

D2RQ

Richard Cyganiak

D2RQ

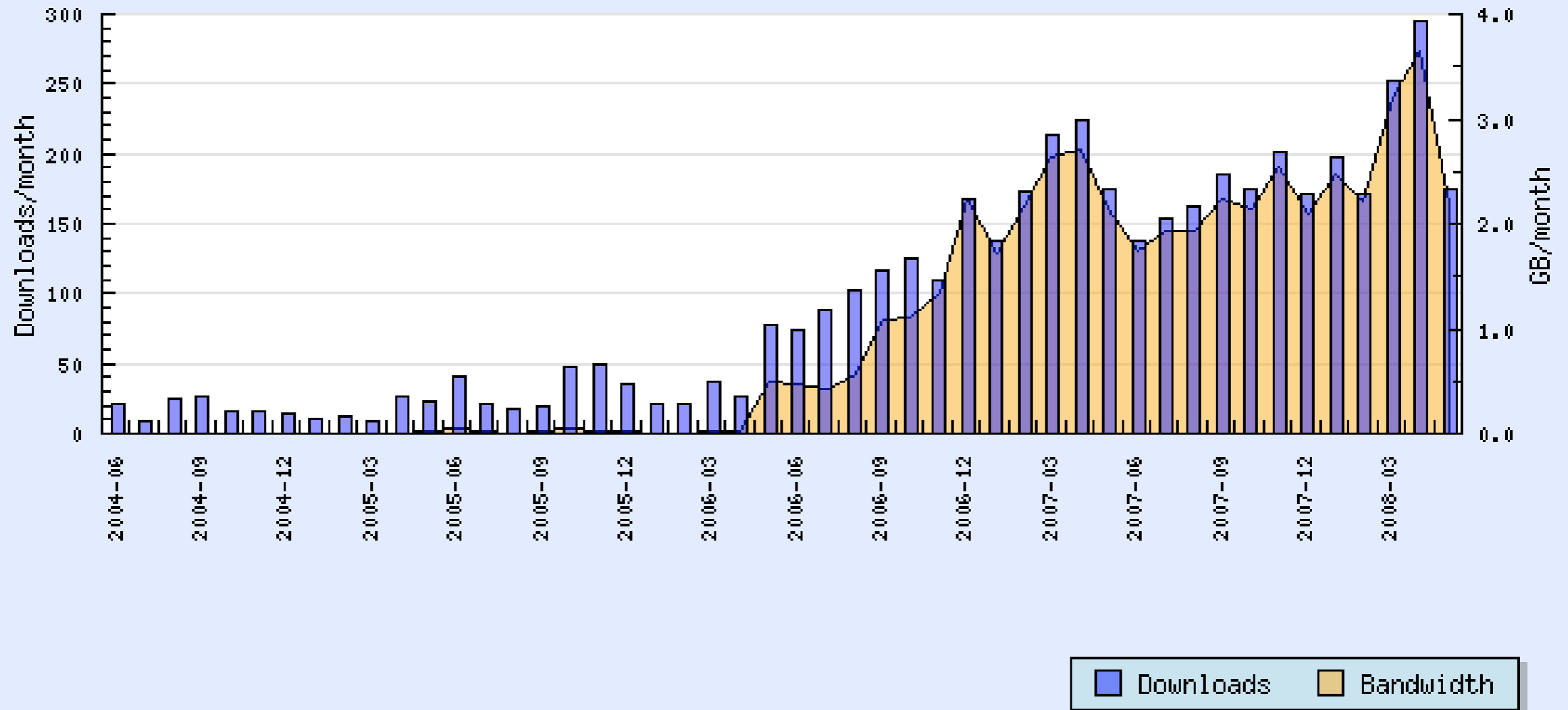
- DB-to-RDF mapper written in Java
- In: any JDBC database
- Middle: flexible, N3-based mapping language
- Out: SPARQL, Linked Data, or Jena API
- Popular, easy to get started
- SPARQL-to-SQL algorithm not state of the art

