

~~DRAFT~~

CityGML 3D CityDB-Loader plugin for QGIS

A quick overview

Giorgio Agugiaro, Kostantinos Pantelios, Camilo León Sanchez

June 2022

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.

Overview

- Motivation
- Plugin overview
- Installation
- Usage
- Advanced options and settings
- Test datasets

Motivation

IDEA:

- Users could benefit from working with the 3DCityDB "directly"
 - No need to work with files
 - Editing of features could become easier
 - SQL/relational model are rather well-known in the user community
 - Eventually.... "3D city models belong best in a database" 😊

Motivation

BUT:

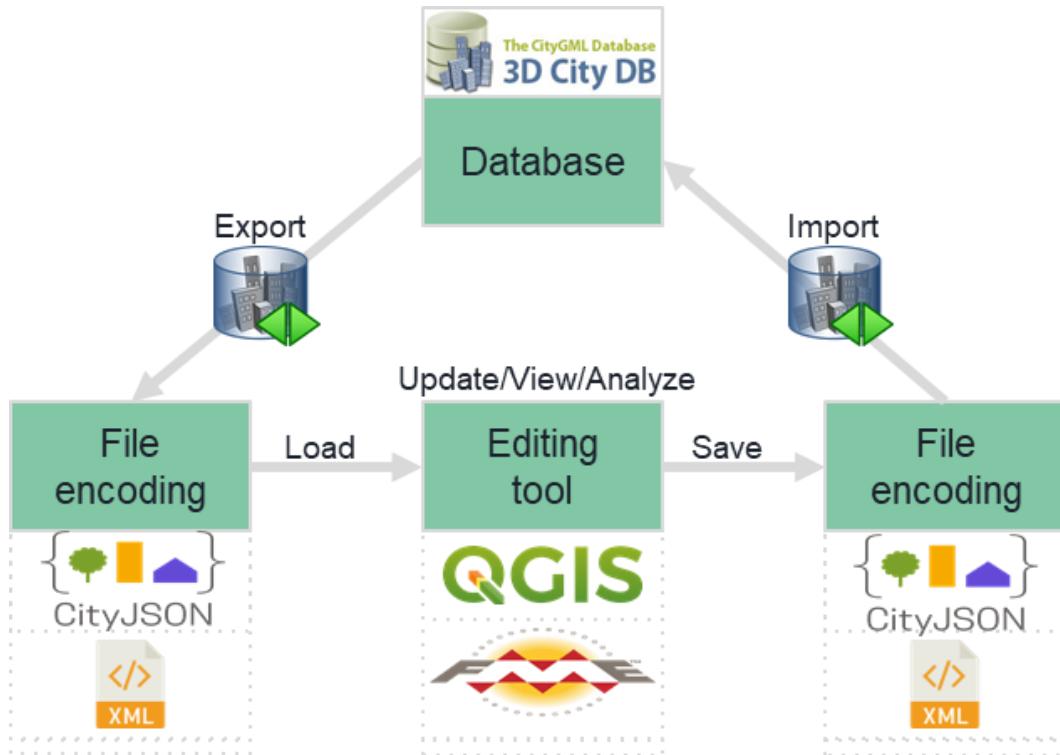
- 3DCityDB structure is rather complex
 - Lots of nested tables, complex structure
 - Data management is complex, although some functions are provided (e.g. delete functions)
 - There can be multiple "scenarios" in the same database instance ("citydb_1, citydb_2, citydb_3, ...)
- CityGML does not follow the Simple Feature Model (SFM)
 - 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

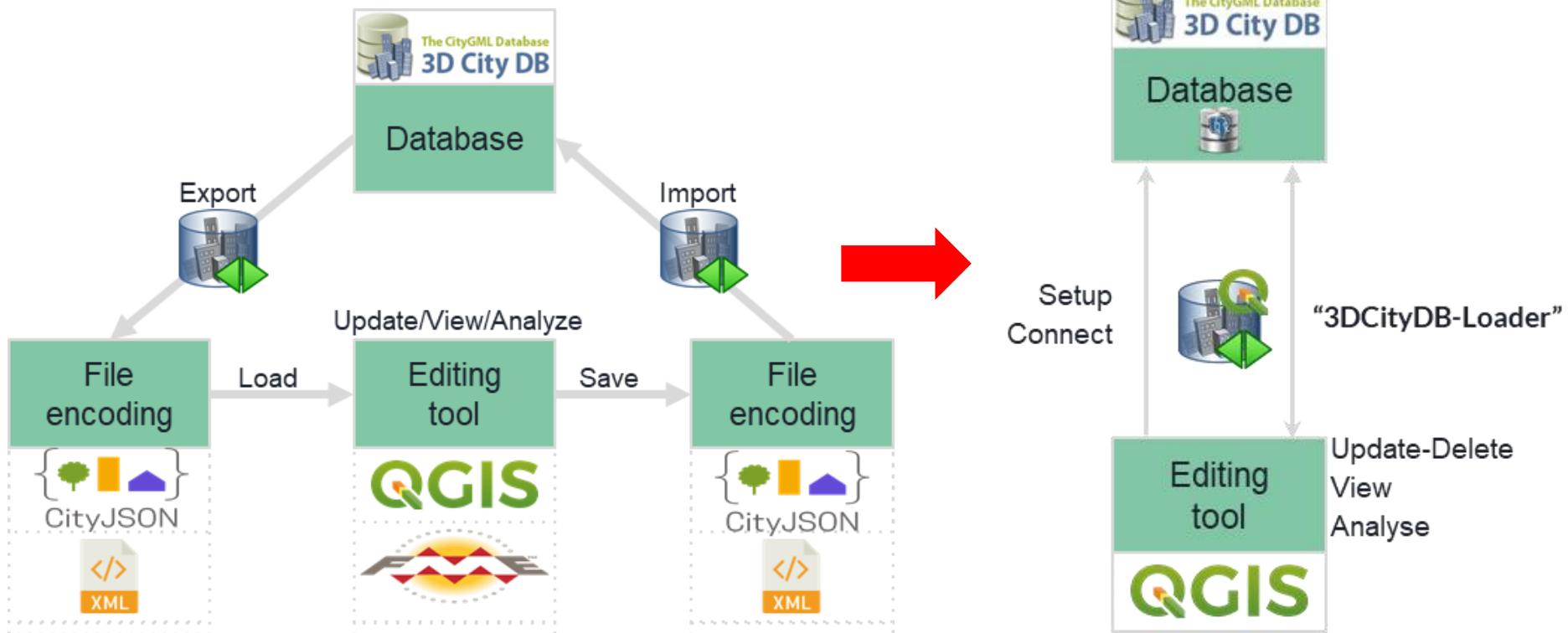
SO:

- Why not using QGIS?
 - Well-known and established open-source software
 - Rather mature, version 3.22 released in autumn 2021, well documented
 - Native support for PostgreSQL/PostGIS, support also for Oracle Spatial
 - Has strong 2D and some (less mature) 3D visualisation functionalities
 - Can be extended with Python-based plugins

Motivation



Motivation



Plugin overview

Main functionalities

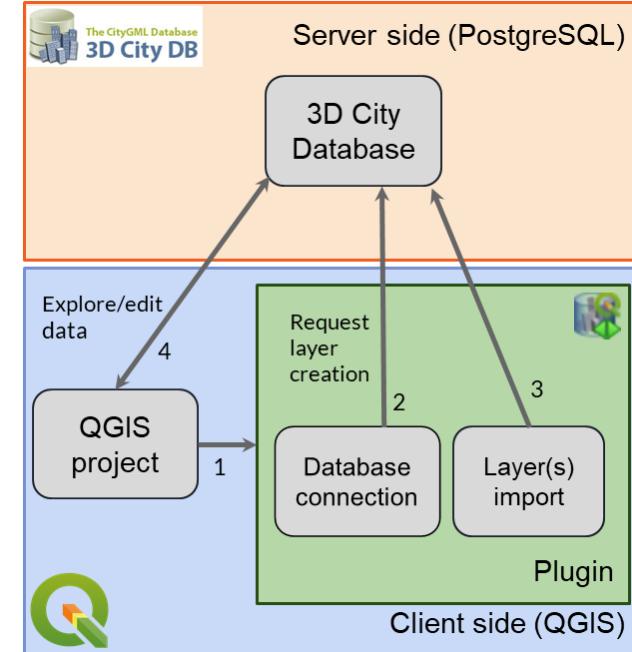
- Create “layers” to hide 3DCityDB complexity when interacting with data
 - Deal with multi-LoD / different geometries / implicit representations
 - Merge all standard attributes of a cityobject into a single “table”
- Support for multiple users with different privileges (read-only, read-write)
- Editing of attributes: possible
- Deletion of features: possible
- Editing of geometries: NOT possible
- Support for multiple 3DCityDB scenarios (aka “citydb” schemas)

Plugin overview

Server-side

PostgreSQL "QGIS Package" ("qgis_pkg")

- Create and manage layers as views (for attributes) linked to materialized views (for geometry) following the SFM
- Manage users and privileges
- Manage multiple scenarios (“citydb” schemas)
- Add default users with ro & rw privileges

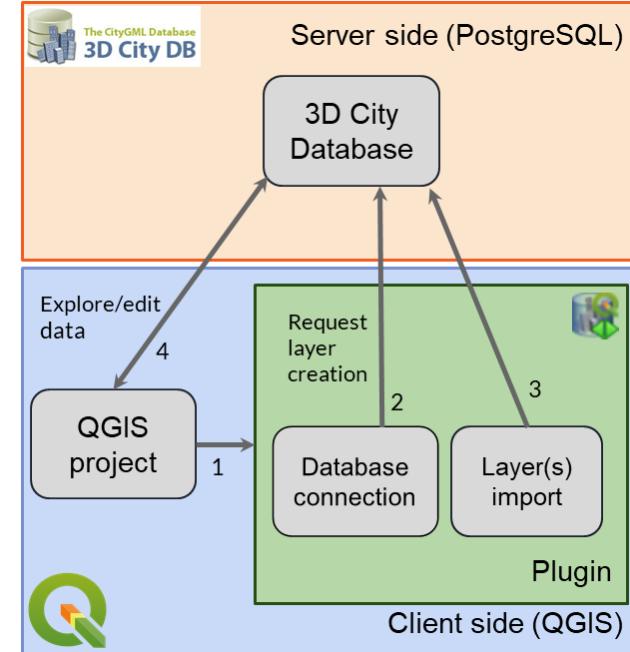


Plugin overview

Client-side

QGIS plugin “3DCityDB-Loader”

- Manage database connections + “qgis_pkg” installation
- GUI-based layer creation and management
- GUI-based management of multiple scenarios
- GUI-based attribute editing via “attribute forms”
- Include CityGML enumerations and children tables (e.g. generic attributes) + optionally codelists
- Create a hierarchical QGIS Table of Contents
- Apply standard colours per feature type



BEWARE

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

In particular you must:

- 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 (aka "scenarios")

Otherwise, here a tutorial: <https://github.com/3dcitydb/tutorials>

Last but not least...

- You need at least a basic knowledge of the main CityGML concepts ☺
- Otherwise, here a crash course: <http://www.urbangeobigdata.it/?p=195>

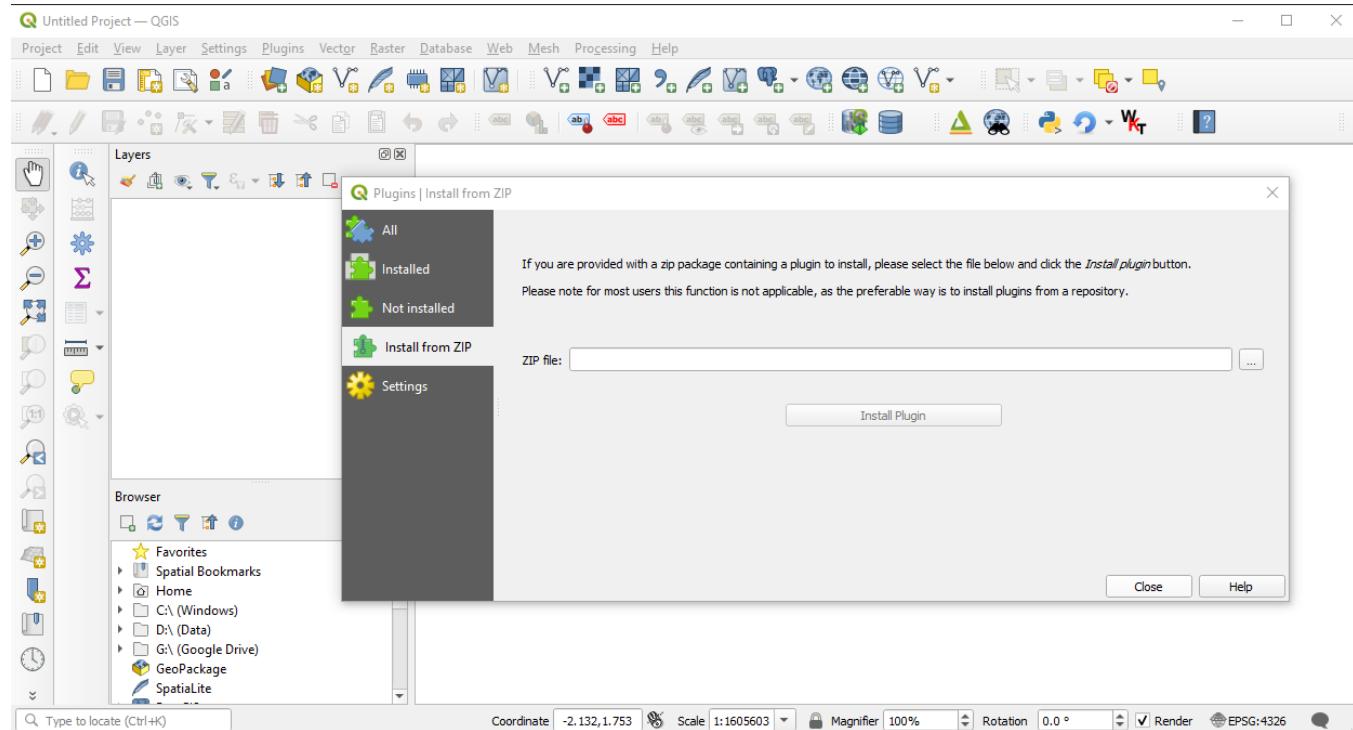
Installation

Software requirements

- CityGML 3D City Database v. 4.x or higher
 - Works on all currently supported versions of PostgreSQL and PostGIS
 - <https://github.com/3dcitydb/3dcitydb-suite/releases>
 - <https://3dcitydb-docs.readthedocs.io/en/latest/>
 - **BEWARE:** 3DCityDB v. 3.x and older NOT supported!
- QGIS 3.22
 - Currently testing on older versions
 - <https://qgis.org/en/site/forusers/download.html>
- PgAdmin (suggested, not required)
 - <https://www.pgadmin.org/download/>

