

VizieR plots : objectifs & état des lieux



Objectifs :

- Améliorer le travail de documentation en offrant une interface plus intuitive et plus simple.
- Améliorer l'ergonomie de la page web résultat

État des lieux :

- Les plots sont le plus souvent issues de fichier FITS (utilisation de fits2a qui permet de transformer les données de type pixels en positions). D'autres formats sont possibles : TSV, ASCII, base de données, données issues de commandes...
- Aujourd'hui les plots sont générés par un SHELL script :
 - AWK, graph (plotutil)
 - sous-scripts (ex : lc_mag.sh)

Architecture et applications

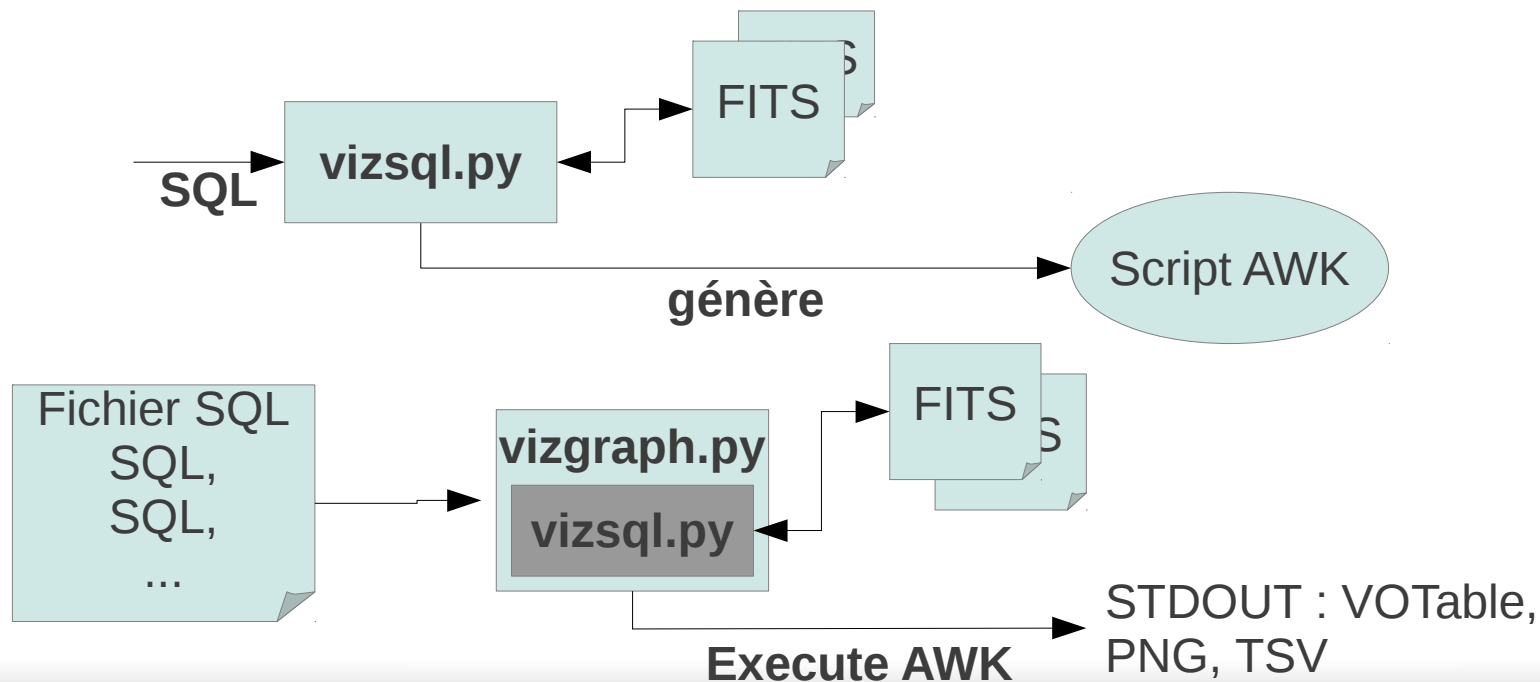


Utilisation d'un **pseudo-SQL** utilisé pour extraire les données.

vizgraph.py : extraction des données à partir d'un fichier contenant les requêtes SQL.

vizsql.py : génère un script AWK à partir d'une requête SQL

Note : Les scripts utilisent le programme "fits2a" pour les fichier FITS !



Accès aux plots via l'interface web

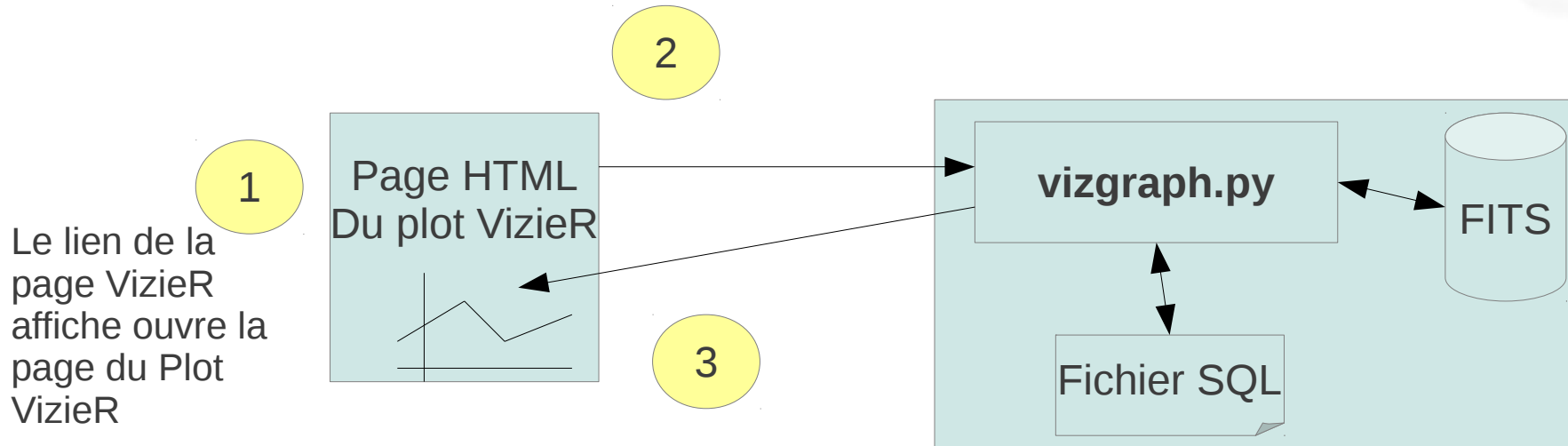
Le même système de liens affiche les Plots VizieR. La page cible étant cette fois une page HTML au lieu d'un CGI.

