



# 3DCityDB Tools

*for*

# QGIS

## Installation and user guide

Version 0.8.8

Last update: 10 July 2024

# License

This presentation is licensed under the [Creative Commons License CC BY-NC-SA 4.0](#). According to CC BY-NC-SA 4.0 permission is granted to share this document, i.e. copy and redistribute the material in any medium or format, and to adapt it, i.e. remix, transform, and build upon the material under the following conditions:



- **Attribution:** You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- **NonCommercial:** You may not use the material for commercial purposes.
- **ShareAlike:** If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.
- **No additional restrictions:** You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

Resources

Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

Resources

# Motivation

## Motivation

Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

### IDEA:

- **CityGML 3D City Database:** Why not letting users benefit from *directly* working with the 3DCityDB?
  - No need to work with files
  - Editing of features attributes could become way easier
  - SQL/relational model are rather well-known in and outside the GIS user community
  - Last but not least.... "3D city models belong best in a database" ☺



# Motivation

BUT:

- 3DCityDB structure is rather complex
  - Lots of nested tables, intricate structure
  - Data management is difficult, although some functions are provided (e.g. delete functions)
  - There can be multiple citydb schemas in the same database instance (aka "scenarios")
- CityGML does not follow the Simple Feature for SQL model (SFS)
  - Nested features
  - One feature can have multiple representations (multiple LoDs, multiple geometry types)
- The existing **Importer/Exporter** offers some functionalities, but its *raison d'être* is basically different (...as the name says!)

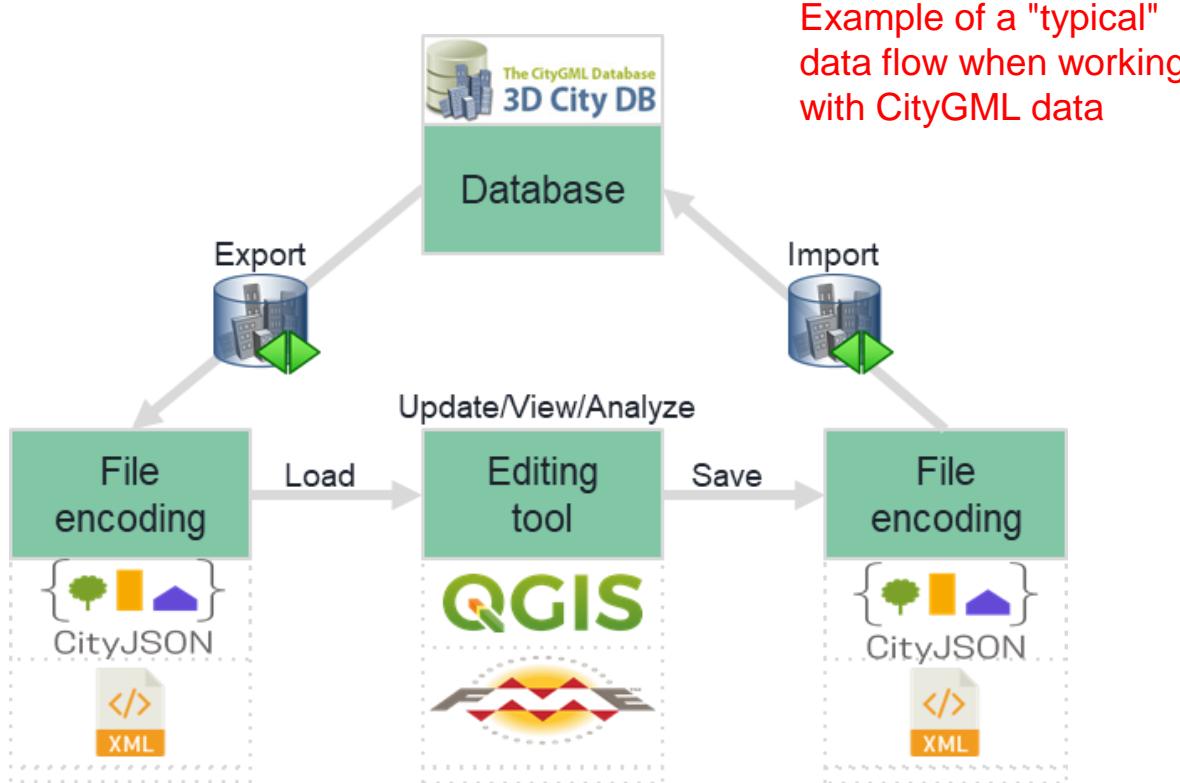
# Motivation

SO:

- Why not using **QGIS**?
  - Well-known and established open-source software
  - Rather mature, version 3.28 LTR released in autumn 2022, well documented
  - Native support for PostgreSQL/PostGIS, and for Oracle Spatial
  - Has strong 2D and some (less mature) 3D visualisation functionalities
  - Can be extended with Python-based plugins

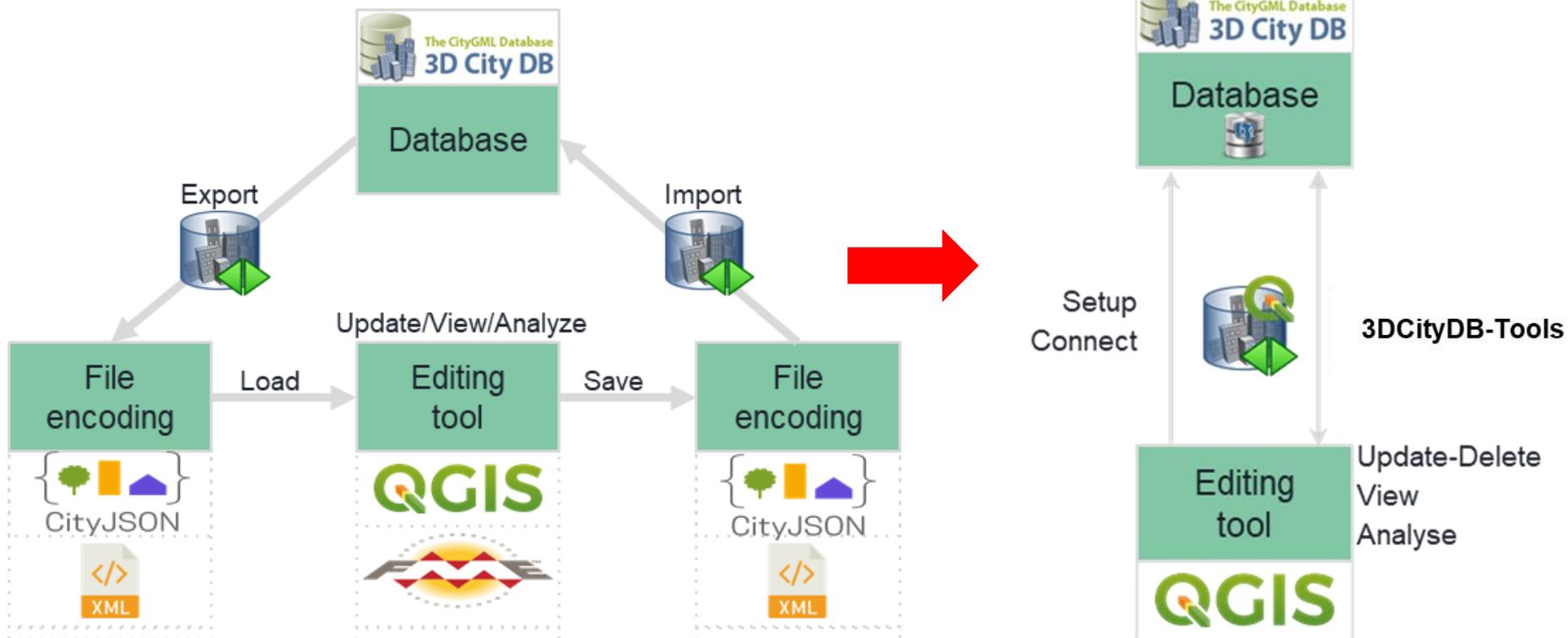


# Motivation



# Motivation

Vision / goal of the plugin



Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

Resources

# Plugin overview

## Main functionalities

Motivation  
**Plugin overview**  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

- Create "**SFS-like layers**" to hide 3DCityDB complexity when interacting with data
  - Deal efficiently with multi-LoD / different geometries / implicit representations
    - Up to ≈600 possible combinations in CityGML!
  - Merge all standard attributes of a CityObject into a single "table"
- **Deal with the possibly huge size of city models** stored in a database
- Support for **multiple citydb schemas** in the same 3DCityDB instance
- Support for **multiple users with different privileges** (read-only, read-write)
- **Editing of attributes:** possible (depending on user privileges)
- **Deletion of features:** possible (depending on user privileges)
- Editing of geometries: NOT possible

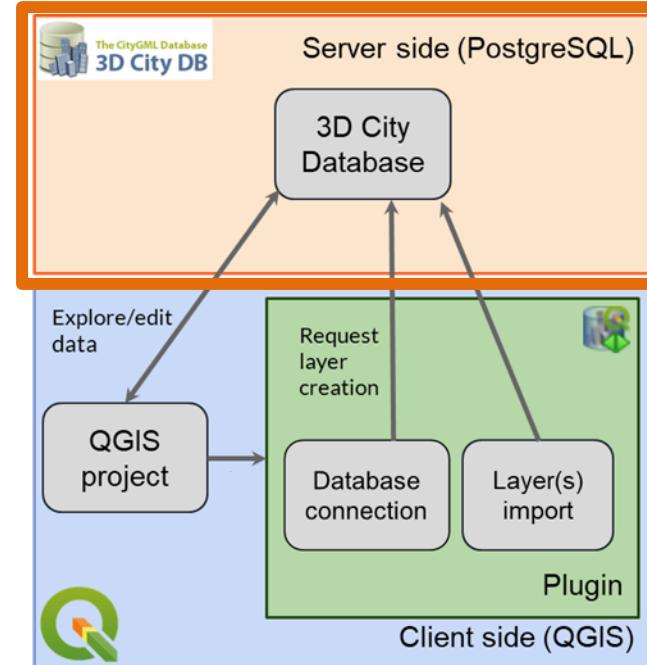
# Plugin overview

Motivation  
**Plugin overview**  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

## Server-side

### PostgreSQL "QGIS Package"

- Creates and manages layers as views (for attributes) linked to materialized views (for geometry) following the SFS model
- Manages
  - users and privileges
  - multiple citydb schemas
- Adds default users with ro & rw privileges



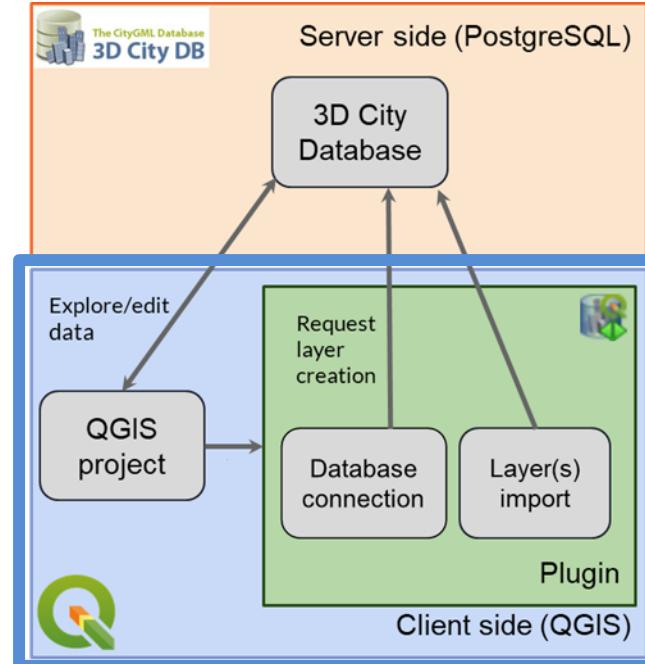
# Plugin overview

Motivation  
**Plugin overview**  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

## Client-side

### QGIS plugin “3DCityDB-Tools”

- Manages database connections + installation of the **QGIS Package**
- Allows for GUI-based
  - layer creation and management
  - management of multiple citydb schemas
  - editing of feature attributes
- GUI includes
  - support for children tables (e.g. generic attributes)
  - CityGML enumerations
  - Codelists
- Creates a hierarchical Table of Contents



# IMPORTANT NOTICE

The following slides assume that you are already familiar with the **3DCityDB Suite**

In particular you should:

- Have an already installed 3DCityDB database instance
- Be able to use the 3DCityDB Importer/Exporter
- Be able to import CityGML data into the 3DCityDB
- (Optionally) be able to create additional citydb schemas

Otherwise:

- Refer to the slides in "**3DCityDB\_Suite\_QuickInstall.pdf**" (also in the same folder of this file) and/or
- Follow the tutorial: <https://github.com/3dcitydb/tutorials>

Last but not least...

- You may profit from a basic knowledge of the main CityGML concepts 😊
- Otherwise, here a crash course for free: <http://www.urbangeobigdata.it/?p=195>

# Installation

## Software requirements

Motivation  
Plugin overview  
**Software installation**  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

- CityGML 3D City Database **v. 4.x** for PostgreSQL
  - <https://github.com/3dcitydb/3dcitydb-suite/releases>
  - <https://3dcitydb-docs.readthedocs.io/en/latest/>
  - **BEWARE:** 3DCityDB v. 3.x and older are NOT supported!
- PostgreSQL **v. 10 or higher**, PostGIS **v. 2.0 or higher**
  - <https://www.postgresql.org/download/>
  - **BEWARE:** NOT compatible with older versions
- QGIS **v. 3.22 LTR, v. 3.28 LTR or 3.34 LTR**
  - <https://qgis.org/en/site/forusers/download.html>
  - **BEWARE:** Not tested/not supported with other versions
- PgAdmin (suggested, not required)
  - <https://www.pgadmin.org/download/>

# Installation

## Compatibility matrix

Motivation  
 Plugin overview  
**Software installation**  
 Software use  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package Resources

QGIS version	3DCityDB-Tools works?	Comments
QGIS 3.36, 3.38	✓	Not supported, but it generally works
<b>QGIS 3.34 LTR</b>	✓	<b>Works</b>
QGIS 3.30, 3.32	✓	Not supported, but it generally works
<b>QGIS 3.28 LTR</b>	✓	<b>Works</b>
QGIS 3.26, 3.24	✓	Not supported, but it generally works
<b>QGIS 3.22 LTR</b>	✓	<b>Works. Reference version used for development</b>
QGIS 3.20	✗	May work, but not supported
QGIS 3.18	✗	"Import selected layers" button always disabled
QGIS 3.16 LTR	✗	User's GUI won't load. Issues with (outdated?) method QgsExtentGroupBox.setMapCanvas()

# Testing machines

Machines used for development and testing:

OS	Processor(s)	HD	RAM	PostgreSQL	PostGIS
Windows 10 21H2 64bit	Core i7-8650U 1.7 GHz	SSD 2 TB	32 GB	16, 64bit	3.4
Windows 10 21H2 64bit	Core i7-8650U 1.7 GHz	SSD 2 TB	32 GB	14, 64bit	3.2
Windows 10 21H2 64bit	Core i7-8650U 1.7 GHz	SSD 2 TB	32 GB	11, 64bit	3.1
Ubuntu 20.04.3 LTS 64bit	Intel i7-7500U (4) 3.500GHz	SSD 250 GB	8 GB	12, 64bit	3.1
Mac OS (11.6.2 64bit)	Core i9-9980HK	HDD 1TB	32 GB	14, 64bit	3.1
Ubuntu 18.4 LTS 64bit	Virtual Machine	HDD 2 TB	16 GB	10, 64bit	3.0
Windows 10 22H2 64bit	Core i7-8565U 1.8 GHz	SSD 250GB + HDD 1 TB	16 GB	15, 64 bit	3.1
Ubuntu 22.04 LTS 64bit	Virtual Machine	100 GB	8 GB	14, 64 bit	3.2

Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

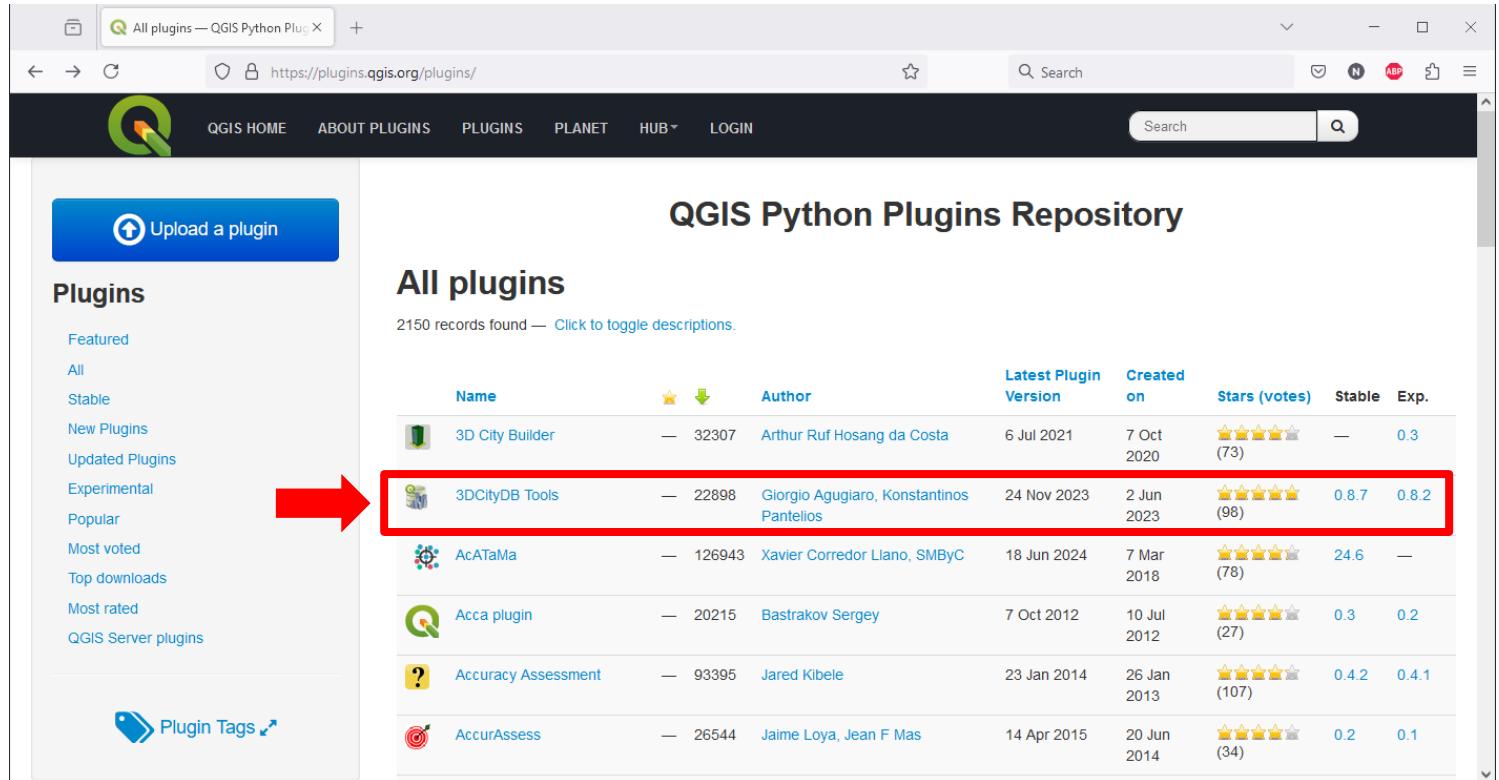
QGIS  
Package

Resources

# Installation via QGIS Plugins repository

Since version 0.8.2, the plugin is available also via the **QGIS Plugins repository**. This is the preferred (and easiest!) way to install it! Link: <https://plugins.qgis.org/plugins/>

Motivation  
Plugin overview  
**Software installation**  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources



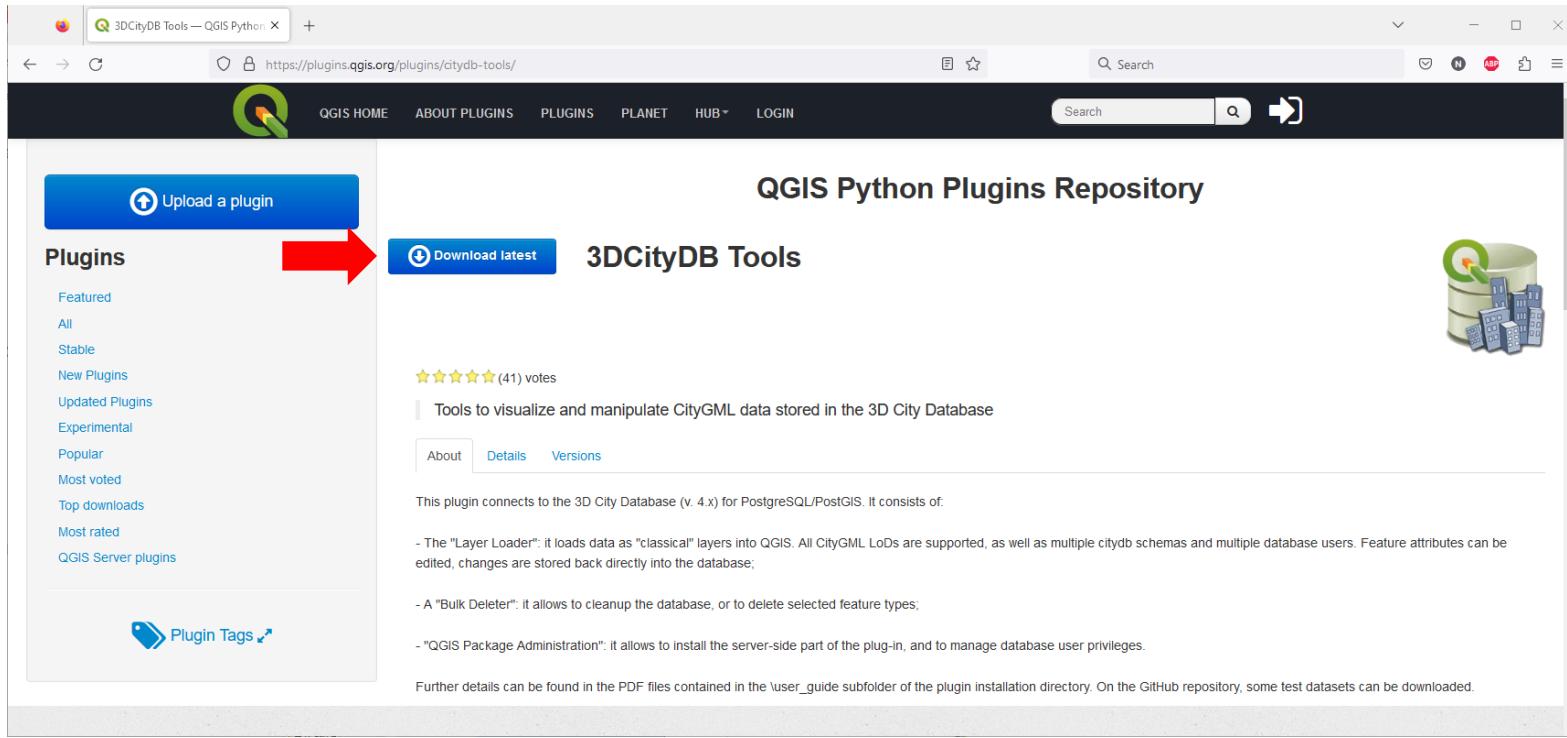
The screenshot shows the QGIS Python Plugins Repository website at <https://plugins.qgis.org/plugins/>. The main page displays a list of all available plugins. A red arrow points to the '3DCityDB Tools' plugin, which is highlighted with a red border. The table below lists the top plugins:

Name	Author	Latest Plugin Version	Created on	Stars (votes)	Stable	Exp.
3D City Builder	Arthur Ruf Hosang da Costa	6 Jul 2021	7 Oct 2020	★★★★★ (73)	—	0.3
3DCityDB Tools	Giorgio Agugliaro, Konstantinos Pantelios	24 Nov 2023	2 Jun 2023	★★★★★ (98)	0.8.7	0.8.2
AcATAma	Xavier Corredor Llano, SMByC	18 Jun 2024	7 Mar 2018	★★★★★ (78)	24.6	—
Acca plugin	Bastrakov Sergey	7 Oct 2012	10 Jul 2012	★★★★★ (27)	0.3	0.2
Accuracy Assessment	Jared Kibele	23 Jan 2014	26 Jan 2013	★★★★★ (107)	0.4.2	0.4.1
AccurAssess	Jaime Loya, Jean F Mas	14 Apr 2015	20 Jun 2014	★★★★★ (34)	0.2	0.1

# Installation via QGIS Plugins repository

Simply select it, and download it. If QGIS is already installed, it will be loaded automatically to the right folder. Done! ☺

Motivation  
Plugin overview  
**Software installation**  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources



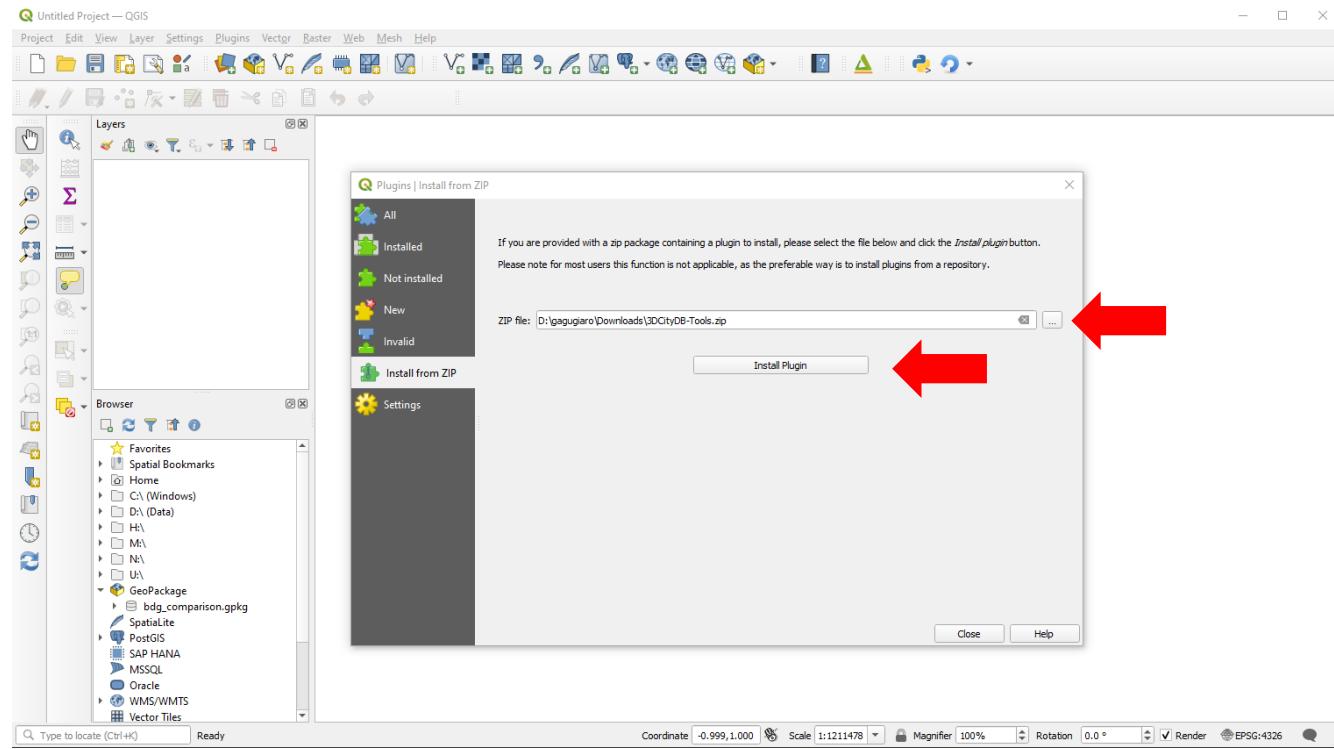
The screenshot shows a web browser window displaying the QGIS Python Plugins Repository at <https://plugins.qgis.org/plugins/citydb-tools/>. The page features a navigation bar with links to QGIS HOME, ABOUT PLUGINS, PLUGINS, PLANET, HUB, and LOGIN. A search bar is also present.

The main content area is titled "QGIS Python Plugins Repository" and displays the "3DCityDB Tools" plugin. A large red arrow points from the "Plugins" sidebar towards the "Download latest" button. The sidebar lists various plugin categories: Featured, All, Stable, New Plugins, Updated Plugins, Experimental, Popular, Most voted, Top downloads, Most rated, and QGIS Server plugins. Below the sidebar is a "Plugin Tags" section.

The "3DCityDB Tools" plugin page includes a rating of 4.1 stars from 41 votes. It is described as "Tools to visualize and manipulate CityGML data stored in the 3D City Database". The "About" tab is selected, showing the following description: "This plugin connects to the 3D City Database (v. 4.x) for PostgreSQL/PostGIS. It consists of: - The "Layer Loader": it loads data as "classical" layers into QGIS. All CityGML LoDs are supported, as well as multiple citydb schemas and multiple database users. Feature attributes can be edited, changes are stored back directly into the database; - A "Bulk Deleter": it allows to cleanup the database, or to delete selected feature types; - "QGIS Package Administration": it allows to install the server-side part of the plug-in, and to manage database user privileges." A note at the bottom states: "Further details can be found in the PDF files contained in the user\_guide subfolder of the plugin installation directory. On the GitHub repository, some test datasets can be downloaded."

# Manual installation

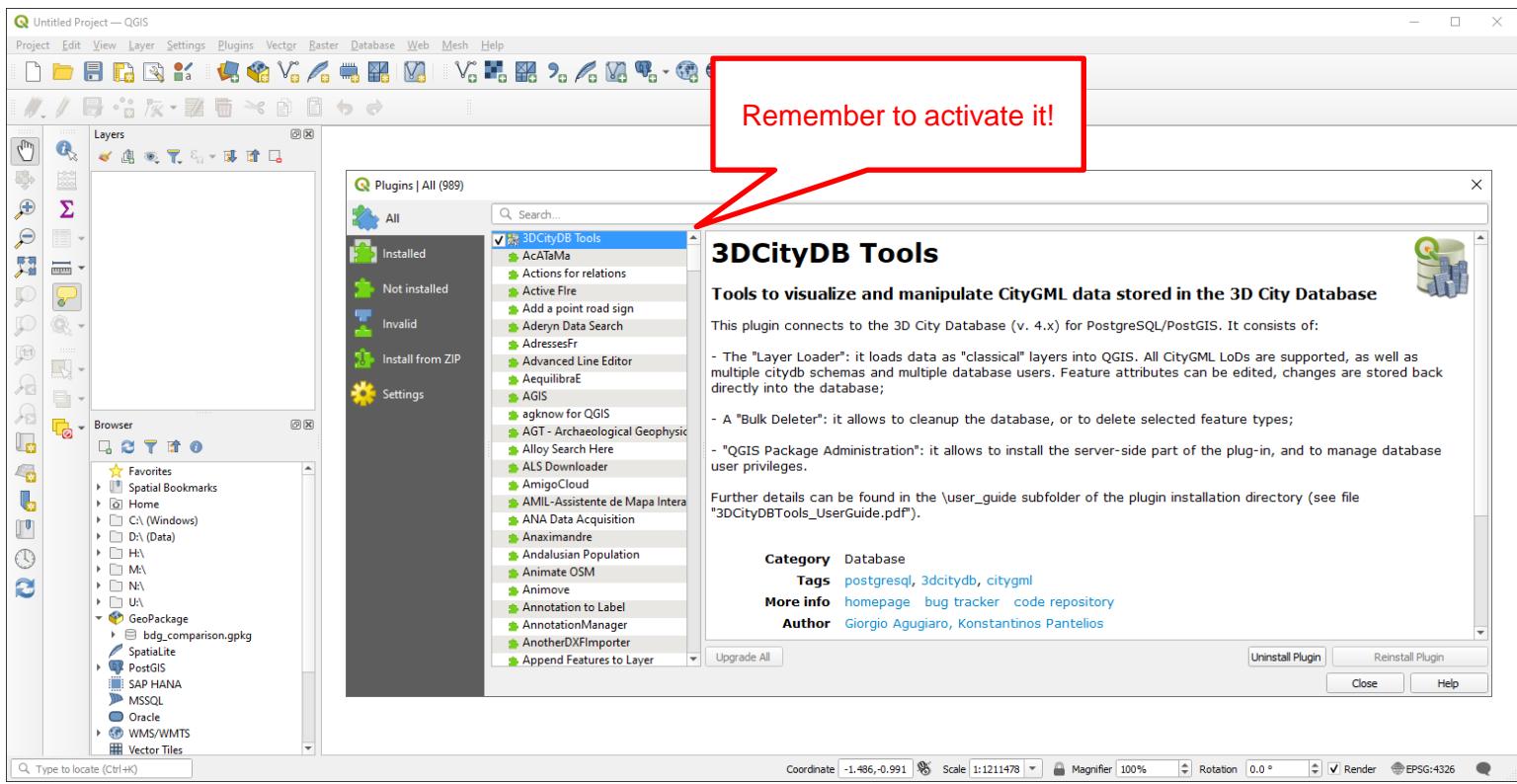
- The plugin is available also as **zip file** from the **GitHub repository**
- In QGIS, open the Plugins\Manage and install plugins window, and choose "Install from ZIP". Select the zip file and click the "Install Plugin" button



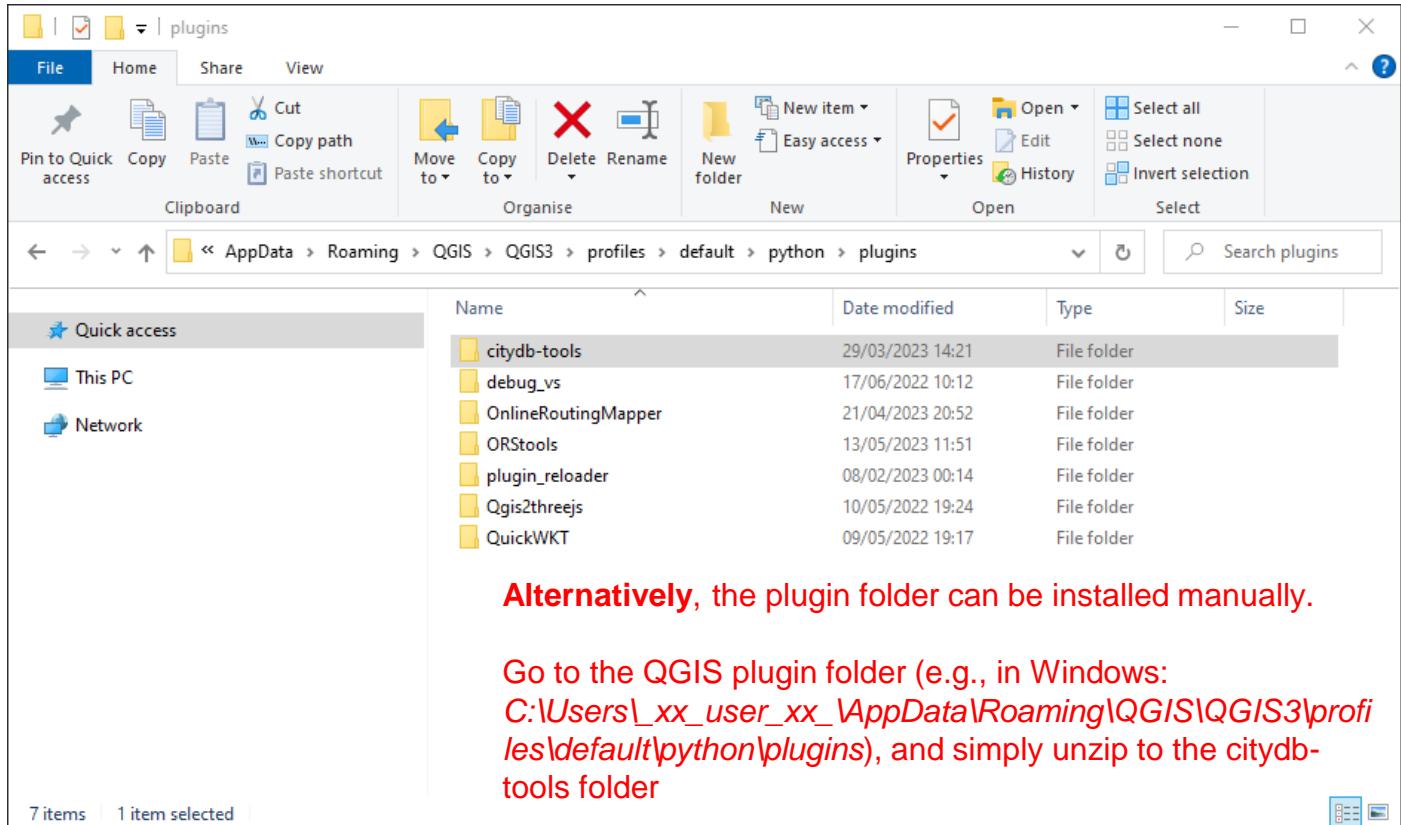
# Manual installation

- Upon installation, you must activate the plugin

Motivation  
Plugin overview  
**Software installation**  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

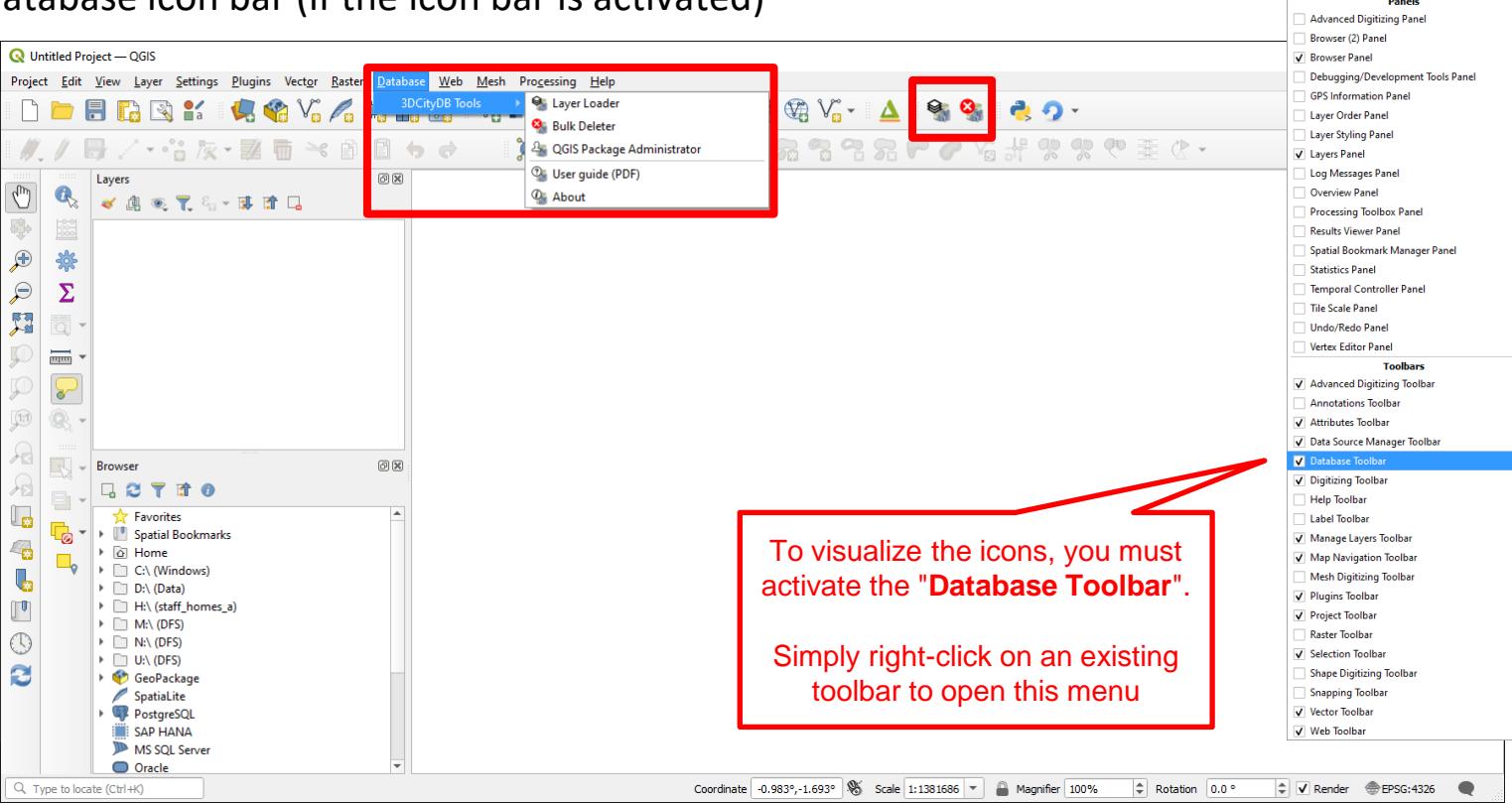


# Manual installation



# Front-end installation

The **Database menu** will now contain a new entry, and the icons will be visible in the database icon bar (if the icon bar is activated)



# Back-end installation

The back-end installation consists in installing the **QGIS Package** into a 3DCityDB instance. It can be carried out using the "QGIS Package Administrator" GUI of the front-end. The **database administrator** is responsible for setting up in advance the server-side for *any* database user.

