Report

# 1. UNSW Lower Campus

## 1.1 Datasets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Raw Type | Data | Size | #Voxels | Description |
| GIS | dtmbot.xyz | 9.22 GB | 641,624,355 | A terrain with holes, in which the buildings fit |
| tree.xyz | 906 MB | 59,640,000 | Tree in lower campus |
| bld1-54.xyz (except for 26 and 46) | 3.52GB | 241,613,693 in total and 4,646,418 per building | 52 buildings in lower campus |
| BIM | be.xyz | 249 MB | 17,460,029 | Built Environment (H13) |
| blockhouse.xyz | 45.4 MB | 3,392,202 | Blockhouse (G6) |
| dalton.xyz | 25.5 MB | 1,887,512 | Dalton (F12) |
| quadrangle.xyz | 43.9MB | 3,161,733 | Quadrangle (E15) |
| roundhouse.xyz | 79.9MB | 6,037,174 | Roundhouse (E6) |
| scithe.xyz | 17.2MB | 1,231,821 | Science Theatre (F13) |

Note that:

* For GIS-based voxels, its resolution is 20cm. All voxels are recorded in same relative coordinate with offset (336000, 6245250, 20).
* For BIM-based voxels, its resolution is 10 cm. Each building is in its own relative coordinate with MINXYZ.

