



Parse, persist and query HL7 CDA documents

Nov 2011 – RIMBAA meeting Amsterdam

Henk Enting <h.d.enting@mgrid.net>

Overview

Introduction

Recap

CDA parsing



About MGRID

- Spin off of Portavita in 2007
- Goal: create a storage layer specifically designed for eHealth applications
- Funding in 2009 by T-Systems Venture Fund
- Release 1.0 in October 2010



Recap



Recap

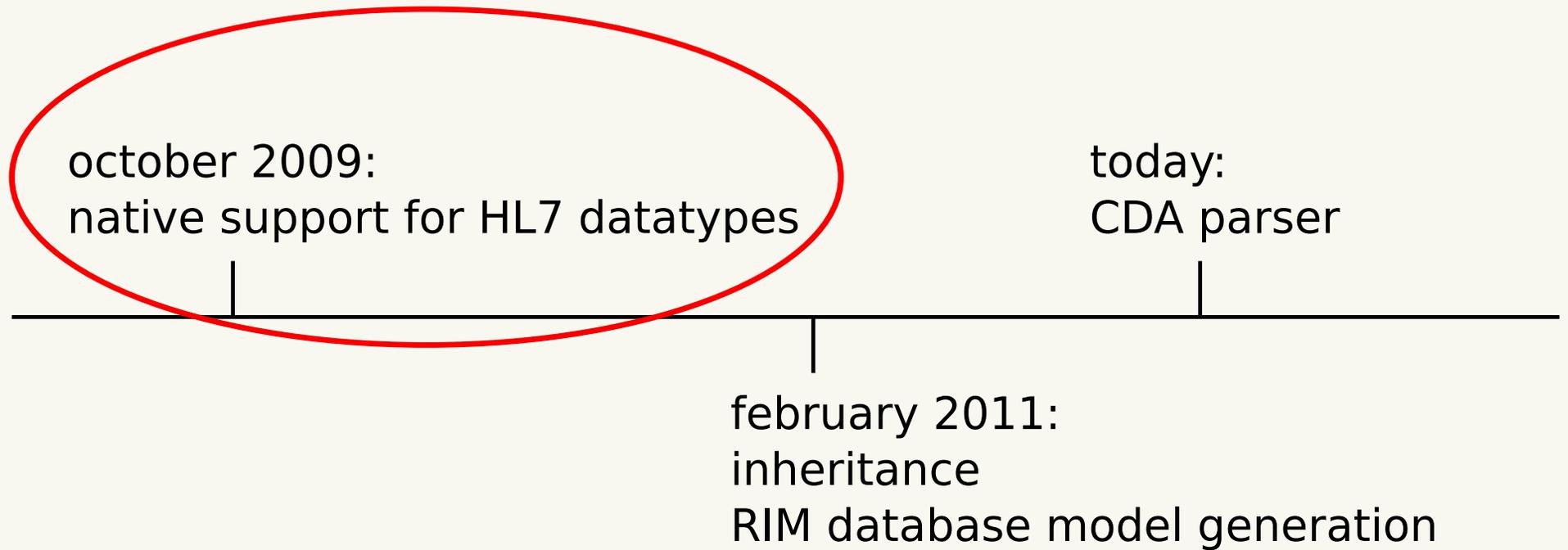
october 2009:
native support for HL7 datatypes

today:
CDA parser

february 2011:
inheritance
RIM database model generation



Recap - datatypes



Native support for ISO datatypes

- Enable precise database mapping of HL7v3 artefacts
 - no locally added constructs or workarounds, no caveats
 - knowing HL7v3 means knowing the database
- Create a query language that is powerful, fast and easy to learn
 - *powerful* query language – SQL & all PL/ languages
 - *fast*; most datatypes support indexes
 - *easy* to learn; the application programmer now has powerful, intuitive primitives



ISO datatypes: PQ

- PQs used to document observations
- Based on Unified Code for Units of Measure
 - 294 units – a.o. units from SI, ISO 1000, ISO 2955, ANSI X3.50, CGS, unified U.S. & British Imperial units
- Operations supported:
 - Comparison: $<$, $>$ and friends
 - Arithmetic: $+$, $-$, $/$, $*$, **power**
 - Aggregation: **min**, **max**, **avg**, **sum**, **var**, **stddev**
- Indexable



ISO datatypes: PQ

```
create table patient (name text, height pq, weight pq);  
CREATE TABLE
```

```
insert into patient values  
 ('Jack', '1.92 m', '92 kg')  
,('Julia', '150 cm', '50 kg')  
,('John', '188 cm', '84.3 kg')  
,('Luke', '78 cm', '11800 g');  
INSERT 0 4
```

```
create or replace function bmi(height pq, weight pq)  
returns pq  
as $$  
  select convert($2, 'kg') / convert($1, 'm')^2;  
$$ language sql immutable;  
CREATE FUNCTION
```

```
select *, bmi(height, weight) from patient where height > '1.70 m'  
order by weight;
```

name	height	weight	bmi
John	188 cm	84.3 kg	23.8512901765504753 kg/m2
Jack	1.92 m	92 kg	24.9565972222222222 kg/m2

(2 rows)