- Pour les petites volumétries: utilisation des Widgets (Anne-Camille+Thomas)
- Pour les grandes volumétries : le plot est exécuté sur le serveur qui renvoie l'image
- (Éventuellement) dans le cas de Plot associé à une table entière, le plot peut être stocké sur le filesystem.

Accès au plots via l'interface web



La page HTML exécute une requête HTTP (javascript) pour récupérer les données sur le serveur



Le javascript de la page HTML génère le graph à partir des données reçues du serveur

Que peut on faire ?



A propos de vizsql.py

- Comprend les instructions SQL simples :

ex : **SELECT** \$1, \$2 **FROM** fichier **WHERE** \$1<100

- Capable de travailler sur des fichiers FITS, TSV, ASCII ou sur le résultat d'une commande UNIX

ex : **SELECT** \$1, \$2 **FROM** "fichier.fits"

SELECT \$1 **FROM** "asu -source=...."

- Plusieurs contraintes possibles

ex : **SELECT** \$1 **FROM** fichier **WHERE** \$1>10 **AND** \$1<11

- Possibilité de faire des opérations arithmétiques :

ex : **SELECT** \$1*2+\$3 **FROM** fichier **WHERE** log(\$2)+1>10

- Possibilité de joindre deux fichiers :

ex : **SELECT** a.\$1, b.\$2 **FROM** "fichier1.fits" a, "fichier.tsv" b **WHERE** a.\$1=b.\$1

- Un graphe peut être composé de plusieurs plot (ou requête SQL)

Que peut on faire ?



Pour les fichiers FITS :

→ Utilisation de "fits2a"

→ Possibilité de choisir la sous-partie du FITS :

ex : `SELECT a.$0.$1 FROM "fichier.fits" a`

(sélection de la colonne 1 de la 1ere sous-partie (=\$0))

ex : `SELECT a.$1.$1 FROM "fichier.fits" a`

(sélection de la colonne 1 de la 2ème sous-partie (=\$1))

Les limitations du pseudo-SQL



Certaines opérations SQL sont impossibles :

- Joindre plus de 2 fichiers
- Les fonctions d'agrégations ne sont pas possibles (avg, sum, count...)
- L'ordre (ORDER BY) et le regroupement (GROUP BY) ne sont pas possibles
- La syntaxe INNER JOIN, LEFT|RIGHT JOIN n'est pas comprise !
- Le 'OR' dans la contrainte n'est pas disponible ('AND' uniquement)

Limitation en terme de capacité :

- La jointure est lente surtout si la jointure comporte une opération arithmétique (pas d'indexation)
- La jointure nécessite de la mémoire
- Le Widget de graphe n'est pas efficace pour les grandes volumétries !
- Le nombre de requêtes SQL dans le fichier SQL est statique (→ problématique avec le catalogue III/45)

Interagir avec le graphe



→ On peut ajouter le nom des axes en utilisant l'instruction SQL "AS"

ex : **SELECT \$1 AS ra, \$2 AS dec FROM**

→ Le titre du plot est indiqué dans le fichier SQL :

ex de ligne SQL :

nom_du_graphe1 : SELECT

nom_du_graphe2 : SELECT

→ Le type du graphe :

→ 2 champs sont sélectionnés : plot de points

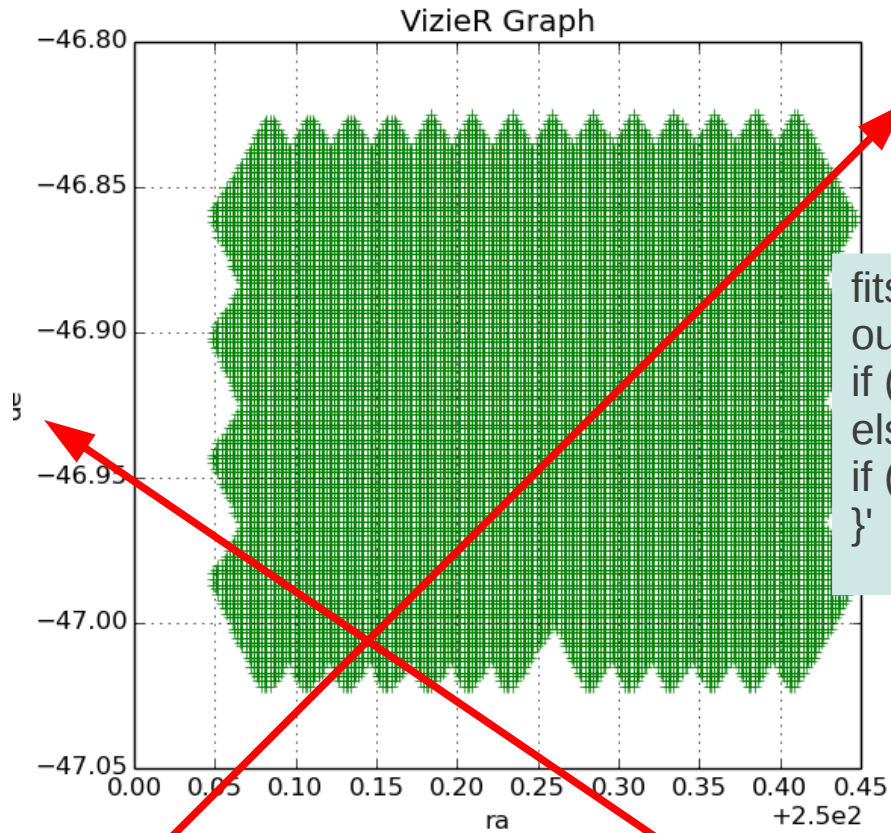
→ 3 champs sont sélectionnés : plot avec barre d'erreur

→ 4 champs sont sélectionnés : plot avec barre d'erreur sur X et Y

Exemple de plots



Catalogue : II/243



```
fits2a -b -d -xten 0 "./rasters/new.08700441.fits.gz"| gawk '{\n  out=1;\n  if ($3>0) {\n  else {out=0}\n  if (out==1){print $1"\t"$2}\n  }'
```

vizsql.py

```
image: SELECT $1 as ra, $2 as de FROM "rasters/${table}" WHERE $3>0
```

