



3DCityDB Tools

for

QGIS

Installation and user guide

Version 0.8.4

Last update: 28 August 2023

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
Changelog

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!)

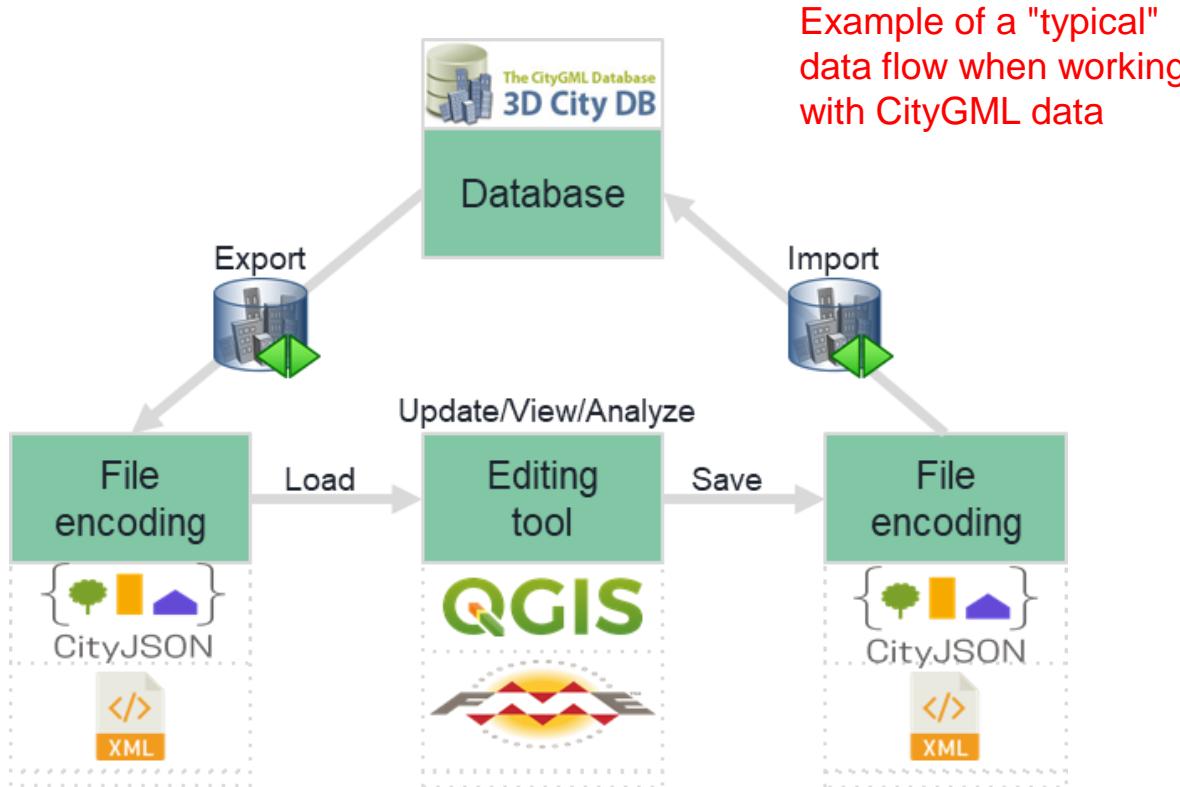
Motivation

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

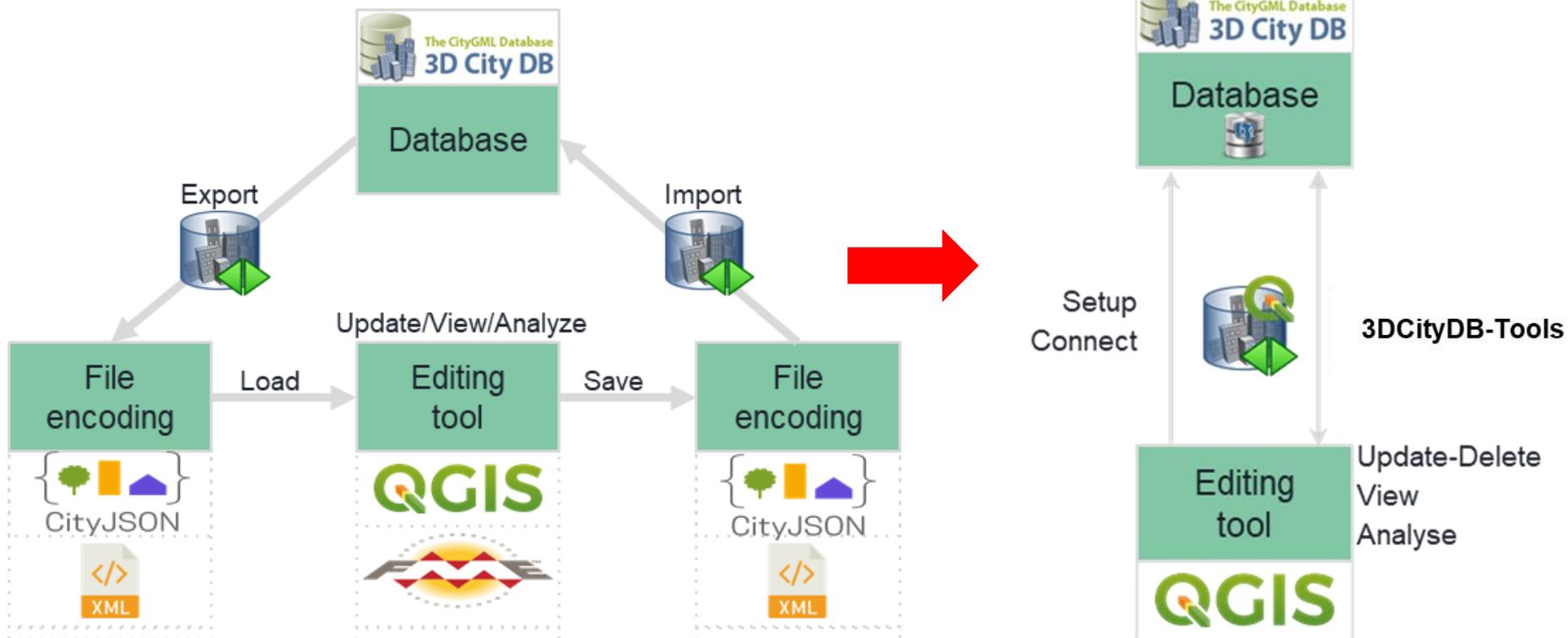


Motivation



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
Changelog

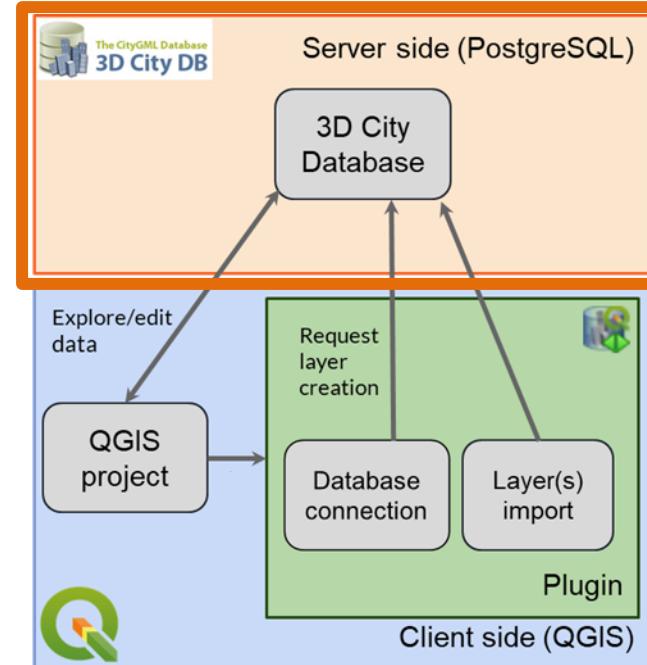
- Create "**SFS-like layers**" to hide 3DCityDB complexity when interacting with data
 - Deal efficiently with multi-LoD / different geometries / implicit representations
 - Up to ≈600 possible combinations in CityGML!
 - Merge all standard attributes of a CityObject into a single "table"
- **Deal with 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
Changelog

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



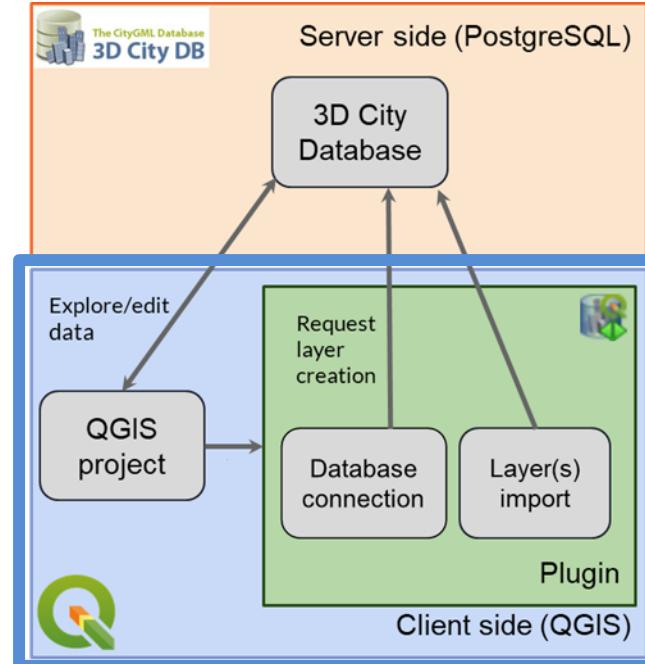
Plugin overview

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

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
Changelog

- 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 or v. 3.28 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

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

Compatibility matrix

QGIS version	3DCityDB-Tools works?	Comments
QGIS 3.30, 3.32	✓	Should work, but not supported
QGIS 3.28 LTR	✓	Works
QGIS 3.26, 3.24	✓	Should work, but not supported
QGIS 3.22 LTR	✓	Works. Reference version used for development
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	14, 64bit	3.2
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

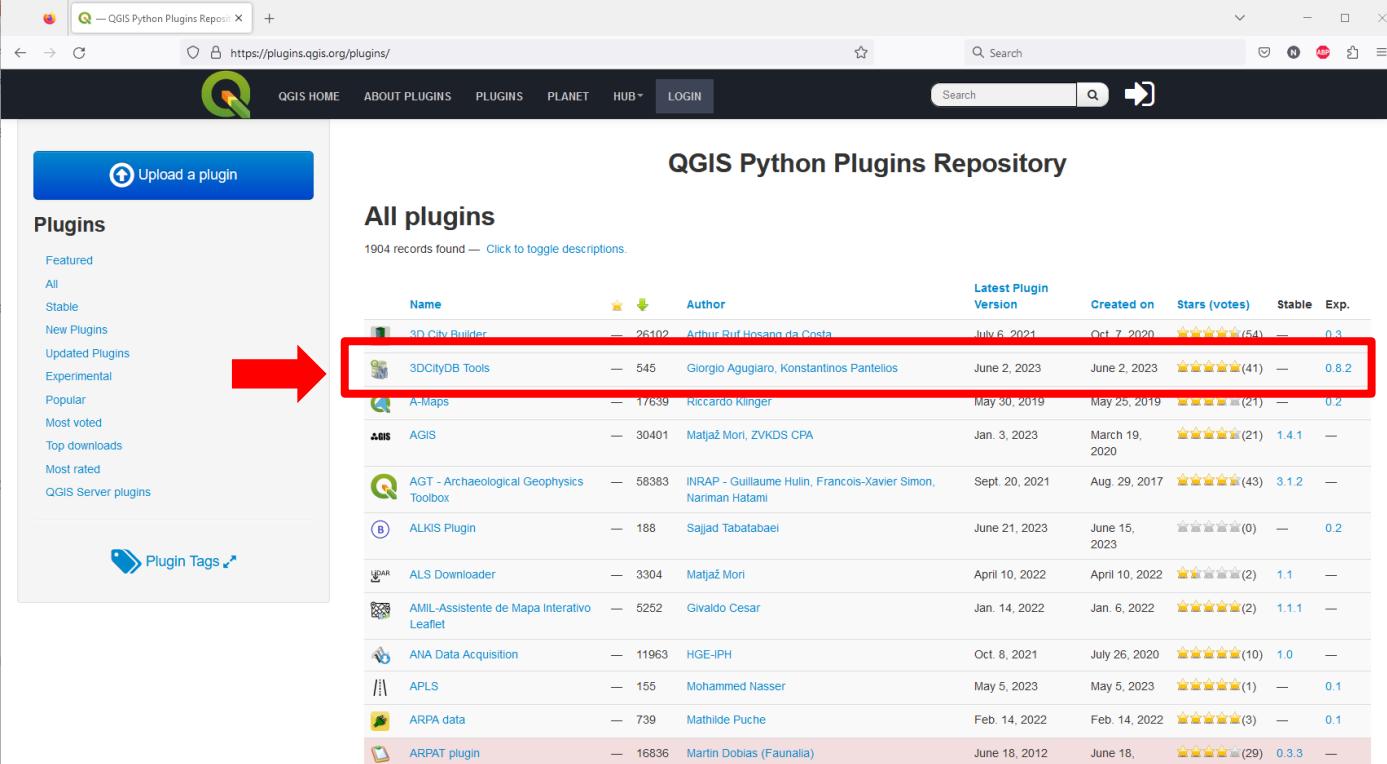
QGIS
Package

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
Changelog

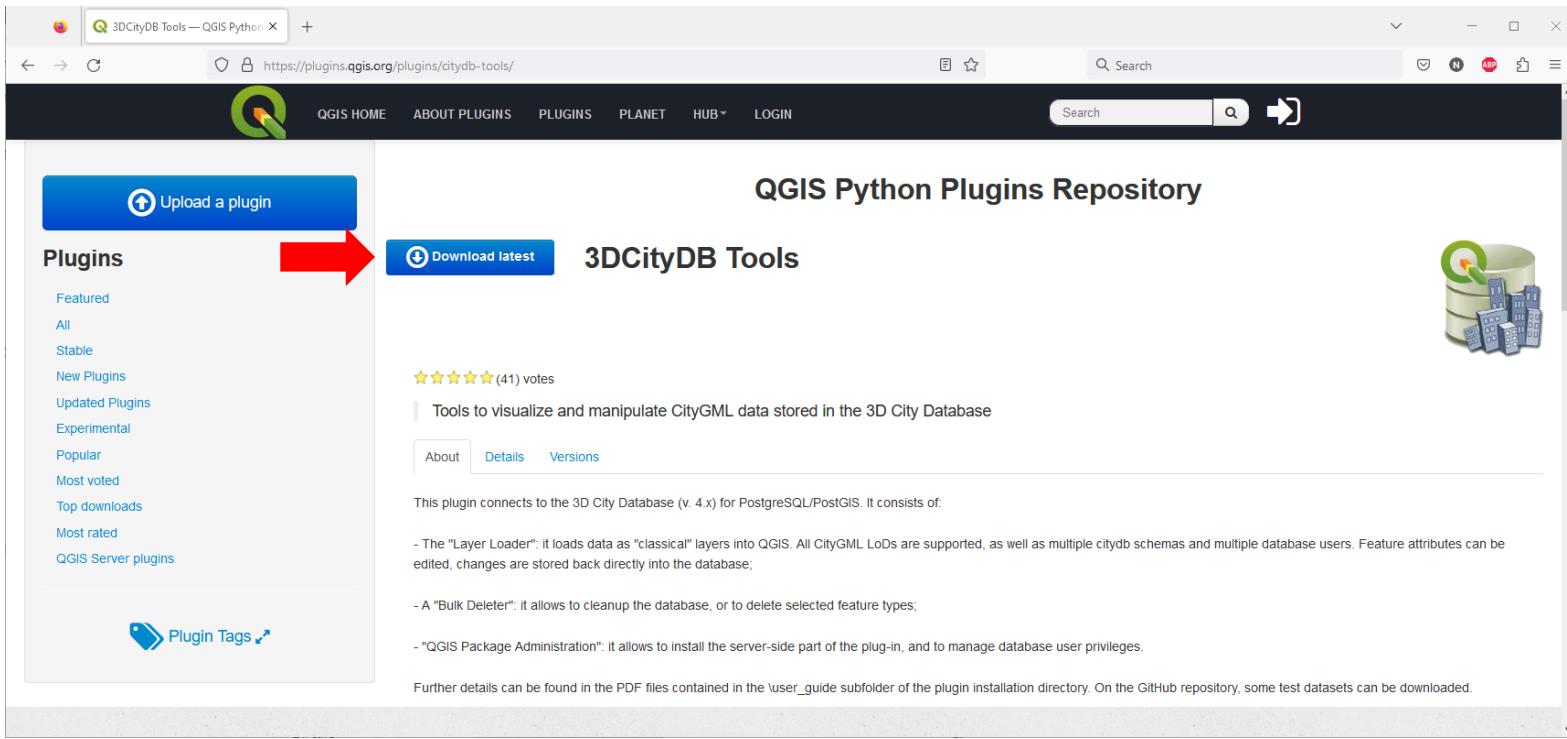


The screenshot shows the QGIS Python Plugins Repository website. On the left, there's a sidebar with a 'Upload a plugin' button and a 'Plugins' section containing various filter options like 'Featured', 'All', 'Stable', 'New Plugins', etc. Below that is a 'Plugin Tags' section. On the right, the main content area is titled 'QGIS Python Plugins Repository' and shows a table of 'All plugins'. The table has columns for Name, Author, Latest Plugin Version, Created on, Stars (votes), Stable, and Exp. A red arrow points from the sidebar towards the plugin list, and a red box highlights the '3DCityDB Tools' row, which is also circled in red. The '3DCityDB Tools' row includes details such as 545 downloads, author Giorgio Agugiaro and Konstantinos Pantelios, latest version 0.8.2 from June 2, 2023, and a 4.1 star rating from 41 votes.

Name	Author	Latest Plugin Version	Created on	Stars (votes)	Stable	Exp.
3DCityDB Tools	Giorgio Agugiaro, Konstantinos Pantelios	0.8.2	June 2, 2023	4.1 (41)	—	0.2
A-Maps	Riccardo Klinger	0.2	May 30, 2019	3.8 (21)	—	0.2
AGIS	Matjaž Mori, ZVKDS CPA	1.4.1	Jan. 3, 2023	4.0 (21)	—	—
AGT - Archaeological Geophysics Toolbox	INRAP - Guillaume Hulin, François-Xavier Simon, Naiman Hatami	3.1.2	Sept. 20, 2021	4.8 (43)	—	—
ALKIS Plugin	Sajjad Tabatabaei	0.2	June 21, 2023	3.5 (0)	—	0.2
ALS Downloader	Matjaž Mori	1.1	April 10, 2022	4.5 (2)	—	—
AMIL-Assistente de Mapa Interativo Leaflet	Givaldo Cesar	1.1.1	Jan. 14, 2022	4.5 (2)	—	—
ANA Data Acquisition	HGE-IPH	1.0	Oct. 8, 2021	4.5 (10)	—	—
APLS	Mohammed Nasser	0.1	May 5, 2023	4.5 (1)	—	0.1
ARPA data	Mathilde Puche	0.1	Feb. 14, 2022	4.5 (3)	—	0.1
ARPAT plugin	Marlin Dobias (Faunalia)	0.3.3	June 18, 2012	4.5 (29)	—	—

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! ☺



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 sidebar on the left with a 'Plugins' section containing links like 'Featured', 'All', 'Stable', etc., and a 'Plugin Tags' section. A red arrow points from the 'Plugins' sidebar to the 'Download latest' button for the '3DCityDB Tools' plugin. The main content area is titled 'QGIS Python Plugins Repository' and shows the '3DCityDB Tools' plugin details, including its rating (4.1 stars), description ('Tools to visualize and manipulate CityGML data stored in the 3D City Database'), and a list of features.

QGIS Python Plugins Repository

3DCityDB Tools

Upload a plugin

Plugins

Download latest

Featured

All

Stable

New Plugins

Updated Plugins

Experimental

Popular

Most voted

Top downloads

Most rated

QGIS Server plugins

Plugin Tags

3DCityDB Tools

(41) votes

Tools to visualize and manipulate CityGML data stored in the 3D City Database

About Details Versions

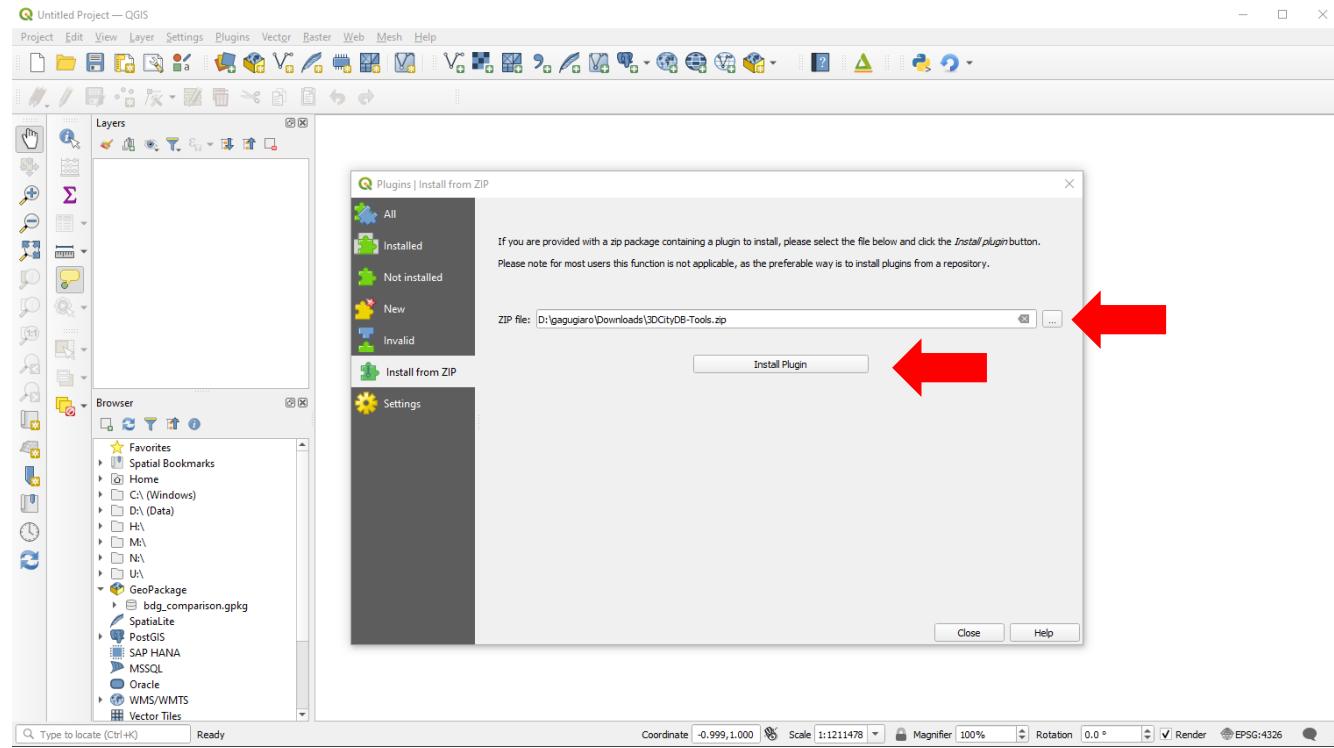
This plugin connects to the 3D City Database (v. 4.x) for PostgreSQL/PostGIS. It consists of:

- 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.

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

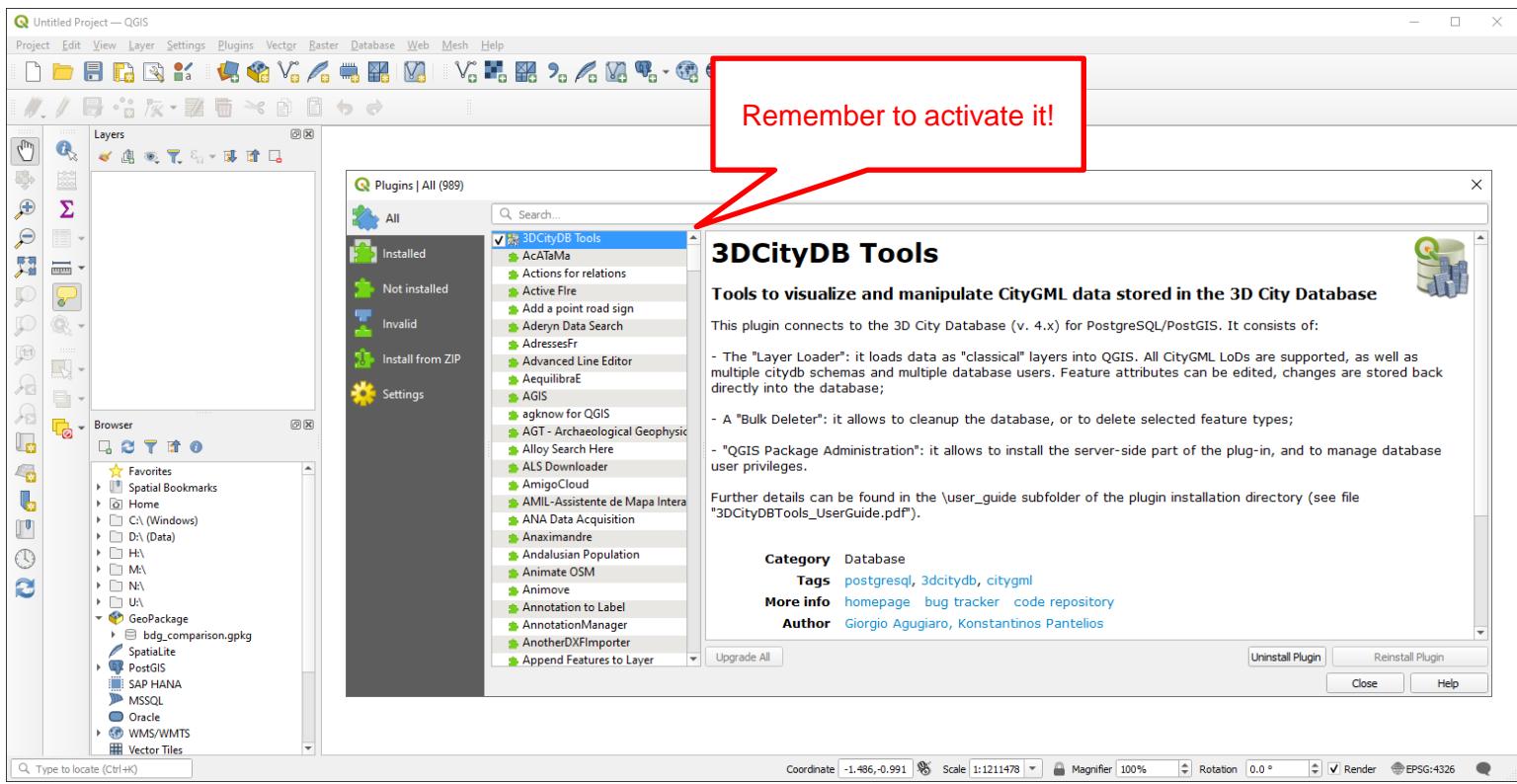
- The plugin is available also as **zip file** from the **GitHub repository**
- In QGIS, open the Plugins\Manage and install plugins window, and choose "Install from ZIP". Select the zip file and click the "Install Plugin" button



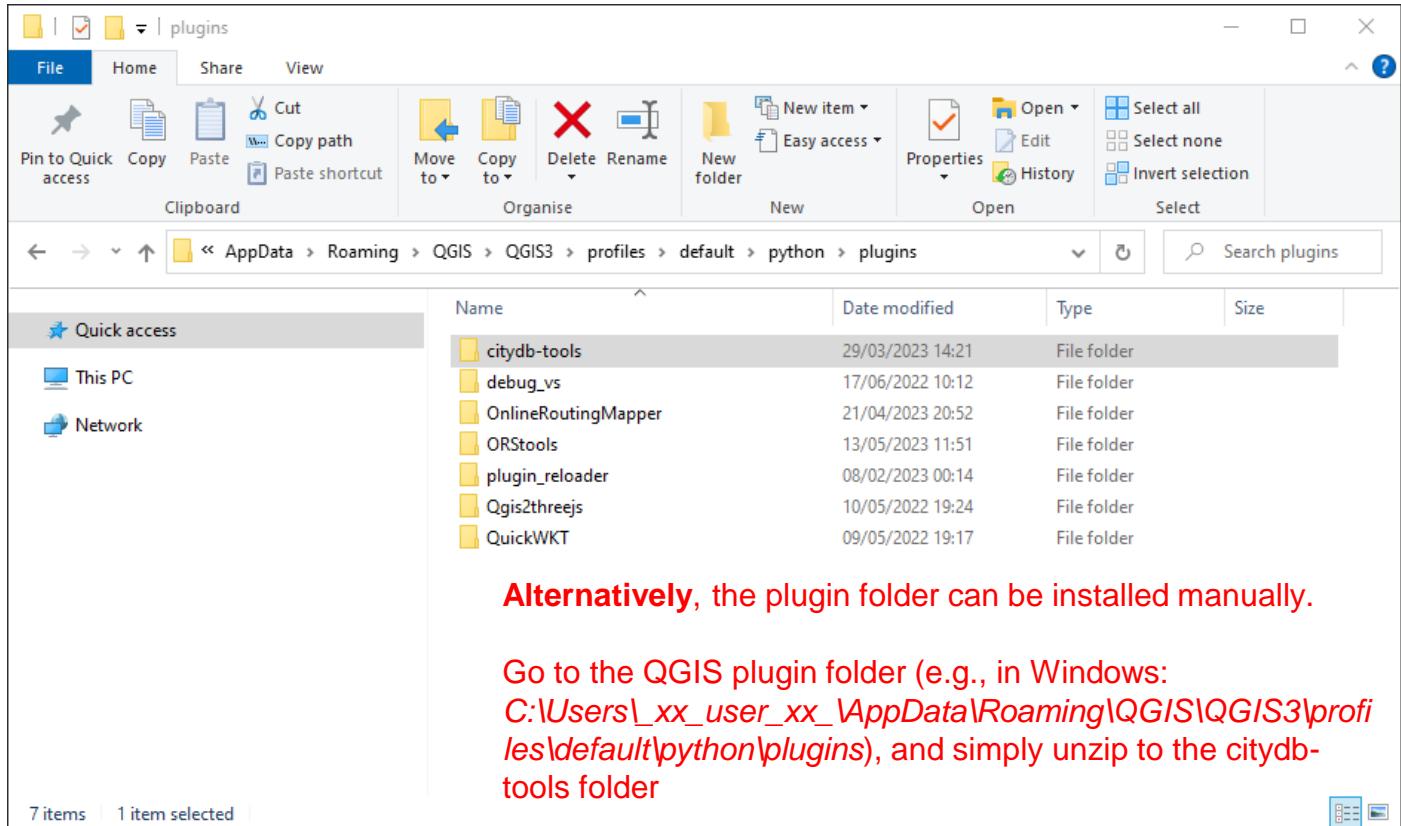
Manual installation

- Upon installation, you must activate the plugin

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

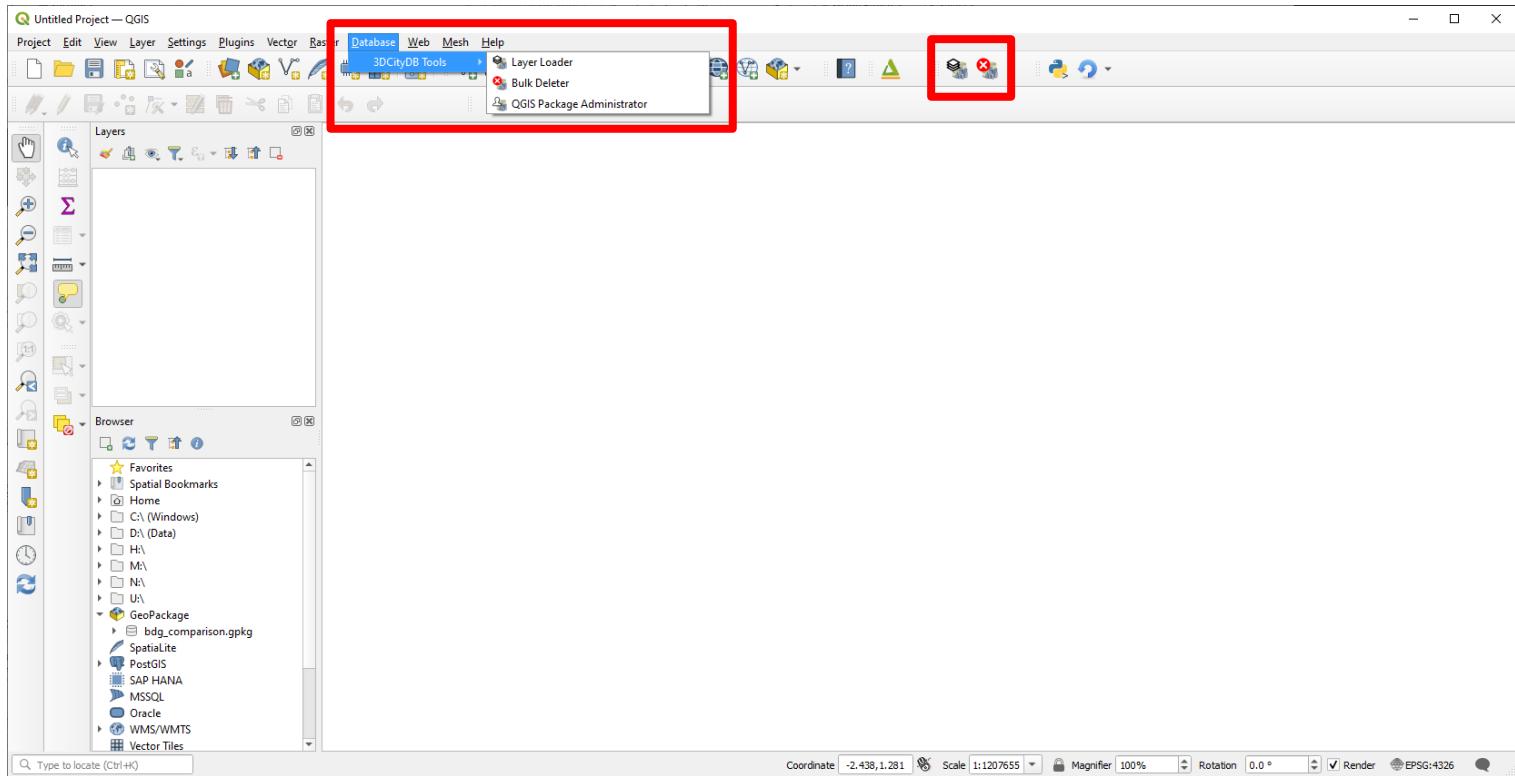


Manual installation



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:

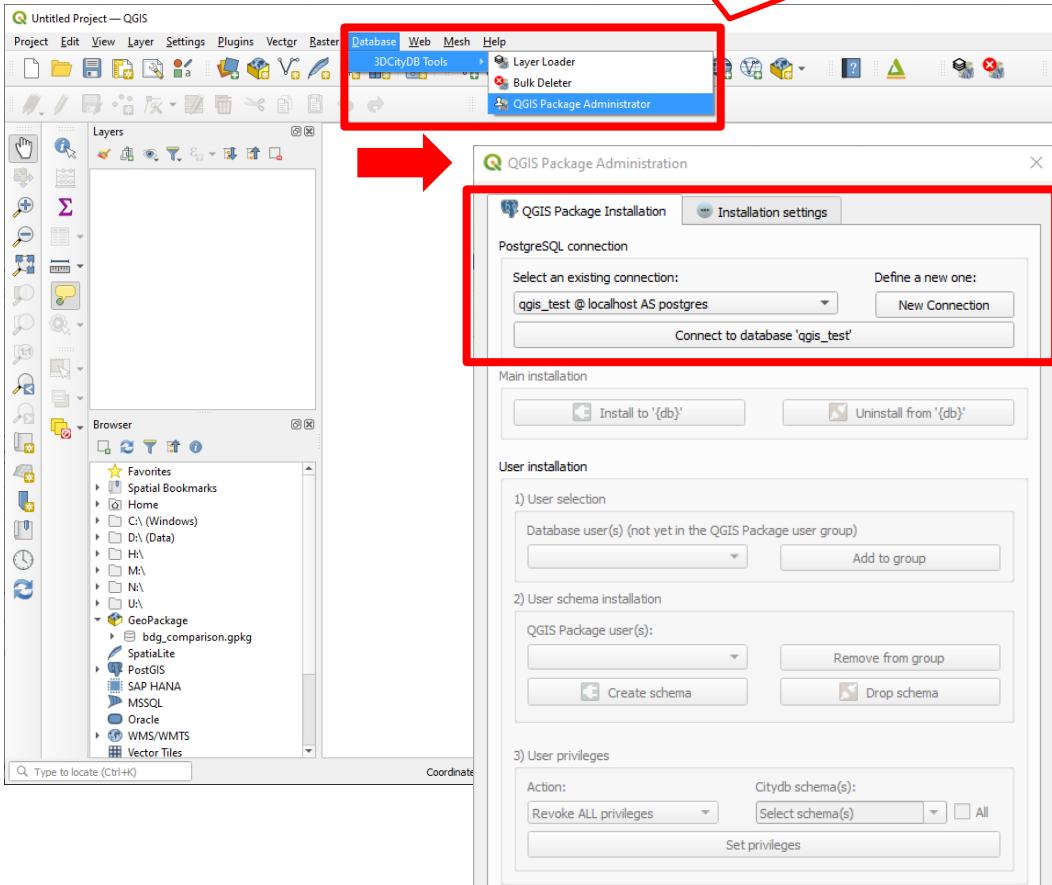
- A "**simplified installation**", which carries out all 4 operation at once and automatically installs 2 default users
- A "**normal installation**", where the administrator has complete control over each step

Back-end installation

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

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

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



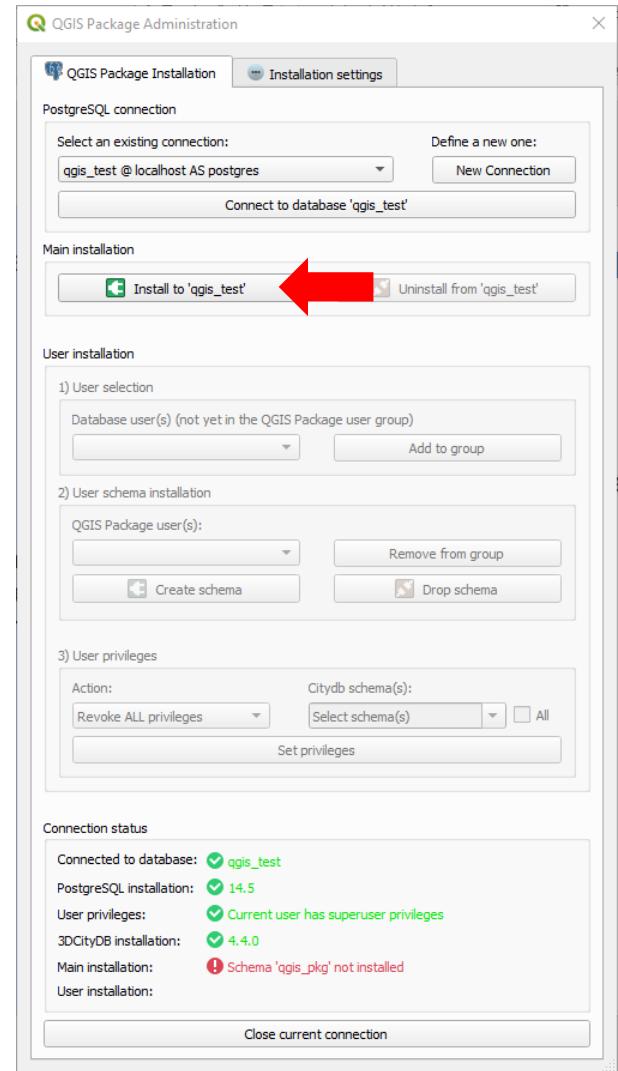
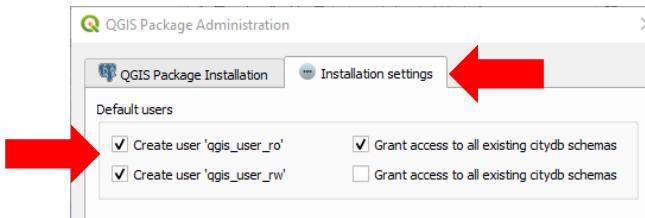
Back-end installation

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



The screenshot shows the 'QGIS Package Administration' dialog. In the 'Main installation' section, there are two buttons: 'Install to 'qgis_test'' (highlighted by a red arrow) and 'Uninstall from 'qgis_test''. Below it is the 'User installation' section with tabs for 'User selection', 'User schema installation', and 'User privileges'. At the bottom, the 'Connection status' section shows 'Connected to database: qgis_test' and other status indicators. A red arrow also points to the 'Installation settings' tab at the top right of the dialog.

Back-end installation

2.1) "Simplified" installation

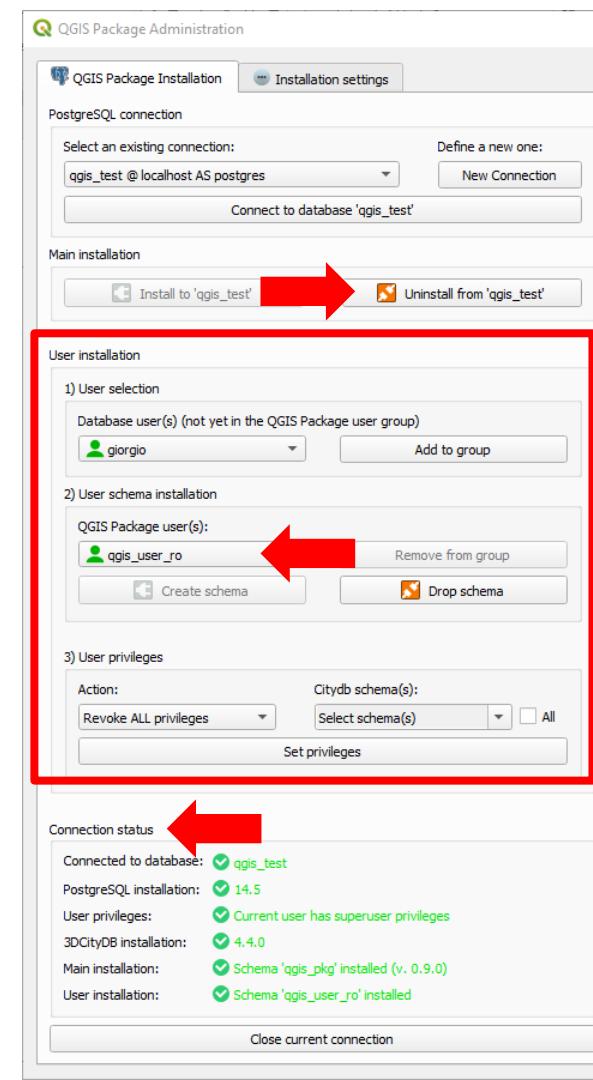
Upon successful installation:

- 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

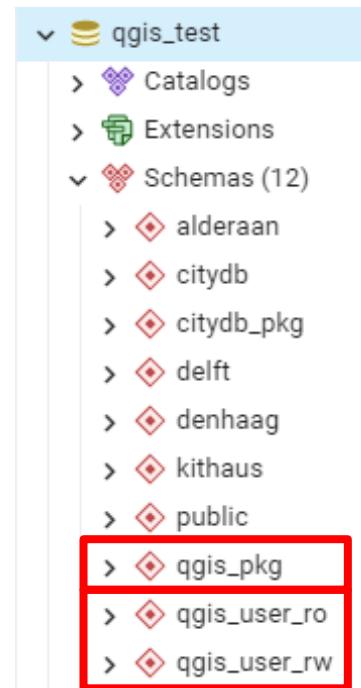
The credentials for the default users are:

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**



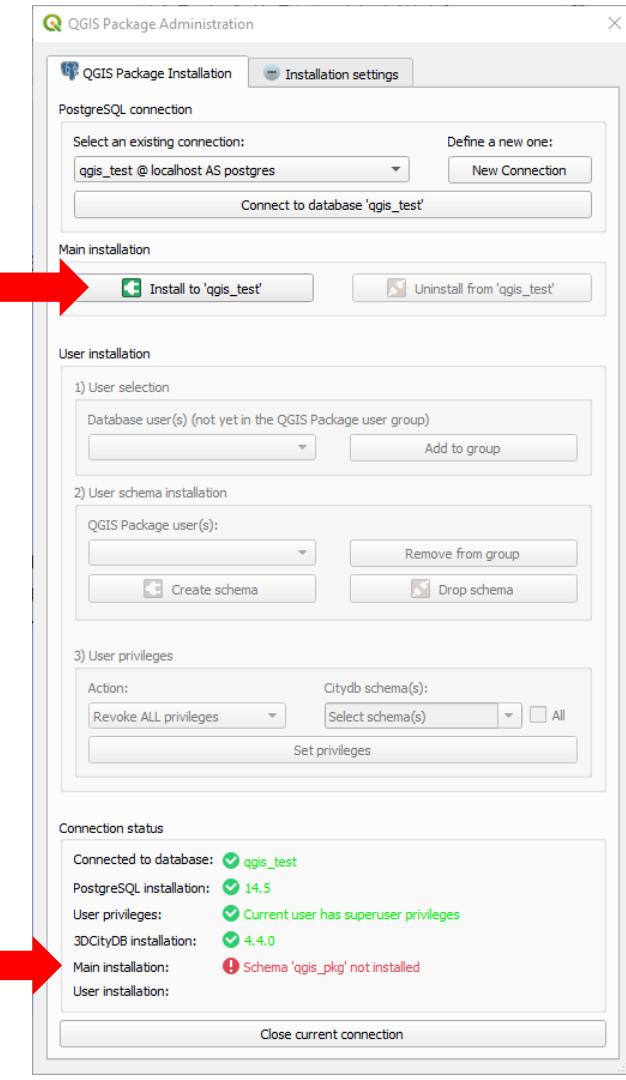
Back-end installation

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.



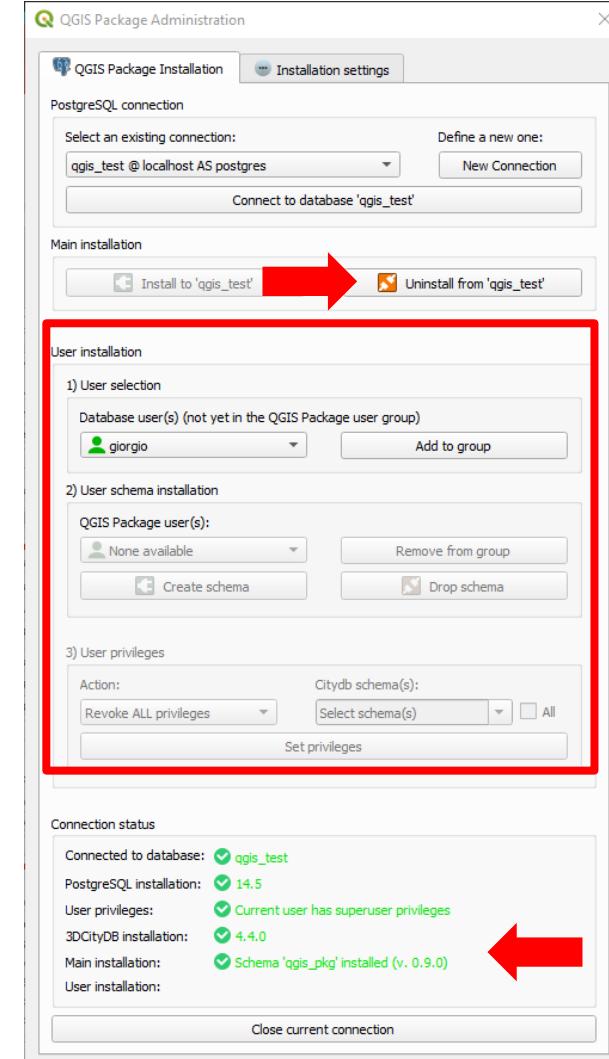
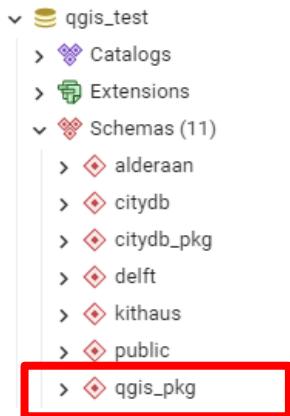
Back-end installation

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



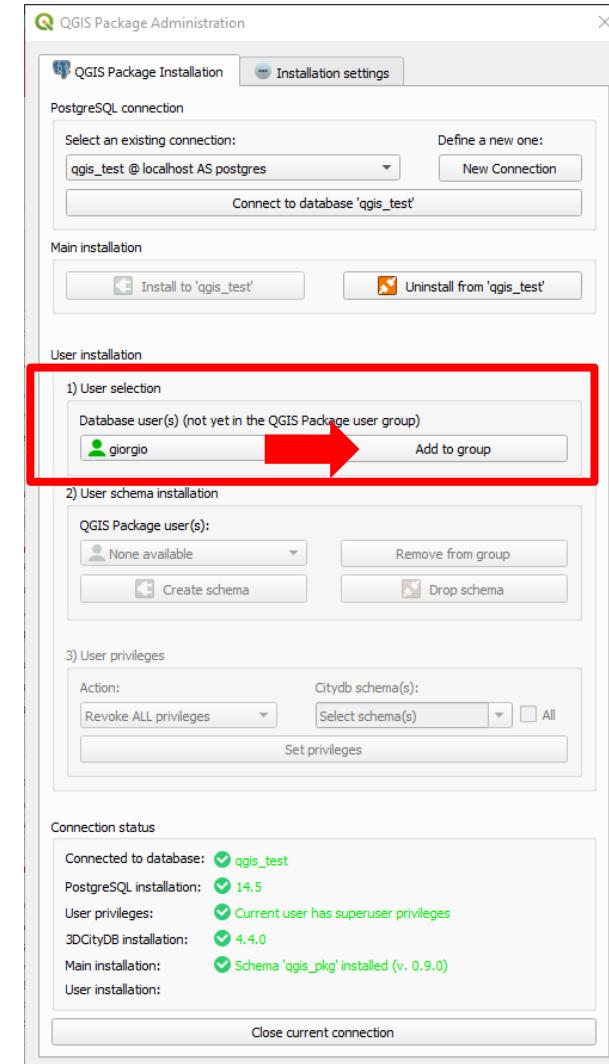
Back-end installation

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



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".



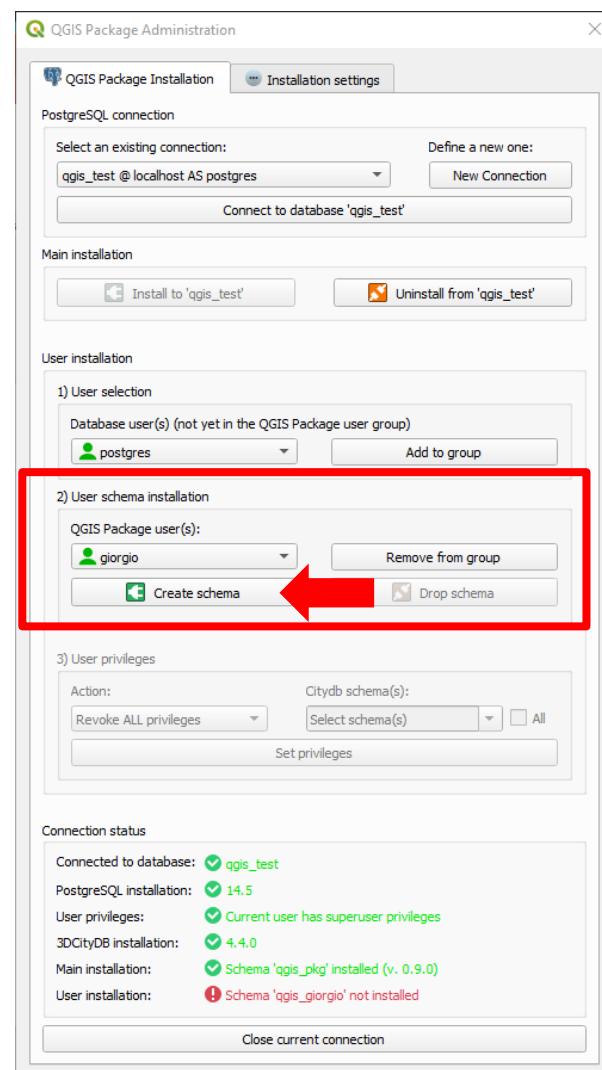
The screenshot shows the 'QGIS Package Administration' window. In the 'User selection' section, a user 'giorgio' is listed in a dropdown menu, with a red arrow pointing to the 'Add to group' button. The 'Main installation' and 'User installation' sections are also visible.

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
Changelog

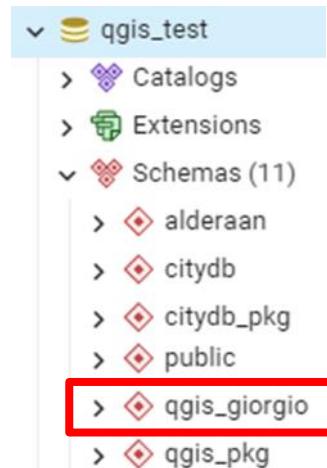


Back-end installation

2.2) "Normal" installation

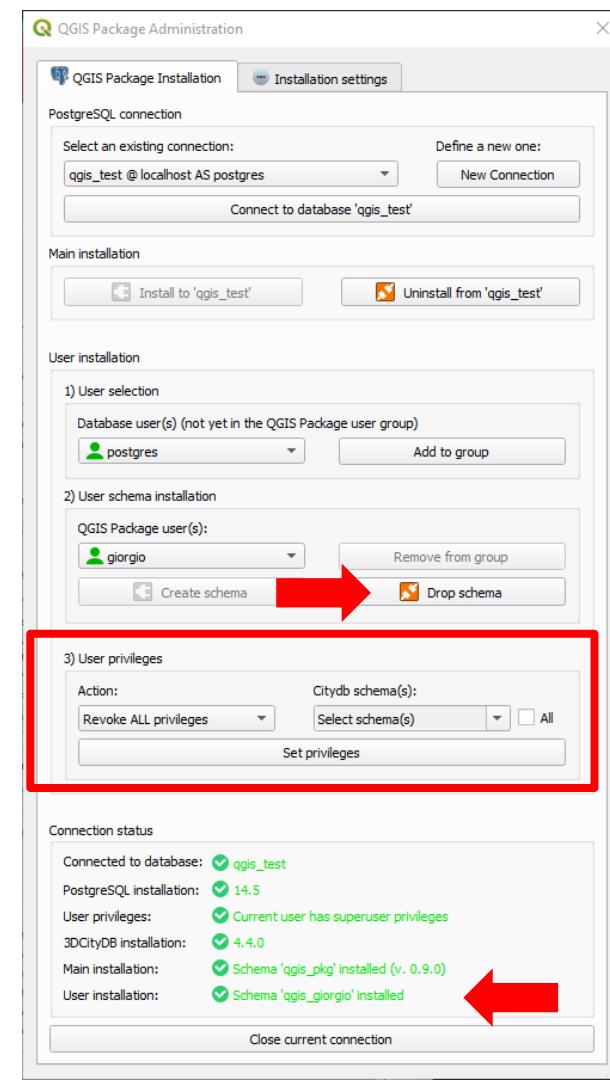
Upon successful creation of the user schema:

- 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

QGIS Package Installation Installation settings

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 Citydb schema(s): Select schema(s) All 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

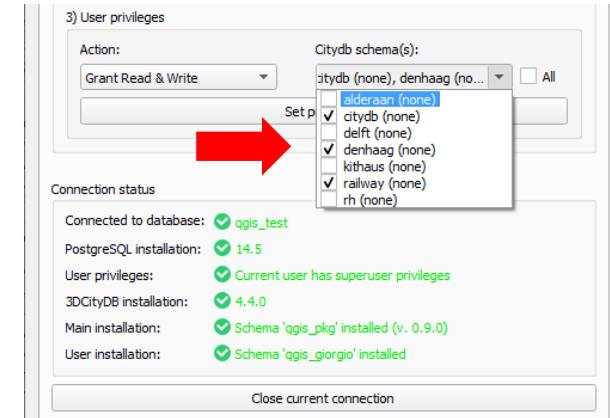
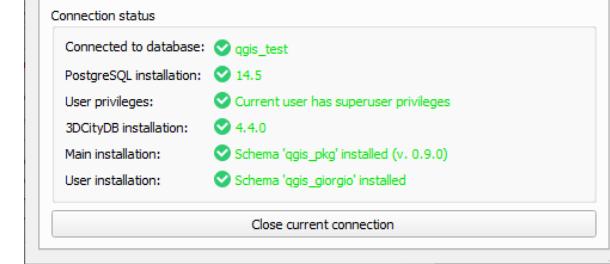
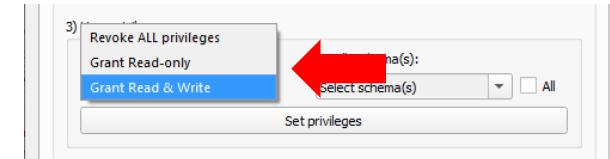
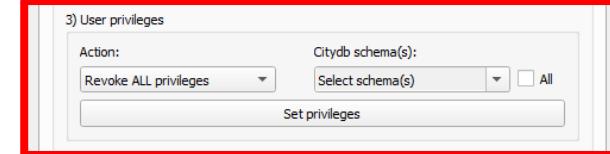
Back-end installation

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

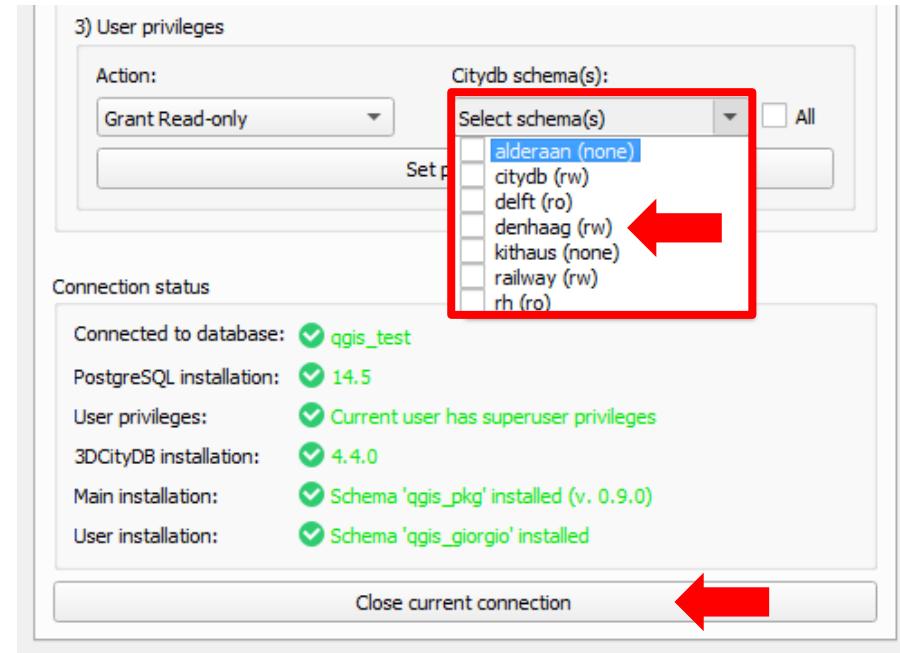
2.2) "Normal" installation

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

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

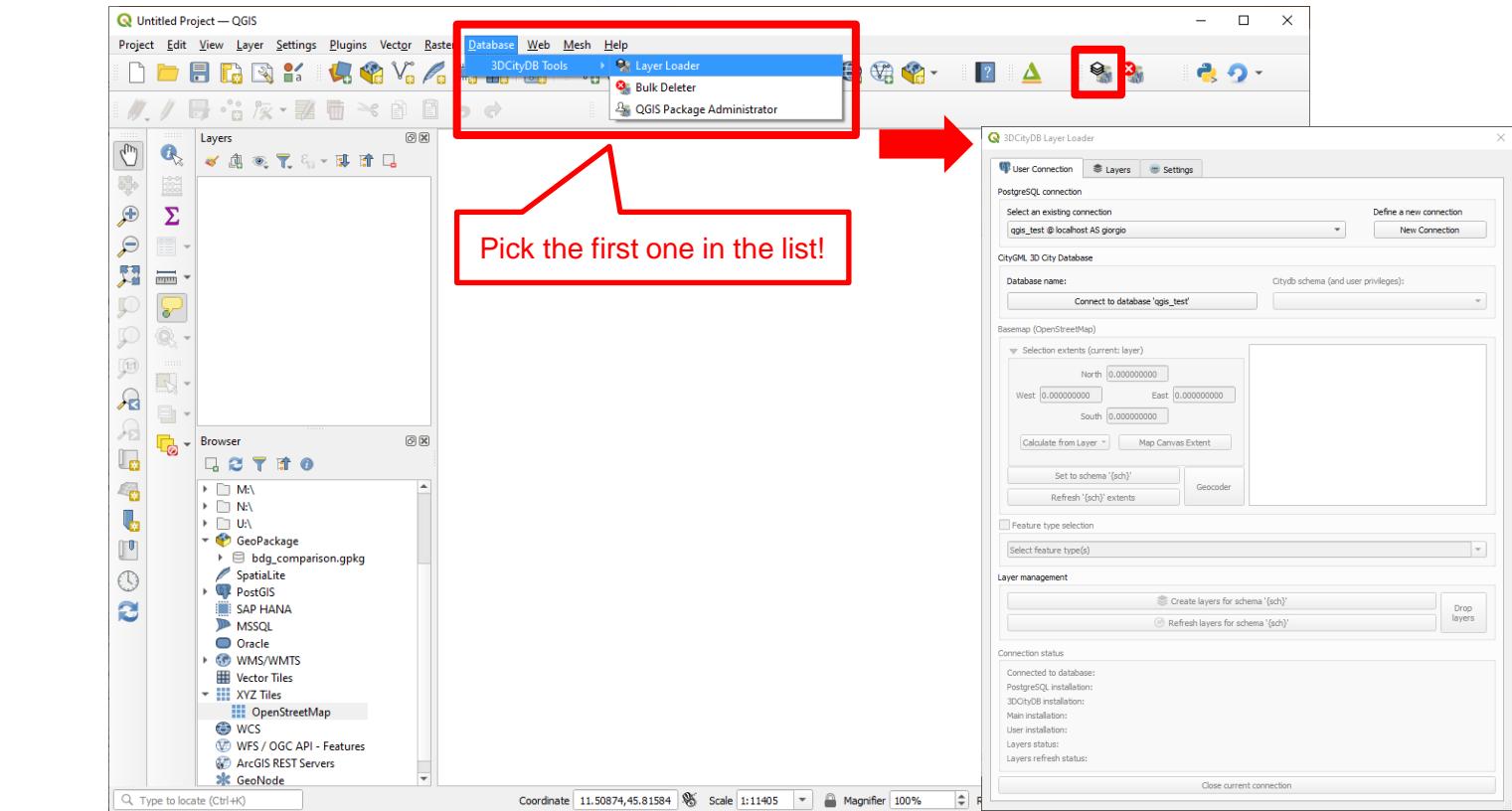
QGIS
Package

Resources

Layer Loader



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



The screenshot shows the QGIS application interface. The top menu bar is visible with options like Project, Edit, View, Layer, Settings, Plugins, Vector, Raster, Database, Web, Mesh, and Help. A red box highlights the 'Database' menu item, which is currently selected. Below the menu is a toolbar with various icons. To the right of the toolbar, a red box highlights the '3DCityDB Layer Loader' icon, which is also highlighted in the status bar. A large red arrow points from the 'Layer Loader' menu item towards the open plugin window. Another red box highlights the text 'Pick the first one in the list!' inside the plugin window. The main QGIS interface shows the 'Layers' panel on the left and the 'Browser' panel below it, both displaying a list of available data sources.

Pick the first one in the list!

QGIS Package Administrator

3DCityDB Layer Loader

User Connection Layers Settings

Select an existing connection qgis_test @ localhost AS giorgio Define a new connection New Connection

CityGML 3D City Database Database name: Connect to database 'qgis_test'

Basemap (OpenStreetMap)

Selection extents (current: layer)

North: 0.0000000000 East: 0.0000000000

West: 0.0000000000 South: 0.0000000000

Calculate from Layer Map Canvas Extent

Set to schema '(sch)' Refresh '(sch)' extents Geocoder

Feature type selection Select feature type(s)

Layer management Create layers for schema '(sch)' Refresh layers for schema '(sch)' Drop layers

Connection status

Connected to database: PostgreSQL installation: 3DCityDB installation: Main installation: User installation: Layers status: Layers refresh status:

Type to locate (Ctrl+K)

Coordinate 11.50874, 45.81584 Scale 1:11405 Magnifier 100%

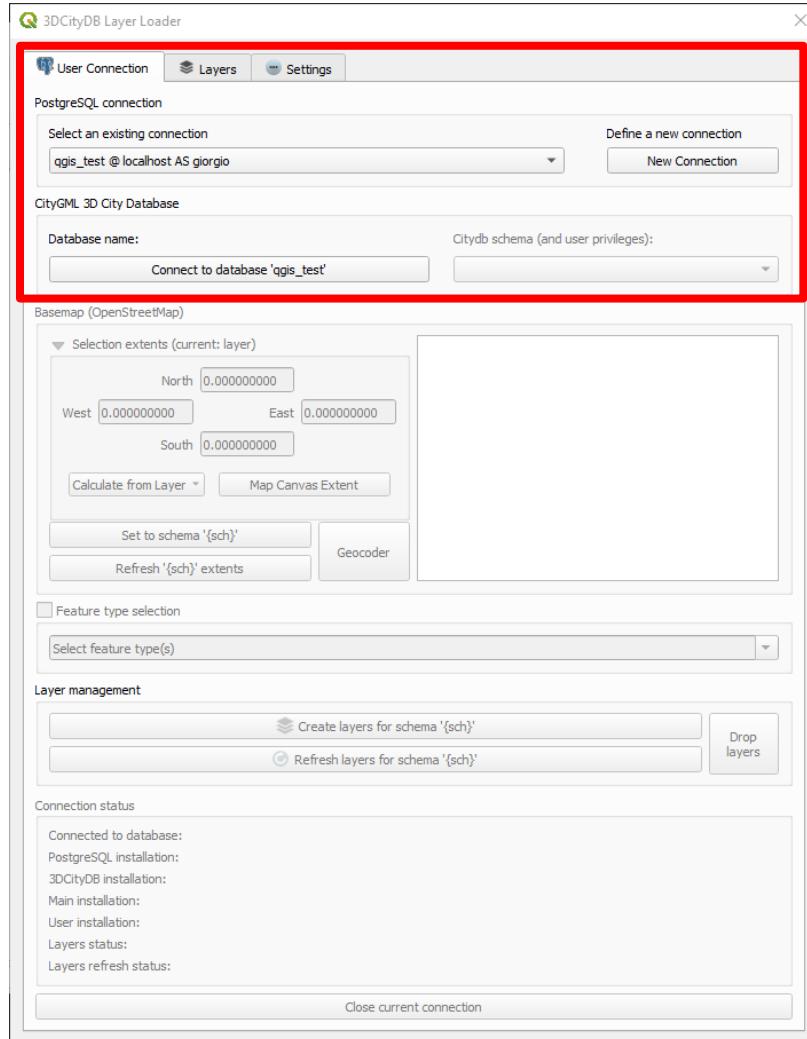
Layer Loader

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)



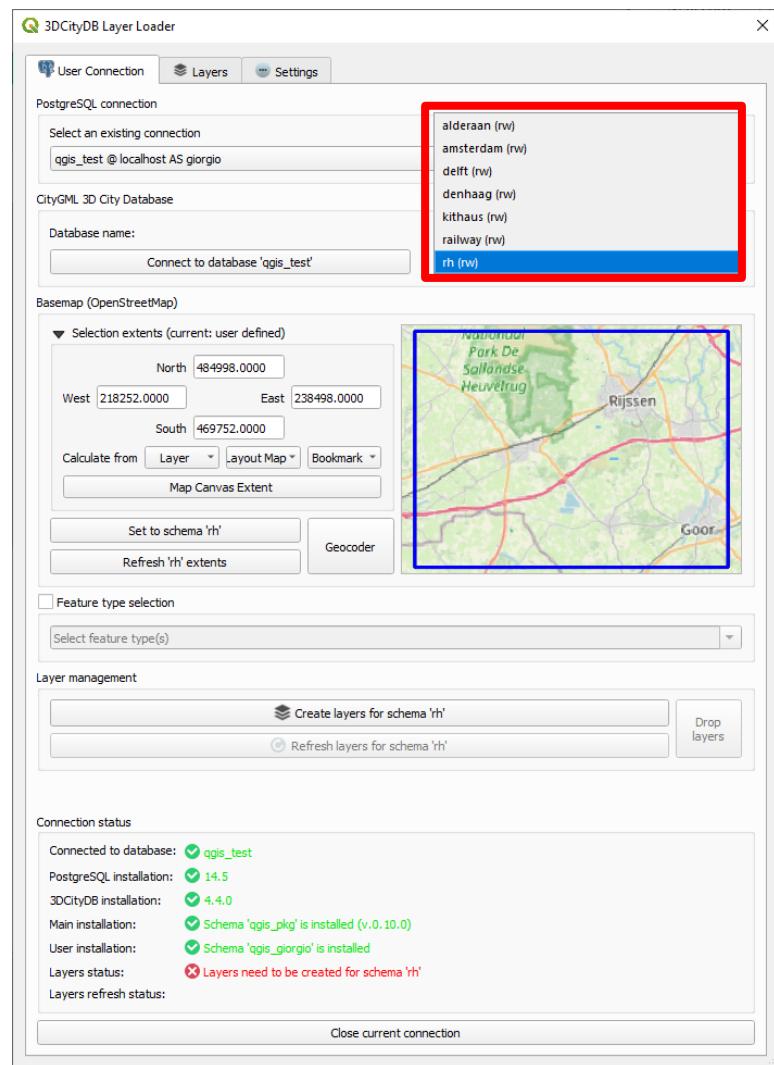
Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

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

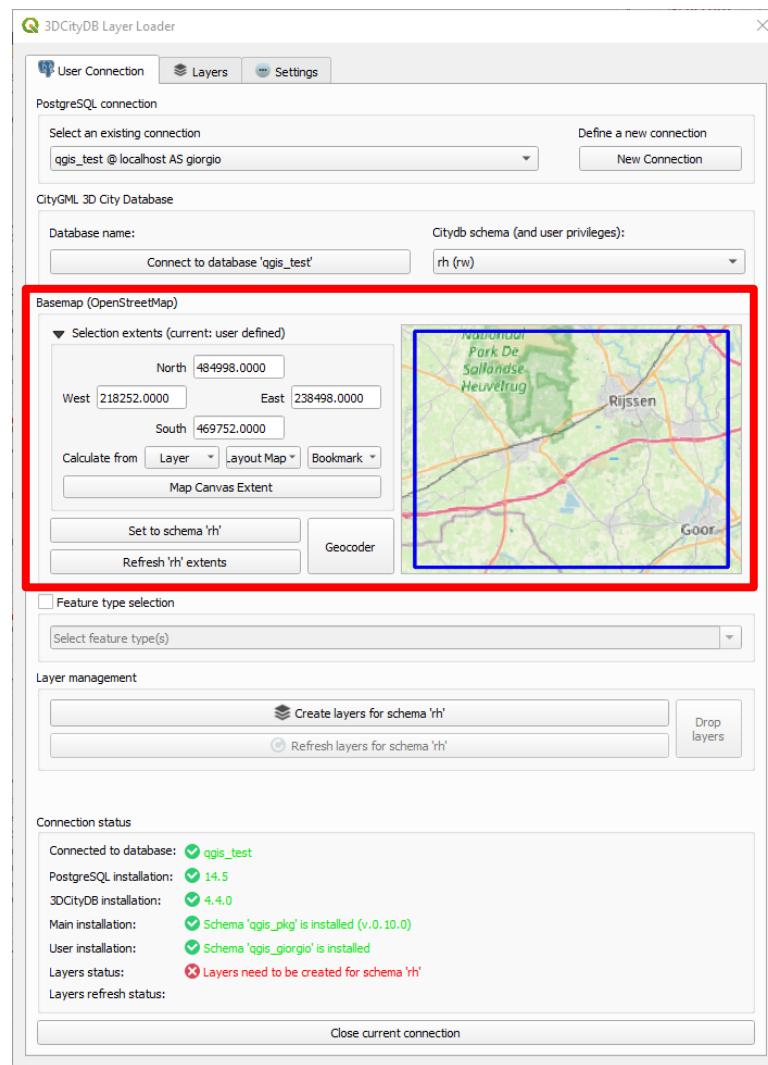
How to create additional citydb schemas
<https://3dcitydb-docs.readthedocs.io/en/latest/3dcitydb/multi-schema.html>



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.

