

CSL 301 - Introduction to Database Systems
Course Project Specification
Faculty Portal at Academic University (Total Weightage: 21%)
2019 -- 2020 Semester I

General Instructions:

- **Allowed team size:** 2 members or 3 members. Projects from teams of size 3 would be more complicated for the same score and same deadline.
- **In case you want to work solo, then contact the instructor. Note that given the open ended nature of the project specification, it's not possible to have a precisely equitable distribution of work across the team sizes.**
- Students should use the given idea and develop it into a full-scale project proposal. We are just giving a just high level idea of the project and by no means a full specification.
- You are allowed C/ C++, JAVA, PHP and Python for these projects. For any other language, you should first get an approval.
- Periodic reviews are mandatory to give a proper direction (and guidance as needed) to the projects.
- Do not disappear after taking a topic and show up only in the week before the final deadline. In such cases, instructors would simply proceed with their discretion on what should have been done!

Important Notes:

- SQL tutorial: <https://www.w3schools.com/sql/>
- Refer material on database triggers, database authorization, functions and procedures.
- Web applications development in Java, Python
- Tutorial on MongoDB: <https://www.tutorialspoint.com/mongodb/index.htm>
- Transactions in Postgresql:
 - Intro to Transactions:
https://www.tutorialspoint.com/postgresql/postgresql_transactions.htm
<https://www.postgresql.org/docs/8.3/static/tutorial-transactions.html>
 - Isolation levels: <https://www.postgresql.org/docs/9.1/static/transaction-iso.html>
 - Setting isolation levels:
<https://www.postgresql.org/docs/9.1/static/sql-set-transaction.html>

Project Description:

You are expected to develop a faculty portal for an academic university. For this project, you are expected to develop code for both the front-end UI and the back-end of the system. Given that this course focuses on topics related to databases systems, we would expect a more thorough effort on the database part of the project. However, there should be a bare minimum effort on front-end UI as well.

Faculty in an academic university are largely divided into two categories: (a) Faculty and (d) cross-cutting faculty (e.g., Deans and Associate Deans etc.). People in each of these categories are formed into a hierarchy with the Director at the top most level. And as expected, people participating in this hierarchy (at various roles) change with time. Your design must allow for such changes and should also keep a track of it. Following is a brief background on the hierarchy of faculty (in an IIT):

Faculty: Faculty are divided on a departments (e.g., CS, EE, ME, Civil, etc.). Each department has a head-of-department (HoD) who is also one of the faculty members in the department. Each HoD appointment is a time bound appointment and is thus associated with a start-date and end-date.

Cross-cutting Faculty: In any institute, we do have some faculty who are not associated with any particular department. Examples of this include, Dean Faculty Affairs, Dean Academic Affairs, Dean Research and Dean Student Affairs. All Deans (and Associate Deans under them) are faculty who have been appointed to the said post for a certain duration.

Concepts relevant to faculty:

- **Personal Profile (to be implemented via a NoSQL):** Each faculty has a portal for storing his/her academic profile (for e.g., refer: <https://research.monash.edu/en/persons/abhinav-dhall>). In this portal, a faculty would like to storing details on his/her background, publications, grants, awards, teaching, etc. As you can guess, not all faculty would want to update this information. So this information should not ideally be modeled as a relational schema. Its best to use noSQL for modeling such information.
- **Leave applications:** From time-to-time, faculty can go on a leave. Depending on the post of the applicant, his/her leave application would go through a specific route. For instance, leave application of a faculty follows the following route for approval: Faculty → HoD → Dean Faculty Affairs. In each stage, the person forwards with comments. Finally Dean Faculty Affairs approves or rejects. After approval, leave is deducted from the available leaves and an intimation is sent to faculty.

Note that leave applications of HoDs and Deans are approved by the Director. Two more things to note here: (a) each employee have a fixed number for leaves per year (this expire at the end of the year). (b) Sometimes, HoD, concerned head, and/or Dean FAA may redirect the application to the employee for more comments. Note that even if the leaves for the current year are finished, the employees may still be granted leave by borrowing some from the coming year.