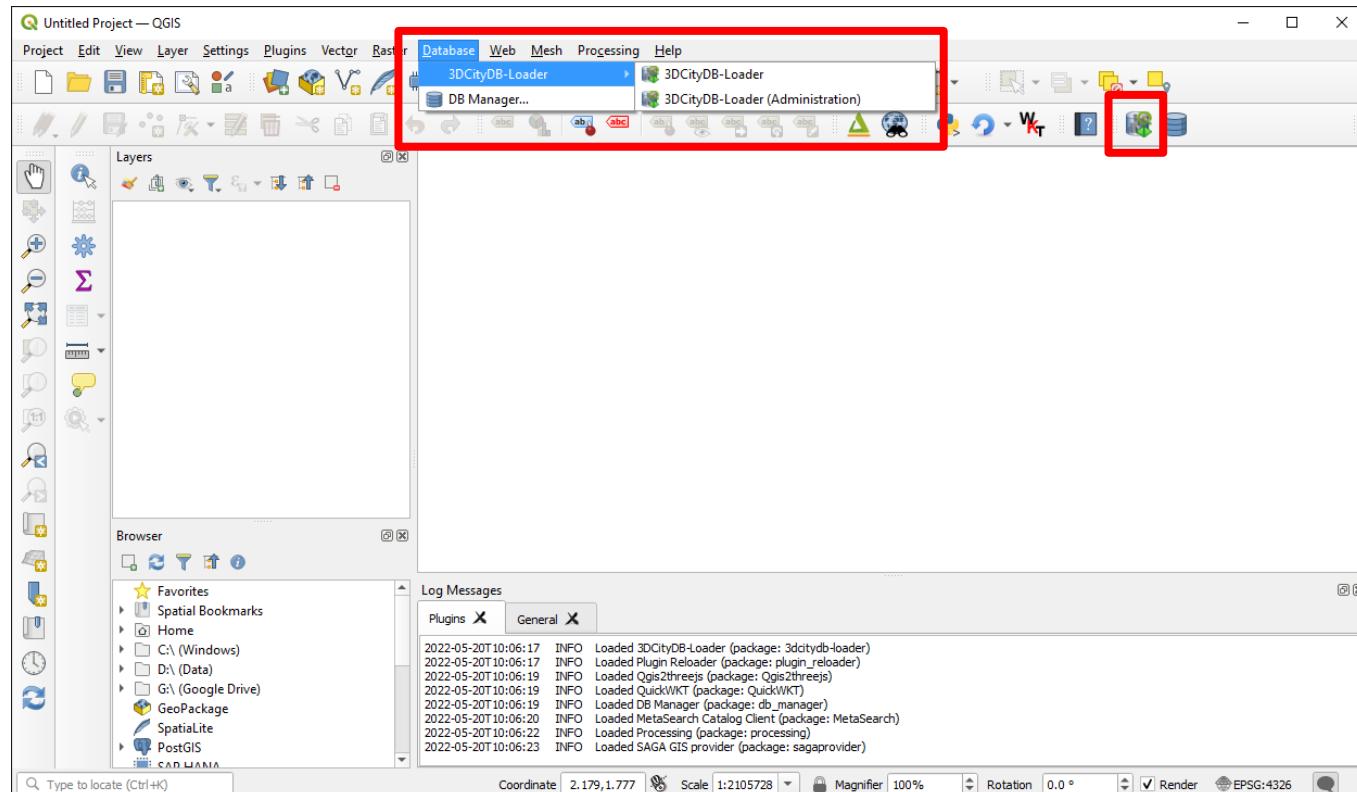
Installation

- The "3DCityDB-Loader" for QGIS is currently available as zip file
- In QGIS, open the Plugins\Manage and install plugins window, and choose "Install from ZIP". Select the provided zip file



Installation

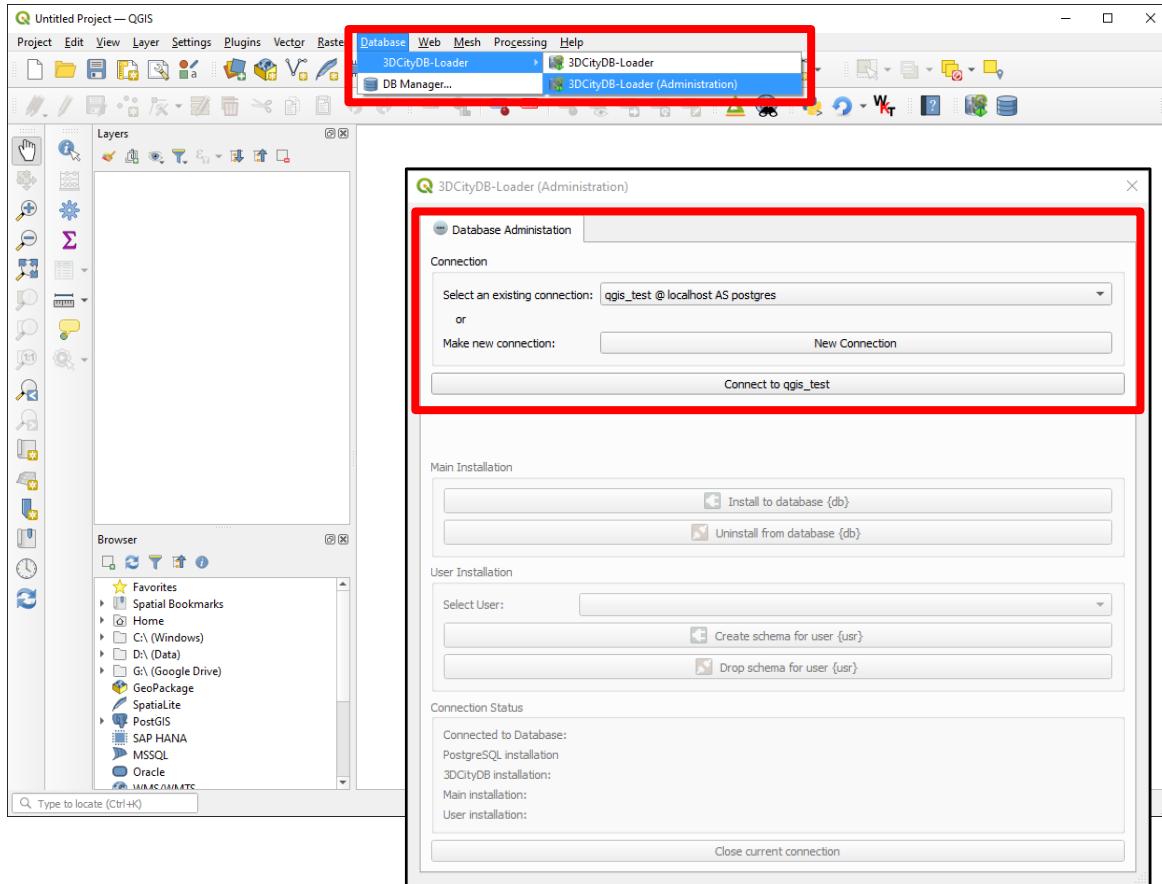
Upon successful installation, the Database menu will contain a new entry, and the icon will be added to the database icon bar



Usage: As database administrator

The database administrator (superuser) is responsible for setting up in advance the server-side part of the plugin for *any* user

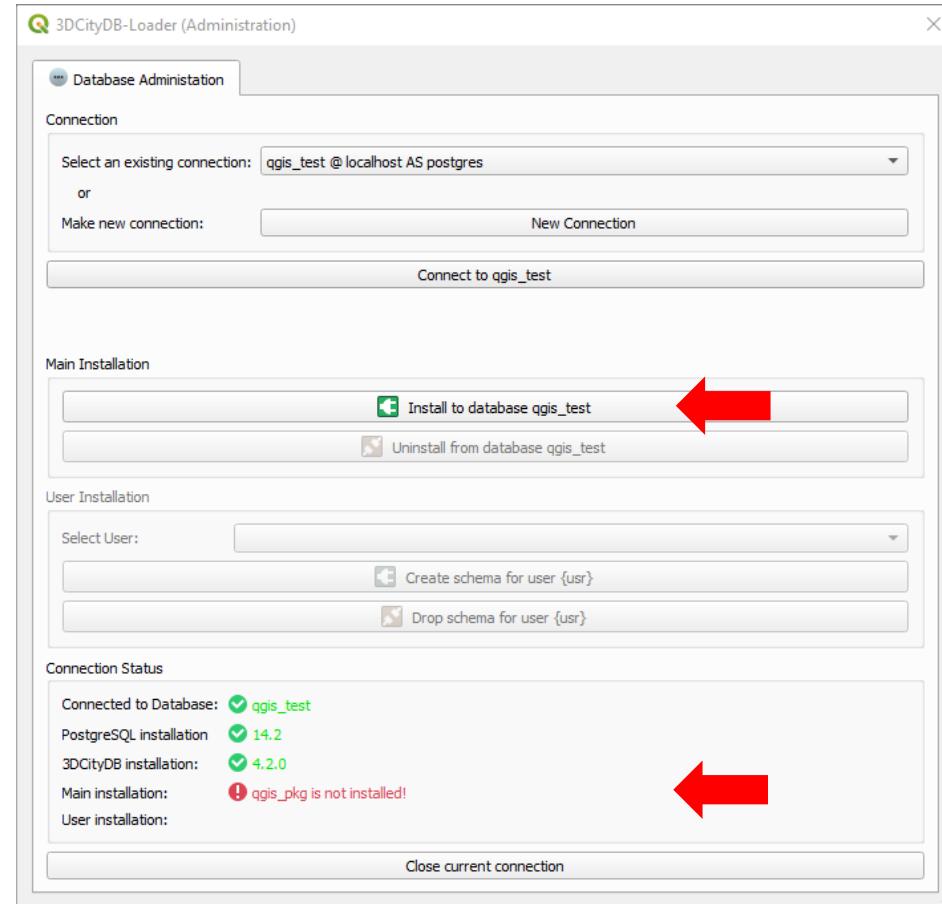
- 1) Create a new connection or use an existing one to the desired 3DCityDB instance (here: "qgis_test")



Usage: As database administrator

- 2) Install the qgis_pk ("Main installation")

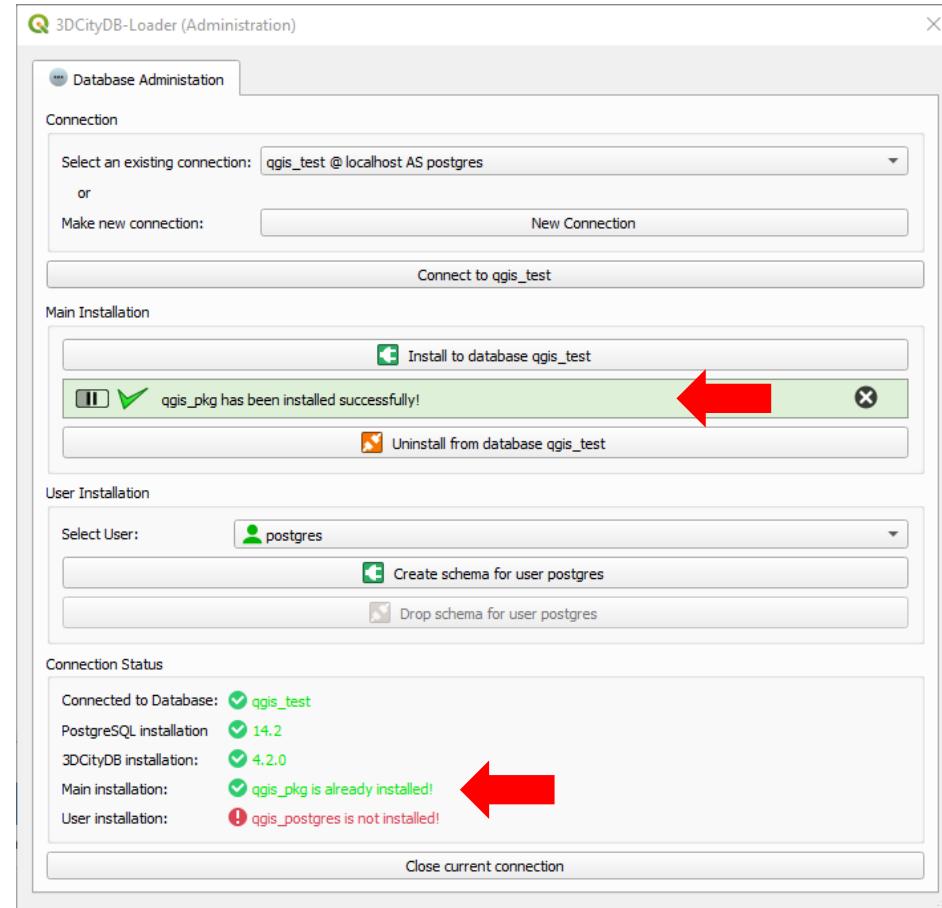
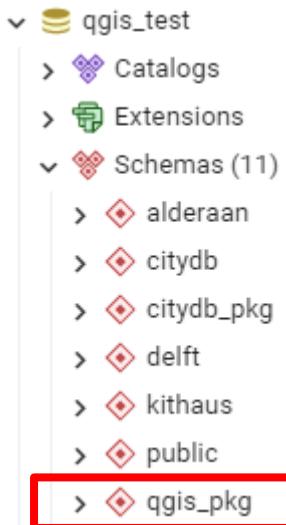
Upon successful installation, in the 3DCityDB, a "qgis_pkg" database schema is created



Usage: As database administrator

2) Install the "qgis_pkg" ("Main installation")

In the 3DCityDB, a "qgis_pkg" database schema is created

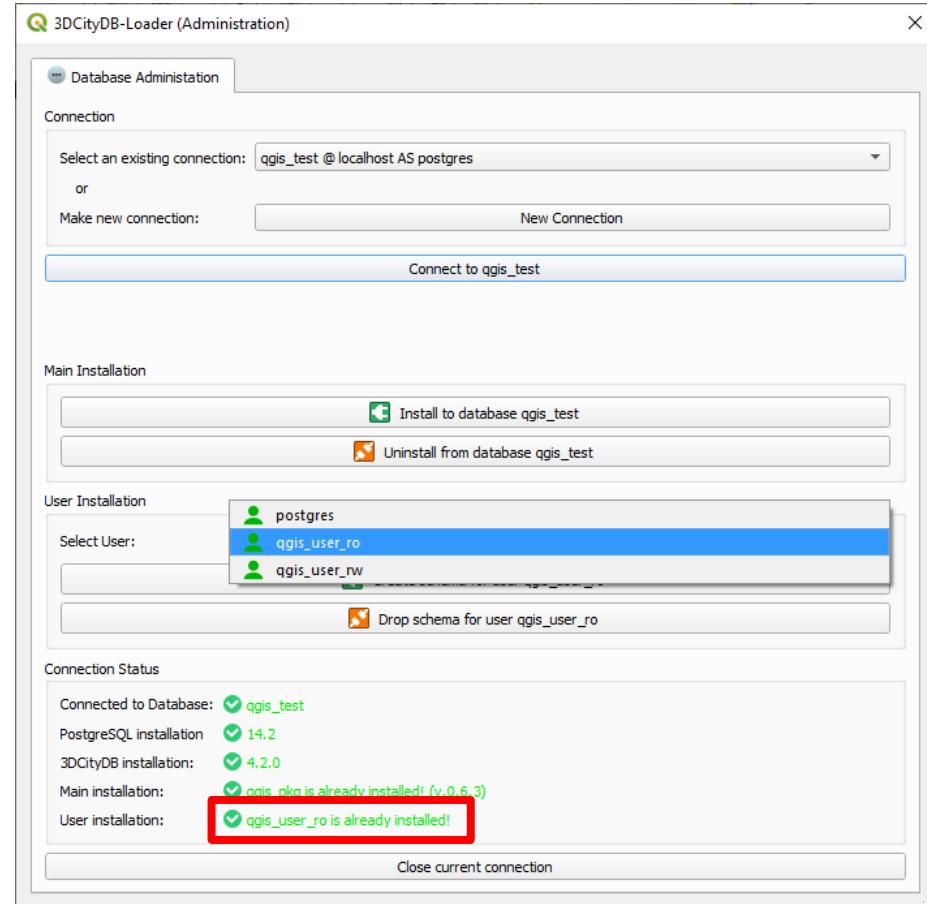


Usage: As database administrator

3) (Optionally) create user schemas.

