



Mastering Medical Data with MGRID SQL

January 2013

The MGRID SQL solution

- MGRID SQL is a clinical data platform targeted at integrating, storing and retrieving large amounts of medical data
- MGRID SQL includes medical features:
 - Native support for ISO-21090 datatypes
 - Terminology support for SNOMED CT, HL7, LOINC, ...
 - HL7v3 RIM Database
 - HL7v3/CDA message parsers
- MGRID SQL enables scale-out of large databases through sharding and parallel query



The MGRID SQL solution

- MGRID SQL is a clinical data platform targeted at integrating, storing and retrieving large amounts of medical data
- MGRID SQL includes medical features:
 - **Native support for ISO-21090 datatypes**
 - Terminology support for SNOMED CT, HL7, LOINC, ...
 - HL7v3 RIM Database
 - HL7v3/CDA message parsers
- MGRID SQL enables scale-out of large databases through sharding and parallel query



Native support for ISO datatypes

- Enable precise database mapping of HL7v3 artefacts
 - no non-standard constructs, no 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

```
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: 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: Time and intervals

```
select canonical(ivl_ts '[2004;2005]' + ivl_ts '[2006;2007]') AS plus,  
       canonical(ivl_ts '[2002;2010]' - ivl_ts '[2004;2005]') AS minus;  
      plus      |      minus
```

```
-----+-----  
[2004;2005];[2006;2007] | [2002;2004[;]2005;2010]  
(1 row)
```

```
create table medication (name text, effectivetime ivl_ts);  
insert into medication values ('Pete', '[20100316;20100514]');  
insert into medication values ('Pete', '[20100420;20100701]');  
insert into medication values ('Pete', '[20101220;20110119]');  
insert into medication values ('John', '[20100516;20100614]');  
insert into medication values ('John', '[20100620;20100801]');  
insert into medication values ('John', '[20101220;20110119]');
```

```
select * from medication where effectivetime @> '20100620';  
name | effectivetime
```

```
-----+-----  
Pete | [20100420;20100701]  
John | [20100620;20100801]  
(2 rows)
```

```
select name, canonical('2010' - SUM(effectivetime)) as nomeds  
from medication  
group by name;
```

