Coursework 2: Individual Report

24978

May 10, 2019

The main challenge of this coursework was the initial difficulty of understanding how to use JDBC and learning the correct syntax for the JDBC related methods. Once past the initial learning curve with JDBC the only real challenge in the coursework was the formulation of some of the more complex SQL queries. There was a strong desire to use a simple SQL query in a loop rather than using a more complex query that required multiple joins.

The later methods involving likes were quite interesting due to the many different cases that had to be considered. Implementing the `getAdvancedPersonView` method was also particularly interesting as it required a complex query that took some planning and though to implement. Another interesting part of this coursework was the ability to use different types of join within queries, whereas it seemed in the first coursework that only inner joins were required. The `createTopic` method was interesting as it required two insertions into two different tables with one of the insertions requiring the auto-generated key from the first insertion. Finding a way to return the generated key without requiring another query was a slight challenge as it appears there are many different methods depending on the SQL version being used.

Generally, a lot of the earlier methods were quite boring to implement as they only required simple queries. However, it is clear that they provide necessary functionality to the forum so must be there. The purpose of the `getLatestPost` and `countPostsInTopic` methods was slightly confusing as they are not used in the web interface, this also made the difficult to test as it seems that writing proper unit tests to ensure the functionality of these methods would be beyond the scope of this coursework. For the coursework I would have liked the opportunity to implement more methods similar to those in B.2 which required interesting SQL queries, so possibly it would have been preferable to be required to implement fewer difficult methods rather than more easier ones.

During this coursework I feel that I have learnt the basics of using JDBC as well as improved my skills at writing database queries. The coursework was also a good reminder for checking and testing for edge cases regarding invalid input. The coursework also did a good job teaching how to correctly handle and catch database errors.

After working with JDBC for a while my opinion of it is fairly positive, it feels fairly easy and intuitive to use. If I were to design my own API I would probably aim to have statements that did not require manually closing. This is something that can be easily forgotten about as there is no real indication that this has been forgotten about. Not having to manually manage resources is a large reason Java is preferred over languages such as C++, so it seems counter-intuitive that a database API for the Java language would require the developer to manually manage resources in this way. Having to manually commit insertions to the database is also something I would probably change as again it is easy to forget to do. Having the process of committing or rolling back the database depending on the outcome of the query may be something which could be encapsulated away from the user and handled internally. I would likely try to keep the syntax used in my own API similar to that used in JDBC as it is easy to learn and also fairly easy to read and understand what is meant to be happening.

As an additional note I believe the forum implementation would benefit greatly from some sort of back button to take the user back to the previous page they were on after performing actions such as liking posts or signing in. This functionality would also make testing the methods much easier as less navigation would be required.