In order to use the plugin, a user needs a schema named "qgis_" + username

- 2 users are **automatically** added, and their schemas created by default (qgis_user_ro, qgis_user_rw)
- You can create the user schema for database user postgres (NOT recommended!)
- For other database users, see advanced options (later on)



Usage: As database administrator

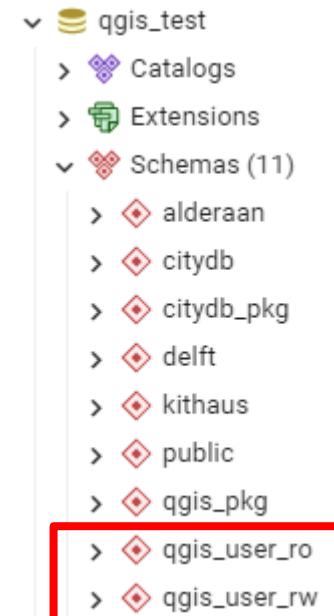
- 3) (Optionally) create user schemas.

In order to use the plugin, a user needs a schema named "qgis_" + username

- 2 users are **automatically** added, and their schemas created by default (qgis_user_ro, qgis_user_rw)
- You can create the user schema for database user postgres (NOT recommended!)
- For other database users, see advanced options (later on)

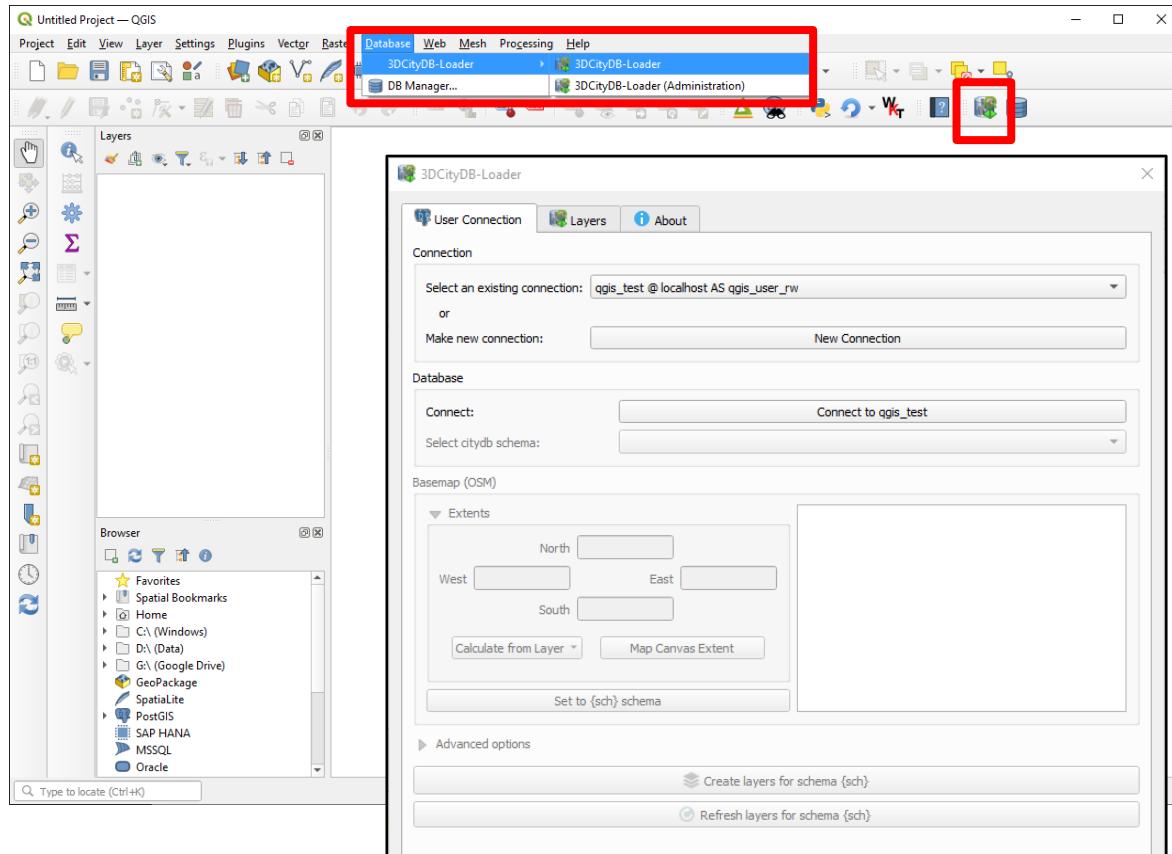
Connection parameter:

- Login/password: "qgis_user_ro"
- Login/password: "qgis_user_rw"



Usage: As user

As "normal" user open the plugin from the Database\3DCityDB-Loader menu or by clicking on the icon



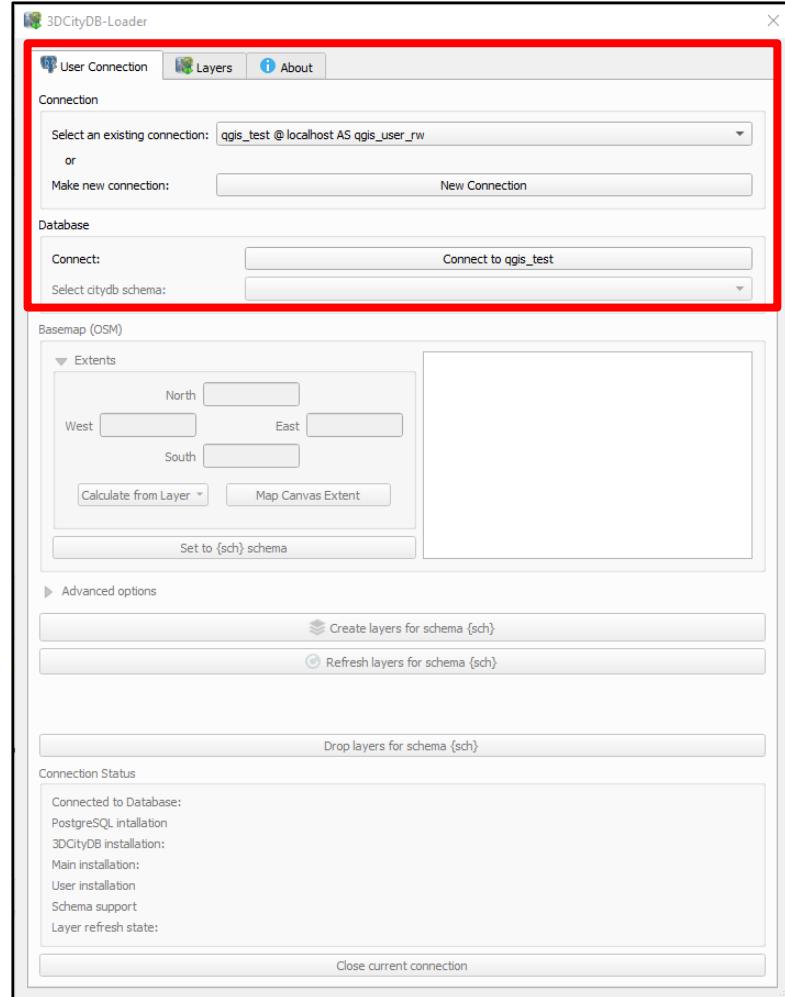
Usage: As user

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" has read-only privileges
 - "qgis_user_rw" has read-write privileges

(You may also connect with your own credentials if the administrator has set up your user schema)



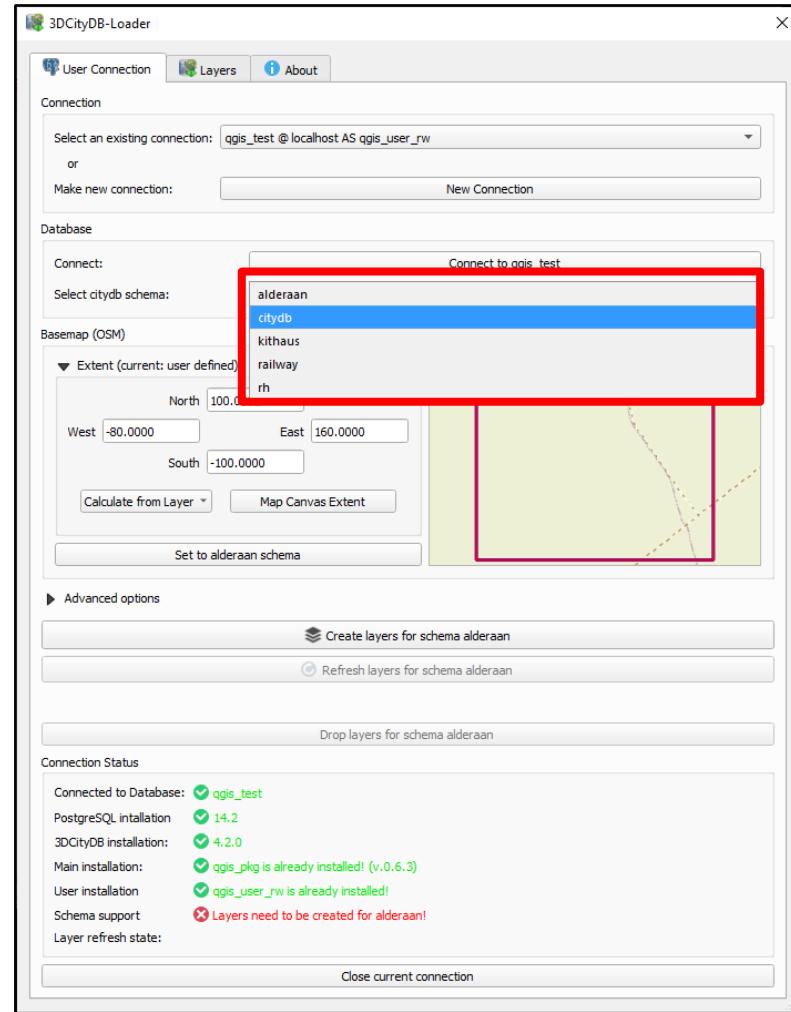
Usage: As user

In the "User connection" tab

- 3) Choose one of the existing scenarios (i.e. citydb schemas). If they contain CityGML data, they will be listed.
The "citydb" one is the most common one

Nota bene: the next slides refer to the "rh" schema

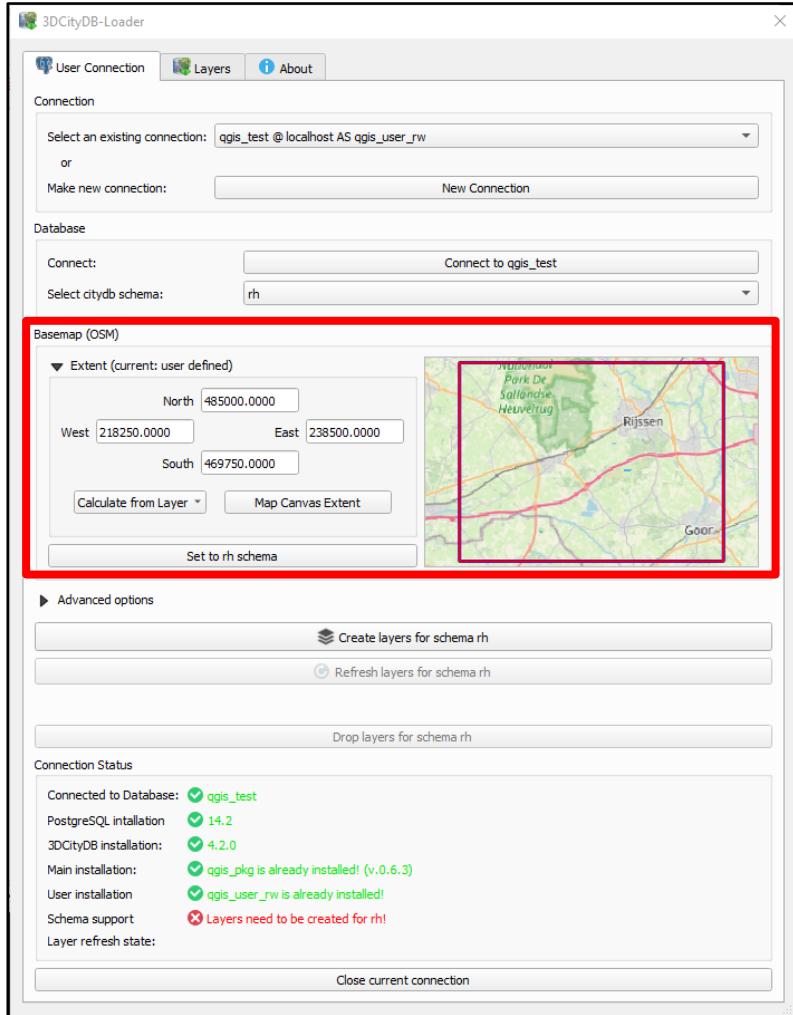
How to create additional schemas
<https://3dcitydb-docs.readthedocs.io/en/latest/3dcitydb/multi-schema.html>



Usage: As user

In the "User connection" tab

- 8) Upon selection of the database schema, you will see the extents of the dataset



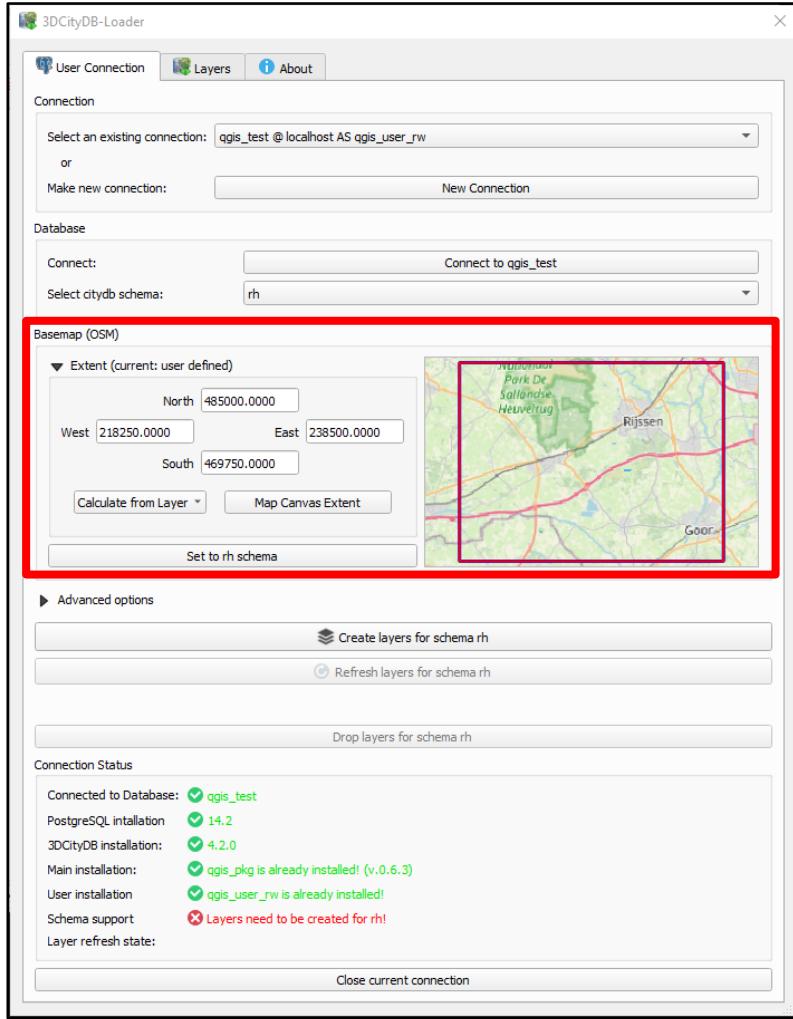
Usage: As user

In the "User connection" tab

- 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

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

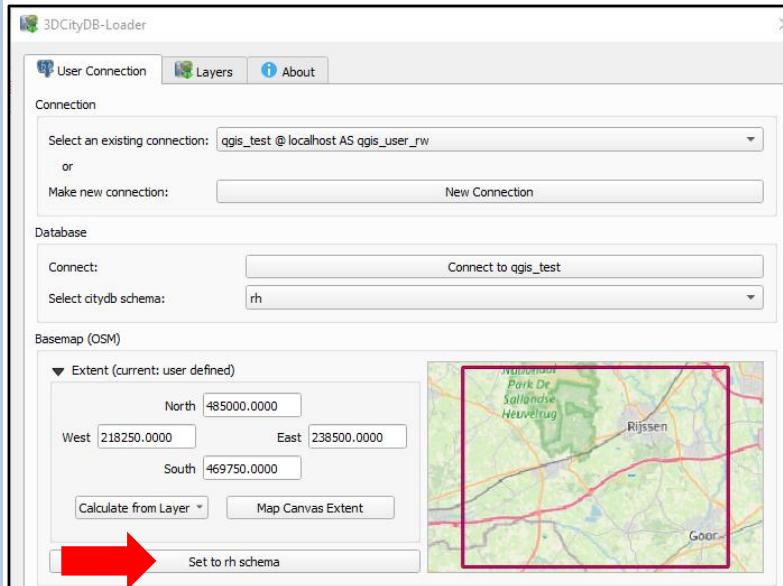
Behind the scenes: In the database, materialized views of the geometries will be generated according to the selected extents. In case of very large cities, it might take a long time (and a lot of space on the server)!



