VO2016

Virtual Observations 2016

WERKCOLLEGE 1: MYSQL

Document identifier:	**VO2016-W1-03**
Date:	**April 11, 2016**
Activity:	
Document status:	
Document link:	

Abstract: This werkcollege introduces mySQL server and gives a first glance on SQL

1. TASKS

Tasks:

start mysql server

create a database

download data from 2MASS and USNO-A2

ingest data into database

cross-identify 2MASS and USNO-A2 regions

2. START MYSQL SERVER

In your home directory (`/Users/users/belikov` through this example) create a subdirectory for MySQL data storage (for example, `data`). Copy configuration file from `/etc/my.cnf`

```
>cp /etc/my.cnf ~/.
```

Edit configuration file. The file should include 4 variables pointing to your home directory. An example:

```
[mysqld_safe]
datadir=/Users/users/belikov/data
socket=/Users/users/belikov/mysql.sock
log-error=/Users/users/belikov/mysqld.log
pid-file=/Users/users/belikov/mysqld.pid
```

Please, note that you have to put full path to your home directory (`/Users/users/belikov`). Do not use relative paths here!

Create service files for your MySQL server:

```
>mysql_install_db --datadir=/Users/users/belikov/data
        --defaults-file=/Users/users/belikov/my.cnf
```

This command should be executed with two parameters: `datadir`, which specify the path to the directory with the data and should be the same as `datadir` in `my.cnf` file, and the full path to your configuration file `my.cnf` in `defaults-file`.

Finally, launch MySQL server with command

```
>mysqld_safe --defaults-file=/Users/users/belikov/my.cnf &
```

and check that it is running

```
>ls /Users/users/belikov/mysql.sock
```

If you can not start `mysqld_safe` due to the TCP port 3306 occupied by other application add port into `my.cfg`:

```
[mysqld_safe]
datadir=/Users/users/belikov/data
socket=/Users/users/belikov/mysql.sock
log-error=/Users/users/belikov/mysqld.log
pid-file=/Users/users/belikov/mysqld.pid
port=<any port, for example, 9999>
```

3. CONNECT TO MYSQL SERVER AND CREATE A DATABASE

To connect to MySQL in this example we will use mysql command line interface (CLI). You should specify a socket which you will use to connect to the server (as it was specified in `my.cnf` in `socket`):

```
>mysql --socket=/Users/users/belikov/mysql.sock -u root
```

With mysql you can execute SQL statements, use `\g` to finish the statement, `\q` to leave mysql. The simple session is shown below.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67 SUSE MySQL RPM

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>create database prob\g
Query OK, 1 row affected (0.03 sec)

mysql>show databases \g
+--------------------+
```

```
| Database           |
+--------------------+
| mysql              |
| prob               |
| test               |
+--------------------+
3 rows in set (0.06 sec)

mysql> use prob \g
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

4. STOPPING MYSQL SERVER

Before you are logging out stop mysql server with following command:

```
>mysqladmin --socket="path_to_your_socket" -u root shutdown
```

not to lose any data.

