

Academic works at IIITH

Intro

- Our mini-world is "The academic works at IIITH".
 - Our mini-world involves various activities involved in university, related to students, Faculty and staff.
 - Our mini-world helps in maintaining the course work and course progress.
-

User

- Professors
 - Students
 - Ta's
 - Other Administration staff
-

Purpose

- The purpose of database is to help students remember/manage their works. And force TA to correct work at earliest allowing a transparent correction. (more time to ask queries)
 - The purpose of this database is to maintain records of students, their progress and followup of works assigned to various people involved in a university.
-

Applications

- Professor can select Ta's.

- Acad Admin can use the database after completion of semester for administrative purposes.
 - Students can register for courses. Database gives information like prerequisite courses and credits.
 - Students they can track their progress and get their grade report.
 - Professor can monitor The Ta's work. No of correction done by them etc.
-

Database Requirements

Entities

- **Professor.**
 - Address (Composite)
 - Country + City + pin-code. Constraint a valid location.
 - Email (multi-valued)
 - All emails of the prof which are for public use.
 - Research Labs (multi-valued)
 - All the Research Labs to which is affiliated. Its values are from a enumerable set. (finite no values)
 - Web-page.
 - Name (composite first Name, last Name)
 - 20 char each.
- **Student**
 - Name (First + last, composite)
 - each 20 char.
 - Roll no Integer with exactly 8 digits
 - Email "Valid institute mail"
- **TA**
 - Email
 - Valid Email

- Name
 - 20 char long for each First name, Last name.

- **Assignment**

- Release Date.
 - dd/mm/yyyy. The date should be after the semester start date.
- End Date.
 - dd/mm/yyyy. The date should be before the semester end date.
- Time given (derived Attribute).
 - Number of days N. $N > 0$ and $N < \text{No of working days in a term}$.
- Assignment file. (link to PDF in cloud)

- **Project**

- Project file. (link to pdf in c)
- Start Date
 - dd/mm/yyyy. The date should be after the semester start date.
- End Date
 - dd/mm/yyyy. The date should be before the semester end date.
- no of members per team (an integer between 3 to 5 (including 3,5))

- **QUIZ**

- platform(codetrantra/moodle/meet in ms)
- Date *dd/mm/yyyy . The date should be before end of semester and after start of semester.
- Time (composite, start time and end time)
 - hr:min. (24 hour format)

- **Course**

- Name. Id
 - It can a finite number of values. Value is the same used by IIITH for a

course.

- Which semester offered.
 - It can either be Monsoon or spring.

- **Acad Admin**

- Admin name.
 - 20 char long for each First name, Last name.
- Qualification
 - Btech(ECE/CSE)/IT/Diploma/Any tech related courses.

Weak Entity Types

- Assingment
- Quiz
- Project

Relationships

- ***Student*** Registered For ***Course***

- Degree 2
- Student entity type has (1, N) participation.
- Course entity type has (0, N) participation.
- Attributes
 - Present Grade (derived) (composite)
 - Score + Done
 - Score The weighted percentage of all the activities till then. $0 \leq \text{Score} \leq \text{Done}$
 - Done The amount of Course activities percentage covered. $0 \leq \text{Done} \leq 100$
 - Attendance.
 - Integer Number of days Absent N. $N \geq 0$. $N < \text{Number of days in a}$

semester.

- **Professor** In charge of **Course**

- Degree 2
- Professor entity type has (0, N) participation.
- Course entity type has (0, 1) participation.

- **TA** For **Course**

- Degree 2
- TA entity type has (1, N) participation.
- Course entity type has (0, N) participation.
- Attributes
 - Office Hour (composite day + start time + end time)
 - day in among {Monday, Tuesday..}
 - start/end time is 24 hour format.

- **Assignment** of a **Course**

- Degree 2
- Assignment has (1, 1) participation.
- Course has (0, N) participation.

- **Quiz** of a **Course**

- Degree 2
- Quiz has (1, 1) participation.
- Course has (0, N) partial participation.

- **Project** of a **Course**

- Degree 2
- Project has (1, 1) participation.

- Course has (0, N) participation.

- **Student** Mentor **TA** in **Course**

- Degree 3
- Student has (1, 1) participation.
- TA has (1, N) participation.
- Course has (0, N) participation.

- **Student** For a **Assignment** who evaluates **TA**

- Degree 3
- Student has (1, N) participation.
- Assignment has (1, N) participation.
- TA has (1, N) participation.
- Attributes
 - Score.
 - A number X which is the percentage. $0 \leq X \leq 100$
 - Verified (yes or no).
 - It can take two values yes or no.
 - Submission.
 - Link to GitHub repo having the submission of assignment. The repo has to be private before the deadline of assignment and turn into public after.
 - Correction Remarks.
 - Link to PDF file in cloud

- **Student** For a **Quiz** who evaluates **TA**

- Degree 3
- Student has (1, N) participation.
- Quiz has (1, N) participation.
- TA has (1, N) participation.
- Attributes

- Score.
 - A number X which is the percentage. $0 \leq X \leq 100$
- Verified (yes or no).
 - It can take two values yes or no.
- Submission. (if subjective quiz)
 - Link to GitHub repo having the submission of quiz.
- Correction Remarks.
 - Link to PDF file in cloud.

- **Student** For a **Assignment** who evaluates **TA**

- Degree 3
- Student has (1, N) participation.
- Project has (1, N) participation.
- TA has (1, N) participation.
- Attributes
 - Score.
 - A number X which is the percentage. $0 \leq X \leq 100$
 - Verified (yes or no).
 - It can take two values yes or no.
 - Submission.
 - Link to GitHub repo having the submission of assignment. The repo has to be private before the deadline of assignment and turn into public after.
 - Correction Remarks.
 - Link to PDF file in cloud

- **Course** Prerequisite **Course**

- Degree 1
- Course has (0, N) participation.

- **Student** Partner of **Student** for **Project** mentored by **TA**

- Degree 4
 - Project has (1, N) participation.
 - TA has (1, N) participation.
 - Student has (1, N) participation.
-

Functionality Requirements

Retrievals

- Selection
 - Student can query for all new corrections given to them. (which they verify)
- Projection
 - Student can get details of the TA who gave him/her score.
 - Retrieval of Emails of all the students above a particular score.
- Aggregate
 - Average Score of a course at the end of semester.
 - Average score in an assignment or quiz.
- Search
 - We can search a prof/student/TA with their mail.
- Analysis
 - Students can get info like all the work they have , all the work they till a particular date.
 - How many students passed their year.
 - How many students failed their year.

Modifications

- Insert
 - Acad admin can insert a new TA/PROFESSOR/STUDENT to our database
 - professor can insert the work and deadline for it.
- Update
 - Professors can update the marks of students if there is any changes in grading(quizzes queries,or like in re-evaluation(Acad admin can do)

- Ta submit scores of students for a work.
- Acad admin can update the present courses as per semester.
- Professor can update the deadline of work.(sometimes)
- Delete
 - The Acad admin can delete any of TA/STUDENT/PROF if they wants to drop.
 - Professor can delete the work assigned for students for the request of students.
 - Acad admin delete work details of Completed courses (We maintain grades of previous year also for prof to select TA).

-
-
- Yadavalli. Anantha Lakshmi
 - P Balaramakrishna Varma
 - Janapati Sai Akash