ISO datatypes: PQ

```
create table patient (name text, height pq, weight pq);
CREATE TABLE
```

```
insert into patient values
 ('Jack', '1.92 m', '92 kg')
,('Julia', '150 cm', '50 kg')
,('John', '188 cm', '84.3 kg')
,('Luke', '78 cm', '11800 g');
INSERT 0 4
```

```
create or replace function bmi(height pq, weight pq)
returns pq
as $$
  select convert($2, 'kg') / convert($1, 'm')^2;
$$ language sql immutable;
CREATE FUNCTION
```

```
select *, bmi(height, weight) from patient where height > '1.70 m'
order by weight;
 name | height | weight |          bmi
-----+-----+-----+-----
 John | 188 cm | 84.3 kg | 23.8512901765504753 kg/m2
 Jack | 1.92 m | 92 kg   | 24.9565972222222222 kg/m2
(2 rows)
```

```
/* And now for something completely different:
 * what is the mean travel time of light, from the sun to the earth?
 */
select convert(pq '1 AU' / '[c]', 's');
          convert
-----
499.0047838061356433 s
(1 row)
```



ISO datatypes: CV

- Controlled vocabularies in medical informatics
 - record information unambiguously
 - allow machine reasoning
- HL7v3 Coded Value implementation
- Support for a large number of codesystems:
 - **S**ystemized **N**omenclature of **M**edicine – **C**linical **T**erms
 - HL7v3 vocabularies all Editions
 - **L**ogical **O**bservation **I**dentifiers **N**ames and **C**odes
- Supports code systems with hierarchies
- Indexable



ISO datatypes: CV

SNOMED Clinical Terms

Ontology Layouts Nodes Arcs Help

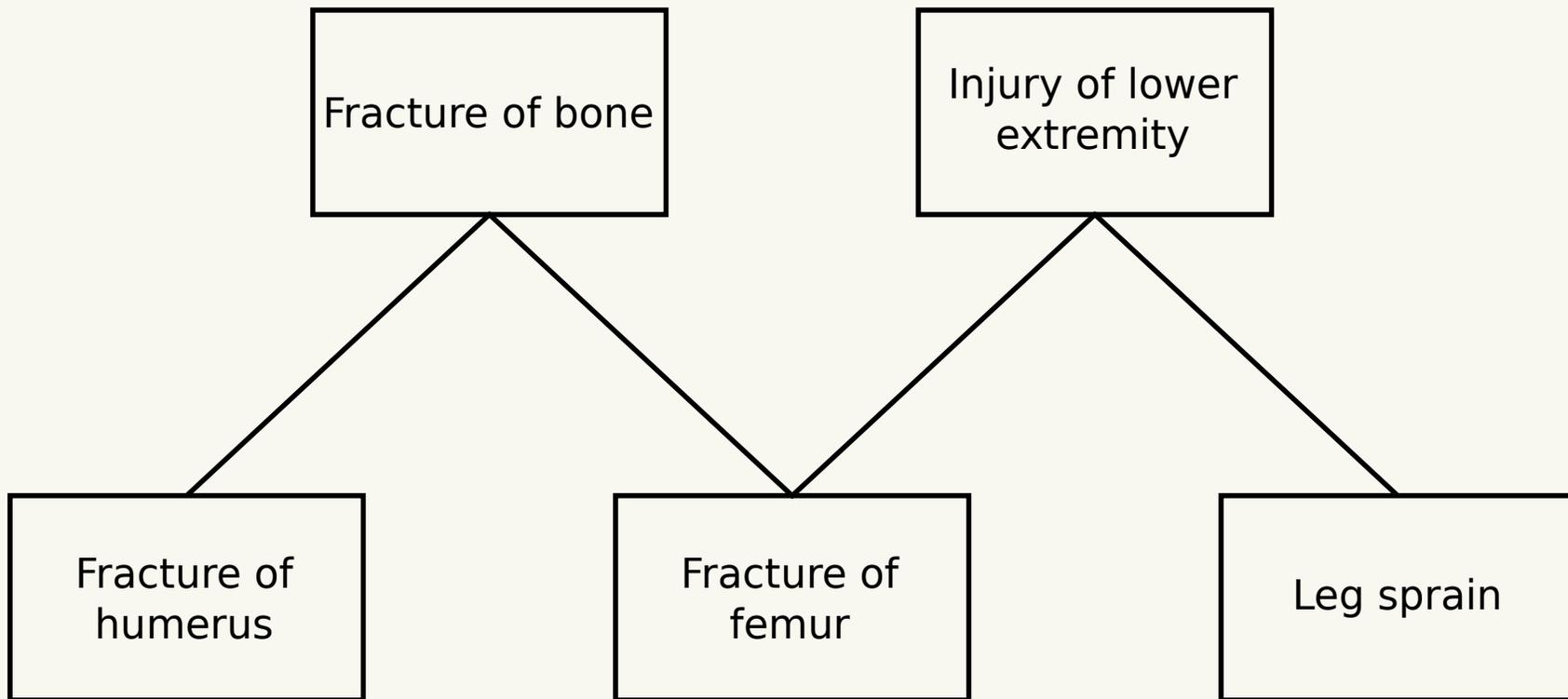
Force Directed Layout 50% Export... Show Roots