5. LOAD TEST DATA FROM VIZIER

Using the web interface at `http://vizier.u-strasbg.fr` select 1 degree area from 2MASS and USNO-A2 and save the data in `"|"-separated` format. Please, note that you have to do this for each catalog separately, so that at the end you have two different files with following header for 2 MASS:

```
#
#   VizieR Astronomical Server: vizier.u-strasbg.fr     2010-02-12T13:51:38
#   (replaces the 'Astrores' format originally described at
#    http://vizier.u-strasbg.fr/doc/astrores.htx)
#   In case of problem, please report to:       question@simbad.u-strasbg.fr
#
#
#Coosys J2000:  eq_FK5 J2000
#INFO   -ref=VIZ4b755c65dc3
#INFO   -out.max=unlimited
#INFO   -c=000.000000+30.000000,rm=60.

#RESOURCE=yCat_2246
#Name: II/246
#Title: 2MASS All-Sky Catalog of Point Sources (Cutri+ 2003)
#Table  II_246_out:
#Name: II/246/out
#Title: The Point Source catalogue of 470,992,970 sources. % Additions to 2MASS Intro (last modif: 27-Nov-2007
#%---------------------------------------==========
#Please {\bf
#\aW{ack}{http://www.ipac.caltech.edu/2mass/releases/allsky/doc/sec1_8b.html}
# {acknowledge the usage of the 2MASS All-Sky Survey}};
# see also the \W{ext}{http://www.ipac.caltech.edu/2mass/}{2MASS Pages}.
#\ifnum\Vphase>3{\fg{red3}\quad
#Note that the magnitudes in red correspond to low quality results
#(upper limits or very poor photometry)}\fi
#
#Column _r      (F8.6)  Distance from center (RAJ2000=000.000000, DEJ2000=+30.000000)   [ucd=pos.angDistance]
#Column _RAJ2000        (F10.6) Right ascension (FK5) Equinox=J2000. (computed by VizieR, not part of the orig
#Column _DEJ2000        (F10.6) Declination (FK5) Equinox=J2000. (computed by VizieR, not part of the original
#Column RAJ2000 (F10.6) (ra) Right ascension (J2000)    [ucd=pos.eq.ra]
#Column DEJ2000 (F10.6) (dec) Declination (J2000) (dec) [ucd=pos.eq.dec]
#Column 2MASS   (a17)   (designation) Source designation (1)    [ucd=meta.id;meta.main]
#Column Jmag    (F6.3)  ?(j_m) J selected default magnitude (2) [ucd=phot.mag;em.IR.J]
#Column e_Jmag  (F6.3)  ?(j_msigcom) J total magnitude uncertainty (4)  [ucd=stat.error;phot.mag;em.IR.J]
#Column Hmag    (F6.3)  ?(h_m) H selected default magnitude (2) [ucd=phot.mag;em.IR.H]
#Column e_Hmag  (F6.3)  ?(h_msigcom) H total magnitude uncertainty (4)  [ucd=stat.error;phot.mag;em.IR.H]
#Column Kmag    (F6.3)  ?(k_m) K selected default magnitude (2) [ucd=phot.mag;em.IR.K]
#Column e_Kmag  (F6.3)  ?(k_msigcom) K total magnitude uncertainty (4)  [ucd=stat.error;phot.mag;em.IR.K]
#Column Qflg    (A3)    (ph_qual) JHK Photometric quality flag (5)      [ucd=meta.code.qual;phot]
#Column Rflg    (A3)    (rd_flg) Source of JHK default mag (6)  [ucd=meta.ref]
#Column Bflg    (A3)    (bl_flg) JHK components fit to source (7)       [ucd=meta.code]
#Column Cflg    (A3)    (cc_flg) Artifact contamination, confusion (8)  [ucd=meta.code]
#Column Xflg    (I1)    [0,2] (gal_contam) Extended source contamination (10)   [ucd=meta.code]
#Column Aflg    (I1)    [0,1] (mp_flg) Association with asteroid or comet (11)  [ucd=meta.code]
_r|_RAJ2000|_DEJ2000|RAJ2000|DEJ2000|2MASS|Jmag|e_Jmag|Hmag|e_Hmag|Kmag|e_Kmag|Qflg|Rflg|Bflg|Cflg|Xflg|Aflg
deg|deg|deg|deg|deg||mag|mag|mag|mag|mag|mag||||||
```

and for USNO-A2

```
#
#   VizieR Astronomical Server: vizier.u-strasbg.fr     2010-02-12T13:53:03
#   (replaces the 'Astrores' format originally described at
#    http://vizier.u-strasbg.fr/doc/astrores.htx)
#   In case of problem, please report to:       question@simbad.u-strasbg.fr
#
#
#Coosys J2000:  eq_FK5 J2000
#INFO   -ref=VIZ4b755d002ab0
#INFO   -out.max=unlimited
#INFO   -c=000.000000+30.000000,rm=60.

#RESOURCE=yCat_1252
#Name: I/252
```

```
#Title: The USNO-A2.0 Catalogue (Monet+ 1998)
#Table  I_252_out:
#Name: I/252/out
#Title: The Full Catalogue
#Column _r      (F8.6)  Distance from center (RAJ2000=000.000000, DEJ2000=+30.000000)   [ucd=pos.angDistance]
#Column _RAJ2000        (F10.6) Right ascension (FK5) Equinox=J2000. (computed by VizieR, not part of the orig
#Column _DEJ2000        (F10.6) Declination (FK5) Equinox=J2000. (computed by VizieR, not part of the original
#Column USNO-A2.0       (a13)   Original designation in USNO-A2.0 catalogue     [ucd=meta.id;meta.main]
#Column RAJ2000 (F10.6) Right ascension (ICRS) mean of blue/red plates  [ucd=pos.eq.ra]
#Column DEJ2000 (F10.6) Declination (ICRS) mean of blue/red plates      [ucd=pos.eq.dec]
#Column ACTflag (A1)    [A] 'A' when star belongs to ACT (Cat. <I/246>) [ucd=meta.code]
#Column Mflag   (A1)    [*] '*' magnitudes are probably wrong   [ucd=meta.code]
#Column Bmag    (F4.1)  Magnitude from blue plate       [ucd=phot.mag;em.opt.B]
#Column Rmag    (F4.1)  Magnitude from red plate        [ucd=phot.mag;em.opt.R]
#Column Epoch   (F8.3)  ? Mean epoch of position        [ucd=time.epoch]
_r|_RAJ2000|_DEJ2000|USNO-A2.0|RAJ2000|DEJ2000|ACTflag|Mflag|Bmag|Rmag|Epoch
deg|deg|deg||deg|deg|||mag|mag|yr
```

