数据库原理与应用

期末综合设计



班级： 2018级电信2班

姓名： 张澍森

学号： 222018601105037

选题： 高校学籍管理系统

Contents

[Abstract 3](#_Toc60242645)

[Design Techniques and Development Environment 4](#_Toc60242646)

[Database Design 4](#_Toc60242647)

[Java Design 8](#_Toc60242648)

[Development Environment 8](#_Toc60242649)

[System Implementation and Users Guide 9](#_Toc60242650)

[Program Testing and Evaluation 10](#_Toc60242651)

[Program Testing 10](#_Toc60242652)

[Program Evaluation 12](#_Toc60242653)

[Conclusion 13](#_Toc60242654)

[Appendix 14](#_Toc60242655)

## Abstract

There are management and coordination systems everywhere in our life, such as the library management system and the car park management system. At their core is database software, such as Access and SQL Server. At the same time, for a student majoring in electronic information engineering or software engineering, database is a basic skill that should be mastered proficiently. In this semester, we finally took the database course.

After learning database theory course for one semester, I have a certain grasp of database theoretical knowledge, including basic operation, structural framework, design ideas and so on. But in fact, I didn't take many experimental courses, but database is a practical course, and computer operation is the most important part of its learning. Therefore, at the end of the semester, I decide to establish a simple student score management system to consolidate the theoretical content of this semester, enrich my experimental knowledge, and cultivate my database thinking.

A simple score management system doesn’t get connection with the Internet, so the database has to be local and packaged with main file. Main file is used to create UI to give instructions to users and show operation results. The knowledge points contained by the whole program include the creation and overall planning of database, triggers, tables and their relationships, database along with UI design and their connections. In this report, the complete process of designing, founding and testing database and main file is shown and principles or theories are also explained. To be specific, in part Design Techniques and Development Environment, I will show my thinking and solution of designing database and java main file, with what software and environment are used while working. In the second part System Implementation and Users Guide, the program and its principles will be explained in detail with some code in the views of users. So through this part users and readers can know how to use the management system. After that, I will test the program, demonstrate it with figures and videos, and evaluate its performance. And finally, a conclusion for the project development and the learning process will be given with prospect to the future.

## Design Techniques and Development Environment

### Database Design

The first and the most fundamental step is to design several tables in the new database called XJGL to satisfy the project requirements. Firstly, table Admin is established to save the information for administrators to log in. To achieve the management of information of students, a table called Student is created with their basic information including their IDs, names, birthdates, enrollment dates, genders (male or female), telephone numbers, classes, academies and majors. As for administering students’ grades, table Grade is set to record students’ IDs, their courses and corresponding grades. In order to regulate the information of reward and punishment of students, table RP is built with reward and punishment IDs, students’ IDs, reward and punishment dates, types (reward or punishment), reasons and degrees. The last table is named Class to count the number of students in each class. The details in design are shown from Fig.1 to Fig. 5.