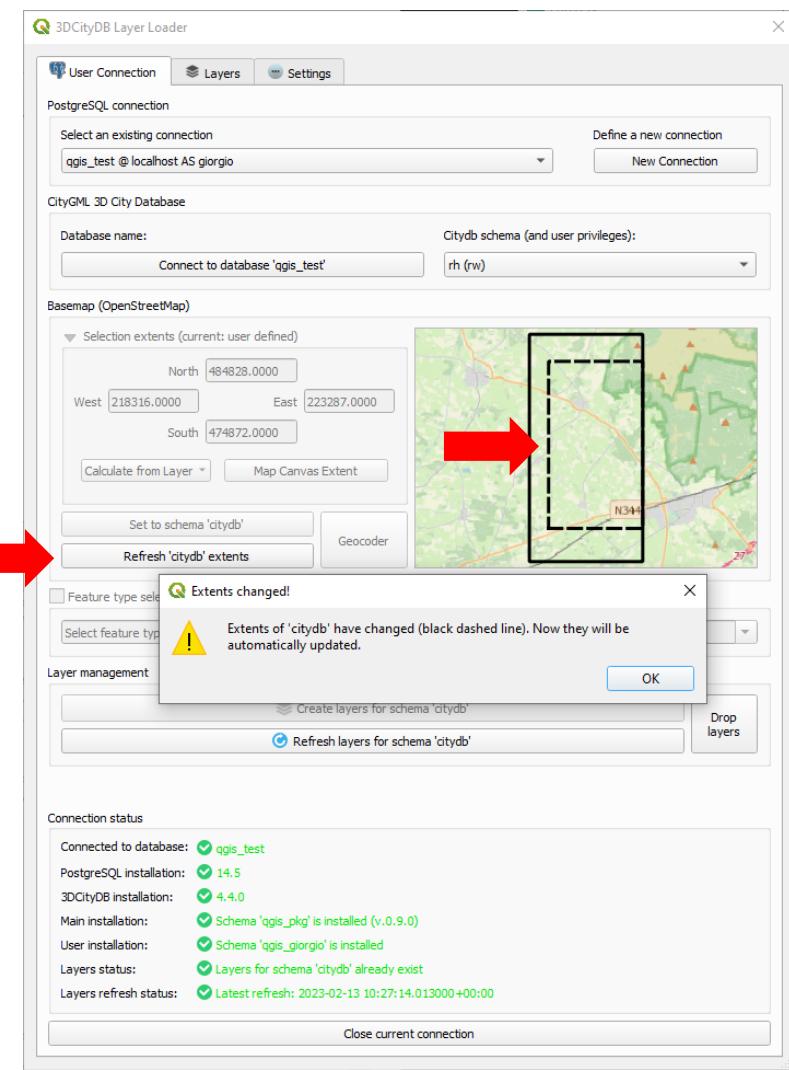


Layer Loader

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)



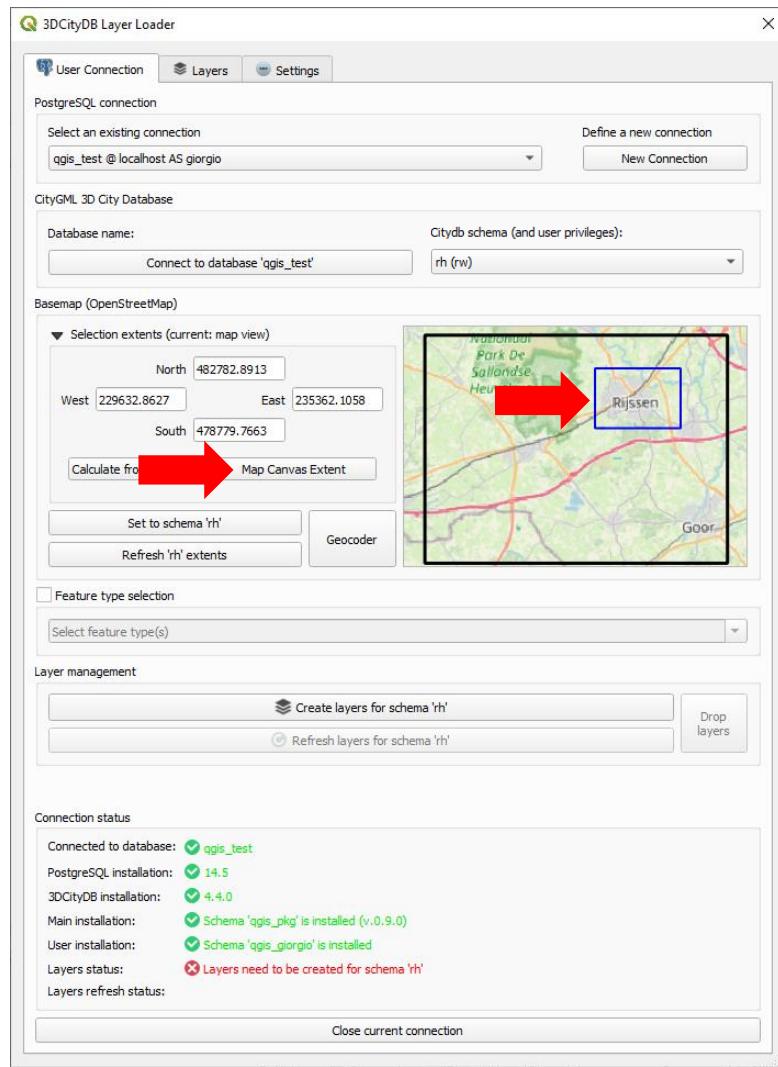
Layer Loader

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)!



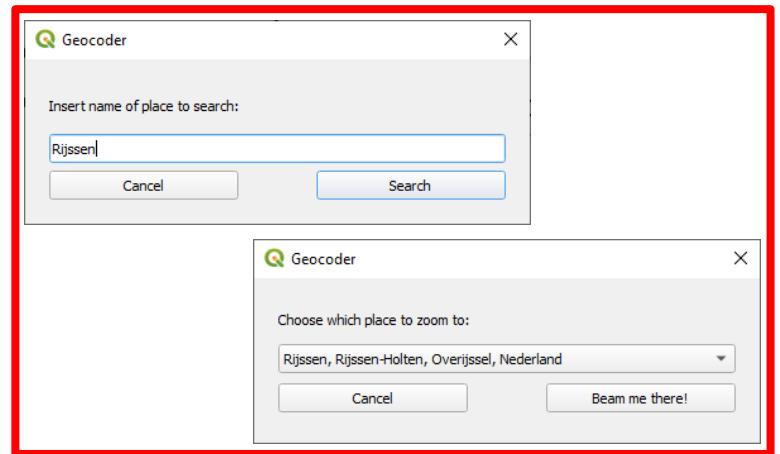
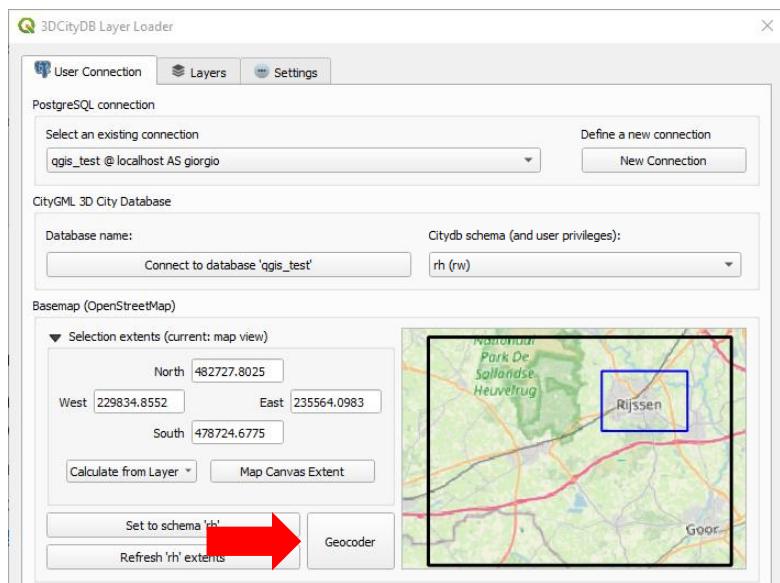
Layer Loader

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

Advanced options

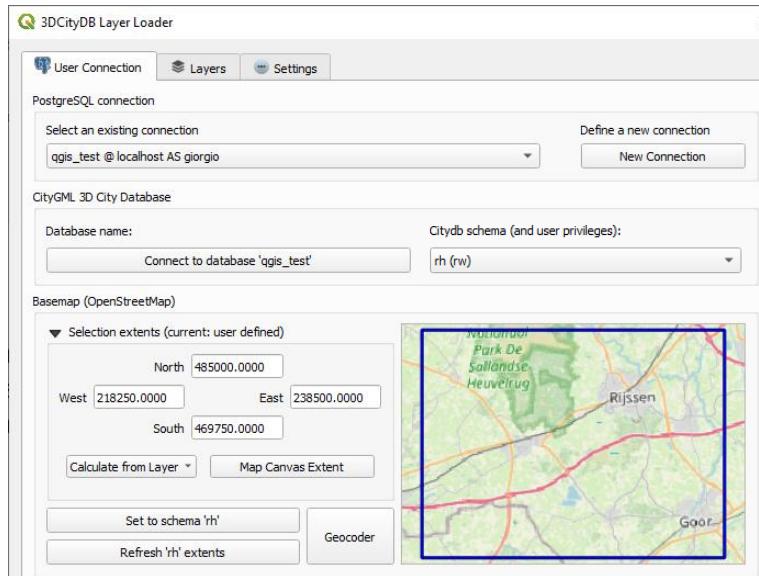
Software uninstall

Current limitations

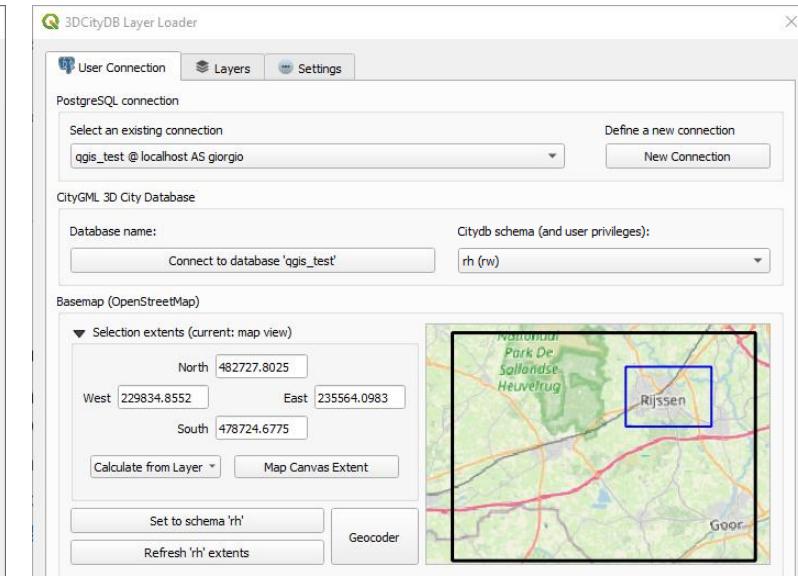
QGIS Package

Resources

Changelog



Both areas coincide (default)



User-selected layers extents

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

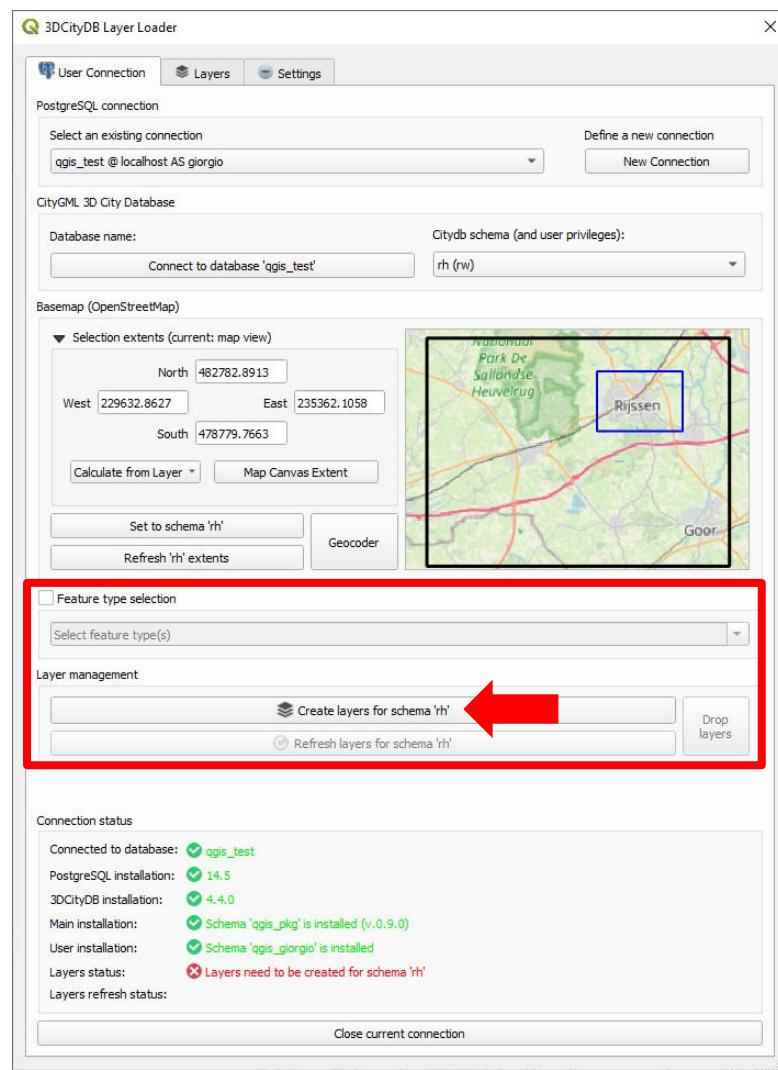
Layer Loader

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
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

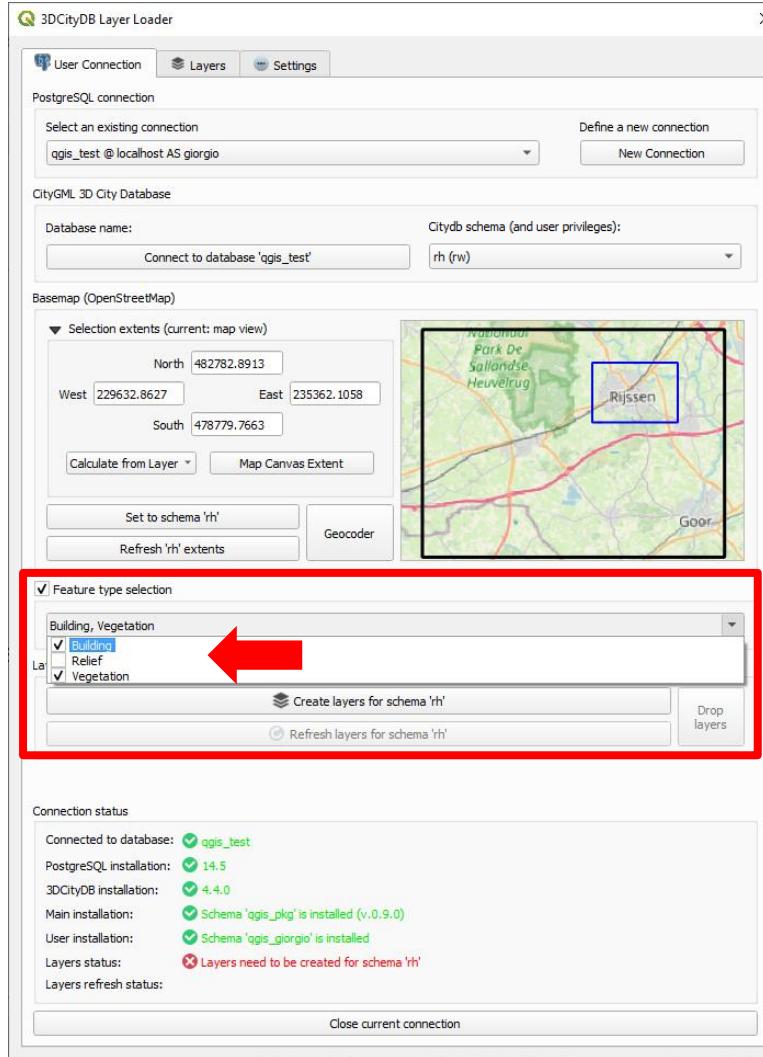
Layer Loader

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.)

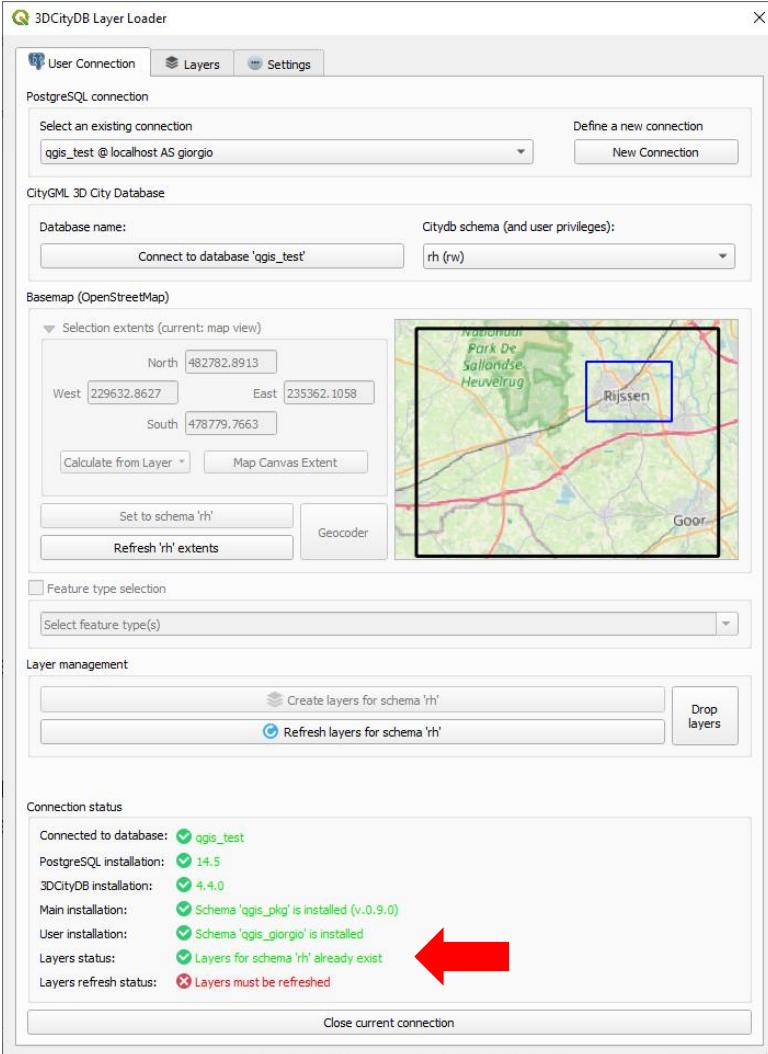


Layer Loader

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

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 set to "rh (rw)".

Under "Basemap (OpenStreetMap)", there is a map view showing a region in the Netherlands with a blue rectangle highlighting a specific area. Below the map are buttons for "Selection extents (current: map view)", "Calculate from Layer", "Map Canvas Extent", "Set to schema 'rh'", "Refresh 'rh' extents", and "Geocoder".

Under "Feature type selection", there is a dropdown menu labeled "Select feature type(s)".

Under "Layer management", there are buttons for "Create layers for schema 'rh'" and "Refresh layers for schema 'rh'".

At the bottom, the "Connection status" section shows the following status items:

- Connected to database: ✓ qgis_test
- PostgreSQL installation: ✓ 14.5
- 3DCityDB installation: ✓ 4.4.0
- Main installation: ✓ Schema 'qgis_pk' is installed (v.0.9.0)
- User installation: ✓ Schema 'qgis_giorgio' is installed
- Layers status: ✓ Layers for schema 'rh' already exist
- Layers refresh status: ✗ Layers must be refreshed

A large red arrow points to the "Layers refresh status" item.

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

Layer Loader

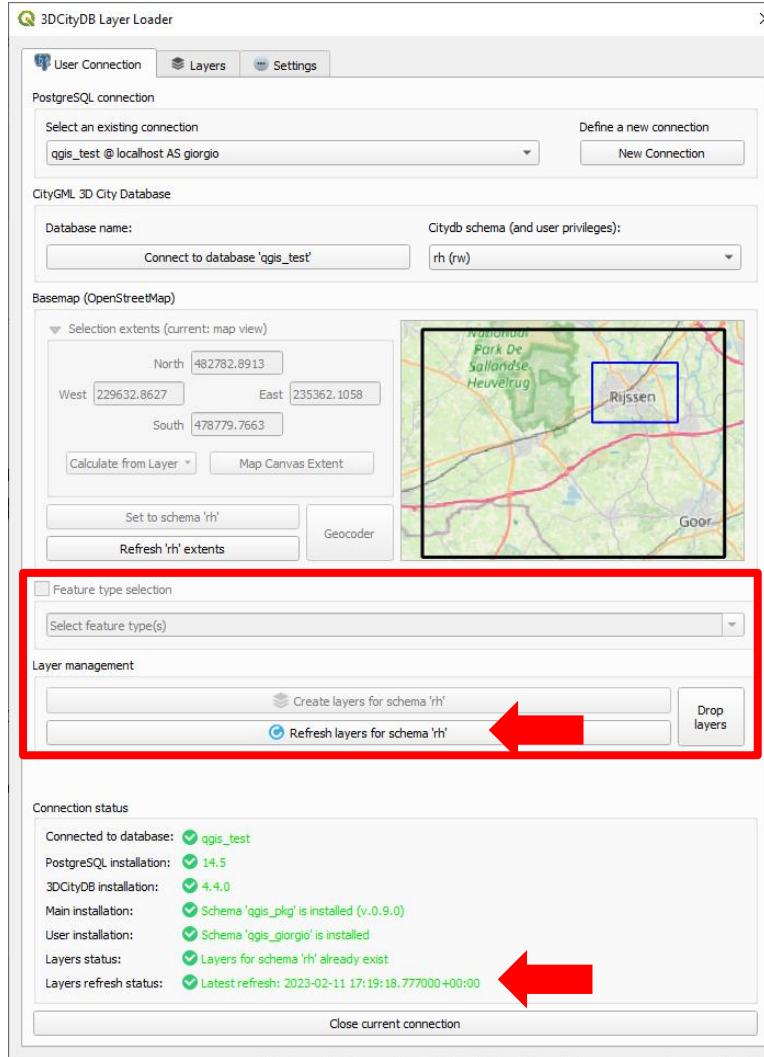
7) Populate/refresh the layers

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
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources
- Changelog

Behind the scenes:

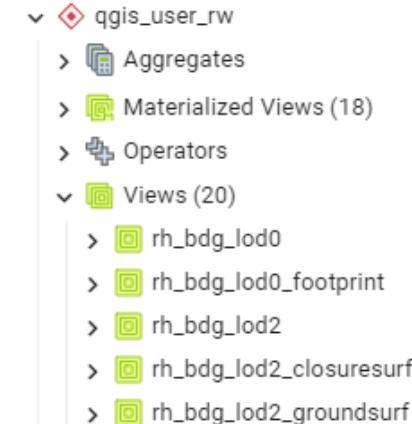
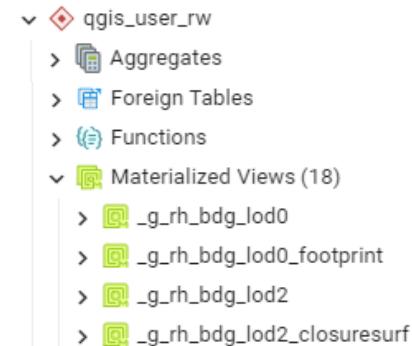
In the database user schema, both materialised views (containing the feature geometries) and the layers (as updatable views, containing the feature attributes linked to the corresponding geometries in the materialised views) can be accessed.

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)

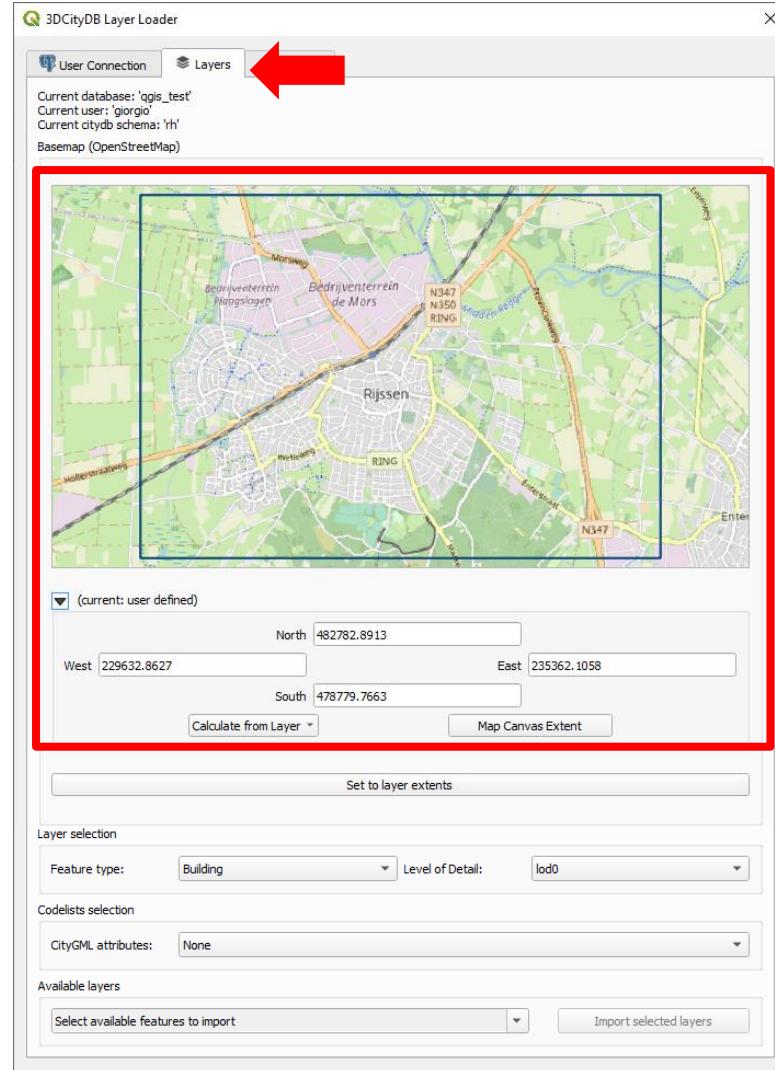


Layer Loader

In the "Layers" tab

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

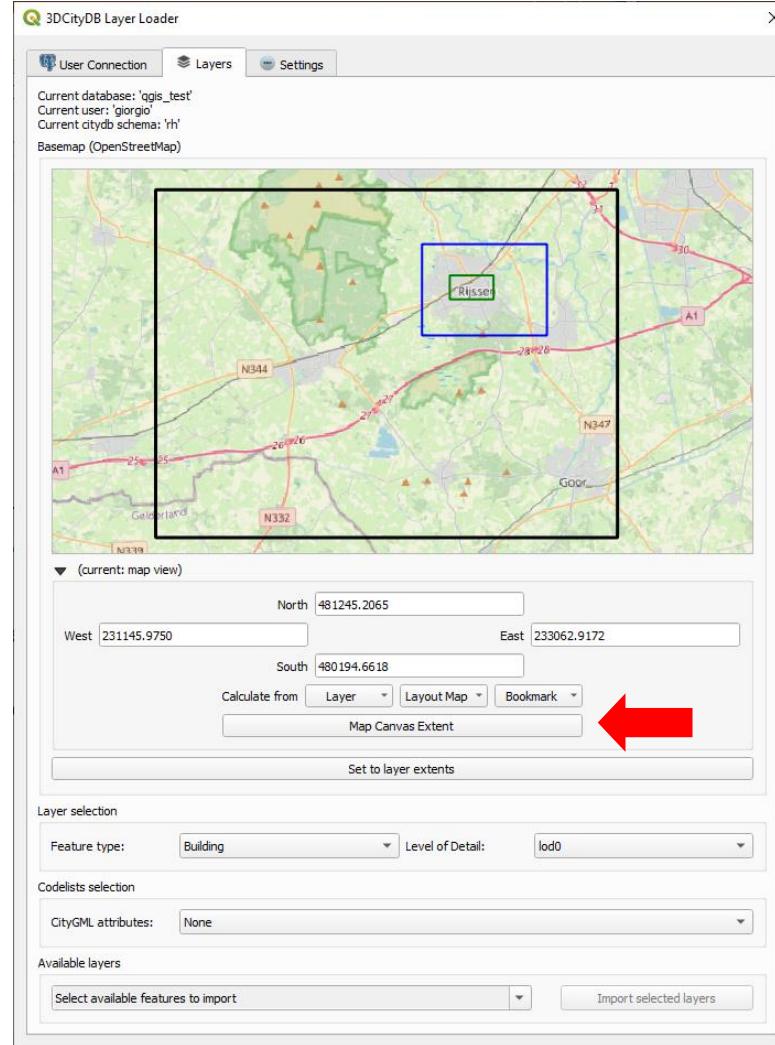
8) You are now directly zoomed to the layers extents (**blue bounding box**)



Layer Loader

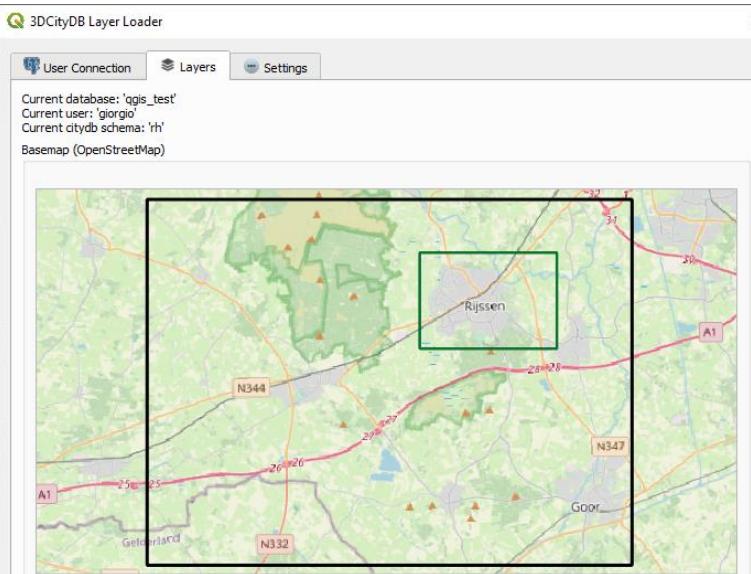
9) Depending on your needs, you can further reduce the extents of the layers to be loaded into QGIS by pressing the **Map Canvas Extent button**. The new extents are represented by the **green bounding box**.

