INTERNATIONAL TRAINING ON TOPONYMY

MANUAL GUIDES

19 - 23 JUNE 2023

BALI, INDONESIA

United Nations
Group of Experts on
Geographical Names

BADAN INFORMASI
GEOSPASIAL

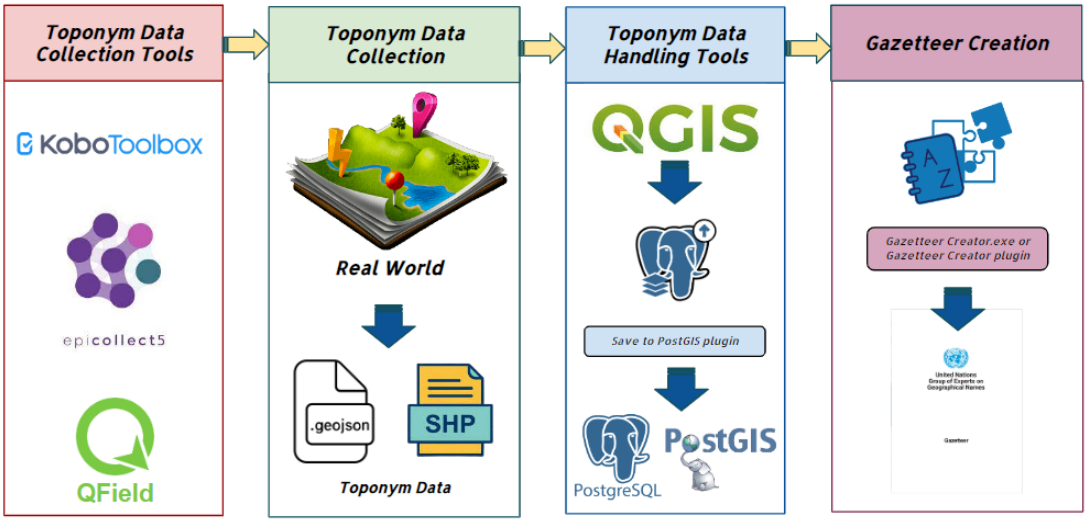
MANUAL GUIDE

Data Processing & Management

(QGIS, PostgreSQL, PostGIS)

3.1 Introduction

The general flow of processing and managing Training Data in PostgreSQL is depicted in the following flowchart:



The collected geographical names data, obtained through surveying tools, undergoes extraction into spatial formats such as GeoJSON or Shapefile to facilitate subsequent data processing and management. During this training, the primary tool employed for processing geographical names data is GIS software, particularly QGIS. Data management is carried out using PostgreSQL, a Database Management System (DBMS), which is enhanced by the spatial capabilities offered by the PostGIS extension. To streamline the migration of geographical feature data into the database, the *Save to PostGIS plugin* is utilized within QGIS.

The spatial data, in the form of geographical names, is imported into QGIS to undergo processing with the principal aim of ensuring compatibility with the PostgreSQL geographical names database format. This includes consistent column naming and attribute completion. Upon successful formatting, the geographical names data is inserted into the database using the *Save to PostGIS plugin*, developed by BIG. This plugin empowers users to insert data into pre-existing tables specifically designed within the PostgreSQL. Data management is performed utilizing a Database Management System (DBMS), taking into account various factors including data accessibility to enable access by multiple users, data security involving

authorization, data administration and management encompassing tasks such as backup, restoration, and data optimization, and supporting scalability.

The geographical names data stored within the database can be accessed by multiple users for a variety of purposes, including processing and presentation. Users have the option to utilize GIS applications such as QGIS and Openjump, GIS servers such as Geoserver, or develop their own web-based or desktop applications. In this training, the stored data is additionally presented in the form of a gazetteer utilizing the *Gazetteer Creator plugin* in QGIS and *Gazetteer Creator* (standalone desktop application). This plugin and application facilitates data retrieval from the database and enables the generation of a gazetteer in PDF format, specifically designed for printing purposes.

The steps of the Data Processing and Management workflows are :

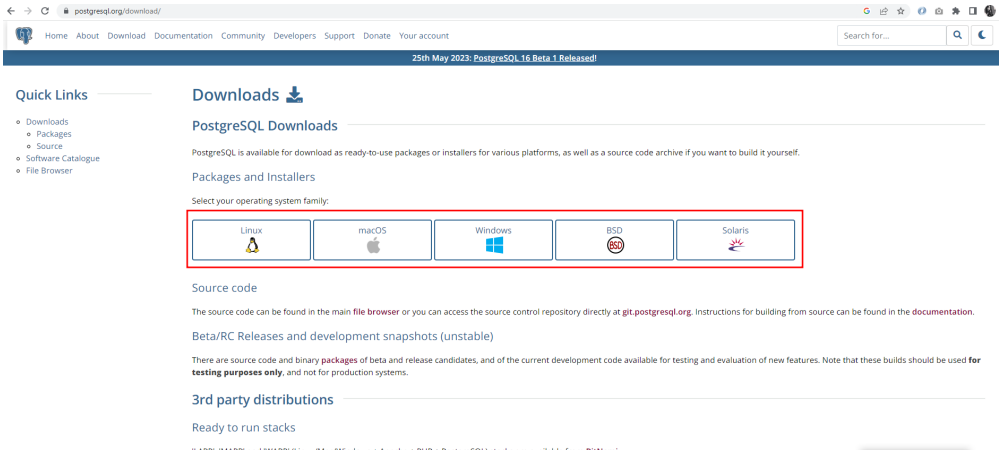
- Database Creation - Create Database using PostgreSQL
- Table Creation - Create Geographical Names Table based on design
- Data Processing - Ensuring compatibility with the PostgreSQL geographical names database format, including consistent column naming and attribute completion
- Save to PostGIS - Migrating data to Geographical Names Table using Save to PostGIS plugin

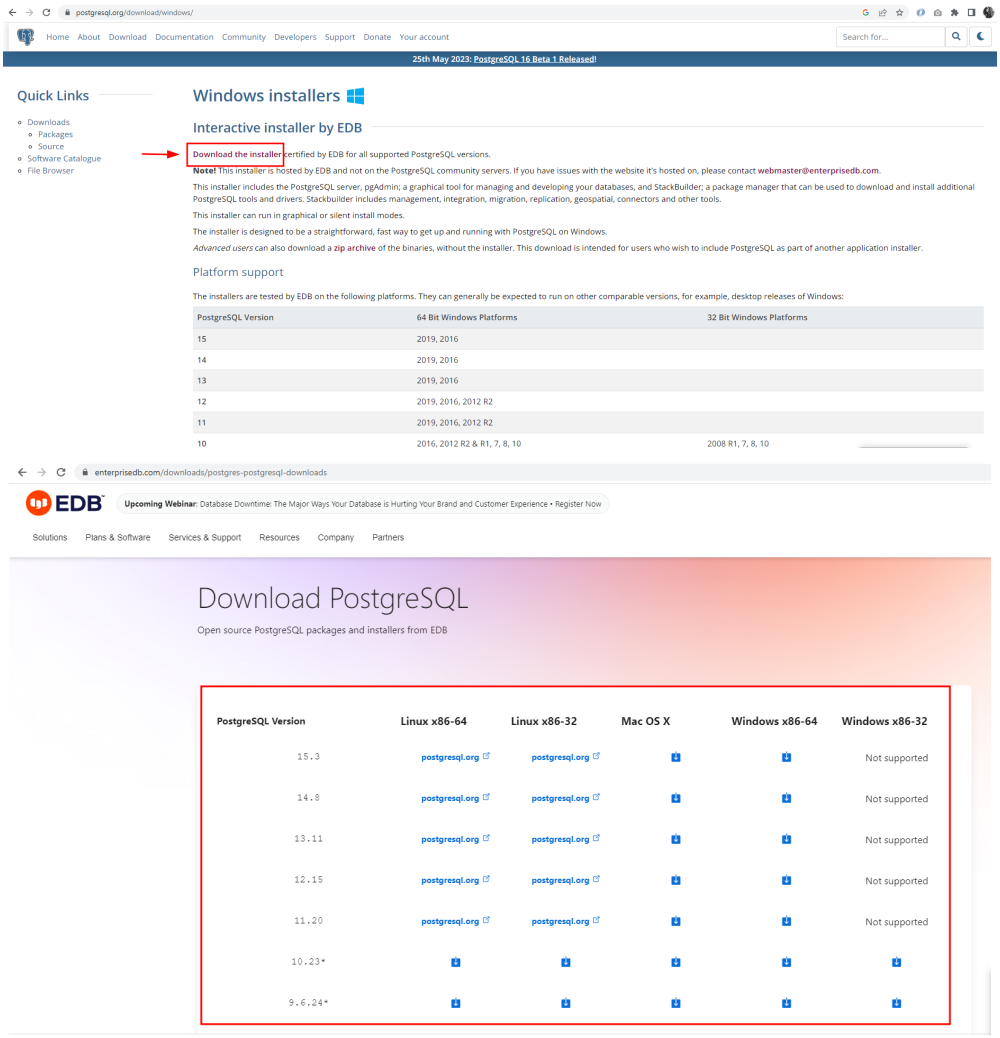
3.2 Installment

1. PostgreSQL

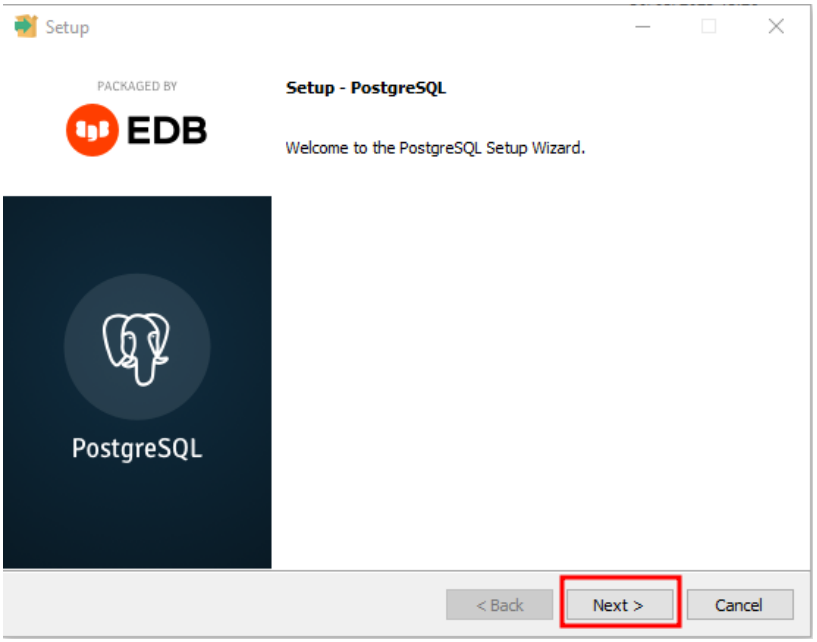
The installation procedure for the PostgreSQL server on a Windows operating system entails the following general steps:

1. Visit the official PostgreSQL website at https://www.postgresql.org/download
2. Proceed to download the appropriate installation file from the provided link.

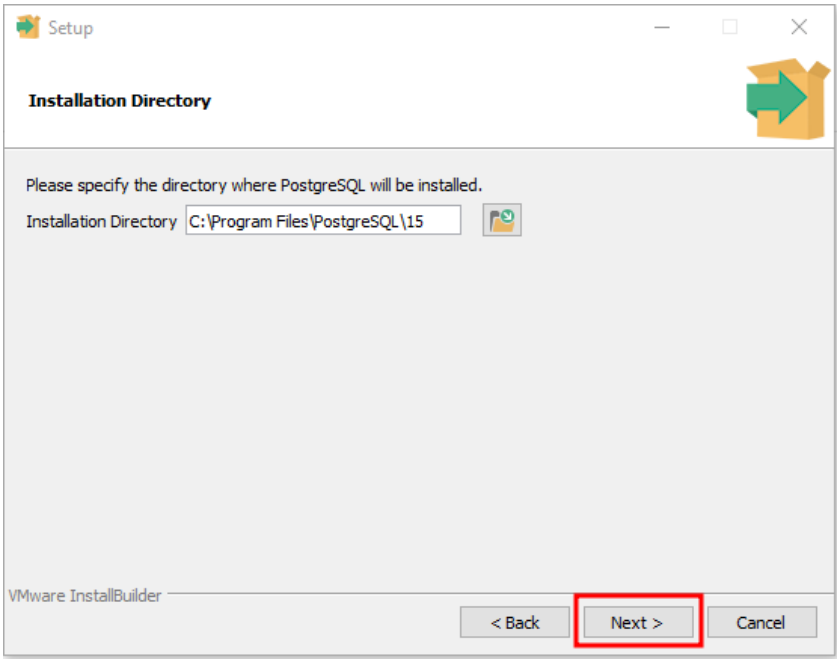




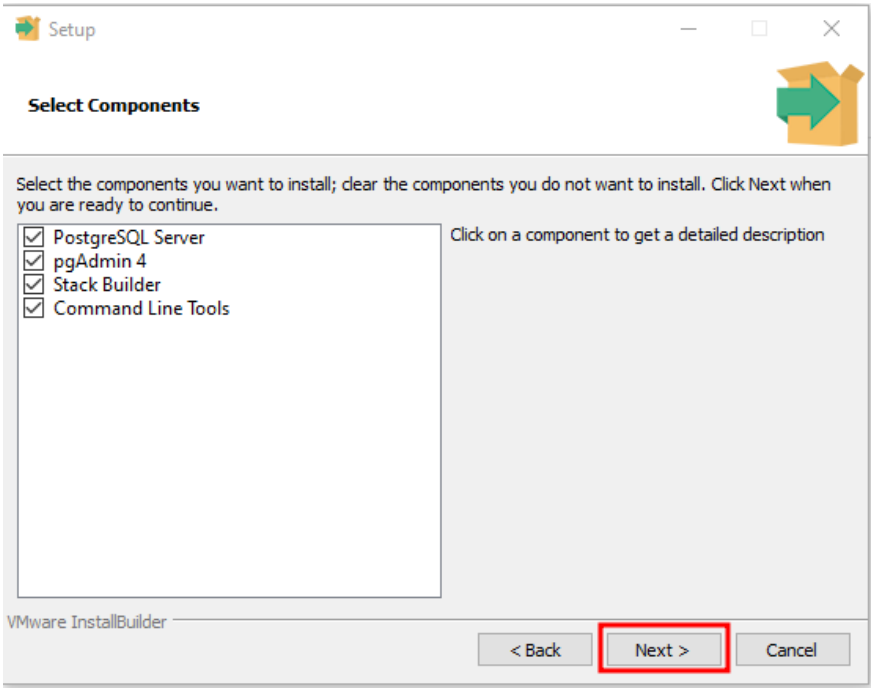
3. Run the installation file: Double-click on the downloaded file to start the installation process. If a security warning appears, click "Run" or "Yes" to proceed
4. Click **Next** button



