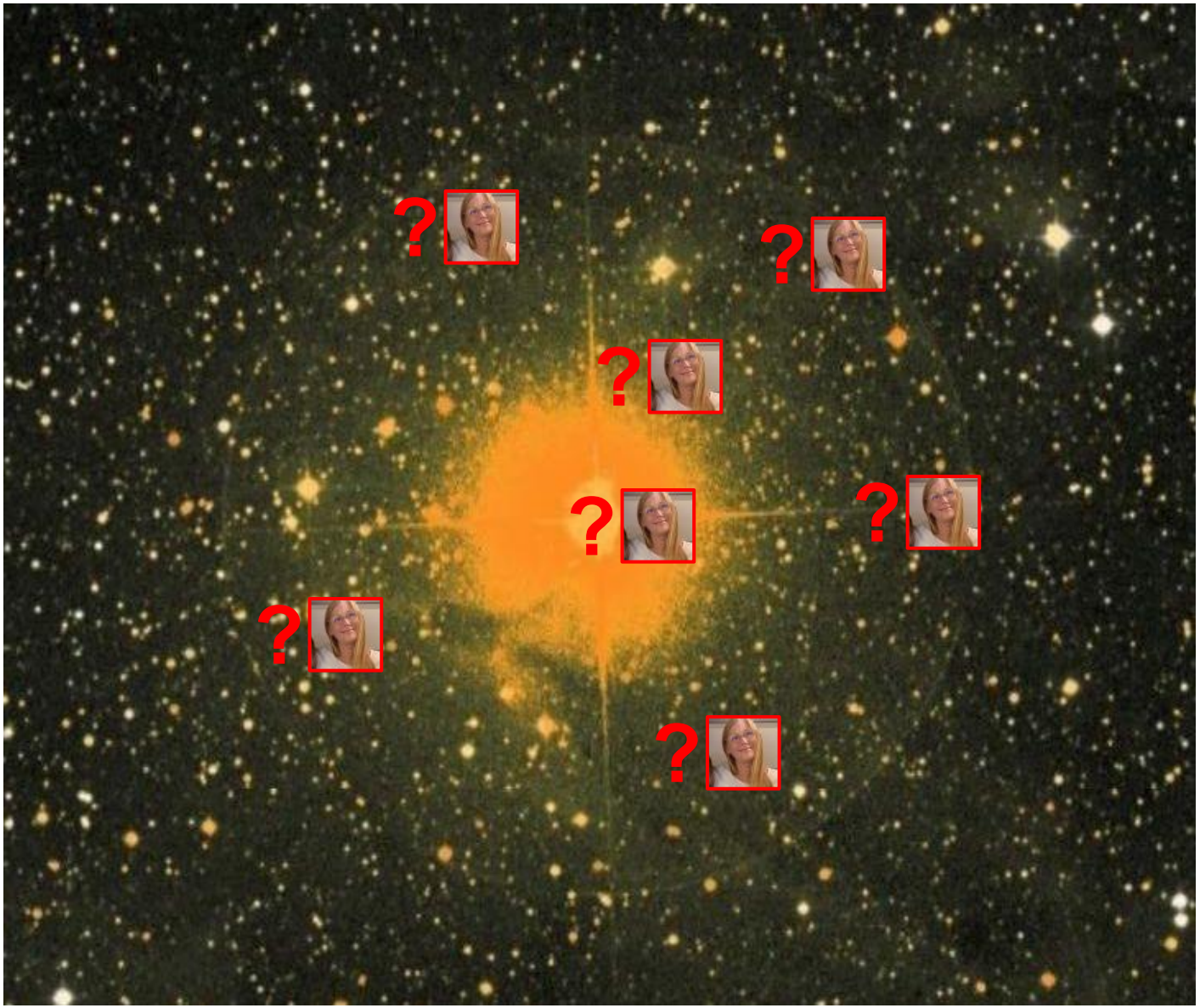
Retrieving Complex Data in TAP Services

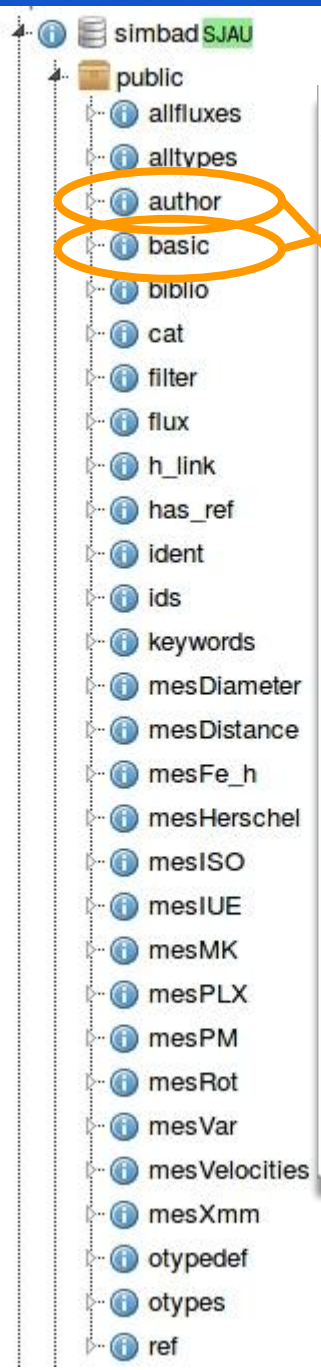
A. Oberto - H. Liao - G. Mantelet - L. Michel
Strasbourg Observatory

Work in Progress

My request



Discovering the right Simbad tables



Columns of table *public.basic* of node *simbad*

Search

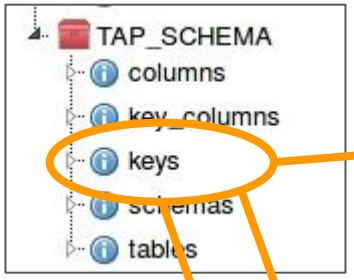
nameattr	unit	ucd	utype	dataType	description
coo_bibcode		meta.bib.bibcode;pos.eq		CHAR	Coordinate reference
coo_err_angle	deg	pos.posAng;pos.errorEllipse;pos.eq		SMALLINT	Coordinate error angle
coo_err_maj	mas	phys.angSize.smajAxis;pos.errorEllipse;pos.eq		REAL	Coordinate error major axis
coo_err_maj_prec				SMALLINT	Coordinate error major axis precision
coo_err_min	mas	phys.angSize.sminAxis;pos.errorEllipse;pos.eq		REAL	Coordinate error minor axis
coo_err_min_prec				SMALLINT	Coordinate error minor axis precision
coo_qual		meta.code.qual;pos.eq		CHAR	Coordinate quality
coo_wavelength		instr.bandpass;pos.eq		CHAR	Wavelength class for the origin of the coordinates (R,I,V,U,)
