

CSL301 - Course Project

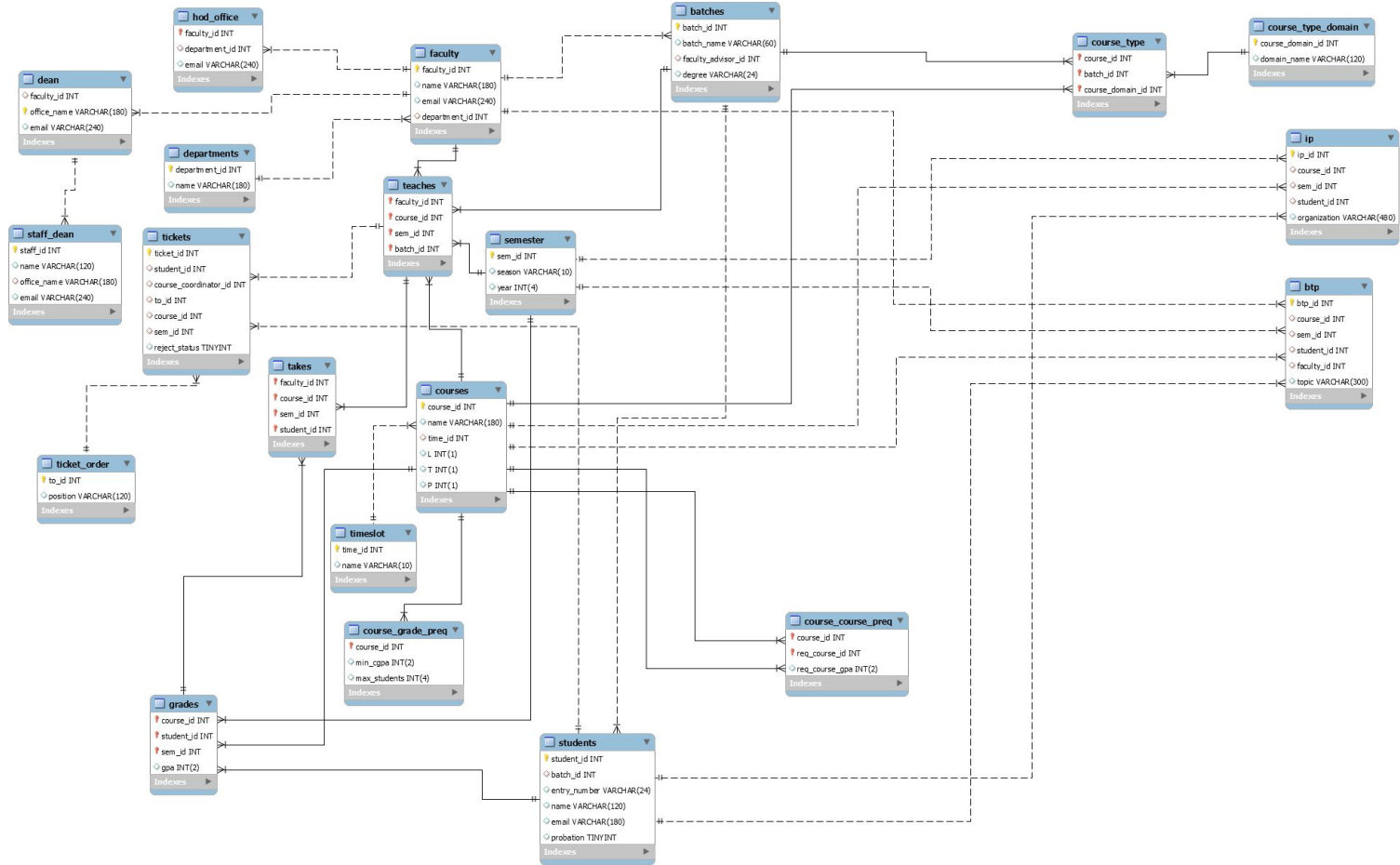
