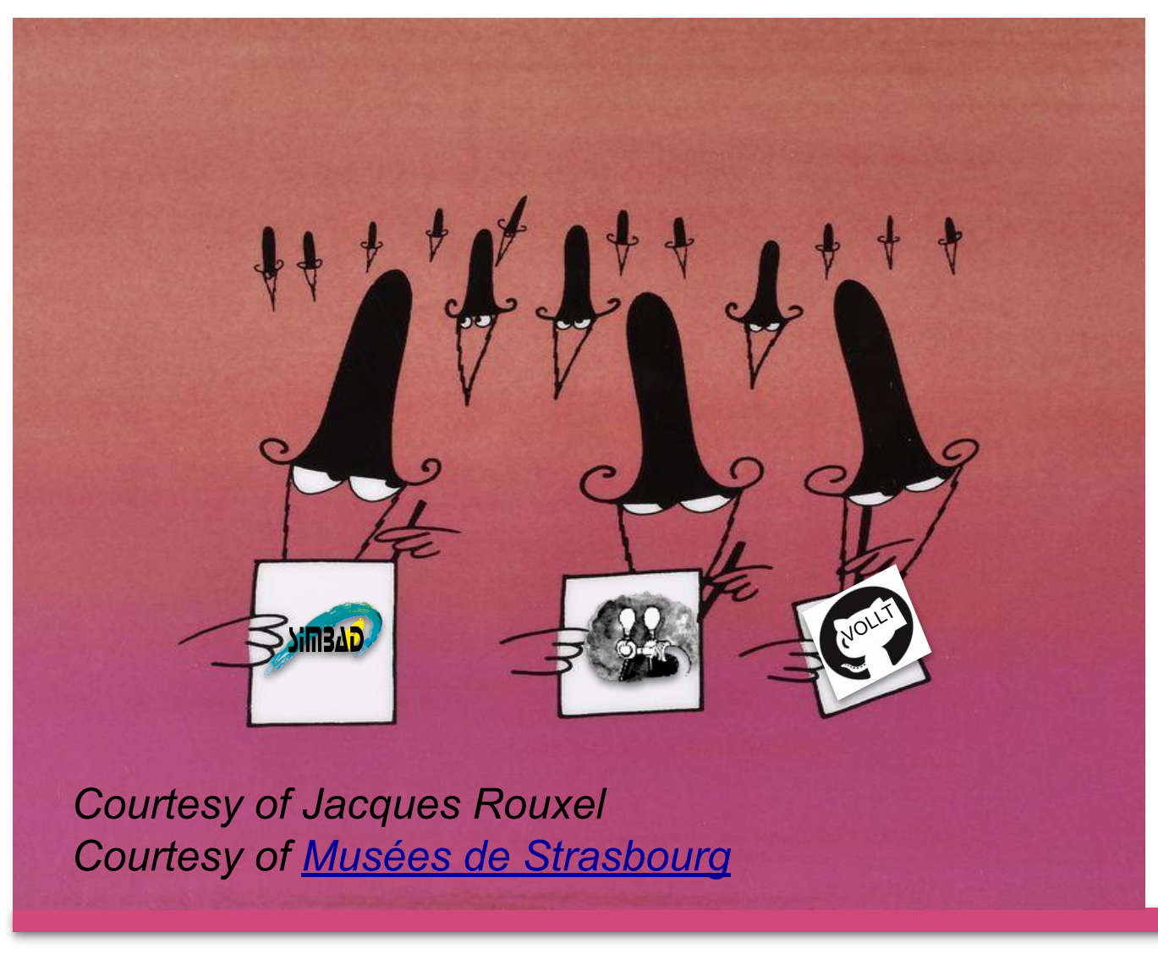
Retrieving Complex Data in TAP Services

{ [Anais.Oberto , Laurent.Michel , Gregory.Mantelet] , @astro.unistra.fr }, { Haoyun LIAO }

