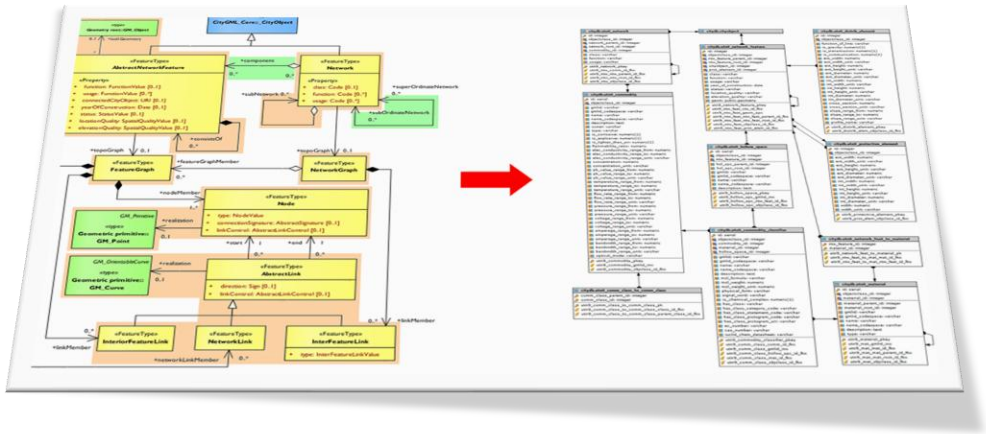
Some updates on the 3DCityDB extension for ADEs

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 next presentation

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 next presentation

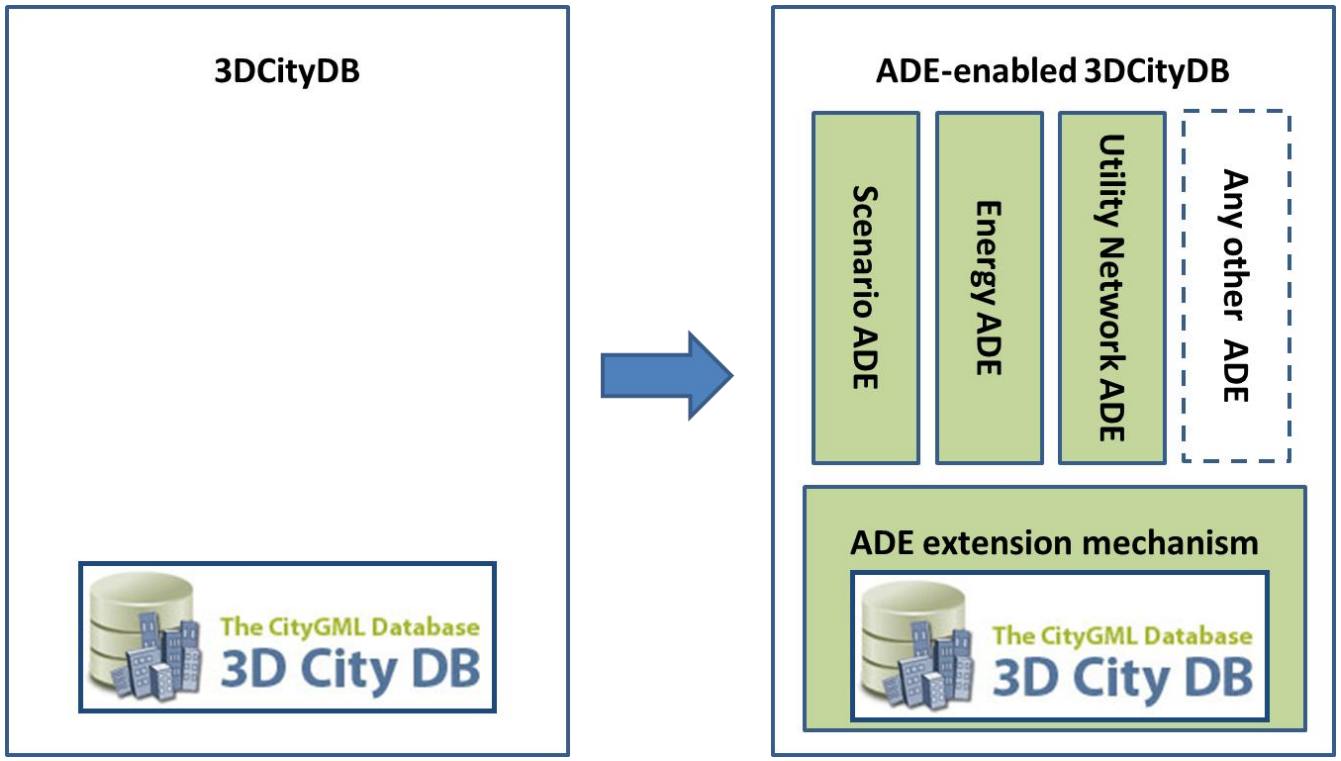
- So far, DB implementations for the Energy/Utility Network ADE:
  - partial AND/OR
  - non-open AND/OR
  - poorly or not documented at all

- Some initial results (for the Energy ADE) presented last May in Grenoble
  - Please refer to those slides for more details
    http://en.wiki.energy.sig3d.org/images/upload/20170523_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.) → See next presentation
- 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

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

Documentation

Last update: 16 September 2017

AUSTRIAN INSTITUTE OF TECHNOLOGY
TOMORROW TODAY

Table of Contents

Source: https://github.com/gioagu/3dcitydb_ade/blob/master/02_energy_ade/manual/3DCityDB_Energy_ADE_0.8_Documentation.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
- 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

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
- 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) → see previous presentation
  - Same criteria of the Energy ADE: Apache 2.0 license and GitHub

Design criteria (excerpt)

- Build upon the existing objects of the 3DCityDB… but keep the original ("vanilla") untouched (for the sake of the Importer/Exporter)
- Define a non-concurrent way of extending the 3DCityDB with *any* ADEs (e.g. Energy ADE + UtilityNetwork 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