coord	deg			POINT	ADQL POINT
dec	deg	pos.eq.dec;meta.main		DOUBLE	Declination
dec_prec				SMALLINT	Declination precision
galdim_angle	deg	pos.posAng		SMALLINT	Galaxy ellipse angle
galdim_bibcode		meta.bib.bibcode;phys.angSize		CHAR	Galaxy dimension reference
galdim_majaxis	arcmin	phys.angSize.smajAxis		REAL	Angular size major axis
galdim_majaxis_prec				SMALLINT	Angular size major axis precision
galdim_minaxis	arcmin	phys.angSize			
galdim_minaxis_prec					
galdim_qual		meta.code.qu			
hpx		meta.id			
main_id		meta.id;meta			
morph_bibcode		meta.bib.bibc			
morph_qual		meta.code.qu			
morph_type		src.morph.ty			
nbref		meta.bib;met			
oid		meta.record;			
otype		src.class;me			
otype_txt		src.class			

Columns of table *public.author* of node *simbad*

Search

nameattr	unit	ucd	utype	dataType	description
name		meta.bib.author		VARCHAR	Author of a bibliographical reference
oidbibref		meta.record;meta.bib		BIGINT	Bibcode internal identifier
pos		meta.number;meta.bib.author		SMALLINT	Position of the author in the bib ref