In general, **4 steps** are necessary:

- a) Installation of the QGIS Package (i.e. the "qgis\_pkg" schema)
- b) Selection of the database users (e.g. "giorgio")
- c) Creation of a user schema for each selected user (e.g. "qgis\_giorgio")
- d) Definition of the database privileges for each user and for each citydb schema (i.e. "read-only", "read & write", "none")

It is possible to perform:

- A "**simplified installation**", which carries out all 4 operation at once and automatically installs 2 default users
- A "**normal installation**", where the administrator has complete control over each step

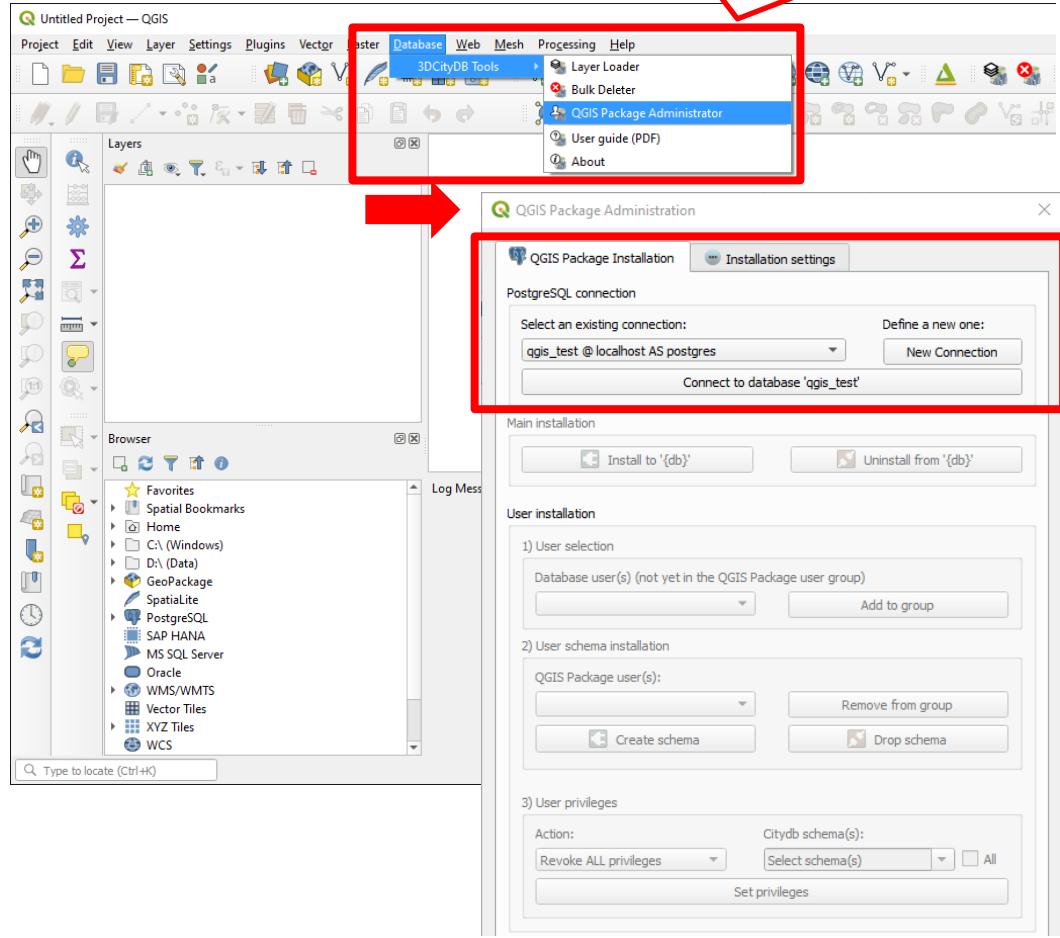
# Back-end installation

Watch out! Pick the QGIS Package Administrator!

Motivation  
Plugin overview  
**Software installation**  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package Resources

As **database administrator**  
(e.g. "postgres"):

1) Create a new connection  
or use an existing one to  
the desired 3DCityDB  
instance (here: "qgis\_test")



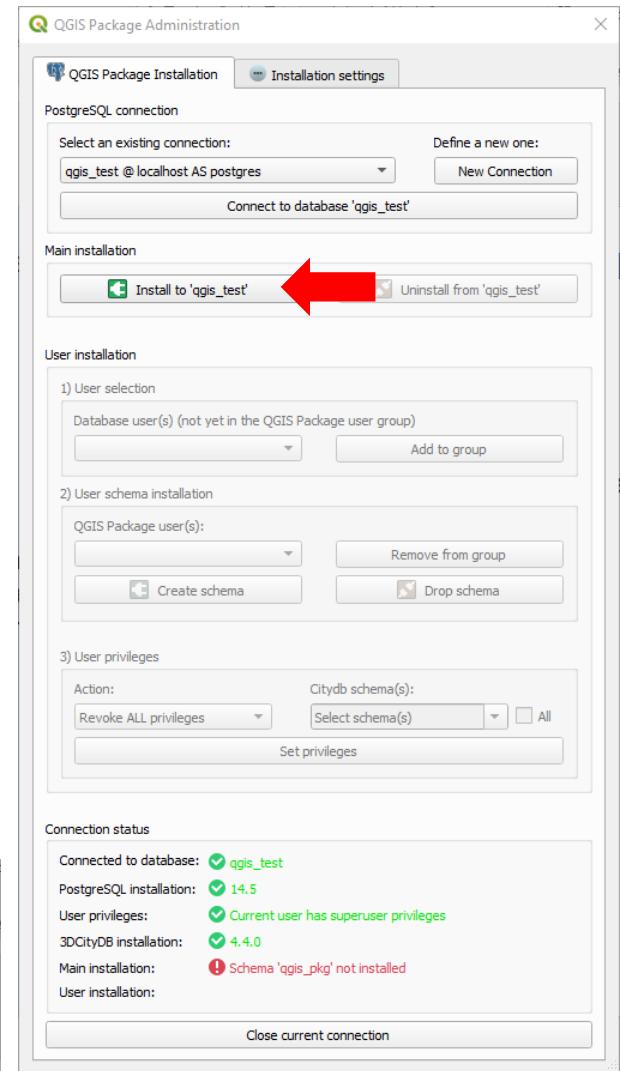
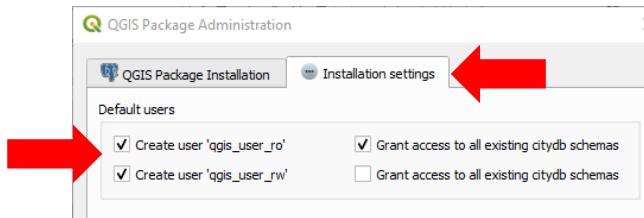
# Back-end installation

## 2.1) "Simplified" installation

The QGIS Package and up to two default users are installed at once:

- User "`qgis_user_ro`" with read-only privileges
- User "`qgis_user_rw`" with read & write privileges
- Both users have access to all citydb schemas in the database at the moment of the installation
- **Note bene:** Their privileges and access rules can be changed at *any* time after installation. See later the "normal"-installation slides

Before clicking the Install button, go to the "Installation settings" tab and check the desired options



A screenshot of the main 'QGIS Package Installation' tab in QGIS Package Administration. It shows a 'PostgreSQL connection' section with a dropdown set to 'qgis\_test @ localhost AS postgres'. Below it is a 'Main installation' section with a 'Install to' button (highlighted by a red arrow) and an 'Uninstall from' button. The 'User installation' section contains three tabs: 'User selection', 'User schema installation', and 'User privileges'. The 'Connection status' section at the bottom shows the database is connected to 'qgis\_test' and PostgreSQL is installed at version 14.5. A red arrow also points to the 'Install to' button in this screenshot.

# Back-end installation

## 2.1) "Simplified" installation

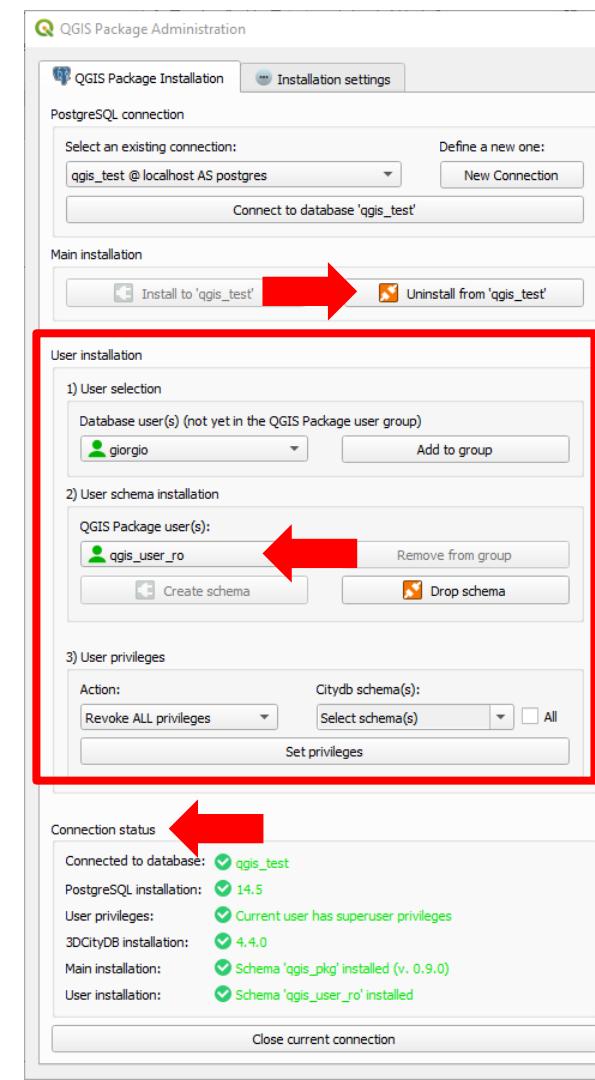
Upon successful installation:

- The **Uninstall button** is activated (in case you want to uninstall the QGIS Package)
- The **User Installation box** is activated
- You are notified in the **Connection status**

**Done!** 😊 Close the "QGIS Package Administration" GUI. You can now start using the plugin (e.g. the "Layer Loader" or the "Bulk Deleter") using the credentials of one of the default users

Please observe that:

- The default user(s) are automatically added to the **QGIS Package user(s)**
- You can edit the privileges in the **User privileges box**, or leave them as they are



# Back-end installation

## 2.1) "Simplified" installation

In the 3DCityDB, the **qgis\_pkg** schema is added and, depending on the chosen options, the **qgis\_user\_ro** and/or the **qgis\_usr\_rw** schemas are created, too

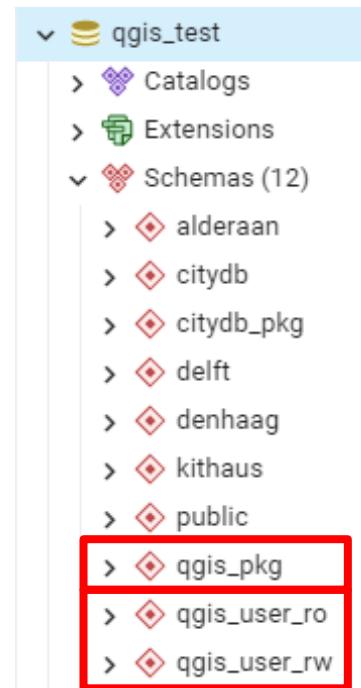
The credentials for the default users are:

### User "qgis\_user\_ro":

- user name: **qgis\_user\_ro**
- password: **qgis\_user\_ro**

### User "qgis\_user\_rw":

- user name: **qgis\_user\_rw**
- password: **qgis\_user\_rw**



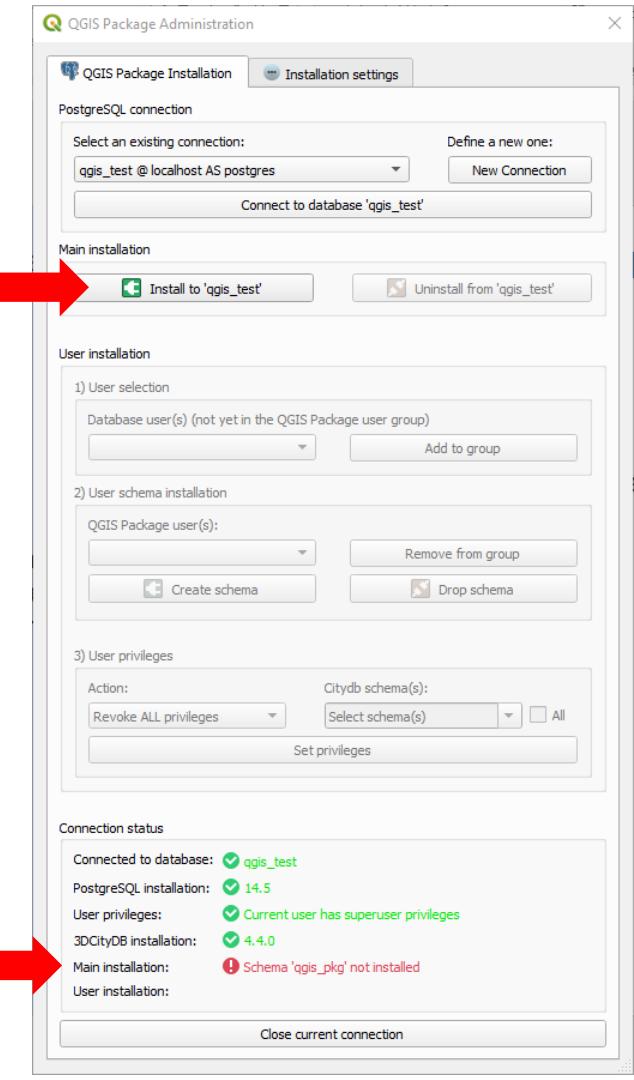
# Back-end installation

## 2.2) "Normal" installation

Using the "normal" installation, the administrator has complete control over each one of the 4 required installation steps

**Step a)** To install the GIS Package, click the **Install button** (here: install to database "qgis\_test")

The **Connection status box** in the lower part of the dialog will keep you informed.



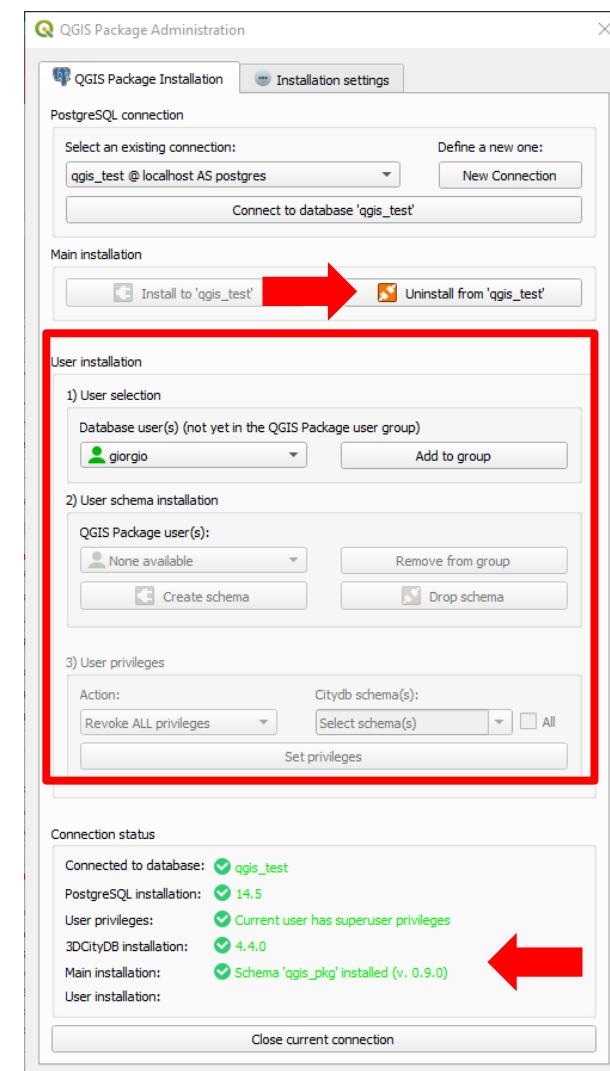
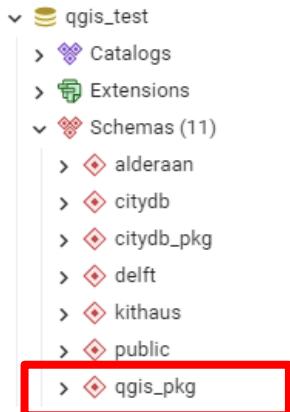
# Back-end installation

## 2.2) "Normal" installation

Upon successful installation:

- The **Uninstall button** is activated (in case you want to immediately uninstall)
- The **User Installation box** is activated
- You are notified in the **Connection status box**

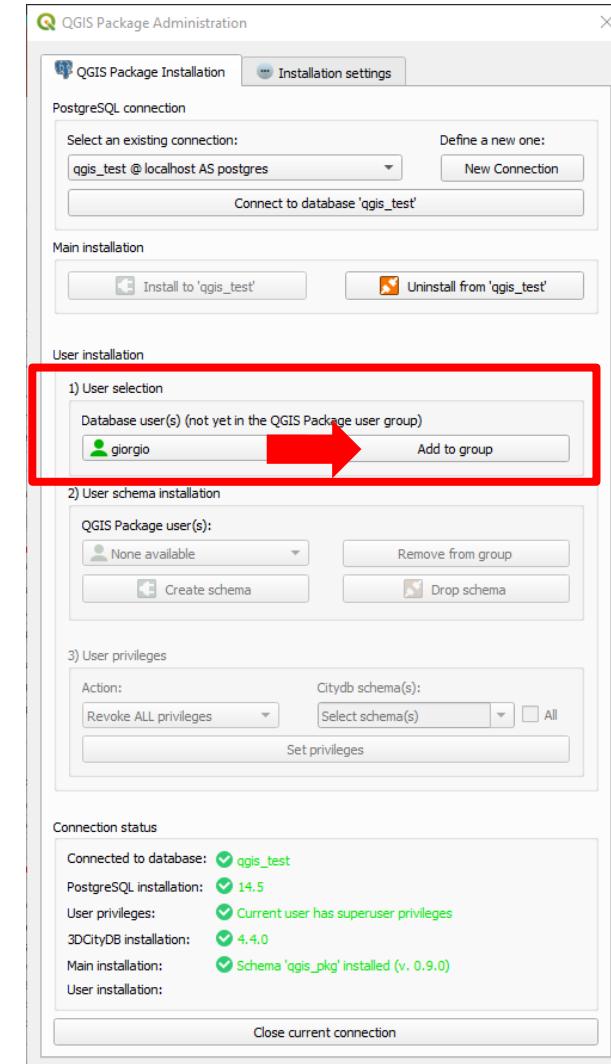
The "**qgis\_pkg**" schema is created in the the selected current database



# Back-end installation

## 2.2) "Normal" installation

**Step b)** Choose from all database users the one(s) to add to the QGIS Package user group. Click the **Add to group** button



The screenshot shows the 'QGIS Package Administration' window. In the 'User selection' section, there is a list of 'Database user(s) (not yet in the QGIS Package user group)' with 'giorgio' selected. A red arrow points to the 'Add to group' button next to it. The window also includes sections for 'User schema installation' and 'User privileges'.

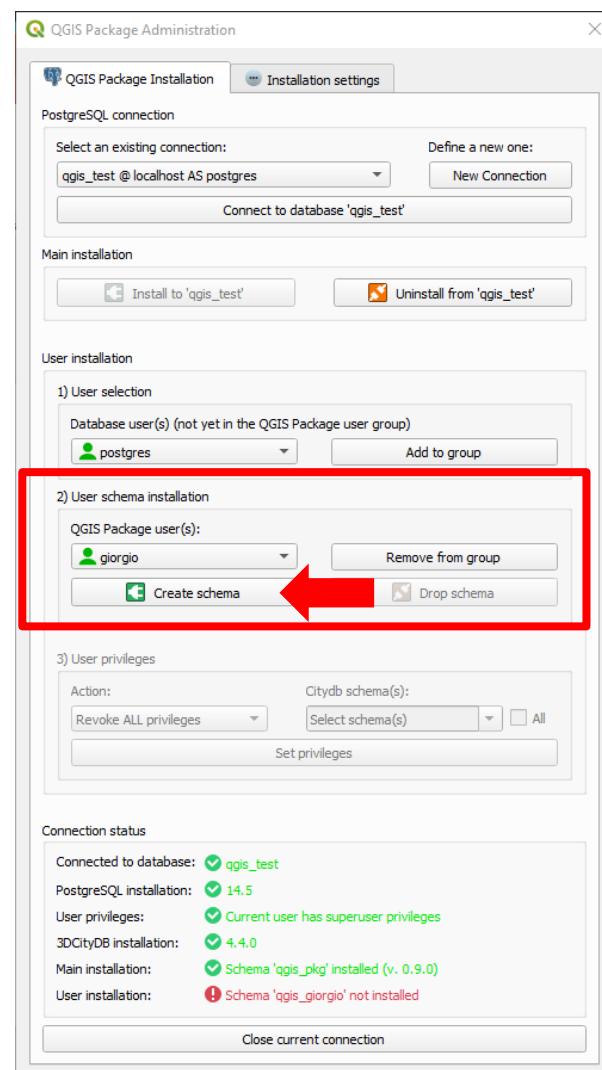
For each 3DCityDB there group named "**qgis\_pkg\_usrgroup\_**" + **database name** is created. It contains those users that will be allowed to interact with the database from the front-end.  
Example: for database "qgis\_test" there is a group called "qgis\_pkg\_usrgroup\_qgis\_test".

# Back-end installation

## 2.2) "Normal" installation

**Step c)** Create the user schema for the selected user(s) belonging to the group

Motivation  
Plugin overview  
**Software installation**  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

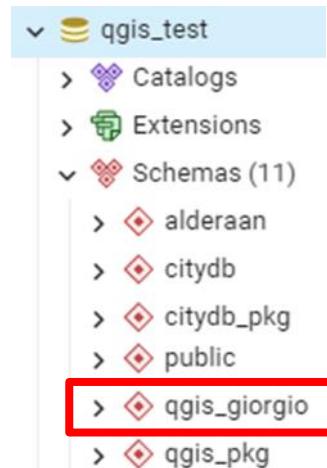


# Back-end installation

## 2.2) "Normal" installation

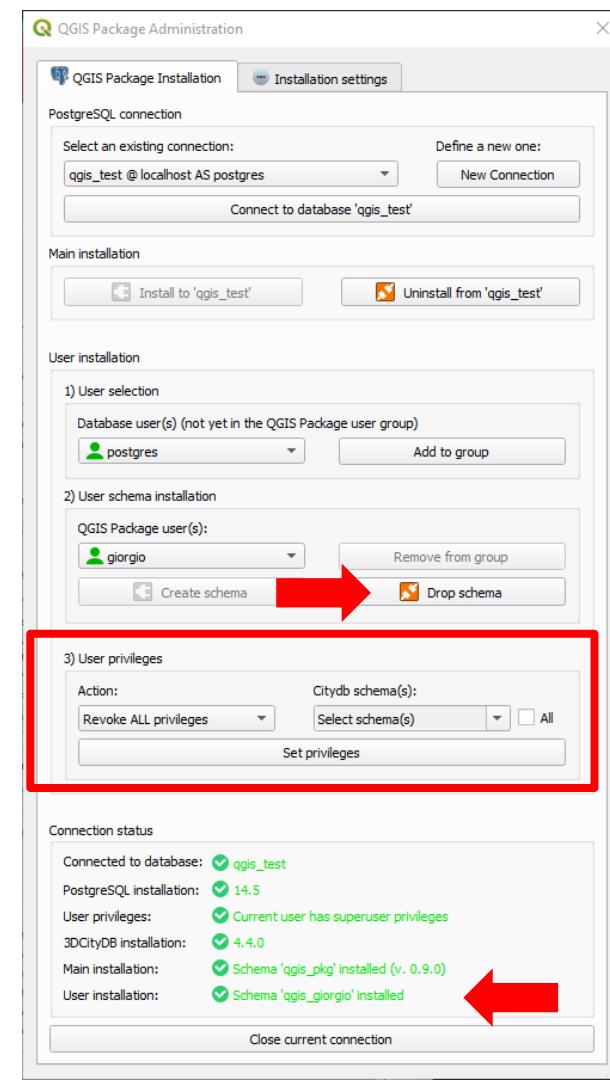
Upon successful creation of the user schema:

- The **Drop schema** button is activated (in case you want to drop the schema you just created)
- the **User privileges box** is activated
- You are notified in the **Connection status box**



A schema named "**qgis\_**" + **user name** is created.

Example: for user "giorgio", schema "qgis\_giorgio" will be created.



**QGIS Package Administration**

**PostgreSQL connection**

Select an existing connection: qgis\_test @ localhost AS postgres Define a new one: New Connection Connect to database 'qgis\_test'

**Main installation**

Install to 'qgis\_test' Uninstall from 'qgis\_test'

**User installation**

- 1) User selection Database user(s) (not yet in the QGIS Package user group): postgres Add to group
- 2) User schema installation QGIS Package user(s): giorgio Remove from group Create schema Drop schema
- 3) User privileges Action: Revoke ALL privileges Select schema(s) Set privileges

**Connection status**

Connected to database: qgis\_test  
 PostgreSQL installation: 14.5  
 User privileges: Current user has superuser privileges  
 3DCityDB installation: 4.4.0  
 Main installation: Schema 'qgis\_pkg' installed (v. 0.9.0)  
 User installation: Schema 'qgis\_giorgio' installed

**Close current connection**

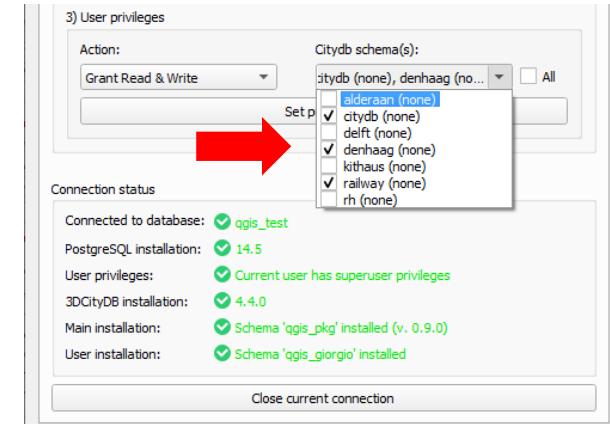
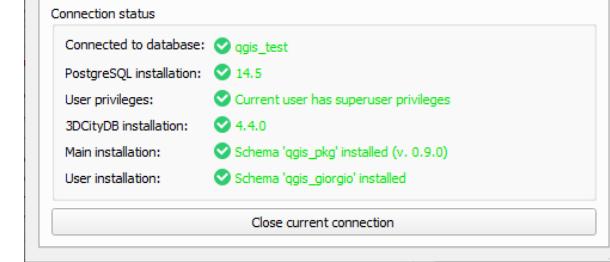
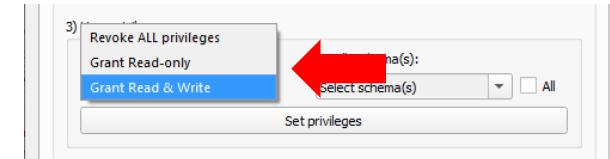
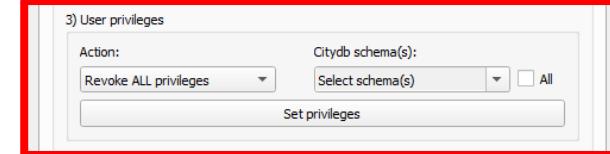
# Back-end installation

## 2.2) "Normal" installation

**Step d)** For the selected user, set the database privileges ("read-only", "read & write", "none") for each of the existing citydb schemas

You can assign different privileges to different citydb schemas – or revoke them.

Click the **Set privileges** button to apply the settings. The privileges status in the drop down menu will be updated accordingly.



# Back-end installation

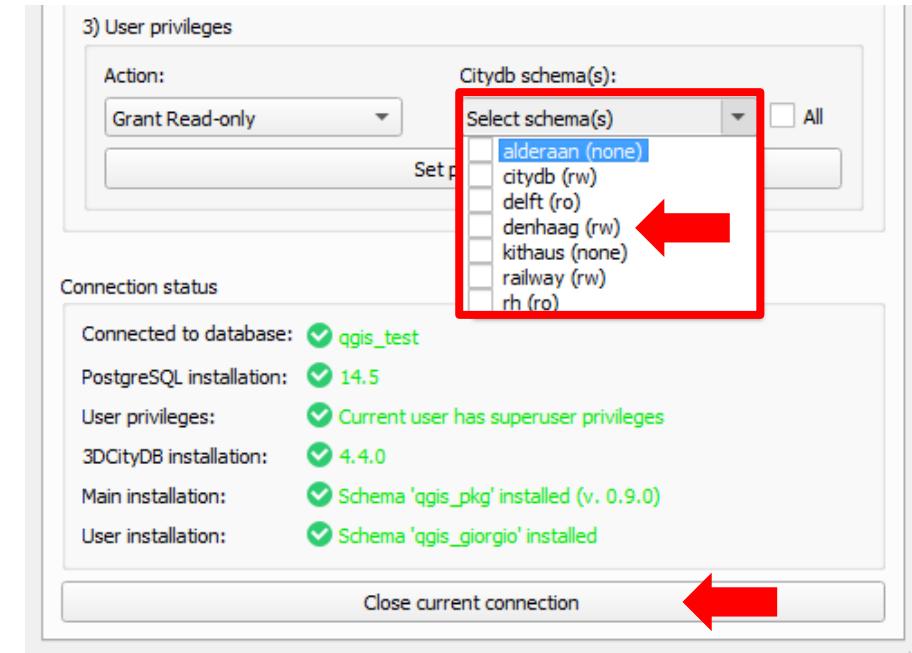
## 2.2) "Normal" installation

Motivation  
Plugin overview  
**Software installation**  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

Every time new privileges are set, the status in the drop down menu is updated with "ro" (read-only), "rw" (read & write) or "none".

Once you are done, you can click the **Close the current connection** button

You can now use the "Layer Loader" or the "Bulk Deleter"



Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

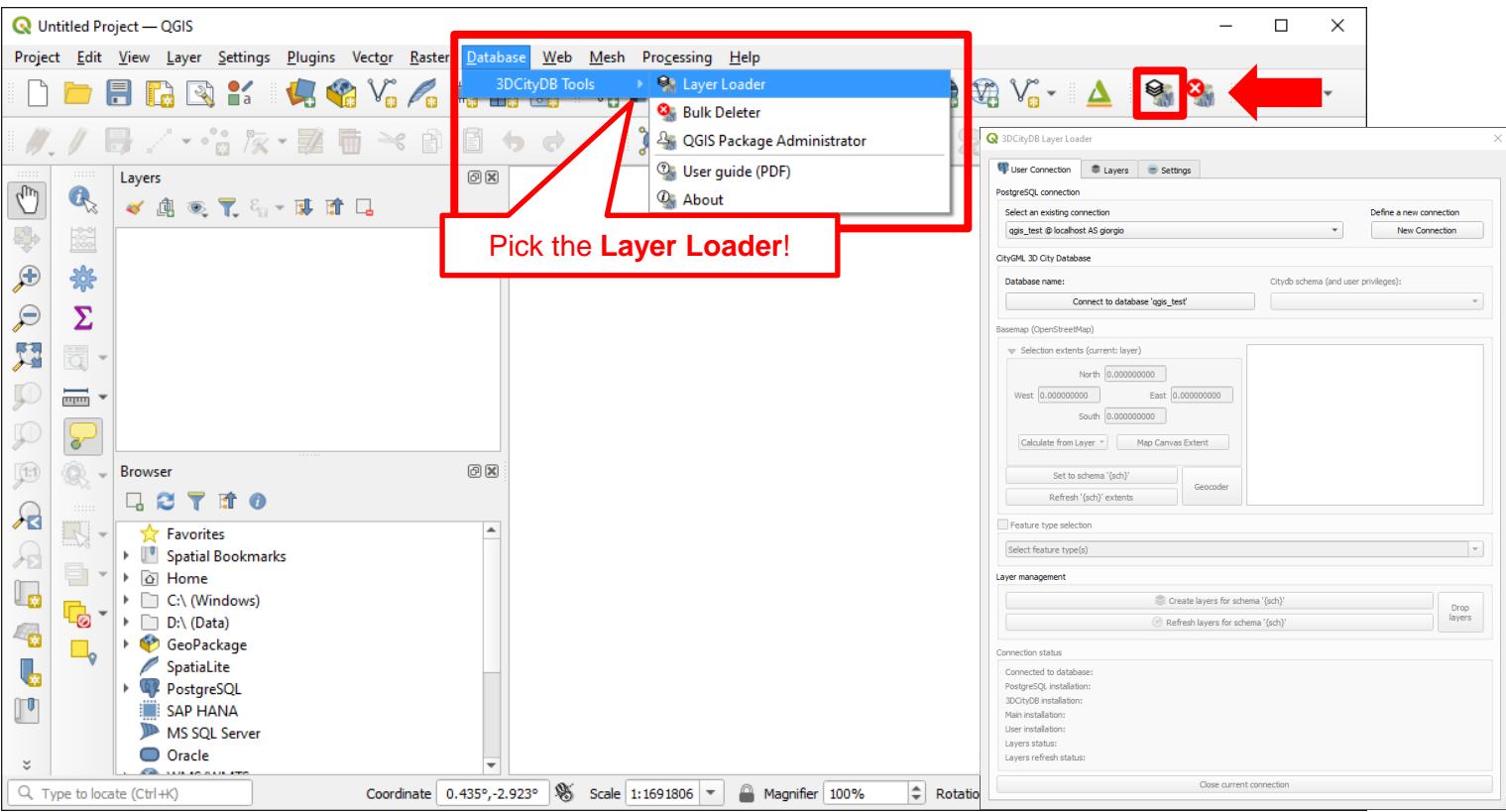
Resources

# Layer Loader



Open the **Layer Loader** from the menu or by clicking on the corresponding icon

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



The screenshot shows the QGIS application interface. The menu bar is open, and the 'Database' menu is selected. Within the 'Database' menu, the '3DCityDB Tools' option is highlighted, and the 'Layer Loader' option is shown as a submenu item. A red box highlights this path: 'Database > 3DCityDB Tools > Layer Loader'. A red arrow points from this box to a red box around the 'Layer Loader' icon in the toolbar. Another red box highlights the 'Layer Loader' icon in the toolbar. A red callout box with the text 'Pick the Layer Loader!' points to the 'Layer Loader' icon in the toolbar. To the right of the toolbar, the '3DCityDB Layer Loader' dialog box is open, showing various configuration options for connecting to a PostgreSQL database and managing layers.

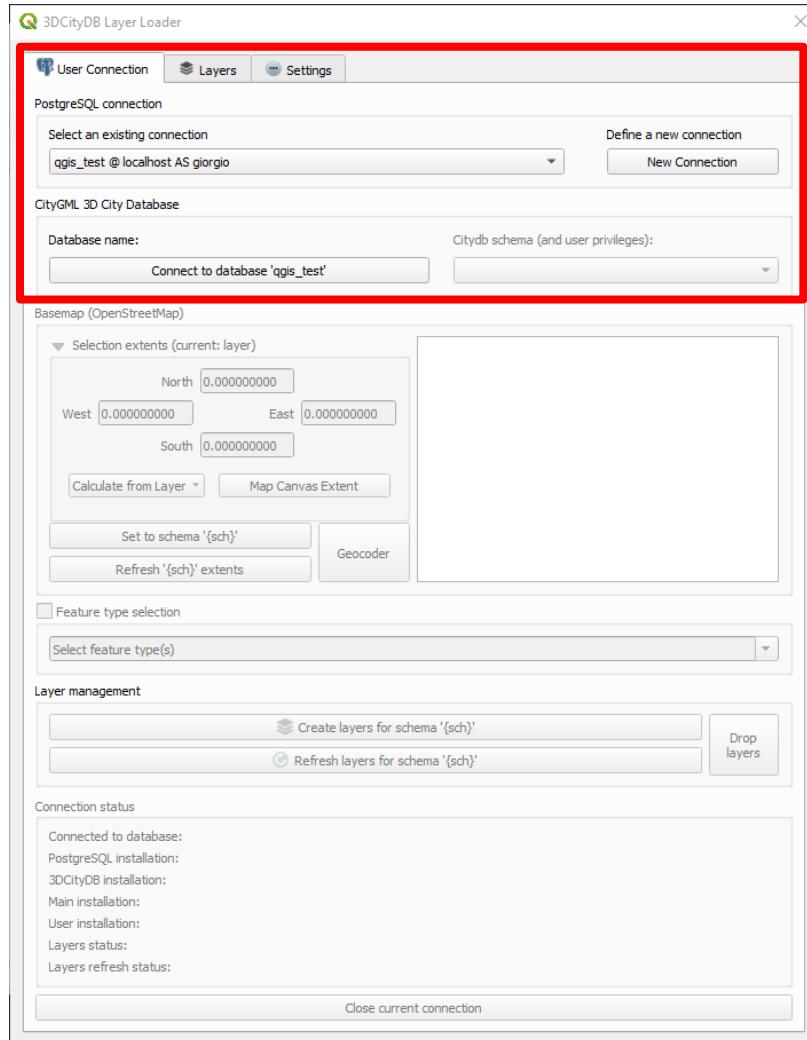
# Layer Loader

In the "User Connection" tab

1) Create a new connection or use an existing one to the desired 3DCityDB instance (here: "qgis\_test")

2) Use the credentials of:

- The default users **qgis\_user\_ro** or **qgis\_user\_rw** (if previously installed)
- Your own credentials (if the administrator has set up your *usr\_schema* before)



Motivation  
Plugin overview  
Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

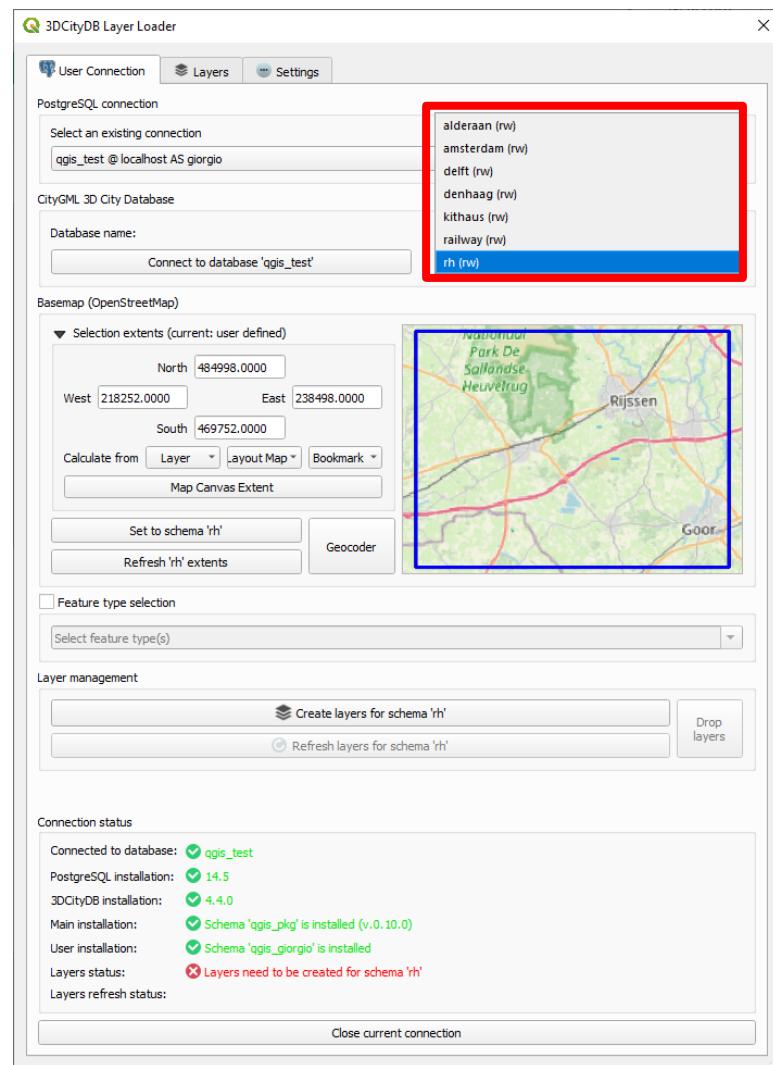
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

# Layer Loader

3) Once connected, choose one of the existing citydb schemas. If they contain CityGML data, they will be listed.  
You will also see your privileges for that citydb schema ("ro" or "rw").

