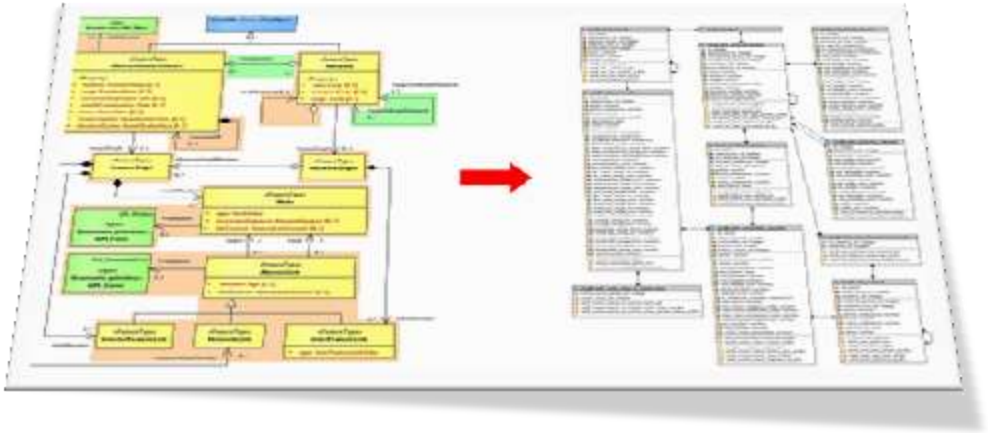
Some updates on the 3DCityDB extension for ADEs

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

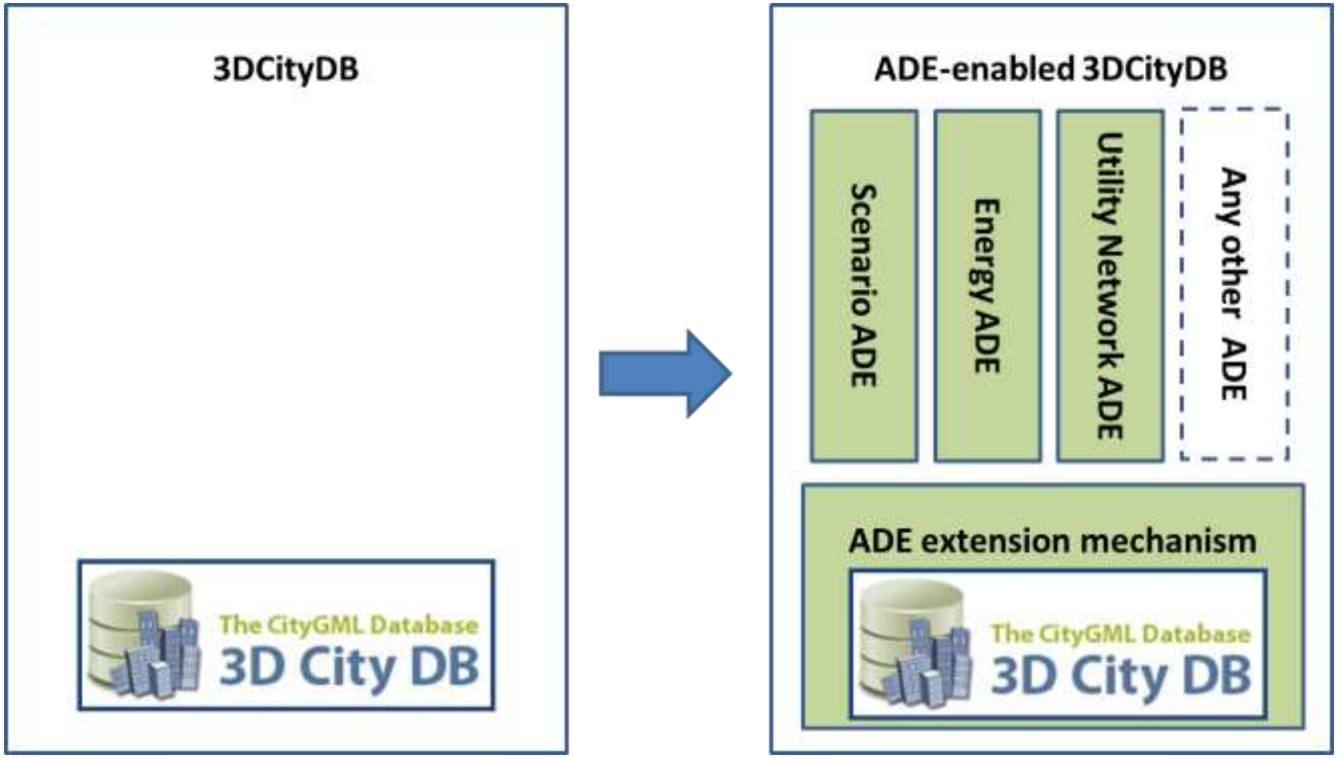
- So far, DB implementations for the Energy/Utility Network ADE:
  - partial AND/OR
  - non-open AND/OR
  - poorly or not documented at all

- Some initial results (for the Energy ADE) presented last May in Grenoble
  - Please refer to those slides for more details: 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



