CS 122A: Introduction to Data Management – Winter 2015

Homework 8: NoSQL (Hands-On) (100 points, +20 extra points)

Due Date: **Tuesday, Mar 17 (5:00 PM)**

Submission

This homework assignment should be submitted to EEE before 5 pm on the due date. Thus, **you should follow the instruction at the end of this document** and submit your TXT file to EEE before 5 pm on that day. Refer to the following table for the submission opportunities. After 5 pm on Thursday, no more submission is allowed. That is, we will not accept assignments after that time (and will be publishing the solutions at that time). **For this assignment, you do not need to submit a hard copy**. Please get all your work in on time! Finally, please recall that **all homework assignments are to be completed individually**.

Date / Time	Place	Remark
Tuesday, Mar 17 (5:00 PM)	EEE	Due date
Wednesday, Mar 18 (5:00 PM)	EEE	20 points will be deducted
Thursday, Mar 19 (5:00 PM)	EEE	40 points will be deducted

NoSQL [100 pts, +20 extra points]

As Deliber's business is expanding day by day in the Orange county area, you are preparing to open branches in 50 major cities. It is clear that you will soon have a vast amount of data to manage. You need to prepare a database system that can accommodate such data, but the co-founders refuse to pay the commercial licensing fees for expensive parallel database systems. As a CS122a taker, you now aware that there is a new, free, open source system that can support your needs – AsterixDB.

By referencing the "101" documentation and help from the AsterixDB team, you have designed an AsterixDB dataverse, datatypes, and several datasets to migrate data from MySQL DB. By designing the following datasets, you have found out that you can reduce the number of datasets to accommodate all data by using more natural, nested data structures. For example, now each user record can include the information both for their customer and driver roles. Since these nested fields are optional, you can include these fields or omit them. This is one example characteristics of a NoSQL data management system. All types such as CustomersType and DriversType are defined in the AQL script. Refer to that script on the class wiki to get the entire schema.

Schema (dataset)

Users (<u>*user_id*: INT</u>, *name*: STRING, *phone_number*: STRING, *customer_data*: CustomersType?, *driver_data*: DriversType?)

Restaurants (*restr_id*: INT, *name*: STRING, *address*: STRING, *bank_account:* BankAccountType, *last_bank_transaction_datetime*: DATETIME, *cuisine*: {{ STRING }}, *dish*: {{ DishesType }})

Customers_review (*cust_id*: INT, *restr_id*: INT, *rating*: INT, *customer_comment*: STRING)

Orders (*order_id*: INT, *cust_id*: INT, *drvr_id*: INT, *restr_id*: INT *order_datetime*: DATETIME, *total_amount*: DOUBLE?, *order_status*: [Orders_track_statusType], *dishes*: {{ DishesQuantityType }})

To understand the notations and syntax, please refer to the AsterixDB online documentation (https://asterixdb.ics.uci.edu/documentation/aql/primer.html), especially the "101" portion. For the entire schema, refer to the AQL script on the class wiki.

To see parts of the actual data, refer to the associated ADM files for this assignment. These four files contain the data for the Deliber dataverse.

In this assignment, you are to use AsterixDB to query the Deliber dataverse and its data and show the results. You need to launch the AsterixDB instance and load the data. Refer to the Windows / OS X section to start AsterixDB instance and load the data based on your OS.

[Windows]

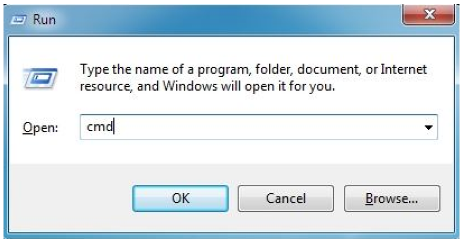
1. Download an AsterixDB Instance file. The file size is relatively big (about 90MB), so it will take some time to download it.

