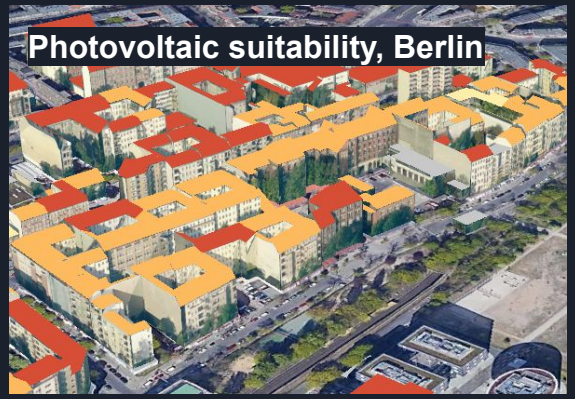
# Development of a QGIS plugin for the CityGML 3D City Database

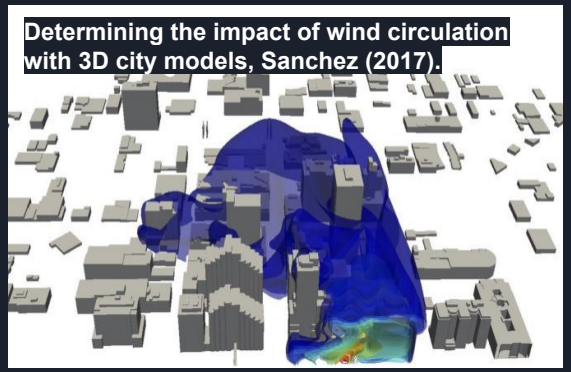
Konstantinos Pantelios
P5 - 29-06-2022

| | |
|---|---|
| TU Delft Supervisors: | Camilo León Sánchez<br>Giorgio Agugiaro |
| External supervisors: | Nagel Claus<br>Zhihang Yao |
| Co-reader: | Martijn Meijers |

# 3D City Models

- Functionality
  - Visualisation
  - Analysis
    and more…
- Applications
  - Urban planning
  - Energy modeling
    and more…
- Data Management
  - CityGML → OGC standard
    - Storage
    - Exchange
  - File encoding
    - E.g. *XML, JSON (CityJSON)*
  - **Database encoding**
    - *E.g. SQL (3D City Database)*



# Database encoding

3D City Models = Large amounts of data (usually)

Database functionalities:

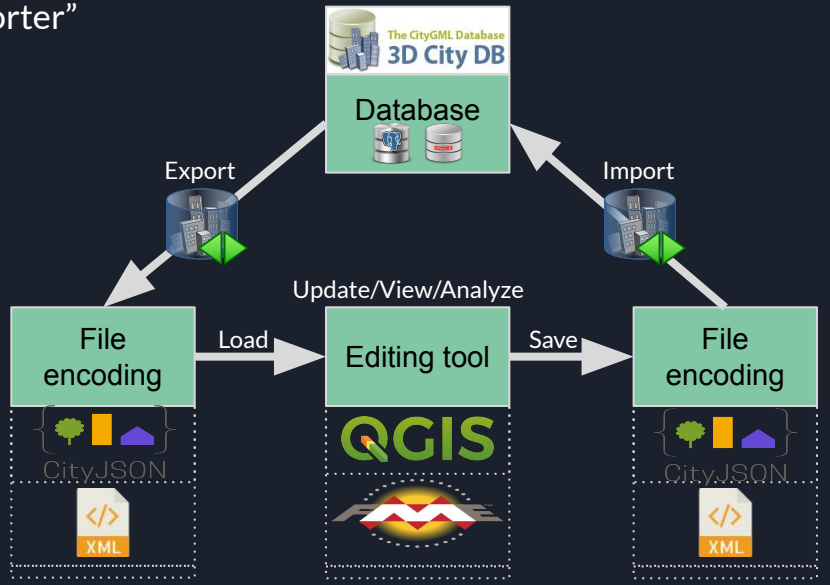
- Dedicated resources (storage, processing)
- Easier queries of large data-sets
- Search, update, insert and delete operations
- Data structures (“native” index support)
- Organize and access data directly or via APIs and more…

# “3D City Database” (1/5)

- Open source software
- PostgreSQL (and Oracle)
- Supported by the “3DCityDB Importer/Exporter” tool and others

However!

- **Indirect** data usage



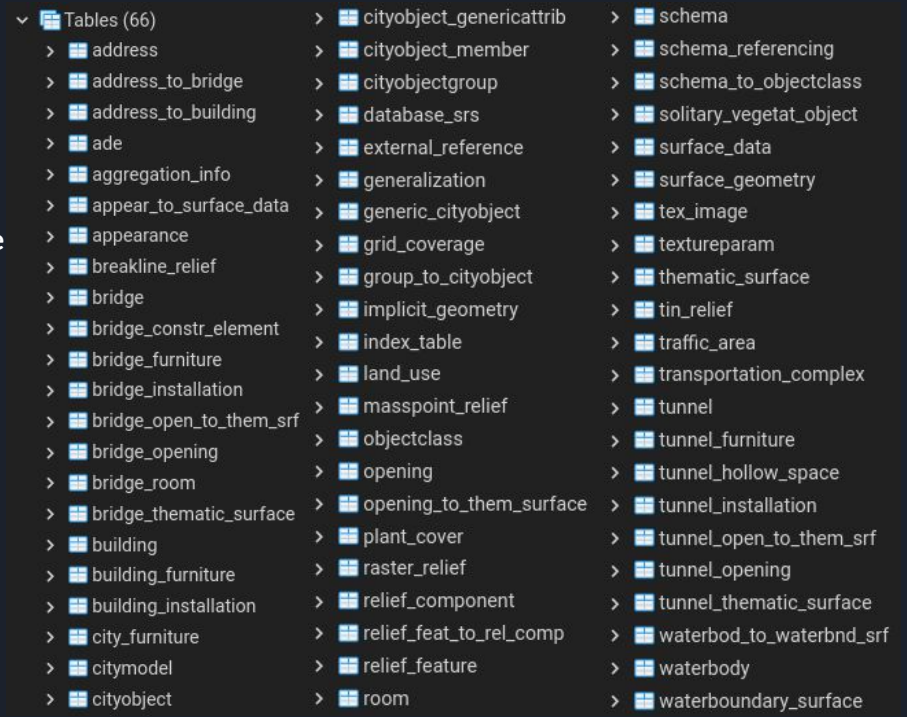
# “3D City Database” (2/5)

Complicated **direct** data usage:

- **66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database**
- Attributes are split over multiple tables
- Complex geometry table structure
- Features do not conform to the “Simple Feature Model” (SFM)

Resulting in:

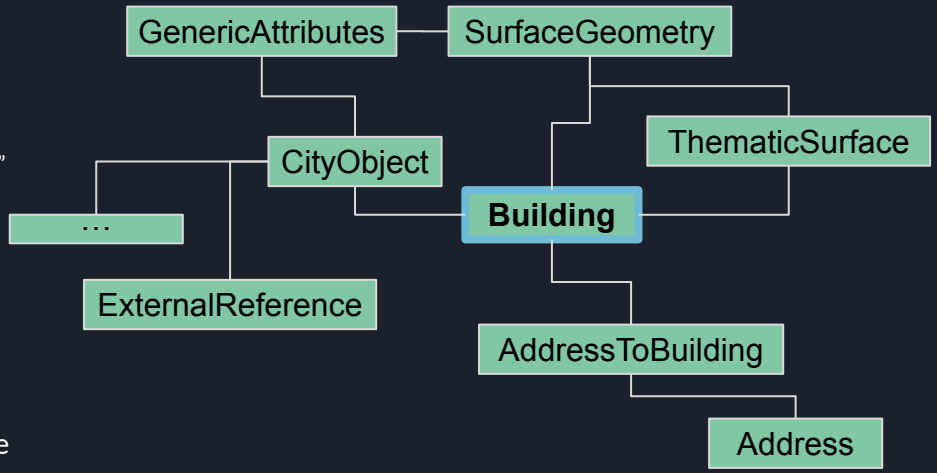
- Requiring complex queries,
- Overall difficult interaction with the database



# “3D City Database” (3/5)

Complicated **direct** data usage:

- 66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database
- **Attributes are split over multiple tables**
- Complex geometry table structure
- Features do not conform to the “Simple Feature Model” (SFM)



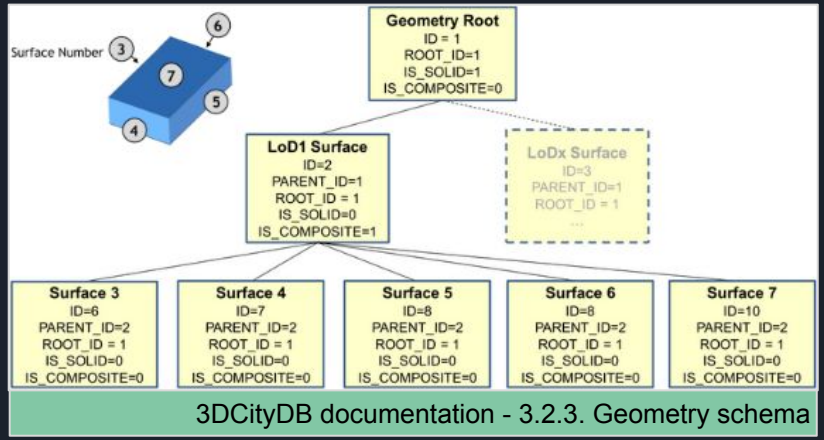

Resulting in:

