

Database Mini-project Documentation

Roll number: 9070

Introduction

The COVID-19 epidemic is closely related to everyone's life. Faced with the massive amount of data generated by various media every second, it is challenging for people to form a comprehensive understanding of it with human brains. Structured query language (SQL), however, can precisely meet this demand under the function of data definition. This mini-project focuses on the realization of two following functions: *checking the country with the largest number of cumulative confirmed cases as of the specified date* and *visualizing the number of cases in some key countries*, by using SQL, direct operation user interface, and heat map to realize information interaction and knowledge production.

Software Design

The software adopts a direct operation user interface and is made based on the pygame and pycharts packages of Python programming language. Users can run the software via different applications, such as command prompt, then open a simple initial interface, which provides a line of prompts and a button that is designed to be clicked to enter the date specified by the user. Type in a regulated date (in the form of Year-Month-Day) and press Enter, then the interface will return the country with the largest number of cumulative confirmed cases till that day and the corresponding number. Besides, the software will generate an HTML file containing a world map (an example shown in fig. 1), intuitively showing COVID-19 situation of the day in some key countries (China, US, United Kingdom, Italy, France, Germany, Spain, and Iran).

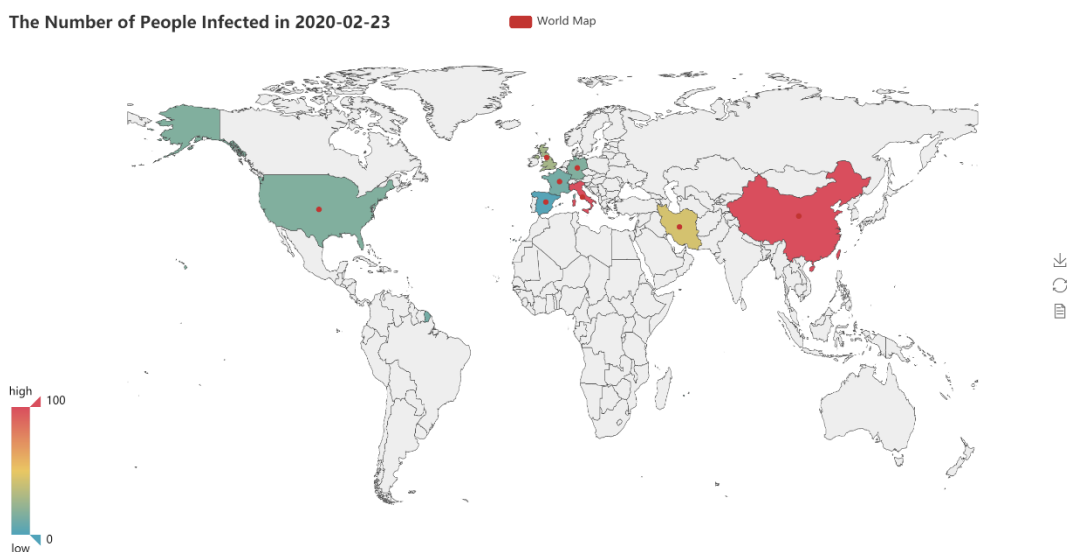


Figure 1: Heat map of COVID-19 cumulative confirmed cases in 8 key countries, 2020-02-23

Database Design

The database files used in this project are countries-aggregated_csv.csv and key-countries-

pivoted_csv.csv, downloaded from <https://datahub.io/core/covid-19>. According to the data features of these databases, table ONECOUNTRYCONFIRMED and KEYCOUNTRYCONFIRMED were created. Detailed attributes and relationships representing by the entity-relationship diagram (ERD) are shown below (fig. 2).

The tables in this project are satisfied amidst the first normal form (1NF), and the query efficiency is relatively high. The date entered by the user is used as the foreign key and a constraint, to participate in data querying in tables.

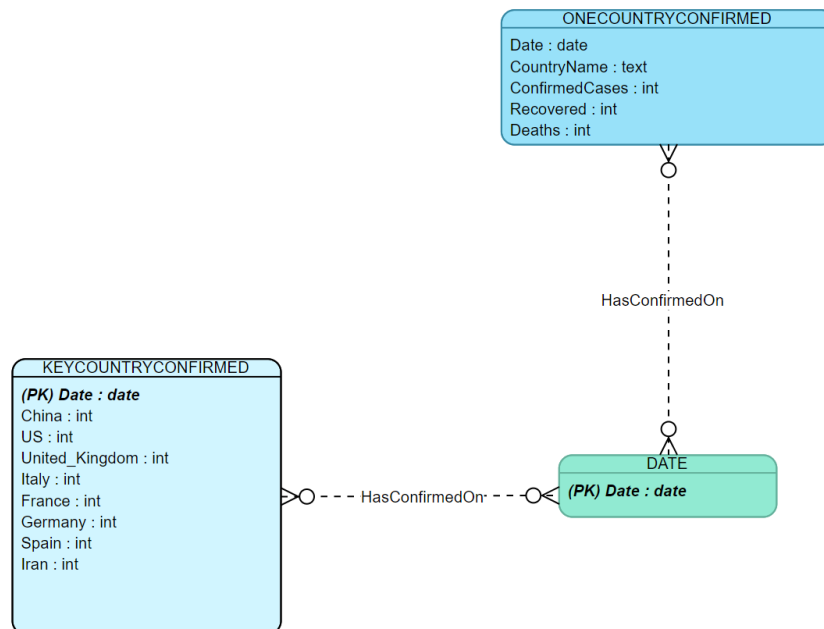


Figure 2: ERD of databases DATE, ONECOUNTRYCONFIRMED, and KEYCOUNTRYCONFIRMED

Cardinality constraints involved are both many-to-many relationships (M: N).

Input

The user needs to type the date in the form of Year-Month-Day, which should be greater than or equal to 2020-01-22 and less than or equal to today's date.

Output

The user interface outputs the country with the largest number of confirmed cases till that day and the number of cases. An HTML file reflecting the COVID-19 situation of 8 key countries on that day will be generated in the same directory of Python code, containing a world map in the form of a heat map.

Discussion

The user interface design of the project is simple and can realize the tasks of querying and visualizing. The software can function as expected, and the exception handling mechanism is

also well designed.

The relational model of this project has a certain degree of redundancy since it only satisfies 1NF. For example, the number of confirmed cases in the two tables overlaps. However, there are discrepancies in databases from different information sources, so the relationship between two tables cannot be easily set up (for example, the values of ONECOUNTRYCONFIRMED.ConfirmedCases cannot be used instead of that of table KEYCOUNTRYCONFIRMED). Furthermore, the database involved in this project is relatively simple, and the query algorithm complexity is relatively low. These defects do not affect the normal function of the software. In fewer entity relationships, the idea of functional dependency analysis can improve query efficiency, whereas, in larger entities and more complex relationships, the balance of non-redundancy and query efficiency should be considered.

Due to lack of time, the project is rough, so as the user interaction. Although the heat map is intuitive to use, it is highly fragmented with the user interface and leads to a bad experience. In the Database Software Technology lessons of the next semester, knowledge and skills about web design need to be used to optimize the project.

Acknowledgment

Thanks to Dr. Wanlu Liu for teaching SQL and the code about pycop2 and pygame. Thanks to some classmates for their help. Names have been hidden.