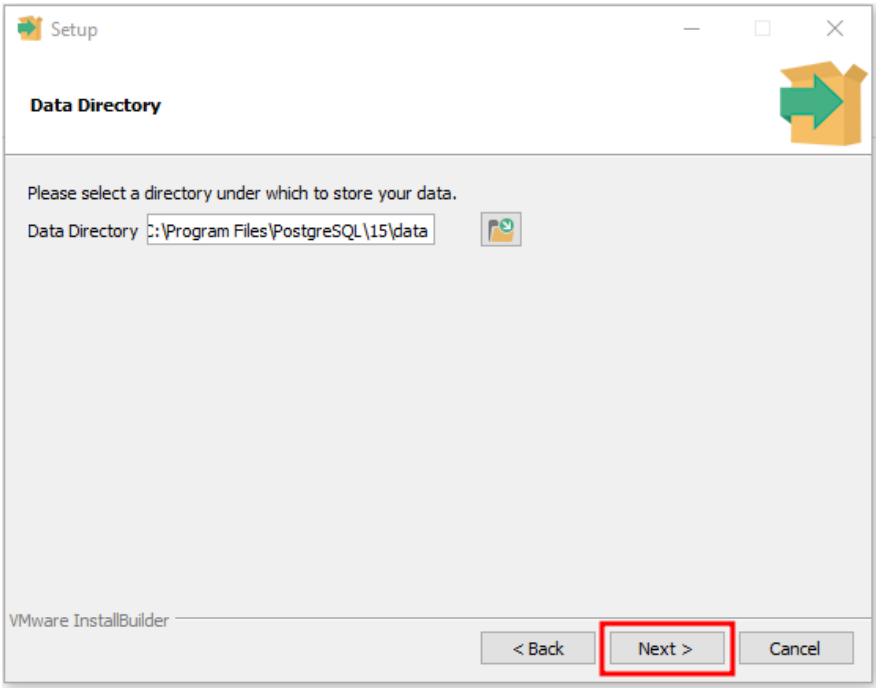
5. Set up the folder installation and click **Next** button



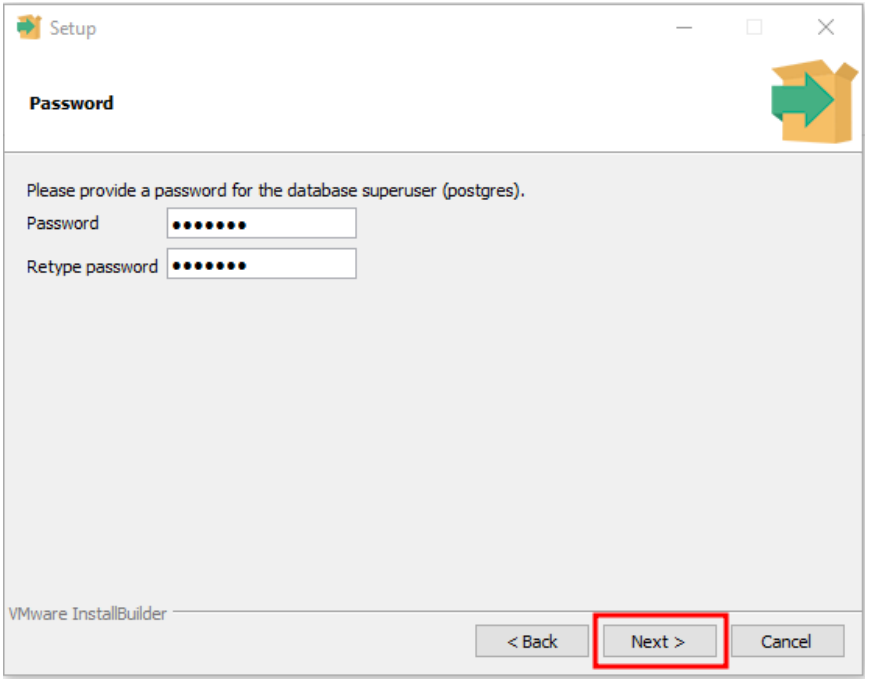
6. Select software component to install :
   a. The PostgreSQL Server to install the PostgreSQL database server
   b. pgAdmin 4 to install the PostgreSQL database GUI management tool.
   c. Command Line Tools to install command-line tools such as psql, pg_restore, etc. These tools allow you to interact with the PostgreSQL database server using the command-line interface.
   d. Stack Builder provides a GUI that allows you to download and install drivers that work with PostgreSQL.



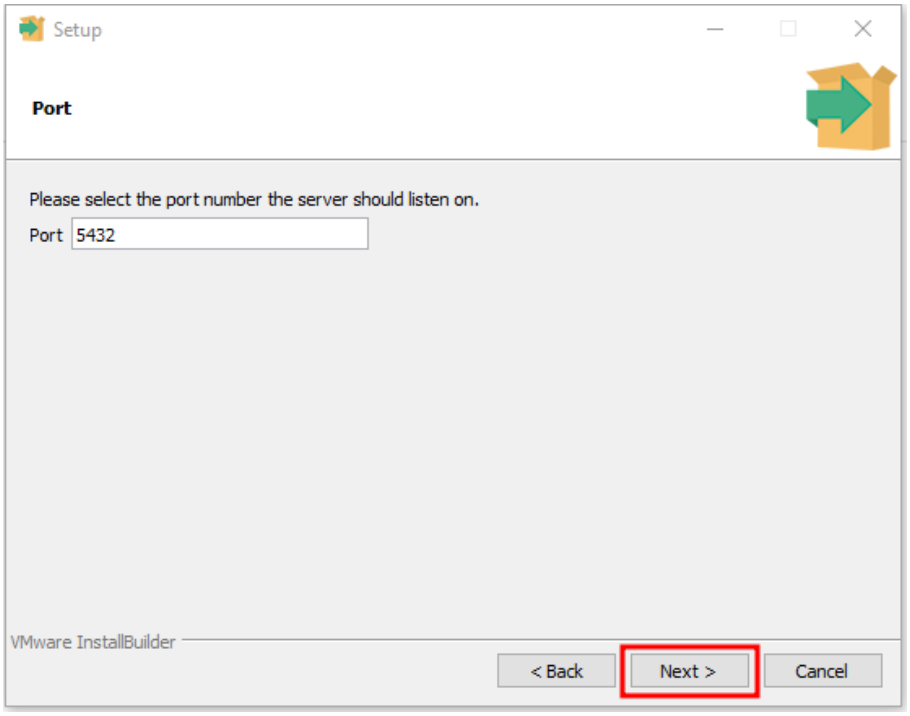
7. Select the database directory to store the data