- Requiring complex queries,
- Overall difficult interaction with the database

# “3D City Database” (4/5)

Complicated **direct** data usage:

- 66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database
- Attributes are split over multiple tables
- **Complex geometry table structure**
- Features do not conform to the “Simple Feature Model” (SFM)



3DCityDB documentation - 3.2.3. Geometry schema

Resulting in:

- Requiring complex queries,
- Overall difficult interaction with the database

# “3D City Database” (5/5)

Complicated **direct** data usage:

- 66 tables for 1 scenario (“citydb” schema) → Possibly multiple scenarios in the same database
- Attributes are split over multiple tables
- Complex geometry table structure
- **Features do not conform to the “Simple Feature Model” (SFM)**

| # | Field name | Field type |
|---|---|---|
| 1 | id | Feature atribute |
| 2 | gmlid | Feature atribute |
| 3 | ….. | Feature atribute |
| 15 | class | Feature atribute |
| 16 | class_codespace | Feature atribute |
| 17 | ….. | Feature atribute |
| 33 | geom | Feature geometry |

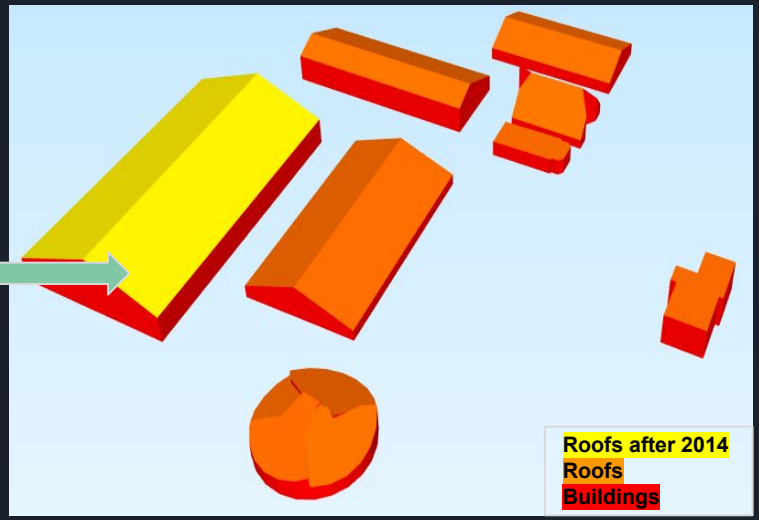
SFM layer example

Resulting in:

- Requiring complex queries,
- Overall difficult interaction with the database

# Example query of building roofs constructed after the year 2014

```sql
SELECT
    ts.id AS roof_id,
    co_ts.gmlid AS roof_gmlid,
    b.id AS building_id,
    co.gmlid AS building_gmlid,
    b.year_of_construction,
    ST_Collect(sg.geometry) AS roof_geom
FROM
    citydb.thematic_surface AS ts
    INNER JOIN citydb.cityobject AS co_ts
        ON (co_ts.id = ts.id)
    INNER JOIN citydb.surface_geometry AS sg
        ON (ts.lod2_multi_surface_id = sg.root_id)
    INNER JOIN citydb.building AS b
        ON (b.id = ts.building_id)
    INNER JOIN citydb.cityobject AS co
        ON (co.id = b.id)
WHERE
    ts.objectclass_id = 33 AND -- roofsurfaces
    b.objectclass_id = 26 AND -- buildings
    b.year_of_construction >= '2015-01-01'::date
GROUP BY
    ts.id,
    co_ts.gmlid,
    b.id,
    co.gmlid,
    b.year_of_construction
ORDER BY
    b.id,
    ts.id;
```



# Objective - Research questions

How to simplify user interaction with the 3DCityDB in terms of:

- Visualization in 2D/3D of multi-LoD geometries?
- Accessing and editing attributes (if user is allowed)?

In terms of implementation:

- How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)?
- Can we “restructure” data in 3DCityDB in order to conform to the SFM?

# Related work (1/3)

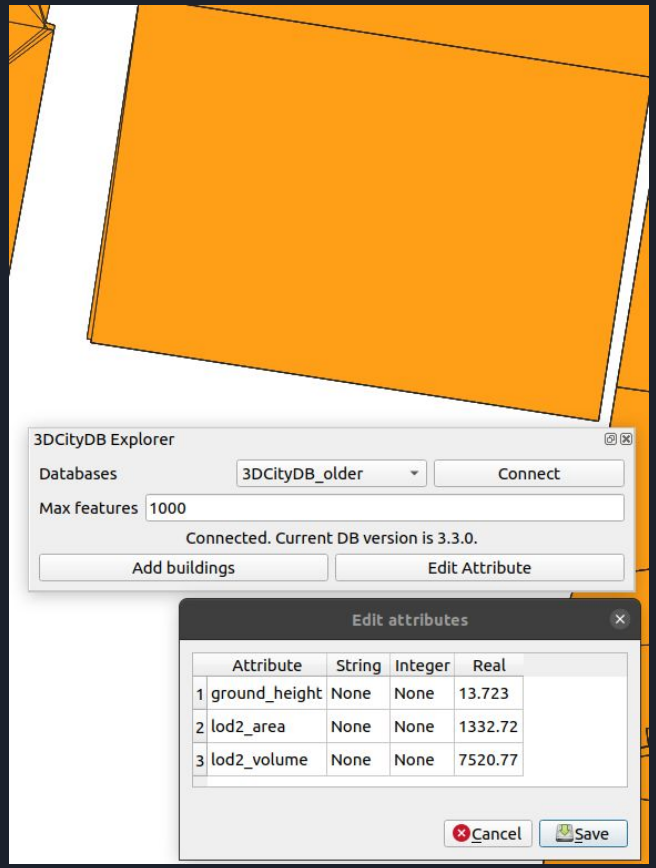
**QGIS plugin: “3DCityDB Explorer” (08-03-2021)**

Functionalities:

- Load data from database
- Update attributes
- Handles data size

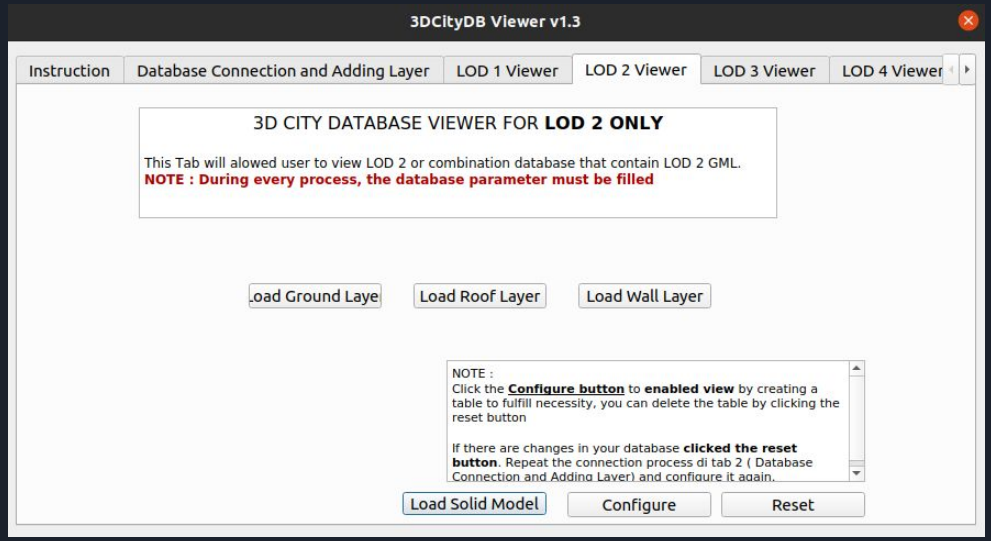
Limitations:

- Loads ONLY the “Building” class of LOD2
- Updates ONLY the “genericAttribute” class
- Works only for the default “citydb” scenario
- Does not account for multiple users/privileges.



