Academic Management System

Design Document

Made using PostgreSQL database engine.

Schema Details

1. CourseCatalogue

Course_id	Title	dept_name	L-T-P-S-C	Credit

Primary key (Course_id)
Foreign key (dept_name) references Department(dept_name)

2. Faculty

name	id	dept_name

Primary key (id)
Foreign key (dept_name) references Department(dept_name)

3. Batch_Advisor

name	id	Batch_year	dept_name

Foreign key (id,dept_name) references Faculty(id,dept_name)

4. PreRequisite

Course_id	preRequisite_course_code

Foreign key(course_id) references courseofferings (Course_id)
Foreign key(preRequisite_course_code) references courseofferings (coursecatalogue)

5. CourseOfferings

Course_id	dept_name	semester	credit	Instructor_id	LTPSC	cgConstraint

foreign key (Course_id,credit) references CourseCatalogue(Course_id,Credit)
Foreign key (dept_name) references Department(dept_name)
Foreign key (id) references Faculty(id)

6.BatchesAllowed

Course_id	yearOfAdmission	dept_name

Foreign key(course_id) references courseofferings (Course_id)
Foreign key (yearOfAdmission,dept_name) referencesStudent(yearOfAdmission,dept_name)

7. Department

dept_name

8. Student

entry_num	student_name	yearOfAdmission	dept_name	total_credit	cg

primary key(entry_num)

9. isGoingToTake

entry_num	Course_id	credit	Sec_id	year	semester

foreign key(entry_num) references student(entry_num),
foreign key(course_id,credit) references courseofferings(course_id,credit)

10. historyOfStudent

entry_num	sem	Course_id	grade	credit	department	yearOfAdmission

foreign key(entry_num) references student(entry_num)

11. studentsTicketRequest

entry_num	sem	Course_id	facultyPermission	BatchAdvisorPermission	DeanPermission

foreign key(entry_num) references student(entry_num),
foreign key(course_id) references courseofferings(course_id)

12. facultyTicketInfo

entry_num	sem	Course_id	facultyPermission	BatchAdvisorPermission	DeanPermission

foreign key(entry_num) references student(entry_num),
foreign key(course_id) references courseofferings(course_id)

13. BatchAdvisorTicketInfo

entry_num	sem	Course_id	facultyPermission	BatchAdvisorPermission	DeanPermission

foreign key(entry_num) references student(entry_num),
foreign key(course_id) references courseofferings(course_id)

14. DeanTicketInfo

entry_num	sem	Course_id	facultyPermission	BatchAdvisorPermission	DeanPermission

foreign key(entry_num) references student(entry_num),
foreign key(course_id) references courseofferings(course_id)

15.TimeSlot

Course_id	Duration	startingTime	endingTime	day

foreign key(course_id) references courseofferings(course_id)

16. courseThroughTicket

entry_num	Course_id	credit	Sec_id	year	semester

foreign key(entry_num) references student(entry_num),
foreign key(course_id) references courseofferings(course_id),
foreign key(day) references Day_table(day)

17. Day_table

day

Triggers and Stored procedures and their accesses

1) A trigger will be generated before insertion and update in CourseOffering by any Faculty which will ensure instructor_id of the course to be inserted should be same as the user and the department of the course_id of the row should be same as the course_id's department in course catalogue and user's department is also same as the course's department that is to be inserted. Also, we will check if L-T-P-S-C is the same for the course_id in CourseOffering and CourseCatalogue.

2) OfferCourse is a stored procedure we implement to insert courses(will be done by faculty) and their details in the table CourseOfferings. Faculties and dean have access to use it.

3) RegisterCourse is a stored procedure we implemented to insert course, entry number of students and some other details in the table isGoingToTake. Students have access to use it.

4) When a student is to register a course then some triggers will be called before it is confirmed that he/she registered successfully .One trigger is to check if a student is fulfilling cg criteria or not . And, one will check for pre-requisites. One other will be to check the credit limit. Prerequisites and credit limit will be checked through historyOfStudent and PreRequisite tables. One of the triggers will check if the time-slot of the course is clashing or not with other courses that the student has already taken .And one other will check if the student is from the allowed batches for taking the course.

5) We have created a stored procedure to upload a timetable slot and import it to the table TimeSlot.Dean has access to use it. And one stored procedure is to upload a grade sheet and import it to the table historyOfStudent. Faculty and dean have the access to use it.

6) One stored procedure is for generating a transcript of a student which creates tables dynamically for each student for storing transcripts and write access is with the dean and students can read only their transcript table.

7) One stored procedure is computing CGPA of the student.It is also updating cg in the Student table. The grant to use this procedure is with faculty,dean, batch advisor and that particular student.

8) We implemented stored procedures to generate ticket generation and propagation.And access to call is with students.
   This works like this:

- There are 4 tables for the use of each stakeholder.
- When a student wants to take a course out of the credit limit and course-offerings' restrictions, he/she would generate the ticket for it through a stored procedure which would take parameters of his/her entry number, course_id, sem.
- So, when a student generates a ticket through calling ticketGenerator stored procedure then a row is inserted to studentsTicketRequest table as well as facultyTicketInfo. In this way, the ticket reaches the faculty .
- After he/she updates the table with the opinion, the copy of the same row gets inserted to the BatchAdvisorTicketInfo table.
- Similarly when the batch advisor updates his/her table then the row gets inserted to the Dean.
- Now, when the dean updates his/her opinion then the course is inserted to courseThroughTicket if dean permits otherwise the exception gets raised that student is still not permitted to take the course.
- There are 3 columns for the status of permission of the faculty, batch advisor and dean respectively in each of the 4 tables. Access to the stakeholders to the column would be given accordingly.

- Students have the access to insert in the table studentsTicketRequest.
- Faculties have the access of column facultyPermission in the table facultyTicketInfo.
- Batch advisors have the access to column BatchAdvisorPermission in the table BatchAdvisorTicketInfo.

Grant And Permissions:

1. CourseCatalogue:

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	No	No	Yes

2. Faculty:

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	No	No	Yes

3. Batch_Advisor:

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	No	No	Yes

4. PreRequisite

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	No	No	Yes

5. CourseOfferings

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	Yes	No	Yes

6.BatchesAllowed

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	Yes	No	Yes

7. Department

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	No	No	Yes

8. Student

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	No	No	Yes

9. isGoingToTake

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	Yes	No	No	Yes

10. historyOfStudent

	Students	Faculty	Batch-Advisor	Dean
Read	No	Yes	Yes	Yes
Write	No	Yes	No	Yes

11. studentsTicketRequest

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	No	No	Yes
Write	Yes	No	No	Yes

12. facultyTicketInfo

	Students	Faculty	Batch-Advisor	Dean
Read	No	Yes	No	Yes
Write	No	Yes	No	Yes

13. BatchAdvisorTicketInfo

	Students	Faculty	Batch-Advisor	Dean
Read	No	No	Yes	Yes
Write	No	No	Yes	Yes

14. DeanTicketInfo

	Students	Faculty	Batch-Advisor	Dean
Read	No	No	No	Yes

Write	No	No	No	Yes

15.TimeSlot

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	Yes	No	Yes

16. courseThroughTicket

	Students	Faculty	Batch-Advisor	Dean
Read	Yes	Yes	Yes	Yes
Write	No	No	No	Yes

NOTE: login and creation of user is done manually through commands.