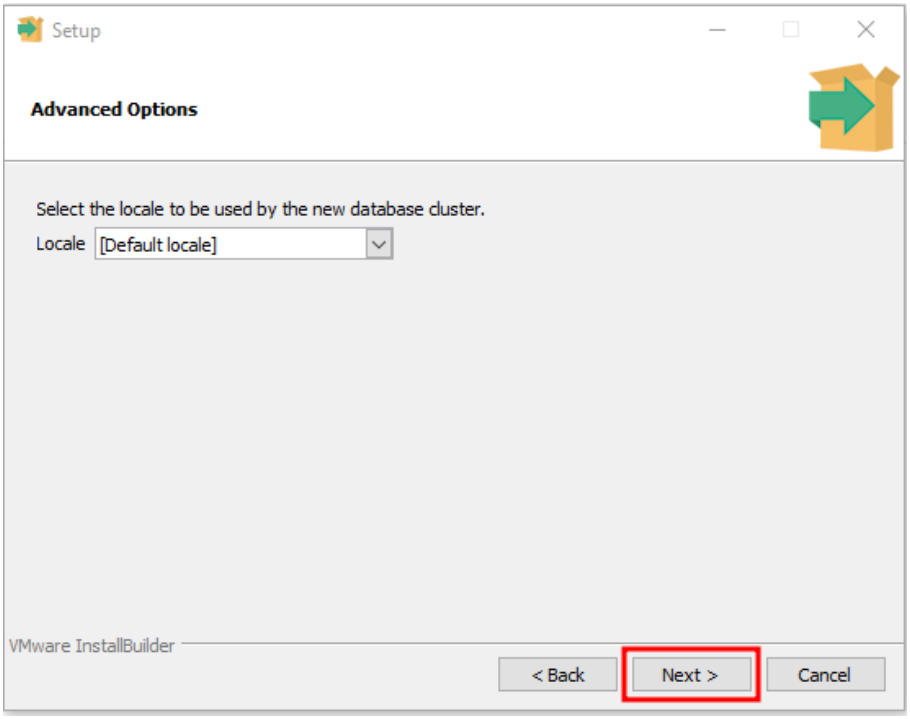


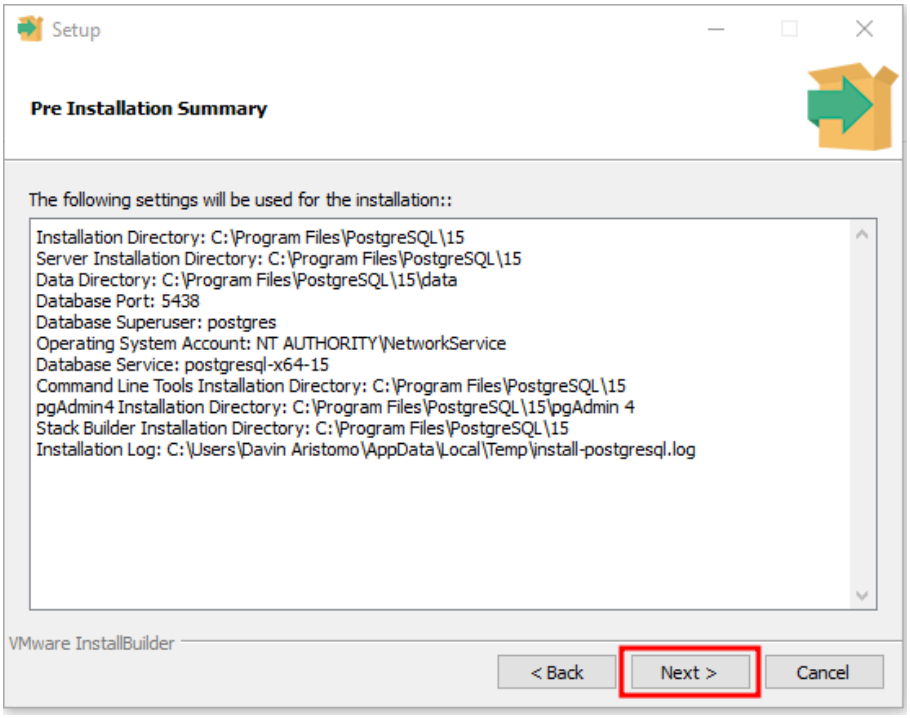
Set up password and click Next

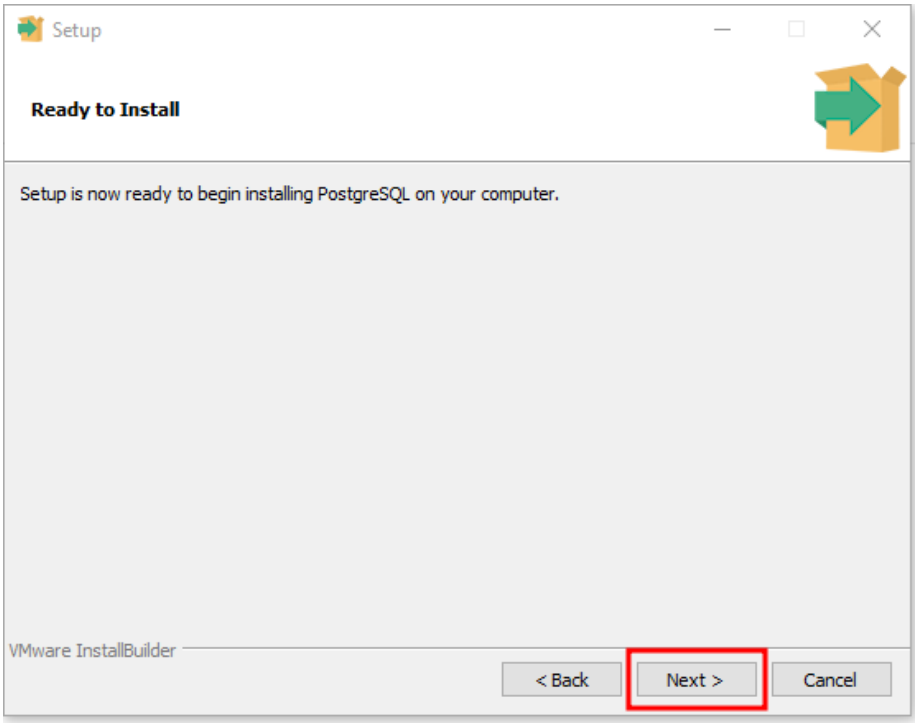


8. Set up the application port and click Next and wait until the installation finish

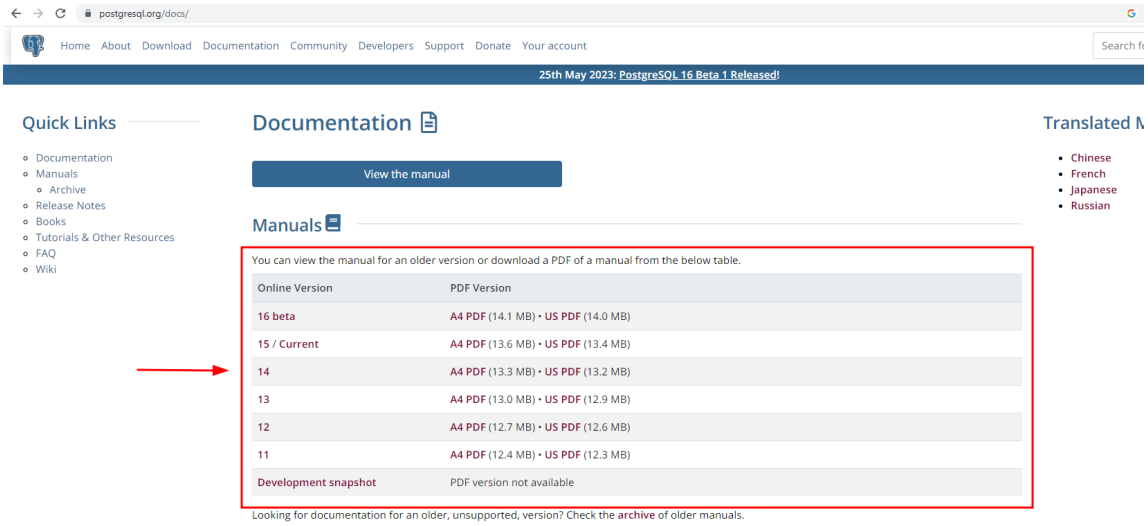








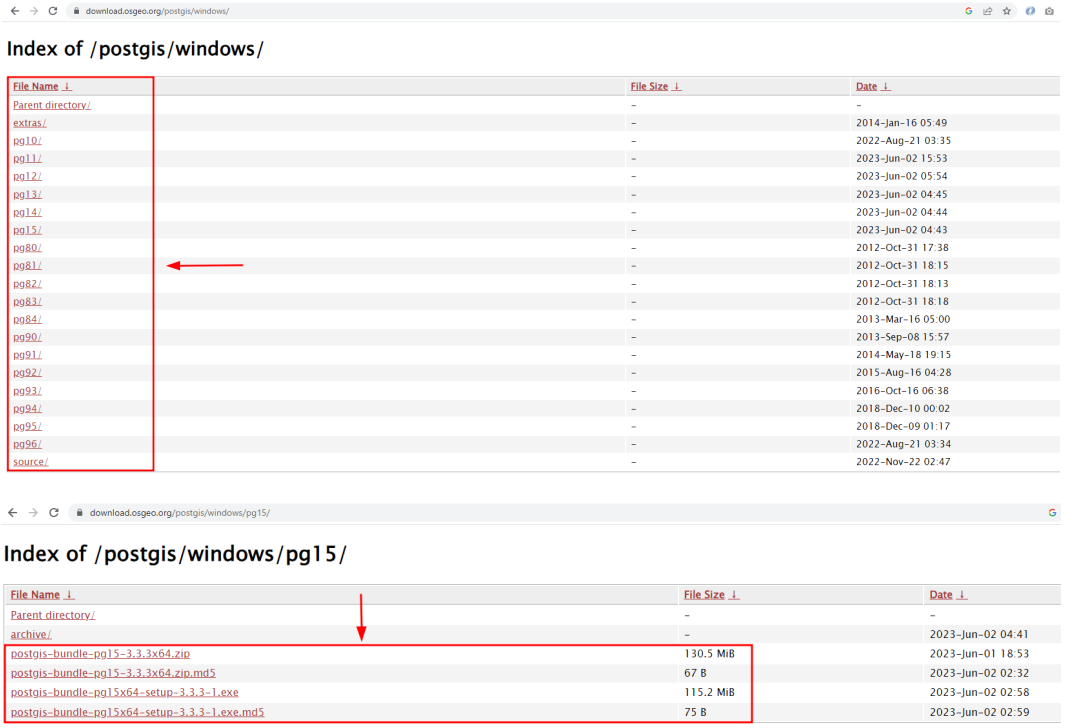
To install PostgreSQL on another operating system, please visit https://www.postgresql.org/docs/ to access the detail instructions based on your operating system.



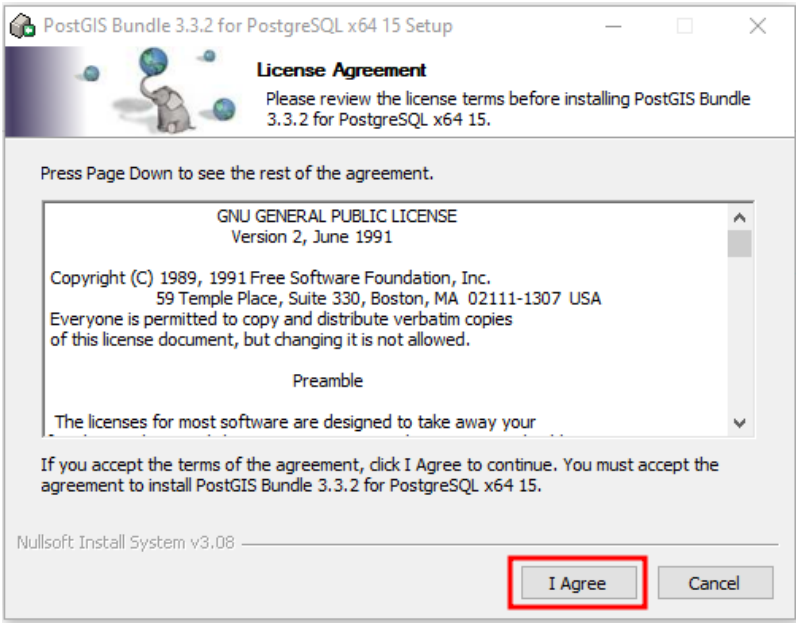
2. PostGIS

The installation procedure for the PostGIS extension on a Windows operating system entails the following general steps:

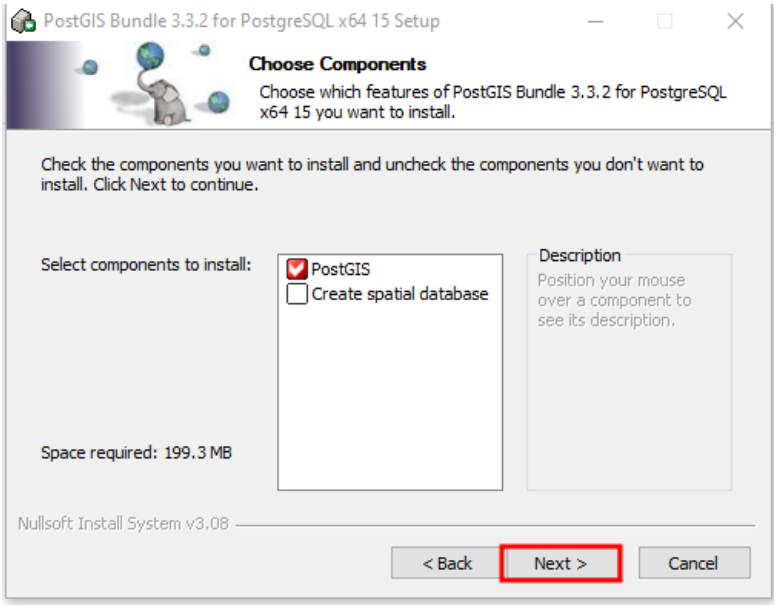
1. Visit https://download.osgeo.org/postgis/windows/ and select the version that is compatible with your postgresql server version.



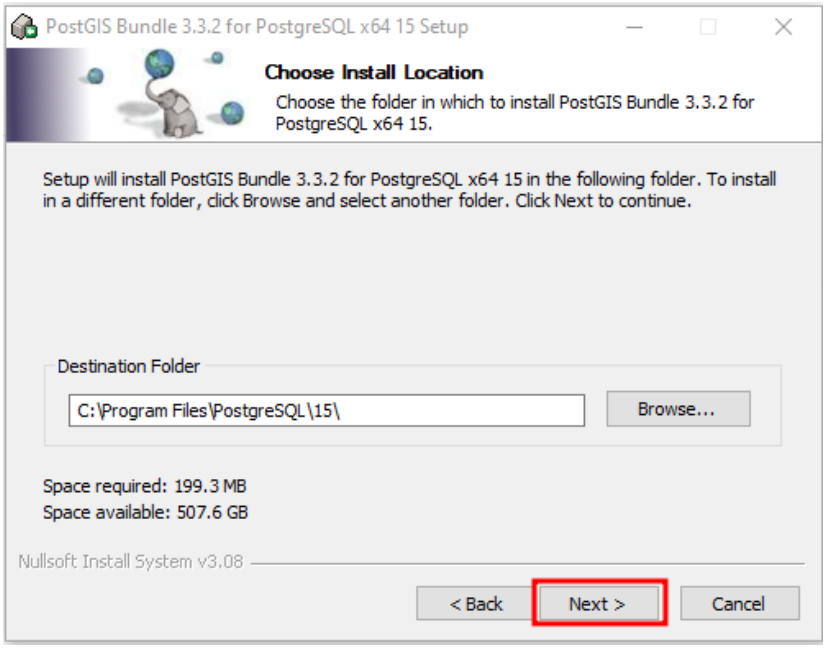
2. Run the installation file and click *I Agree*

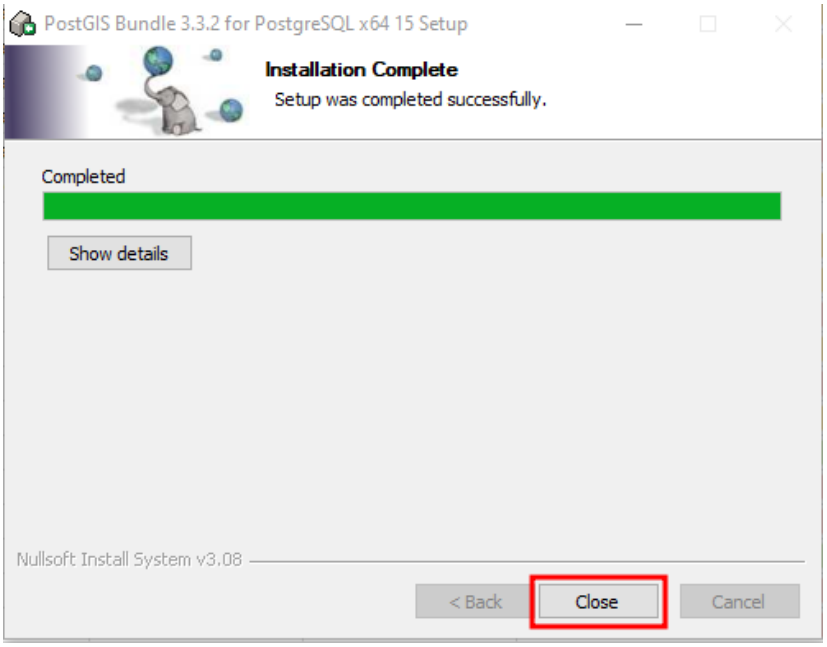


3. Select component to install > PostGIS and click Next



4. Set the destination folder, click Next and wait until the installation finishes.



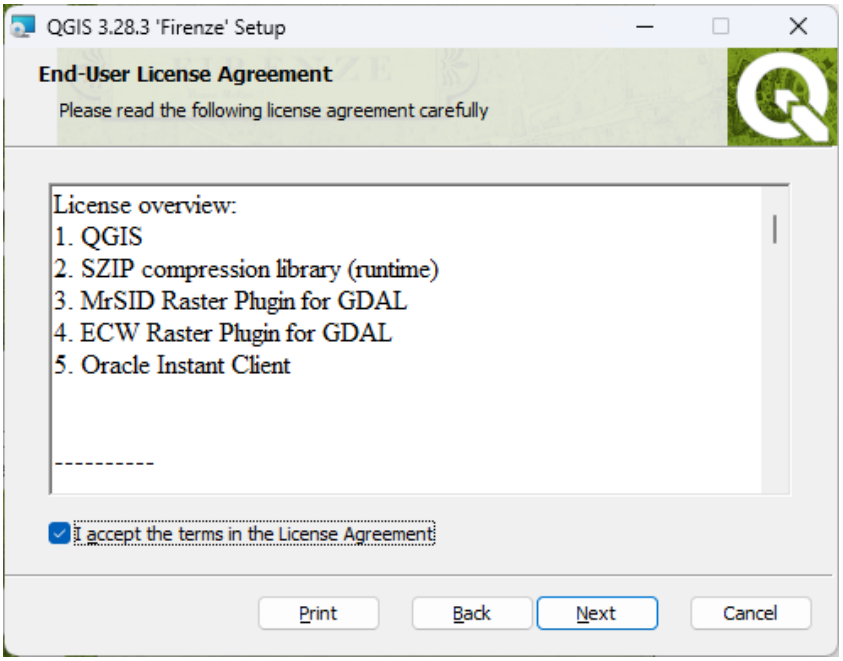


3. Quantum GIS (QGIS)

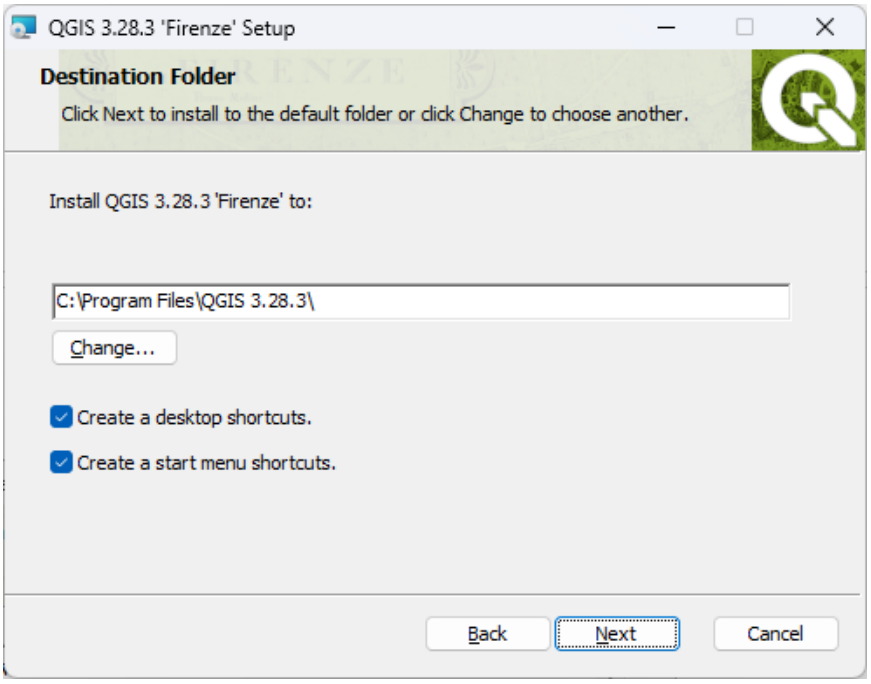
The installation procedure for the PostgreSQL server on a Windows operating system entails the following general steps:

1. Download the QGIS software from this link.
2. After you downloaded it, double click on the QGIS installation. Then the computer will compute your space requirements → click Next.