Université de Strasbourg, CNRS, Observatoire astronomique de Strasbourg, UMR 7550, F-67000 Strasbourg, France



Courtesy of Jacques Rouxel
Courtesy of Musées de Strasbourg

We are working on a client module for TAP Web interfaces . The database schema, namely the TAP_SCHEMA, is analysed to build a JSON representation of complex data that can be annotated with query constraints set by the user. So that complex queries can be easily setup and run. Query results are parsed out to reconstruct the data structure in JSON strings, where top level data are explicitly listed and subcomponents can be fetched with appropriate synchronous TAP queries. These queries are generated by the parser and included within the JSON output. Thus, users can unfold on demand any part of the searched data. This mechanism is being implemented in Javascript to be proposed as a new overlay for both SIMBAD and TAPHandle interfaces.

SIMBAD is a database with +25 tables and support ADQL queries.

TapHandle is a generic Web interface for TAP services.

VOLLT is a Java library to expose a database through TAP protocol

Standard TAP interface

Flat representation of the TAP tables

Schema: `data`

- **basic** (OID, main_id, ra, dec)
- **identifier** (OIDREF, id)
- **has_ref** (OIDREF, OIDBIBREF)
- **ref**: (OIDBIB, bibcode)

Hand-written ADQL Query

```
-- Gets objects and bibcodes around a
'Antares' and having a publication from
'OBERTO'
SELECT main_id, bibcode
FROM
    (SELECT ra,dec FROM basic JOIN Ident ON
oid=oidref WHERE id='ANTARES') AS antares,
    basic AS star
    JOIN has_ref ON oid=oidref
    JOIN ref ON oidbibref=oidbib
    JOIN author USING(oidbibref)
WHERE CONTAINS(POINT('ICRS', star.ra,
star.dec), CIRCLE('ICRS', antares.ra,
antares.dec, 0.1)) = 1
AND author.name LIKE 'OBERTO%'
```

Result table

main_id	bibcode
* alf Sco B	1992BICDS..40...71P
* alf Sco B	1976A&A....46...11A
* alf Sco	1995ApJ...440L..93B
* alf Sco	2005A&A...431..773R
* alf Sco	2002AJ....124.1636K
* alf Sco	1999ApJ...516..817K
* alf Sco	1999ApJS..122..221M
* alf Sco	1994A&A...281..161A
* alf Sco	1994BAAS...26.1455D
* alf Sco	1992BICDS..40...71P
* alf Sco	1984ApJ...276L..21B

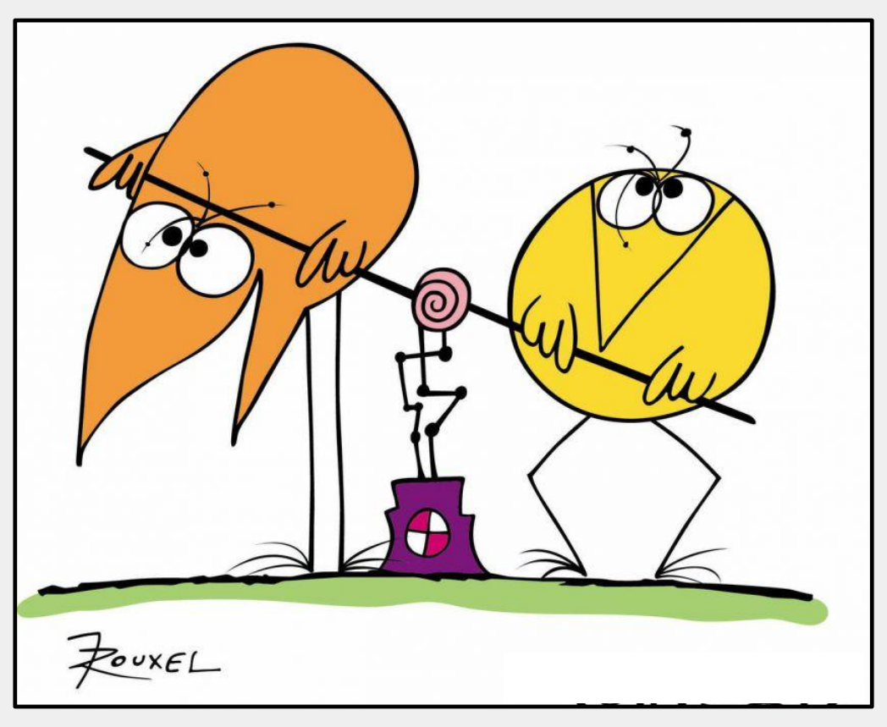
In a standard TAP service, users have to:

- Work with a flat representation of the database schema (TAP_SCHEMA tables).
- Deal with cryptic information about the relationships between tables.
- Build ADQL queries "by hand", including joins.
- Get **denormalized** tables.

```
TAP REQUEST TO http://.../tap/sync
---
REQUEST = doQuery
LANG    = ADQL
FORMAT  = VOTable
QUERY   = SELECT main_id FROM ...
```

Flat table

```
   main_id     |        bibcode
---------------|-----------------------
"* alf Sco B"|"1992BICDS..40...71P"
"* alf Sco B"|"1976A&A....46...11A"
"* alf Sco"  |"1995ApJ...440L..93B"
"* alf Sco"  |"2005A&A...431..773R"
"* alf Sco"  |"2002AJ....124.1636K"
"* alf Sco"  |"1999ApJ...516..817K"
"* alf Sco"  |"1999ApJS..122..221M"
"* alf Sco"  |"1994A&A...281..161A"
"* alf Sco"  |"1994BAAS...26.1455D"
"* alf Sco"  |"1992BICDS..40...71P"
"* alf Sco"  |"1984ApJ...276L..21B"
```

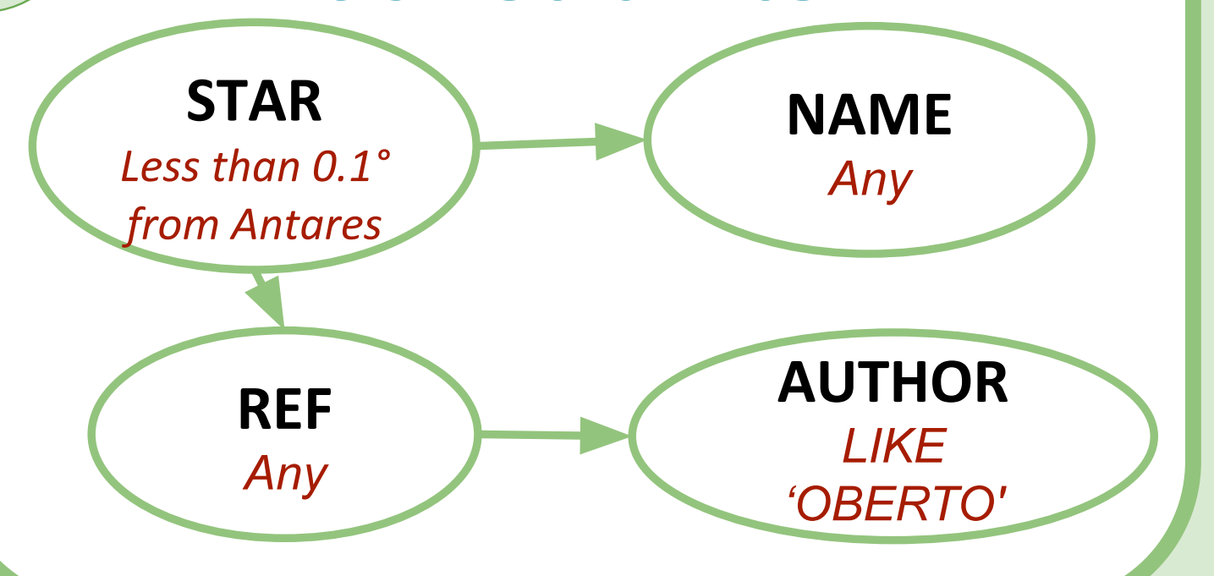


TAP database

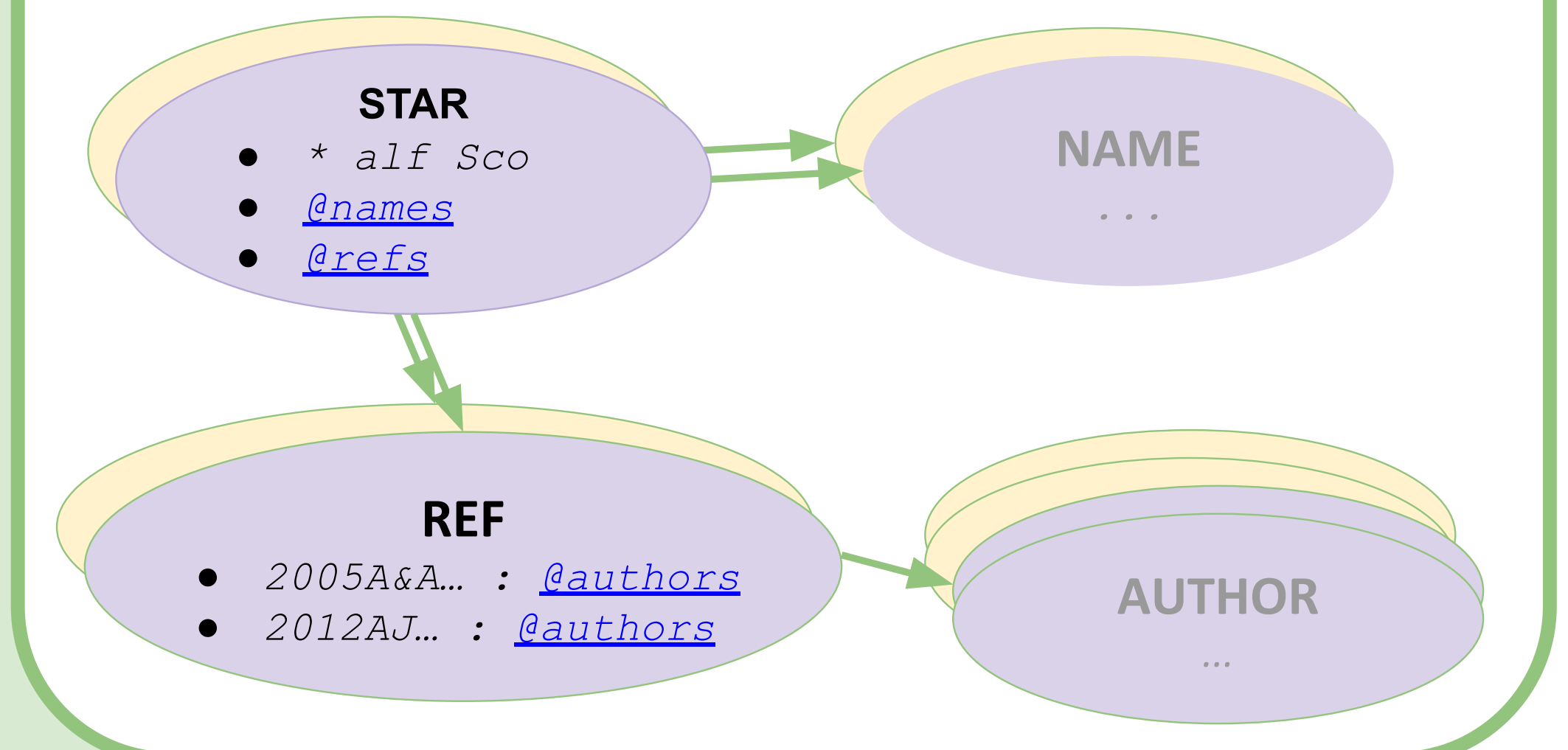
Graph based TAP Web interface

1. Conversion of the TAP_SCHEMA into a graph
2. Conversion of user constraints on graph nodes into regular ADQL query(ies)
3. Conversion of denormalized tables into a JSON structure based on the graph
4. Setup of TAP URLs requesting the content of individual graph sub-nodes