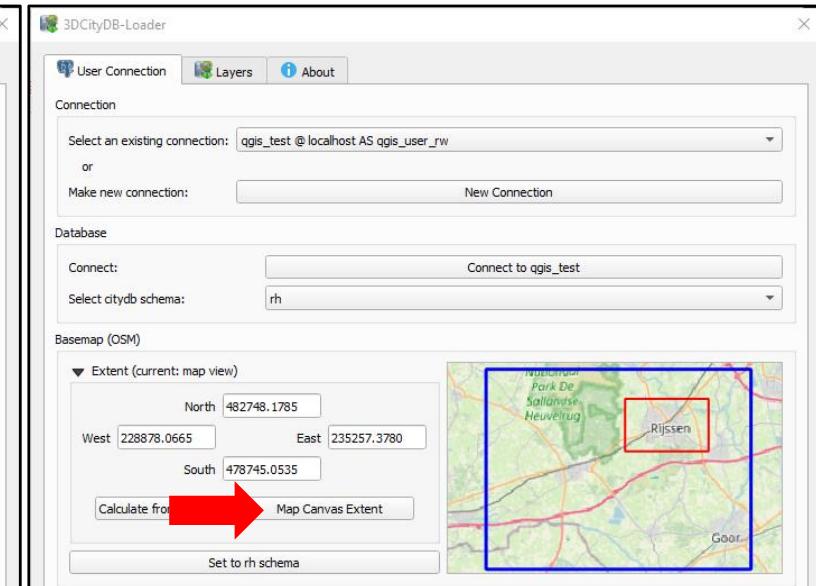
Usage: As user

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

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



Both areas coincide (default)



User-selected layers extents

Usage: As user

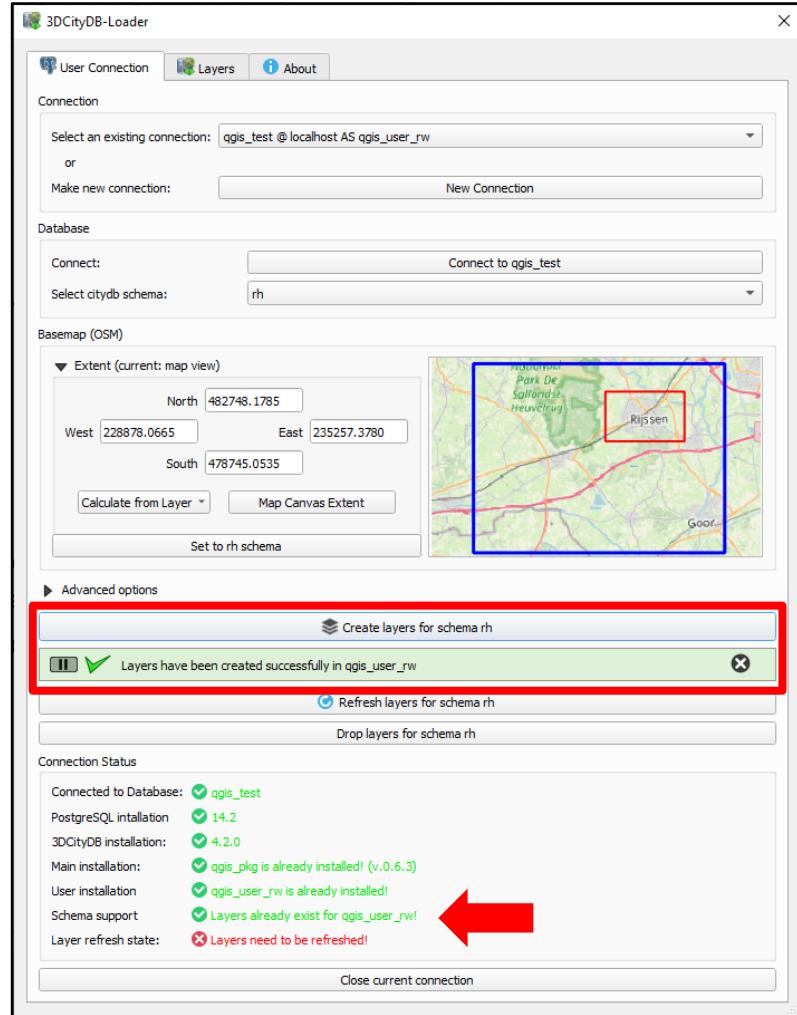
In the "User connection" tab

6) Create the layers

Otherwise, if available, layers created in a previous session can be used

Behind the scenes: In the database, (empty) materialized views of the geometries and views (for the attributes) will be created and joined into layers.

Only layers for *existing* data will be created.



Usage: As user

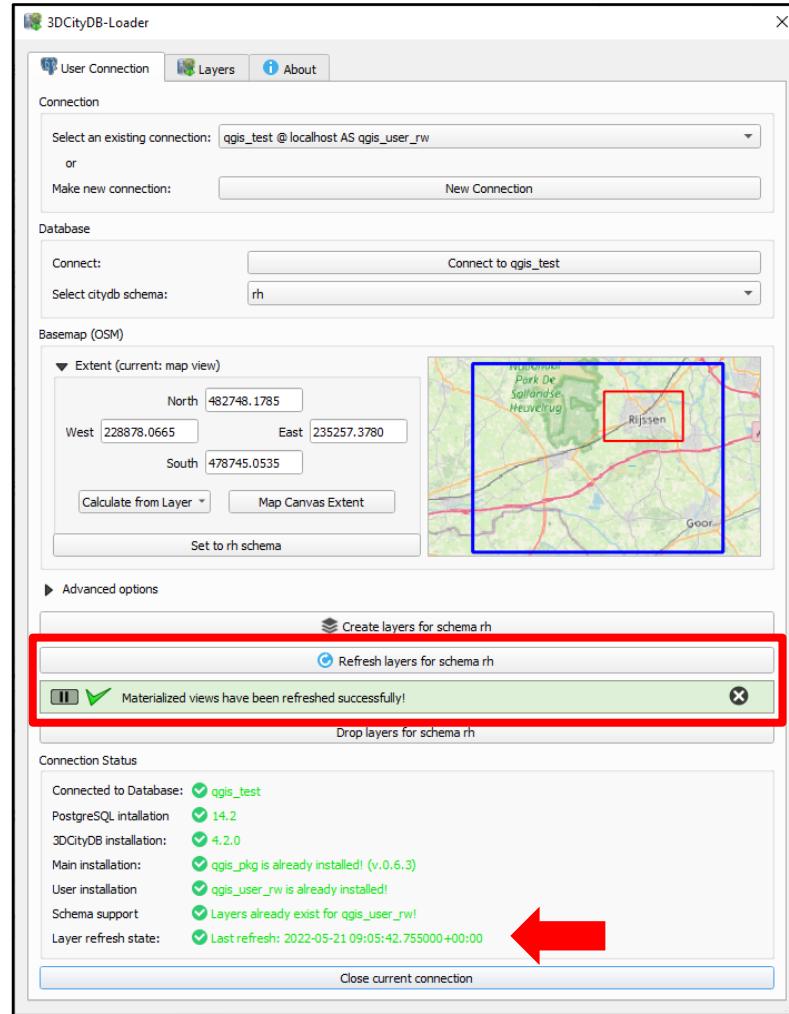
In the "User connection" tab

7) Populate/refresh the layers

Otherwise, if available, layers created in a previous session can be used (or refreshed again)

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

Behind the scenes: In the database, the materialized views of the geometries are refreshed.



Usage: As user

Behind the scenes:

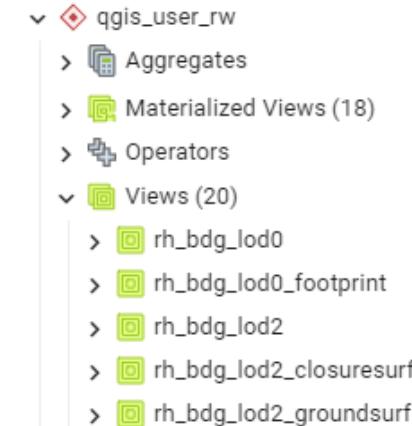
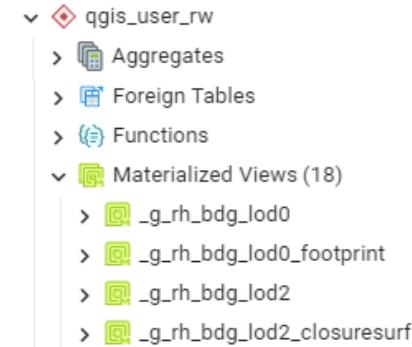
In the database user schema, both materialised views (containing the feature geometries) and the layers (as views, containing the feature attributes linked to the corresponding geometries in the materialised views) can be accessed.

Materialised views name coding:

- "_g" prefixed + citydb schema name + feature name + lodx + (optional) semantic details

Views name coding:

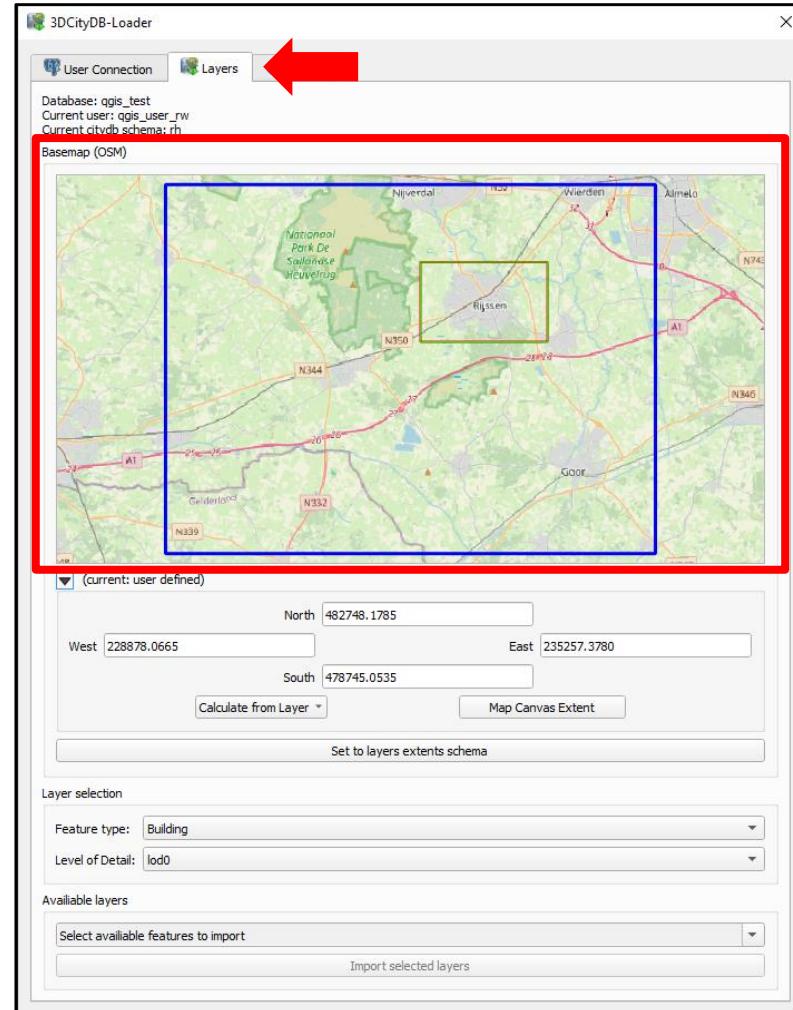
- citydb schema name + feature name + lodx + (optional) semantic details
- Linked via column co_id (PK and FK to the materialised views)



Usage: As user

In the "Layers" tab

- 8) The dataset and the (database-side) layer extents are shown again in the map



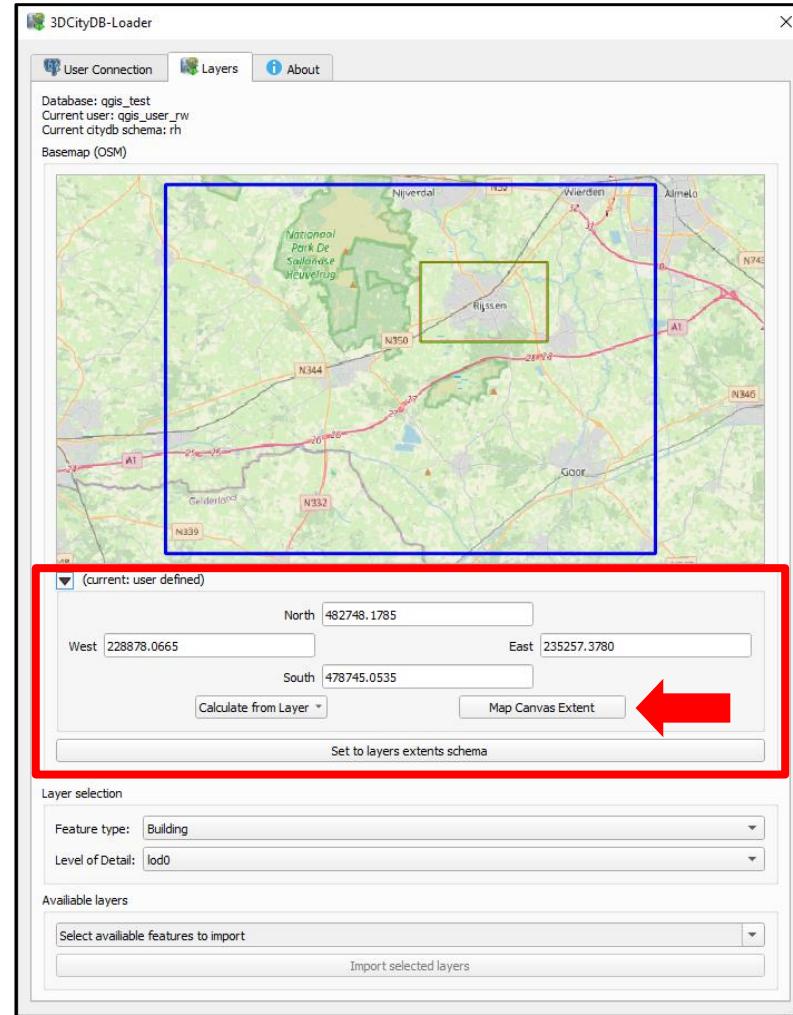
Usage: As user

In the "Layers" tab

9) Depending on your needs, you can further reduce the extents of the layers to be loaded into QGIS

- Default: same size of the materialised view extents
- Otherwise: zoom in the map and choose your own area

Behind the scenes: In the database, materialized 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)!



