3D City DB and Database Bench marking

Agnes Folga*
University of Illinois Chicago

Fahad Ahmad†
University of Illinois Chicago

ABSTRACT

In this paper, we bench mark common operations performed on 3D City Database, a 3D geo database to store, represent and manage 3d city models on top of a relational database. The common operations were performed were

- selecting and dragging any box on a map
- export all the buildings present within that selected box
- the exported data, in a GML format can be viewed by any 3d GML viewer
- Finding all buildings with a height over 50 feet

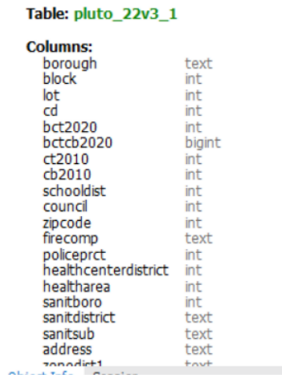
1 INTRODUCTION

In this paper, we examine 3D City Database, a geo database solution to store, represent, and manage virtual 3D city models on top of the spatial relational database systems' . Existing query languages for Building Information Modeling (BIM) only performed comparisons on individual attributes of those in object-oriented mode [1].

2 PRIOR WORK

Before investigating and analyzing 3DCity Database, we multiple examined prior works which queried and discussed other 3d Data. We also pulled the 3dcitydb [2] and CityGml [3] GitHub repositories. There, the developers have put in some documentation and examples to show how they set up the database with the data sets. There is also a quite extensive tutorial written by the original developers, which outlines in detail the method they chose [4]. While this guide is very comprehensive, it is easy to get lost in it as it is 99 pages. The first paper examined was 'Querying 3D Cadastral Information from BIM Models'. In this paper, the authors aimed to develop BIM (building information modeling) based queries for interrogating questions about the legal ownership of properties inside multistory buildings. [5] If there is more interest in prior work, please refer to papers named 'Geospatial Data Management Research' [6] and 'A Spatial Query Language for 3D Building Models and 3D City Models' [7].

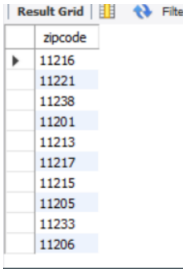
Along with studying prior research papers, we performed basic MySQL operations on a New York City Pluto Dataset in CSV format. Pluto is 'Extensive land use and geographic data at the tax lot level in comma–separated values (CSV) file format. The PLUTO files contain more than seventy fields derived from data maintained by city agencies.' This is quite a large dataset, so it was decided to trim it and take only the first 1000 rows (instead of 856,000). Before loading the data into MySQL WorkBench, we inspected the data and looked to see what kind of queries could be performed. After inspection and loading into WorkBench, The schema should look like this:



The first query we tested on the NYC Pluto Schema was selecting distinct zipcodes from New York City neighborhoods where the building had 2 floors, was in Brooklyn ('BK') and in school district 13 :

```
select distinct zipcode from pluto_22v3_1
where numfloors=2 and schooldist=13 and borough='BK';
```

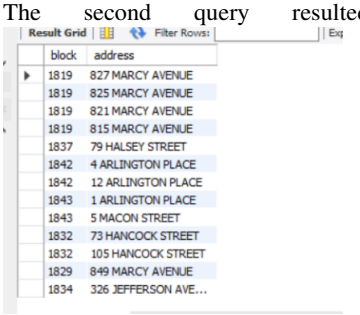
Here is the result of running the first query:



The second query we ran on the NYC Pluto data set was selecting the block and address from New York City neighborhoods that were in the Bedford Historical District

```
select distinct block, address from pluto_22v3_1
where histdist='Bedford Historic District';
```

The second query resulted in the following:

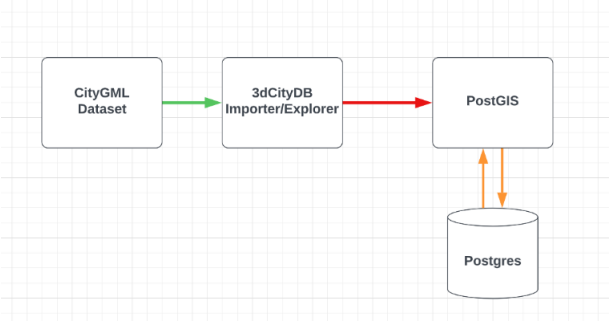


3 METHODOLOGY

In this section, we will outline how we conducted the operations on 3DCity DB. The data set used was linked in the

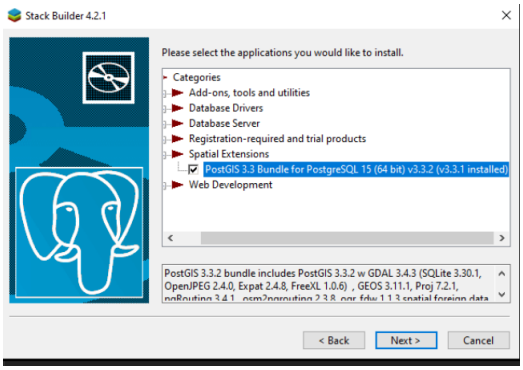
*e-mail:afolga3@uic.edu
†e-mail:fahmad27@uic.edu

original tutorial, and the download is linked in the references [8]. We illustrate the pipeline used to query the data here:

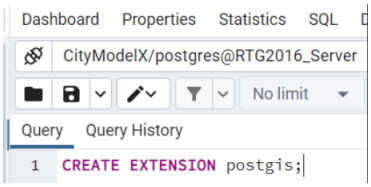


The process is outlined here:

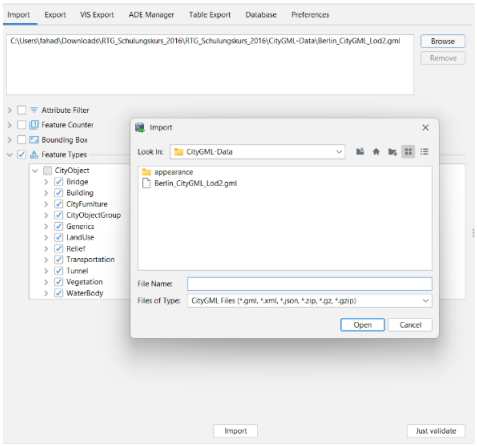
1. Set up PostGres, PostGis



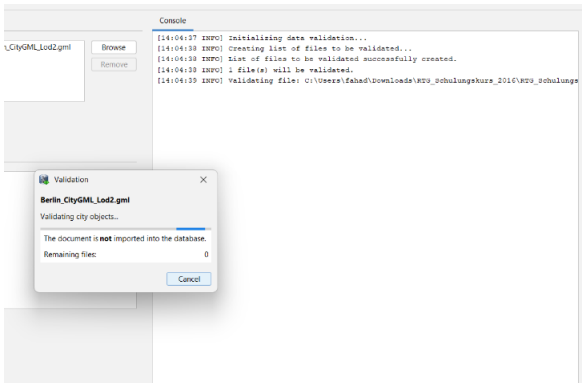
2. Enable PostGis Extension into database



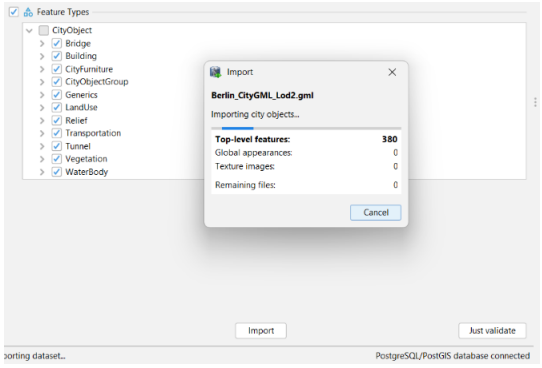
3. Load any GML Data into 3D City Importer