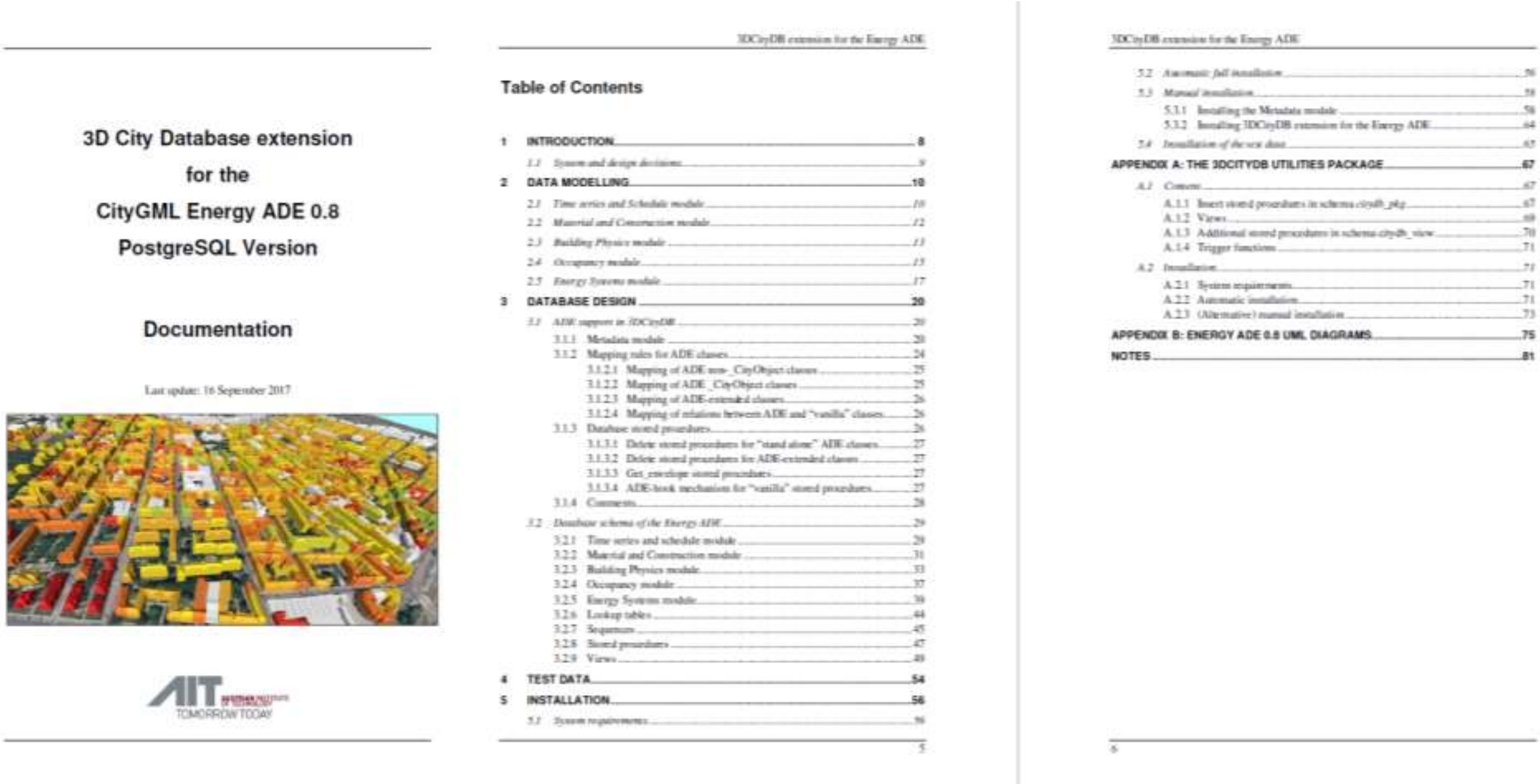
Motivation & goals

- Gather and share experience on how to extend to 3DCityDB for *any* ADE
  - For further tools (citygml4j, Importer/Exporter, etc.)

- 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



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

Documentation

Last update: 16 September 2017

Table of Contents

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

Design criteria (excerpt)

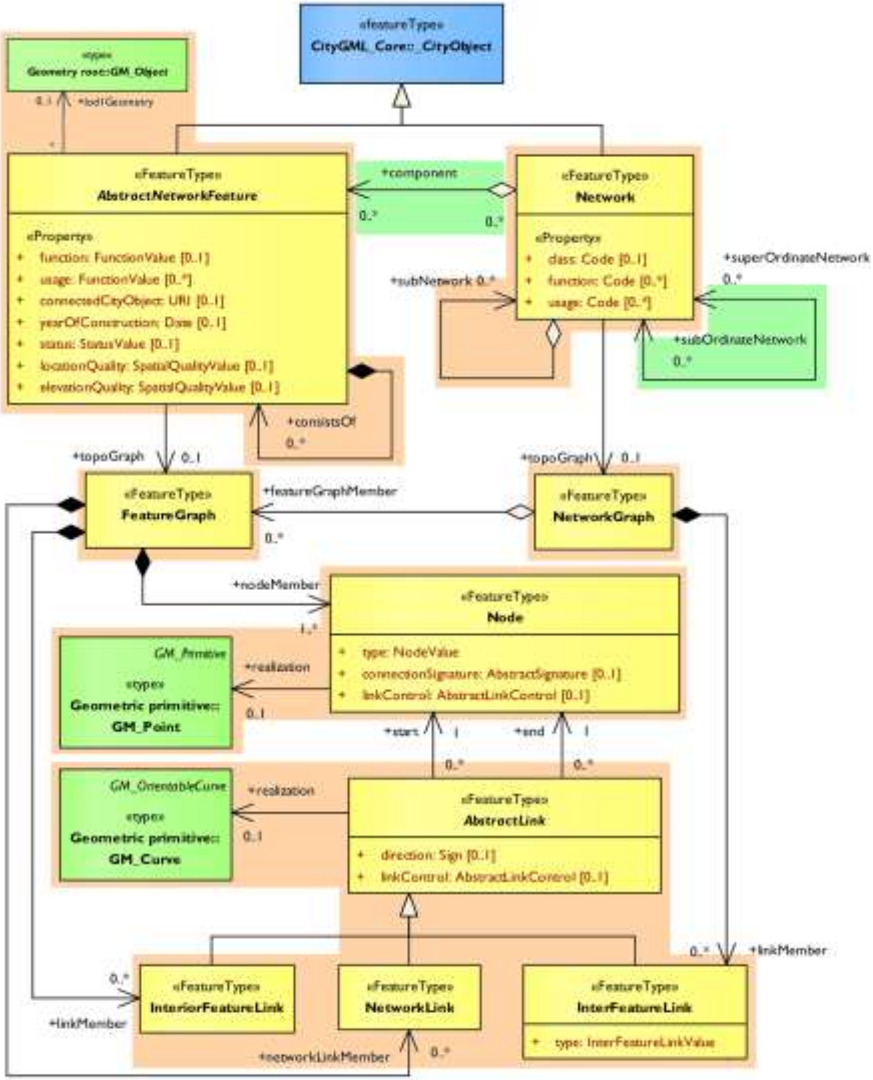
- Build upon the existing objects of the 3DCityDB… but keep the original ones untouched (for the sake of the Importer/Exporter)
- Define a non-concurrent way of extending the 3DCityDB with *any* ADEs (e.g. Energy ADE + Utility Network ADE)
- Stay close to the original “style” of the 3DCityDB when it 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)

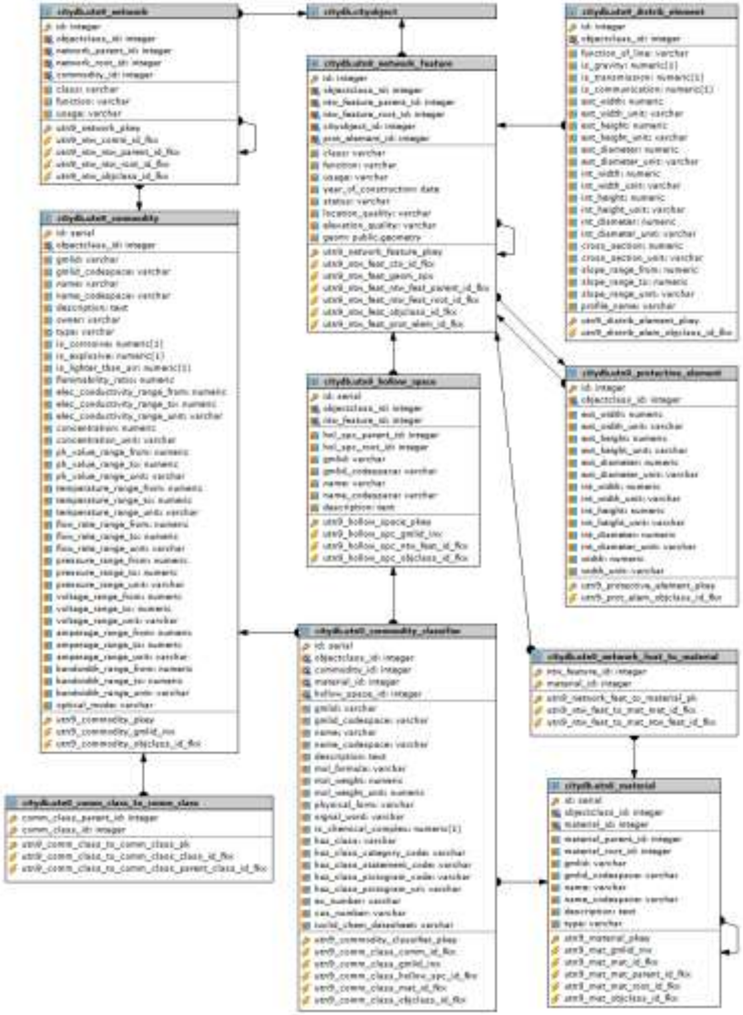
Implementation steps

- 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




citydb.utn9_network
citydb.cityobject
citydb.utn9_distrib_element
citydb.utn9_network_feature
citydb.utn9_commodity
citydb.utn9_hollow_space
citydb.utn9_protective_element
citydb.utn9_commodity_classifier
citydb.utn9_network_feat_to_material
citydb.utn9_comm_class_to_comm_class
citydb.utn9_material