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

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

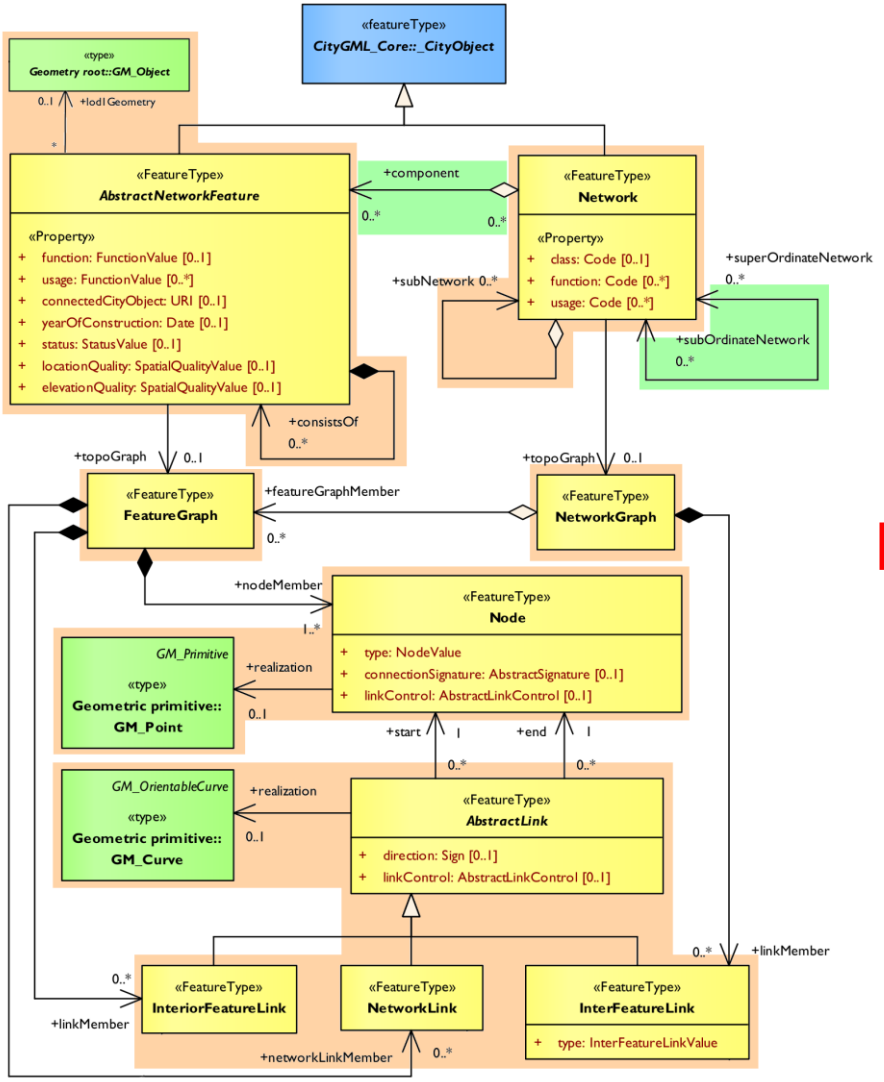
Implementation steps

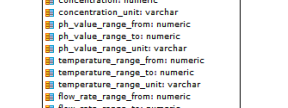
- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - 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 next presentation

Implementation steps

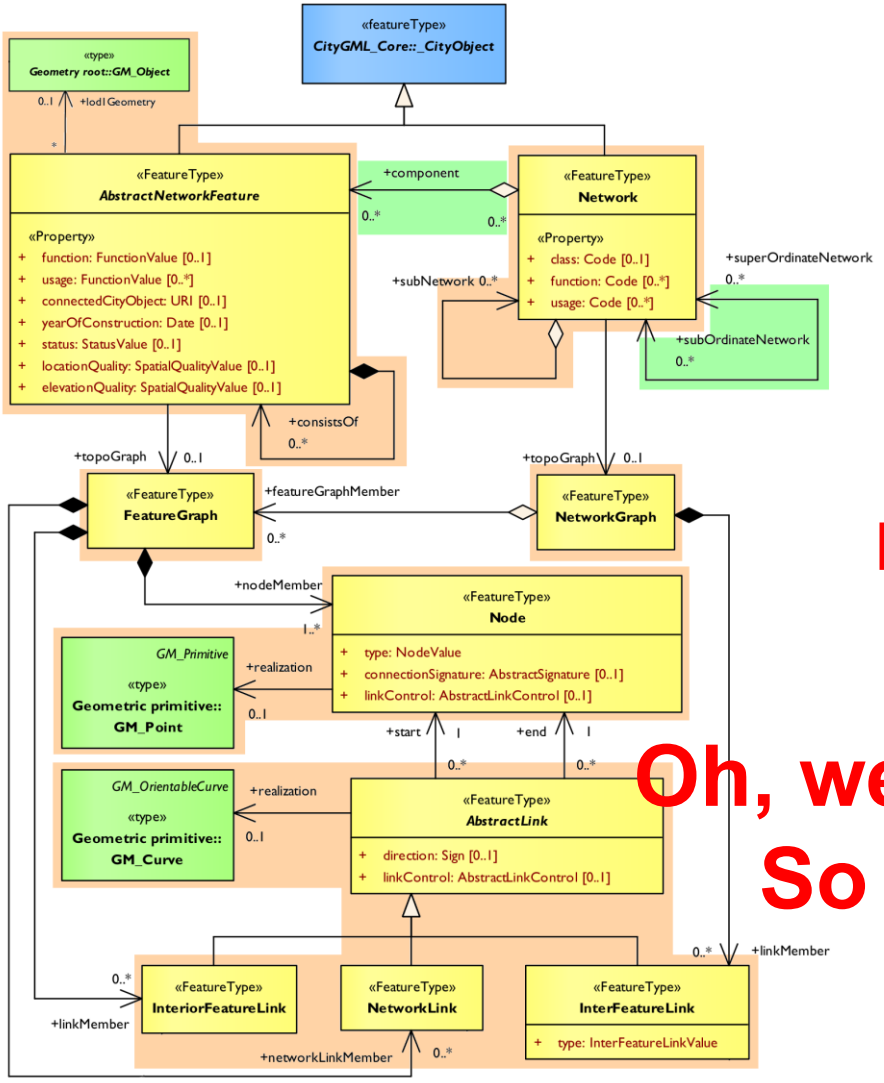
- Define and agree upon rules to make the 3DCityDB "ADE-compatible"
  - Enable to "register" *any* ADE
    - Add a metadata module
    - Add functions to help installing/removing an ADE
  - Define rules how to map ADE-classes to new/existing tables
    - Adopt naming convention for new DB entities
  - 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 next presentation
- All rules are agreed upon within the 3DCityDB development team and are being further tested and implemented for the next 3DCityDB release
- Further details → my presentation in Grenoble and in the next presentation

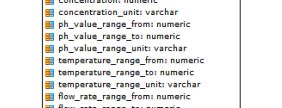
From OO-Model to ER-Model





From OO-Model to ER-Model





Oh, well, nice...

So what?

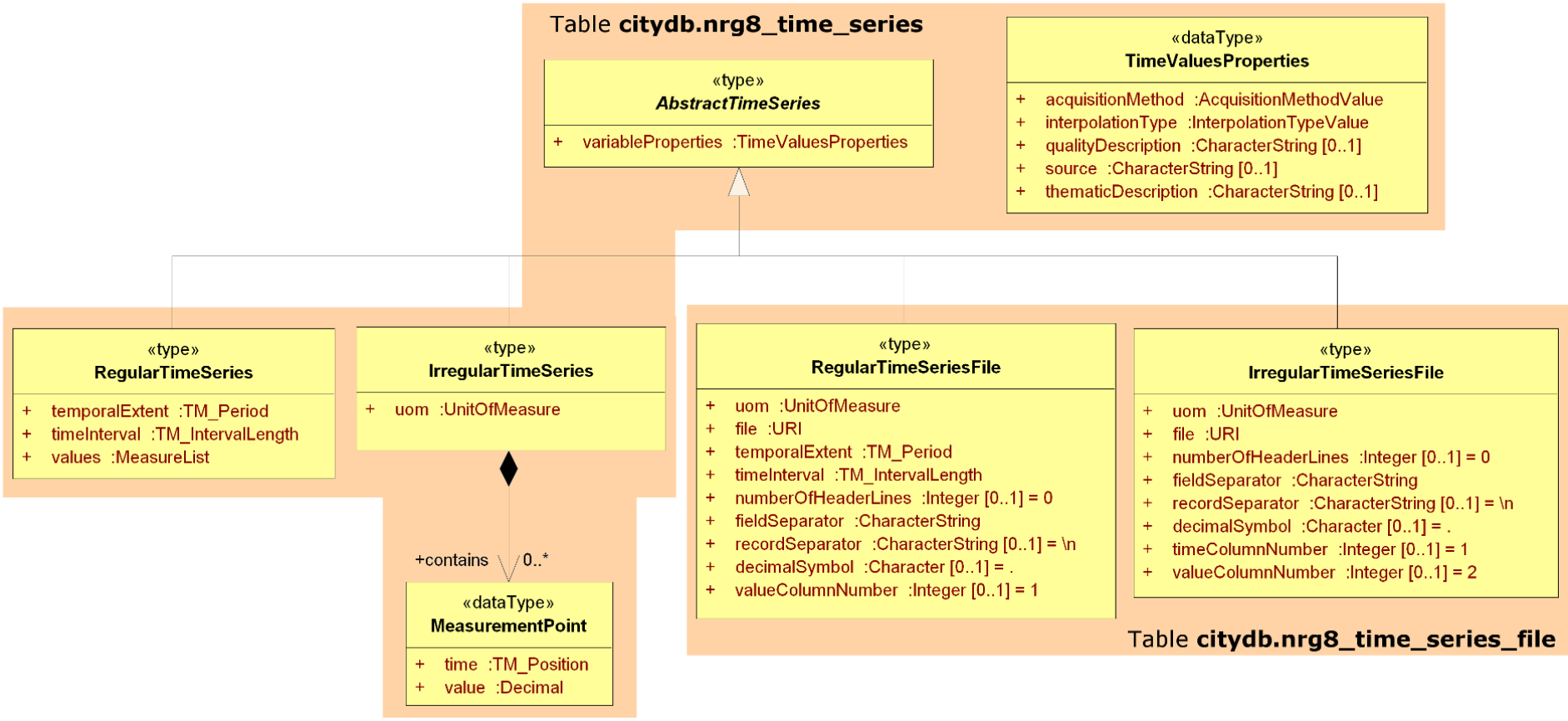
Interacting with the (extended) 3DCityDB

