

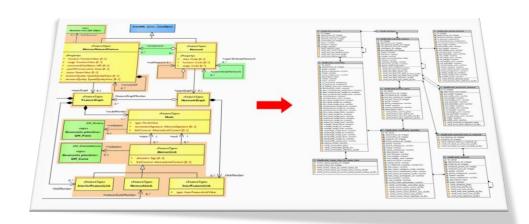
# Some updates on the 3DCityDB extension for ADEs

Giorgio Agugiaro

CityGML Joint Workshop Energy + Utility Network ADE 7 December 2017, Karlsruhe

#### giorgio.agugiaro@ait.ac.at

Smart and Resilient Cities Unit Center for Energy AIT - Austrian Institute of Technology Vienna, Austria





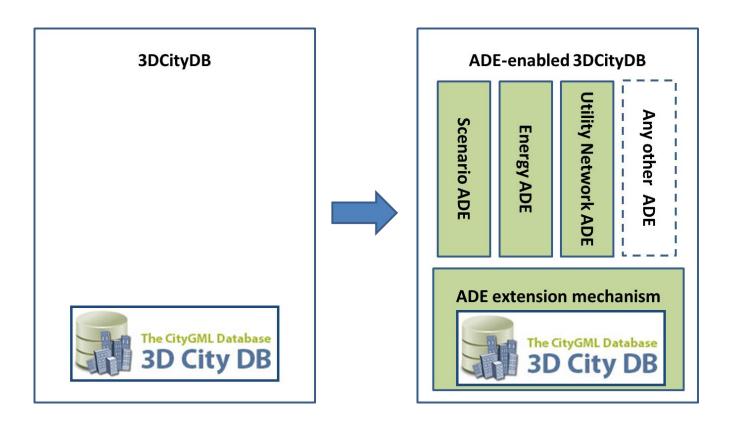
## Motivation & goals

- Growing demand for an ADE-aware DB solution
  - Work in progress by the 3DCityDB development team
    - Automatic mapping from OO to ER model
    - Automatic generation of DB schema
    - Extension of the Importer/Exporter
    - ...
    - All these fantastic goodies for any ADE!
    - BUT: it will take time till it is ready → See presentation by C. Nagel
- So far, DB implementations for the Energy/Utility Network ADE:
  - partial AND/OR
  - non-open AND/OR
  - poorly or not documented at all
- Some initial results (for the Energy ADE) presented last May in Grenoble
  - Please refer to those slides for more details:
     <a href="http://en.wiki.energy.sig3d.org/images/upload/20170523">http://en.wiki.energy.sig3d.org/images/upload/20170523</a> Agugiaro Energy ADE Workshop 7\_3DCityDB.pdf



#### Motivation & goals

- Gather and share experience on how to extend to 3DCityDB for any ADE
  - For further tools (citygml4j, Importer/Exporter, etc.) → See next presentation!
- Foster adoption and further development of the Energy & Utility Network ADEs





#### Motivation & goals

- Gather and share experience on how to extend to 3DCityDB for any ADE
  - For further tools (citygml4j, Importer/Exporter, etc.)
- Foster adoption and further development of the Energy & Utility Network ADEs
- First test case: implementation of the Energy ADE (for PostgreSQL)
  - (Manual) mapping from OO to ER
  - Complete implementation of v. 0.8, but 99% compatible with v.0.9.
  - Particular care of documentation
  - Released in July 2017 under to Apache 2.0 license on GitHub
    - https://github.com/gioagu/3dcitydb\_ade
- Follow up: test methodology also on
  - Utility Network ADE, released September 2017, updated yesterday
  - Scenario ADE (work in progress)
  - Same criteria of the Energy ADE: Apache 2.0 license and GitHub