Windows: https://drive.google.com/file/d/0BzeFLVaOjXQYS2EzelBIdXpPT3c/view?usp=sharing

2. Uncompress the ZIP file to a folder in which your account has the read/write access.
(e.g., **Windows** - C:\users\youraccount*\Downloads)

* youraccount should be your actual account. (e.g., taewookim)

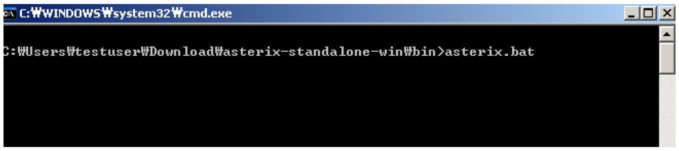
3. Execute CMD and type **cd C:\Users\youraccount\Downloads\asterix-standalone-win\bin** in the command prompt. You can execute CMD by pressing Windows Key + [R], typing cmd and pressing Enter key.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

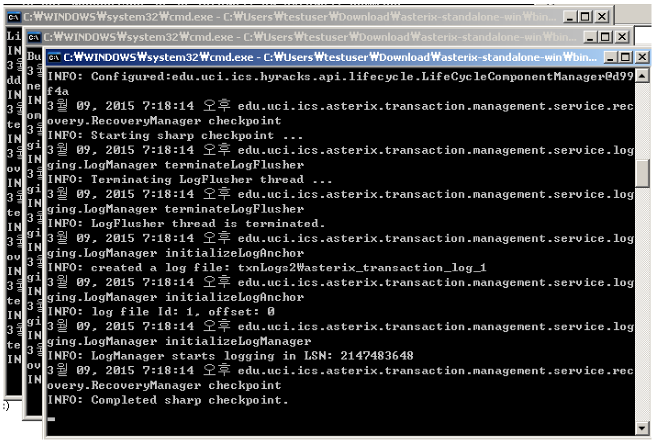
Open: cmd

4. Execute asterix.bat and wait until you see the following message (GLOBAL Recovery Completed) or (INFO: Result state cleanup instance successfully completed). You should be able to see three Windows. **If one of Windows shows continuous message, then close all windows and execute asterix.bat again.**



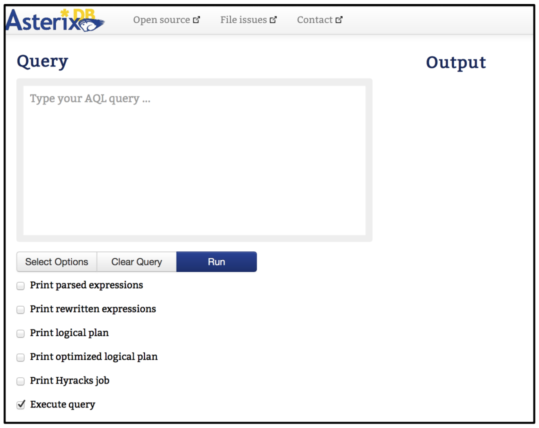
```
C:₩Users₩testuser₩Download₩asterix-standalone-win₩bin>asterix.bat
```

…


C:₩WINDOWS₩system32₩cmd.exe - C:₩Users₩testuser₩Download₩asterix-standalone-win₩bin...
C:₩WINDOWS₩system32₩cmd.exe - C:₩Users₩testuser₩Download₩asterix-standalone-win₩bin...
C:₩WINDOWS₩system32₩cmd.exe - C:₩Users₩testuser₩Download₩asterix-standalone-win₩bin...

