CS 122A: Introduction to Data Management – Fall 2021

Homework 8: NoSQL (100 points)

Due Date: **Wednesday, Dec 1 (6:00 PM)**

Submission

All HW assignments must be submitted online as a PDF through the associated dropbox on Gradescope. See the table below for the HW submission opportunities. Note that after 6:00 PM on Thursday, December 2 no further HW submissions will be accepted. (We will be releasing the solution at that time so you can study it for the Endterm exam.) Please strive to get your work in on time!

Date / Time	**Grade Implications**
Wednesday, Dec 1 (6:00 PM)	Full credit will be available
Thursday, Dec 2 (6:00 PM)	10 points will be deducted

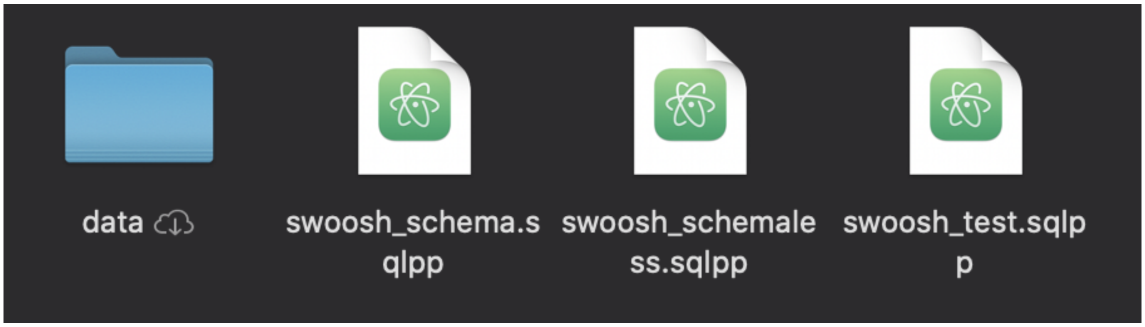
NoSQL (SQL) [100 pts]

Congratulations! If you are reading this, you **must have successfully finished reading and running the exercises in the SQL++ Primer**. ***If you haven't, stop now and go do that first!*** Business is booming, and SWOOSH needs to scale beyond PostgreSQL's capabilities. In this assignment you will explore some of the virtues of "NoSQL" systems for applications like this. Enter AsterixDB…! As you read through the assignment, answer its questions, and write queries, you will want to refer to two other sources of information -- namely, the solution to HW #1 (the E-R schema) and the corresponding translated relational schema that you have used for all the SQL-based HW assignments. You may also find that the queries that you're asked to write here seem hauntingly familiar (*deja vu!*), so you can refer to the relevant earlier assignment(s) if you are wondering if your answers here are on track. (We will also specify the number of expected answers for each query, as usual, and show a sample output record to help you with that.)

Step 0. Load the Dataset

To save you time, we have transformed the relational SWOOSH database schema into ADM (the Asterix Data Model). You need to download our scripts [HERE] and populate the dataset. Unzip the 'hw8.zip' file; there should be 1 folder (containing the JSON data) and 3 SQL++ scripts inside the folder like the following picture. (**IGNORE** the DS.store and MACOS files, those are automatically generated by our

MacOS laptop when compressing a folder :-)!)



data

swoosh_schema.sqlpp

swoosh_schemaless.sqlpp

swoosh_test.sqlpp

After you've started your sample AsterixDB cluster (through the **AsterixDB instructions doc**), the next thing you should do is run the initial test script (swoosh_test.sqlpp). The test script creates a test dataverse, which only contains the dataset for the **Course** entity, and loads the **Course** JSON data into this dataset. To do so, first open the `swoosh_test.sqlpp` file. **Replace the text "PATH_TO_HW8_FOLDER_GOES_HERE" to point to the above *hw8* folder on your machine.**

Windows Users: There is **NO** need to change '/'. The path you should use will look like this:
`("path"=":///C:/Users/XXX/Downloads/hw8/data/course.json")`

