

## Table of Contents

Short description of BK database: .....	1
Using OpenCities Map.....	3

## Short description of BK database:

The database contains the building of the Bouwkunde faculty of the Technical University Delft. The database has been created on PostgreSQL and the tables of the database had three different sources as shown in Figure 1:

- 1- Tables related to Revit software (Semantic and schedule information).
- 2- Tables related to IFC model (3D geometry).
- 3- Tables related to LADM-IndoorGML model (Integration model to database).

### • Generating Database and visualizing the data

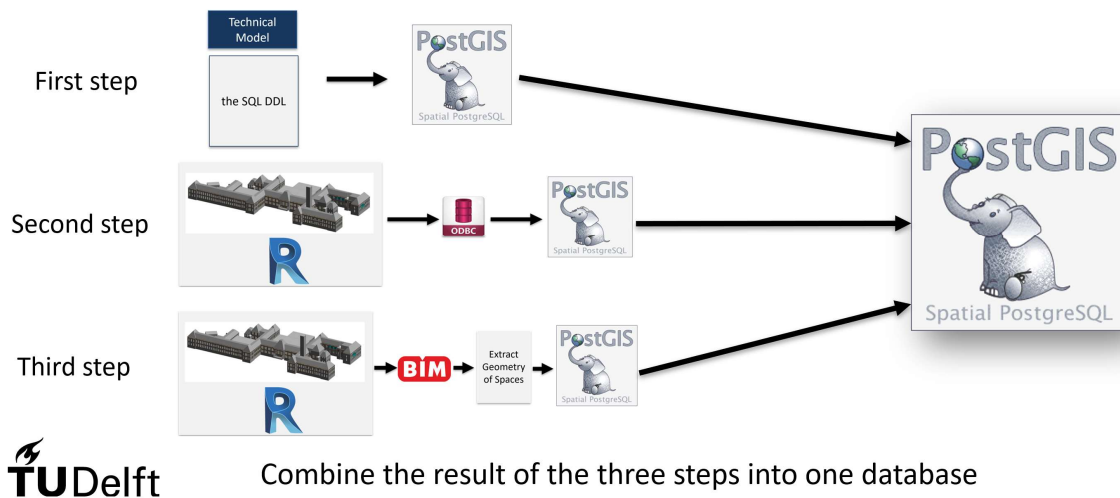


Figure 1: shows the three source types for the database tables

The tables that have been created from Revit consists of semantic information without any geometry such as WindowTypes, Walls, Levels, etc... as shown in Figure 2.

The screenshot shows the pgAdmin 4 interface with the 'WindowTypes' table selected in the browser. The SQL editor displays the query: `SELECT * FROM public."WindowTypes" ORDER BY "Id" ASC`. The table data is as follows:

ThermalResistanceR double precision	HeatTransferCoefficientU double precision	OmniClassNumber character varying	AssemblyCode character varying	FamilyName character varying	TypeName character varying	TypeMark character varying	RoughWidth double precision	RoughHeight double precision
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0406 x 0610mm	10	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0406 x 1220mm	11	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0610 x 0610mm	12	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0610 x 1220mm	13	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0915 x 0610mm	14	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0915 x 1220mm	15	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0406 x 1830mm	16	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0610 x 1830mm	17	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0915 x 1830mm	18	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0406 x 1220mm 2	19	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0610 x 1220mm RES	20	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0610 x 1220mm RES...	21	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0610 x 1220mm RES...	22	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	0710 x 1900 front_s...	23	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	1600 x 4100mm RES...	24	[null]	[null]
0.271105568508377	3.6886	23.30.20.17.11	[null]	M_Fixed	1600 x 1500mm.sma...	25	[null]	[null]

Figure 2: shows the WindowTypes table

The ifcspace table includes the 3D geometry of spaces and it has been created by extracting the geometry from IFC model by using local tool as shown in Figure 3.

