Specifications of the project:

Feature 1: Add a course

Input: Course name, Program id, Number of Credits, Grading format, whether the course is required, room type, number of sections

Output: if the course with same name does not exists, print out new course id. Otherwise, just update that course with the input.

Exec add\_course ('Advanced Database Project',2,3,1,1,1,1);

Feature 2: Add a instructor

Input: Instructor name, Department id, Instructor type (full time or part time).

Output: 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.

Exec add\_instructor (' Dr. Chen',1,1);

Feature 3: Instructor’s teaching preferences

Input: Instructor id, Year, Semester, Course load, a list of courses the instructor is willing to teach, the number of sections.

Output: checking the condition for validation of the instructor ID, course load is greater than list of courses and the leave days for the instructor should be at least 2 days. Print an error message if one of these cases occurs. Otherwise insert these inputs to appropriate tables.

Exec add\_instructor\_pref (1,date’2019-08-28’,1,2,2,1);

Feature 4: Courses offered to students

Input: Year, Semester, Program id.

Output: prints out for each scheduled course section the name of the course, number of credits, grading format, schedule id, section id, name of instructor, name of classroom, days of class, start and end time of each class, and whether the class is open or full. The results should order by course id and section number. Check whether input program id is valid.

Exec add\_stu\_course(date’2019-08-28’,1,2);

Feature 5: Checks the number of courses taught by each instructor

Input: List of the course for the Instructor, Year, Semester

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

Exec add\_number\_of\_course(2,date’2019-08-28’,1);

Feature 6: Add course to instructors schedule who are willing to teach an extra course.

Input: ID of a course, year, and semester

Output: Checking whether the course needs more sections. If so, assign new sections to instructor who is willing to teach that course and create schedule for this course. Otherwise print a message there are enough sections.

Exec add\_extra\_course(4,date’2019-08-28’,1);

Feature 7: Goes through every program in department and assigns the required courses first if not done so.

Input: Course ID, Program ID, instructor ID, year, and semester

Output: it will check whether all instructor has been assigned enough courses and print out the names of instructors who have not.

Exec add\_course\_dept (3,2,1,date’2019-08-28’,1);

Feature 8: Assign room and time to a scheduled section

Input: schedule id

Output: Checking 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.

Exec add\_sroom\_time (3);

Feature 9: Assign rooms and time blocks for courses in a department

Input: department id, year, and semester

Output: Assign for each section a room and time block.

Exec add\_droom\_time (1, date’2019-08-28’,1);

Feature 10: Special permission for student and schedule section

Input: student id, schedule id.

Output: check whether both are valid. If not print a message. If both are valid enter special permission to special permission table.

Exec add\_permission (3, 3);

Feature 11: Checks prerequisite of student

Input: student ID and schedule ID

Output: Returns one if the student has taken prerequisites of the scheduled course, 0 otherwise.

Exec add\_prerquisite (3, 3);

Feature 12: Student registers a course given a schedule ID

Input: schedule id and student id

Output: 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 add\_register\_course(3, 3);

Feature 13: Allows a student to drop a course

Input: student id and schedule id

Output: Student’s registration status should be checked and print a message ‘print a message the student is not registered with that course’, if student is not registered. If student is on waitlist remove the student, change registration status to dropped, and move up anyone after the student on wait list. If the student is enrolled, automatically enroll the first student on waiting list and update the waiting list. If there is no student on waiting list, set the course to be open if the course capacity is not reached.

Exec add\_drop\_course (3, 3);

Feature 14: Allows a student to print course schedule

Input: student id, year, and semester

Output: Print course id, course name, section id, and status of all courses the student has registered in that year and semester.

Exec add\_course\_sch (3, date’2019-08-28’,1);

Feature 15: Enrollment Statistics

Input: department id, year, and semester

Output: print out total number of students enrolled for at least one course in any courses in that year and semester in programs in the department. print out total number of courses in that department, year and semester, total number of course sections, number of students enrolled and wait listed in each course section along with course id, course name, section id

Exec add\_enroll(1, date’2019-08-28’,1);

Feature 16: Longest waiting list, rooms with fewest schedule class sections, time blocks with the fewest schedule class sections.

Input: year, semester, and k.

Output: print out the id and name of classes, their waiting list length, room id, room name, and #of scheduled class sections. For time blocks print out time block id, days of the week, and start time.

Exec add\_k(date’2019-08-28’,1,k);