J-PAS Tools School

ADQL

Tamara Civera & Alessandro Ederoclite

Ayuda ICTS-MRR-2021-03-CEFCA, financiada por:

What is a database?

“a **database** is an organized collection of data stored and accessed electronically”

https://en.wikipedia.org/wiki/Database

The simplest database: a table

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italia	46	Inter F.C.	Italy	2 :-(
J. Klopp	Alemania	54	Liverpool	UK	2
P. Guardiola	Spagna	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

Curating your data!

What are the types of your columns?

Name
Nationality
Age
Team
Team Nation
This year rank

The simplest database: a table

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italia	46	Inter F.C.	Italy	2 :-(
J. Klopp	Alemania	54	Liverpool	UK	2
P. Guardiola	Spagna	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

How many italian managers?

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italia	46	Inter F.C.	Italy	2
J. Klopp	Alemania	54	Liverpool	UK	2
P. Guardiola	Spagna	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

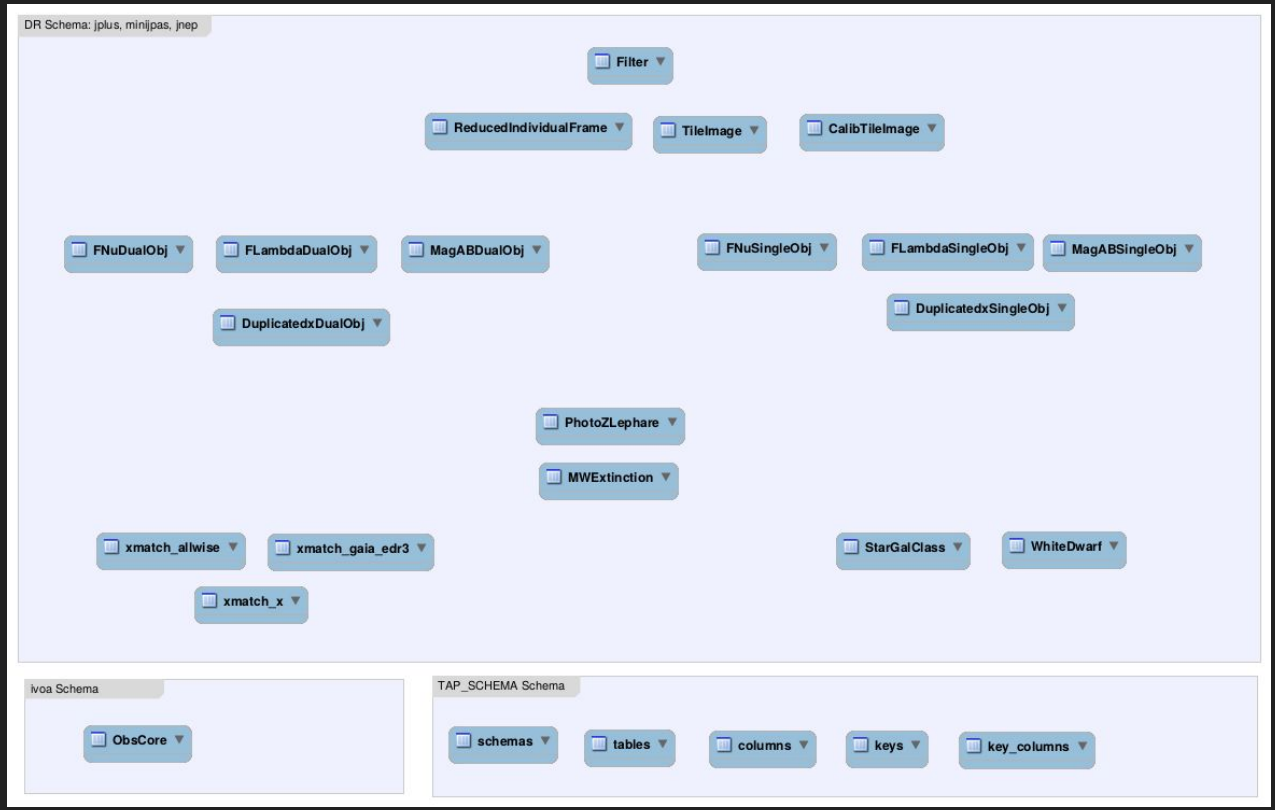
What about English teams?