Interacting with the (extended) 3DCityDB

- "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

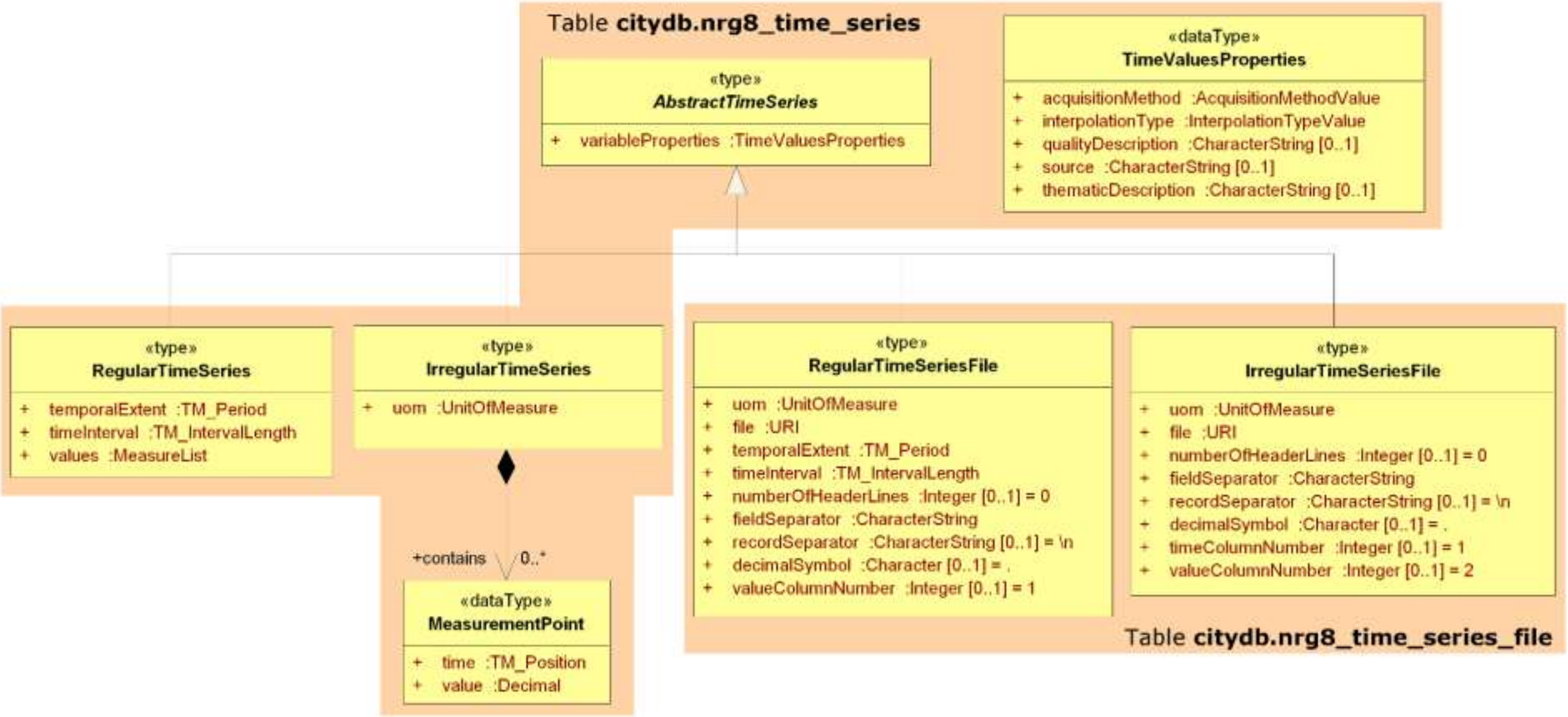


Table **citydb.nrg8_time_series**

«type»
AbstractTimeSeries
+ variableProperties :TimeValuesProperties

«dataType»
TimeValuesProperties
+ acquisitionMethod :AcquisitionMethodValue
+ interpolationType :InterpolationTypeValue
+ qualityDescription :CharacterString [0..1]
+ source :CharacterString [0..1]
+ thematicDescription :CharacterString [0..1]

«type»
RegularTimeSeries
+ temporalExtent :TM_Period
+ timeInterval :TM_IntervalLength
+ values :MeasureList

«type»
IrregularTimeSeries
+ uom :UnitOfMeasure

+contains 0..*

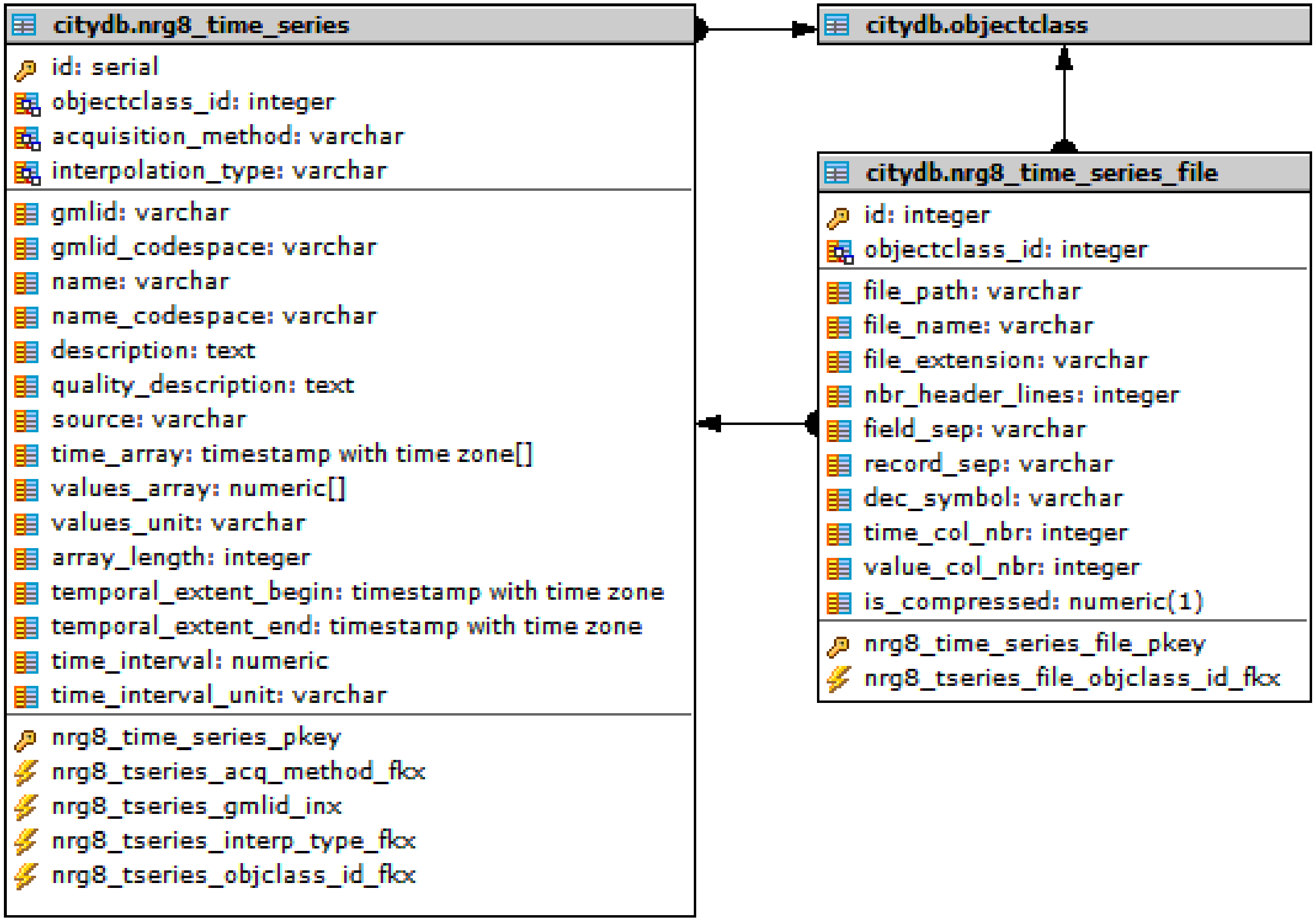
«dataType»
MeasurementPoint
+ time :TM_Position
+ value :Decimal

