

For the record – we only have one language

The specification defines the vocabulary and the grammar

- Shapes
 - Circle
 - Square
 - Triangle
- Colours
 - Red
 - Green
 - Blue
- Metadata query
 - Do you have <shape> <colour> ?
 - Do you support <colour> ?
- Data request
 - Please send me <count> <shape> <colour>.
- Data response
 - Here are the <count> <shape> <colour> you asked for.
- Error response
 - I'm sorry we don't have <count> <shape> <colour>.
 - I'm sorry, we don't support <colour>.

This is one language, not nine.



STC REGION

- 1) Import the text from TAP 1.0 as an appendix in ADQL 1.2
- 2) Update the text in response to comments
- 3) Good enough for 2.1 ?





BOX

- 1) Rough straw poll indicates no one strongly *needs* it
- 2) Remove BOX from the specification





CIRCLE(POINT, radius)

- 1) Sounds sensible, need to hear from implementors
- 2) Does it **need** to be in 1.2, or can it wait for 1.3 ?





Set operations

Questions about the accuracy of the BNF

- Need to hear from implementors.
- Please send suggested fixes.
- Is it “good enough” for 2.1 ?





CAST

- 1) Happy to drop CAST from 2.1
- 2) If someone really wants CAST, we can add it to 2.2





TIMESTAMP

TIMESTAMP was added to maintain compatibility with VOTable

Constructor syntax was defined to match geom constructors

Add text recommending when to use TIMESTAMP
and when to use more accurate types ?





Boolean functions

The intention was to replace

```
SELECT .. WHERE 1 = CONTAINS(...)
```

With user friendly

```
SELECT .. WHERE CONTAINS(...)
```

Gain – simpler queries

Cost – obscure error messages when we get it wrong

What do we want ?





Bitwise operators

We have been through many iterations of this

- Functions only
- Functions and operators
- Operators only

Keep operators unless there is a strong case against.

Gain – consistent with existing databases and documentation

Cost – adding the operators to our grammar parsers

(* I know the BNF in the document has issues with operator precedence.

Updated BNF proposed on the mailing list fixes the issue.





SQL comparison

Request from implementaors

Appendix listing the differences between ADQL and SQL

Probably not ready for 2.1, bounce to 2.2

PEG grammar in 3.x is more important





SQL comparison

Request from new implementaors

Appendix listing the differences between ADQL and SQL

Probably not ready for 2.1, bounce to 2.2

PEG grammar in 3.x is more important





SQL comparison

Request from new implementors

Appendix listing the differences between ADQL and SQL

Probably not ready for 2.1, bounce to 2.2

PEG grammar in 3.x is more important





String functions

Request from implementors

UPPER
LOWER
STR_LEN
SUB_STR ..

Probably not ready for 2.1, bounce to 2.2

PEG grammar in 3.x is more important





Next steps

Only accepting minor changes for 2.1

Work in progress for 2 independent implementations

Non-trivial changes will get bounced to 2.2

Version 3.x will have PEG grammar

