

Group 6

Features Description:

Feature 1:

Add a new course. Input includes course name, program id, number of credits, grading format, whether the course is required, room type, number of sections. The feature checks if a course with the same name exists in the program. If so, update the existing course with the values of the input parameters. Otherwise, insert a new course and generate a new course id. Please print out the new course id.

Input : Course name , Program ID , Number of Credits, Grading Format , Course Requirement, Room-Type, Number of Sections.

Output : Check if same course name exists. If yes , update the course of input . If no , generate a new course and a new course ID

Exec: add_course('Advanced Database Management System',1,4,'GPA',1,'Computer Type',1)

Feature 2:

Add a new instructor. Input includes instructor name, department id, and instructor type (full time or part time). Please check if an instructor with the same name exists at the same department with same instructor type. If so print an error message. Otherwise insert the instructor and print out the generated instructor id.

Input : Instructor name, Department ID, Instructor type(full or part)

Output: If instructor present with the same name , department , and type, print Error. If not Add a new instructor and a new Instructor ID

Exec: add_instructor('Prof.Zhiyuan Chen',1,'Full-Time')

Feature 3:

Allow and instructor to enter teaching preferences

Input: instructor id, year, semester, course load, instructor_course, no_of_sections, not_available.

Output: if the instructor id is valid, the list of course is smaller than the course load, the number of blackout days is over two. Print an error message if one of these cases occurs. Otherwise insert these input to appropriate tables.

Exec allow_instructor (1,'2019', 'Fall 2019',2,'Advanced Database Project',2,'Monday');

Feature 4:

Allow a student to search for all courses offered in a year, semester, and program id.

Group 6

Input: course_credits, grading_format, schedule_id, section_id, instructor_name, classroom_name, days_of_class, start_time, end_time.

Output: If the class is open or full. Whether input program is valid.

Exec course_search (3,'A',1,1,'Prof.Chen','ITE 201',1,'4:30 PM','7:00 PM');

Feature 5 :

Compute the number of assigned courses for all instructors given a year and semester. This feature checks the number of scheduled courses taught by each instructor for that year and semester and store this number in the course_load table. Note that multiple sections in the same course is counted as one course. This feature will be called by other features.

Input:

Output: the number of assigned courses for all instructors given a year and semester.

Exec: assigned_courses

Feature 7:

Assign courses in a department for a given year and semester. This feature can call feature 6 to assign a course to instructors. This feature will go through every program in the department and tries to assign required courses first. If there are still instructors who have not reached their course load, this feature will assign the electives in these programs. Finally it will check whether all instructor has been assigned enough courses and print out the names of instructors who have not.

Input:

Output: Assign courses in a department for a given year and semester. Check all instructor who are assigned and names of instructor who are not

Exec: assigned_instructors

Feature 8

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

Group 6

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

Input: scheduled_id

Output: Check if scheduled_id is valid or not then check if the scheduled_id is assigned a room and time block if not then assign a room and time block.

Exec book_room(1)

Feature 9:

Assign rooms and time blocks for courses in a department given a year and semester. Input includes department id, year, and semester. This feature can call feature 8 to assign a section to a room and time block. This feature will go through every scheduled section in that department, year, and semester and assign for each section a room and time block. Please first schedule graduate courses and then undergraduate ones. For courses are all graduate or undergraduate, schedule required courses first.

Input : Department id , semester, year

Output: To assign a section and time block. It will check every schedule section in department , year and semester and assign rooms and time block.

Exec: assigned_rooms(1,'Fall-2019',2019)

Feature 10:

Enter special permission for a student and a scheduled section. The input includes student id and schedule id. First check whether both are valid. If not print a message. If both are valid enter special permission to special permission table. The special permission includes 1)

Input: student_id, scheduled_id

Output: Check if student_id and scheduled_id is valid then insert permission(1) into permission table.

Exec permission_for_student(1003, 3);

Feature 12: Allow a student to register a course given schedule id.

Input: schedule_id, student_id

Output: a) check whether schedule_id and student_id is valid otherwise print an error message.

b) whether student has taken the same class before. Retaking is not allowed unless the student got a D before. Print an error message.

c) whether the student has taken prerequisites of this class. If not and is student doesn't have special permission to enroll without prerequisite, print an error message.

Group 6

d) Finally check whether the class is still open. If so or the student has special permission to enroll in a closed class, add the student to registration table with enrolled status. Please check whether the class reaches capacity now. If so, update the status of schedule. If the class is full and there is still room on waiting list put the student on the next position of the waiting list. If the waiting list is full as well, print an error message.

Exec allow_student (1,2);

Feature 13:

Allow a student to drop a course. Input includes student id and schedule id. First check whether the student has registered for that scheduled section (the status could be enrolled or wait listed). If not, print a message the student is not registered with that course. Otherwise if the student is on wait list, remove the student, change registration status to dropped, and move up anyone after the student on wait list.

Input : Student id and Schedule ID

Output: Check If the student is registered for the schedule section. If not , the student has not registered for the course. If the student is found to be on wait list , remove the student ,change registration status to drop or move student up the order in the wait list.

Exec: drop_course(1001,1)

Feature 14:

Allow a student to print a course schedule for a given year and semester. Input includes student id, year, and semester. Print course id, course name, section id, and status of all courses the student has registered in that year and semester. For any course the student is on the waiting list, also print out the waiting list position.

Input: student_id, year, semester

Output: check if the student_id entered is on waitlist. If on waitlist the print the waitlist position for that course else and print all the course related information of the student and message saying not on waitlist.

Exec student_schedule(1001, 2019, Fall);

*****Feature 16:**

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.

Group 6

Input: Year , Semester , K(Top course with long waiting list)

Output: We can print id and name of classes, waiting list length. For room we print out room id, room name and number of scheduled class section. For the time block process print out block id , days of the week occupied and start time.

Exec: schedule_details()