# Related work (2/3)

**QGIS plugin: “3DCityDB Viewer” (16-06-2021)**



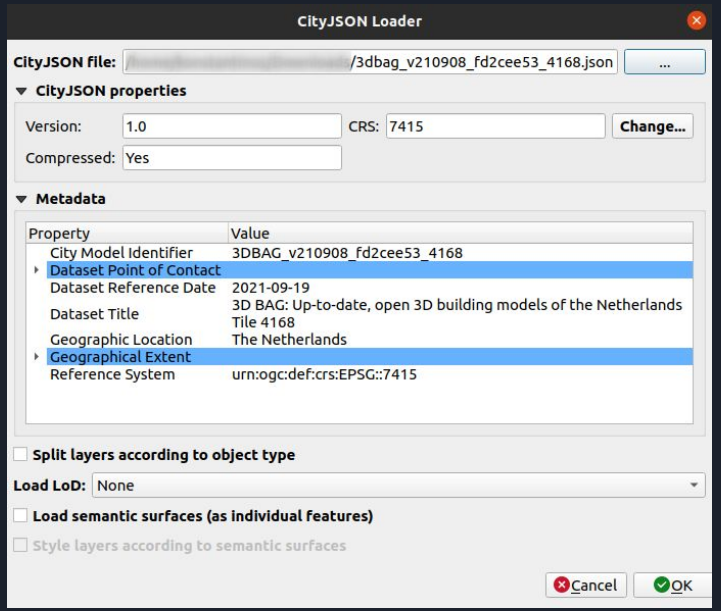

Functionalities:

- Load data from database

Limitations:

- Loads ONLY the “Building” module (of all representations)
- ONLY the geometries of “Building” module classes are loaded.
- Works only for the default “citydb” scenario.
- Does not account for multiple users/privileges.
- Does not handle data size
- Confusing GUI (IMO)

# Related work (3/3)




**QGIS plugin: “CityJSON Loader” (23-03-2021)**

Functionalities:

- Load only CityJSON files
- Allow for multiple layer representations
- Allow use of standard color schema

Limitations:

- It is not possible to store updates back at the original file.
- Does not handle data size

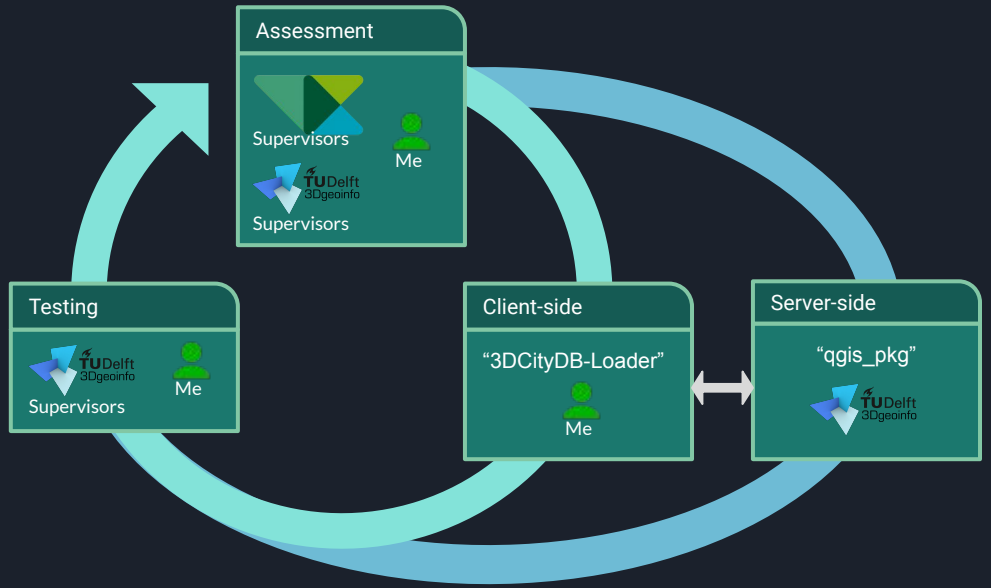
# Methodology (1/2): Requirements

Explore/discover requirements:

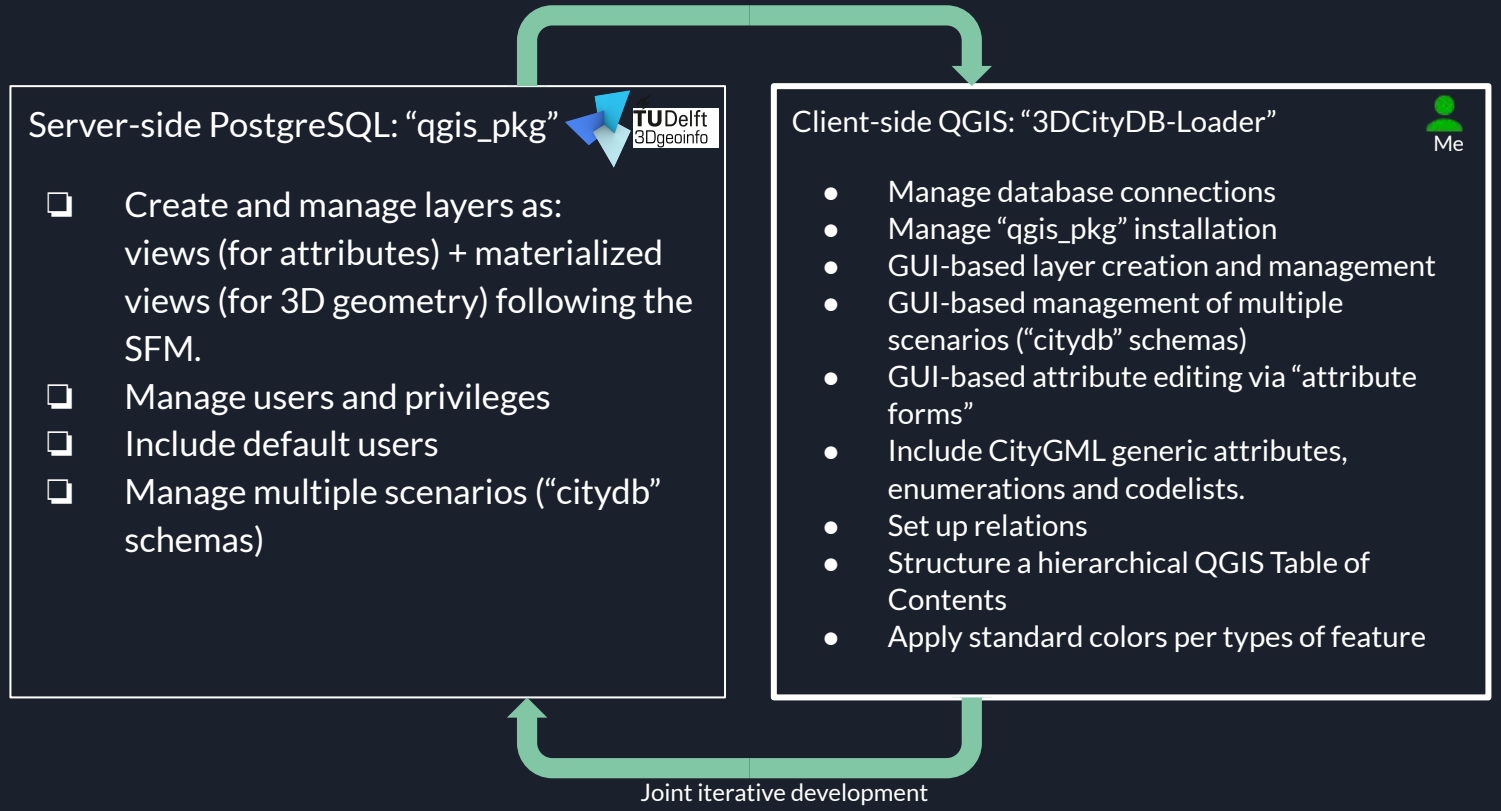
- Meetings with “TU Delft” and “VirtualCitySystems” supervisors
- Iterative process (4 iterations)

Primary requirements:

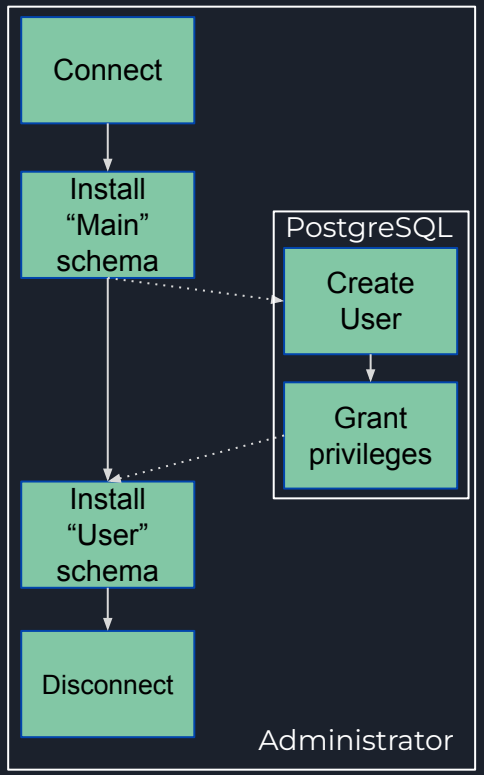
- “Data layers” able to interact with 3DCityDB data
- Multi-user with different privileges (Read-only, Read-write)
- Allow multiple scenarios (“citydb” schema).
- Edit ONLY attributes, no geometries
- Deal with multi-LoD, multi-geometry representations
- Operate from a GUI (QGIS)