**Nota bene:** Generally, "**citydb**" is the default, and, very often, the only one citydb schema! Nevertheless, the next slides refer to the "**rh**" schema

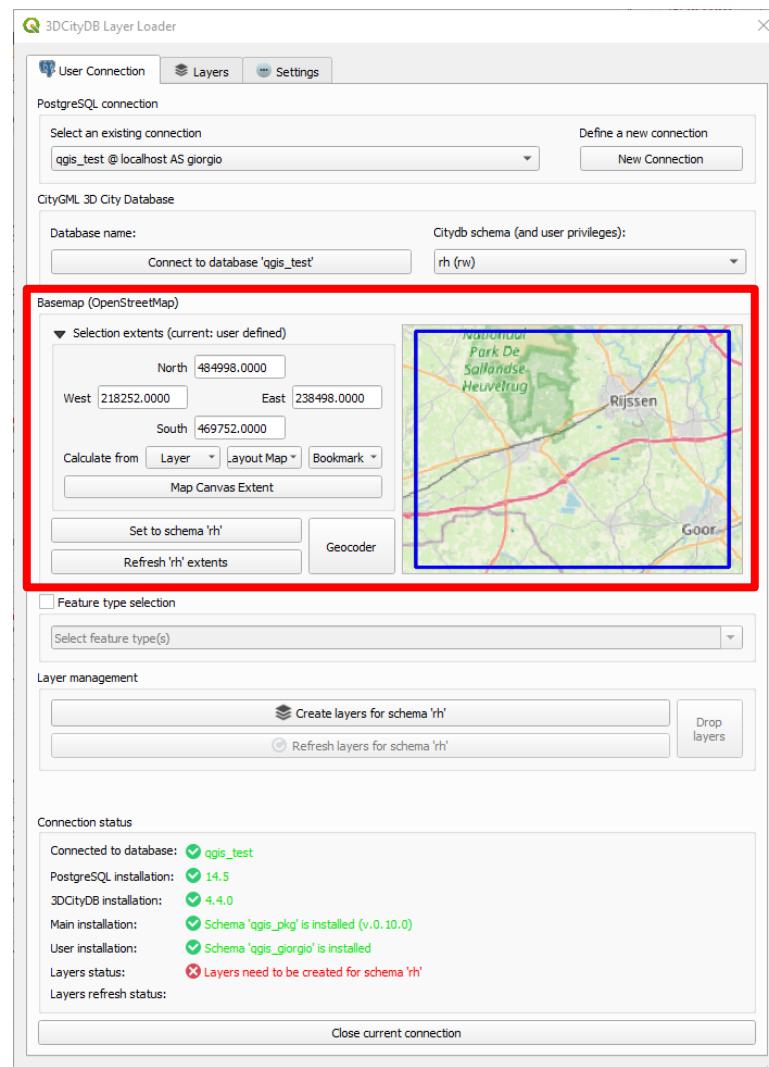
How to create additional citydb schemas  
<https://3dcitydb-docs.readthedocs.io/en/latest/3dcitydb/multi-schema.html>



# Layer Loader

4a) Upon selection of the citydb schema, you will see the extents of the dataset. They correspond to the extents of all currently loaded data in the selected citydb schema (here, for example, schema "rh")

**Please note:** the very first time you load a citydb schema, and depending on the size of the city model, it might take a while to load as the bounding boxes are being computed. From the second time onwards, it will load nearly instantly.

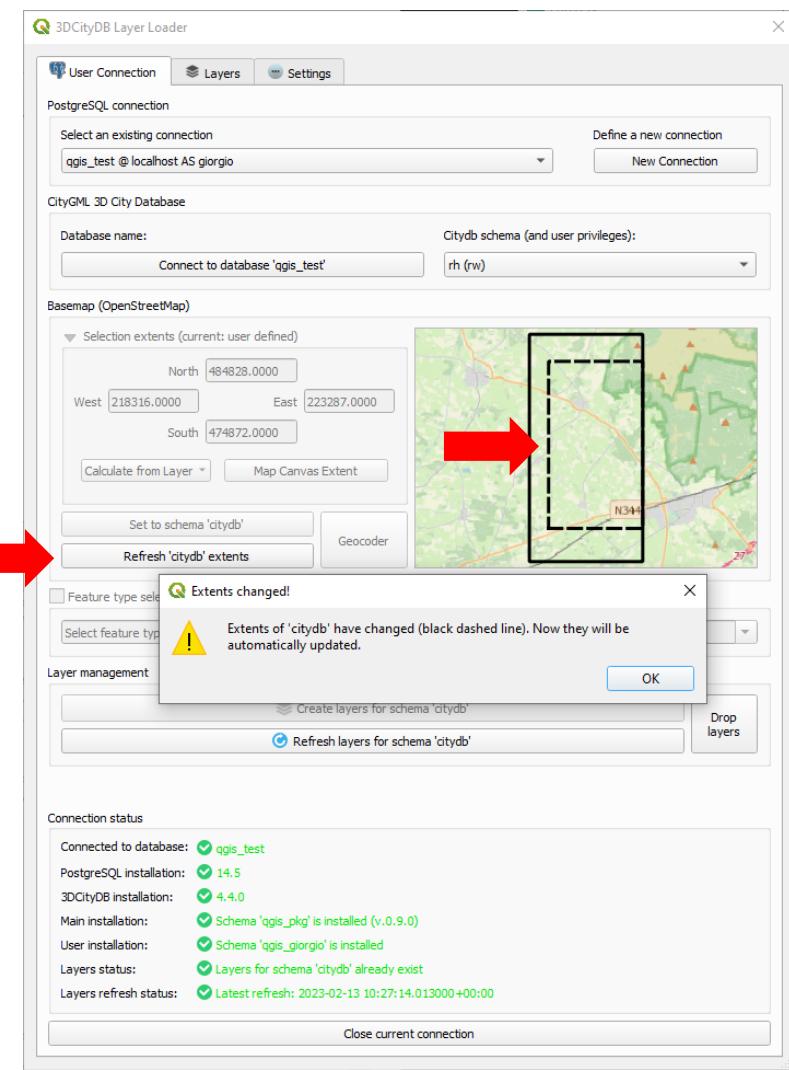


# Layer Loader

4b) If data has been added or removed in the current citydb schema, you can refresh the extents by pressing the **Refresh {cdb\_schema} extents** button.

The new extents will be temporarily shown with a **black dashed line**, before being updated.

**Note bene:** Depending on how the extents have changed, you may have to recreate, refresh and reload the layers in QGIS (see next slides)



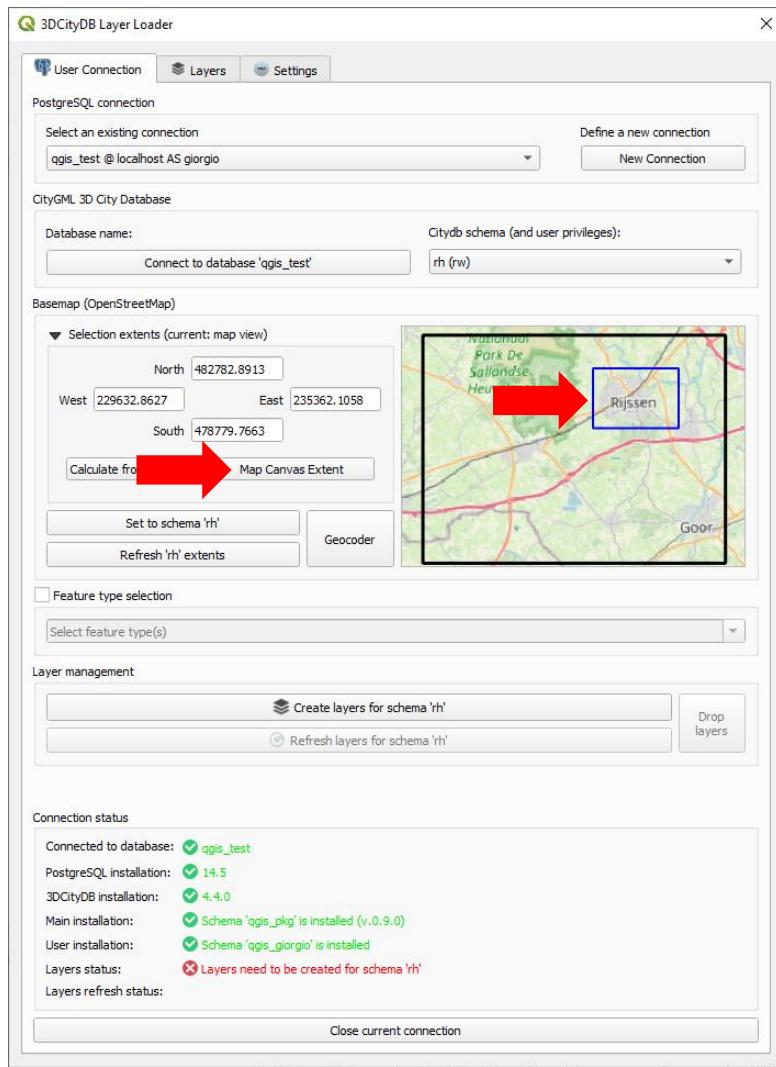
# Layer Loader

5) Depending on your needs, you can set the spatial extents of the study area for which the layers will be created

- Default: same size of the whole dataset
- Otherwise: zoom in the map and choose your own area by clicking the **Map Canvas Extent button**. The **blue bounding box** shows the layers extents.

**Beware:** The bigger the size, the more time it will take to populate the layers!

**Behind the scenes:** In the database, materialised views of the geometries will be generated according to the selected extents. In case of very large cities, it might take a long time (and a lot of space on the server)!



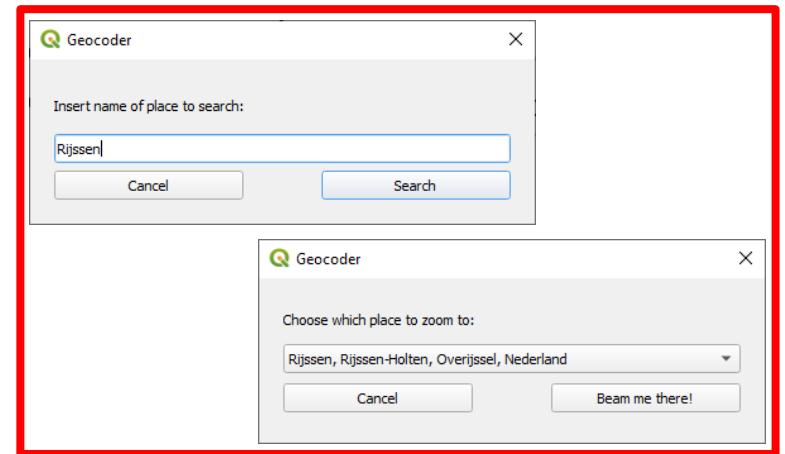
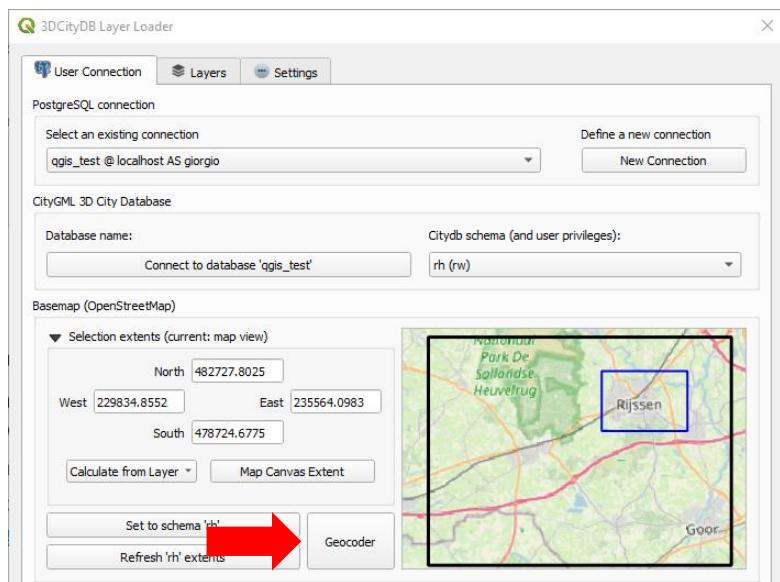
# Layer Loader

5) Depending on your needs, you can set the spatial extents of the study area for which the layers will be created

- Default: same size of the whole dataset
- Otherwise: zoom in the map and choose your own area by clicking the **Map Canvas Extent button**. The **blue bounding box** shows the layers extents.

If you are looking for a specific place inside the citydb extents, you can also use the Geocoder that will zoom you directly there.

Simply click the **Geocoder button**.



# Layer Loader

**Black:** database schema extents (i.e. extents of the whole city model/dataset)

**Blue:** database-side layers extents (i.e. extents of the materialised views)

Motivation

Plugin overview

Software installation

**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

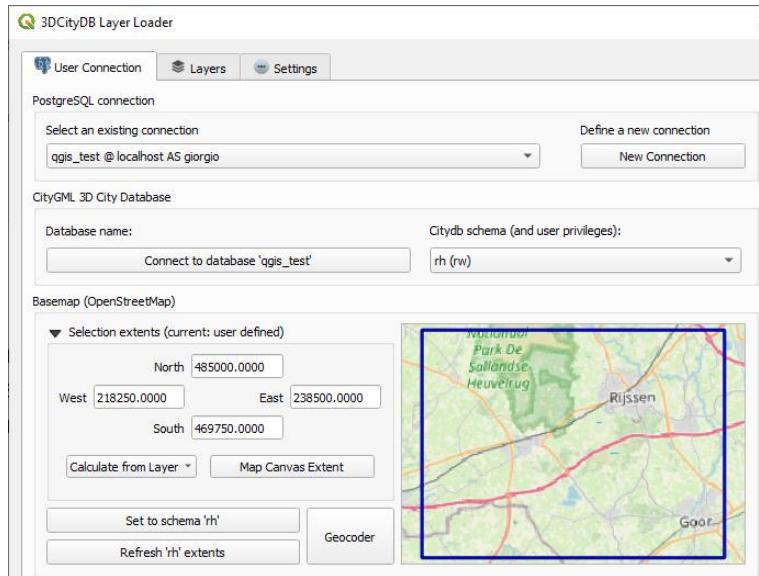
Advanced options

Software uninstall

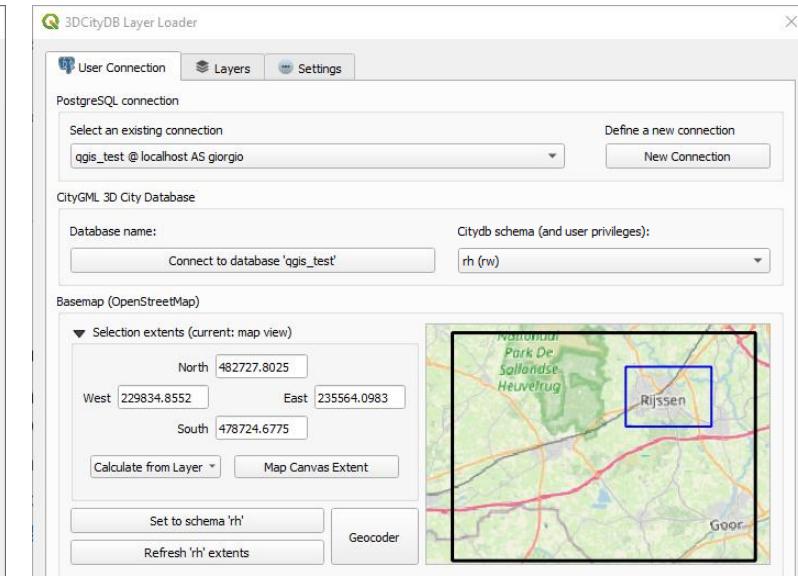
Current limitations

QGIS Package

Resources



Both areas coincide (default)



User-selected layers extents

Motivation  
 Plugin overview  
 Software installation  
**Software use**  
 • Layer Loader  
 • Bulk Deleter  
 • About, etc.  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

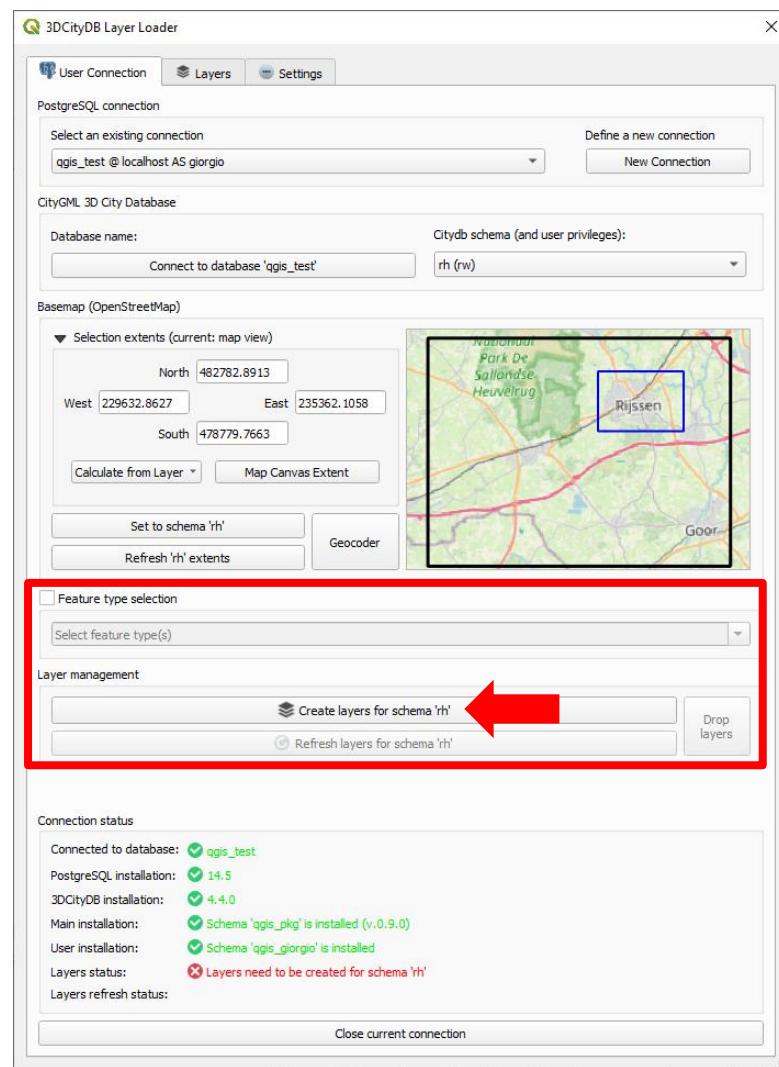
# Layer Loader

## 6a) Create the layers

Layers for all CityObjects available  
 within the Layer extents will be created

**Behind the scenes:** In the database, (empty)  
 materialised views of the geometries and  
 views will be created as layers.

Only layers for *existing* data will be created.



Motivation  
 Plugin overview  
 Software installation  
**Software use**  
 • Layer Loader  
 • Bulk Deleter  
 • About, etc.  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

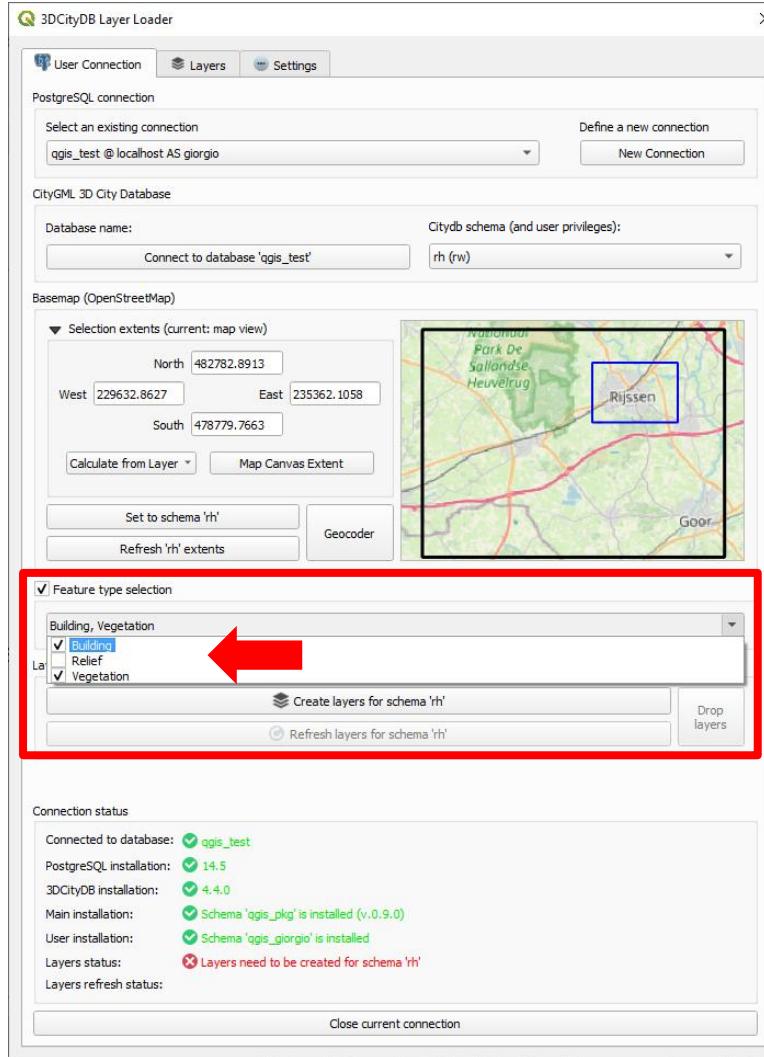
# Layer Loader

## 6a) Create the layers

Layers for all CityObjects available within the Layer extents will be created.

Optionally, you can further refine your selection and choose for which Feature Types the layers will be generated. Open the **Feature type selection box** and check the desired Feature types.

**Note bene:** Feature Types correspond to the CityGML modules (Building, Vegetation, Transportation, LandUse, Relief, etc.)

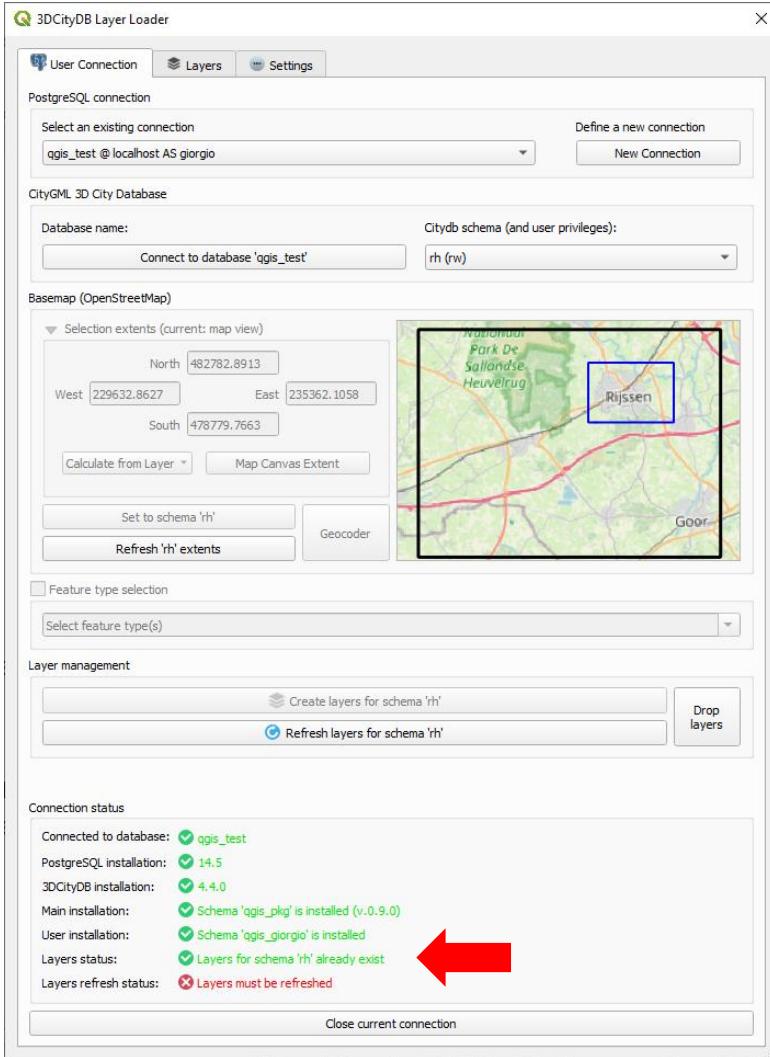


# Layer Loader

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

## 6b) Create the layers

Upon successful creation of the layers,  
you will be notified in the **Connection  
Status box**



The screenshot shows the 3DCityDB Layer Loader interface. At the top, there are tabs for User Connection, Layers, and Settings. Under User Connection, it says "PostgreSQL connection" and "Select an existing connection: qgis\_test @ localhost AS giorgio". There are buttons for "Define a new connection" and "New Connection". In the main area, under "CityGML 3D City Database", the "Database name" is set to "qgis\_test" and the "Citydb schema (and user privileges)" is "rh (rw)". Below this is the "Basemap (OpenStreetMap)" section, which includes a map view of a rural area with roads and green spaces. A blue rectangle highlights a specific location near "Rijssen". To the left of the map are coordinates: North (482782.8913), West (229632.8627), East (235362.1058), and South (478779.7663). Buttons for "Calculate from Layer", "Map Canvas Extent", "Set to schema 'rh'", "Refresh 'rh' extents", and "Geocoder" are present. Below the map is a "Feature type selection" section with a dropdown menu. The "Layer management" section contains buttons for "Create layers for schema 'rh'" and "Refresh layers for schema 'rh'". At the bottom, the "Connection status" section lists various system checks with their status: Connected to database (green checkmark), PostgreSQL installation (green checkmark), 3DCityDB installation (green checkmark), Main installation (green checkmark), User installation (green checkmark), Layers status (green checkmark), and Layers refresh status (red X). A red arrow points to the "Layers refresh status" line, which says "Layers must be refreshed". At the very bottom right is a "Close current connection" button.

Motivation  
 Plugin overview  
 Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

# Layer Loader

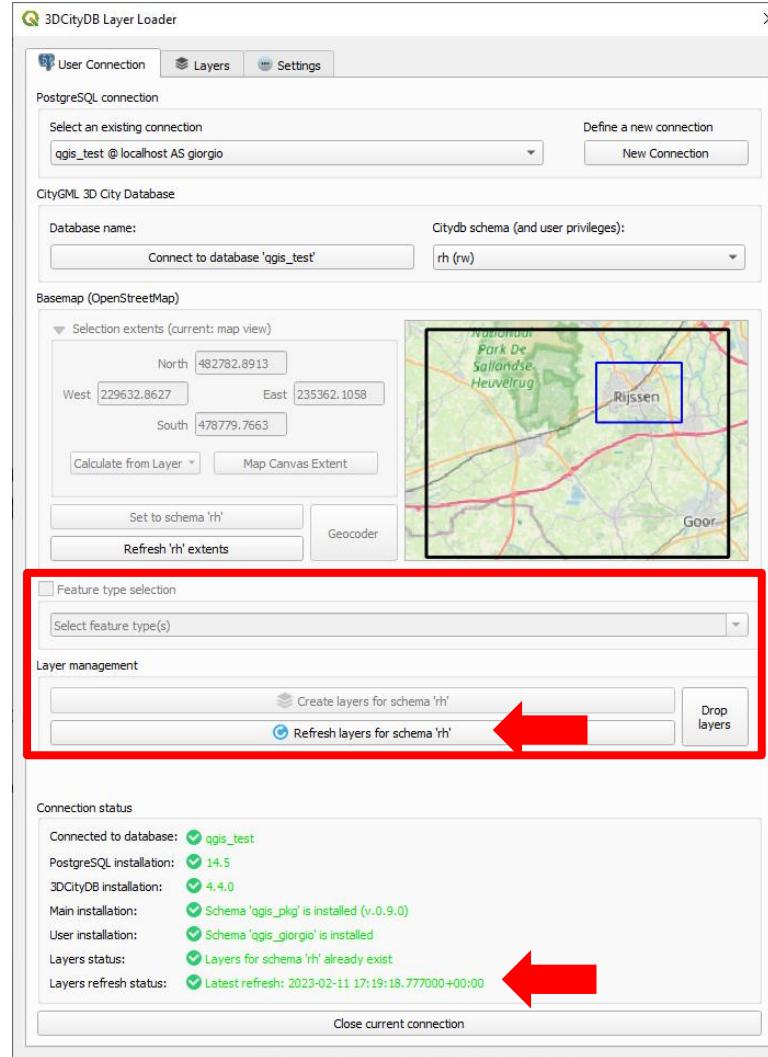
## 7) Populate/refresh the layers

Click on the **Refresh layers button**. Once the operation is complete, you will be notified in the **Connection status box**.

The following "Layers" tab is now activated and you can open it.

Alternatively, layers created in a previous session may be used (and/or refreshed again), or dropped.

**Beware:** Depending on the size of the selected area and the amount of data in the city model, this operation might take long.



# Layer Loader

Motivation  
Plugin overview  
Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

## Behind the scenes:

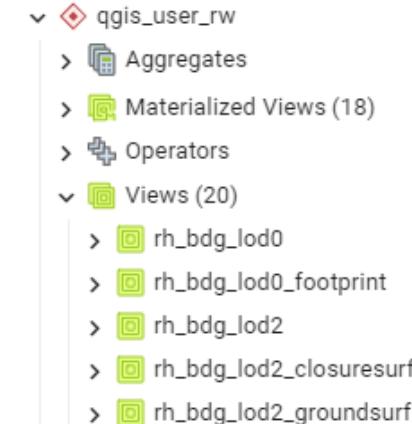
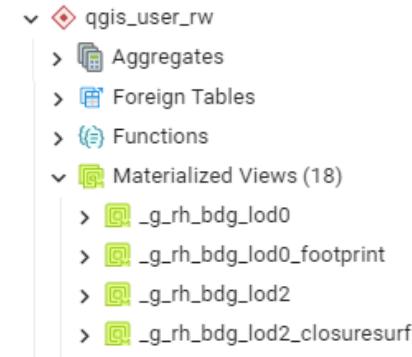
In the database user schema, both materialised views (containing the feature geometries) and the layers (as updatable views, containing the feature attributes linked to the corresponding geometries in the materialised views) can be accessed.

### Materialised views name coding:

- "\_g\_" prefix + citydb schema name + feature name + lidx + (optional) semantic details

### Views name coding:

- citydb schema name + feature name + lidx + (optional) semantic details
- Linked via column co\_id (PK and FK to the materialised views)

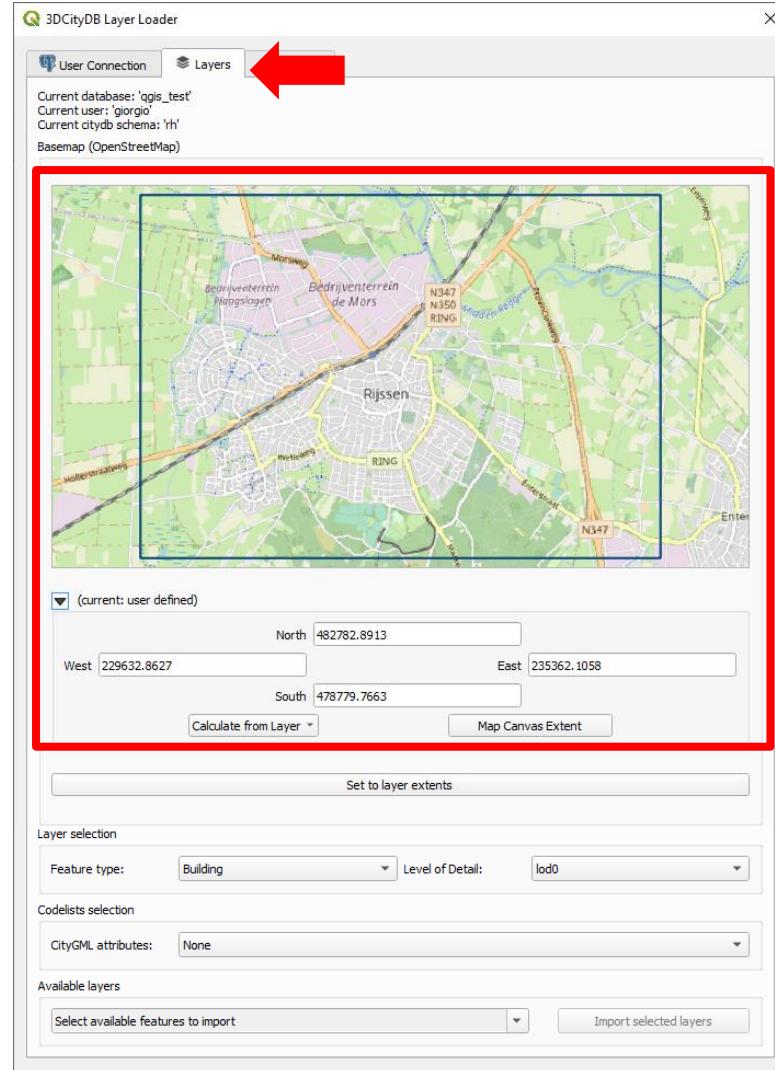


# Layer Loader

In the "Layers" tab

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

8) You are now directly zoomed to the layers extents (**blue bounding box**)



Motivation  
Plugin overview  
Software installation  
**Software use**

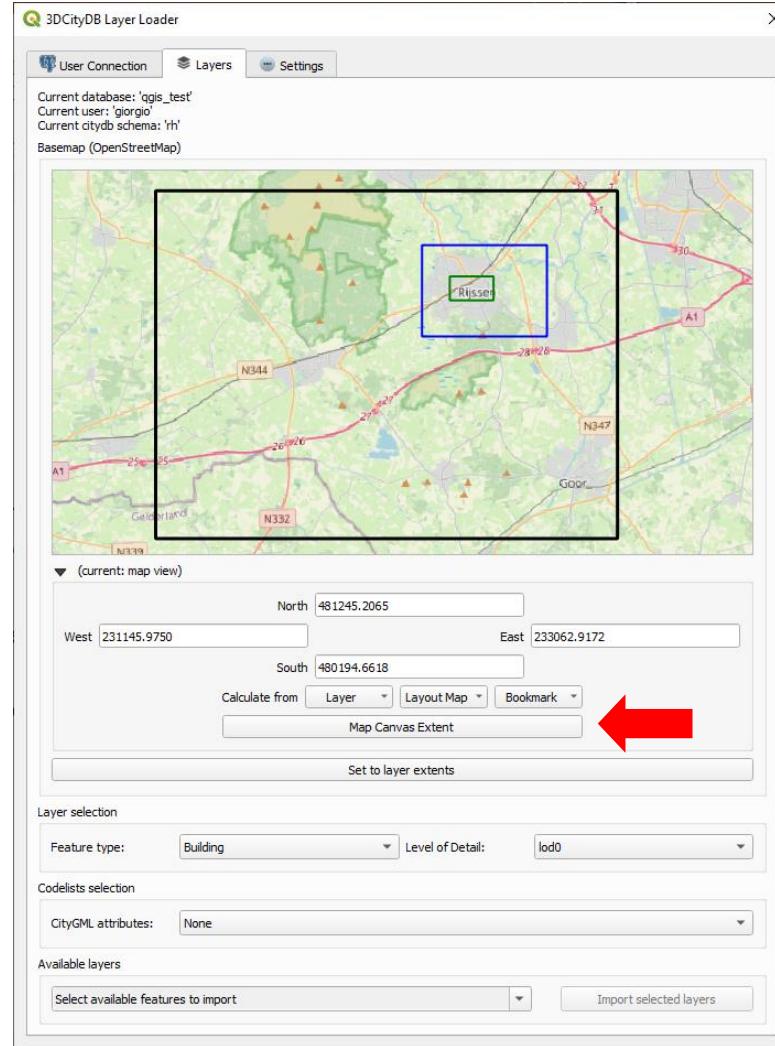
- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

# Layer Loader

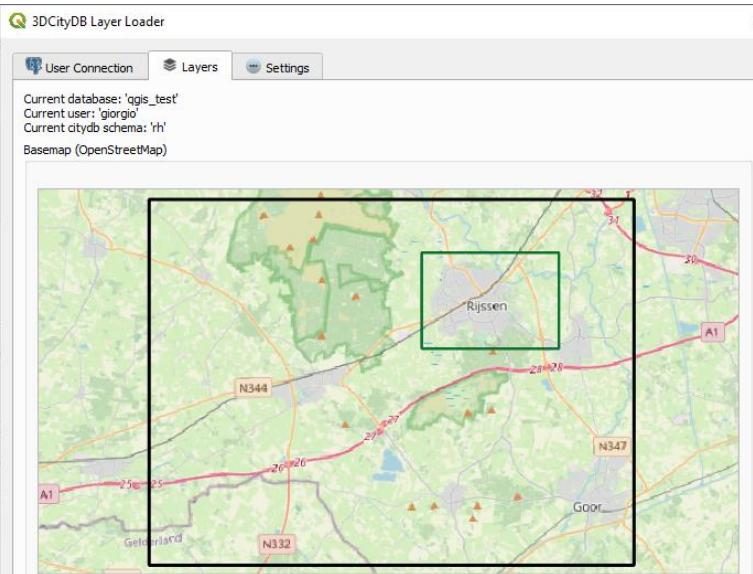
9) Depending on your needs, you can further reduce the extents of the layers to be loaded into QGIS by pressing the **Map Canvas Extent button**. The new extents are represented by the **green bounding box**.

- Default: same size of the layer extents (**blue bounding box**)
- Otherwise: zoom in and choose your own area

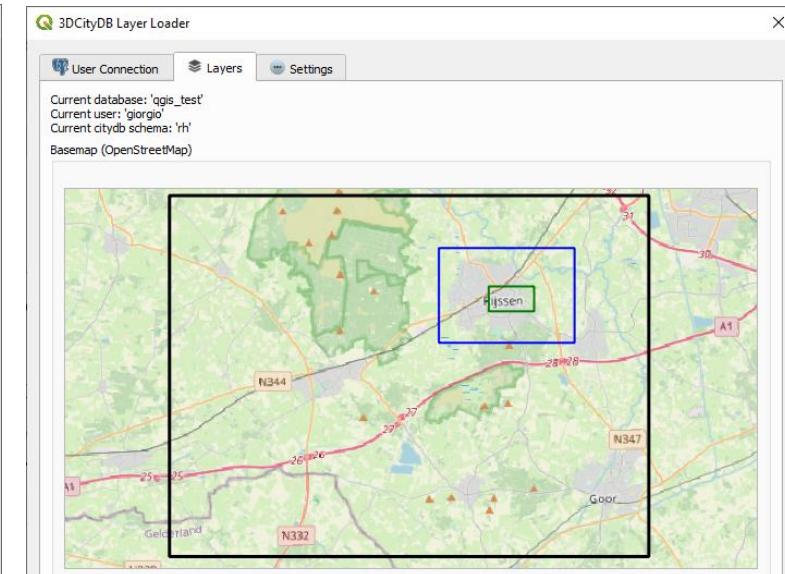


# Layer Loader

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



Database- and QGIS-side extents coincide (default)

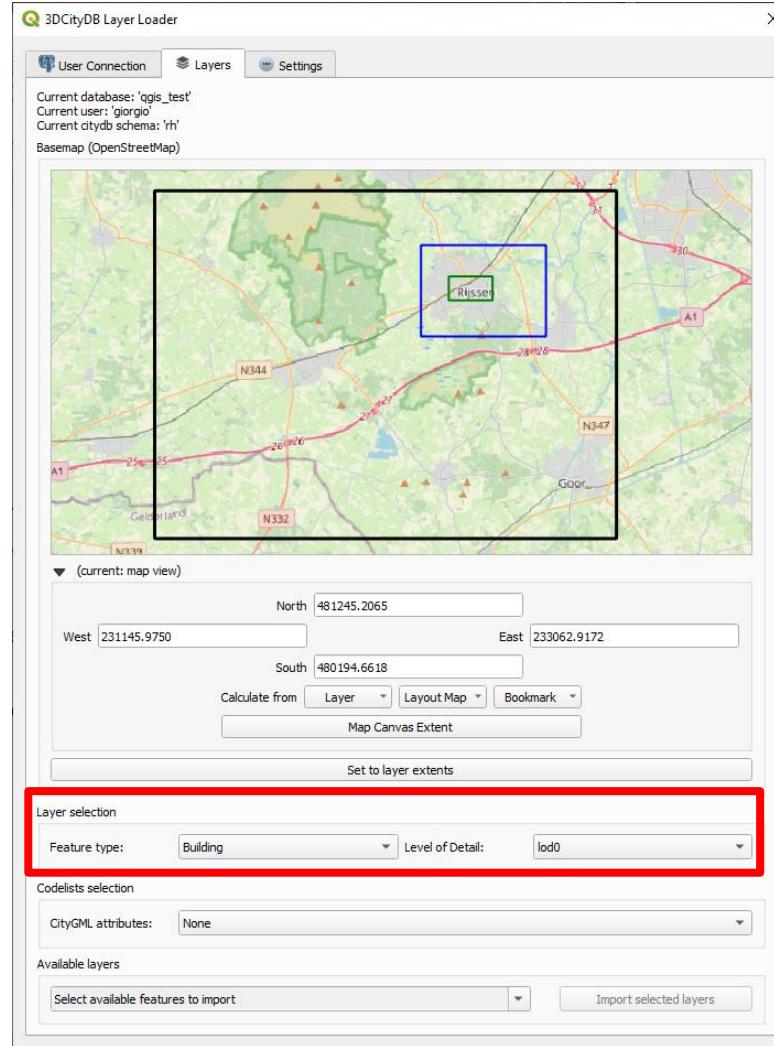


User-selected QGIS-layers extents

# Layer Loader

- 10) Select available Feature type and LoD
- Layers are grouped according to the CityGML Feature Types (e.g. "Bridge", "Building", "Tunnel", "Relief", etc.)

