SCHEMA DESIGN -

Entities:

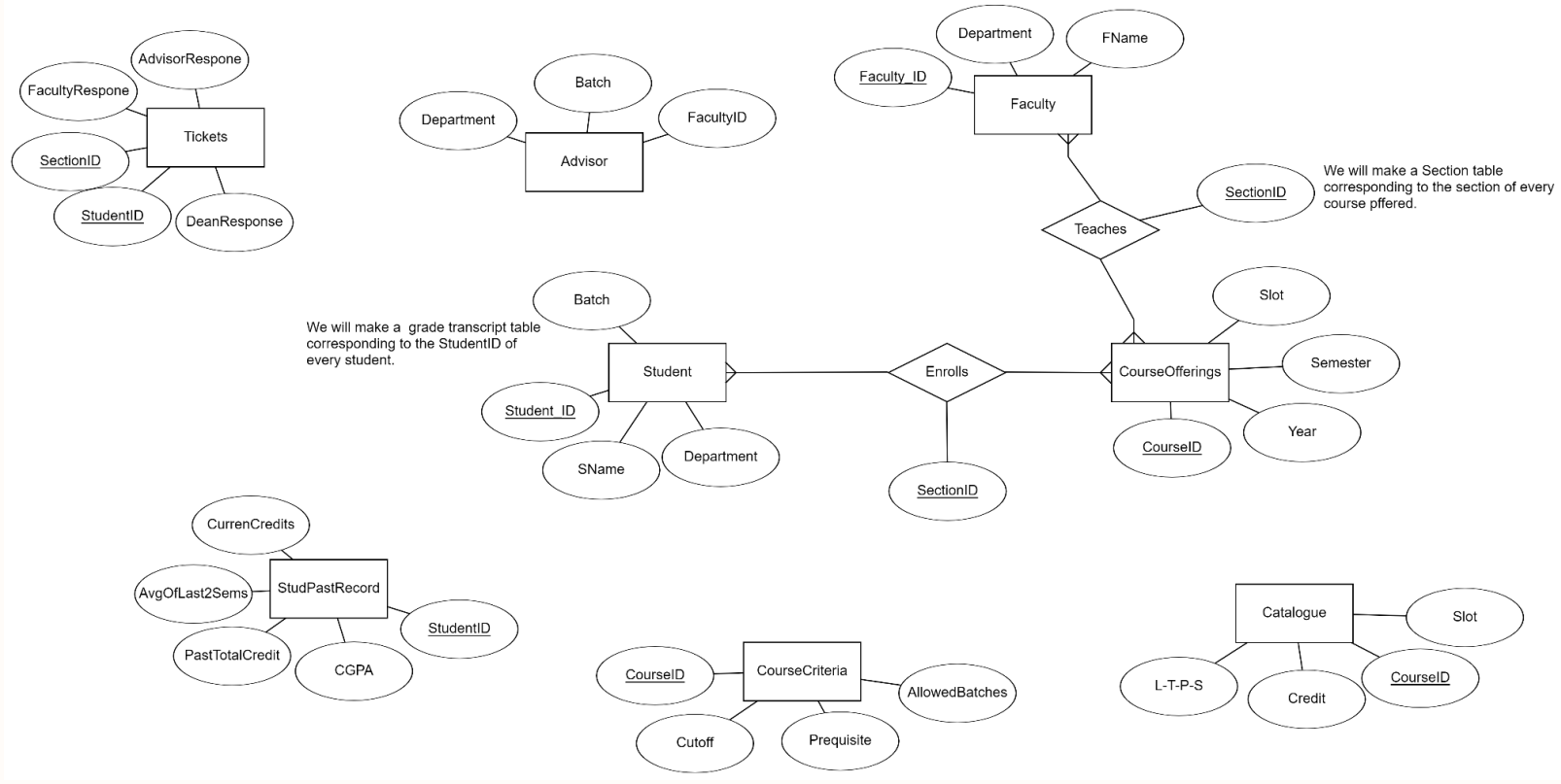
- Student
  - StudentID Varchar(20)
  - SName Varchar(20)
  - Department Varchar(30)
  - Batch Integer
- Catalogue:
  - Course ID Varchar(20)
  - Slot Varchar(20)
  - Credit Integer
  - L-T-P-S Varchar(20)
- Faculty:
  - FacultyID Varchar(20)
  - FName Varchar(20)
  - Department Varchar(30)
- CourseOfferings:
  - CourseID Varchar(20)
  - Slot Varchar(20)
  - Semester Integer
  - Year Integer
- StudPastRecord:
  - StudentID Varchar(20)
  - CGPA Float
  - PastTotalCredit Integer
  - AvgofLast2Sems Integer
  - CurrCredits Integer
- CourseCriteria:
  - CGCutOff Float
  - Prerequisite Varchar(20)
  - CourseID Varchar(20)
  - AllowedBatches Integer
- Advisor:
  - FacultyID Varchar(20)
  - Batch Integer
  - Department Varchar(30)
- Tickets:
  - StudentID Varchar(20)
  - SectionID Varchar(20)
  - FacultyResponse Varchar(5)