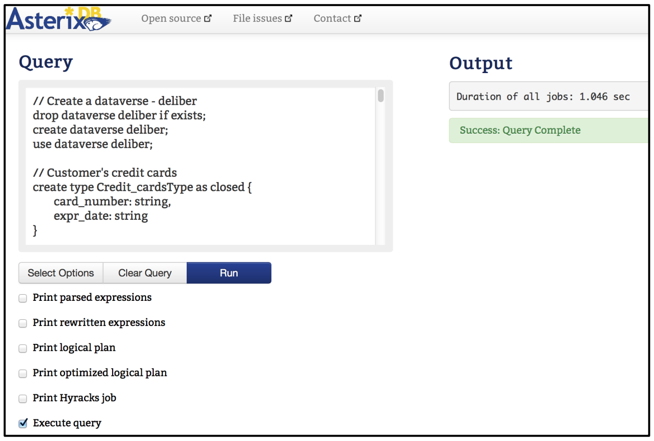
```
INFO: Configured:edu.uci.ics.hyracks.api.lifecycle.LifeCycleComponentManager@d99
f4a
3월 09, 2015 7:18:14 오후 edu.uci.ics.asterix.transaction.management.service.rec
overy.RecoveryManager checkpoint
INFO: Starting sharp checkpoint ...
3월 09, 2015 7:18:14 오후 edu.uci.ics.asterix.transaction.management.service.log
ging.LogManager terminateLogFlusher
INFO: Terminating LogFlusher thread ...
3월 09, 2015 7:18:14 오후 edu.uci.ics.asterix.transaction.management.service.log
ging.LogManager terminateLogFlusher
INFO: LogFlusher thread is terminated.
3월 09, 2015 7:18:14 오후 edu.uci.ics.asterix.transaction.management.service.log
ging.LogManager initializeLogAnchor
INFO: created a log file: txnLogs2₩asterix_transaction_log_1
3월 09, 2015 7:18:14 오후 edu.uci.ics.asterix.transaction.management.service.log
ging.LogManager initializeLogAnchor
INFO: log file Id: 1, offset: 0
3월 09, 2015 7:18:14 오후 edu.uci.ics.asterix.transaction.management.service.log
ging.LogManager initializeLogManager
INFO: LogManager starts logging in LSN: 2147483648
3월 09, 2015 7:18:14 오후 edu.uci.ics.asterix.transaction.management.service.rec
overy.RecoveryManager checkpoint
INFO: Completed sharp checkpoint.
_
```

5. Open a Web browser and connect to the following page.
http://localhost:19001 or http://127.0.0.1:19001



□ Print parsed expressions
□ Print rewritten expressions
□ Print logical plan
□ Print optimized logical plan
□ Print Hyracks job
☑ Execute query

6. Copy and paste the HW8 AQL script and click “Run” to see the result.



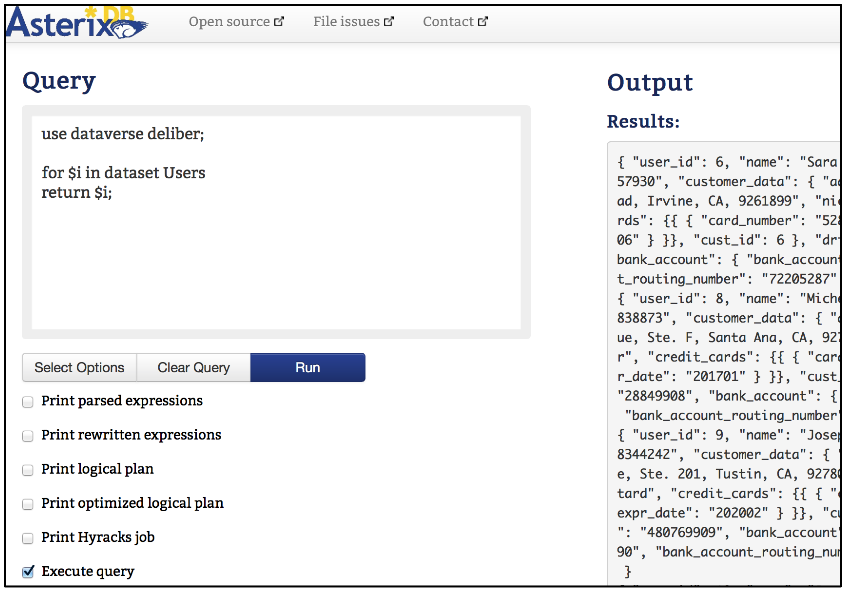
```
// Create a dataverse - deliber
drop dataverse deliber if exists;
create dataverse deliber;
use dataverse deliber;

// Customer's credit cards
create type Credit_cardsType as closed {
     card_number: string,
     expr_date: string
}
```

```
Duration of all jobs: 1.046 sec
```