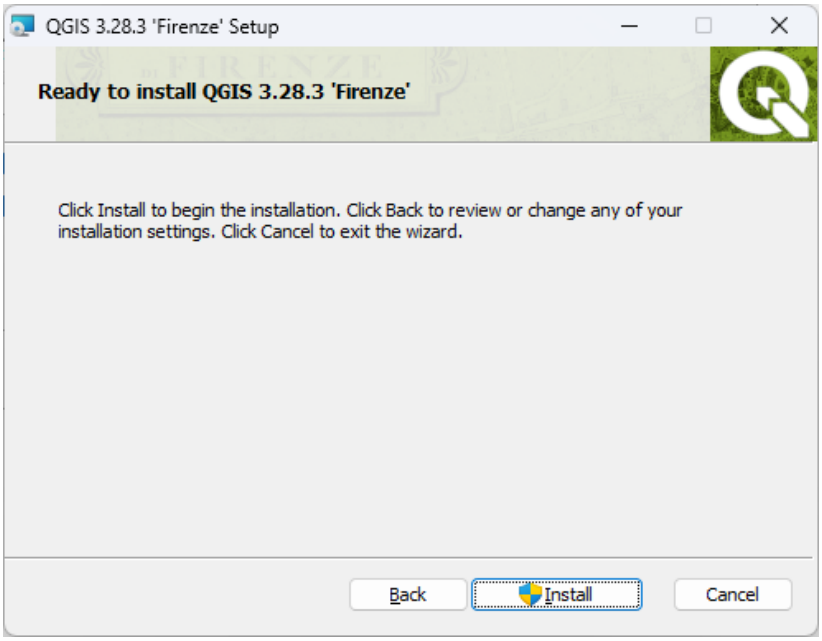
3. The next dialog is license overview → after you read it click accept the License Agreement box → Next.

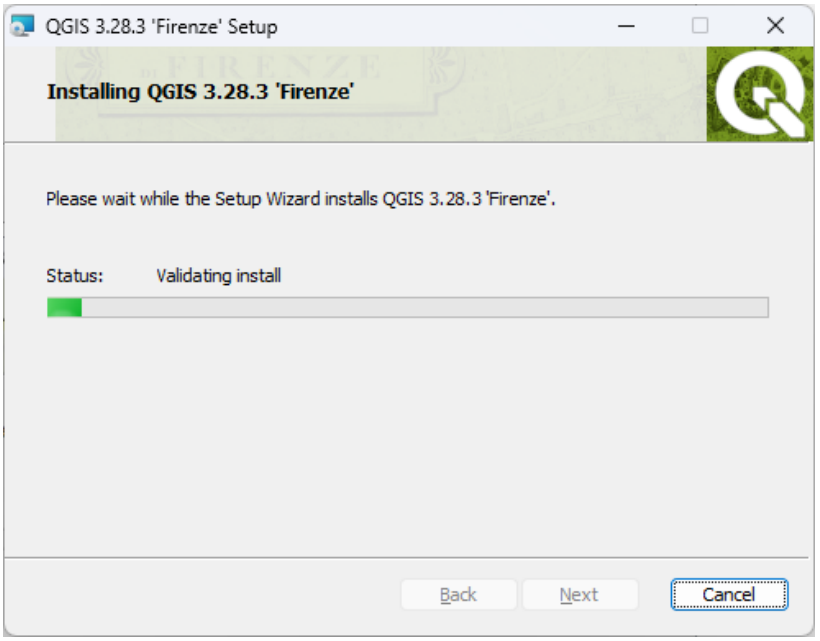


4. You can choose a custom folder path installation, the default folder path was saved in C:\Program Files.



5. Then QGIS is ready to install → click Install → please wait for a couple minutes until the program finished

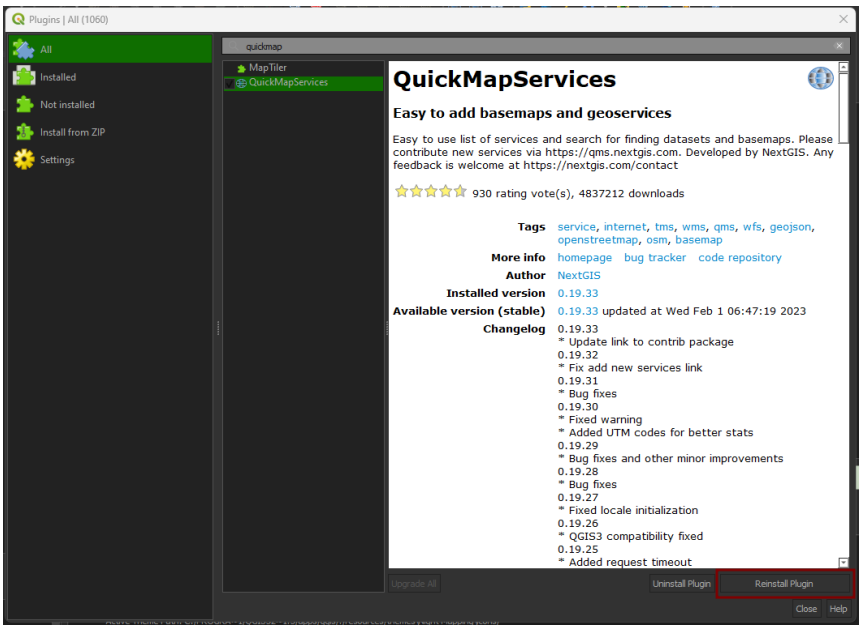




6. QGIS has been successfully installed and we are ready to use QGIS for data processing.

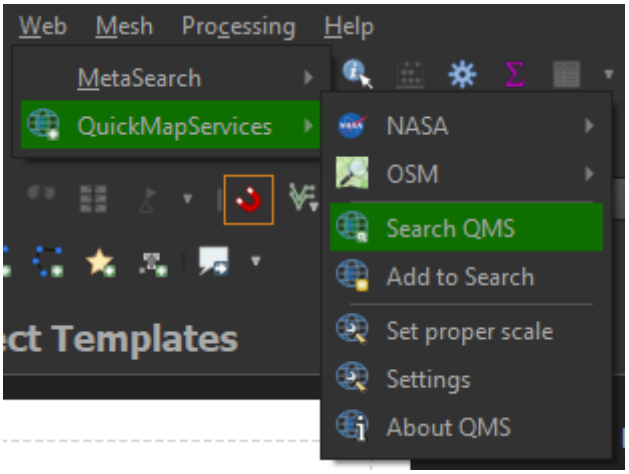
4. QGIS Plugins

1. QuickMapServices
   a. Open QGIS, click on the Plugins → Manage and Install Plugins..
   b. Select All → search QuickMapServices plugin → click Install Plugin.



c. The plugin will appear in the Web menu.

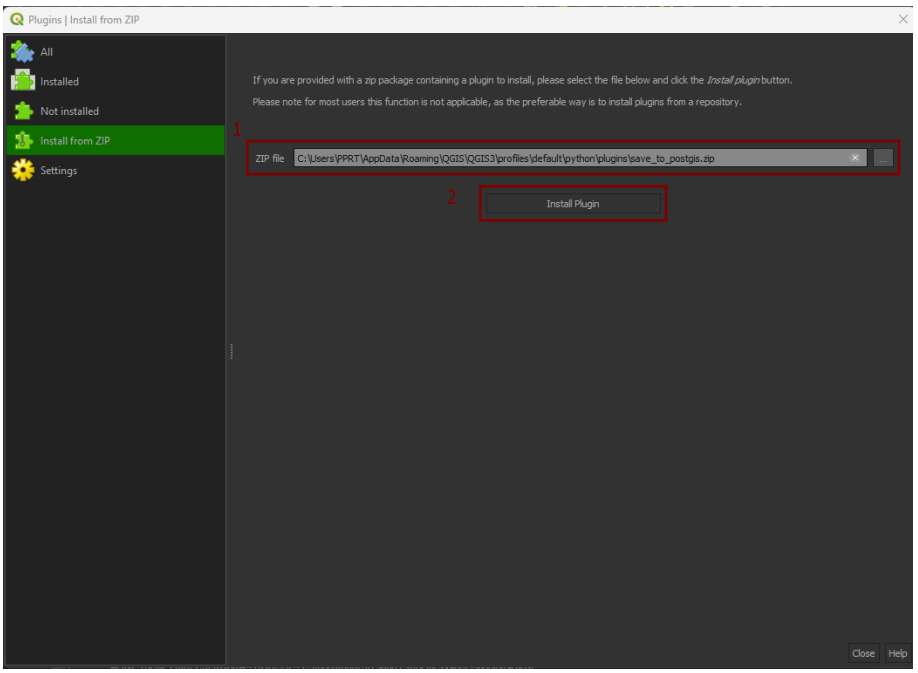
d. To add basemap services to the QGIS layer click Web → QuickMapServices → Search QMS → search Bing → click Add.



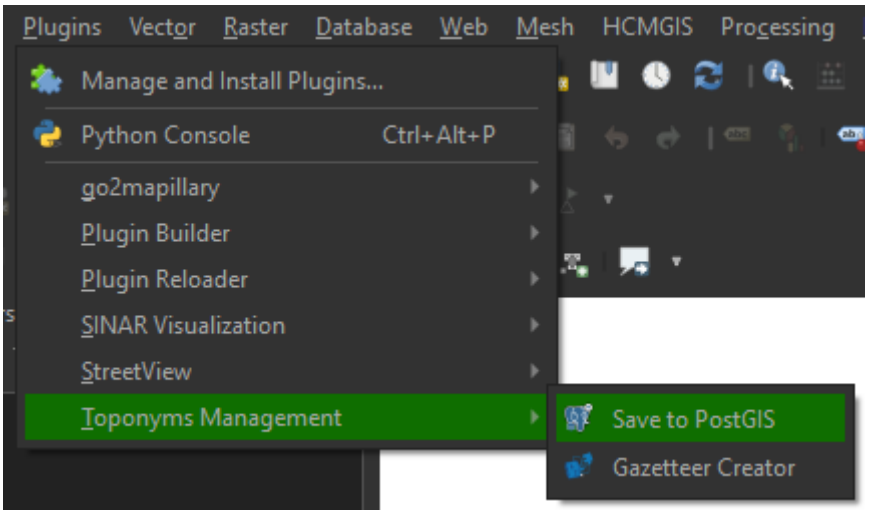
2. Save to PostGIS

a. Open QGIS, click on the Plugins → Manage and Install Plugins..

b. Select Install from ZIP → browse save_to_postgis.zip file → click Install Plugin.



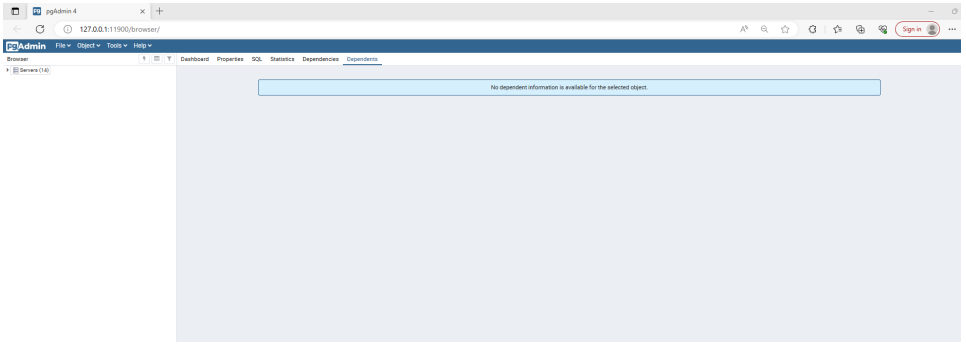
c. The plugin will appear in the Plugins menu.