3DCityDB extension for the Energy ADE

#### 3D City Database extension for the CityGML Energy ADE 0.8 PostgreSQL Version

#### Documentation

Last update: 16 September 2017





#### **Table of Contents**

1	INT	RODUCTION	8
	1.1	System and design decisions	9
2	DAT	TA MODELLING	10
	2.1	Time series and Schedule module	10
	2.2	Material and Construction module	12
	2.3	Building Physics module	
	2.4	Occupancy module	
	2.5		
3		TABASE DESIGN	
3			
	3.1	ADE support in 3DCityDB	
		3.1.1 Metadata module	
		3.1.2 Mapping rules for ADE classes	
		3.1.2.1 Mapping of ADE nonCityObject classes	
		3.1.2.2 Mapping of ADE _CityObject classes	
		3.1.2.3 Mapping of ADE-extended classes	
		3.1.2.4 Mapping of relations between ADE and "vanilla" classes	
		3.1.3 Database stored procedures	
		3.1.3.1 Delete stored procedures for "stand alone" ADE classes	27
		3.1.3.2 Delete stored procedures for ADE-extended classes	
		3.1.3.3 Get_envelope stored procedures	
		3.1.3.4 ADE-hook mechanism for "vanilla" stored procedures	
		3.1.4 Comments	28
	3.2	Database schema of the Energy ADE	29
		3.2.1 Time series and schedule module	29
		3.2.2 Material and Construction module	31
		3.2.3 Building Physics module	33
		3.2.4 Occupancy module	37
		3.2.5 Energy Systems module	39
		3.2.6 Lookup tables	44
		3.2.7 Sequences	45
		3.2.8 Stored procedures	
		3.2.9 Views	49
4	TES	ST DATA	54
5	INS	TALLATION	56
	5.1	System requirements	56

3DCityDB extension for the Energy ADE

	5.2	Automatic full installation	56		
	5.3	Manual installation	58		
		5.3.1 Installing the Metadata module	58		
		5.3.2 Installing 3DCityDB extension for the Energy ADE	64		
	5.4	Installation of the test data	65		
APF	PEND	IX A: THE 3DCITYDB UTILITIES PACKAGE	67		
	A.I	Content	67		
		A.1.1 Insert stored procedures in schema citydb_pkg	67		
		A.1.2 Views	69		
		A.1.3 Additional stored procedures in schema citydb_view	70		
		A.1.4 Trigger functions			
	A.2	Installation	71		
		A.2.1 System requirements	71		
		A.2.2 Automatic installation	71		
		A.2.3 (Alternative) manual installation	73		
APF	APPENDIX B: ENERGY ADE 0.8 UML DIAGRAMS				
NO	OTES				



## Design criteria (excerpt)

- Build upon the existing objects of the 3DCityDB... but <u>keep the original ones</u> untouched (for the sake of the Importer/Exporter)
- Define a non-concurrent way of extending the 3DCityDB with any ADEs (e.g. Energy ADE + Utility Network ADE)
- Stay close to the original "style" of the 3DCityDB when it comes to tables, constraints, naming conventions, data types, etc.
- Possibly keep the number of new tables in check
- Implementation for PostgreSQL, but avoid potential technology lock-ins for future conversions to other DBs (as far as possible)

