

# keyDB - MS3 Progress Report

keyDB is a console-based Database Management System (DBMS) built in OCaml

## Vision

Our final project turned out relatively on par with our initial vision. Although we were concerned that being down a team member may have impacted our productivity, our final system met our initial expectations. It was easier to communicate, take accountability, and make decisions. We had no major restructuring this sprint, so our system/vision did not change as much as previous sprints. Most of the structure changes were quite small and had to do with adding a function somewhere to moving exceptions or functions from one module to another.

## Summary of Progress

During our last sprint, our time was mostly focused on finishing Table, implementing the DBMS log, and enabling select with conditions. Our table functions were mainly used to analyze and manipulate data - adding/deleting columns, counting cells or null values in a column, and summing a column if possible. The DBMS log captures every user command in a separate log file so the user can see what they have done. Unfortunately, we did not get to implement undo.

During our demo we got to show off most of our DBMS's functionality, except for some exception handling as the discussion section was running late. Running the commands in INSTALL.txt demonstrates most of the database functionality. Bad commands are often caught with helpful error messages.

## Activity Breakdown

Ben:

- Implemented Table.sum\_column, Table.count, Table.count\_null
- Implemented log module: Log.make\_log, Log.get\_log, Log.clear, toplevel access
- Code quality cleanup and documentation, fixed broken test cases

Donal:

- Implemented Table.select with conditions taken into account
- Implemented Table.add\_columns, Table.delete\_columns
- Polished off the OUnit test suite

## Productivity Analysis

As mentioned before, we were still able to make quite a bit of progress even down to two people. We met often to discuss the project, and if we could not meet we still made sure to communicate our progress and any issues that came up. Additionally, we were quite impressed

with the accuracy of our road map, as our final sprint objectives actually matched those of our initial project charter. We wrote a lot of new code and, since we were almost done with the core functionality, didn't run into too much trouble. We met our good scope goals successfully and efficiently. Overall, we are very satisfied with our productivity and time estimations.

## Scope Grade

We believe we achieved good scope for this final sprint. We satisfied all the requirements we laid out for ourselves in satisfactory and good scope, but did not quite reach excellent scope. To do so, we would've needed to implement undo and enable it through the toplevel. We thought about how to implement this, and came across a major issue. Basically, we would have had to save instances of the database at every call that could potentially require an undo and involve bringing back data not stored in the log (eg. drop table, delete column, remove row). Overall, we came to the conclusion that it was not entirely necessary for our DBMS to be complete, and our time would be better spent elsewhere. Another feature we hoped to add (that we didn't foresee in MS2) was pretty printing. Ideally, when viewing a table in our DBMS toplevel, white space would be automatically added so the columns line up. We didn't implement this due to time. Ultimately, it's a UI improvement and has no effect on the actual database functionality (i.e. one could take the generated csv files and open them in a spreadsheet program to see the table nicely formatted). Overall, we did a significant amount of work during MS3 but there are still plenty of ways to improve our system.