## Overview

Your assignment is to design a course scheduling system. You will design the database, insert some sample data, and implement a set of required features. Each feature will be implemented as one or more Oracle PL/SQL procedures/functions. You do **NOT** need to write a graphic user interface.

**Assumptions:**

You can make the following assumptions in this project.

1. The system needs to store information about academic departments. Each department has an ID and a name.
2. The department has a few programs. E.g., the IS department as IS BS program, IS BTA program, IS MS program, HCC MS program, etc. Each program has a program ID, program name, and program type (1 for undergraduate and 2 for master’s, 3 for PhD).
3. Each program has a number of courses. Each course has course id, course name, number of credits, grading format (letter, pass), required (1 is required for program, 0 is elective), room type (whether it needs computer lab or regular room), number of sections (number of sections needed for a year and semester), section size (number of students per section, status (1 open, 0 close). If a course is scheduled for a given year and semester, it can have 1 or more sections. E.g., IS 620 has one section in Fall 2019 but IS 633 has two sections in Fall 2019.
4. Each course may have one or more courses as prerequisites. A student needs to complete all prerequisites before taking that course.
5. The system stores information about buildings. Each building has a building ID and a name.
6. Each building has a number of classrooms. Each classroom has room id, room name (e.g., ITE 468), number of seats, and room type (1 as computer lab, 0 as regular room).
7. Each department has a number of instructors. Each instructor has an ID, name, and is either part-time or full-time.
8. Each instructor needs to teach a certain number of courses in a given year and semester. This is called course load. For example, Dr. Chen needs to teach two courses in Fall 2019. Note that multiple sections of the same course will be counted as one course. E.g., if Dr. Chen teaches two sections of IS 633 it is still counted as one course (he will get a TA but you don’t need to consider that). So he needs to teach another course to fulfill his load. The system also needs to store the number of courses actually assigned to that instructor.
9. Each instructor can specify a list of courses that he or she can teach as well as the number of sections for each course for a given year and semester. E.g., Dr. Chen may specify he can teach one section of IS 620 and two sections of IS 633 for fall 2019.
10. Each instructor can specify up to two day’s of a week that he or she cannot teach. E.g., Dr. Chen can specify he cannot teach on Thursday, and Dr. Karabatis may specify he cannot teach on Thursday or Friday.
11. Each course section is scheduled at a time block. These time blocks are not overlapping. Each time block has an ID, start time, length (2.5 hour or 1 hour 15 minutes), day1, day2 (day 1 and day 2 are the days of week the class will be offered). 1 means Monday, 2 means Tuesday, and so on. If the class is 2.5 hour long, it will be scheduled on one day (so day2 is null). If the class is 1 hour 15 minutes, it will be scheduled on two days. E.g., one time block is 10-11:15 am Monday and Wednesday. Another time block is 4:30 pm to 7 pm Tuesday.
12. One or more sections of a course can be scheduled at a given year and semester. The schedule includes schedule id, course ID, section id, instructor id, year, semester, capacity (#of students allowed in the section), time block id, room id, waiting list capacity, and status (open = 1, 0 = full).
13. Each student has a student ID, name, and enrolled in a program.
14. Each scheduled section can have a waiting list. The list keeps IDs of any student on the list, the schedule ID, and position of the student on the waiting list.
15. A student can register for a scheduled course section. There can be three different registration status: 1 means enrolled, 2 means dropped, and 0 means wait listed. A student will have a grade for that course section: (4: A, 3: B, 2: C, 1: D or pass, 0: F).
16. Special permission can be given to students to allow them to enroll in a section that is either closed or the student has not taken prerequisites. The system stores ID of student, ID of the scheduled section, and the type of special permission (1: enroll in closed class, 2: enroll without prerequisite).

**Features (those with \*\*\* are more difficult and can be counted as 2-3 features when assigning features among group members)**

## Course scheduling features to be implemented

\*\*\* Given the ID of a course, year, and semester, assign the course to instructors who are willing to teach that course. Please first check whether the course needs more sections (compare number of sections needed for the course from the course table to the number of sections already scheduled). If so assign new sections to instructor who is willing to teach that course and create schedule for this course (you can leave classroom unassigned). Othewise print a message there are enough sections.

You need to make sure that   
1) enough course sections are assigned (the number of sections is in course table). Print a message if this is not possible.

2) only instructors who are willing to teach that course will teach the course.

3) The instructor's course load is not exceeded.

4) the new schedule's capacity equals section size of the course and waiting list size is 10.

You can use the following greedy method to assign the courses.

You can also use a temporary table to store intermediate results (e.g., the set of instructors, the number of sections they are willing to teach, the course load, and their weight as described in step 4)

Psuedo code of the method to assign course c.

1) Find a set I of instructors who are willing to teach course c and have not reached their course load (the number of courses they suppose to teach) and have not scheduled to teach that course.