In such a case, two requests are raised by the faculty. One request is for borrowing of leaves and other is regular leave. Both of these requests are encapsulated as one request and go together and follow the same route described previously. Once approved, an appropriate note (mentioning that leaves have been borrowed) is attached with the approval.

Assume the following Depts in Faculty: CSE, ME and EE. Each Dept has an HoD.

Assume the following Deans: Dean Faculty Affairs, Associate Dean Faculty Affairs

Assume one Director: And everybody comes under the Director.

Portals to be implemented:

- Basic Employee Portals: Each of the employees would have their own personal portals. Portals should have the following: (a) Personal academic profile (and options to edit it), (b) Total number leaves available this year, (c) Status of the leave applications (including the comments added by various entities, (d) Options to start new leave applications, (e) Respond to comments made on leave applications.
- Specialized Portals: Each of the named positions such as HoD, Dean and Director would have specialized portals for handling the applications. Note that all the specialized portal logins must be tied up with an employee (implicitly).

Constraints:

- **Complete Paper Trail Needs to be Maintained in the system:** “Who signed what and when.” Even if an employee leaves the institute, there should be a record on what all did he/she approve. Similar is the case when HoDs or Deans change. Note that all the specialized portal login must be tied up with an employee. For instance, if a faculty signs an application via his/her Deans login, then appropriate information regarding this must be stored in the database.
- **Route of the leave applications should not be hard coded into your code.** These things change with time. You can assume the presence of your DB-ADMIN who can change these routes as needed in the database without the need for re-compiling the code.
- An employee can launch only one leave application at a time.
- Your design should have relevant security features. For instance, a faculty should not have write access to the field/table containing Dean’s comments (or HoD comments) on leave applications, he/she can only read it.
- Make sure you commit the transactions!!

Extra Work for Team size-3 (team size-2 and size-1 can ignore this portion)

Extend the previously developed solution to include the concepts of the sponsored projects. From time-to-time, faculty write proposal for sponsored projects from external funding agencies such as DST, DBT, etc. Typically, these projects have budget to hire project associates, travel (for conferences) and equipment (s). Note that, not all projects would have budget for each of these components. For instance, exchange-visit grants typically do not have any budget for buying equipment.

Following concepts need to be addressed while designing this component of the project:

1. A project can have multiple PIs. Out of these, one is designated as the main PI and others as Co-PIs. Each of them would be a faculty in the institute.
2. Each project would have several budget heads, e.g., manpower, equipment and travel.
3. A PI can hire a project associate if money is available in the manpower budget. A manpower has a fixed salary e.g., 31,000 per month (for JRF) or 40,000 (for SRF). These salaries are fixed by the funding agency.
4. Before a project associate hiring can be done, PIs need to get an approval from the institute. Approval is also similar to leave application. A PI (Co-PI or main PI) puts in the application. This application (containing the number of months for which associate would be hired) goes to main PI (if applicable). Following this, the application is forwarded to Dean Sponsored Projects for approval. Post approval, the application is sent back to faculty.
5. PIs need to follow a similar route of approval for using travelling allowance and equipment.
6. **Note that the projects should be tied up with the personal portal of the faculty. In other words, you should not make another portal for PIs.**
7. Complete expenditure report of all the projects must be stored (and also visible to PIs)
8. Complete paper trail on “who approved what (and when)”, “who bought what must be maintained.” This must be stored even if a faculty leaves the institute.

Key Deliverables:

1. ER diagram, schema of the tables, plans on triggers and stored procedures. Your diagram should have **least possible redundancy of data and attributes must be stored in appropriate entities. Appropriate cardinality constraints (E.g., many-to-one, one-to-one, etc) must be clearly shown. Improper creation of entities, relationships, attributes, or not using ER diagram symbols would lead to significant loss of points.**
2. Final Demo after implementing all the required portals.