CSCI3343 Database Systems (Fall 2025)

Application Flavor Project

Topic:

This is a semester-long course project of building a database-driven, Web-based (optional) application for a real-world domain of your interest. The project will be done in several stages: First, you will identify an application of your interest that requires database systems as backbends. Then, you will design your underlying database and key functionalities you may provide with the database. You will create an actual database using the database of your choice (e.g., PostgreSQL, MySQL, MongoDB) and populate the database. After that, you will write queries and modifications on the database to support your application logic. As a bonus feature, you can build a Web-based interface to support readily interaction with your real-world application. You will demo your system and possibly showcase it to the whole class.

Those who do not have a project can do the default project as described at the end of this document.

This programming project takes up 40% for your final score. If you can implement a project with Web interface and access, there will be extra points (25%). The programming project is a multi-stage project and for each stage, you have deliverables to submit. You cannot proceed to the next step until you get an approval (with comments and suggestions) from the instructor.

- **Stage 1**: Group formation and Team Name (5%)
- **Stage 2**: A detailed description of your real-world application and the Entity-Relationship diagram for SQL database or data modeling graph for NoSQL database (20%)
- **Stage 3**: Development plan (20%)
- **Stage 4**: Demo and final report (55%)

Part I: Group Formation and Prospectus

You must form a group of up to three students.

Create a cool **name for your group** and include **student names**, **emails** information.

- A paragraph describing the project.
- The project cannot be trivial in that you must have at least 5 tables/data models with at least a one-to- many relationship among some of the tables/data models.

Part 2: Application description and ER diagram

Your group identifies an application you would like to manage with your database. Please pick an application of your interest, as you'll be working with it for the entire semester. Your application should be substantial, but not too complicated.

You need to submit a project summary including:

- Description of the application domain of your choice. State as clearly as possible what you want to do, and pay attention to the following factors: (1) Usefulness. State as clearly as possible why your chosen application is useful. Make sure to answer the following questions: Are there any similar or equivalent web sites/applications out here? If so, what are they and how are yours different? (2) Realness. Describe what your data is and where you will get them.
- Description of the functionality that you plan to offer. This is where you talk about how to meet the functionality requirements. There are two types of functions you need to offer: (1) Basic Functions: see Stage 4 for what basic functions you need to offer. (2) Advanced Functions: Remember to include at least one creative thing (advanced functions), i.e., something that doesn't exist in equivalent web sites/applications. The advanced functions should go beyond the basic functions listed above; in addition, an advanced function should also be something that is technically challenging, meaning you would need to spend some significant time (at least a few days' work) to implement it. Of course, such functions should be relevant and therefore useful for your application. Make clear what the features are, and explain why you think it is cool to have them.

You should submit the specification of your application (functions, each table/collection, row/document, column/attribute, primary key), plus descriptions on the assumptions you make.

Part 3: Development Plan

Each group must submit a detailed development plan including the following:

- The relational schema of your database. Remember to include all keys and dependencies (e.g., functional dependencies) as appropriate.
- The final choice of databases and platforms/framework that you will be using.
- Where and how you will get data for your application. Do you get it from the Web, some other application, or do you make it up?
- Describe the labor division among group members.
- A project timeline with milestones.

Part 4: Demo and Final Report

Each group must demo its application. The demonstration takes 15 minutes, and will contain all of the following:

- **Basic Functions**: (1) show how to insert records to the database; (2) show how to search the database and list or print returned results; (3) show a few different interesting queries over your database. One of the queries must involve joining multiple tables; (4) show how to update records; (5) show how to delete records.
- **Advanced Functions**: the advanced functions should go beyond the basic functions listed above; in addition, an advanced function should also be something that is technically challenging, meaning you would need to spend some significant time (at least a few days' work) to implement it. Of course, such functions should be relevant and therefore useful for your application.

Clarifications on advanced functions

An advanced function should be: (1) useful for your users; (2) technically more challenging than the basic functions; (3) creative or novel with respect to other similar applications in your domain. Examples: It's an advanced function because of novelty: as an example, there was a project that archived all the weather forecasts from several major weather sites. One of its advanced functions predicted which weather site is the most accurate, given the current weather state. It's an advanced function because of implementation. As another example, there was a project that does recipe search. One of its advanced functions is to recommend recipes based on the user's dietary constraints. Users enter their constraints, such as calorie intake, ingredient preferences, etc., and the function returns a list of recipes that meet the constraints. The novelty comes from their search implementation. The search is an exhaustive search over 1000+ recipes, involving multiple table joins and pair comparisons, and they optimized this search with indexes and views.

Essentially, advanced functions are opportunities for your group to be creative and fun. At demo time, your group needs to make a case why your functions are advanced.

Report requirement:

As a complete project report, you need to include all the detailed contents of your project and write it as a complete story. The report is expected to be at least 8 pages in length (including references). You need to address the following key components in your report:

- **Introduction**: What is your application? Why do you want to choose this application, instead of others, as the course project (motivation)? What are the key components in your project?
- **Database details**: How do you design the database? What is your E-R model and relational model? What are the tables you include in the database?
- **Implementation details**: What languages and platform you've chosen to do the implementation? How do you implement the front-end Web interface and the application logic? How does the front-end Web interface interact with the backend database?

- **Experiences**: What have you learnt from this project? How have you solved hard problems in this project? How to extend your project to more advanced, mature systems in the future?
- **References**: Any resource you have turned to for help during the implementation of your project.