**Behind the scenes:** The plugin shows only the available Feature types and LoDs of data *within* the QGIS extents (**green bounding box**).



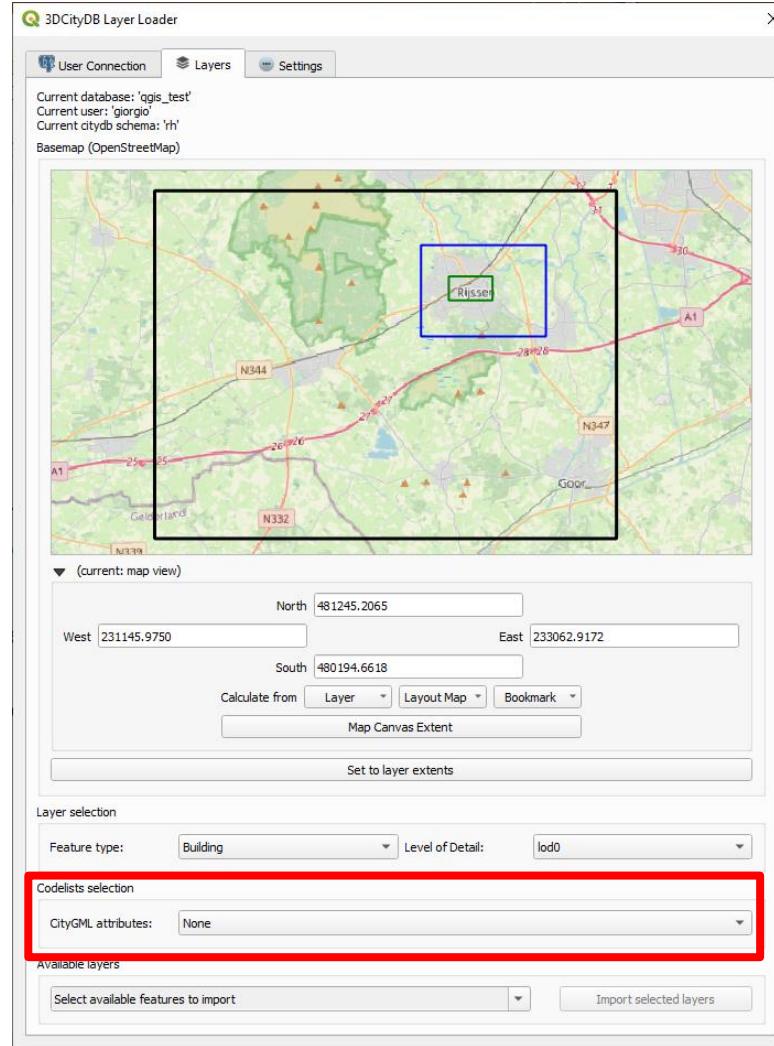
Motivation  
Plugin overview  
Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

# Layer Loader

11) The **Codelist selection** allows to optionally load codelists and set up the attribute forms accordingly (see next slides about "Use in QGIS" for more details)



# Layer Loader

Motivation  
Plugin overview  
Software installation  
**Software use**

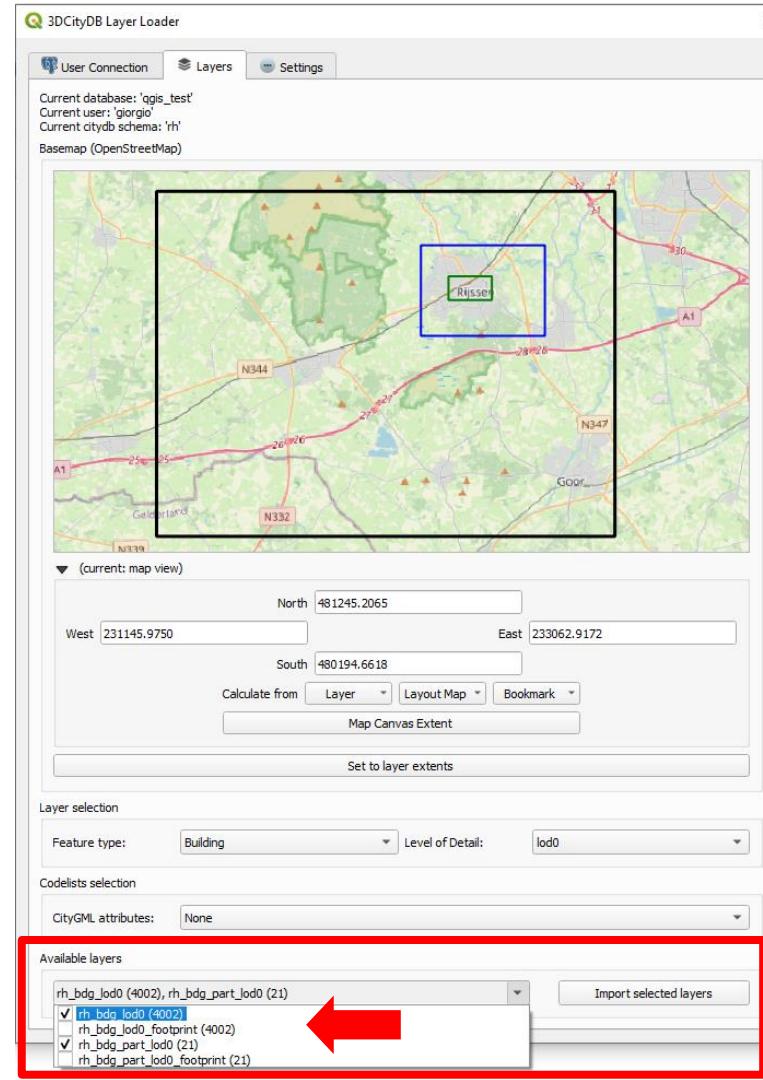
- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

## 12) Select the layer(s) to import into QGIS

- The number of available features is shown next to the layer name

**Behind the scenes:** The plugin shows only the available Layers *within* the QGIS extents (**green bounding box**).

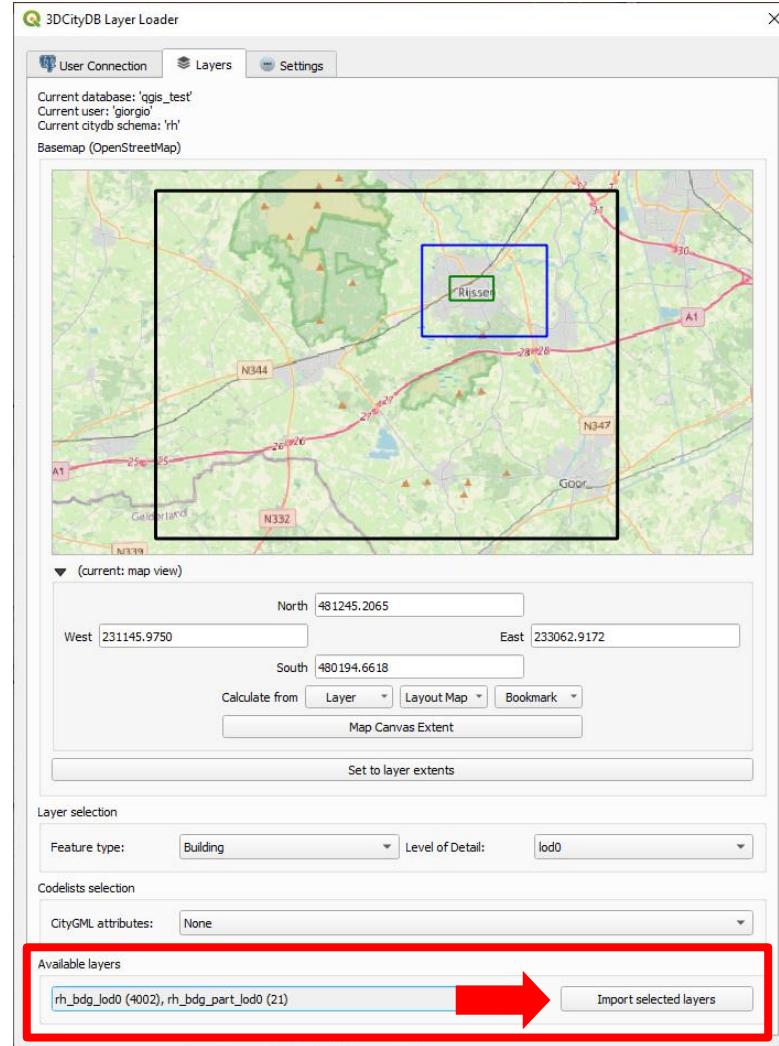


# Layer Loader

13) Import the selected layers to QGIS by clicking on the **Import selected layers** button

- The import operation can be repeated with different layers
- The layers will be automatically added to the QGIS Layers Tree / Table of Contents
- The Plugin window can be closed, the connection parameters and settings will be kept until the connection is intentionally closed by the user (in the "User Connection" tab)

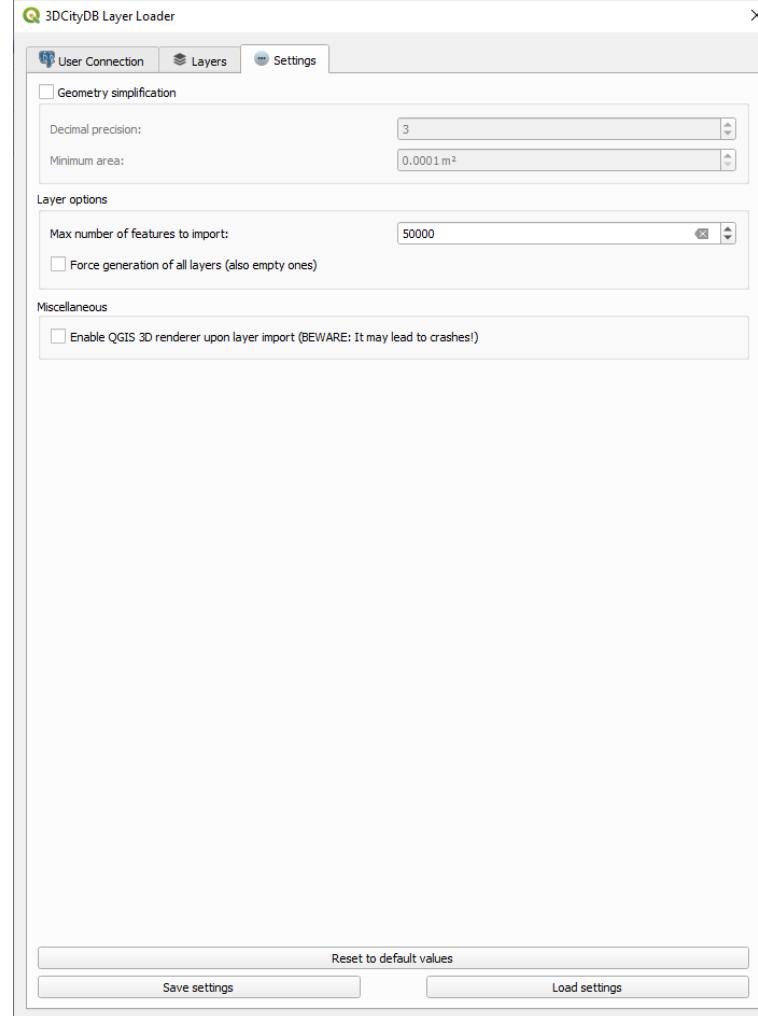
**Nota bene:** Layers that have already been loaded can be selected, but won't be loaded again



# Layer Loader

The "Settings" tab allows to enable specific options.

- The **Geometry simplification box** contains details about the coordinates precisions and the minimum area of the geometries to be generated in the materialized views
- The **Layer options box** allows to set the maximum number of features to be imported in each import action and to force the generation of the empty layers
- In the **Miscellaneous box**, the user can force the 3D rendered to be enabled upon import of the selected layers, although this might lead to instabilities (see next slides)
- Settings can be saved, (re)loaded and reset to the default values.



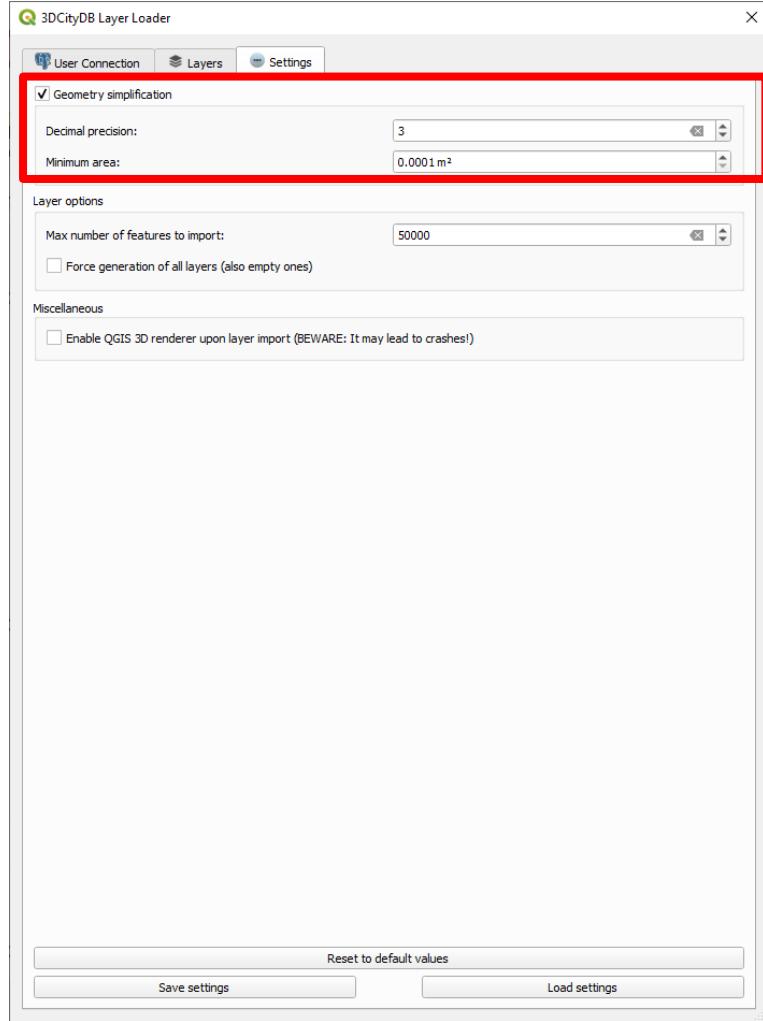
# Geometry simplification

In order to simplify geometries and (partially) cope with the 3D visualisation issues of the 3D View Map in QGIS, the user can set some simplification parameters *before* generating the layers.

All polygons composing the geometries will be checked. The user can set the number of decimal positions in the coordinates. Resulting degenerate geometries are filtered out. The second parameter filters out all polygons smaller than the chosen threshold.

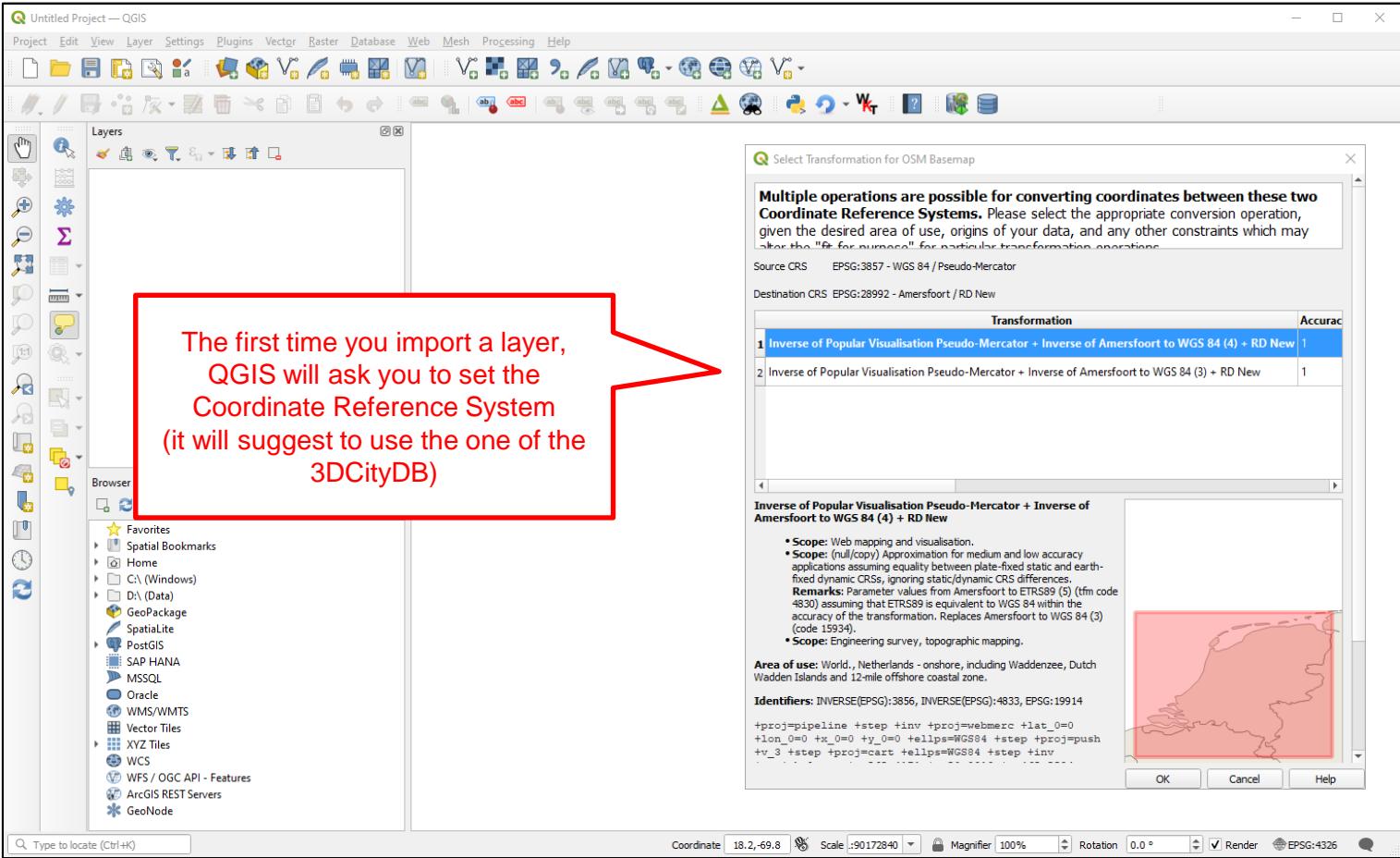
**Beware!** This operation:

- can significantly increase the time needed to refresh the layers
- does NOT change the original data in the database!



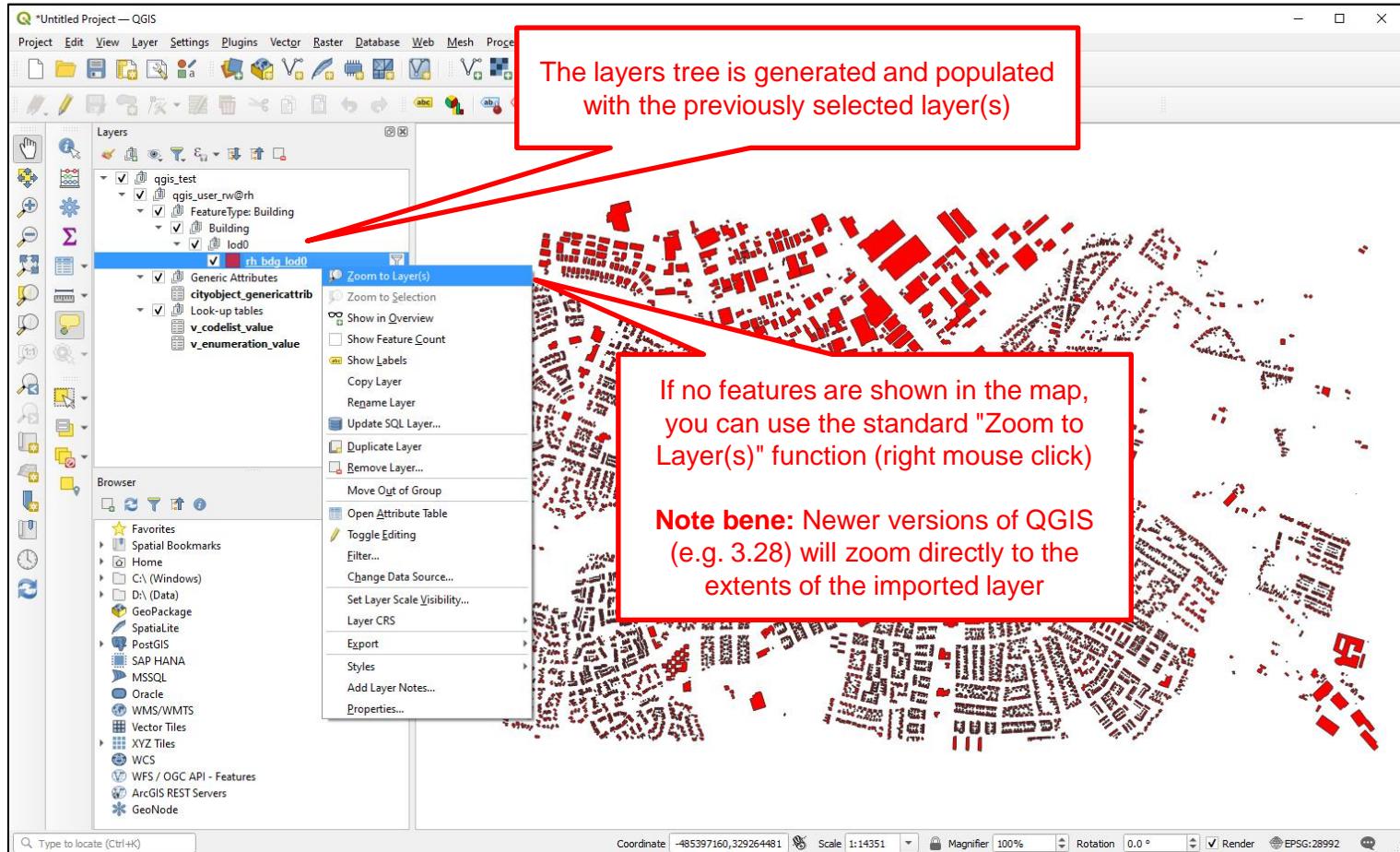
# Use in QGIS: Layers

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



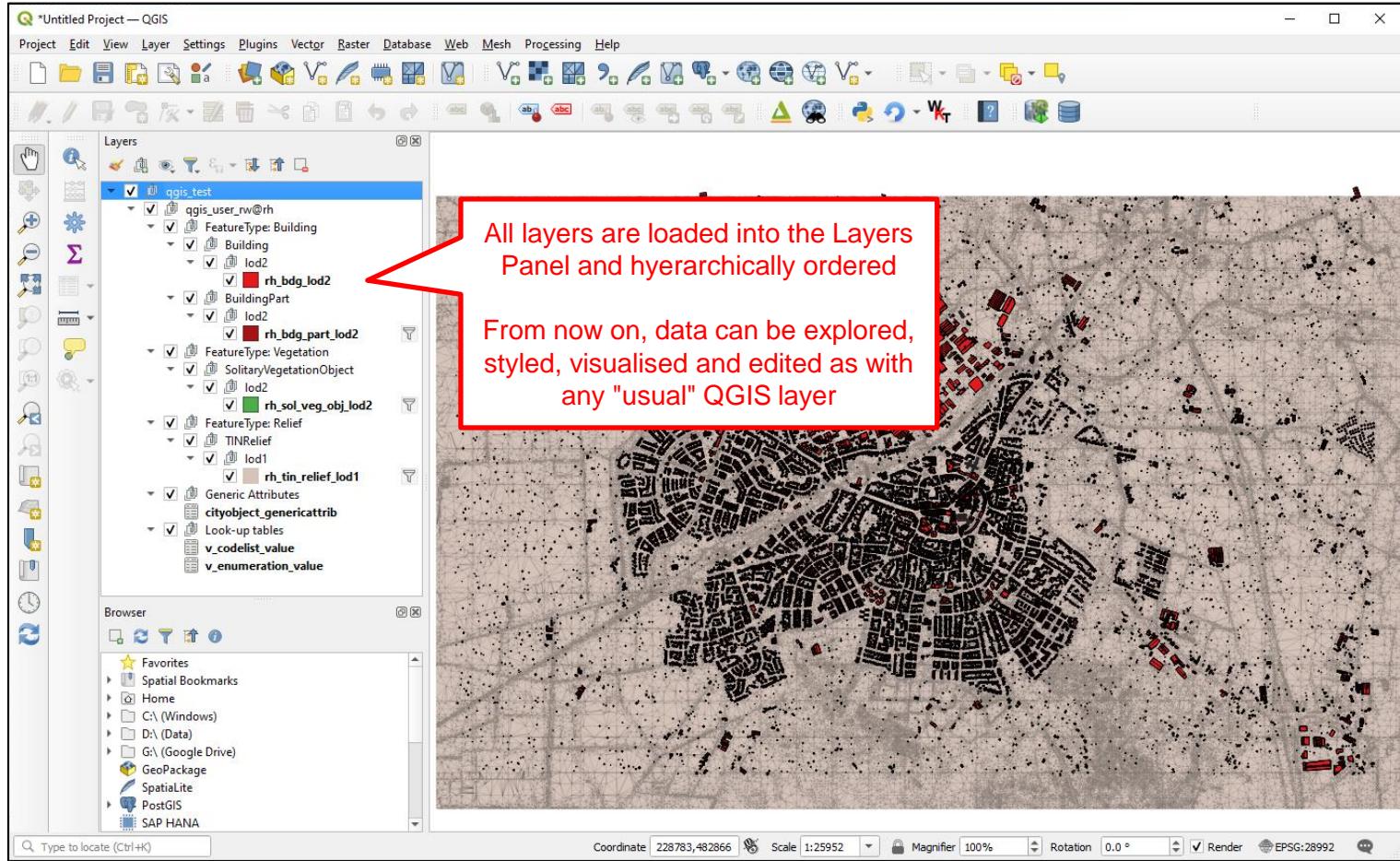
# Use in QGIS: Layers

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



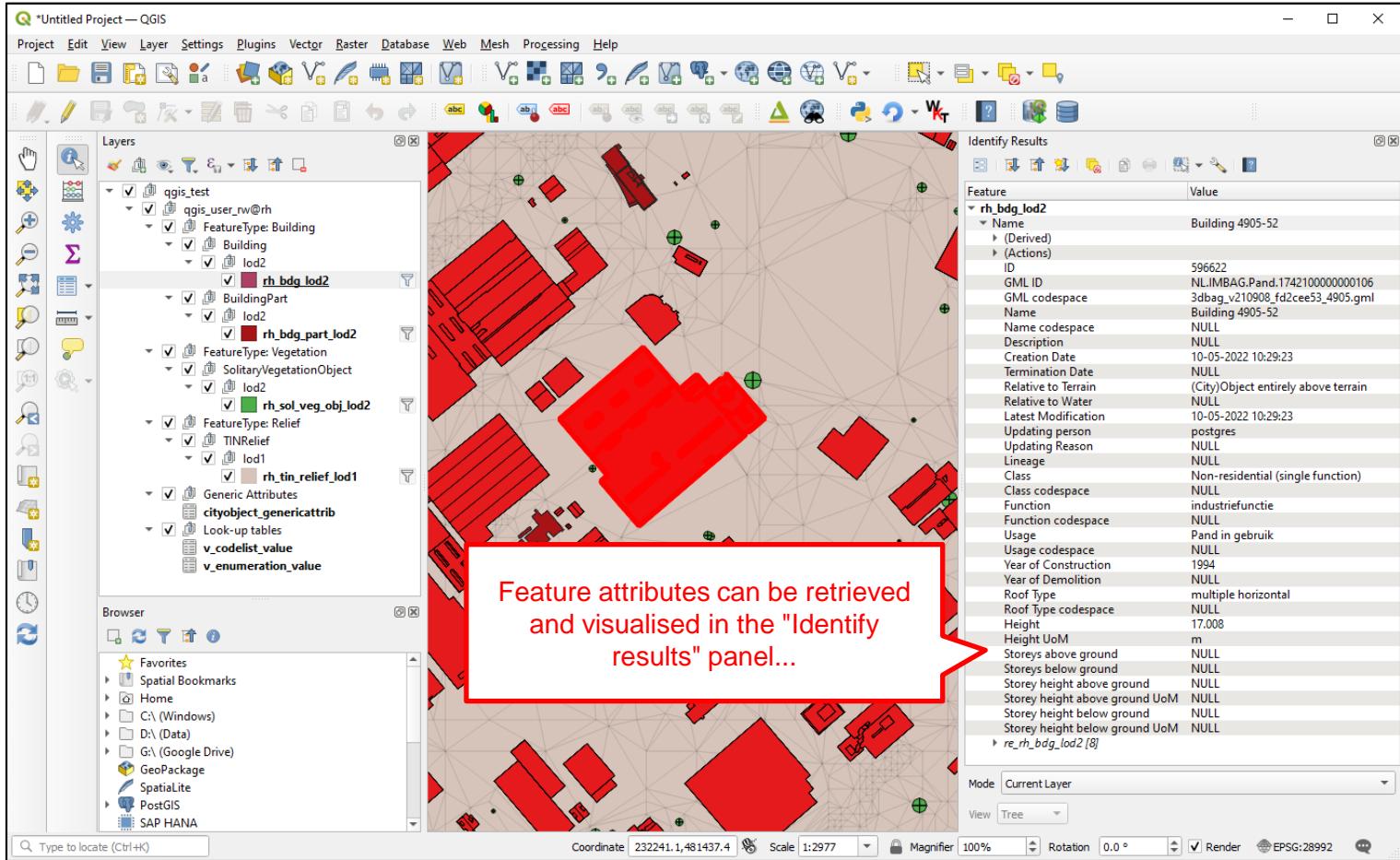
# Use in QGIS: Layers

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



# Use in QGIS: Attributes

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



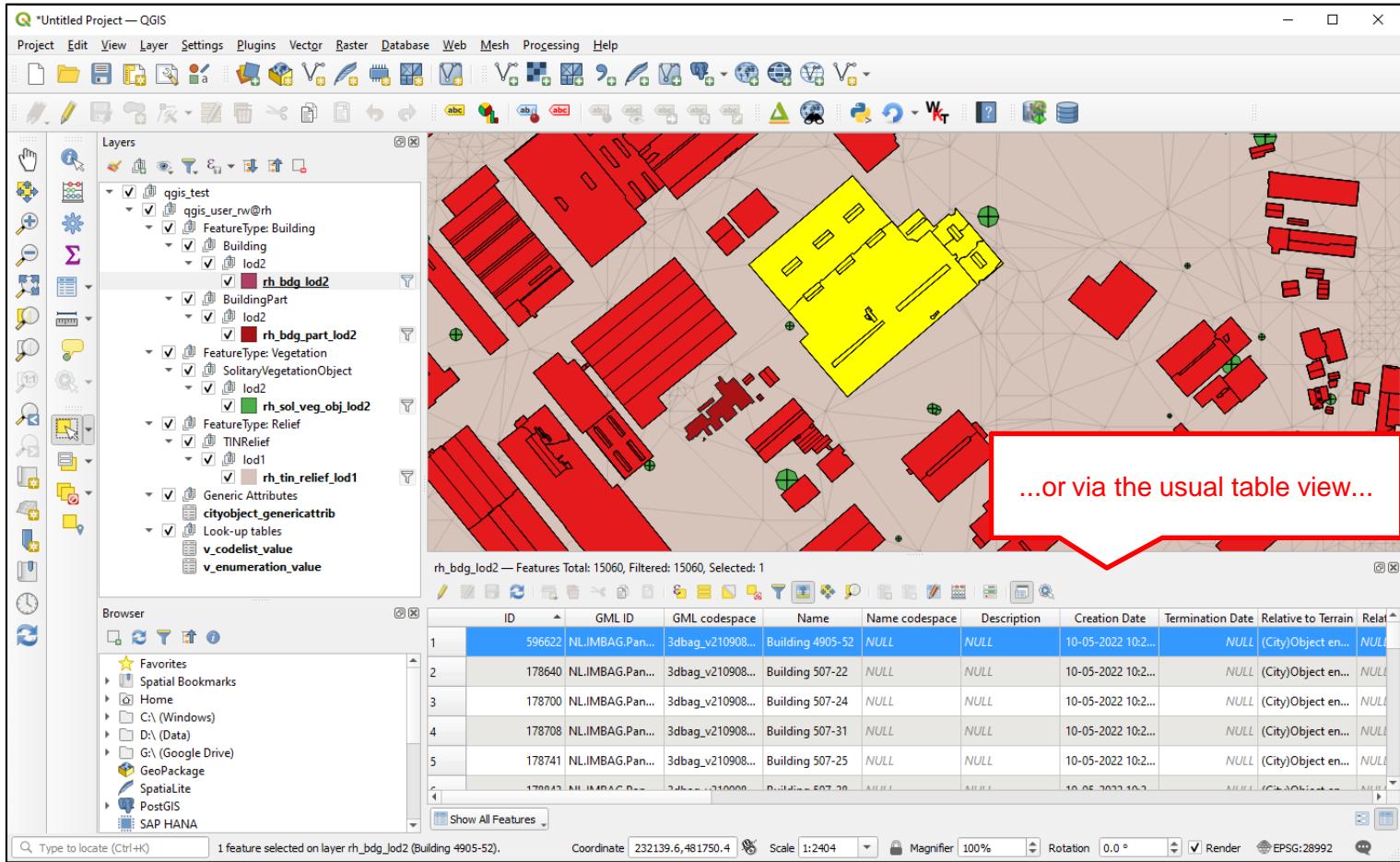
The screenshot shows the QGIS interface with a project titled "Untitled Project — QGIS". The left sidebar contains the "Layers" panel, which lists several layers including "qgis\_test", "qgis\_user\_nv@rh", "Building", "Vegetation", "Relief", and "Generic Attributes". The "Identify Results" panel on the right displays the attributes for a selected feature, specifically a building. A red callout box highlights the text: "Feature attributes can be retrieved and visualised in the \"Identify results\" panel...".

Feature	Value
rh_bdg_lod2	
Name	Building 4905-52
(Derived)	
Actions	
ID	596622
GML ID	NLIMBAG.Pand.174210000000106
GML codespace	3dbag_v210908_fd2ceef53_4905.gml
Name	Building 4905-52
Name codespace	NULL
Description	NULL
Creation Date	10-05-2022 10:29:23
Termination Date	NULL
Relative to Terrain	(City)Object entirely above terrain
Relative to Water	NULL
Latest Modification	10-05-2022 10:29:23
Updating person	postgres
Updating Reason	NULL
Lineage	NULL
Class	Non-residential (single function)
Class codespace	NULL
Function	industriefunctie
Function codespace	NULL
Usage	Pand in gebruik
Usage codespace	NULL
Year of Construction	1994
Year of Demolition	NULL
Roof Type	multiple horizontal
Roof Type codespace	NULL
Height	17.008
Height UoM	m
Storeys above ground	NULL
Storeys below ground	NULL
Storey height above ground	NULL
Storey height above ground UoM	NULL
Storey height below ground	NULL
Storey height below ground UoM	NULL
re_rh_bdg_lod2 [8]	

Mode: Current Layer  
View: Tree  
Coordinate: 232241.1, 481437.4  
Scale: 1:2977  
Magnifier  
100%  
Rotation: 0.0°  
Render  
EPSG:28992

# Use in QGIS: Attributes

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



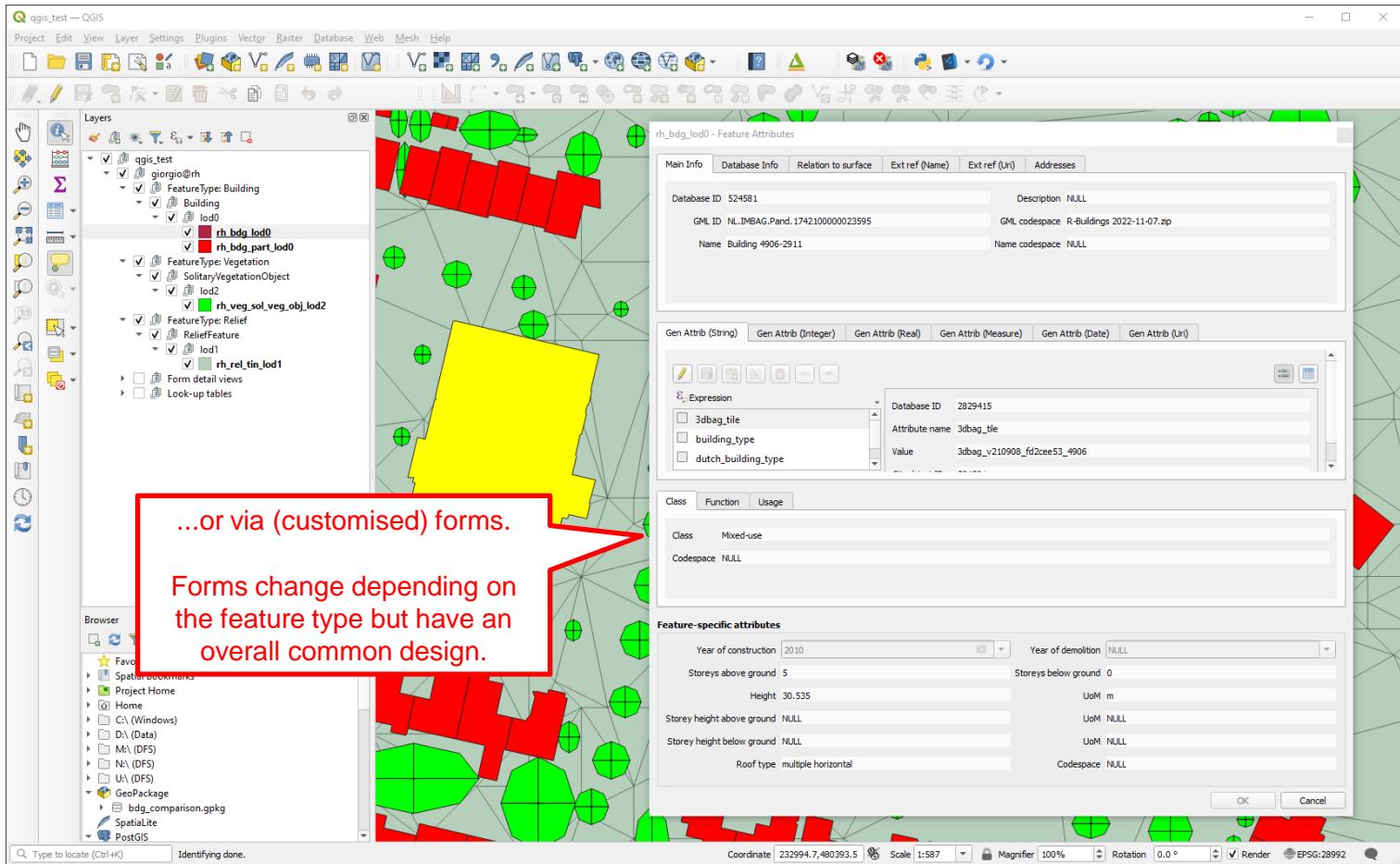
The screenshot shows the QGIS interface with a map view and a table view. The map view displays several red building footprints and a yellow polygon representing vegetation. A callout box highlights the table view, which shows a list of building features with columns for ID, GML ID, GML codespace, Name, Name codespace, Description, Creation Date, Termination Date, and Relative to Terrain. The first feature in the table is highlighted with a blue background.

ID	GML ID	GML codespace	Name	Name codespace	Description	Creation Date	Termination Date	Relative to Terrain	Relat
1	596622	NL.1MBAG.Pan...	3dbag_v210908..	Building 4905-52	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...
2	178640	NL.1MBAG.Pan...	3dbag_v210908..	Building 507-22	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...
3	178700	NL.1MBAG.Pan...	3dbag_v210908..	Building 507-24	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...
4	178708	NL.1MBAG.Pan...	3dbag_v210908..	Building 507-31	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...
5	178741	NL.1MBAG.Pan...	3dbag_v210908..	Building 507-25	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...
	178640	NL.1MBAG.Pan...	3dbag_v210908..	Building 507-20	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...

...or via the usual table view...