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	UK	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

Ok, we are good

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	England	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

J-PAS Tables structure



SQL: Structured Query Language

It is the language which is used to query a database.

It has a “simple” syntax:

SELECT *some content*

FROM *a table*

WHERE *some condition is met*

SELECT * FROM FootballManagers

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	England	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

SELECT Age FROM FootballManagers

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	England	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

result

Age
62
56
46
54
51
50

SELECT Age FROM FootballManagers WHERE Age > 60

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	England	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

result

Age
62

SELECT Name FROM FootballManagers WHERE Age > 60

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	England	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

result

Name
C.Ancelotti

SELECT * FROM FootballManagers WHERE Age > 50 and This_year_rank > 1

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	England	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

result

Name	Nationality	Age	Team	Team Nation	This year rank
J. Klopp	Germany	54	Liverpool	England	2

A table made of all strings (many repeated) occupies **a lot** of space.

Can I fix this?

Relational databases.

In a relational database, repeated information is stored in another table and “linked” to the main table.

ID	**Nation**
1	Argentina
2	England
3	France
4	Germany
5	Italy
6	Spain

How does our table look like now? From this…

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	Italy	62	Real Madrid	Spain	1
S. Pioli	Italy	56	Milan A.C.	Italy	1
S. Inzaghi	Italy	46	Inter F.C.	Italy	2
J. Klopp	Germany	54	Liverpool	England	2
P. Guardiola	Spain	51	Manchester City	England	1
M. Pochettino	Argentina	50	PSG	France	1

… to this

Name	Nationality	Age	Team	Team Nation	This year rank
C.Ancelotti	5	62	Real Madrid	6	1
S. Pioli	5	56	Milan A.C.	5	1
S. Inzaghi	5	46	Inter F.C.	5	2
J. Klopp	4	54	Liverpool	2	2
P. Guardiola	6	51	Manchester City	2	1
M. Pochettino	1	50	PSG	3	1

JOIN: the core or relational databases

SELECT *some field*

FROM *some table*

JOIN *some table* **WITH** *another table* **ON** *some condition*

WHERE *some other conditions*

Astro Example

In astronomy we use the Astronomy Data Query Language (ADQL)

http://tapvizier.cds.unistra.fr/adql/

https://www.gaia.ac.uk/data/gaia-data-release-1/adql-cookbook

https://www.cosmos.esa.int/web/gaia-users/archive/writing-queries

https://archive.cefca.es/catalogues/minijpas-pdr201912/help_adql.html

```
select top 1000 Name, RAJ2000, DEJ2000 from "V/50/catalog"
```

```
select top 1000 "V/50/catalog".recno, HD, Name, RAJ2000,
DEJ2000, Parallax, Vmag, "B-V"  from "V/50/catalog"
```

```
select top 1000 "V/50/notes".recno, HD, Name, RAJ2000,
DEJ2000, Parallax, Vmag, "B-V" , Remark
from "V/50/catalog"
join "V/50/notes" on "V/50/notes".recno =
"V/50/catalog".recno
```

Gaia

https://gea.esac.esa.int/archive/

Important notes

- Parallax_error / parallax < 0.2
- RUWE <= 1.4

Gaia examples

```
select top 10000 * from gaiaedr3.gaia_source
```

```
select top 10000 * from gaiaedr3.gaia_source
where ruwe <= 1.4
```

```
select top 10000 * from gaiaedr3.gaia_source
where ruwe <= 1.4 and parallax_over_error >= 5
```

```
select top 10000  source_id,  phot_g_mean_mag , parallax ,
bp_rp from gaiaedr3.gaia_source
where ruwe <= 1.4 and parallax_over_error >= 5
```