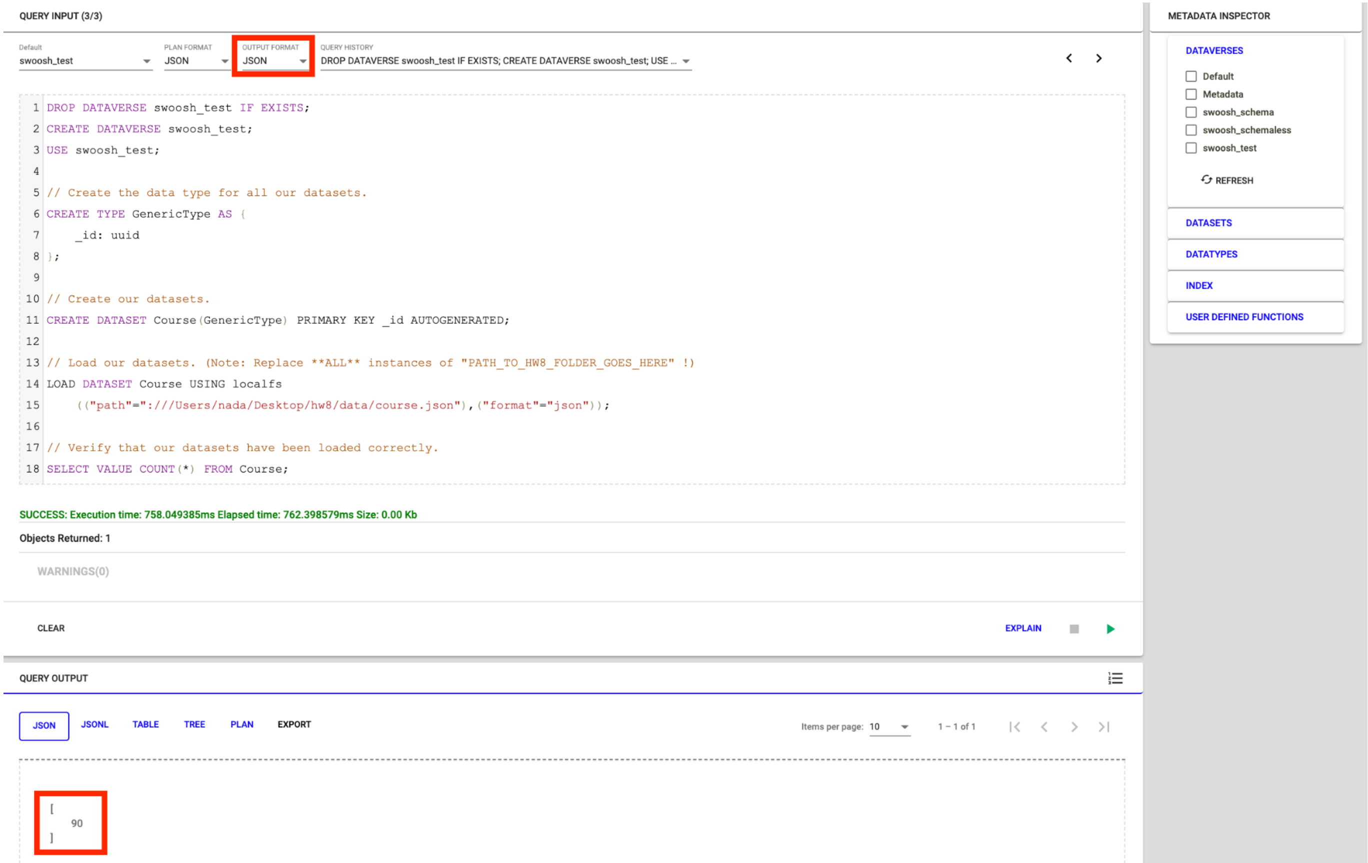
OSX, Linux Users: Change the **PATH_TO_HW8_FOLDER_GOES_HERE** to match with the location where you placed the hw8 folder. (E.g., `"path"=":///Users/XXX/Downloads/hw8/data/course.json"`)

All users: Beware of errors in these paths -- particularly of having the wrong number of leading slashes! Also make sure your quotation marks do not curl (e.g.: **"** instead of **"**). Your load commands will fail if you get those things wrong. Please double-check your path before posting questions on Piazza!