- "Vanilla" 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

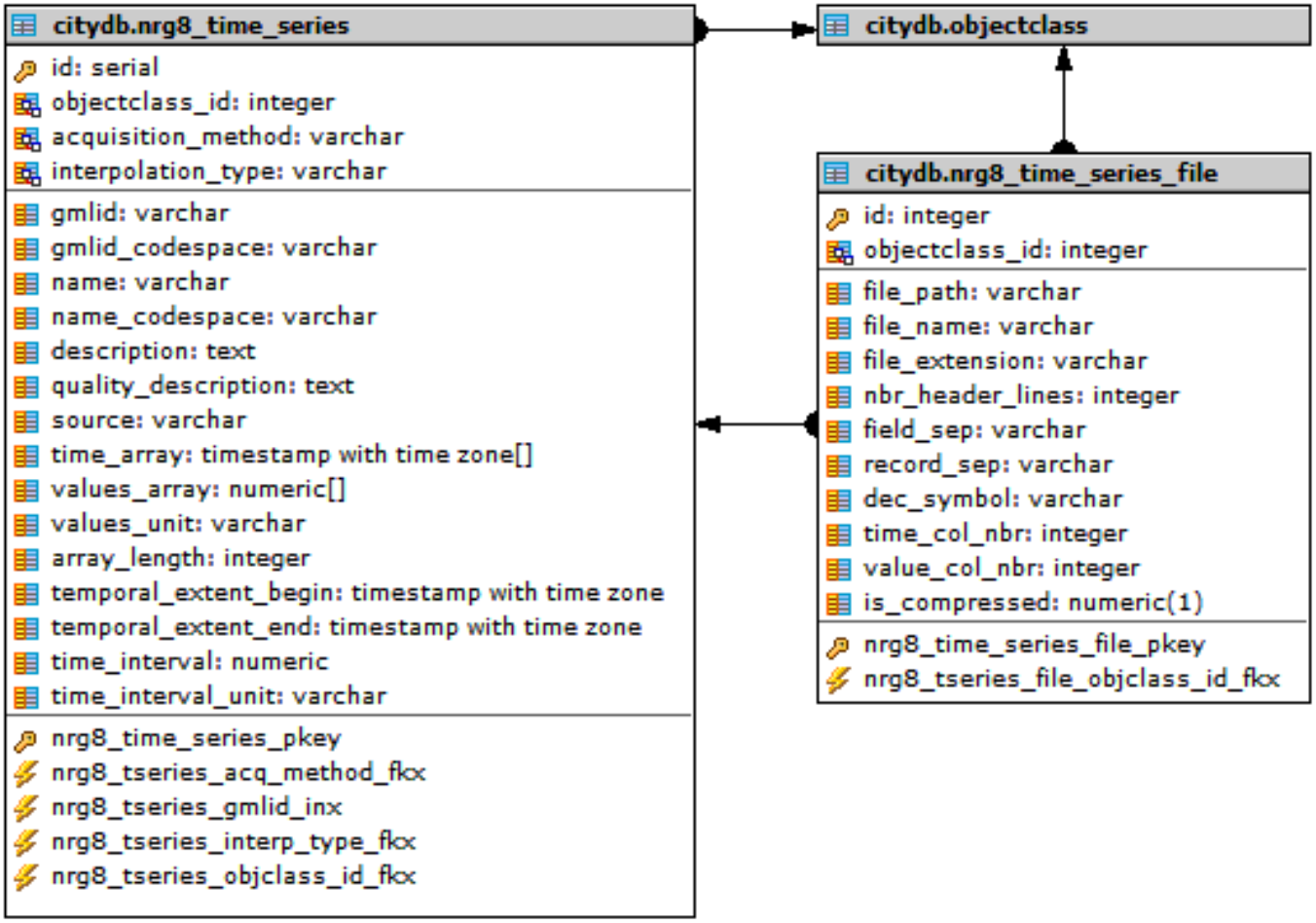
Interacting with the (extended) 3DCityDB