Discovering the data path with the TAP_SCHEMA



null	alltypes	otypesToBasic	basic
null	otypedef	otypedefToBasic	basic
null	ident	identToBasic	basic
null	flux	fluxToBasic	basic
null	allfluxes	allfluxesToBasic	basic
null	has_ref	hasRefToBasic	basic
null	mesDistance	distanceToBasic	basic
null	mesDiameter	diameterToBasic	basic
null	mesFe_h	Fe_hToBasic	basic
null	mesISO	isoToBasic	basic
null	mesIUE	iueToBasic	basic
null	mesMK	mkToBasic	basic
null	mesPLX	plxToBasic	basic
null	mesPM	pmToBasic	basic
null	mesRot	rotToBasic	basic
null	mesVar	varToBasic	basic
	mesVelocities	velocitiesToBasic	basic

e query
SUBMIT | Select What | Where | Position | Plain Text Query | Job Control

Result Limit 100

```
SELECT TOP 100 *
FROM TAP_SCHEMA.keys
WHERE TAP_SCHEMA.keys.target_table = 'basic'
```

basic

has_ref

ref

author

description	from_table	key_id	target_table
null	has_ref	hasRefToRef	ref
null	has_ref	hasRefToBasic	basic

Hide query
SUBMIT | Select What | Where | Position | Plain Text Query | Job Control

Result Limit 100

```
SELECT TOP 100 *
FROM TAP_SCHEMA.keys
WHERE TAP_SCHEMA.keys.from_table = 'has_ref'
```

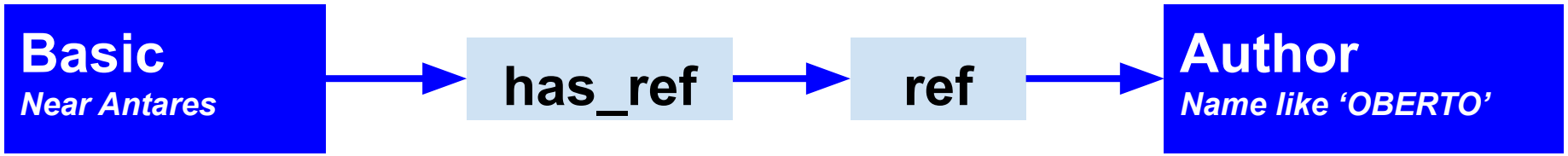
description	from_table	key_id	target_table
null	has_ref	hasRefToRef	ref
null	author	authorToRef	ref
	keywords	KeywordsToRef	ref

Hide query
SUBMIT | Select What | Where | Position | Plain Text Query | Job Control

Result Limit 100

```
SELECT TOP 100 *
FROM TAP_SCHEMA.keys
WHERE TAP_SCHEMA.keys.target_table = 'ref'
```

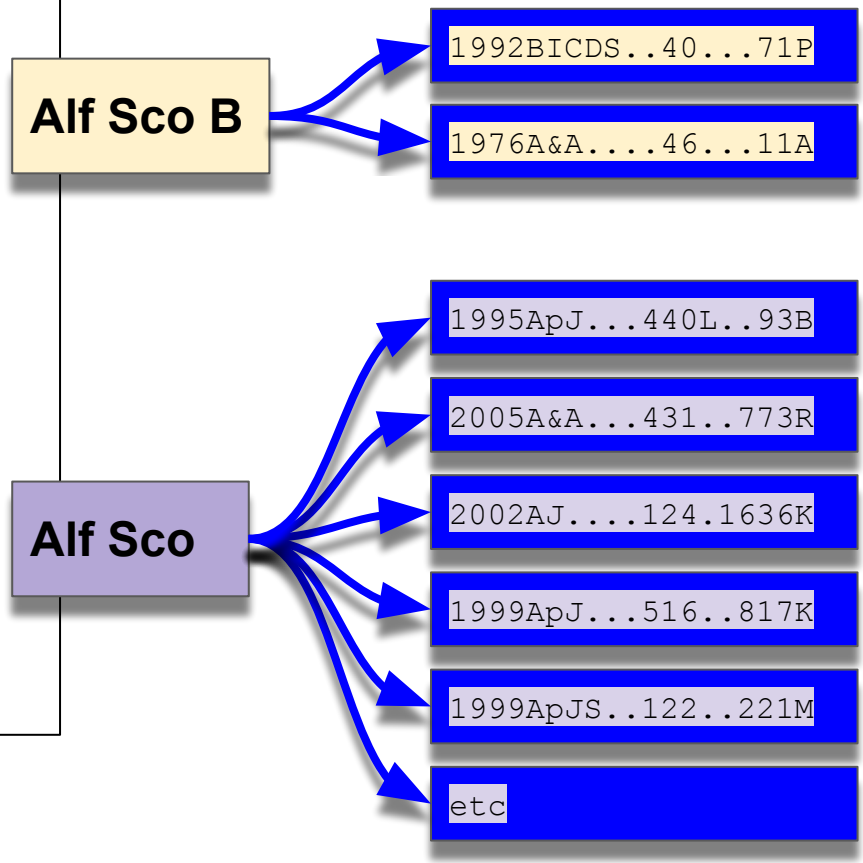
Writing the Query




```
-- 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%'
```