After you change the path, **copy and paste the DDL** you just edited into the Query box of the AsterixDB query interface. Make the following adjustments to the settings above the query box (if required):

1. Make sure that `Output Format` is "JSON" (the default setting).

Finally, click 'Run' (the play button). After execution, you should see something that resembles the following result (with the number 90 under `Query Output`):



Default: swoosh_test
PLAN FORMAT: JSON
OUTPUT FORMAT: JSON
QUERY HISTORY: DROP DATAVERSE swoosh_test IF EXISTS; CREATE DATAVERSE swoosh_test; USE ...

```
DROP DATAVERSE swoosh_test IF EXISTS;
CREATE DATAVERSE swoosh_test;
USE swoosh_test;

// Create the data type for all our datasets.
CREATE TYPE GenericType AS {
    _id: uuid
};

// Create our datasets.
CREATE DATASET Course(GenericType) PRIMARY KEY _id AUTOGENERATED;

// Load our datasets. (Note: Replace **ALL** instances of "PATH_TO_HW8_FOLDER_GOES_HERE" !)
LOAD DATASET Course USING localfs
    (("path"=":///Users/nada/Desktop/hw8/data/course.json"),("format"="json"));

// Verify that our datasets have been loaded correctly.
SELECT VALUE COUNT(*) FROM Course;
```

SUCCESS: Execution time: 758.049385ms Elapsed time: 762.398579ms Size: 0.00 Kb

Objects Returned: 1

WARNINGS(0)

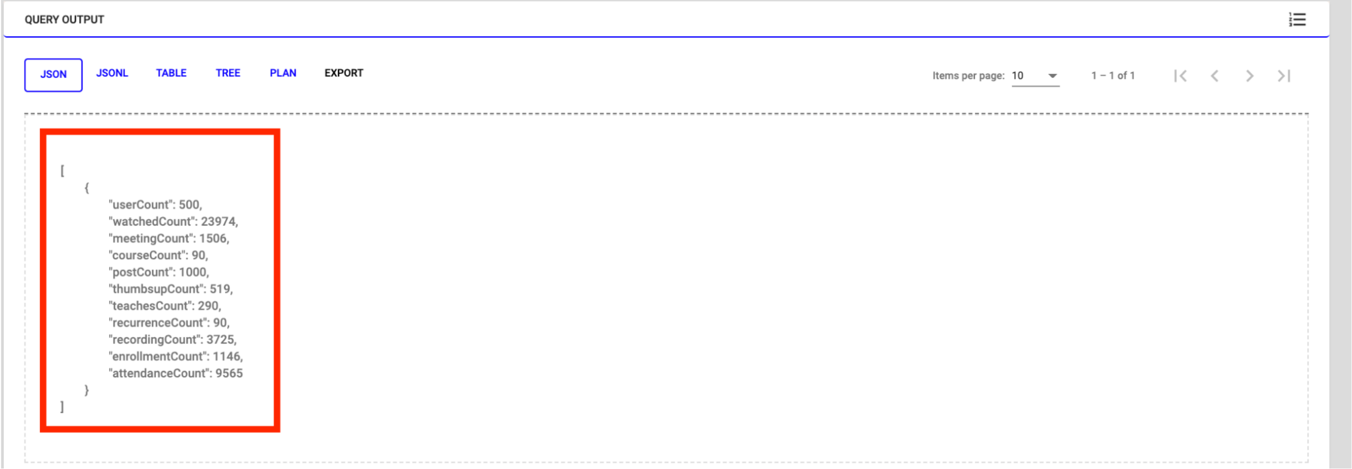
Items per page: 10   1 – 1 of 1

```
[
  90
]
```

If the Output doesn't show "SUCCESS" and the query results as listed here, please post your issue on Piazza. Again, beware of bad LOAD paths (usually due to too many or too few slashes). Please check for that error before reaching out for help on Piazza. ***NOTE: Under no circumstances should you try to load the full datasets until you get loading working perfectly here first!***