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

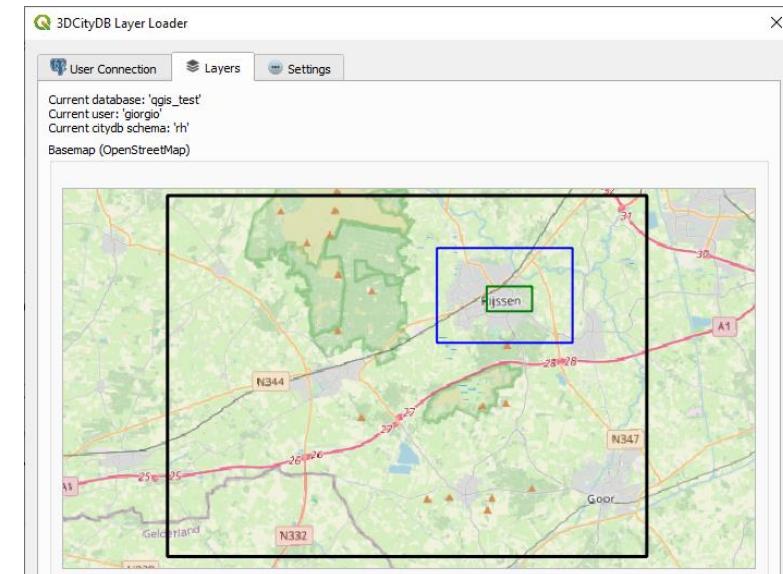


Layer Loader

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog



Database- and QGIS-side extents coincide (default)



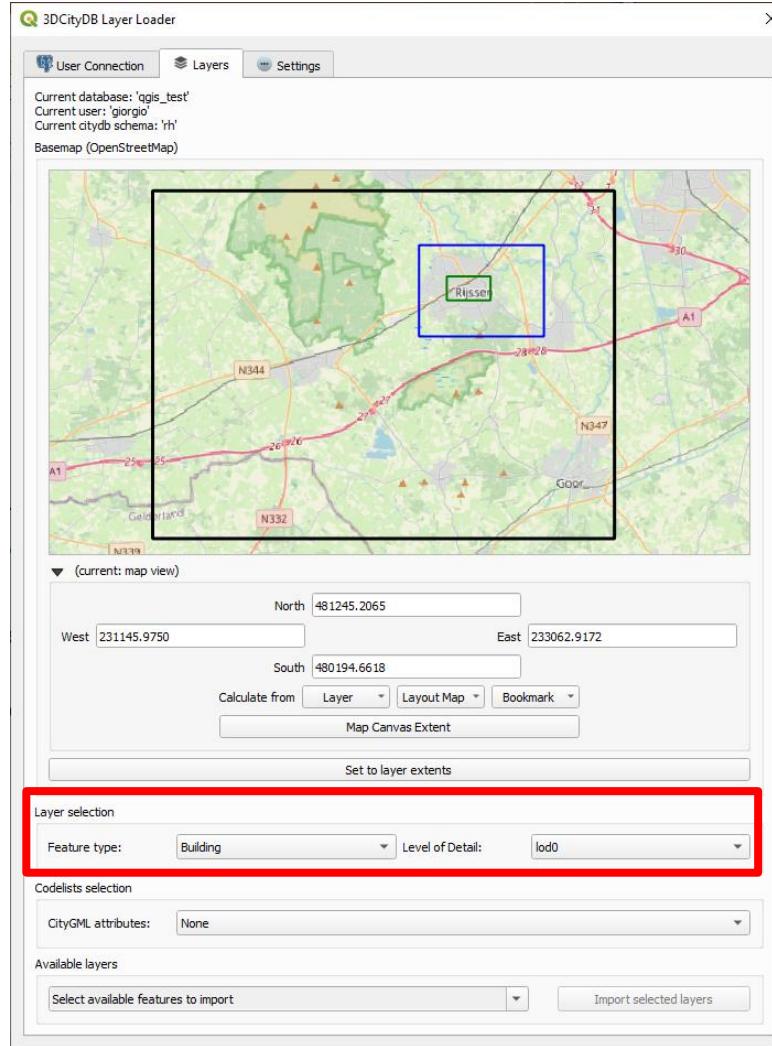
User-selected QGIS-layers extents

Layer Loader

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

- 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**).



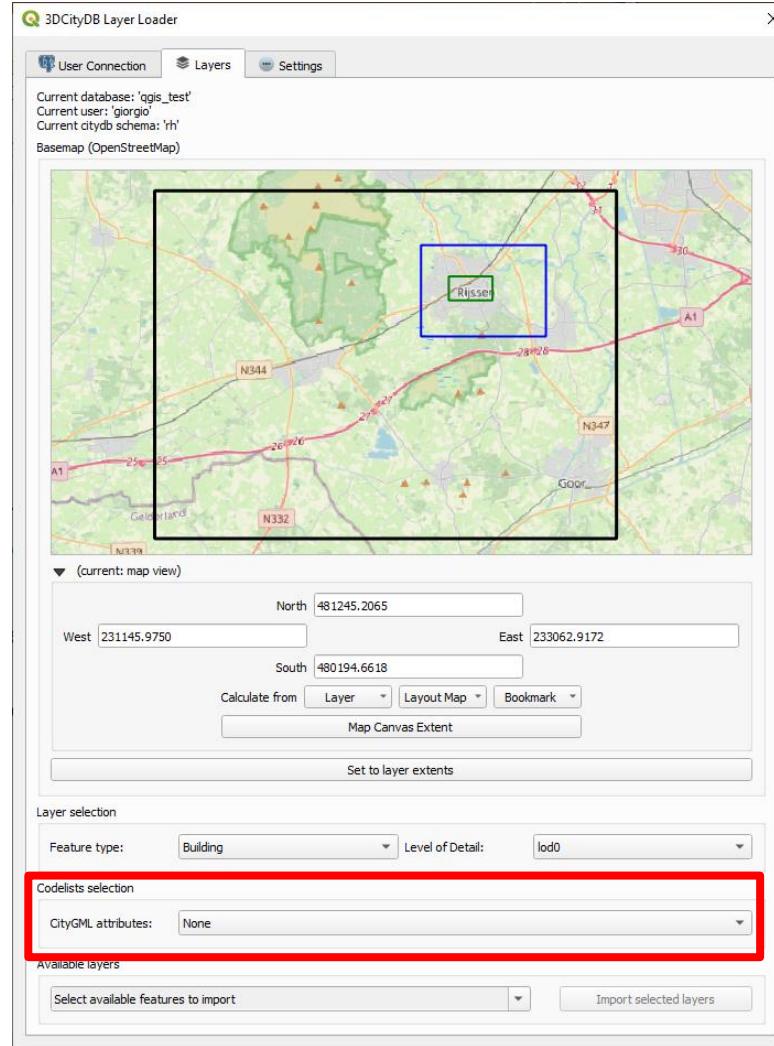
Motivation
Plugin overview
Software installation
Software use

- Layer Loader
- Bulk Deleter

Advanced options
Software uninstall
Current limitations
QGIS Package
Resources
Changelog

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)

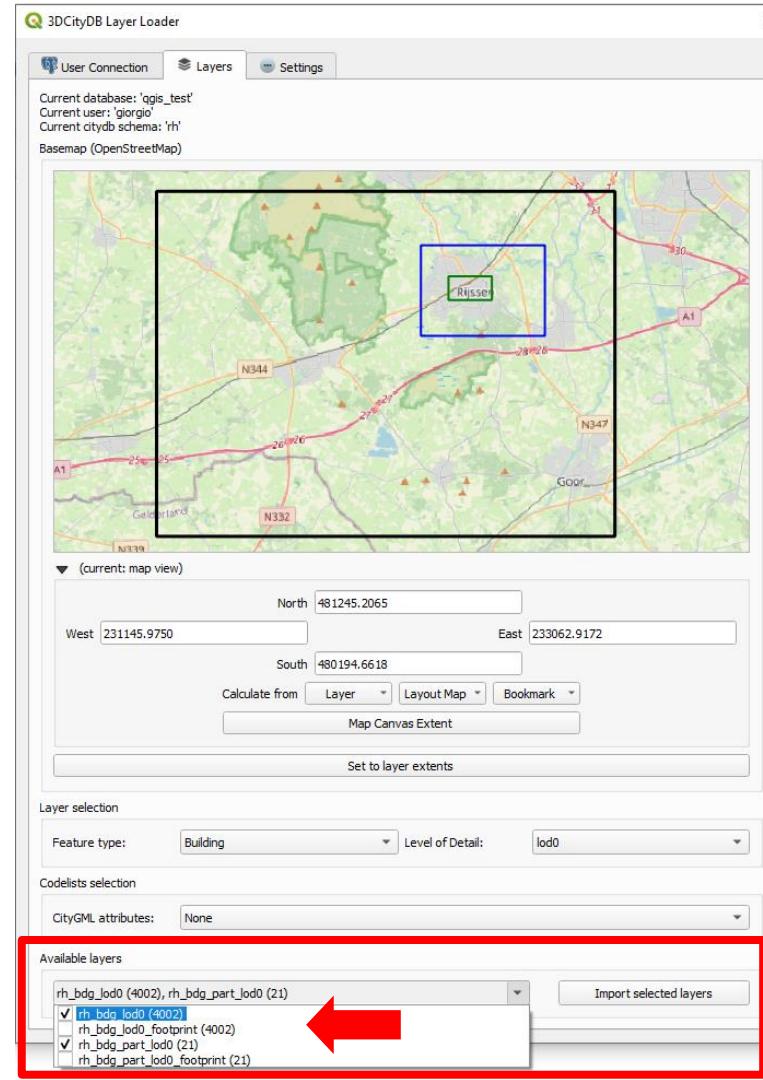


Layer Loader

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**).

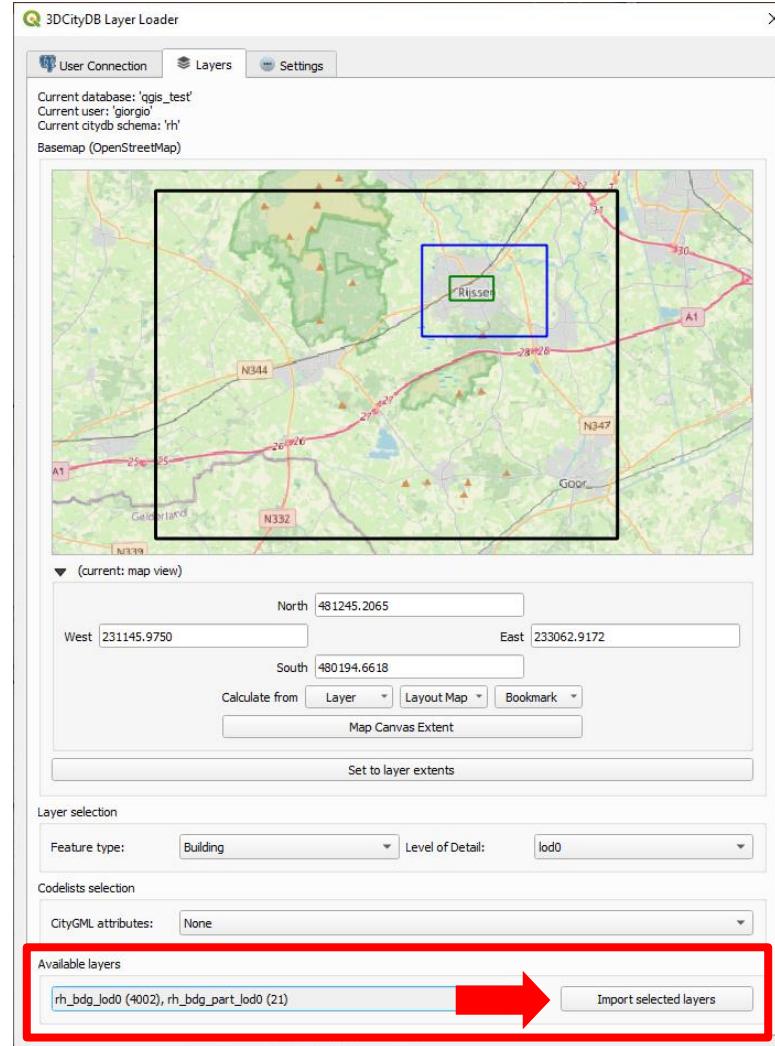


Layer Loader

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)

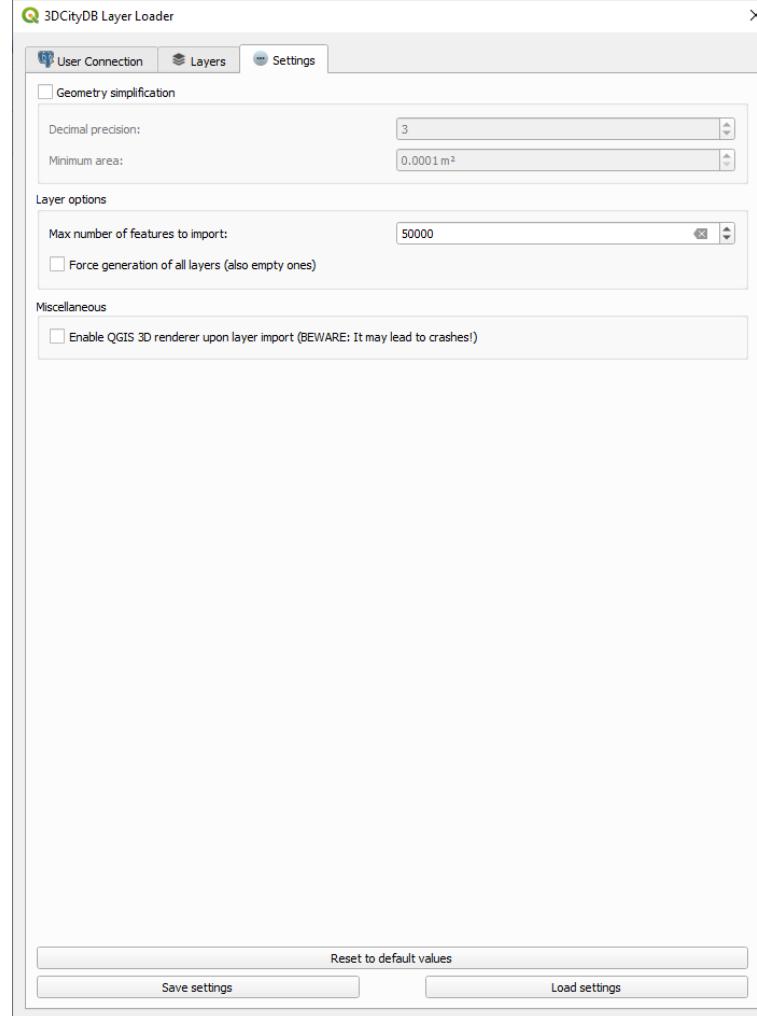
Nota bene: Layers that have already been loaded can be selected, but won't be loaded again



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.



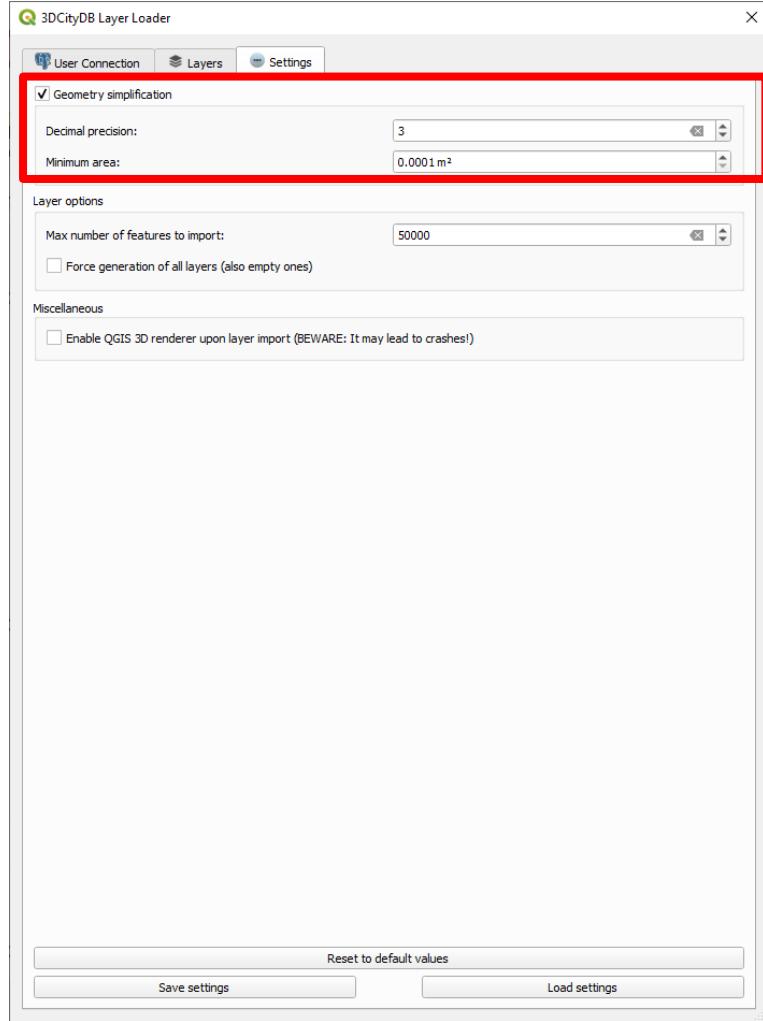
Geometry simplification

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:

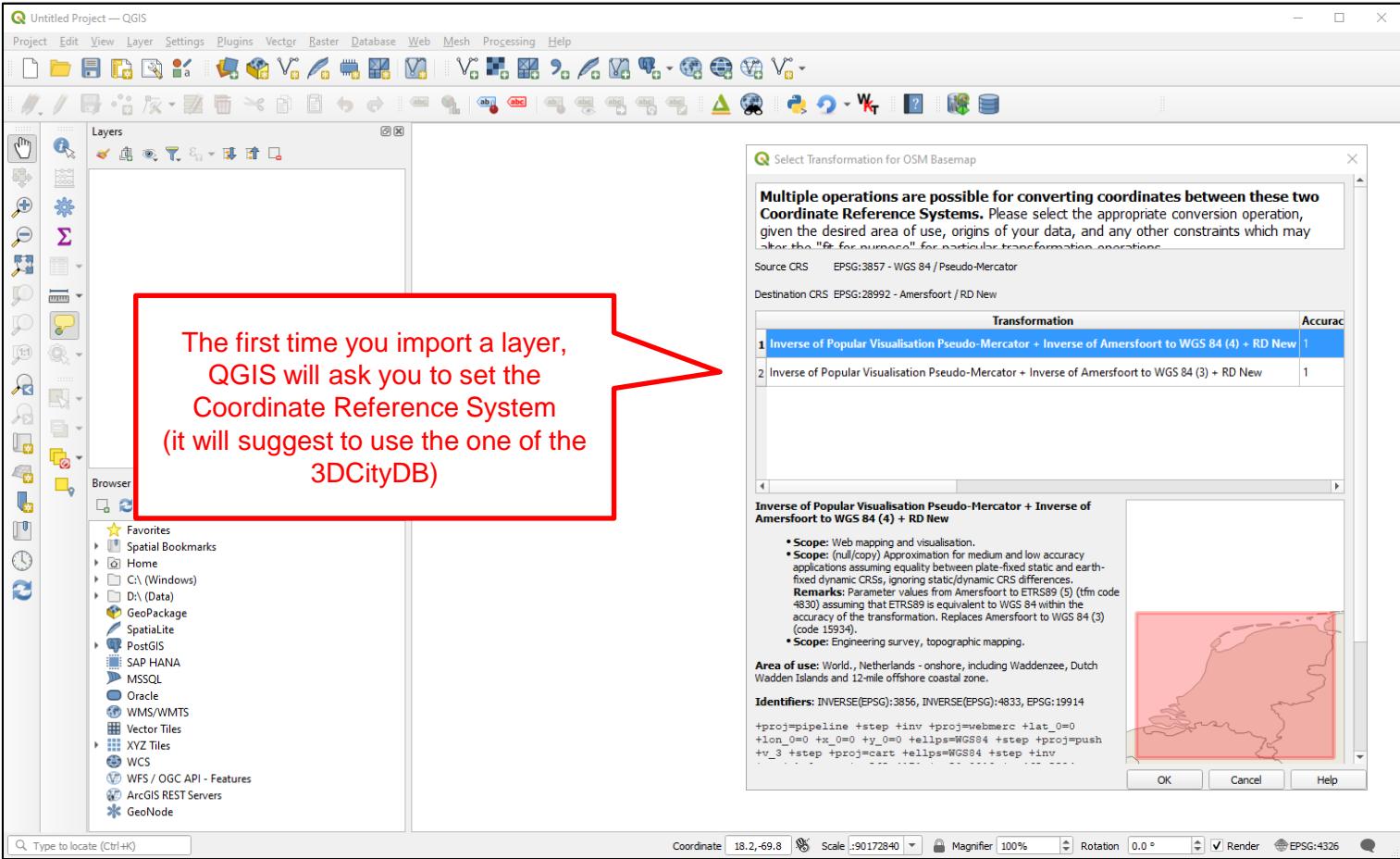
- can significantly increase the time needed to refresh the layers
- does NOT change the original data in the database!



Use in QGIS: Layers

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

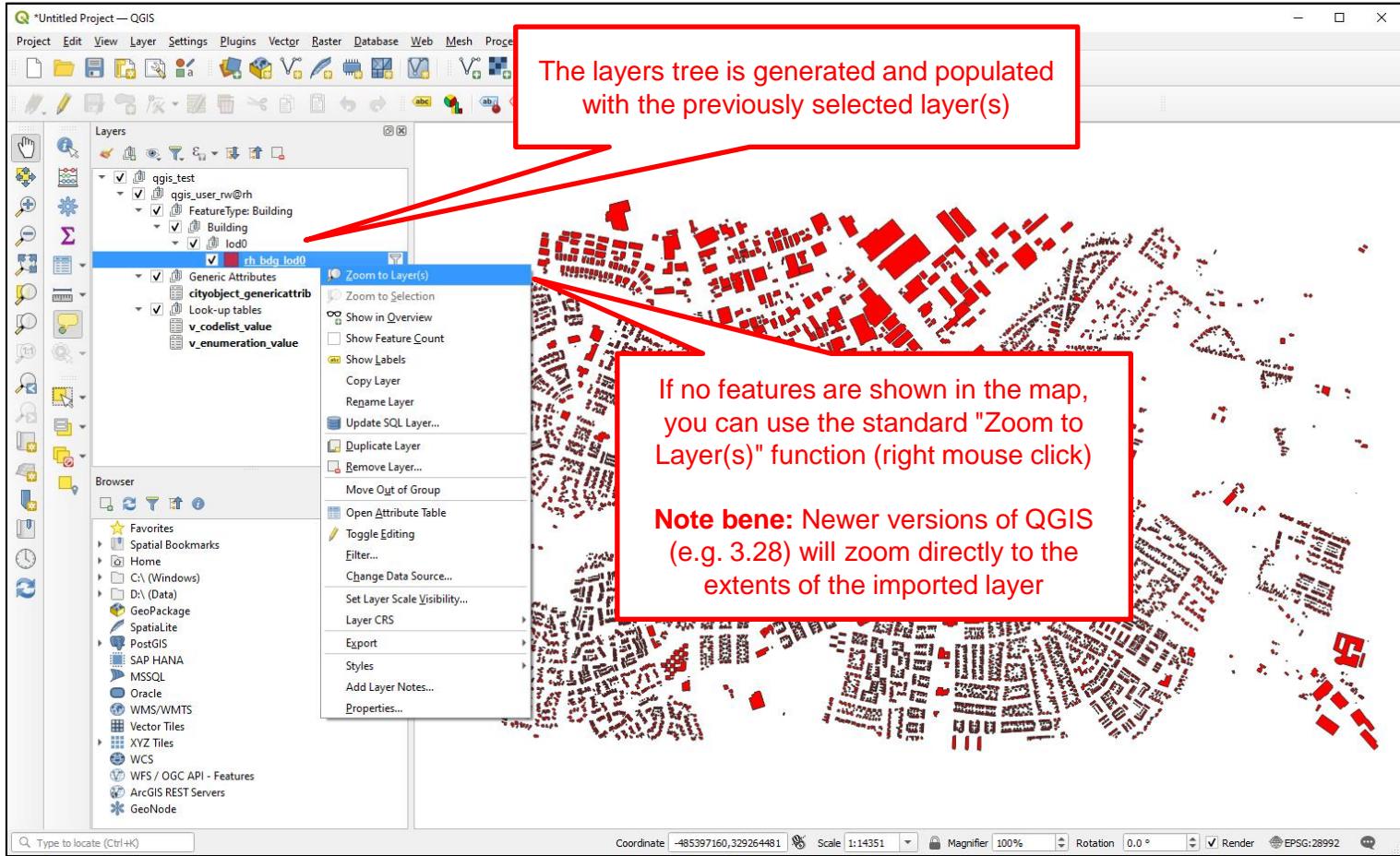
The first time you import a layer, QGIS will ask you to set the Coordinate Reference System (it will suggest to use the one of the 3DCityDB)



The screenshot shows the QGIS interface with the 'Layers' panel open on the left. A red box highlights a message in the center of the screen: "The first time you import a layer, QGIS will ask you to set the Coordinate Reference System (it will suggest to use the one of the 3DCityDB)". To the right, a 'Select Transformation' dialog box is displayed, showing two transformation operations listed under 'Transformation': "Inverse of Popular Visualisation Pseudo-Mercator + Inverse of Amersfoort to WGS 84 (4) + RD New" and "Inverse of Popular Visualisation Pseudo-Mercator + Inverse of Amersfoort to WGS 84 (3) + RD New". The first operation is selected. The dialog also contains detailed information about the transformation, including its scope, remarks, area of use, identifiers, and a preview map.

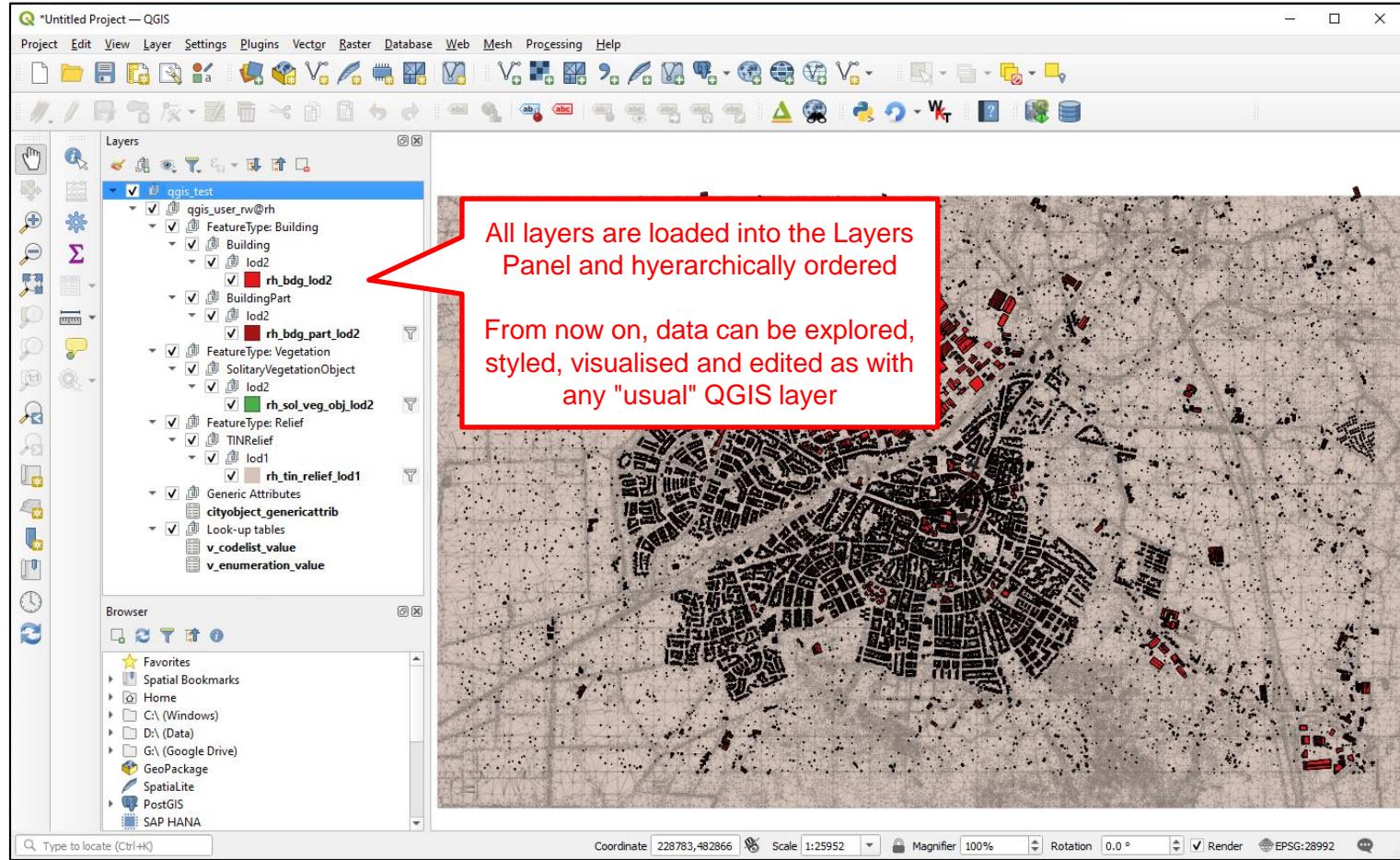
Use in QGIS: Layers

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



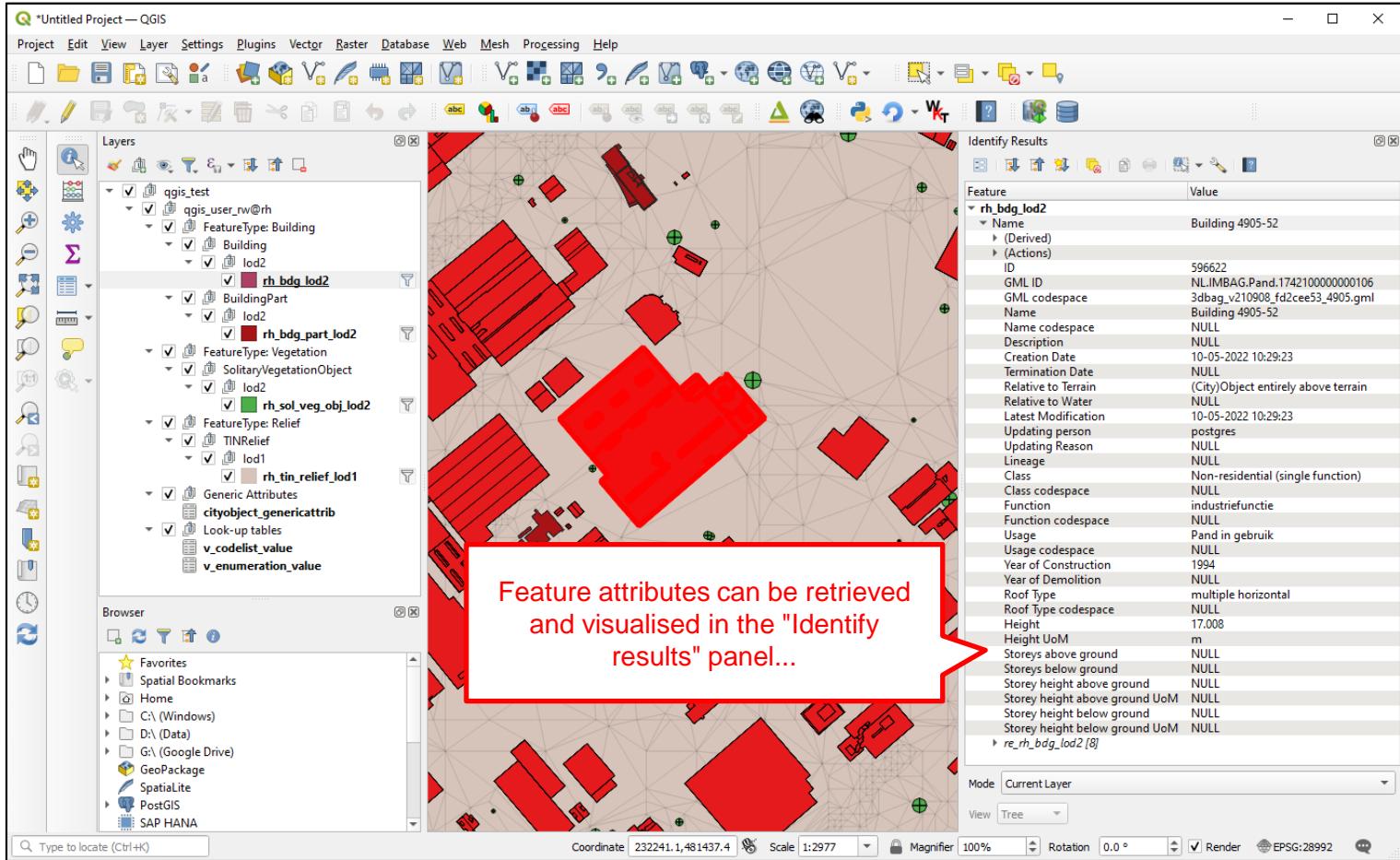
Use in QGIS: Layers

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



Use in QGIS: Attributes

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog



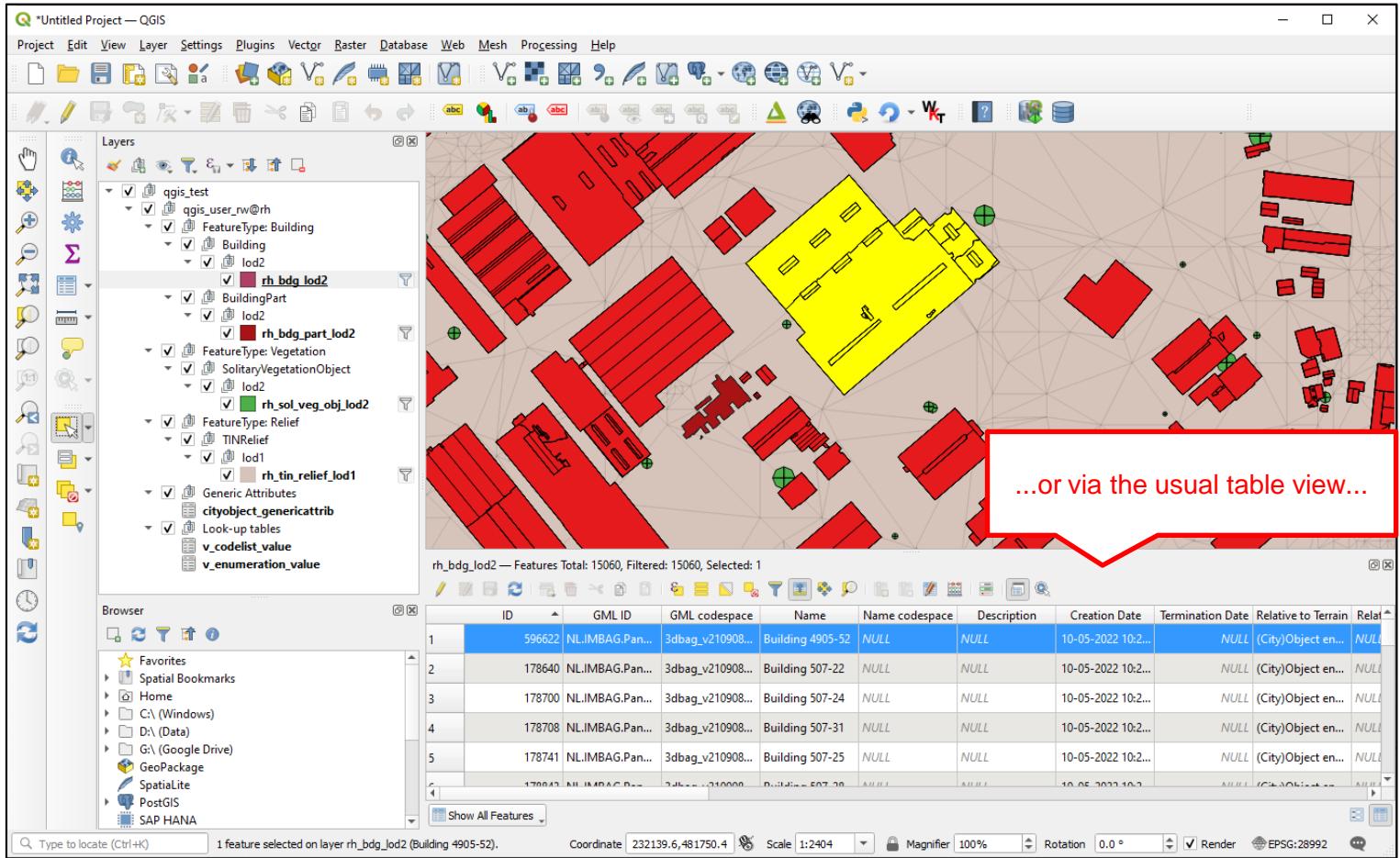
The screenshot shows the QGIS interface with a map view containing red polygonal buildings on a grey terrain background. The 'Identify Results' panel on the right displays detailed attributes for a selected feature. A red callout box points to this panel with the text: "Feature attributes can be retrieved and visualised in the 'Identify results' panel..."

