CS251: Final Assignment

Nov 10, 2013
2:00-5:00pm

Total Marks: 100

General Instructions

There are eight (8) phases in the assignment which together make a complete work-flow. Each of the first seven phases should produce an output, and the eighth phase binds them all together. Each phase requires an input which is produced by one or more of the previous phases.

If you cannot complete a phase, you can ask the TA for a solution to that phase. However, your output from the previous phases must comply strictly to the format required by this phase; otherwise, the solution provided by the TA will not work. You, of course, will lose all the marks for that phase.

You are allowed to use the Internet for only two purposes: *moodle* and searching. If, at any time, you are found using it for any other purpose, it will be treated as a case of copying, and will be dealt accordingly.

Submission Instructions

Create a folder with your roll number. Make sure that everything that you work on is inside this folder.

Zip the folder named after your roll number as `rollnumber.zip`. Ensure that the zipping is done correctly, i.e., it contains all the files, sub-folders, etc.

Upload this zip file to moodle *in front of* a TA *after* the scheduled time ends. Only the contents of this zip file will be graded.

Phase-1: Octave (10)

Write an **Octave** script to create a list of 100 roll numbers randomly for 100 students. The roll number must be a 5-digit positive integer.

Assume a set of 9 courses, 1 to 9. Assign 3 courses to each student randomly from this pool. A student cannot enroll in the same course more than once.

For each course, there are three components, exams, assignments and projects. For each student and each course she is enrolled in, generate marks randomly for each component between 0 and 50 (both inclusive). The marks should be integers.

Export the entire data as a *csv* (comma-separated-values) file `phase1.out`. The columns should comply to the following format.

roll, course1, course2, course3, asgn1, proj1, exam1, asgn2, proj2, exam2, asgn3, proj3, exam3

This entire **Octave** script should be named `phase1.m`.

Phase-2: Perl (15)

Write a **Perl** script that creates the table named `marks` in the **MySQL** database `test`. The `marks` table has the following fields, all of which are integers:

roll	course	assignment	project	exam

The primary key of this table is `roll` and `course` together.

The script then parses the csv file produced in phase-1, i.e., `phase1.out`, and inserts the data appropriately in the table `marks`.

Additionally, it creates another table `names` in the same database according to the format

roll	name

with `roll` as the primary key. It then inserts all the names from the file `names.txt` along with the roll numbers. The name corresponding to the i^{th} line in the file should have the i^{th} roll number in ascending order.

This entire **Perl** script should be named `phase2.pl`.

Phase-3: MySQL (15)

Write a **MySQL** script to modify the table `marks` created in phase-2 by adding two columns `total` (integer) and `grade` (string).

Update the `total` field in each row as the total marks of the three components in the course.

Update the `grade` field according to the following formula:

$$grade = \begin{cases} A* & \text{if} \quad total \geq 95\% \\ A & \text{if} \quad 80\% \leq total < 95\% \\ B & \text{if} \quad 60\% \leq total < 80\% \\ C & \text{if} \quad 45\% \leq total < 60\% \\ D & \text{if} \quad 30\% \leq total < 45\% \\ F & \text{if} \quad total < 30\% \end{cases}$$

Calculate the number of students for each grade category in each course. Collect the output in the file `phase3.out`.

This entire **MySQL** script should be named `phase3.sql`.

Phase-4: Formatting (Bash / Perl / Octave) (10)

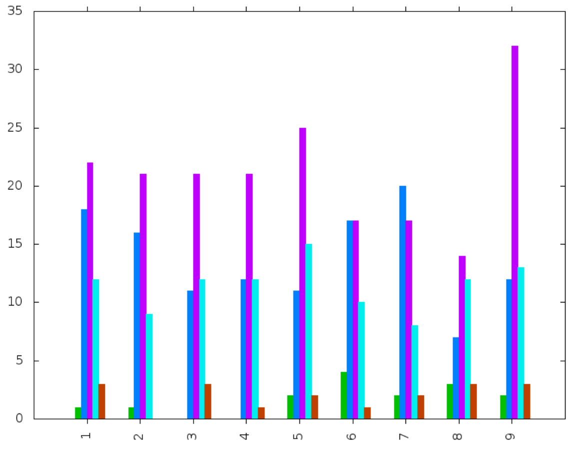
Write a script in either **Bash** or **Perl** or **Octave** that uses the output of the last query of phase-3, i.e., `phase3.out`, and prepares the data in a format that the `GnuPlot` script of phase-5 can consume. The output of this phase is the file `phase4.out`.

This entire script should be named `phase4.sh` or `phase3.pl` or `phase3.m` depending on what you have used.

Phase-5: GnuPlot (10)

Write a **GnuPlot** script that plots the output of phase-3 formatted correctly in phase-4, i.e., the file `phase4.out` as a histogram.

For each course, the plot shows a histogram of the number of students in each grade. The scheme of the figure should be a set of 6 histogram bars in 9 courses as in



The plot should contain proper title, legends, fonts, scale, etc. and should be named `phase5.png` or `phase5.jpg` or `phase5.eps` or `phase5.pdf`.

This entire **GnuPlot** script should be named `phase5.gnu`.

Phase-6: LaTeX (15)

Write a **LaTeX** file that describes the entire process so far including how the marks for students are generated, how the grades are assigned, etc. It should contain your analysis about the marks distribution as well.

You can use the template file `analysis.tex` and `analysis.bib`. However, the files may contain errors and, thus, may not compile as is.

Ensure that your submitted **LaTeX** file is complete and contains tables, figures (the plot produced in phase-5), equations, referencing, etc.

Also, cite *all* the tools and the tutorials that you have used using **BibTeX**.

The **LaTeX** file should be named as `phase6.tex` and the **BibTeX** file should be named as `phase6.bib`. The output of this phase is `phase6.pdf`.

Phase-7: Bash (15)

Write a **Bash** script to produce a ranked list of all students according to their total marks for every course. Extract the raw data out from the **MySQL** table `marks` *without* sorting the students based on `total`.

Segregate, sort and find the ranks using **Bash**. You can use auxiliary **Bash** tools such as **Grep**, **Sed**, **Awk**, etc.

There will be a single output file. that contains the ranked lists of the 9 courses *in order*. It should be named as `phase7.out`. The columns in each file should be

course ⟨ tab ⟩ roll ⟨ tab ⟩ name ⟨ tab ⟩ rank ⟨ tab ⟩ total

This entire **Bash** script should be named `phase7.sh`. If you need an intermediate **MySQL** script, it should be named `phase7.sql`.

Phase-8: Automation (Bash) (10)

Write a **Bash** script that connects the first 7 phases together. This file should contain the commands of how to run each phase, what it inputs, what it outputs, etc.

Also, create symbolic (soft) links of the output of phase-1 as the input of phase-2, etc. In other words, the file `phase2.in` should be a soft link to `phase1.out`. Changing `phase1.out` thus automatically changes `phase2.in`.

Create similar soft links of `phase3.out` to `phase4.in` and `phase4.out` to `phase5.in`.

This entire **Bash** script should be named `phase8.sh`.

If all the phases are done correctly, just running this file should, thus, complete the entire assignment.