

Inf 551-Final Report

Linlin Sun:8249831190 Zepei Zhao:5635405542

YouTube: <https://www.youtube.com/watch?v=na99jO3SsZY>

Source Code: https://drive.google.com/open?id=1YY3gpo6v_4tkE40V0T81MHknXsyhCF4f

Introduction

This project develops a web browser for exploring relational data stored in firebase. It provides an user friendly interface to retrieve data driven by keywords. In this project, we work on importing data from MySQL to Firebase and develop a keyword driven interface for users to explore data via foreign-key relationships. In the whole project, we complete several vital processes, database selection, index construction, data import, user interface design, major functions implementation and project optimization. Besides, sort and navigation are also realized in our web browser.

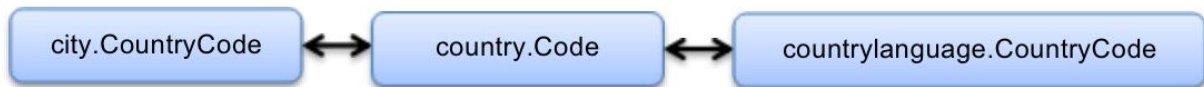
The project requires a web browser interface for user interaction, so a search box is designed for getting user input. Then the input delivered as the keywords to request databases, and results generated by inverted index will be shown in the front page. Meanwhile, tuples from different tables are related to each other by foreign key, so that users can navigate to other tables. Besides, if the input keyword is not a single word, the results are required to be sorted by frequency.

To accomplish this project, we develop a Django framework and complete front-end and back-end components. Basically, python is applied for framework construction, and we use html, css to design the web page. Besides, javascript is utilized to dynamically generate tables and implement search and databases select functions.

Database Selection

There are three databases selected in our project, world, sakila and basketball women. World database contains city, country and country language tables, which show details of different countries and cities in the world. Sakila is a large database, and we choose actor, film_actor and film table, providing film description and relationship between films and actors. The third database is talking about women basketball players' basic information, including their awards history, team situation and personal information.

World:



Sakila:



Basketball Woman:



Chart-1 Foreign keys of three databases

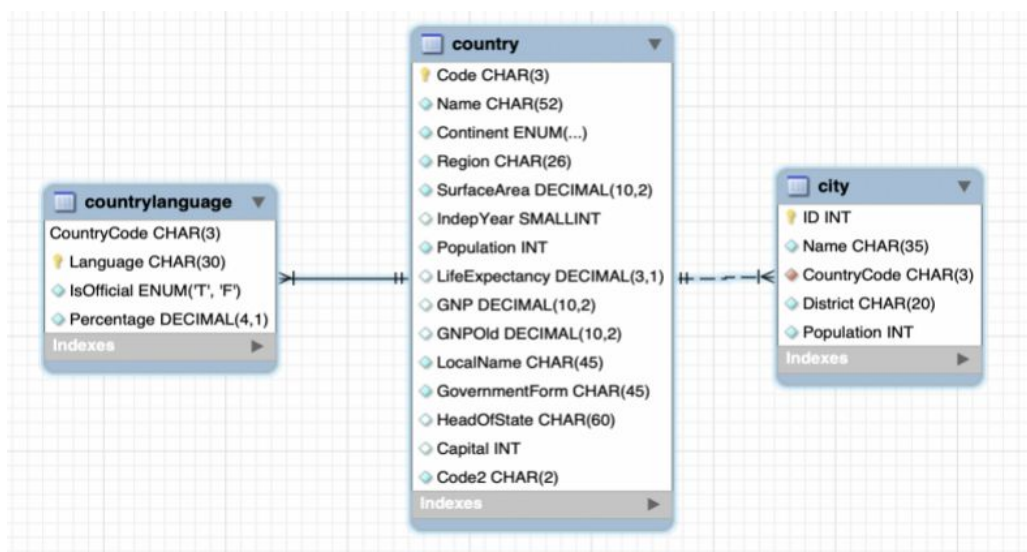


Chart-2 ER Model of World

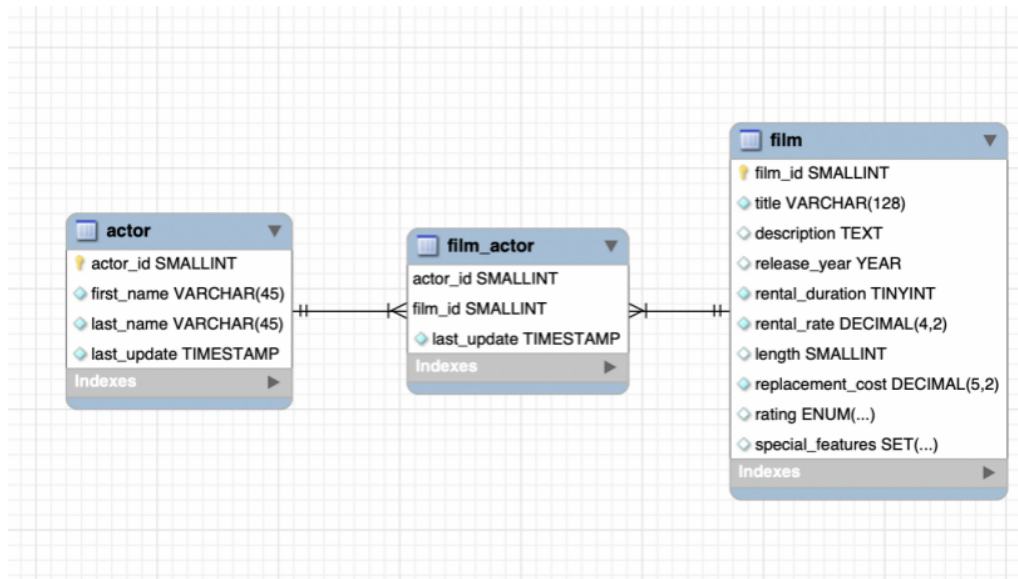


Chart-3 ER Model of Sakila

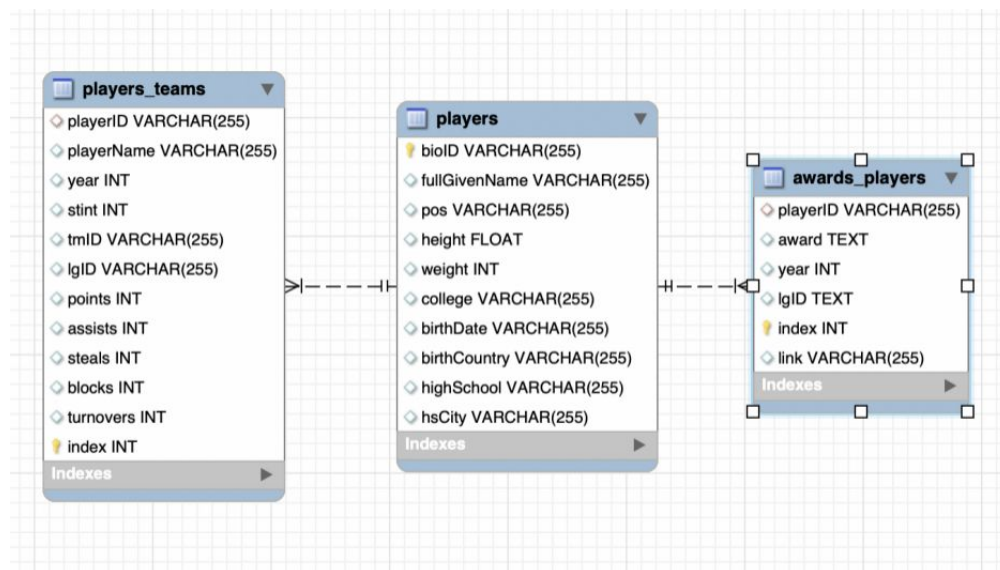


Chart-4 ER Model of Basketball Woman

Data Structure in Firebase

1. Data Structure for Searching

When users input keywords and click the search button, the request will be sent to firebase and access certain databases. Therefore, all the words having practical meaning are extracted and constructed to an inverted index. In the structure of inverted index, the key is a single word, and value is all the objects who contain this word. The structure is shown as the following figure.