check the number of columns and their order. Do not forget to set "Number of entries" to "unlimited" before submitting the request.

6. INGEST DATA IN MYSQL

Prepare file for ingestion removing the header with format. The new data file should start with the data with record, separated by delimiter, i.e.,

```
0.013723|000.012392|+30.008553|000.012392|+30.008553|00000297+3000307 |
15.090| 0.048|14.600| 0.068|14.669| 0.120|AAB|222|111|000|0|0
```

Use python program to ingest the data (all programs are available on page of the course[1]:

```
import MySQLdb
import csv

def pN(inp):
 if len(str(inp).strip())==0:
  return -99.99
 return inp


conn = MySQLdb.connect(host = "localhost",user = "root",passwd = "",
                       unix_socket="/home/belikov/mysql/mysql.sock",
                       read_default_file="/home/belikov/mysql/my.cnf")
cursor = conn.cursor ()
cursor.execute ("USE PROB")
cursor.execute ("DROP TABLE TWOMASS")
create_str ="CREATE TABLE TWOMASS (ID INT,RA2000 DOUBLE,DEC2000 DOUBLE,TWOMASSID CHAR(18),
                                   JMAG DOUBLE,EJMAG DOUBLE,HMAG DOUBLE,EHMAG DOUBLE,
                                   KMAG DOUBLE,EKMAG DOUBLE,Qflg CHAR(3),Rflg CHAR(3),
                                   Bflg CHAR(3),Cflg CHAR(3),Xflg INT, Aflg INT, PRIMARY KEY (ID))"

cursor.execute (create_str)


inp_file=csv.reader(open('2mass.data'),delimiter="|",quoting=csv.QUOTE_NONE)
i=0
for row in inp_file:
  if len(row)==0: break
  i=i+1
  ins_str="INSERT INTO TWOMASS VALUES(%i,%s,%s,\'%s\',%s,%s,%s,%s,%s,%s,
          \'%s\',\'%s\',\'%s\',\'%s\',%s,%s)" %
        (i,row[3],row[4],row[5],pN(row[6]),pN(row[7]),pN(row[8]),pN(row[9]),
        pN(row[10]),pN(row[11]),row[12],row[13],
       row[14],row[15],row[16],row[17])
```

[1] http://www.astro.rug.nl/∼belikov/VO2016

```
  print ins_str
  cursor.execute(ins_str)

cursor.close ()
conn.close ()
```

This program uses packages MySQLdb and csv. First you have to create a connection with your database (`conn=MySQLdb.connect()`, then with established connection create a cursor (`cursor=conn.cursor()`) and use this cursor to execute SQL statements (verbcursor.execute()).

Change all parameters (`unix_socket`,`read_default_file`, filename in `cvs`,`reader(open(filename))`) to your parameters. If you create the table for the first time, omit the statement

```
cursor.execute("DROP TABLE TWOMASS")
```

Repeat the same for USNO

7. FAST DATA LOAD/UNLOAD

The data ingest row-by-row is time consuming. There is a way to load data with an DBMS optimized loader. To do this create a new table for 2MASS data, which will correspond exactly to the format of the input file.

```
>cat > 2mass_load.sql
CREATE TABLE TWOMASS2 (RD      DOUBLE,
                       RAVIZIER DOUBLE,
                       DECVIZIER DOUBLE,
                       RA2000  DOUBLE,
                       DEC2000 DOUBLE,
                       TWOMASSID CHAR(18),
                       JMAG DOUBLE,
                       EJMAG DOUBLE,
                       HMAG DOUBLE,
                       EHMAG DOUBLE,
                       KMAG DOUBLE,
                       EKMAG DOUBLE,
                       Qflg CHAR(3),
                       Rflg CHAR(3),
                       Bflg CHAR(3),
                       Cflg CHAR(3),
                       Xflg INT,
                       Aflg INT)
```

