CS258 Assignment (2016/17)

Submit a single file named `Assignment.java`
via Tabula by **noon on Thursday 8th December 2016**

The storage of teaching-related data at a Midlands university is governed by the database schema given in Figure 1. To identify students, the university uses 4-digit identifiers (Student_id) in STUDENT, which makes it possible to distinguish students with identical names. Every module is controlled by a single university department, as detailed in the MODULE table. Different modules have different module codes, but also different names. The fact that a module runs in a particular year is noted in HISTORY along with the name of its organiser. A student can be examined on a specific module at most once: EXAM stores the year in which the exam took place (if applicable) along with a numerical score. HISTORY and EXAM records may only contain data about modules listed in MODULE and students from STUDENT. Similarly, EXAM and HISTORY should be consistent in that an exam entry should only mention modules that actually ran. We assume that each module is examined in the same year in which it was delivered.

Initialisation (20%)

Write SQL code to create the tables from Figure 1. Use the following types for the respective attributes: `char(4)` (Student_id), `varchar(6)` (Module_code), `smallint` (Year, Delivery_year, Exam_year, Score), `varchar(30)` (any attribute of the shape 'Bla_name').

- Specify key and referential integrity constraints so as to capture the description given above as accurately as possible. Provide justifications for your choices.

Include the SQL code (and your justifications) as a comment at the beginning of the submitted `.java` file.

Application (80%)

- Write (and test) a Java program that repeatedly displays the menu below and allows the user to select one of the options. Each option is specified below along with the expected output format. Document your solutions and justify any decisions you make in comments.

STUDENT

Student_name	Student_id	Course_name	Year

MODULE

Module_name	Module_code	Department_name

HISTORY

Module_code	Delivery_year	Organiser_name

EXAM

Student_id	Module_code	Exam_year	Score

Figure 1: Database schema

```
MENU
(1) Modules by student
(2) Ghost modules
(3) Popularity ratings
(4) Top student(s)
(5) Harshness ranking
(6) Leaf modules
(7) Risky exams
(8) Twisted prerequisites
(0) Quit
Enter your choice:
```

Option 1: Modules by student

For each student, print a single line containing Student_id followed by a colon, a space and a list of module codes (separated by spaces) that the student has taken exams in.

```
Enter your choice: 1
8125: CS134 CS432 CS465
1235: IB321 CS432 CS215 CS765
```

Option 2: Ghost modules

Print on a single line the list of all module codes (separated by spaces) for which no exam has ever taken place.

```
Enter your choice: 2
CS456 CS654 MA543
```

Option 3: Popularity ratings

Print out the list of all module names (one name per line) in decreasing order of popularity. Popularity is taken to mean the number of students that have been examined on the module.

```
Enter your choice: 3
Programming for Maths
Relativity in Economics
Geometry in Motion
English Literature
```

Option 4: Top student(s)

Print out (in any order) the names of all students with the highest average examination score (one name per line).

```
Enter your choice: 4
Alex Jones
Roger Federer
Andy Gordon
```

Option 5: Harshness ranking

List the names of all module organisers in decreasing order of harshness. Harshness is quantified by the average exam score calculated across all of the modules that the person was in charge of. Use one line per name.

```
Enter your choice: 5
Prof. John Harsh
Dr Jenny Mild
Prof. Helen Lenient
```

Options 6-8

For the last three options we need to create a new table, which specifies dependencies between modules.

PREREQUISITES

Module_code	Prerequisite_code

Option 6: Leaf modules

List on a single line (in any order) the codes of modules that do not have any prerequisites. Use a space to separate the items.

```
Enter your choice: 6
CS118 CS101 CS100 MA104
```

Option 7: Risky exams

List on a single line the Id's of all students who have been examined on a module but have not taken exams in all of the prerequisites. Exam performance does not matter in this question. Use a space to separate the items.

```
Enter your choice: 7
1001 1235 2345 3421
```

Option 8: Twisted prerequisites

Note that nothing prevents us from specifying circular dependencies among prerequisites. For example, a module could be specified as its own prerequisite! And there are many more ways of creating circular dependencies! Naturally, in such cases, the prerequisites cannot be satisfied. This option should print (on a single line) the codes of all modules with unsatisfiable prerequisites separated by spaces.

```
Enter your choice: 8
PH124 MA435
```

Final remarks

Please make sure your programs work on daisy. Keep a copy of everything you submit! You may discuss with fellow students the material covered in lectures and seminars, but you are not allowed to collaborate on the assignment. The University of Warwick takes plagiarism seriously, and penalties will be incurred if any form of plagiarism is detected. Copying, or basing your work on, solutions written by people who have not taken this module is also counted as plagiarism. This includes material that has been downloaded from the internet.

Good luck!