Search: Contains Search by: Name Search Found 1 matching term

Fracture of femur
ID 71620000
Children 8
CTV3ID XA0HC
ISPRIMITIVE 0
SYNONYM IS Fracture of femur, NOS, Fracture of thigh, NOS, Fracture of upper leg, NOS
SNOMEDID DD-13100
SYNONYM SY Fracture of thigh, Fracture of upper leg
CONCEPTSTATUS 0
SYNONYM FN Fracture of femur (disorder)
Synonym Fracture of thigh, Fracture of femur (disorder), F
TUI T037
UMLS_CUI C0015802
Semantic_Type Injury or Poisoning



ISO datatypes: CV – simplified



ISO datatypes: CV

```
select name, code(disorder), codesystemname(disorder),
       displayname(disorder) from observation;
```

name	code	codesystemname	displayname
Willem	71620000	SNOMED-CT	Fracture of femur
Yeb	66308002	SNOMED-CT	Fracture of humerus
Henk	262994004	SNOMED-CT	Leg sprain

(3 rows)

```
select name, displayname(disorder) from observation
       where disorder << '284003005|Fracture of bone'::cv('SNOMED-CT');
```

name	displayname
Willem	Fracture of femur
Yeb	Fracture of humerus

(2 rows)

```
select name, displayname(disorder) from observation
       where disorder << '127279002|Injury of lower extremity'::cv('SNOMED-CT');
```

name	displayname
Willem	Fracture of femur
Henk	Leg sprain

(2 rows)



Recap - Inheritance

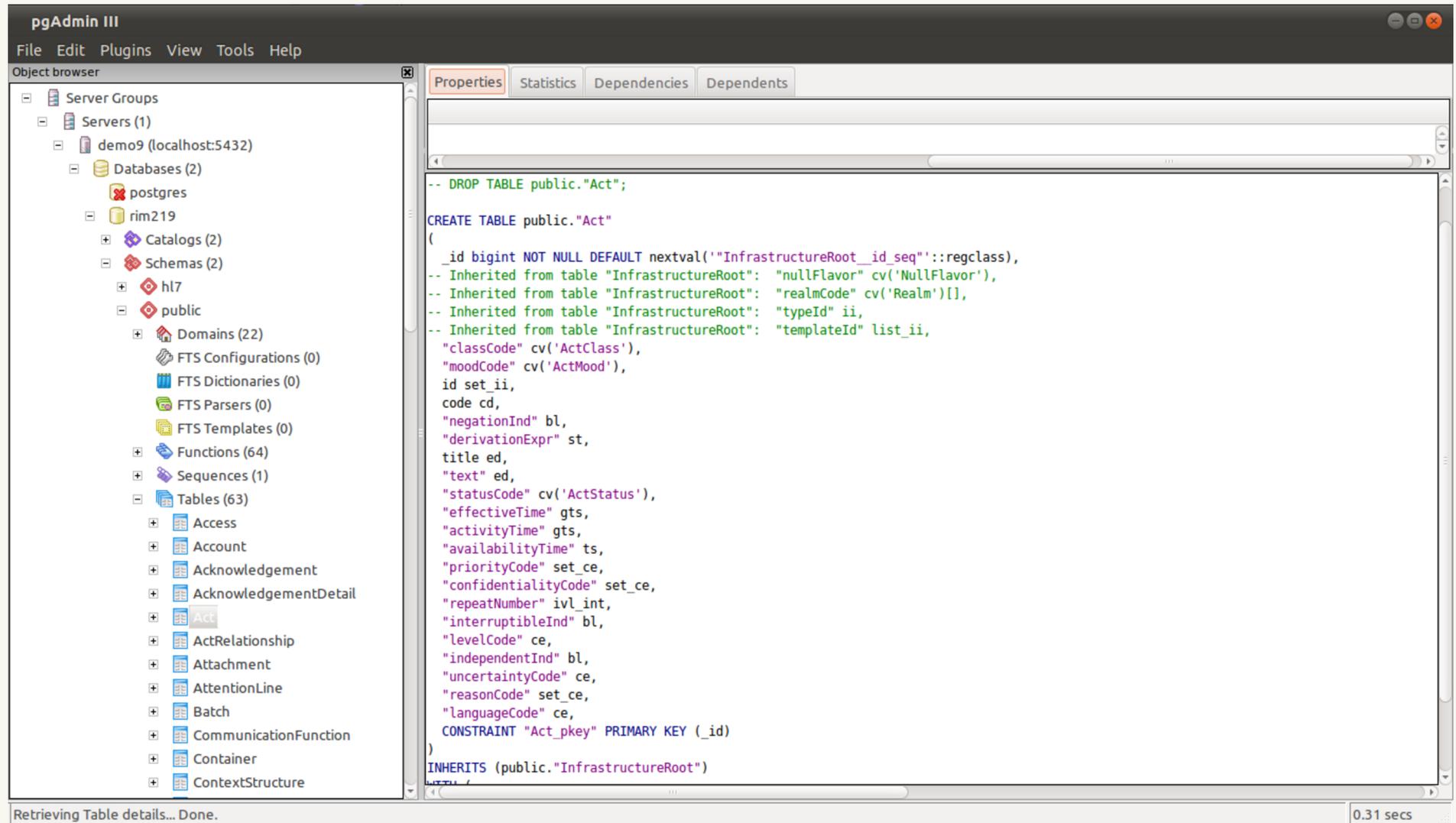
october 2009:
native support for HL7 datatypes