The screenshot shows the pgAdmin 4 interface with the 'ifcspace' table selected in the browser. The SQL editor displays the query: `SELECT * FROM public.ifcspace`. The table data is as follows:

ifcguid character varying	geometry_wkt geometry
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...
1EcckFUEX4Ghhefn...	01060000A0E61...

Figure 2: shows the ifcspace table

The third type of tables are related to the integration model. The la\_party table include information about the users of the building such as name, role, pID, and type. The **generalspace** table contain information about the function of the spaces, usage, class, and we have included the generalspaceid and geometry of the space to this table to simplify the SQL queries as shown in Figure 4.

function	usage	class	generalspaceid (PK)	geom
A4 Study room / area	student	A Education	1195344	01060000A...
H2 Hall	student, staff, visitor	H Horizontal traffic	1195345	01060000A...
A4 Study room / area	student	A Education	1195346	01060000A...
H1 Times	student, staff, visitor	H Horizontal traffic	1195349	01060000A...
H1 Times	student, staff, visitor	H Horizontal traffic	1195350	01060000A...
A4 Study room / area	student	A Education	1195351	01060000A...
H1 Times	student, staff, visitor	H Horizontal traffic	1195352	01060000A...
V1 Stairs	student, staff, visitor	V Vertical traffic	1195353	01060000A...
V2 Lift	student, staff, visitor	V Vertical traffic	1195354	01060000A...
V1 Stairs	student, staff, visitor	V Vertical traffic	1195355	01060000A...
V1 Stairs	student, staff, visitor	V Vertical traffic	1195356	01060000A...
A2-4 drawing room	student	A Education	1195357	01060000A...
A4 Study room / area	student	A Education	1195358	01060000A...
A4 Study room / area	student	A Education	1195359	01060000A...
S1 Toilet room	student, staff, visitor	S Sanitary	1195360	01060000A...
S1 Toilet room	student, staff, visitor	S Sanitary	1195361	01060000A...

Figure 2: shows the generalspace table

## Using OpenCities Map

- PostGIS, data is loaded with:  
**"C:\Program Files\PostgreSQL\10\bin\psql.exe" -h localhost -U postgres -d postgres -p 5432 -f "D:\Data\Bentley Map\PostGIS\Delft\restore.sql"**  
 ()
- The geometry is **not** georeferenced.
- Editing and posting of modified and new objects to the database is also possible, when a numeric primary key is available.
- DB Connection: **postgres / admin** and **localhost postgres**:

**Map Imports/Exports**

Imports: Import 1

- postgres@postgres localhost
- postgres public: "Rooms"
- postgres public: "SpaceGeometry"
- postgres public: boundary
- postgres public: generalspace
- postgres public: ifcspace
- postgres public: la\_spatialunit
- postgres public: space

**PostGIS Connect**

User: postgres  
 Password:   
☐ Use Windows Integrated Authentication  
 Host: localhost  
 Database: postgres

**New Connection Parameters**

Selected Features:

- PostGIS
- Current Session
- postgres public: "Rooms"
- postgres public: "SpaceGeometry"
- postgres public: boundary
- postgres public: generalspace
- postgres public: ifcspace
- postgres public: la\_spatialunit
- postgres public: space

**General**

User: postgres  
 Host: localhost  
 Database: postgres

**Connection Parameters**

Pessimistic Locking: True  
 Target Model: Untitled Design

**Feature Parameters**

Data Streaming: False

**Analyze Result**

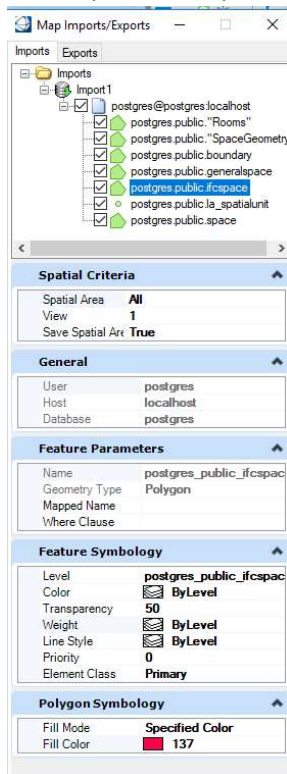
postges public: "Rooms"