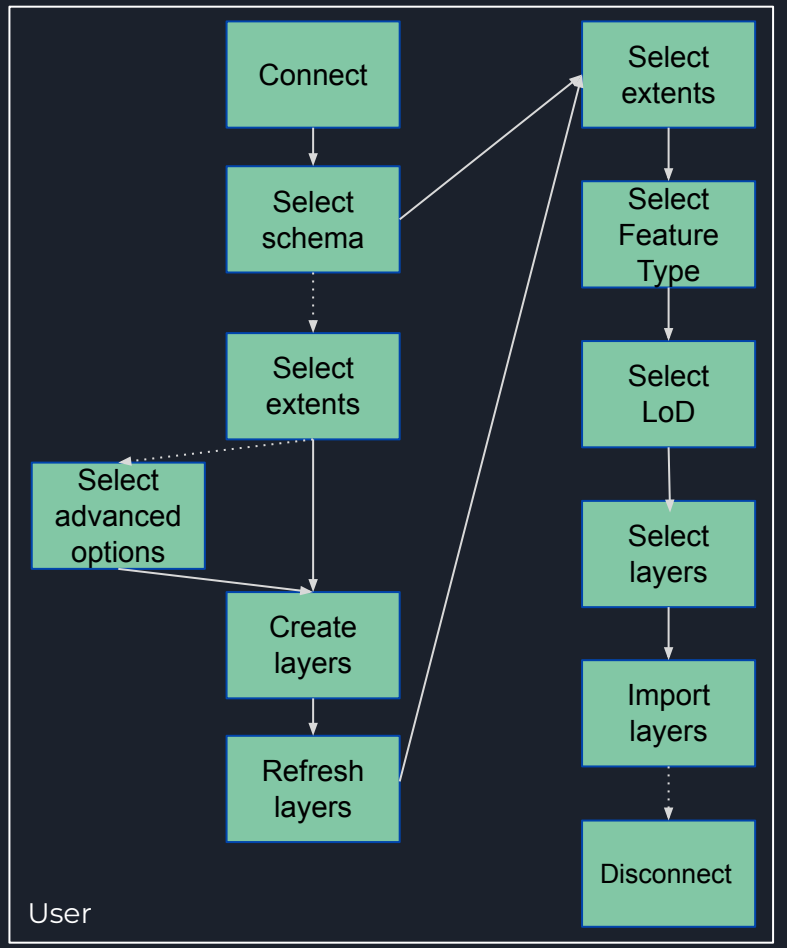


# Methodology (2/2): Design decisions

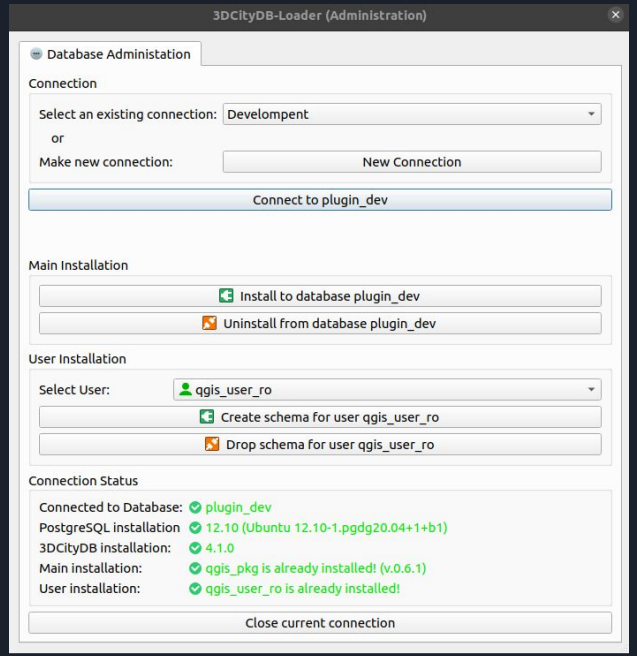


# “3DCityDB-Loader” Workflow

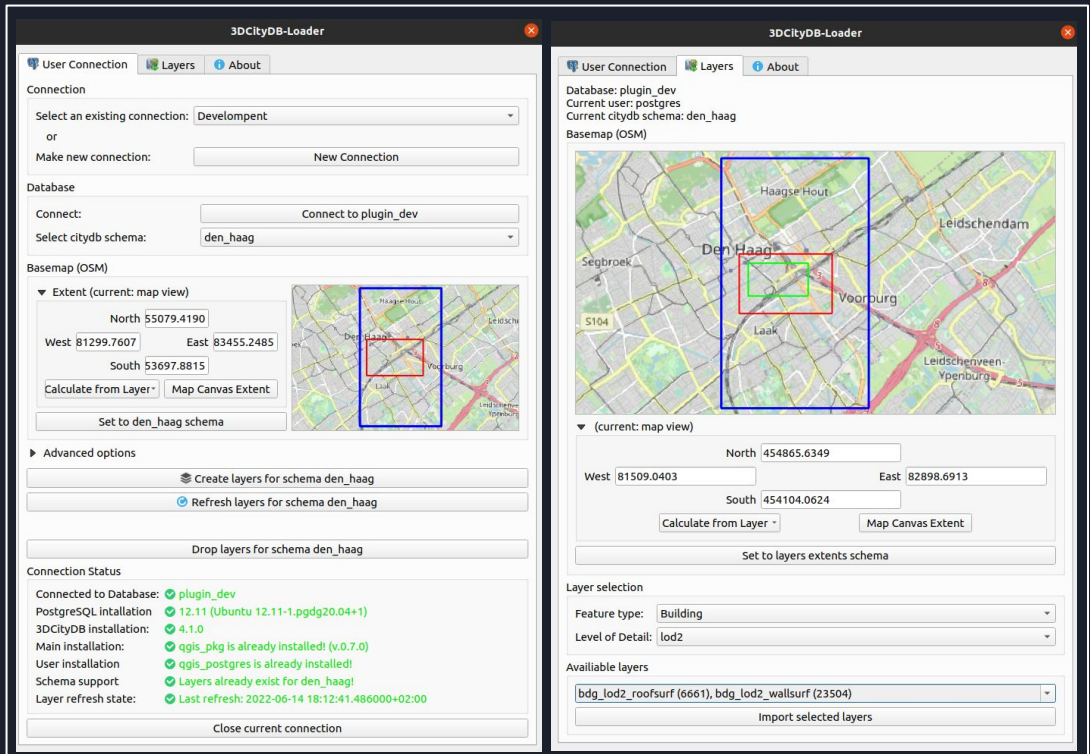


# “3DCityDB-Loader” GUI



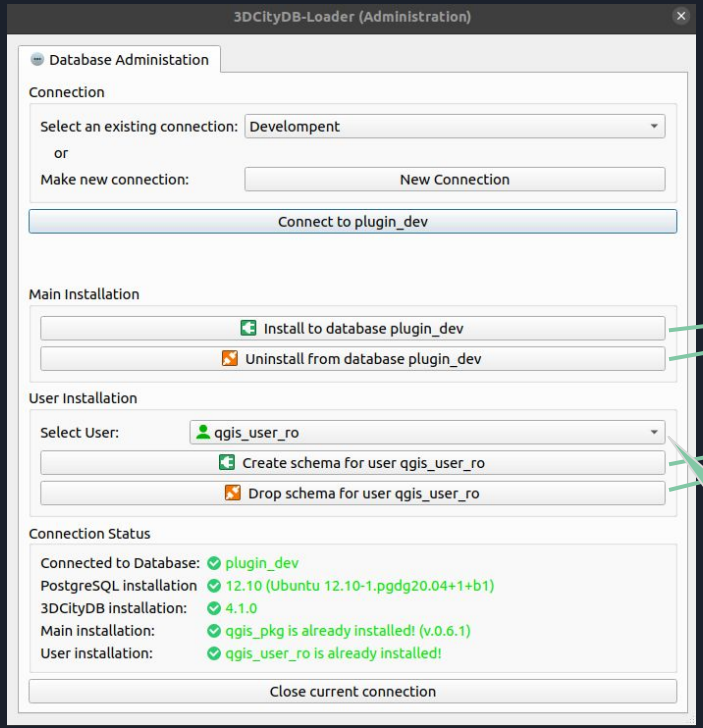

Administrator

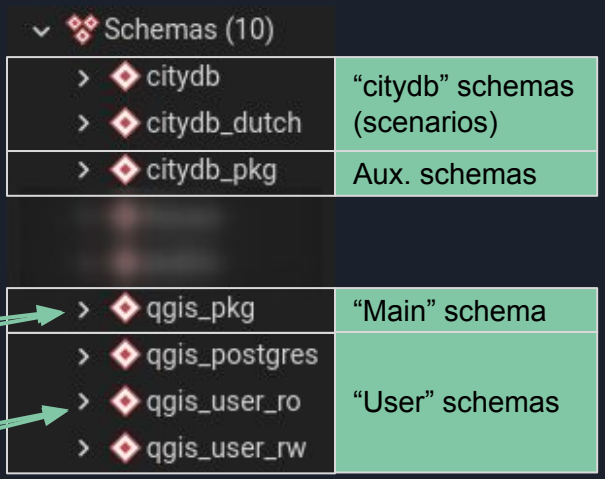


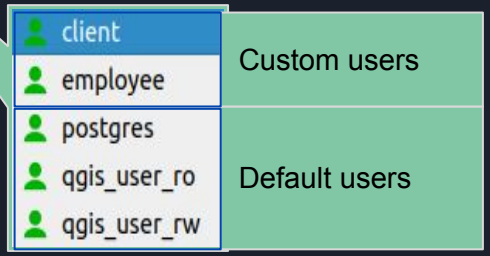

User

# “3DCityDB-Loader” in practice (1/7)

Database administration example

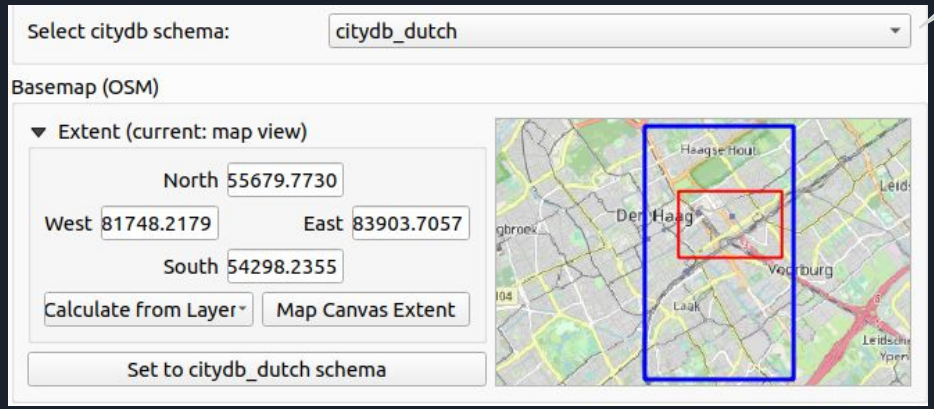


# “3DCityDB-Loader” in practice (2/7)

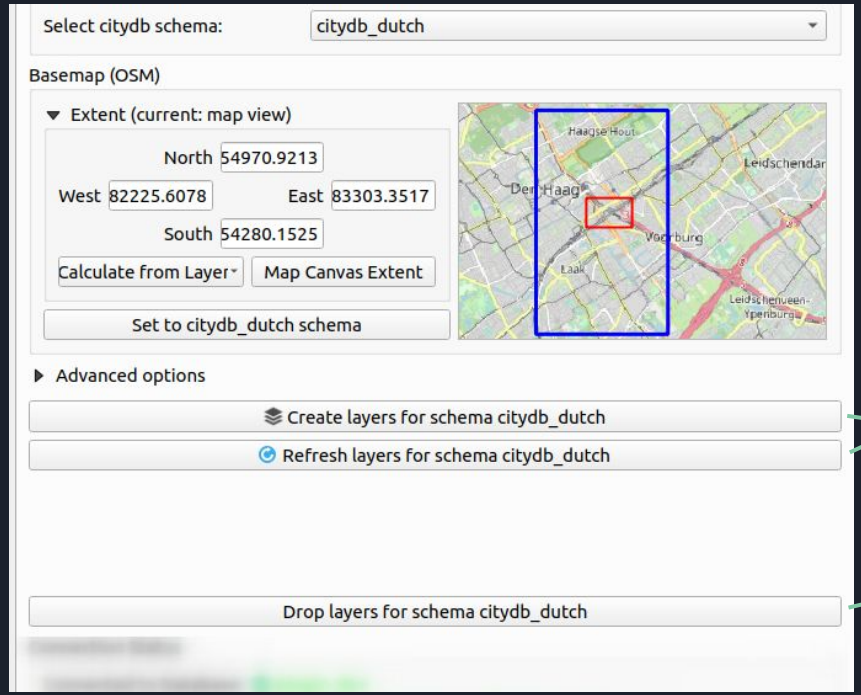
Use example as database regular user

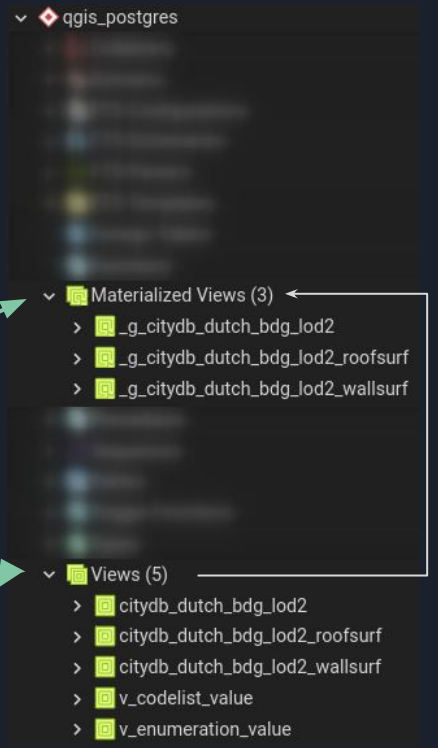