# Use in QGIS: Attributes

Motivation  
 Plugin overview  
 Software installation  
**Software use**  
 • Layer Loader  
 • Bulk Deleter  
 • About, etc.  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources



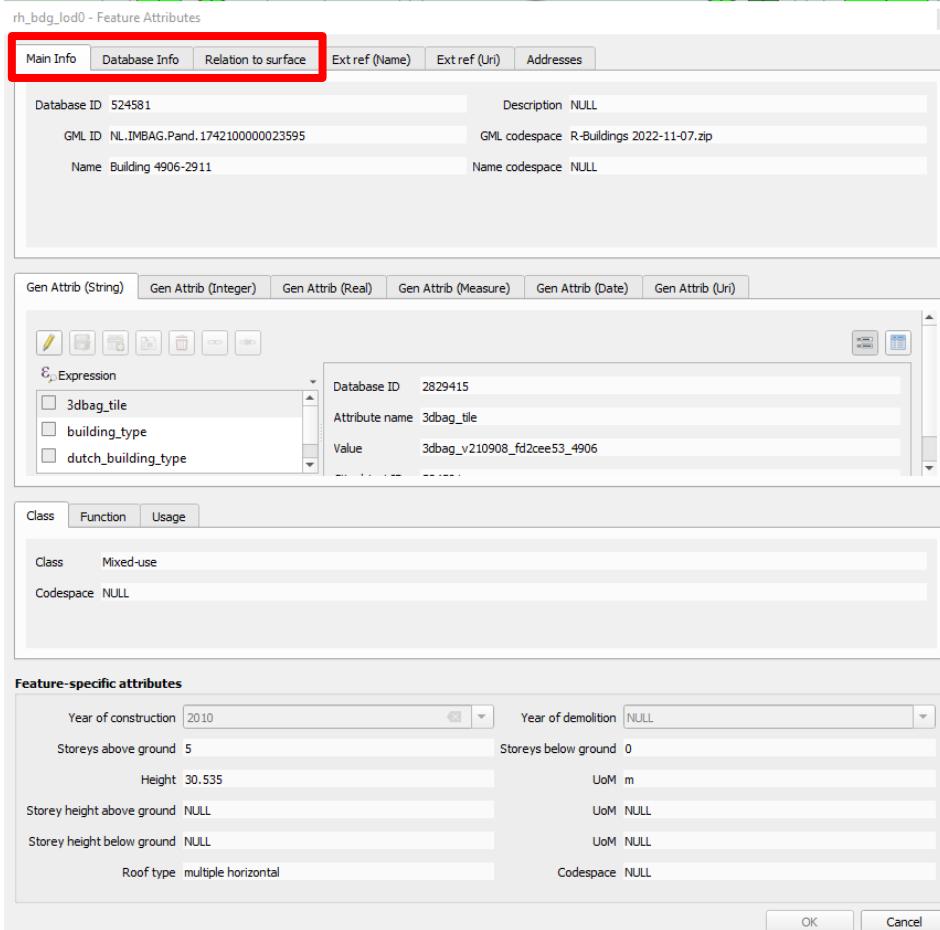
# Use in QGIS: Attributes

Motivation  
Plugin overview  
Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

Attributes are grouped into tabs.



The screenshot shows the QGIS Attribute Editor for a feature named "Building 4906-2911". The top tab bar includes "Main Info", "Database Info", "Relation to surface", "Ext ref (Name)", "Ext ref (Uri)", and "Addresses". The "Main Info" tab is selected, displaying basic information like Database ID (524581), GML ID (NL.IMBAG.Pand.1742100000023595), and Name (Building 4906-2911). Below this is a "Gen Attrib" section with tabs for String, Integer, Real, Measure, Date, and Uri. The "String" tab is active, showing an expression builder with fields for Database ID (2829415), Attribute name (3dbag\_tile), and Value (3dbag\_v210908\_fd2cee53\_4906). Further down are tabs for Class (Mixed-use) and Function. The bottom section, titled "Feature-specific attributes", contains fields for Year of construction (2010), Year of demolition (NULL), Storeys above ground (5), Height (30.535), Storey height above ground (NULL), Storey height below ground (NULL), Roof type (multiple horizontal), and Codespace (NULL).

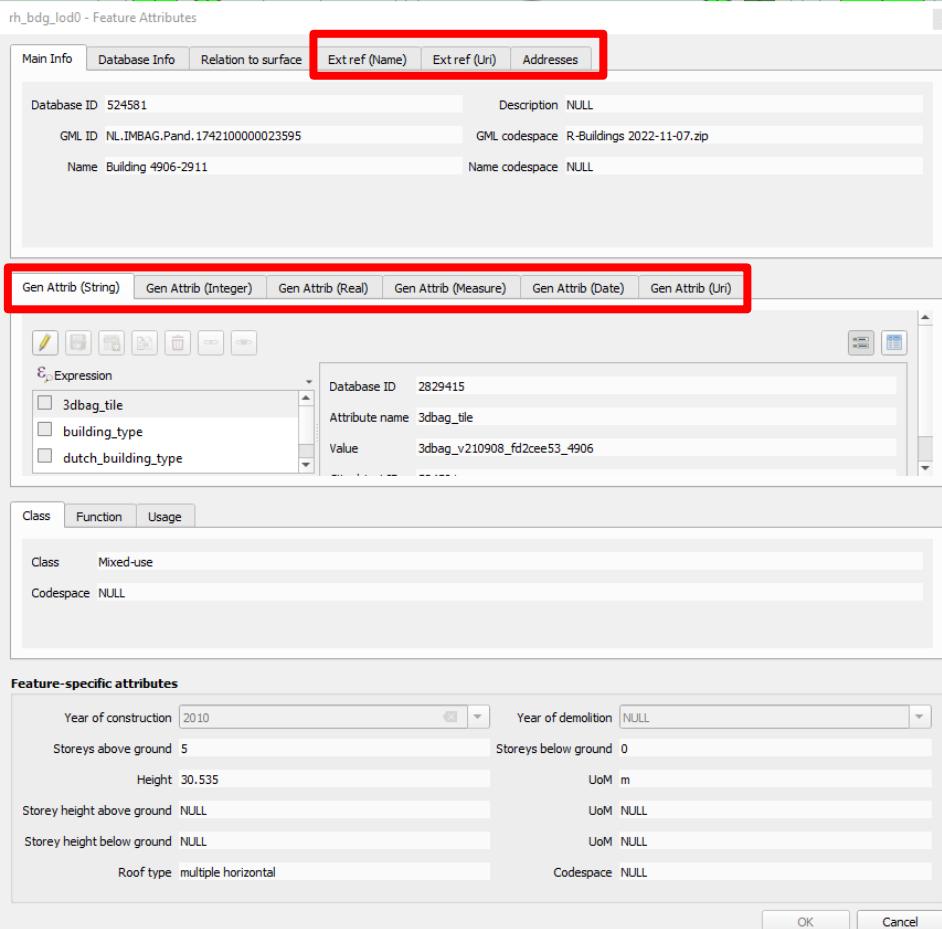
Motivation  
Plugin overview  
Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

# Use in QGIS: Attributes

Additional related tables are connected (e.g. External References, Addresses, Generic Attributes)



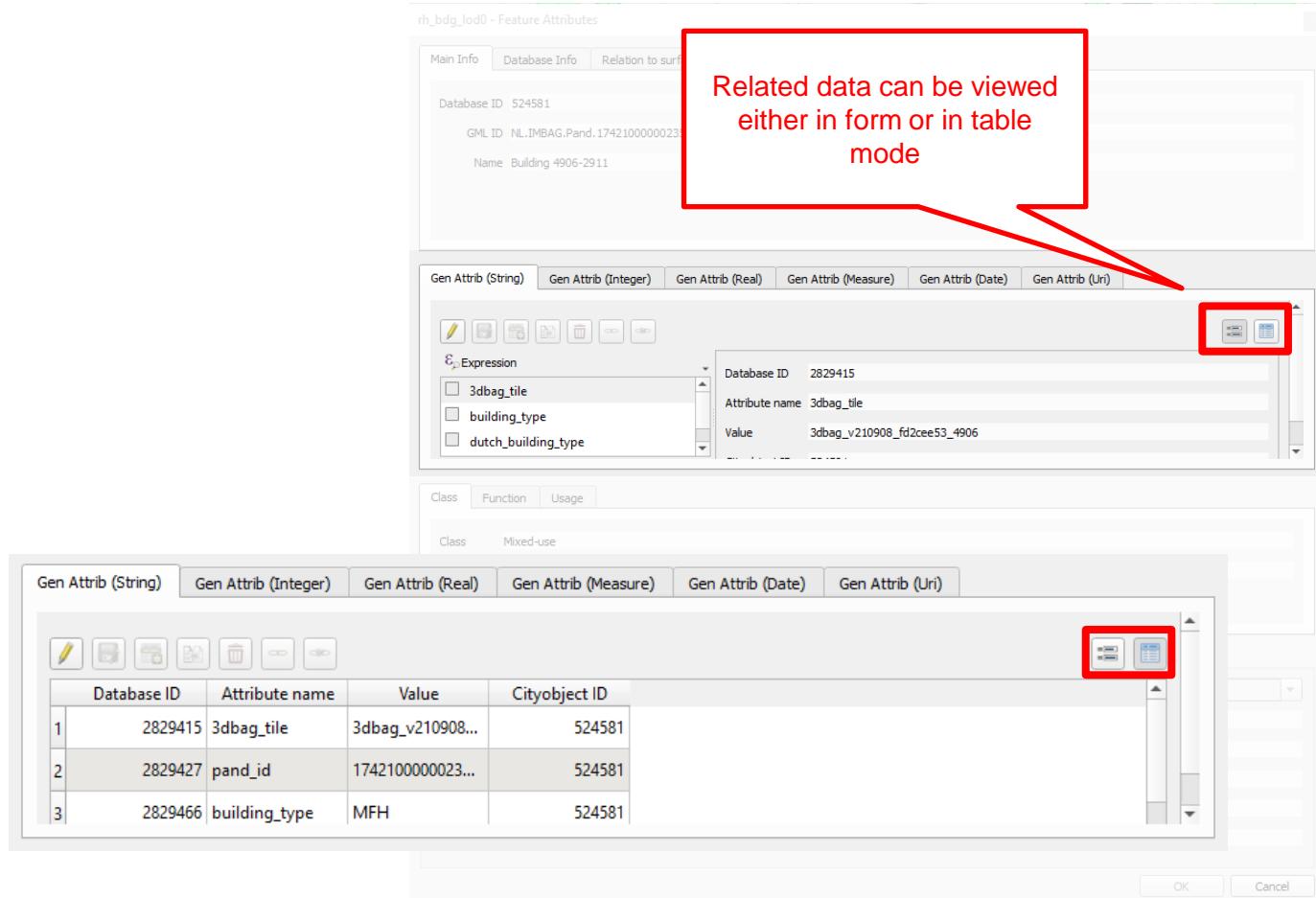
The screenshot shows the 'Feature Attributes' dialog box for a layer named 'rh\_bdg\_lod0'. The top navigation bar includes tabs for 'Main Info', 'Database Info', 'Relation to surface', 'Ext ref (Name)', 'Ext ref (Uri)', and 'Addresses'. The 'Ext ref (Name)' tab is highlighted with a red border. Below the tabs, there are sections for 'Database ID', 'GML ID', 'Name', 'Description', and 'Codespace'. The 'Ext ref (Name)' section displays a table with columns for 'Attribute name' and 'Value'. The 'Attribute name' column lists '3dbag\_tile', 'building\_type', and 'dutch\_building\_type'. The 'Value' column shows the corresponding values. The 'Addresses' tab is also visible at the bottom of the dialog. At the bottom right, there are 'OK' and 'Cancel' buttons.

Motivation  
Plugin overview  
Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

# Use in QGIS: Attributes

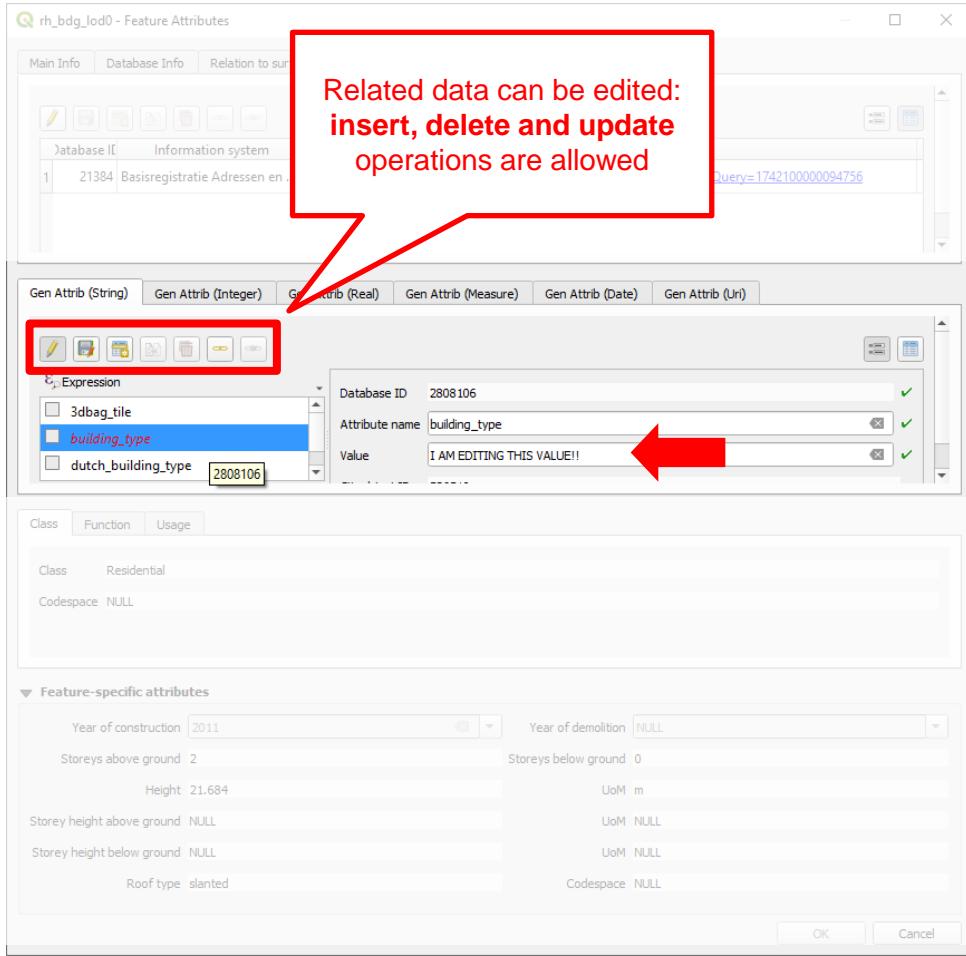


The screenshot shows the QGIS Attribute Editor for a feature named "Building 4906-2911". The top part displays "Main Info" with fields like Database ID (524581), GML ID (NL.IMBAG.Pand.1742100000023), and Name. A red box highlights this area with the text: "Related data can be viewed either in form or in table mode". Below this, the "Gen Attrib (String)" tab is selected, showing an expression builder with fields 3dbag\_tile, building\_type, and dutch\_building\_type, and their corresponding values. The bottom part shows a table view with columns Database ID, Attribute name, Value, and Cityobject ID, containing three rows of data. Red boxes highlight the "Form" and "Table" buttons in the top right of each tab section.

Database ID	Attribute name	Value	Cityobject ID
1	2829415	3dbag_tile	3dbag_v210908...
2	2829427	pand_id	1742100000023...
3	2829466	building_type	MFH

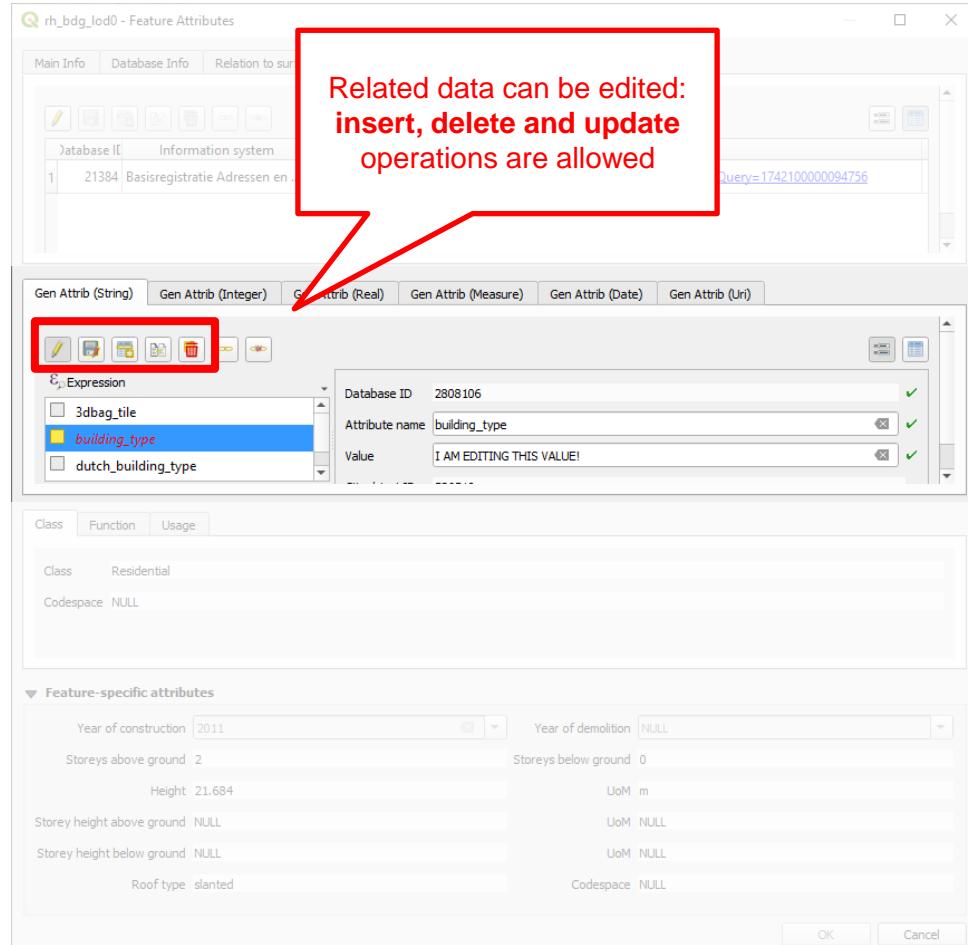
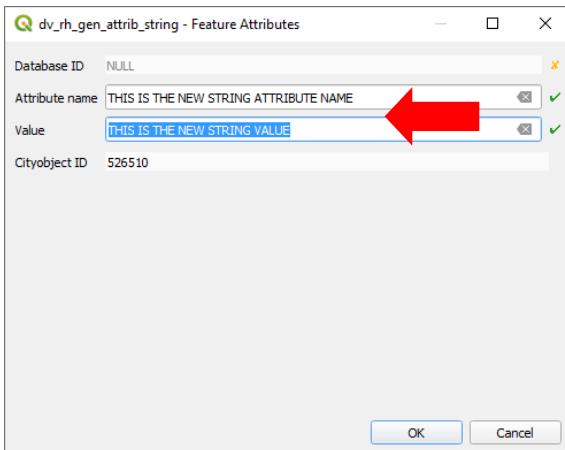
- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

# Use in QGIS: Attributes



- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

# Use in QGIS: Attributes

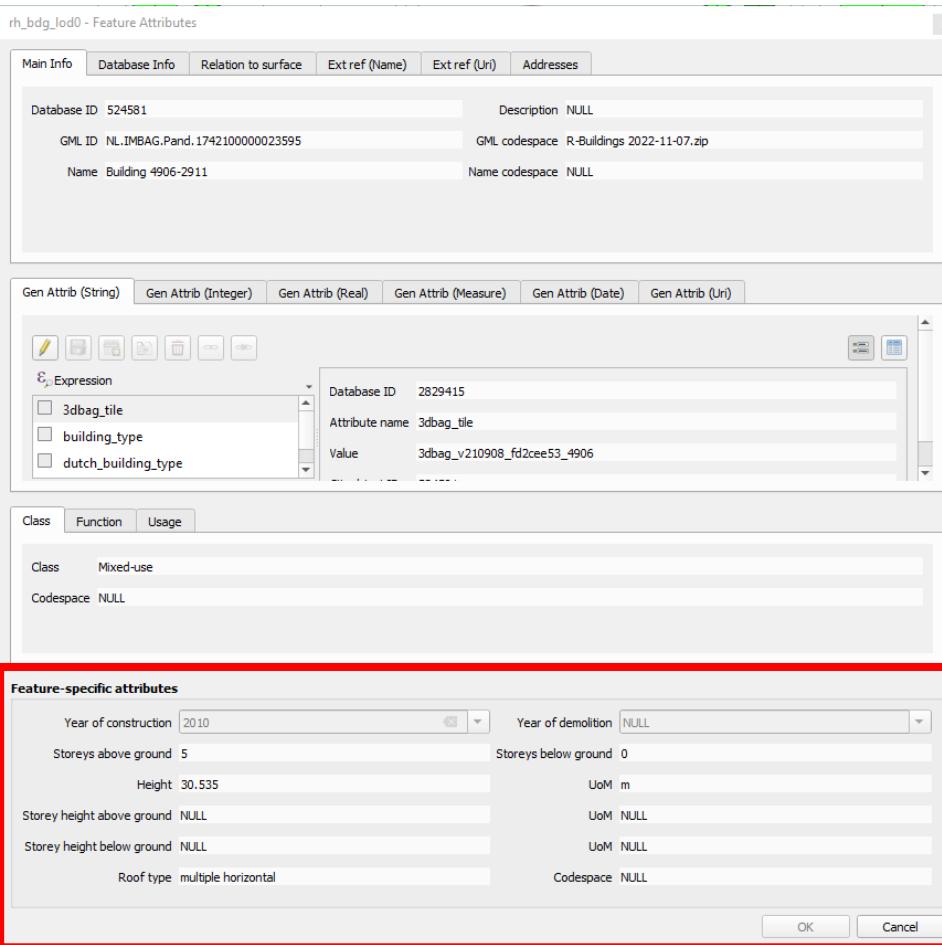


Motivation  
Plugin overview  
Software installation  
**Software use**

- Layer Loader
- Bulk Deleter
- About, etc.

Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
Resources

# Use in QGIS: Attributes



The screenshot shows the 'Feature Attributes' dialog for a feature named 'Building 4906-2911'. The 'Main Info' tab displays basic information like Database ID (524581), GML ID (NL.IMGAG.Pand.1742100000023595), and Name (Building 4906-2911). The 'Feature-specific attributes' tab is highlighted with a red box and contains fields for Year of construction (2010), Storeys above ground (5), Height (30.535), and other building characteristics.

rh\_bdg\_lod0 - Feature Attributes

Main Info Database Info Relation to surface Ext ref (Name) Ext ref (Uri) Addresses

Database ID 524581 Description NULL

GML ID NL.IMGAG.Pand.1742100000023595 GML codespace R-Buildings 2022-11-07.zip

Name Building 4906-2911 Name codespace NULL

Gen Attrib (String) Gen Attrib (Integer) Gen Attrib (Real) Gen Attrib (Measure) Gen Attrib (Date) Gen Attrib (Uri)

Expression

3dbag\_tile  
building\_type  
dutch\_building\_type

Database ID 2829415 Attribute name 3dbag\_tile Value 3dbag\_v210908\_fd2cee53\_4906

Class Function Usage

Class Mixed-use Codespace NULL

Feature-specific attributes

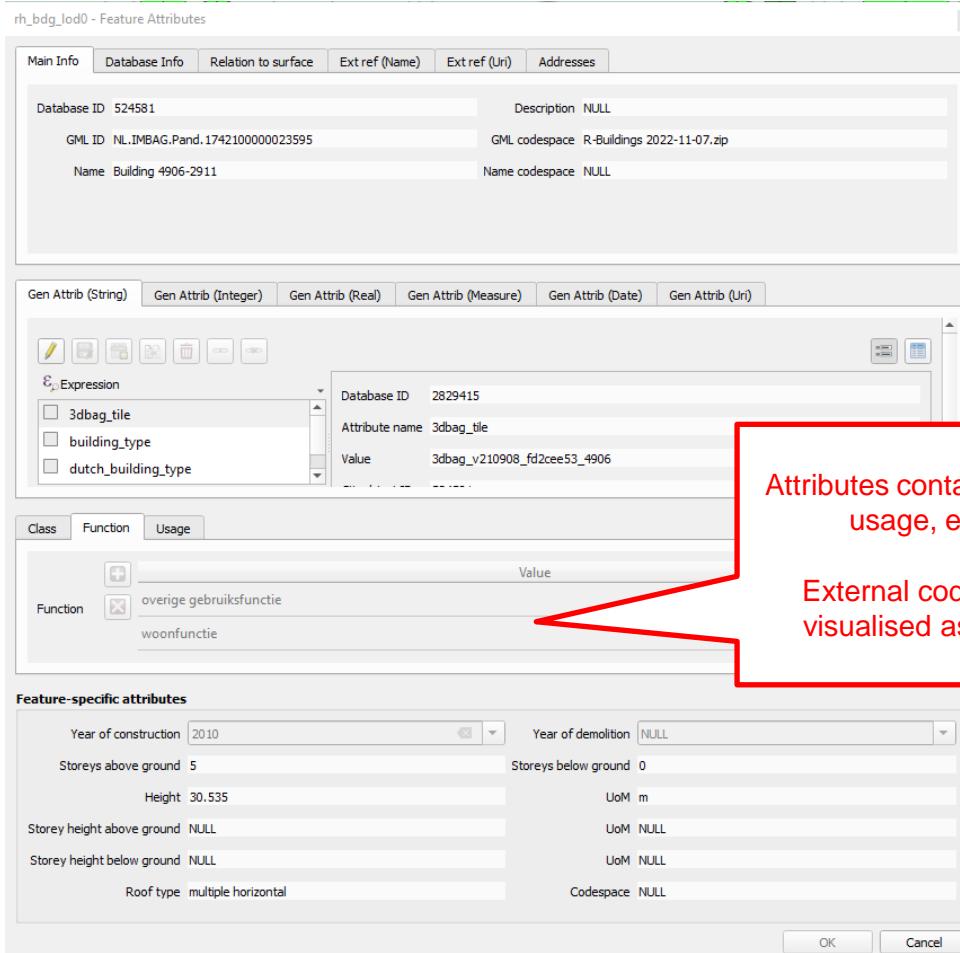
Year of construction	2010	Year of demolition	NULL
Storeys above ground	5	Storeys below ground	0
Height	30.535	UoM	m
Storey height above ground	NULL	UoM	NULL
Storey height below ground	NULL	UoM	NULL
Roof type	multiple horizontal	Codespace	NULL

OK Cancel

Feature-specific attributes  
are grouped in the lower  
part of the form.

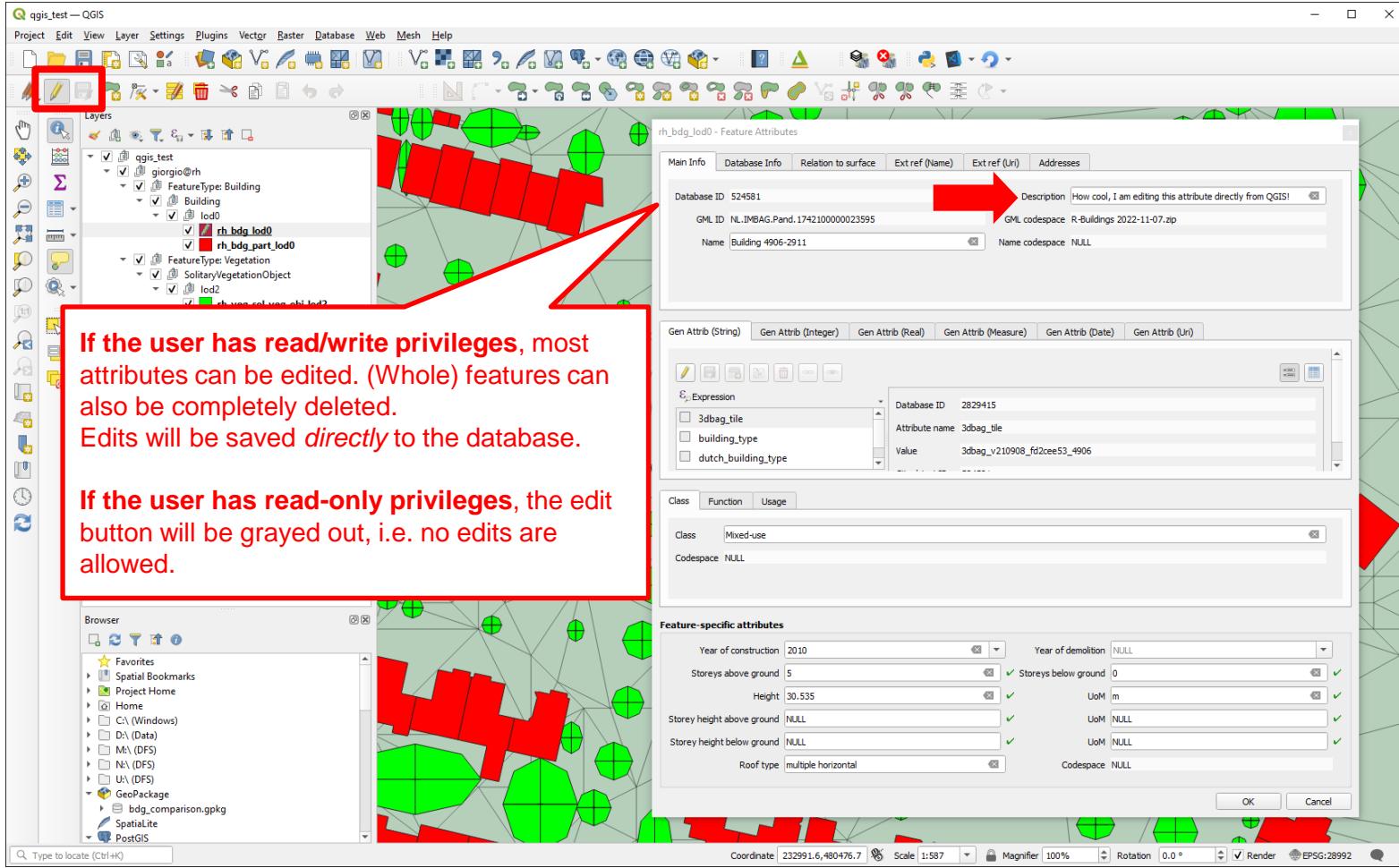
# Use in QGIS: Attributes

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



# Use in QGIS: Attribute editing

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



If the user has read/write privileges, most attributes can be edited. (Whole) features can also be completely deleted. Edits will be saved *directly* to the database.

If the user has read-only privileges, the edit button will be grayed out, i.e. no edits are allowed.

The 'Feature Attributes' dialog shows the following details for the selected building:

Attribute	Value
3dbag_tile	2829415
building_type	3dbag_v_210908_fd2cee53_4906
dutch_building_type	

Feature-specific attributes:

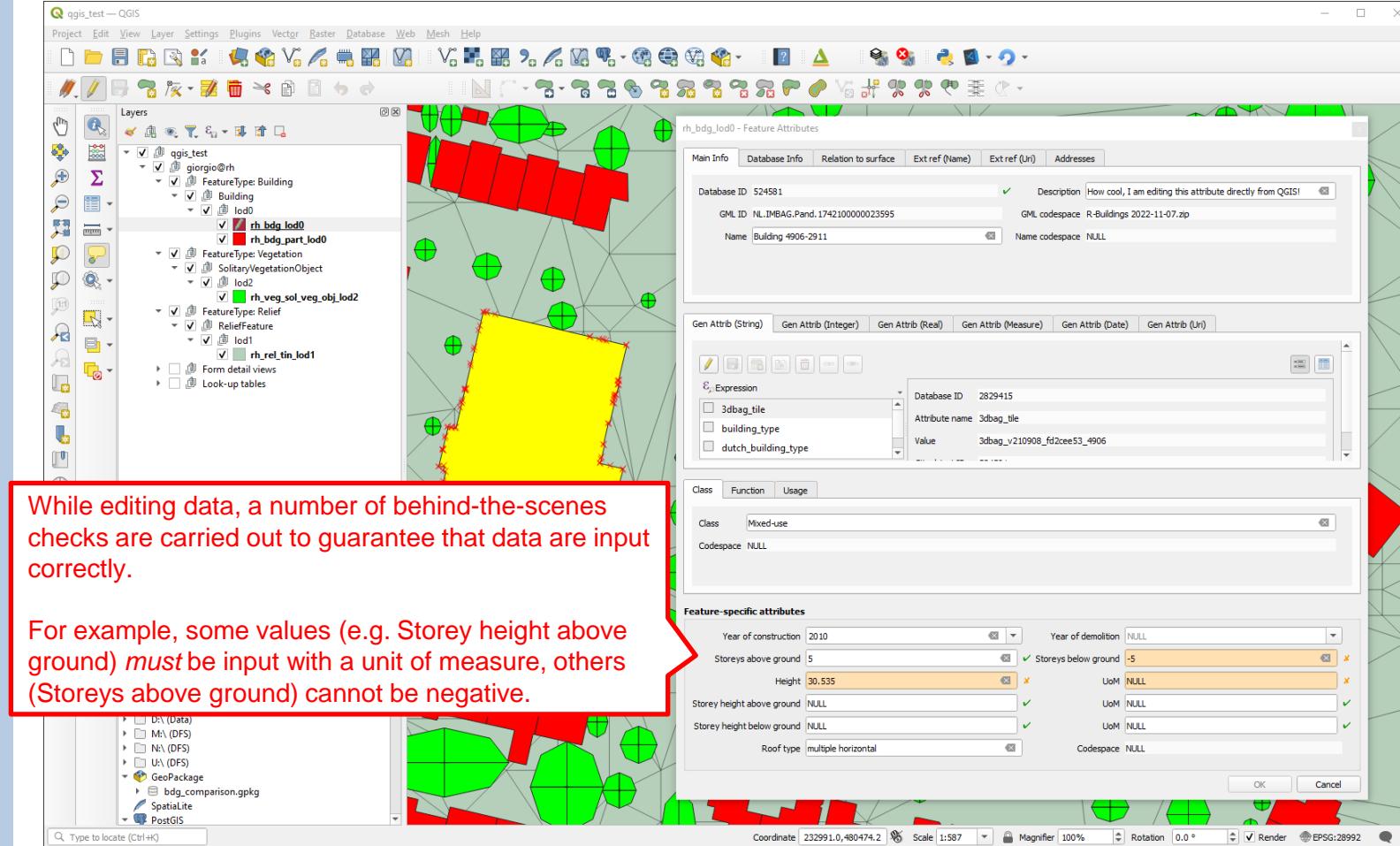
Attribute	Value
Year of construction	2010
Storeys above ground	5
Height	30.535
Storey height above ground	NULL
Storey height below ground	NULL
Roof type	multiple horizontal
Year of demolition	NULL
Storeys below ground	0
UoM	m
UoM	NULL
UoM	NULL
Codespace	NULL

# Use in QGIS: Attribute editing

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

While editing data, a number of behind-the-scenes checks are carried out to guarantee that data are input correctly.

For example, some values (e.g. Storey height above ground) *must* be input with a unit of measure, others (Storeys above ground) cannot be negative.

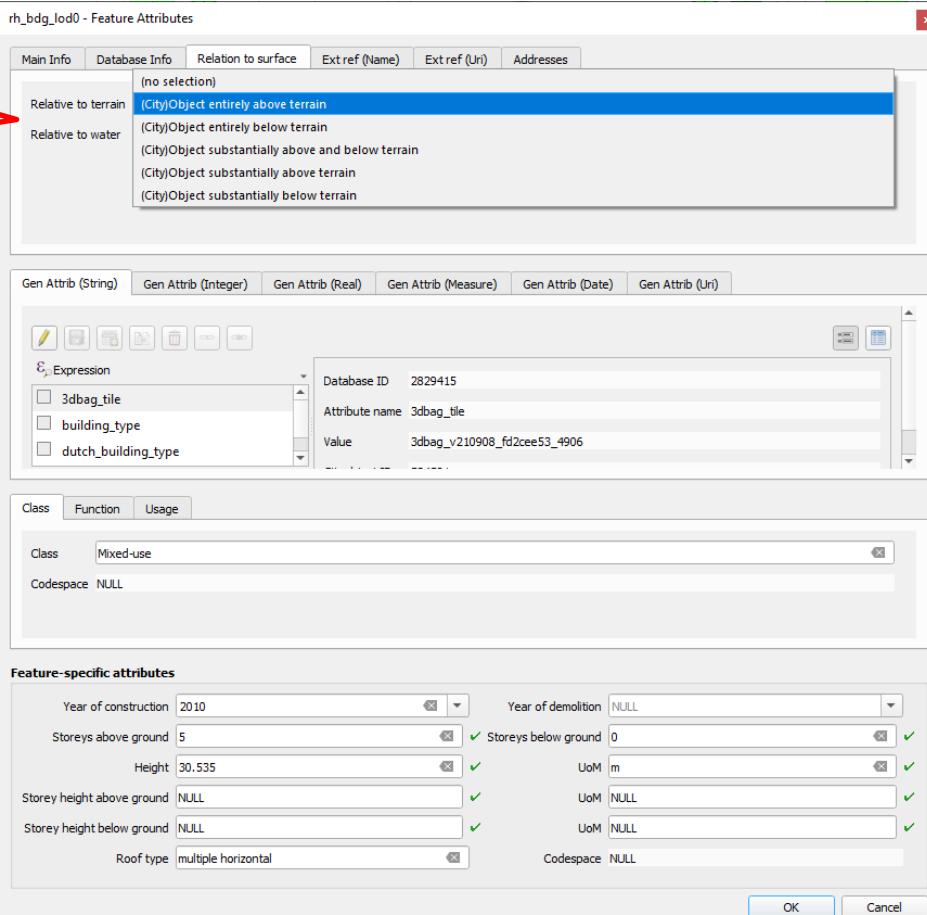


Attribute	Value	Type
Database ID	524581	
GML ID	NL.IMGAG.Pand.1742100000023595	
Name	Building 4906-2911	
Height	30.535	Real
Storeys above ground	5	Integer

# Use in QGIS: Attribute editing

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

While editing data, input of enumeration values is made via drop-down lists

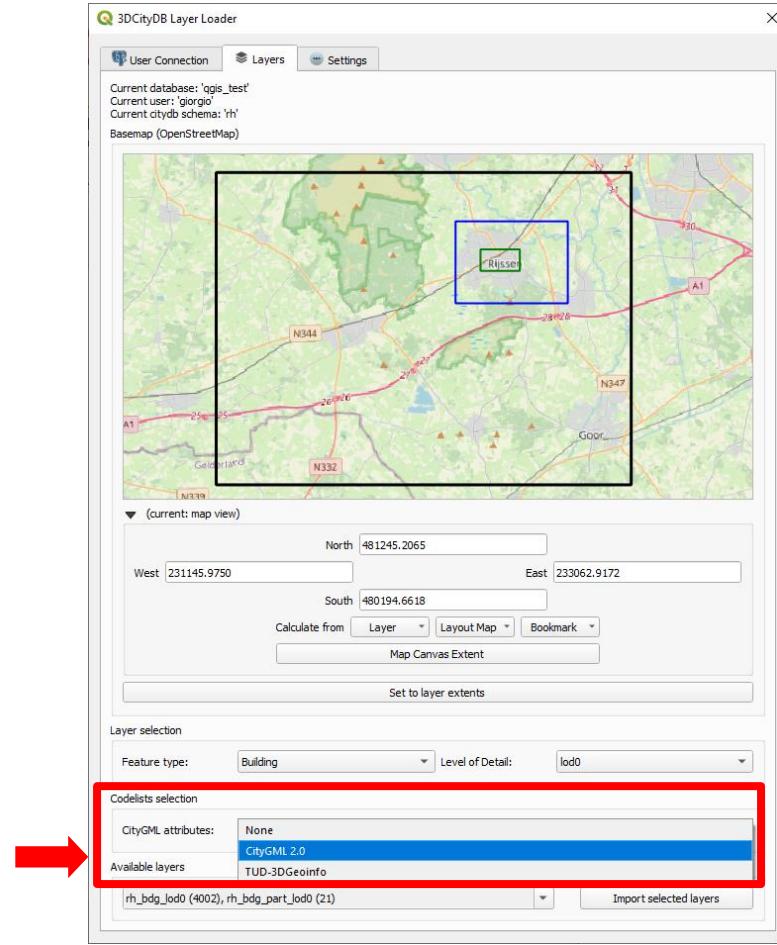


# Use in QGIS: Codelist support

Codelists can be selected upon layer import

The attribute forms will be formatted automatically into drop-down lists or multiple-selection lists

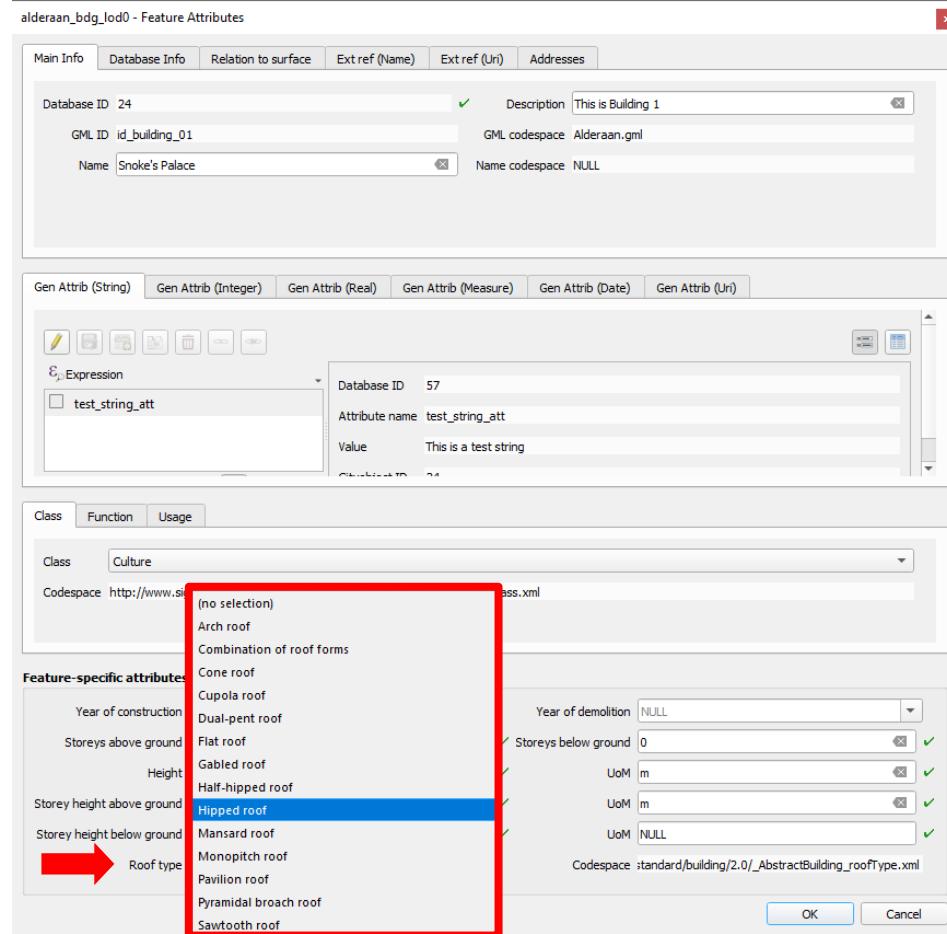
Codelists can be added and customised either by the database administrator or by the user (see **Advanced options**)