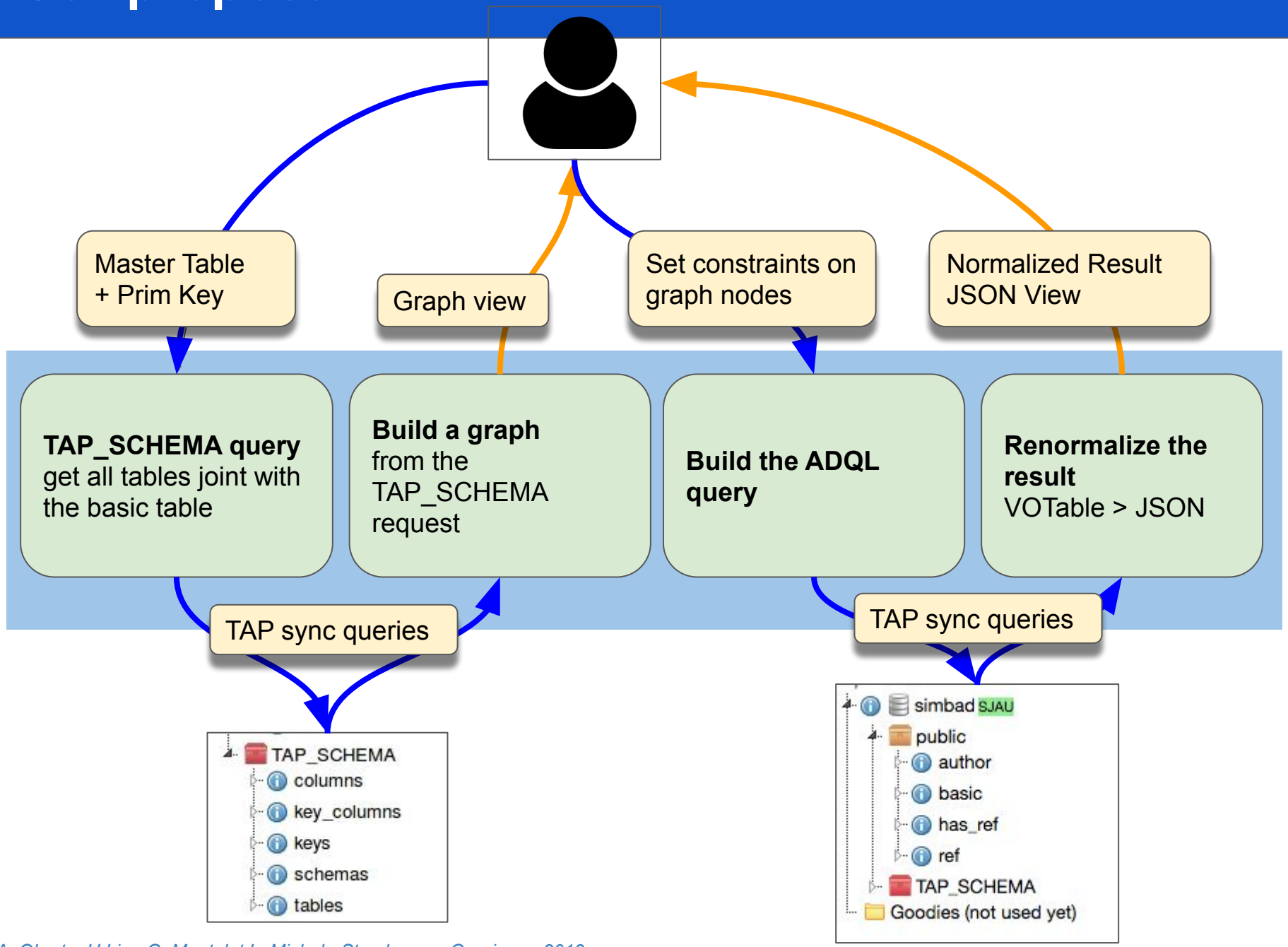
What I get finally: a denormalized table

