**TEAM EVALUATION: TEAM 13 EVALUATING TEAM 23**

For the duration of this project, our team(team 13) was assigned and used the database from team 23 for building our application and running test cases. Members of Team 13 include Ryan Coverick, Curtis Wendel, and Ryan Pesek. Team 23 members include Rachel Glockenmeier, Colton Revia, and Alec Woelffer.

**COMMUNICAITON:** The rating that we decided to give team 23 for their communication is **4 out of 5.** The members of team 23 did not give us any problems when it came to contacting them. The primary mode of contact was email, though a few text messages were exchanged about little questions regarding the use of their library. During the development stages, team 23 was quick to send us their API and header files, and gave us prompt updates when changes were made to the API. The only reason that they did not earn a 5/5 is because they forgot to send us their finished database. However, when we asked them for the database they responded quickly and apologized for the error, so it was only a minor issue and it was handled effectively and in a timely manner.

**API STAGE:** The API for team 23 earns a rating of **3 out of 5** for their API. The API was generally clear, though there was a bit of confusion about what exactly happened during some of the functions, such as the CrossJoin function of their table class. The primary problem we ran into was that a number of their functions had a return value of ‘void’ instead of an object, which made those functions not accomplish what they were intended to do and thus impossible to test. This delayed our group in the writing of our test cases, and was more than a minor issue. However, because team 23 was quick to respond and made the appropriate changes in a reasonable amount of time, it was not a serious drawback.

**TEST CASES:** the database for team 23 earns a **4 out of 5** in the area of test cases. Though we later discovered some bugs that weren’t previously found in the test cases, their code passed most of the cases that we gave them. The database query function and the table join functions were the only functions that we ran into issues with during testing, however the rest of the functions in their database, table, attribute, and record class worked fine.

**FINAL DATABASE LIBRARY:** The final database library from team 23 earns a **2 out of 5.** During the writing of the database application, it was discovered that the query function of their database would not recognize names of tables in the database, even though the table had been successfully added to the database and verified by the GetTables() function. Because of this problem, the writing of the database application involved numerous roundabout fixes to this problem to properly carry out the queries, which was time consuming and involved a significantly larger amount of code than would have been otherwise needed. Because this proved to be a significant problem in building our database application, we decided that it was worth a lower score than other areas.

**PROBLEM HANDLING:** Team 23 earns **5 out of 5** for their problem handling. When we came to them with major issues regarding their database, they were quick to respond and generally got the problems fixed. When we had questions and bugs in their initial API they quickly fixed them and emailed us back with the corrections in a timely manner.