Default Topic:

Groups or individuals that do not have a project by the established deadline will be required to do the default topic described below:

Description:

Groups or individuals will develop a Department Course Scheduling system. The requirements are described below:

- Course information will include
  - Course prefix (CSCI, COMM, ART, etc.) which will not be confined to a specific department
  - Course Number (four digits)
  - Course Description (Text field)
  - Whether the course is a small lecture, large lecture with recitations, large lecture without recitations
- Classroom information will include
  - Building
  - Room
  - Capacity (number of seats)
  - Whether the classroom is technology enhanced or not
- Faculty member
  - Title (Dr. Prof, Mr., Mrs, Ms, etc.)
  - First Name
  - Last Name
  - Building
  - Room
  - Email
  - phone
- Schedule information
  - Course id
  - Section
  - Whether it is a lab or lecture
  - Classroom assigned
  - Day(s) class meetings
  - Start time
  - Ending Time
  - Assigned course instructor
- Should allow the user to assign rooms/times to a class or recitation
- Should not allow recitations to conflict with associated lecture
- Should allow the user to print out the schedule
- Should allow the user to sort the schedule based upon faculty, course type and number, assigned room, times.
- Should show classes that conflict
- Should report any errors in the entered schedule

Research Implementation Flavor Project

Topic:

There is a semester-long course project that is meant to be a substantial independent research or engineering effort related to the real-world data management issues. Students will select one (or several) published, **full** research paper from the leading database or data processing conferences (the short list is mentioned below), e.g., SIGMOD, VLDB, ICDE, published on or after 2010, implement the core algorithms and systems specified in the paper, and carry out some experimental studies performed in the paper. The list of potential conferences are listed as follows:

- SIGMOD ACM Conference on Management of Data
- SIGKDD ACM Knowledge Discovery and Data Mining
- ICDE IEEE International Conference on Data Engineering
- SIGIR International Conference on Research on Development in Information Retrieval
- VLDB International Conference on Very Large Databases
- CIKM ACM International Conference on Information and Knowledge Management
- WSDM ACM International Conference on Web Search and Data Mining
- PODS ACM Symposium on Principles of Database Systems
- DASFAA Database Systems for Advanced Applications
- ECML-PKDD European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases
- ISWC IEEE International Semantic Web Conference
- ICDM International Conference on Data Mining
- ICDT International Conference on Database Theory
- EDBT International Conference on Extending DB Technology
- CIDR International Conference on Innovative Data Systems Research
- SDM SIAM International Conference on Data Mining

Students can form teams of at most three people. Students are welcome to discuss their problems, ideas, and potential solutions with the TA, the instructor, and even other faculty members throughout the semester.

Milestones

- Group formation (0%): find a project partner in the class, if needed, and begin to discuss project problems and ideas.

- Project proposal (10%): your proposal is one or two pages long and should explicitly state the following: 1. Your project content indicates the paper you want to implement, including the paper title, conference name, author name, and the year the paper was published); 2. The problem this paper proposes to address; 3. The (rough) methodology and plan for your project. Be sure to structure your plan into a set of incremental, implementable milestones and include a schedule for meeting them; 4. The resource needed to carry out your project; 5. The workload distribution if more than one member is involved in the group.

- Literature survey (20%): You should determine the exact paper/idea you want to implement at this stage. Your survey is between two or three pages long and should place a particular focus on the technical discussions about HOW existing algorithms, methods, and solutions differ from the work you implement and why it is effective for solving the problem, compared with others. The survey should include comparative justification for the pros and cons of different work with technical details. Through this survey, you should be able to convince others that you are implementing something fundamentally new, either a brand new problem or a novel approach to a known problem, and you have known the existing state of the art for this problem.

- Status report (10%): Your status report is one or two pages long and should contain enough implementation, data, and analysis to show that your project is on the right track. You should revise your original proposal to accommodate the instructor's comments, along with any surprising results or changes in the direction, schedule, etc. You sometimes also need to have a refined version of the problem statement.

  Basically, the following items are expected in your report: 1. A very clear and specific problem you want to solve (you've finalized the problem statement so far); 2. Basic goal of the project (what do you want to achieve at the end of the semester); 3. Your assumptions and methods and how they differ from others (in brief) 4. Your software/tools/data sets used in the project; 5. The detailed plan of experimental studies you want to perform (in accordance with the experimental studies mentioned in the paper); 6. Your current status and partial results; 7. Your brief plan for the remaining time.

- Final report and software/source code (60%): the final report should extend your previous write-ups into a conference-style paper with up to ten pages. The report should: 1. present the research problem and summarize your contributions in the first section; 2. survey related work in the related work section; 3. include a detailed description of your algorithms, analysis, and implementation in the technical section; 4. describe evaluation methodology and significant results in the evaluation section; 5. finally present your conclusions (in the summary section); 6. For team work, the report should also include a paragraph explaining, for each group member, their contributions and duties in the project. 7. Please specify a hyperlink through which we can download your source code, software, and data set for reproducing your experimental results.

Example Papers

- Yang, Zongheng, Badrish Chandramouli, Chi Wang, Johannes Gehrke, Yinan Li, Umar Farooq Minhas, Per-Åke Larson, Donald Kossmann, and Rajeev Acharya. "Qd-tree: Learning data layouts for big data analytics." In Proceedings of the 2020 ACM SIGMOD international conference on management of data, pp. 193-208. 2020. https://dl.acm.org/doi/pdf/10.1145/3318464.3389770

- Ren, Huimin, Menghai Pan, Yanhua Li, Xun Zhou, and Jun Luo. "St-siamesenet: Spatio-temporal siamese networks for human mobility signature identification." In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 1306-1315. 2020. https://dl.acm.org/doi/pdf/10.1145/3394486.3403183