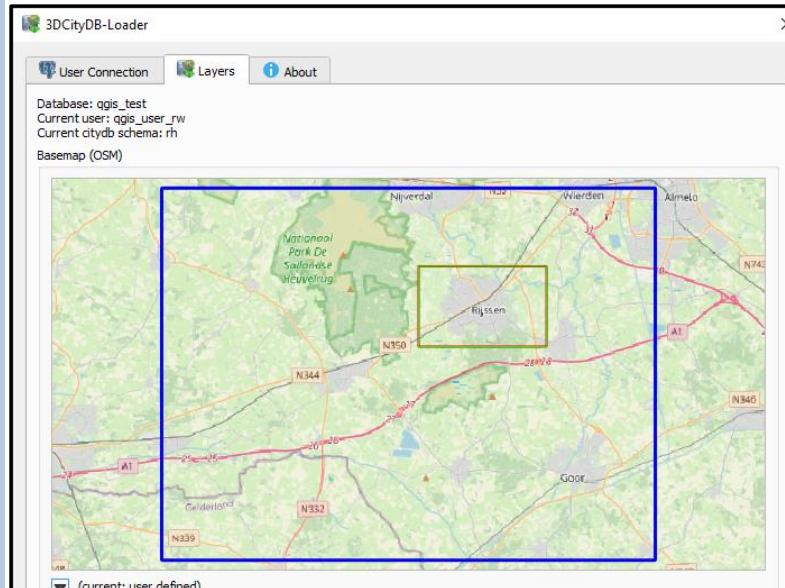
Usage: As user

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

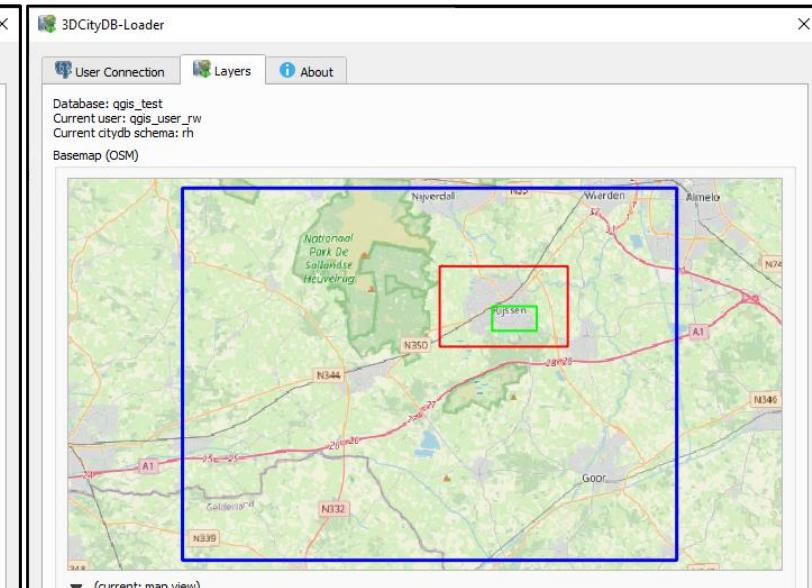
Red: database-side layers extents (i.e. extents of the materialised views)

Green: QGIS-side layers extents (i.e. extents of the data loaded into QGIS)

Remember: **QGIS-extents \subseteq Mat. views extents \subseteq City model extents**



Database- and QGIS-side extents coincide (default)



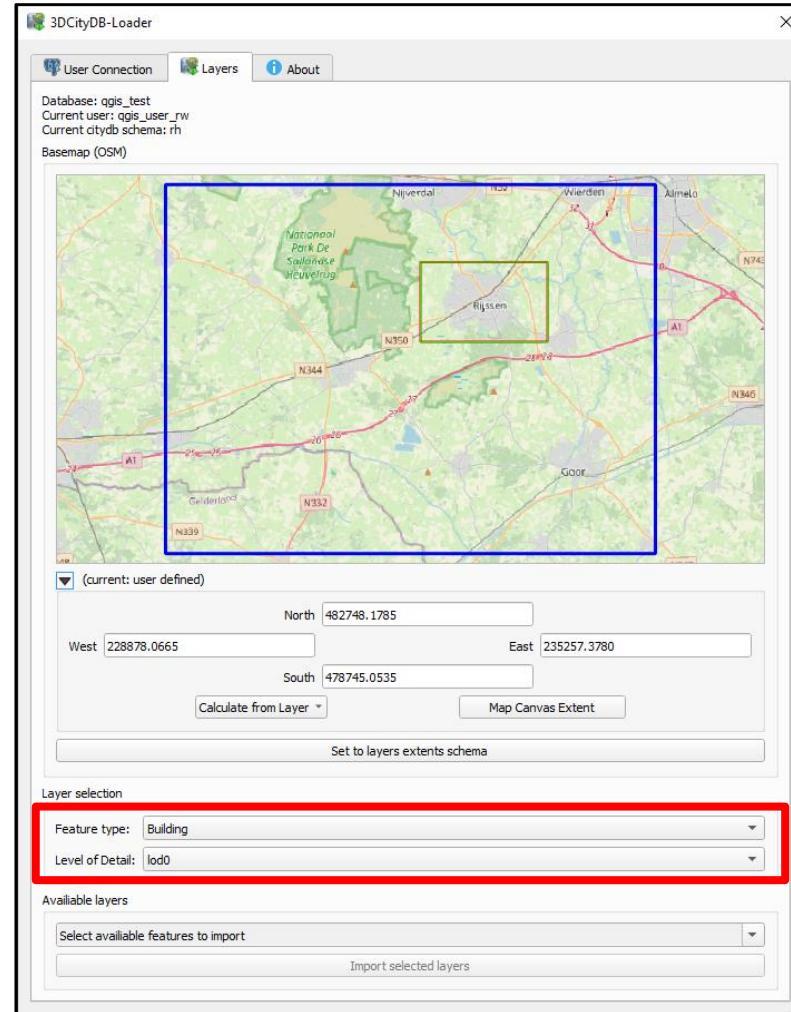
User-selected QGIS-layers extents

Usage: As user

In the "Layers" tab

10) Select feature type and LoD

- Layers are grouped according to the CityGML modules (or "Feature Types", such as "Bridge", "Building", "Tunnel", "Relief", etc.)
- Depending on the data, only certain Feature Types and LoDs will be selectable



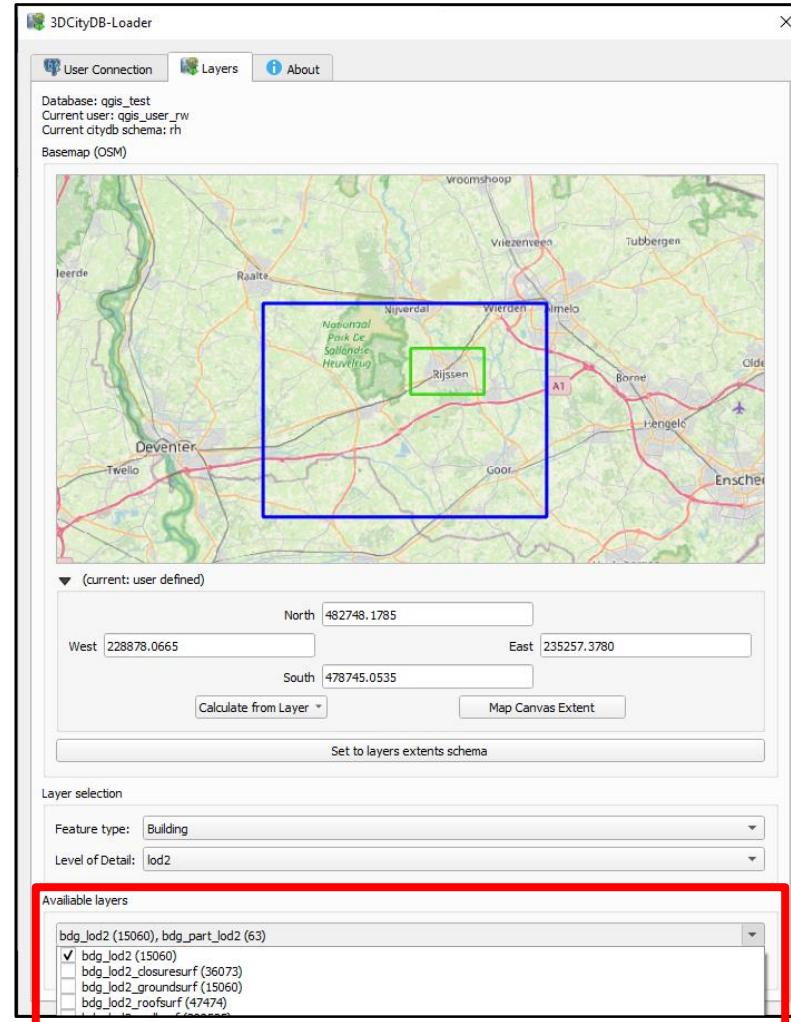
Usage: As user

In the "Layers" tab

11) Select the layer(s) to import in QGIS

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

Behind the scenes: Depending on the QGIS layers extents, the plugin shows only the available layers for which there are data in the selected area.

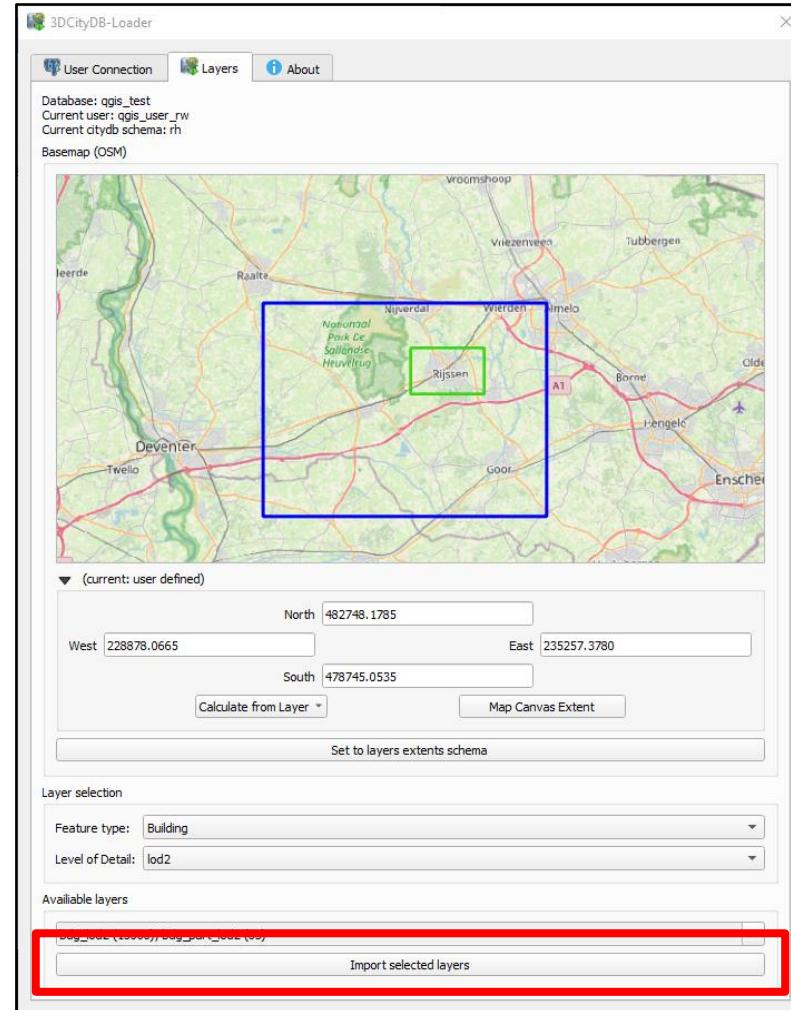


Usage: As user

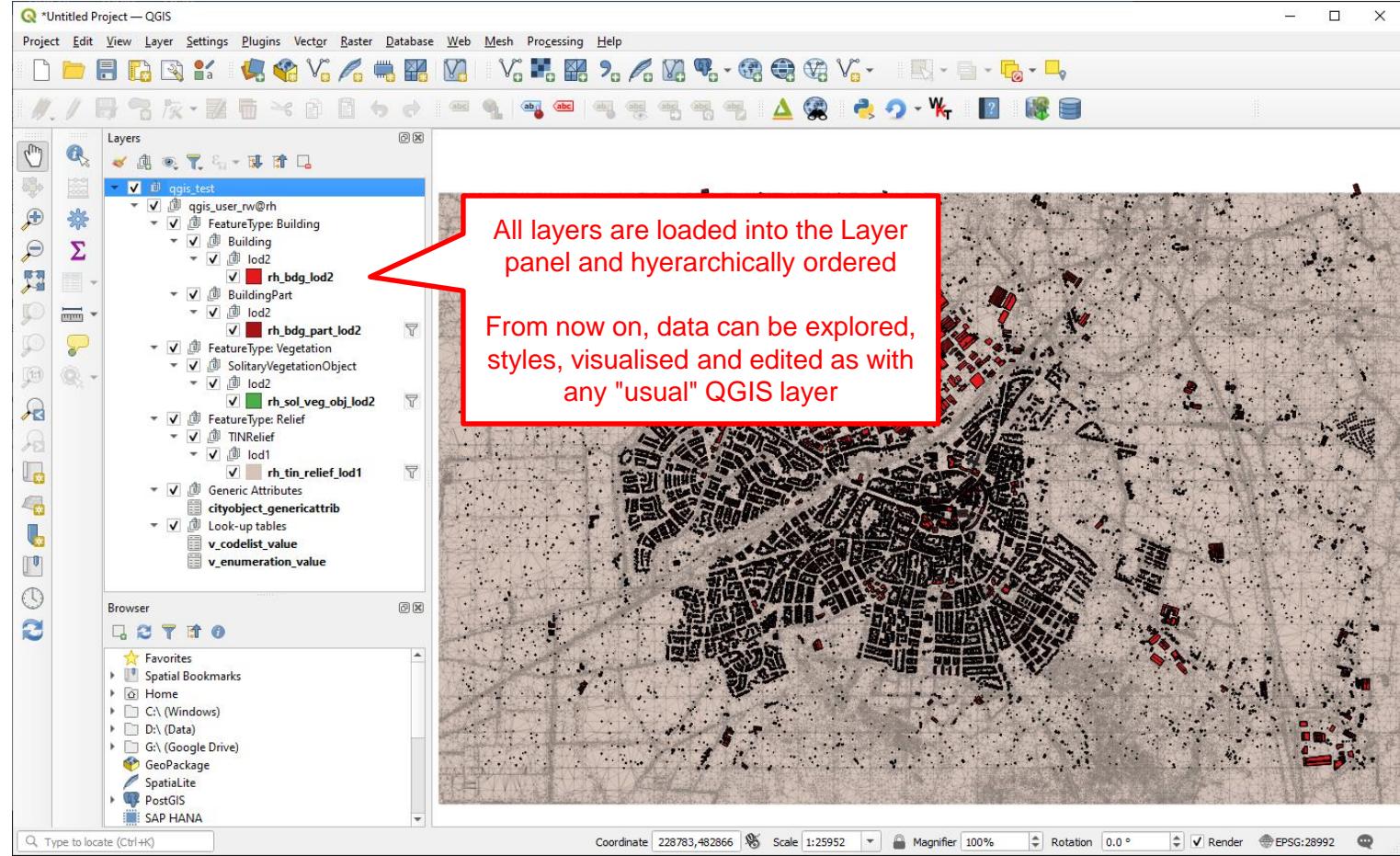
In the "Layers" tab

12) Import the selected layers into QGIS

- The import operation can be repeated with different layers
- The layers will be automatically added to the QGIS 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)



Usage in QGIS: Layers



The screenshot shows the QGIS interface with a project titled "Untitled Project — QGIS". The left side features the "Layers" panel, which displays a hierarchical tree of loaded layers. The tree includes categories like "FeatureType: Building", "FeatureType: Vegetation", "FeatureType: Relief", and "Generic Attributes", each with its own sub-layers. A red box highlights this panel with the text: "All layers are loaded into the Layer panel and hierarchically ordered". The main canvas area shows a detailed map of a city with buildings and terrain. A red box highlights the map area with the text: "From now on, data can be explored, styles, visualised and edited as with any "usual" QGIS layer".