Advisor: Dr. Viswanath Gunturi

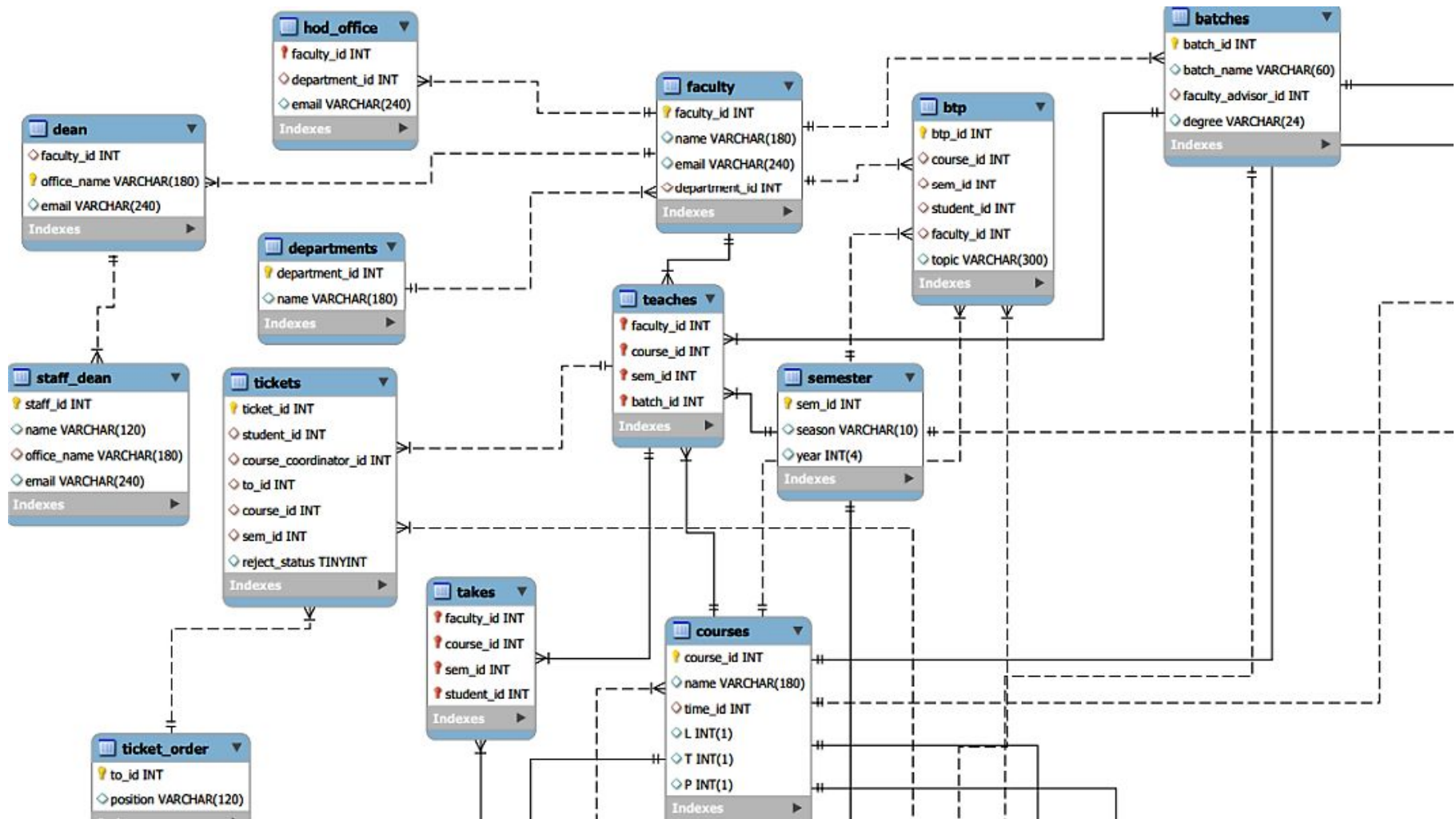
Team:

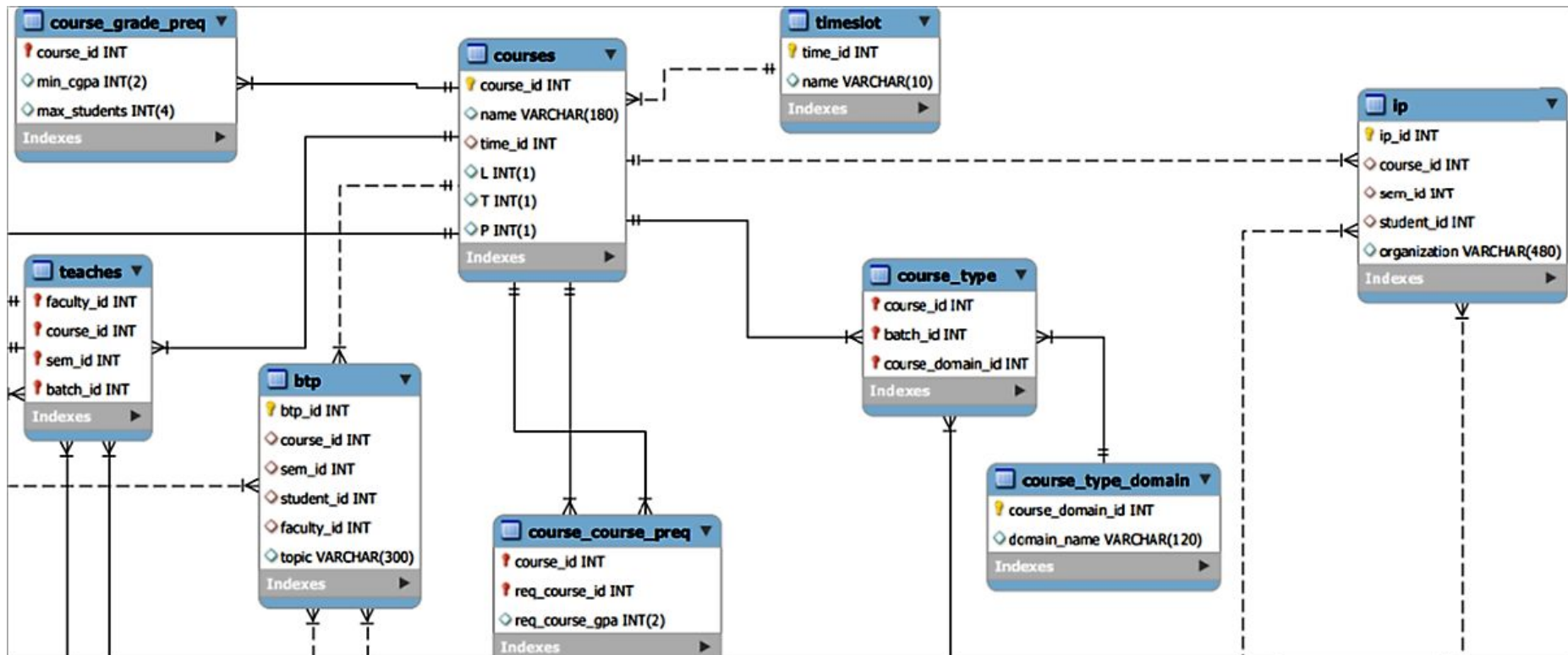
Piyush Jain	2015csb1023
Pratham Gupta	2015csb1024
Pratik Chhajer	2015csb1025
Rajat Sharma	2015csb1026