Feature	Value
rh_bdg_lod2	Building 4905-52
Name	(Derived)
ID	596622
GML ID	NLIMBAG.Pand.174210000000106
GML codespace	3dbag_v210908_fd2ceef53_4905.gml
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

Use in QGIS: Attributes

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



The screenshot shows the QGIS interface with a map view and a table view.

Map View: The map displays several red polygonal features representing buildings. One specific building feature is highlighted in yellow. A callout box points to this yellow feature with the text "...or via the usual table view...".

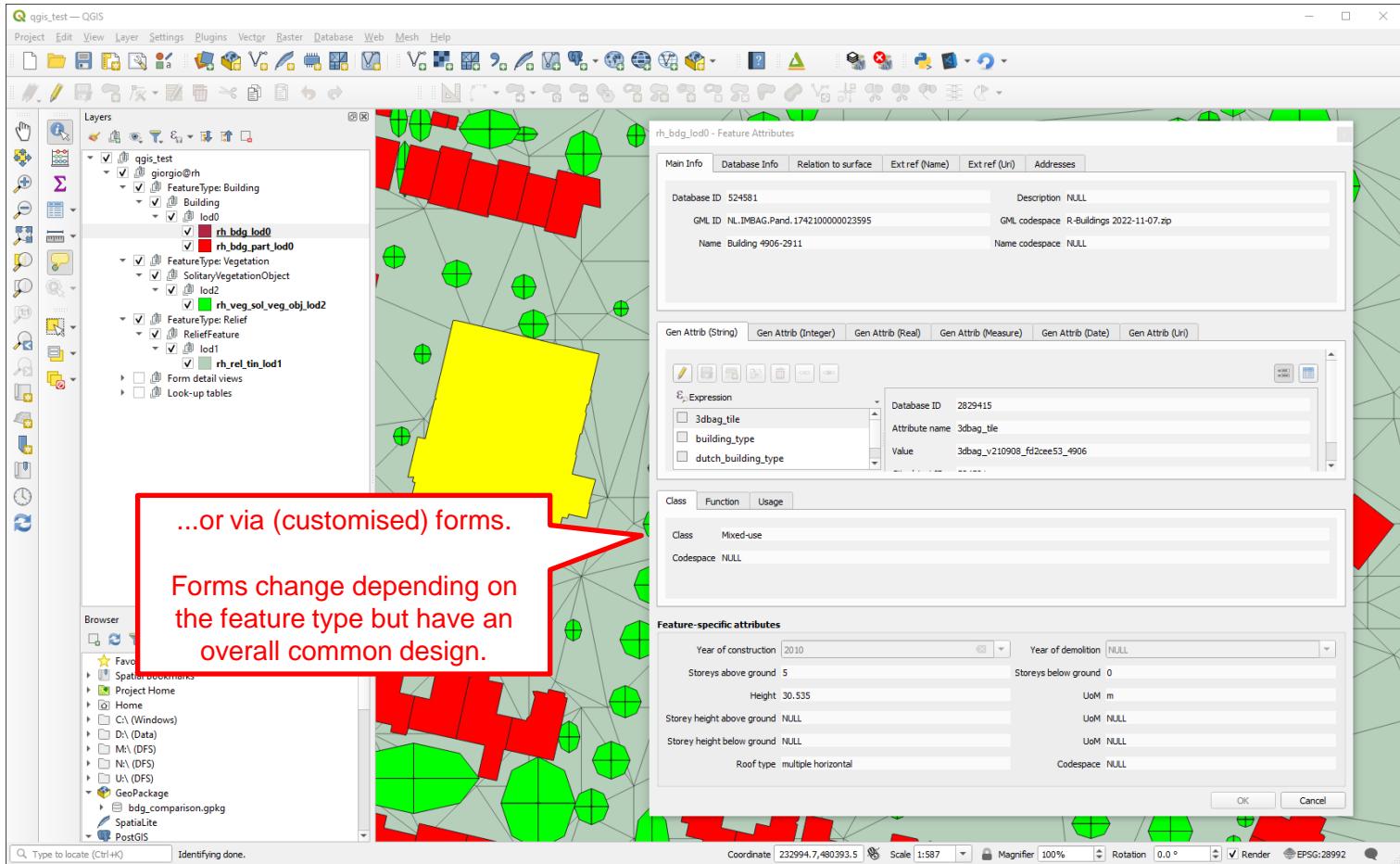
Table View: Below the map, a table titled "rh_bdg_lod2" shows the attributes of the selected building. The table has columns: ID, GML ID, GML codespace, Name, Name codespace, Description, Creation Date, Termination Date, Relative to Terrain, and Relative to Water. The first five rows of data are shown:

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

At the bottom of the table, there is a button labeled "Show All Features".

Use in QGIS: Attributes

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



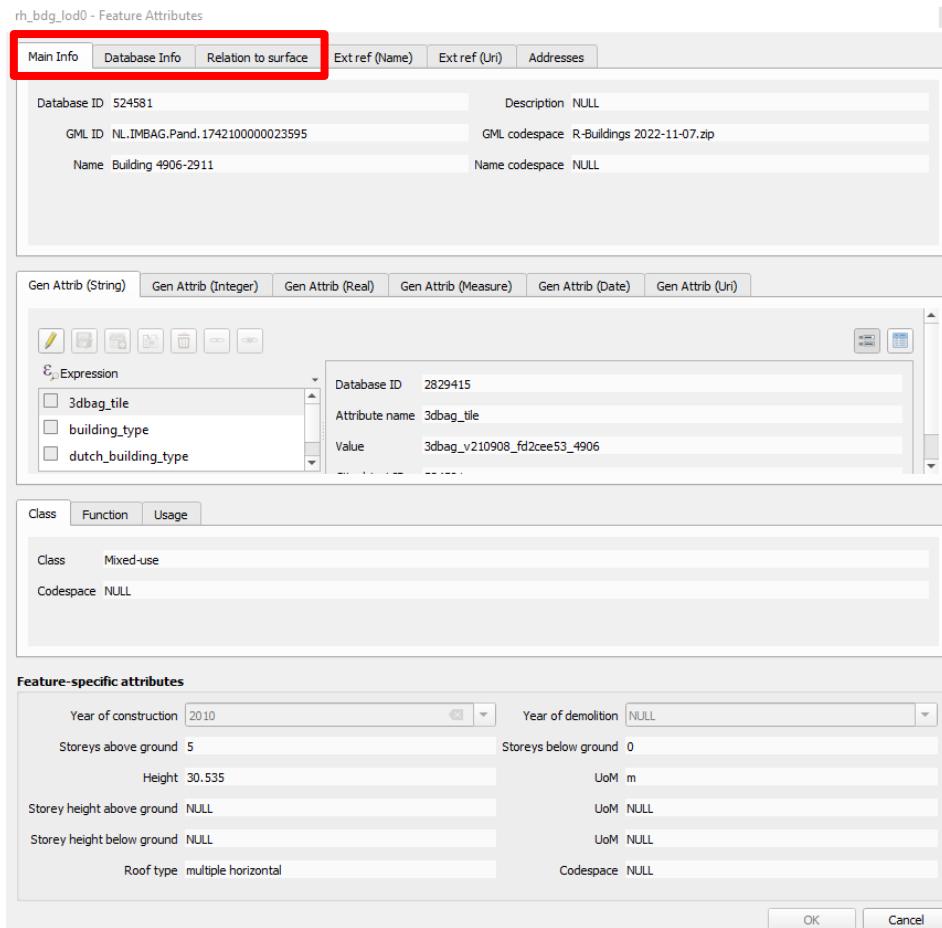
Use in QGIS: Attributes

Motivation
Plugin overview
Software installation
Software use

- Layer Loader
- Bulk Deleter

Advanced options
Software uninstall
Current limitations
QGIS Package
Resources
Changelog

Attributes are grouped into tabs.

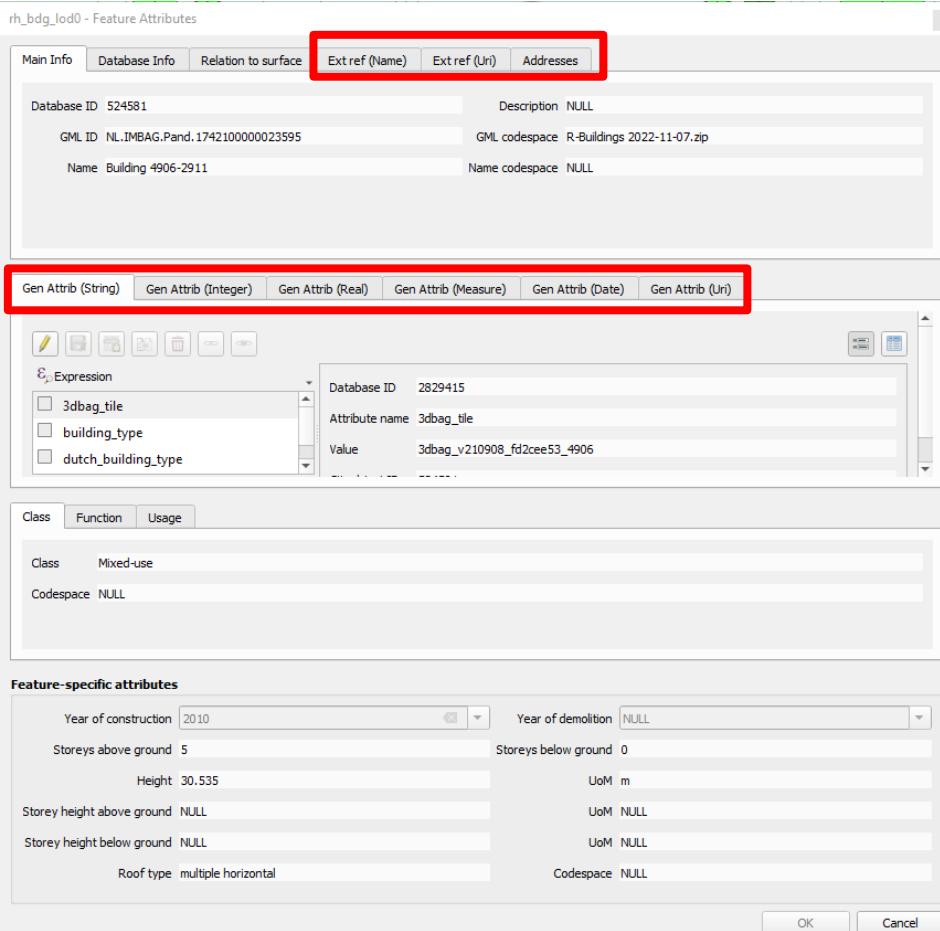


The screenshot shows the QGIS Attribute Editor for a feature named "Building 4906-2911". The top tab bar includes "Main Info", "Database Info", "Relation to surface", "Ext ref (Name)", "Ext ref (Uri)", and "Addresses". A red box highlights the first three tabs. Below the tabs, there are sections for "Database ID" (524581), "GML ID" (NL.IMGAG.Pand.1742100000023595), "Description" (NULL), "Name" (Building 4906-2911), and "Name codespace" (NULL). The "Relation to surface" tab is expanded, showing fields for "Attribute name" (3dbag_tile) and "Value" (3dbag_v210908_fd2cee53_4906). The "Main Info" tab is expanded, showing "Class" (Mixed-use) and "Codespace" (NULL). The "Database Info" tab is expanded, showing "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). Buttons for "OK" and "Cancel" are at the bottom right.

Use in QGIS: Attributes

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

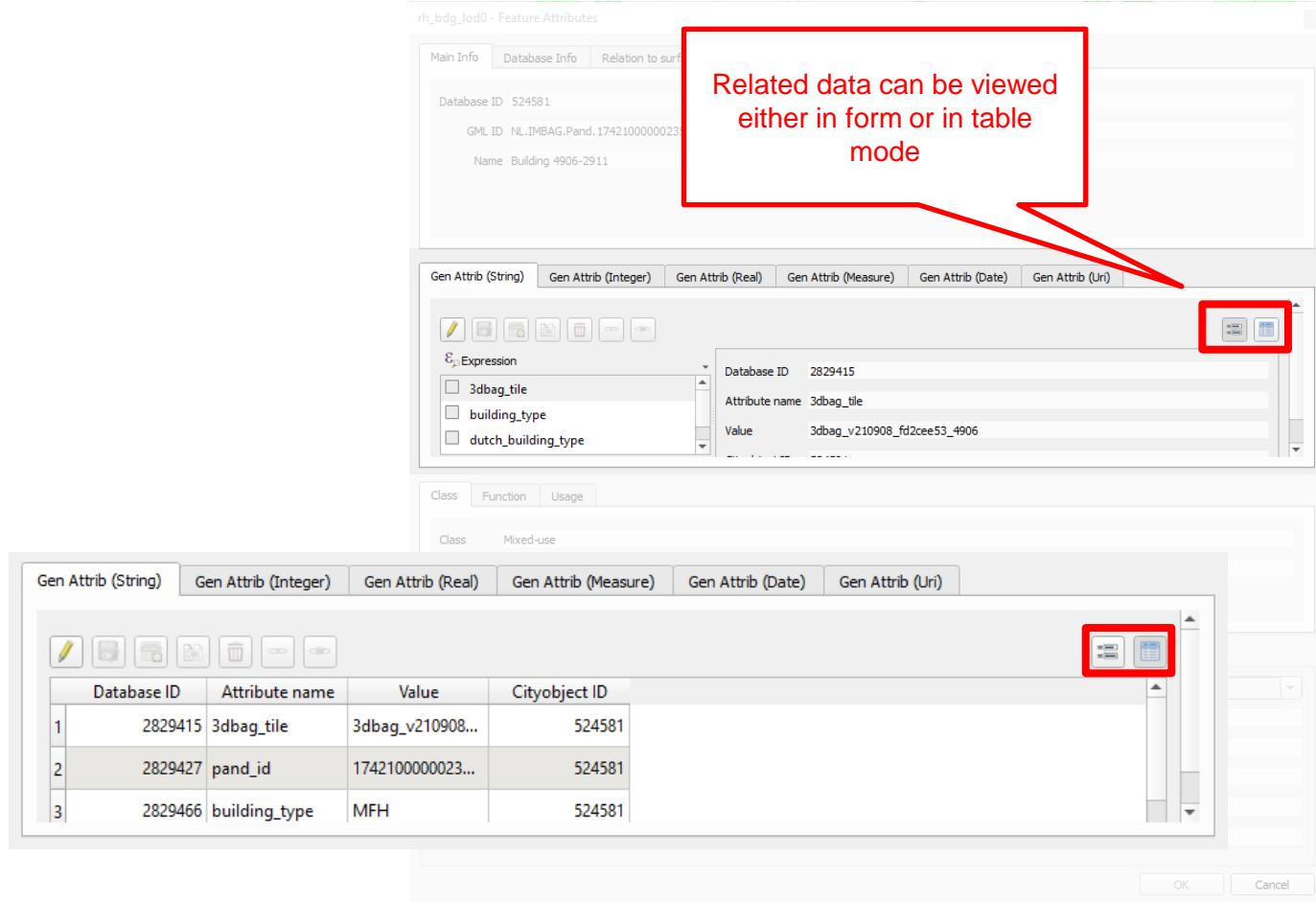
Additional related tables are connected (e.g. External References, Addresses, Generic Attributes)



The screenshot shows the 'Feature Attributes' dialog box in QGIS. 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 currently selected and highlighted with a red border. Below this, there are sections for 'Gen Attrib (String)', 'Gen Attrib (Integer)', 'Gen Attrib (Real)', 'Gen Attrib (Measure)', 'Gen Attrib (Date)', and 'Gen Attrib (Uri)'. The 'Expression' section displays attribute names and values: '3dbag_tile' (Value: 3dbag_v210908_fd2cee53_4906), 'building_type' (Value: NULL), and 'dutch_building_type' (Value: NULL). The 'Class' section shows 'Mixed-use' and 'Codespace' (NULL). The 'Feature-specific attributes' section contains fields for '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), 'Height UoM' (m), 'Storey height UoM' (NULL), 'Storey height below ground UoM' (NULL), and 'Roof type Codespace' (NULL). Buttons for 'OK' and 'Cancel' are at the bottom right.

Use in QGIS: Attributes

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



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 (Building 4906-2911). A red box highlights the "Form" and "Table" buttons in the bottom right corner of this panel. The bottom part shows the "Table" mode, where attributes are listed in a table:

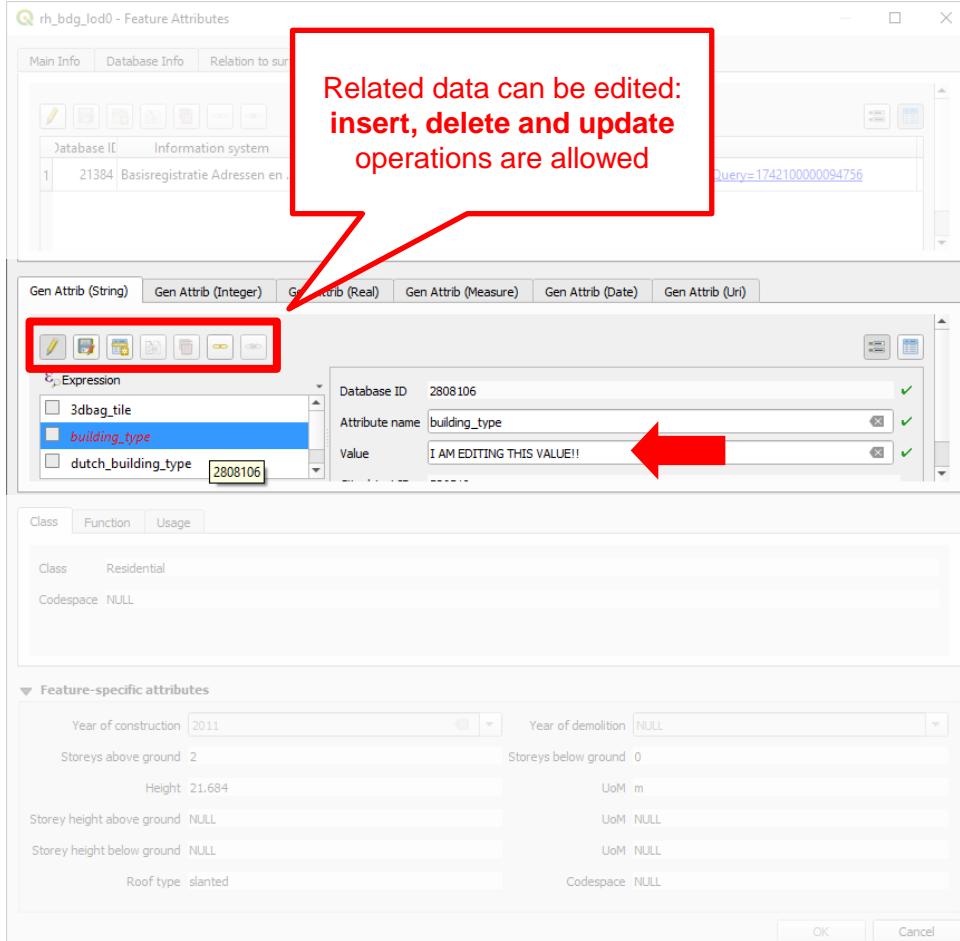
	Database ID	Attribute name	Value	Cityobject ID
1	2829415	3dbag_tile	3dbag_v210908...	524581
2	2829427	pand_id	1742100000023...	524581
3	2829466	building_type	MFH	524581

A red box highlights the "Form" and "Table" buttons in the bottom right corner of the Attribute Editor interface.

Related data can be viewed
either in form or in table
mode

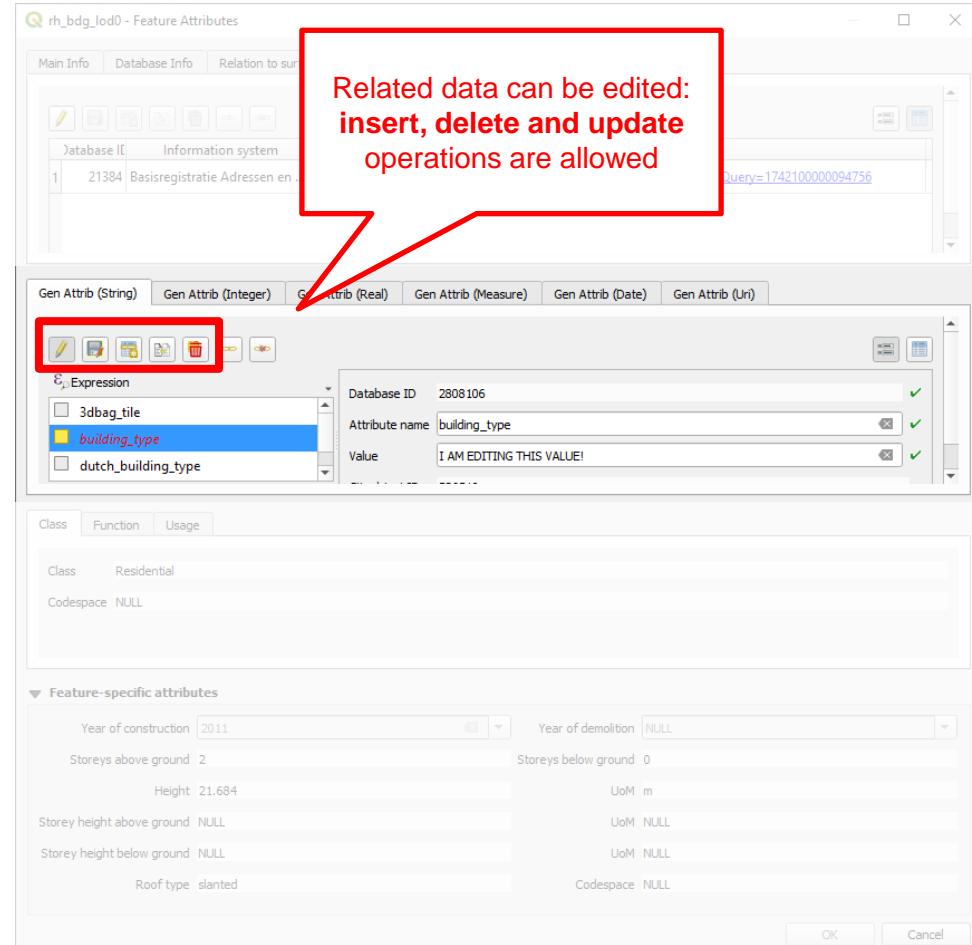
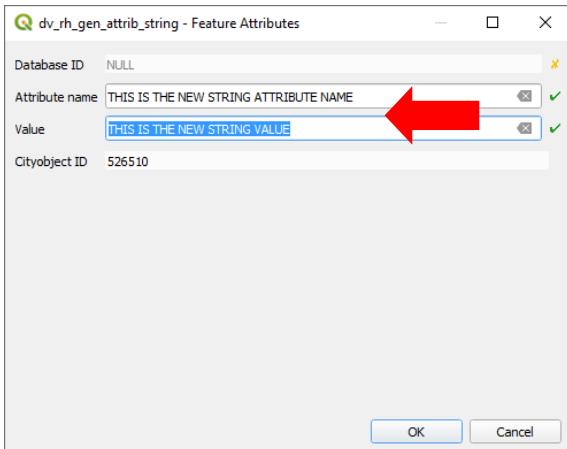
Use in QGIS: Attributes

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



Use in QGIS: Attributes

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

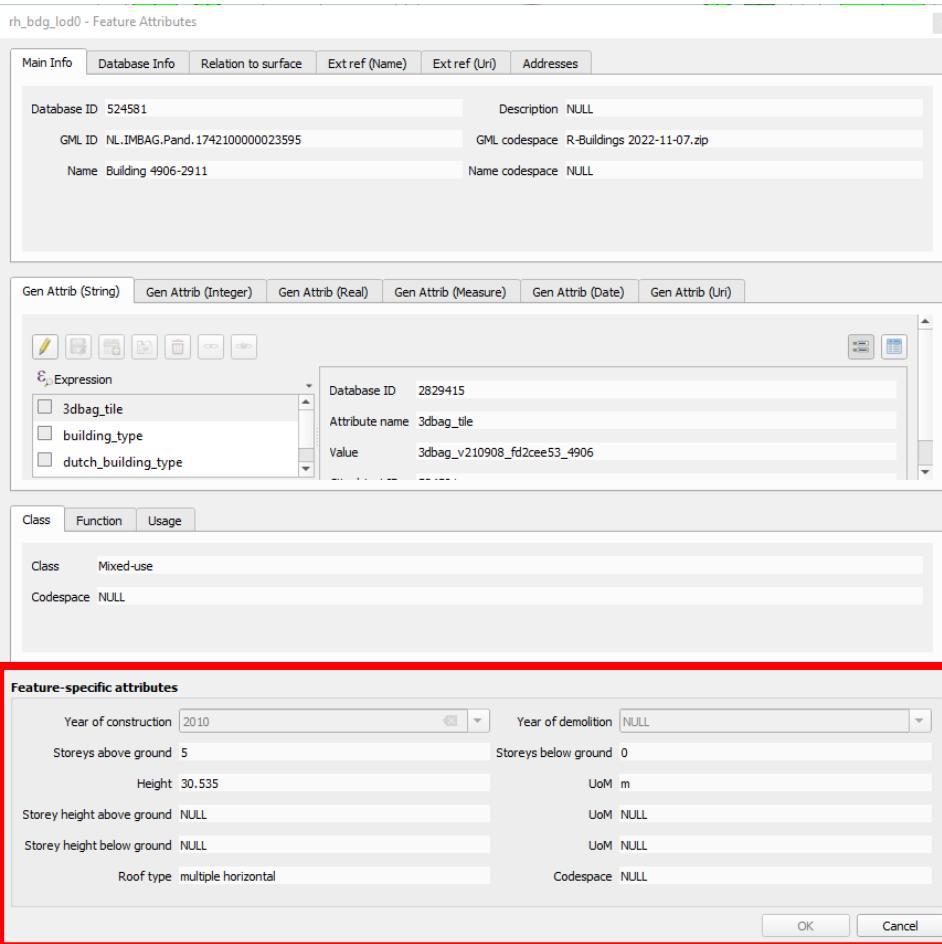


Motivation
Plugin overview
Software installation
Software use

- Layer Loader
- Bulk Deleter

Advanced options
Software uninstall
Current limitations
QGIS Package
Resources
Changelog

Use in QGIS: Attributes



The screenshot shows the QGIS Attribute Editor for a feature named "Building 4906-2911". The top section displays general information such as Database ID (524581), GML ID (NL.IMBAG.Pand.1742100000023595), Description (NULL), and Name (Building 4906-2911). Below this, the "Gen Attrib (String)" tab is selected, showing an expression builder with three items: "3dbag_tile", "building_type", and "dutch_building_type". The "Value" field for "3dbag_tile" is set to "3dbag_v210908_fd2cee53_4906". The bottom section, titled "Feature-specific attributes", contains fields for 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), and Codespace (NULL).

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.IMBAG.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

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

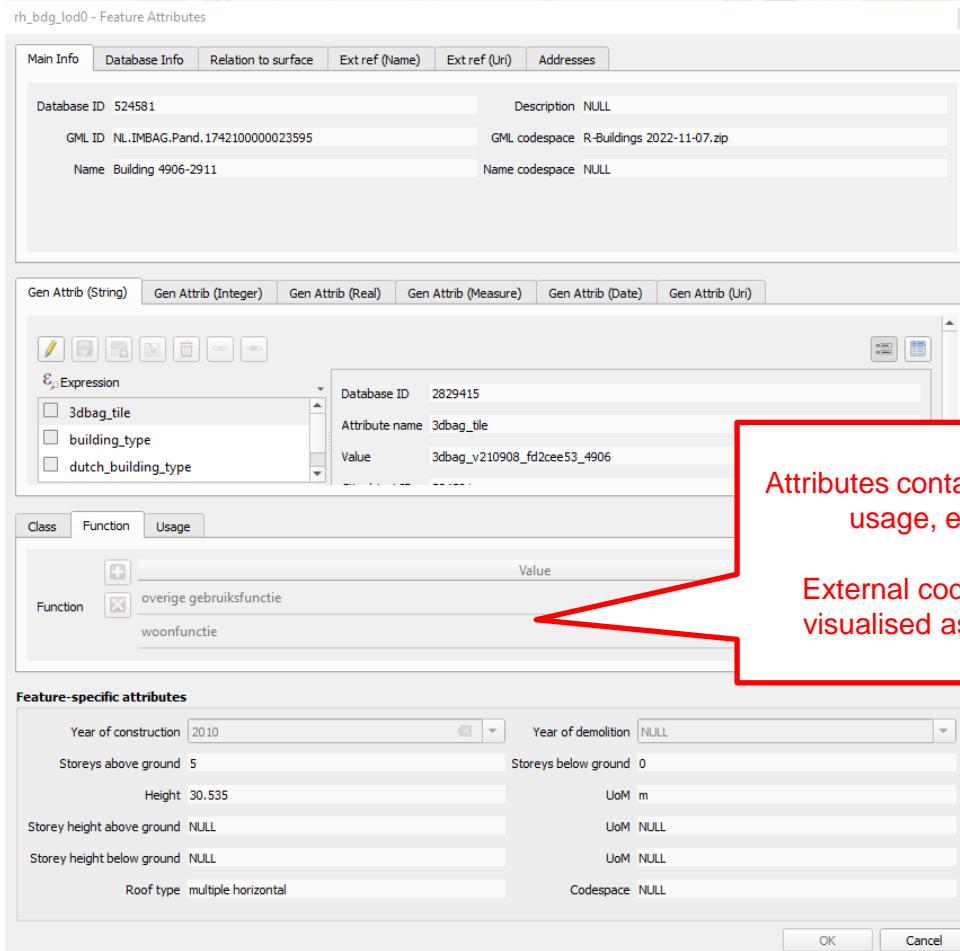
Storey height below ground NULL UoM NULL

Roof type multiple horizontal Codespace NULL

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
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources
- Changelog

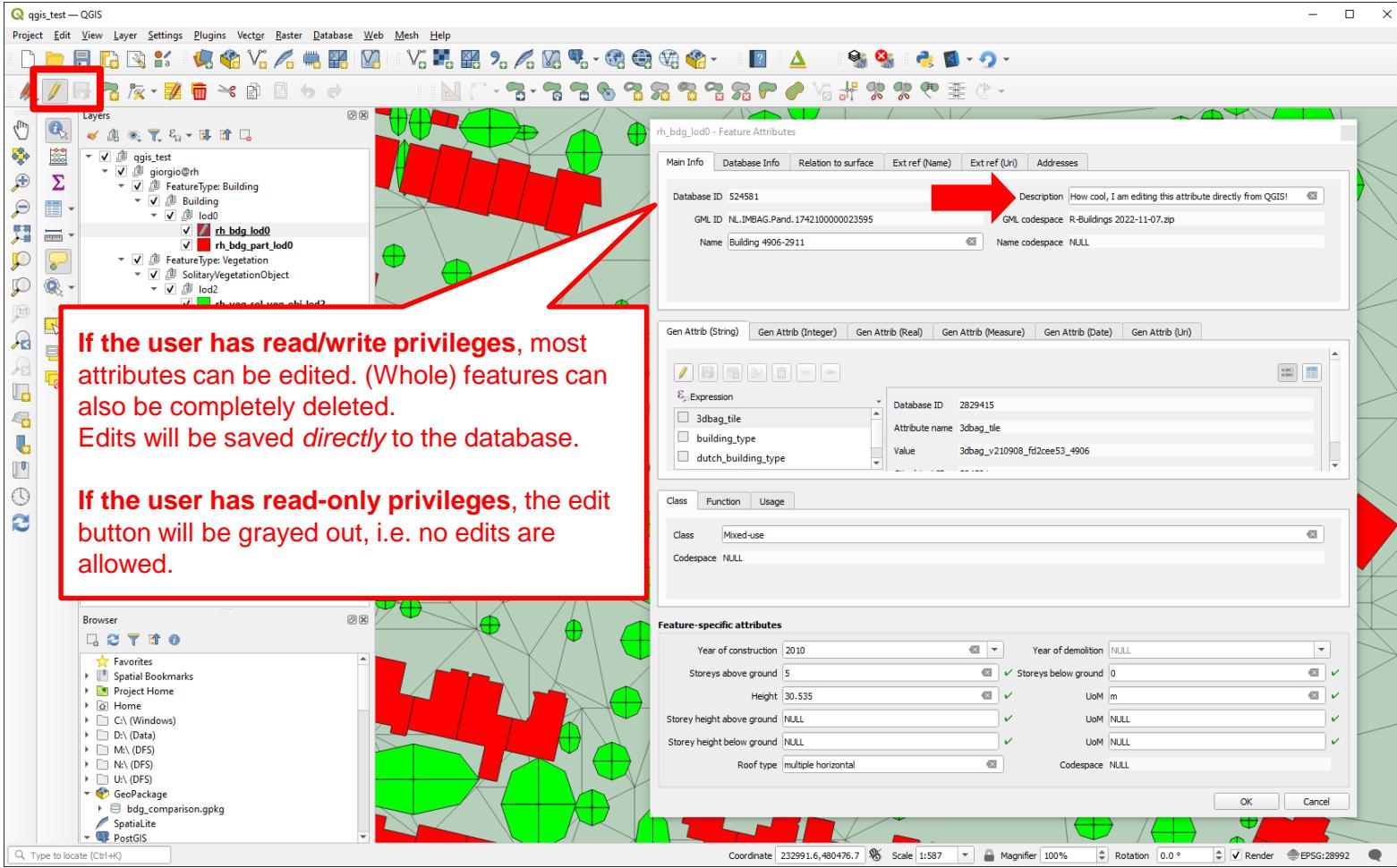


Attributes containing [0..*] entries (e.g. function, usage, etc.) are presented as lists.

