

Assignment 1 T3  
Zhengyu Pan  
April 16th, 2020

**First of all, the link to github**

<https://github.com/zacharyPan/zhengyuPanBUSA8090A1T3>

**To download it,**

```
git clone https://github.com/zacharyPan/zhengyuPanBUSA8090A1T3
```

**Before run the code,**

```
chmod u+x zhengyuPanBUSA8090A1T3/*.sh
```

To check the code, it is better to copy and paste from A1T3commandLine.txt Which is in the github repository with link above.

Additionally, I also made all mysql codes together in a .bash file and separately in .sql file as well as available in A1T3commandLine.txt

**Question 1 a**

**In command line,**

```
mysql -u awkologist -p compbiol -e 'select * from expression where  
expr_value is not null;' >question1a.sql
```

```
less question1a.sql > question1a.tab
```

**I follow book “computational Biology”, page 314(pi page) example but forgot change “mysql” to mysqldump” and forgot to drop “-e”, but magically got the right answer. Additionally, I take out all the NULL from the database**

```
vi question1a.tab
```

```
gene      expr_value  
alr1207 8303  
alr2938 10323  
alr3395 1432  
all3556 8043  
alr4392 729  
alr4851 633  
alr5000 5732
```

## Question 1 b

Some of my command line is only for demonstrating, to check my work ,

In command line,

```
mysql -u awkologist -p compbiol -e 'select * from
annotation where function is not null;' > question1b.sql
```

```
less question1b.sql > question1b.tab
```

```
vi question1b.tab
```

```
gene      function      metabolism
alr2938 iron superoxide dismutase      Detoxification
alr4392 nitrogen-responsive regulator      Nitrogen assimilation
alr4851 preprotein translocase subunit      Protein and peptide secretion
alr3395 adenylosuccinate lyase      Purine biosynthesis
alr1207 uridylylate kinase      Pyrimidine biosynthesis
alr5000 CTP synthetase      Pyrimidine biosynthesis
all3556 succinate-dehydrogenase      TCA cycle
```

**I took out the NULL from the database**

## Question 1 c

In command line,

```
./joinTwofile.sh
```

Or

```
./joinTwofile.sh > question1c_result.tab
```

The explanation is inside the code after “#”

```
#!/bin/bash
# join two file with select columns
# author Zhengyu Pan
# April 14th, 2020

# first of all, let me sort both of my table with the header
# (Need to sort text keeping first line always first, n.d.)
(head -1 question1a.tab;sort <(sed -n '2,$p' question1a.tab))> question1a_sorted.tab
(head -1 question1b.tab;sort <(sed -n '2,$p' question1b.tab))> question1b_sorted.tab

# then join them together
join -t $'\t' question1b_sorted.tab question1a_sorted.tab > foo.tab

# last drop the column we don't need(metabolism, which is the third column)
# (How to remove a column or multiple columns from file using shell command?, n.d.)
awk '{ $3="" ; print $0 }' foo.tab
```

```

gene function  expr_value
all3556 succinate-dehydrogenase  cycle 8043
alr1207 uridylate  Pyrimidine biosynthesis 8303
alr2938 iron  dismutase Detoxification 10323
alr3395 adenylosuccinate  Purine biosynthesis 1432
alr4392 nitrogen-responsive  Nitrogen assimilation 729
alr4851 preprotein  subunit Protein and peptide secretion 633
alr5000 CTP  Pyrimidine biosynthesis 5732

```

## Question 1d

**Consider the problem of listing all duplicate metabolisms in the annotation table using SQL.**

i

$$\pi_{\text{metabolism}}(\sigma_{\text{annotation} \bowtie \text{metabolism} = \text{metabolism}} (\text{annotation} \text{ AND } \text{annotation}_{\text{gene} <> \text{gene}}))$$

ii

**In command line,**

I put all mysql code in one mysql.sh file, either run all mysql together

```
./mysql.sh
```

Or input the following

```
mysql -u awkologist -p compbiol -e 'select distinct a1.metabolism
from annotation a1 INNER Join annotation a2 on\
a1.metabolism = a2.metabolism where a1.gene <>a2.gene;'
```

```

+-----+
| metabolism |
+-----+
| Pyrimidine biosynthesis |
+-----+

```

Or

```
mysql -p -u awkologist compbiol < question1_partd_ii.sql
```

Additionally, to check in mysql,

```

$ mysql -p -u awkologist compbiol
mysql> source question1_partd_ii.sql