«type»
RegularTimeSeriesFile
+ uom :UnitOfMeasure
+ file :URI
+ temporalExtent :TM_Period
+ timeInterval :TM_IntervalLength
+ numberOfHeaderLines :Integer [0..1] = 0
+ fieldSeparator :CharacterString
+ recordSeparator :CharacterString [0..1] = \n
+ decimalSymbol :Character [0..1] = .
+ valueColumnNumber :Integer [0..1] = 1

«type»
IrregularTimeSeriesFile
+ uom :UnitOfMeasure
+ file :URI
+ numberOfHeaderLines :Integer [0..1] = 0
+ fieldSeparator :CharacterString
+ recordSeparator :CharacterString [0..1] = \n
+ decimalSymbol :Character [0..1] = .
+ timeColumnNumber :Integer [0..1] = 1
+ valueColumnNumber :Integer [0..1] = 2

Table **citydb.nrg8_time_series_file**

Time series: ER model



citydb.nrg8_time_series

id: serial
objectclass_id: integer
acquisition_method: varchar
interpolation_type: varchar

gmlid: varchar
gmlid_codespace: varchar
name: varchar
name_codespace: varchar
description: text
quality_description: text
source: varchar
time_array: timestamp with time zone[]
values_array: numeric[]
values_unit: varchar
array_length: integer
temporal_extent_begin: timestamp with time zone
temporal_extent_end: timestamp with time zone
time_interval: numeric
time_interval_unit: varchar

nrg8_time_series_pkey
nrg8_tseries_acq_method_fkx
nrg8_tseries_gmlid_inx
nrg8_tseries_interp_type_fkx
nrg8_tseries_objclass_id_fkx

citydb.objectclass

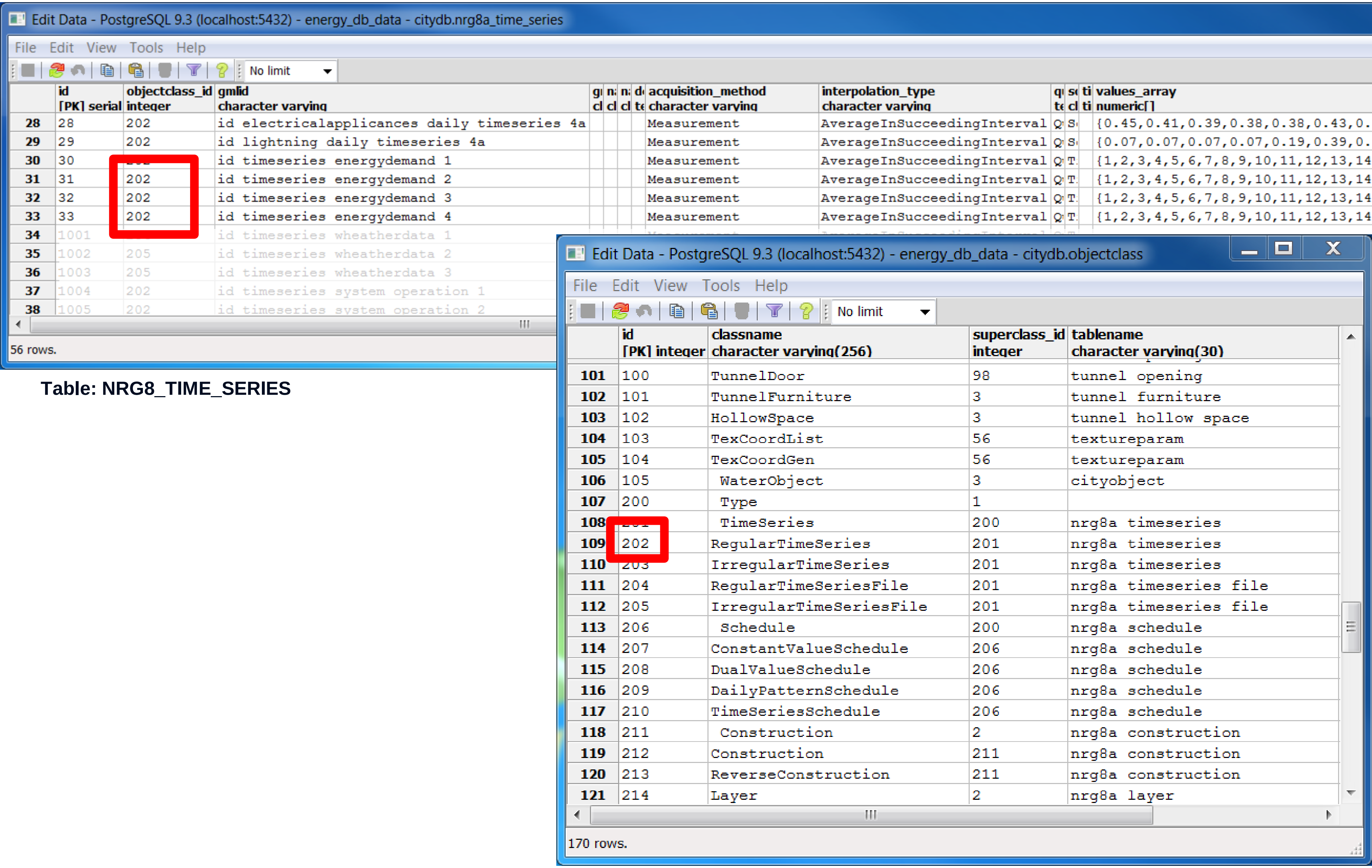
citydb.nrg8_time_series_file

id: integer
objectclass_id: integer

file_path: varchar
file_name: varchar
file_extension: varchar
nbr_header_lines: integer
field_sep: varchar
record_sep: varchar
dec_symbol: varchar
time_col_nbr: integer
value_col_nbr: integer
is_compressed: numeric(1)

nrg8_time_series_file_pkey
nrg8_tseries_file_objclass_id_fkx

(Ir)regular time series



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

	id [PK] serial	objectclass_id integer	gmlid character varying	gı 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		id timeseries wheatherdata 1										
35	1002	205	id timeseries wheatherdata 2										
36	1003	205	id timeseries wheatherdata 3										
37	1004	202	id timeseries system operation 1										
38	1005	202	id timeseries system operation 2										

56 rows.