- "Vanilla" 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

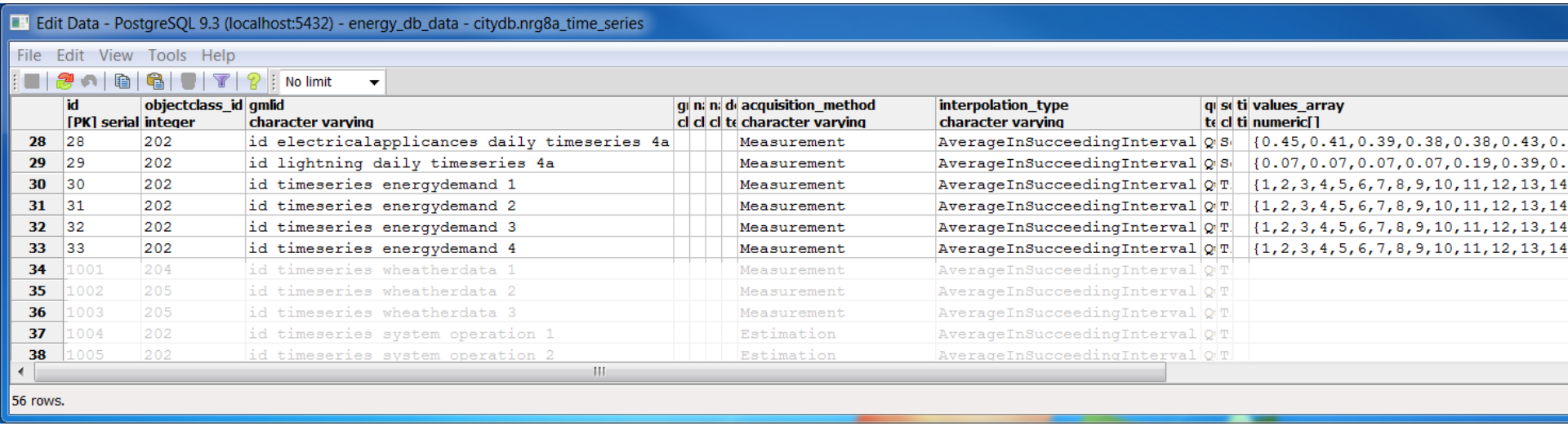
Time series: UML Diagram & mapping



Time series: ER model



(Ir)regular time series



	id [PK] serial	objectclass_id integer	gmlid character varying	acquisition_method character varying	interpolation_type character varying	values_array numeric[]
28	28	202	id electricalapplicances daily timeseries 4a	Measurement	AverageInSucceedingInterval	{0.45,0.41,0.39,0.38,0.38,0.43,0.
29	29	202	id lightning daily timeseries 4a	Measurement	AverageInSucceedingInterval	{0.07,0.07,0.07,0.07,0.19,0.39,0.
30	30	202	id timeseries energydemand 1	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
31	31	202	id timeseries energydemand 2	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
32	32	202	id timeseries energydemand 3	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
33	33	202	id timeseries energydemand 4	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
34	1001	204	id timeseries wheatherdata 1	Measurement	AverageInSucceedingInterval	
35	1002	205	id timeseries wheatherdata 2	Measurement	AverageInSucceedingInterval	
36	1003	205	id timeseries wheatherdata 3	Measurement	AverageInSucceedingInterval	
37	1004	202	id timeseries system operation 1	Estimation	AverageInSucceedingInterval	
38	1005	202	id timeseries system operation 2	Estimation	AverageInSucceedingInterval	

Table: NRG8_TIME_SERIES

(Ir)regular time series

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series

File Edit View Tools Help

No limit

	id [PK] serial	objectclass_id integer	gmlid character varying	g cl	n: cl	n: cl	d te	acquisition_method character varying	interpolation_type character varying	q te	s cl	ti ti	values_array numeric[]
28	28	202	id electricalapplicances daily timeseries 4a					Measurement	AverageInSucceedingInterval	Q	S		{0.45,0.41,0.39,0.38,0.38,0.43,0.
29	29	202	id lightning daily timeseries 4a					Measurement	AverageInSucceedingInterval	Q	S		{0.07,0.07,0.07,0.07,0.19,0.39,0.
30	30	[redacted]	id timeseries energydemand 1					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
31	31	202	id timeseries energydemand 2					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
32	32	202	id timeseries energydemand 3					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
33	33	202	id timeseries energydemand 4					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
34	1001		id timeseries wheatherdata 1					Measurement	AverageInSucceedingInterval	Q	T		
35	1002	205	id timeseries wheatherdata 2					Measurement	AverageInSucceedingInterval	Q	T		
36	1003	205	id timeseries wheatherdata 3					Measurement	AverageInSucceedingInterval	Q	T		
37	1004	202	id timeseries system operation 1					Estimation	AverageInSucceedingInterval	Q	T		
38	1005	202	id timeseries system operation 2					Estimation	AverageInSucceedingInterval	Q	T		

56 rows.

Table: NRG8_TIME_SERIES

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.objectclass

File Edit View Tools Help

No limit

	id [PK] integer	classname character varying(256)	superclass_id integer	tablename character varying(30)
101	100	TunnelDoor	98	tunnel opening
102	101	TunnelFurniture	3	tunnel furniture
103	102	HollowSpace	3	tunnel hollow space
104	103	TexCoordList	56	textureparam
105	104	TexCoordGen	56	textureparam
106	105	WaterObject	3	cityobject
107	200	Type	1	
[redacted]	[redacted]	TimeSeries	200	nrg8a timeseries
[redacted]	202	RegularTimeSeries	201	nrg8a timeseries
[redacted]	203	IrregularTimeSeries	201	nrg8a timeseries
111	204	RegularTimeSeriesFile	201	nrg8a timeseries file
112	205	IrregularTimeSeriesFile	201	nrg8a timeseries file
113	206	Schedule	200	nrg8a schedule
114	207	ConstantValueSchedule	206	nrg8a schedule
115	208	DualValueSchedule	206	nrg8a schedule
116	209	DailyPatternSchedule	206	nrg8a schedule
117	210	TimeSeriesSchedule	206	nrg8a schedule
118	211	Construction	2	nrg8a construction
119	212	Construction	211	nrg8a construction
120	213	ReverseConstruction	211	nrg8a construction
121	214	Layer	2	nrg8a layer

170 rows.

Table: OBJECTCLASS

Table: OBJECTCLASS

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.objectclass

File Edit View Tools Help

No limit

	id [PK] integer	classname character varying(256)	superclass_id integer	tablename character varying(30)
101	100	TunnelDoor	98	tunnel opening
102	101	TunnelFurniture	3	tunnel furniture
103	102	HollowSpace	3	tunnel hollow space
104	103	TexCoordList	56	textureparam
105	104	TexCoordGen	56	textureparam
106	105	WaterObject	3	cityobject
107	200	Type	1	
108	201	TimeSeries	200	nrg8a timeseries
109	202	RegularTimeSeries	201	nrg8a timeseries
110	203	IrregularTimeSeries	201	nrg8a timeseries
111	204	RegularTimeSeriesFile	201	nrg8a timeseries file
112	205	IrregularTimeSeriesFile	201	nrg8a timeseries file
113	206	Schedule	200	nrg8a schedule
114	207	ConstantValueSchedule	206	nrg8a schedule
115	208	DualValueSchedule	206	nrg8a schedule
116	209	DailyPatternSchedule	206	nrg8a schedule
117	210	TimeSeriesSchedule	206	nrg8a schedule
118	211	Construction	2	nrg8a construction
119	212	Construction	211	nrg8a construction
120	213	ReverseConstruction	211	nrg8a construction
121	214	Layer	2	nrg8a layer

170 rows.

File-based (ir)regular time series

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series

File Edit View Tools Help

No limit

	id [PK] serial	objectclass_id integer	gmlid character varying	g cl	n: cl	n: cl	d te	acquisition_method character varying	interpolation_type character varying	q te	s cl	ti ti	values_array numeric[]
28	28	202	id electricalapplicances daily timeseries 4a					Measurement	AverageInSucceedingInterval	Q	S		{0.45,0.41,0.39,0.38,0.38,0.43,0.
29	29	202	id lightning daily timeseries 4a					Measurement	AverageInSucceedingInterval	Q	S		{0.07,0.07,0.07,0.07,0.19,0.39,0.
30	30	202	id timeseries energydemand 1					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
31	31	202	id timeseries energydemand 2					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
32	32	202	id timeseries energydemand 3					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
33	33	202	id timeseries energydemand 4					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
34	1001	204	id timeseries wheatherdata 1					Measurement	AverageInSucceedingInterval	Q	T		
35	1002	205	id timeseries wheatherdata 2					Measurement	AverageInSucceedingInterval	Q	T		
36	1003	205	id timeseries wheatherdata 3					Measurement	AverageInSucceedingInterval	Q	T		
37	1004	202	id timeseries system operation 1					Estimation	AverageInSucceedingInterval	Q	T		
38	1005	202	id timeseries system operation 2					Estimation	AverageInSucceedingInterval	Q	T		

56 rows.