Exemple de plots


(plusieurs graph pour 1 catalogue)

Catalogue : II/262

- Un graphe illustrant les filtres utilisés sur la table II/262/summary
- Un SED pour chaque record de la table II/262/batc

[II/262/summary](#) [BATC Data Release One - BATC DR1 \(Zhou+ 1995-2005\)](#) [ReadMe+ftp](#)
[Post annotation](#) Basic information for the following files ( [filter characteristics](#)) (70 rows) [filter](#) 

<u>Full</u>	<u>RAJ2000</u> "h:m:s"	<u>DEJ2000</u> "d:m:s"	<u>Field</u>	<u>RAJ2000</u> "h:m:s"	<u>DEJ2000</u> "d:m:s"	<u>FN</u>	<u>Nsrc</u>	<u>Filter</u>
<u>1</u>	12 30 49.31	+12 23 42.4	N023	12 30 49.31	+12 23 42.4	14	4990	.bcdefghijklmnop
<u>2</u>	12 29 45.81	+08 00 00.3	N024	12 29 45.81	+08 00 00.3	13	3234	.bcde.ghijklmnop

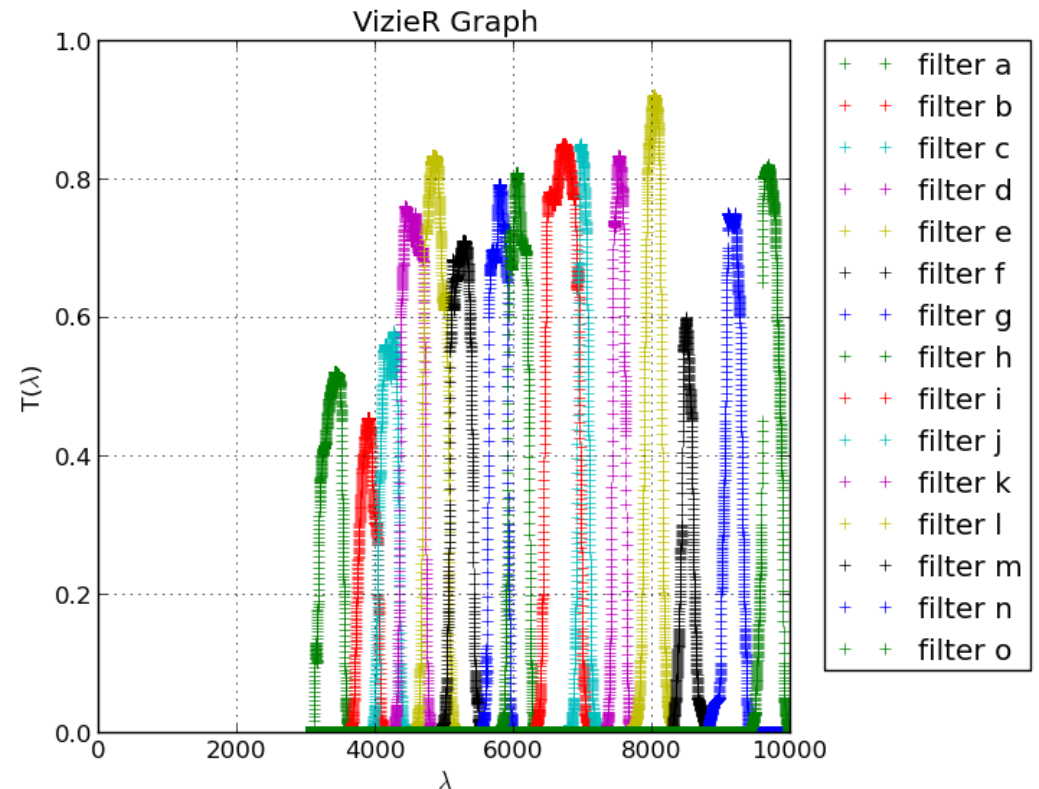
[II/262/batc](#) [BATC Data Release One - BATC DR1 \(Zhou+ 1995-2005\)](#) [ReadMe+ftp](#)
[Post annotation](#) The contents of the catalog (64932 rows) [spectrum/SED](#) 

<u>Full</u>	<u>RAJ2000</u> "h:m:s"	<u>DEJ2000</u> "d:m:s"	<u>Field</u>	<u>RAJ2000</u> "h:m:s"	<u>DEJ2000</u> "d:m:s"	<u>Rem</u>	<u>Seq</u>	<u>cMag</u> mag	<u>eMag</u> mag	<u>gMag</u> mag	<u>iMag</u> mag	<u>kMag</u> mag	<u>nMag</u> mag	<u>oMag</u> mag	<u>pMag</u> mag	<u>SED</u>
<u>1</u>	12 29 13.55	+12 33 29.4	N023	12 29 13.55	+12 33 29.4		1	20.64		21.17	22.13	20.58		19.90	18.48	SED
<u>2</u>	12 29 04.53	+12 45 50.9	N023	12 29 04.53	+12 45 50.9		2	20.22			21.13	19.67	19.47	18.72	18.98	SED

Exemple de plots (plusieurs plots pour 1 catalogue)

Catalogue : II/262 – graphe des filtres (table II/262/summary)

```
filter a: SELECT $1 as '$\lambda$', $2 as 'T($\lambda$)' FROM "filters.dat"  
filter b: SELECT $1 as '$\lambda$', $3 as 'T($\lambda$)' FROM "filters.dat"  
filter c: SELECT $1 as '$\lambda$', $4 as 'T($\lambda$)' FROM "filters.dat"  
filter d: SELECT $1 as '$\lambda$', $5 as 'T($\lambda$)' FROM "filters.dat"  
filter e: SELECT $1 as '$\lambda$', $6 as 'T($\lambda$)' FROM "filters.dat"  
filter f: SELECT $1 as '$\lambda$', $7 as 'T($\lambda$)' FROM "filters.dat"  
filter g: SELECT $1 as '$\lambda$', $8 as 'T($\lambda$)' FROM "filters.dat"  
filter h: SELECT $1 as '$\lambda$', $9 as 'T($\lambda$)' FROM "filters.dat"  
filter i: SELECT $1 as '$\lambda$', $10 as 'T($\lambda$)' FROM "filters.dat"  
filter j: SELECT $1 as '$\lambda$', $11 as 'T($\lambda$)' FROM "filters.dat"  
filter k: SELECT $1 as '$\lambda$', $12 as 'T($\lambda$)' FROM "filters.dat"  
filter l: SELECT $1 as '$\lambda$', $13 as 'T($\lambda$)' FROM "filters.dat"  
filter m: SELECT $1 as '$\lambda$', $14 as 'T($\lambda$)' FROM "filters.dat"  
filter n: SELECT $1 as '$\lambda$', $15 as 'T($\lambda$)' FROM "filters.dat"  
filter o: SELECT $1 as '$\lambda$', $16 as 'T($\lambda$)' FROM "filters.dat"
```