3.3 Create Database

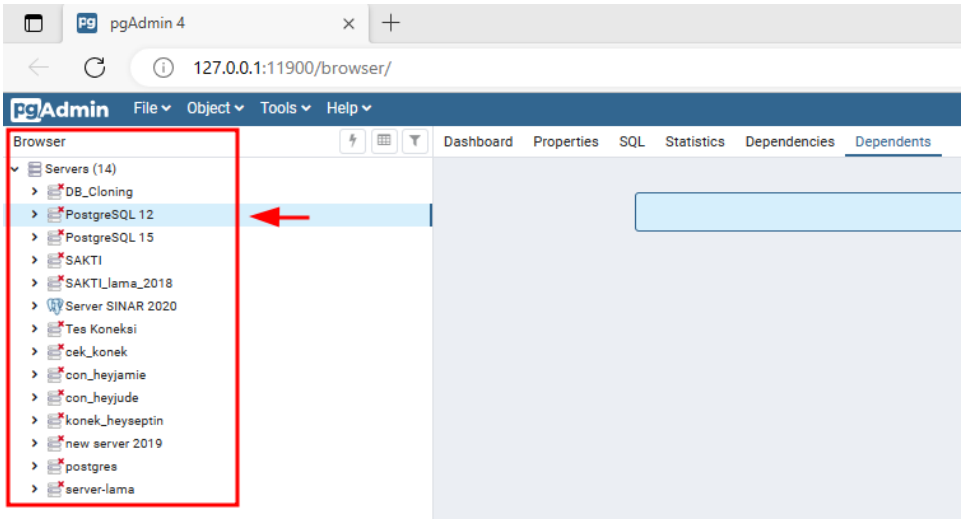
The database is created using PGAdmin, a popular software used for managing databases in PostgreSQL. PGAdmin provides features that enable users to perform tasks such as creating, editing, and deleting tables, managing users and access permissions, executing SQL queries, as well as viewing and analyzing data within the PostgreSQL database. The details procedure in creating a database is explained as follows:

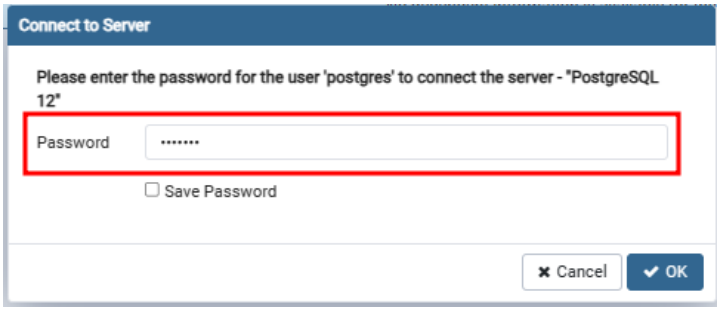
1. **Launch pgAdmin**



2. **Connect to a PostgreSQL server**

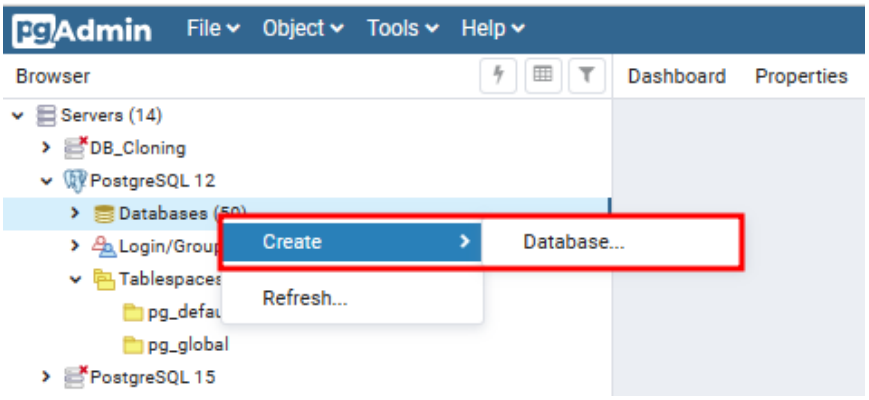
In the pgAdmin interface, expand the "Servers" group in the left sidebar. Locate and select the PostgreSQL server you want to work with. Enter your server credentials if prompted.



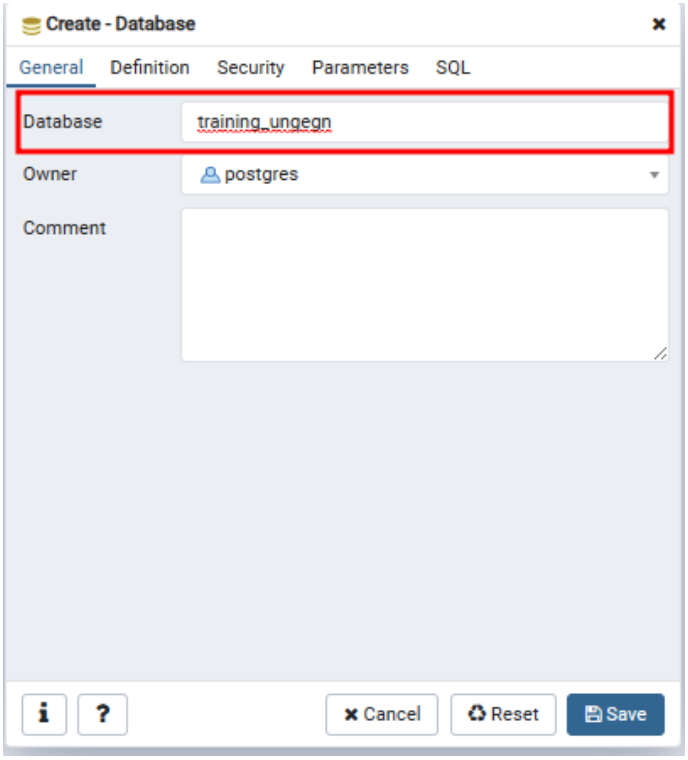


3. **Create a new database**

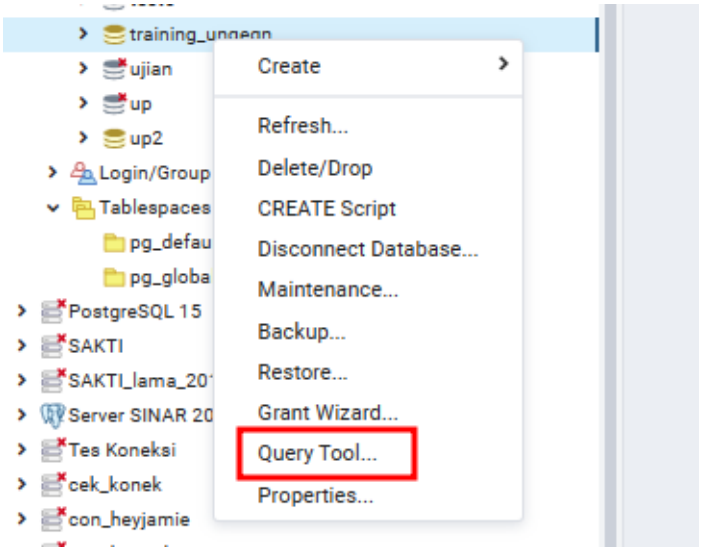
Right-click on the selected server and choose "Create" from the context menu. Then, select "Database" from the submenu.



4. **Configure the new database**
   Enter a name for your new database, for example *training_ungegn*



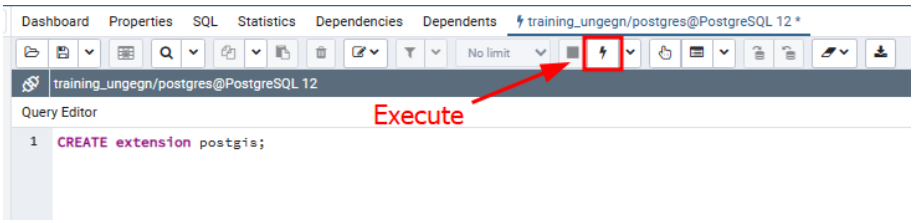
5. **Add PostGIS extension on the database**
   Right click on the database and choose *Query Tool*



Add postgis extension on the database by running SQL query on SQL editor below :

```
CREATE extension postgis;
```

Execute the script



3.4 Create Table

At this stage, the creation of a new table is carried out in the database that was created in the previous step. The table is created based on the following design as stated below:

Column Name	Mandatory (YES/NO)	TYPE	LENGTH
geom_type	Yes	TEXT	40
fclass	Yes	TEXT	40
ftype	Yes	TEXT	75
lat	Yes	Double	
long	Yes	Double	
remark	No	TEXT	254
prov_state	Yes	TEXT	254
regency	Yes	TEXT	254

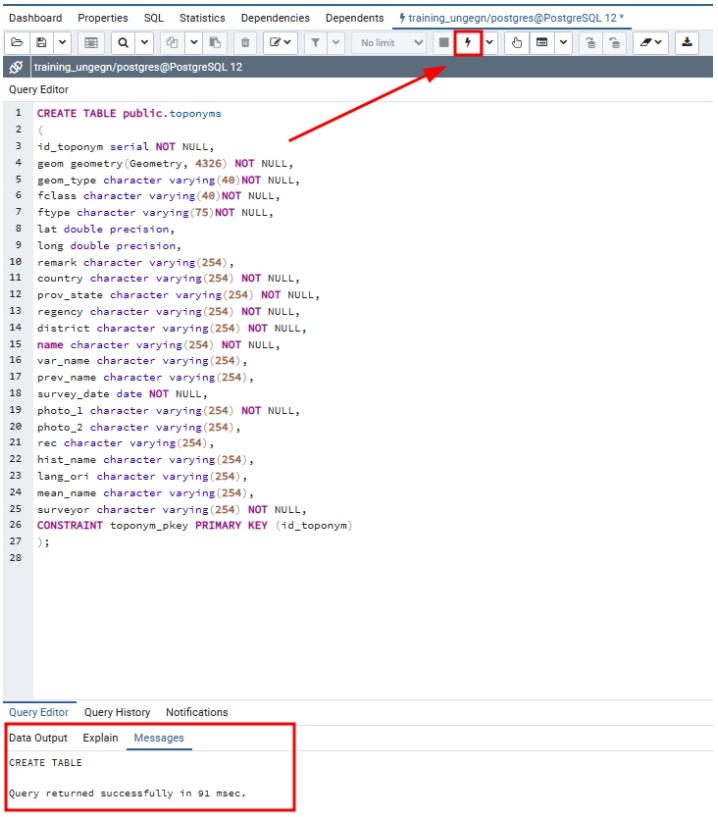
district	Yes	TEXT	254
name	Yes	TEXT	254
var_name	No	TEXT	254
prev_name	No	TEXT	254
survey_date	Yes	Date	
photo_1	Yes	TEXT	254
photo_2	No	TEXT	254
rec	No	TEXT	254
hist_name	No	TEXT	254
lang_ori	No	TEXT	254

To create a table in PostgreSQL, we can use SQL scripts or utilize the user interface (UI) in pgAdmin. In this training, the table creation is done using SQL scripts. The table mentioned above is translated into the SQL script below to generate toponyms table :

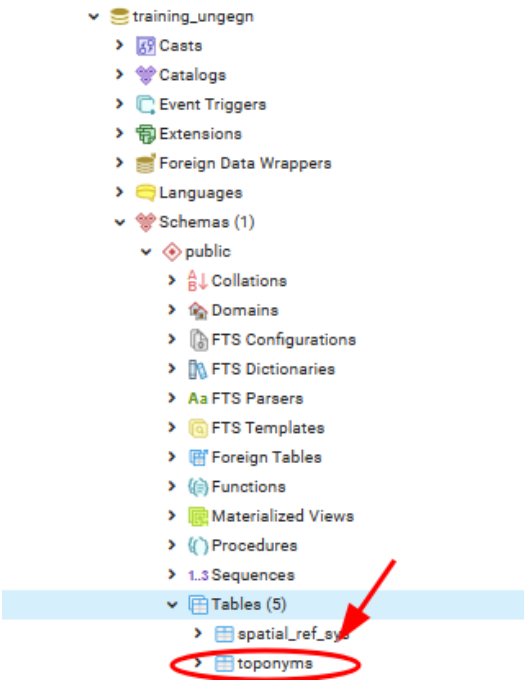
```
CREATE TABLE public.toponyms
(
id_toponym serial NOT NULL,
geom geometry(Geometry, 4326) NOT NULL,
geom_type character varying(40)NOT NULL,
fclass character varying(40)NOT NULL,
ftype character varying(75)NOT NULL,
lat double precision NOT NULL,
long double precision NOT NULL,
remark character varying(254),
prov_state character varying(254) NOT NULL,
regency character varying(254) NOT NULL,
district character varying(254) NOT NULL,
name character varying(254) NOT NULL,
var_name character varying(254),
prev_name character varying(254),
survey_date date NOT NULL,
photo_1 character varying(254) NOT NULL,
photo_2 character varying(254),
rec character varying(254),
hist_name character varying(254),
lang_ori character varying(254),
mean_name character varying(254),
surveyor character varying(254) NOT NULL,
CONSTRAINT toponym_pkey PRIMARY KEY (id_toponym)
);
```

The steps in creating toponyms table are explained as follows:

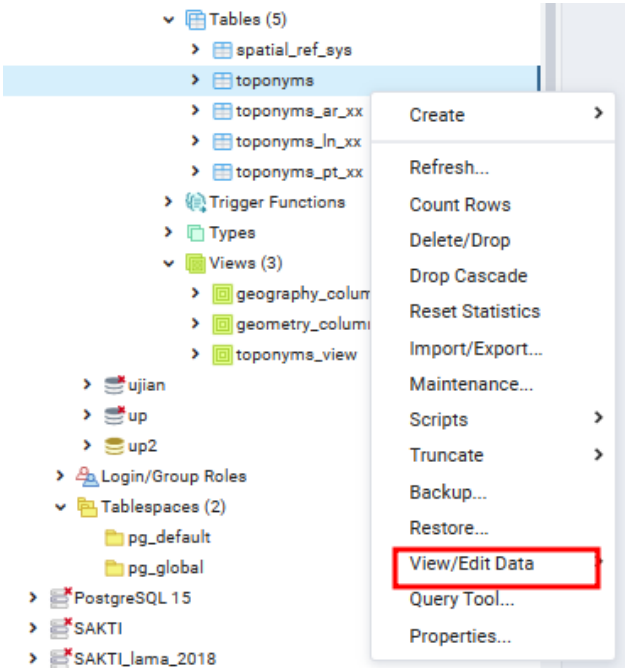
a. Run the script above in the SQL Editor until the successful message appear



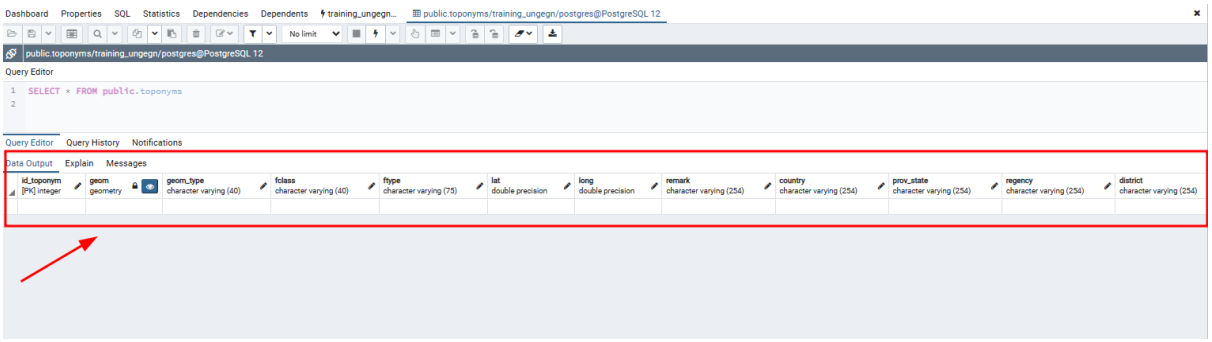
b. Access the toponyms within the training_ungegn database by expanding to the "Schema" node and locating the "Public" Schema. Expand the "Public" Schema and the toponyms table will be found in this node.



c. To view the table, right-click on the toponyms table and select "View/Edit Data" from the context menu. This will open a new window displaying the table data



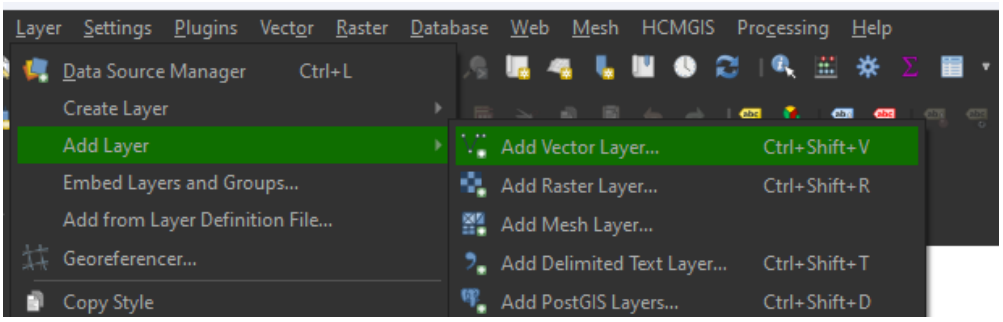
An empty table will be displayed in the data output section including the column name and the column type. The toponyms table contains several columns, primary key as the unique identifier in which id_toponym column, and geometry column named geom. In the next section of training, the participant will input data to this table.