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

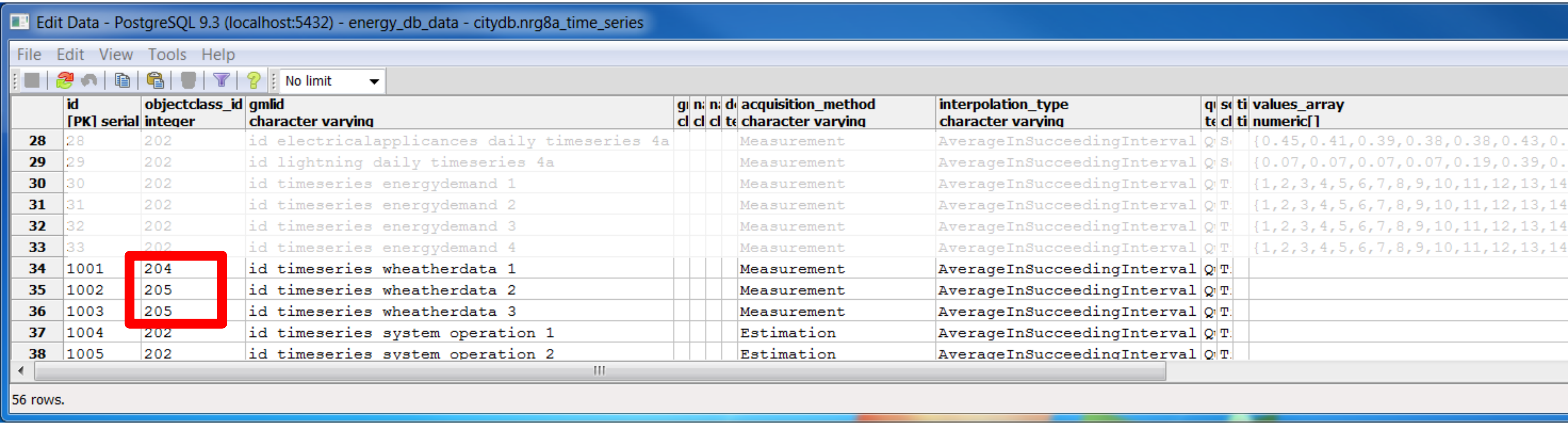
	id [PK] integer	classname character varying(256)	superclass_id integer	tablename character varying(30)
101	100	TunnelDoor	98	tunnel opening
102	101	TunnelFurniture	3	tunnel furniture
103	102	HollowSpace	3	tunnel hollow space
104	103	TexCoordList	56	textureparam
105	104	TexCoordGen	56	textureparam
106	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.

Table: NRG8_TIME_SERIES

Table: OBJECTCLASS

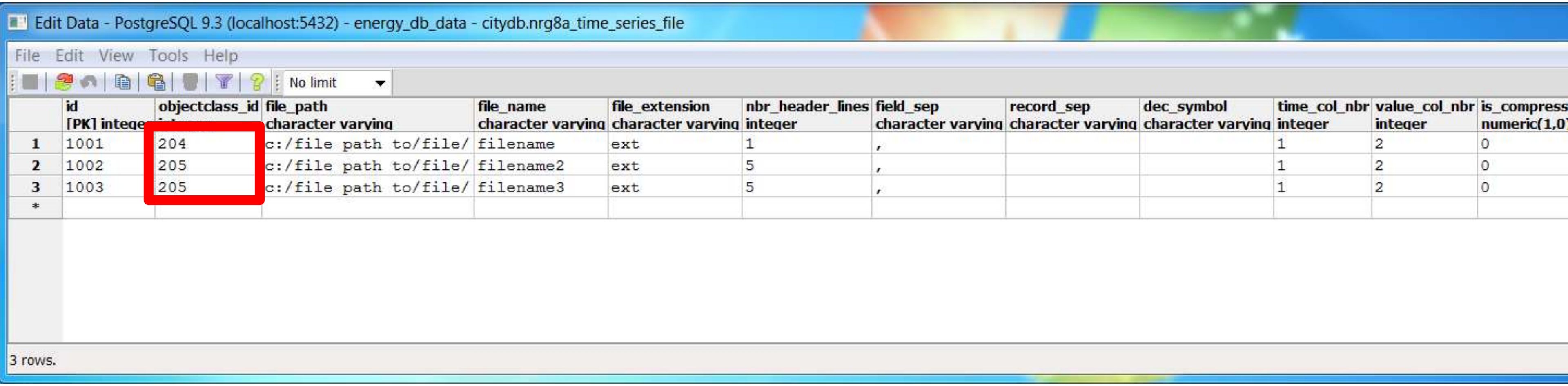
File-based (ir)regular time series


Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series
File Edit View Tools Help
No limit

	id	objectclass_id	gmlid	gı	n:	n:	d	acquisition_method	interpolation_type	q	s	ti	values_array
	[PK] serial	integer	character varying	cl	cl	cl	te	character varying	character varying	te	cl	ti	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

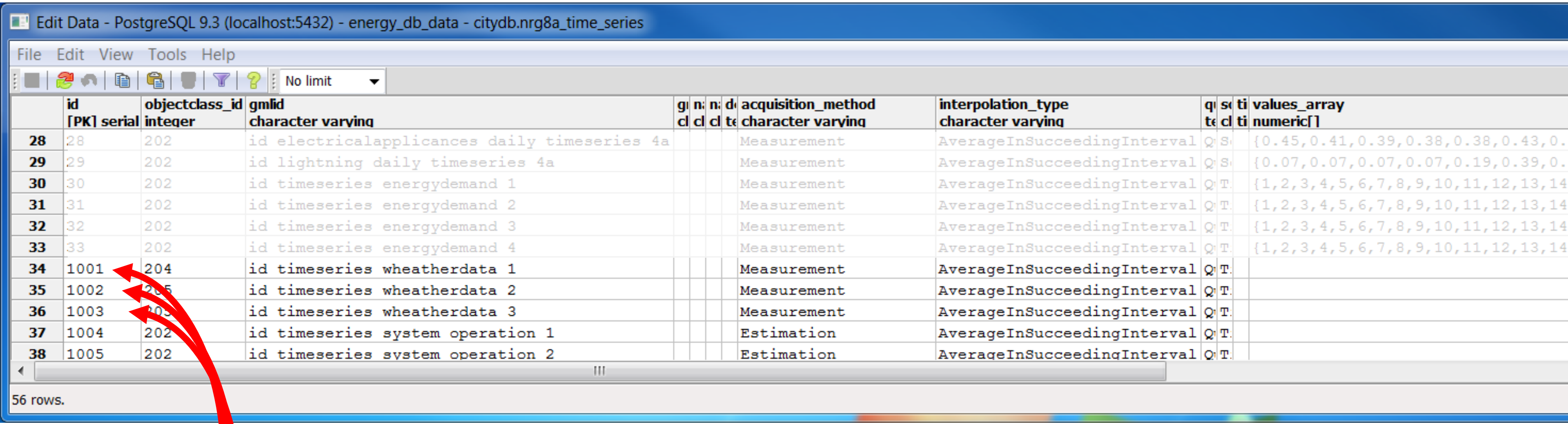

Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series_file
File Edit View Tools Help
No limit

	id	objectclass_id	file_path	file_name	file_extension	nbr_header_lines	field_sep	record_sep	dec_symbol	time_col_nbr	value_col_nbr	is_compress
	[PK] intege		character varying	character varying	character varying	integer	character varying	character varying	character varying	integer	integer	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

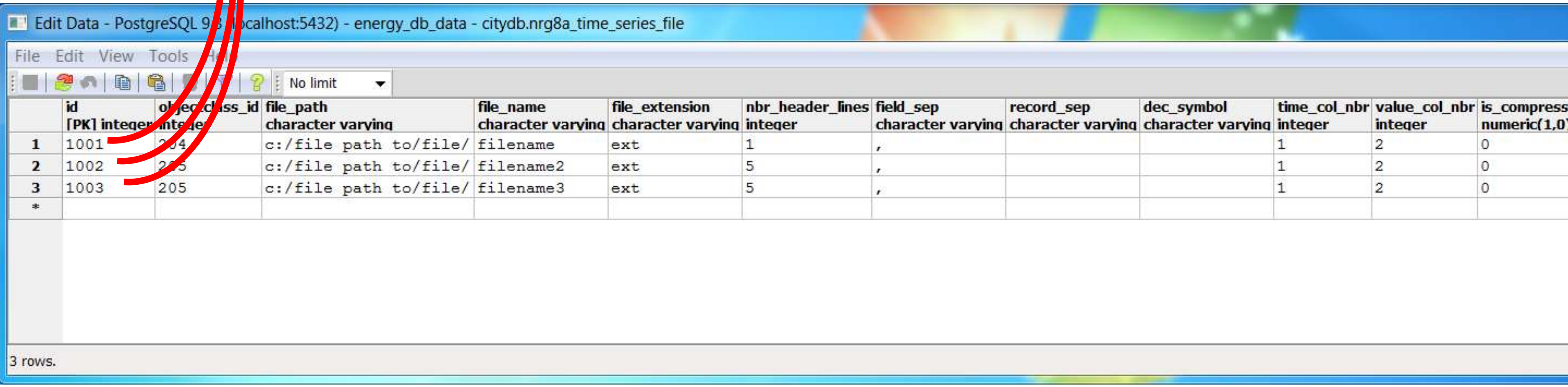


Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series

	id [PK] serial	objectclass_id integer	gmlid character varying	gı 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