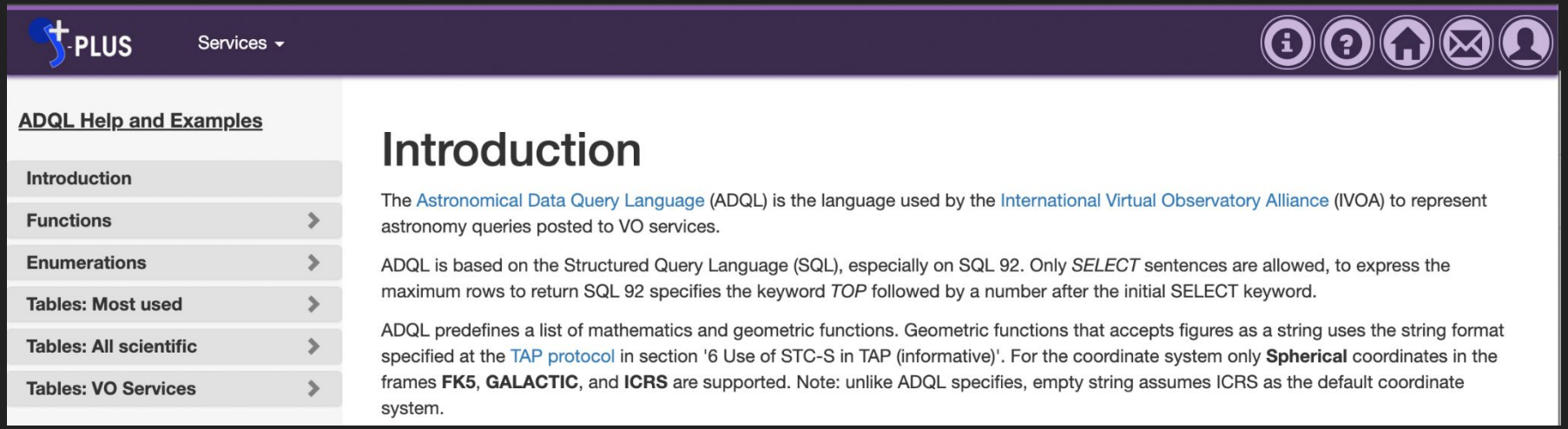
Gaia examples

```
select top 10000  source_id,  phot_g_mean_mag + 5 *
log10(parallax)- 10 as g_mag_abs,  bp_rp
from gaiaedr3.gaia_source
where parallax_over_error >= 5 and
phot_bp_mean_flux_over_error > 0
and phot_rp_mean_flux_over_error > 0 and
sqrt(power(2.5/log(10) / phot_bp_mean_flux_over_error, 2) +
power(2.5/log(10) / phot_rp_mean_flux_over_error, 2)) <=
0.05
order by random_index
```

ADQL access

V.O. Asynchronous Queries (ADQL)	Lets you search the database for all objects that meet any criteria you can think of, then returns whatever object data you request. Database queries are in Astronomical Data Query Language (ADQL), which is basically a standardised version of SQL (ADQL help and examples).	You want to answer a specific astronomical research question.

ADQL help


-PLUS
Services
ADQL Help and Examples
Introduction
Functions
Enumerations
Tables: Most used
Tables: All scientific
Tables: VO Services
Introduction
The Astronomical Data Query Language (ADQL) is the language used by the International Virtual Observatory Alliance (IVOA) to represent astronomy queries posted to VO services.
ADQL is based on the Structured Query Language (SQL), especially on SQL 92. Only SELECT sentences are allowed, to express the maximum rows to return SQL 92 specifies the keyword TOP followed by a number after the initial SELECT keyword.
ADQL predefines a list of mathematics and geometric functions. Geometric functions that accepts figures as a string uses the string format specified at the TAP protocol in section '6 Use of STC-S in TAP (informative)'. For the coordinate system only Spherical coordinates in the frames FK5, GALACTIC, and ICRS are supported. Note: unlike ADQL specifies, empty string assumes ICRS as the default coordinate system.