Success: Query Complete

□ Print parsed expressions
□ Print rewritten expressions
□ Print logical plan
□ Print optimized logical plan
□ Print Hyracks job
☑ Execute query

7. Enter the following sample query in the box and click “Run” to see the result.



```
use dataverse deliber;

for $i in dataset Users
return $i;
```

```
{ "user_id": 6, "name": "Sara
57930", "customer_data": { "a
ad, Irvine, CA, 9261899", "ni
rds": {{ { "card_number": "52
06" } }}, "cust_id": 6 }, "dr
bank_account": { "bank_accoun
t_routing_number": "72205287"
{ "user_id": 8, "name": "Mich
838873", "customer_data": { "
ue, Ste. F, Santa Ana, CA, 92
r", "credit_cards": {{ { "car
r_date": "201701" } }}, "cust
"28849908", "bank_account": {
 "bank_account_routing_number
{ "user_id": 9, "name": "Jose
8344242", "customer_data": {
e, Ste. 201, Tustin, CA, 9278
tard", "credit_cards": {{ { "
expr_date": "202002" } }}, "c
": "480769909", "bank_account
90", "bank_account_routing_nu
 }
```

□ Print parsed expressions
□ Print rewritten expressions
□ Print logical plan
□ Print optimized logical plan
□ Print Hyracks job
☑ Execute query

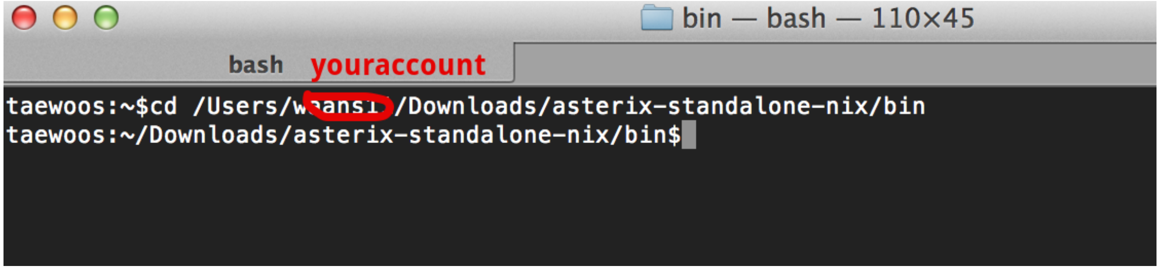
[OS X]

1. Download an AsterixDB Instance file. The file size is relatively big (about 90MB) so it takes time to download.

OS X: https://drive.google.com/file/d/0BzeFLVaOjXQYajdNNVJtY1dqTXM/view?usp=sharing

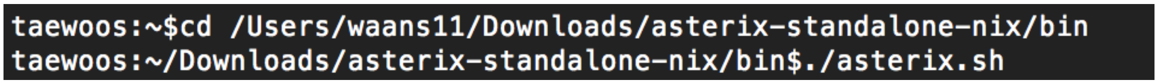
2. Uncompress the ZIP file to a folder that your account has read/write access.
(e.g., **OS X** - /Users/youraccount*/Downloads)
* youraccount should be your actual account. (e.g., taewookim)

3. Open a terminal and go to /Users/youraccount/Downloads/asterix-standalone-nix/bin directory. If you can't find how to execute the terminal, use Spotlight feature to find terminal and execute it.