```

## Question 2

**a .**

**i.**

$\pi_{LastName, FirstName}(\sigma_{coach = NULL}(Member))$

**ii.** {m.LastName, m.FirstName | Member(m) and m.Coach = NULL}

**iii.**

**In command line,**

I put all mysql code in one mysql.sh file, either run all mysql together  
./mysql.sh

```
mysql -u awkologist -p compbiol -e 'select \
m.LastName, m.FirstName from Member m where m.Coach is null;'
```

```
+-----+-----+
| LastName | FirstName |
+-----+-----+
| Stone    | Michael   |
| Nolan    | Brenda    |
| Branch   | Helen     |
| Beck     | Sarah     |
| Spence   | Thomas    |
| Olson    | Barbara   |
| Wilcox   | Daniel    |
| Young    | Betty     |
| Willis   | Carolyn   |
| Kent     | Susan     |
+-----+-----+
```

or

```
mysql -p -u awkologist compbiol < question2_parta_iii.sql
```

Additionally, to check in mysql,

```
$ mysql -p -u awkologist compbiol
mysql> source question2_parta_iii.sql
```

**b .**

**i.**  $\pi_{LastName, FirstName}(\sigma_{JoinDate \text{ like } 2010}(Member))$

**ii.** {m.LastName, m.FirstName | Member(m) and m.JoinDate like 2010}

**iii.**

**In command line,**

I put all mysql code in one mysql.sh file, either run all mysql together  
./mysql.sh  
Or do the following

```
mysql -u awkologist -p compbiol -e 'select m.LastName,
m.FirstName from Member m where JoinDate LIKE "%2010%";'
```

```
+-----+-----+
| LastName | FirstName |
+-----+-----+
| Beck     | Sarah    |
| Kent     | Susan    |
+-----+-----+
```

Or

```
mysql -p -u awkologist compbiol < question2_partb_iii.sql
```

Additionally, to check in mysql,

```
$ mysql -p -u awkologist compbiol
mysql> source question2_partb_iii.sql
```

## C.

i.

$$\pi_{LastName, FirstName, MemberID}(\sigma_{(Member \bowtie_{MemberID = MemberID} Entry) - \sigma_{year = 2014}(Member \bowtie_{MemberID = MemberID} Entry)})$$

*Citation, ("NOT EXISTS in Relational Algebra and QBQL," 2012)*

```
ii. {m.LastName, m.FirstName, m.MemberID | Member(m), Entry(e)
and m.MemberID = e.MemberID NOT  $\exists$  (e) Entry(e)
(m.MemberID = e.MemberID and e.Year = 2014)}
```

iii.

**In command line,**

I put all mysql code in one mysql.sh file, either run all mysql together

```
./mysql.sh
```

Or do the following

```
mysql -u awkologist -p compbiol -e 'select distinct m.LastName,
m.FirstName, m.MemberID from Member m inner join Entry e on
e.MemberID = m.MemberID where not exists(select * from Entry e
where e.MemberID = m.MemberID and e.Year = 2014);'
```

```
+-----+-----+-----+
| LastName | FirstName | MemberID |
+-----+-----+-----+
| Burton   | Sandra   | 228      |
| Spence   | Thomas   | 239      |
+-----+-----+-----+
```

Or

```
mysql -p -u awkologist compbiol < question2_partc_iii.sql
```

Additionally, to check in mysql,

```
$ mysql -p -u awkologist compbiol
mysql> source question2_partc_iii.sql
```

**d. i.** {m.LastName, m.FirstName, m.MemberID | Member(m) AND  
Not  $\exists$ (e) Entry(e)  
(Not  $\exists$ (e) Entry(e) And e.MemberID = m.MemberID  
and e.Year = e.Year)}

**ii.**

**In command line,**

I put all mysql code in one mysql.sh file, either run all mysql together

./mysql.sh

**Or do the following**

```
mysql -u awkologist -p compbiol -e 'select m.LastName,
m.FirstName, m.MemberID\
from Member m where not exists\
(select * from Entry e1 where not exists\
(select * from Entry e2 where e2.MemberID = m.MemberID and \
e1.Year = e2.Year));'
```

```
+-----+-----+-----+
| LastName | FirstName | MemberID |
+-----+-----+-----+
| Cooper   | William   | 235      |
| Pollard  | Robert    | 286      |
| Taylor   | William   | 415      |
+-----+-----+-----+
```

Or

```
mysql -p -u awkologist compbiol < question2_partd_ii.sql
```

Additionally, to check in mysql,

```
$ mysql -p -u awkologist compbiol
mysql> source question2_partd_ii.sql
```

## Reference:

*How to remove a column or multiple columns from file using shell command?* (n.d.). Unix & Linux Stack Exchange. Retrieved April 14, 2020, from <https://unix.stackexchange.com/questions/222121/how-to-remove-a-column-or-multiple-columns-from-file-using-shell-command>

*Need to sort text keeping first line always first.* (n.d.). Retrieved April 14, 2020, from <https://www.unix.com/shell-programming-and-scripting/244756-need-sort-text-keeping-first-line-always-first.html>

NOT EXISTS in Relational Algebra and QSQL. (2012, September 24). *Quasi Believable Quantifier Laws*. <https://vadimtropashko.wordpress.com/2012/09/24/not-exists-in-relational-algebra-and-qsql/>