Roll #: xxxxxxxxx

Email : xxxxxxxxxxxxxxxxxx

YOUR NAMEA long, thin rectangle to divide sections of the document

# **QUESTION 1:**

## Part (a) *— Create a tab-separated file contain the data from the ‘expression’ table*

Place the data in the **Question1A.txt** file that retrieves from the expression table. As the data are arranged in the genes so we needed to answer the question that is mentioned in the bottom in Part(c). Keep in mind, you need to include "> **Question1A.txt**" in the code when using the run mode.

|  |
| --- |
| Part (a):  mysql -u UserName -p Password -e 'select \* from **expression** order by gene;' > **Question1A.txt** |

## 

## Part (b) *— Create a tab-separated file contain the data from the ‘annotation’ table*

Place the data in the **Question1B.txt** file that retrieves from the expression table. As the data are arranged in the genes so we needed to answer the question that is mentioned in the bottom in Part(c). Keep in mind, you need to include "> **Question1B.txt**" in the code when using the run mode.

|  |
| --- |
| Part (b):  mysql -u UserName -p Password -e 'select \* from **annotation** order by gene;' > **Question1B.txt** |

## 

## Part (c) *— Perform the computation*

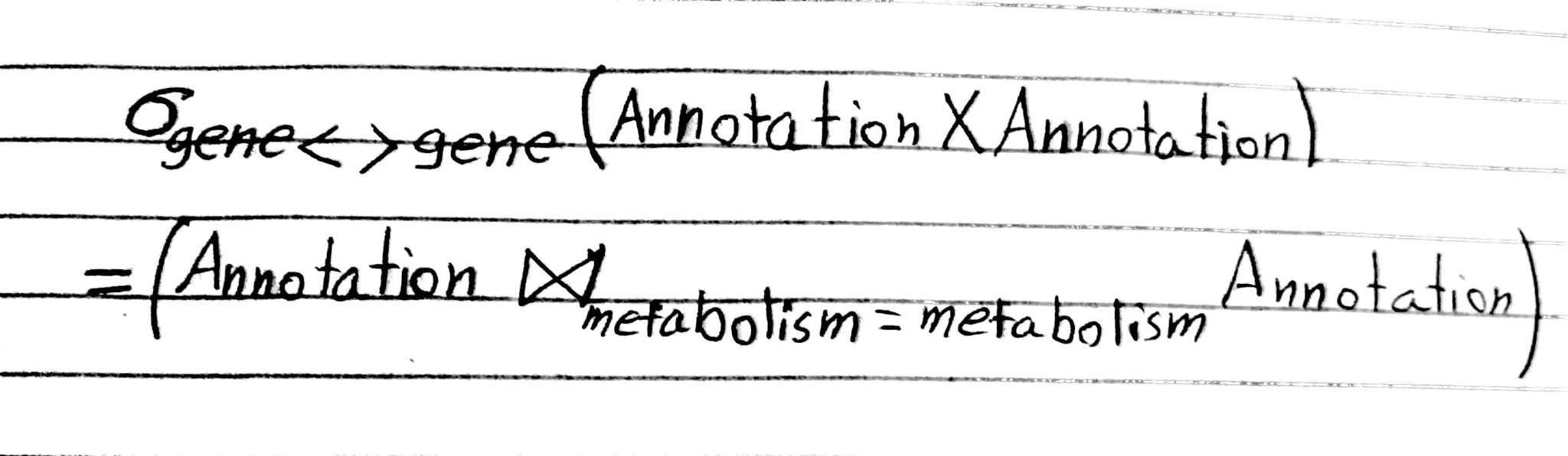
A file with the name of **Question1A.txt** that contains the expression table**,** and the other file with the name of **Question1B.txt** that contain the annotation table. We will combine both of these files. The command **“-j 1”** is used to merge two files based on the first column of the two files. The expected result is the gene, function, and the value of the expression, ending with “-o 1.1, 2.2, 1.2”,It specifies the following columns: gene (first rank\Columns of the **Question1A.txt** file), function (second rank\Columns of the **Question1B.txt** file) and the expression value from the second rank\columns of the **Question1A.txt** file.

After run command that [present in part a and then run command from part b Now for this task (perform computation) use the following command

|  |
| --- |
| Part (c):  join -j 1 -o 1.1,2.2,1.2 **Question1A.txt** **Question1B.txt** |

## Part (d) *— listing all duplicate metabolisms*

**(i)**



**(ii)**

Keep in mind that the annotation table is linked with itself, with the same metabolism but different genes. If we don’t do that in this way then each gene will link or join with itself**.**

|  |
| --- |
| part(d)  mysql -u UserName -p Password -e ' b.metabolism where a.gene <> b.gene = a.gene, a.function, a.metabolism, b.gene from annotation a inner join annotation b on a.metabolism |

# **QUESTION 2:**

# **Part (a)** ***—*** Listing all members who don’t have a coach.

1. R1 <- σcoach is NULL(Member)

R <- ∏ FirstName, memberID, LastName, (R1)

1. Let mem = member table so

{mem.FirstName, mem.LastName, mem.memberID | mem AND mem.Coach is NULL}

1. mysql -u username -p password –e

‘select LastName, mem.LastName, FirstNamefrom Member

where Coach is NULL’

# **Part (b)** ***—*** Listing all members who joined during the 2013 calendar year.

1. R1 <- σjoindate >= ’2013**/**1**/**1’ AND joindate <’2014-1-1’(Member)

R2 <- ∏ FirstName, memberID, LastName (R1)

1. Let mem = member Table so

{mem.LastName, mem.memberID, mem.FirstName | mem and mem.joindate >= ’2013/1/1’ AND mem.joindate <’2014/1/1’

1. Let mem = member table so

select LastName, memberID, FirstName

from Member

where JoinDate >’2013/1/31’ and JoinDate < ’2014/1/1’

# **Part (c)** ***—*** Finding the member names and IDs of those who didn’t compete in any tournaments in 2013, but have competed in other year

1. R1 <- ∏ MemberID(σyear-2013(Entry))

R2 <- ∏ LastName,FirstName,MemberID((∏ MemberID(Member) U ∏ MemberID(Entry))

R <- R2 - R1

1. Let mem = member table so

{mem.LastName, mem. memberID, mem.FirstName

|Member(mem),.∀(en) Entry(en)∄ (f)Entry(f) and

mem.MemberID = en.MemberID and f.year = 2013

and en.MemberID = f.MemberID

1. mysql -u username -p password -e’

Select Distinct mem.LastName, mem.FirstName, mem.MemberID

From Member inner join Entry mem on mem. MemberID = mem.MemberID Where mem.MemberID NOT exists (Select mem.MemberID

From Entry mem1 Where mem1.Year=2013);'

# **Part (d)** ***—*** Member names and IDs of those who have competed in every year that club members have competed.

1. Let mem = member table so.

{mem.LastName, mem.FirstName, mem.MemberID |

Member(mem),Entry(ent),Entry(a),Entry(h)

and mem.MemberID = ent.MemberID

and ent.MemberID = a.memberID

and a.MemberID = h.MemberID and ent.Year = 2013 and

a.Year = 2014 and h.year =2015}

1. mysql -u username -p password -e '

Select Distinct mem.LastName, mem.FirstName, ent.MemberID

from ((Entry ent inner join Entry a on ent.MemberID =

a.MemberID) inner join Entry h on a.MemberID = h.MemberID)

inner join Member mem on m.MemberID = e.MemberID where

e.Year = 2013 and a.Year = 2014 and h.Year = 2015;’