Execute this statement with mysql CLI:

```
>mysql --socket=/Users/users/belikov/mysql.sock -u root prob < 2mass_load.sql
```

The general form of the command for the data load from file is

```
LOAD DATA [LOW_PRIORITY | CONCURRENT] [LOCAL] INFILE 'file_name'
    [REPLACE | IGNORE]
    INTO TABLE tbl_name
    [CHARACTER SET charset_name]
    [{FIELDS | COLUMNS}
        [TERMINATED BY 'string']
        [[OPTIONALLY] ENCLOSED BY 'char']
        [ESCAPED BY 'char']
    ]
    [LINES
        [STARTING BY 'string']
        [TERMINATED BY 'string']
    ]
    [IGNORE number LINES]
    [(col_name_or_user_var,...)]
    [SET col_name = expr,...]
```

or, as in this example:

```
>mysql --socket=/Users/users/belikov/mysql.sock -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.0.67 SUSE MySQL RPM

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>use prob\g
mysql>LOAD DATA INFILE '/Users/users/belikov/2mass.data' INTO TABLE TWOMASS2 FIELDS TERMINATED BY
"|"\g
```

Note the difference between the load of the data with this statement and python program.

The complementary command for data unload is

```
SELECT (col_name,col_name,...) INTO OUTFILE 'file_name'
  [FIELDS TERMINATED BY 'char'] [OPTIONALLY ENCLOSED BY 'char']
  [LINES TERMINATED BY 'char']
  FROM tbl_name
```

8. CROSS-IDENTIFICATION

Wtite SQL statement to create CROSSID table, take into account that you have to create one primary key and two foreign keys in this table:

```
CREATE TABLE CROSSID (CROSSID INT,
                      IDTWOMASS INT,
                      IDUSNOA2 INT,
                      PRIMARY KEY (CROSSID),
                      FOREIGN KEY (IDTWOMASS) REFERENCES TWOMASS(ID),
                      FOREIGN KEY (IDUSNOA2) REFERENCES USNOA2(ID))
```

To cross-identify between these catalogs will be used a simplest cross-identification by coordinates with the check of the distance between 2 sources in each catalog, i.e,

```
(COS(RA1)*COS(RA2)+SIN(RA1)*SIN(RA2))*COS(DEC1)*COS(DEC2)+SIN(DEC1)*SIN(DEC2) > COS(ALPHA)
```

where (RA1,DEC1) and (RA2,DEC2) are coordinates of two objects and ALPHA is a cross-identification radius. If the angle between objects less than cross-identification radius, they are marked as cross-identified, if the angle is bigger than ALPHA, they are treated as 2 different objects. **Please, note that the use of spherical trigonometry for cross-identification is extremely inefficient in the case of big datasets and should be avoided.**

Use `cross_id.py` to create a table for cross-identification and populate this table.

```
import MySQLdb

conn = MySQLdb.connect(host = "localhost",
                       user = "root",
                       passwd = "",
                       unix_socket="/home/belikov/mysql/mysql.sock",
                       read_default_file="/home/belikov/mysql/my.cnf")
conn2 = MySQLdb.connect(host = "localhost",
                        user = "root",
                        passwd = "",
                        db="PROB",
                        unix_socket="/home/belikov/mysql/mysql.sock",
                        read_default_file="/home/belikov/mysql/my.cnf")
conn3 = MySQLdb.connect(host = "localhost",
                        user = "root",
                        passwd = "",
                        db="PROB",
                        unix_socket="/home/belikov/mysql/mysql.sock",
                        read_default_file="/home/belikov/mysql/my.cnf")
cursor  = conn.cursor()
cursor_in = conn3.cursor()
cursor.execute ("USE PROB")
cursor.execute ("DROP TABLE CROSSID")
create_str ="CREATE TABLE CROSSID (CROSSID INT, IDTWOMASS INT, IDUSNOA2 INT,
             PRIMARY KEY (CROSSID),
             FOREIGN KEY (IDTWOMASS) REFERENCES TWOMASS(ID), FOREIGN KEY (IDUSNOA2) REFERENCES USNOA2(ID))"

cursor.execute (create_str)

cursor.execute ("SELECT ID, RA2000, DEC2000 FROM TWOMASS")


sr=1.0/60.0/60.0*15.0

id_cross=1
while(1):
 row=cursor.fetchone()
 if row==None: break
 cursor2=conn2.cursor()
 select_str="SELECT ID FROM USNOA2 WHERE (COS(RADIANS(%s))*COS(RADIANS(RA2000))+
    SIN(RADIANS(%s))*SIN(RADIANS(RA2000)))*COS(RADIANS(%s))*COS(RADIANS(DEC2000))+
    SIN(RADIANS(%s))*SIN(RADIANS(DEC2000)) > COS(RADIANS(%s))" % (row[1],row[1],row[2],row[2],sr)
 print select_str
 cursor2.execute(select_str)
```

```
 row2=cursor2.fetchall()
 if len(row2)>0:
  for newid in row2:
   ingest_statement="INSERT INTO CROSSID VALUES(%s,%s,%s)" % (id_cross,row[0],newid[0])
   print ingest_statement
   cursor_in.execute(ingest_statement)
   print id_cross, row[0], newid
   id_cross=id_cross+1
 cursor2.close()

cursor.close ()
cursor_in.close()
conn.close ()
conn2.close()
conn3.close()
```