- Blue square: scenario extents (“citydb” schema in database - e.g. citydb_dutch)
- Red square: User-defined layer extents (layers in database)

# “3DCityDB-Loader” in practice (3/7)

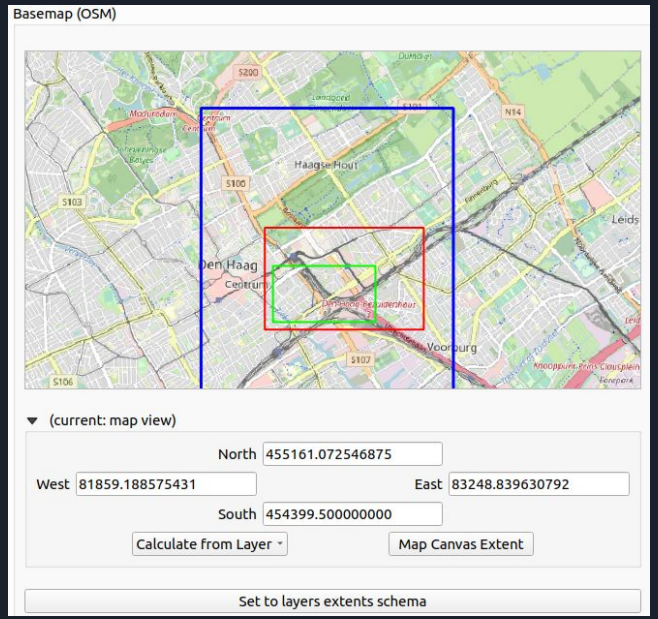
Use example as database regular user



# “3DCityDB-Loader” in practice (4/7)

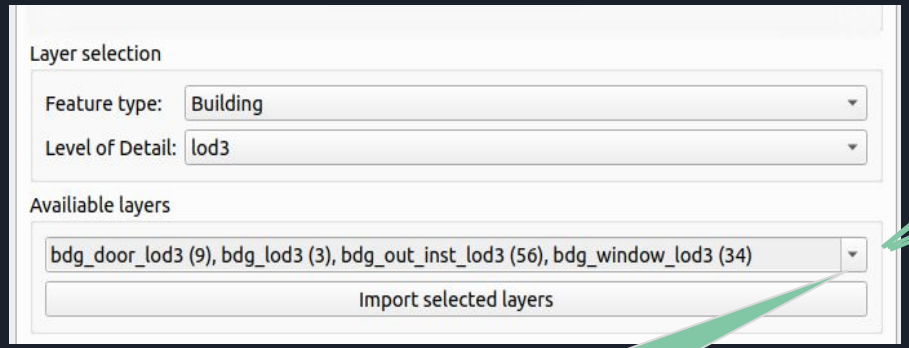
Use example as database regular user



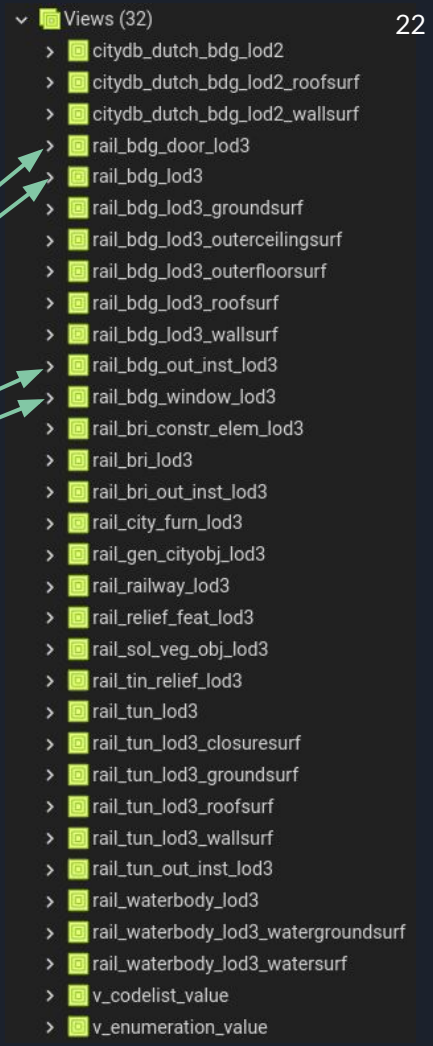

- Blue square: scenario extents (“citydb” schema in database - e.g. citydb_dutch)
- Red square: layer extents (layers in database)
- Green square: User-defined layer extents (QGIS layers)