Chart-5 Data structure for searching

2. Data Structure for Navigation

The implementation of navigation depends on foreign keys, so that one word can point to records in the other table. Therefore, the data structure we create for navigation is using table name as the key, and table it references to stored as the value. For example, regarding the table `players_teams`, its foreign key `playerID` references `bioID` in table `players`. So, taking foreign key of `players_teams` as index, the records whose foreign key in the player table equal to the index are stored under the index. The data structure is built as the following figure.

Chart-6 Data Structure for Navigation

Web App-Data Explore

We built the web app using the Django web framework, and all back-end functions are developed based on Python3. In order to connect the web app with Firebase database, we also use the library 'pyrebase' which supports Firebase API and helps us to format the query which can be sent to Firebase and get the data back.

1.Details in building Django project

To start a Django project, we need to activate a virtual environment for the project and download all libraries the project needs, such as django, pyrebase, json and so on. When the django project has been set up, we get a directory in the structure below.

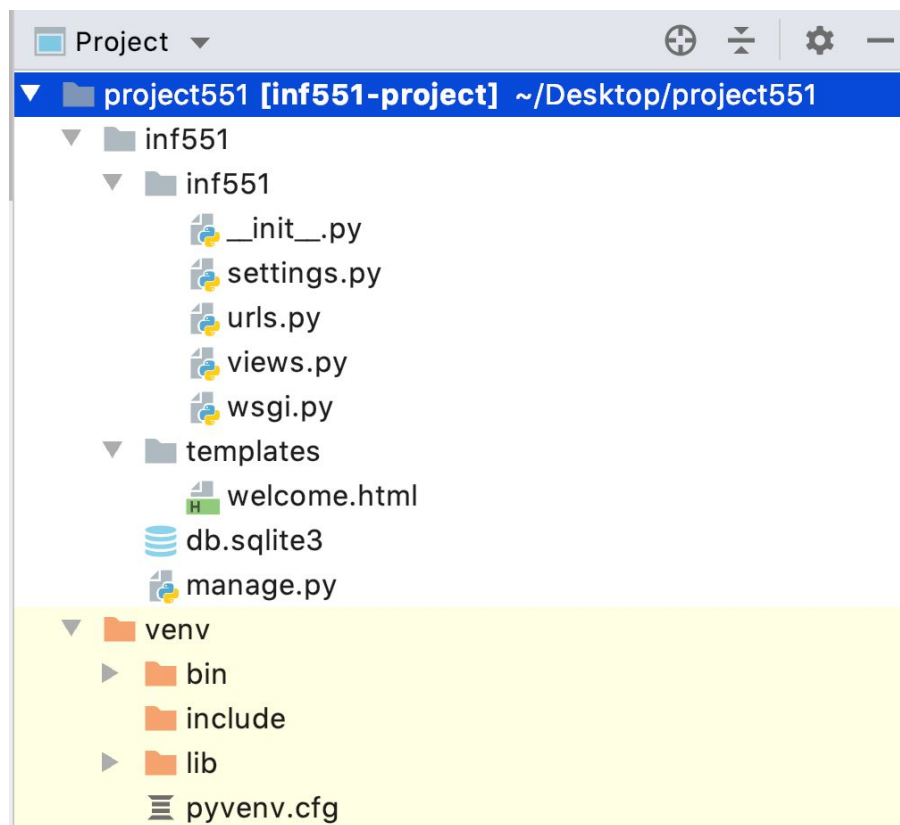


Chart-7 Directories for Django Project

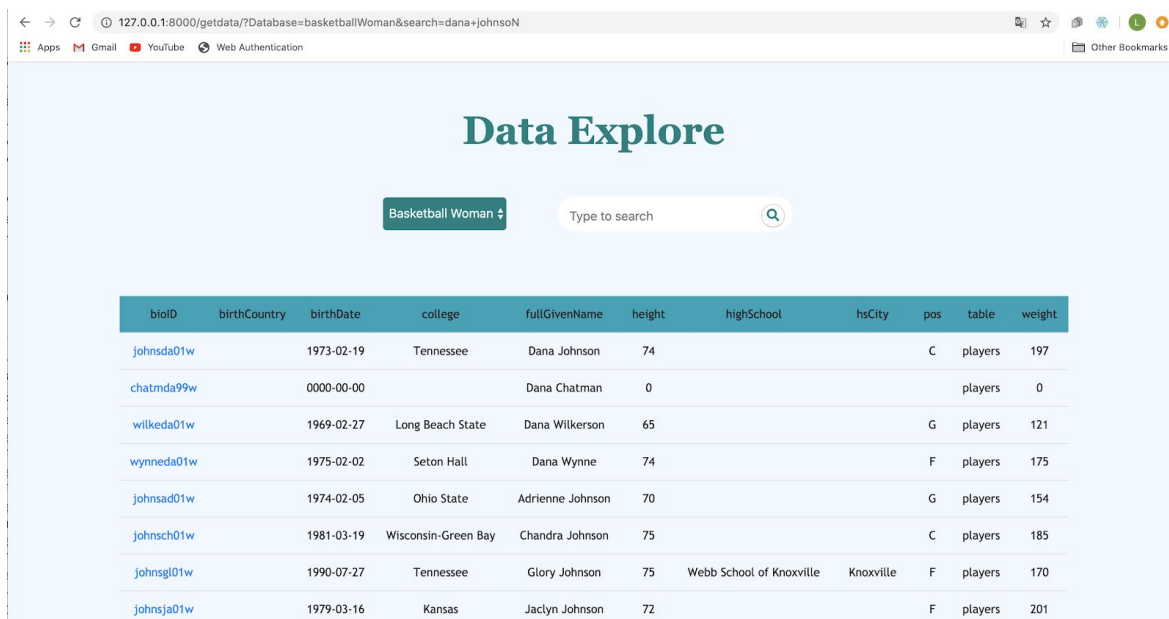
2.Start App and create functions in Views.py

When the django project has been set up, we need to create an app and install it into the Django project. In order to get connection between front end and back end, we write the function inside of views.py which can render html files under the directory templates.

3.Searching process

When all the environments for the django project get prepared, we can start the server in the terminal using 'python manage.py runserver' and then visit the web page which has been rendered in views.py.