2) If I is empty print an error message “not enough sections are assigned due to lack of faculty”. Stop the procedure.

3) Sort instructors in I in descending order of a weight. The weight = min(#of sections of c an instructor is willing to teach, #of unassigned sections of c) multiply (the instructor's course load - #of assigned courses to that instructor). E.g., suppose both instructor A and B need to teach two courses. A is assigned 1 course, and B is assigned nothing so far. Both A and B are willing to teach 2 sections of c. c has 3 unassigned sections. Now A's weight = min(2,3)\*(2-1)=2, B's weight = min(2,3)\*2=4. So the sort order is B, A.

4) for each instructor i in the sorted I

5) Assign c to i. The number of sections is the minimal of unassigned sections of c and number of sections i is willing to teach. Create a course schedule for this assignment (you don't need to assign classroom or time at this moment)

6) exit when there is no more section to assign.

7) end for;

8) If there is still sections to assign print there is not enough instructors.

E.g., suppose course c needs 3 sections, instructor A and B both are willing to teach 2 sections of c. Both A and B have a load of two courses and A has been assigned one course so far and B has been assigned 0.

The method will sort A, B by their weight. A's weight = 2, B's weight = 4. So the first round B is assigned to teach 2 sections of c. In the next round only A remains so A will be assigned to teach the remaining section of c.

**Implemented as:**

Input Includes Course ID , Year, Semester

Output: First check whether the course needs more sections (compare number of sections needed for the course from the course table to the number of sections already scheduled). If so assign new sections to instructor who is willing to teach that course and create schedule for this course (you can leave classroom unassigned). Otherwise print a message there are enough sections.

You need to make sure that   
1) enough course sections are assigned (the number of sections is in course table). Print a message if this is not possible.

2) only instructors who are willing to teach that course will teach the course.

3) The instructor's course load is not exceeded.

4) the new schedule's capacity equals section size of the course and waiting list size is 10.

Exec assign\_course\_ins\_willing(121, 2018,’ Spring’);

**Explanation:**

By default each course has a certain number of sections for all semesters. E.g., IS 410 may have 3 sections per semester. This information should be stored in course table and you don't have to change it in feature 6.

The check is to deal with the case when feature 6 is called multiple times on the same course, year, and semester.

E.g., if feature 6 is first called to assign IS 410 in spring 2020, the feature will assign all 3 sections to some instructors.

However, when feature 6 is called again with the same input: IS 410 in spring 2020, the feature should not assign new sections as all 3 are already assigned.

1. \*\*\* Assign room and time to a scheduled section. Input includes a schedule id. First check whether the schedule id is valid. If not print an error message. Next check whether the scheduled section already has a room and time block. If so print an error message saying that the course is already assigned. Otherwise find a room and a time block pair that satisfies the following conditions

1) The day of week is not one of the blackout days of the instructor.

2) The instructor is not teaching at that time block.

3) The room has no class scheduled at that time block.

4) If the course has multiple sections, no other section is at the same time block.

5) If the course belongs to a graduate program, choose a time block that is after 4:00 pm.

6) The number of seats in the room is greater than or equal to class capacity

7) If the course requires in a computer lab, the room must be of computer lab type.

If there are multiple (room, time block) pairs satisfying the condition, then you can compute a score for each pair and assign the section to the pair with the lowest score. If there are no pair satisfying the above conditions, print out cannot find such pair.

The score is computed as following. First find out the total number of sections already scheduled in that room at the same year/semester.

Next find out the total number of sections scheduled at the same time block at the same year/semester.

The score = product of these two.

E.g., if a room A has 3 sections scheduled and a room B has 4 sections scheduled, and a time block T1 has 2 sections scheduled and a time block T2 has 3 sections scheduled.

Suppose two pairs (room A, T1) and (room B, T2) satisfy the above conditions (1 to 7).

Score of (room A, T1) = 3 \* 2 = 6

Score of (room B, T2) = 4 \* 2 = 8

The class should be assigned to room A at T1.

Again, you can create a temporary table to store the room, time block pair and score.

**Statistics Features**

1. Print enrollment statistics for a department given department id, year, and semester.
   1. Please print out total number of students enrolled for at least one course (do not count dropped or wait listed cases) in any courses in that year and semester in programs in the department.
   2. Please also print out total number of courses in that department, year and semester, total number of course sections, #of students enrolled and wait listed in each course section along with course id, course name, section id.
2. \*\*\* Identify for a given year and semester, the top k courses with the longest waiting list, the top k rooms with the fewest scheduled class sections, and the top k time blocks that have the fewest scheduled class sections. Input includes year, semester, and k. Please print out the id and name of classes, their waiting list length. For rooms, please print out room id, room name, and #of scheduled class sections. For time blocks print out time block id, days of the week, and start time.

Please pay special attention to the case when a time block has no class scheduled or a room has no class scheduled (so they should be in the result).