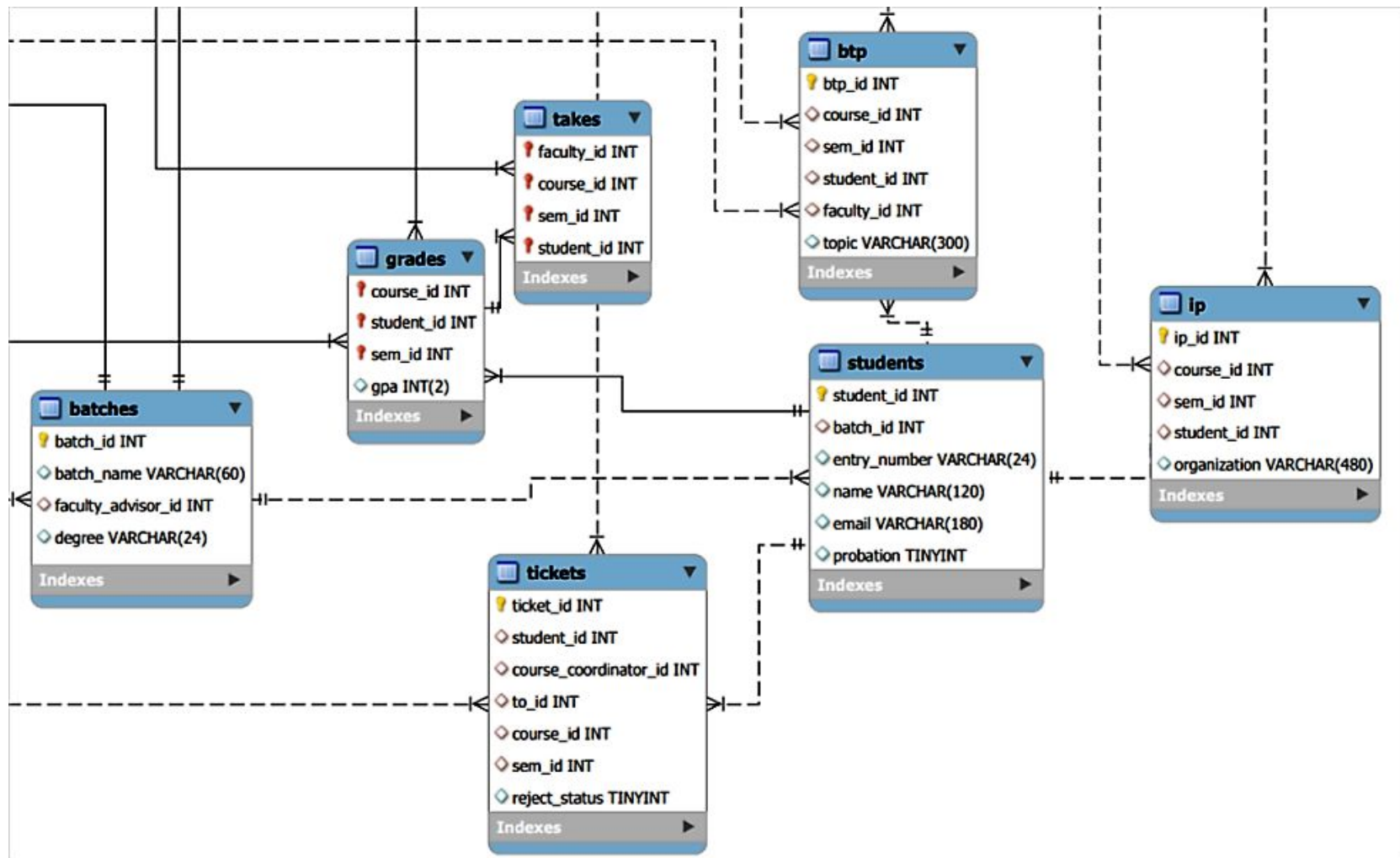
Content

- ER Diagram
- Stored Procedures
- Triggers
- Assert & Constraints
- Authorization