```
name | nomeds  
-----+-----  
John | [20100101;20100516[;]20100614;20100620[;]20100801;20101220[  
Pete | [20100101;20100316[;]20100701;20101220[  
(2 rows)
```



The MGRID SQL solution

- MGRID SQL is a clinical data platform targeted at integrating, storing and retrieving large amounts of medical data
- MGRID SQL includes medical features:
 - Native support for ISO-21090 datatypes
 - **Terminology support for SNOMED CT, HL7, LOINC, ...**
 - HL7v3 RIM Database
 - HL7v3/CDA message parsers
- MGRID SQL enables scale-out of large databases through sharding and parallel query

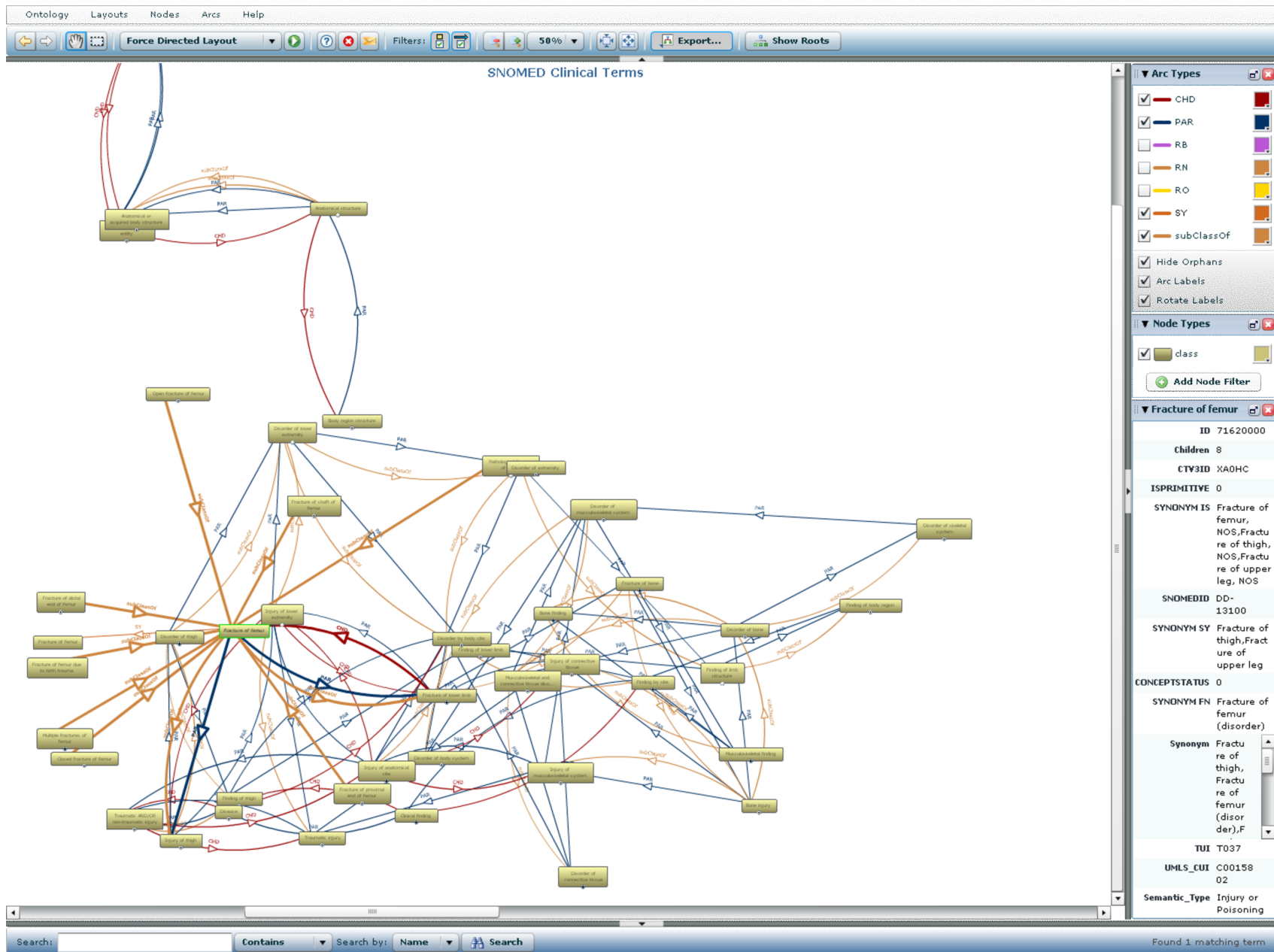


Terminology Support

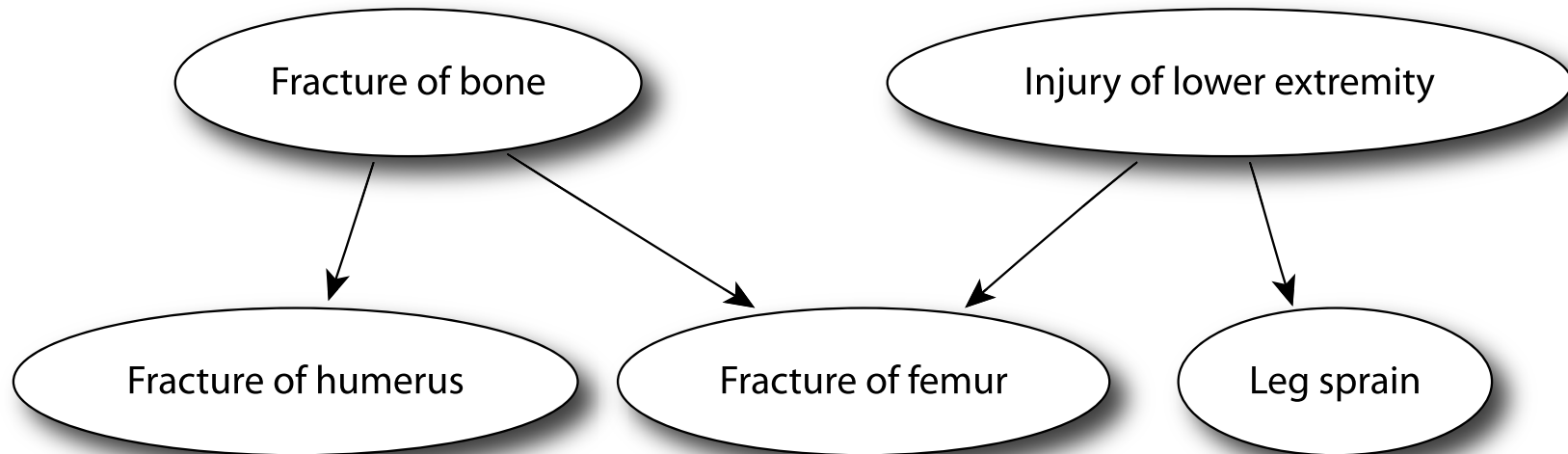
- Controlled vocabularies in medical informatics
 - record information unambiguously
 - allow codesystem based information retrieval
- ISO datatype Coded Value (CV) implementation
 - Supports code systems with hierarchies
 - Indexable
- Support for a large number of codesystems:
 - ~300 HL7 codesystems per edition
 - SNOMED-CT
 - LOINC



Terminology support: SNOMED CT



Terminology support: SNOMED CT



Terminology support: SNOMED CT