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



The client has to reconstruct the 1-to-N associations from the denormalized table

Our proposal

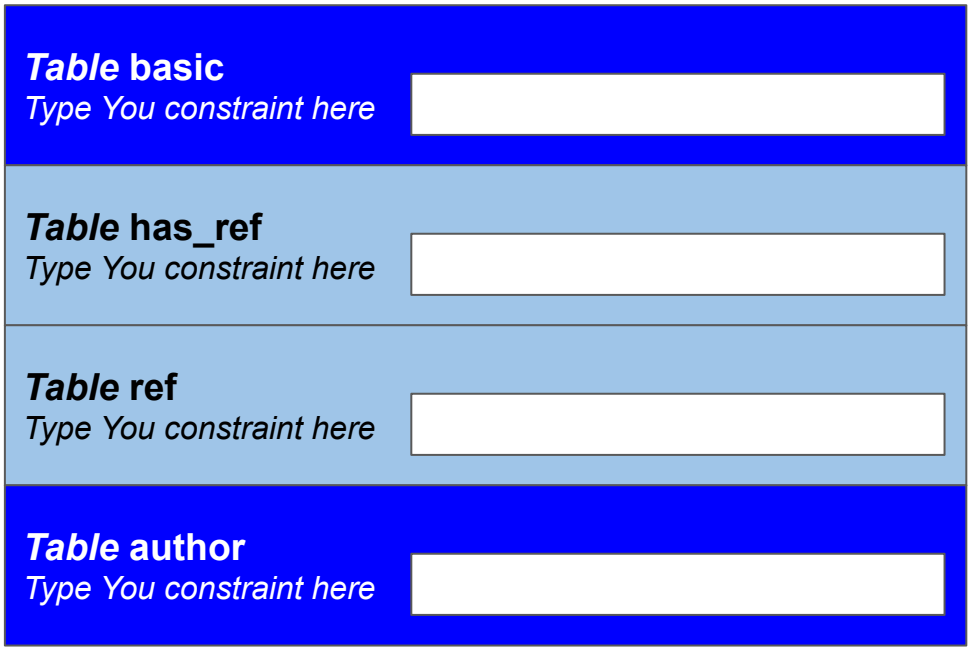


JSON Graph

```
{
"basic" :{
    "Keys":{"from": "oid"},
    "rows": [ rows ],
    "joint_tables": [
        "has_ref" :{
           "keys":{"from": "oid", "target":"oidref"},
           "rows": [ rows ],
           "joint_tables": [
                "ref" :{
                   "keys":{"from": "oidref", "target":"oidbib"},
                   "rows": [ rows ],
                    "joint_tables": [
                        "authors" :{
                            "keys":{"from": "oidbib", "target":"oidbibref"},
                            "rows": [ rows ],
                            "joint_tables": []
                         }
                    ]
                }
            ]
         },
         {….}
    ]
}
}
```

- The graph construction follows the table hierarchy
- The exact structure is not validated yet