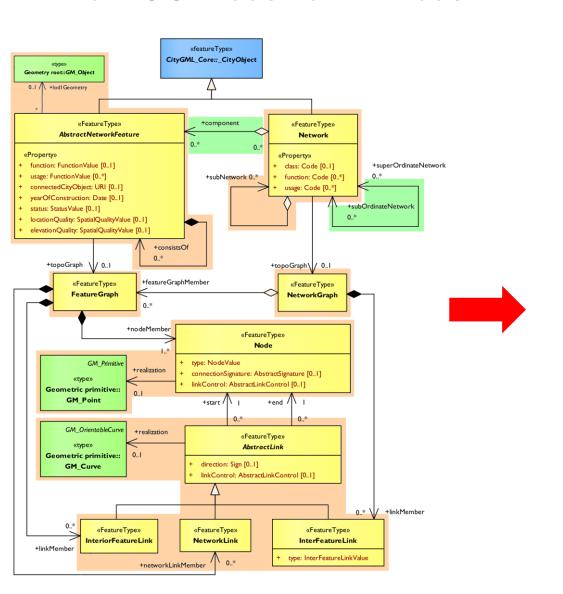


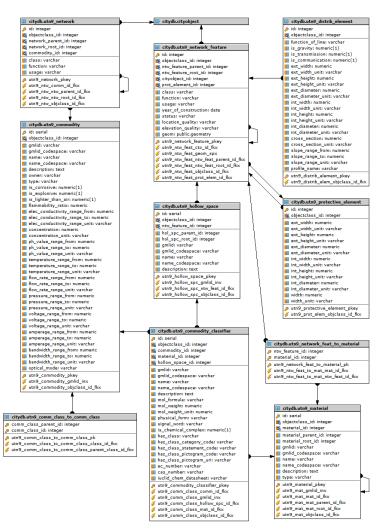
#### Implementation steps

- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" any ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - Make some existing stored procedures ADE-aware. E.g.:
    - delete\_building() → must work also with ADE-AbstractBuilding
    - delete\_cityobject() → must work also with new CityObjects
    - delete\_cityobjectgroup() → must work also with new CityObjects
    - get\_envelope\_cityobject() → same as above
  - Enable/extend existing tools to be ADE-compatible: citygml4j,
     Importer/Exporter, etc. → See presentation by C. Nagel
- All rules are agreed upon within the 3DCityDB development team and are being further tested and implemented for the next 3DCityDB release



#### From OO-Model to ER-Model



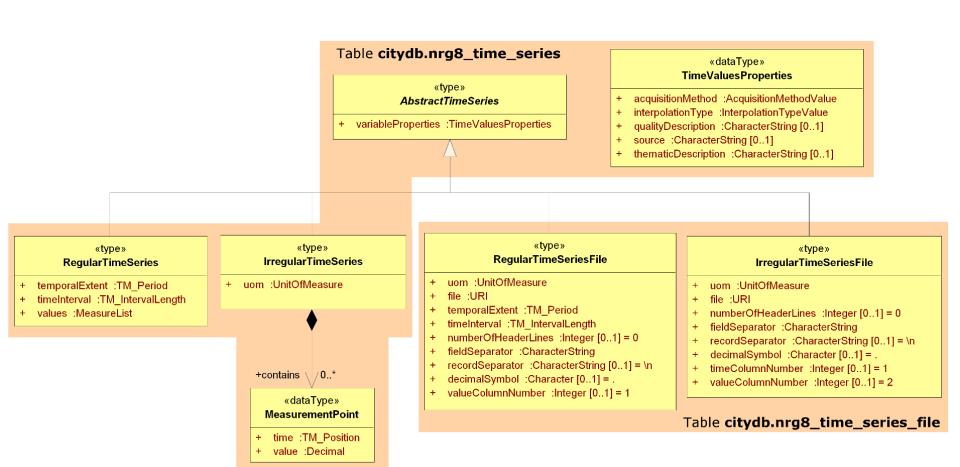




- "Pure" CityGML data can be already importer/exported using the 3DCityDB importer/exporter
- For ADE data, no "out-of-the-box" tools (yet)
- Data import into the 3DCityDB (sometimes) difficult, due to the very rich and complex database structure
- A couple of examples:
  - A TimeSeries object ("plain" and file-based) from the Energy ADE
  - A building with additional Energy ADE attributes