```
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)



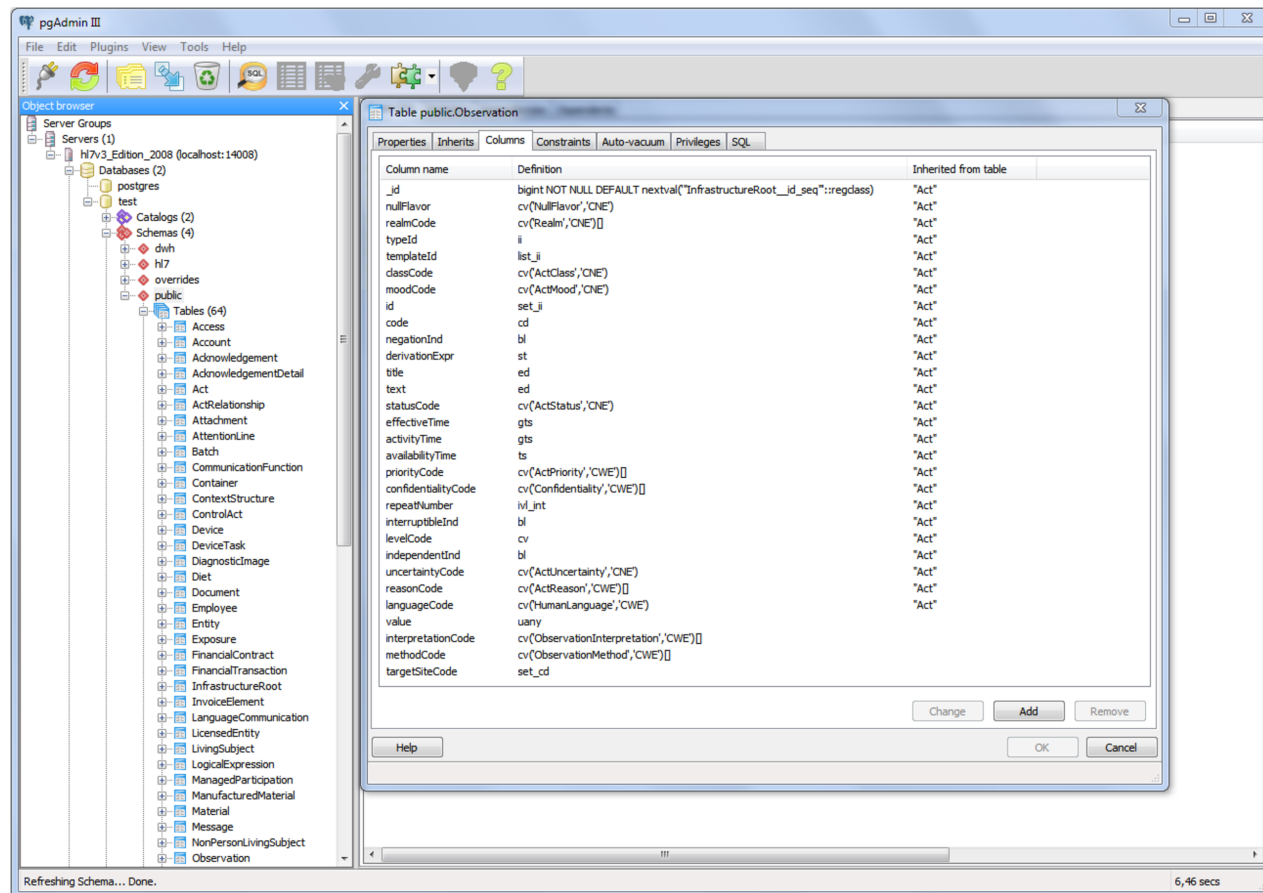
The MGRID SQL solution

- MGRID SQL is a clinical data platform targeted at integrating, storing and retrieving large amounts of medical data
- MGRID SQL includes medical features:
 - Native support for ISO-21090 datatypes
 - Terminology support for SNOMED CT, HL7, LOINC, ...
 - **HL7v3 RIM Database**
 - HL7v3/CDA message parsers
- MGRID SQL enables scale-out of large databases through sharding and parallel query



HL7v3 RIM Database

- MGRID SQL RIM = HL7v3 RIM
- Database structure and inheritance *generated* straight from the specification

