CS348: Introduction to Database Systems

(Fall 2019)

Assignment 3 (due Tuesday November 12 by 5pm via submit)

You are to use your accounts and DB2 to implement the **RegistrarInfo** system. **RegistrarInfo** consists of there application programs with simple command line interfaces. The requirements for the programs are given in the following sections. A specification of the underlying schema for the database is the same as for Assignments 1 and 2 and appropriate DB2 `create table` commands are provided on the assignment web page. Note that all submissions must use this schema. Either C (or C++) must be used together with SQL and the static embedded SQL protocol to implement **RegistrarInfo**.

1 Assignment Submission

A submission of the following items are to be made before or on the assignment due date:

1. Source of each of the application programs comprising **RegistrarInfo**. These are to be submitted online in files named `schedule.sqc`, `conflicts.sqc`, and `gpa.sqc`.

2. A shell script named `compile`. Typing "`./compile`" should suffice to compile all application programs.

2 Applications Programs

2.1 `schedule`

This application is to print a schedule of current classes for each student and/or professor whose `snum`s and/or `pnum`s are given on the command line (schedules for multiple persons can be requested). Assume that the `snum`s and `pnum`s are *social insurance numbers* (SIN) represented as integers. Note that there may be persons that are *both* a professor and a student in the current term: in that case a *combined schedule* should be produced. For each SIN the program should print the following information:

```
Schedule for "name" (sin) and term <term>
----
MONDAY:    cname (cnum) section time room
...
MONDAY:    cname (cnum) section time room
TUESDAY:   cname (cnum) section time room
...
----
```

The schedules must be in order of days and, within days, in order by time. An example output may look as follows:

```
$ ./schedule 123456789
Schedule for "David Toman" (123456789) and term <F19>
----
TUESDAY:   "Intro to Databases" (CS348) 1 10:00 MC4059
TUESDAY:   "Intro to Databases" (CS348) 3 11:30 MC4059
TUESDAY:   "Intro to Databases" (CS348) 2 16:00 E21732
THURSDAY:  "Intro to Databases" (CS348) 1 10:00 MC4059
THURSDAY:  "Intro to Databases" (CS348) 3 11:30 MC4059
THURSDAY:  "Intro to Databases" (CS348) 2 16:00 E21732
----
```

In the case when the SIN on the command line does not identify a student nor a professor, the application should output

```
Schedule for (sin) does not exist.
----
```

2.2 conflicts

This application is to find conflicts in the schedule of current classes for a single student and/or professor whose `snum` and/or `pnum` is given on the command line and fix them.

The policy is as follows: Two distinct schedule records conflict if they are scheduled at the same day and time in the same term; each such conflict should be resolved by applying conflict resolution policies in the following order:

1. schedule for a professor's class assignment takes precedence over schedules for student's class assignment (in the cases when a person both teaches and takes classes),
2. schedule lower for level classes in the same department takes precedence over schedule upper classes,
3. schedule for a section of a class with a lower number is preferred over a section of the same class with higher number, and
4. schedule conflicts for classes in different departments or conflicts due to concurrent schedule in multiple rooms cannot be automatically corrected.

Note that the application *does not* need to worry about overbooking rooms. For each conflict found the application should print out the pair of conflicting schedule records as follows:

```
DAY      time: cnum1(section1) room1 -- cnum2(section2) room2 -- <resolution>
```

where `<resolution>` is one of "`cnum(section) deleted`" or "`cannot be resolved`". An example output may look as follows:

```
$ ./conflicts 349143576
TUESDAY     10:00: PM245(1) MC4556 -- CS123(2) MC1111 -- cannot be resolved
WEDNESDAY   11:30: CS348(1) MC1234 -- CS448(2) DC3245 -- CS448(2) deleted
THURSDAY    16:00: CS245(1) DC2289 -- CS245(2) MC3245 -- CS245(2) deleted
FRIDAY      16:00: CS245(1) MC4556 -- CS245(1) MC3245 -- cannot be resolved
```

The conflicts should be presented in chronological order and report invalid ids similarly to the output of `schedule`.

2.3 gpa

This application is to report the overall GPA and the GPA for each past term for a student whose `snum` is given on the command line as follows:

```
GPA for student "name" (sin) is overall-gpa
term gpa
...
```

where the terms are ordered chronologically and the gpa is calculated to one decimal place. An example output may look as follows:

```
$ ./gpa 345612345
GPA for student "John Doe" (345612345) is 77.2
F02 75.3
W03 77.1
S03 76.3
W04 80.0
```