## 1.2 Database Schema

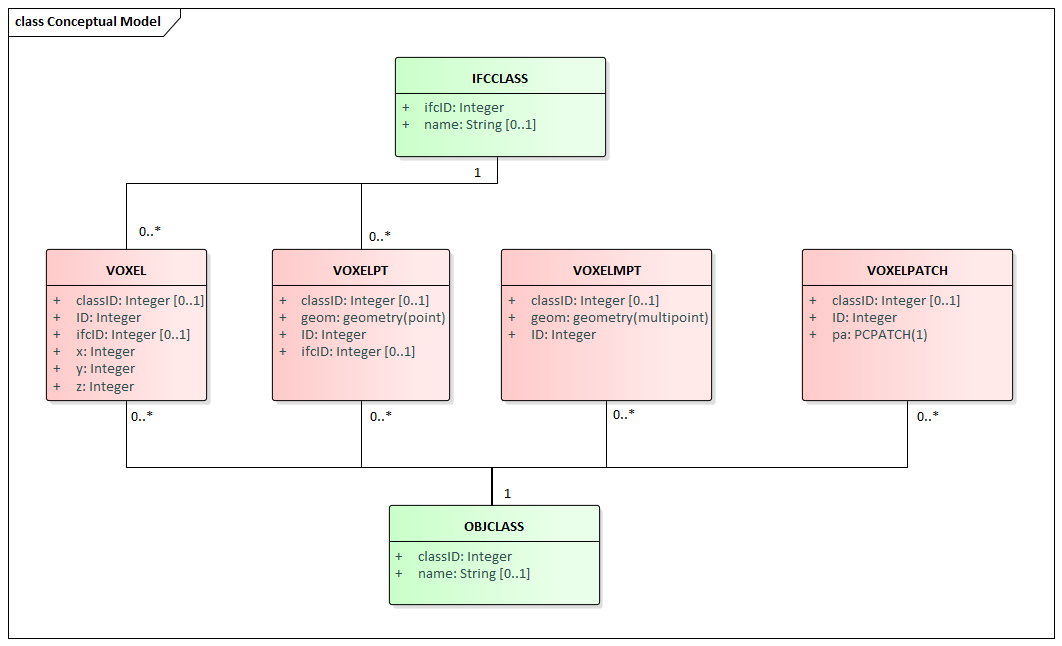


Figure 1. Conceptual Model

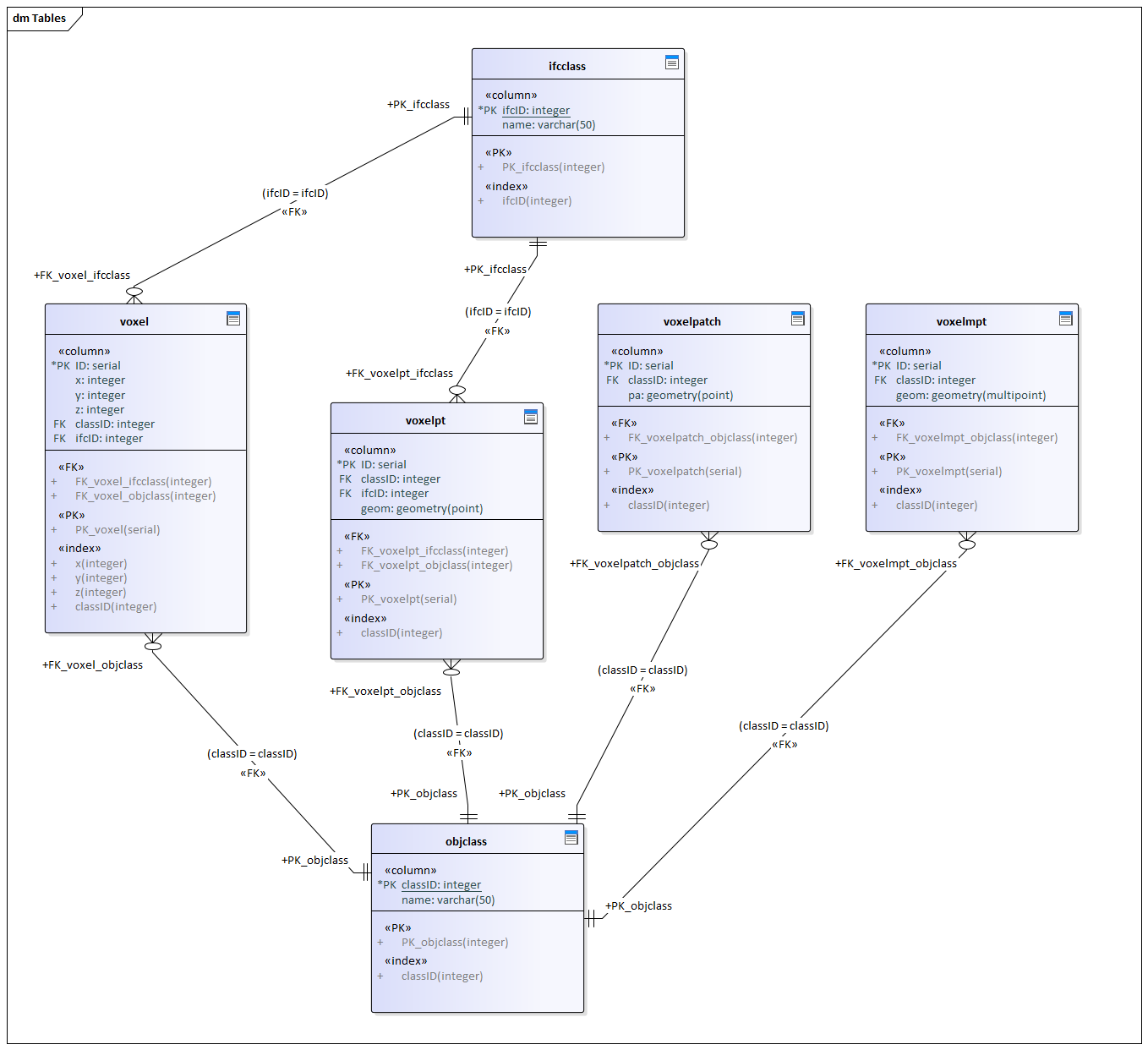


Figure 2. Physical Model

## 1.3 Data Layout in PostgreSQL

Four Main Tables for voxel storage:

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Columns | Geometry | Description |
| voxel | id, x, y, z, classid, ifcid | N/A | One voxel per row/record. |
|  | | | |
| voxelpt | id, classid, ifcid, geom | POINT(x,y,z) | One voxel per row/record with geometry. |
|  | | | |
| voxelmpt | id, classid, geom | MULTIPOINT | One building per row/record.  20,000,000 point per row/record for “tree” and “dtm”. |
|  | | | |
| voxelpatch | id, classid, pa | PCPATCH(1) | One building per row/record.  20,000,000 point per row/record for “tree” and “dtm”. |
|  | | | |

Two Semantic Tables for IFC and class info:

|  |  |  |
| --- | --- | --- |
| Table | Columns | Description |
| ifcclass | ifcid, name | 26 IFC class with corresponding name |
|  | | |
| objclass | classid, name | classid 1-54 (except for 26 and 46) are building ID.  classid 55 is tree ID.  classid 56 is dtmbot ID. |
|  | | |

# 2. QGIS Visualization

The data query is processed using a HP laptop. Its processor is Intel(R) Core (TM) i7-7600 CPU @ 2.80GHz and its installed memory is 16.0 GB. Its operating system is 64-bit Windows 10. And the test is performed on PostgreSQL (11.2), PostGIS (2.5.2), and QGIS (3.6.1).

## 2.1 Simple Visualization

Considering bld52 (we don’t have building name at this moment), 29.6MB and 1,881,847 voxels, we extract “geom” first, and then convert its coordinate into EPSG:28356.

1. **SELECT** ST\_MakePoint(336000+ST\_X(geom)\*0.2, 6245250+ST\_Y(geom)\*0.2, 20+ST\_Z(geom)\*0.2) **AS** geom
2. **FROM** voxelpt
3. **WHERE** classid=52;

Figure 3 shows how to execute query in QGIS.