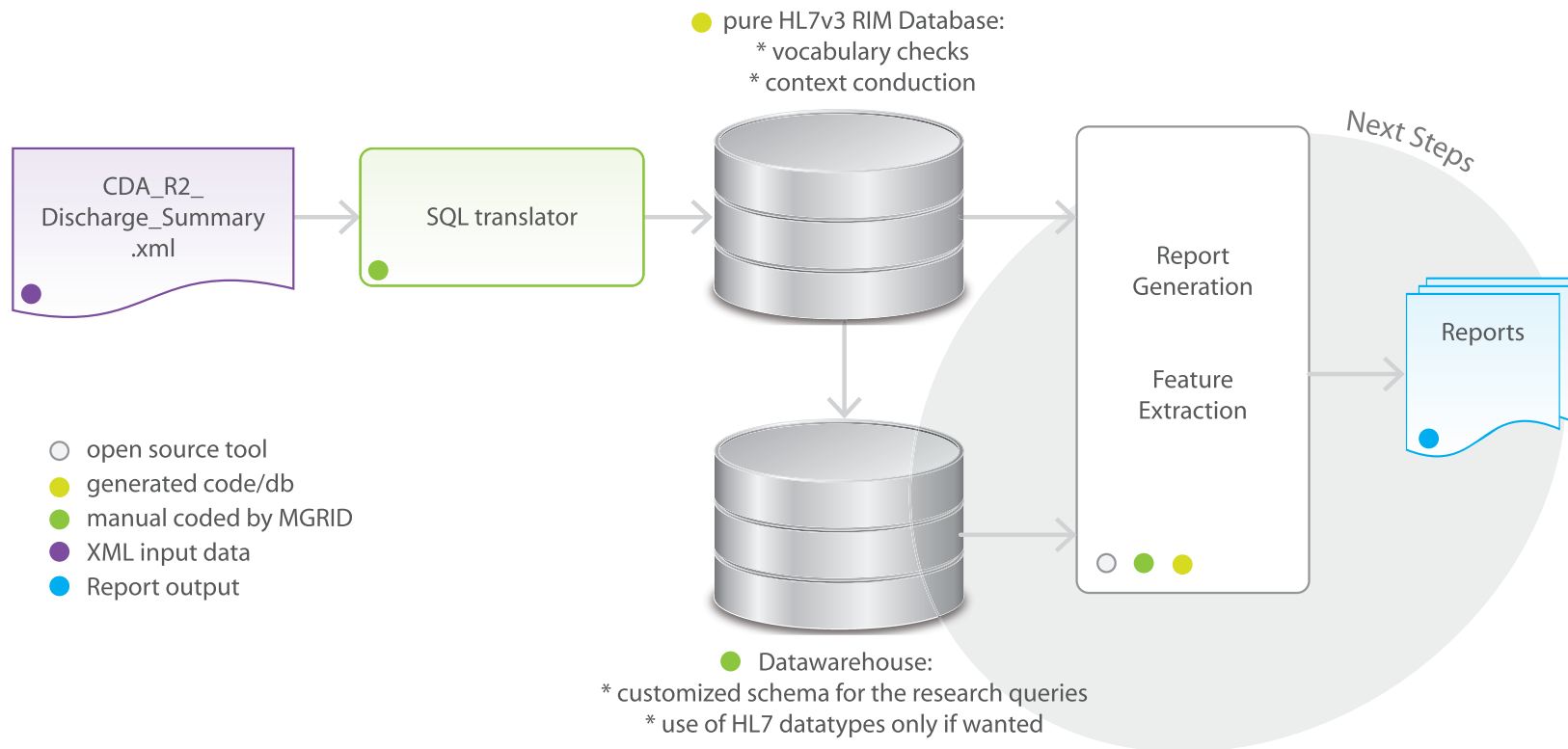


The MGRID SQL solution

- MGRID SQL is a clinical data platform targeted at integrating, storing and retrieving large amounts of medical data
- MGRID SQL includes medical features:
 - Native support for ISO-21090 datatypes
 - Terminology support for SNOMED CT, HL7, LOINC, ...
 - HL7v3 RIM Database
 - **HL7v3/CDA message parsers**
- MGRID SQL enables high performance and scalability through sharding and parallel query

