This document is about setting up SpatiaLite version 4.2.0 and its installation on Linux and Mac OS.

If you are a windows user visit link below for installation:

* Windows installation:
  + <https://www.gaia-gis.it/gaia-sins/>
  + <https://www.gaia-gis.it/spatialite-2.3.1/install-windows.html>

**Steps for installation**

* **Install SQLite3**, the SQLite is the core DBMS for setting up the SpatiaLite.
* **Install Homebrew**, this helps you to install packages easily.
* **Install gcc/C++ compiler**, this helps you to run GIS packages for SpatiaLite.
* **Install SpatiaLite**

**SQLite3 installation**

Depending on the operating system that you are using, download the SQLite package from the following link and save it on your computer.

SQLite Download Page: <http://sqlite.org/download.html>

Open the terminal and go to the directory that you saved the file. Then follow the steps bellow:

* Unzip the file: tar xvfz sqlite-autoconf-\*.tar.gz
* Go to the file directory: cd sqlite-autoconf-\*
* The run this command: ./configure
* After that type: make
* And at the end type: make install

**Homebrew installation**

Based on the operating system that you are running on your machine, follow steps for installation from bellow links for each OS:

* Linux: <http://linuxbrew.sh/>
* Mac OS: <https://brew.sh/>

**GCC/C++ Compiler installation**

Most of the time this compiler is installed and comes with your OS. If the compiler is not installed on your machine then go to this link: <https://gcc.gnu.org/wiki/InstallingGCC> and follow the instruction for installation.

**SpatiaLite Installation**

**Install SpatiaLite libraries**

Now, it’s time to install SpatiaLite. To begin this part, first we need to install required libraries which are GIS packages and extension for SQLite database. Follow the steps bellow for this part:

* Get the latest SpatiaLite library source bundle from this link: <http://www.gaia-gis.it/gaia-sins/libspatialite-sources/>

Or run this command on the terminal:

wget https://www.gaia-gis.it/gaia-sins/libspatialite-sources/libspatialite-4.2.0.tar.gz

Then follow bellow steps one by one:

tar xvfz libspatialite-4.2.0.tar.gz

cd libspatialite-4.2.0

./configure

make

sudo make install

* The SpatiaLite library version that is explained in this document is 4.2.0. By the time that you are reading this document, there might be newer version released.

If you are getting an error after running ./configure during the installation like the following error:

configure: error: 'libxml2' is required but it doesn't seem to be installed on this system.

Then run this command:

brew install libxml2

If still getting same error even though you installed the package, then run this command on terminal:

brew install pkg-config

Then rerun the command ./configure and this time it should complete the installation.

**Install SpatiaLite Tools**

**Using Homebrew**

Homebrew handles all SpatiaLite related packages. You can install them by running the following commands on the terminal one by one:

brew update

brew install spatialite-tools

brew install gdal

**Using command line on terminal**

* First download the tool package form this link: <https://www.gaia-gis.it/fossil/spatialite-tools/index>
* Then go to the downloaded file directory and run the following commands in terminal:

For Linux:

tar xvfz spatialite-tools-4.2.0.tar.gz

cd libspatialite-4.2.0

./configure

make

sudo make install

For Mac:

tar xvfz spatialite-tools-4.2.0.tar.gz

cd libspatialite-4.2.0

./configure --target=macosx

make

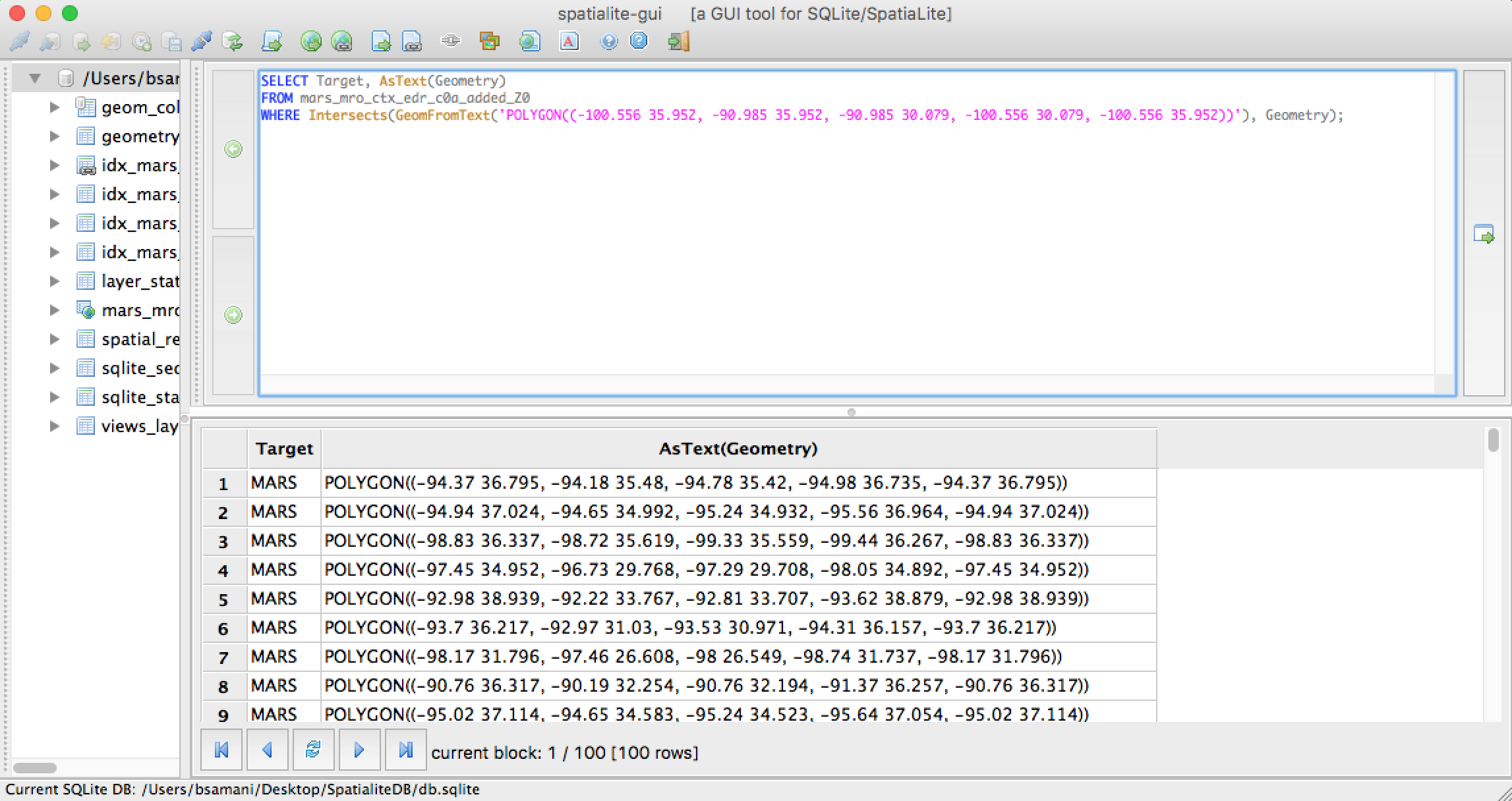
sudo make install

After installation, you will see a folder named “Spatialite Tools”. Now you can go to this folder and open another folder called “tool” and setup your SpatiaLite database inside this folder by running the following command in the terminal:

spatialite <database name>.spatialite

The spatialite tools inside this folder will create the database and now the database is reading to crate tables, setup their relations and insert data into tables.

There is another option for creating database and tables. Using GUI software like spatialite-gui helps you to setup the database and tables graphically and more interactively. You can download the application from this link: <https://www.gaia-gis.it/fossil/spatialite_gui/index>



It is available for Windows, Mac and Linux operating systems. Using this application helps users to test spatial queries and check the result. It also helps to visualize the spatial data.