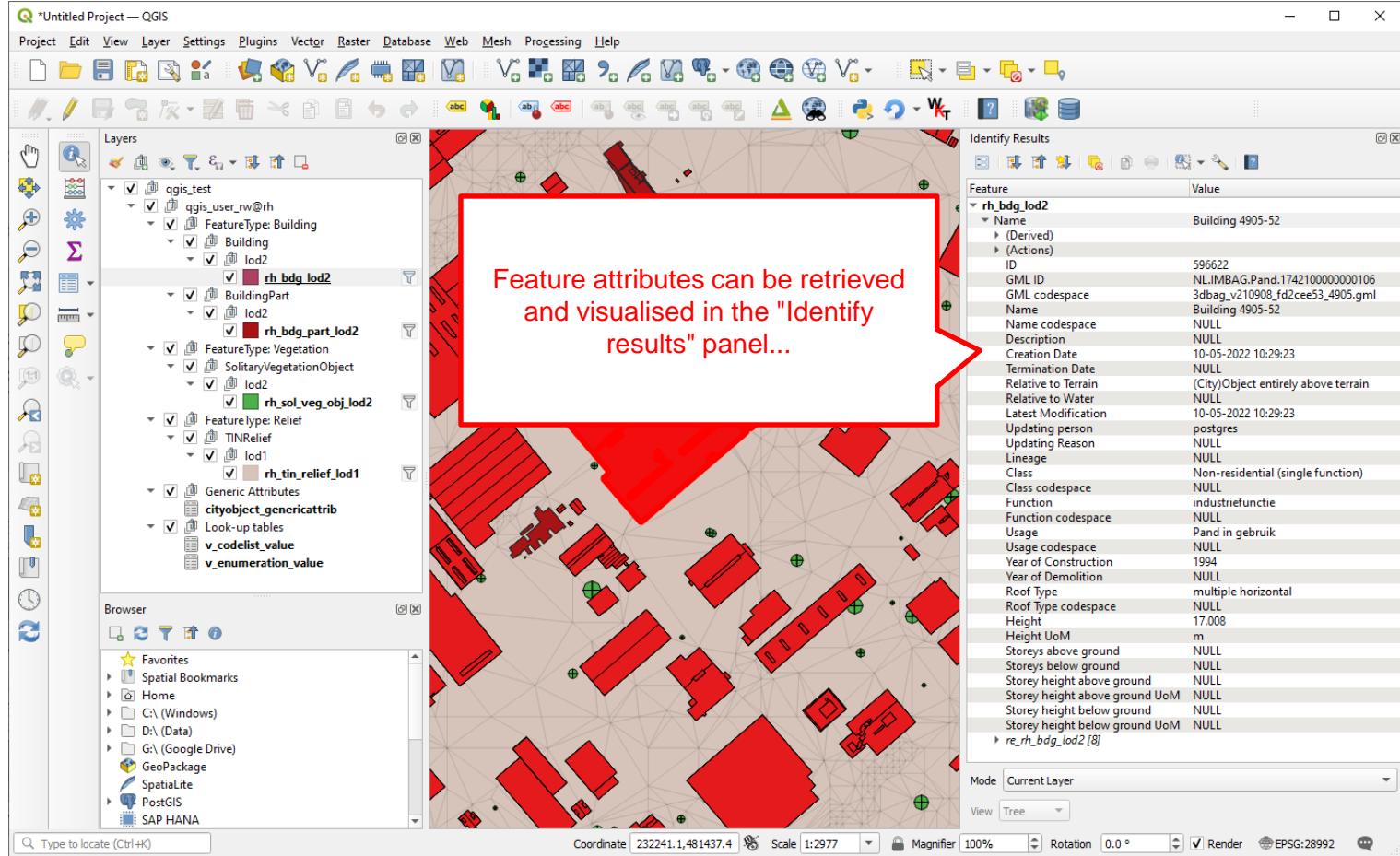
All layers are loaded into the Layer panel and hierarchically ordered

From now on, data can be explored, styles, visualised and edited as with any "usual" QGIS layer

Coordinate 228783,482866 Scale 1:25952 Magnifier 100% Rotation 0.0° Render EPSG:28992

Usage in QGIS: Attributes

Feature attributes can be retrieved and visualised in the "Identify results" panel...



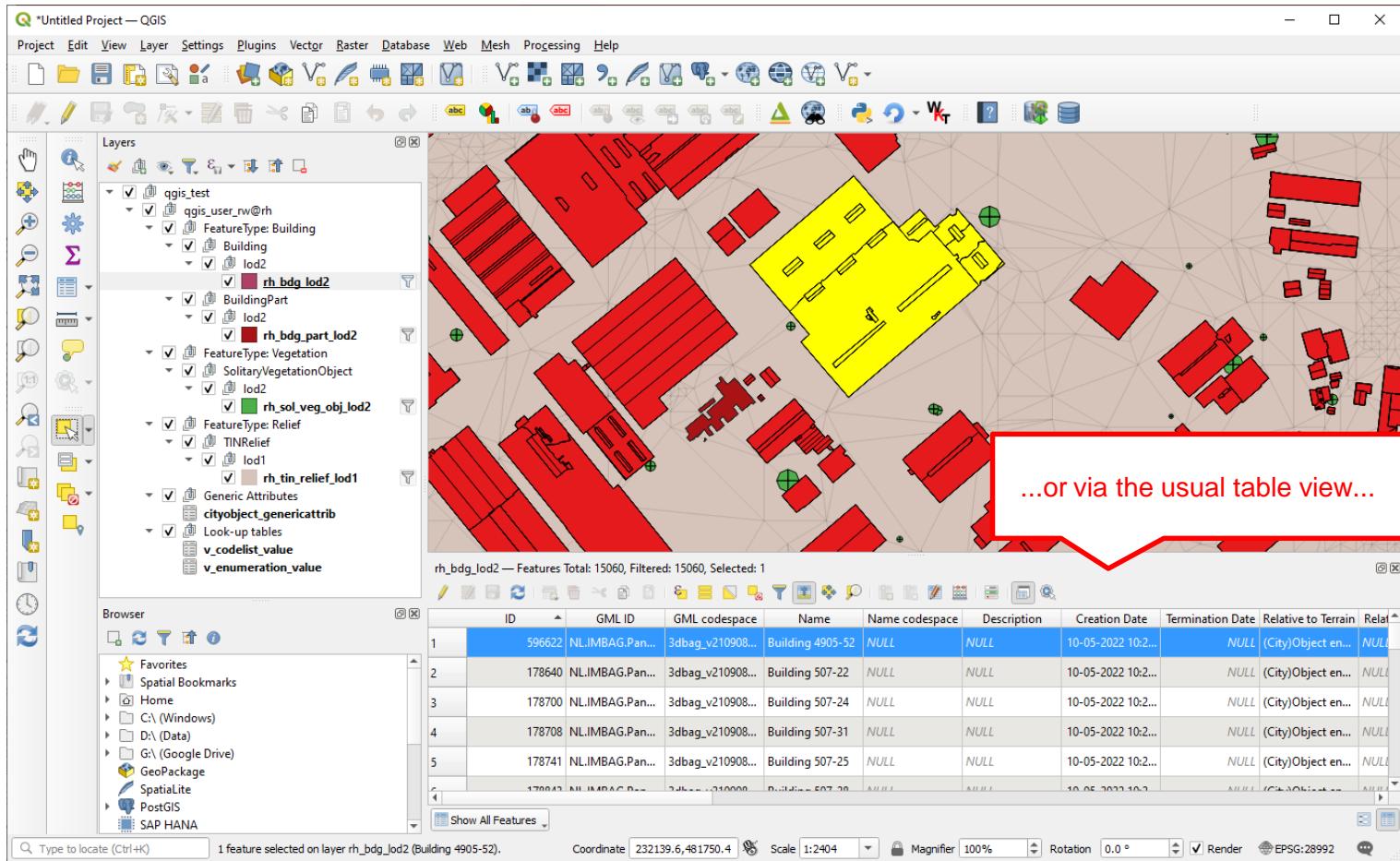
The screenshot shows the QGIS application interface with the following components visible:

- Project Bar:** Standard QGIS menu bar with options like Project, Edit, View, Layer, Settings, Plugins, Vector, Raster, Database, Web, Mesh, Processing, Help.
- Toolbar:** A large set of icons for various geoprocessing tools and functions.
- Layers Panel:** Shows a tree view of layers in the project. One layer is expanded to show its sub-components, including "rh_bdg_lod2" which is highlighted.
- Browser Panel:** Shows a list of spatial bookmarks and file paths.
- Map Canvas:** Displays a map with red polygonal features representing buildings, overlaid on a grey terrain base.
- Identify Results Panel:** A floating panel on the right containing a table titled "Identify Results". It lists attributes for the selected feature "rh_bdg_lod2". The table includes columns for "Feature" and "Value".

Identify Results Panel Data:

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

Usage in QGIS: Attributes



The screenshot shows the QGIS interface with a map of a residential area. Buildings are represented by red polygons, and a single building is highlighted in yellow. Vegetation is shown as green polygons. A red callout box points to the attribute table at the bottom right of the interface.

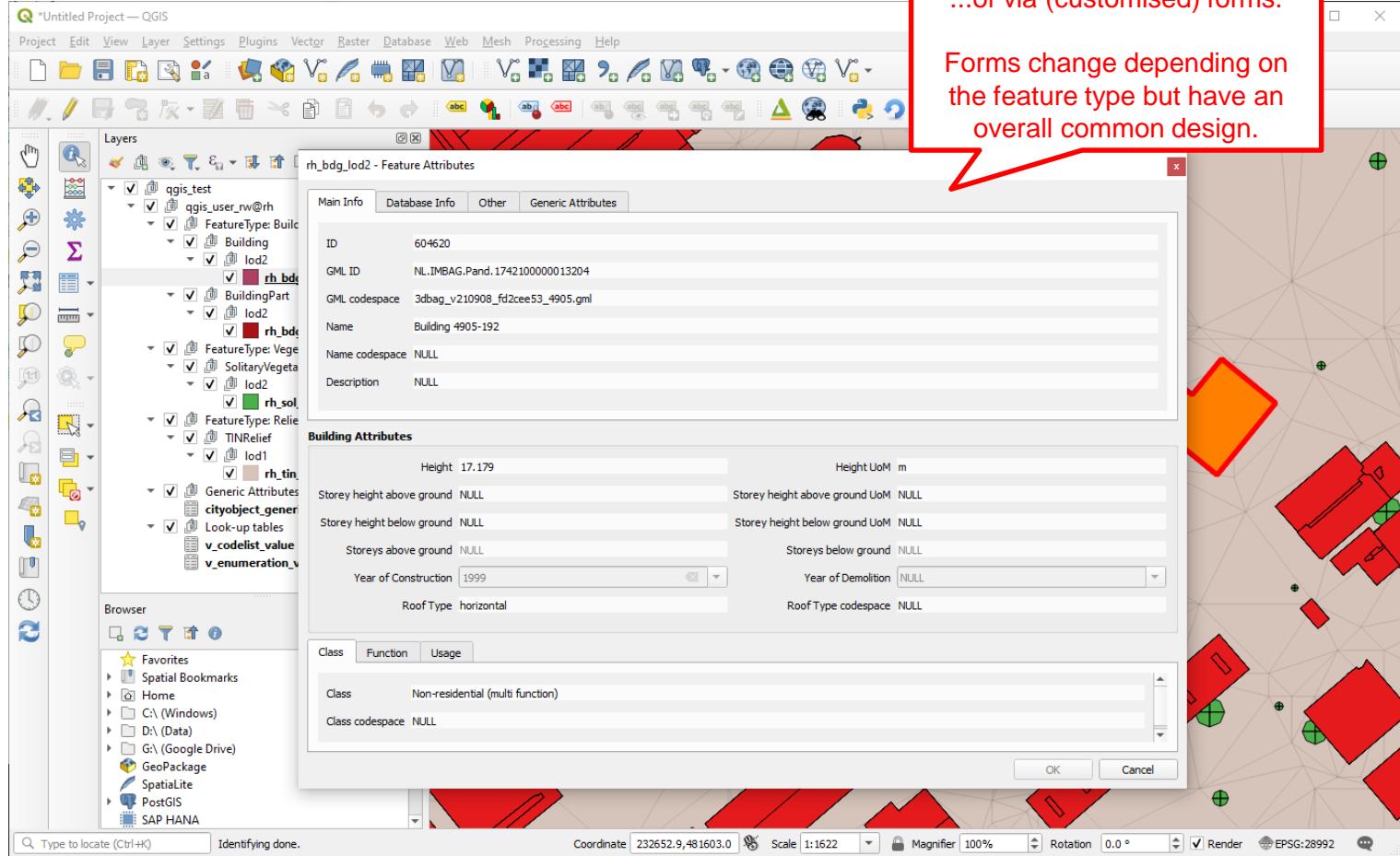
...or via the usual table view...

ID	GML ID	GML codespace	Name	Name codespace	Description	Creation Date	Termination Date	Relative to Terrain	Relat...
1	596622	NL.3DGeoInfo.2010...	Building 4905-52	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...	NULL
2	178640	NL.3DGeoInfo.2010...	Building 507-22	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...	NULL
3	178700	NL.3DGeoInfo.2010...	Building 507-24	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...	NULL
4	178708	NL.3DGeoInfo.2010...	Building 507-31	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...	NULL
5	178741	NL.3DGeoInfo.2010...	Building 507-25	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...	NULL
...	178640	NL.3DGeoInfo.2010...	Building 507-20	NULL	NULL	10-05-2022 10:2...	NULL	(City)Object en...	NULL

Usage in QGIS: Attributes

...or via (customised) forms.

Forms change depending on the feature type but have an overall common design.



