0652 The Principle of Database (Fall 2021)

Report

Xiong Boya, Zhang Jun, An Jinyi, Yuan Yue, Tan Feng, Xiang Yiran

1. **Completion of the Project (10%).** The first part of the score comes from the completion of the project, i.e., all functions claimed in the report must be run with no serious issues. However, you may submit incomplete project as long as it is runnable.

It shows the origin, development and branches of surnames, and the development of different surnames presents China's long-standing culture. It mainly includes the following functions:

1) User registration and login

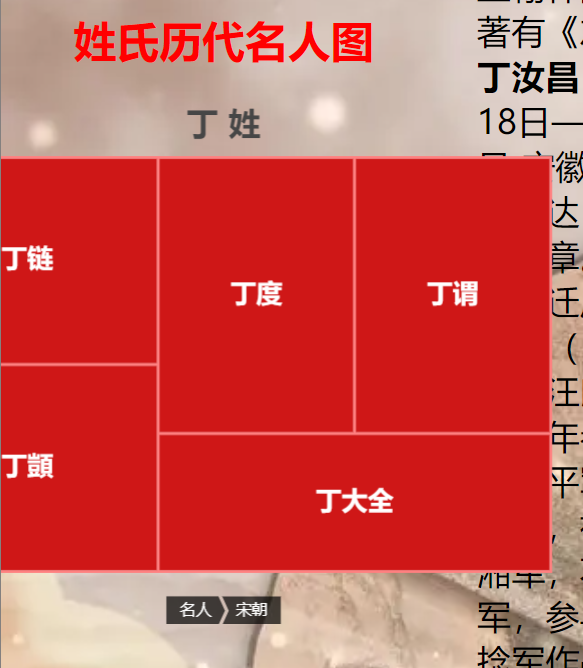
2) Search for the corresponding last name

3) Surname culture in written form

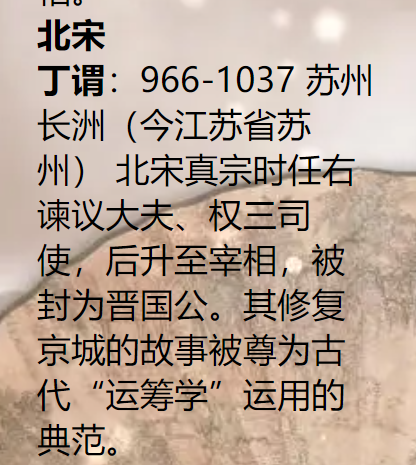
4) The map format shows the proportion of surnames in the population distribution of various provinces in the country:



5) Select a celebrity with the surname in a dynasty through the chart to display the celebrity information. There are dynasties divided into Tang, Song, Yuan, Ming, Qing, Republic of China and the People's Republic of China:



Here are all Ding names in the Song Dynasty included in the database. After selecting "Ding Wei", you can see Ding Wei's introduction:



6) Surname word cloud: It displays the words that appear frequently in all the information corresponding to the surname. Users can have a preliminary and intuitive understanding of the surname through the word cloud:

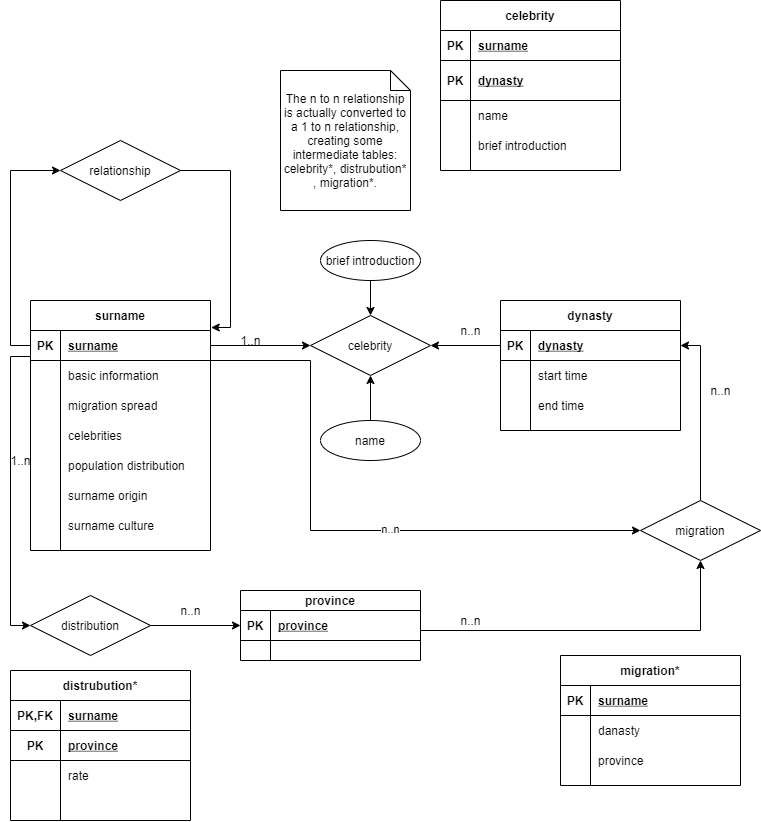


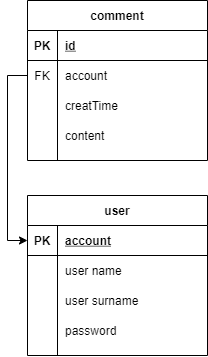
7) Since each surname has multiple surname origins, we use text to show the origin of each surname

8) Surname relationship diagram: show what surname evolved from, or what surname evolved into. It can be understood as using a diagram to show the source and evolution relationship between surnames

9) Surname path map: Use the map to visually display the migration process of surnames in different dynasties

2. **Database Design (10%).** Show ER diagram and explain the normalization forms used in the project and why you use it.





(ER pic)

The normalization forms we used in the project:

**1NF**-Each column of a database table is an indivisible basic data item. The same column cannot have multiple values, that is, an attribute in an entity cannot have multiple values or duplicate attributes.

1NF is the basement of a database system, we have split all attributes that contains more than one item in our database(as ER picture) and we create different names for each attribute which not only fit their functions, but also help us distinguish them, means we used 1NF.

**2NF**-The attributes of the entity are required to be completely dependent on the primary key.

Since most tables in our database contain only one attributes for primary key, and other attributes in the same table are dependent on the primary key. So the attributes in these tables are all completely dependent on the primary key. Otherwise we guarantee the non-primary attributes are completely dependent on the primary key consists of more than one attribute, for instance, in the table migration, means we used 2NF.

**3NF**-Each non-primary attribute is neither partially dependent nor transitionally dependent on the primary key.

Since we used 2NF, and in one table, the non-primary attributes are not connected or dependent on other attributes, so there is no transitionally dependence,

In the table celebrity:

(surname, dynasty, name) → brief introduction

In the table surname:

surname → basic information, migration spread, celebrities, population distribution, surname origin, surname culture

In the table dynasty:

dynasty → start time, end time

In the table distribution:

(surname, province) → rate

In the table migration:

(surname, dynasty) → province

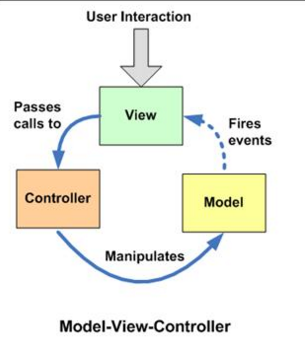
,means we used 3NF(also proved we used 1NF and 2NF).

**BCNF**-All non-primary attributes are fully functional dependent on each candidate key; All primary attributes are fully function dependent on each candidate key that does not contain it; No attribute complete function depends on any set of attributes that are not candidate keys.(We may not think about this field.)

3. **Architecture (10%).** You must explain the architecture of your application. In addition, you are expected to develop application that is dynamic and must show aggregation of data.

In this programme, we use MVC.

The full name of MVC is Model View Controller, which is the abbreviation of model-view-controller. The business logic is aggregated into one component, and the business logic does not need to be rewritten while improving and customizing the interface and user interaction. MVC was uniquely developed to map traditional input, processing and output functions in a logical graphical user interface structure.



**Data relationship**

View accepts user interaction requests.

View forwards the request to the Controller.

Controller operates Model to update data.