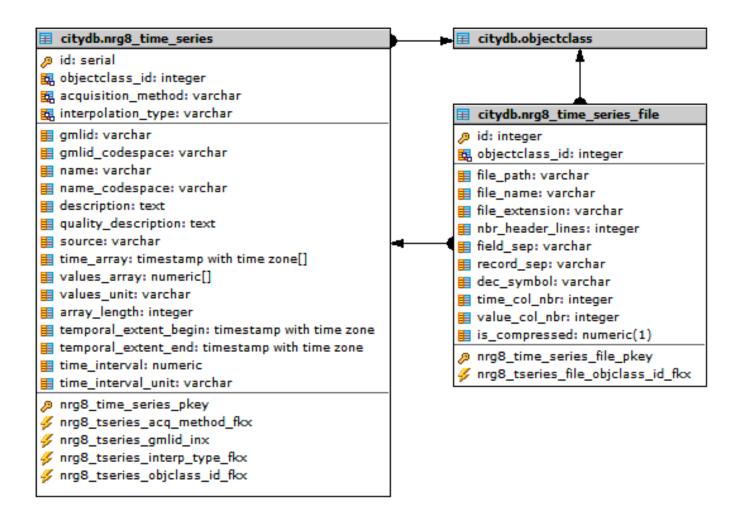


## Time series: UML Diagram & mapping





#### Time series: ER model



## (Ir)regular time series



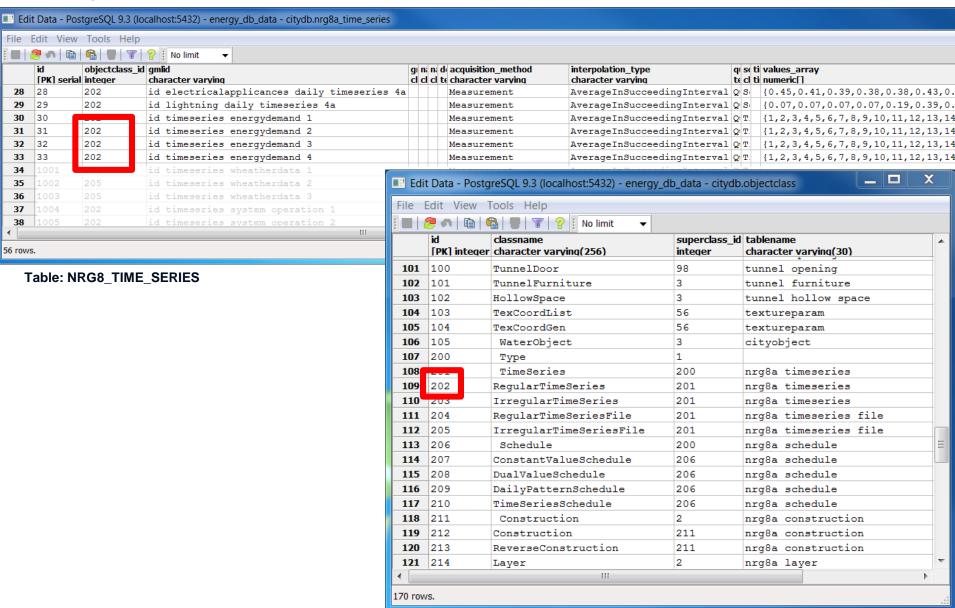


Table: OBJECTCLASS

## File-based (ir)regular time series



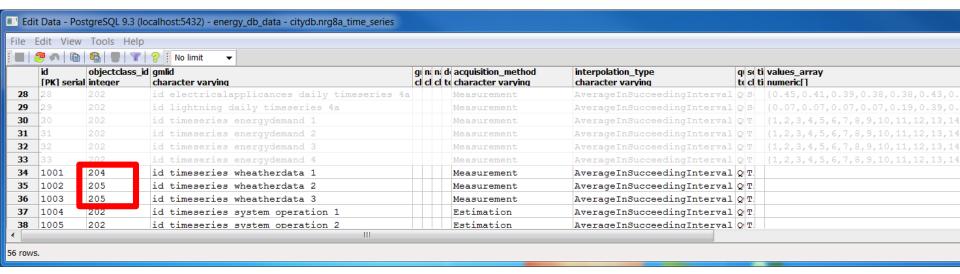