today:
CDA parser

february 2011:
inheritance
RIM database model generation



Inheritance - generated model



The screenshot displays the pgAdmin III interface. On the left, the 'Object browser' shows a tree view of the database structure: Server Groups > Servers (1) > demo9 (localhost:5432) > Databases (2) > rim219 > Schemas (2) > public > Tables (63). The 'Act' table is highlighted in the 'Tables' folder.

The main window shows the SQL definition for the 'Act' table, with tabs for Properties, Statistics, Dependencies, and Dependents. The SQL code is as follows:

```
-- DROP TABLE public."Act";  
  
CREATE TABLE public."Act"  
(  
  _id bigint NOT NULL DEFAULT nextval('"InfrastructureRoot_id_seq'::regclass),  
  -- Inherited from table "InfrastructureRoot": "nullFlavor" cv('NullFlavor'),  
  -- Inherited from table "InfrastructureRoot": "realmCode" cv('Realm')[],  
  -- Inherited from table "InfrastructureRoot": "typeId" ii,  
  -- Inherited from table "InfrastructureRoot": "templateId" list_ii,  
  "classCode" cv('ActClass'),  
  "moodCode" cv('ActMood'),  
  id set_ii,  
  code cd,  
  "negationInd" bl,  
  "derivationExpr" st,  
  title ed,  
  "text" ed,  
  "statusCode" cv('ActStatus'),  
  "effectiveTime" gts,  
  "activityTime" gts,  
  "availabilityTime" ts,  
  "priorityCode" set_ce,  
  "confidentialityCode" set_ce,  
  "repeatNumber" ivl_int,  
  "interruptibleInd" bl,  
  "levelCode" ce,  
  "independentInd" bl,  
  "uncertaintyCode" ce,  
  "reasonCode" set_ce,  
  "languageCode" ce,  
  CONSTRAINT "Act_pkey" PRIMARY KEY (_id)  
)  
INHERITS (public."InfrastructureRoot")
```

At the bottom of the window, a status bar indicates 'Retrieving Table details... Done.' and a timer shows '0.31 secs'.



Recap - Inheritance

october 2009:
native support for HL7 datatypes

today:
CDA parser

february 2011:
inheritance
RIM database model generation



HL7 CDA parsing

- Objective:
 - Parse HL7 CDA documents
 - Persist them in a RIM database
 - Query the database for reporting
- Focus today: parse and persist



