

ADQL-2.1

Grégory Mantelet¹

¹CDS (Centre de Données astronomiques de Strasbourg)

Semi Hack-A-Thon ASOV - 23 Mars 2021



Université

de Strasbourg

□ 1.a. État général

- **1^{er} draft:** Mai 2016
- **PR** depuis Janvier 2018
- **État actuel:** ajouts et modifications terminés
- **Prochaine étape:** RFC



1.b. En bref

Feature	Status
UNION / INTERSECT / EXCEPT	✓
CAST(...) + TIMESTAMP(...)	✓
Optional features	✓
WITH (or Common Table Expression = CTE)	✓
OFFSET	✓
Recommended x-match syntax (now DISTANCE(...))	✓
Grouping/Ordering by an expression	✓
ILIKE	✓
LOWER(...), UPPER(...)	✓
IN_UNIT(...)	✓
Fix MOD(...), RAND(...), ROUND(...) and TRUNCATE(...)	✓
Deprecation of BOX	✓
Geometry constructor with "exploded" points	✓
Deprecation of coord. sys. argument + COORDSYS(...)	✓
UDF Catalogue	✓
Clarifications on datatypes	✓
BLOB and CLOB	✓
Hexadecimal values/operations	Ⓛ
Bitwise operators	Ⓛ
Boolean datatype	Ⓛ
Interval functions (intersects/overlaps)	Ⓛ

□ 2.a. Syntaxe de cross-match

~~ADQL-2.0:~~

```
1 = CONTAINS( POINT(' ', ra, dec),  
              CIRCLE(' ', 1.0, 2.0, 1./60.))
```

ADQL-2.1+:

```
DISTANCE(ra, dec, 1.0, 2.0) < 1./60.
```

□ 2.b. UNION, ...

```
SELECT id, ra, dec FROM table1
UNION
SELECT id, ra, dec FROM table2
INTERSECT
SELECT id, ra, dec FROM table3
```

□ 2.c. WITH

```
WITH tenFirst AS ( SELECT TOP 10 id, ra, dec
                    FROM atable
                    ORDER BY id ASC),
     tenLast  AS ( SELECT TOP 10 id, ra, dec
                    FROM atable
                    ORDER BY id DESC)
SELECT * FROM tenFirst
UNION
SELECT * FROM tenLast
```

□ 2.d. IN_UNIT

```
IN_UNIT(sqrt(power(pm_ra,2)+power(pm_dec,2)), 'rad/yr')
```

□ 2.e. Régions et coord. sys.

- **Construction de région:**
 - Constructeurs: POINT/CIRCLE/POLYGON(...)
 - Fonction REGION('...')
 - sérialisation: STC/s par défaut (*possiblement d'autres selon implémentation*)
 - paramètre: chaîne littérale (=> *pas de colonne*)
 - Fonction CAST(... AS POINT/CIRCLE/POLYGON)
(*sérialisation DALI et STC/S*)
 - sérialisation: DALI par défaut, STC/s autorisée (*possiblement d'autres selon implémentation*)
 - paramètre: chaîne littérale, colonne ou expression
- **Support/Conversion de système de coordonnées:**
 - 1^{er} paramètre des constructeurs déprécié
 - Par défaut, celui de la base de données
 - *Info à préciser dans les métadonnées de TAP ?*
 - Pour conversion, utiliser une UDF (*cf "f. Catalogue d'UDFs"*)

□ 2.f. Catalogue d'UDFs

Proposed Endorsed Note - Catalogue of ADQL User Defined Functions, *Jon Juaristi Campillo, et al.*

- **Objectif:** regrouper la définition (surtout la signature) des UDFs les plus fréquentes afin d'encourager leur usage dans les implémentations nouvelles ou déjà existantes.
- **État:** actuellement encore en [RFC](#)

□ 3. Prochaines modifications ?

- Grammaire PEG
- Tableaux
- Time + Interval (*cf DALI*)
- MOC
- Type booléen
- Notation hexadecimal + Opérations binaires
- Annotation d'unité
- Fonction `substring(...)` (*semble déjà existée dans SQL-92*)
- Géométries:
 - *cf OGC standard (OpenGIS® Implementation Standard for Geographic information)*
 - DISTANCE entre d'autres géométries que seulement des POINTs
 - définition de régions complexes par le biais d'intersections

□ 4.a. Implémentations d'ADQL

- **CADC** : ADQL-2.0
- **DACHS** : ADQL-2.1 (*à ajuster*)
- **VOLLT/ADQL-Lib** : ADQL-2.1 (*en cours de finalisation*)

□ 4.b. Collaboration

- GitHub project (*Issues, PullRequests, and Projects overview*)
- dal@ivoa.net (*email*)
- #adql channel on SLACK (*chat*)