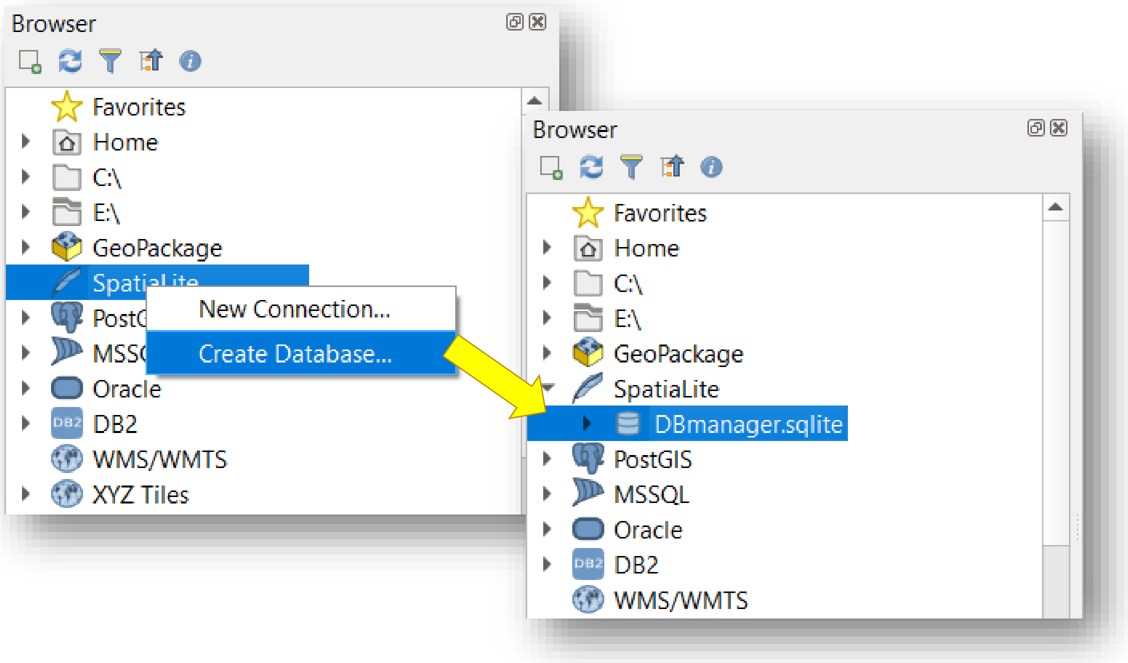
DB Manager: Duplicate values in vector layers, QGIS

DB Manager is a very powerful tool to manage and query tables. This instruction is only a very straightforward guide of how to check duplicate values in vector layer's fields, and not a general introduction to DB Manager. If you want to know more about DB Manager, take a look at the introductory document **DB Manager: Short introduction** that is available for download in Climate Risk Database.

Duplicate values are not allowed in fields that should only contain unique values. These kind of fields are, for example, the FID (field identifier) or OID (object identifier) and other thematic fields, such as unique names or codes of the objects in question.

