Gaia Data Queries with TAP/ADQL and TOPCAT

Markus Demleitner & Hendrik Heinl

Adapted by the SVO (Dic 2020)

svo-support@cab.inta-csic.es

TAP: T(able) A(ccess) P(rotocol)

ADQL: A(stronomical) D(ata) Q(uery) L(anguage)

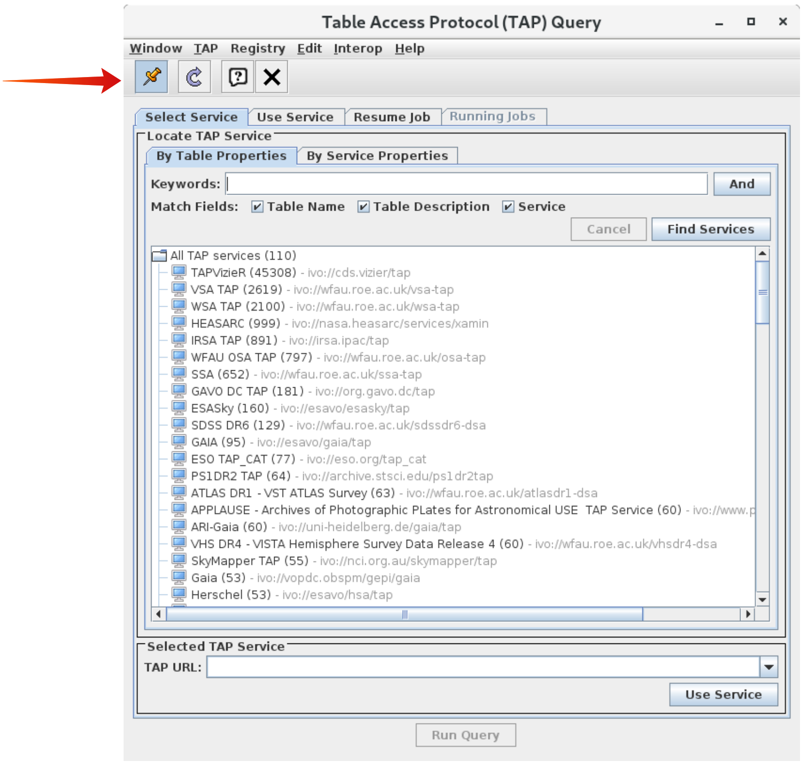
VO protocols like ConeSearch, SSAP or SIAP only allow simple queries, mainly by position and search radius. The table access protocol (TAP[1]) defines a service protocol to make more complex queries to astronomical catalogs as well as general database tables. TAP uses ADQL (Astronomical Data Query Languaje) to build expressions of relational algebra.

The basic syntax is composed by a *SELECT* command to select the number of rows and columns wanted for the output and a *FROM* command to indicate the table we want to ask for.

In this tutorial we will see a few introductory examples of queries with TAP/ADQL.

Note that all names in ADQL (column names, table names, commands, etc) are case-insensitive.

- Launch TOPCAT
- In the menu on top, go to *VO → Table Access Protocol (TAP) query.* A new window (*"Table Access Protocol (TAP) Query"*) will pop up. If you wish, click the pin icon in the upper left corner of this window to keep it open even while the query is executing (background must be blue).



All TAP services (110)
- TAPVizieR (45308) - ivo://cds.vizier/tap
- VSA TAP (2619) - ivo://wfau.roe.ac.uk/vsa-tap
- WSA TAP (2100) - ivo://wfau.roe.ac.uk/wsa-tap
- HEASARC (999) - ivo://nasa.heasarc/services/xamin
- IRSA TAP (891) - ivo://irsa.ipac/tap
- WFAU OSA TAP (797) - ivo://wfau.roe.ac.uk/osa-tap
- SSA (652) - ivo://wfau.roe.ac.uk/ssa-tap
- GAVO DC TAP (181) - ivo://org.gavo.dc/tap
- ESASky (160) - ivo://esavo/esasky/tap
- SDSS DR6 (129) - ivo://wfau.roe.ac.uk/sdssdr6-dsa
- GAIA (95) - ivo://esavo/gaia/tap
- ESO TAP_CAT (77) - ivo://eso.org/tap_cat
- PS1DR2 TAP (64) - ivo://archive.stsci.edu/ps1dr2tap
- ATLAS DR1 - VST ATLAS Survey (63) - ivo://wfau.roe.ac.uk/atlasdr1-dsa
- APPLAUSE - Archives of Photographic PLates for Astronomical USE TAP Service (60) - ivo://www.p
- ARI-Gaia (60) - ivo://uni-heidelberg.de/gaia/tap
- VHS DR4 - VISTA Hemisphere Survey Data Release 4 (60) - ivo://wfau.roe.ac.uk/vhsdr4-dsa
- SkyMapper TAP (55) - ivo://nci.org.au/skymapper/tap
- Gaia (53) - ivo://vopdc.obspm/gepi/gaia
- Herschel (53) - ivo://esavo/hsa/tap