Main Info

- ID: 604620
- GML ID: NL.IMGAG.Pand.1742100000013204
- GML codespace: 3dbag_v210908_fd2cee53_4905.gml
- Name: Building 4905-192
- Name codespace: NULL
- Description: NULL

Building Attributes

Height: 17.179	Height UoM: m
Storey height above ground: NULL	Storey height above ground UoM: NULL
Storey height below ground: NULL	Storey height below ground UoM: NULL
Storeys above ground: NULL	Storeys below ground: NULL
Year of Construction: 1999	Year of Demolition: NULL
Roof Type: horizontal	Roof Type codespace: NULL

Class

- Class: Non-residential (multi function)
- Class codespace: NULL

OK Cancel

Type to locate (Ctrl+K) Identifying done. Coordinate 232652.9, 481603.0 Scale 1:1622 Magnifier 100% Rotation 0.0° Render EPSG:28992

Usage in QGIS: Attributes

rh_bdg_lod2 - Feature Attributes

Main Info	Database Info	Other	Generic Attributes
Creation Date	10-05-2022 10:29:26		
Termination Date	NULL		
Latest Modification	10-05-2022 10:29:26		
Updating person	postgres		
Updating Reason	NULL		
Lineage	NULL		

rh_bdg_lod2 - Feature Attributes

Main Info	Database Info	Other	Generic Attributes
Surface Relation			
Relative to Terrain (City)Object entirely above terrain			
Relative to Water (no selection)			

rh_bdg_lod2 - Feature Attributes

Main Info	Database Info	Other	Generic Attributes										
<div style="border: 1px solid #ccc; padding: 5px;"> ✚ Expression <div style="margin-top: 5px;"> <input type="checkbox"/> 3dbag_tile <input type="checkbox"/> bag_net_internal_area <input type="checkbox"/> footprint_area <input type="checkbox"/> is_single_part <input type="checkbox"/> lod2_volume </div> </div>													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">id</td> <td>2892542</td> </tr> <tr> <td>parent_genattrib_id</td> <td>NULL</td> </tr> <tr> <td>root_genattrib_id</td> <td>2892542</td> </tr> <tr> <td>attrname</td> <td>3dbag_tile</td> </tr> <tr> <td>datatype</td> <td></td> </tr> </table>				id	2892542	parent_genattrib_id	NULL	root_genattrib_id	2892542	attrname	3dbag_tile	datatype	
id	2892542												
parent_genattrib_id	NULL												
root_genattrib_id	2892542												
attrname	3dbag_tile												
datatype													

Data is further organized in tabs and additional relations tables are connected via Generic Attributes

Usage in QGIS: Attributes

rh_bdg_lod2 - Feature Attributes

Main Info Database Info Other Generic Attributes

ID	604620
GML ID	NL.IMBAG.Pand.1742100000013204
GML codespace	3dbag_v210908_fd2cee53_4905.gml
Name	Building 4905-192
Name codespace	NULL
Description	NULL

Building Attributes

Height	17.179	Height UoM	m
Storey height above ground	NULL	Storey height above ground	NULL
Storey height below ground	NULL	Storey height below ground	NULL
Storeys above ground	NULL	Storeys	Year
Year of Construction	1999	Roof Type	Horizontal
Roof Type	Horizontal	Year	Roof Ty

Class Function Usage

Function	industriefunctie, winkelfunctie
Function codespace	

Attributes containing [0..*] entries (e.g. function, usage, etc.) are "linearised" to a comma-separated string.

Codelists can also be visualised as multi-value combo boxes (see advanced settings later on)

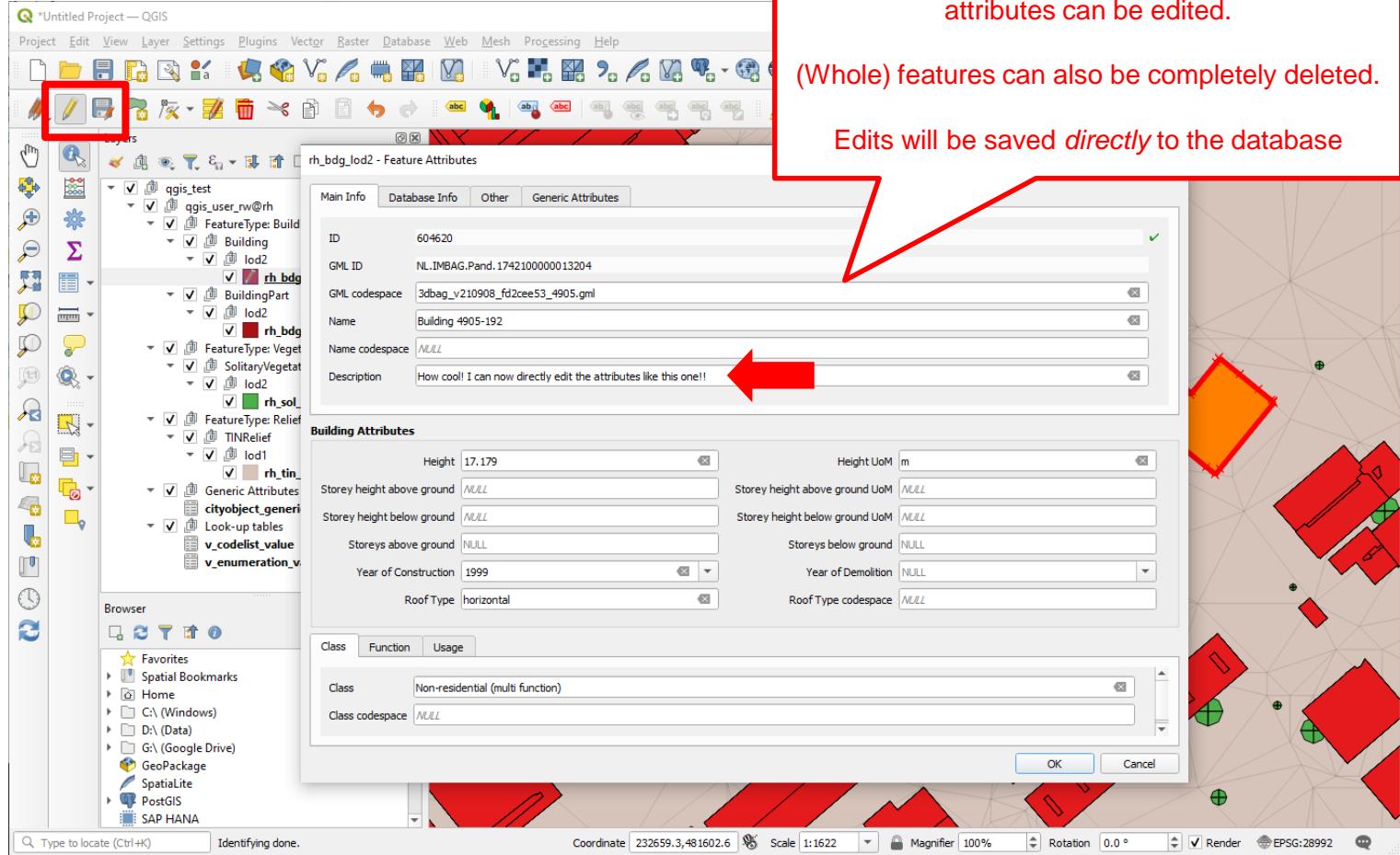
OK Cancel

Usage in QGIS: Attribute editing

If the user has read/write privileges, most feature attributes can be edited.

(Whole) features can also be completely deleted.

Edits will be saved *directly* to the database



The screenshot shows the QGIS interface with a red box highlighting the 'Edit' button in the toolbar. The 'Attribute Table' dialog is open for a feature named 'rh_bdg'. The 'Main Info' tab displays various attribute fields:

- ID: 604620
- GML ID: NL.UMBAG.Pand.1742100000013204
- GML codespace: 3dbag_v210908_fd2ce53_4905.gml
- Name: Building 4905-192
- Name codespace: NULL
- Description: How cool! I can now directly edit the attributes like this one!!

The 'Building Attributes' section contains the following values:

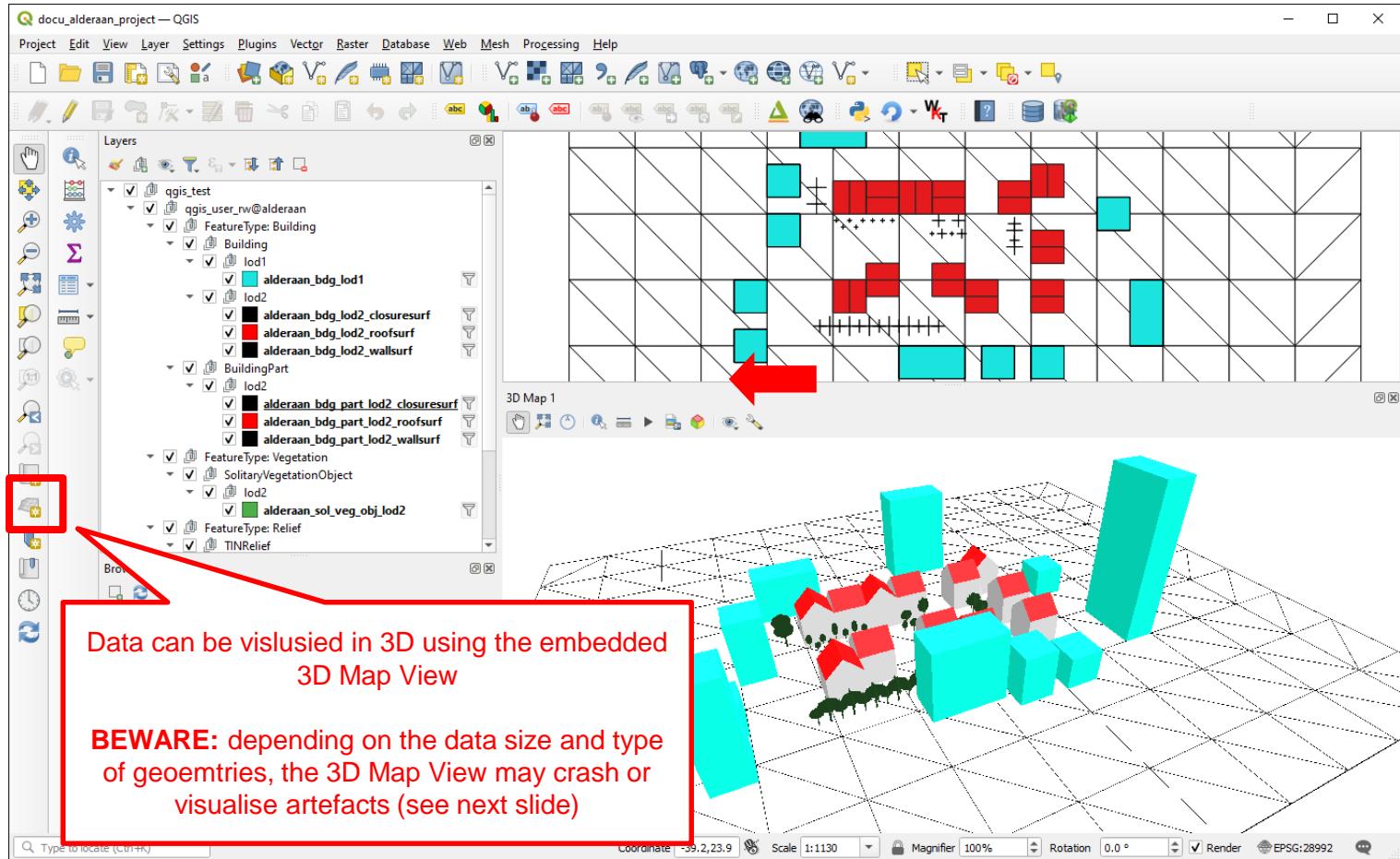
- Height: 17.179
- Height UoM: m
- Storey height above ground: NULL
- Storey height above ground UoM: NULL
- Storey height below ground: NULL
- Storey height below ground UoM: NULL
- Storeys above ground: NULL
- Storeys below ground: NULL
- Year of Construction: 1999
- Year of Demolition: NULL
- Roof Type: horizontal
- Roof Type codespace: NULL

The 'Class' tab shows the class information:

- Class: Non-residential (multi function)
- Class codespace: NULL

A red arrow points from the text 'Edits will be saved *directly* to the database' to the 'Description' field in the attribute table. Another red arrow points from the text '(Whole) features can also be completely deleted.' to the map view where several buildings are highlighted in red.

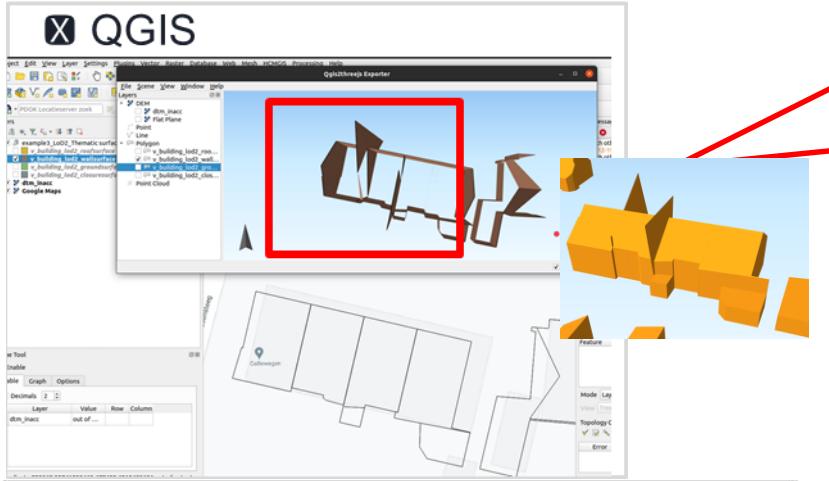
Usage in QGIS: 3D visualisation



Data can be visualised in 3D using the embedded 3D Map View

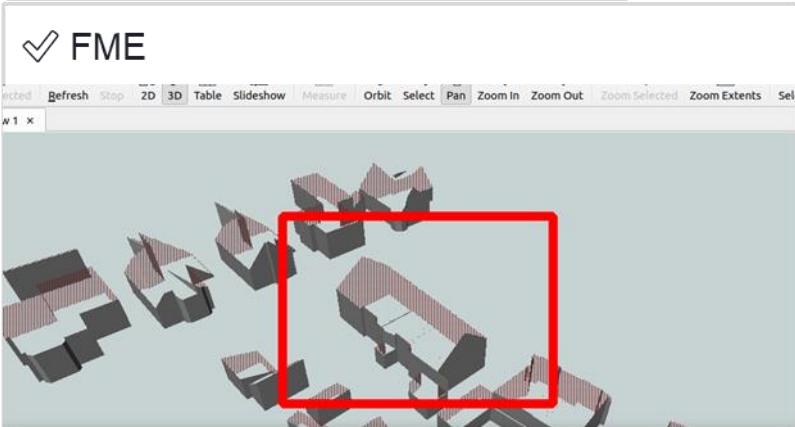
BEWARE: depending on the data size and type of geometries, the 3D Map View may crash or visualise artefacts (see next slide)

Usage in QGIS: 3D visualisation



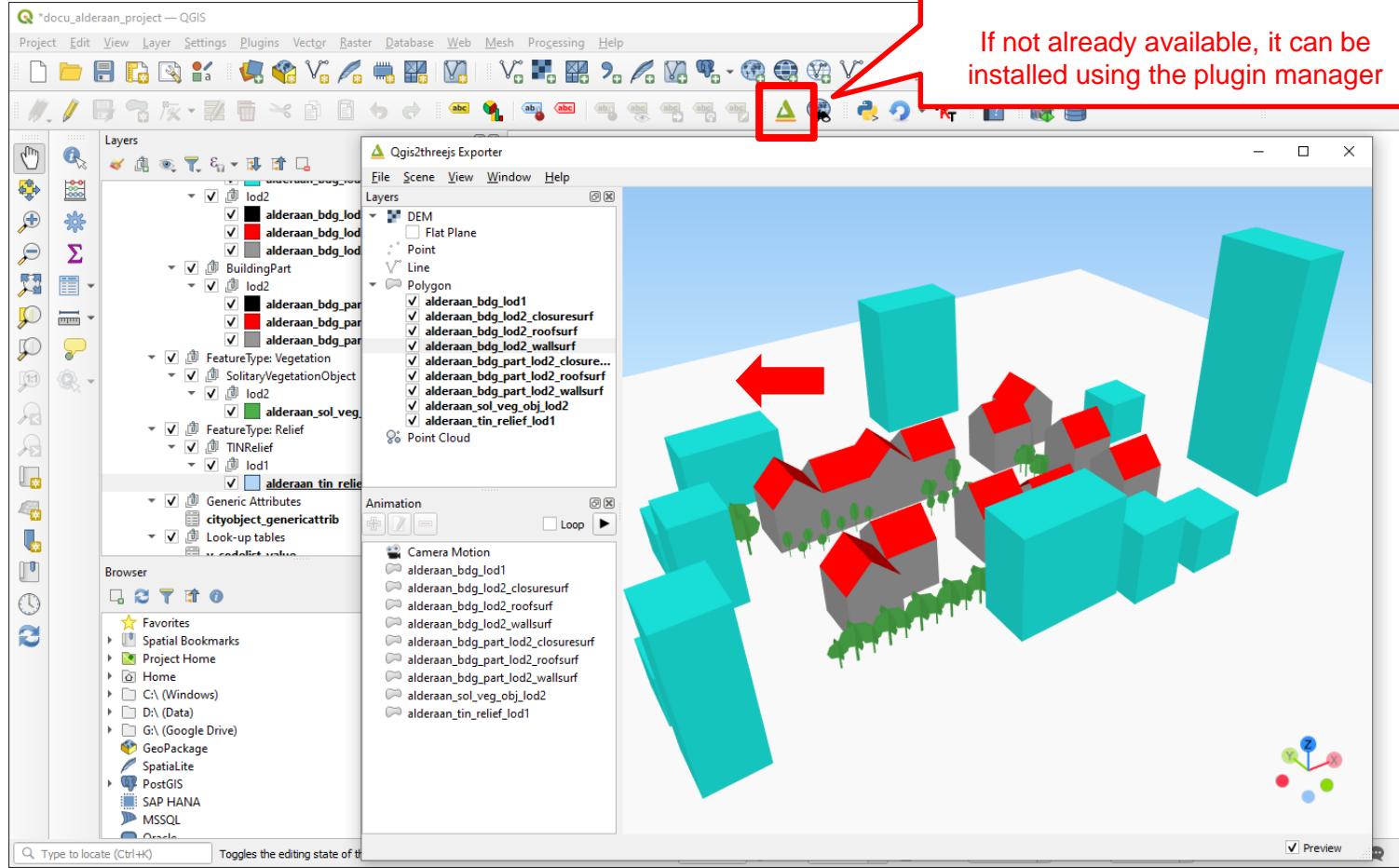
In our tests, such artefacts are however only a 3D visualisation issue (QGIS 3D renderer?) and *not* related to the data. 3D visualisation in FME and in Google Earth show indeed correct results.