Table: NRG8\_TIME\_SERIES

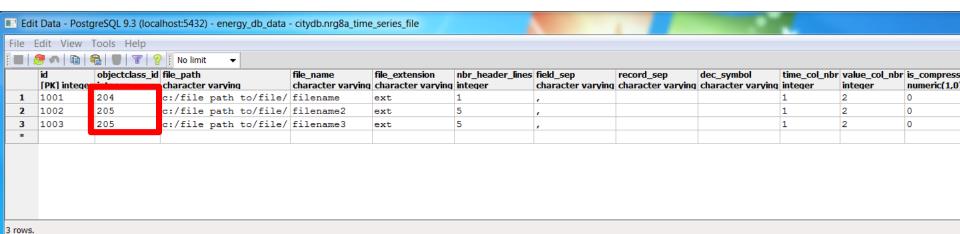


Table: NRG8\_TIME\_SERIES\_FILE

### File-based (ir)regular time series



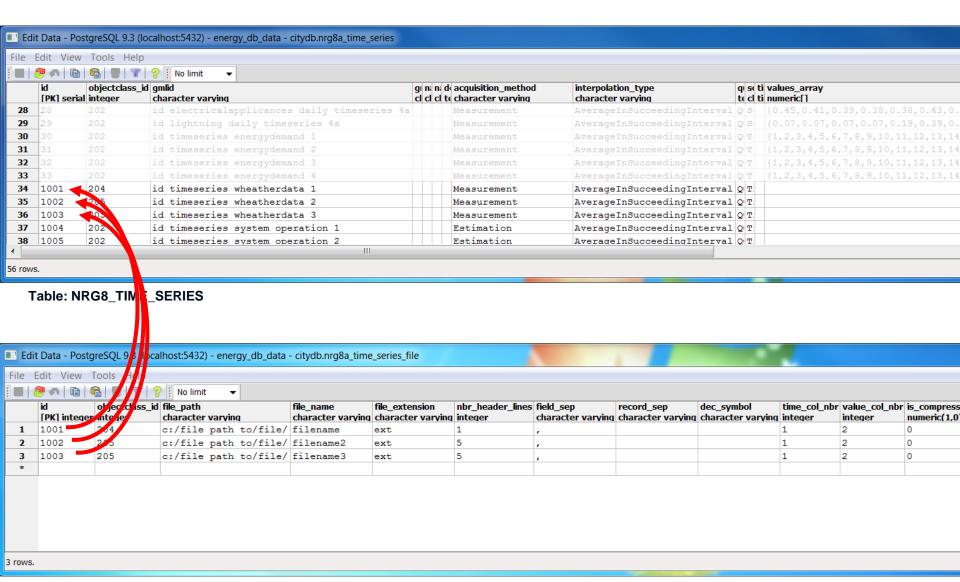
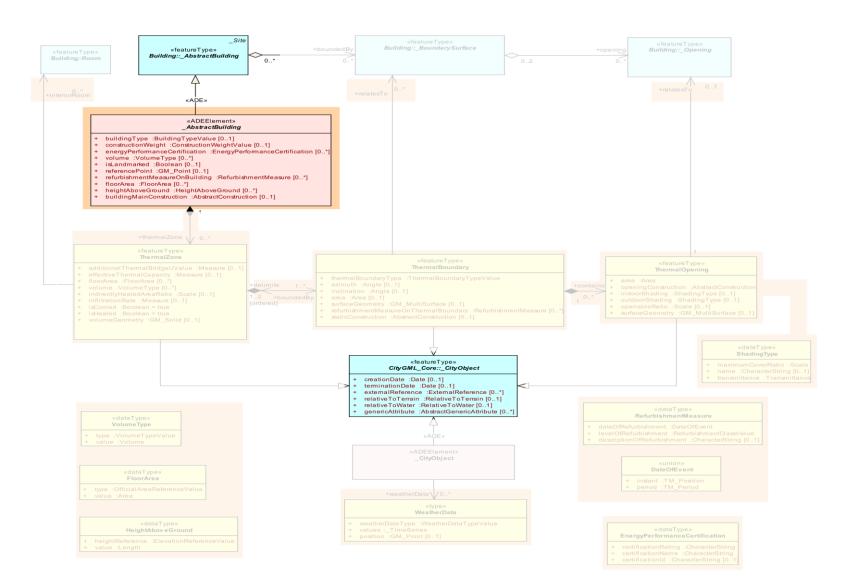