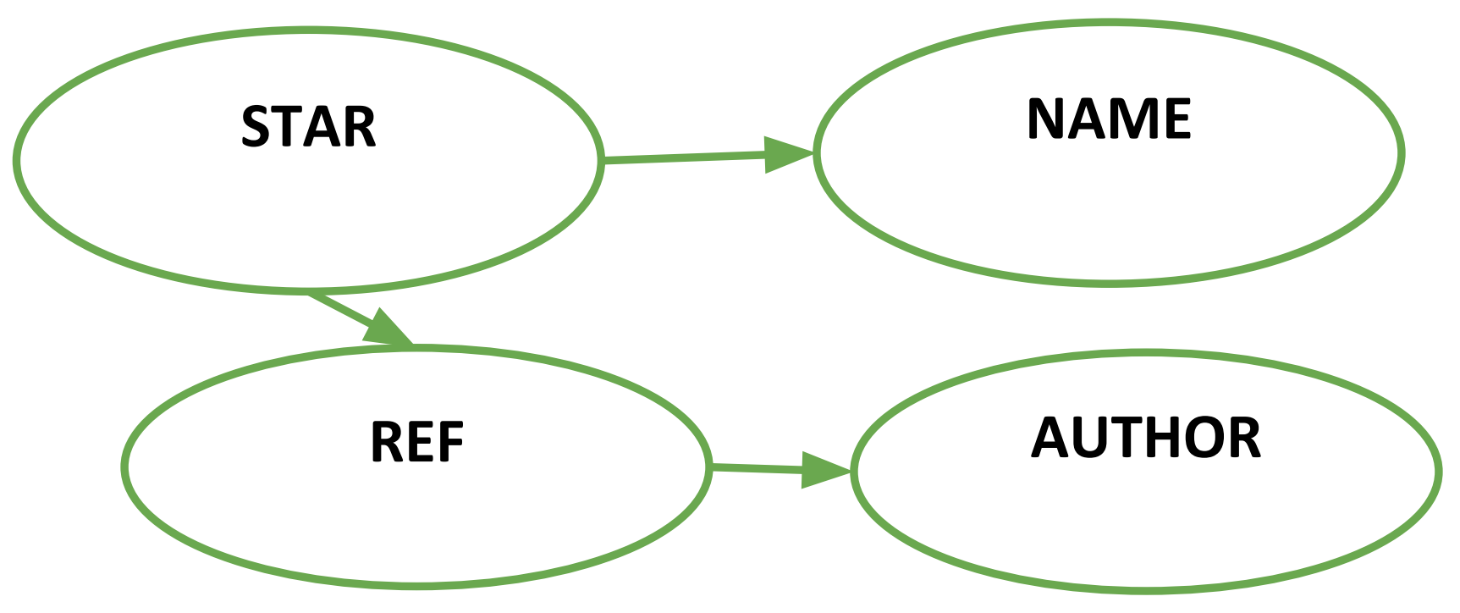
2 Graph annotated with constraints



3 4 Graph filled with results



1 Graph representation of the TAP_SCHEMA



To retrieve hierarchical data in a TAP service, users can:

- Work with a **graph representation** of the database schema.
- Get a clear view on the relationships between tables.
- Put local constraints directly on **graph nodes**.
- Get data in a structured representation.

ADASS 2019

JS TAP graph based module
Python TAP_SCHEMA explorer
https://github.com/lmichel/TAP-complex-data