Gaia Data Queries with TAP/ADQL and TOPCAT

Francisco Jiménez-Esteban

Centro de Astrobiología (INTA-CSIC).
Spanish Virtual Observatory, Madrid. Spain.

CENTRO DE ASTROBIOLOGÍA

CSIC
Consejo Superior de Investigaciones Científicas

INTA Instituto Nacional de Técnica Aeroespacial

EXCELENCIA MARÍA DE MAEZTU

ADQL is a dialect of SQL

SQL has been chosen as a base because:

- **Solid theory behind it (relational algebra)**
- **Lots of high-quality engines available**
- **Expressions similar to other programming languages**

  SQRT, POWER, ect….

SELECT

ADQL defines just one statement, the SELECT statement, which lets you write down expressions of relational algebra. Roughly, it looks like this:

SELECT [TOP setLimit] selectList FROM fromClause [WHERE conditions] [GROUP BY columns] [ORDER BY columns]

SELECT

ADQL defines just one statement, the SELECT statement, which lets you write down expressions of relational algebra. Roughly, it looks like this:

SELECT [TOP setLimit] selectList FROM fromClause [WHERE conditions] [GROUP BY columns] [ORDER BY columns]

SELECT: what?
The select list has column names or expressions involving columns.

SELECT

ADQL defines just one statement, the SELECT statement, which lets you write down expressions of relational algebra. Roughly, it looks like this:

SELECT [TOP setLimit] selectList FROM fromClause [WHERE conditions] [GROUP BY columns] [ORDER BY columns]

SELECT: FROM
This specify the table or tables containing the data.

SELECT

ADQL defines just one statement, the SELECT statement, which lets you write down expressions of relational algebra. Roughly, it looks like this:

SELECT [TOP setLimit] selectList FROM fromClause [WHERE conditions] [GROUP BY columns] [ORDER BY columns]

SELECT: WHERE?
Behind the WHERE is a logical expression; these are similar to other languages as well, with operators AND, OR, and NOT.

SELECT

ADQL defines just one statement, the SELECT statement, which lets you write down expressions of relational algebra. Roughly, it looks like this:

SELECT [TOP setLimit] selectList FROM fromClause [WHERE conditions] [GROUP BY columns] [ORDER BY columns]

SELECT: GROUP BY & ORDER BY

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.

SELECT

ADQL defines just one statement, the SELECT statement, which lets you write down expressions of relational algebra. Roughly, it looks like this:

SELECT [TOP setLimit] selectList FROM fromClause [WHERE conditions] [GROUP BY columns] [ORDER BY columns]

SELECT: JOIN

The tricky point in ADQL is the FROM clause when you add more tables: JOIN. It is a combination of Cartesian product and a select.