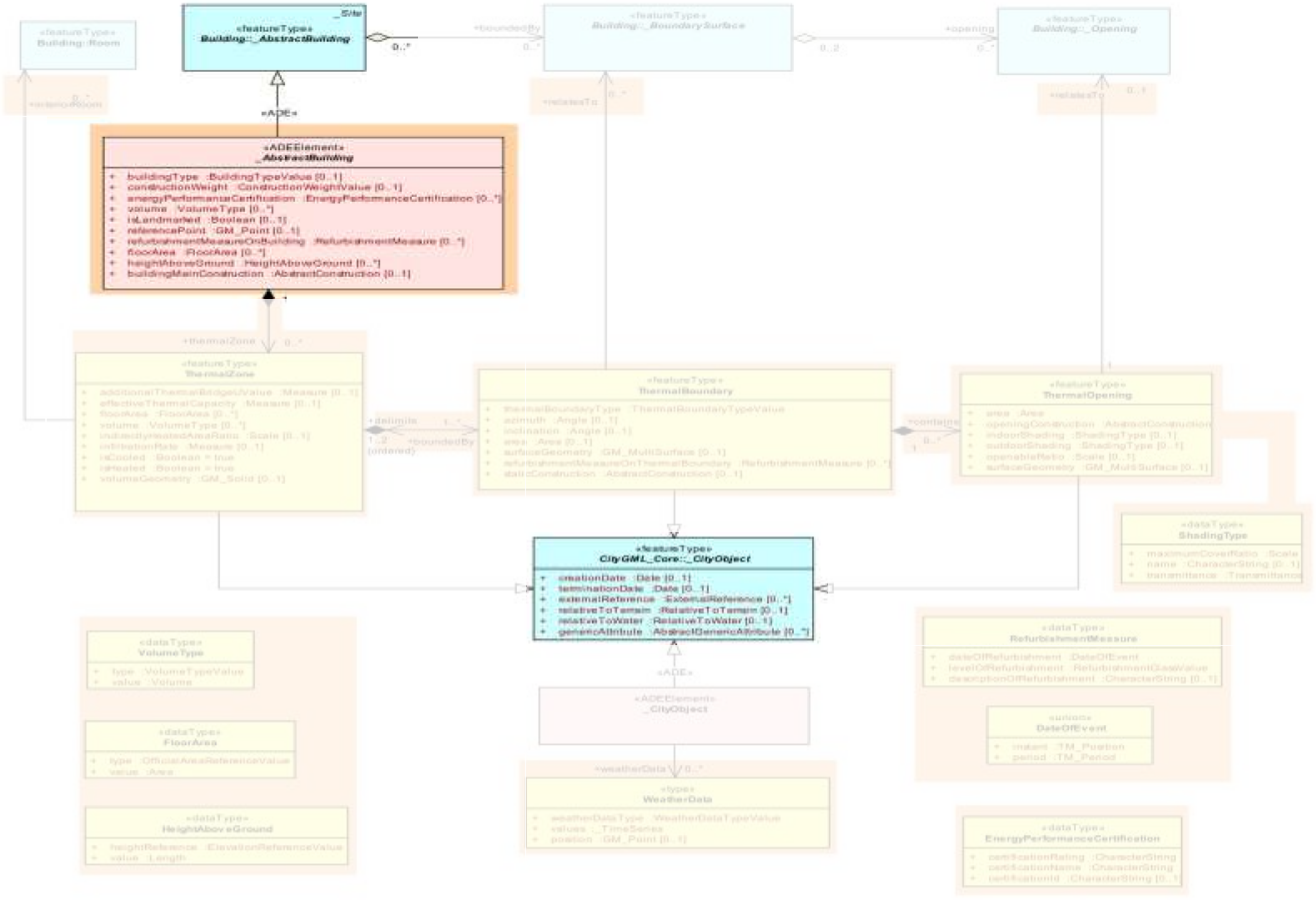
Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb.nrg8a_time_series_file

	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



«featureType»
Building::Room

+interiorRoom 0..*

_Site
«featureType»
Building::_AbstractBuilding

0..*

+boundedBy 0..*

«featureType»
Building::_BoundarySurface

0..2

+opening 0..*

«featureType»
Building::_Opening

+relatesTo 0..*

+relatesTo 0..1

«ADE»

«ADEElement»
_AbstractBuilding

- + buildingType :BuildingTypeValue [0..1]
- + constructionWeight :ConstructionWeightValue [0..1]
- + energyPerformanceCertification :EnergyPerformanceCertification [0..*]
- + volume :VolumeType [0..*]
- + isLandmarked :Boolean [0..1]
- + referencePoint :GM_Point [0..1]
- + refurbishmentMeasureOnBuilding :RefurbishmentMeasure [0..*]
- + floorArea :FloorArea [0..*]
- + heightAboveGround :HeightAboveGround [0..*]
- + buildingMainConstruction :AbstractConstruction [0..1]

1

+thermalZone 0..*

«featureType»
ThermalZone

- + additionalThermalBridgeUValue :Measure [0..1]
- + effectiveThermalCapacity :Measure [0..1]
- + floorArea :FloorArea [0..*]
- + volume :VolumeType [0..*]
- + indirectlyHeatedAreaRatio :Scale [0..1]
- + infiltrationRate :Measure [0..1]
- + isCooled :Boolean = true
- + isHeated :Boolean = true
- + volumeGeometry :GM_Solid [0..1]

+delimits 1..*

1..2 +boundedBy
{ordered}

«featureType»
ThermalBoundary

- + thermalBoundaryType :ThermalBoundaryTypeValue
- + azimuth :Angle [0..1]
- + inclination :Angle [0..1]
- + area :Area [0..1]
- + surfaceGeometry :GM_MultiSurface [0..1]
- + refurbishmentMeasureOnThermalBoundary :RefurbishmentMeasure [0..*]
- + staticConstruction :AbstractConstruction [0..1]

+contains 0..*

1

1

«featureType»
ThermalOpening

- + area :Area
- + openingConstruction :AbstractConstruction
- + indoorShading :ShadingType [0..1]
- + outdoorShading :ShadingType [0..1]
- + openableRatio :Scale [0..1]
- + surfaceGeometry :GM_MultiSurface [0..1]

«dataType»
ShadingType

- + maximumCoverRatio :Scale
- + name :CharacterString [0..1]
- + transmittance :Transmittance

«featureType»
CityGML_Core::_CityObject

- + creationDate :Date [0..1]
- + terminationDate :Date [0..1]
- + externalReference :ExternalReference [0..*]
- + relativeToTerrain :RelativeToTerrain [0..1]
- + relativeToWater :RelativeToWater [0..1]
- + genericAttribute :AbstractGenericAttribute [0..*]

«ADE»

«ADEElement»
_CityObject

+weatherData 0..*

«type»
WeatherData

- + weatherDataType :WeatherDataTypeValue
- + values :_TimeSeries
- + position :GM_Point [0..1]