Once this test has executed successfully, great! You are now ready to populate the ***actual*** datasets. Open `swoosh_schema.sqlpp` and change the path's value to your downloaded 'hw8' folder just like what you did with the test dataset. Make sure that this is the absolute path and ***very carefully*** change **ALL** of the occurrences of the path variable as you did above. Each file will populate a dataset that you will have previously created. (*Hint:* If you get the path variable replacement wrong, you will probably get an error message that says something like: `ASX3077: /wrong_path/hw8/data/user.json: path not found [HyracksDataException]`.)

After changing all of the path values, go ahead and copy and paste the **ENTIRE** content of the `swoosh_schema.sqlpp` file into the query interface. As before, execute it by clicking `Run`. You should now see something similar to the following result:



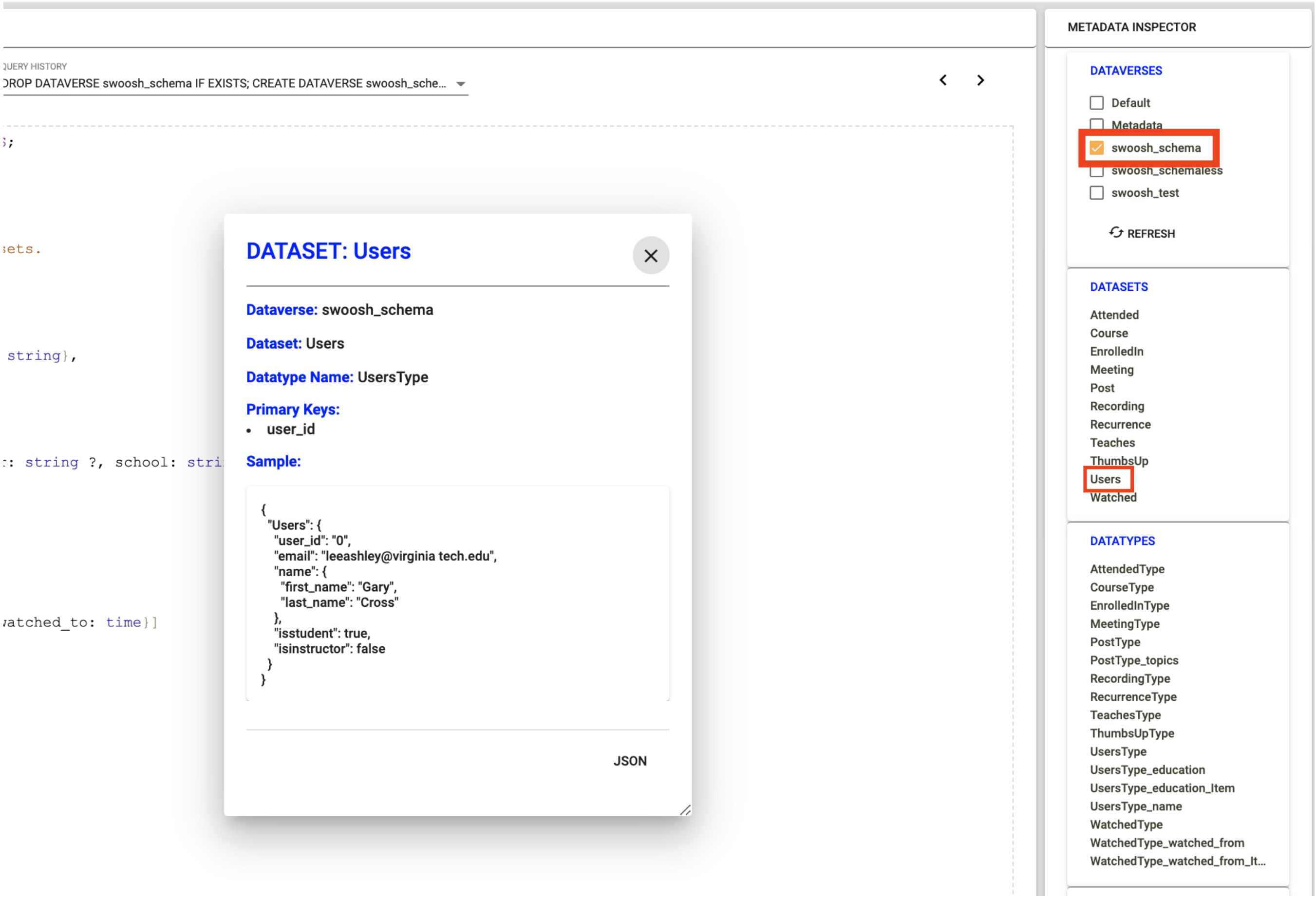
```
[
    {
        "userCount": 500,
        "watchedCount": 23974,
        "meetingCount": 1506,
        "courseCount": 90,
        "postCount": 1000,
        "thumbsupCount": 519,
        "teachesCount": 290,
        "recurrenceCount": 90,
        "recordingCount": 3725,
        "enrollmentCount": 1146,
        "attendanceCount": 9565
    }
]
```

