

CS 564 Minirel Project: Overview

Introduction

These project stages will enable you to learn how to build a database system. You will build a working single-user DBMS that can execute certain simple SQL queries. The objective is to learn how a DBMS is organized and what goes on inside it when queries are executed. We will help you out by supplying the topmost and lowermost DBMS layers. The topmost layer is a parser that parses SQL queries and calls appropriate functions in the lower layers to perform relational operations. The lowermost layer is the disk I/O layer which reads and writes pages from and to the disk. In your case, the disk will just be the UNIX file system.

At the end of each stage we will take a checkpoint and evaluate your work. In Stage 3 you will implement a *buffer manager*. Stage 4 involves implementing *heap files* - a collection of pages that is used to hold a relation. Stage 5 implements the database *catalogs* and various *utilities*. Finally, in Stage 6 you will implement a number of *relational operators*. At the end of it all, you will have implemented a miniature DBMS. It will not do everything that a commercial DBMS does, but it should be able to handle a fairly useful set of queries. While doing these stages you shall learn how to tackle large programming projects and in the process have fun as well! Good luck!

Note: Since time is short, you will not have to do Stage 5. After doing Stage 4, you will skip directly to Stage 6.

Logistics

These stages will be done in C++. For many of you, this is likely to be the biggest programming project you have ever done so far and so it might be useful to keep the following points in mind.

- **Platform** : The stages will be compiled and tested on the CS department Unix machines using the latest supported g++ compiler. You are free to do some development using other platforms but you must make sure that your project works with the official configuration.
- **Warnings** : One of the strengths of C++ is that it does a lot of compile time checking of the code (consequently reducing run-time errors). Try to take advantage of this by turning *on* as many compiler warnings as possible. The makefile that we will supply will have `-Wall` on as default.
- **Software Engineering** : A large project such as this requires significant design effort. Spend some time thinking before you start writing code.
- **Partners** : Each stage should be done by the entire project team. It is unlikely that one person will be able to finish the stages (no concession will be made to single-person teams). In real life, all software development is a team effort and you will be well-served if you can pick up some team management skills during this project. Disagreements can and will happen and it is your responsibility to get the project going regardless of differences. All members of the team will get the same grade for each project stage.
- **Due Dates** : Your group must submit the solution by the due dates. **No late submission will be accepted.** This is because the code for the next stage will contain the solution to the previous stage. So we cannot release the next stage until all groups have submitted the solution for the previous stage. Thus, if you submit late, you are delaying the start of the next stage. This is not acceptable because we are already on a very tight project schedule. Hence, if you do not submit by the deadline, we will have no choice but to grade your submission as zero. Even if you only have partial code, you should still submit by the deadline.
- **Exams** : The final thing that you should note is that **all topics covered in the stages will be fair game for questions during the midterm and final**. So make sure you know about what you (and your partner) did for the project.

A Reminder about Cheating

We believe in the highest standards of academic integrity. The project implementations of each group must be completely distinct from each other. You will not share any code across groups. Nor will you attempt to use any code from previous offerings of this course. All the code that you submit must be written by you and you only. Deviations from this will be punished to the fullest extent possible. Usually this means reporting to the department chair and the college, as required by UW. **We will use software tools built into Canvas as well as our own tool to insure that your project represents your work.**

Project Stages and Due Dates

The due dates will be announced by your instructor on Canvas.

- **Project Stage 3.** To work on this stage, read an [Overview of the Minirel I/O Layer](#).
- **Project Stage 4.** To work on this stage, read an [Overview of the Minirel Page Class](#).
- **Project Stage 5.** **You will not have to work on this stage. Your instructor will give further instructions.**
- **Project Stage 6.** **This brief note provides some tips for doing Stage 6:** [AnHai's note](#).