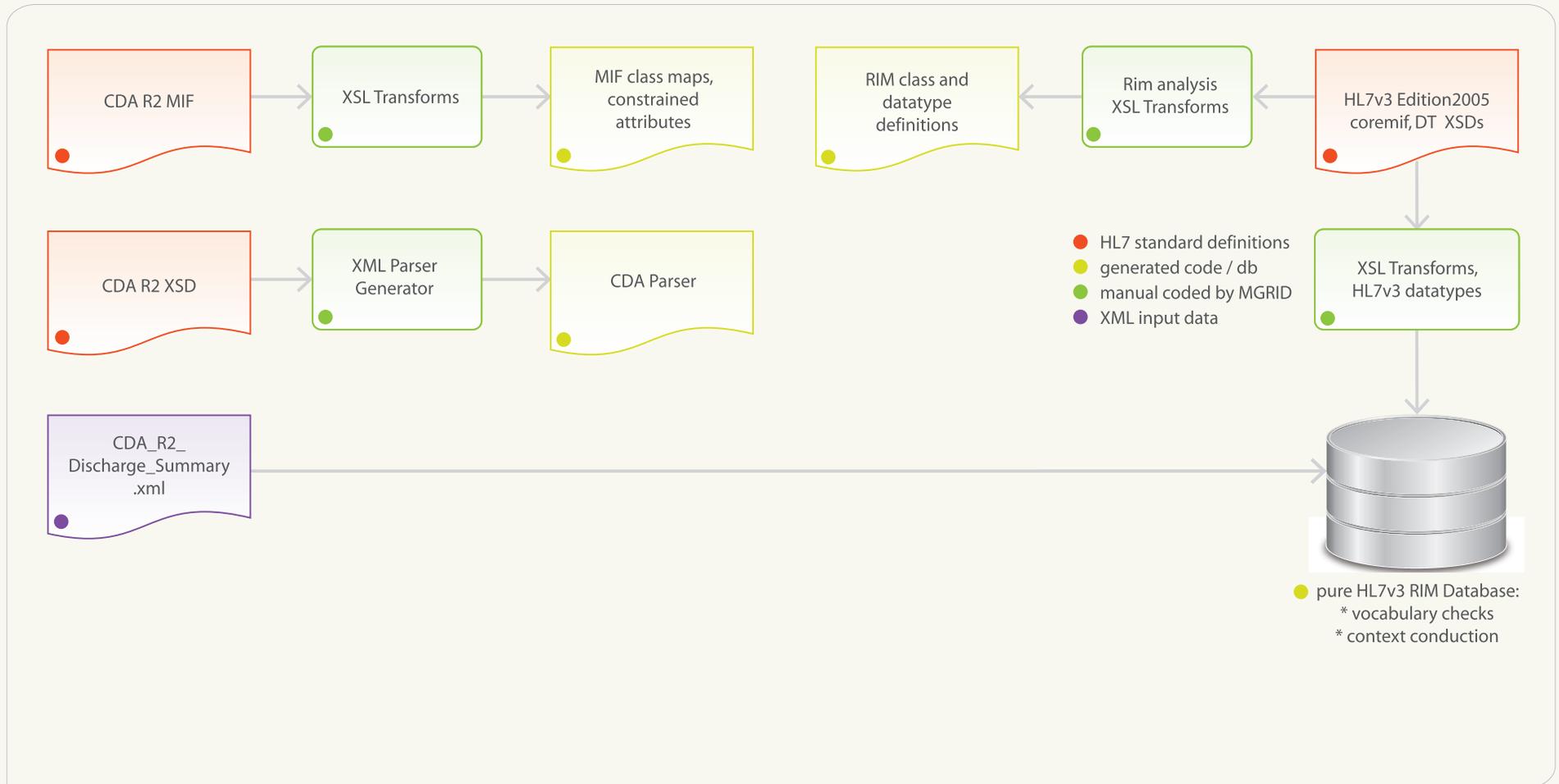
Step 1: create the database



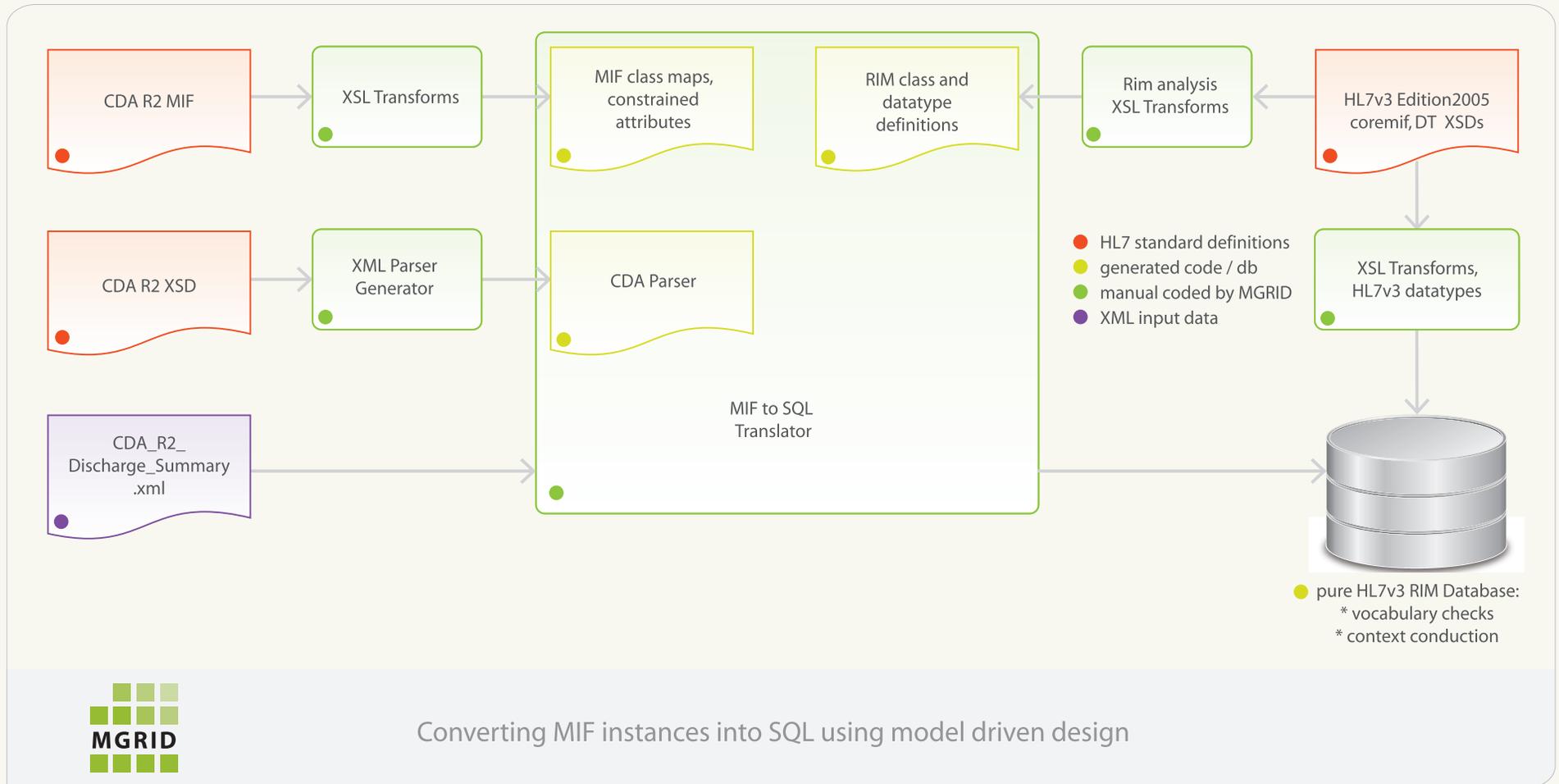
Step 2: create the parser



Step 3: determine the mappings



Step 4: combine parser with mapper



Example CDA

- CDA R2 test instances provided by Tiani SPIRIT

```
<?xml version="1.0" encoding="utf-8"?>
<ClinicalDocument xmlns="urn:hl7-org:v3">
  <typeId extension="POCD_HD000040" root="2.16.840.1.113883.1.3"/>
  <templateId root="1.3.6.1.4.1.19376.1.5.3.1.1.1"/>
  <templateId root="1.3.6.1.4.1.19376.1.5.3.1.1.2"/>
  <templateId extension="7d062328-1f36-102c-b83a-000c2915b919" root="1.2.40.0.32.6.1.10.1.1"/>
  <templateId root="1.3.6.1.4.1.19376.1.5.3.1.1.4"/>
  <id extension="20E8558A-9A0C-2A02-5611-0944F9DEE4F2" root="2.16.17.710.777.1001.902.1.1.3.2"/>
  <code code="18842-5" codeSystem="2.16.840.1.113883.6.1" codeSystemName="LOINC"
    displayName="SUMMARIZATION OF EPISODE NOTE"/>
  <title>Discharge Summarization </title>