Exemple de plots (concaténation)

Catalogue : II/262 – SED (table II/262/batc)

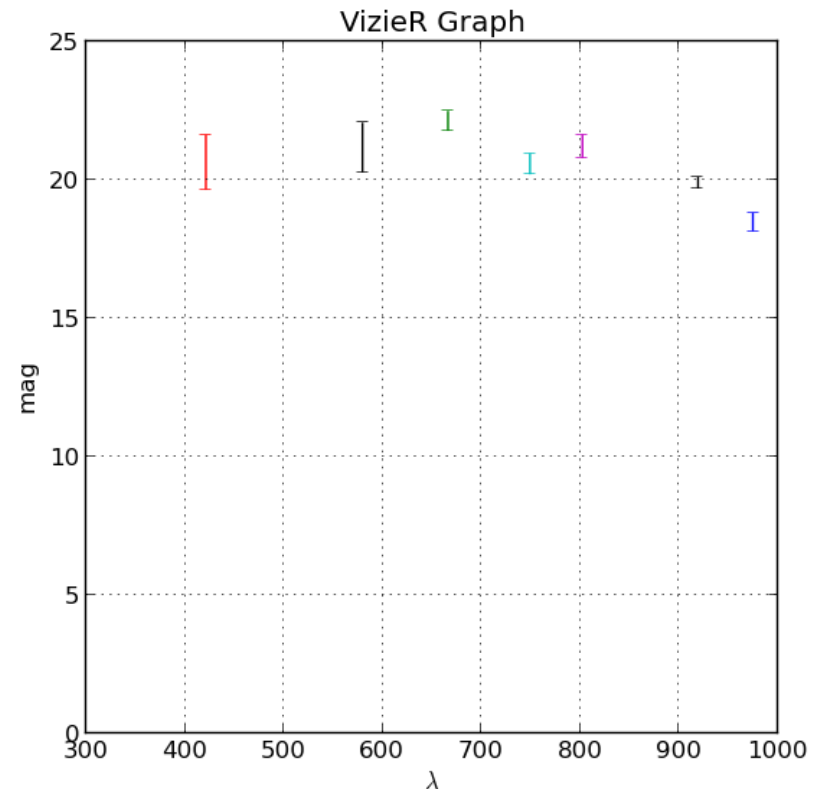
```

set filterchar="gawk 'BEGIN{pline=0;n=0}{if (/^Filter char/){pline=1}if(pline==1){if (/---/){n+=1}else if (n==2){print
$0}}}' /home/cats/II/262/ReadMe"
SED_a: SELECT $2 as '$\lambda$' FROM "${filterchar}" WHERE $1='a'
SED_a: SELECT $2 as mag, $3 as err FROM "asu -source=II/262/batc recno=${recno} -out=recno,aMag,e_aMag"
SED_b: SELECT $2 as '$\lambda$' FROM "${filterchar}" WHERE $1='b'
SED_b: SELECT $2 as mag, $3 as err FROM "asu -source=II/262/batc recno=${recno} -out=recno,bMag,e_bMag"
SED_c: SELECT $2 as '$\lambda$' FROM "${filterchar}" WHERE $1='c'
SED_c: SELECT $2 as mag, $3 as err FROM "asu -source=II/262/batc recno=${recno} -out=recno,cMag,e_cMag"
SED_d: SELECT $2 as '$\lambda$' FROM "${filterchar}" WHERE $1='d'
SED_d: SELECT $2 as mag, $3 as err FROM "asu -source=II/262/batc recno=${recno} -out=recno,dMag,e_dMag"
SED_e: SELECT $2 as '$\lambda$' FROM "${filterchar}" WHERE $1='e'
SED_e: SELECT $2 as mag, $3 as err FROM "asu -source=II/262/batc recno=${recno} -out=recno,eMag,e_eMag"
...
    
```

Filter characteristics:

Table
`${filterchar}` →

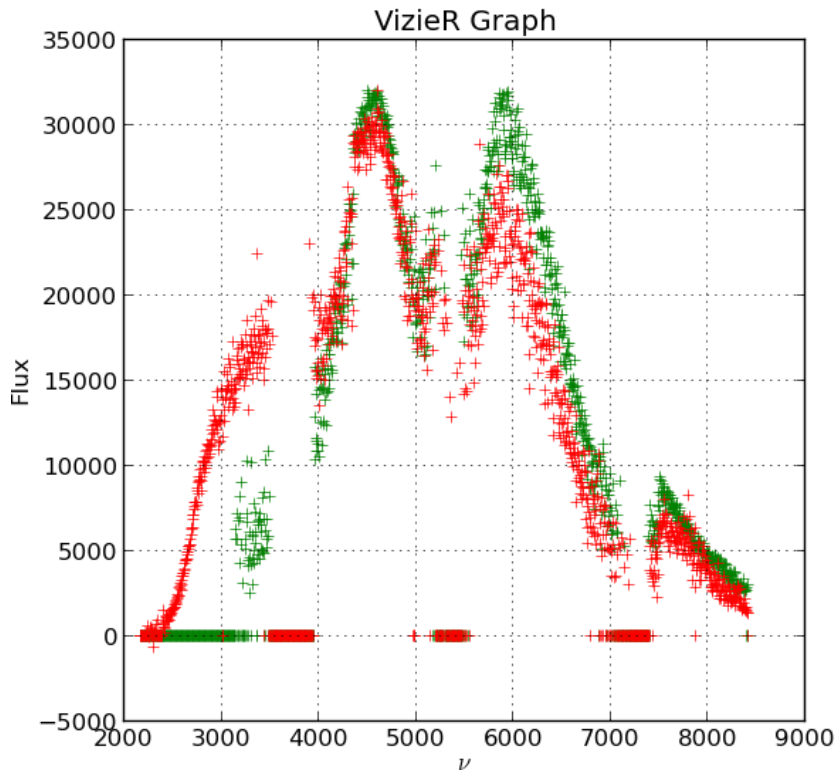
Filter Code	cw(nm)	FWHM(nm)
a	336.0	36
b	389.0	34
C	421.0	32
.....		