**Spatial queries with SpatiaLite:**

A query for calculating the area of each polygons in the table “mars\_mro\_ctx\_edr\_c0a\_added\_Z0”:

SELECT PK\_UID, AsGeoJSON (ST\_Centroid (Geometry)) AS center

FROM mars\_mro\_ctx\_edr\_c0a\_added\_Z0;

A query for finding the intersection of given polygon with all the polygons that we have in the table “mars\_mro\_ctx\_edr\_c0a\_added\_Z0”:

SELECT Target, ST\_AsText (Geometry)

FROM mars\_mro\_ctx\_edr\_c0a\_added\_Z0

WHERE ST\_Intersects (ST\_GeomFromText ('POLYGON ((-100.556 35.952, -90.985 35.952, -90.985 30.079, -100.556 30.079, -100.556 35.952))'), Geometry);

A query for finding all polygons that completely exist in a defined bounding box from table “mars\_mro\_ctx\_edr\_c0a\_added\_Z0”:

SELECT PK\_UID, AsText(Geometry)

FROM mars\_mro\_ctx\_edr\_c0a\_added\_Z0

WHERE MBRContains (BuildMBR (-100.556,35.952, -90.985,30.079), Geometry);

A query to create bounding box around polygons that are in the table “mars\_mro\_ctx\_edr\_c0a\_added\_Z0”:

SELECT PK\_UID, ASGeoJSON (ST\_Envelope(Geometry)) FROM mars\_mro\_ctx\_edr\_c0a\_added\_Z0;

A query to find polygons from table “mars\_mro\_ctx\_edr\_c0a\_added\_Z0” that the distance of center point of those polygons from a defined point/location is more than 300 kilometers:

SELECT PK\_UID, ODEId, ST\_Distance(ST\_Centroid(Geometry), ST\_GeomFromText ('POINT (-100.556 35.952)')) AS Distance

FROM mars\_mro\_ctx\_edr\_c0a\_added\_Z0

WHERE ST\_Distance (ST\_Centroid (Geometry), ST\_GeomFromText ('POINT (-100.556 35.952)')) > 300;

The following table shows some of frequently used GIS function in both database systems and how the function can be different in each one.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **GIS Function Des.** | | **GIS Function def.** | **SpatiaLite** | **PostGIS** | **Difference** | |
| **SpatiaLite** | **PostGIS** |
| 1 | See the results as text | ST\_AsText |  |  | ----- | ------ |
| 2 | Converting results to GeoJSON | ST\_AsGeoJSON |  |  | AsGeoJSON | ST\_AsGeoJSON |
| 3 | Finding center point of a polygon | ST\_Centroid |  |  | ----- | ----- |
| 4 | Finding distance between two points | ST\_Distance |  |  | ----- | ----- |
| 5 | Finding the rectangle bounding the geometry as a Polygon | ST\_Envelope |  |  | ----- | ----- |
| 6 | Check if two polygons intersecting each other | ST\_Intersects |  |  | ----- | ----- |
| 7 | Find all polygons that have intersection with  another polygon | ST\_Intersection |  |  | ----- | ----- |
| 8 | Finding all polygons that are in a bounding box | ST\_MakeEnvelope |  |  | BuildMbr | ST\_MakeEnvelope |
| 9 | Returns a geometric object that is the convex hull of a geometry | ST\_ConvexHull |  |  | ----- | ----- |
| 10 | Finding a geometric object that is the set union of two polygons | ST\_Union |  |  | ----- | ----- |

**References:**

* SpatiaLite installation: <https://docs.djangoproject.com/en/2.0/ref/contrib/gis/install/spatialite/>
* SpatiaLite manual: <http://www.gaia-gis.it/gaia-sins/spatialite-manual-2.3.1.html>
* SpatiaLite tutorial: <https://www.gaia-gis.it/gaia-sins/spatialite-tutorial-2.3.1.html>
* SpatiaLite functions reference list: <http://www.gaia-gis.it/gaia-sins/spatialite-sql-4.2.0.html>
* SpatiaLite cookbook: <https://www.gaia-gis.it/spatialite-3.0.0-BETA/spatialite-cookbook/html/python.html>