  <effectiveTime value="20091028102753+0200"/>
  <confidentialityCode code="N" codeSystem="2.16.840.1.113883.5.25"
    codeSystemName="Confidentiality" displayName="Normal"/>
  <languageCode code="en-ZA"/>
  <setId root="2.16.17.710.777.1001.902.1.1.3.2"/>
  <versionNumber value="1"/>
  <recordTarget>
    <patientRole>
      <id extension="005.12567212651421" root="1.3.6.1.4.1.21367.2009.2.2.795"/>
      <addr>
        <country>AUSTRIA</country>
        <state>Wien</state>
        <city>Wien</city>
        <postalCode>1010</postalCode>
        <streetName/>
      </addr>
      <telecom use="HP" value="tel:+43123456789"/>
      <telecom use="WP" value="tel:+43198765432"/>
      <patient>
        <name>
```



CDA persisted in MGRID

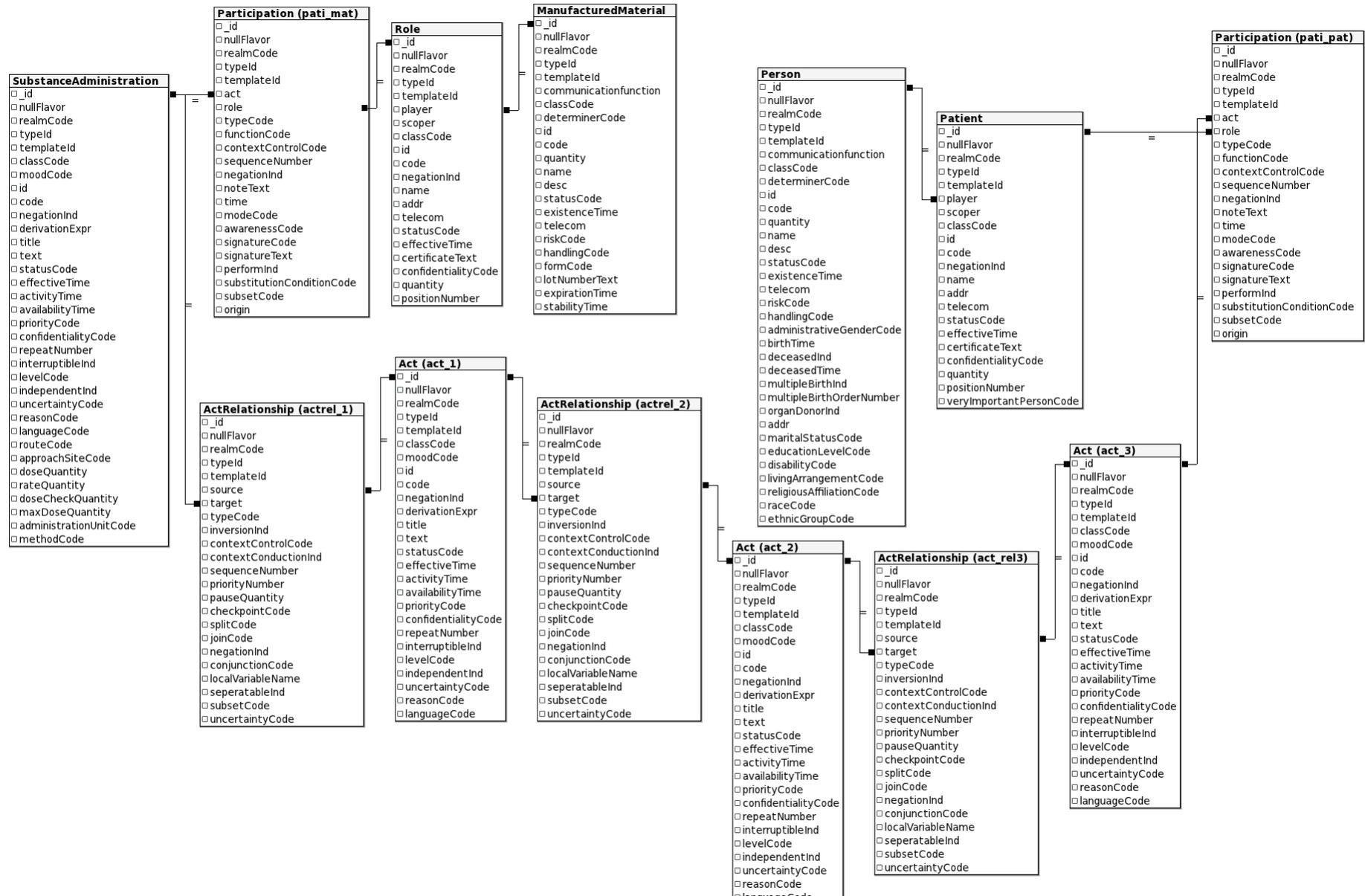
relname	reltuples
PatientEncounter	1
SubstanceAdministration	3
Organization	4
Patient	1
Observation	10
Person	4
Document	1
ManufacturedMaterial	3
ActRelationship	39
Act	25
Participation	9
Entity	1
Role	8