Exemple de plots



Catalogue : III/45



```
fcat "/home/cats/III/45/sp19.dat"|
tail -n +1|cut -c1-11,12- |sed -n 's/ *\t/p'| gawk '{
out=1;
if (out==1){print $0"\t"$1}
}'

fcat "/home/cats/III/45/sp20.dat"|
tail -n +1|cut -c1-11,12- |sed -n 's/ *\t/p'| gawk '{
out=1;
if (out==1){print $0"\t"$1}
}'
```

```
spectra1: SELECT $1 as '$\nu$', $2 as Flux FROM "sp${theTab}.dat"
spectra2: SELECT $1 as '$\nu$', $2 as Flux FROM "sp${theTab2}.dat"
```

Exemple de plots



Catalogue : III/45

<u>Full</u>	<u>RAJ2000</u>	<u>DEJ2000</u>	<u>RAB2000</u>	<u>DEB2000</u>	<u>Names</u>	<u>Files</u>
	"h:m:s"	"d:m:s"	"h:m:s"	"d:m:s"		
<u>1</u>	02 19 20.8	-02 58 28	02 19 20.7	-02 58 28	HR 681 = HD 14386 = omicron Cet	<u>19,20</u>
<u>2</u>	03 05 10.1	+38 50 30	03 05 10.0	+38 50 30	HR 921 = HD 19058 = rho Per	<u>14</u>
<u>3</u>	04 35 55.1	+16 30 43	04 35 55.0	+16 30 43	HR 1457 = HD 29139 = alpha Tau	<u>07,08</u>
<u>4</u>	05 05 23.7	+01 10 39	05 05 23.6	+01 10 39	HR 1648 = HD 32736 = W Ori	<u>36</u>
<u>5</u>	05 55 10.3	+07 24 25	05 55 10.2	+07 24 25	HR 2061 = HD 39801 = alpha Ori	<u>25,26</u>
<u>6</u>	06 14 53.0	+22 30 25	06 14 52.9	+22 30 25	HR 2216 = HD 42995 = eta Gem	<u>12</u>
<u>7</u>	06 22 57.5	+22 30 54	06 22 57.4	+22 30 54	HR 2286 = HD 44478 = mu Gem	<u>13</u>
<u>8</u>	06 36 32.8	+38 26 44	06 36 32.8	+38 26 44	HR 2405 = HD 46687 = UU Aur	<u>35</u>
<u>9</u>	06 45 10.9	-16 41 58	06 45 10.8	-16 41 58	HR 2491 = HD 48915 = alpha CMa	<u>02,03</u>
<u>10</u>	07 22 58.3	-25 46 03	07 22 58.2	-25 46 03	HR 58061 = VV CMa	<u>34</u>

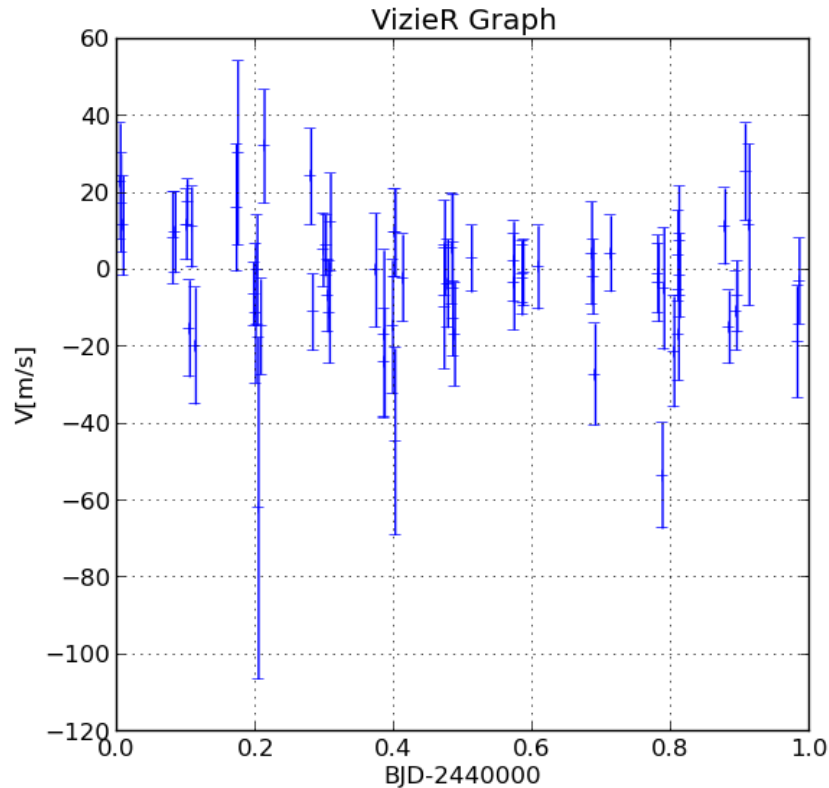
Problème : le nombre de fichiers FITS est variable !...

```
spectra1: SELECT $1 as '$\nu$', $2 as Flux FROM "sp${theTab}.dat"  
spectra2: SELECT $1 as '$\nu$', $2 as Flux FROM "sp${theTab2}.dat"
```

Exemple de plots (avec barre d'erreurs)



Catalogue : III/185



```
asu -data -source=III/185/table1 -out=JD-2440000,RV,e_RV HR==509 | gawk '{\n  out=1;\n  if (out==1){print ($1/10)-int($1/10)"\t"$2"\t"$3}\n}'
```

```
Set asu="asu -source=III/185/table1 -out=JD-2440000,RV,e_RV"
```