Stored Procedures - 1

getCredits

Arguments	course_id
Returns	credits

getSGPA

Arguments	students_id, sem_id
Returns	SGPA

getCGPA

Arguments	student_id
Returns	CGPA

Stored Procedures - 2

creditsCount

Arguments	student_id, sem_id
Returns	Credits count

creditsAllowed

Arguments	students_id, sem_id
Returns	Allowed credits count

insertTicket

Arguments	student_id, course_id, course_coord_id, sem_id
Returns	Ticket id

Stored Procedures - 3

transcriptGen

Arguments	student_id
Returns	Transcript list

coursesDone

Arguments	students_id
Returns	List of courses

gradeSheetGen

Arguments	Student_id, sem_id
Returns	List of grades, SGPA

Stored Procedures - 4

domainCoursesDone

Arguments	student_id, domain_type
Returns	List of courses, total credits

btpInfo

Arguments	students_id
Returns	Btp topic, instructor

ipInfo

Arguments	student_id
Returns	Internship Organization

Trigger - 1

```
CREATE OR REPLACE FUNCTION proc() RETURNS trigger AS $proc$
BEGIN
    IF (EXISTS(SELECT name, email, department_id FROM faculty WHERE faculty_id = OLD.faculty_id AND NOT EXISTS (SELECT * FROM
        faculty AS T1 WHERE T1.name = name AND T1.email = email AND T1.department_id = department_id AND T1.faculty_id =
        0))) THEN

        INSERT INTO faculty VALUES(0, (SELECT name, email, department_id FROM faculty WHERE faculty_id = OLD.faculty_id
            AND NOT EXISTS (SELECT * FROM faculty AS T1 WHERE T1.name = name AND T1.email = email AND T1.
            department_id = department_id AND T1.faculty_id = 0)));

    END IF;

    UPDATE hod_office SET faculty_id = 0 WHERE faculty_id = OLD.faculty_id;
    UPDATE dean SET faculty_id = 0 WHERE faculty_id = OLD.faculty_id;
    UPDATE batches SET faculty_advisor_id = 0 WHERE faculty_advisor_id = OLD.faculty_id;
    UPDATE btp SET faculty_id = 0 WHERE faculty_id = OLD.faculty_id;

    IF (EXISTS(SELECT course_id, sem_id, batch_id FROM teaches WHERE faculty_id = OLD.faculty_id AND NOT EXISTS (SELECT *
        FROM teaches AS T1 WHERE T1.course_id = course_id AND T1.sem_id = sem_id AND T1.batch_id = batch_id AND T1.
        faculty_id = 0))) THEN

        INSERT INTO teaches VALUES(0, EXISTS(SELECT course_id, sem_id, batch_id FROM teaches WHERE faculty_id = OLD.
            faculty_id AND NOT EXISTS (SELECT * FROM teaches AS T1 WHERE T1.course_id = course_id AND T1.sem_id =
            sem_id AND T1.batch_id = batch_id AND T1.faculty_id = 0)));

    END IF;
    UPDATE takes SET faculty_id = 0 WHERE faculty_id = OLD.faculty_id;

RETURN OLD;
END;
```

```
RETURN OLD;  
END;  
$proc$ LANGUAGE plpgsql;  
  
CREATE TRIGGER trig1  
BEFORE DELETE ON faculty  
FOR EACH ROW  
EXECUTE PROCEDURE proc();
```

- Deletion of an entry from 'faculty' (i.e. setting faculty_id 0)
- Foreign Key Updation Problem (Circular Problem)
- Addition of new entries with faculty_id 0 in 'faculty', then updation in other tables, then deleting the original entries from 'faculty'.

Trigger - 2

- On updating 'takes', if constraints not satisfied, add a corresponding entry in the table 'ticket'.
- Check in corresponding stored procedure of trigger and insert corresponding entry in 'ticket' if not satisfied.

Constraints - 1

```
ALTER TABLE takes
ADD CONSTRAINT different_timeslots
CHECK NOT EXISTS (
(SELECT time_id FROM courses WHERE courses.course_id = course_id)
EXCEPT
(SELECT courses.time_id FROM takes, courses WHERE takes.course_id = courses.course_id AND takes.student_id AND takes.student_id =
      student_id AND takes.sem_id = sem_id)
);
```

- Student takes a new course.
- Time Slots Collision.
- Constraints avoid Collision.
- Tables Involved: takes.
- Attributes Involved: time_id, course_id.

Constraints - 2

- Course Prerequisites
 - courses, course_course_preq
- GPA Prerequisites
 - courses, course_course_preq
- CGPA Criteria
 - courses, course_grade_preq
- Max Students Limit
 - courses, course_grade_preq
- Credits Limit
 - students, takes, courses, sem
- Different Time Slots
 - courses, time_slot

Authorization - 1

- Course Catalog
 - insert, delete, update
 - HOD, Dean, Dean-Staff
- Faculty, HOD, Dean, Dean-Staff
 - insert, delete
 - Admin
 - update
 - Faculty, HOD, Dean, Dean-Staff
- Transcript
 - select
 - Faculty, HOD, Dean, Dean-Staff

Authorization - 2

- Student
 - insert, update
 - Student
 - delete
 - Admin, Dean, Dean-Staff
- HOD, Dean
 - insert, update, delete
 - Admin
- Department
 - insert, update, delete
 - Admin

Tools / Frameworks to be used

- Front end
 - HTML
 - CSS
 - JavaScript
 - Materialize CSS
 - jQuery
- Back end
 - Apache Server
 - PostgreSQL Database
 - PHP

Thank You