REGION 1.0

Why is Region in the language?

- important for astronomical queries
  - indexable query predicates are generally expressed as language elements rather than functions in SQL
- region is a type
  - it can be used as a column type or a parameter type to the functions
  - can be considered a binary or blob for consistency with SQL92
  - can actually be implemented that way, not required

REGION values

- can be used as a constant, e.g.:
  - CIRCLE(ICRS, 123.0, 45.0, 0.5)
- can be used with columns, e.g.:
  - CIRCLE(t.coordsys, t.ra, t.dec, t.radius)
- other standard shapes, e.g.:
  - RECTANGLE(coordsys, x1, y1, x2, y2)
  - POLYGON(coordsys, x1, y1, x2, y2, x3, y3[, ...])

Coordinate Systems

- option 1: define constants in the language

  <coordsys> ::= ICRS | GAL | <column_reference>

  - service can use anything internally (column)
  - only standard systems used *on the wire*
- option 2: allow any string, deal with runtime errors

Predicates

- support for two predicates: boolean valued expressions used in WHERE clause:

  CONTAINS(region, coordsys, longitude, latitude)

  INTERSECTS(region1, region2)

Point vs Region -- CONTAINS

- Find all objects in this circle:

```
SELECT * FROM SomeTable t
WHERE CONTAINS(CIRCLE(ICRS, 180, 32.5, 0.5),
      t.coodsys, t.ra, t.dec)
```

- Find all “good” objects (not in masked areas)

```
SELECT * FROM SomeTable t, Mask m
WHERE NOT CONTAINS(m.shape, t.coodsys, t.ra, t.dec)
```

- Is this point in any of the surveys?**

```
SELECT * FROM Surveys s
WHERE CONTAINS(s.region, ICRS, 182.5, 32.1)
```

Region vs Region -- INTERSECTS

- Find observations that overlap this 4x1 degree rectangle

```
SELECT * FROM observation o
WHERE INTERSECTS(o.shape, RECTANGLE(ICRS, 10,
  20, 14, 21))
```

- Find objects with error circles intersecting region

```
SELECT * FROM Catalog c
WHERE INTERSECTS(CIRCLE(c.coordsys, c.longitude,
  c.latitude, c.error), RECTANGLE(...))
```

Other Functions

- AREA(region)?
  - useful in SELECT, WHERE, or ORDER BY
  - implementation could (should) compute and store in column or be able to index computed values
- DISTANCE(long1, lat1, long2, lat2)?
  - useful in SELECT or ORDER BY
  - use in WHERE clause can be duplicated using CONTAINS and CIRCLE

Summary (Reserved Words)

- regions:

  CIRCLE(coordsys, a, b, radius)

  RECTANGLE(coordsys, a1, b1, a2, b2)

  POLYGON(coordsys, a1, b1, a2, b2, a3, b3, ...)

- predicates

  CONTAINS(region, coordsys, a, b)

  INTERSECTS(region1, region2)

- coordinate system constants

  ICRS | GAL