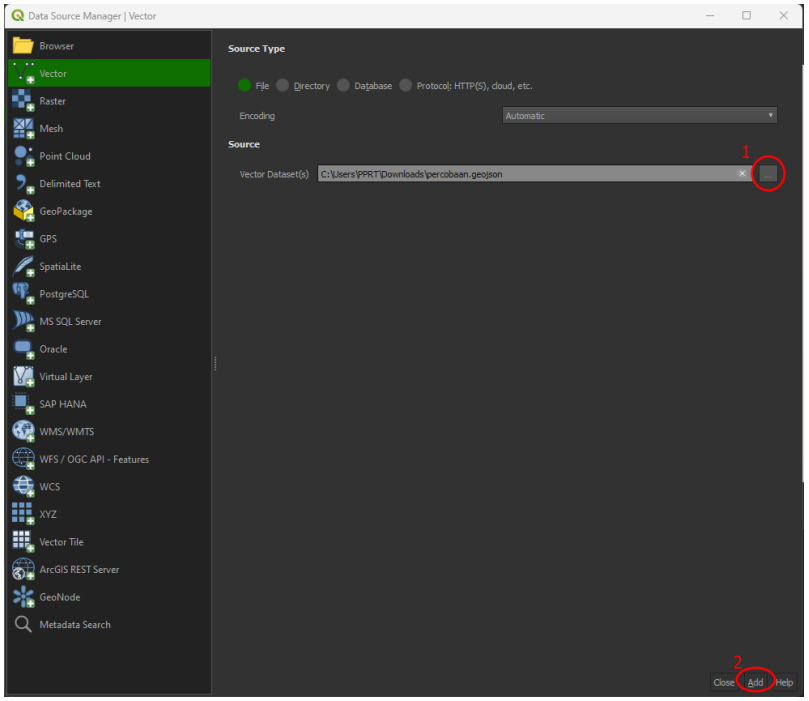
3.5 Data Processing

In this part, we will add data from the field survey and process the data, which includes spatial data processing and attribute data processing. Therefore, we must check the spatial data obtained from the field survey. For the attribute data, our focus on checking the mandatory columns are not missing, verifying that the data type of mandatory columns are correct, and confirming the values of columns are not null.

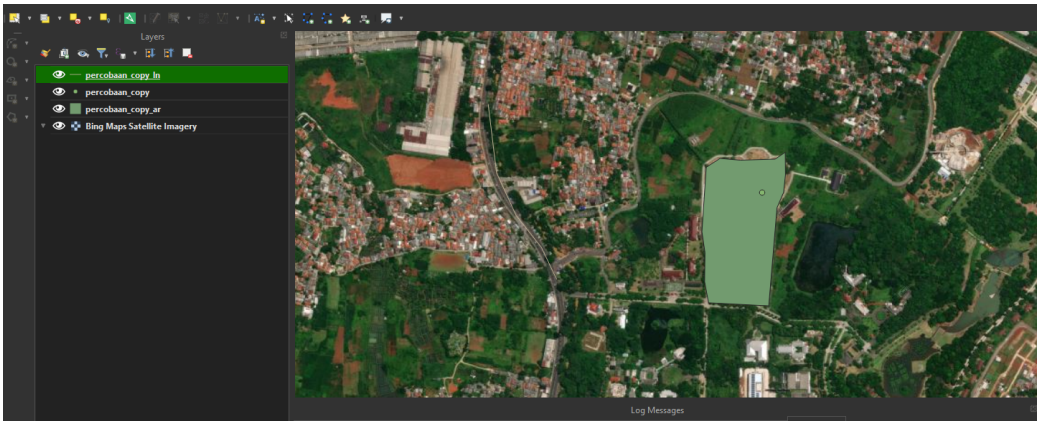
1. Add the data to QGIS from Layer Menu → Add Layer → Add Vector Layer :



Then QGIS will show the dialog to ask the filepath of the vector data (read: field survey data) on the Vector Dataset(s) row → Click Add.



The data will show in the QGIS layer.



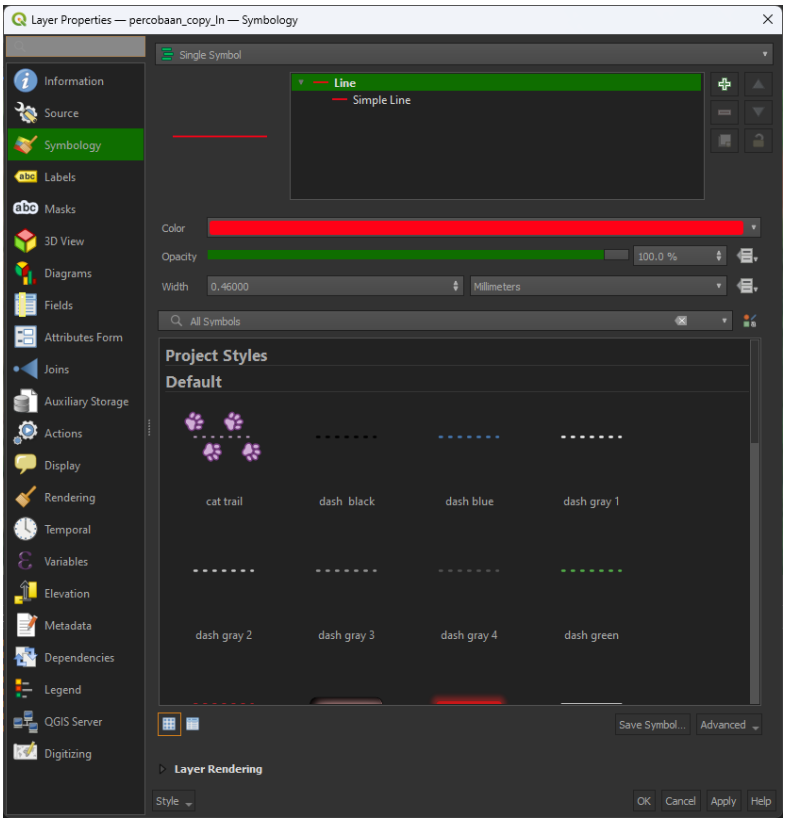
2. Sometimes a lot of data needs to be corrected in the office which can be challenging to processing in the field. The processing can be divided into two types: spatial data processing and attribute data processing.

   a. Spatial Data Processing

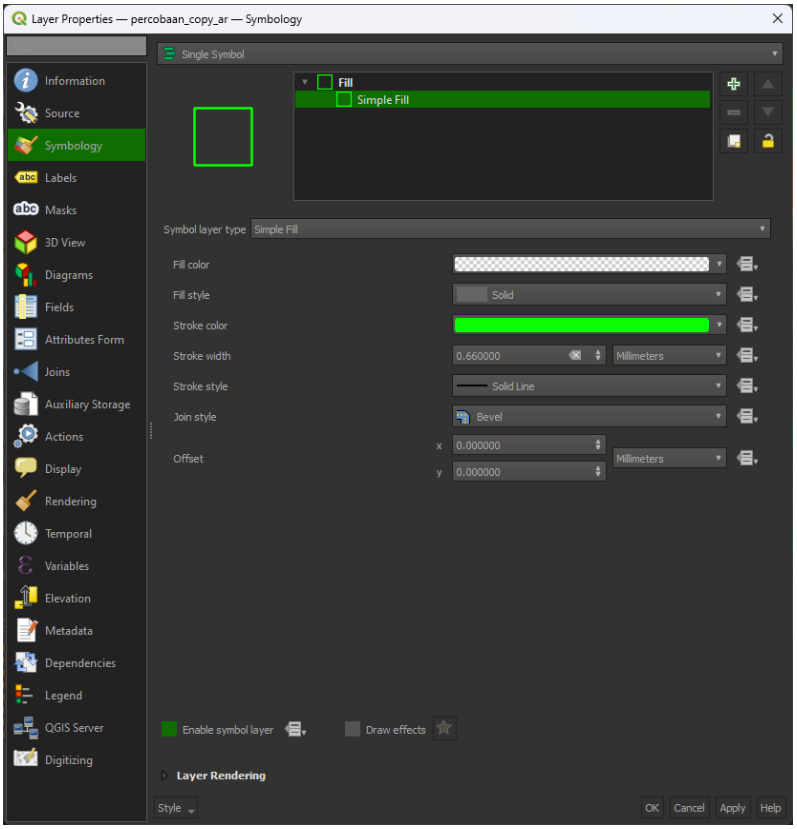
      In this part, we will edit the spatial data, such as repositioning/changing the geometry of the data, because it's very difficult to reposition in the field.

      i. Setting the Symbology

         Before we edit spatial data, we must set the symbology of the data. It means we set the color of the data to easily see the data by right clicking on the layer → Properties → Symbology.

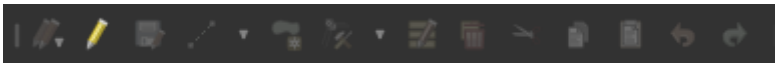


         Then choose the color on the right box → OK. For polygon data, you can click simple fill then choose for fill color and stroke color (outside color).

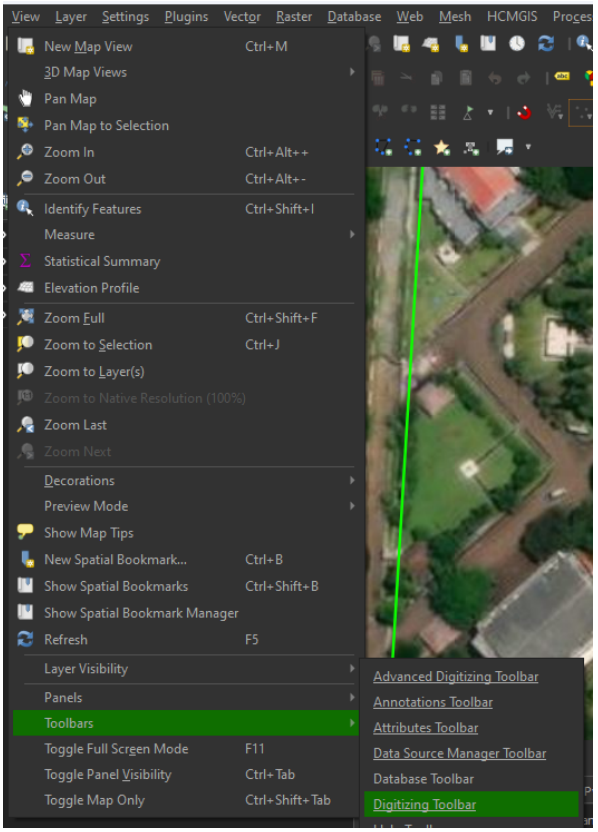


ii. Edit Spatial Vertex Editor

Please ensure that the digitizing toolbar has been added to your QGIS interface.



If the digitizing toolbar is not visible in your QGIS interface, you can enable it by going to the View Menu → Toolbars → Digitizing Toolbar. This will display the digitizing toolbar in your QGIS interface.



After the digitizing toolbar display in your QGIS interface, follow these steps to Start Editing the spatial data:

- Click the Start Editing icon 
- Then click Select Features icon  (on the Selection Toolbar), you can select the data that you want to edit.
- To reposition/editing the geometry of the data, you can click Vertex Tool icon  (on Digitizing Toolbar).
- To edit the point data you can click the point that you have selected.

To move the point data to the correct position in QGIS, follow these steps:

- Move your mouse cursor to the desired position until the point data icon resembles an "X".
- Click on the buildings or points that you want to move.
- You will see that the point data has now been successfully moved to the new position.

To editing geometry of line or polygon data, follow these steps:

- Click the vertex. The vertex is the point/node in line or polygon data to connect a line or polygon. In QGIS, it is presented with a red “X” icon.

- After you click the vertex, if you want to delete the vertex just click “Delete” on the keyboard. If you want to move it, then move your mouse to the desired position. If you want to add the vertex, there is “+” icon in the middle of two vertices.
- You will see that the vertex data has now been successfully moved to the new position.