Change all parameters in connect() to your ones. As result you will fill in the cross-identification tables with pairs (IDTWOMASS, IDUSNOA2). Please, note that the search radius in the program is 15 arcsec.

9. WORK WITH SQLITE

SQLite is a Python library which allows to work with SQL-like database stored in the file.

To start working with sqlite import it in Python

```
>python
Python 2.7.3 (default, Apr 14 2012, 08:58:41) [GCC] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>import sqlite3
```

Create a file with database

```
>>>conn=sqlite3.connect("test.db")
```

You can execute SQL statements via cursor

```
>>>cursor=conn.cursor()
>>>cursor.execute('''CREATE TABLE test (test1 DOUBLE, test2 INTEGER, test3 CHAR(10))''')
```

Repeat exersize with crossidentification createing tables in sqlite, loading data, createing cross-id table. Please, note that number of functions implemented in SQLite is very limited.

10. FIND CARDINALITY OF CROSS-IDENTIFIED PAIRS

In the case of cross-identification you did you will have a number of pairs with multiple sources, i.e., to single star from 2MASS corresponds a number of stars from USNO-A2 (and vice versa).

Find how many 1-1 cross-identifications, 1-2 crossidentifications and non-cross-identified stars you have.

Hint: use mysql and SELECT SQL statement with GROUP BY and HAVING classes.

```
SELECT CROSSID, IDTWOMASS, IDUSNOA2 FROM CROSSID GROUP BY IDTWOMASS HAVING COUNT(IDTWOMASS) = 1
```

Please, note that you are grouping records by 2MASS identifiers. You are counting how many times the same 2MASS identifier appears in the CROSSID table.

11. SELECTING SUBSET WITH ALL MAGNITUDES FROM CROSS-IDENTIFIED TABLE

```
SELECT T1.CROSSID, T1.IDTWOMASS, T1.IDUSNOA2, T2.JMAG, T2.HMAG, T2.KMAG, T3.BMAG, T3.RMAG
      FROM CROSSID T1, TWOMASS T2, USNOA2 T3
      WHERE T2.ID=T1.IDTWOMASS AND T3.ID=T1.IDUSNOA2
```

Rewrite this statement to select cross-identification with $R_{USNO} - J_{2MASS} < 1.0$.

```
SELECT T1.CROSSID, T1.IDTWOMASS, T1.IDUSNOA2, T2.JMAG, T2.HMAG, T2.KMAG, T3.BMAG, T3.RMAG
      FROM CROSSID T1, TWOMASS T2, USNOA2 T3
      WHERE T2.ID=T1.IDTWOMASS AND T3.ID=T1.IDUSNOA2 AND T3.RMAG-T2.JMAG < 1.0
```

Find sources from USNOA2 which were not crossidentified

```
select ID, RA2000, DEC2000, BMAG, RMAG FROM USNOA2 where ID NOT IN (SELECT IDUSNOA2 FROM CROSSID)
```

12. ADDITIONAL TASKS

select 1 deg region from the galactic plane and from one of galactic poles from 2MASS and USNO-A2

crossidentify these regions

find completeness limit for each magnitude in original catalog and in cross-identified catalog

find completeness limit for each range of colors in RMAG-JMAG (from -1.0 till 3.0 with 1.0 mag step)

Hint: to find a completeness limit you have to build a historgam: number of sources per magnitude. Try to do it with SQL statement (SELECT with COUNT() function and a magnitude in some interval in WHERE statement).