Table: NRG8\_TIME\_SERIES\_FILE

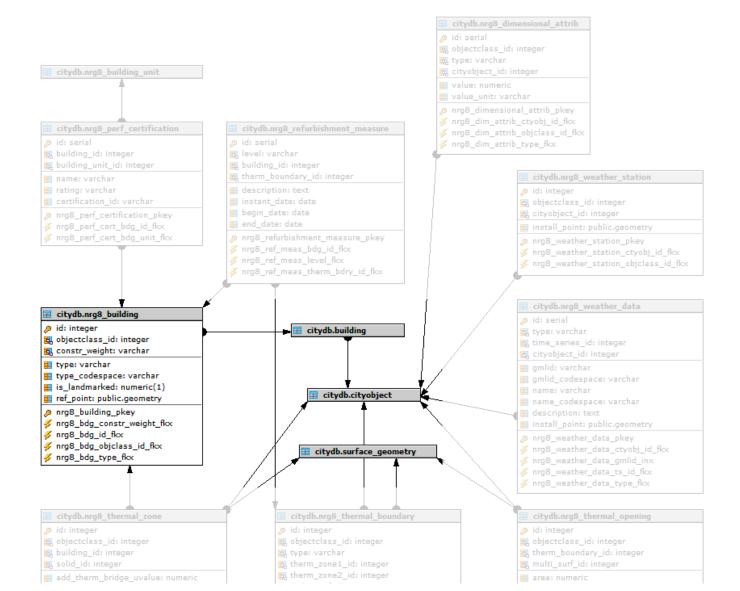


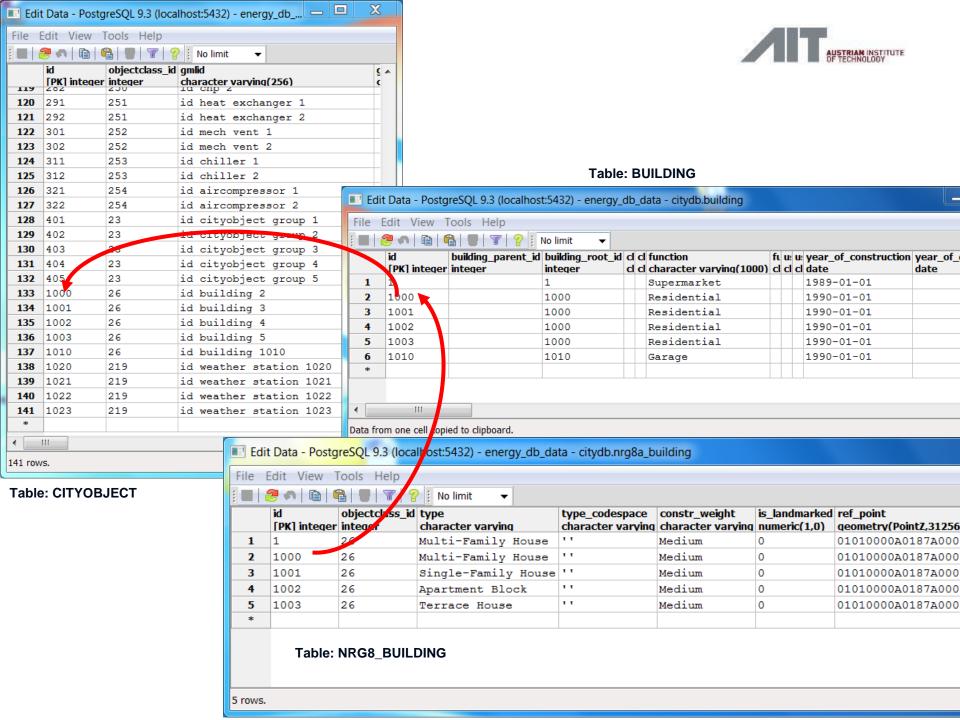
## AbstractBuilding: UML Diagram & mapping





#### AbstractBuilding: ER model







- How to delete data from the database?
  - Use the delete stored procedures (refer to the documentation)
- How to insert ADE-data into the database?
  - 1. Write your own SQL code to access the tables directly
  - 2. Use the *insert* stored procedures
  - 3. Use the "smart" *insert* stored procedures
  - 4. Use the updatable views

**Nota bene**: The following examples are also available on GitHub <a href="https://github.com/gioagu/3dcitydb\_ade/blob/master/02\_energy\_ade/test\_data/Energy\_ADE\_Insert\_data\_aexample\_scripts.sql">https://github.com/gioagu/3dcitydb\_ade/blob/master/02\_energy\_ade/test\_data/Energy\_ADE\_Insert\_data\_aexample\_scripts.sql</a>



1. Write your own SQL code to access the tables directly

LEFT SIDE:

**RIGHT SIDE:** 

Example with a **RegularTimeSeries** object

Example with a **RegularTimeSeriesFile** object



1. Write your own SQL code to access the tables directly

```
INSERT INTO citydb.nrg8_time_series
(id, objectclass_id, name, acquisition_method, interpolation_type, values_array, values_unit, temporal_extent_begin, temporal_extent_end, time_interval, time_interval_unit)

VALUES
(10001, 202, 'Test_time_series_insert_1a', 'Estimation', 'AverageInSucceedingInterval', '{1,2,3,4,5,6,7,8,9,10,11,12}', 'kWh/m^2/month', '2015-01-01 00:00', '2015-12-31 23:59', 1, 'month')

RETURNING id;
```

```
WITH s AS (
 INSERT INTO citydb.nrg8 time series
 (id, objectclass id, name, acquisition method, interpolation type,
values_unit, temporal_extent_begin, temporal_extent_end,
time_interval, time_interval_unit)
 VALUES
 (10011, 204, 'Test time series file insert 1', 'Estimation',
'AverageInSucceedingInterval', 'kWh/m^2/month', '2015-01-01 00:00', '2015-
12-31 23:59', 1, 'month')
 RETURNING id, objectclass_id
INSERT INTO citydb.nrg8_time_series_file
(id, objectclass id, file path, file name, file extension,
nbr header lines, field sep, record sep, dec symbol, value col nbr,
is compressed)
 SELECT s.id, s.objectclass id, 'file path XXXXX', 'file name XXXXX',
'file_ext_XXXXX', 1, ',', '/n', '.', 1, 0
 FROM s
RETURNING id:
```