- After you editing the spatial data to the truth geometry, then save it by click Save Layer Edits icon 
- Don’t forget to click Toggle Editing icon  , to stop editing of the data.

b. Attribute Data Processing

In this part, we will edit the attribute data especially to check the mandatory columns name, mandatory columns data type, and the values of the mandatory columns. There are differences in attribute data processing collected using KoboToolbox and EpiCollect based on the column names and data types that are generated by their respective systems. Therefore we will discuss both:

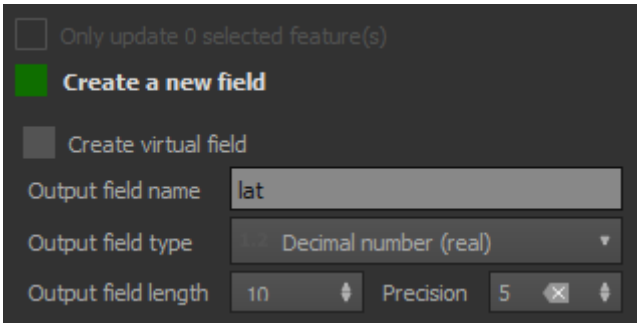
1. Data Collection using KoboToolbox

   Processing the data collected by KoboToolbox is particularly challenging due to being unable to get latitude and longitude data values as decimal numbers that depend on mandatory columns. Then, we must check the appropriate values for the mandatory columns, like photo_1 and photo_1_URL.

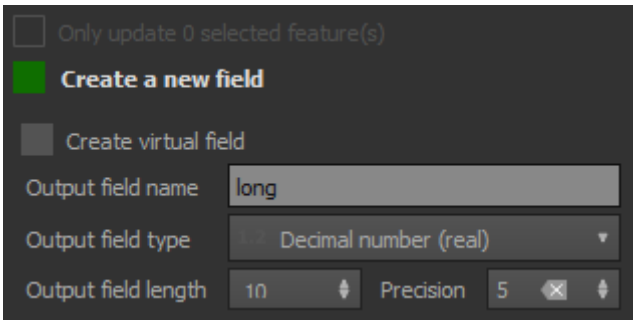
   a. Processing the lat and long columns

We can get the values of latitude and longitude as decimal numbers from the QGIS field calculator. To add lat and long columns, follow these steps:

- Right click on the layer → Open Attribute Table
- Click Open field calculator icon 
- To add new field you can check the box of “Create new field”, then fill the output field with the following steps:
  - For lat columns
    - Output field name : lat
    - Output field type : Decimal number (real)
    - Output field length : 10
    - Precision : 5

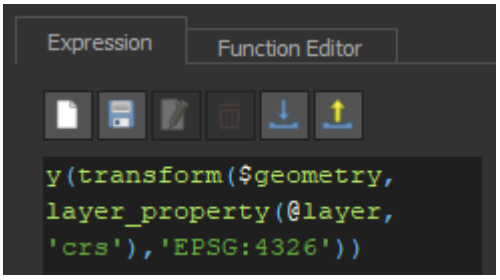


  - For long columns
    - Output field name : long
    - Output field type : Decimal number (real)
    - Output field length : 10
    - Precision : 5



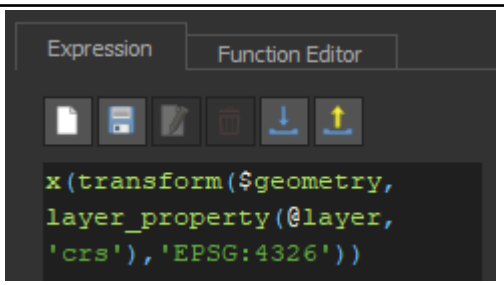
- Fill the expression box with the following scripts:
  - For point data
    - For lat columns

```
y(transform($geometry, layer_property(@layer, 'crs'),'EPSG:4326'))
```



- For long columns

```
x(transform($geometry, layer_property(@layer,
'crs'),'EPSG:4326'))
```

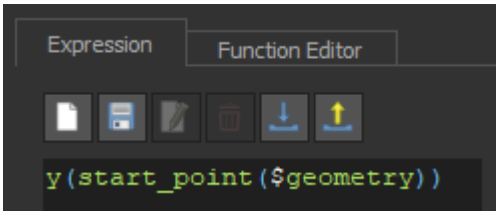


- For line data

  In this case, we will calculate latitude and longitude values of line data from the first coordinate of the vertex line.

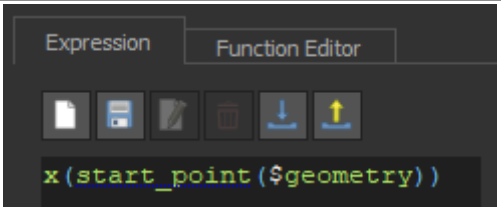
  - For lat columns

```
y(start_point($geometry))
```



  - For long columns

```
x(start_point($geometry))
```

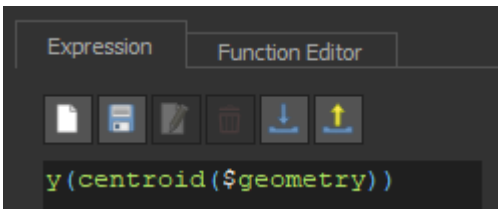


- For polygon data

  In this case, we will calculate latitude and longitude values of polygon data from the centroid of the polygon.

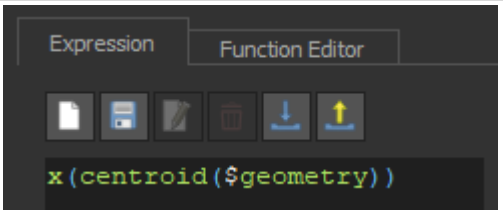
  - For lat columns

```
y(centroid($geometry))
```



- For long columns

```
x(centroid($geometry))
```

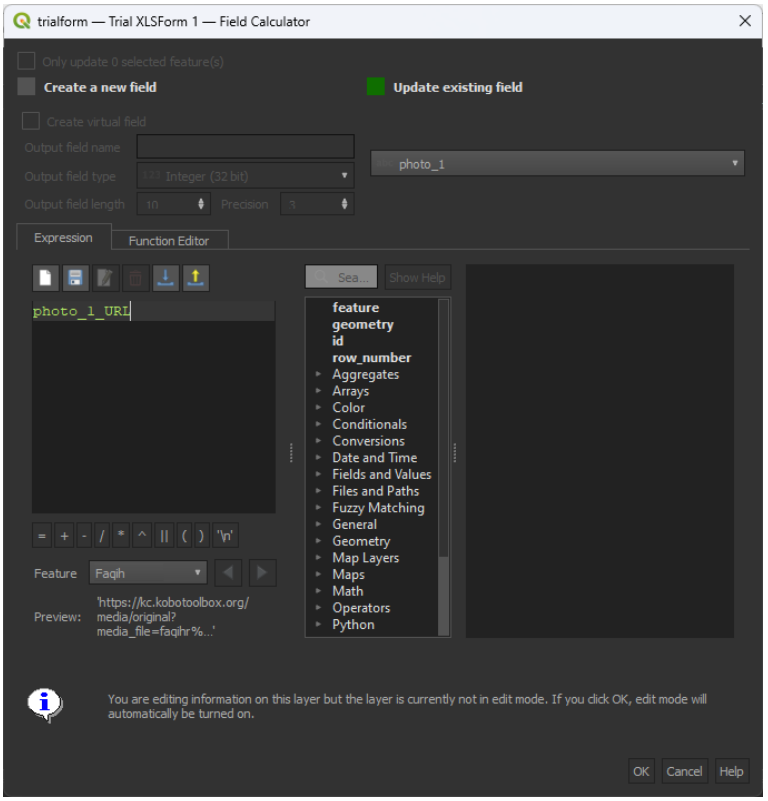


- Click OK.

b. Processing the photo_1 and photo_2 columns

If we see the values of the photo_1 and photo_2 columns, there is a filename not linked to the photo web page. But we need the photo web page not a filename and photo_1_URL/photo_2_URL wouldn't be saved in PostGIS because there are no photo_1_URL/photo_2_URL columns in the PostGIS table. Therefore, we must update the values of photo_1 and photo_2 to the URL. To update the photo_1 and photo_2 values, follow these steps:

- Right click on the layer → Open Attribute Table
- Click Open field calculator icon 
- Check Update existing field, choose photo_1 column. In the expression box, type the photo_1_URL/photo_2_URL column. Therefore, the form of a field calculator will be like this.



- Click OK

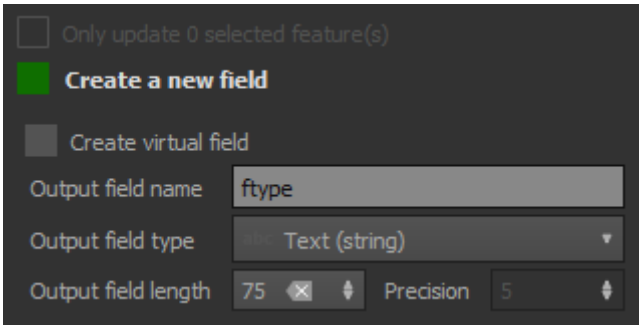
2. Data Collection using EpiCollect

   Processing the data collected by EpiCollect is particularly challenging due to the presence of various ftype columns. Additionally, the date type column which is represented as text in EpiCollect, needs to be converted to the appropriate date format in the database.

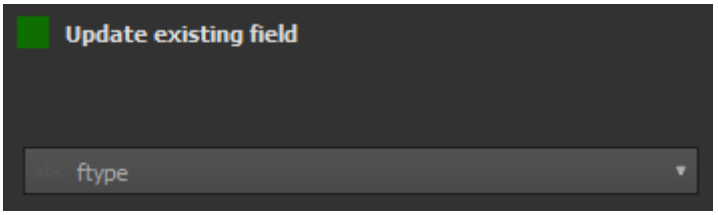
   a. Processing the ftype columns

      If we see the data structure from epicollect, especially for ftype, there are many ftype columns to represent ftype like ftype_adm, ftype_hydro, etc. Therefore we need, to update (if ftype column exist) or we create a ftype column (if ftype column not exist) , follow these steps:

      - Right click on the layer → Open Attribute Table
      - Click Open field calculator icon 
      - To add new field you can check the box of “Create new field”, then fill the output field with the following steps:
        - Output field name : ftype
        - Output field type : Text (string)
        - Output field length : 75

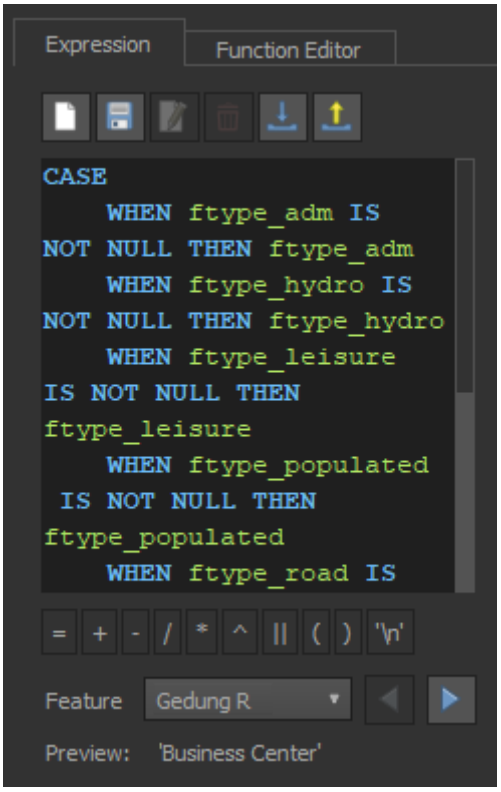


If ftype column is exist, check the box of “Update existing field” then choose ftype column below it.



- Fill the expression box with the following scripts.

```
CASE
      WHEN ftype_adm IS NOT NULL THEN ftype_adm
      WHEN ftype_hydro IS NOT NULL THEN ftype_hydro
      WHEN ftype_leisure IS NOT NULL THEN
ftype_leisure
      WHEN ftype_populated IS NOT NULL THEN
ftype_populated
      WHEN ftype_road IS NOT NULL THEN ftype_road
      WHEN ftype_build IS NOT NULL THEN ftype_build
      WHEN ftype_topo IS NOT NULL THEN ftype_topo
    WHEN ftype_undersea IS NOT NULL THEN
ftype_undersea
      WHEN ftype_vege IS NOT NULL THEN ftype_vege
END
```

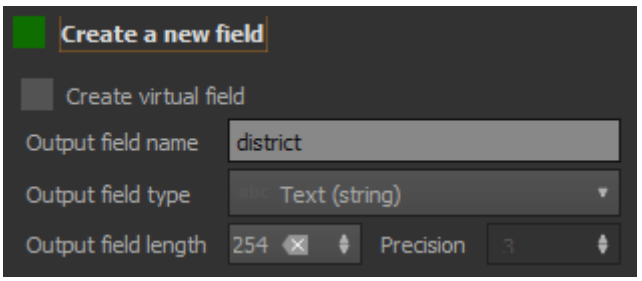


The result from this script is that the following various ftype columns are not empty.

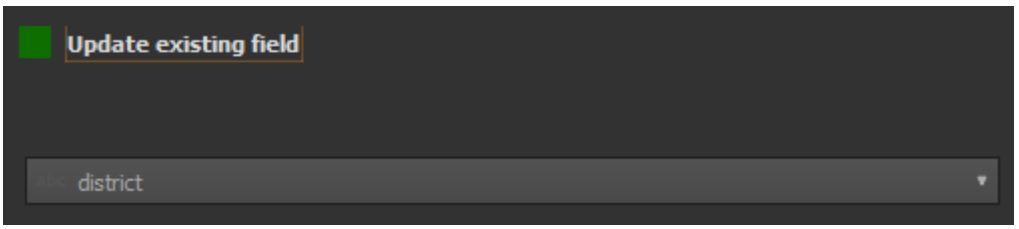
b. Processing the district columns

If we see the data structure from epicollect, especially for districts, there are many district columns to represent districts like district_gianyar, district_klungkung, etc. Therefore we need to update (if district column exist) or we create a district column (if district column not exist) , follow these steps:

- Right click on the layer → Open Attribute Table
- Click Open field calculator icon 
- To add new field you can check the box of “Create new field”, then fill the output field with the following steps:
  - Output field name : district
  - Output field type : Text (string)
  - Output field length : 254

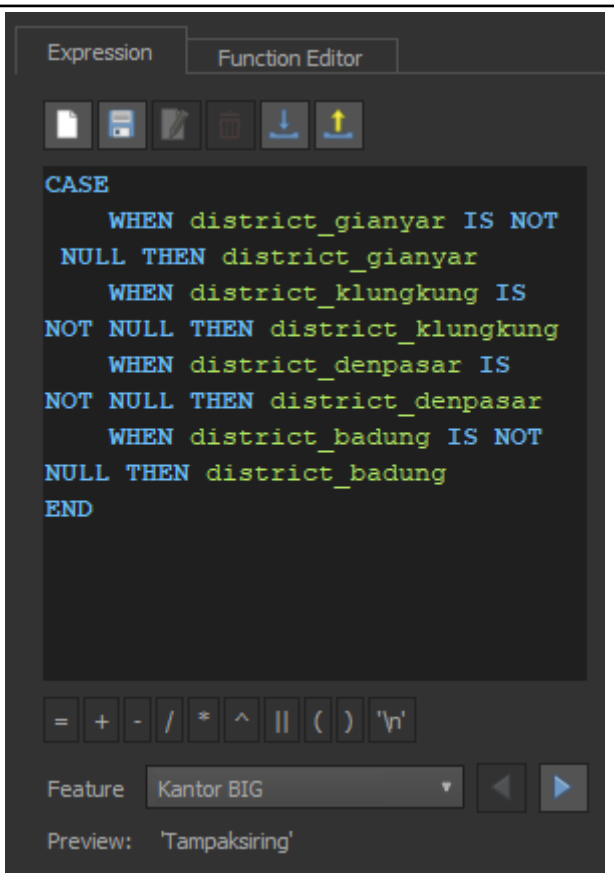


If a district column exists, check the box of “Update existing field” then choose the district column below it.



