COMP SCI 4094/4194/7094 - Distributed Databases and Data Mining
Assignment 2

DUE: 23:59 Monday, 16th Oct, 2017

Important Notes

- Handins:
  - The deadline for submission of your assignment is **23:59 Monday, 16th Oct, 2017**.
  - You must do this assignment individually and make individual submissions.
  - Your program should be coded in **C++** and pass test runs on the two test files. The sample input and output files are downloadable from the track of Assignment 2 at MyUni (`https://myuni.adelaide.edu.au/courses/25366/assignments/40065`).
  - You need to use `svn` to upload and run your source code in the web submission system following Web-submission instructions stated at the end of this sheet. You should attach your name and student number in your submission.
  - Late submissions will attract a penalty: the maximum mark you can obtain will be reduced by 25% per day (or part thereof) past the due date.
- Marking scheme:
  - 8 marks for testing the data cube.
  - 7 marks for testing the query.
  - **Note:** If it is found your code did not implement the required computation tasks in this assignment, you will receive zero mark regardless of the correctness of test output.

If you have any questions, please send them to the student discussion forum. This way you can all help each other and everyone gets to see the answers.

The assignment

In the recent years, the number of new enrolments in South Australian public universities has increased significantly. For the purpose of making university programs more competitive and providing better service, we wish to learn the insights of student preferences on different programs within a discipline. To do so, in this assignment, you are required to design and implement two algorithms for on-line analysis of student preferences in the information technology discipline containing PhD, MCS, MSE, BCS and BSE five programs. The first algorithm builds a data cube (Page 136 in textbook, 3rd ed) from the SA student on the enrolment information database with 4 dimensions/features: University, Program, Semester, Nationality. Specifically, your first algorithm should construct a 4-dimensional data cube of Number of Enrolments on the 4 dimensions which enables aggregate quires. Your second algorithm shall answer two types of aggregate quires: standard deviation (SD) query (SD of Number of Enrolments on a given dimension) and top-k query (top k numbers of enrolments on a given dimension), where the Standard Deviation (SD) of n numbers, $x_1, x_2, \ldots, x_n$ is defined as:

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}, \tag{1}$$

where $\bar{x}$ is the mean (arithmetic average) of the n x_i's, i.e., $\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$.

When calculating SD, your code should round off the decimals in the results to two digits only.

Database format

- Each line (tuple) in the database contains values in 5 dimensions (attributes), split by `Tab`, in the following order:
  University → Program → Semester → Nationality → Enrolment.

- Dimension “University” has 3 keys (index in code):
  UofA(0), UniSA(1), Flinders(2).

- Dimension “Program” has 5 keys:
  PhD(0), MCS(1), BSE(2), MSE(3), BCS(4).

- Dimension ”Semester” has 4 keys :
  s1/2016(0), s2/2016(1), s1/2017(2), s2/2017(3).

- Dimension “Nationality” has 5 keys :
  Australia(0), Brazil(1), China(2), India(3), Egypt(4).

- Dimension “Enrolment” contains positive integers stating enrolment numbers.

Example

Inputs

- Student enrolment information database:

University	Program	Semester	Nationality	Enrolment
Flinders	PhD	s1/2017	Australia	13
Flinders	PhD	s2/2017	Brazil	1
UofA	MCS	s2/2017	China	14
Flinders	MSE	s2/2016	India	13
UniSA	BSE	s2/2017	China	16
UniSA	MSE	s1/2016	China	14
Flinders	MCS	s1/2017	India	22
UofA	MSE	s1/2016	Egypt	2
UniSA	MCS	s1/2017	Egypt	17

- Queries:
  - Standard deviation query:
    - SD UofA ALL-PROGRAM ALL-SEMESTER ?-NATIONALITY
      (Find the stand deviation of enrolments from different nations in UofA.)

    - ∗ SD ?-UNIVERSITY ALL-PROGRAM s2/2017 China
      (Find the stand deviation of enrolments in different universities in s2/2017 from China.)
  - Top-k query:
    - ∗ TOP_2 ?-UNIVERSITY MCS ALL-SEMESTER ALL-NATIONALITY
      (Find the top two universities that have most total enrolments of MCS.)
    - ∗ TOP_1 ALL-UNIVERSITY ?-PROGRAM s1/2017 India
      (Find the top program which contributes the most enrolments in s1/2017 from India.)

Outputs

- 4-dimensional data cube
  - 0-D cuboid × 1:
    All = 112
  - 1-D cuboid × 4:
    Uni[3], Prog[5], Sem[4], Nat[5]
    E.g. Prog[0] = 14 (Program == PhD)
  - 2-D cuboid × 6:
    UniProg[3][5], UniSem[3][4], UniNat[3][5], ProgSem[5][4], ProgNat[5][5], SemNat[4][5],
    E.g. ProgSem[0][2] = 13 (Program == PhD, Semester == s1/2017)
  - 3-D cuboid × 4:
    UniProgSem[3][5][4], UniProgNat[3][5][5], UniSemNat[3][4][5], ProgSemNat[5][4][5]
    E.g. UniProgNat[0][3][4] = 2 (University == UofA, Program == MSE, Nationality == Egypt)
  - 4-D cuboid × 1:
    UniProgSemNat[3][5][4][5]
    E.g. UniProgSemNat[0][1][3][2] = 14 (University == UofA, Program == MCS, Semester == s2/2017, Nationality == China)
- Standard deviation query:
  - 6.00
  - 1.00
- Top-k query:
  - Flinders UniSA
  - MCS

Web-submission instructions

- First, type the following command, all on one line (replacing xxxxxxx with your student ID):
  ```
  svn mkdir --parents -m "DDDM"
  https://version-control.adelaide.edu.au/svn/axxxxxxx/2017/s2/dddm/assignment2
  ```
- Then, check out this directory and add your files:
  ```
  svn co https://version-control.adelaide.edu.au/svn/axxxxxxx/2017/s2/dddm/assignment2
  cd assignment2
  ```

```
svn add DatacubeBuilder.cpp
svn add DatacubeBuilder.h
svn add QueryResponder.cpp
svn commit -m "assignment2 solution"
```

- Next, go to the web submission system at:
  `https://cs.adelaide.edu.au/services/websubmission/`
  Navigate to 2017, Semester 2, Distributed Databases and Data Mining, Assignment 2. Then, click Tab "Make Submission" for this assignment and indicate that you agree to the declaration. The automark script will then check whether your code compiles. You can make as many resubmissions as you like. If your final solution does not compile you will not get any marks for this solution.

- **Note:**

  1. The auto-marker script will generate a random database with around 650 records and a random query set for each online test.
  2. There will be no `main()` function in `DatacubeBuilder.cpp`. We have prepared a driver to call function `buildCuboid` in `DatacubeBuilder.cpp`. `buildCuboid` should take the database as the parameter. We use `standard naming space` in our driver.
  3. Your QueryResponder.cpp should have a `main()` function which accepts two input files in the order of database and query, then print the answer as the output.
  4. Please follow the detailed instructions in all the sample files.
  5. Your local file path will not work with our web-submission system. Do not upload your test files, we have prepared them at the web-submission server.