The primary idea about this UI is to make it straightforward and easy for users to search the information based on three datasets provided, so the interface is more like a search engine. After choosing the database that uses what to explore, typing the key words inside of the search box which is case-insensitive. Then all the records which are relevant to the key words will show in the tables.



bioID	birthCountry	birthDate	college	fullGivenName	height	highSchool	hsCity	pos	table	weight
johnsda01w		1973-02-19	Tennessee	Dana Johnson	74			C	players	197
chatmda99w		0000-00-00		Dana Chatman	0				players	0
wilkeda01w		1969-02-27	Long Beach State	Dana Wilkerson	65			G	players	121
wynneda01w		1975-02-02	Seton Hall	Dana Wynne	74			F	players	175
johnsad01w		1974-02-05	Ohio State	Adrienne Johnson	70			G	players	154
johnsch01w		1981-03-19	Wisconsin-Green Bay	Chandra Johnson	75			C	players	185
johnsg01w		1990-07-27	Tennessee	Glory Johnson	75	Webb School of Knoxville	Knoxville	F	players	170
johnsja01w		1979-03-16	Kansas	Jaclyn Johnson	72			F	players	201

Chart-8 User Interface for Searching

Chart 8 gives us an example about searching 'dana johnsoN' in the 'Basketball Woman' dataset. When users click on the search button, the information they chose or typed in the search box will be sent to the back-end via url(<http://127.0.0.1:8000/getdata/?Database=basketballWoman&search=dana+johnsoN>). Functions related to the form in html file will get the information provided by url and extract the database and key words inside of it.

Inside of the views.py, we have set up the configuration which allows the query to get to the specific Firebase project. After getting the information from the front-end, the function will help us to format the query and send it to the Firebase.

```

from django.contrib import auth
config = {
    "apiKey": "AIzaSyCE3aBjH1Li0cAZ_GG3d7lgFhXZ3Q8gIb8",
    "authDomain": "inf551-d17d1.firebaseio.com",
    "databaseURL": "https://inf551-d17d1.firebaseio.com",
    "projectId": "inf551-d17d1",
    "storageBucket": "inf551-d17d1.appspot.com",
    "messagingSenderId": "1051148253779",
    "appId": "1:1051148253779:web:105a4f728be6f062db64a6",
    "measurementId": "G-4B5X2KPSJQ"
}