# “3DCityDB-Loader” in practice (5/7)

Use example as database regular user

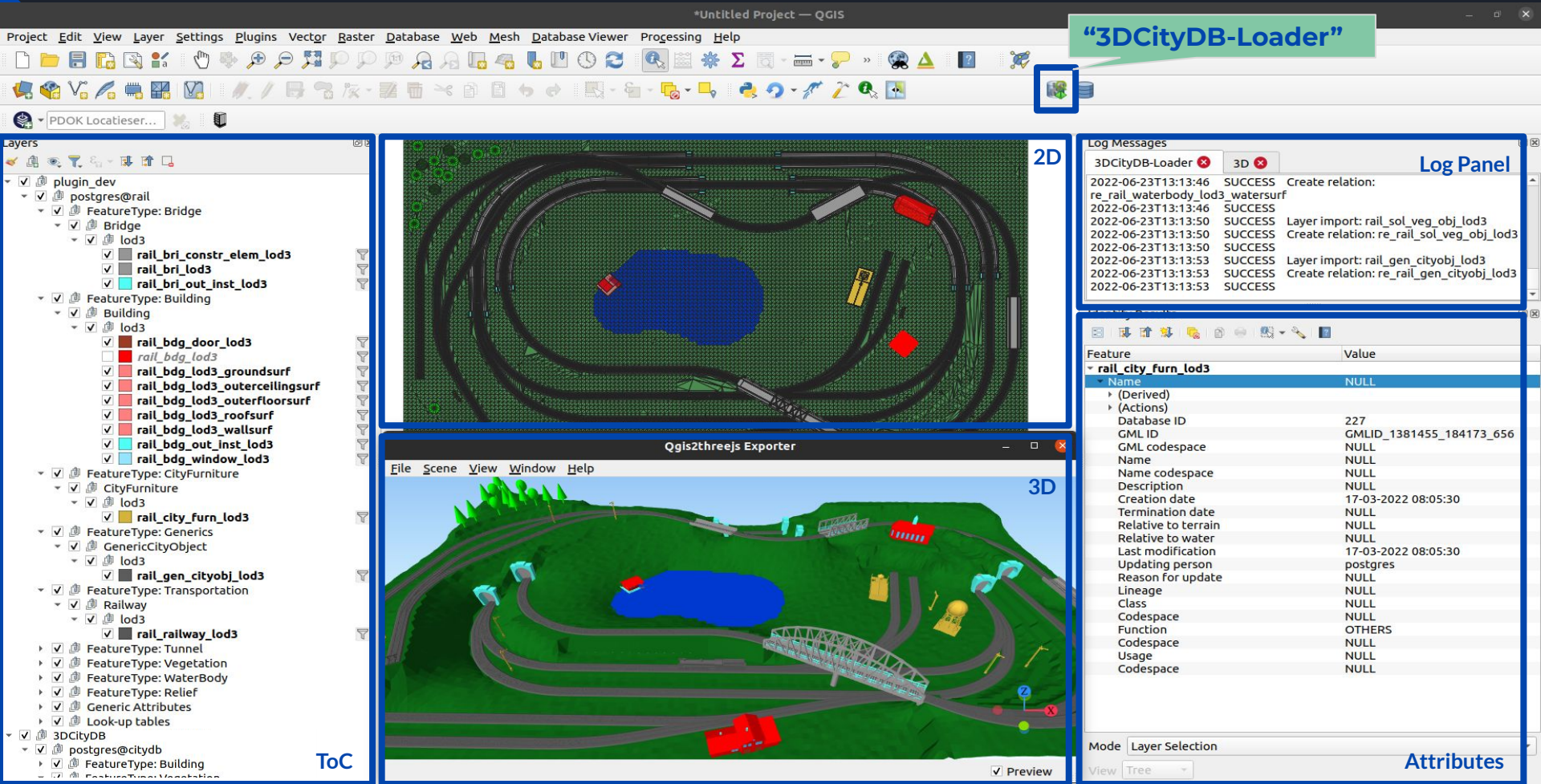




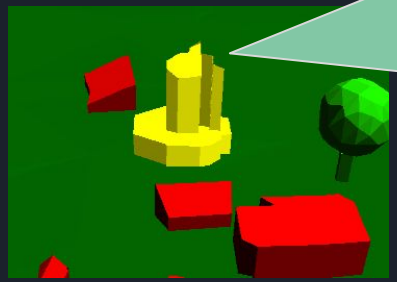


# “3DCityDB-Loader” in practice (6/7)





# “3DCityDB-Loader” in practice (7/7)



citydb_bdg_lod2 - Feature Attributes

Main Info | Database Info | Relation to surface | Generic Attributes

| | |
|---|---|
| Database ID | 24918 |
| GML ID | NL.IMBAG.Pand.1742100000008466 |
| GML codespace | NULL |
| Name | Building 480-835 |
| Name codespace | NULL |
| Description | NULL |

Class | Function | Usage

| | |
|---|---|
| Class | Non-residential (multi function) |
| Codespace | NULL |

**Feature-specific attributes**

| | | | |
|---|---|---|---|
| Year of construction | 1889 | Year of demolition | NULL |
| Storeys above ground | 1 | Storeys below ground | -5 |
| Height | 45 | UoM | m |
| Storey height above ground | 3 | UoM | NULL |
| Storey height below ground | 3 | UoM | m |
| Roof type | slanted | Codespace | NULL |

Cancel | OK

Attribute form

# Conclusions (1/5)

How to simplify user interaction with the 3DCityDB in terms of:

- Visualization in 2D/3D of multi-LoD geometries. ✅
- How to access attributes and edit them (if user is allowed)? ✅

In terms of implementation:

- How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)? ✅
- Can we “restructure” data in 3DCityDB in order to conform to the SFM? ✅

# Conclusions (2/5)

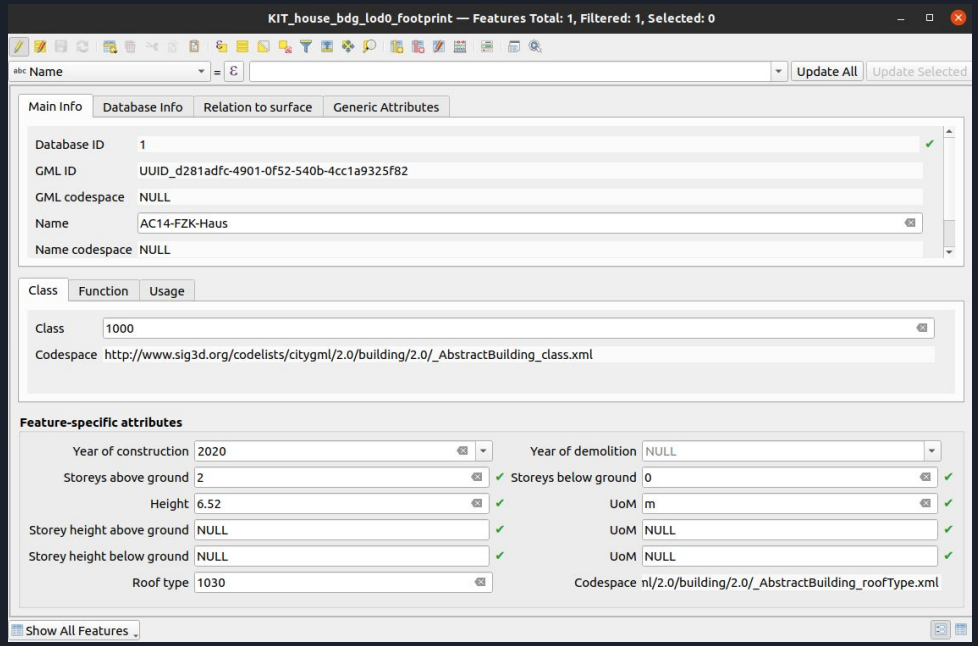
How to simplify user interaction with the 3DCityDB in terms of:

- **Visualization in 2D/3D of multi-LoD geometries.** ✅
- How to access attributes and edit them (if user is allowed)? ✅

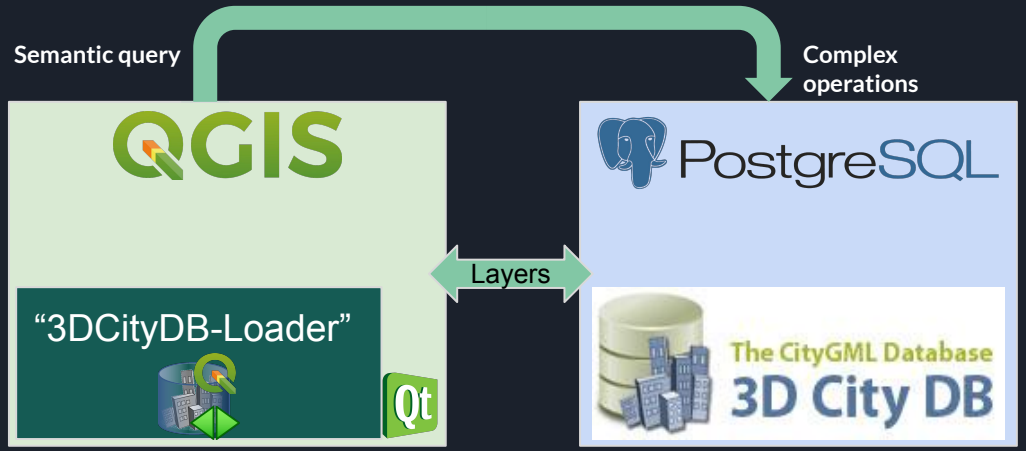
# Conclusions (3/5)

How to simplify user interaction with the 3DCityDB in terms of:

- Visualization in 2D/3D of multi-LoD geometries. ✅
- **How to access attributes and edit them (if user is allowed)? ✅**