In order to minimise this problem, please refer to advanced settings (see later)



✓
Google
earth
(as KML)

Usage in QGIS: 3D visualisation



Advanced options

- To be added
- Administrator
 - Add and manage database users
- User
 - Simplify geometries in Materialised views
 - Load and set look-up tables (codelists)

Plugin: Server-side

GOODIES

Columns with the "beloved" --\-- field separator can be transformed via views into simple arrays (which QGIS likes!)

Trigger functions take care of data conversion from/to arrays ✓

**Table BUILDING
(excerpt)**

id [PK] bigint	class character varying (256)	function character varying (1000)
1531	Mixed-use	overige gebruiksfunctie--\--woonfunctie
4431	Mixed-use	winkelfunctie--\--woonfunctie
7839	Non-residential (multi function)	gezondheidszorgfunctie--\--kantoorfunctie--\--winkelfunctie
15423	Non-residential (multi function)	gezondheidszorgfunctie--\--kantoorfunctie
15499	Mixed-use	gezondheidszorgfunctie--\--kantoorfunctie--\--winkelfunctie--\--woonf...
16545	Mixed-use	overige gebruiksfunctie--\--woonfunctie
14490	Mixed-use	bijeenkomstfunctie--\--gezondheidszorgfunctie--\--kantoorfunctie--\--...
15956	Mixed-use	overige gebruiksfunctie--\--woonfunctie

View BUILDING (excerpt)

function_array character varying[]
{'overige gebruiksfunctie', 'woonfunctie'}
{'winkelfunctie', 'woonfunctie'}
{'gezondheidszorgfunctie', 'kantoorfunctie', 'winkelfunctie'}
{'gezondheidszorgfunctie', 'kantoorfunctie'}
{'gezondheidszorgfunctie', 'kantoorfunctie', 'winkelfunctie', 'woonfunctie'}
{'overige gebruiksfunctie', 'woonfunctie'}
{'bijeenkomstfunctie', 'gezondheidszorgfunctie', 'kantoorfunctie', 'overi...'}
{'overige gebruiksfunctie', 'woonfunctie'}



Plugin: Front-end

_geom_citydb_bdg_lod0_footprint - Feature Attributes

Actions

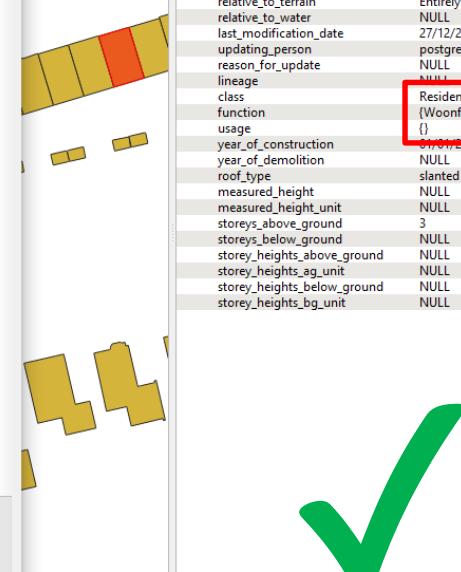
db_id	690643
gmlid	NL.IMBAG.Pand.1742100000095946
gmlid_codespace	RijssenHolten_all_lod.zip
name	Building 2039-1622
name_codespace	NULL
description	NULL
creation_date	27/12/2021 10:54:17
termination_date	NULL
relative_to_terrain	Entirely above terrain
relative_to_water	(no selection)
last_modification_date	27/12/2021 10:54:17
updating_person	postgres
reason_for_update	NULL
lineage	NULL
class	Residential
function	<input type="checkbox"/> Bijeenkomstfunctie <input type="checkbox"/> Celfunctie <input type="checkbox"/> Gezondheidszorgf... <input type="checkbox"/> Industriefunctie <input type="checkbox"/> Kantoorfunctie <input type="checkbox"/> Logiesfunctie <input type="checkbox"/> Onderwijsfunctie <input type="checkbox"/> Overige ... <input type="checkbox"/> Sportfunctie <input type="checkbox"/> Winkelfunctie <input checked="" type="checkbox"/> Woonfunctie
usage	<input type="checkbox"/> Bijeenkomstfunctie <input type="checkbox"/> Celfunctie <input type="checkbox"/> Gezondheidszorgf... <input type="checkbox"/> Industriefunctie <input type="checkbox"/> Kantoorfunctie <input type="checkbox"/> Logiesfunctie <input type="checkbox"/> Onderwijsfunctie <input type="checkbox"/> Overige ... <input type="checkbox"/> Sportfunctie <input type="checkbox"/> Winkelfunctie <input type="checkbox"/> Woonfunctie
year_of_construction	01/01/2014
year_of_demolition	NULL
roof_type	slanted
measured_height	NULL
measured_height_unit	NULL
storeys_above_ground	3
storey_heights_above_ground	NULL
storey_heights_ag_unit	NULL
storey_heights_below_ground	NULL
storey_heights_bg_unit	NULL

OK Cancel

Identify Results

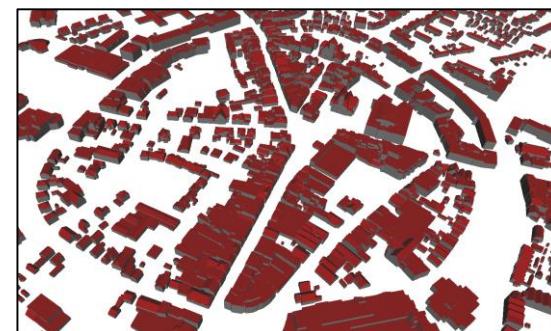
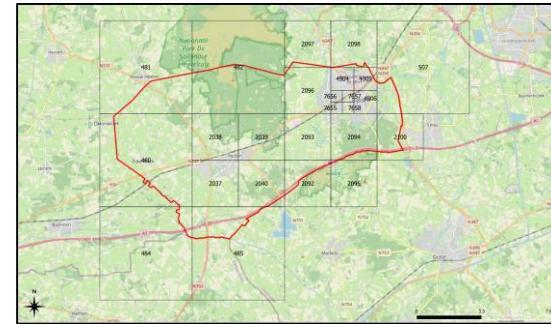
Feature	Value
db_id	690643
db_id	690643
gmlid	NL.IMBAG.Pand.1742100000095946
gmlid_codespace	RijssenHolten_all_lod.zip
name	Building 2039-1622
name_codespace	NULL
description	NULL
creation_date	27/12/2021 10:54:17 (UTC+01:00)
termination_date	NULL
relative_to_terrain	Entirely above terrain
relative_to_water	NULL
last_modification_date	27/12/2021 10:54:17 (UTC+01:00)
updating_person	postgres
reason_for_update	NULL
lineage	NULL
class	Residential
function	{ "Woonfunctie" }
usage	{} 01/01/2014
year_of_construction	NULL
year_of_demolition	NULL
roof_type	slanted
measured_height	NULL
measured_height_unit	NULL
storeys_above_ground	3
storeys_below_ground	NULL
storey_heights_above_ground	NULL
storey_heights_ag_unit	NULL
storey_heights_below_ground	NULL
storey_heights_bg_unit	NULL

Mode Current Layer

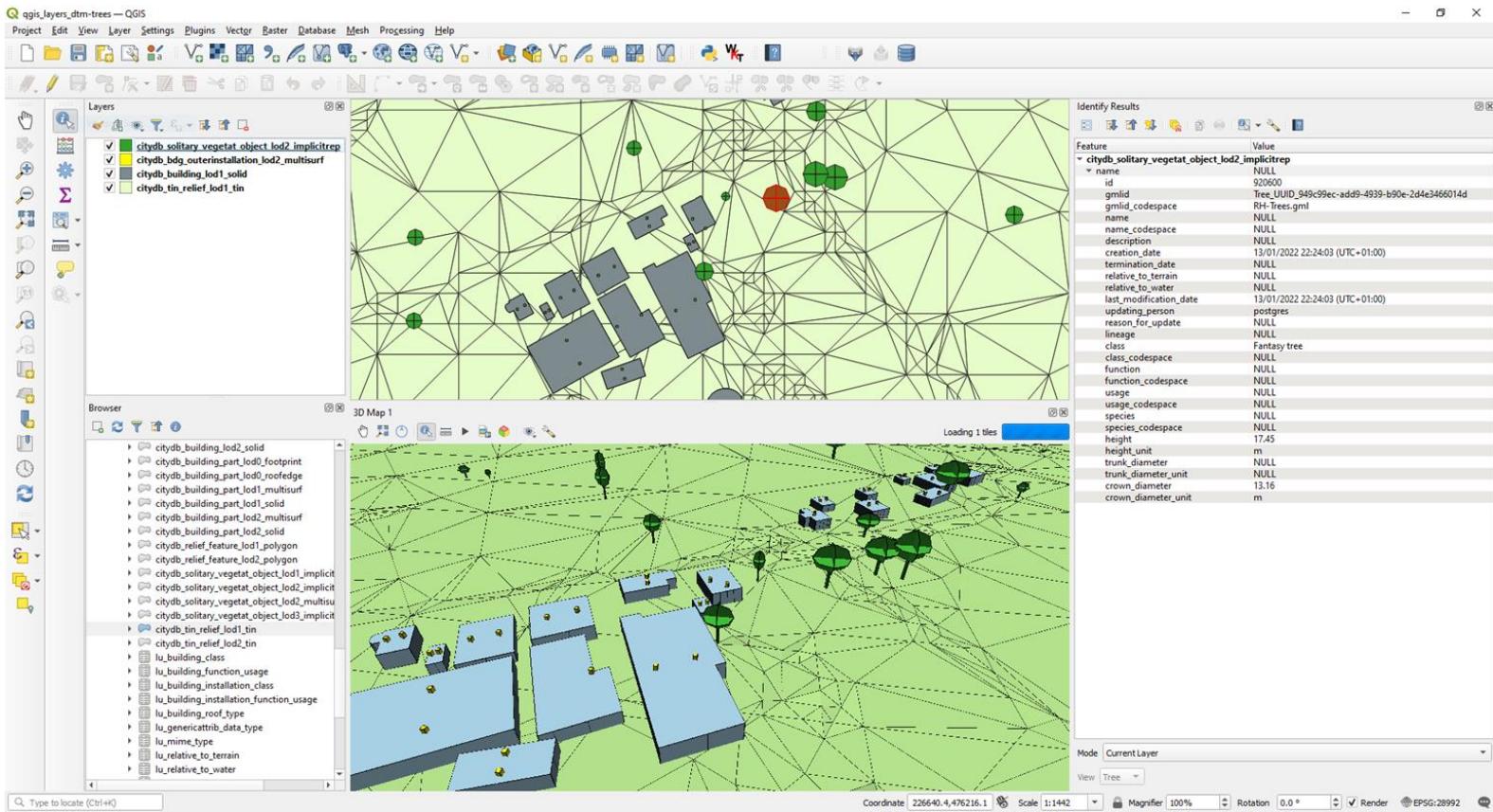


Test data

- "Customised" version of the 3D city model of the Dutch municipality of Rijssen-Holten
 - Source: 3D BAG dataset: <https://3dbag.nl/en/viewer>
 - 25 tiles, for circa 35000 buildings
 - Circa 50000 (Outer)BuildingInstallations
 - Circa 50000 SolitaryVegetationObjects (in LoD2/3 as implicit representation)
 - LoD1 ReliefFeature containing circa 5000 LoD1 TINReliefs
- Main characteristics:
 - Single-part buildings and (few) multi-part buildings
 - Both building and building part objects have some **generic attributes**
 - There are **global appearances** to help distinguish the different geometries



Test data



Comments, open ideas, ...

- At the moment, the **Approach 2** (mat views + views in PostgreSQL) seems the most convincing one
 - More SQL scripting, but it can be generated automatically
 - Already able to deal with multiple citydb schemas
 - Hopefully less QGIS-side programming for "tables", more focus on GUI/UX
 - To be further tested
- What about **ADEs**?
 - With new ADE-Features: no problem. Same approach: one layer per feature *AND* LoD *AND* geometry type
 - E.g. Energy ADE ThermalZones, UsageZones, ThermalBoundaries, WeatherStations
 - ADE attributes of existing CityGML classes:
 - Approach 2 creates many hard-coded views. With new ADE-properties of existing CityGML classes, this might become tricky as the views must be rewritten
 - Approach 3 may be more flexible, in that ADEs table are joined (from QGIS)
 - **_GEOM_BDG_LOD1_SOLID + CITYOBJECT + NRG_CITYOBJECT + BUILDING + NRG_BUILDING?**

Requires more reasoning, but out of scope of the MSc thesis

Testing machines

OS	Proc	HD	RAM	PostgreSQL	PostGIS	QGIS
Windows 10 2009 64bit	Core I7-8650U 1.7 GHz	SSD 500 GB	32 GB	14 64bit	3.2	QGIS 3.22 64bit
Linux (Ubuntu 20.04.3 LTS x86_64)	Intel i7-7500U (4) 3.500GHz	SSD 250 GB	8 GB	12 64bit	3.1	QGIS 3.22
Mac OS (11.6.2 64bit)	Core i9-??	1TB SSD	32 GB	13 64bit	3.1	
Linux server (Ubuntu 18.4)	TUD VM			10 64bit	3.0	N/A

Thank you for your attention!



Dr. Giorgio Agugiaro

g.agugiaro@tudelft.nl

3D Geoinformation Group

TU Delft

The Netherlands

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