

Region and STC

- ADQL provides geometry constructs: 2D spatial
 - roughly follows OpenGIS specification
 - REGION(<string>) is the general purpose geometry
 - string format: NOT specified by ADQL
 - consistent with SQL treatment of dates, binary/hex, etc.
 - ADQL independent of serialisation/versions
 - a service that consumes ADQL specifies the standard it supports
 - added footnote that the only current standard is STC
 - note for TAP: this would be metadata for the REGION function specifically

Region and STC

- ADQL provides geometry constructs
 - roughly follows OpenGIS specification
 - REGION(<string>) is the general purpose geometry
 - string format: NOT specified by ADQL
 - consistent with SQL treatment of dates, binary/hex, etc.
 - ADQL independent of serialisation/versions
 - a service that consumes ADQL specifies the standard it supports
 - added footnote that the only current standard is STC
 - note for TAP: this would be metadata for the REGION function specifically

Region and STC

- ADQL also provides some explicit types
 - POINT(<string>, ...)
 - CIRCLE(<string>, ...)
 - POLYGON(<string>, ...)
 - RECTANGLE(<string>, ...)
 - coordinate system string: NOT specified by ADQL
 - an enumerated type: defined by an external standard
 - acceptable values specified by service
 - note for TAP: this would be metadata for these functions

RECTANGLE

- RECTANGLE was supposed to be shorthand for POLYGON
 - two opposite corners, edges all great circles
 - poor name when large and/or away from equator (spherical coordinates only)
 - same concept as BOX in STC
- PROPOSAL: remove? replace?
 - BOX(<string>, center, width, height)
 - edges remain great circles

LONGITUDE/LATITUDE

- late additions to standard
- poorly named as they limit ADQL to spherical
- PROPOSAL: rename to something generic
 - COORD1/COORD2?
 - CVAL1/CVAL2?
- PROPOSAL: add for completeness?
 - COORDSYS(<geometry_expression>)

Note about examples

- in RFC comments:
 - POINT('ICRS', t.ra, t.dec)
 - this is legal in SQL (and ADQL) but dangerous
 - we cannot make ADQL semantically safe
- preferable:
 - use t.position (a column of type POINT)
 - use POINT(t.coordsys, t.ra, t.dec)

cross-match

```
SELECT *  
FROM myTable m JOIN someCatalog c  
ON INTERSECTS(  
    CIRCLE(m.csys,m.ra.m.dec,m.err),c.position  
) = 1
```

- JOIN: qualified
- myTable: job for TAP to enable upload