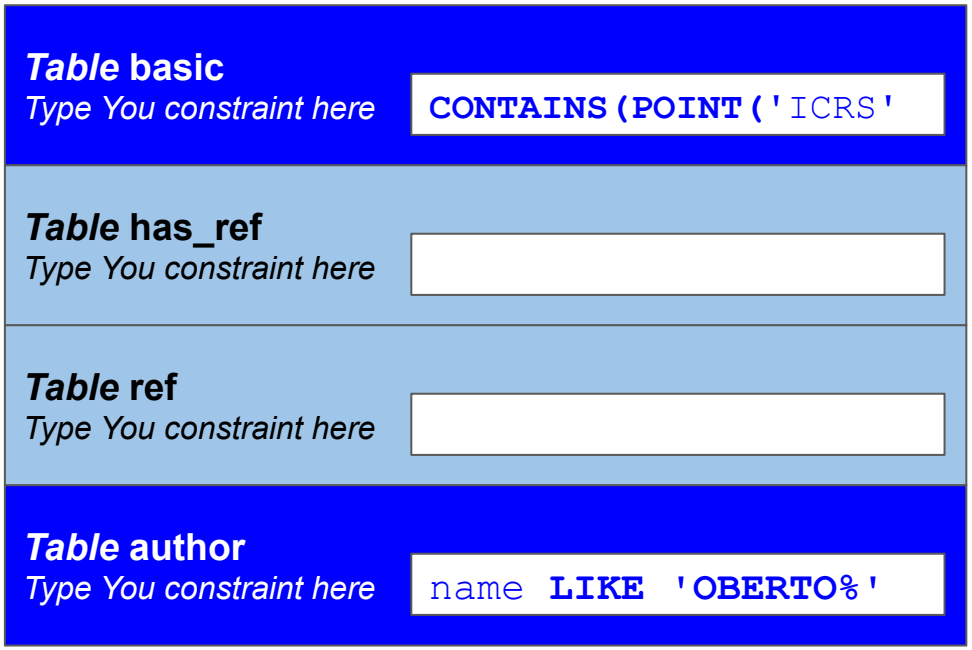
The graph view shown up to the user



- One form for each table
- One constraint editor for each table

- The HTML/CSS design still needs a bit of work :=)

The graph view shown up the user



- One form for each table
- One constraint editor for eadch table

```
{
"basic" :{
    "Keys":{"from": ""},
    "rows": [ rows ],
    "Contraint": "CONTAINS(POINT('ICRS', star.ra, star.dec)
            , CIRCLE('ICRS', antares.ra, antares.dec, 0.1)) = 1",
    "joint_tables": [
        "has_ref" :{
            "keys":{"from": "", "target":""},
            "rows": [ rows ],
            "joint_tables": [
                "ref" :{
                    "keys":{"from": "", "target":""},
                    "rows": [ rows ],
                     "joint_tables": [
                         "authors" :{
                             "keys":{"from": "", "target":""},
                             "rows": [ rows ],
                             "constraint": "name LIKE 'OBERTO%'"
                             "joint_tables": []
                         }
                     ]
                }
            ]
        },
        {….}
    ]
}
}
```