```

Chart-9 Create Configuration for Firebase

4. Ranking the results:

When the function in views.py get the keywords in the url, it will format a query:

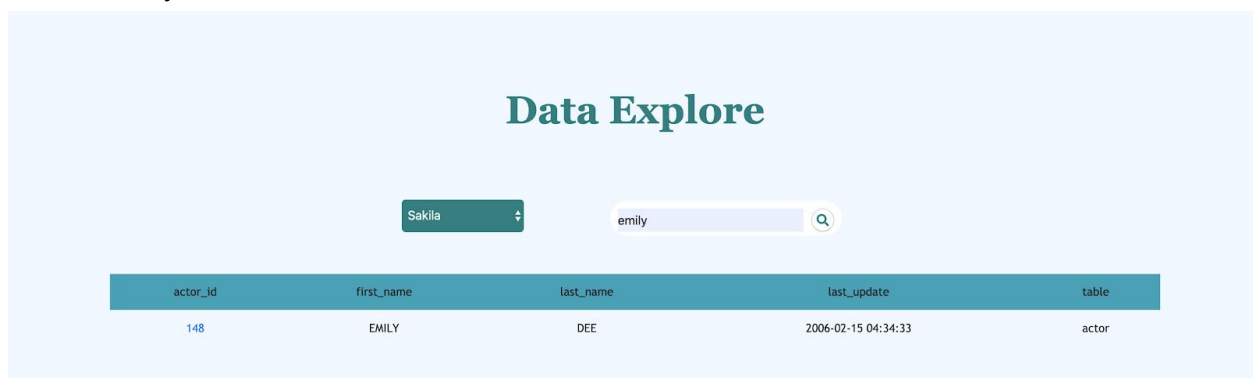
```
value = database.child(dataset).child(keyword).get().val()
```

Based on the query, for every request, only the corresponding records will be retrieved.

In which, the value contains all records containing keywords and, in the same function, we need to rank the record based on the frequency of keywords and whether all keywords appear in the same attribute. The records with the highest frequency of key words and some of keywords in the same attribute will come first. The following records only contain part of the keywords or less relevant to the keywords.

5. Data navigation:

Data navigation can be shown by using the example “emily” in the Sakila database. When users would like to get the basic information of someone called “emily”, they can search the actor’s name directly.



Data Explore

Sakila emily 🔍

actor_id	first_name	last_name	last_update	table
148	EMILY	DEE	2006-02-15 04:34:33	actor

Chart-10 UI of Navigation

By clicking on the actor_id. they can get more information about the films that this actor has attended.

Data Explore

Sakila
↓

Q

actor_id	film_id	last_update	table
148	27	2006-02-15 05:05:03	film_actor
148	57	2006-02-15 05:05:03	film_actor
148	133	2006-02-15 05:05:03	film_actor
148	149	2006-02-15 05:05:03	film_actor
148	226	2006-02-15 05:05:03	film_actor
148	342	2006-02-15 05:05:03	film_actor
148	368	2006-02-15 05:05:03	film_actor
148	422	2006-02-15 05:05:03	film_actor
148	468	2006-02-15 05:05:03	film_actor
148	633	2006-02-15 05:05:03	film_actor

Chart-11 UI of Navigation

If the users would like to get more details about the specific movie, they can get the information by clicking on the file_id.

Data Explore

Sakila
↓

Q

description	film_id	length	rating	release_year	rental_duration	rental_rate	replacement_cost	special_features	table	title
A Stunning Epistle of a Man And a Husband who must Reach a Mad Scientist in A Jet Boat	57	90	PG-13	2006	4	2.99	18.99	Deleted Scenes	film	BASIC EASY

Chart-12 UI of Navigation

For each navigation, only send one query to firebase, which will request for the data in the inverted index node, so only the records relevant will be retrieved.

Responsibility:

Linlin Sun:

- Buildup up the web app framework in Django.
- Write the back-end function which support data searching and data navigation
- Connect the front-end to the back-end using JavaScript
- Write the final report

Zepei Zhao

- Write import.py to upload three datasets to Firebase
- Create inverted index for three datasets
- Beautify the interface using CSS
- Write the final report