The project

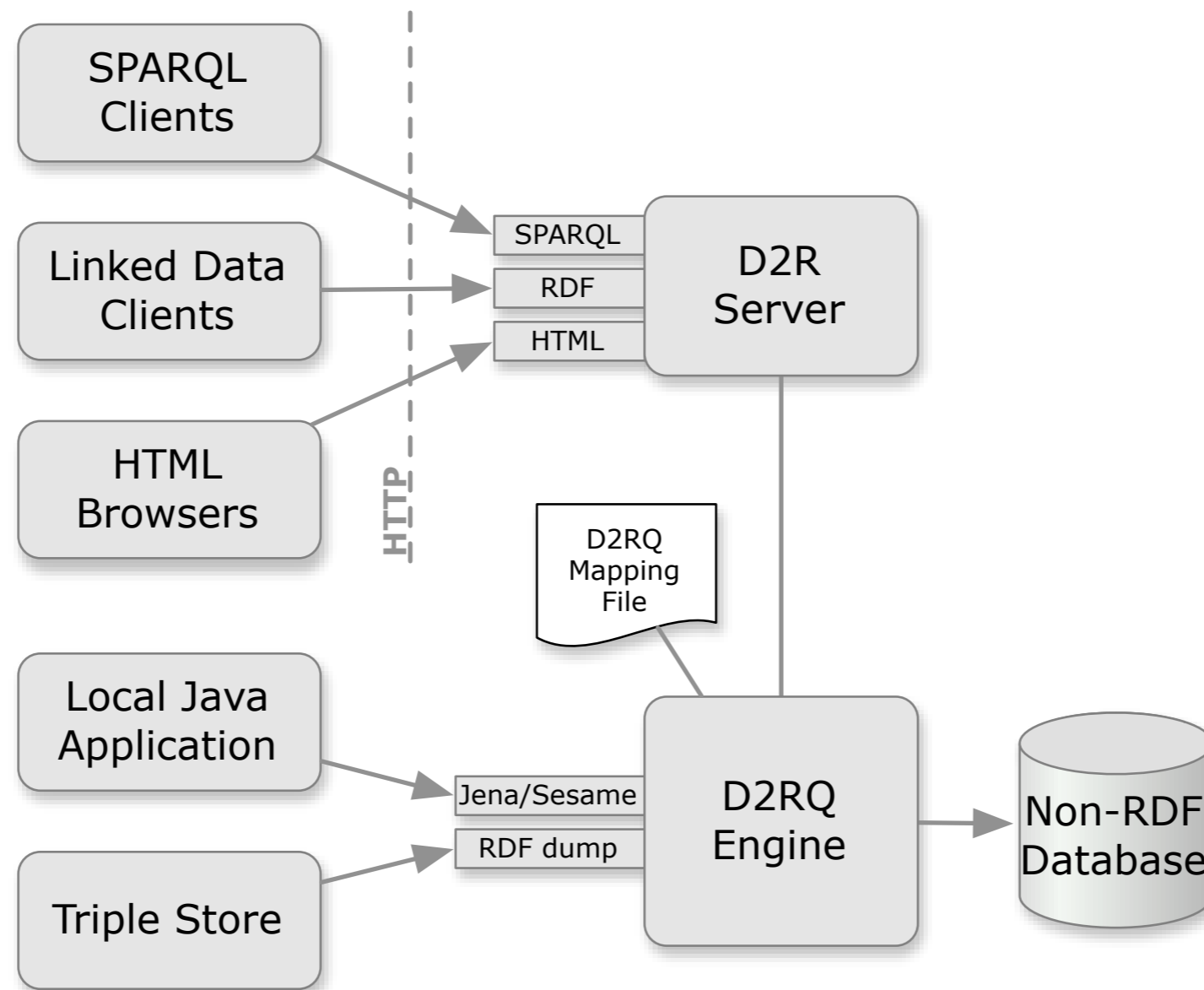
- Started 2004 (roots: 2002) by Chris Bizer at FU Berlin; later me at FU and HP Labs
- 200+ downloads/month, 4600+ total
- mailing list at ~20 msgs/month, 700+ total
- In LOD cloud, TopBraid Composer etc

Download History For D2RQ and D2R Server

All Time



Architecture



Architecture (2)

- maps DB to *virtual* RDF graph
- easy to offer arbitrary interfaces to the RDF graph
- most requested: SPARQL and RDF dumps

Mapping language

Mapping language

- N3 based syntax
- Very flexible
- Language is not easy, wish we had a GUI
- Usually auto-generate mapping from DB schema, then customize
- Defined in D2RQ manual

Flexible mappings!

- Properties of one class from multiple tables
- Several classes in the same table
- Value translations, SQL expressions
- Arbitrary joins and SQL conditions
- *This is a MUST HAVE!* Users need it

To SQL or not to SQL

- Users want to deal with complexity by using their SQL knowledge
- They want to write arbitrary SQL queries
- We don't want to parse SQL (*painful!*)
- We force users to decompose their query into small fragments

Mapping process

1. Define your entities
2. Add properties to entites
3. Link entities together
4. Get fancy with conditions, joins, value translations

I. Define your entities

```
map:People a d2rq:ClassMap;  
  d2rq:uriPattern "http://.../people/@@User.ID@@";  
  d2rq:class foaf:Person;  
  d2rq:condition "User.deleted=0".
```

2. Add properties

```
map:name a d2rq:PropertyBridge;  
  d2rq:column "User.name";  
  d2rq:property foaf:nick .
```

```
map:mbox a d2rq:PropertyBridge;  
  d2rq:uriPattern "mailto:@@User.email@@";  
  d2rq:property foaf:mbox .
```

3. Link your entities

```
map:photo a d2rq:PropertyBridge;  
d2rq:refersToClassMap map:Photos;  
d2rq:property foaf:made;  
d2rq:join "User.ID = Photo.UserID".
```

(also d2rq:alias for self-joins)