Table: NRG8_TIME_SERIES

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series_file

File Edit View Tools Help

No limit

	id [PK] integer	objectclass_id integer	file_path character varying	file_name character varying	file_extension character varying	nbr_header_lines integer	field_sep character varying	record_sep character varying	dec_symbol character varying	time_col_nbr integer	value_col_nbr integer	is_compress numeric(1,0
1	1001	204	c:/file path to/file/	filename	ext	1	,			1	2	0
2	1002	205	c:/file path to/file/	filename2	ext	5	,			1	2	0
3	1003	205	c:/file path to/file/	filename3	ext	5	,			1	2	0
*												

3 rows.

Table: NRG8_TIME_SERIES_FILE

File-based (ir)regular time series

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series

File Edit View Tools Help

No limit

	id [PK] serial	objectclass_id integer	gmlid character varying	g cl	n: cl	n: cl	d te	acquisition_method character varying	interpolation_type character varying	q te	s cl	ti ti	values_array numeric[]
28	28	202	id electricalapplicances daily timeseries 4a					Measurement	AverageInSucceedingInterval	Q	S		{0.45,0.41,0.39,0.38,0.38,0.43,0.
29	29	202	id lightning daily timeseries 4a					Measurement	AverageInSucceedingInterval	Q	S		{0.07,0.07,0.07,0.07,0.19,0.39,0.
30	30	202	id timeseries energydemand 1					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
31	31	202	id timeseries energydemand 2					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
32	32	202	id timeseries energydemand 3					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
33	33	202	id timeseries energydemand 4					Measurement	AverageInSucceedingInterval	Q	T		{1,2,3,4,5,6,7,8,9,10,11,12,13,14
34	1001	204	id timeseries wheatherdata 1					Measurement	AverageInSucceedingInterval	Q	T		
35	1002	205	id timeseries wheatherdata 2					Measurement	AverageInSucceedingInterval	Q	T		
36	1003	205	id timeseries wheatherdata 3					Measurement	AverageInSucceedingInterval	Q	T		
37	1004	202	id timeseries system operation 1					Estimation	AverageInSucceedingInterval	Q	T		
38	1005	202	id timeseries system operation 2					Estimation	AverageInSucceedingInterval	Q	T		

56 rows.

Table: NRG8_TIME_SERIES

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series_file

File Edit View Tools Help

No limit

	id [PK] integer	objectclass_id	file_path character varying	file_name character varying	file_extension character varying	nbr_header_lines integer	field_sep character varying	record_sep character varying	dec_symbol character varying	time_col_nbr integer	value_col_nbr integer	is_compress numeric(1,0
1	1001	204	c:/file path to/file/	filename	ext	1	,			1	2	0
2	1002	205	c:/file path to/file/	filename2	ext	5	,			1	2	0
3	1003	205	c:/file path to/file/	filename3	ext	5	,			1	2	0
*												

3 rows.

Table: NRG8_TIME_SERIES_FILE

File-based (ir)regular time series

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series

File Edit View Tools Help

No limit

	id [PK] serial	objectclass_id integer	gmlid character varying	acquisition_method character varying	interpolation_type character varying	values_array numeric[]
28	28	202	id electricalapplicances daily timeseries 4a	Measurement	AverageInSucceedingInterval	{0.45,0.41,0.39,0.38,0.38,0.43,0.
29	29	202	id lightning daily timeseries 4a	Measurement	AverageInSucceedingInterval	{0.07,0.07,0.07,0.07,0.19,0.39,0.
30	30	202	id timeseries energydemand 1	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
31	31	202	id timeseries energydemand 2	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
32	32	202	id timeseries energydemand 3	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
33	33	202	id timeseries energydemand 4	Measurement	AverageInSucceedingInterval	{1,2,3,4,5,6,7,8,9,10,11,12,13,14
34	1001	204	id timeseries wheatherdata 1	Measurement	AverageInSucceedingInterval	
35	1002	205	id timeseries wheatherdata 2	Measurement	AverageInSucceedingInterval	
36	1003	205	id timeseries wheatherdata 3	Measurement	AverageInSucceedingInterval	
37	1004	202	id timeseries system operation 1	Estimation	AverageInSucceedingInterval	
38	1005	202	id timeseries system operation 2	Estimation	AverageInSucceedingInterval	

56 rows.

Table: NRG8_TIME_SERIES

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series_file

File Edit View Tools Help

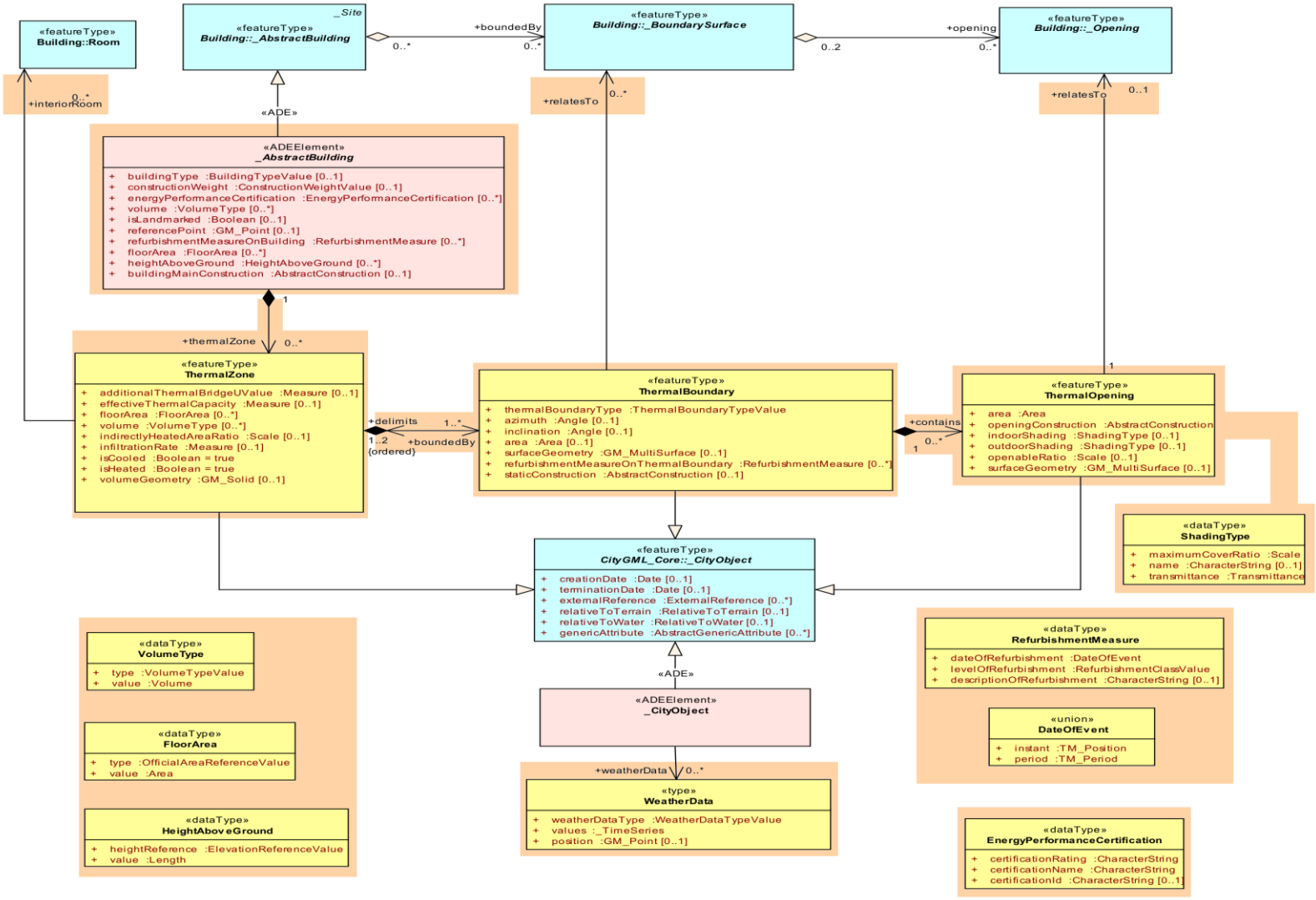
No limit

	id [PK] integer	objectclass_id integer	file_path character varying	file_name character varying	file_extension character varying	nbr_header_lines integer	field_sep character varying	record_sep character varying	dec_symbol character varying	time_col_nbr integer	value_col_nbr integer	is_compress numeric(1,0
1	1001	204	c:/file path to/file/	filename	ext	1	,			1	2	0
2	1002	205	c:/file path to/file/	filename2	ext	5	,			1	2	0
3	1003	205	c:/file path to/file/	filename3	ext	5	,			1	2	0
*												