Notes:

The id, objectclass\_id MUST be set by the user



2. Use the *insert* stored procedures (in citydb\_pkg schema)

```
SELECT citydb_pkg.nrg8_insert_time_series(
  objectclass_id := 202,
  name := 'Test_time_series_insert_2a',
  acquisition_method := 'Estimation',
  interpolation_type := 'AverageInSucceedingInterval',
  values_array := '{1,2,3,4,5,6,7,8,9,10,11,12}',
  values_unit := 'kWh/m^2/month',
  temporal_extent_begin := '2015-01-01 00:00',
  temporal_extent_end := '2015-12-31 23:59',
  time_interval := 1,
  time_interval_unit := 'month');
```

#### Notes:

The **objectclass\_id** MUST be set by the user The **id** and **gmlid**, if null, are set automatically The **id** value is returned by the stored procedure

```
WITH s AS (
 SELECT citydb_pkg.nrg8_insert_time_series(
  objectclass id := 204,
  name := 'Test_time_series_file_insert_2a',
  acquisition method := 'Estimation',
  interpolation_type := 'AverageInSucceedingInterval',
  values unit := 'kWh/m^2/month',
  temporal_extent_begin := '2015-01-01 00:00',
  temporal extent end := '2015-12-31 23:59',
  time interval := 1,
  time_interval_unit := 'month') AS ts_id
SELECT citydb pkg.nrg8 insert time series file(
 id := s.ts id.
 objectclass id := 204,
 file_path := 'file_path_XXXXX',
 file name := 'file name XXXXX',
 file extension := 'file ext XXXXX',
 nbr header lines := 1,
 field_sep := '.'.
 record_sep := '/n',
 dec_symbol := '.',
 value col nbr := 1,
 is compressed := 0)
FROM s;
```



3. Use the "smart" *insert* stored procedures (in citydb\_view schema)

```
SELECT citydb_view.nrg8_insert_regular_time_series(
name := 'Test_time_series_insert_4a',
acquisition_method := 'Estimation',
interpolation_type := 'AverageInSucceedingInterval',
values_array := '{1,2,3,4,5,6,7,8,9,10,11,12}',
values_unit := 'kWh/m^2/month',
temporal_extent_begin := '2015-01-01 00:00',
temporal_extent_end := '2015-12-31 23:59',
time_interval := 1,
time_interval_unit := 'month');
```

```
SELECT citydb view.nrg8 insert regular time series file(
name := 'Test time series file insert 4a',
acquisition method := 'Estimation',
interpolation type := 'AverageInSucceedingInterval',
values unit := 'kWh/m^2/month',
temporal_extent_begin := '2015-01-01 00:00',
temporal_extent_end := '2015-12-31 23:59',
time interval := 1,
time interval unit := 'month',
file path := 'file path XXXXX',
file name := 'file name XXXXX',
file extension := 'file ext XXXXX',
nbr header lines := 1,
field sep := ',',
record sep := '/n',
dec_symbol := '.',
value col nbr := 1,
is compressed := 0):
```

#### Notes:

The **id** and the **gmlid**, if null, are set automatically The **objectclass\_id** is set automatically The **id** value is returned by the stored procedure



4. Use the updatable views (in citydb\_view schema)

#### INSERT INTO citydb view.nrg8 time series regular

(name, acquisition\_method, interpolation\_type, values\_array, values\_unit, temporal\_extent\_begin, temporal\_extent\_end, time\_interval, time\_interval\_unit)

#### **VALUES**

('Test\_time\_series\_insert\_3c', 'Estimation', 'AverageInSucceedingInterval', '{1,2,3,4,5,6,7,8,9,10,11,12}', 'kWh/m^2/month', '2015-01-01 00:00', '2015-12-31 23:59', 1, 'month')