External codelists can also be loaded and visualised as look-up tables (see later on)

Use in QGIS: Attribute editing

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



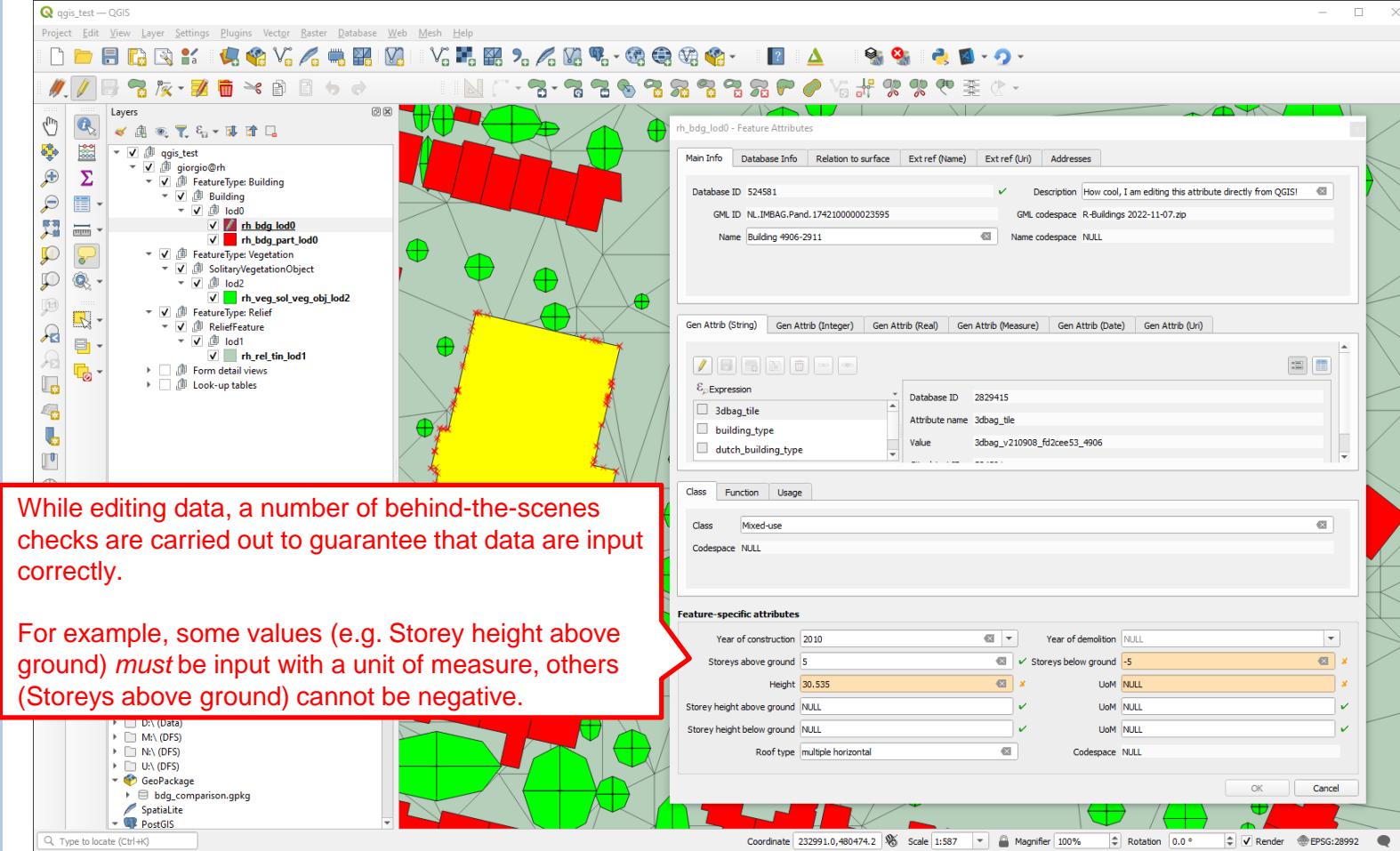
If the user has read/write privileges, most attributes can be edited. (Whole) features can also be completely deleted. Edits will be saved *directly* to the database.

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

The screenshot shows the QGIS interface with a building layer selected. The 'Edit' button in the toolbar is highlighted with a red box. The 'Feature Attributes' dialog is open for a selected building feature, showing fields like 'Name' and 'Description'. The 'Edit' button in this dialog is also highlighted with a red box. Below it, the 'Feature-specific attributes' dialog is shown with various building properties like 'Year of construction', 'Storeys above ground', and 'Roof type'. The 'Edit' button here is also highlighted with a red box.

Use in QGIS: Attribute editing

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



The screenshot shows the QGIS interface with a project named "qgis_test". The left panel displays a tree view of layers, including "giorgio@rh" which contains "FeatureType: Building", "FeatureType: Vegetation", and "FeatureType: Relief". A specific layer, "rh_bdg_lod0", is selected and highlighted in red. The main canvas shows a 3D rendering of buildings and vegetation. A context menu is open over a yellow-shaded building, showing options like "Edit feature" and "Attribute table". To the right, a "Feature Attributes" dialog box is open for the selected feature. The dialog has tabs for "Main Info", "Database Info", "Relation to surface", "Ext ref (Name)", "Ext ref (URI)", and "Addresses". Under "Main Info", the "Name" field is set to "Building 4906-2911". The "Gen Attrib (String)" tab is active, showing fields for "Expression", "Database ID" (2829415), "Attribute name" (3dbag_tile), and "Value" (3dbag_v210908_fd2ceef5_4906). The "Feature-specific attributes" tab shows fields for "Year of construction" (2010), "Storeys above ground" (5), "Height" (30.535), and "Roof type" (multiple horizontal). A red callout box points from the text below to the "Height" field.

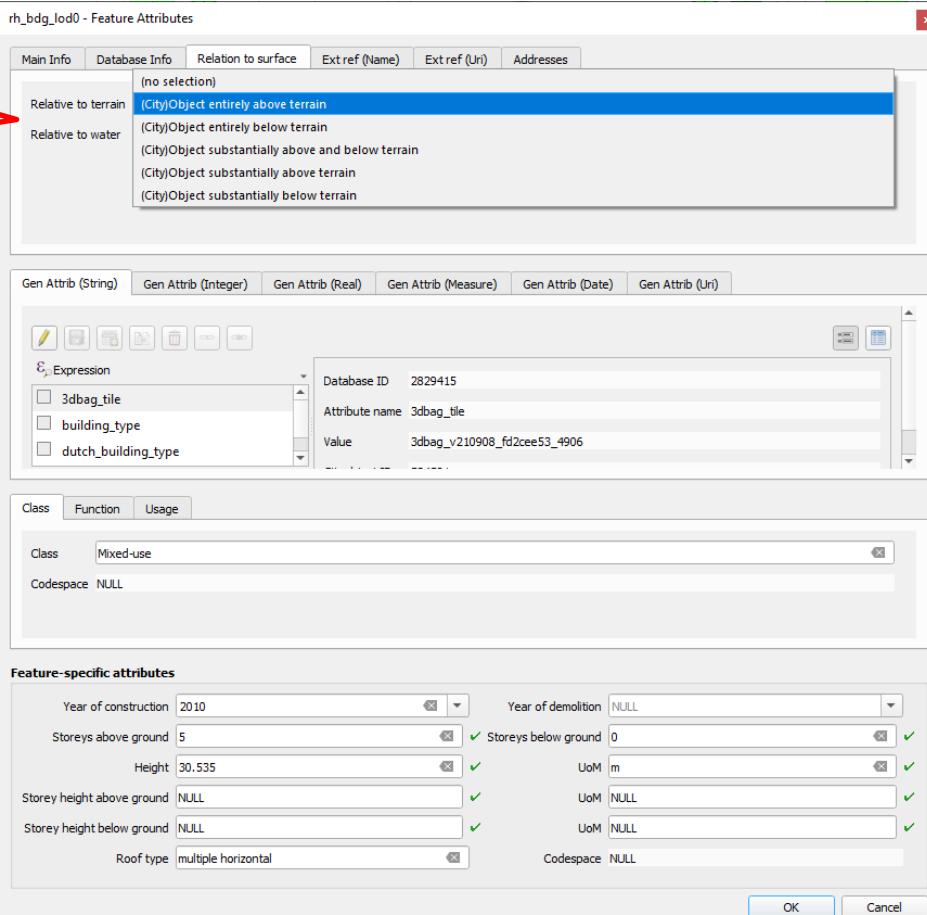
While editing data, a number of behind-the-scenes checks are carried out to guarantee that data are input correctly.

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.

Use in QGIS: Attribute editing

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

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



Use in QGIS: Codelist support

Motivation
Plugin overview
Software installation
Software use

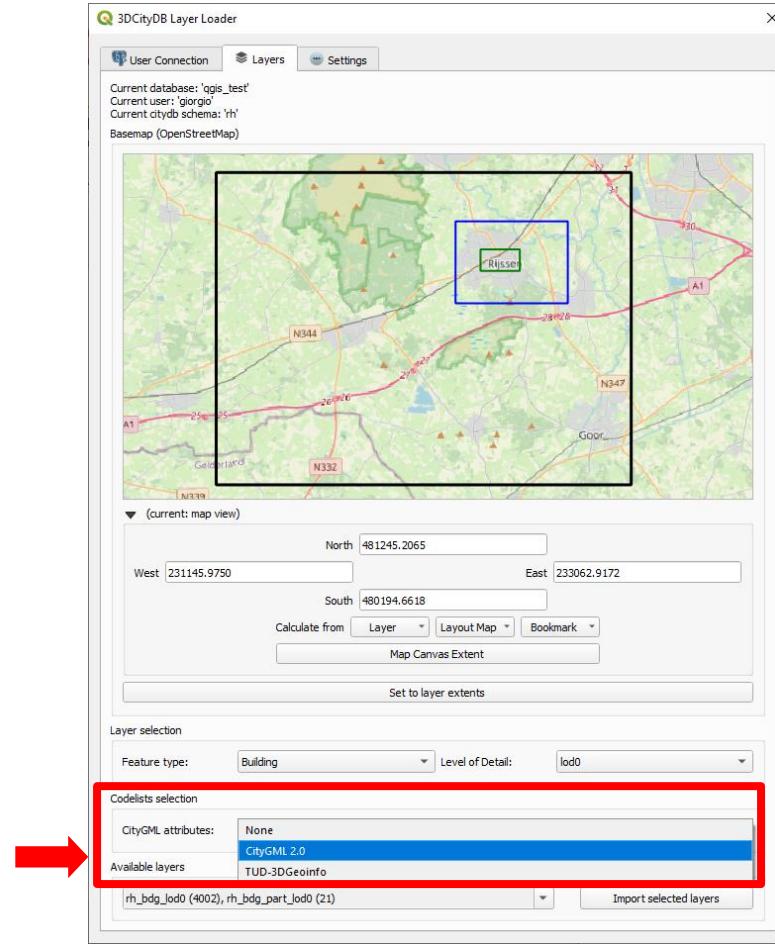
- Layer Loader
- Bulk Deleter

Advanced options
Software uninstall
Current limitations
QGIS Package
Resources
Changelog

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**)

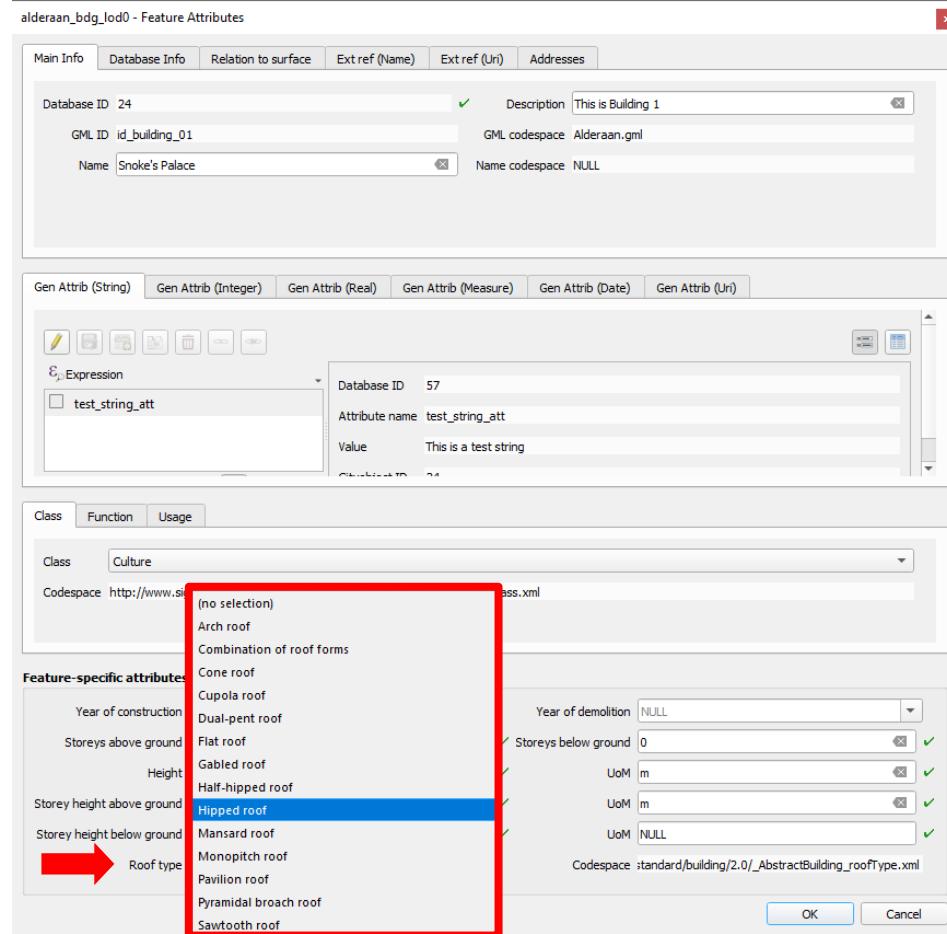


Use in QGIS: Codelist support

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

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**

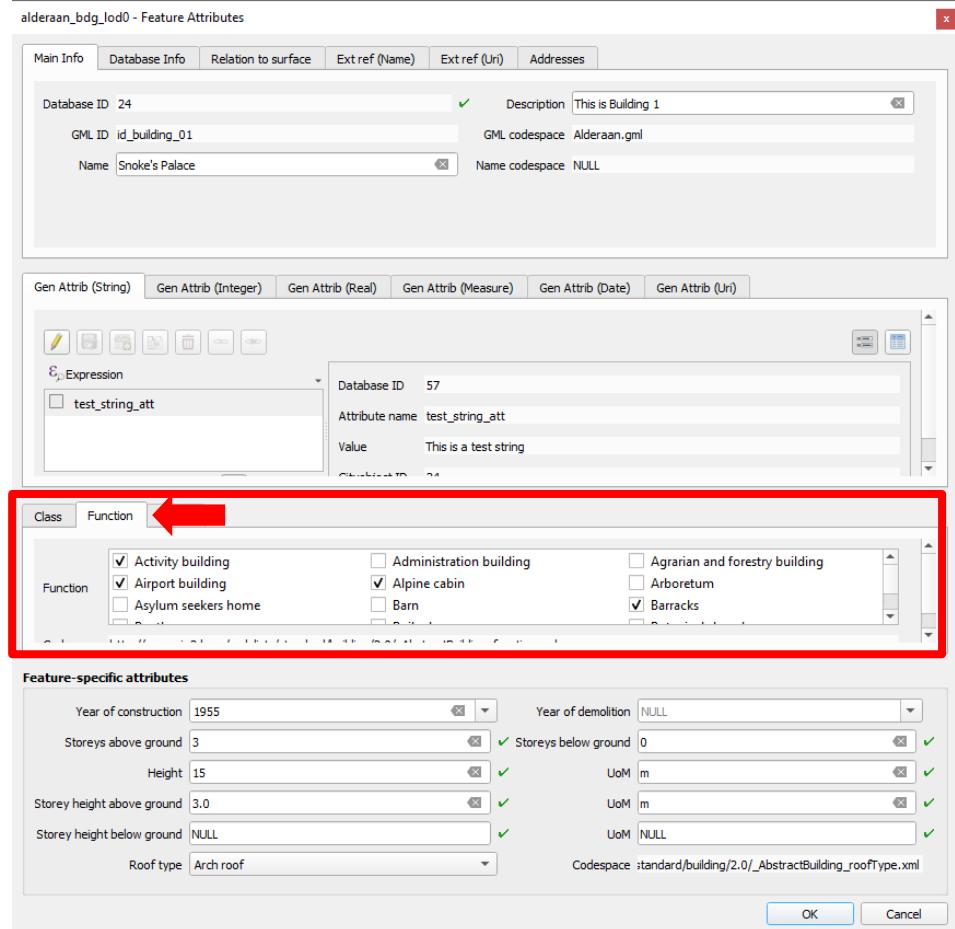


Use in QGIS: Codelist support

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

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

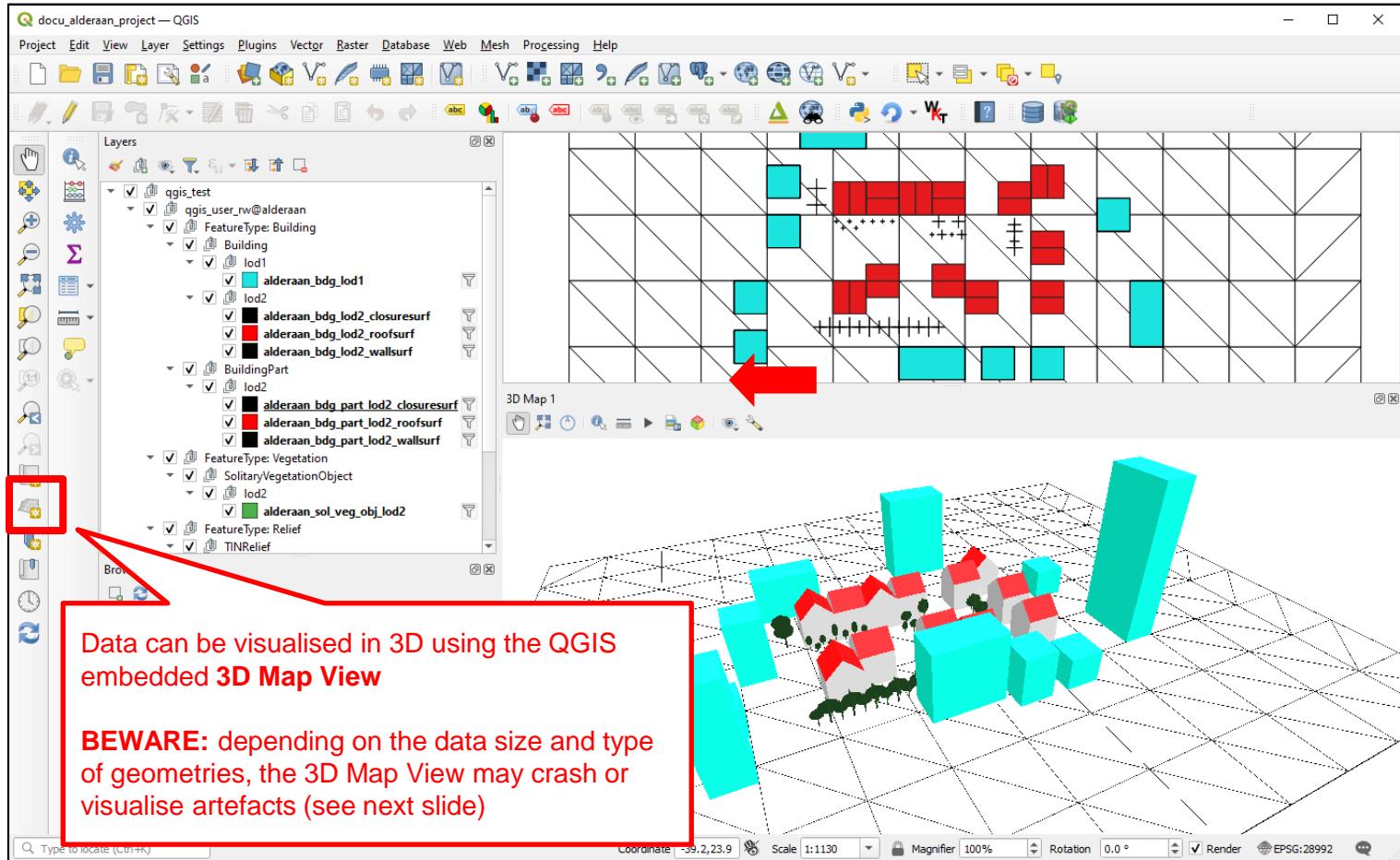


Use in QGIS: 3D visualisation

Motivation
Plugin overview
Software installation
Software use

- Layer Loader
- Bulk Deleter

Advanced options
Software uninstall
Current limitations
QGIS Package
Resources
Changelog

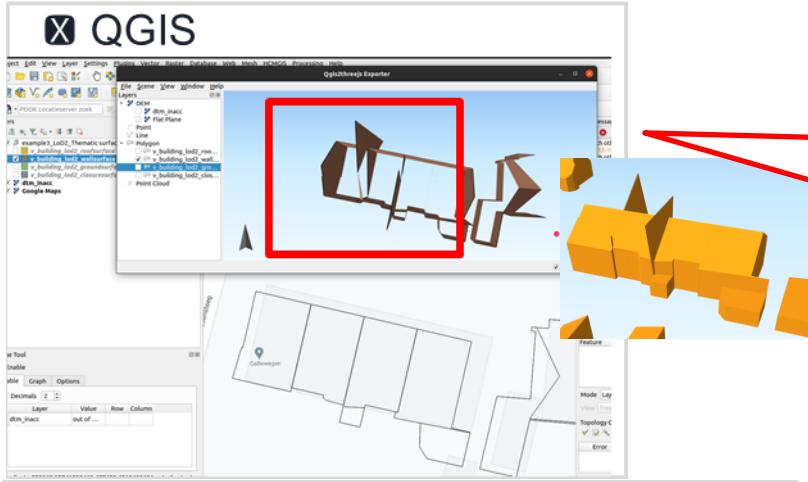


Use in QGIS: 3D visualisation

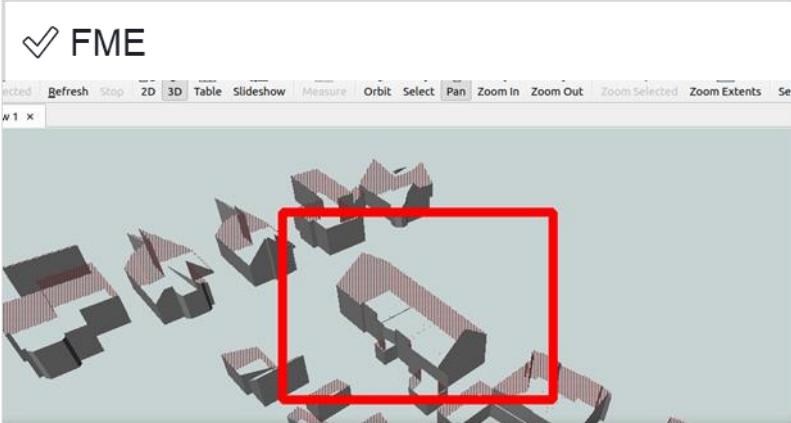
Motivation
 Plugin overview
 Software installation
Software use

- Layer Loader
- Bulk Deleter

 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog



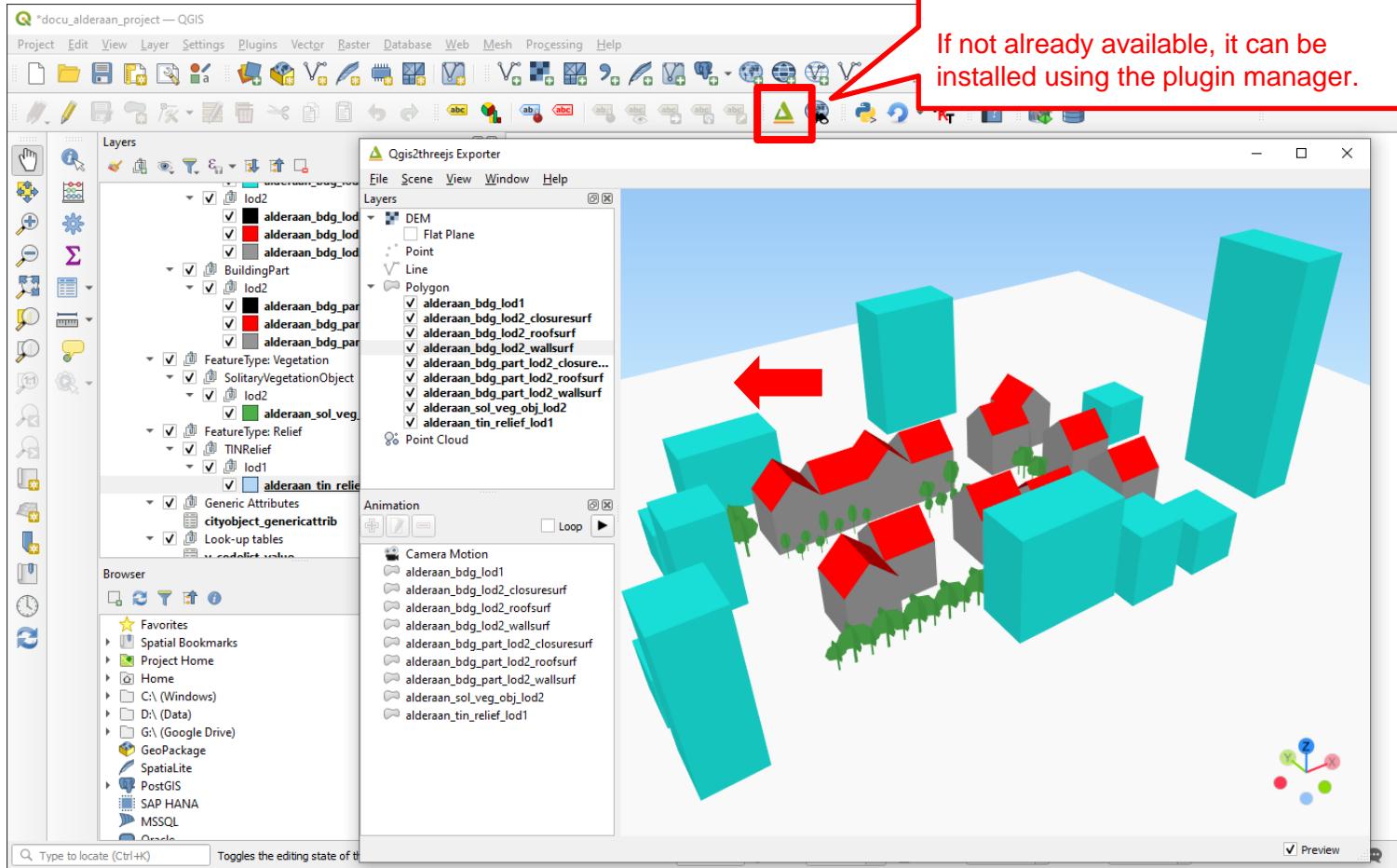
In our tests, such artefacts are however a 3D visualisation issue (QGIS 3D renderer?) and *not* related to the actual data. 3D visualisation in FME and in Google Earth show indeed correct results.



✓
Google earth
 (as KML)

Use in QGIS: 3D visualisation

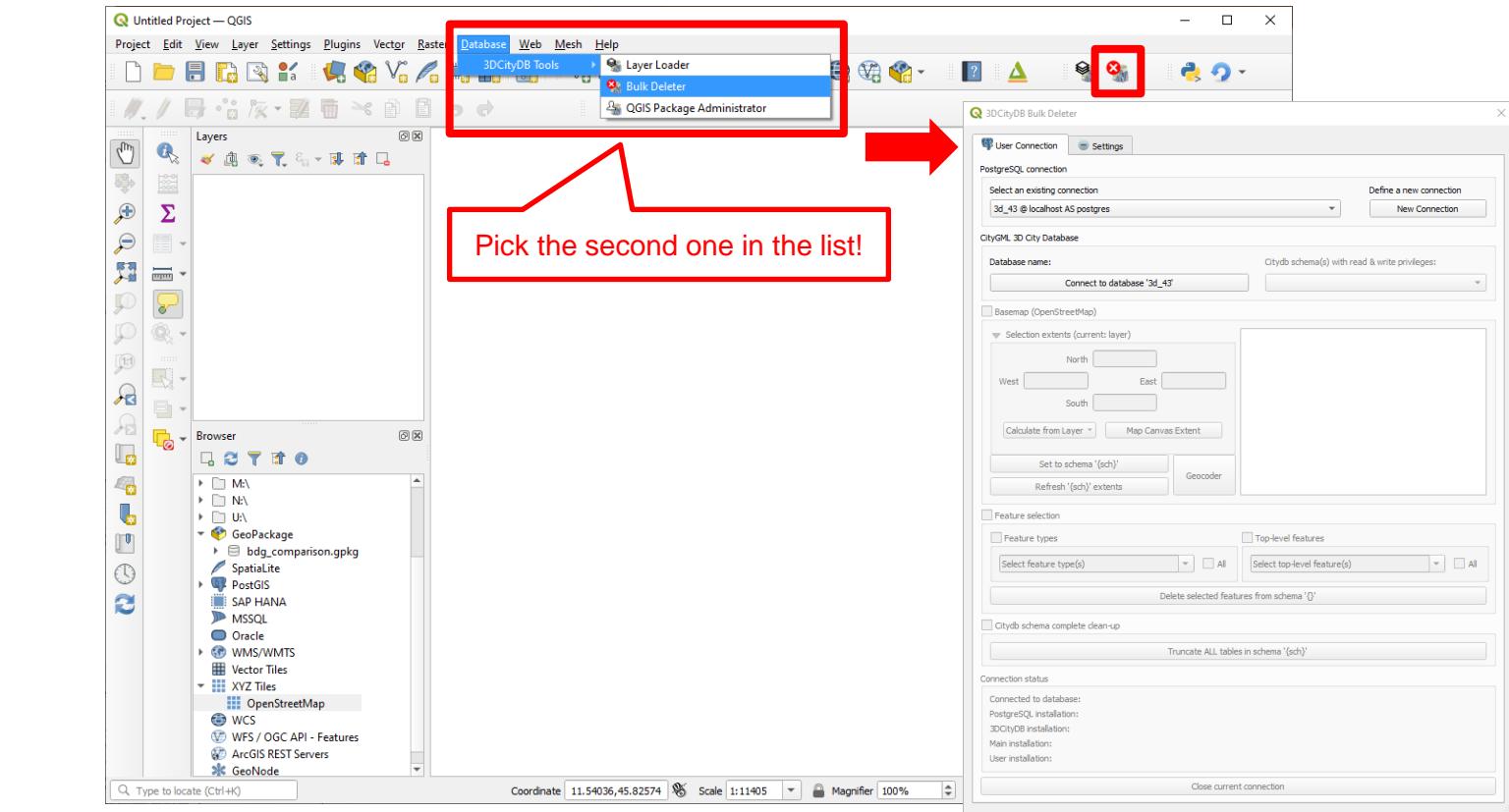
- Motivation
- Plugin overview
- Software installation
- Software use**
 - Layer Loader
 - Bulk Deleter
- Advanced options
- Software uninstall
- Current limitations
- QGIS Package
- Resources
- Changelog



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

Advanced options
Software uninstall
Current limitations
QGIS Package
Resources
Changelog

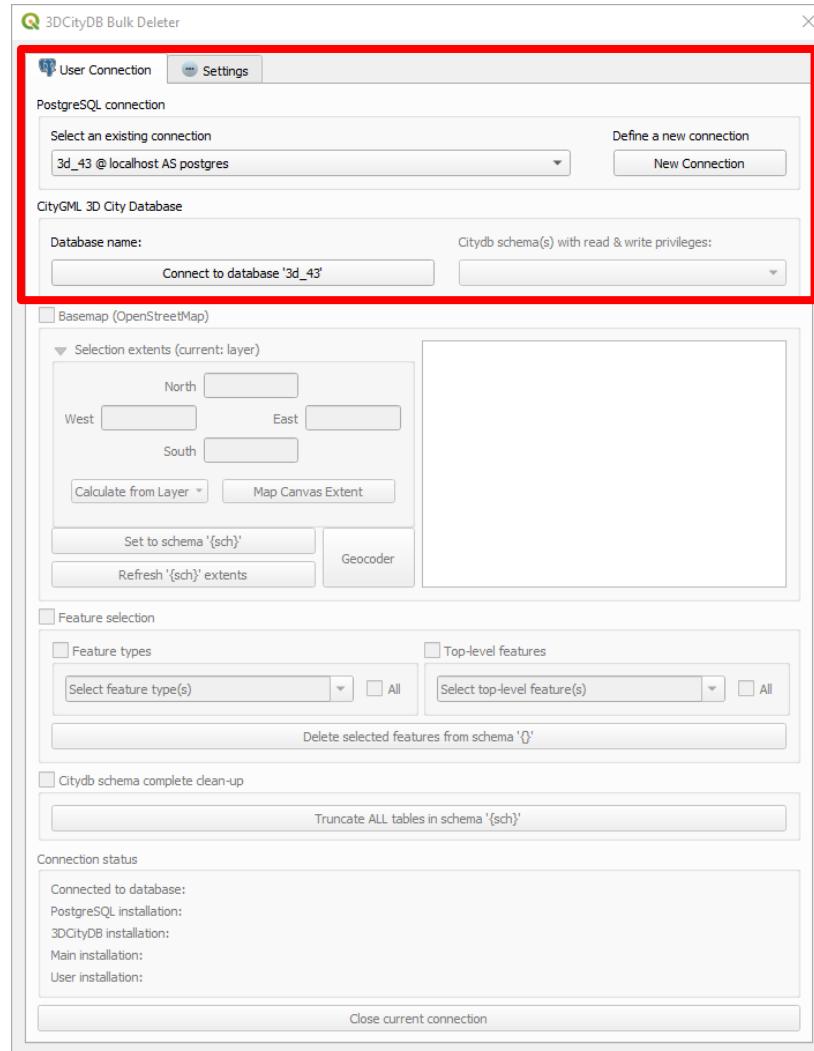
82

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")

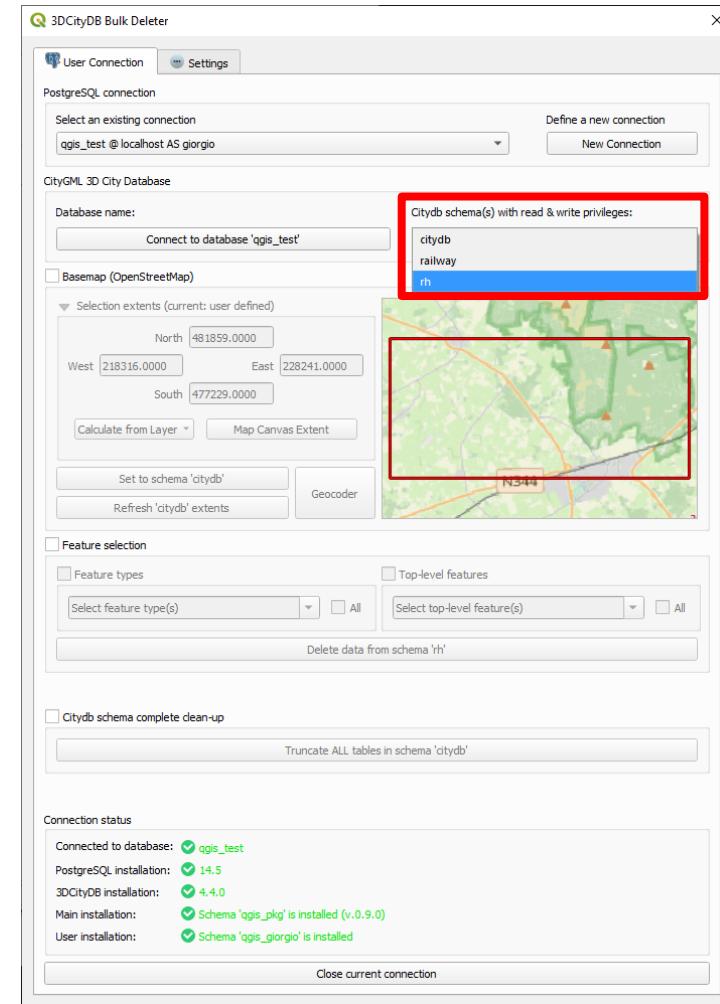
2) Connect to the chosen database



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

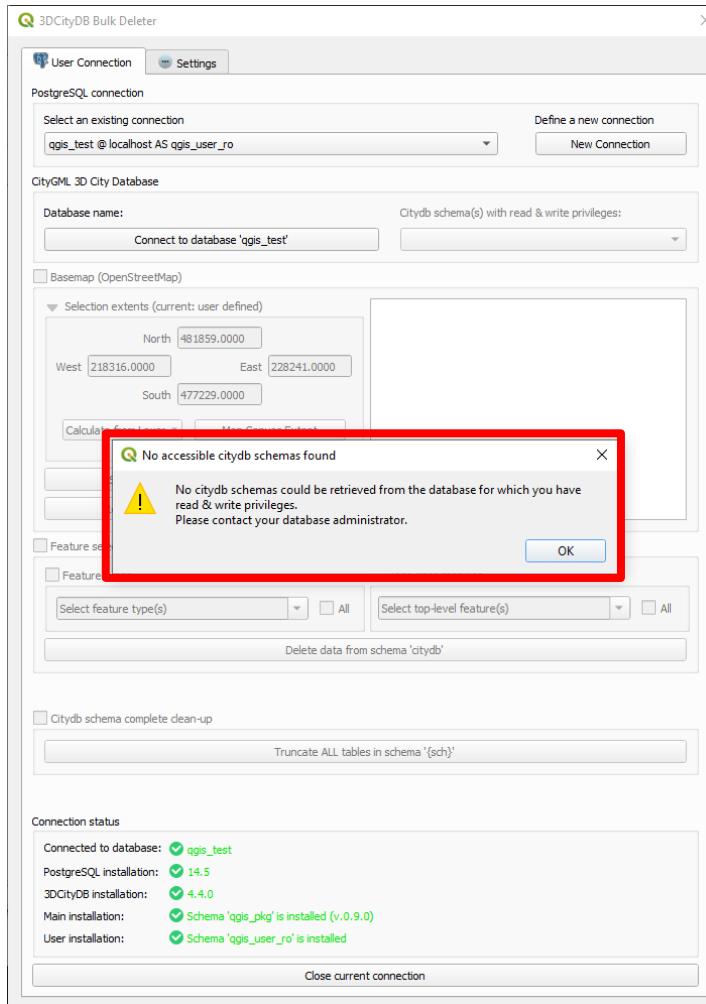


Bulk Deleter

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.