[1] http://www.ivoa.net/documents/TAP/

- Enter Gaia in the **Keywords** box . Click **Find Services.**
- Click on **ARI-Gaia**. The corresponging TAP URL will appear in the TAP URL box (at the bottom of the window). Hit **Use Service**. You will be addressed to the **Use service** tab.
- In the tab **Use service** you will see all the tables available from the Gaia database. Select gaiadr2.gaia_source on the left panel. If you click on the **Columns** tab to the right, you will get information of all the columns available in this particular table.

  - **SELECT / TOP**
    - In the bottom box enter:

      SELECT TOP 5 * FROM gaiadr2.gaia_source

    - Hit **Run Query.**

      The *TOP* command followed by an integer is used to indicate how many rows we want returned.

      The "*" indicates that we want returned all columns in the table that we specify before the *FROM* command.

      Once the query is finished, a new table should have been created in TOPCAT with five rows and 94 columns:

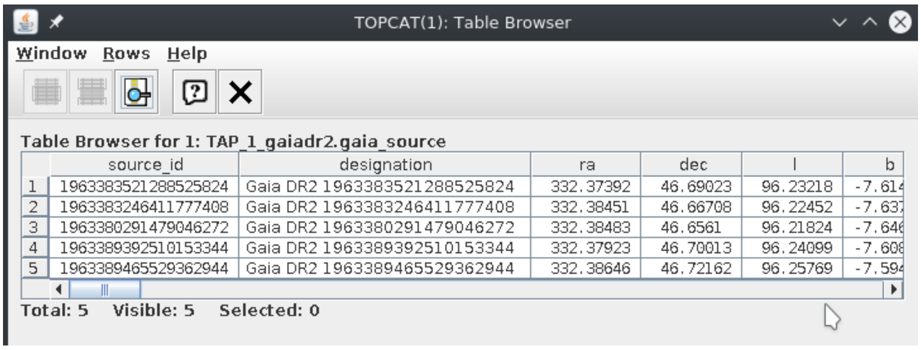


Table Browser for 1: TAP_1_gaiadr2.gaia_source

	source_id	designation	ra	dec	l	b
1	1963383521288525824	Gaia DR2 1963383521288525824	332.37392	46.69023	96.23218	-7.614
2	1963383246411777408	Gaia DR2 1963383246411777408	332.38451	46.66708	96.22452	-7.637
3	1963380291479046272	Gaia DR2 1963380291479046272	332.38483	46.6561	96.21824	-7.646
4	1963389392510153344	Gaia DR2 1963389392510153344	332.37923	46.70013	96.24099	-7.608
5	1963389465529362944	Gaia DR2 1963389465529362944	332.38646	46.72162	96.25769	-7.594

Total: 5 Visible: 5 Selected: 0

  - **SELECT / ORDER BY**
    - In the bottom box enter:

      SELECT TOP 5 source_id, phot_g_mean_mag FROM gaiadr2.gaia_source ORDER BY phot_g_mean_mag

    - Hit **Run Query**.

    Note that we are replacing the "*" in the previous example with the name of the two columns we want to be returned. We use one of those columns to indicate how we want the output to be ordered by using *ORDER BY*.

    You will get the five brightest stars in the Gaia DR2 source catalogue.

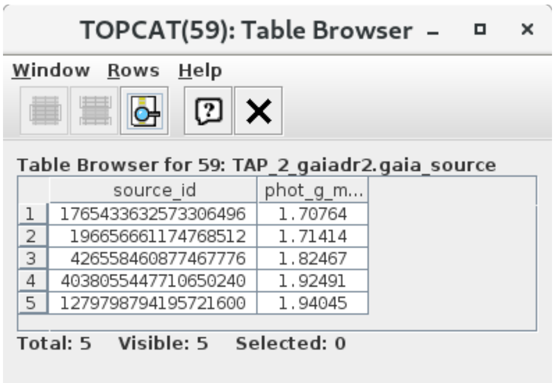


Table Browser for 59: TAP_2_gaiadr2.gaia_source

	source_id	phot_g_m...
1	1765433632573306496	1.70764
2	196656661174768512	1.71414
3	426558460877467776	1.82467
4	4038055447710650240	1.92491
5	1279798794195721600	1.94045

Total: 5 Visible: 5 Selected: 0

- If you now enter:

  SELECT TOP 5 source_id,phot_g_mean_mag FROM gaiadr2.gaia_source ORDER BY phot_g_mean_mag DESC

- and click **Run Query** you will get the five faintest stars in the Gaia DR2 source catalogue.