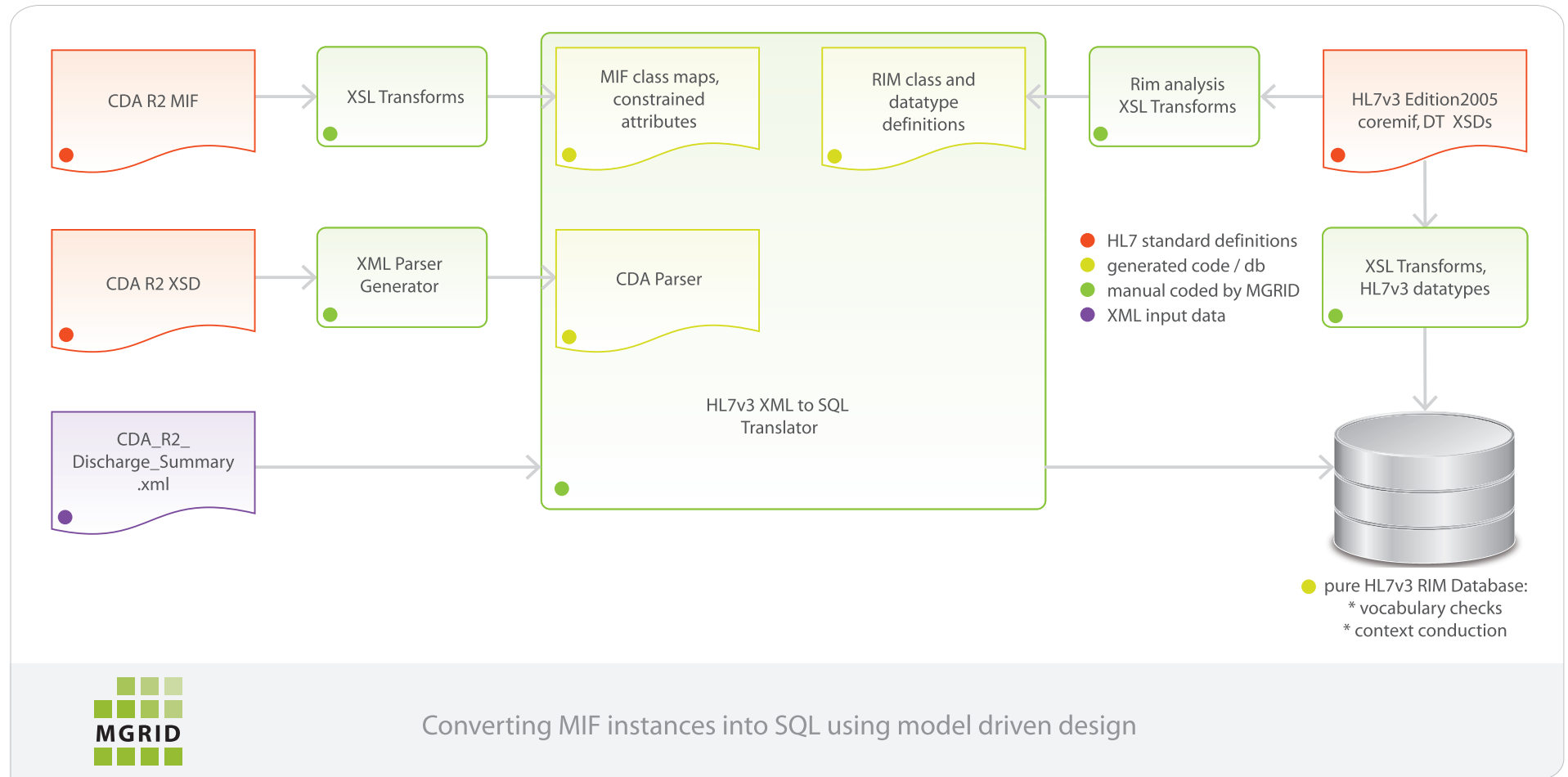


From documents to reports



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.

Message parsers using MDA



Model Driven Architecture

- MGRID SQL includes generators for RIM Databases and Message Parsers
- All HL7v3 Normative Editions supported
- Keep up with HL7v3's rapid pace of change