Property	Value
"Id"	1195563
"Area"	195.98170365505675
"BaseFinish"	
"Building"	08 Architecture and the Built Environment (A+BE)
"Capacity"	0
"CeilingFinish"	
"Comments"	
"Department"	
"DesignOption"	
"Floor"	Gietvloer
"FloorFinish"	
"IfcGUID"	1EcnkFUEX4Ghnefn8c6750
"Level"	322664
"Name"	02 Med.100
"Number"	218
"Occupancy"	
"Occupant"	
"Perimeter"	65.4631573910076
"PhaseId"	86961
"RoomNumber"	08.01.02.100
"SpaceName"	Kraaiennest
"Type"	H2 Hall
"UsageFunction"	H Horizontal traffic
"Volume"	470.3560887721361
"WallFinish"	
unit	

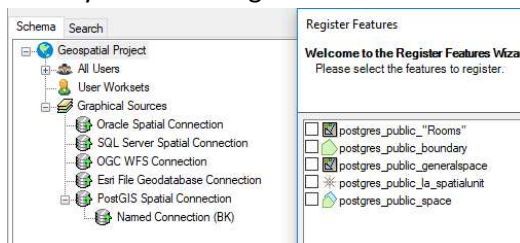
**postges public: generalspace**

Property	Value
generalspaceid	1195563
"function"	H2 Hall
class	H Horizontal traffic
usage	student, staff, visitor

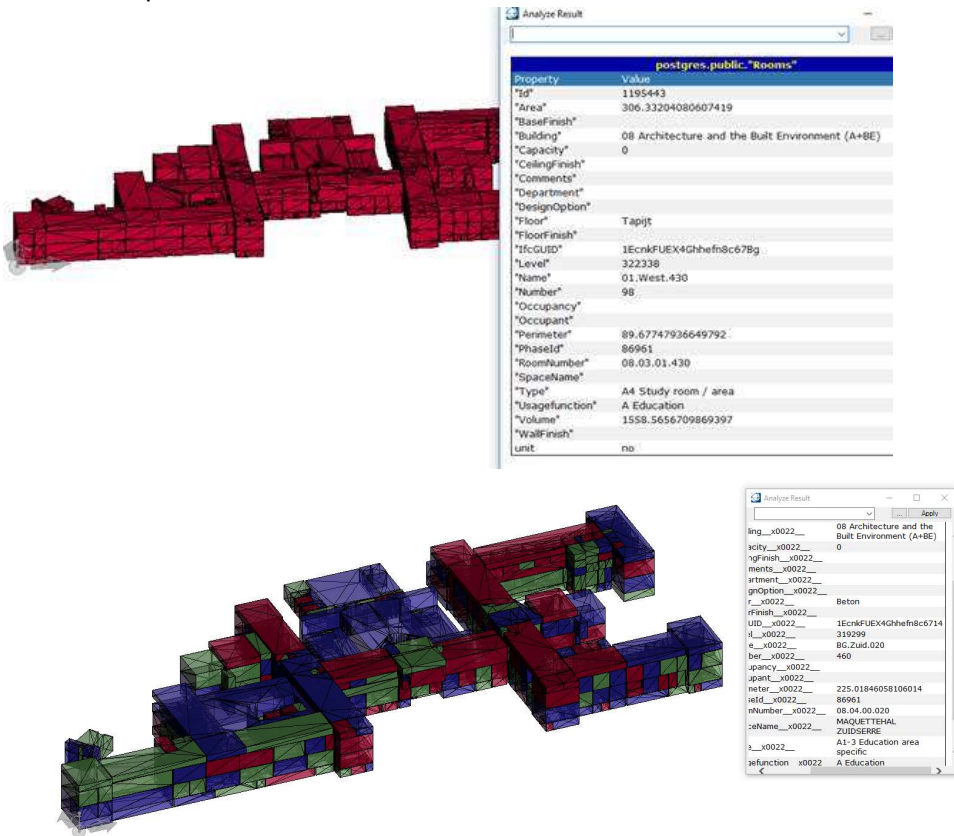
- **Generalspace:** Query Complete (673 features in 2.882 seconds). But what table?
- **Rooms:** Query Complete (673 features in 2.839 seconds).
- **ifcspace** , SpaceGeometry, space:  
Querying postgres.public."SpaceGeometry" for update is not supported. This is usually because of a missing primary key specification.
- Note the table space has 2 geometry columns, maybe make a view with a single geometry column.
- Naming of table and column names is not very consequent, initcase, quotes etc.
- Editing and posting of modified and new objects to the database is also possible, when a numeric primary key is available.
- The table *space* has 2 geometry columns. Probably views should be created with a single geometry column to make a successful import.
- Import using IMPX file D:\Data\Bentley Map\PostGIS\Delft\BK\_Import.impx.  
Four layers are imported into a model in the DGN file BK Delft.dgn.