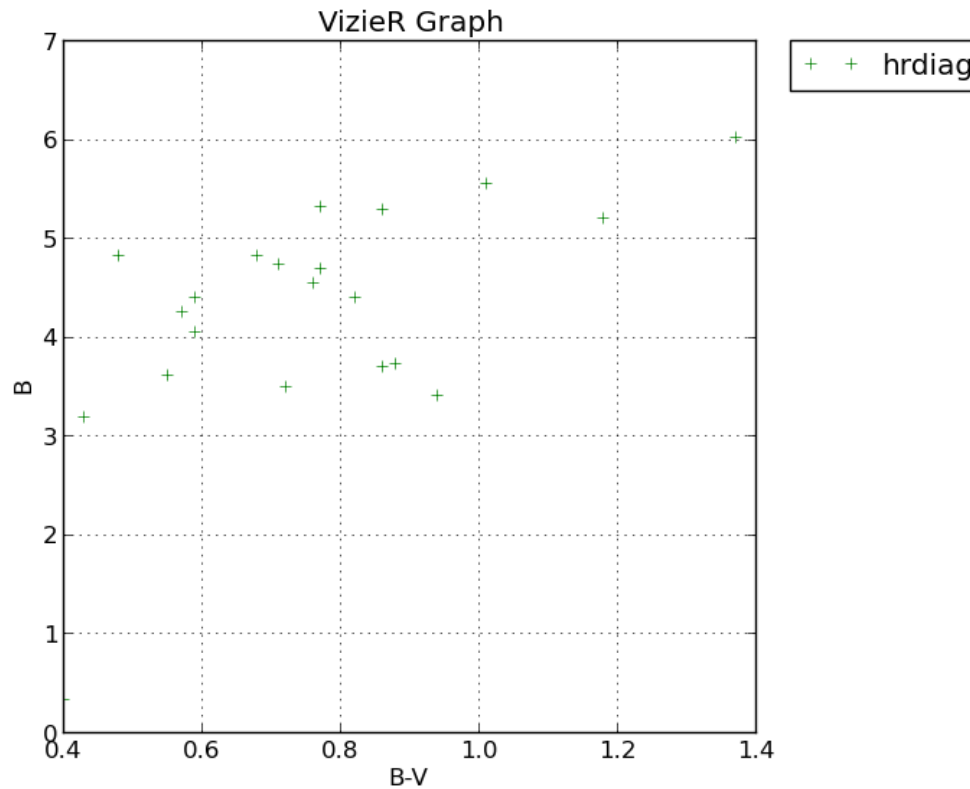
```
Light_curves: SELECT ($1/10)-int($1/10) as 'Phase', $2 as 'V[m/s]', $3 FROM "${asu} HR==${HR}"
```


Exemple de plots



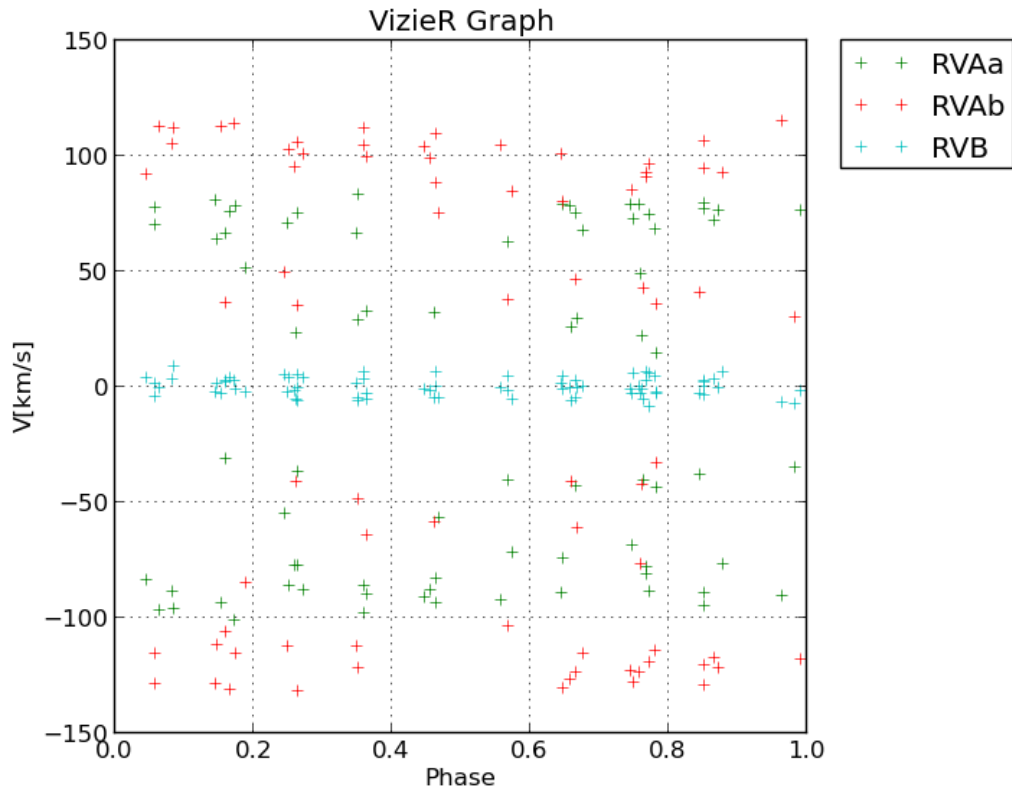
Catalogue : III/185 – diagram HR

```
set asu="asu -source=III/185/stars -out=Bmag,Vmag"  
hrdiag: SELECT $1-$2 as 'B-V', $2 as B from "${asu}"
```



Exemple de plots

Catalogue : J/APJ/640/1018 - table1



```
asu -data -n- -source=J/ApJ/640/1018/table1
-out=HJD,RVAa| gawk '{\
out=1;\
if (out==1){print $1/10-int($1/10)"\t"$2}\
}'
asu -data -n- -source=J/ApJ/640/1018/table1
-out=HJD,RVAb| gawk '{\
out=1;\
if (out==1){print $1/10-int($1/10)"\t"$2}\
}'
asu -data -n- -source=J/ApJ/640/1018/table1
-out=HJD,RVB| gawk '{\
out=1;\
if (out==1){print $1/10-int($1/10)"\t"$2}\
}'
```

```
set asu="asu -n- -source=J/ApJ/640/1018/table1"
```

```
RVAa:SELECT $1/10-int($1/10) as Phase, $2 as 'V[km/s]' from "${asu} -out=HJD,RVAa"
```