ADQL Help and Examples

Introduction

Functions >

Enumerations v

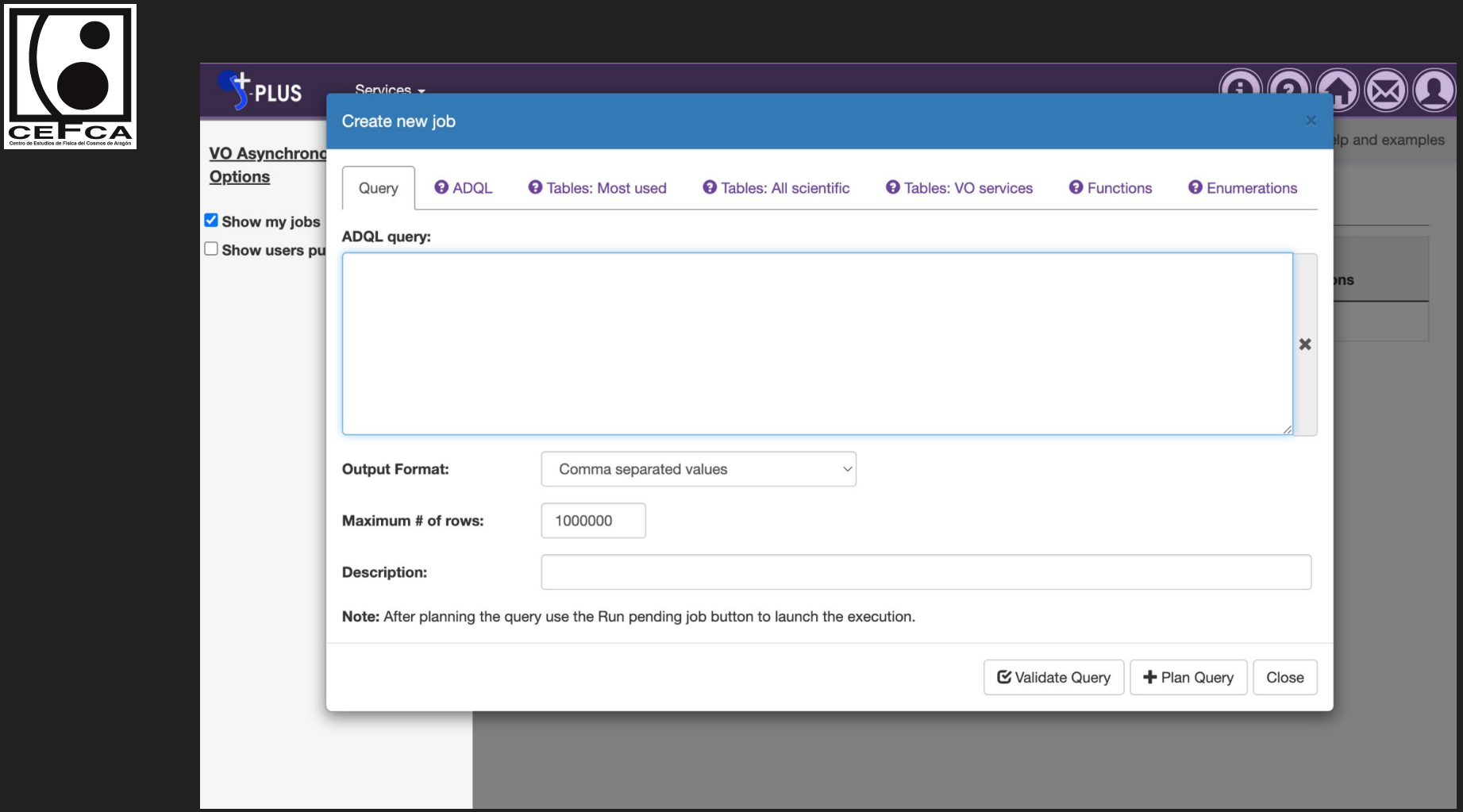
- calibration_method
- jplus
- jplus_img_flag

Tables: Most used v

- jplus.CalibTileImage
- jplus.Filter
- jplus.MagABDualObj
- jplus.MagABSingleObj
- jplus.PhotoZLephare
- jplus.ReducedIndividualFrame
- jplus.TileImage

Tables: All scientific >

Tables: VO Services >


PLUS
Services
VO Asynchronc
Options
Show my jobs
Show users pu
lp and examples
ons
Create new job
Query
ADQL
Tables: Most used
Tables: All scientific
Tables: VO services
Functions
Enumerations
ADQL query:
Output Format:
Comma separated values
Maximum # of rows:
1000000
Description:
Note: After planning the query use the Run pending job button to launch the execution.
Validate Query
Plan Query
Close

ADQL limited to 1,000,000 objects *see also Tamara's comment on Jupyter

Default output is a CSV (I normally change to fits)

Let's run a first easy query

```
select * from filters
```

or

```
select * from tileimage
```

Tips on tables: MagABDualObj

The identifier of each object is **TILE_ID** and **NUMBER**

There are many magnitudes. For your convenience, we just use MAG_AUTO. Magnitudes are stored as arrays!

