



## ADQL specification 2.1 and beyond

D.Morris  
Institute for Astronomy,  
Edinburgh University  
May 2016



## ADQL-2.1 – Changes since Sydney #1

- **BOOLEAN** data type

WHERE enabled = True

- **HEX** literals

WHERE flags & 0xFF = 0x01

- **OFFSET** clause

```
SELECT  
  <TOP n>  
  ...  
FROM  
  ...  
WHERE  
  ...  
<OFFSET m>
```



## ADQL-2.1 – Changes since Sydney #2

- Overloaded DISTANCE

```
SELECT
```

```
... .
```

```
FROM
```

```
... .
```

```
WHERE
```

```
  - Coordinate form
```

```
DISTANCE(t1.ra, t1.dec, t2.ra, t2.dec) < 0.01
```

```
OR
```

```
  - Point form
```

```
DISTANCE(t1.p1, t2.p1) < 0.01
```

*Services registering support for DISTANCE means they support BOTH forms.*





## ADQL-2.1 – Are we done ?

Working draft WD-ADQL-2.1-20160502

Available on the IVOA documents page

<http://www.ivoa.net/documents/ADQL/20160502/>

Is this good enough to move to PR ?

Close version 2.1, and anything new goes version 2.2.



## ADQL-2.2 - Overloading geometry functions #1

- We accepted overloaded DISTANCE based on number of params
- We have deprecated COORD-SYS and recommend empty string

Can we have overloaded versions of the geometric constructors without the COORD\_SYS parameter ?

```
BOX(coordsys, ra, dec, width, height)  
BOX(ra, dec, width, height)
```

```
CIRCLE(coordsys, ra, dec, radius)  
CIRCLE(ra, dec, radius)
```

```
POINT(coordsys, ra, dec)  
POINT(ra, dec)
```



## ADQL-2.2 - Overloading geometry functions #2

- We accepted overloaded DISTANCE based on number of params
- We have deprecated COORD-SYS and recommend empty string

Can we have versions of the geometric constructors that accept POINT parameters ?

```
BOX(coordsys, ra, dec, width, height)
BOX(ra, dec, width, height)
BOX(Point, width, height)
```

```
CIRCLE(coordsys, ra, dec, radius)
CIRCLE(ra, dec, radius)
CIRCLE(Point, radius)
```

```
POINT(coordsys, ra, dec)
POINT(ra, dec)
```



## ADQL-2.2 - Overloading geometry functions #3

- We accepted overloaded DISTANCE based on number of params
- We have deprecated COORD-SYS and recommend empty string

What do we do with POLYGON ?

If expressions can resolve to either numeric or Point,  
then overloading on number of params does not work.

Is this four coordinates or four Points ?

POLYGON(a, b, c, d)

Can we use different names ?

POLYCOORD(a, b, c, d)

POLYPOINT(a, b, c, d)



## ADQL-2.2 – Point literal

Use {} brackets to indicate a Point literal.

```
CIRCLE(POINT(a, b), 7)
```

becomes

```
CIRCLE({a, b}, 7)
```

Benefits

- Smaller and more concise

Costs

- Departure from SQL syntax







## ADQL-2.2 – BLOBs

Add CLOB and BLOB data types to the ADQL specification.

Add explicit clauses to the text and the BNF to restrict where they can occur.

- Allowed in the SELECT fields but not in the WHERE clause.
- Allowed in the WHERE clause but only as function parameters.





## ADQL-2.2 – Value expression in ORDER BY

The BNF for ADQL 2.1 only allows column name or column number in the ORDER BY and GROUP BY clauses.

```
ORDER BY <column_name> | <unsigned_decimal>  
GROUP BY <column_name> | <unsigned_decimal>
```

We have specific requests from end users to be able to have the same value expressions in the ORDER BY and GROUP BY clauses as the SELECT fields.

Initial testing suggests this is supported on all of the database platforms we have looked at so far.

The only restriction is that the ORDER BY or GROUP BY expression must be the same as one of the SELECT fields.

```
SELECT  
    ra,  
    dec,  
    count(*)  
FROM  
    ....  
WHERE  
    ....  
ORDER BY  
    count(*)  
GROUP BY  
    count(*)
```





## ADQL-2.1 – Are we done ?

Working draft WD-ADQL-2.1-20160502

Available on the IVOA documents page

<http://www.ivoa.net/documents/ADQL/20160502/>

Are any of these simple enough to include in version 2.1 ?

- Geometric constructors without COORD\_SYS
- Geometric constructors with Point
- What do we do with POLYGON ?

Propose we close version 2.1, and anything else goes in version 2.2.





## Unnamed columns #1

Unnamed columns occur when a query SELECT field contains an expression or function call, and does not have an alias.

```
SELECT
    ra + 2
FROM
    table
WHERE
    . . . .
```

- What should the column name be ?
- Should the response be uniform across the whole of the IVOA ?
- Should the behavior be defined in the ADQL or TAP standards ?
- Currently this is not defined in TAP/ADQL.



Is this important enough to  
delay 2.1 until we have a  
solution ?



## Unnamed columns #2

Initial survey of database platforms and TAP services shows a range of different solutions.

```
SELECT
    ra + 2
FROM
    table
WHERE
    . . . .
```

- Column with no name : "
- Simple name generator : COL\_1
- Random name generator : ahblwwnbtm
- Wrap the expression in quotes : "ra + 2"



Is this important enough to  
delay 2.1 until we have a  
solution ?



## Duplicate columns

Duplicate column names occur when a query selects the same fields from two or more tables.

```
SELECT
    T1.ra,
    T1.dec,
    T2.ra
    T2.dec
FROM
    T1, T2
WHERE
    . . . .
```

- What should the column name be ?
- Should the response be uniform across the whole of the IVOA ?
- Should the behavior be defined in the ADQL or TAP standards ?
- Currently this is not defined in TAP/ADQL.





## Round trip - query, download, upload, query

Use case :

A user runs a TAP/ADQL query, downloads the results and uploads them as the input for another query.

What happens if the original results contain unnamed or duplicate columns ?



Is this important enough to  
delay 2.1 until we have a  
solution ?



## Round trip - query, download, upload, query

Use case :

A user runs a TAP/ADQL query, downloads the results and uploads them as the input for another query.

What happens if the original results contain unnamed or duplicate columns ?

End user would expect to be able to use the results from one service as upload input to another (without having to edit the VOTable).

Should the standards require that downloaded results from one TAP service be valid for use as TAP upload and ADQL query in another ?

If so, does that change how we handle unnamed or duplicate columns in our results ?







## Round trip - query, download, upload, query

Use case :

A user runs a TAP/ADQL query, downloads the results and uploads them as the input for another query.

Does this mean new rules on column names in the results ?