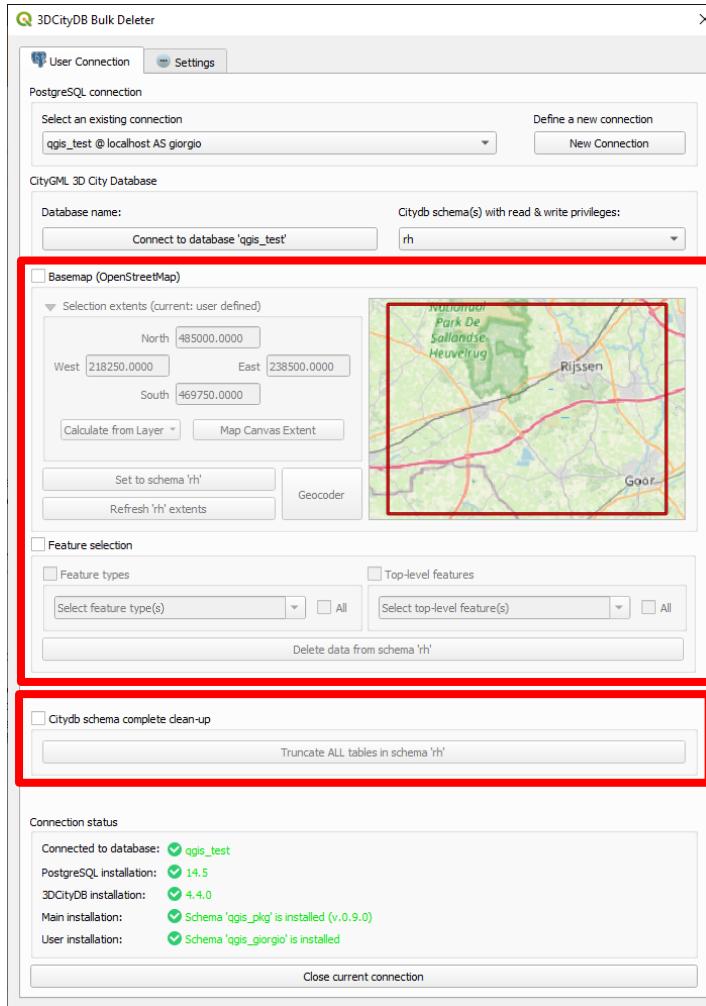


Bulk Deleter

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

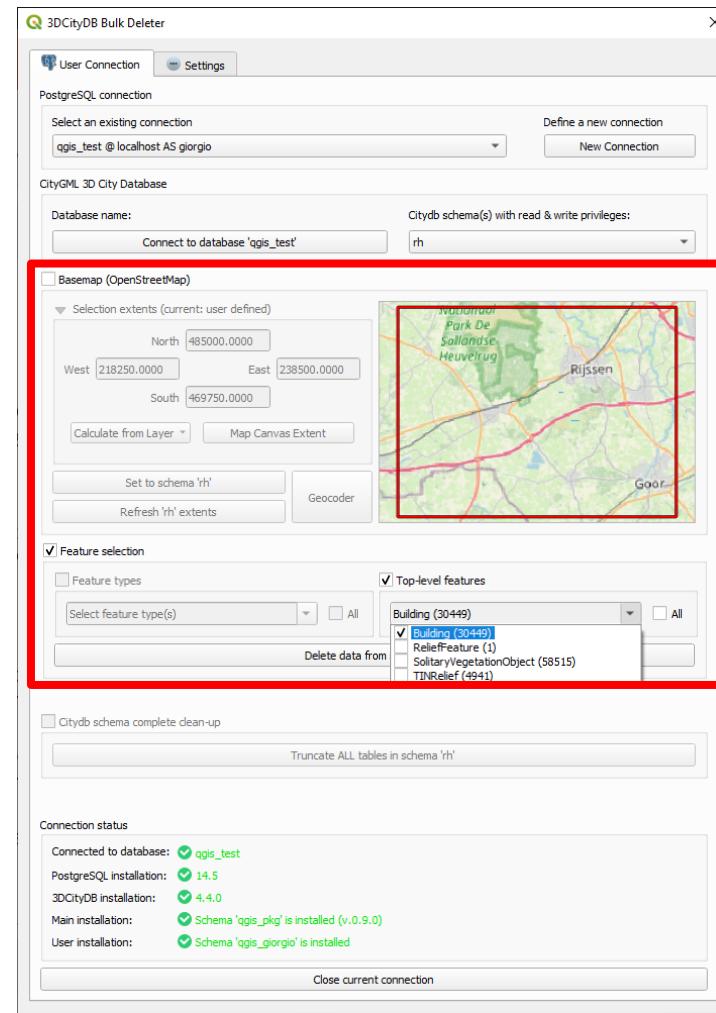
Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

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



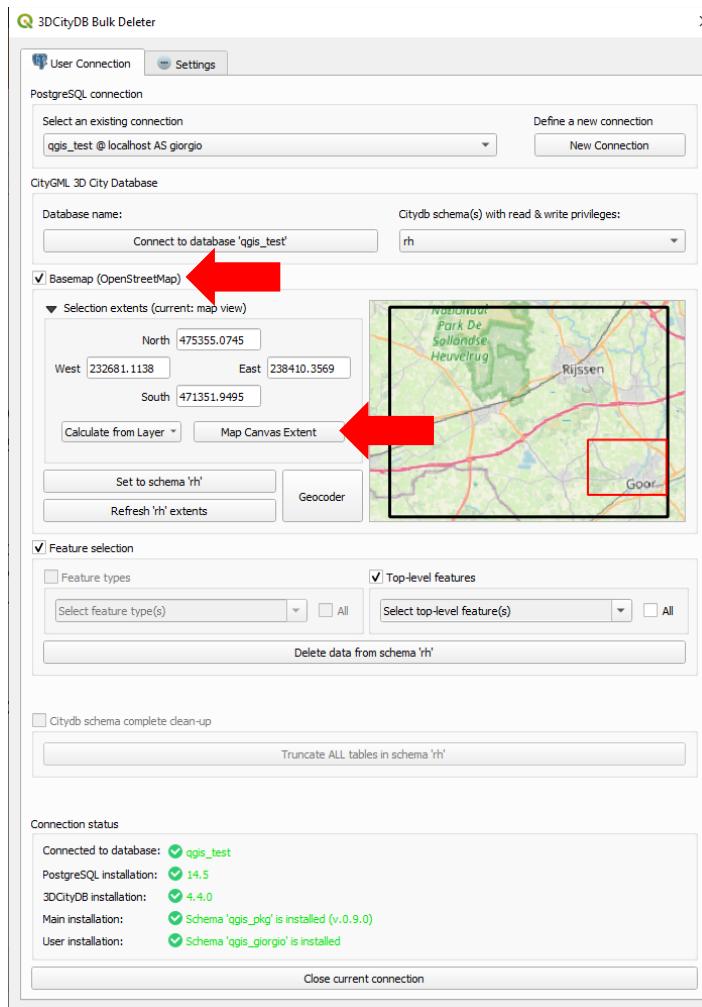
Bulk Deleter

Motivation
 Plugin overview
 Software installation
Software use
 • Layer Loader
 • Bulk Deleter
 Advanced options
 Software uninstall
 Current limitations
 QGIS Package
 Resources
 Changelog

- 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



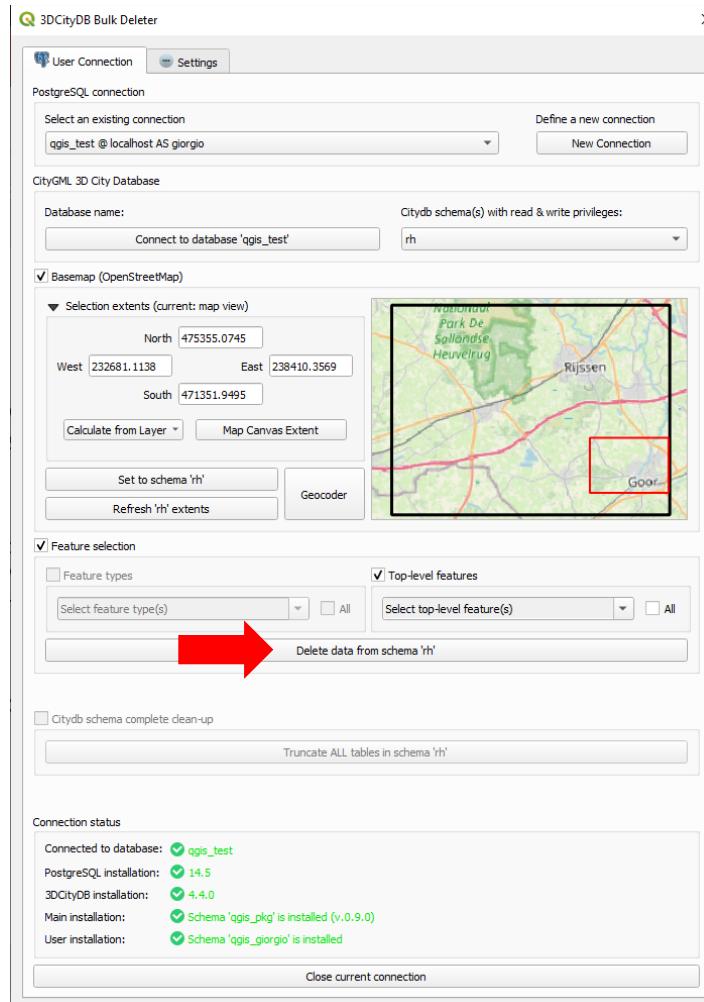
Bulk Deleter

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

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!**

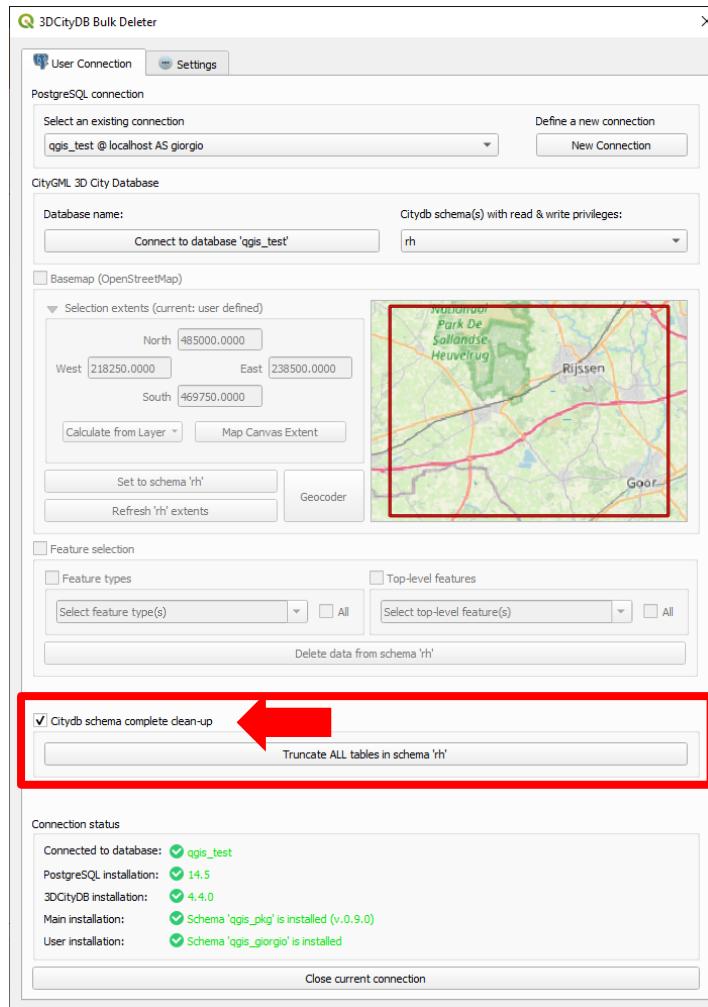


Bulk Deleter

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

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"



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)

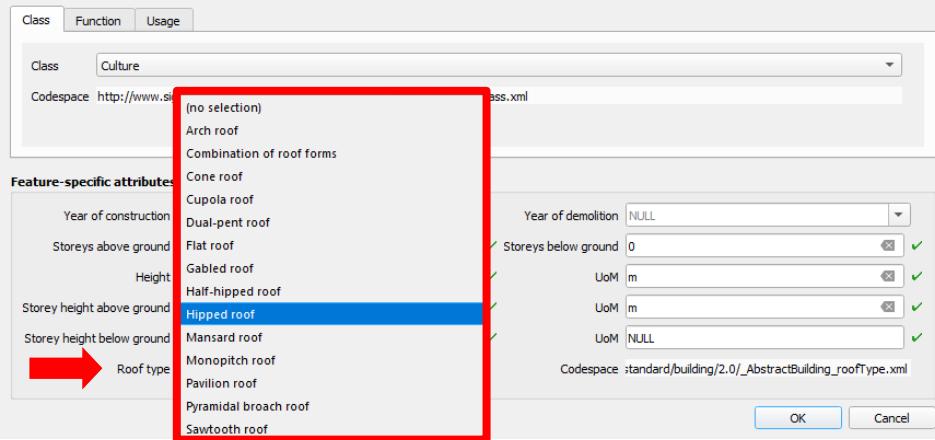
In this way the user does not have to "remember" specific codes when typing, thus reducing the chances of wrong data input

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

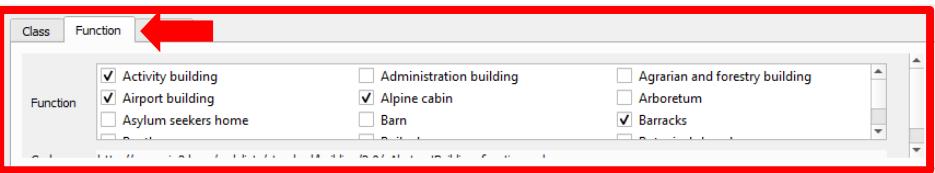
Customised codelists

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

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

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
> codelist_value
> enumeration
> enumeration_value
> extents
> layer_metadata
> settings

Table CODELIST (excerpt)

id [PK] bigint	data_model character varying	name character varying	name_space character varying	description 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)

id [PK] bigint	code_id integer	value character varying	description 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)

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

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

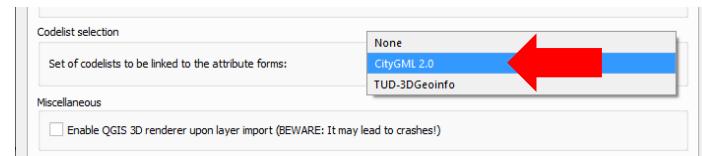
Tables (8)	
>	codelist
>	codelist_lookup_config
>	codelist_value
>	enumeration
>	enumeration_value
>	extents
>	layer_metadata
>	settings

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.

id [PK]	name character varying	ade_prefix character var	source_class character varying	source_table character varying	source_column character varying	target_table character varying	key_column character var	value_column character varyir	filter_expression character varying
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

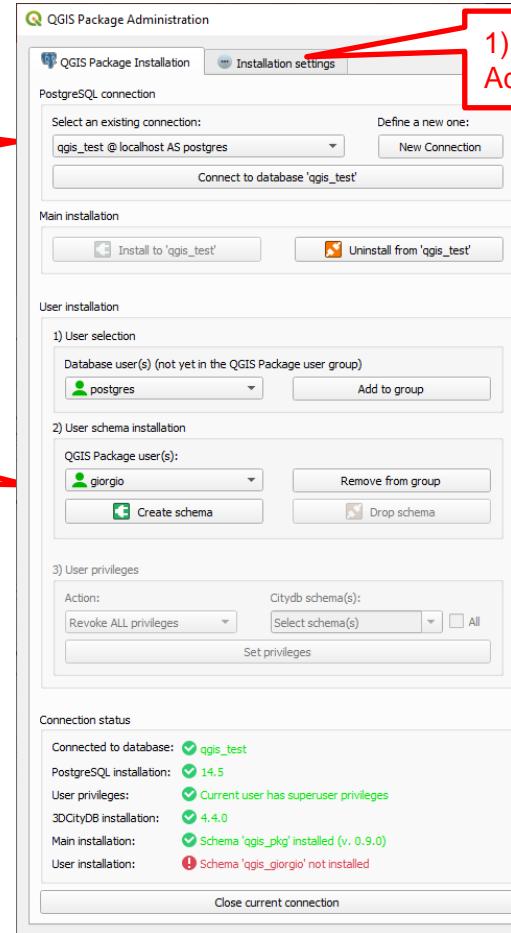
2) **Removal of the 3DCityDB-Tools plugin** from QGIS

- 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
Changelog

2) Connect as administrator



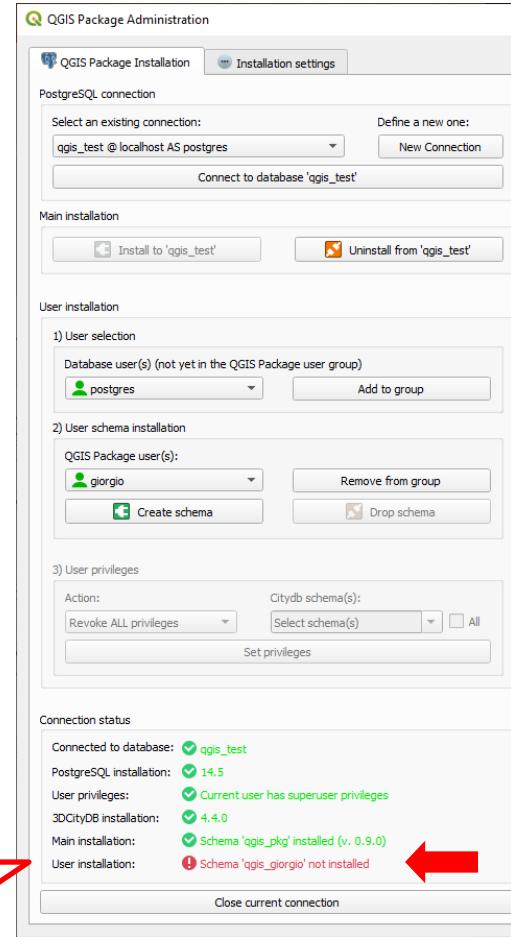
1) Load the "QGIS Package Administration" GUI

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
Changelog

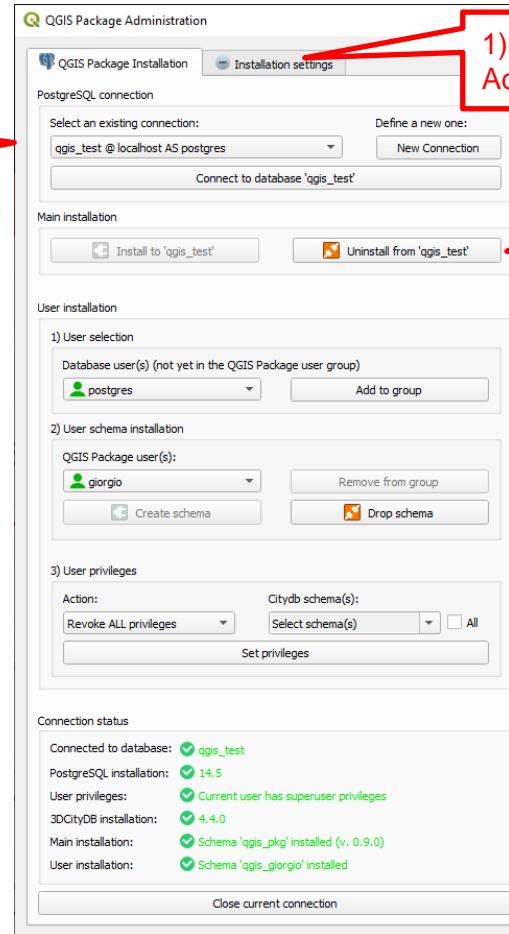


5) The user installation field
is now red again

Software uninstall: Uninstall QGIS Package

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

2) Connect as administrator

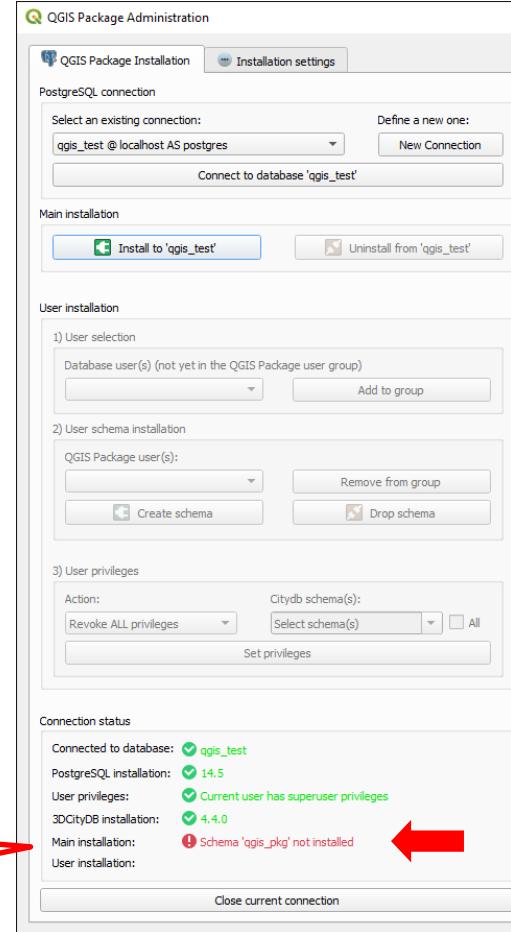


1) Load the "QGIS Package Administration" GUI

3) Uninstall the QGIS Package from the current database

Uninstall QGIS Package

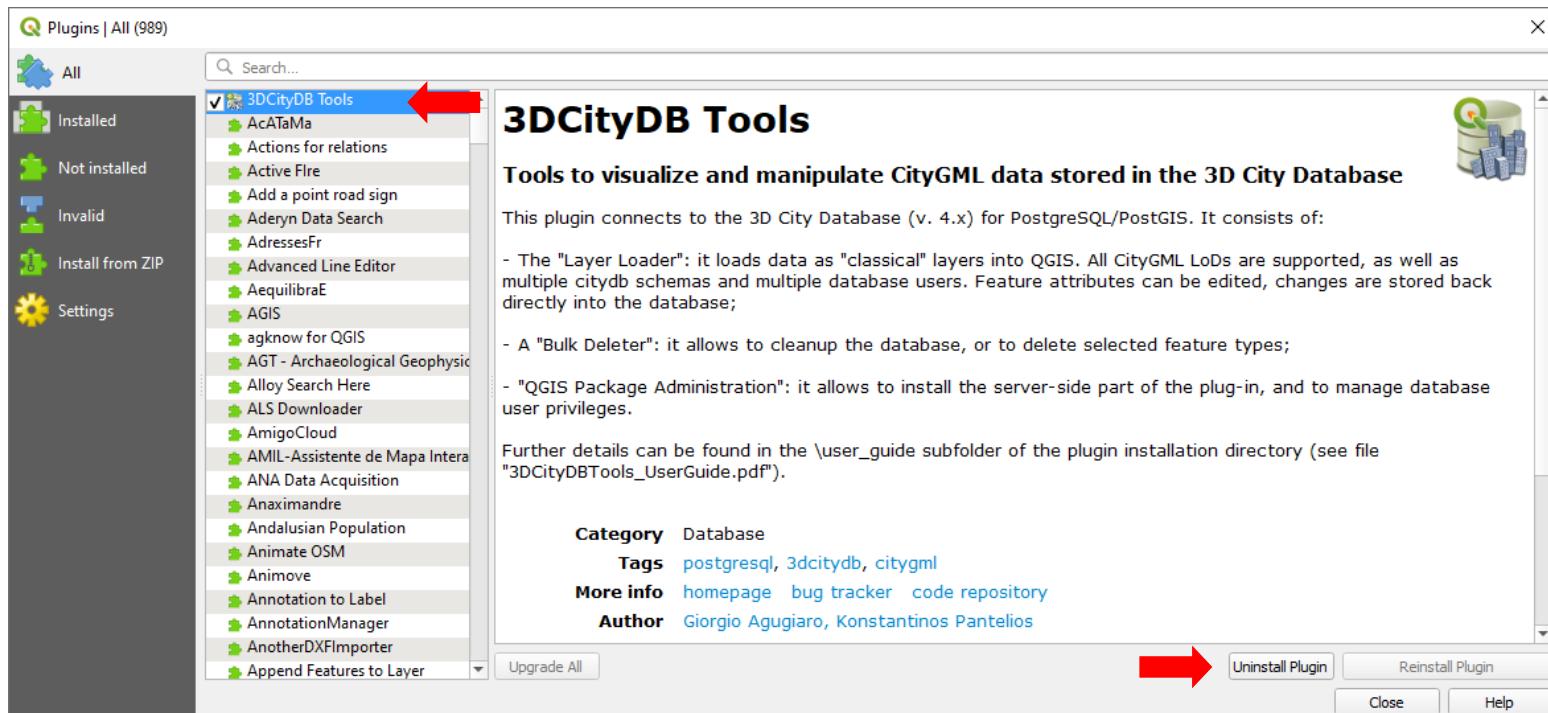
Motivation
Plugin overview
Software installation
Software use
Advanced options
Software uninstall
Current limitations
QGIS Package
Resources
Changelog



Uninstall 3DCityDB-Tools

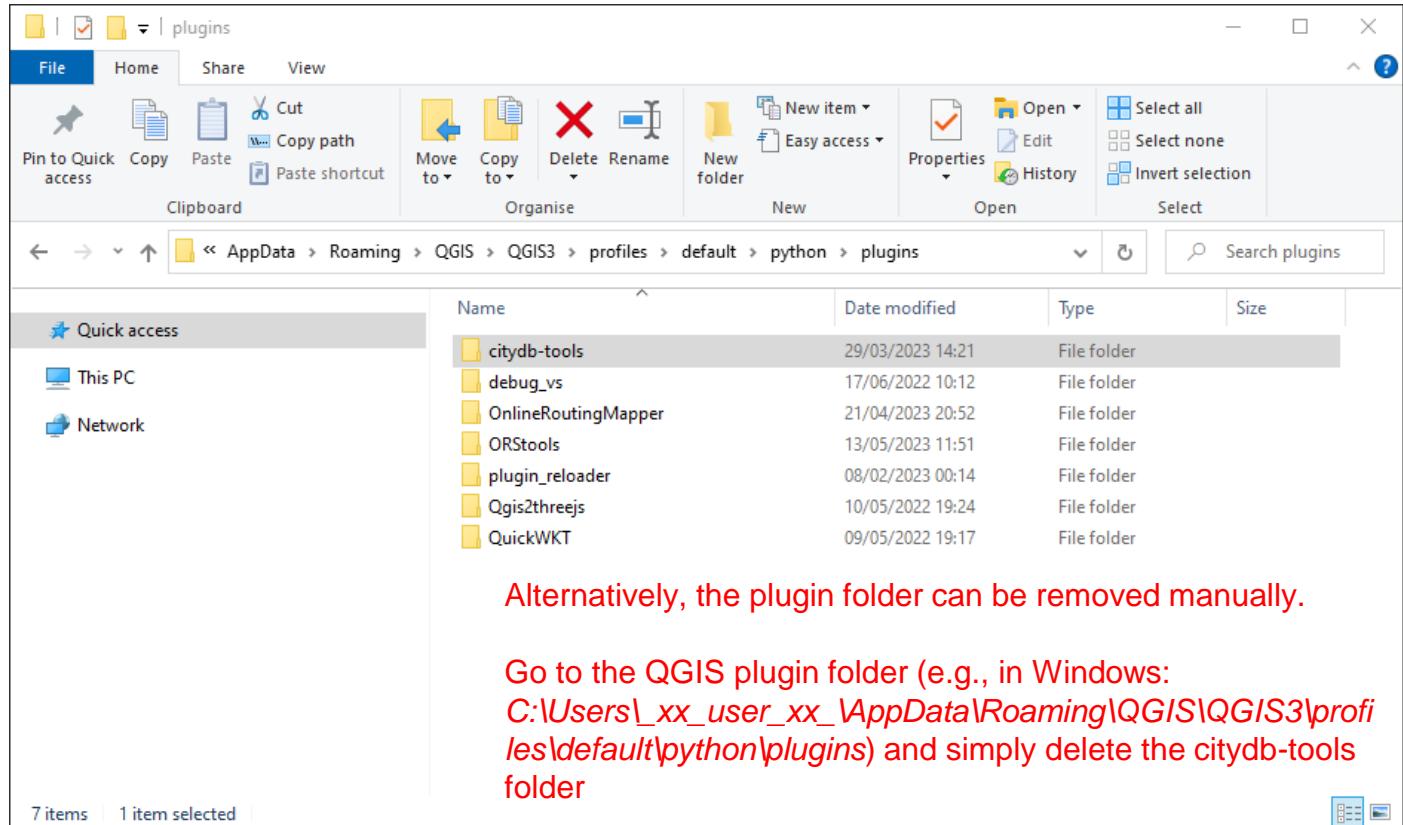
The plugin can be uninstalled from the \Plugins\Manage and Install Plugins... Menu in QGIS

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



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

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
Changelog

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

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

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');
```

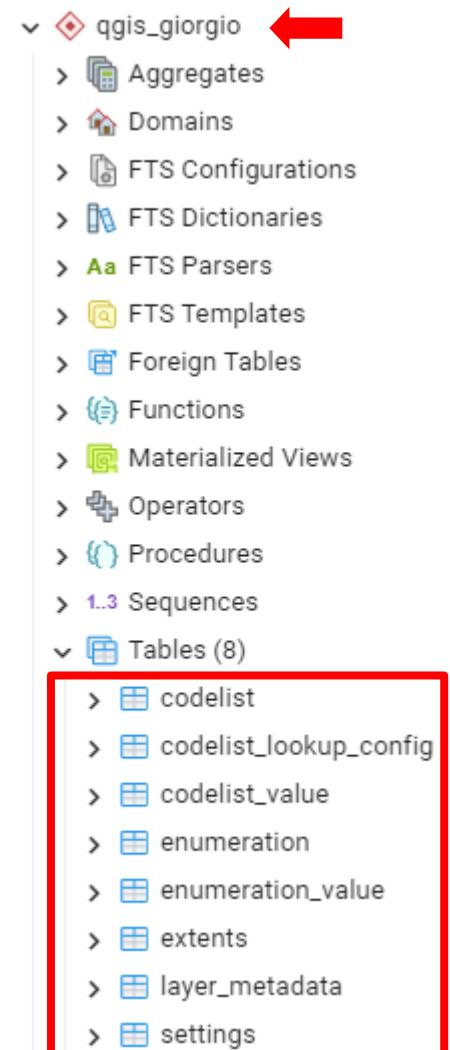
QGIS Package: Create user schemas

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

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
>	Aa FTS Parsers
>	✉ FTS Templates
>	Foreign Tables
>	Functions
>	Materialized Views
>	Operators
>	Procedures
>	1..3 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

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

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
Changelog

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
usr_schema	varchar	The database user schema, e.g. "qgis_giorgio".
cdb_schema	varchar	the citydb schema where data are stored, e.g. "citycb", or "citydb2".
perform_snapping	integer	DEFAULT 0 (i.e. disabled). If 1, geometry simplification is performed. Unused if perform_snapping is set to 0.
digits	integer	DEFAULT 3. Number of decimal positions to keep during geometry simplification. Unused if perform_snapping is set to 0.
area_poly_min	numeric	DEFAULT 0.001 [m ²]. Minimum polygon area during geometry simplification. Unused if perform_snapping is set to 0.
bbox_corners_array	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> !
is_geographic	boolean	DEFAULT False. True is the coordinate system of the citydb schema has geographic coordinates, False if it has projected coordinates.
force_layer_creation	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
Changelog

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 m² 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
Changelog

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

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

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

Function

qgis_pkg.refresh_layers(usr_schema, cdb_schema)

Parameter	Type	Description
usr_schema	varchar	The database user schema, e.g. "qgis_giorgio".
cdb_schema	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
Changelog

Function

qgis_pkg.drop_layers(usr_schema, cdb_schema)

Parameter	Type	Description
usr_schema	varchar	The database user schema, e.g. "qgis_giorgio".
cdb_schema	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.)

Optionally, they can be created programmatically 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.

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
Changelog

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');

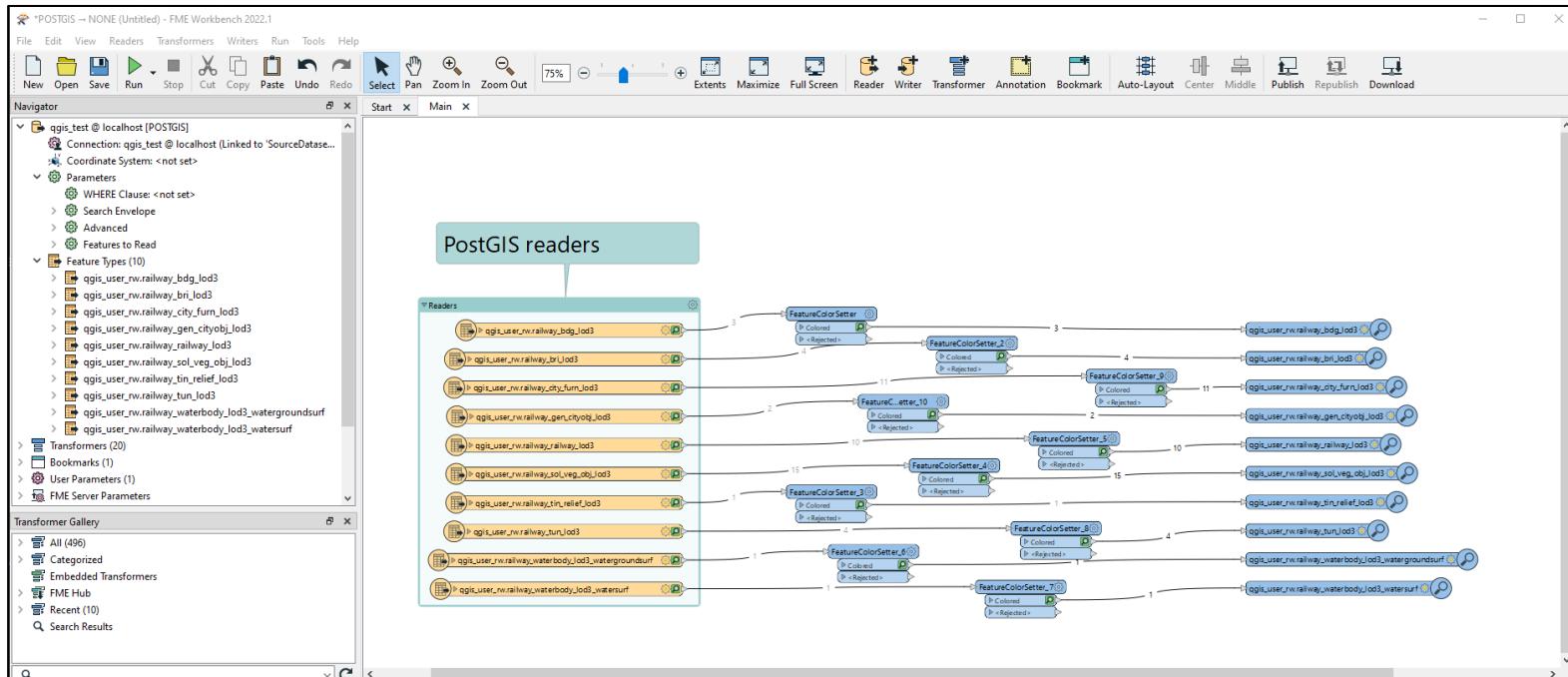
...
-- Then drop the user schema
DROP SCHEMA qgis_giorgio CASCADE;
-- Optionally (if necessary) remove user "giorgio" from the "qgis_pkg_usrgroup_qgis_test" associated to database
-- "qgis_test"
REVOKE qgis_pkg_usrgroup_qgis_test FROM giorgio;
```