4. Validate data

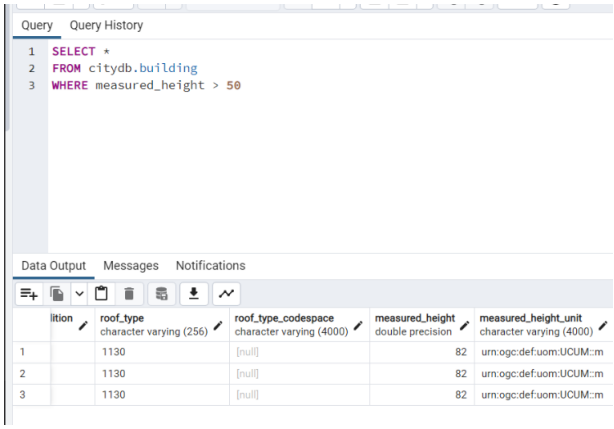


5. Import data into PostGres



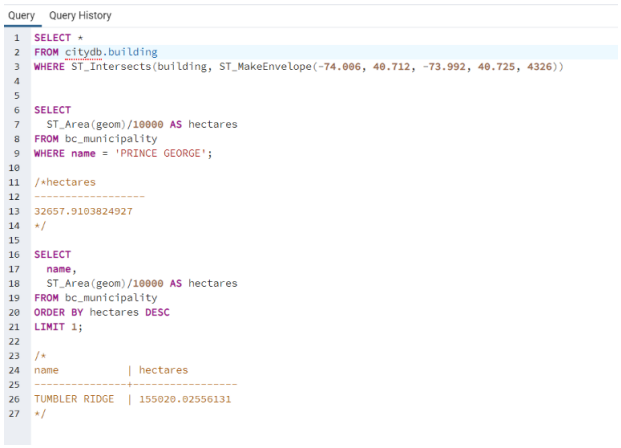
6. Run queries in PostGres on the data The following sample query was performed,

```
select * from citydb.building where measured_height>50;
```



4 RESULTS

In this section, we show what queries can be made in PostGres using the 3DCity DB importer/exporter once everything is set up properly. We have performed the following sample queries:



```
Query  Query History
1  SELECT *
2  FROM citydb.building
3  WHERE ST_Intersects(building, ST_MakeEnvelope(-74.006, 40.712, -73.992, 40.725, 4326))
4
5
6  SELECT
7    ST_Area(geom)/10000 AS hectares
8  FROM bc_municipality
9  WHERE name = 'PRINCE GEORGE';
10
11 /*hectares
12 ------------------
13 32657.9103824927
14 */
15
16 SELECT
17   name,
18   ST_Area(geom)/10000 AS hectares
19 FROM bc_municipality
20 ORDER BY hectares DESC
21 LIMIT 1;
22
23 /*
24 name            | hectares
25 ---------------+-----------------
26 TUMBLER RIDGE  | 155020.02556131
27 */
```

5 Challenges

When conducting these operations on 3DCity DB,we encountered some issues and challenges. First, the space and memory requirements to download the datasets, load them into Postgres, and query them can be very large. The data extraction step can be very computational expensive for personal machines. Another challenge we faced was difficulty with setting up Postgres with the proper 3DCity DB extension. In order to properly set up the PostGres database, we consulted the hands on tutorial. However, this turned out to be quite a tedious process as this tutorial is very long and has some links which were disabled/not working. We hope with our GitHub, future users will face less difficulty.

6 Conclusion

In this paper, we outlined a baseline process for examining and querying 3D Data using PostGis with the 3D City DB extension. We were able to successfully connect to the 3D City Importer/Exporter. Throughout the course of the semester we were able to create an easy to follow, up to date document for anyone who wants to get started with PostGres, PostGIs and GML datasets. We queried the City GML data that we originally planned on using in the proposal. We hope that this guide can be extended to other 3D datasets and databases.

7 GitHub Repository

Our GitHub is available here: `https://github.com/readysetgit24/citygml-docs`

Acknowledgments

We would like to most importantly thank our mentor and professor Dr. Fabio Miranda for his guidance, evaluations, and patience throughout the course of us conducting this project. We would like to thank our peers for their suggestions and comments throughout the semester.