Again, if the Output doesn't show "SUCCESS", please post your issue on Piazza after first trying to resolve the problem yourself. Don't wait until you finish your script! (And if you later happen to see other students struggling on Piazza at this step -- students making the same mistake you did -- please chime in and give them a hand! Loading the data successfully isn't a part of the graded exercise, so helping one another out with the ***loading*** specifics will be lauded rather than being disapproved of as inappropriate collaboration. :-))

Compare the counts in the output above with those in your own output to make sure that you have the complete set of datasets loaded. All of the datasets for the "schema`ed" dataverse reside in the `swoosh_schema` dataverse, so be sure to put the statement `USE swoosh_schema;` in front of each one of your queries when you run them. Here is an example of how that looks:

```
USE swoosh_schema;
SELECT *
FROM User U
WHERE U.user_id = '0';
```

You might find it helpful at time to use the **"Metadata Inspector"** tab in order to explore your datasets. To get information about a specific dataset (such as the datatype name, the primary key(s), and a sample object), check the box by the **Dataverse** name that you would like to work in, which will cause the corresponding **Datasets** to appear. Click on the dataset you'd like, for example "Users", and a window like the one below will pop up:



QUERY HISTORY

DROP DATAVERSE swoosh_schema IF EXISTS; CREATE DATAVERSE swoosh_sche...

```
;

sets.

 string},

: string ?, school: stri

watched_to: time}]
```

DATASET: Users

Dataverse: swoosh_schema

Dataset: Users

Datatype Name: UsersType

Primary Keys:

- user_id

Sample:

```
{
 "Users": {
  "user_id": "0",
  "email": "leeashley@virginia tech.edu",
  "name": {
   "first_name": "Gary",
   "last_name": "Cross"
  },
  "isstudent": true,
  "isinstructor": false
 }
}
```

METADATA INSPECTOR

DATAVERSES

- ☐ Default
- ☐ Metadata
- ☑ swoosh_schema
- ☐ swoosh_schemaless
- ☐ swoosh_test

DATASETS

Attended
Course
EnrolledIn
Meeting
Post
Recording
Recurrence
Teaches
ThumbsUp
Users
Watched

DATATYPES

AttendedType
CourseType
EnrolledInType
MeetingType
PostType
PostType_topics
RecordingType
RecurrenceType
TeachesType
ThumbsUpType
UsersType
UsersType_education
UsersType_education_Item
UsersType_name
WatchedType
WatchedType_watched_from
WatchedType_watched_from_It...

Note: For this assignment, you must turn in a PDF that you have created by copying/pasting from the query interface into a copy of the HW8 template and then PDF-printing the results. When you copy and paste your query and result, do ***not*** take a screenshot. Instead, use the text copy and paste feature.

NOTE: As you work on this HW, you should check the DDL of the schema-fied versions of the datasets to determine their attribute names and structures (for working with both versions).