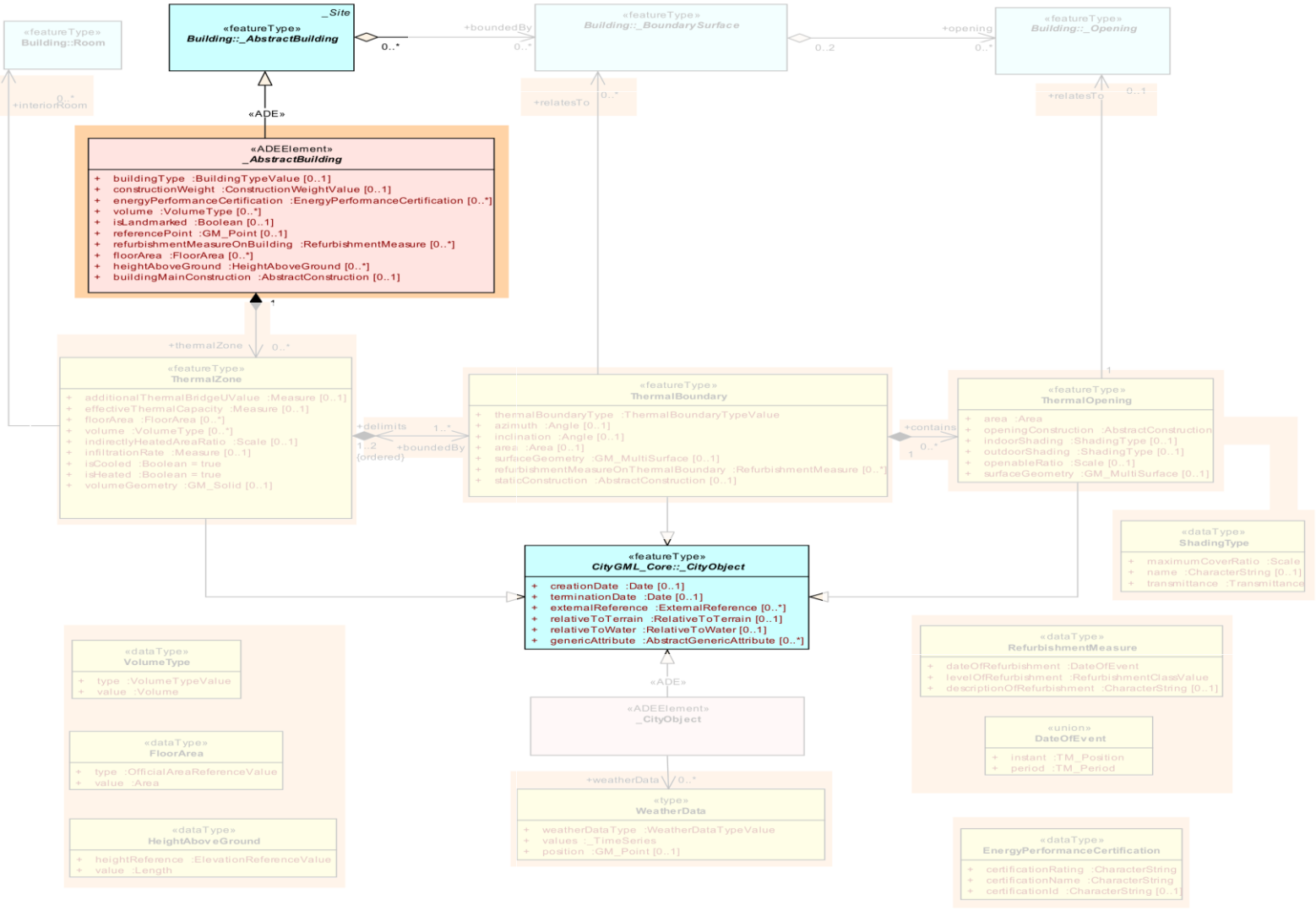
3 rows.

