What We Changed

When refactoring the project, we added the functionality of a default option when creating DDL. Since our project has MySQL capabilities built in, you can create DDL in MySQL without needing to import anything. So, we made it easier for the user by allowing them to create MySQL DDL without having to find the class files needed to do so. We made this change throughout the system by including the default option in all of the menus, including the menu option to list all available products.

Aside from this, we also made some changes to improve the readability of the code. We removed some unnecessary statements in the code and some characters that were causing errors. When we were learning how the code functioned, we were very confused by how the code worked. Part of this confusion stemmed from different statements that had no function in the code. So, we removed as many of these statements as we could find so that future developers would be less confused when being introduced to the code. In “EdgeConvertFileParser”, many little issues were refactored. First, we changed many ArrayList types to use the correct types. We also changed multiple attributes to local variables because many attributes were only used in one place. We removed numLine, stNatRelFields, the booleanValue function, and multiple occurrences of casting in the makeArrays and isTableDup functions because they were never used or were just unnecessary. Finally, we used the contains function instead of the indexOf function in a few areas because it is easier to read.

How Our Code is Extensible

Our code is extensible for a couple of reasons. First of all, when we inherited the project, new database products could be added. We did not remove this capability, we simply made a default option available. So, if a user wanted to change the product they are using, they could simply select the necessary class files to accomplish this. Our changes just make it easier for a user to create MySQL DDL specifically.

Another part of our code that is extensible is the help system. The help system is built so that the menu is created based on a list of article titles and the information that is associated with those titles. So, if a developer wanted to add a new article, they could just add a new item onto that list, and the menu would be updated accordingly. This is extensible because it doesn’t require developers to hard-code the new articles that they want to add aside from the actual text that they want to add. This also reduces repeated code by looping through the list to create each article instead of hard-coding each article.