# Use in QGIS: Codelist support

In the case of a CityGML property with cardinality [0..1], the associated codelist values are presented as a **drop-down list**

Example: **Roof type**

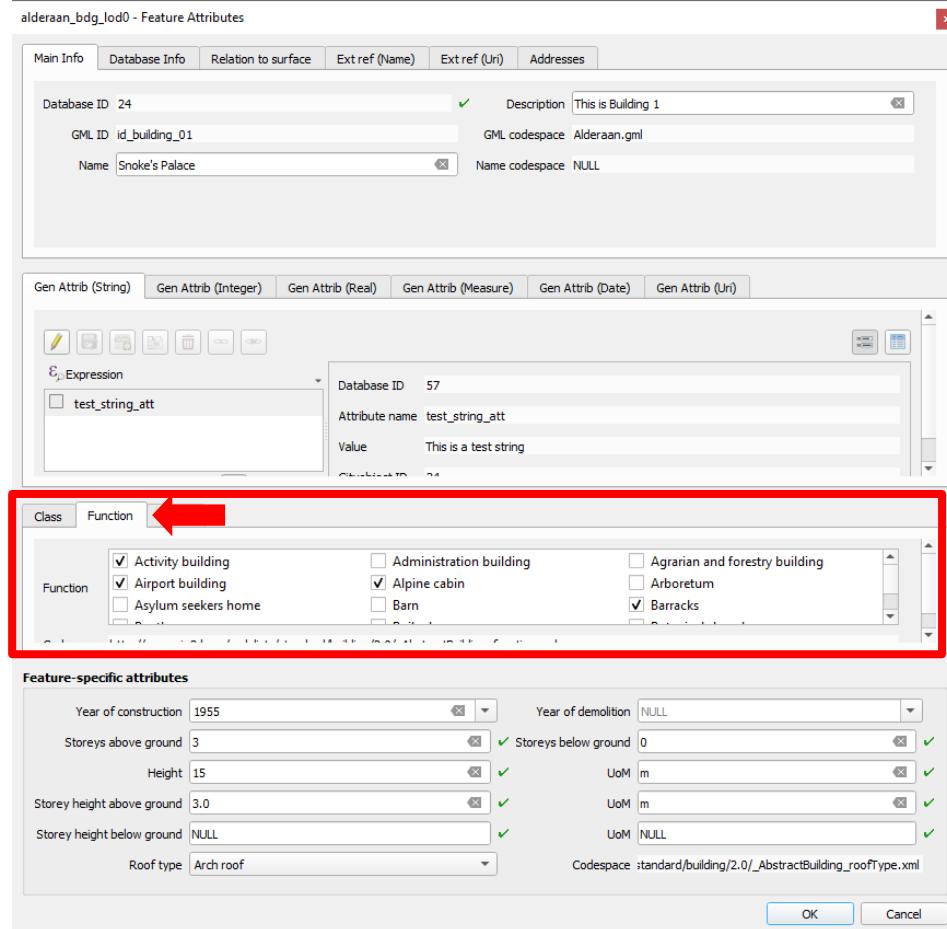


Motivation  
 Plugin overview  
 Software installation  
**Software use**  
 • Layer Loader  
 • Bulk Deleter  
 • About, etc.  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

# Use in QGIS: Codelist support

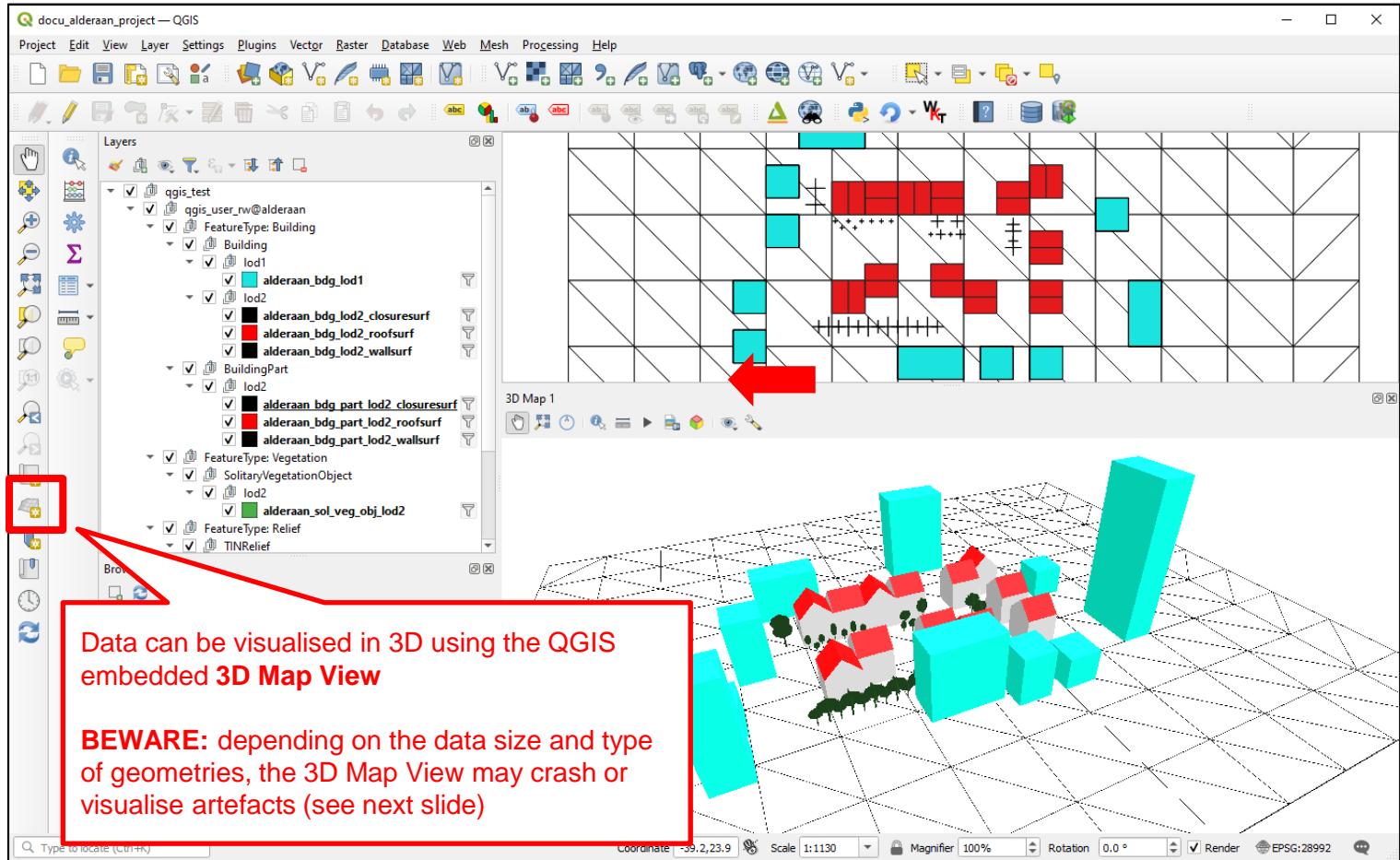
In the case of a CityGML property with cardinality [0..\*], the associated codelist values are presented as a **multiple-selection list**

Example: property (Building) function



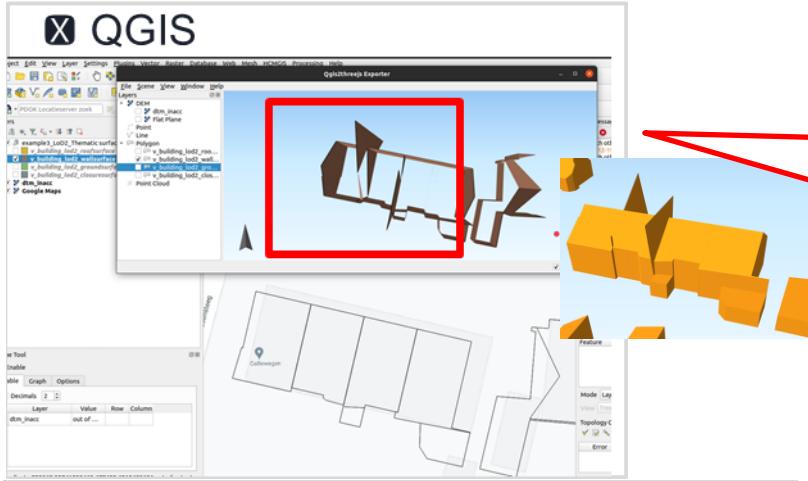
# Use in QGIS: 3D visualisation

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

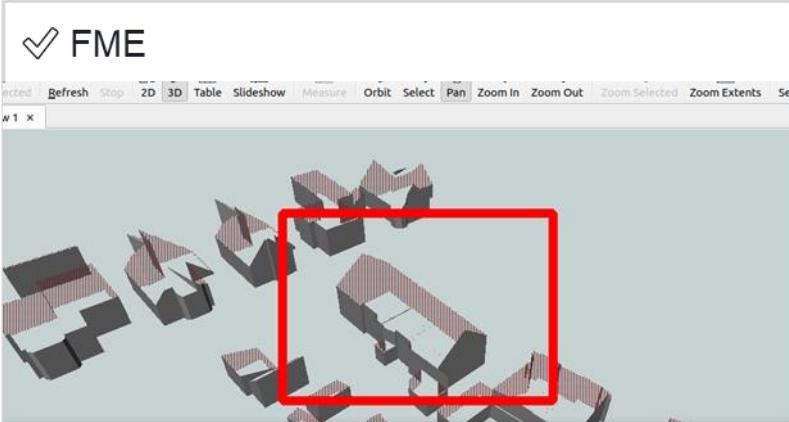


# Use in QGIS: 3D visualisation

Motivation  
 Plugin overview  
 Software installation  
**Software use**  
 • Layer Loader  
 • Bulk Deleter  
 • About, etc.  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources



In our tests, such artefacts are a 3D visualisation issue (QGIS 3D renderer?) and *not* related to the actual data. 3D visualisation in FME and in Google Earth shows indeed correct results.

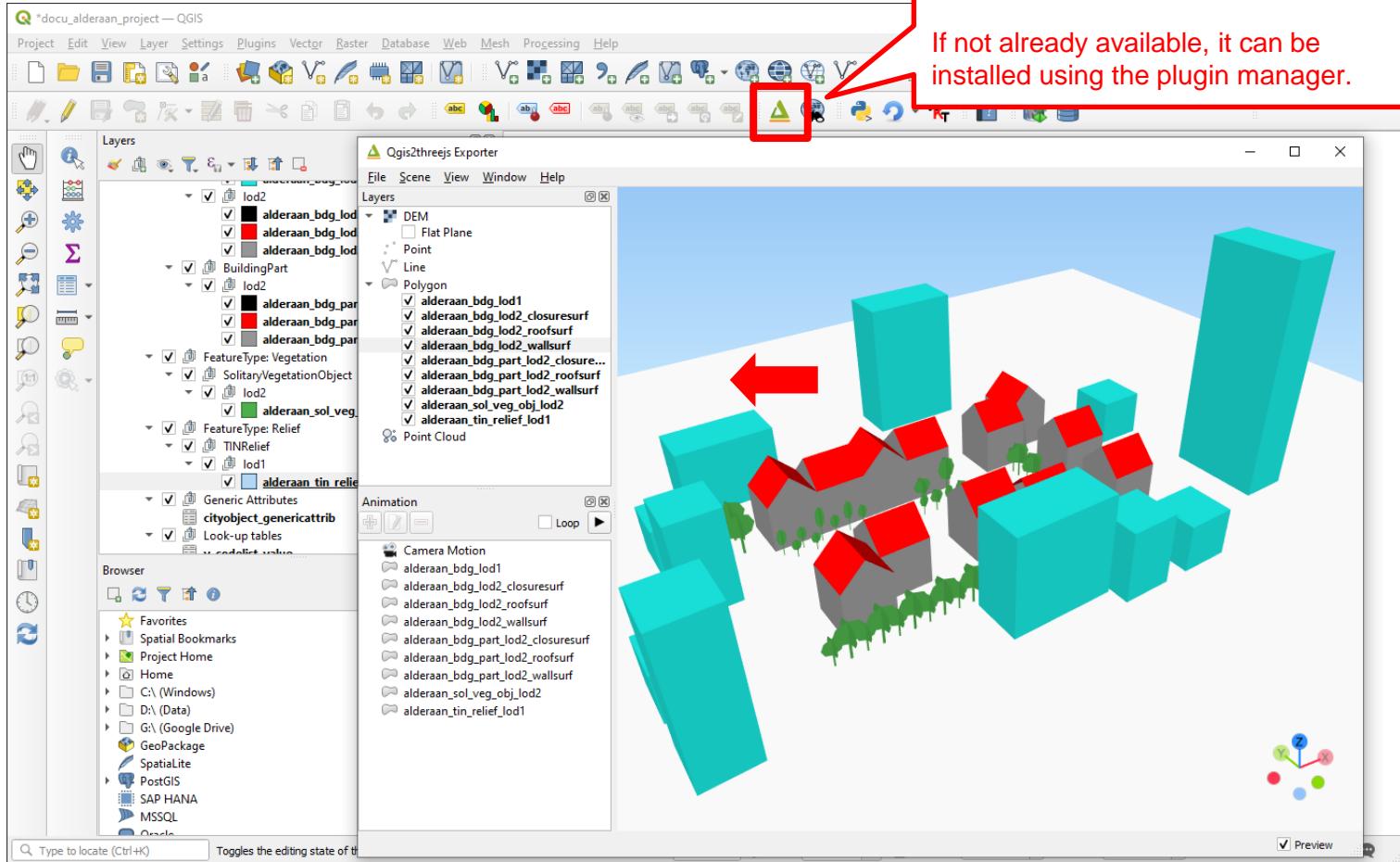


✓  
 Google earth  
 (as KML)

# Use in QGIS: 3D visualisation

Alternatively, the **Qgis2threejs** plugin can be used for 3D visualisation.

If not already available, it can be installed using the plugin manager.

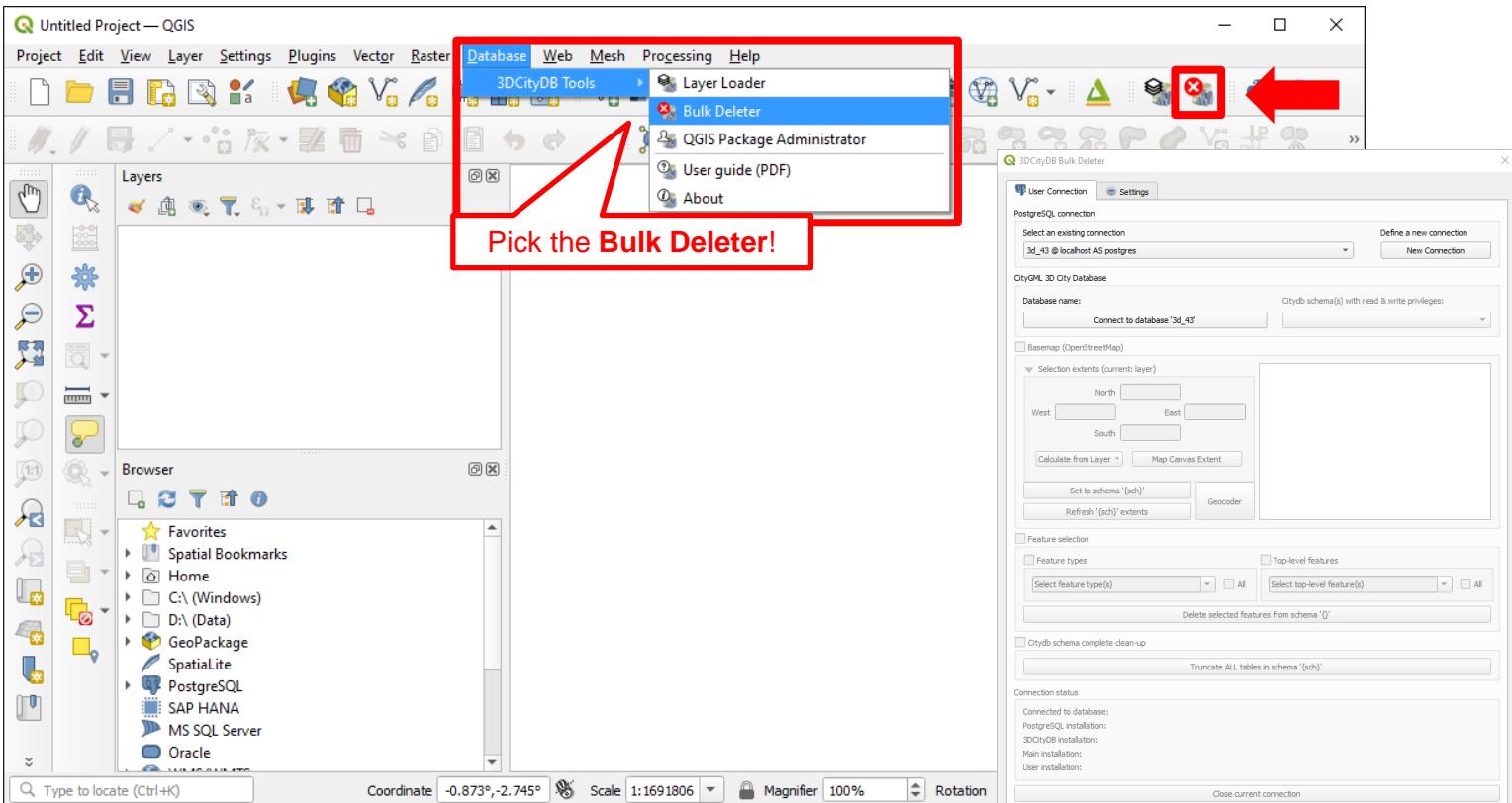


# Bulk Deleter



Open the **Bulk Deleter** from the menu or by clicking on the corresponding icon

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

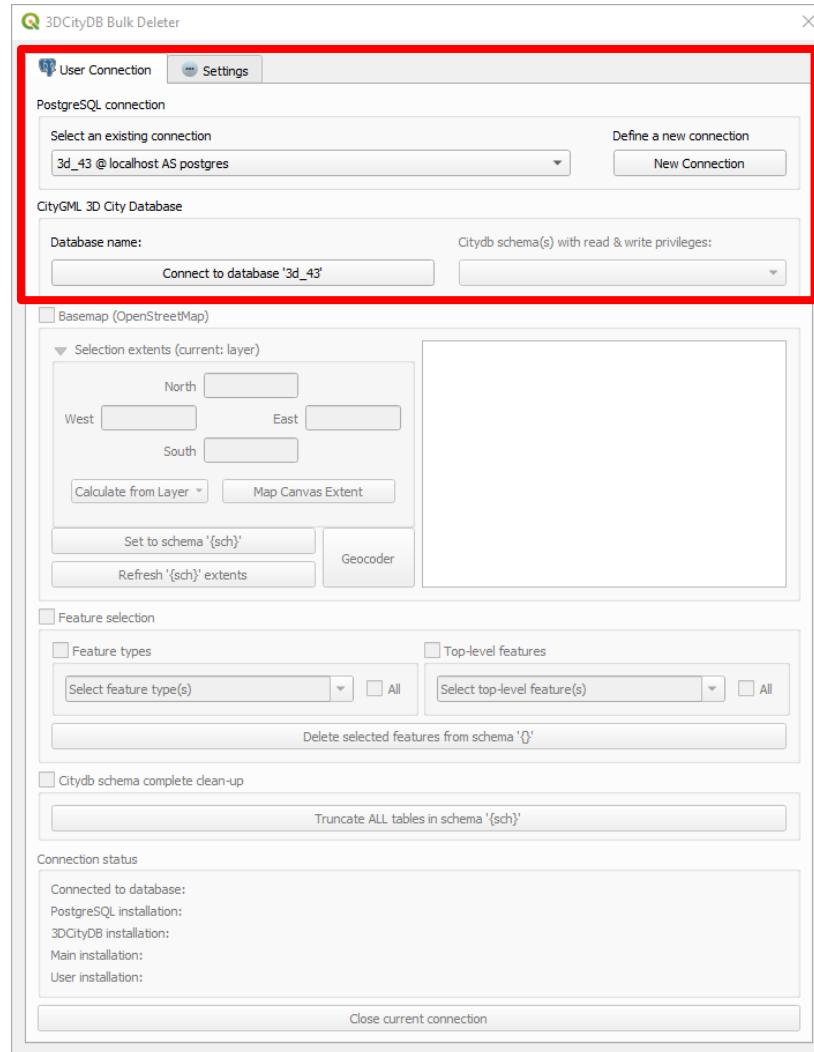


# Bulk Deleter

In the "User Connection" tab

1) Create a new connection or use an existing one to the desired 3DCityDB instance (here: "qgis\_test")

2) Connect to the chosen database

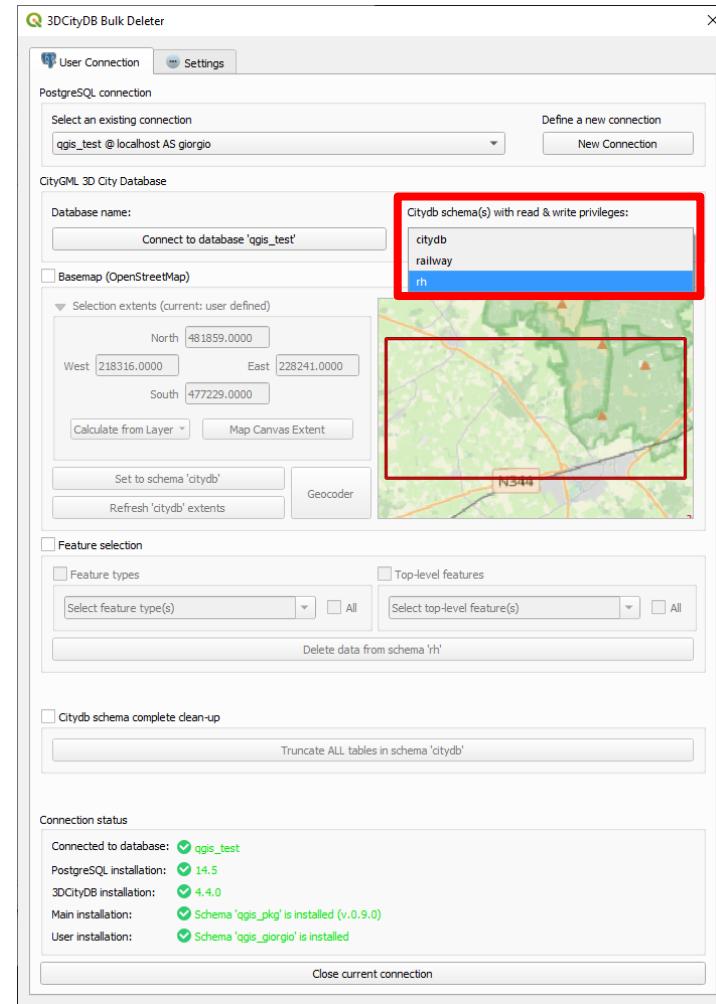


# Bulk Deleter

3) Once connected, choose one of the available citydb schemas

**Nota bene:** Only the citydb schemas for **which you have read & write ("rw") privileges** are listed

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

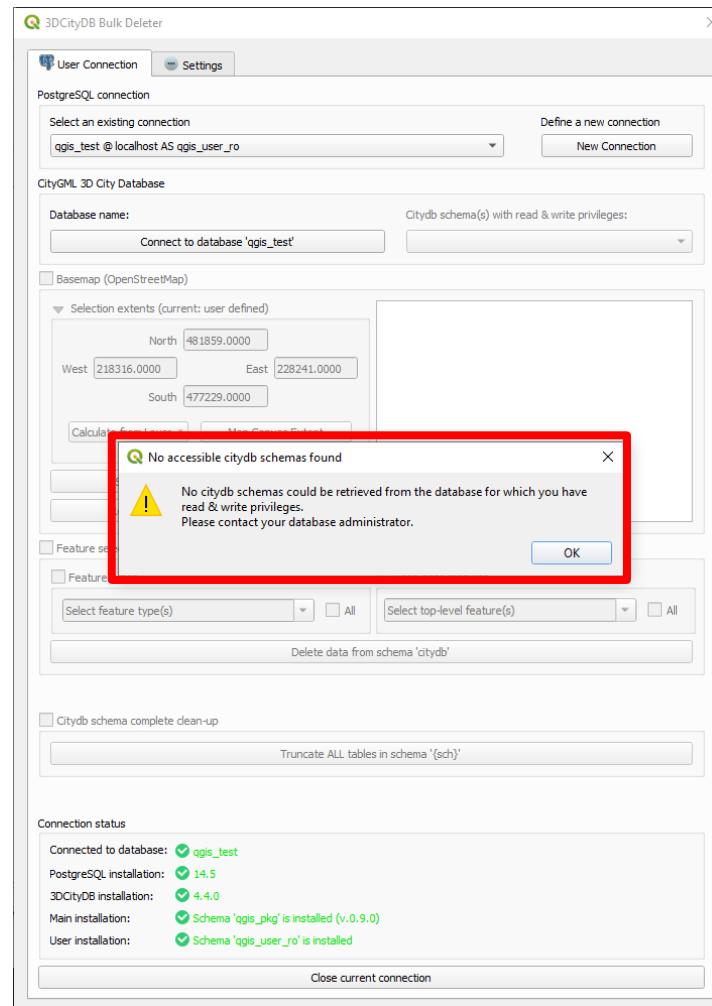


# Bulk Deleter

3) Once connected, choose one of the available citydb schemas

**Nota bene:** Only the citydb schemas for which you have read & write ("rw") privileges are listed

If there are no citydb schemas for which you have "rw" privileges, you will be notified before the connection is closed and the GUI completely disabled.

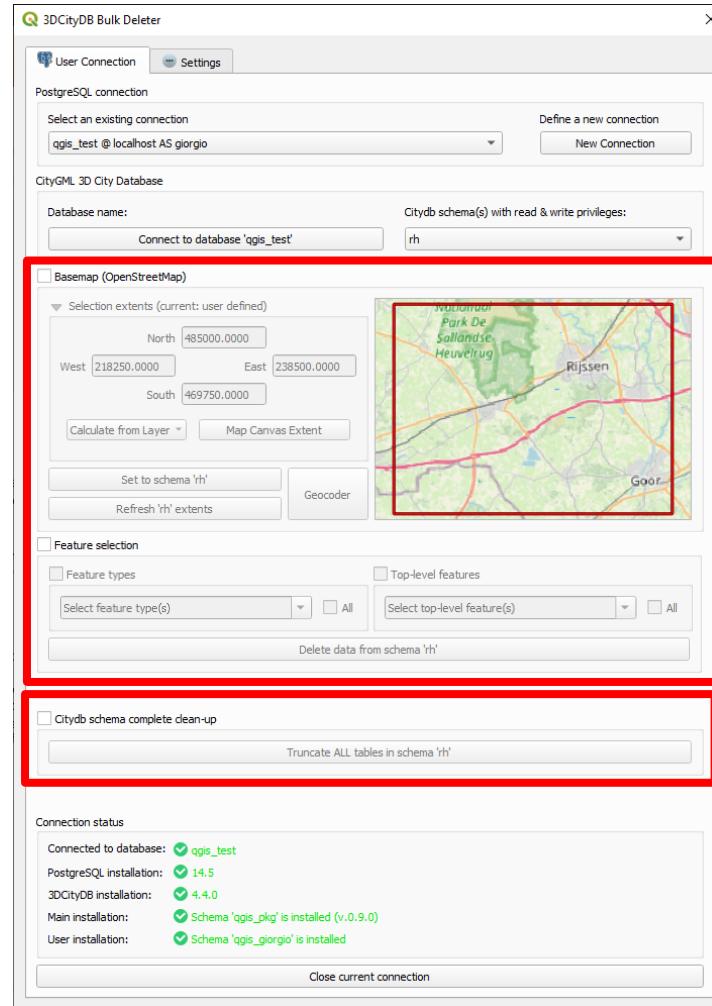


# Bulk Deleter

4) Once you have selected the citydb schema, you can perform 2 types of actions:

- Select specific features to delete (and optionally define a spatial filter, too)
- Clean up the whole schema, i.e. truncate all tables of the selected citydb schema

The GUI will prevent you from choosing both at the same time



# Bulk Deleter

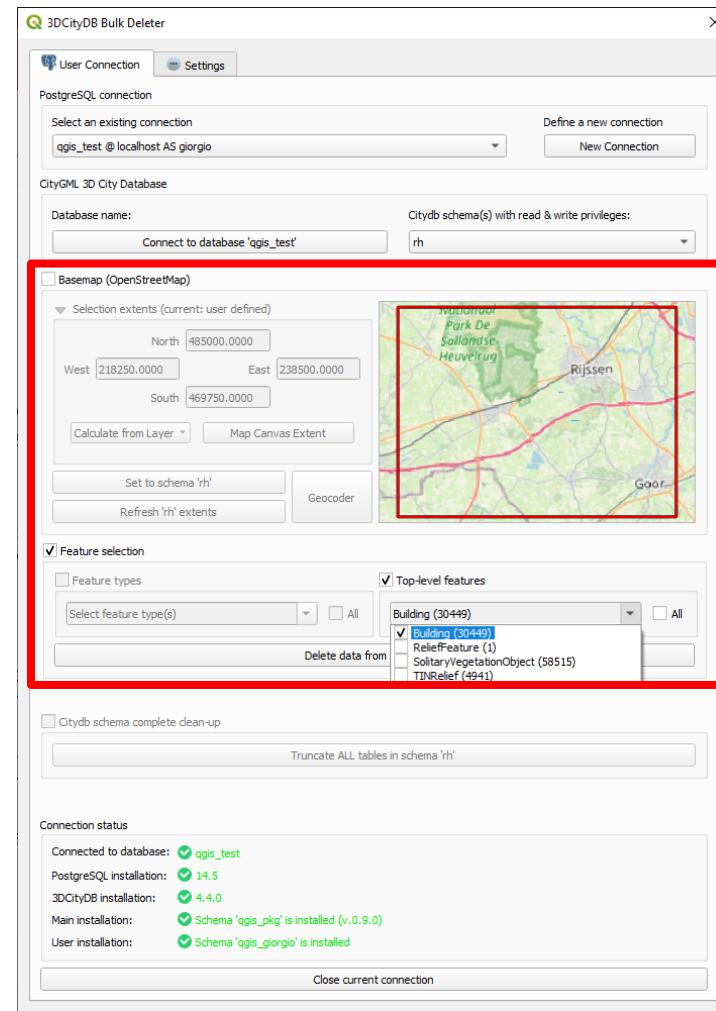
Motivation  
 Plugin overview  
 Software installation  
**Software use**  
 • Layer Loader  
 • Bulk Deleter  
 • About, etc.  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

5a) Select the features to delete. Activate the **Feature selection box**. You can now select:

- either CityGML Features types
- or top-level features

**Remember:** Feature Types correspond to CityGML modules, i.e. they may contain multiple top-level features. For example:

- Feature Type "Vegetation" includes "Solitary Vegetation Object" and "PlantCover" top-level features
- Feature Type "Relief" includes "TINRelief", "BreakLineRelief" and "MassPointRelief" top-level features



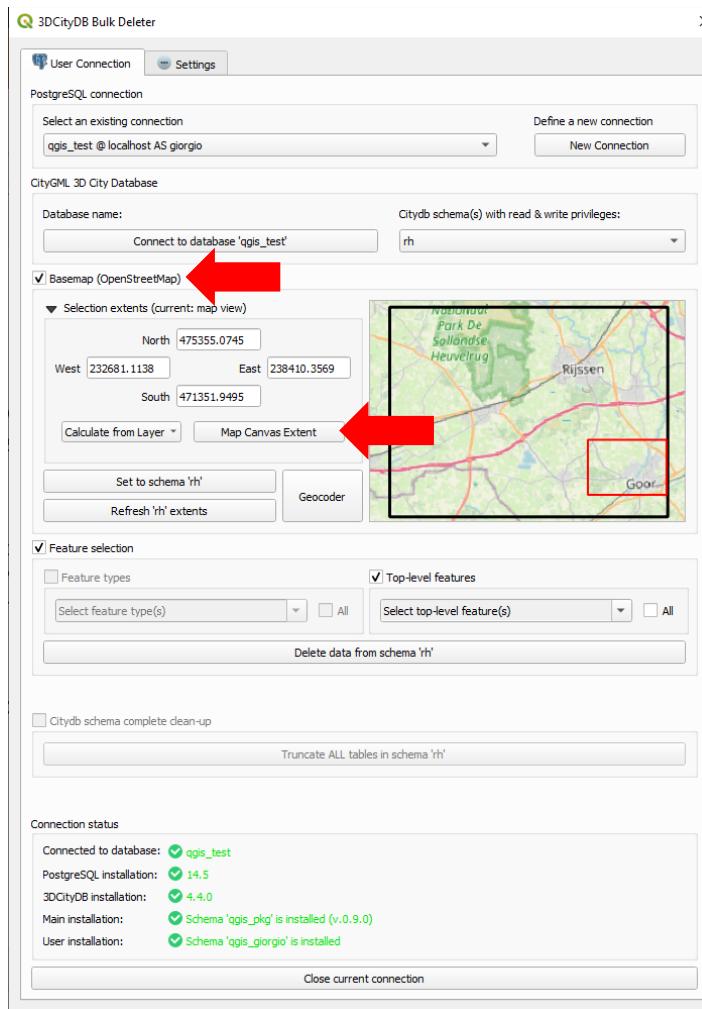
# Bulk Deleter

Motivation  
 Plugin overview  
 Software installation  
**Software use**  
 • Layer Loader  
 • Bulk Deleter  
 • About, etc.  
 Advanced options  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

- 5a) Select the features to delete. Activate the **Feature selection box**. You can now select:
- either CityGML Features types
  - or top-level features
  - and, optionally, define the extents of the area where to delete the selected feature. You must then also activate the **Basemap box** and press the **Map Canvas Extent button**

The delete extents are represented by the **red bounding box**.

**Please note:** The Set to schema {cdb\_schema}, Refresh {cdb\_schema} and GeoCoder buttons follow the same logic as in the Layer Loader



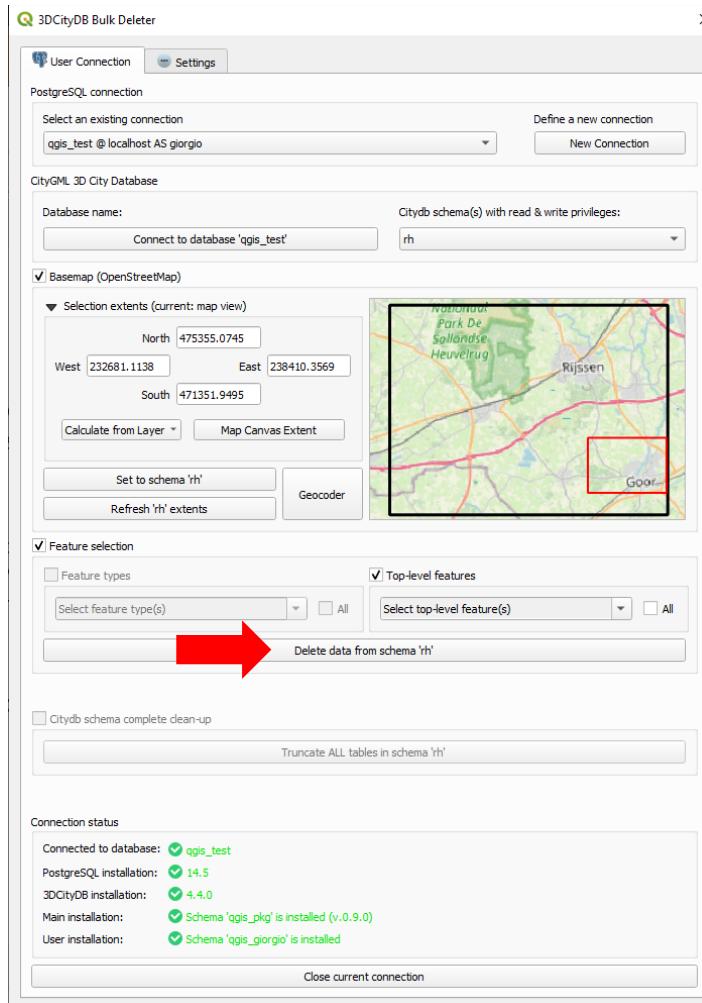
# Bulk Deleter

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - About, etc.
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources

6a) Press the **Delete data from schema {cdb\_schema}** button

**Beware:** Depending on the quantity of selected features, the operation might take some time.

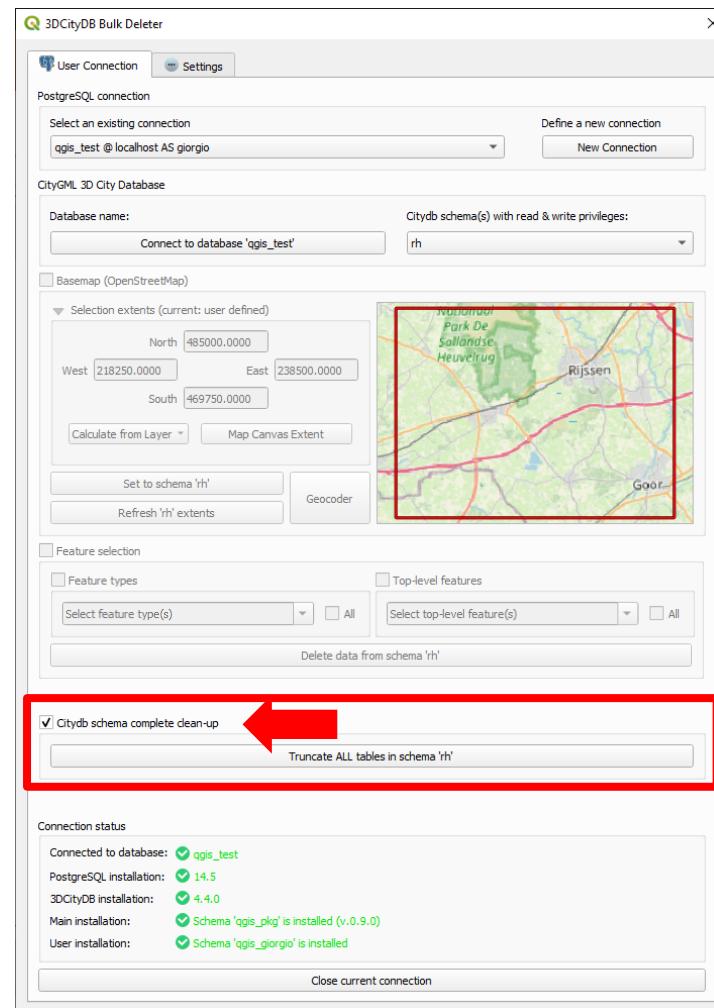
If you want to completely delete the database, you are recommended to use the other option (see next slides) which is **much faster!**



# Bulk Deleter

5b) Enable the **Citydb schema complete clean-up box** and press the **Truncate ALL tables in schema {cdb\_schema}** button.