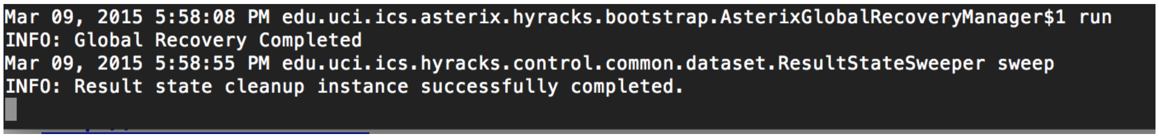
```
taewoos:~$cd /Users/waans11/Downloads/asterix-standalone-nix/bin
taewoos:~/Downloads/asterix-standalone-nix/bin$
```

4. Execute asterix.sh file and wait until you see the following message (GLOBAL Recovery Completed) or (INFO: Result state cleanup instance successfully completed).



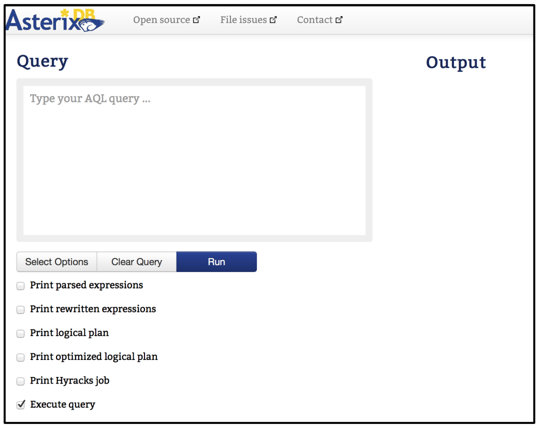
```
taewoos:~$cd /Users/waans11/Downloads/asterix-standalone-nix/bin
taewoos:~/Downloads/asterix-standalone-nix/bin$./asterix.sh
```

…



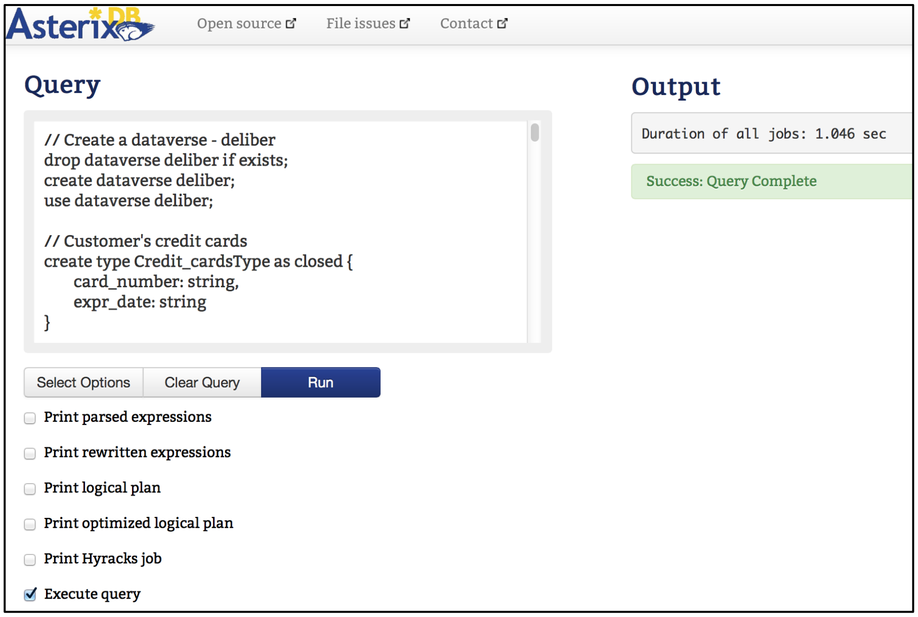
```
Mar 09, 2015 5:58:08 PM edu.uci.ics.asterix.hyracks.bootstrap.AsterixGlobalRecoveryManager$1 run
INFO: Global Recovery Completed
Mar 09, 2015 5:58:55 PM edu.uci.ics.hyracks.control.common.dataset.ResultStateSweeper sweep
INFO: Result state cleanup instance successfully completed.
```