  By using *DESC* at the end of the query, we are ordering the apparent magnitude in descending order before making the selection of the first five entries.

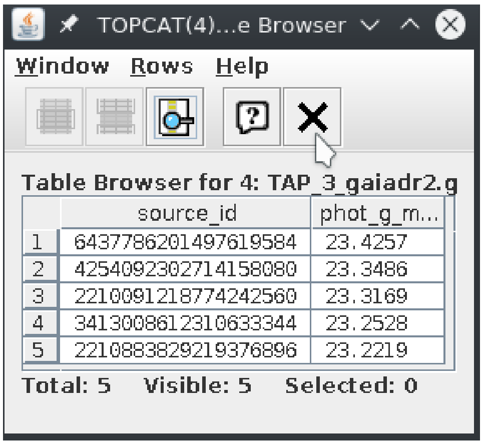


Table Browser for 4: TAP_3_gaiadr2.g

	source_id	phot_g_m...
1	6437786201497619584	23.4257
2	4254092302714158080	23.3486
3	2210091218774242560	23.3169
4	3413008612310633344	23.2528
5	2210883829219376896	23.2219

Total: 5 Visible: 5 Selected: 0

- **SELECT / WHERE**
  - In the bottom box enter:

    SELECT source_id, parallax, parallax_error FROM gaiadr2.gaia_source WHERE parallax>100 AND parallax_error/parallax<0.1 ORDER BY parallax DESC

  - Hit **Run Query**.

    This query returns all objects (note that no *TOP* command is used here) that satisfy the imposed criteria with the *WHERE* command: to be at a shorten distance than 10 pc and with good parallax determinations, defined as relative error lower than 10% (parallax_error/parallax<0.1). We ordered the selection by descending parallax (i.e., closest objects come first) using *DESC*.

The output will be a table with 1722 objects and the three columns indicated after the *SELECT* command.