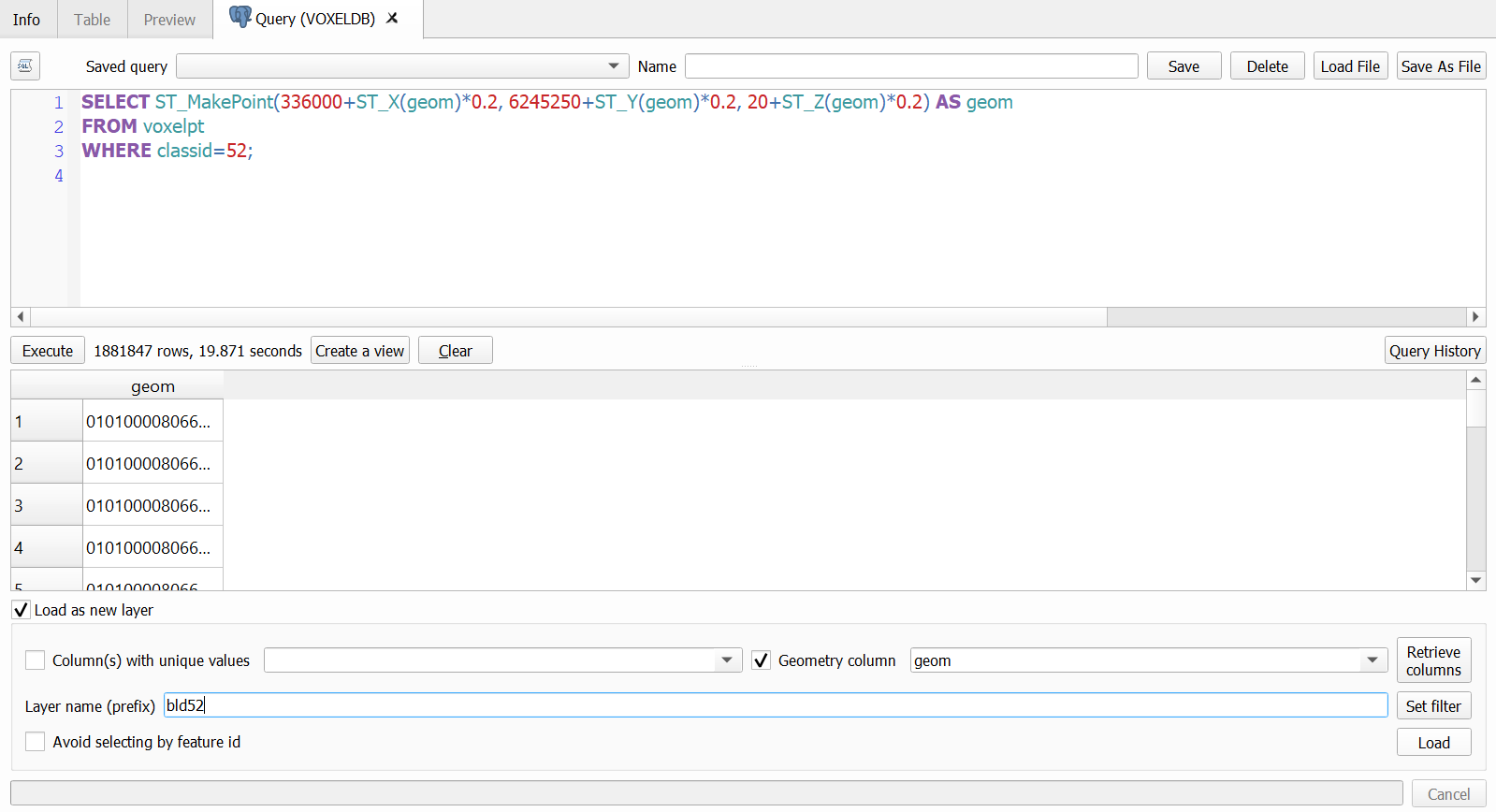


Figure 3. Query execution in QGIS

Note that, loading the above query result as a new layer in QGIS may take several minutes and 3D view is as well. Once choosing 3D view, please keep an eye out for your GPU and memory changes, if you crash, kill the task or stop doing the work at hand and continue to wait patiently. If not necessary or you are not confident in your PC, don’t try 3D view.



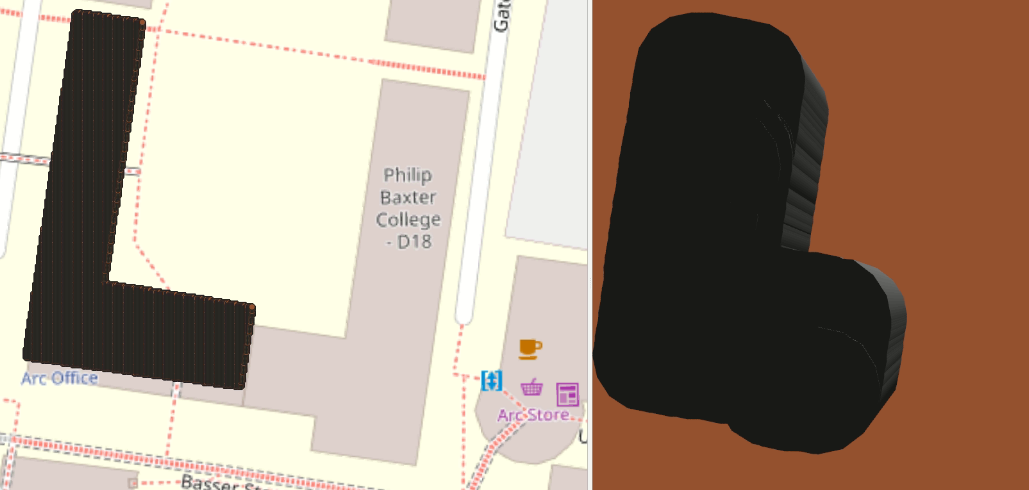


Figure 3. 2D and 3D visualization of “bld52” in QGIS