Table: NRG8_TIME_SERIES_FILE

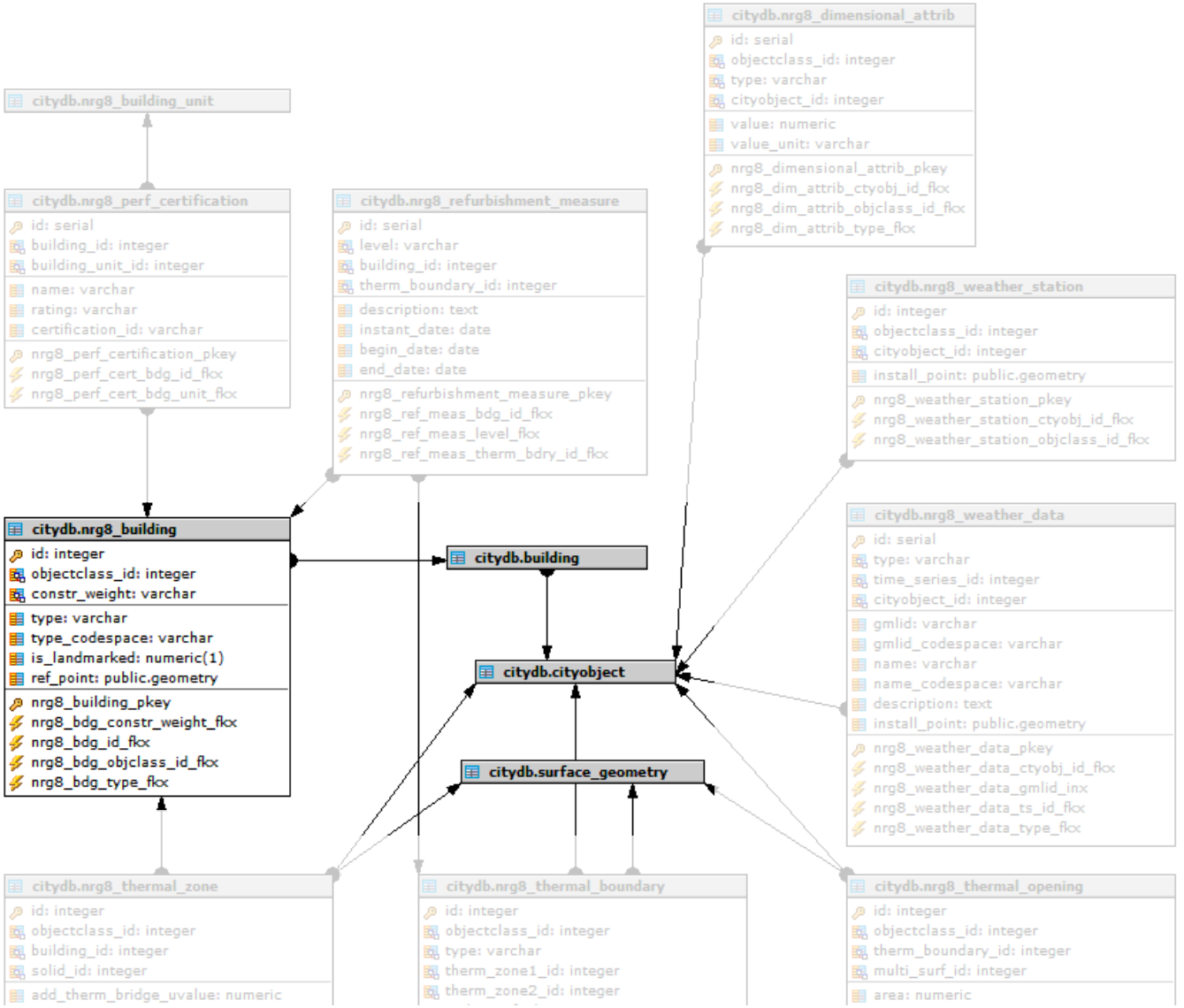
AbstractBuilding: UML Diagram & mapping



AbstractBuilding: UML Diagram & mapping



AbstractBuilding: ER model



Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_...

File Edit View Tools Help

No limit

	id [PK] integer	objectclass_id integer	gmlid character varying(256)
120	291	251	id heat exchanger 1
121	292	251	id heat exchanger 2
122	301	252	id mech vent 1
123	302	252	id mech vent 2
124	311	253	id chiller 1
125	312	253	id chiller 2
126	321	254	id aircompressor 1
127	322	254	id aircompressor 2
128	401	23	id cityobject group 1
129	402	23	id cityobject group 2
130	403	23	id cityobject group 3
131	404	23	id cityobject group 4
132	405	23	id cityobject group 5
133	1000	26	id building 2
134	1001	26	id building 3
135	1002	26	id building 4
136	1003	26	id building 5
137	1010	26	id building 1010
138	1020	219	id weather station 1020
139	1021	219	id weather station 1021
140	1022	219	id weather station 1022
141	1023	219	id weather station 1023
*			

141 rows.

Table: CITYOBJECT

Table: BUILDING

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.building

File Edit View Tools Help

No limit

	id [PK] integer	building_parent_id integer	building_root_id integer	cl cl	cl cl	function character varying(1000)	fu cl	u: cl	u: cl	year_of_construction date	year_of_ date
1	1		1			Supermarket				1989-01-01	
2	1000		1000			Residential				1990-01-01	
3	1001		1000			Residential				1990-01-01	
4	1002		1000			Residential				1990-01-01	
5	1003		1000			Residential				1990-01-01	
6	1010		1010			Garage				1990-01-01	
*											

Data from one cell copied to clipboard.

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_building

File Edit View Tools Help

No limit

	id [PK] integer	objectclass_id integer	type character varying	type_codespace character varying	constr_weight character varying	is_landmarked numeric(1,0)	ref_point geometry(PointZ,31256
1	1	26	Multi-Family House	''	Medium	0	01010000A0187A000
2	1000	26	Multi-Family House	''	Medium	0	01010000A0187A000
3	1001	26	Single-Family House	''	Medium	0	01010000A0187A000
4	1002	26	Apartment Block	''	Medium	0	01010000A0187A000
5	1003	26	Terrace House	''	Medium	0	01010000A0187A000
*							

5 rows.

Table: NRG8_BUILDING

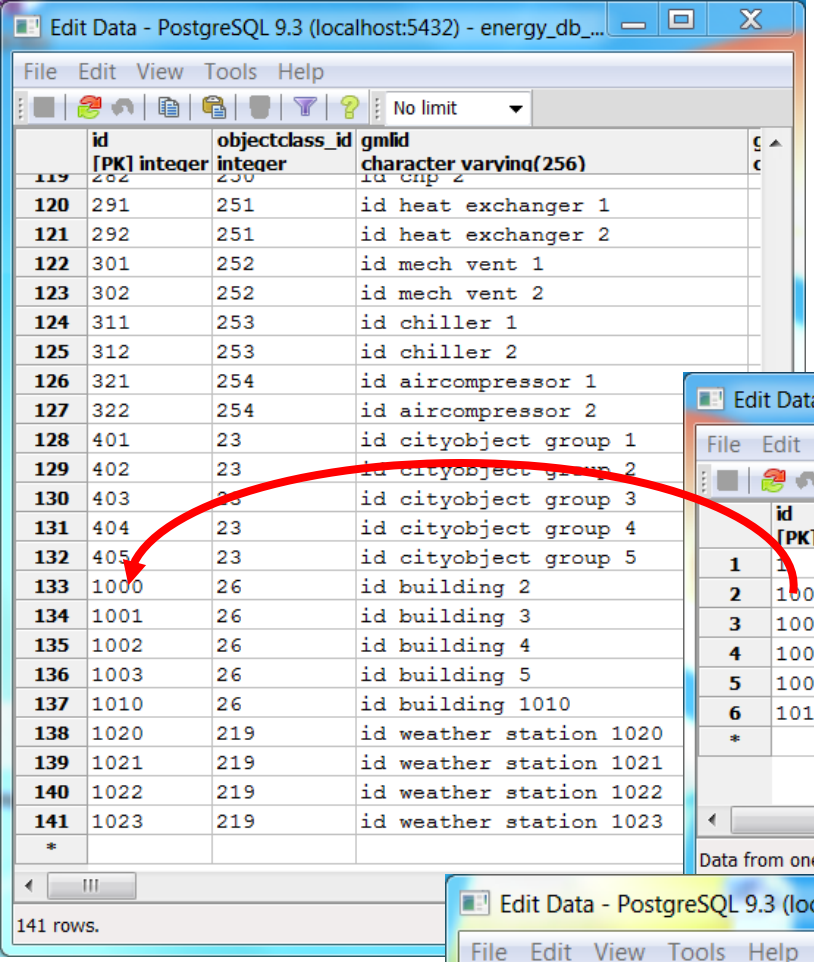
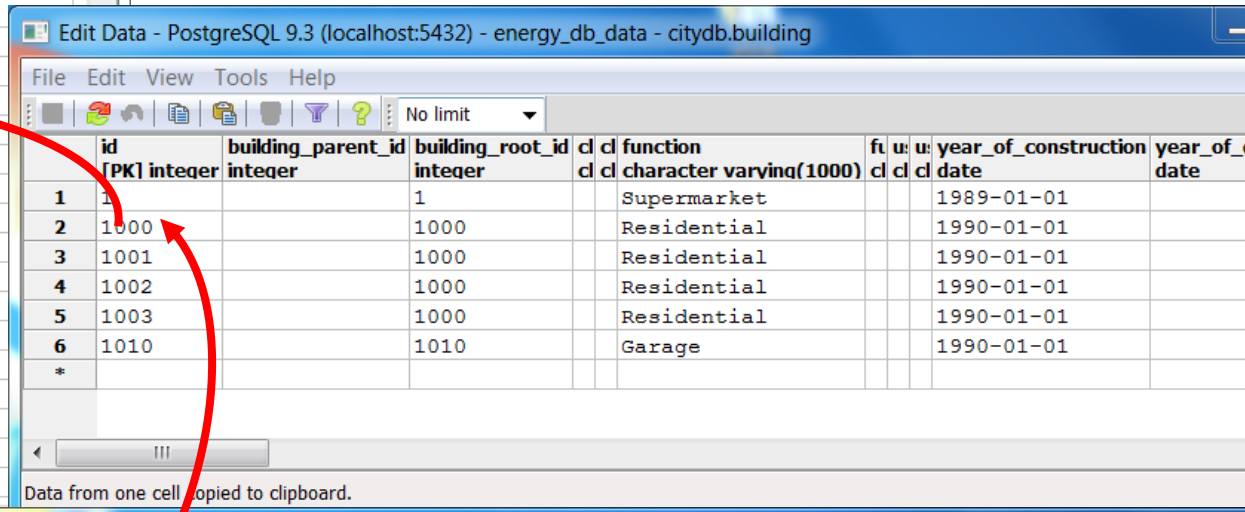