5. Open a Web browser and connect to the following page.
http://localhost:19001 or http://127.0.0.1:19001



☐ Print parsed expressions
☐ Print rewritten expressions
☐ Print logical plan
☐ Print optimized logical plan
☐ Print Hyracks job
☑ Execute query

6. Copy and paste the HW8 AQL script and click “Run” to see the result.



```
// Create a dataverse - deliber
drop dataverse deliber if exists;
create dataverse deliber;
use dataverse deliber;

// Customer's credit cards
create type Credit_cardsType as closed {
      card_number: string,
      expr_date: string
}
```

```
Duration of all jobs: 1.046 sec
```

Success: Query Complete

□ Print parsed expressions

□ Print rewritten expressions

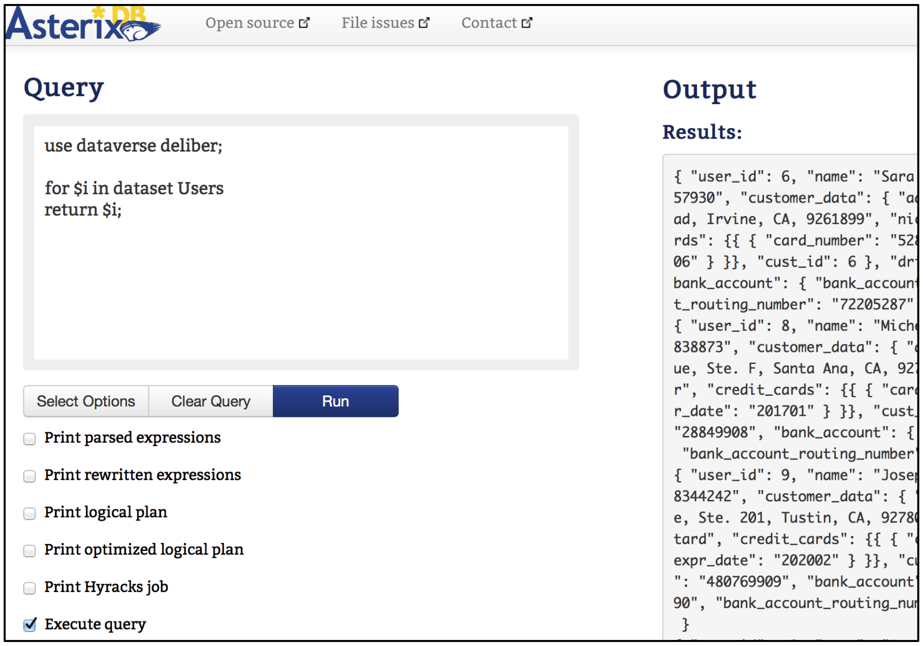
□ Print logical plan

□ Print optimized logical plan

□ Print Hyracks job

☑ Execute query

7. Enter the following sample query in the box and click “Run” to see the result.



```
use dataverse deliber;

for $i in dataset Users
return $i;
```

```
{ "user_id": 6, "name": "Sara
57930", "customer_data": { "a
ad, Irvine, CA, 9261899", "ni
rds": {{ { "card_number": "52
06" } }}, "cust_id": 6 }, "dr
bank_account": { "bank_accoun
t_routing_number": "72205287"
{ "user_id": 8, "name": "Mich
838873", "customer_data": { "
ue, Ste. F, Santa Ana, CA, 92
r", "credit_cards": {{ { "car
r_date": "201701" } }}, "cust
"28849908", "bank_account": {
 "bank_account_routing_number
{ "user_id": 9, "name": "Jose
8344242", "customer_data": {
e, Ste. 201, Tustin, CA, 9278
tard", "credit_cards": {{ { "
expr_date": "202002" } }}, "c
": "480769909", "bank_account
90", "bank_account_routing_nu
 }
```

□ Print parsed expressions

□ Print rewritten expressions

□ Print logical plan

□ Print optimized logical plan

□ Print Hyracks job

☑ Execute query

Now write each of the following statements in AQL against the Deliber dataverse. Turn in both the AQL statements and their results. Please note that you will not get points for correct results if the AQL itself is not correct. Since you have a “live” AsterixDB instance at your disposal, however, this should not be an issue – as you can test all of your work. **For this assignment, you are to create a TXT file to include your AQL statements as well as their results and submit it to the EEE dropbox. Refer to the end of this homework for more information about this step.**

1. [20 pts] Which dataset in the Deliber dataverse can still be classified as 1NF (in the relational database design theory sense) based on the DDL statements? Write the dataset name and briefly state your reasoning.

