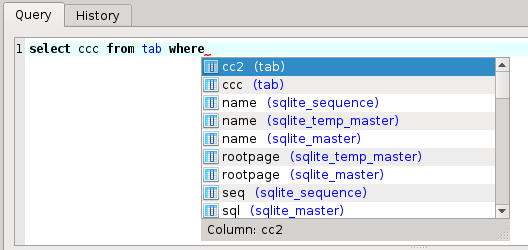
SQLiteStudio provides code assistant, so it can help you with the SQL syntax. When you hit Ctrl+Space (it's a configurable shortcut), it will show you all possible values for the current cursor position. Let say you typed "SELECT c" and hit Ctrl+Space:



You can see it shows only valid entries that start with the letter "c". Let's pick "ccc", then ask the assistant about next suggestions:

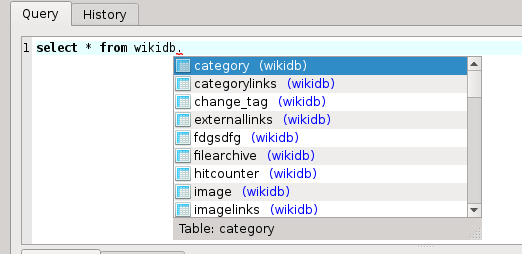


First position (with a bullet icon) is a pseudo value, which tells you that you can put an alias name for table or column here. You typed "ccc" before, which is a column name, so here you could type an alias name for it. We won't do that. Instead we pick "FROM" and add "tab where" and ask the assistant again:



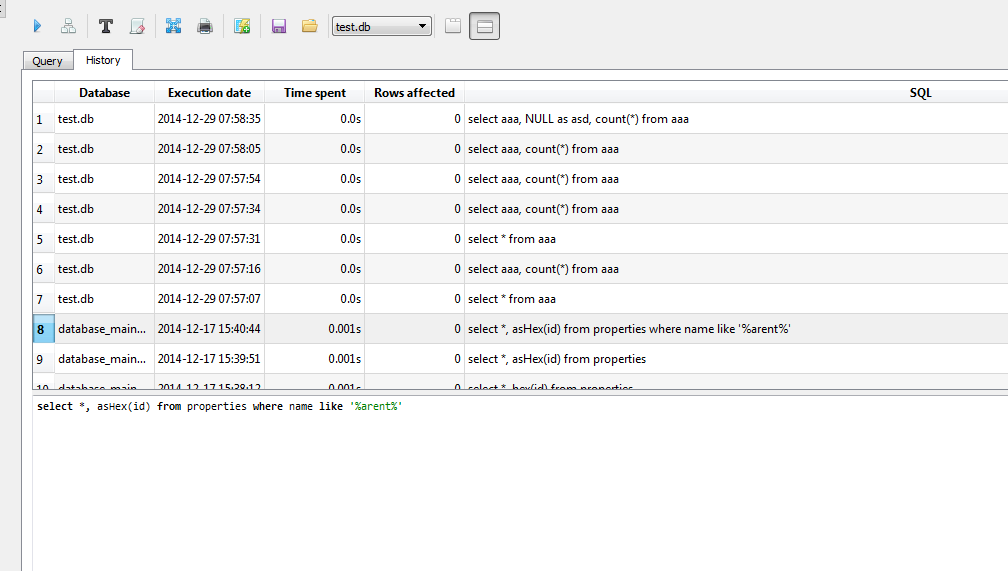
The assistant proposes several columns which you can use in your "where" clause, but notice, that columns from "tab" table are listed at the begining, as you use that table in the query already. This is how the assistant tries to be smart with proposing values - when they are more likely to be used in the context, it will list them at the top.

The assistant is triggered manually by the keyboard shortcut mentioned above, but it also triggers automatically when you type "name.", where the name is name of existing database or table. In following example I didn't use keyboard shortcut, this just appeared after I types "wikidb.":



### Query execution history

SQL history tab keeps the history of SQL queries you have ever executed. The number of queries kept in history is configurable in [Configuration dialog](https://github.com/pawelsalawa/sqlitestudio/wiki/Configuration_dialog) and defaults 1000 entries. When that limit is exceeded, the oldest entry gets deleted.



You can browse and preview queries from the history. Double-click on the entry to automatically copy it into SQL editor, so you can re-execute it.

You can erase entire SQL queries history by clicking on "Erase history" toolbar button and confirming your choice.

**For further information**

<https://www.sqlitetutorial.net/>

<https://likegeeks.com/python-sqlite3-tutorial/>