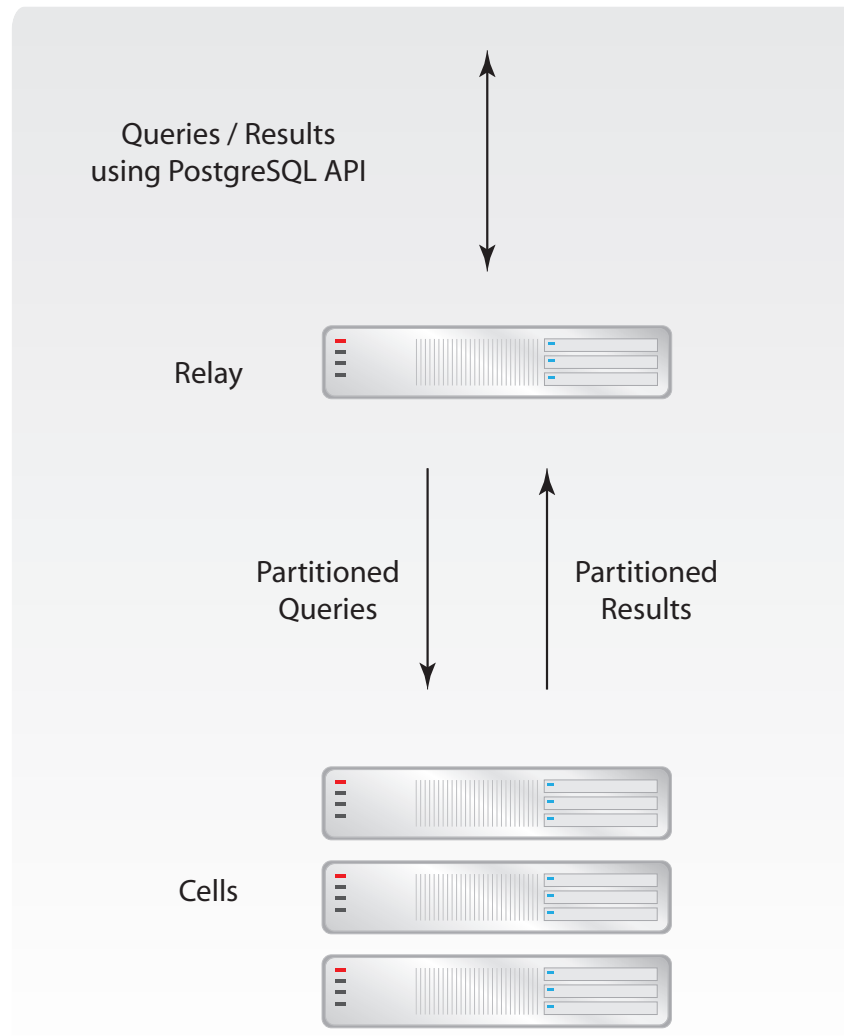


The MGRID SQL solution

- MGRID SQL is a clinical data platform targeted at integrating, storing and retrieving large amounts of medical data
- MGRID SQL includes medical features:
 - Native support for ISO-21090 datatypes
 - Terminology support for SNOMED CT, HL7, LOINC, ...
 - HL7v3 RIM Database
 - HL7v3/CDA message parsers
- **MGRID SQL enables scale-out of large databases through sharding and parallel query**



Sharding and parallel query



- **layout** defines distribution
 - of tables
 - on cells
 - via attributes
 - using a degree of parallelism (dop)
- **relay** grid gateway
 - provides a standard PostgreSQL interface for clients
 - plans distributed queries
 - combines grid results
- **cells** hold partitioned data
- **redundancy group**
 - one complete copy of the data

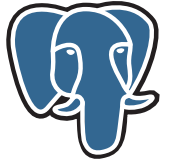


Sharding and parallel query

- Data distribution is transparent to the application and feature compatible with the latest standard PostgreSQL.
- MGRID SQL includes an optimized distribution method for RIM databases, that ensures that persisted RIM documents do not span multiple partitions.
- MGRID SQL sharding will speed up OLAP / BI style workloads.



MGRID SQL is based on PostgreSQL



- 15 years of active development, runs on all major platforms
- Proven architecture, with a strong reputation for reliability, data integrity and correctness
- Sophisticated enterprise class features; e.g. hot backups, (a)synchronous replication, nested transactions and foreign data wrappers.
- Prominent users: Reddit, Skype, OpenStreetMap, International Space Station



MGRID SQL Interface options

- Use common PostgreSQL connection libraries to connect to MGRID, e.g.:
 - Java: JDBC
 - Python: psycopg2
 - .NET: Npgsql
 - C++: libpqxx
 - ODBC: psqLODBC
 - Perl: DBD:pg



More Information

- website: www.mgrid.net
- email: info@mgrid.net
- Watch our blog for upcoming features in 2013 on www.mgrid.net/blog