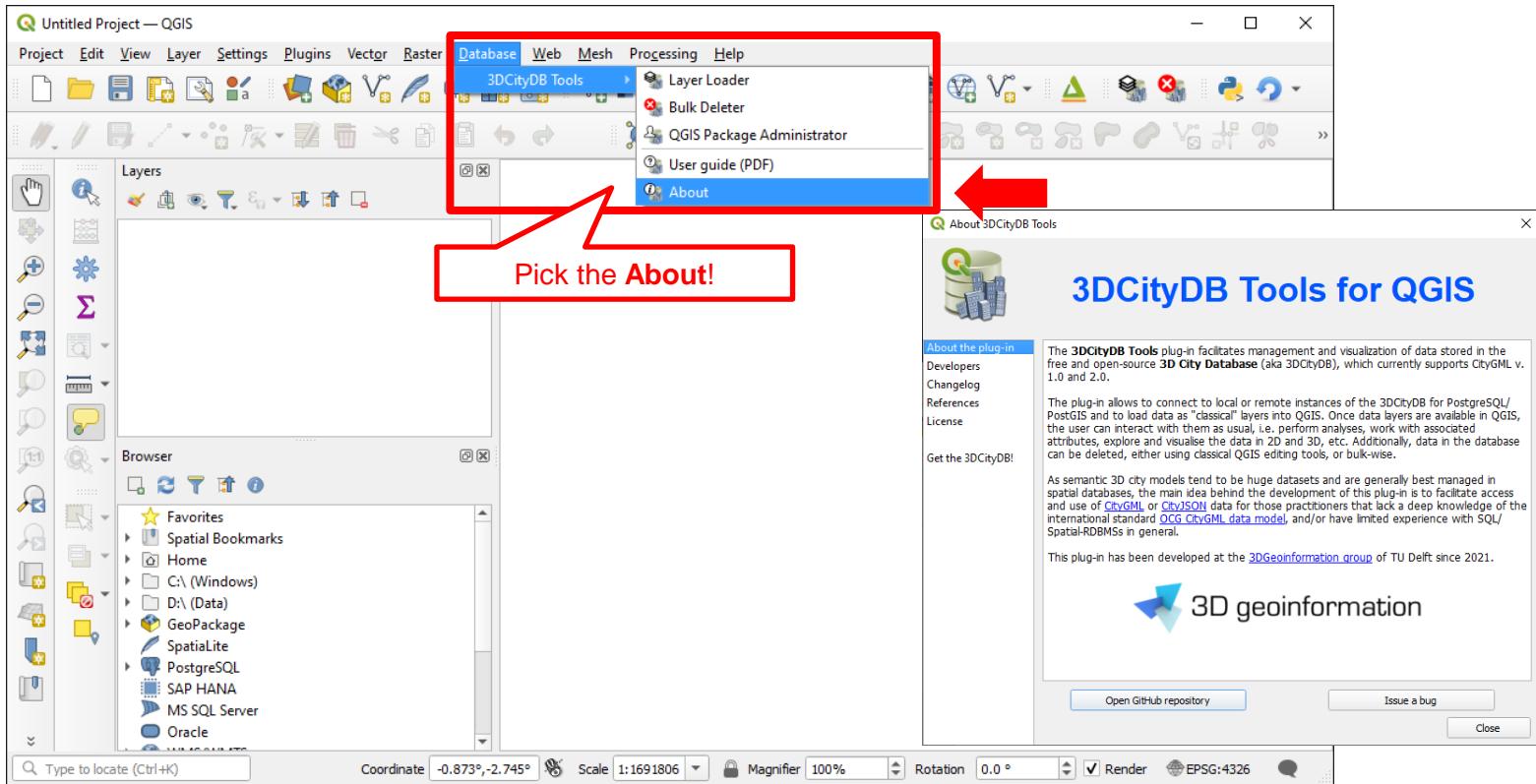
**Beware:** The selected citydb schema will be completely emptied and reset to its initial state. In addition, all preexisting privileges (also of other users) will be reset to "None"



# About

The **About** dialog provides additional information about the plug-in, the possibility to access the GitHub repository (and submit a bug), or to get the 3DCityDB Suite

- Motivation
- Plugin overview
- Software installation
- Software use**
  - Layer Loader
  - Bulk Deleter
  - **About, etc.**
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources



Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

Resources

# Customised codelists

For certain CityGML properties (e.g. class, function, usage, roof type, etc.) the QGIS attribute forms in the Layer Loader can be linked to look-up tables containing

- Codelist values from the non-normative CityGML specifications
- Codelist values optionally defined by the user

This applies to properties containing single (e.g. class, roof type) or, possibly, multiple values (e.g. function, usage)

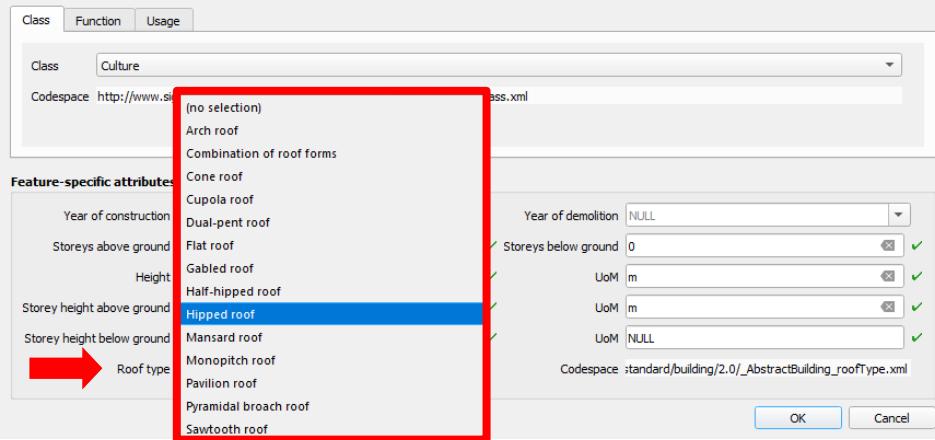
In this way the user does not have to "remember" specific codes when typing, thus reducing the chances of wrong data input

The 3DCityDB-Tools plugin already contains the codelists from the CityGML 2.0 specifications.

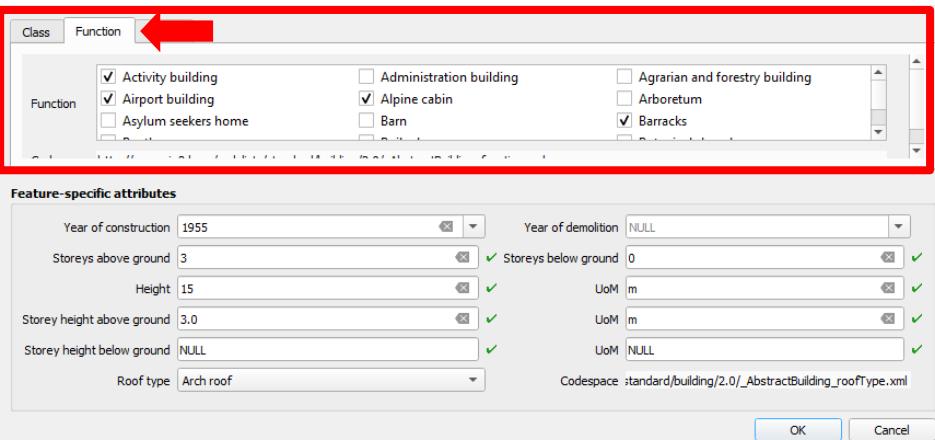
# Customised codelists

Motivation  
 Plugin overview  
 Software installation  
 Software use  
**Advanced options**  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

Property with cardinality [0..1]:  
 drop-down list



Property with cardinality [0..\*]:  
 multiple-selection list



# Customised codelists

Adding customised codelists requires **two steps**:

- 1) Load the actual codelist data** (codelist entries and codelist metadata). Such data must be stored in two predefined tables of the QGIS Package in the 3DCityDB
- 2) Add mapping rules** for automatic setup of the lists in the attribute forms. Such data must be stored in a predefined table of the QGIS Package in the 3DCityDB

Both operations can be carried out either by the database administrator or by the user

1) Administrator:

- The predefined tables are in schema **qgis\_pkg**
- All codelists and rules are made available to each newly created `qgis_{usr}` schema

2) User:

- The predefined tables are in schema **qgis\_{usr\_name}**
- All codelists and rules are available only to user

# Customised codelists

Adding customised codelists requires **two steps**:

**1) Load the actual codelist data** (codelist entries and codelist metadata). Such data must be stored in two predefined tables of the QGIS Package in the 3DCityDB

- The tables are named **CODELIST** and **CODELIST\_VALUE** in the `qgis_{usr_name}` and **CODELIST\_TEMPLATE** and **CODELIST\_VALUE\_TEMPLATE** in the `qgis_pkg` schema, respectively. Their structure is exactly the same
- The following examples are based on tables CODELIST and CODELIST\_VALUE but the procedure is the same for the \*\_TEMPLATE ones

# Customised codelists

Motivation  
 Plugin overview  
 Software installation  
 Software use  
**Advanced options**  
 Software uninstall  
 Current limitations  
 QGIS Package  
 Resources

Table **CODELIST** contains some metadata values such as the codelist name, its name\_space and data\_model.

It is referenced by table **CODELIST\_VALUE** which contains the actual values

Tables (8)
> codelist
> codelist_lookup_config
> <b>codelist_value</b>
> enumeration
> enumeration_value
> extents
> layer_metadata
> settings

**Table CODELIST (excerpt)**

<b>[PK]</b> id bigint	<b>data_model</b> character varying	<b>name</b> character varying	<b>name_space</b> character varying	<b>description</b> text
1	CityGML 2.0	MimeType	https://www.sig3d.org/codelists/standard/core/2.0/_ImplicitGeometry_mimeType.xml	[null]
2	CityGML 2.0	_AbstractBridgeClass	https://www.sig3d.org/codelists/standard/bridge/2.0/_AbstractBridge_class.xml	
3	CityGML 2.0	_AbstractBridgeFunctionUsage	https://www.sig3d.org/codelists/standard/bridge/2.0/_AbstractBridge_function.xml	
4	CityGML 2.0	_AbstractBuildingClass	https://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_class.xml	
5	CityGML 2.0	_AbstractBuildingFunctionUsage	https://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuilding_function.xml	
6	CityGML 2.0	_AbstractBuildingRoofType	https://www.sig3d.org/codelists/standard/building/2.0/_AbstractBuildingRoof_type.xml	
7	CityGML 2.0	RoomClass	https://www.sig3d.org/codelists/standard/building/2.0/_Room_class.xml	
8	CityGML 2.0	RoomFunctionUsage	https://www.sig3d.org/codelists/standard/building/2.0/_RoomFunction_usage.xml	
9	CityGML 2.0	BuildingFurnitureClass	https://www.sig3d.org/codelists/standard/building/2.0/_BuildingFurniture_class.xml	
10	CityGML 2.0	BuildingFurnitureFunctionUsage	https://www.sig3d.org/codelists/standard/building/2.0/_BuildingFurnitureFunction_usage.xml	

**Table CODELIST\_VALUE (excerpt)**

<b>[PK]</b> id bigint	<b>code_id</b> integer	<b>value</b> character varying	<b>description</b> text
1	1	model/vrml	VRML97
2	1	application/x-3ds	3ds max
3	1	application/dxf	AutoCad DXF
4	1	application/x-autocad	AutoCad DXF
5	1	application/x-dxf	AutoCad DXF
6	1	application/acad	AutoCad DWG
13	1	image/tiff	*.tiff, *.tif images
14	1	image/bmp	*.bmp images
15	2	1000	Arced bridge
16	2	1010	Cable-stayed bridge
17	2	1020	Deck bridge
18	2	1030	Cable-stayed overpass

# Customised codelists



In QGIS, all codelists values are retrieved from view **v\_codelist\_value** in the user schema of every user (e.g. "qgis\_giorgio")

**View V\_CODELISTS\_VALUE (excerpt)**

<b>id</b> bigint	<b>data_model</b> character varying	<b>name</b> character varying	<b>value</b> character varying	<b>description</b> text	<b>name_space</b> character varying
1	CityGML 2.0	MimeType	model/vrml	VRML97	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
2	CityGML 2.0	MimeType	application/x-3ds	3ds max	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
3	CityGML 2.0	MimeType	application/dxf	AutoCad DXF	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
4	CityGML 2.0	MimeType	application/x-autocad	AutoCad DXF	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
39	CityGML 2.0	_AbstractBuildingClass	1110	Maintainence and waste mana...	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
40	CityGML 2.0	_AbstractBuildingClass	1120	Healthcare	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
41	CityGML 2.0	_AbstractBuildingClass	1130	Communicating	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
42	CityGML 2.0	_AbstractBuildingClass	1140	Security	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
43	CityGML 2.0	_AbstractBuildingClass	1150	Storage	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
44	CityGML 2.0	_AbstractBuildingClass	1160	Industry	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
45	CityGML 2.0	_AbstractBuildingClass	1170	Traffic	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
46	CityGML 2.0	_AbstractBuildingClass	1180	Other function	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
47	CityGML 2.0	_AbstractBuildingClass	9999	Unknown	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
48	CityGML 2.0	_AbstractBuildingFunct...	1000	Residential building	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>
49	CityGML 2.0	_AbstractBuildingFunct...	1010	Tenement	<a href="https://www.sig3d.org/codelists/stan...">https://www.sig3d.org/codelists/stan...</a>

# Customised codelists

To add values to the **CODELIST** and **CODELIST\_VALUE** tables, the user can for example issue a SQL statement such as:

```
-- Optionally, delete previously loaded values belonging to the same codelist in user schema "qgis_giorgio"  
DELETE FROM qgis_giorgio.codelist WHERE data_model = 'NL-BAG';  
-- Insert first the codelist metadata and then the values in one single SQL statement  
WITH cl AS (  
    INSERT INTO qgis_giorgio.codelist (data_model, name, name_space, description)  
    VALUES  
    ('NL-BAG', 'BAG', 'https://..some_url_here.....', 'Codelist containing the values of the Dutch Basisregistratie  
    Adressen en Gebouwen')  
    RETURNING id)  
INSERT INTO qgis_giorgio.codelist_value (code_id, value, description)  
SELECT cl.id, v.value, v.description FROM cl, (VALUES  
    ('apple' , 'Codelist value for "apple"' ),  
    ('orange' , 'Codelist value for "orange"' ),  
    ('pear' , 'Codelist value for "pear"' ),  
    ('banana' , 'Codelist value for "banana"' )  
) AS v(value, description);
```

This SQL statement can be adapted by changing only the parts in red

# Customised codelists

Adding customised codelists requires **two steps**:

- 1) Load the actual codelist data (codelist entries and codelist metadata). Such data must be stored in two predefined tables of the QGIS Package in the 3DCityDB
- 2) **Add mapping rules** for automatic setup of the lists in the attribute forms. Such data must be stored in a predefined table of the QGIS Package in the 3DCityDB
  - The table is named **CODELIST\_LOOKUP\_CONFIG** in the qgis\_{usr\_name} and **CODELIST\_LOOKUP\_CONFIG\_TEMPLATE** in the qgis\_pkg schema, respectively. Their structure is exactly the same
  - The following examples are based on tables CODELIST\_LOOKUP\_CONFIG but the procedure is the same for the \*\_TEMPLATE one

# Customised codelists

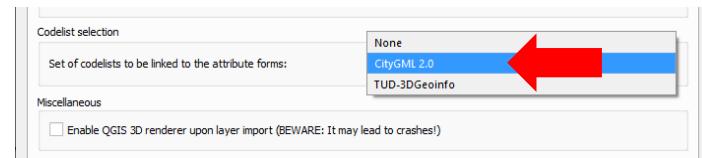
Tables (8)	
>	codelist
>	<b>codelist_lookup_config</b>
>	codelist_value
>	enumeration
>	enumeration_value
>	extents
>	layer_metadata
>	settings

Table **CODELIST\_LOOKUP\_CONFIG** contains fields used to map Codelist values to the corresponding form attributes in QGIS.  
Explanation of the necessary fields is given in the next slide.

<b>id [PK]</b>	<b>name</b> <b>character varying</b>	<b>ade_prefix</b> <b>character var</b>	<b>source_class</b> <b>character varying</b>	<b>source_table</b> <b>character varying</b>	<b>source_column</b> <b>character varying</b>	<b>target_table</b> <b>character varying</b>	<b>key_column</b> <b>character var</b>	<b>value_column</b> <b>character varyir</b>	<b>filter_expression</b> <b>character varying</b>
1	CityGML 2.0	[null]	Bridge	bridge	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeClass'
2	CityGML 2.0	[null]	Bridge	bridge	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'
3	CityGML 2.0	[null]	Bridge	bridge	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'
4	CityGML 2.0	[null]	BridgePart	bridge	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeClass'
5	CityGML 2.0	[null]	BridgePart	bridge	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'
6	CityGML 2.0	[null]	BridgePart	bridge	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'
7	CityGML 2.0	[null]	Building	building	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingClass'
8	CityGML 2.0	[null]	Building	building	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingFunctionUsage'
9	CityGML 2.0	[null]	Building	building	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingFunctionUsage'
10	CityGML 2.0	[null]	Building	building	roof_type	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingRoofType'
11	CityGML 2.0	[null]	BuildingPart	building	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingClass'
12	CityGML 2.0	[null]	BuildingPart	building	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingFunctionUsage'
13	CityGML 2.0	[null]	BuildingPart	building	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingFunctionUsage'
14	CityGML 2.0	[null]	BuildingPart	building	roof_type	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBuildingRoofType'
15	CityGML 2.0	[null]	BuildingRoom	room	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = 'RoomClass'
16	CityGML 2.0	[null]	BuildingRoom	room	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = 'RoomFunctionUsage'
17	CityGML 2.0	[null]	BuildingRoom	room	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = 'RoomFunctionUsage'
18	CityGML 2.0	[null]	BuildingFurnit...	building_furnit...	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = 'BuildingFurnitureClass'
19	CityGML 2.0	[null]	BuildingFurnit...	building_furnit...	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = 'BuildingFurnitureFunctionUsage'
20	CityGML 2.0	[null]	BuildingFurnit...	building_furnit...	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = 'BuildingFurnitureFunctionUsage'

# Customised codelists

Those in yellow are the fields the must be added by the user. The other ones will be added automatically



**NAME:** The name indicating the set of mapping rules. It must be the same for all rules belonging to the same group. This label will be the selectable one in the QGIS GUI.

**SOURCE\_CLASS:** The CityGML/ADE\* class the CodeList will be associated to (\*ADEs are currently not supported)

**SOURCE\_TABLE:** The corresponding citydb table which contains the column to be associated to a codelist

**SOURCE\_COLUMN:** The column to be associated to a codelist

**ALLOW\_MULTI:** FALSE if the cardinality is 0..1, TRUE if it is 0..\*

**NUM\_COLUMNS:** Number of column presented in the widget and containing look-up values. Default: 1 when ALLOW\_MULTI is FALSE, 3 when ALLOW\_MULTI is TRUE.

**FILTER\_EXPRESSION:** String containing the expression to filter the values of the desired codelist in the GUI form. It refers to the values of the columns DATA\_MODEL and NAME of view V\_CODELIST (see previous slides).

integer	name character varying	ade_prefix character var	source_class character varying	source_table character varying	source_colur character var	target_table character var	key_column character var	value_column character var	filter_expression character varying	num_columns integer	allow_multi boolean	allow_null boolean	order_by_value boolean	use_comp boolean
1	CityGML 2.0	[null]	Bridge	bridge	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeClass'	1	false	true	true	false
2	CityGML 2.0	[null]	Bridge	bridge	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'	3	true	true	true	false
3	CityGML 2.0	[null]	Bridge	bridge	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'	3	true	true	true	false
4	CityGML 2.0	[null]	BridgePart	bridge	class	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeClass'	1	false	true	true	false
5	CityGML 2.0	[null]	BridgePart	bridge	function	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'	3	true	true	true	false
6	CityGML 2.0	[null]	BridgePart	bridge	usage	v_codelist	value	description	data_model = 'CityGML 2.0' AND name = '_AbstractBridgeFunctionUsage'	3	true	true	true	false

# Codelists and look-up tables

To add values to the **CODELIST\_LOOKUP\_CONFIG** table, the user can for example issue a SQL statement such as:

```
-- Optionally, delete previously loaded values belonging to the same codelist group in user schema "qgis_giorgio"  
DELETE FROM qgis_giorgio.codelist_lookup_config WHERE name = 'StarWarsCodelist';  
-- Insert the mapping rules in one single SQL statement  
INSERT INTO qgis_giorgio.codelist_lookup_config  
(name, source_class, source_table, source_column, allow_multi, num_columns, filter_expression)  
VALUES  
('StarWarsCodelist', 'Building' , 'building', 'class' , FALSE, 1, 'data_model = "StarWarsCoruscant" AND name =  
"CoruscantBdgClass"),  
('StarWarsCodelist', 'Building' , 'building', 'function' , TRUE , 3, 'data_model = "StarWarsCoruscant" AND name =  
"CoruscantBdgFunction"),  
('StarWarsCodelist', 'Building' , 'building', 'usage' , TRUE , 3, 'data_model = "StarWarsCoruscant" AND name =  
"CoruscantBdgUsage"),  
('StarWarsCodelist', 'Building' , 'building', 'roof_type' , FALSE, 1, 'data_model = "StarWarsCoruscant" AND name =  
"CoruscantBdgRoofType");
```

This SQL statement can be adapted by changing only the parts in red. For the qgis\_pkg, use table CODELIST\_LOOKUP\_CONFIG\_TEMPLATE instead.

Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

Resources

# Software uninstall

Uninstallation consists of two parts:

## 1) Partial/complete **removal of the QGIS Package** from PostgreSQL

- This operation can be carried out only by the database administrator
- The administrator can choose to drop only the schema of a selected user (e.g. "qgis\_giorgio")
- The administrator can remove all user schemas AND the qgis\_pkg schema

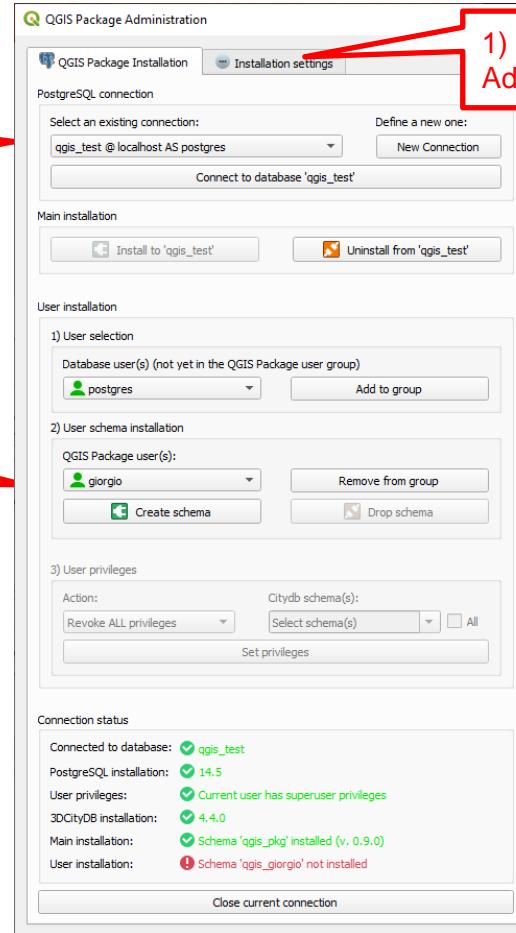
## 2) **Removal of the 3DCityDB-Tools plugin** from QGIS

- This operation can be carried out by any user
- The plugin can be uninstalled from the \Plugins\Manage and Install Plugins... Menu in QGIS
- Alternatively, it can be uninstalled manually by simply removing the plugin folder

# Drop user schema

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
**Software uninstall**  
Current limitations  
QGIS Package  
Resources

2) Connect as administrator



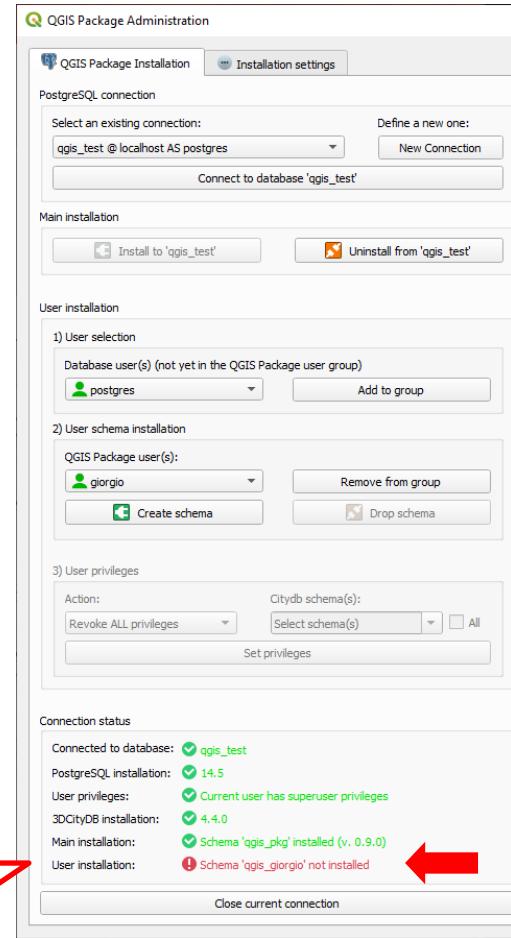
1) Load the "QGIS Package Administration" GUI

3) Select the user

4) Drop the schema of the selected user.  
All privileges will be automatically reset to "None"

# Drop user schema

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
**Software uninstall**  
Current limitations  
QGIS Package  
Resources

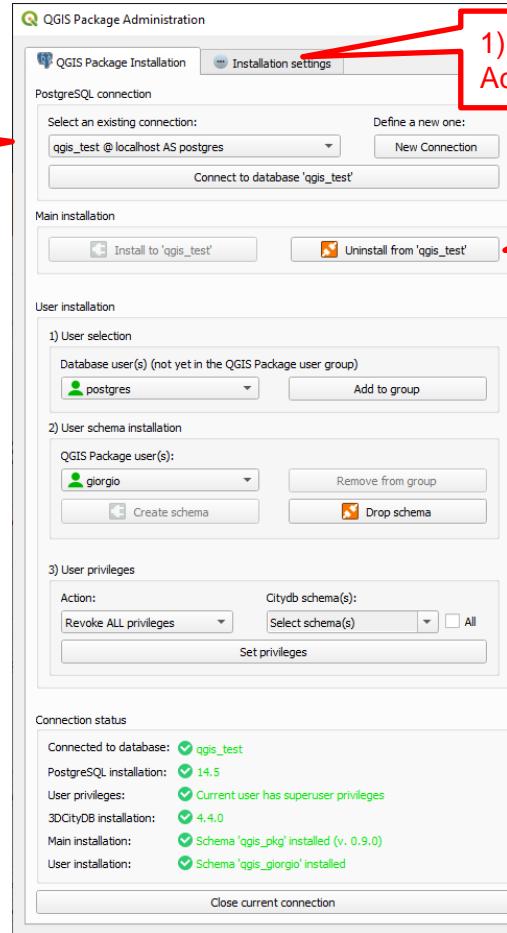


5) The user installation field  
is now red again

# Software uninstall: Uninstall QGIS Package

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
**Software uninstall**  
Current limitations  
QGIS Package  
Resources

2) Connect as administrator

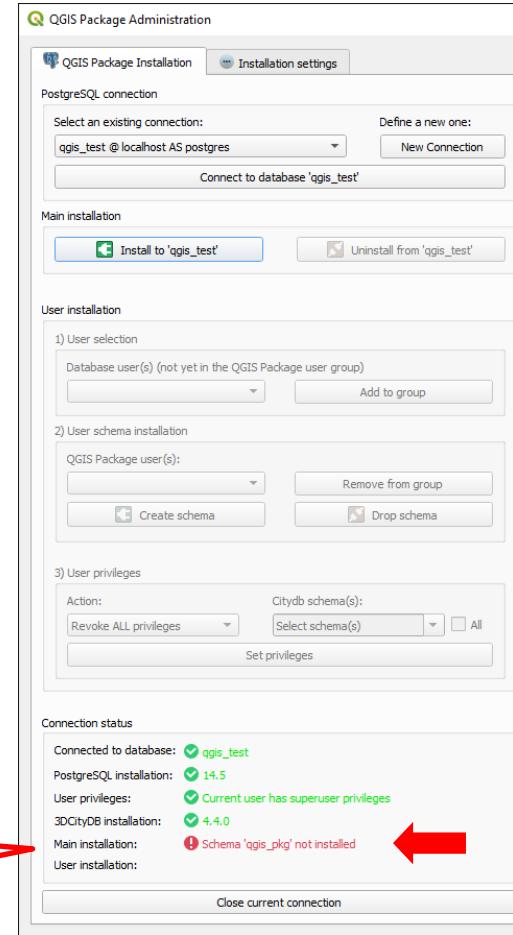


1) Load the "QGIS Package Administration" GUI

3) Uninstall the QGIS Package from the current database

# Uninstall QGIS Package

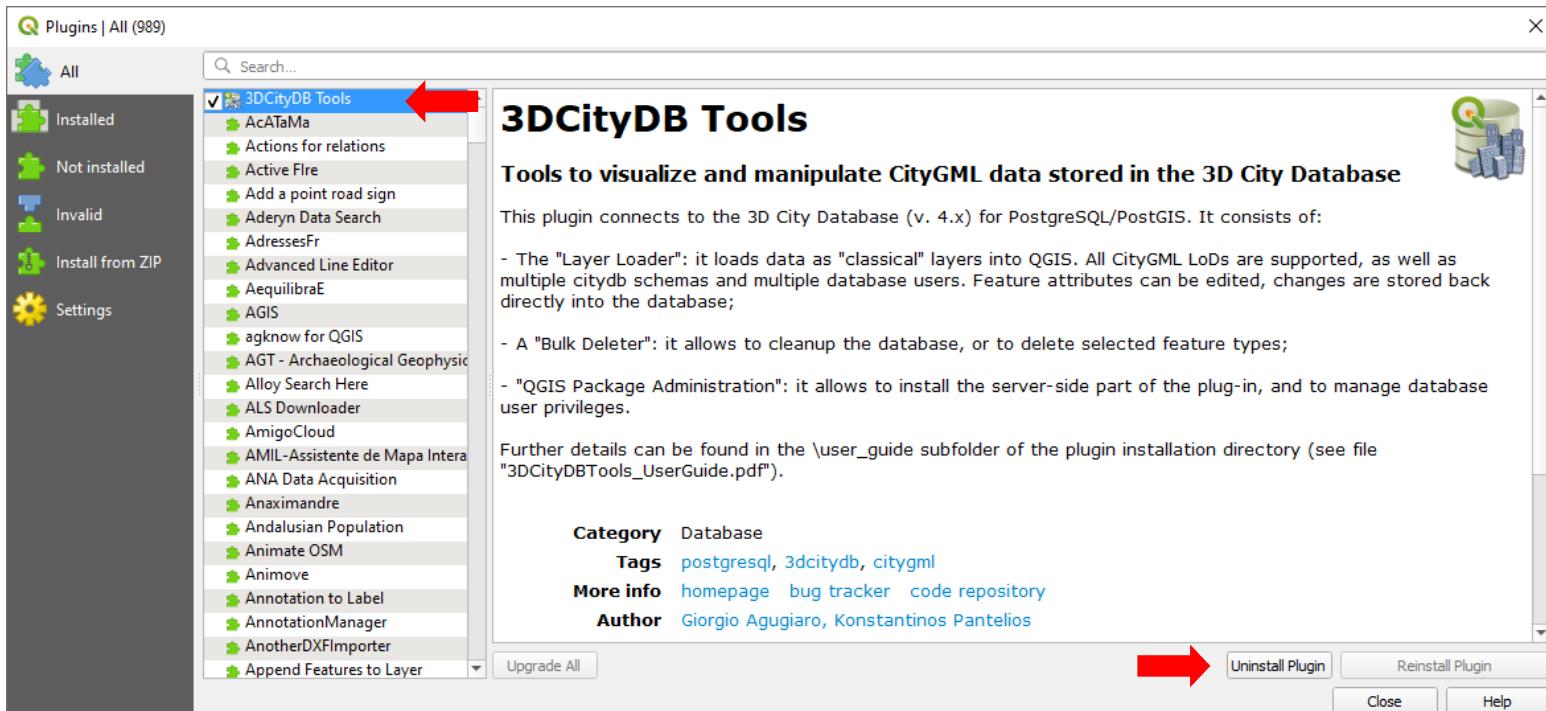
Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
**Software uninstall**  
Current limitations  
QGIS Package  
Resources



# Uninstall 3DCityDB-Tools

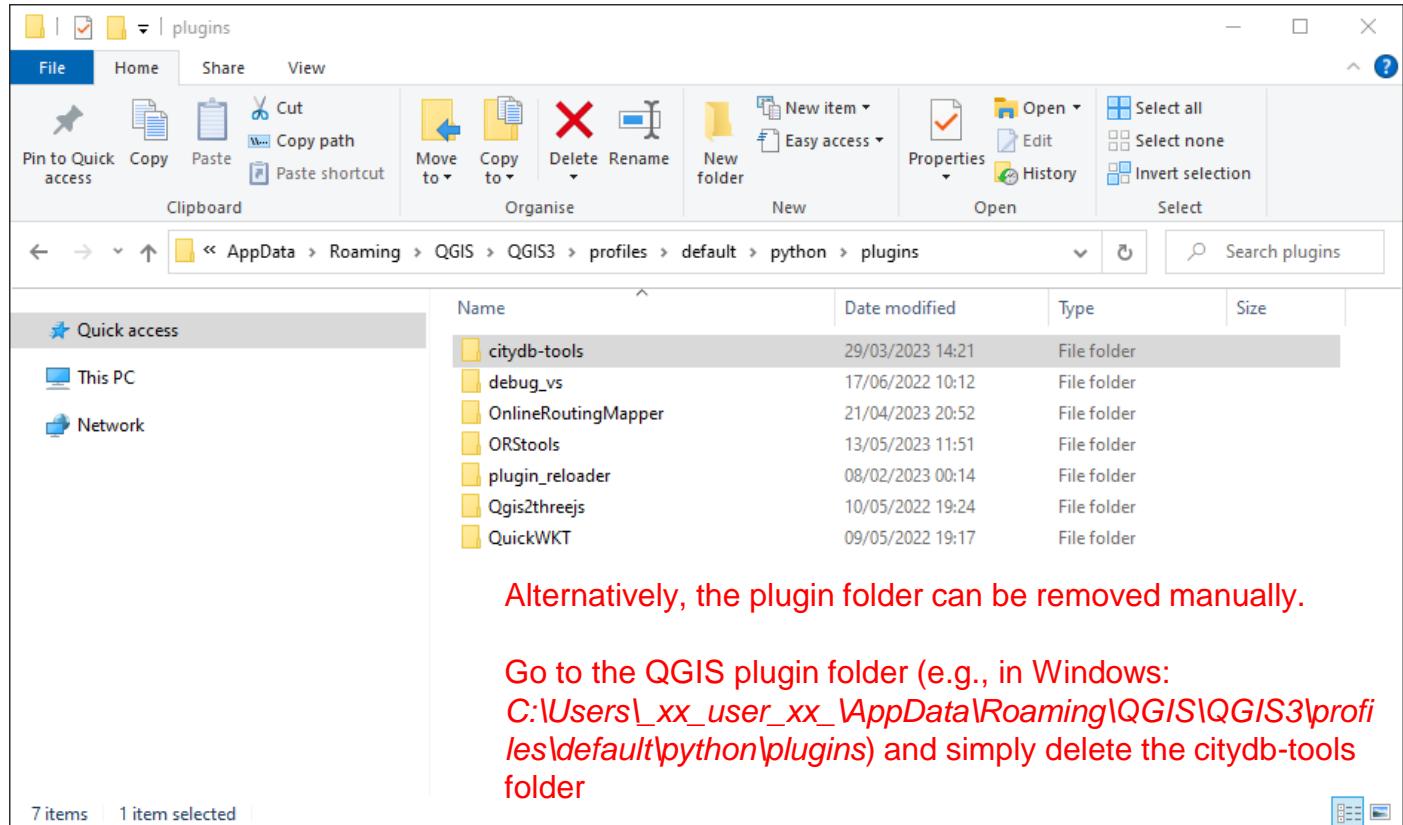
The plugin can be uninstalled from the \Plugins\Manage and Install Plugins... Menu in QGIS

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
**Software uninstall**  
Current limitations  
QGIS Package  
Resources



Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
**Software uninstall**  
Current limitations  
QGIS Package  
Resources

# Uninstall 3DCityDB-Tools



Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

Resources

# Current limitations

In general:

- CityGML appearances are not supported
- The Layer Loader does not support CityObjectGroups
- CityGML ADEs (Application Domain Extensions) are not supported

The QGIS Package does not support:

- Raster-based Relief features
- Generation of layers for CityObjectGroups

Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

QGIS  
Package

Resources

# QGIS Package in a nutshell

- It represents the **server-side part** of the QGIS plugin
  - Most of the actions carried out from the QGIS GUI can be actually performed also by interacting directly with the database (e.g. using **PgAdmin**)
- It can be installed and used independently from the QGIS client-side part
  - E.g. with FME, or programmatically via Python, SQLAlchemy, etc.
- It requires
  - PostgreSQL **v. 10 or higher**
  - An existing installation of the 3DCityDB **v. 4.x**
- All relevant entities are installed in the "**qgis\_pkg**" database schema
  - Database types
  - Triggers and trigger functions
  - Functions
  - Tables, mostly used as templates for the user schemas

# QGIS Package in a nutshell

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
**QGIS Package**  
Resources

The database administrator can:

- Create user schemas
- Grant/revoke privileges per user and per citydb schema
- Create, refresh, drop layers
- Drop user schemas

The required SQL functions are all available in schema **qgis\_pkg**.

# QGIS Package: Create user schemas

The database administrator can create user schemas for specific database users.

For example:

- User "giorgio" -> schema "qgis\_giorgio"
- User "konstantinos" -> schema "qgis\_konstantinos"
- User "postgres" -> schema "qgis\_postgres"

Each user schema will be used only by the corresponding user

Each user schema is accessible only by the corresponding user (and the database superusers)

In a user schema all necessary tables, updatable views, materialized views etc. will be created

## SQL example

```
-- Create the schema for user "giorgio". It will create schema "qgis_giorgio" in the current database
SELECT qgis_pkg.create_qgis_usr_schema('giorgio');
```

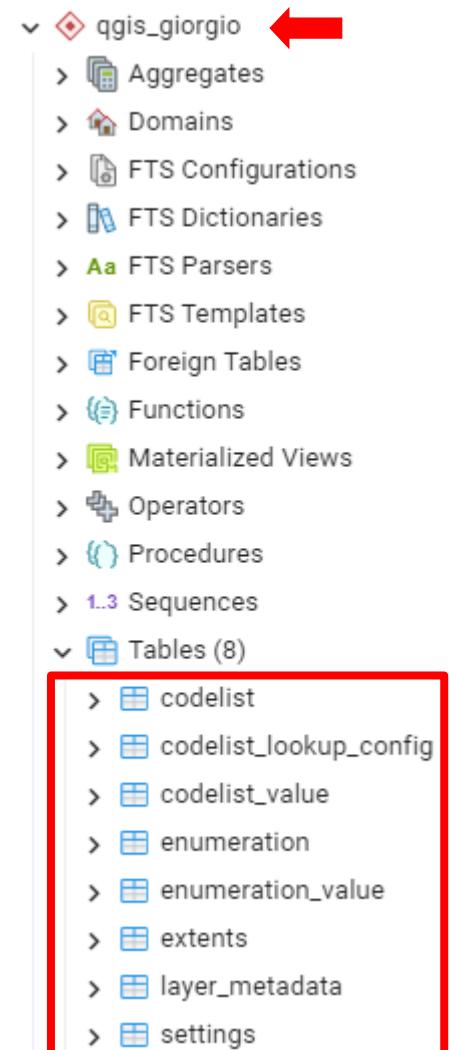
# QGIS Package: Create user schemas

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
**QGIS Package**  
Resources

## User schema overview

In each user schema (e.g. "qgis\_giorgio") some tables are generated upon schema creation:

- Tables **CODELIST**, **CODELIST\_LOOKUP\_CONFIG** and **CODELIST\_VALUE** are used to store codelists and related settings. See slides on **advanced options** for more details
- Tables **ENUMERATION** and **ENUMERATION\_VALUE** are used to store enumerations
- Table **EXTENTS** contains the bounding boxes of the citydb schemas and those of the materialized views
- Table **LAYER\_METADATA** contains information about generated and refreshed layers
- Table **SETTINGS** is used to store the user's settings (from the QGIS GUI)



qgis_giorgio
Aggregates
Domains
FTS Configurations
FTS Dictionaries
FTS Parsers
FTS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (8)
codelist
codelist_lookup_config
codelist_value
enumeration
enumeration_value
extents
layer_metadata
settings

# QGIS Package: User management

The database administrator can grant read-only (ro) or read-write (rw) privileges to the users.