2. [20 pts] Show the id, phone-number, and the address of a customer whose name is “James M. Schwan”.

The format of the result should be the following:
{ “id”: xxxx, “phone_number”: “xxxxxxxxxx”, “address”: “xxxxxxxxxxxxxxxxxxxxxxxxxxxx” }

a) [15 pts] AQL
b) [5 pts] The result

3. [20 pts] Show the name and the price of dishes, and also their offering restaurant name, whose spiciness is “hot”. Sort the result by the dish name. You can see that “spiciness” of a dish is not defined in the DDL. However, dishes stored in actual restaurant instances can still store that information. This illustrates one interesting characteristic of No-SQL systems such as AsterixDB.

The format of the result should be the following:
{ “name”: xxxx, “price”: xxxxxxxxxx, “from”: xxxxxxxxx} …

a) [15 pts] AQL
b) [5 pts] The result

4. [20 pts] Return the names, addresses, and cuisines of restaurants that serve ‘Mexican’ cuisine. Sort the result by the name.

The format of the result should be the following:
{“name”: “xxxxxxxx”, “address”: “xxxxxxxxxxxxxx”, “cuisines”: {{ …. }} } …

a) [15 pts] AQL
b) [5 pts] The result

5. [20 pts] Return the minimum total amount of any order among all orders.

The format should be the following:
{ “total_amount”: xxxxxxx.xxx }

a) [15 pts] AQL
b) [5 pts] The result

6. [**Extra 20 pts question**] Return the customer ids, and number of reviews that each customer posted. Only return the customers who have posted at least two reviews. Sort the result by the id.

The format of the result should be the following:
{“id”: x, “count”: x} …

a) [15 pts] AQL
b) [5 pts] The result

Final Note: For this assignment, you need to create a TXT file to include your queries and its results and submit your file to the EEE dropbox. You are not required to submit a hard copy. Points may be deducted if you don’t follow the instruction. Here are the instructions for this file.

1. Open a blank text file in a text editor and copy the following template into the file. Then, replace each query in the template with your query. (e.g., 1Q, 2Q, …)
use dataverse deliber;

// 1A
// Put your answer here. Do not remove // in the beginning of this question.

// 2Q
Your query for Q2

// 2A
Your Answer for Q2

// 3Q
Your query for Q3

// 3A

Your Answer for Q3

// 4Q
Your query for Q4

// 4A
Your Answer for Q4

// 5Q
Your query for Q5

// 5A
Your Answer for Q5

// 6Q - EXTRA
Your query for Q6

// 6A - EXTRA
Your Answer for Q6

2. If you have skipped a question, put **// X: skipped (e.g., // 2: skipped**) between the previous question and the next question. Delete the query and answer in the question that you skipped.

3. For the question 1, replace "Put your answer here. Do not remove // in the beginning of this question." with your answer. Do not remove // since your answer is not a query.

4. In the AsterixDB Webpage (http://localhost:19001), manually execute one query at a time. When the result is returned for a query, copy the result into the corresponding answer part (e.g., Your answer for the question 1 goes to 1A.). Repeat this step for the entire queries.

5. Save it as "YourStudentID_Last_FirstName.txt" (e.g. 12345678_Kim_Taewoo.txt). Do not convert it to other format such as DOC or PDF.

6. Submit the TXT file to EEE. (https://eee.uci.edu/toolbox/dropbox/)