(13 rows)



Query on persisted CDA

- List of medication administrations per person:



Context conduction

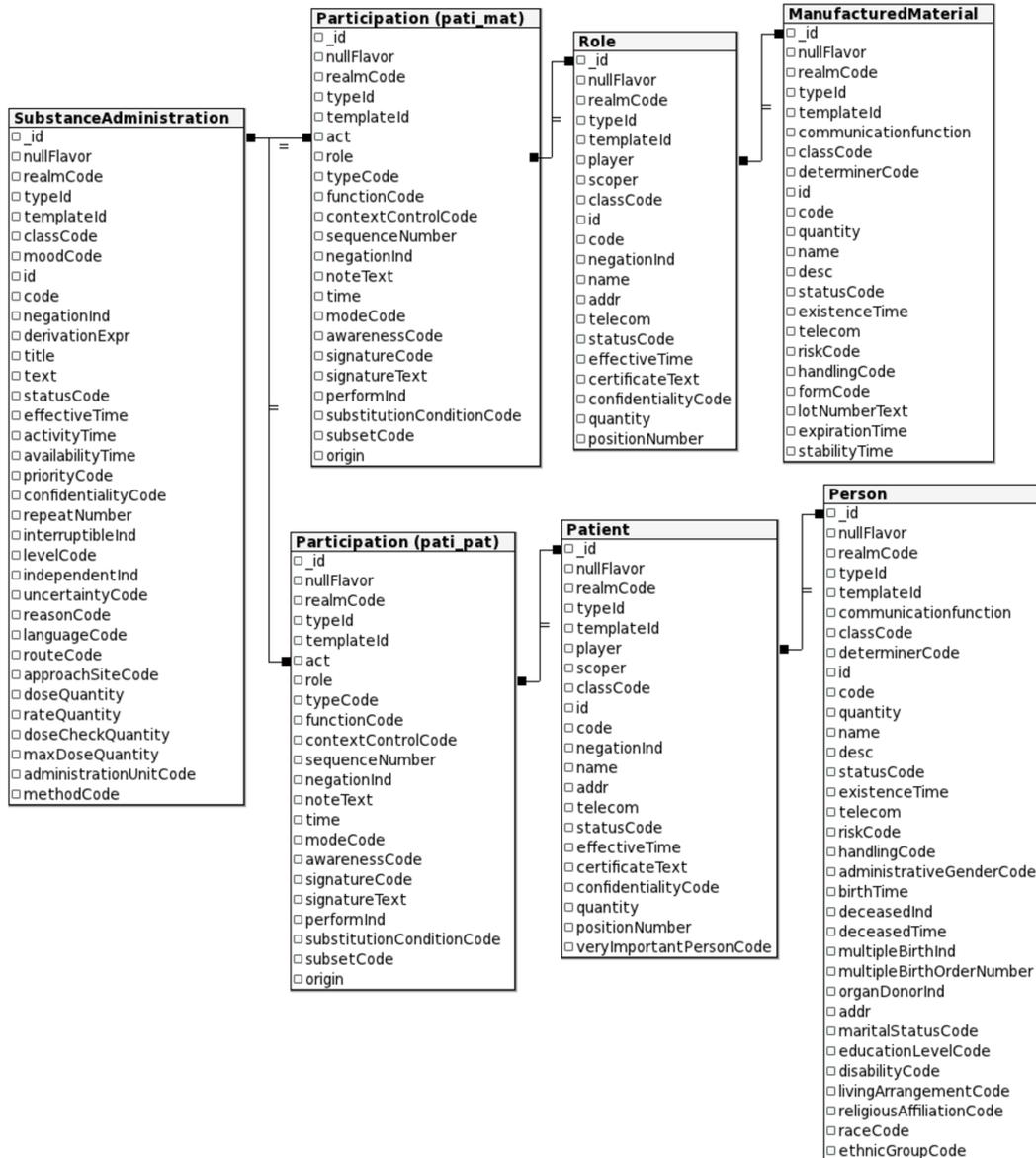
- In this case: participations
- Performed on insert time
- Less joins necessary, so
 - more concise query
 - better performance

relname	reltuples
PatientEncounter	1
SubstanceAdministration	3
Organization	4
Patient	1
Observation	10
Person	4
Document	1
ManufacturedMaterial	3
ActRelationship	39
Act	25
Participation	123
Entity	1
Role	8