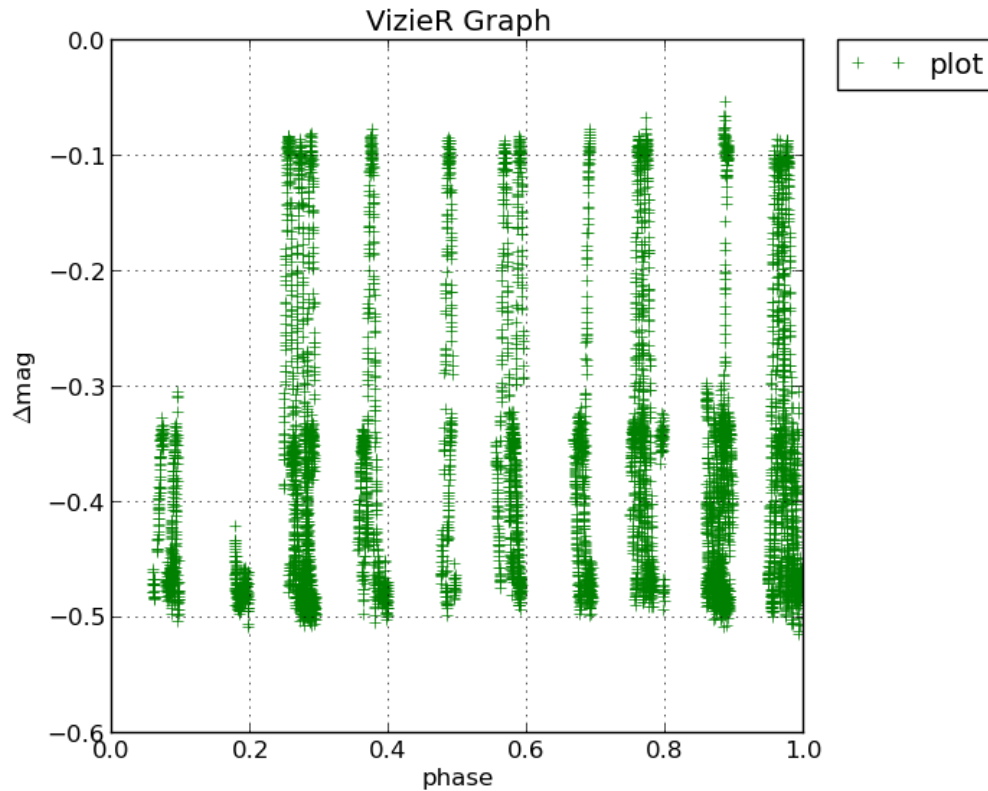
```
RVAb:SELECT $1/10-int($1/10) as Phase, $2 as 'V[km/s]' from "${asu} -out=HJD,RVAb"
```

```
RVB: SELECT $1/10-int($1/10) as Phase, $2 as 'V[km/s]' from "${asu} -out=HJD,RVB"
```

Exemple de plots



Catalogue : J/APJ/640/1018 – tableX, X>1..

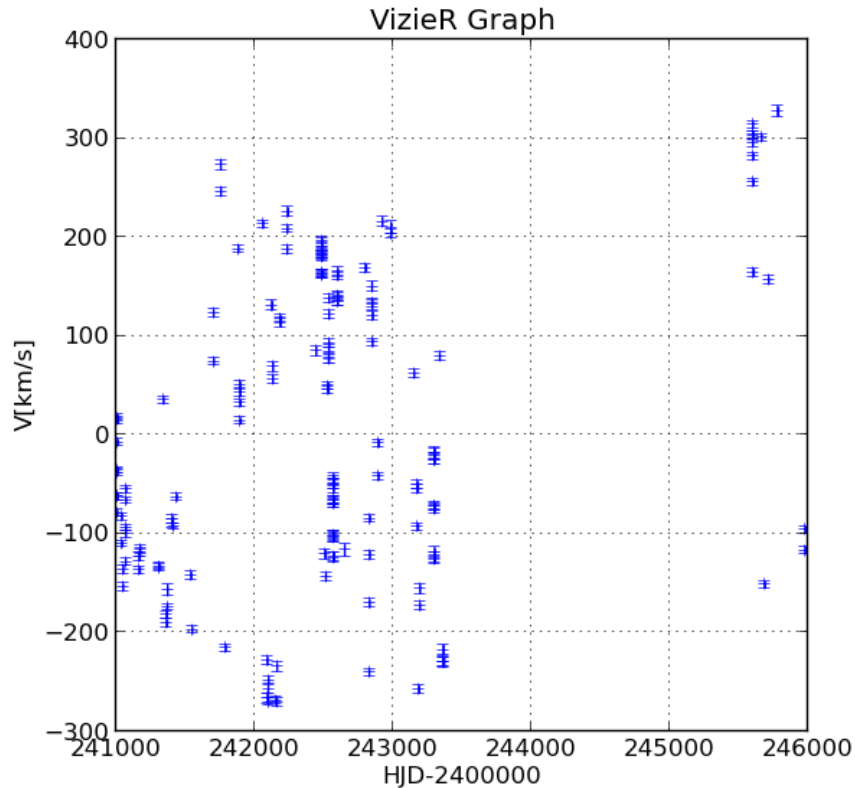


```
set asu="asu -source=J/APJ/640/1018"  
plot: SELECT $1/($P)-int($1/$P) as phase, $2 as '$\Delta$mag' FROM "${asu}/${theTab}" -out=HJD,Delmag"
```