«dataType»
VolumeType

- + type :VolumeTypeValue
- + value :Volume

«dataType»
FloorArea

- + type :OfficialAreaReferenceValue
- + value :Area

«dataType»
HeightAboveGround

- + heightReference :ElevationReferenceValue
- + value :Length

«dataType»
RefurbishmentMeasure

- + dateOfRefurbishment :DateOfEvent
- + levelOfRefurbishment :RefurbishmentClassValue
- + descriptionOfRefurbishment :CharacterString [0..1]

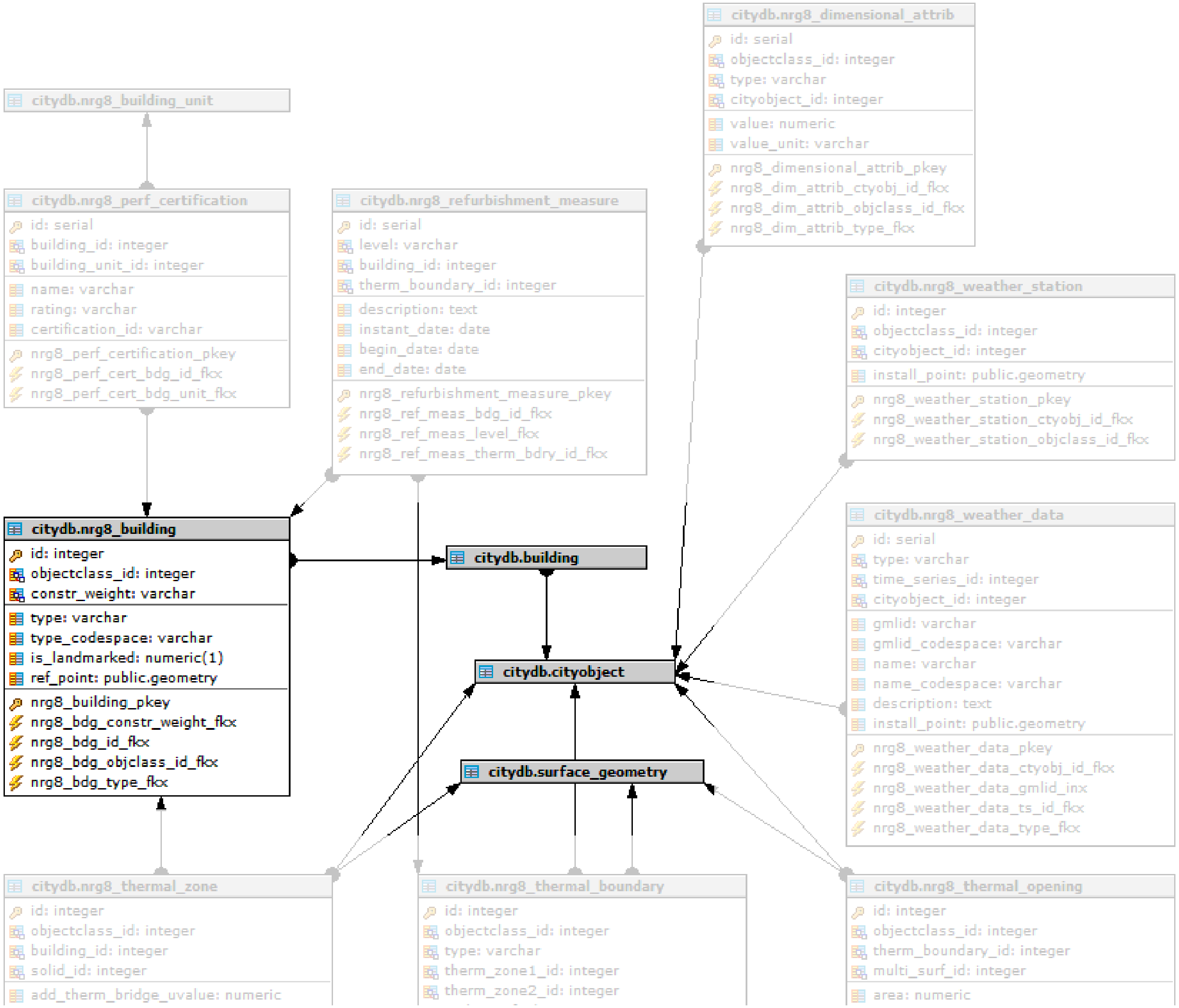
«union»
DateOfEvent

- + instant :TM_Position
- + period :TM_Period

«dataType»
EnergyPerformanceCertification

- + certificationRating :CharacterString
- + certificationName :CharacterString
- + certificationId :CharacterString [0..1]

AbstractBuilding: ER model



citydb.nrg8_dimensional_attrib
id: serial
objectclass_id: integer
type: varchar
cityobject_id: integer
value: numeric
value_unit: varchar
nrg8_dimensional_attrib_pkey
nrg8_dim_attrib_ctyobj_id_fkx
nrg8_dim_attrib_objclass_id_fkx
nrg8_dim_attrib_type_fkx

citydb.nrg8_building_unit

citydb.nrg8_perf_certification
id: serial
building_id: integer
building_unit_id: integer
name: varchar
rating: varchar
certification_id: varchar
nrg8_perf_certification_pkey
nrg8_perf_cert_bdg_id_fkx
nrg8_perf_cert_bdg_unit_fkx

citydb.nrg8_refurbishment_measure
id: serial
level: varchar
building_id: integer
therm_boundary_id: integer
description: text
instant_date: date
begin_date: date
end_date: date
nrg8_refurbishment_measure_pkey
nrg8_ref_meas_bdg_id_fkx
nrg8_ref_meas_level_fkx
nrg8_ref_meas_therm_bdry_id_fkx

citydb.nrg8_weather_station
id: integer
objectclass_id: integer
cityobject_id: integer
install_point: public.geometry
nrg8_weather_station_pkey
nrg8_weather_station_ctyobj_id_fkx
nrg8_weather_station_objclass_id_fkx

citydb.nrg8_building
id: integer
objectclass_id: integer
constr_weight: varchar
type: varchar
type_codespace: varchar
is_landmarked: numeric(1)
ref_point: public.geometry
nrg8_building_pkey
nrg8_bdg_constr_weight_fkx
nrg8_bdg_id_fkx
nrg8_bdg_objclass_id_fkx
nrg8_bdg_type_fkx