After the data is updated, the Model notifies the View to update the data changes.

View update change data.

**Way**

All methods are one-way communication.

**Structural realization**

View : Use Composite mode.

View and Controller: Use the Strategy pattern.

Model and View: Synchronize information using the Observer pattern.

**Use**

View in MVC can directly access Model! Therefore, the View will contain Model information and inevitably some business logic. In the MVC model, more attention is paid to the constant of the Model, and there are multiple different displays of the Model and View at the same time. So, in the MVC model, the Model does not depend on the View, but the View depends on the Model. Not only that, because some business logic is implemented in the View, it is difficult to change the View, at least those business logic cannot be reused.

Since the program is presented in the form of a web page program, we use some techniques to achieve dynamic development as follows:

1) Django build program architecture

2) HTML, CSS, JS, vue construct front-end

3) Data visualization through python-pyecharts

4) Store data via SQL

4. **Security (10%).** Your system is supposed to be safe from standard hacking tricks, e.g., SQL injection.

①User identity authentication:

The system identifies the user identity according to the user identification, and the legal user is allowed to enter the computer system.

②Access control:

The database management system carries out access control and only allows users to perform legal operations. The access control mechanism consists of: defining the user authority and registering the user authority in the data dictionary. Legal authority check: the user sends a request to access the database, and the DBMS looks up the data dictionary to check the legal authority.

③Prevent SQL injection and csrf injection:

Use ORM to prevent SQL injection. At the same time, the server will generate some values to put in the form to prevent CSRF injection. That is, if you want to submit the form, you must first obtain a value from the server before submitting it to the server. Otherwise, the server will refuse to accept it. This is to prevent CSRF injection.

5. **Complexity and Performance (10%).** You must show-off your ability by developing a system that has enough complexity, and you need to defend it in your report. Your system must not be too slow, and you must argue how you optimize your queries other parts of the system.

**Note:** The best way to show-off the complexity and performance is to have some data analytic system.

First, our system consist of an intact database and a high-connected front-end with back-end system was built on our own server. Our database contain several entity tables, such as surname; dynasty; province and relation tables, such as celebrity; distribution; migration and so on. Meanwhile we build the database, we used 1NF, 2NF, 3NF and BCNF, which means it is easily to add or delete or change or select information we need to the database.

Second, in order to shift the database, we make to role in our system, one is visitors(also called vip) who can surf the web and add comment, the other is manage(maybe only the developer ourselves) who can shift the database in the system. That also keep the security of our system.

Third, we create the function of query; we clean the text and data(for instance, we make an array that store the keyword of the place and the dynasty of China, and find them in the text data we get by python) before we build database which can improve the system performance. Since the query of surname only contain one or two letter(in Chinese), to optimize our system is to simply provide an exact search for users.

Forth, in the aspect of storage, we use the following five methods to optimize our system:

1. Reduce data access (reduce disk access) by directly got which information users need to search on the navigation.

2. Return less data (less network transfers or disk access) by the same method as last.

3. Reduce the number of interactions (reduce network transmission) by store suitable information in the database for users’ querying.

4. Reduce server CPU overhead (reduce CPU and memory overhead)

5. Use more resources (increase resources)

6. **Bonus ( 10%).** If you are able to show-off your ability to develop a high-level application and use some modern tools, then you are eligible to get an extra score up to 10%.

TF IDF technology is used to measure the weight of each word, and a word cloud is generated according to the weight;

The acquisition of database data resources is obtained from web pages by using web crawler technology under the framework of selenium;

The news details page of our website does not crawl data temporarily, but directly jumps to another page

Since we have climbed out much information on the website, the information that we require to illustrate in our database still maintain much vacancy. So we use RNN(Recurrent Neural Network) which means a type of recursive neural network that input a sequence of data, and make recursion on the developing direction of the sequence and all cycle units linked by chain, to create the name, surname and other information.

The project (code) of our program is submitted at the website:

http://sandbox.bundit-lae.me/~u2019111329/

In case the project folder cannot be found, please visit：

<http://sandbox.bundit-lae.me/~u2019110278/>

OR <http://sandbox.bundit-lae.me/~u2019110734/>

NOTE：Please see GROUP-PROJECT-README.txt