Exemple de plots



Catalogue : J/APJ/634/625 - table1



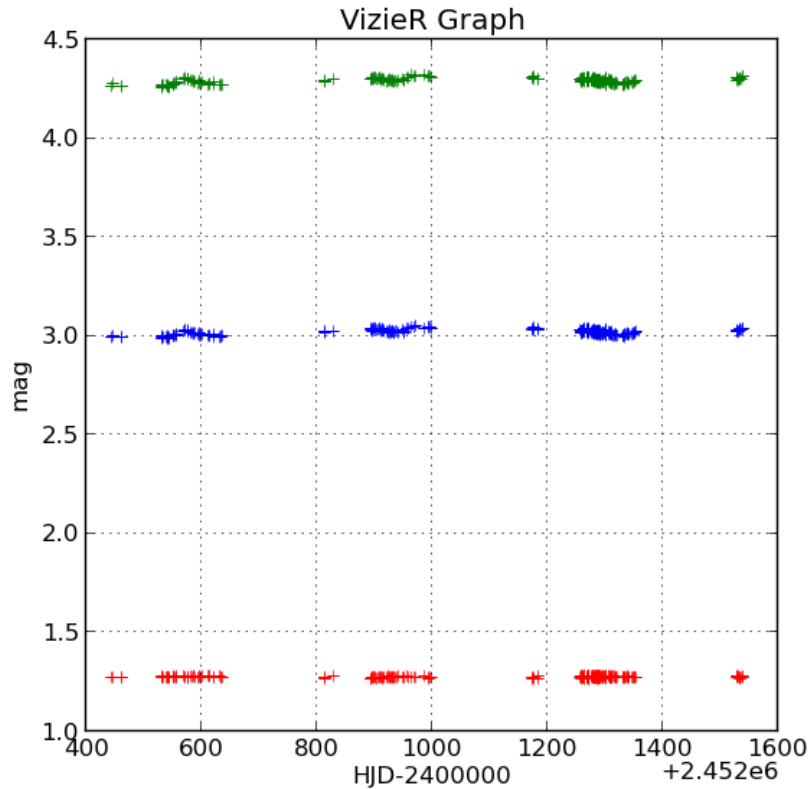
```
asu -data -n- -source=J/ApJ/634/625/table1  
-out=HJD,RV,e_RV| gawk '{\  
out=1;\  
if (out==1){print $1"\t"$2"\t"$3}\  
}'
```

```
set asu="asu -n- -source=J/ApJ/634/625/table1 -out=HJD,RV,e_RV"  
velocity_curve: SELECT $1 as 'HJD-2400000',$2 as 'V[km/s]', $3 FROM "${asu}"
```

Exemple de plots



Catalogue : J/APJ/634/625 - table5



```
set asu="asu -source=J/ApJ/634/625/table5"
```

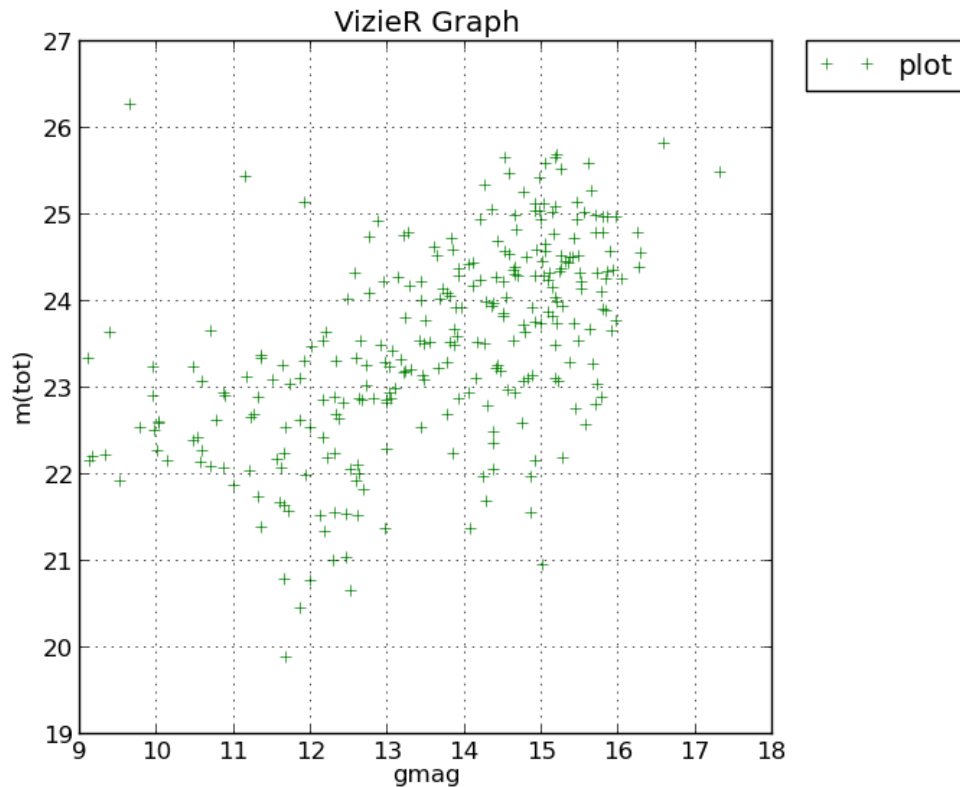
```
P-C1: SELECT $1 as 'HJD-2400000', $2 as mag, $3 FROM "${asu} -out=HJD,P-C1"
```

```
P-C2: SELECT $1 as 'HJD-2400000', $2 as mag, $3 FROM "${asu} -out=HJD,P-C2"
```

```
C1-C2:SELECT $1 as 'HJD-2400000', $2 as mag, $3 FROM "${asu} -out=HJD,C1-C2"
```

Exemple de SQL - jointures

```
set vcc="asu -source=VII/266/vcc -out=VCC,gmagT"  
set bdd="asu -source=VII/266/bdd_g -out=VCC,mu50"  
plot: SELECT vcc.$2 as gmag, bdd.$2 as mu50 FROM "${vcc}" vcc, "${bdd}" bdd WHERE vcc.$1=bdd.$1
```



La jointure peut se faire sur n'importe quel type de ressource : commande asu, fichier ASCII, FITS, etc...

METAdonnées



Les METAdonnées propres aux plots sont comme aujourd'hui stipules dans les champs d'explication des colonnes (METAcou) ou des tables (METAtab).

Travail du documentaliste :

1) Écriture du fichier SQL

2) Remplir .status pour exécuter le fichier SQL en utilisant une nouvelle marque. **A définir !**

Ex : `\Plot{nom_fichier}{table}{param liste}`

`\Plot{.graph}{I/234/table1}{ID=${ID}&Num=${Num}}`

Note : possibilité d'ajouter des macros comme : **(TODO)**

`\Plot-line-curve, \Plot-sed, \Plot-time-serie, ?????`

Exemple : <http://cdsarc/viz-bin/lightcurve?-s=II/247&-i=graph1&-d=true&theTab=1/3319.10>

Le script vizsql :

Exemple : `vizsql II/243 'select $1 as x, $2 as y from "rasters/new.08700441.fits"`

(attention : ne pas oublier les simple quotes ' ... ')

TODO



- Ajouter les macro \plotsed,...
- Traiter le cas SQL : OR
- Traiter le cas du catalogue III/45 ?? (nombre de fits variables..)
- Catalogue J/A+A/558/A18 : inversion des colonnes ?