DATABASE SYSTEMS PROJECT-2

Design Document

The project was a replica of a university database system which has the details of all the students, classes, courses and classes in which the students are enrolled in.

In the beginning there is a login page. User has to enter the username and password to login. Once the user has logged in, all the modules appear on the top bar and the functions under each module appear on the left hand side of the page. The user can switch between the modules just by clicking on the module button. To switch between the functions of the module the user can click on any function that the module carries out. For eg. the user wants to see all the courses offered. To do so he can click on the courses button on the top and then click on the display courses at the side. All the courses offered will be displayed on the screen. In some cases the user will have to enter values to get specific data. For eg. To find the details of any student the user will have to enter the student_id in the box. Find_student procedure will be called and the details of the student will be displayed. If incase there is no student with that Id then it will show error. So when ever user clicks on any function in the module, the respective procedure is called and the result is displayed. The modules along with its procedures are explained below.

The project has 5 modules :

- **Students module**: In Students module, we have used 4 procedures to display, add, search and delete a student entry. The procedures are as follows:
    1. PROCEDURE display_students : This procedure is used to display all the students.
    2. PROCEDURE add_student : This procedure takes data from the user as input which is entered here. The inserted data is stored in the students table.
    3. PROCEDURE find_student : This procedure takes the SID as the input and returns all details of the student. If the SID is not present it shows invalid SID error. Also if the student is enrolled in any class then it shows those details as well and if not then it displays only student’s personal details.
    4. PROCEDURE delete_student : This procedure also takes SID as an input and deletes that particular students record. It first checks if the SID is present or not. If it’s not present then it shows invalid SID error and if that SID is present then it deletes the entry of that student.
- **Courses module**: In Courses module, we have used 3 procedures, which are as follows:
    1. PROCEDURE display_courses : This procedure shows all the courses that are offered.
    2. PROCEDURE display_prerequisites: This procedure displays all the courses and their prerequisites.
    3. PROCEDURE find_course : It takes dept_code and course_no as input and displays the details of that specific course. It first checks if the course is present and if it doesn’t find the course then it shows course not found error. If it finds the course then it displays all the prerequisites of that course along with that course.
- **Classes module**: In Classes module, we have used 2 procedures, which are as follows:
    1. PROCEDURE display_classes : It displays all classes with it’s details.
    2. PROCEDURE find_class : This procedure returns the class details and details of students who have taken that class. It takes classid as input from user. It checks if classid is available in DB, if not it shows invalid classid as error. It also checks how many students have taken that course. If none of the students have taken that course then it displays only the class details and if there are students in that class then it displays the students details as well.
- **Enrollments module**: In Enrollments module, we have used 3 procedures, which are as follows:
    1. PROCEDURE display_enrollments : It displays the SID of the students and classid of the classes they are enrolled in. It also displays the grade received by the student in that class.

2. PROCEDURE enroll_student : This procedure takes SID, CID and CLASSID as input from the user and checks a number of conditions before enrolling the student.
    - Checks if the sid is valid. If not returns invalid sid error.
    - Checks if the classid is valid. If not returns invalid classid error.
    - Checks if the limit of the class is not reached. If yes then it returns class full.
    - Checks if the student has already taken that class. If yes then it returns class already taken.
    - Checks if the student has already taken 3 courses. If yes then it returns courses overloaded.
    - Checks if the student has completed all the prerequisite courses with a minimum of C grade. If not then it returns prerequisite courses not completed.
    - If all the above conditions are false and if the student has taken only 2 classes then it enrols the student for that class.
3. PROCEDURE drop_enrollment : This procedure also takes SID, CID and CLASSID as input from users. It checks a few conditions before dropping the student from that class.
    - Checks if the sid is valid. If not returns invalid sid error.
    - Checks if the classid is valid. If not returns invalid classid error.
    - Checks if the student is enrolled in that class, if not then it returns student not enrolled.
    - Checks if that class is a prerequisite for any other class the student has taken then it does not permit the drop as it's a prerequisite class.
    - If the above conditions are false then it drops the student from that class.
    - After dropping it checks all the other classes the student is enrolled in.
    - It also checks the number of students still enrolled in that class, if none then it returns class does not have any students.

- **Log module**: In Log module, we have used 1 procedures, which are as follows:
    1. PROCEDURE display_logs : It displays the log of activities- who altered which table , the time and date and also the operation carried on that table.

There are also triggers that are triggered before carrying out any procedure or after execution of a procedure.

- TRIGGER students_updated_insert : This triggered is activated before every insert operation is done on enrolments. It increases the class size by one and then enrols the student.
- TRIGGER students_updated_delete : This triggered is activated before every student is drops the class. It decreased the class size by one and then drops the student.
- TRIGGER delete_student_enrollments : Before every student entry is deleted, this trigger deletes all the classes the student is enrolled in and then deletes the student entry.
- TRIGGER enrollments_updated_insert : Before any insert operation is done on enrolment, this trigger is triggered and it updates the log table.
- TRIGGER enrollments_updated_delete : Before any student is dropped(deleted) from the class(enrolments table), this trigger is triggered and it updates the log table.
- TRIGGER enrollment_added_classes : Before a new student is added, this trigger updates the log table.
- TRIGGER enorllment_dropped_classes : Before a new student is deleted, this trigger updates the log table.

Also a sequence (SEQUENCE generate_logid) is used to generate the logid.