



Project Specification

1 Expectations

This project is intended to be a *complex* database application. As a guide, the database component of the project should include at least 10 tables and at least 5 non-trivial queries used for reporting (explained below). Furthermore, you must implement a user interface in a programming language of your choosing.

2 Teams

Students will work in project teams, typically with 2-3 members. A student can work individually if given approval by the instructor. Larger teams are expected to produce more impressive final products.

3 Deliverables

The team must produce ...

1. a **proposal**, describing the project work to be completed;
2. a **presentation**, to be delivered to the class at the end of the semester;
3. and a final **packet**, including all digital artifacts

Additionally, each team member will individually submit a **peer evaluation**.

During the semester there are four milestones (see the Syllabus) when components of the final packet are due. These deadlines are intended to facilitate project progress, as well as serve as structured opportunities for feedback. It is expected that your final work may differ from what was submitted at these checkpoints; however, these submissions will be graded for quality and completeness.

3.1 Proposal

The proposal lays out the work that the team will do for the class project. This document is to be **at most 2 pages** and must include at least the following information:

- A. The motivation for the project and the problem to be solved
- B. The main sets of information (“entities”) that you foresee being stored in the database, as well as what information about them and an estimate of how many resulting tables (should be at least 10, but not significantly more)
- C. A set of at least 5 *interesting* report queries you foresee asking of your database (see guidance below on levels of complexity) – if this component is difficult, you may consider amending what will be stored, or choose a different project

- D. The digital artifact(s) that will be produced (e.g. program(s)/websites), how you foresee them functioning, and implementation language/tool(s)
- E. The DBMS to be used
- F. How you plan to divide up the work in your team

The proposal should be turned in by only one member of the team as a single PDF document. The team will receive prompt feedback; if the project is inappropriately complex, the team will need to meet with the instructor to discuss the project and submit an amended proposal.

3.2 Presentation

The team will demonstrate their project in the last weeks of class. Students are expected to attend all presentations; not attending or coming to class late will result in a reduced individual project grade. All members of the group are required to be present during, and contribute to, their group presentation; if a group member is not present or does not contribute, s/he will receive a reduced individual project grade.

3.2.1 Format

The presentation will be made using PowerPoint (or equivalent) and must cover at least the following aspects of the project:

- a) Project Motivation: what problem are you addressing?
- b) System Description: what does the system do? how does this solve the problem?
- c) System Architecture: what dbms, front-end language(s), and any other component(s)? why? how does everything talk?
- d) Database Design: ERD, relational model, source of the data for each table, physical design
- e) Reports: description and demonstration of each query

You should provide a live demonstration of the running system, including running parameterized queries via an application interface.

Your presentation, including demo, should last 12 minutes. You will then have 3 minutes for a question/answer session with the other members of the class. The audience is expected to act as a panel of database experts: they should try to understand the database design, question aspects that don't seem right, and contribute constructively. Audience members that ask great questions will receive an increased individual project grade.

You must include a PDF version of your slides with your final presentation packet. Additionally, one member of the team must submit a PDF version of your slides *by 8am, the morning of the presentation*.

3.3 Packet

The packet should be turned in by one member of the team and must contain at least the following items:

- A cover sheet with the project name, team members, class/semester, and date
- Abstract: a 1-page overview of the problem being solved and the solution delivered

- A detailed textual description of the problem, including what entities are involved, how they relate/constrain each other, and what information needs to be extracted/presented – if this is reasonable and precise, this description will be the basis by which the database design is evaluated
- ER diagrams: 1 global, as many logically separated local views as is appropriate
- Normalized relations with primary/secondary/foreign keys clearly identified; any deviations from the ERD and/or 3NF must be justified in text (including denormalization)
- Physical design: including justification of any [not] included indexes
- Sample screenshots of your running system, with text captions indicating why they were selected, and what is being presented
- Sample reports: for each query, an example output, with accompanying text description of what was the parameterized input, what is being output, and why this report helps solve the core problem addressed by the project
- Source code & structure/data dumps (i.e. DDL & DML)
- A user manual, indicating all of the program names, what they do, and how they are compiled/configured/used; this must be sufficient to get the system up and running in a VM
- A section discussing how the project unfolded, including
 - what you liked and disliked the most
 - what you found the easiest and the hardest
 - what you learned from doing a large-scale project
- A conclusion statement: what was produced, what still needs to be done

Submit the packet as a single archive. By default, packet components should be in PDF format, including text and diagrams. You must include source code and build scripts to reproduce the final system.

