Revised Project Proposal

Weisheng Wang ww2609 (working alone)

For my project, I'd like to continue my midterm paper's topic about "database-backed web application performance anti-pattern." This topic aims to help developers to address APs that hurt their applications' performance. The topic is relevant to the course not only because it has been well studied by many SE researchers, but also because it shares the same mission with the course: finding and developing tools to help software developers work on various tasks.

For the project, in particular, I want to focus on SQL-based APs and extend the original SQLCheck tool by enhancing its AP detection correctness (i.e. reducing its false-positive rate). In my midterm paper, I experimented with SQLCheck using one dataset schema from Kaggle and found that it has a high false-positive rate. After examining the false-positive cases, I find it solvable by implementing additional rule-based approaches and analyzing raw data's contents (this is proposed in SQLCheck, but the open-sourced SQLCheck tool does not support inputting raw data. Besides, SQLCheck uses data analysis to help detect APs. Here, I will extend the SQLCheck's data analysis algorithm (or propose a new algorithm) to reduce the false-positive rate.)

For the project, I will first examine and report on SQLCheck's performance in database schemas(The SQLCheck paper reports finding logical design APs(No Primary Key, Data in Metadata, Generic Primary Key, Multi-Valued Attribute, No Foreign Key APs) in its collection of Kaggle datasets' schemas). In the ideal case, I will experiment with 10 popular datasets from Kaggle. Next, I will implement my tool SQLCheck+ which will use the schema and perform data analysis on dataset contents to validate and improve the SQLCheck's results and yield the final AP detection report. For evaluation, I will mostly compare my tool with the original SQLCheck to measure how much the false-positive rate drops. (The SQLCheck paper also reports finding data APs in Kaggle datasets (APs detected in stored data such as Missing Timezone and Incorrect Data Type APs). However, since the SQLCheck tool does not support inputting data, I cannot compare the performance difference on data APs).

If time allows, I will also create query statements for 1~2 datasets from the 10 datasets, evaluate SQLCheck's performance on query APs(in my midterm paper, it's also found that SQLCheck may detect false-positive AP in query statements), and extend SQLCheck+ to reduce such false-positive detections.