Data models for GitHub's big data
Master Thesis — Summary

1 Justification

GitHub being the largest collaborative hosting platform for software engineering projects provide researchers an opportunity to analyze the publicly available data to answer interesting research questions, for example, 1) code evolving rate, 2) topics occurring most in desktop web-app development, 3) interactions distinguishing between developers joining/remaining outside project teams. All these questions try to achieve the same goal, i.e., to find answers in data. Data management is the crucial topic here, because the decision on how the big data from GitHub is managed has effects on the planned evaluation and the analysis on the data. For example, the data structure lying underneath the data has effects on performance of queries executed on the dataset. Many initiatives, such as GHTorrent/GHArchive, have been introduced in the past to support researchers in the data management. However, the successful ones of them provide data packed in the relational data model and none of them provides data management solution as a graph database on which researchers can execute graph queries efficiently. This is an interesting context, because of the following reasons, 1) GitHub stores the Version Control System (VCS) history in graph data structure, 2) GitHub attribute relations can be structured as a forest as depicted in Figure 1, 3) graph algorithms have their strong part in computer science for finding answers for problems that fit well in a graph data structure.

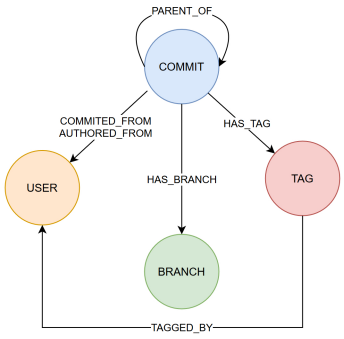

Figure 1: GitHub attributes and their relations expressed as a forest

2 Contribution

In order to open new research perspectives and define a base ground for future research questions, 1) we evaluate papers from the Mining Software Repositories (MSR) conference to define GitHub attributes relevant to researchers, 2) we define the data models and the DBMS that are used by researchers in their analysis, 3) we estimate if it is possible to transform the researchers' analysis in SQL queries to define the evaluation from the view point of the relational model. Our contribution can be summarized as answers to the following research questions.

- **RQ1** *Relevant GitHub Attributes*: Which data attributes from GitHub are relevant for researchers in the Software Engineering community?

  RQ1 answer: Attributes providing information about commits are most relevant. After that, attributes revealing information about repositories, such as number of files, hold priority. The third most relevant attributes are those that give information about the user.

- **RQ2** *Available Data Models and DBMS*: Which data models or database management system (DBMS) do researchers use for storing their data?

  RQ2 answer: JSON and the relational data model both emerge with the frequency of 36.36%. The graph data model shows up only at 9.09%. The contrast of the relational and the graph data model results to be of 300%. Frequency of all other data models results 4.54% each. No DBMS usage accounts to be at 53.33% frequency, MySQL at 33.33%, Google's BigQuery and SQLite both at 6.66% each.

- **RQ3** *SQL transformation*: Is it possible to transform the researchers' evaluation on GitHub attributes in form of SQL queries?

  RQ3 answer: For all papers, the transformation of the researcher's evaluation in SQL queries is *limitedly* possible, because of lacking schema or non accessible data. The estimated Extract, Transform, and Load (ETL) cost for 57% of the papers is high and for 42% average.

3 Conclusion

we have discovered that the big data researchers collect from GitHub mostly lacks proper data management. However, the transit of data management for software engineering research analysis to a DBMS is limitedly possible. The results inform that there are cases where the relational data model emerges, albeit mostly without well-defined schema. On the other side, the use of the other popular data models, such as the graph data model, rarely comes in the focus. The aforementioned results are especially interesting, because we have also discovered 1) commit, 2) repository, and 3) user related attributes to be the most relevant for the software engineering research questions that fit well in context of a *forest* (graph data model).