Accéder aux données de Gaia DR1

I-Chun Shih et Nicolas Leclerc

Vendredi 16 septembre 2016

Qu'est ce que ADQL ?

- Astronomical Data Query Language
- IVOA (International Virtual Observatory Alliance), version 2 d'avril 2008
- Repose sur la norme SQL 92 (Structured Query Language) : un langage d'interrogation de base de données
- Ajoute des fonctionnalités spatiales à SQL :

area, centroid, circle, contains, distance, intersects, latitude, longitude, point, polygon, rectangle, region

Le SQL : SELECT ... FROM pour extraire des données

Pour extraire des données d'une table, il faut 2 informations :

- Ce que vous voulez extraire (SELECT)
- D'où vous voulez l'extraire (FROM)

```
Select source_id, ra, dec
from gaiadr1.gaia_source;
```

```
Select *
from gaiadr1.gaia_source;
```

l'opérateur *, qui retourne toutes les colonnes de la table, fait perdre du temps. Ne l'utiliser qu'en cas de nécessité

Le SQL : ORDER BY pour trier les données extraites

ORDER BY permet de trier les données et doit se situer en fin de requête

- Possible d'utiliser plusieurs colonnes
- Par défaut ascendant (asc), mais descendant possible (desc)

```
Select source_id, ra, dec
from gaiadr1.gaia_source
order by source_id;
```

```
Select source_id, ra, dec
from gaiadr1.gaia_source
order by source_id, ra, dec;
```

```
Select ra, dec, souce_id
from gaiadr1.gaia_source
order by 3, 1;
```

Le SQL : WHERE pour filtrer les données

Voici les opérateurs décris dans la documentation ADQL :

- Les opérateurs logique standard : AND, OR, NOT
- Les opérateurs de comparaison : =, !=, <>, <, >, <=, >=
- BETWEEN
- LIKE
- NULL
- EXISTS

Le SQL : WHERE (suite)

- NOT inverse le résultat de la condition qui suit
- AND est prioritaire par rapport à OR source_id = "1" OR source_id = "2" AND dec_error < 0.1
  est différent de :
  (source_id = "1" OR source_id = "2") AND dec_error < 0.1
- L'opérateur IN est plus rapide qu'une série de OR
- L'opérateur LIKE est puissant, mais peut être très lent :
  - "_" remplace un caractère
  - "%" remplace plusieurs caractères
  - "[ ]" permets de spécifier des ensembles
    (ex : [JM%] retourne tout les champs commençant par JM)

Le SQL : Créer des champs calculés

- Opérateur de concaténation || (ex : SELECT '(' || ra || ', ' || dec || ')')
- Calculs arithmétiques + - x /
- Fonctions mathématiques : cos, sin, abs ...
- Fonctions statistiques : avg, count, max, min, sum
- Pour count, si la colonne est précisée, les NULL ne sont pas comptés
- Les alias avec AS, permettent de donner un nom à un champ calculé (ex gaiadr1.gaia_source AS g)

http ://www.w3schools.com/sql/sql_functions.asp

Le SQL : GROUP BY pour grouper les données

- GROUP BY peut contenir un nombre illimité de colonnes
- Pas d'opérateur d'agrégation (COUNT, AVG, MIN ...) dans le GROUP BY
- Excepté l'agrégation, toutes les colonnes dans le SELECT doivent être dans le GROUP BY
- Si NULL est présent, il formera un groupe
- Pour filtrer les GROUP BY, il faut utiliser HAVING

Le SQL : GROUP BY exemple

Que contient un source_id codé sur 64 bits ?