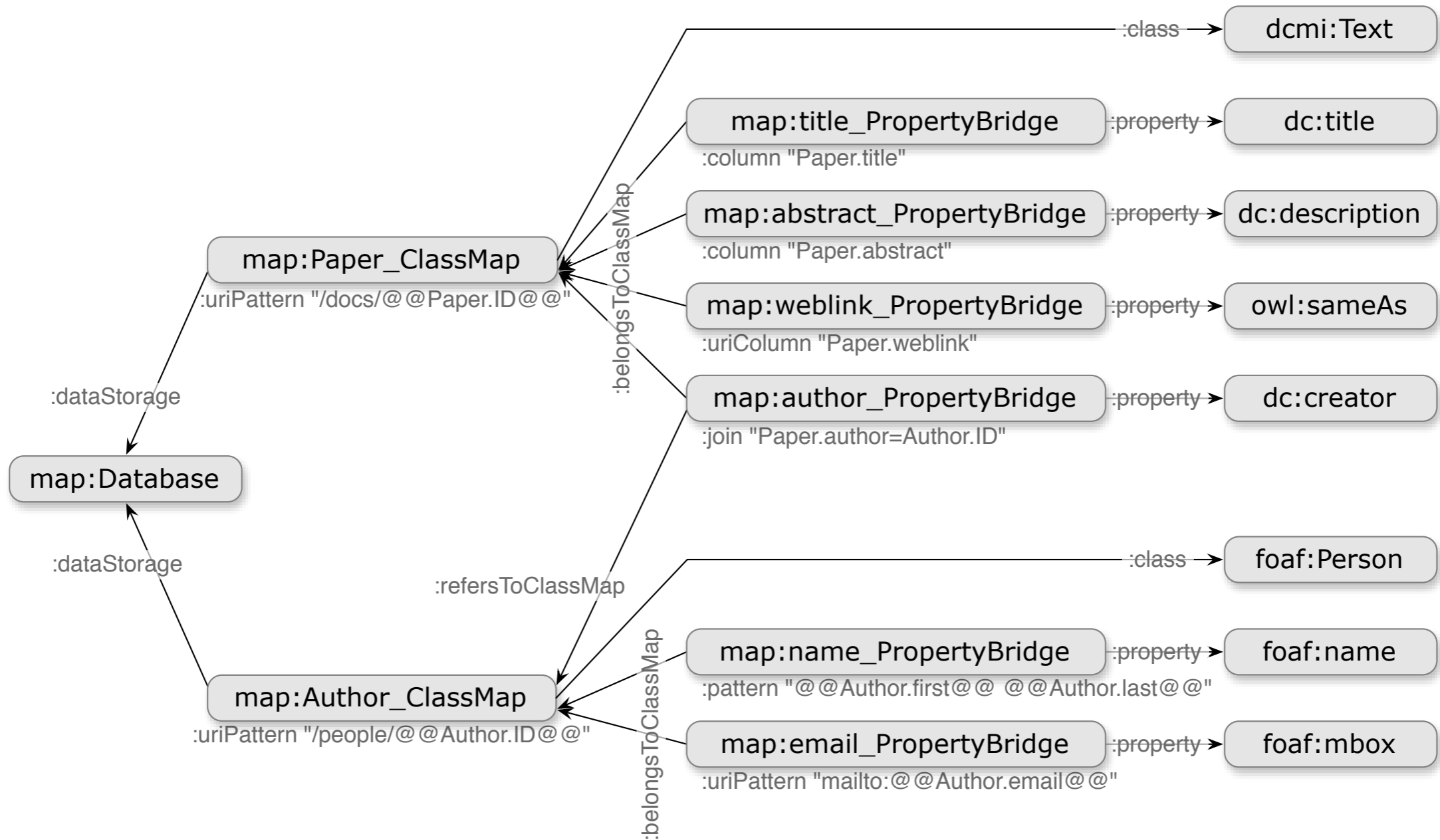
4. Get fancy

```
map:mbox_sha1 a d2rq:PropertyBridge;  
d2rq:sqlExpression  
    "SHA1(CONCAT('mailto:', User.email))";  
d2rq:property foaf:mbox_sha1sum .
```

Terminology warning!

- A ClassMap doesn't necessarily correspond to an RDFS/OWL class
- Neither necessarily to a single table
- Rather: A set of entities/resources generated in the same way from the DB

Mapping file overview



SQL composition

- Take one or more PropertyBridges with its ClassMap
- **SELECT** *columns/expressions*
FROM *all mentioned tables*
WHERE *joins*
AND *conditions*
- (and deal with aliases)

Mapping language warts

- Some suboptimal terminology (ClassMap)
- Remnants from pre-SPARQL days (constraints)
- Remnants from early lack of JDBC schema introspection (column types)

Still missing

- Determine property of a PropertyBridge based on type code
- SQL subqueries in WHERE or FROM

Summary

- D2RQ: four years, 4600+ downloads
- Powerful mapping language is important
- To SQL or not to SQL?
- In the future there will be GUIs