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**

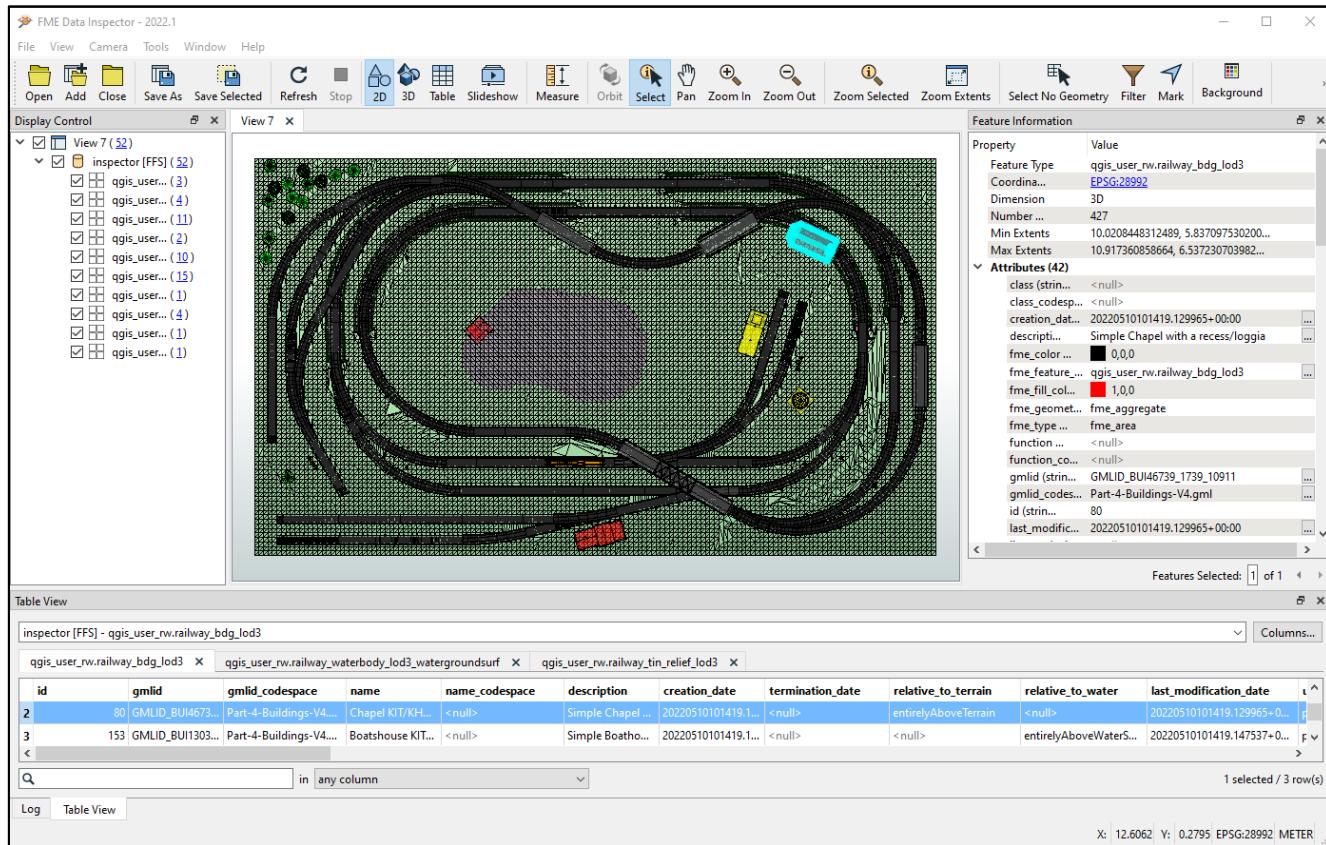
Motivation
 Plugin overview
 Software installation
 Software use
 Advanced options
 Software uninstall
 Current limitations
QGIS Package
 Resources
 Changelog



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
 Changelog



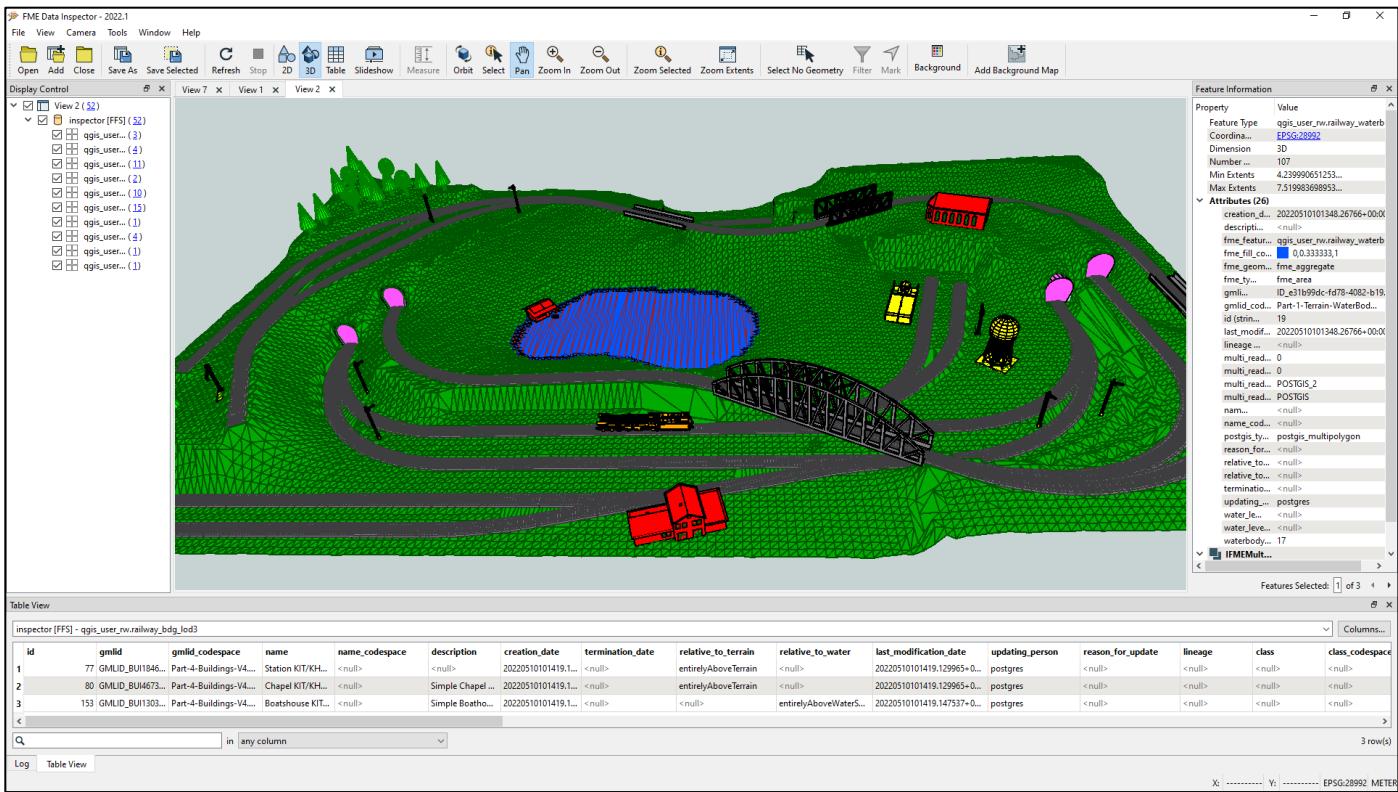
The screenshot shows the FME Data Inspector application window. The main area displays a 2D map of a railway network with various tracks and buildings. The map is overlaid with a grid pattern. On the left, a tree view shows multiple layers under 'View 7 (52)', including 'inspector [FFS]' which contains several sub-layers labeled 'qgis_user...' followed by numbers from 1 to 15. The top menu bar includes File, View, Camera, Tools, Window, Help, and various toolbar icons for selection, measurement, and navigation. To the right, there are two panes: 'Feature Information' showing properties like Feature Type (qgis_user_rw.railway_bdg_lod3), Coordinate System (EPSG:28992), Dimension (3D), and Number (427); and 'Attributes (42)' listing columns such as class, class_codespace, creation_date, description, fme_color, fme_fill_color, fme_geometries, fme_type, function, gmlid, gmlid_codespace, id, last_modification_date, name, name_codespace, and termination_date. At the bottom, a 'Table View' pane shows a detailed table for the selected feature, with columns including id, gmlid, gmlid_codespace, name, name_codespace, description, creation_date, termination_date, relative_to_terrain, relative_to_water, and last_modification_date. The table currently displays two rows of data.

Table View											
inspector [FFS] - qgis_user_rw.railway_bdg_lod3											
qgis_user_rw.railway_bdg_lod3			qgis_user_rw.railway_waterbody_lod3_watergroundsurf			qgis_user_rw.railway_tin_relief_lod3					
id	gmlid	gmlid_codespace	name	name_codespace	description	creation_date	termination_date	relative_to_terrain	relative_to_water	last_modification_date	
2	80	GMLID_BU14673...	Part-4-Buildings-V4...	Chapel KIT/KH...	<null>	Simple Chapel...	20220510101419.1...	<null>	entirelyAboveTerrain	<null>	20220510101419.12995+0...
3	153	GMLID_BU1130...	Part-4-Buildings-V4...	Boathouse KIT...	<null>	Simple Boatho...	20220510101419.1...	<null>	entirelyAboveWaterS...	20220510101419.147537+0...	

QGIS Package via FME

- 3D visualisation via FME Data Inspector

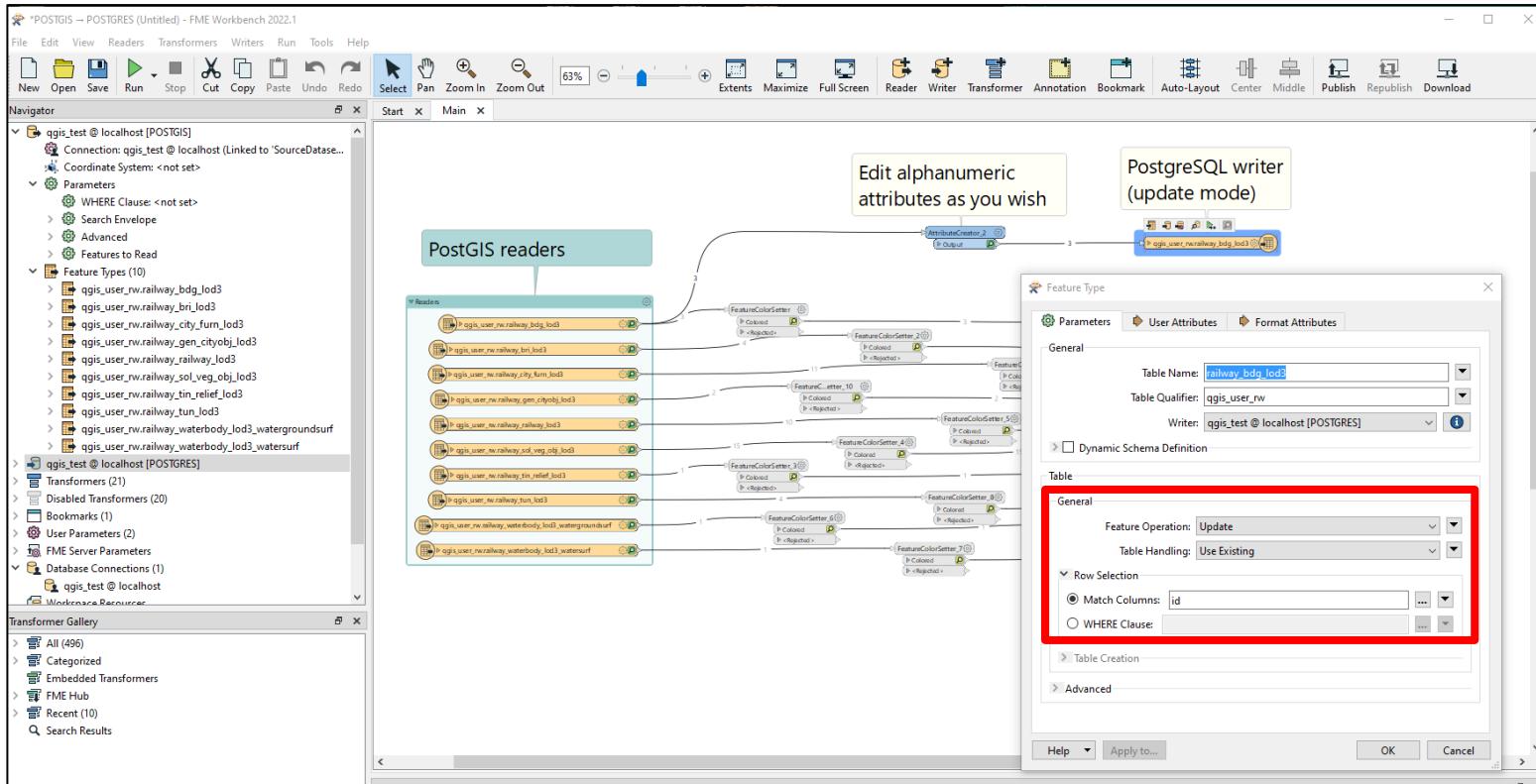
Motivation
 Plugin overview
 Software installation
 Software use
 Advanced options
 Software uninstall
 Current limitations
QGIS Package
 Resources
 Changelog



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
 Changelog



Motivation

Plugin
overview

Software
installation

Software
use

Advanced
options

Software
uninstall

Current
limitations

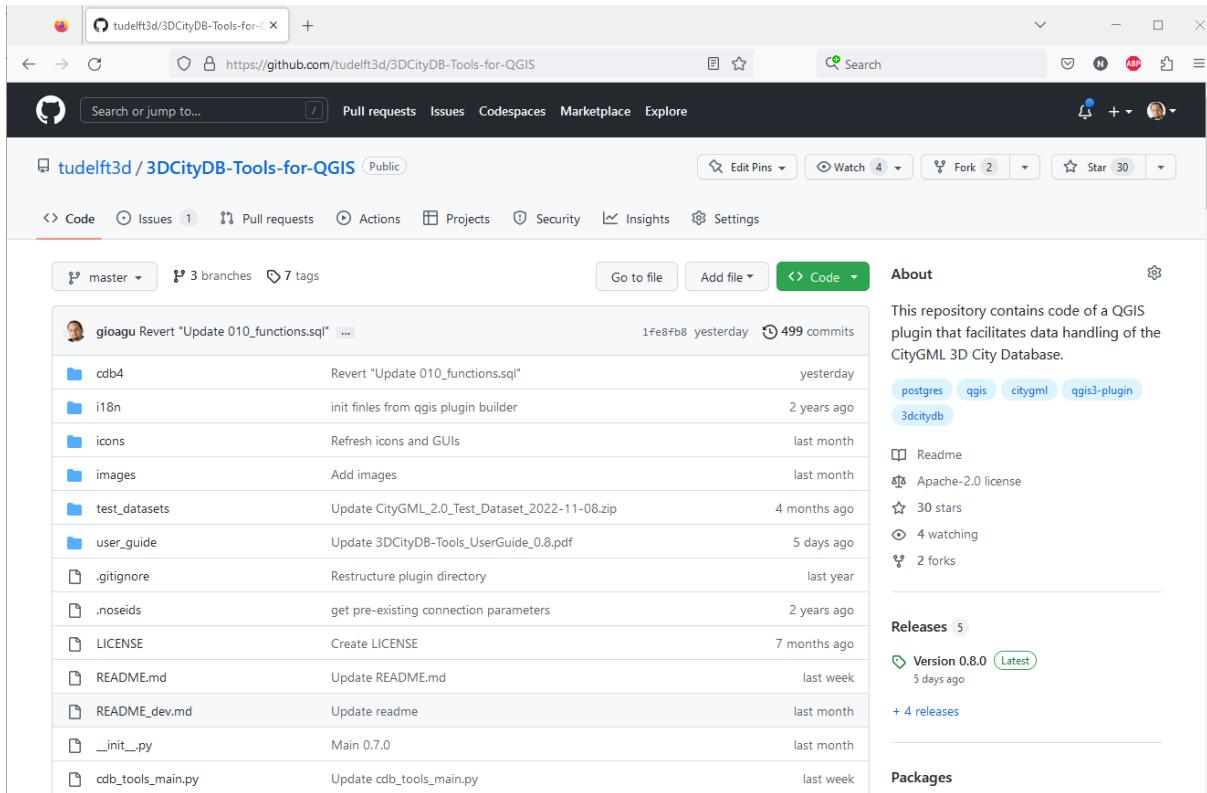
QGIS
Package

Resources

Resources

Source code and GitHub repository

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



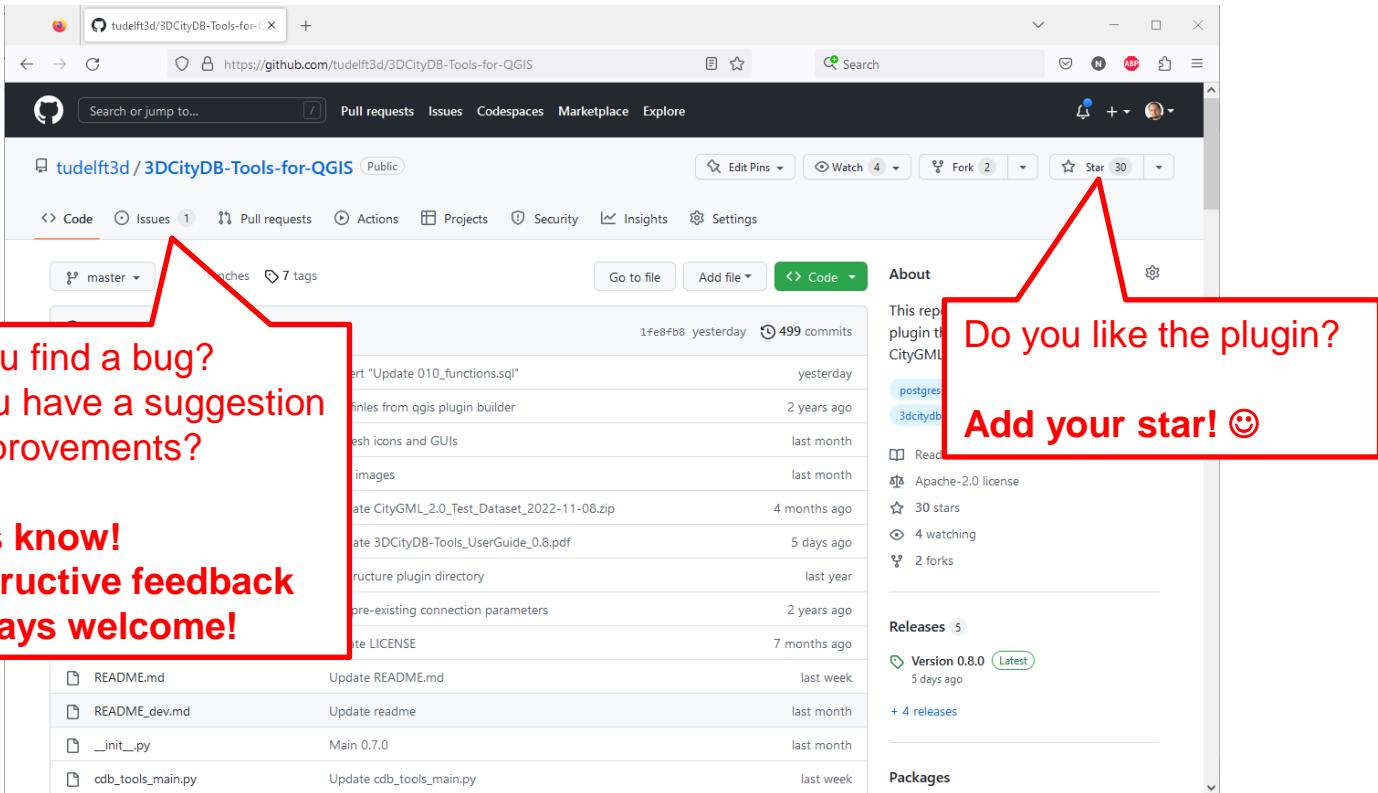
The screenshot shows the GitHub repository page for `tudelft3d / 3DCityDB-Tools-for-QGIS`. The page includes a navigation bar with links for Pull requests, Issues, Codespaces, Marketplace, and Explore. Below the navigation is a search bar and a header with the repository name, its status as Public, and metrics for forks (4), stars (30), and issues (1). The main content area displays a list of recent commits, showing changes made by users like `gioagu`, `cdb4`, and `i18n`. The commits are dated from yesterday to last month. To the right of the commit list is an 'About' section describing the repository as containing code for a QGIS plugin to handle CityGML 3D City Database data. It also lists tags such as `postgis`, `qgis`, `citygml`, `qgis3-plugin`, and `3dcitydb`. The 'Releases' section shows a latest version of 0.8.0 and four other releases. The 'Packages' section is also visible at the bottom.

Resources

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

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! ☺

GitHub repository details:

- Repository: tudelft3d / 3DCityDB-Tools-for-QGIS (Public)
- Code: master
- Issues: 1
- Pull requests: 0
- Actions: 0
- Projects: 0
- Security: 0
- Insights: 0
- Settings: 0
- Go to file: Add file ▾
- Code: ▾
- 1fe8fb8 yesterday 499 commits
- Update "010_functions.sql" yesterday
- binaries from qgis plugin builder 2 years ago
- fresh icons and GUIs last month
- images last month
- update CityGML_2.0_Test_Dataset_2022-11-08.zip 4 months ago
- update 3DCityDB-Tools_UserGuide_0.8.pdf 5 days ago
- structure plugin directory last year
- more-existing connection parameters 2 years ago
- update LICENSE 7 months ago
- 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

About

This repository plugin to support QGIS with CityGML

postgreSQL
3dcitydb

Apache-2.0 license

30 stars
4 watching
2 forks

Releases 5

Version 0.8.0 (Latest)
5 days ago

+ 4 releases

Packages

Resources

QGIS Plug-in repository

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



The screenshot shows a web browser 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 brief description: "Tools to visualize and manipulate CityGML data stored in the 3D City Database". It also has tabs for "About", "Details", and "Versions". Below the card, there is a list of features: "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;". Another feature is listed: "A 'Bulk Deleter': it allows to cleanup the database, or to delete selected feature types;". A third feature is listed: "'QGIS Package Administrator': it allows to install the server-side part of the plug-in, and to manage database user privileges.". At the bottom, a note 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."

Resources

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

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/>

Changelog

ChangeLog for version 0.8.4

Main changes

- Minor bug fixes
- Layer Loader
 - Corrected new connections not being updated in Layer Loader dialog
- QGIS Package (version 0.10.2)
 - WaterBody module: Added support for MultiCurve LoD0-1 layers
 - Transportation and Vegetation modules: Corrected minor bugs
 - Added CityGML 2.0 Codelists for WaterBody module
 - Some code refactoring
- Test datasets
 - Updated Railway dataset to contain WaterBody LoD1 MultiCurve
- Installation and user guide (this document)
 - QGIS Package: Corrected/enhanced examples with functions to set user's privileges

Changelog

ChangeLog for version 0.8.3

Main changes

- The plugin is not anymore "experimental"; thus, it can be installed by anybody from QGIS
- Minor edits to the Layer Loader dialog GUI
 - Moved Codelist selection combo box from Settings tab to Layers tab
- Updated this document
 - Added installation from QGIS Plugins repository
 - Updated figures with new Layer Loader dialog GUI
 - QGIS Package: Corrected SQL examples to refresh views
 - QGIS Package: Added SQL examples for detail views
 - QGIS Package: Added SQL examples to upsert the user-defined bounding box extents
 - Corrected some typos and other minor errors

Changelog

ChangeLog for version 0.8.2

Main changes

- Minor internal changes to comply with the QGIS Plugin online repository

ChangeLog for version 0.8.1

Main changes

- Layer loader: Detail View layers are now ro/rw depending on user privileges
- QGIS Package: Version 0.10.1

Bug fixes

- QGIS Package: Fixed bug to set user privileges

Changelog

ChangeLog for version 0.8.0

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

Client-side: Layer Loader

- Added support for Address features as layers
- Added support for LoD-independent layers (i.e., "LoDx")
- Redesigned and extended attribute forms to support:
 - External References (per type)
 - Generic Attributes (per type)
 - Addresses
 - All subforms allow insert, update and delete operations
- Added support for CodeLists selection via GUI
- Improved support of datasets with geographic coordinates (bbox)

Client side: Bulk Deleter

- Improved support of datasets with geographic coordinates (bbox)
- Renamed root-class to top-level features

Client-side: QGIS Package Administration

- Minor bug fixes and code clean-up

Changelog

Changelog for version 0.8.0 (ctd)

Server-side (QGIS Package):

- New version 0.10
- Updates to layer_metadata table
- Updates to layer creation functions
- Added support for (updatable) Detail Views
- Added support for look-up tables metadata
- Improved support to datasets with geographic coordinates (bbox precision)
- Improved speed to compute the bbox and list cdb_schemas
- Minor code clean-up and bug fixes

Changelog

Changelog for version 0.7.1

Main changes

- Code restructuring to allow for better modularization of current (and future) modules
- Allow for concurrent dialogs, but added logic to:
 - Prevent having user dialogs (and connections) open when using the "QGIS Package Administration"
 - Prevent having user dialogs connected to the same database *and* citydb schema at the same time
- Ships with QGIS Package v. 0.9.1

Bug fixes

- When cleaning up the database, sequences are now correctly restarted

Changelog

Changelog for version 0.7.0

The 3DCityDB-Loader is growing and gets therefore a new name: **3DCityDB-Tools**

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

Server-side (QGIS Package):

- Version 0.9.0
- Each 3DCityDB instance gets not its own database group
- Added support for new client-side functionalities:
 - Refactored layer_metadata table
 - Added functions to manage groups
 - Added functions to check Feature Types availability
- Minor code clean-up and bug fixes

Client-side: Admin GUI

- QGIS Package Installation tab: Major rework, added user management
- Installation settings tab (NEW)
- Minor GUI improvements to the New connection dialog
- QGIS Package uninstall procedure is now much faster
- User schema uninstall procedure is now much faster
- Rewritten all SQL queries to follow psycopg2 syntax
- Minor code refactoring and clean-up

Changelog

Changelog for version 0.7.0 (ctd)

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

Client-side: Loader GUI

- Connection tab
 - Added Feature Type selection in layer generation
 - Added Geocoding function (based on Nominatim API)
 - Creating and dropping layers is now faster
 - Minor redesign of the GUI
- Settings tab (NEW)
- Editing of layers from read-only citydb schemas is now blocked in QGIS
- QML files are split and loaded separately for forms, 2D styles and (optionally) 3D styles, respectively
- Rewritten all SQL queries to follow psycopg2 syntax
- Major code refactoring and clean-up

Client side: Bulk Deleter GUI (NEW)

- A user with rw privileges can delete features selecting them:
 - via a spatial filter
 - and/or via a Feature Type/Root-class feature filters
- A user with rw privileges can delete ALL DATA from the selected citydb schema (i.e., truncate all tables)
- Geocoding function (based on Nominatim API)

Changelog

Changelog for **version 0.6.0**

Server-side (QGIS Package)

- Version 0.8.0
- Fixed bug in delete trigger functions to extract the name of the cdb_schema
- Rewritten functions to compute and upsert the extents
- Added layer support for (#8):
 - Terrain Intersection Curve geometries (bridge, building, tunnel, generics, city_furniture)
 - MultiCurve geometries (bridge, building, tunnel)
 - MasspointRelief, BreaklinesRelief Features
- Minor code clean-up and bug fixes

Client-side (Admin):

- Reworked the series of checks carried out upon connection
 - Added check and warning if PostgreSQL < 10.0
 - Added check and warning if QGIS Package < 0.8.0
- Changes to the GUI dialog
 - Reshaped connection buttons
 - Minor other visual improvements (labels, tooltips, etc.)
- Minor code refactoring and clean-up

Changelog

Changelog for version 0.6.0 (ctd)

Client-side (User):

- Added compatibility for QGIS v. 3.28 LTR
- Reworked the series of checks carried out upon connection
 - Added check and warning for outdated QGIS Package versions
 - Added check and warning in case there are no accessible citydb schemas
 - Added check and warning in case there are only empty citydb schemas
- Added functionality to update the cdb_extents in case data are added/removed to/from the citydb schemas
- Solved bug of bboxes being wrongly resized (enlarged) after canvases are set/changed
- Improved function to import layers
 - Drop QGIS spatial filter if QGIS bbox = Layers bbox
 - Added check to avoid loading multiple times the same layer
- Changes to the GUI dialog, forms, etc.
 - Reshaped connection buttons
 - Added button "Refresh {sch} extents"
 - In the "Layer" tab, the Feature types in the combo box are now ordered alphabetically
 - Minor other visual improvements (labels, tooltips, etc.)
 - Added forms for new Relief Features, etc.
- Further major code refactoring and clean-up

Changelog

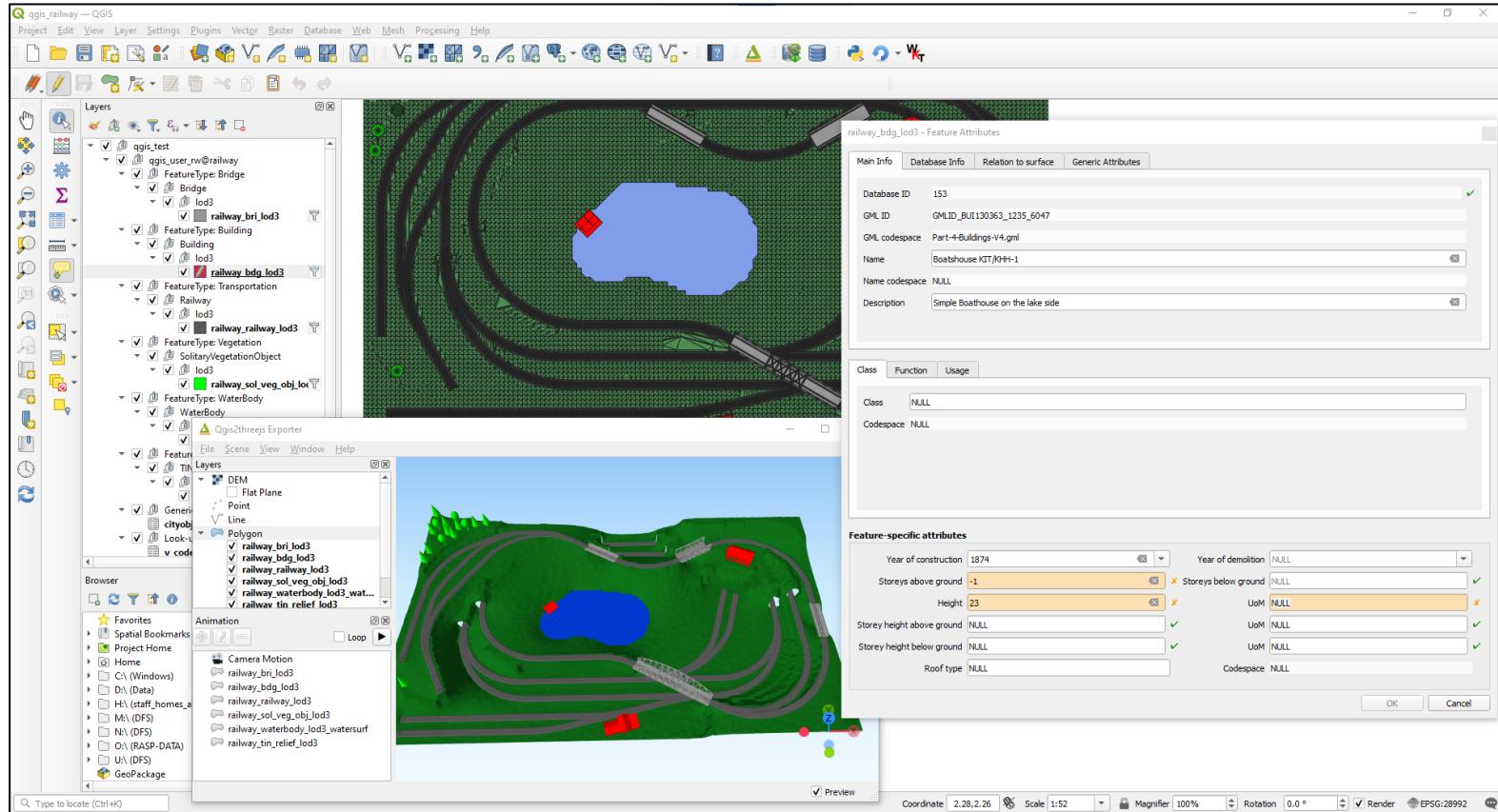
Changelog for **version 0.5.0**

First public release of the 3DCityDB-Loader plugin for QGIS

Main changes

- Major code refactoring and cleaning up
- Minor bug fixes
- User guide documentation updated and extended with FME examples
- Switched to TUD GitHub repository

Enjoy! ☺



Main developers



Dr. Giorgio Agugiaro
g.agugiaro@tudelft.nl

3D Geoinformation Group
TU Delft
The Netherlands
<https://3d.bk.tudelft.nl/gagugiaro>



Konstantinos Pantelios
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