

INF 551 – Spring 2020

Project Description

Keyword-Driven Exploration of Relational Data Using Firebase

The goal of this project is to import data from MySQL to Firebase and develop a keyword-driven interface to explore the data via their foreign-key relationships. As a motivating example, consider a MySQL database “enrollment” with tables: students, courses, and take, as shown below.

id	name	program	age	gender
100	John	Applied data science	20	M
200	Mary	Computer science	25	F
300	David	Applied data science	35	M

(a) Students

id	title
INF551	Databases
INF552	Machine learning
INF553	Data mining

(b) Courses

sid	cid
100	INF551
200	INF551
100	INF553

(a) Take

The project consists of the following tasks.

1. Develop a Python program “import.py” that imports the data from a MySQL database to a Firebase realtime database.
 - Execution format: `python import.py <mysql-database-name> <firebase-database-node-name>`
 - For example, “import.py enrollment enrollment” will connect to a MySQL database “enrollment”, transform and store its data in `https://<your-firebase-database-url>/enrollment`.
2. Develop a Web browser based user interface or mobile app that allows users to explore the content of a relational database stored in Firebase. Specific requirements are as follows.
 - The interface should have a search box that allows users to enter keywords. It should tokenize the search query and return users the tuples of tables that contain at least one keyword in the query. The tuples may come from different tables and should be ordered by the number of keywords appearing in the tuple. Note that it is possible that different keywords appear in the values of different attributes in a tuple. In this case, the tuple should be ranked after the tuple which contains all keywords in the same attribute (or a smaller number of attributes). The search should be **case-insensitive**.
 - For example, for the query “john science”, the tuple for student “100” will be ranked higher (since it contains both keywords: one in name attribute; the other program) than the tuple for students “200” (which contains only one keyword).
 - It should allow users to continuously navigate from returned/current tuples to other tuples through the foreign key relationships. For example, the tuple for student “100” would have a link on its “id” value (100) and when user clicks on it, he/she will be presented with tuples in the “take” table whose “sid” are 100; or vice versa (that is,

there should be a link from sid of take to id of students on the corresponding value). If users click on cid value “inf551”, they will be presented with the courses tuple for “inf551” (i.e., first row in the table).

In other words, imagine an expanding graph that starts with search results (tuples that contain the keywords) and grows to include more and more related tuples when users follow the “hyperlink” between tuples that capture their FKs or referential relationships.

Additional requirements are:

- There should be no more than two persons in a project group (individual project is ok).
- You need to structure the data in Firebase in such a way that allows efficient processing of queries. For example, you need to build index for efficient keyword search of tuples.
- You are required to test and show your app using at least three databases: two will be chosen by yourself and the third will be provided to you. Each database should have at least 3 tables and at least two foreign key relationships.
- You should not download the entire database into the browser and app and implement the search and navigation using the “local” data. Instead, it should send requests to Firebase server to retrieve only the data necessary to complete the search or navigation.
- For the evaluation purpose, report the amount of data (communication overhead) retrieved from the server on every request (search or navigation).

The project consists of 3 phases: proposal, midterm report, final report & demo. The total point of the project is 100, broken down as follows.

- Proposal: 10 points
- Midterm report: 10 points
- Final report: 20 points
- Demo: 10 points
- Project implementation & system performance: 50 points

Proposal (1-2 pages):

Your proposal should include the following content. Please also prepare 1-3 slides for a short presentation (1-2 minutes) of your project idea. All members should be present.

- Two databases you choose, including some example data.
- Initial design of your app/interface. How do you plan to implement the features?
- Group formation: who are in your group? What will be each member’s responsibility? Is your group equipped to implement the application by the end of the semester?
- Milestones: a project timeline with milestones. (no need to present this, but include it in your proposal report).

Midterm progress report (1-2 pages):

- Provide a checklist showing the items in your timeline and the status on each time (complete, on-going, etc.).
- Provide a screenshot of the components you have completed.
- Are you on track to achieve your milestones?
- Any challenges you have encountered? Any helps that you will need?
- Any other things you think should be reported in the midterm?

Final report (5-10 pages):

It should be a comprehensive report. You may include the contents from your proposal and midterm report, with changes to reflect the final implementation of your project. The final report should have the following parts.

- Project idea.
- Screenshot for each working component with a description.
- Implementation details.
- Performance analysis on query processing & exploration process.
- Responsibility and work of each group member.

Final Demo:

- Demo of your app (10-minute) will be done in the last week.
- Show the working of each component of your system.
- All group members should be present at the presentations.

Deliverables:

Your phase & final reports and project codes.