- Constraints added on graph nodes

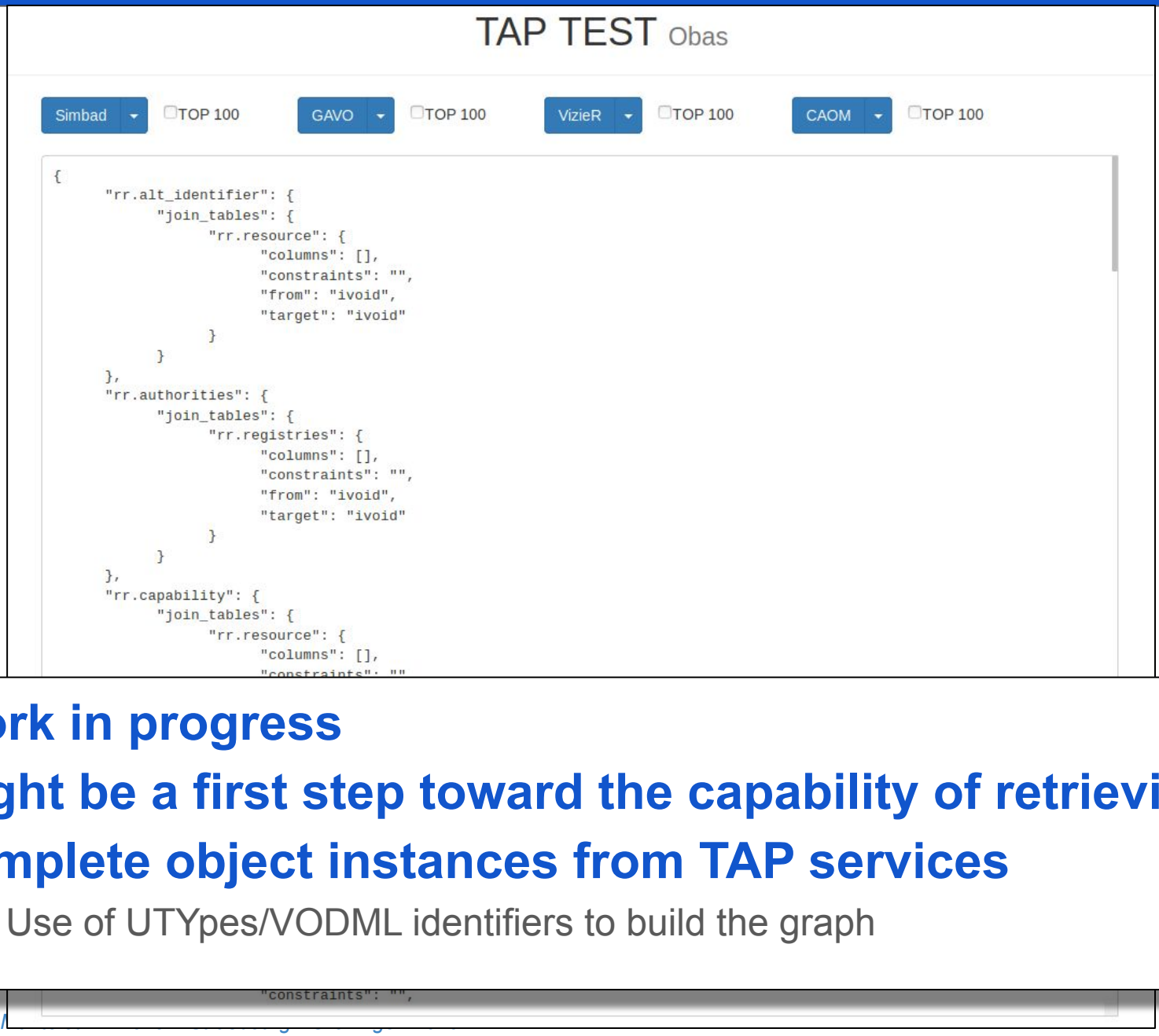
Renormalized result

```
[
{ id : "1234",
  table : "basic",
  fields: { oid: 1234, name= "alf Sco B", ra : 247.35, dec: -26.43]
  join_data:[ { id:"409987",
                table: "has_ref",
                query: "SELECT… FROM has_ref WHERE …. id='1234'
                    },
                   ….
              ]
},
{ id : "9988",
  table : "basic",
  fields: { oid: 1234, name= "alf Sco B", ra : 247.35, dec : -26.43}
  join_data:[ { id:"409987",
                    table: "has_ref",
                    query: "SELECT… FROM has_ref WHERE …. id=9988
                   },
                  ….

}
]
```

- A collection of unique primary objects from the master table
- Component can be show up by URLs
- Thes limits the resulst size and avoid to gte the whole database for service highly connected (e.g. Simbad)

Status



- **Work in progress**
- **Might be a first step toward the capability of retrieving complete object instances from TAP services**
  - Use of UTYpes/VODML identifiers to build the graph

- **Work in progress**
- **TAP services targeted**
  - Simbad
  - R Registry
  - CAOM
  - XCatDB
  - Any other service with connected tables
- **Implementation**
  - Module Javascript
  - To be used with TAPSimbad and TAPHandle
  - Could be used with other interfaces