- Privileges can be granted to a specific citydb schema. Please note the **cdb\_schema** parameter, requiring a single varchar value with the specific schema name

## SQL examples

```
-- Database user "giorgio" is added to group "qgis_pkg_usrgroup_qgis_test", can access data in citydb schema "citydb" of database "qgis_test" with read-only privileges
```

```
SELECT qgis_pkg.grant_qgis_usr_privileges(usr_name := 'giorgio', priv_type := 'ro', cdb_schema := 'citydb');
```

```
-- Database user "konstantinos" is added to group "qgis_pkg_usrgroup_qgis_db", can access data in citydb schema "citydb_2" of database "qgis_db" with read-write privileges
```

```
SELECT qgis_pkg.grant_qgis_usr_privileges(usr_name := 'konstantinos', priv_type := 'rw', cdb_schema := 'citydb_2');
```

# QGIS Package: User management

The database administrator can grant read-only (ro) or read-write (rw) privileges to the users

- Alternatively, privileges can be granted to multiple (or all) citydb schemas at the same time; Please note the **cdb\_schemas** (with s!) parameter, requiring an array of varchar values containing the schema names, or NULL for all schemas

## SQL examples

```
-- Database user "camilo" is added to group "qgis_pkg_usrgroup_starwars", can access data in citydb schemas  
"alderaan" and "hoth" of the current database "starwars" with read-only privileges
```

```
SELECT qgis_pkg.grant_qgis_usr_privileges(usr_name := 'camilo', priv_type := 'ro', cdb_schemas :=  
ARRAY['alderaan', 'hoth']);
```

```
-- Database user "giorgio" is added to group "qgis_pkg_usrgroup_starwars", can access data in ALL citydb schemas of  
the current database "starwars" with read-write privileges
```

```
SELECT qgis_pkg.grant_qgis_usr_privileges(usr_name := 'giorgio', priv_type := 'rw', cdb_schemas := NULL);
```

# QGIS Package: User management

The database administrator can revoke privileges from the users

## SQL examples

-- Database user "giorgio" cannot access anymore data in citydb schema "citydb" of the current database

```
SELECT qgis_pkg.revoke_qgis_usr_privileges('giorgio', 'citydb');
```

-- Database user "camilo" cannot access anymore ANY citydb schemas of the current database

```
SELECT qgis_pkg.revoke_qgis_usr_privileges('camilo');
```

**IMPORTANT:** The database users are NOT automatically removed from the group "qgis\_pkg\_usrgroup\_{cdb\_schema}". If required, the administrator must remove them manually (or use the QGIS plug-in GUI).

-- Database user "giorgio" is removed from group "qgis\_pkg\_usrgroup\_starwars" and won't be able to use the QGIS plugin anymore for the database "starwars"

```
REVOKE qgis_pkg_usrgroup_starwars FROM giorgio;
```

# QGIS Package: Layer management

The database administrator can create layers with function **qgis\_pkg.create\_layers(...)**.

- All materialized views and updatable views are created, but only if corresponding data exist in the database
- The user can create layers only for selected CityGML modules using the similar functions:
  - **qgis\_pkg.create\_layers\_bridge(...)**
  - **qgis\_pkg.create\_layers\_building(...)**
  - ...
  - **qgis\_pkg.create\_layers\_waterbody(...)**
- All functions are in schema **qgis\_pkg** and have the same signature (see next slide)

# QGIS Package: Layer management

Motivation  
 Plugin overview  
 Software installation  
 Software use  
 Advanced options  
 Software uninstall  
 Current limitations  
**QGIS Package**  
 Resources

## Function

```
qgis_pkg.create_layers(usr_schema, cdb_schema [, perform_snapping] [, digits]
[, area_poly_min] [, bbox_corners_array] [, is_geographic] [, force_layer_creation])
```

Parameter	Type	Description
<b>usr_schema</b>	varchar	The database user schema, e.g. "qgis_giorgio".
<b>cdb_schema</b>	varchar	the citydb schema where data are stored, e.g. "citycb", or "citydb2".
<b>perform_snapping</b>	integer	DEFAULT 0 (i.e. disabled). If 1, geometry simplification is performed. Unused if perform_snapping is set to 0.
<b>digits</b>	integer	DEFAULT 3. Number of decimal positions to keep during geometry simplification. Unused if perform_snapping is set to 0.
<b>area_poly_min</b>	numeric	DEFAULT 0.001 [m <sup>2</sup> ]. Minimum polygon area during geometry simplification. Unused if perform_snapping is set to 0.
<b>bbox_corners_array</b>	numeric[]	DEFAULT Null, i.e. the extents of the whole <i>cdb_schema</i> . Otherwise, extents of the materialized views to be created, e.g. {x_min, y_min, x_max, y_max}. Coordinates must be in the same SRS as the <i>cdb_schema</i> !
<b>is_geographic</b>	boolean	DEFAULT False. True is the coordinate system of the citydb schema has geographic coordinates, False if it has projected coordinates.
<b>force_layer_creation</b>	boolean	DEFAULT False. Otherwise: force creation of all layers, also the empty ones.

# QGIS Package: Layer management

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
**QGIS Package**  
Resources

## SQL examples

-- For user "giorgio", create all layers for existing data in citydb schema "citydb"

```
SELECT qgis_pkg.create_layers('giorgio', 'citydb');
```

-- For user "giorgio", create all layers for existing data in citydb schema "citydb2" and perform geometry simplification with 2 decimal places and 0.01 m<sup>2</sup> minimum area for polygons

```
SELECT qgis_pkg.create_layers('giorgio', 'citydb', 1, 2, 0.01);
```

-- For user "camilo", create all building module layers for existing data in citydb schema "vienna"

```
SELECT qgis_pkg.create_layers_building('camilo', 'vienna');
```

-- For user "konstantinos", create all waterbody module layers for existing data in citydb schema "alderaan" within a certain bounding box. The bounding box is defined by the lower-left and upper-right coordinates (i.e. x\_min, y\_min, x\_max, y\_max)

```
SELECT qgis_pkg.create_layers_waterbody('konstantinos', 'alderaan', bbox_corners_array := '{10, 20, 110, 220}');
```

# QGIS Package: Layer management

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
**QGIS Package**  
Resources

## SQL examples

-- The following query works, but actually it is not written in a user-friendly way.

```
SELECT qgis_pkg.create_layers('giorgio', 'citydb', 1, 2, 0.01);
```

-- In general, therefore, it is always a good habit to use **named parameters** when calling functions!

```
SELECT qgis_pkg.create_layers(  
    usr_name := 'giorgio',  
    cdb_schema := 'citydb',  
    perform_snapping := 1,  
    digits := 2,  
    area_poly_min := 0.01)
```

# QGIS Package: Layer management

After creating the layers, you may (optionally) want to register also their bounding box in the EXTENTS table of the respective qgis\_schema. This will enable the plugin in QGIS to "see" and represent the extents also from the plugin GUI.

## SQL examples

```
-- In user schema "qgis_giorgio", update the extents for the whole dataset in citydb schema "citydb"  
SELECT qgis_pkg.upsert_extents('qgis_giorgio', 'citydb', 'db_schema');
```

```
-- In user schema "qgis_giorgio", update the extents for the user-defined bounding box in citydb schema "citydb"  
SELECT qgis_pkg.upsert_extents(  
    usr_schema := 'qgis_giorgio',  
    cdb_schema := 'citydb',  
    cdb_bbox_type := 'm_view', -- must be one of ('db_schema', 'm_view', 'qgis')  
    cdb_envelope := ST_Envelope('LINESTRING(232038 480366, 232600 480856)::geometry),  
    is_geographic := FALSE);
```

**Please note:** the *cdb\_envelope* parameter requires a PostGIS geometry that is a polygon. As long as this requirement is fulfilled, other PostGIS functions can be used, for example:

- `ST_GeomfromText('POLYGON(...))')`
- `ST_MakePolygon(...)`

The SRID can be omitted as it is automatically set based on the one from the selected citydb schema. The coordinates must be however in the same CRS of the selected citydb!

# QGIS Package: Layer management

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
**QGIS Package**  
Resources

The database administrator can refresh the materialized views with function **qgis\_pkg.refresh\_layers(...)**.

- All materialized views created before will be refreshed. This is necessary every time the layers are generated (or re-generated using a different bounding box).
- The user can refresh the materialized views only for selected CityGML modules using the similar functions:
  - **qgis\_pkg.refresh\_layers\_bridge(...)**
  - **qgis\_pkg.refresh\_layers\_building(...)**
  - ...
  - **qgis\_pkg.refresh\_layers\_waterbody(...)**
- All functions are in schema qgis\_pkg and have the same signature (see next slide)

# QGIS Package: Layer management

## Function

**qgis\_pkg.refresh\_layers(usr\_schema, cdb\_schema)**

Parameter	Type	Description
<b>usr_schema</b>	varchar	The database user schema, e.g. "qgis_giorgio".
<b>cdb_schema</b>	varchar	the citydb schema where data are stored, e.g. "citycb", or "citydb2".

## SQL examples

-- In user schema "qgis\_giorgio", refresh all materialized views in citydb schema "citydb"

```
SELECT qgis_pkg.refresh_layers('qgis_giorgio', 'citydb');
```

-- In user schema "qgis\_konstantinos", refresh all waterbody module materialized views in citydb schema "alderaan"

```
SELECT qgis_pkg.refresh_layers_waterbody('qgis_konstantinos', 'alderaan');
```

# QGIS Package: Layer management

The database administrator can drop layers with function **qgis\_pkg.drop\_layers(...)**.

- All existing layers in the selected user schema and related to the selected citydb schema will be dropped
- The user can drop layers only for selected CityGML modules using the similar functions:
  - **qgis\_pkg.drop\_layers\_bridge(...)**
  - **qgis\_pkg.drop\_layers\_building(...)**
  - ...
  - **qgis\_pkg.drop\_layers\_waterbody(...)**
- All functions are in schema **qgis\_pkg** and have the same signature (see next slide)

# QGIS Package: Layer management

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
**QGIS Package**  
Resources

## Function

### **qgis\_pkg.drop\_layers(usr\_schema, cdb\_schema)**

Parameter	Type	Description
<b>usr_schema</b>	varchar	The database user schema, e.g. "qgis_giorgio".
<b>cdb_schema</b>	varchar	the citydb schema where data are stored, e.g. "citycb", or "citydb2".

## SQL examples

-- In user schema "qgis\_giorgio", drop all layers related to citydb schema "citydb"

```
SELECT qgis_pkg.drop_layers('qgis_giorgio', 'citydb');
```

-- In user schema "qgis\_konstantinos", drop all waterbody module layers related to citydb schema "alderaan"

```
SELECT qgis_pkg.drop_layers_waterbody('qgis_konstantinos', 'alderaan');
```

# QGIS Package: Detail views

Detail views are children tables containing additional layer data (e.g. generic attributes, external references, etc.)

They can be created and dropped in a similar way as normal layers. If created, they will be used (also) by the plugin GUI in QGIS to link these views to the attribute forms as nested tables.

**Important:** they must be created with the same extents (bounding box) used for the layers

## SQL examples

-- For user "giorgio", create the detail views for citydb schema "citydb"

```
SELECT qgis_pkg.create_details_view('giorgio', 'citydb');
```

-- For user "giorgio", create the detail views for citydb schema "citydb" inside the user-defined bounding box

```
SELECT qgis_pkg.create_detail_view(  
    usr_name := 'giorgio',  
    cdb_schema := 'citydb',  
    bbox_corners_array := ARRAY[232038, 480366, 232600, 480856],  
    is_geographic := FALSE);
```

-- In user schema "qgis\_giorgio", drop the detail views for citydb schema "citydb"

```
SELECT qgis_pkg.drop_detail_view('qgis_giorgio', 'citydb');
```

# QGIS Package: Drop user schema

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
**QGIS Package**  
Resources

The database administrator can drop a user schema

FIRST, revoke privileges of the user for all citydb schemas, THEN drop the user schema.  
Please refer to the previous slides for more details about user privileges.

SQL example

```
-- First revoke all ro/rw privileges of user "giorgio" for all citydb schemas
SELECT qgis_pkg.revoke_qgis_usr_privileges('giorgio');

-- Then drop the layers using the drop_layer_x functios
SELECT qgis_pkg.drop_layers_building('qgis_giorgio');
SELECT qgis_pkg.drop_layers_bridge('qgis_giorgio');

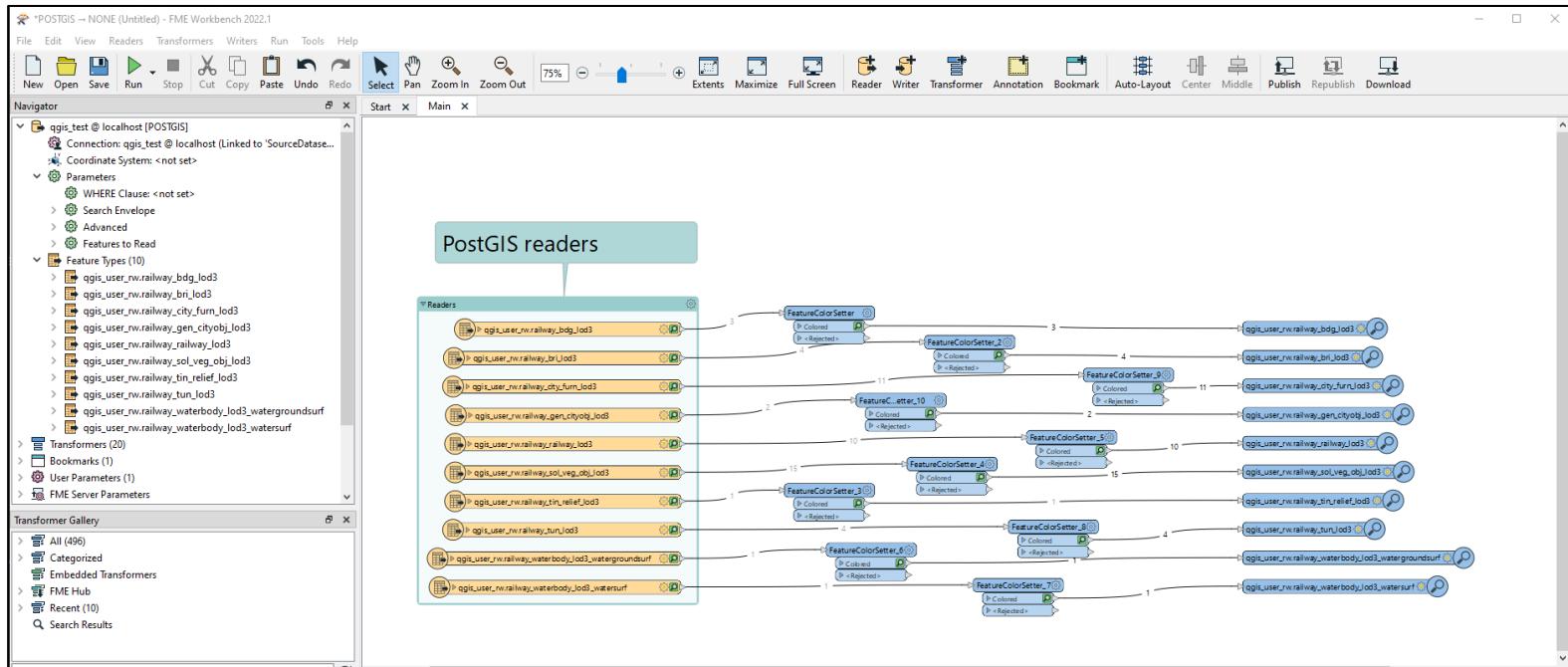
...
-- Then drop the user schema
DROP SCHEMA qgis_giorgio CASCADE;
-- Optionally (if necessary) remove user "giorgio" from the "qgis_pkg_usrgroup_qgis_test" associated to database
-- "qgis_test"
REVOKE qgis_pkg_usrgroup_qgis_test FROM giorgio;
```

# QGIS Package via FME

This is a simple example of how the QGIS package can be used via FME

- Simply connect to the 3D City Database and import the views with **PostGIS readers**

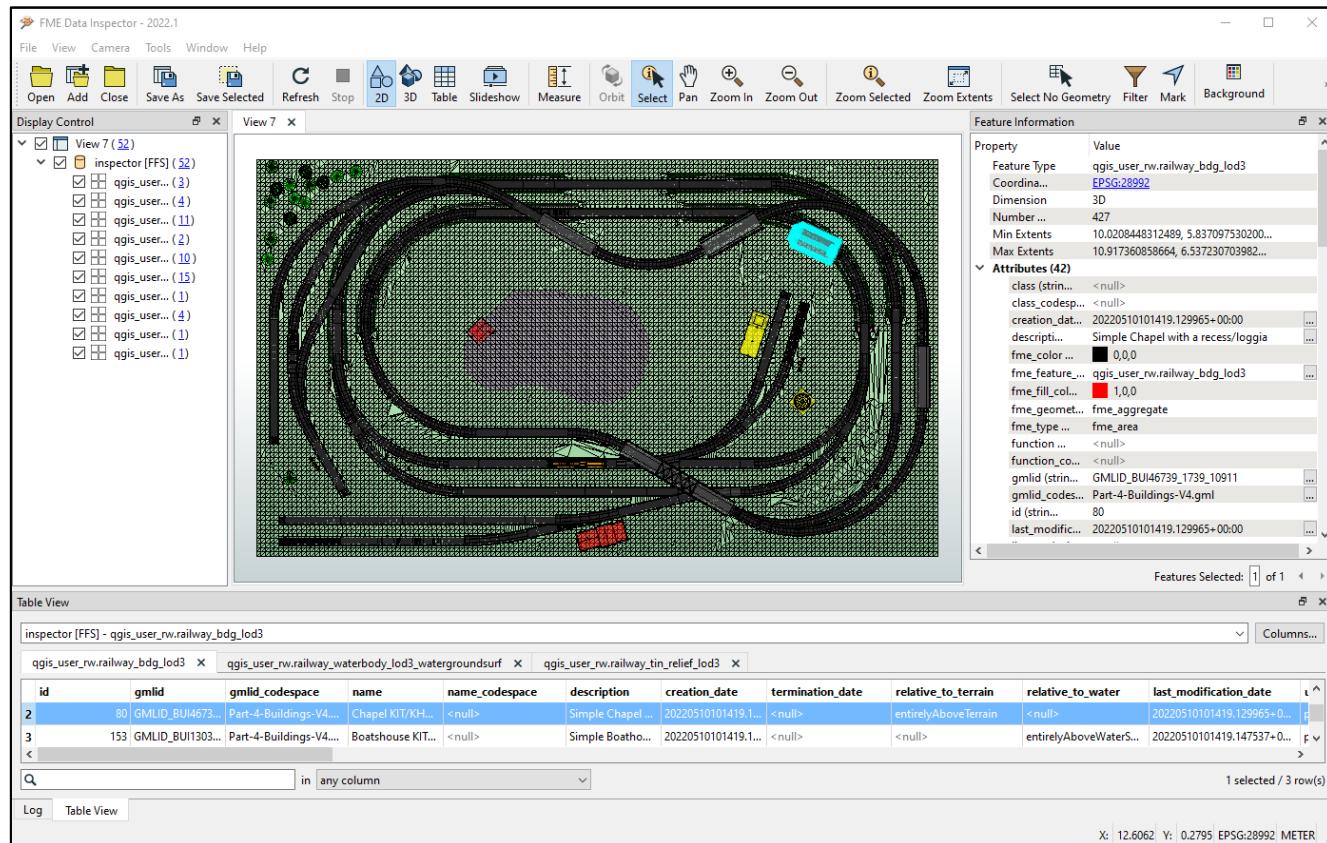
Motivation  
 Plugin overview  
 Software installation  
 Software use  
 Advanced options  
 Software uninstall  
 Current limitations  
**QGIS Package**  
 Resources



# QGIS Package via FME

- 2D visualisation via FME Data Inspector

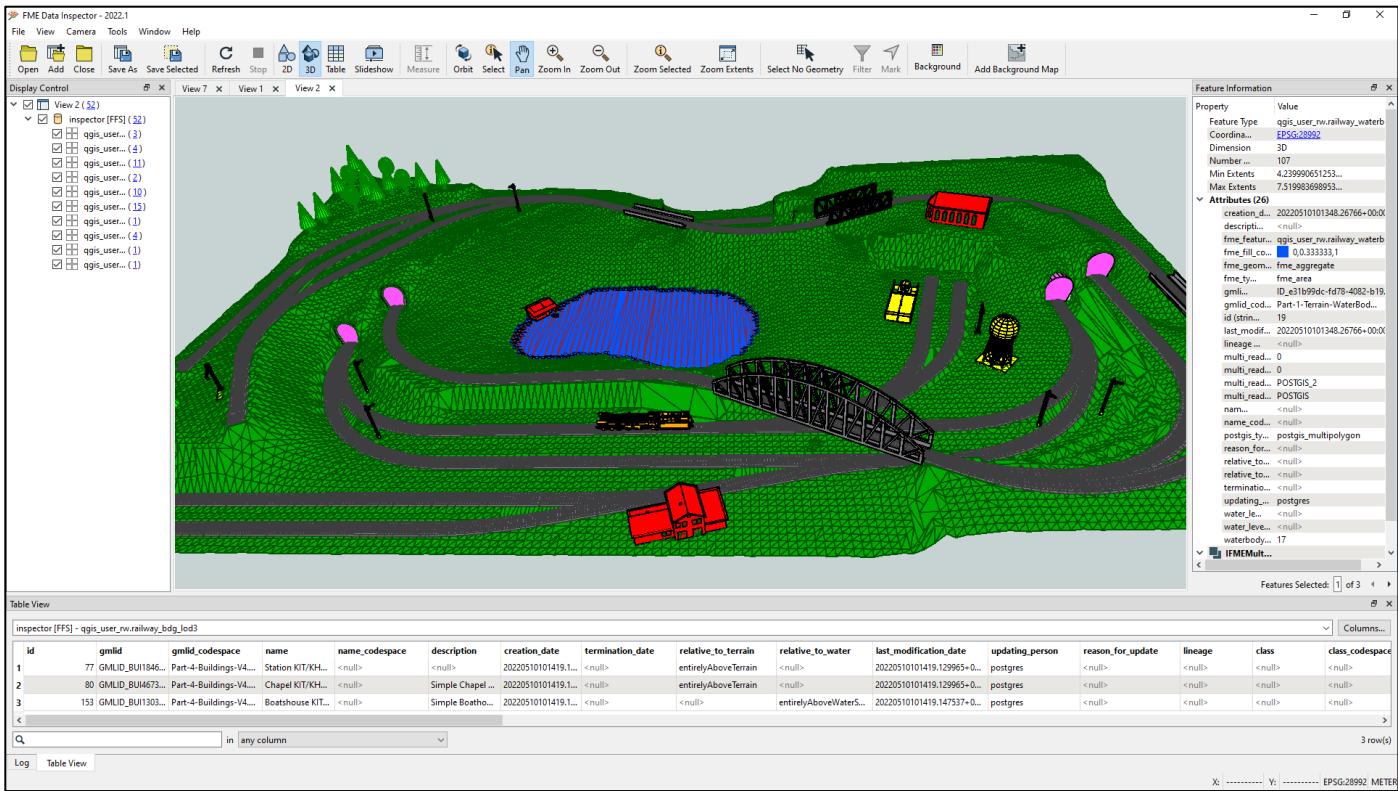
Motivation  
 Plugin overview  
 Software installation  
 Software use  
 Advanced options  
 Software uninstall  
 Current limitations  
**QGIS Package**  
 Resources



# QGIS Package via FME

- 3D visualisation via FME Data Inspector

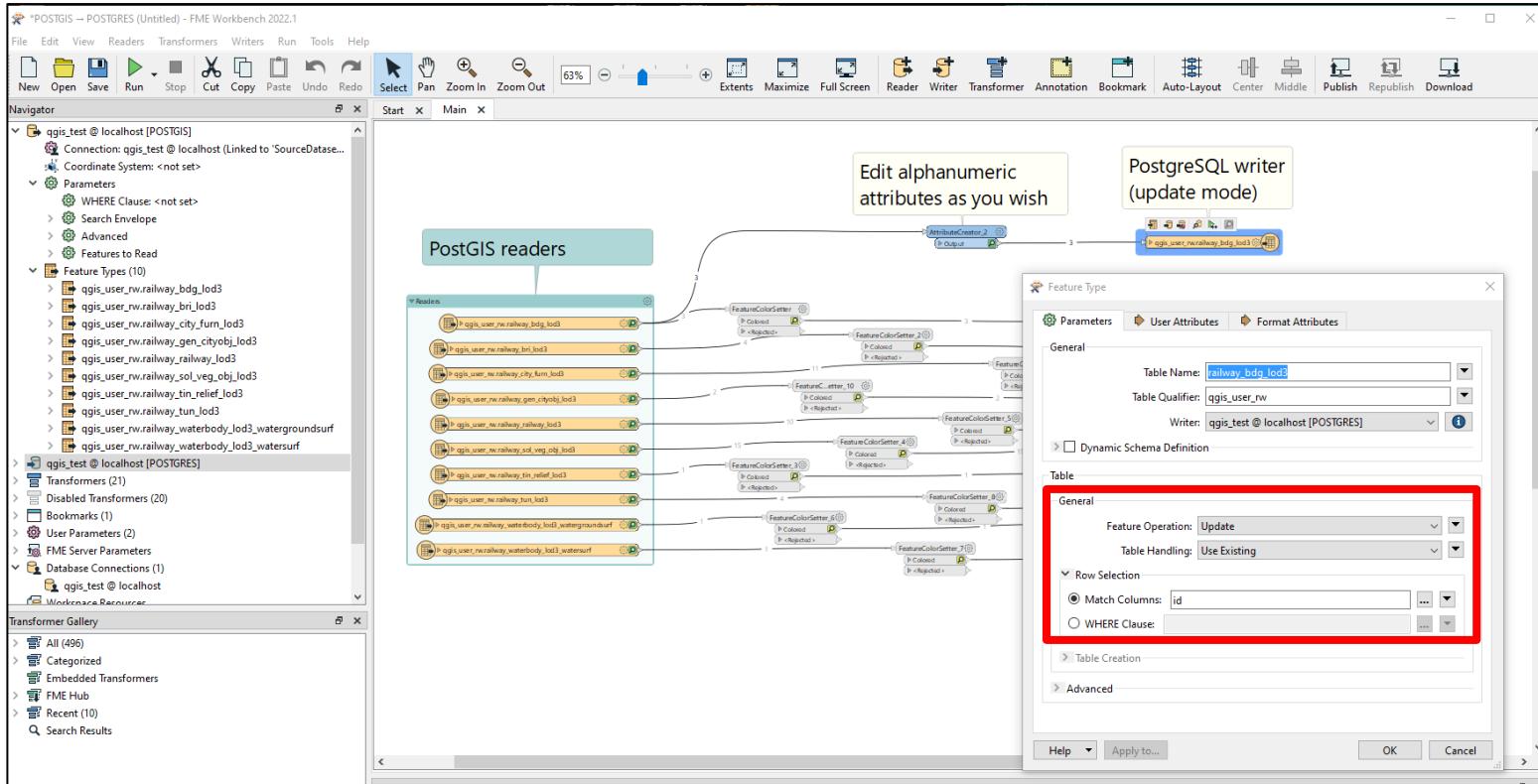
Motivation  
 Plugin overview  
 Software installation  
 Software use  
 Advanced options  
 Software uninstall  
 Current limitations  
**QGIS Package**  
 Resources



# QGIS Package via FME

- Remember: alphanumeric attributes in the views are updatable! 😊
- You will need a PostgreSQL writer in update mode

Motivation  
 Plugin overview  
 Software installation  
 Software use  
 Advanced options  
 Software uninstall  
 Current limitations  
**QGIS Package**  
 Resources



Motivation

Plugin  
overview

Software  
installation

Software  
use

Advanced  
options

Software  
uninstall

Current  
limitations

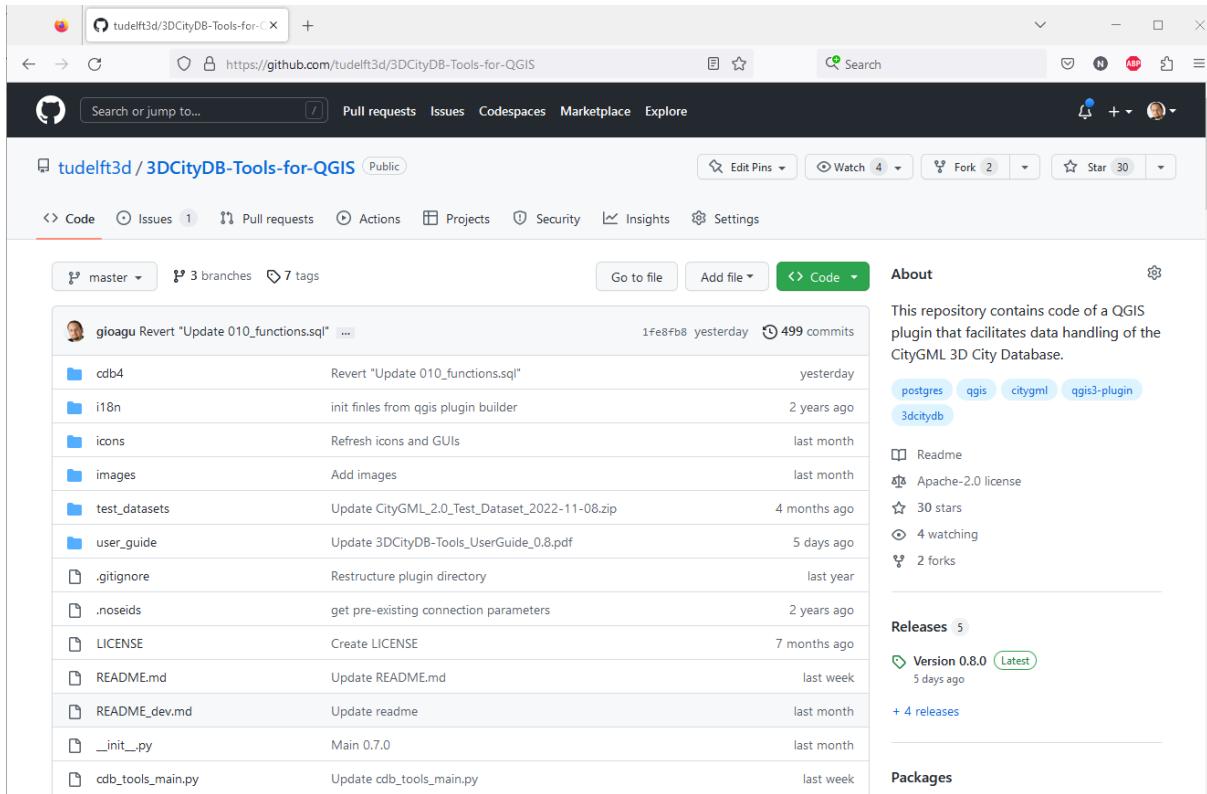
QGIS  
Package

Resources

# Resources

## Source code and GitHub repository

- GitHub: <https://github.com/tudelft3d/3DCityDB-Tools-for-QGIS>



The screenshot shows the GitHub repository page for `tudelft3d / 3DCityDB-Tools-for-QGIS`. The repository is public and has 499 commits. The main page displays a list of recent commits, including updates to functions.sql, i18n files, icons, images, test datasets, user guides, and plugin restructuring. The repository is associated with PostgreSQL, QGIS, CityGML, and QGIS3-Plugin. It has 30 stars, 4 forks, and 4 watchers. A release section shows Version 0.8.0 (Latest) and 4 other releases. A packages section is also present.

This repository contains code of a QGIS plugin that facilitates data handling of the CityGML 3D City Database.

Tags: postgres, qgis, citygml, qgis3-plugin, 3dcitydb

Releases: 5

Version 0.8.0 (Latest)  
5 days ago

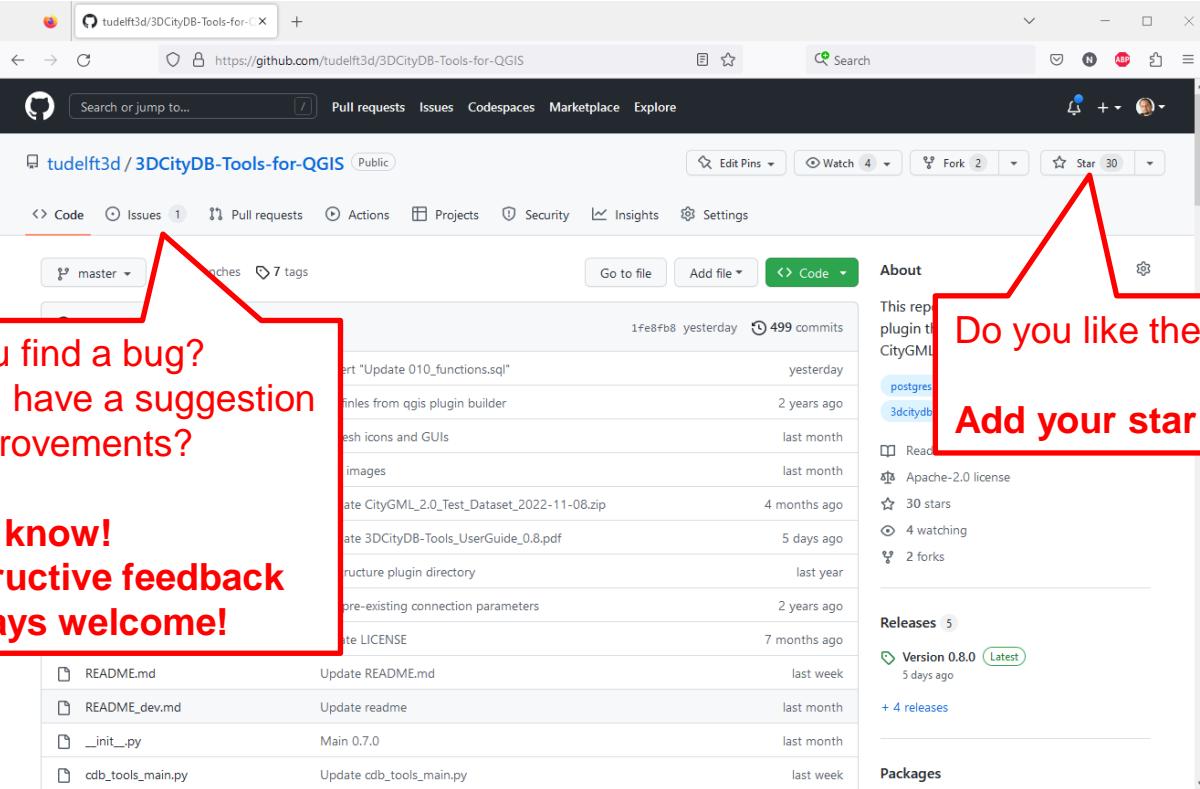
+ 4 releases

Packages

# Resources

## Source code and GitHub repository

- GitHub: <https://github.com/tudelft3d/3DCityDB-Tools-for-QGIS>



Did you find a bug?  
Do you have a suggestion  
for improvements?

Let us know!  
Constructive feedback  
is always welcome!

Do you like the plugin?  
Add your star! ☺

This repository provides a QGIS plugin for working with CityGML files. It includes support for PostgreSQL and 3DCityDB databases. The plugin is released under the Apache-2.0 license.

Releases: 5

Version 0.8.0 (Latest)

+ 4 releases

Packages

File	Description	Last Commit
README.md	Update README.md	last week
README_dev.md	Update readme	last month
__init__.py	Main 0.7.0	last month
cdb_tools_main.py	Update cdb_tools_main.py	last week

# Resources

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
**Resources**

## QGIS Plug-in repository

- <https://plugins.qgis.org/plugins/citydb-tools/>



The screenshot shows a web browser window displaying the QGIS Python Plugins Repository. The URL in the address bar is <https://plugins.qgis.org/plugins/citydb-tools/>. The page title is "QGIS Python Plugins Repository". On the left, there is a sidebar with a "Upload a plugin" button and a "Plugins" section containing links for "Featured", "All", "Stable", "New Plugins", "Updated Plugins", "Experimental", "Popular", "Most voted", "Top downloads", "Most rated", and "QGIS Server plugins". Below this is a "Plugin Tags" section. The main content area features a large "3DCityDB Tools" plugin card. The card includes a "Download latest" button, a 5-star rating with "(52) votes", and a description: "Tools to visualize and manipulate CityGML data stored in the 3D City Database". It has tabs for "About", "Details", and "Versions". The "About" tab is selected, showing text about the plugin connecting to the 3D City Database (v. 4.x) for PostgreSQL/PostGIS. It lists three features: "Layer Loader", "Bulk Deleter", and "QGIS Package Administrator". At the bottom, it says "Further details can be found in the PDF files contained in the \user\_guide subfolder of the plugin installation directory. On the GitHub repository, some test datasets can be downloaded."

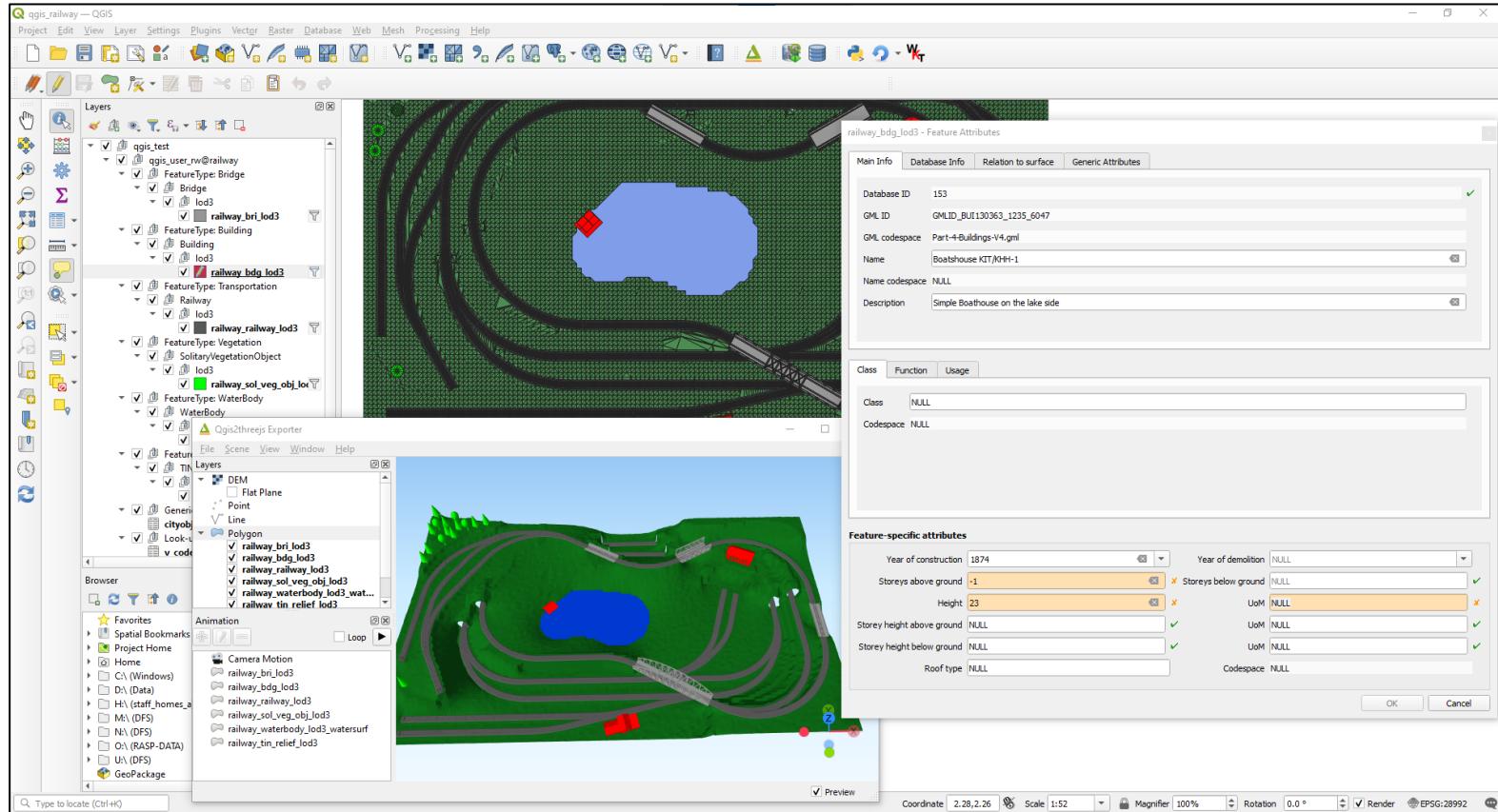
# Resources

Motivation  
Plugin overview  
Software installation  
Software use  
Advanced options  
Software uninstall  
Current limitations  
QGIS Package  
**Resources**

## Test datasets

- In the GitHub repository, you will find test datasets that you can import into the 3DCityDB (using the Importer/Exporter) to test the 3DCityDB-Tools plugin. They are located in subfolder \test\_datasets
  - The test datasets are:
    - CityGML\_2.0\_Test\_Dataset\_2022-03-11.zip (aka "Railway")
    - FZK-Haus-LoD-all-KIT-IAI-KHH-B36-V1.zip (aka "Kit House")
    - DenHaag\_bdg\_lod2.zip
  - You can find links to many additional free and open CityGML/CityJSON datasets at:
    - Awesome CityGML: <https://github.com/OloOcki/awesome-citygml>
    - 3D Geoinformation group @ TU Delft: <https://3d.bk.tudelft.nl/opendata/opencities/>

# Enjoy! ☺



# Main developers



**Dr. Giorgio Agugiaro**

[g.agugiaro@tudelft.nl](mailto:g.agugiaro@tudelft.nl)

3D Geoinformation Group

TU Delft

The Netherlands

<https://3d.bk.tudelft.nl/gagugiaro>



**Konstantinos Pantelios**

[konstantinospantelios@yahoo.com](mailto:konstantinospantelios@yahoo.com)

MSc Geomatics graduate

TU Delft

The Netherlands

## Acknowledgements / suggestions and contributions by:

Tendai Mbwanda, Camilo León-Sánchez (TU Delft), Claus Nagel and Zhihang Yao (Virtual City Systems GmbH)



# 3DCityDB Tools

for



QGIS