**RETURNING** id;

INSERT INTO citydb\_view.nrg8\_time\_series\_regular\_file

(name, acquisition\_method, interpolation\_type, values\_unit, temporal\_extent\_begin, temporal\_extent\_end, time\_interval, time\_interval\_unit,

file\_path, file\_name, file\_extension, nbr\_header\_lines, field\_sep, record\_sep, dec\_symbol, value\_col\_nbr, is\_compressed)

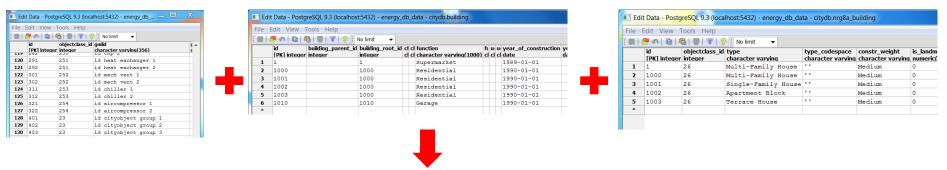
#### **VALUES**

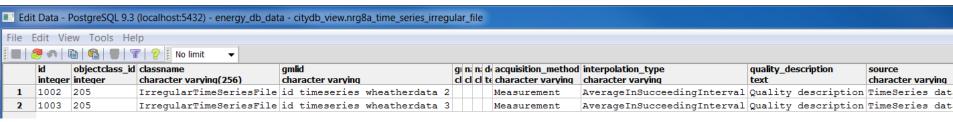
('Test\_time\_series\_insert\_file\_3c', 'Estimation', 'AverageInSucceedingInterval', 'kWh/m^2/month', '2015-01-01 00:00', '2015-12-31 23:59', 1, 'month',

'file\_path\_XXXXX', 'file\_name\_XXXXX', 'file\_ext\_XXXXX', 1, ',', '/n', '.', 1, 0) **RETURNING** id:



- 4. Use the updatable views (in citydb\_view schema)
- Views hide the complexity of data stored in multiple tables by defining a virtual joined table which can be accessed like any other "standard" table.
- As the 3DCityDB views are built upon the "smart" insert stored procedures, the same benefits still apply.
- In addition: UPDATE and DELETE operations are allowed, too.







 All methods shown so far can be embedded in functions written in any programming language (Python, Java, etc.)

OR

By using an ETL tool (like FME by Safe Software)

OR

A combination thereof

There are also additional views and stored procedures to ease management of objects connected to time series.

For more details and more examples, please refer to the documentation or the resources on GitHub!!



#### Conclusions 1/2

- Current implementation extends 3DCityDB for
  - Energy ADE
  - Utility Network ADE
  - Scenario ADE (soon)
- Included are some additional features to ease data input/editing with a quite high degree of granularity
  - Insert stored procedures
  - "Smart" insert stored procedures
  - Updatable views



#### Conclusions 2/2

- A final word/note of caution
  - Implementation did not focus on performance
  - There is room for further improvements
    - Focus on automatic code generation, performance, scalability, etc.
      - → see next presentation
  - But...
- As of now, first and only available free and open implementation
  - Already being tested/used by EIFER, HFT, TU Delft ...and IntegrCiTy consortium
  - Feedback, further testing, help are always welcome!



#### Upcoming conferences

- 1-5 October 2018: GeoDelft 2018 "triple" conference
  - ISPRS Comm IV Midterm Symposium <a href="http://www.isprs.org/tc4-symposium2018/">http://www.isprs.org/tc4-symposium2018/</a>
  - 3D GeoInfo 2018 <a href="https://www.utwente.nl/en/3dgeoinfo2018/">https://www.utwente.nl/en/3dgeoinfo2018/</a>
  - Smart Data and Smart Cities 2018 <a href="http://www.udms.net/">http://www.udms.net/</a>

#### Deadlines:

- Full paper submission (full paper double blind review) 31 March 2018
- Abstract submission (abstract blind review) 30 April 2018
- Notification of authors 15 May 2018
- Final full paper 15 June 2018



## AIT Austrian Institute of Technology

your ingenious partner

Dr. Giorgio Agugiaro
Smart and Resilient Cities Unit
Center for Energy
AIT - Austrian Institute of Technology
giorgio.agugiaro@ait.ac.at