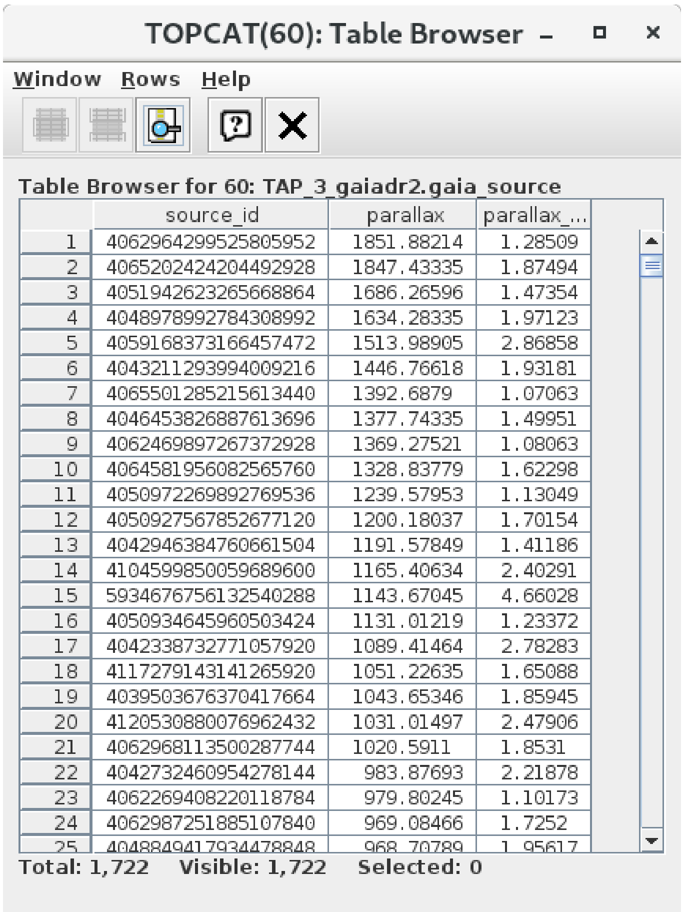


Table Browser for 60: TAP_3_gaiadr2.gaia_source

	source_id	parallax	parallax_...
1	4062964299525805952	1851.88214	1.28509
2	4065202424204492928	1847.43335	1.87494
3	4051942623265668864	1686.26596	1.47354
4	4048978992784308992	1634.28335	1.97123
5	4059168373166457472	1513.98905	2.86858
6	4043211293994009216	1446.76618	1.93181
7	4065501285215613440	1392.6879	1.07063
8	4046453826887613696	1377.74335	1.49951
9	4062469897267372928	1369.27521	1.08063
10	4064581956082565760	1328.83779	1.62298
11	4050972269892769536	1239.57953	1.13049
12	4050927567852677120	1200.18037	1.70154
13	4042946384760661504	1191.57849	1.41186
14	4104599850059689600	1165.40634	2.40291
15	5934676756132540288	1143.67045	4.66028
16	4050934645960503424	1131.01219	1.23372
17	4042338732771057920	1089.41464	2.78283
18	4117279143141265920	1051.22635	1.65088
19	4039503676370417664	1043.65346	1.85945
20	4120530880076962432	1031.01497	2.47906
21	4062968113500287744	1020.5911	1.8531
22	4042732460954278144	983.87693	2.21878
23	4062269408220118784	979.80245	1.10173
24	4062987251885107840	969.08466	1.7252
25	4048849417934478848	968.70789	1.95617

Total: 1,722 Visible: 1,722 Selected: 0

- **SELECT / COUNT**
  - Use *COUNT(*)* to figure out how many rows there are in a table:

    SELECT COUNT(*) FROM gaiadr1.tgas_source

  - Hit **Run Query**.

    The output will be a new table in TOPCAT with one row indicating this number (2057050).

- **CREATING NEW COLUMNS**
  - In the bottom box enter:

    SELECT TOP 5 source_id, p m r a , pmdec, sqrt(power(pmra,2)+power(pmdec,2)) AS pm_tot FROM gaiadr2.gaia_source

  - Hit **Run Query**.

You will obtain a new table in TOPCAT with five rows (indicated with the *TOP* command) and the specified four columns, one of which is an algebraic operation and that we have named "pm_tot" using *AS*.

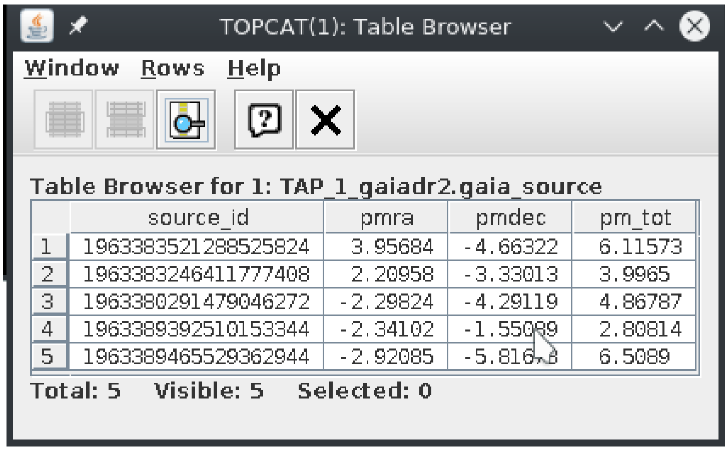


Table Browser for 1: TAP_1_gaiadr2.gaia_source

	source_id	pmra	pmdec	pm_tot
1	1963383521288525824	3.95684	-4.66322	6.11573
2	1963383246411777408	2.20958	-3.33013	3.9965
3	1963380291479046272	-2.29824	-4.29119	4.86787
4	1963389392510153344	-2.34102	-1.55089	2.80814
5	1963389465529362944	-2.92085	-5.816[illegible]8	6.5089

Total: 5 Visible: 5 Selected: 0

- **GROUPING**

  For histogram-like functionality, you can compute factor sets, i.e., subsets that have identical values for one or more columns, and you can compute aggregate functions for them.

  - In the bottom box enter:

    SELECT COUNT(*) AS n, ROUND(phot_g_mean_mag) AS bin, AVG(parallax) FROM gaiadr1.tgas_source GROUP BY bin ORDER BY bin

  - Hit **Run Query**.

    Here we use the GROUP BY command to group all objects with the same G magnitude taken as an integer by running *ROUND*. For each bin, we have calculated the average parallax with *AVG* and required it as an ouptut, together with the number of sources in that bin.