- Symbology of layer *SpaceGeometry* can be based on the property Class.
- Optionally an XFM schema can be created using the Geospatial Administrator.  
Five layers can be registered.



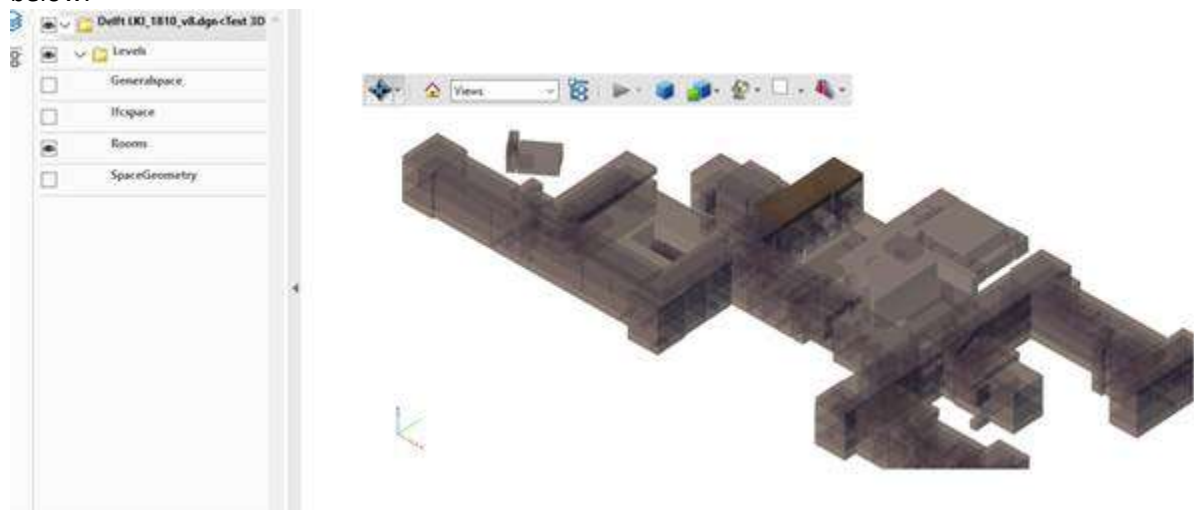
- Result of import tables.



or

## Publishing

- The data can be printed to 3D PDF, see the attached PDF file with 4 levels, and the screenshot below.



- Viewing the model on a mobile device is possible by creating an iModel (and placing the model on a cloud server) and use a Bentley app (available for Android and Apple).
- The model can also be exported and published via [Cesium](#).