(13 rows)



Query with context conduction



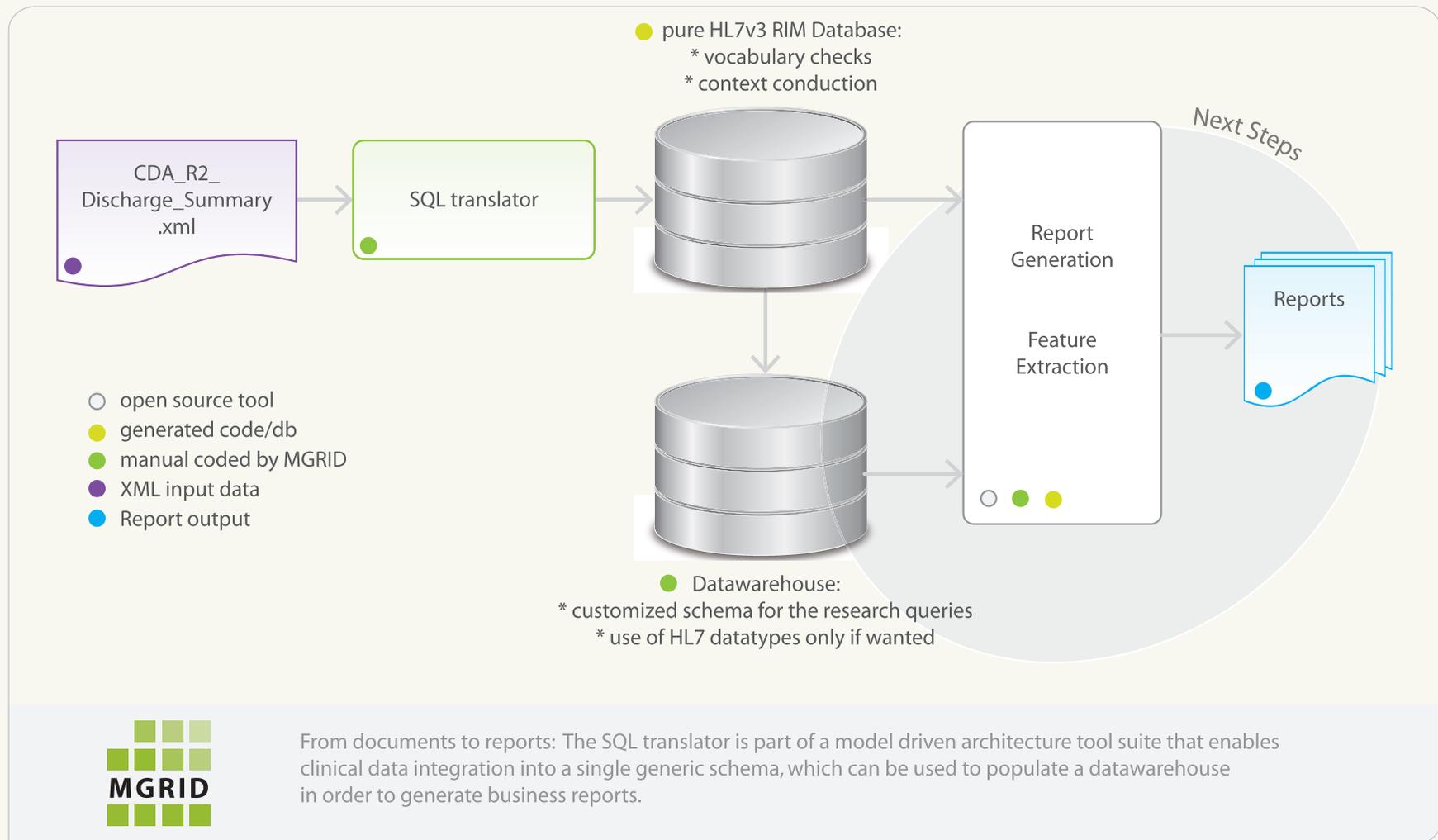
Query using implies operator

```
test=# select displayname('7947003'::CV('SNOMED-CT'));
displayname
-----
Aspirin
```

```
test=# select displayname(code) from "ManufacturedMaterial"
       where (code) << '7947003'::cv('SNOMED-CT');
displayname
-----
Aspirin 300mg soluble tablet
(1 row)
```



Next steps



From documents to reports: The SQL translator is part of a model driven architecture tool suite that enables clinical data integration into a single generic schema, which can be used to populate a datawarehouse in order to generate business reports.

Questions?