- Fill the expression box with the following scripts.

```
CASE
      WHEN district_gianyar IS NOT NULL THEN
district_gianyar
      WHEN district_klungkung IS NOT NULL THEN
district_klungkung
      WHEN district_denpasar IS NOT NULL THEN
district_denpasar
      WHEN district_badung IS NOT NULL THEN
district_badung
END
```



The result from this script is that the following various district columns are not empty.

c. Processing the survey_date columns

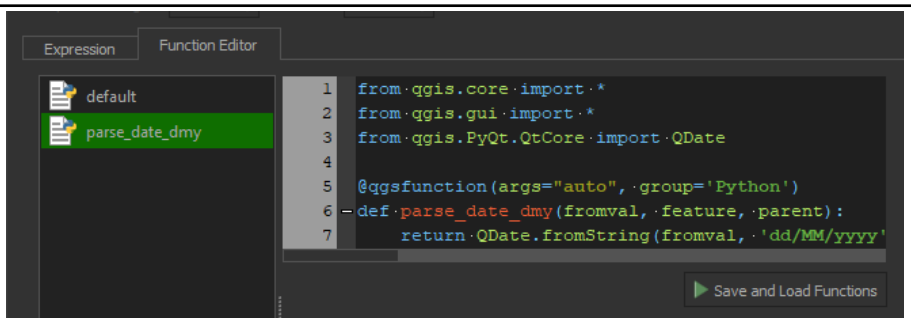
If we see the date columns from right click on the layer → Properties → Fields. The date column is a text, but in the database is a date. Therefore, we need survey_date column with date type with the following steps:

- Right click on the layer → Open Attribute Table
- Click Open field calculator icon 
- To add new field you can check the box of “Create new field”, then fill the output field with the following steps:
  - Output field name : survey_date
  - Output field type : Date
- As we can see from this link discussion, from the function editor box we can add by clicking the green “+” icon and typing “parse_date_dmy” to name it. Then copy this script and paste on the right box → click Save and Load Functions.

  You can customize the date format appropriate to your date column from the red script.

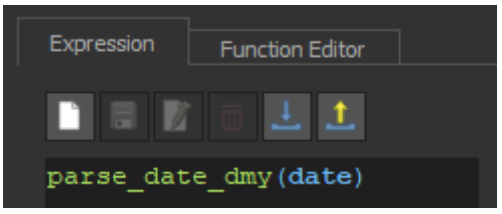
```
from qgis.core import *
from qgis.gui import *
from qgis.PyQt.QtCore import QDate

@qgsfunction(args="auto", group='Python')
def parse_date_dmy(fromval, feature, parent):
    return QDate.fromString(fromval, 'dd/MM/yyyy')
```



- Fill the expression box with the following script → click OK.

```
parse_date_dmy(date)
```



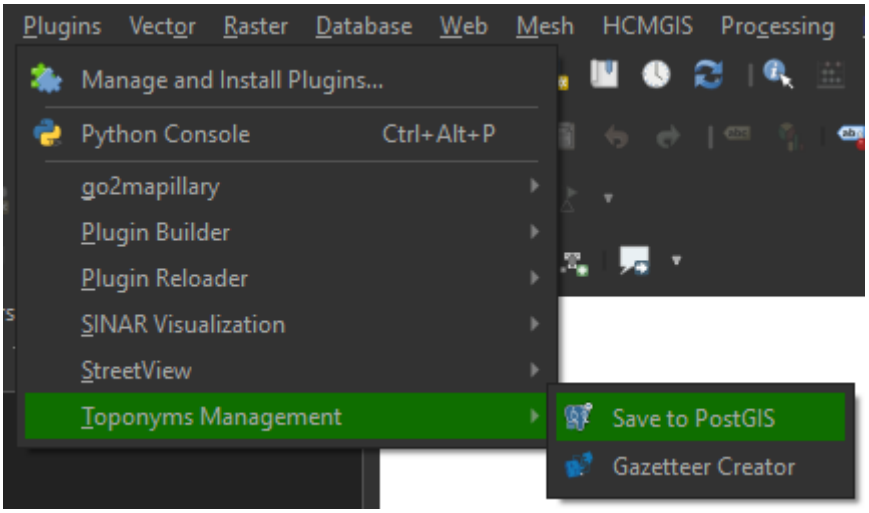
It’s an example of data processing from EpiCollect and KoboToolbox, maybe we have some data collection problems for other applications. At least, we have learned

the basics of data processing including spatial data processing and attribute data processing.

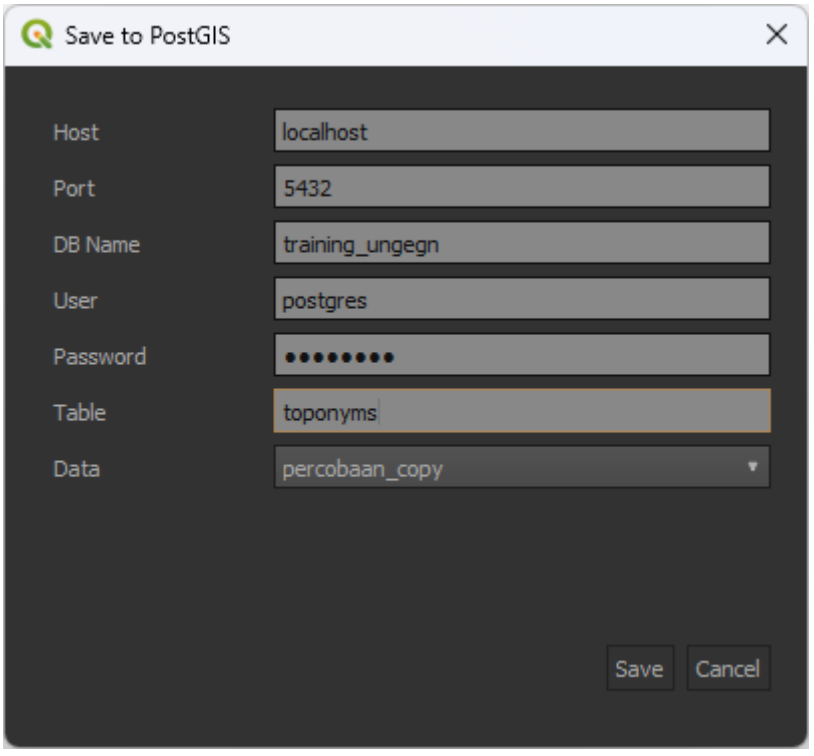
3.6 Save to PostGIS

In this part, we will save the data that we have processed to PostGIS. Therefore, others can see our data from the database by database credential. To Save the data to the PostGIS, follow these steps:

1. Open Save to PostGIS plugin by click Plugins Menu → Toponyms Management → Save to PostGIS



2. Fill the forms that include the credential of the database that you have created the table → click Save.



a. Host : Host of the database. If you want to save on your local computer, the host value is usually “localhost”. If you want to save on the server, sometimes its value is a number.

b. Port : Port of the database. The default value is 5432, maybe it has a different number depending on your database server.

c. DB Name : Database name. In this training, we have set it to “training_ungegn”. It depends on the database creation name.

d. User : User of the database. The default value is “postgres”, maybe you have changed it depending on your database credential.

e. Password : Password of the database. It depends on your password in the database.

f. Table : Table name in the database that you want to save the data. In this training we have set it to “toponyms”.

g. Data The vector data on the QGIS layer that is ready to save in PostGIS.

3. Successful, Info, and Warning Messages in QGIS Message

After you click Save on the “Save to PostGIS” plugin, it will return messages. The messages are divided into two categories. The categories are successful and warning messages.

a. Successful messages.

This message will show when your data is perfect and your data has been saved in PostGIS.

2023-05-29T13:57:33 SUCCESS Success : Your data has been successfully saved in PostGIS!

b. Info messages.

This message will show when the optional columns are missing in your data. The data would be saved in PostGIS when all of the mandatory columns exist. But if the mandatory columns are missing, it wouldn’t saved in PostGIS

2023-05-29T14:15:33 INFO Info : The following optional columns are missing in the layer data : rec.

c. Warning messages.

The warning messages will raise python exception errors. There are 3 categories of warning messages when the plugin found the mandatory columns are not appropriate in the database. Inappropriate mandatory columns are divided based on :

1. Mandatory columns are missing

   The plugin will detect the selected data for the availability of mandatory columns in the database to your data. The plugin will raise an exception python error when it finds the mandatory columns are missing. This is an example of the message when the mandatory columns are missing.

   > 2023-05-29T14:24:13 WARNING Warning : The following mandatory columns are missing in the layer data : ftype, geom_type.

2. Mandatory columns data type is not appropriate

   The plugin will detect the selected data for a data type of mandatory columns in the database to your data. The plugin will raise an exception python error when it found the data type of mandatory columns is not appropriate with the data type in the database. This is an example of the message when the data type of mandatory columns is not appropriate.

   > 2023-05-29T14:25:43 WARNING Warning : Please check the following mandatory columns data type : ftype, survey_date. The data types are different with the database!

3. NULL values in mandatory columns

   The plugin will detect the values of selected data. The plugin ensures that the mandatory column values are not null. It's because we have set the mandatory columns that are not null in the database. This is an example of the message when the plugin detects NULL values of the data.

   > 2023-05-29T14:25:43 WARNING Warning : Please check the following mandatory columns value : geom_type. There are NULL values!

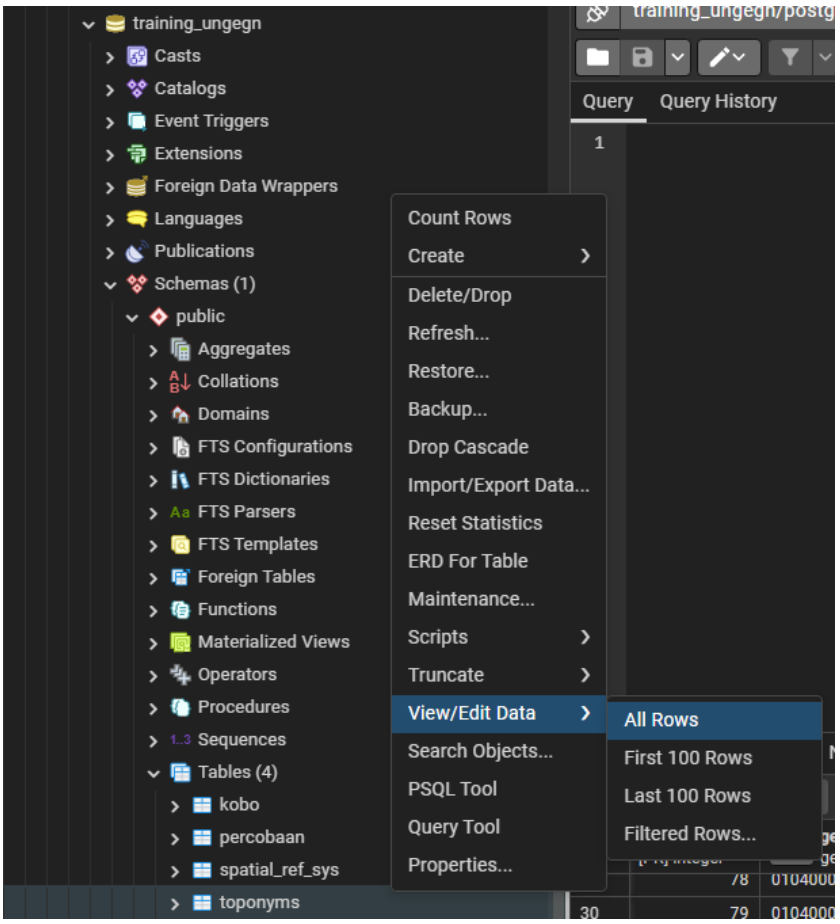
The warnings serve as a reminder that if the mandatory columns are not appropriately set, the data will not be saved in PostGIS. Therefore, it's crucial to thoroughly check the mandatory columns to ensure the data can be successfully stored in PostGIS. Here are the table guidelines for the mandatory columns.

Column Name	Data Type	Character Maximum Length
geom_type	character varying/text/string	40
fclass	character varying/text/string	40

ftype	character varying/text/string	75
lat	double precision/real	-
long	double precision/real	-
prov_state	character varying/text/string	254
regency	character varying/text/string	254
district	character varying/text/string	254
name	character varying/text/string	254
survey_date	date	-
photo_1	character varying/text/string	254
surveyor	character varying/text/string	254

4. View Data in PostGIS

   To ensure the data has been successfully saved in PostGIS, you can open pgAdmin → input your master password → expand the “training_ungegn” database by username and password → expand Schemas → expand public → expand Tables → right click on toponyms → View/Edit Data → click All Rows.



Therefore, if the data has been successful, we can see the table will add a new row marked by id_toponym row number addition.