CLASS_STAR is SExtractor's default galaxy/star classifier (probability of being point source)

FLAGS is a binary code of photometric quality (<4 is good and between 2048 and 2051 is a good known variable). **FLAGS** are stored as arrays!

Arrays

Arrays are in “order”: rSDSS, gSDSS, iSDSS, zSDSS, uJAVA, J0378, J0395, J0410, J0430, J0515, J0660, J0861

There are some “array specific” ADQL functions

- ARRAY_ADD
- ARRAY_ADD_SCALAR
- ARRAY_DIV
- ARRAY_DIV_SCALAR
- ARRAY_MAX_FLOAT
- ARRAY_MAX_INT
- ARRAY_MIN_FLOAT
- ARRAY_MIN_INT
- ARRAY_MULT
- ARRAY_MULT_SCALAR
- ARRAY_SUB
- ARRAY_SUB_SCALAR

My quality criteria

```
(array_max_int(flags) < 3 OR flags[jplus::rSDSS] BETWEEN
2048 AND 2051)

AND

array_min_float(NORM_WMAP_VAL) > 0.8

AND

array_max_int(MASK_FLAGS) < 1
```

Let’s try to make a colour magnitude diagram

Use tile 99390

```
select top 10000 tile_id,number,alpha_J2000,delta_j2000,
mag_auto

from magABdualobj

where tile_id = 99390 and
(array_max_int(flags) < 3 OR flags[jplus::rSDSS] BETWEEN
2048 AND 2051) AND array_min_float(NORM_WMAP_VAL) > 0.8 AND
array_max_int(MASK_FLAGS) < 1
```