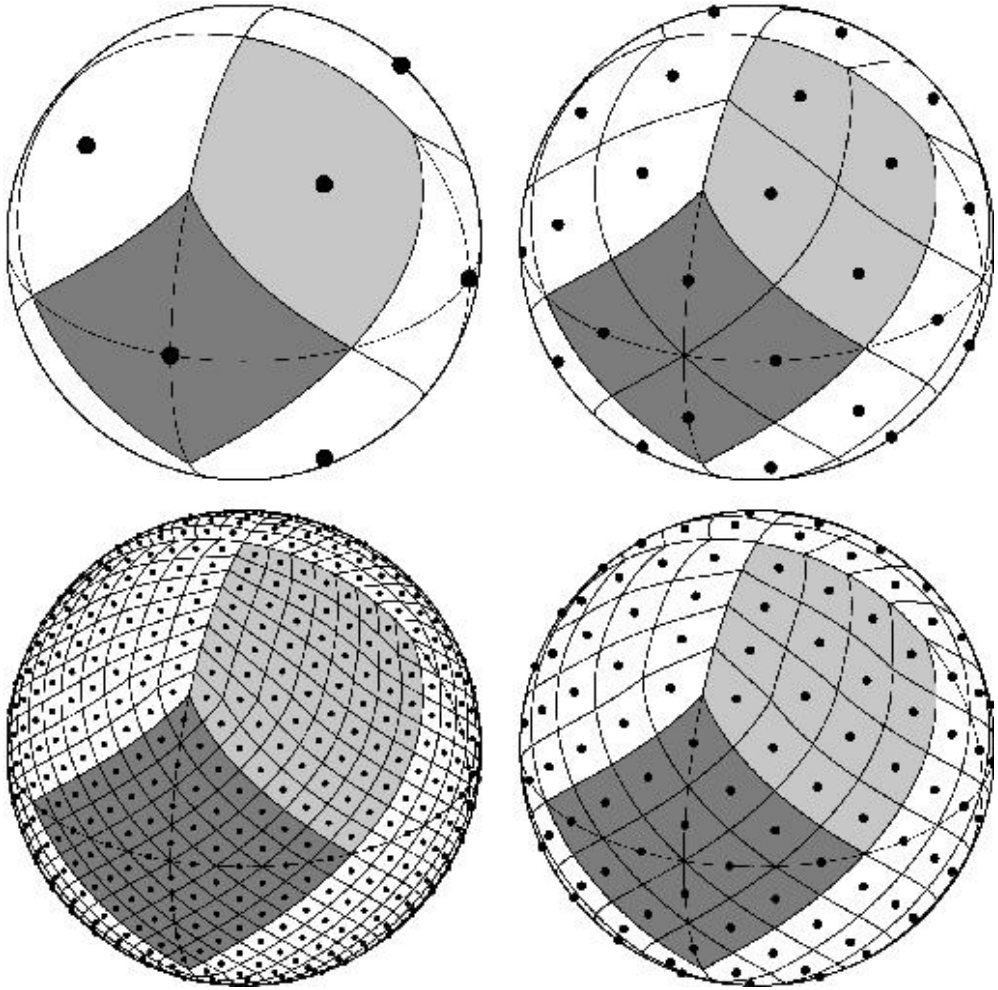
- Healpix (Hierarchical Equal Area isoLatitude Pixelisation) d'ordre 12 (201 326 592 Npix)
- Code DPC
- Numéro de création dans le Healpix
- Numéro de la source

Pour chaque cellule d'un Healpix de niveau 7, compter le nombre d'étoile et la valeur moyenne de ra, dec :

```
SELECT source_id/35184372088832 as hpx7,
       count(*) as gaia_nbsrc,
       avg(ra) as ra,
       avg(dec) as dec
FROM gaiadr1.tgas_source
GROUP BY hpx7
HAVING count(*) > 1
ORDER BY hpx7 asc
```

Le SQL : GROUP BY exemple (2)

A quoi ressemble un Healpix :



Le SQL : GROUP BY exemple (3)

Comprendre source_id/35184372088832 :
Le Healpix d'ordre 12 est codé dans le source_id à partir du 35^e bit, ainsi $source\_id \gg 35$ ou $\frac{source\_id}{2^{35}}$ retourne la valeur du Healpix d'ordre 12.

Ordre	*Nside*	*Npix*
1	2	48
2	4	192
3	8	768
4	16	3072
5	32	12288
6	64	49152
7	128	**196608**
8	256	786432
9	512	3145728
10	1024	12582912
11	2048	50331648
12	4096	**201326592**

Avec $\mathbf{Nside} = \mathbf{2^{Ordre}}$ et
$\mathbf{Npix} = \mathbf{12} \times \mathbf{Nside^2}$

Pour passer de l'ordre 12 à l'ordre 7, il faut $\frac{201326592}{196608} = 1024 = 2^{10}$

Donc pour l'ordre 7 : $\frac{source\_id}{2^{45}}$ avec $2^{45} = 35184372088832$

Le SQL : Travailler avec des sous-requêtes

La requête traitée en premier est celle qui est la plus à l'intérieur

```
SELECT hip, b_v, e_b_v
FROM gaiadr1.hipparcos
WHERE hip IN (SELECT hip
   FROM gaiadr1.tgas_source
   WHERE parallax/parallax_error > 5) ;
```

```
SELECT h.hip, h.b_v, h.e_b_v
FROM gaiadr1.hipparcos AS h,
   gaiadr1.tgas_source AS g
WHERE h.hip = g.hip
   AND g.parallax/g.parallax_error > 5
```

```
SELECT h.hip, h.b_v, h.e_b_v
FROM gaiadr1.hipparcos AS h
      INNER JOIN gaiadr1.tgas_source AS g
      ON h.hip = g.hip
WHERE g.parallax/g.parallax_error > 5
```

Le SQL : Les jointures