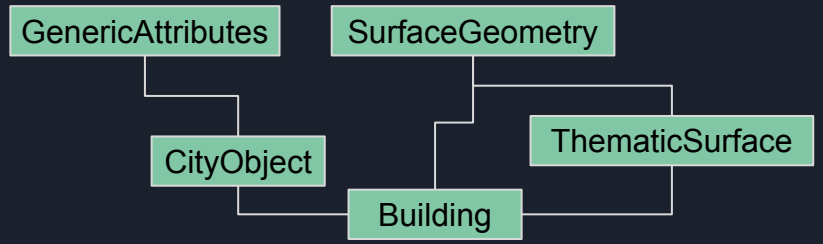
# Conclusions (4/5)

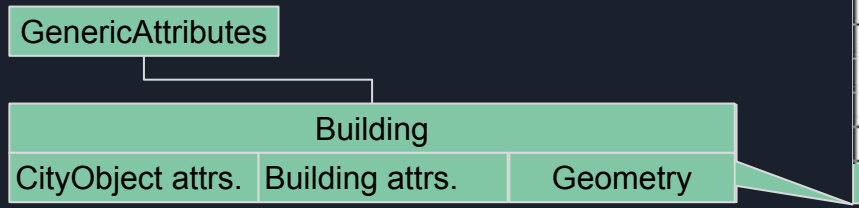



In terms of implementation:

- **How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)? ✅**
- Can we “restructure” data in 3DCityDB in order to conform to the SFM? ✅

# Conclusions (5/5)



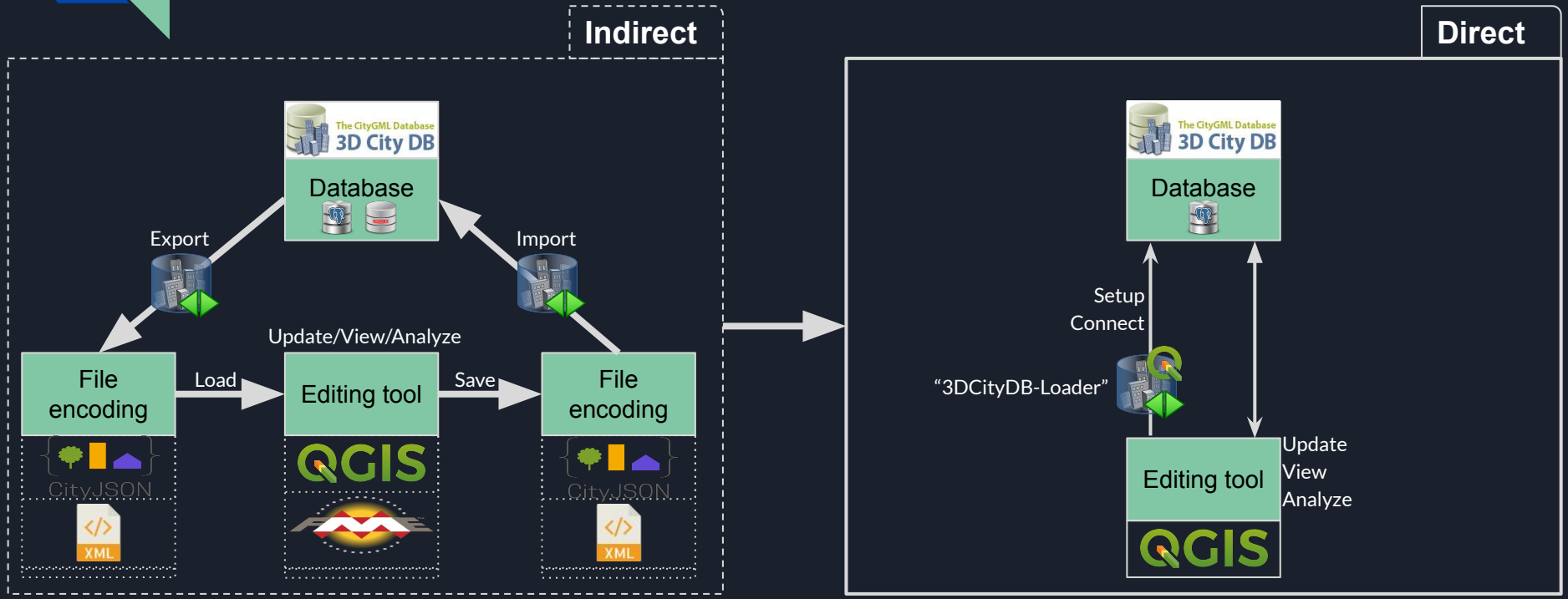




In terms of implementation:

- How to create a bi-directional interface between QGIS GUI (front-end) and the 3DCityDB (back-end)? ✅
- **Can we "restructure" data in the 3DCityDB in order to conform to the SFM?** ✅

| # | Field name | Field type |
|---|---|---|
| 1 | id | Feature atribute |
| 2 | gmlid | Feature atribute |
| 3 | ….. | Feature atribute |
| 15 | class | Feature atribute |
| 16 | class_codespace | Feature atribute |
| 17 | ….. | Feature atribute |
| 33 | geom | Feature geometry |

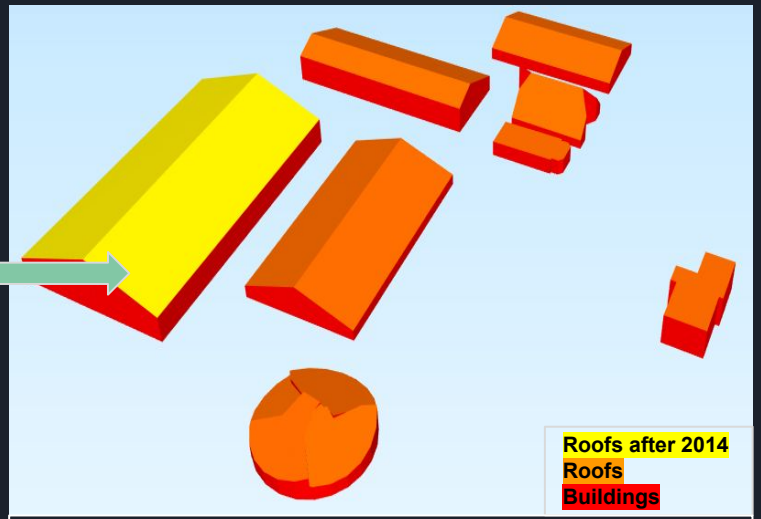
SFM layer example


Indirect
The CityGML Database
3D City DB
Database
Export
Import
Update/View/Analyze
File encoding
Load
Editing tool
Save
File encoding
CityJSON
QGIS
CityJSON
XML
XML
Direct
The CityGML Database
3D City DB
Database
Setup
Connect
"3DCityDB-Loader"
Editing tool
Update
View
Analyze
QGIS

# Example query of building roofs constructed after the year 2014.

```sql
SELECT
    ts.id AS roof_id,
    co_ts.gmlid AS roof_gmlid,
    b.id AS building_id,
    co.gmlid AS building_gmlid,
    b.year_of_construction,
    ST_Collect(sg.geometry) AS roof_geom
FROM
    citydb.thematic_surface AS ts
    INNER JOIN citydb.cityobject AS co_ts
        ON (co_ts.id = ts.id)
    INNER JOIN citydb.surface_geometry AS sg
        ON (ts.lod2_multi_surface_id = sg.root_id)
    INNER JOIN citydb.building AS b
        ON (b.id = ts.building_id)
    INNER JOIN citydb.cityobject AS co
        ON (co.id = b.id)
WHERE
    ts.objectclass_id = 33 AND -- roofsurfaces
    b.objectclass_id = 26 AND -- buildings
    b.year_of_construction >= '2015-01-01'::date
GROUP BY
    ts.id,
    co_ts.gmlid,
    b.id,
    co.gmlid,
    b.year_of_construction
ORDER BY
    b.id,
    ts.id;
```

*Using vanilla 3DCityDB*

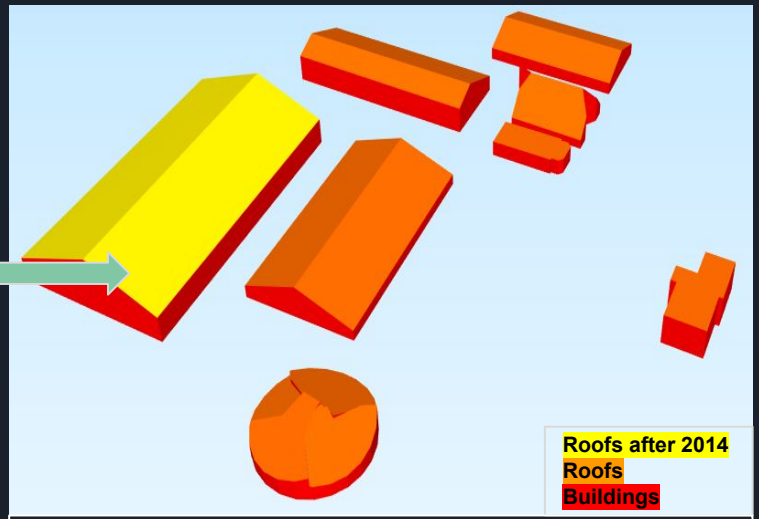



*this queries all of the roofs stored in the database scenario. The image shows only a small sub-area