    A new table will be created in TOPCAT with 16 rows and the following information:

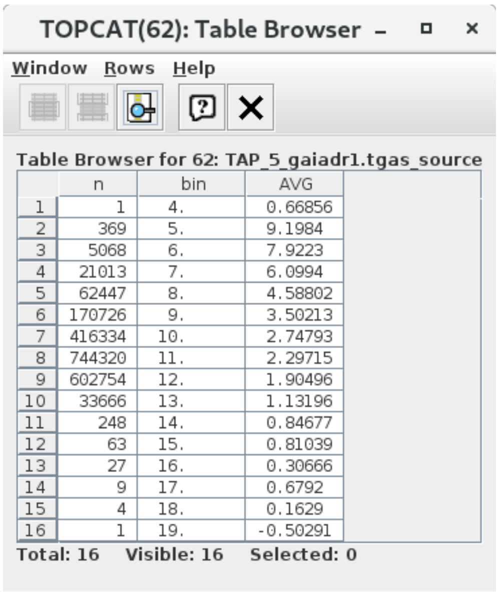


Table Browser for 62: TAP_5_gaiadr1.tgas_source

	n	bin	AVG
1	1	4.	0.66856
2	369	5.	9.1984
3	5068	6.	7.9223
4	21013	7.	6.0994
5	62447	8.	4.58802
6	170726	9.	3.50213
7	416334	10.	2.74793
8	744320	11.	2.29715
9	602754	12.	1.90496
10	33666	13.	1.13196
11	248	14.	0.84677
12	63	15.	0.81039
13	27	16.	0.30666
14	9	17.	0.6792
15	4	18.	0.1629
16	1	19.	-0.50291

Total: 16 Visible: 16 Selected: 0

As expected, brighter stars tend to have larger parallaxes.

- **JOINING:**

  So far, we have worked with a single table but we can join more tables as in the following example.

  - In the bottom box enter:

    SELECT TOP 10 h1.ra, h1.dec, t1.source_id FROM gaiadr2.gaia_source AS h1 JOIN gaiadr2.tmass_best_neighbour AS t1 USING(source_id)

  - Hit **Run Query**.

    We are taking the first 10 rows (*TOP* 10) in the table resulting from the match between tables.

    Note that we are renaming the first table (gaiadr2.gaia_source) to “h1” and the second table (gaiadr2.tmass_best_neighbour) to “t1” with the *AS* command. Because of this, the columns ra, dec and source_id from the first table that we want to obtain in the output are preceded by the new name given to it (e.g., “h1.ra”).

    The *JOIN* command performs a cross-match between the two tables. To indicate the column that we want to use as a reference for the cross-match, we use the *USING* command.

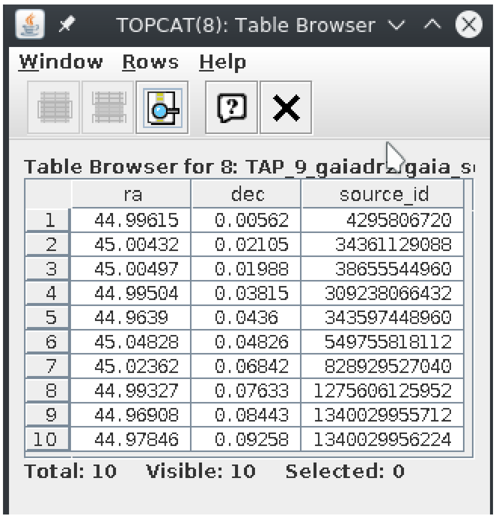


Table Browser for 8: TAP_9_gaiadr2_gaia_s

	ra	dec	source_id
1	44.99615	0.00562	4295806720
2	45.00432	0.02105	34361129088
3	45.00497	0.01988	38655544960
4	44.99504	0.03815	309238066432
5	44.9639	0.0436	343597448960
6	45.04828	0.04826	549755818112
7	45.02362	0.06842	828929527040
8	44.99327	0.07633	1275606125952
9	44.96908	0.08443	1340029955712
10	44.97846	0.09258	1340029956224

Total: 10   Visible: 10   Selected: 0

All the above is just a very basic introduction to TAP/ADQL. If you want to know more, the following URLs can be useful:

- http://docs.g-vo.org/adql-gaia/html/twoup.pdf
- http://tapvizier.u-strasbg.fr/adql/help.html

More examples of queries can be found by clicking “Examples” in the “Table Access Protocol (TAP) Query” window of TOPCAT.

Moreover, the brown dwarf case made using ADQL can be found at the school web page.