Table: CITYOBJECT

Table: BUILDING



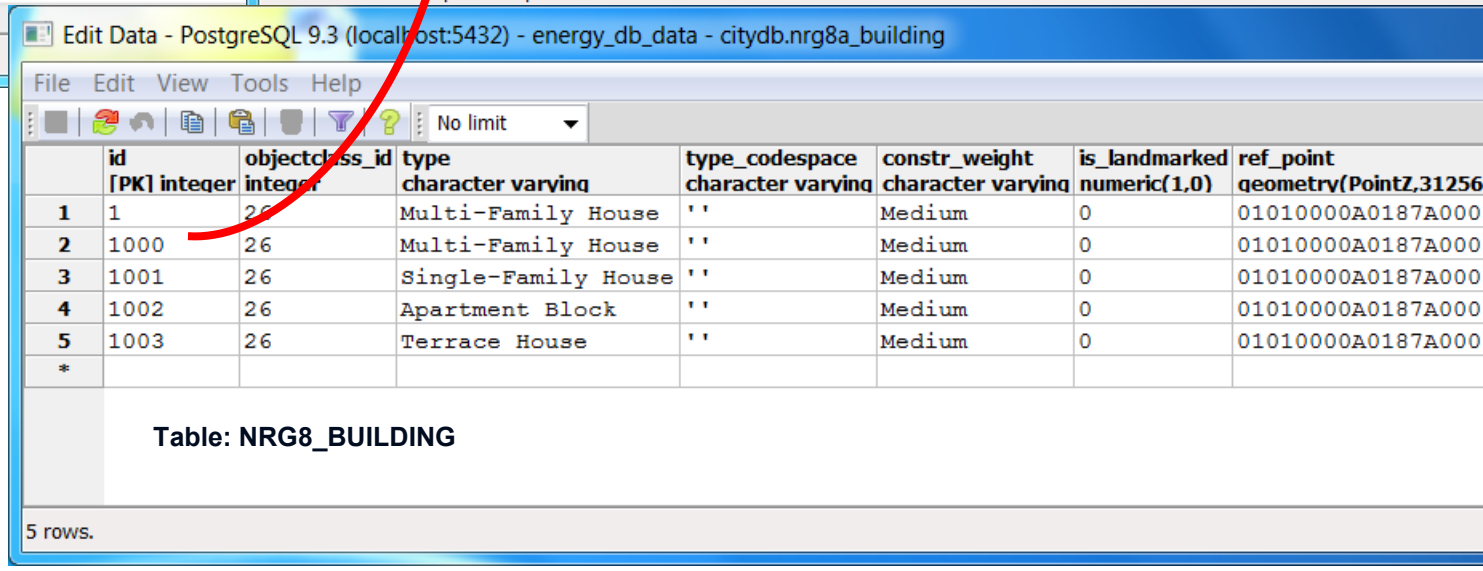


Table: NRG8_BUILDING

Interacting with the (extended) 3DCityDB

- How to delete data from the database?
  - Use the *delete* stored procedures (refer to the documentation)

- How to input 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
https://github.com/gioagu/3dcitydb_ade/blob/master/02_energy_ade/test_data/Energy_ADE_Insert_data_example_scripts.sql

Interacting with the (extended) 3DCityDB

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

LEFT SIDE:
Example with a **RegularTimeSeries** object

RIGHT SIDE:
Example with a **RegularTimeSeriesFile** object

Notes:
The **id**, **objectclass_id** MUST be set by the user

Interacting with the (extended) 3DCityDB

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

Interacting with the (extended) 3DCityDB

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');
```

```
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;
```

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

Interacting with the (extended) 3DCityDB

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

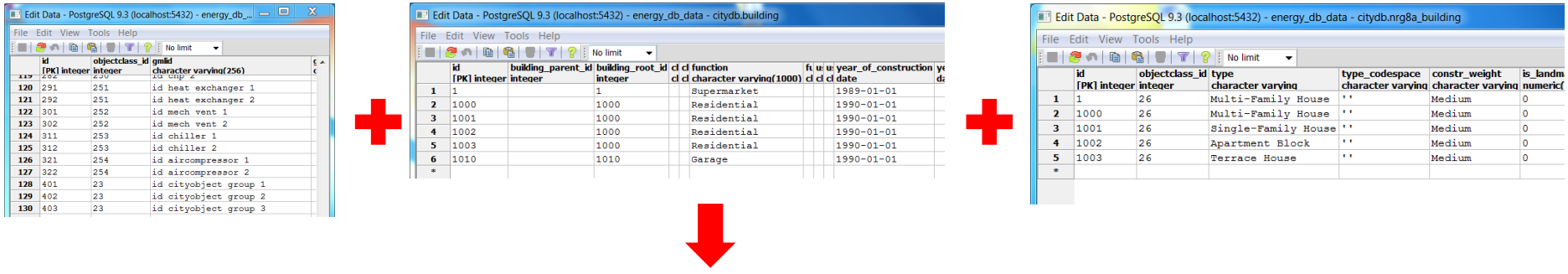
The **id** value is returned by the stored procedure

Interacting with the (extended) 3DCityDB

4. Use the updatable views (in citydb_view schema)

- Updatable views hide the complexity of data spread over multiple tables by defining a virtual joined table which can be accessed (if updatable) like any other "standard" table.

- As the views are built upon the "smart" insert stored procedures, the same benefits still apply. In addition: UPDATE and DELETE operations are allowed, too.



Interacting with the (extended) 3DCityDB

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;
```

Interacting with the (extended) 3DCityDB

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

Interacting with the (extended) 3DCityDB

- 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…

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 http://www.isprs.org/tc4-symposium2018/
  - **3D GeoInfo 2018** https://www.utwente.nl/en/3dgeoinfo2018/
  - **Smart Data and Smart Cities 2018** http://www.udms.net/

- 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

WG 10: **Advanced Geospatial Applications for Smart Cities and Regions**

- Spatial information challenges in modelling urban air quality, noise mapping, micro-climate/heat islands, citizens' health
- Data models for estimation/simulation/co-simulation of energy demand and supply, exploitation of local renewable energies, 3D solar cadastres;
- Mobility: transportation, logistics, transport-oriented development;
- Socio-economic and legal aspects of 3D cadastre, population growth/decline, land use and soil consumption;
- Emergency response and planning: disaster mapping and prevention/relief, public safety, cascading effects due to infrastructure failures;
- Tourism-related applications with focus on virtual tourism, web-based and mobile applications;

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