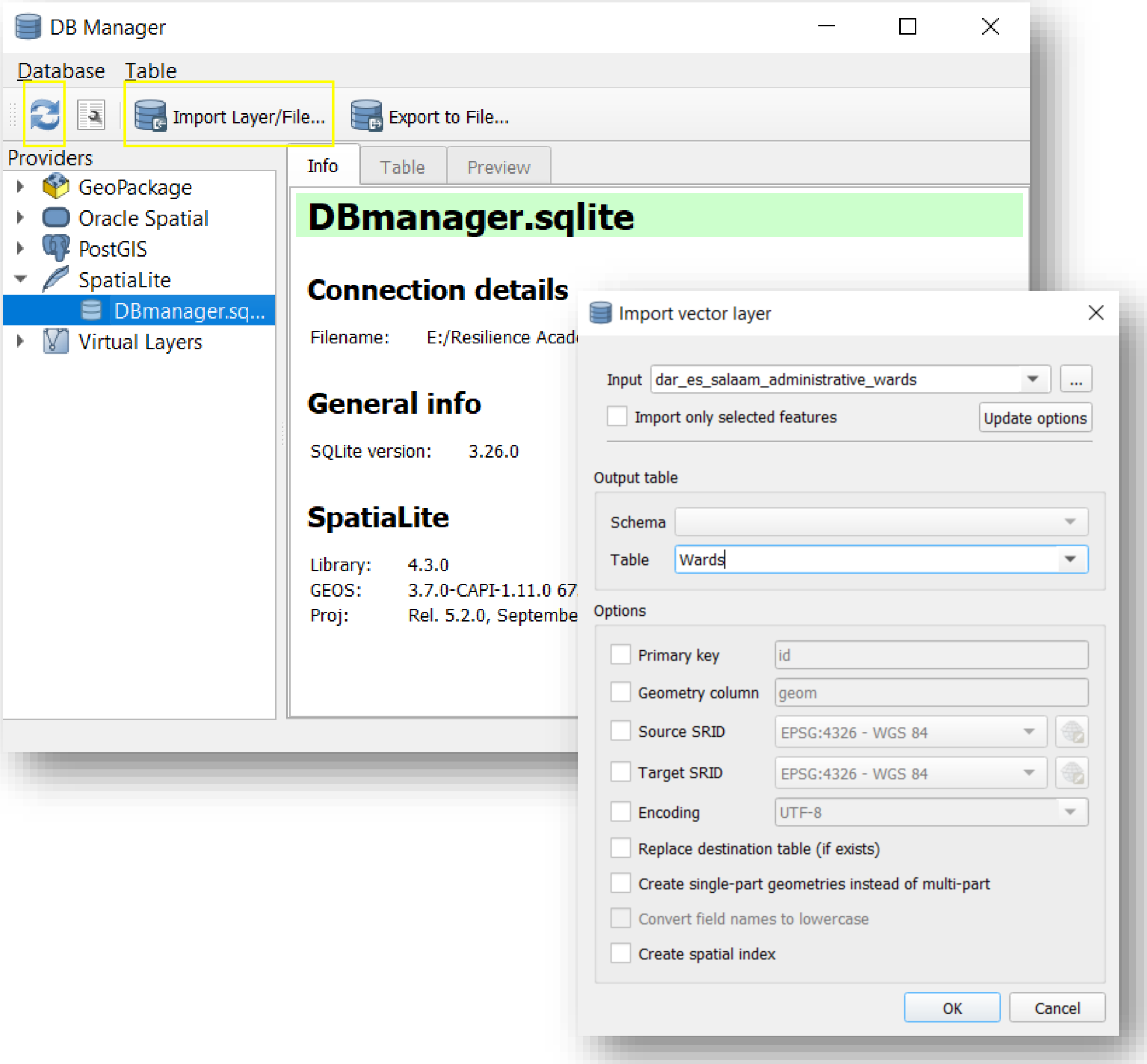
Preparations

1. Open your vector layer in QGIS.
2. You must create a new database connection in order to view the attribute table in DB Manager. This can be done in QGIS by using PostIGS or SpatiaLite databases. Let's use SpatiaLite this time.
3. Find *SpatiaLite* from your browser window in QGIS. Right-click it and select *Create Database…*
4. Browse to the folder where you want to create the database. The folder can be any of your working folders. Give the database a name. In this example, the database got name "DBmanager".
5. Click *Save*. Now, the new database connection appeared below *SpatiaLite* in QGIS browser window.



DB Manager

1. Open DB Manager by clicking *Database →DB Manager…* from the top navigation panel of QGIS.
2. Expand *SpatiaLite* and you find the database you just created under it.
3. If instead of the *Connection details* you see text that indicates the database is not connected to DB manager, don't worry. Just wait a moment, or close and re-open the DB Manager and *Refresh* the databases. Sometimes it takes a few moments to create the connection.
4. Import your layer to the database you created. Select the database by clicking it, and then click *Import Layer/File…* from the top panel. Select your layer from the drop-down menu or search it from your folders. You only have to give the imported table a name. In this example, the table is named as "Wards". No other settings are needed.