# Example query of building roofs constructed after the year 2014.

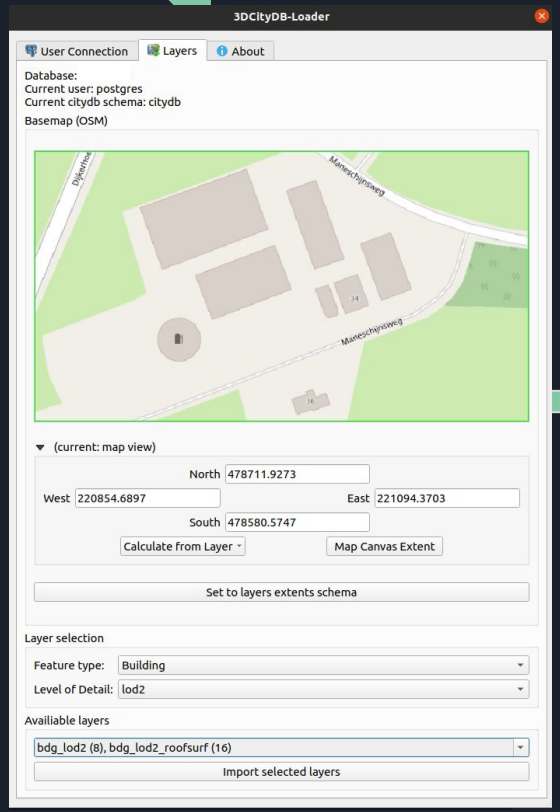
```sql
SELECT
    rs.id AS roof_id,
    rs.gmlid AS roof_gmlid,
    rs.building_id AS bdg_id,
    b.gmlid AS bdg_gmlid,
    b.year_of_construction,
    rs.geom AS roof_geom
FROM
    qgis_user_ro.citydb_bdg_lod2_roofsurf AS rs
    INNER JOIN qgis_user_ro.citydb_bdg_lod2 AS b
        ON b.id = rs.building_id
WHERE
    b.year_of_construction >= '2015-01-01'::date
ORDER BY
    b.id,
    rs.id;
```

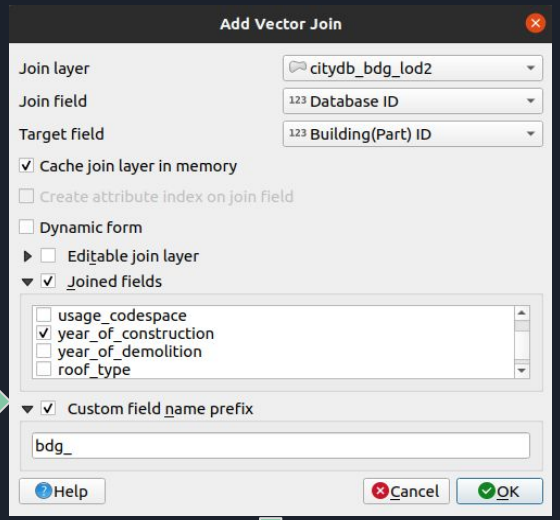
*Using server-side of "3DCityDB-Loader"*



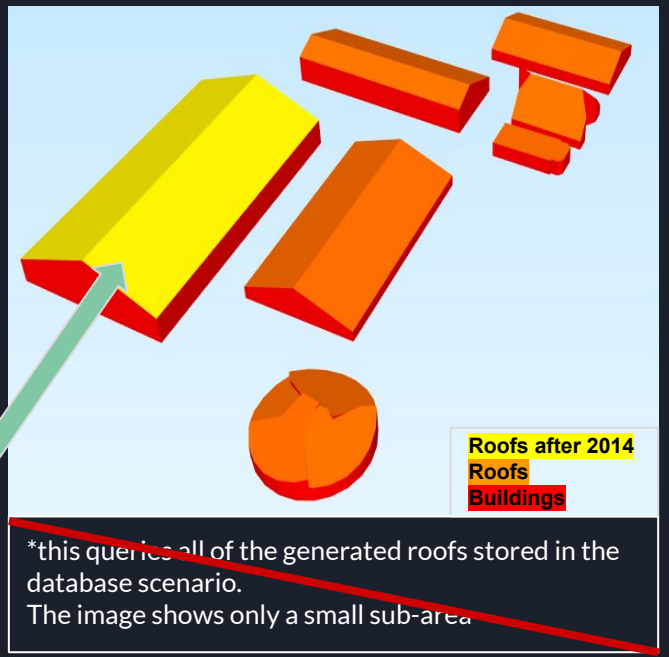

*this queries all of the generated roofs stored in the database scenario.
The image shows only a small sub-area

# Example query of building roofs constructed after the year 2014.




```
"bdg_year_of_construction" >= make_date(2015,1,1)
```

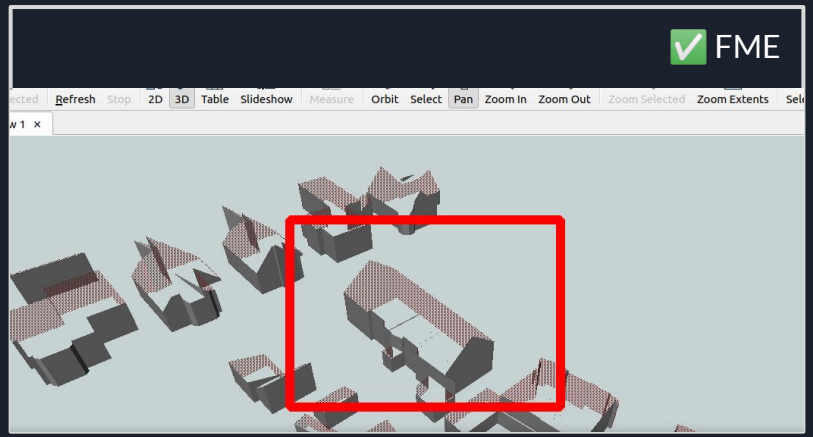



*this queries all of the generated roofs stored in the database scenario.
The image shows only a small sub-area

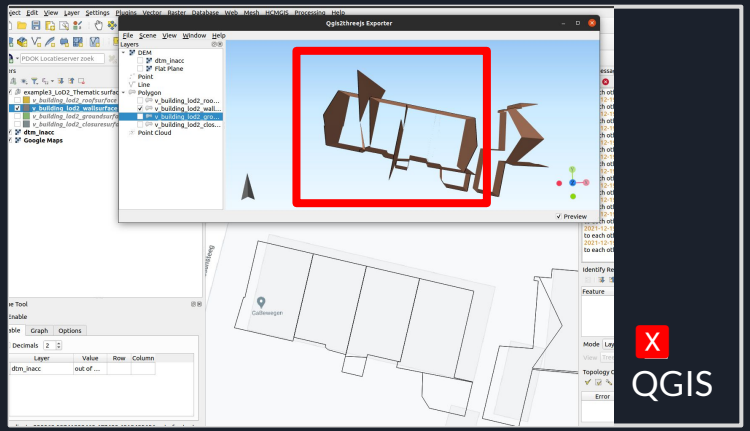
# Limitations

- Not developed to insert or update geometries
- Custom users (along with privileges) currently can only be set outside of the GUI workflow
- No maximum size limit for layers (large datasets)
- Base-map tool to select extents by drawing a square on map is unavailable
- Potential issues with 3D visualization due (probably) to precision loss from projected coordinates

# 3D Visualization issue







# Outlook

- ❏ Further testing with more and heterogeneous data-sets
- ❏ Add management of the CityGML “Appearances” class
- ❏ Collect more feedback from selected early adopters
- ❏ Explore and add new functionalities
  (tracked in the project’s GitHub repository)

# Feedback from test users

**22 Open** ✓ 14 Closed

**Some proposals for improvement**
#37 opened 2 days ago by msanchezaparicio

**Add option to Delete and Edit existing connection** enhancement
#36 opened 2 days ago by Konstantinos-Pantelios

**QGIS 3.22: it doesnt work**
#35 opened 9 days ago by msanchezaparicio

**Add check for QGIS version before loading** enhancement
#34 opened 10 days ago by gioagu

**QGIS 3.20: it works**
#33 opened 11 days ago by gioagu

**QGIS 3.24: it works**
#32 opened 11 days ago by gioagu

**QGIS 3.16: Cannot load main puglin window (for user)** bug
#31 opened 11 days ago by gioagu

**QGIS 3.18: Cannot import selected layers** bug question
#30 opened 12 days ago by yaozhihang

## Some proposals for improvement #37

Open **msanchezaparicio** opened this issue 12 days ago · 0 comments

**msanchezaparicio** commented 12 days ago

As a beginner on 3DCityDB - Loader, i suggest some improvements that would be interesting:

- At least in me case, i have to activate the tool manually in the toolbar. Its would be interesting if this step were automatic.
- Have a different icon for the administration than for the normal user to be able to identify them without any problems.
- If the user decide to desinstall the plugins, give the possibility to delete all the connections created.
- In the *administration window*:

```
1.  A botton to delete or edit connections already created.
2. When the user creates a new connection, in the window "Establish new postgreSQL connection", an
3. Once the values have been introduced in the different line edit, it would be nice that when cor
```

I hope it will help you

TU Delft

Thank you for your attention!