**IIIT Guwahati**

CS 241 DBMS Lab

Winter 2017

Design and implement a database. Some topics are suggested below. If you have any other ideas, please suggest. You will have to get approval for the topic you choose. The lab will be on individual basis. There will evaluations at the end of each phase as per the set time table. There will be a viva for the complete system at the end.

Phase I - Table Design – In this phase you will design the tables in your application. You will look at alternatives and justify your choices. You will implement your tables in Mariadb. You may have to revisit this phase while doing phase III. 19/1/2017

Phase II – in this phase you will design the screens for data entry; You will also design formats for bulk loading of data (say CSV format data / excel format data to be uploaded at once). 2/2/2017

Phase III – in this phase you will design the screens for queries to your database. You will also write all the SQL queries that are required. 9/2/2017

Phase IV – in this phase you will implement the designs of phases II and III in html and PHP. 16/2/2017

Phase V – you will test your implementation thoroughly. You will design test cases and justify why the test cases were chosen. You will run the test cases and encure the implementation is correct. 2/3/2017

Phase VI – you will tune your database by inserting indices as you may feel necessary. You will run experiments to show differences in performance with and without indexing. You will use some of the test cases to see the performance differences. 9/3/2017

Phase VII – You will add concurrency control features to allow concurrent updates to your DBMS. You will again use test cases to see the effect on performance. 16/3/2017

Assignments on Advanced features of Mariadb and other DBMS systems – the rest of the lab sessions.

1. **Social Network database:** The goal is to create a social network database, like Twitter, where you will store the network (say who-follows-whom) and the associated data of each person like tweets, tags, photos and more. Users should be able to view a chronological listing of anyone's tweets, find tweets by certain people, group tweets by topics, find friends who tweet about some topic, find friends of friends who are not friends (to close social triangles!) so on.
2. **Bibliography database:** Develop a system that will improve a research group's ability to track its publications and publications of interest to the group. Track information such as papers, authors, projects, conferences and journals. Readers should be able to view chronological listings, find papers by certain authors, group by projects, recover lists of papers based on keywords, etc. It should be easy for group members to add new papers, both written by the group and published by others in the literature. Examples of such systems include [Connotea](http://www.connotea.org) and [CiteULike](http://www.citeulike.org).
3. [**The Green500 list**](http://www.green500.org)**:** This list of the most energy-efficient supercomputers was created and is maintained by professors in the Department of Computer Science at Virginia Tech. This project involves creating and maintaining the database that drives the website. Dowload the data at https://www.top500.org/files/green500/green500\_top\_201611.xlsx, create a schema for this data, storing the data using this schema, and enabling queries to be made, as in these [examples](http://www.top500.org/sublist).
4. **Nobel Awards Database:** The goal is to model and populate information about the awards made in the various fields (Physics, Chemistry, Physiology or Medicine Literature, Peace and the Economic Sciences), the recipients, their countries, their year of birth etc. Your system should be able to answer questions such as "When was the first time an Asian won an award for the economic sciences?" (the answer to this particular question is 1998). The Nobel Foundation maintains such an [interface.](http://www.nobel.se/index.html) You could also work on variants of this idea such as the recipients of the [ACM awards](http://www.acm.org/awards/) (unfortunately, there is not too much information online about this). Interesting queries then could be "Name people who have won at least two different awards" (the answer would include Knuth, Thompson, Ritchie, Engelbart etc.) Or the people "who were ACM Fellows before becoming Turing Award Winners" and so on.
5. **Books Database:** This domain is another popular one. Just look at [barnesandnoble.com](http://www.barnesandnoble.com) or [amazon.com](http://www.amazon.com) for excellent examples. You could model entities such as books, their authors, topics (which may be a complex hierarchy). You may also model various attributes of the authors and the institutions they belong to. You can support a service for buying and selling used books or books used in specific university courses. Your system can build a personal profile of people (and the books they like) and your database application could form the basis for a "recommender system", such as those supported by the commercial sites. The goal here is to "cluster" similar preferences together and the system can then make recommendations: "Since you liked *Harry Potter and the Sorcerer's Stone*, I recommend that you try *Harry Potter and the Chamber of Secrets*".
6. **Movies Database:** There are several excellent movie resources on the web, such as the [hollywood.com movies site](http://regal.hollywood.com) or the [Internet Movie Database](http://www.imdb.com). You could model entities such as movies, their actors, directors, genres, playing times, and reviews. There are several sources on the web from which you could get data to populate such a database. You can support various queries such as finding specific playing times, finding movies playing in Blacksburg directed by a given director. You can also support updates to the reviews section of the database (e.g., viewers giving their own opinions). Another functionality is to provide personal profiles of people (i.e., the movies they like) and then try to recommend movies to them based on profiles of viewers with similar tastes. You could also create a database of OSCAR or Golden Globe nominations and awards and answer queries such as "Find all the sitcoms that have been nominated three years in a row".
7. **Personal Photos database:** With the advent of cheap digital cameras, everybody has piles of digital photos. People need a way to organize, access, and show off their photos.
8. **Apartment Homes:** Our friendly neighborhood web guide is [here](http://www.apt-guide.com/). This domain would require modeling apartments and their attributes, areas of town and their various characteristics (e.g., BT bus lines, crime rates, distance from various landmarks). You would provide an interface for offering apartments for rent, finding apartments based on various requirements ("gas heating + pets allowed + rent less than $500 + close to campus + BEV modem facility").
9. **Census Database:** Can you make a census data dissemination system for the [Census Bureau](http://www.census.gov/)? A census gathers data about people, business, [geographic regions](http://quickfacts.census.gov/qfd/), etc. Different types of users need to gain different types of answers from the data. Homeowners want to know statistics about their region, such as crime rates. Business owners want to find holes in the competition. Government decision makers want to learn about demographic trends, and where to focus resources.
10. **Web Sites:** How do you think web search engines such as [Google](http://www.google.com) model their domain? You could think of them as a glorified database system where the basic entities modeled are web sites. You could then model the various properties of a web site: Topic, URL, domain name, other sites it links to, the background colour, etc. Retrieval could be for sites that have similar characteristics and properties.
11. Student Registration Record and Grades Record
12. Student Information (personal, hostel, cultural activities, games, clubs, prizes, projects, papers)
13. Employee Information – posts; promotion record; leave record; loans;
14. Pharmacy database (users; doctors; medicines – description including expiry data; stock of medicines; issue record; procurement record)
15. Hotel Booking System (bookings, cancellations, rates (varying), availability, payement)
16. A simple banking system handling different kinds of accounts – current, savings, fixed deposits, recurring. Account creation, withdrawals, deposits; transfers
17. Cricket Statistics based system
18. Football Results (scores, scorers, time of scores, yellow and red cards, substitutions)
19. Music DBMS
20. Election Results
21. R&D Accounting System – vouchers : payment, receipt; budgets; tax deductions.
22. Online Examination System
23. Training and Placement Management
24. Airline Reservation System (simplified)
25. Online Student Feedback
26. M.Tech Online Applications
27. Payroll System
28. Library management system database project
29. Store Management System
30. Sports League Management System.