citydb.building

citydb.cityobject

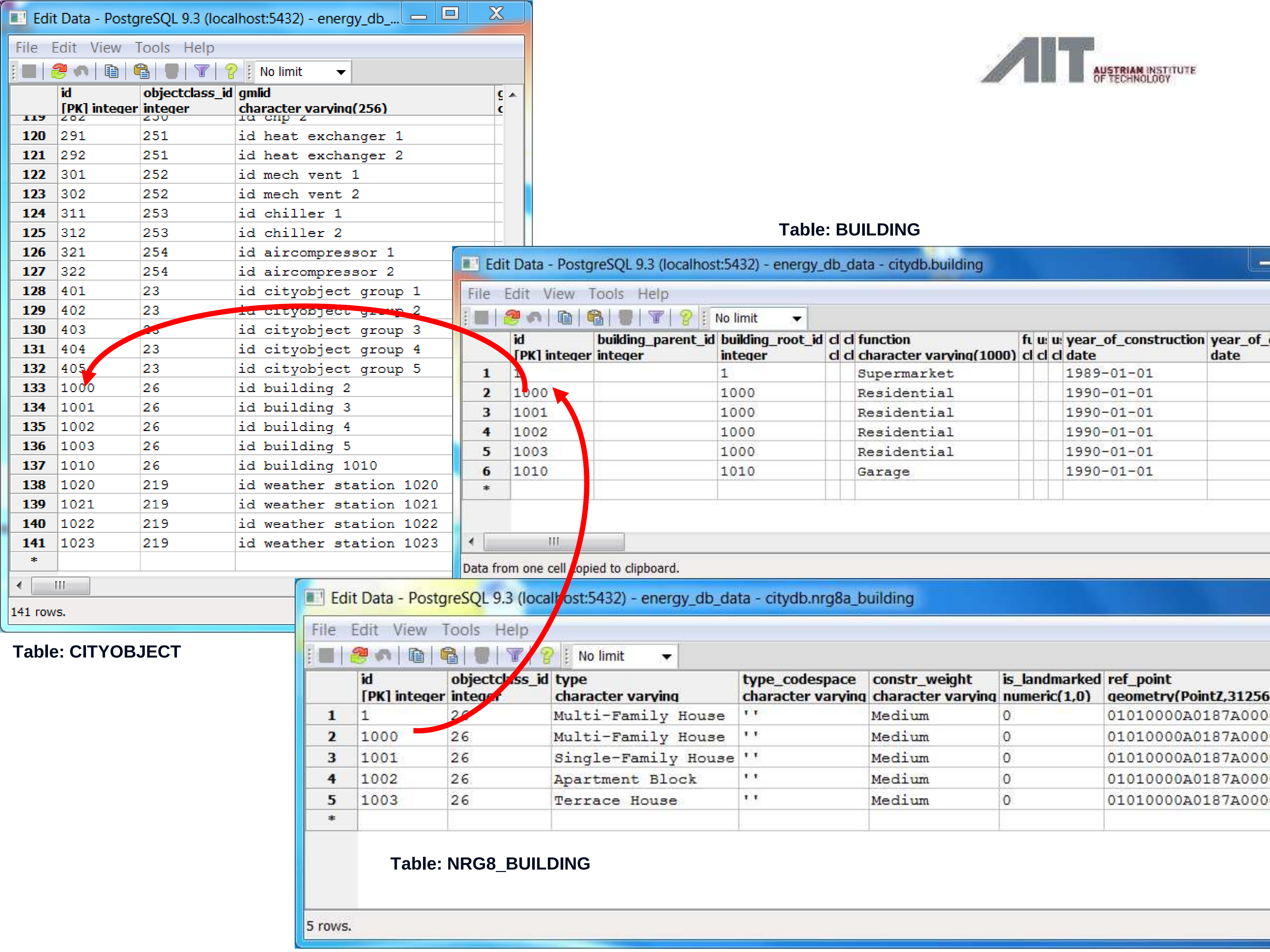
citydb.surface_geometry

citydb.nrg8_weather_data
id: serial
type: varchar
time_series_id: integer
cityobject_id: integer
gmlid: varchar
gmlid_codespace: varchar
name: varchar
name_codespace: varchar
description: text
install_point: public.geometry
nrg8_weather_data_pkey
nrg8_weather_data_ctyobj_id_fkx
nrg8_weather_data_gmlid_inx
nrg8_weather_data_ts_id_fkx
nrg8_weather_data_type_fkx

citydb.nrg8_thermal_zone
id: integer
objectclass_id: integer
building_id: integer
solid_id: integer
add_therm_bridge_uvalue: numeric

citydb.nrg8_thermal_boundary
id: integer
objectclass_id: integer
type: varchar
therm_zone1_id: integer
therm_zone2_id: integer

citydb.nrg8_thermal_opening
id: integer
objectclass_id: integer
therm_boundary_id: integer
multi_surf_id: integer
area: numeric



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)
119	282	250	id chp 2
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
*							

Table: NRG8_BUILDING

5 rows.

Interacting with the (extended) 3DCityDB

- 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
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

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');
```

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;
```

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)

```
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;
```

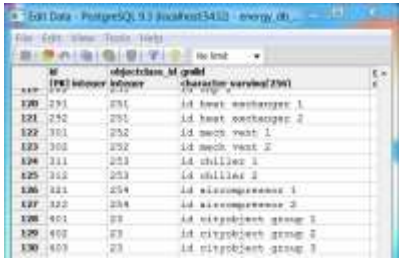
Interacting with the (extended) 3DCityDB

4. Use the updatable views (in citydb_view schema)

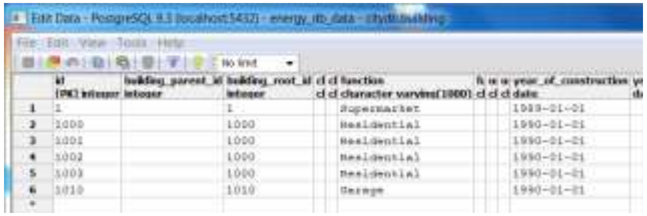
- Views hide the complexity of data stored in multiple tables by defining a *virtual* joined table which can be accessed like any other "standard" table.

- As the 3DCityDB views are built upon the "smart" insert stored procedures, the same benefits still apply.

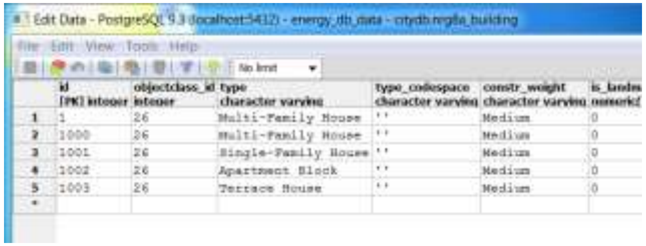
- In addition: UPDATE and DELETE operations are allowed, too.


Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_...
File Edit View Tools Help
No limit

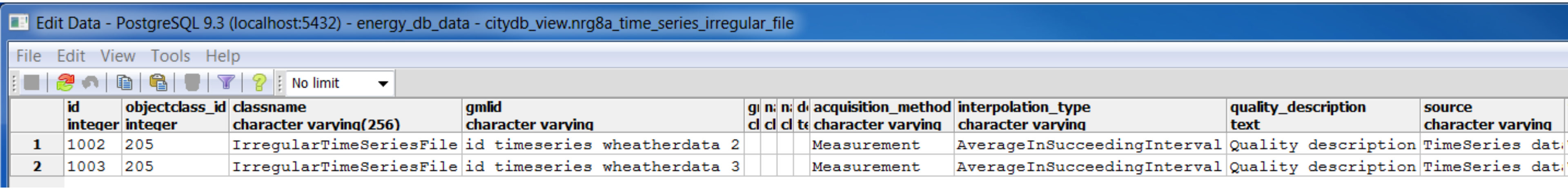
	id [PK] integer	objectclass_id integer	gmlid character varying(256)
120	291	251	id heat exchanger 1
121	292	251	id heat exchanger 2
122	301	252	id mech vent 1
123	302	252	id mech vent 2
124	311	253	id chiller 1
125	312	253	id chiller 2
126	321	254	id aircompressor 1
127	322	254	id aircompressor 2
128	401	23	id cityobject group 1
129	402	23	id cityobject group 2
130	403	23	id cityobject group 3


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)	f cl	u cl	u cl	year_of_construction date	y d
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	
*											


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_landm numeric(
1	1	26	Multi-Family House	''	Medium	0
2	1000	26	Multi-Family House	''	Medium	0
3	1001	26	Single-Family House	''	Medium	0
4	1002	26	Apartment Block	''	Medium	0
5	1003	26	Terrace House	''	Medium	0
*						


Edit Data - PostgreSQL 9.3 (localhost:5432) - energy_db_data - citydb_view.nrg8a_time_series_irregular_file
File Edit View Tools Help
No limit

	id integer	objectclass_id integer	classname character varying(256)	gmlid character varying	gi cl	n: cl	n: cl	d te	acquisition_method character varying	interpolation_type character varying	quality_description text	source character varying
1	1002	205	IrregularTimeSeriesFile	id timeseries wheatherdata 2					Measurement	AverageInSucceedingInterval	Quality description	TimeSeries dat
2	1003	205	IrregularTimeSeriesFile	id timeseries wheatherdata 3					Measurement	AverageInSucceedingInterval	Quality description	TimeSeries dat

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…

- 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

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