    - AdvisorResponse Varchar(5)
    - Dean Response Varchar(5)
- Advisor_Tickets (Dynamic table):
    - StudentID Varchar(20)
    - SectionID Varchar(20)
    - FacultyResponse Varchar(5)
    - AdvisorResponse Varchar(5)
    - Dean Response Varchar(5)
- Instructor_Tickets (Dynamic table):
    - StudentID Varchar(20)
    - SectionID Varchar(20)
    - FacultyResponse Varchar(5)
    - AdvisorResponse Varchar(5)
    - Dean Response Varchar(5)
- Section (Dynamic Table):
    - StudentID Varchar(20)
    - Grade Integer
- Transcript_StudentID (Dynamic Table):
    - SectionID Varchar(20)
    - CourseID Varchar(20)
    - Grade Integer
- Result_StudentID (Dynamic Table):
    - CGPA Float
    - SGPA Float
    - Total Credits Integer

Relations:

- Teaches:
    - SectionID Varchar(20)
    - FacultyID Varchar(20)
    - CourseID Varchar(20)
- Enrolls:
    - StudentID Varchar(20)
    - SectionID Varchar(20)
    - CourseID Varchar(20)

ER DIAGRAM -



TRIGGER IMPLEMENTATION -

Faculty_teach -

Faculty Teaches trigger is used to insert a course into the CourseOfferings table as soon as a faculty wants to offer a course and he/she inserts the CourseID, SectionID and FacultyID into the Teaches relationship table. It creates a dynamic table with the name of the table as the SectionID. As soon as a tuple gets inserted the dynamic table is created and the course gets added to the CourseOfferings table.

This trigger also generates the Instructor_Tickets table for the corresponding faculty of the current course being added into the CourseOfferings table.

Student_enroll -

This trigger is fired before any course is inserted in the enrolls table. It checks the necessary criteria which include CG cutoff, prerequisites, course slots and the credit

limit of the student. Once the student is found to be eligible to take the course, we can use the StudendID and add this student to the corresponding SectionID table which was created during the insertion of a course into the Teaches table.

If by taking the current course, the current credits of the student exceed the credit limit, then a ticket is generated and that ticket is inserted into the Instructor_Tickets table corresponding to the instructor who is teaching the current course and an Advisor_Tickets table corresponding to the department of the student enrolling in the current course.

This trigger also creates the dynamic Transcript_StudentID table for every student as soon as he/she successfully registers for a course.

Add_grade -
This trigger is used to add the grade of the particular student into his/her dynamic Transcript_StudentID table as soon as a professor adds grade to the Section table in which the current student is enrolled.

Advisor_ticket_trigger -
This trigger is used to insert the ticket into the respective Advisor_Tickets table as soon as the ticket present in the Instructor_Tickets table gets updated. As soon as the Instructor updates the value of his response in the current Instructor_Tickets table, the table gets deleted from the Instructor_Tickets table and gets inserted into the Advisor_Tickets table corresponding to the Batch advisor of the student to whom the ticket belongs.

Dean_ticket_trigger -
We use this trigger to insert the ticket into the respective main Tickets table as soon as the ticket present in the Advisor_Tickets table gets updated. As soon as the Batch Advisor updates the value of his response in the current Advisor_Tickets table, the table gets deleted from the Advisor_Tickets table and gets inserted into the main Tickets table.

Get_ticket_result -
This trigger gets us the final result of the ticket and once the current ticket in the main Tickets table gets updated, then we can delete the current ticket from the Tickets table. This will help in optimizing the space required for the ticket generation process.

STORED PROCEDURES -

Faculty_teaches, Student_enrolls, Advisor_tickets, **Dean_tickets, Ticket_result** stored procedures are executed when the triggers with similar names as mentioned above are called, and their functioning is also described above. Apart from these stored procedures, we have some other stored procedures which are useful for functions like uploading the grade, fetching transcript, etc.

Course_catalogue_add -
We make this stored procedure to add the courses to the course catalogue in the format already defined by the academic section. The Academic section has to insert the information into both the Catalogue, CourseCriteria table.

Upload_grade -
This function basically uses the csv file and adds the grades of the whole current section when the Faculty corresponding to that section uploads the grade.

Generate_report -
This stored procedure makes a report of the students by using the Transcript_StudentID table and the StudPastRecord table simultaneously to calculate the CGPA, SGPA of the student and the total credits of the student at the end of the current semester and make a new dynamic Result_StudentID table. Then we export the Result_StudentID, Transcript_StudentID tables in .csv format as the final result.

ROLES AND PERMISSIONS:

- Catalogue, CourseCriteria - Edit access to Dean Academic Office
- Transcript_StudentID-
    - View access to particular students,
    - Edit access to Dean Academic Office
- Teaches - INSERT only access to all faculties
- Enrolls - INSERT only access to all students
- SectionID tables - Edit access to faculty of the respective course
- Result_StudentID -
    - View access of the individual result table to the particular student
    - Edit access to Dean Academic Office
- Instructor_Tickets Table - Edit access only to the concerned professor
- Advisor_Tickets - Edit access to the faculty advisor of the particular batch
- Tickets- Edit access to the Dean
- CourseOfferings-
    - View access to the Dean Academic Office

    - Edit access to faculty
- Advisor- Edit access to the Dean
- Student - Edit access to the Dean Academic Office
- Faculty- Edit access to the Dean Academic Office