The packet is due the last day of class, and late work will not be accepted.

3.4 Peer Evaluation

Group projects are sometimes looked upon as being “unfair.” To combat contribution inequity, each team member’s perception of the quantity of work that s/he performed and that of each team member will be analyzed against the perceptions of the team member(s). Through this process, hopefully equity will be achieved that reflects each member’s contribution to the group effort.

Each team member will submit a report rating the relative contributions of each team member (including her/himself) using a single number, as well as optional commentary. The aggregate rating for each student will determine the grade that individual receives, relative to the group grade. In order for this process to work effectively there is the need for each group member to be honest and objective; these ratings and comments will be kept confidential.

4 Grading

The group project grade is based upon ...

Group presentation (20%) Includes aspects of professionalism (how the group presented itself, was the presentation rehearsed and did it fit well into the time allowed, how were questions fielded) and the degree to which the required sections were covered (see above).

Database design (20%) Includes the ERD and resulting relational schema (was the design correct semantically, as well as technically both in diagrams and code; were tables normalized to at least 3NF; was a reasonable physical design applied and justified, relative to the highlighted queries).

Project complexity (20%) Includes number of tables and complexity of the report queries (see below).

Product and documentation quality (20%) Includes the sample data (quantity, quality, strategy and implementation for gathering/producing the data), the manual (did it have all required content, was the writing grammatically correct and professionally presented), the source code (was it commented; were user inputs sanitized; were good security measures taken, such as salting passwords), and the ability to install, run, and use the system. Note that a “pretty” and usable interface is encouraged, and may be given bonus points for outstanding work, but is not required (i.e. if command line tools yield a fully functional system, all associated points will be awarded).

Intermediate Milestones (20%) Each of the four (4) submitted components will count for 5% based upon the criteria in this section. While it is understood that the work will change by final submission, final submission quality is expected.

As described above, each team member will submit a peer evaluation report. These evaluations are a serious statement and are used to re-distribute up to 50% of the grade on the project: if all group members agree they put in equal share, each individual grade will be equivalent to the group grade, but those who did less will receive up to 50% lower individual grade, and those who did more will receive up to 50% greater. No late submissions will be accepted for this evaluation. If you do not submit an evaluation it will be assumed that you did not perform your fair share of the work and your grade will suffer as a result.

4.1 Report Query Complexity

You will likely need many queries for your system to run (e.g. creating tables, populating tables, and simple queries to add/update/remove entities). However, you must come up with at least 5 queries that you highlight as “reports,” that is they extract useful information out of your database and present it in a useful fashion.

To help you gauge the complexity of these report queries, please reference the following approximate scoring algorithm. Note: actual grading of queries will be subjective, and so the following guidelines do not guarantee that your queries will be considered “complex” (especially if you try to game the system).

4.1.1 Score Your Query

Start with a score of 0, add points for each applicable question:

- # Tables joined (1:0 points, 2:1 point, >2:2 points)
- Non-inner/natural join? (no:0 points, yes:1 point)
- # of subqueries (n/a:0 points, 1:1 point, >2:2 points)
- # queries comprising result via union/intersect (0:0 points, ≥ 1 :1 point)
- Aggregate function(s)? (no:0 points, yes:1 point)
- Grouping? (no:0 points, yes:1 point)
- # ordering fields (≤ 1 :0 points, > 1 :1 point)
- # WHERE/HAVING conditions not for joins (≤ 1 :0 points, > 1 :1 point)
- Non-aggregation functions or expressions in SELECT/WHERE? (no:0 points, yes:1 point)
- Strong motivation/justification for the query in the domain? (no:0 points, yes:1 point)

4.1.2 Score Meaning

Your score falls into one of the following buckets:

Trivial 0 - 1 points

Easy 2 - 3 points

Moderate 4 - 5 points

Complex >5 points

You need at least 5 complex queries, as well as 10 tables, to be considered a complex project. Queries that are insufficiently complex will receive partial credit. Only the 5 most complex queries will be scored (i.e. many trivial/easy/moderate queries will not add up to fewer complex queries).