Create an alias for the table

```
select top 10000
jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000, jp.mag_auto

from magABdualobj jp

where jp.tile_id = 99390 and
(array_max_int(jp.flags) < 3 OR jp.flags[jplus::rSDSS] BETWEEN
2048 AND 2051) AND array_min_float(jp.NORM_WMAP_VAL) > 0.8 AND
array_max_int(jp.MASK_FLAGS) < 1
```

```
select top 10000
jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000,
jp.mag_auto, sgc.sglc_prob_star

from magABdualobj  jp

join StarGalClass sgc on jp.number = sgc.number and
jp.tile_id = jp.tile_id

where jp.tile_id = 99390
and sgc.sglc_prob_star > 0.8 and (array_max_int(jp.flags) <
3 OR jp.flags[jplus::rSDSS] BETWEEN 2048 AND 2051) AND
array_min_float(jp.NORM_WMAP_VAL) > 0.8 AND
array_max_int(jp.MASK_FLAGS) < 1
```

```
select top 10000
jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000,
jp.mag_auto, jp.mag_err_auto, sgc.sglc_prob_star

from magABdualobj  jp

join StarGalClass sgc on jp.number = sgc.number and
jp.tile_id = jp.tile_id

where jp.tile_id = 99390 and sgc.sglc_prob_star > 0.8 and
array_max_float(jp.mag_err_auto) < 0.1
and (array_max_int(jp.flags) < 3 OR jp.flags[jplus::rSDSS]
BETWEEN 2048 AND 2051) AND array_min_float(jp.NORM_WMAP_VAL)
> 0.8 AND array_max_int(jp.MASK_FLAGS) < 1
```

```
select top 10000 jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000,
jp.mag_auto[jplus::rSDSS] as rSDSS, jp.mag_auto[jplus::gSDSS] as
gSDSS, mag_err_auto, sgc.sglc_prob_star

from magABdualobj  jp

join StarGalClass sgc on jp.number = sgc.number and jp.tile_id =
jp.tile_id

where jp.tile_id = 99390 and sgc.sglc_prob_star > 0.8 and
array_max_float(jp.mag_err_auto) < 0.1 and (array_max_int(jp.flags) <
3 OR jp.flags[jplus::rSDSS] BETWEEN 2048 AND 2051) AND
array_min_float(jp.NORM_WMAP_VAL) > 0.8 AND
array_max_int(jp.MASK_FLAGS) < 1
```

```
select top 10000 jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000,
jp.mag_auto[jplus::rSDSS] as rSDSS, jp.mag_auto[jplus::gSDSS] as gSDSS,
mag_err_auto, sgc.sglc_prob_star , gaia.parallax , gaia.parallax_error

from magABdualobj  jp

join StarGalClass sgc on jp.number = sgc.number and jp.tile_id = jp.tile_id

join xmatch_gaia_edr3 gaia on jp.number = gaia.number and jp.tile_id =
gaia.tile_id

where jp.tile_id = 99390 and sgc.sglc_prob_star > 0.8 and
array_max_float(jp.mag_err_auto) < 0.1 and (array_max_int(jp.flags) < 3 OR
jp.flags[jplus::rSDSS] BETWEEN 2048 AND 2051) AND
array_min_float(jp.NORM_WMAP_VAL) > 0.8 AND array_max_int(jp.MASK_FLAGS) < 1
```

```
select top 10000 jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000,
jp.mag_auto[jplus::rSDSS] as rSDSS, jp.mag_auto[jplus::gSDSS] as gSDSS,
mag_err_auto, sgc.sglc_prob_star , gaia.parallax , gaia.parallax_error

from magABdualobj  jp

join StarGalClass sgc on jp.number = sgc.number and jp.tile_id = jp.tile_id

join xmatch_gaia_edr3 gaia on jp.number = gaia.number and jp.tile_id =
gaia.tile_id

where jp.tile_id = 99390 and sgc.sglc_prob_star > 0.8 and
array_max_float(jp.mag_err_auto) < 0.1  and gaia.parallax_over_error > 5  and
(array_max_int(jp.flags) < 3 OR jp.flags[jplus::rSDSS] BETWEEN 2048 AND
2051) AND array_min_float(jp.NORM_WMAP_VAL) > 0.8 AND
array_max_int(jp.MASK_FLAGS) < 1
```

```
select top 10000 jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000,
jp.mag_auto[jplus::gSDSS] - jp.mag_auto[jplus::rSDSS]  as g_minus_r ,
jp.mag_auto[jplus::gSDSS] + 5 - 5 * log10(1000/gaia.parallax) as g_abs,
mag_err_auto, sgc.sglc_prob_star , gaia.parallax , gaia.parallax_error

from magABdualobj  jp

join StarGalClass sgc on jp.number = sgc.number and jp.tile_id = jp.tile_id

join xmatch_gaia_edr3 gaia on jp.number = gaia.number and jp.tile_id =
gaia.tile_id

where jp.tile_id = 99390 and sgc.sglc_prob_star > 0.8 and
array_max_float(jp.mag_err_auto) < 0.1 and gaia.parallax_over_error > 5 and
(array_max_int(jp.flags) < 3 OR jp.flags[jplus::rSDSS] BETWEEN 2048 AND 2051) AND
array_min_float(jp.NORM_WMAP_VAL) > 0.8 AND array_max_int(jp.MASK_FLAGS) < 1
```

```
select top 10000 jp.tile_id,jp.number,jp.alpha_J2000,jp.delta_j2000,
jp.mag_auto[jplus::gSDSS]  - mwe.ax[jplus::gSDSS]  - jp.mag_auto[jplus::rSDSS]   +
mwe.ax[jplus::rSDSS]  as g_minus_r ,  jp.mag_auto[jplus::gSDSS]  -
mwe.ax[jplus::gSDSS]  + 5 - 5 * log10(1000/gaia.parallax) as g_abs, mag_err_auto,
sgc.sglc_prob_star , gaia.parallax , gaia.parallax_error

from magABdualobj  jp

join StarGalClass sgc on jp.number = sgc.number and jp.tile_id = jp.tile_id

join xmatch_gaia_edr3 gaia on jp.number = gaia.number and jp.tile_id =
gaia.tile_id

join MWExtinction mwe on jp.number = mwe.number and jp.tile_id = mwe.tile_id

where jp.tile_id = 99390 and sgc.sglc_prob_star > 0.8 and
array_max_float(jp.mag_err_auto) < 0.1 and gaia.parallax_over_error > 5 and
(array_max_int(jp.flags) < 3 OR jp.flags[jplus::rSDSS] BETWEEN 2048 AND 2051) AND
array_min_float(jp.NORM_WMAP_VAL) > 0.8 AND array_max_int(jp.MASK_FLAGS) < 1
```

… the sky is the limit!

Remember to study the tables before start using them!