DBmanager.sqlite

Connection details

Filename: E:/Resilience Acad

General info

SQLite version: 3.26.0

SpatiaLite

Library: 4.3.0
GEOS: 3.7.0-CAPI-1.11.0 67
Proj: Rel. 5.2.0, Septembe

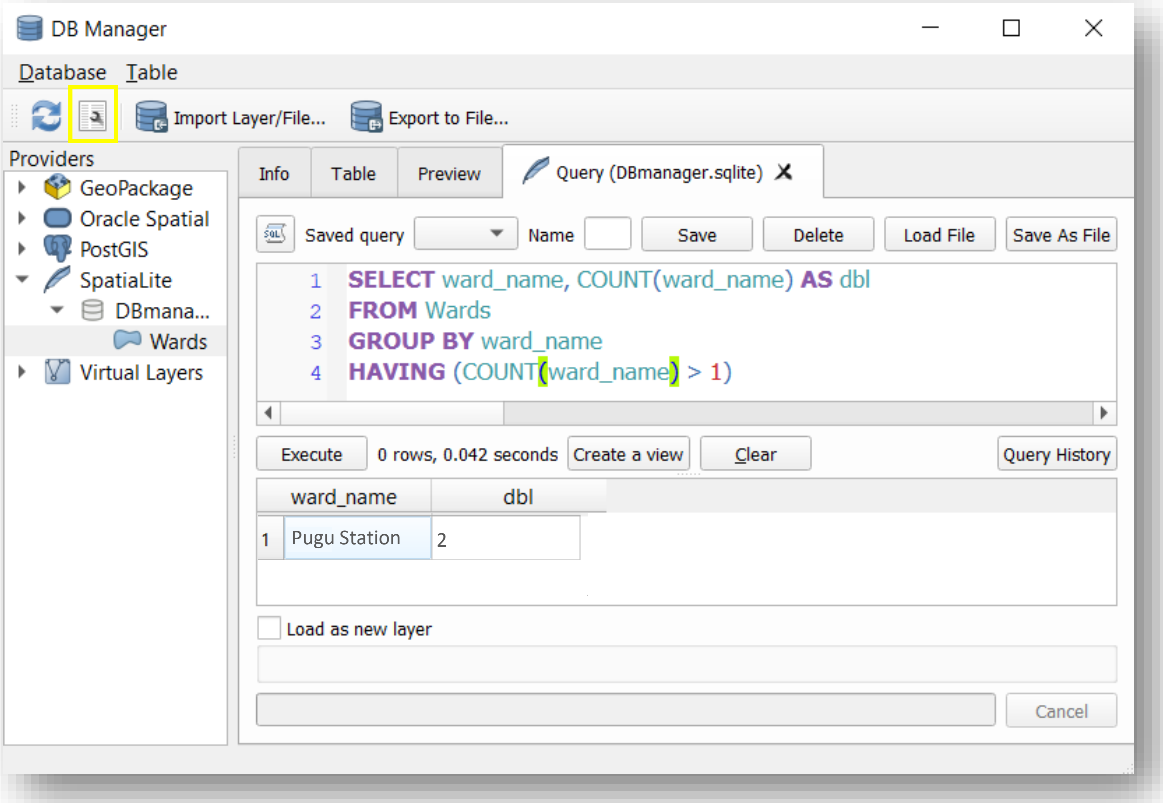
Input: dar_es_salaam_administrative_wards
□ Import only selected features

Output table
Schema:
Table: Wards

Options
□ Primary key: id
□ Geometry column: geom
□ Source SRID: EPSG:4326 - WGS 84
□ Target SRID: EPSG:4326 - WGS 84
□ Encoding: UTF-8
□ Replace destination table (if exists)
□ Create single-part geometries instead of multi-part
□ Convert field names to lowercase
□ Create spatial index

5. Look for the duplicate values by opening *SQL Window*.
6. Attach the following query to the query field. Change "column_name" to match with the name of the column in your dataset you want to query, and "table_name" to match with the name of your table.
7. Click *Execute*.

```
SELECT column_name, COUNT(column_name) AS dbl
FROM table_name
GROUP BY column_name
HAVING (COUNT(column_name) > 1)
```



```
SELECT ward_name, COUNT(ward_name) AS dbl
FROM Wards
GROUP BY ward_name
HAVING (COUNT(ward_name) > 1)
```

0 rows, 0.042 seconds

	ward_name	dbl
1	Pugu Station	2

□ Load as new layer

8. The results are printed to the lower window. If there are no duplicates in the column you are querying, this results window is left empty. If there are duplicates, the results shows which value is in the column multiple times, and how many times it occurs. In this example, the ward Pugu Station seems to be in the dataset twice. This is not desired, since the ward names should be unique. Next, the duplicate value should be examined from the attribute table, and fixed accordingly.