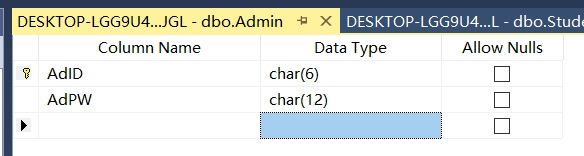


Fig.1 The Design of Table Admin

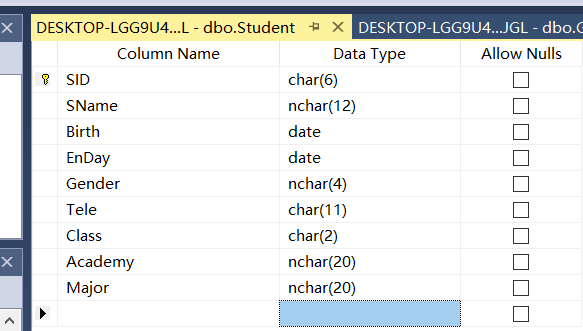


Fig.2 The Design of Table Student

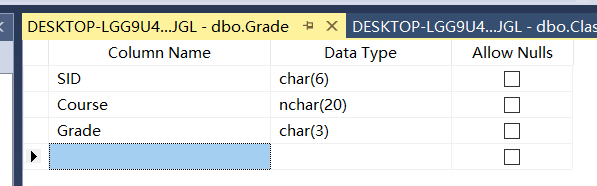


Fig.3 The Design of Table Grade

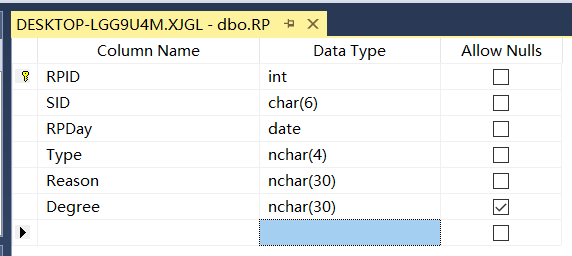


Fig.4 The Design of Table RP

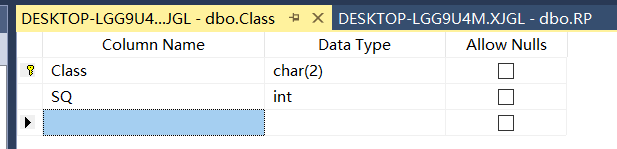


Fig.5 The Design of Table Class

And their initializations are in from Fig.6 to Fig. 10.

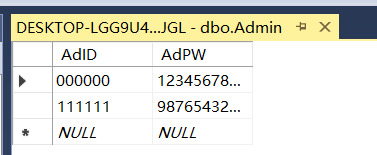


Fig.6 The Initialization of Table Admin

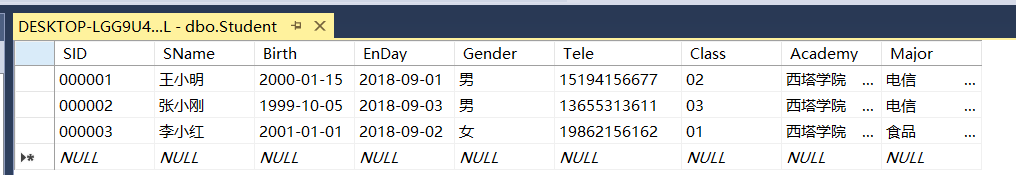


Fig.7 The Initialization of Table Student

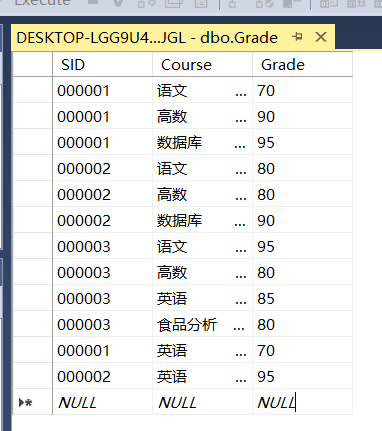


Fig.8 The Initialization of Table Grade

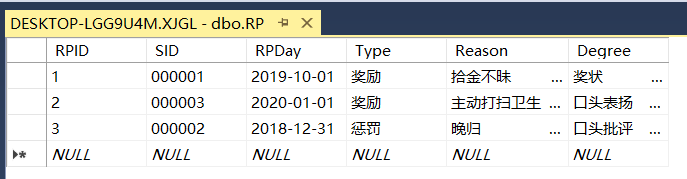


Fig.9 The Initialization of Table RP

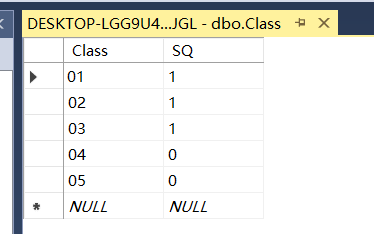


Fig.10 The Initialization of Table Class

Secondly, to set gender only able to be female or male, and type only to be reward or punishment, rules are needed in table Student and RP. SQL to set the rules is in

Fig.11. Besides, both of the two constraints can be checked through SQL Server.

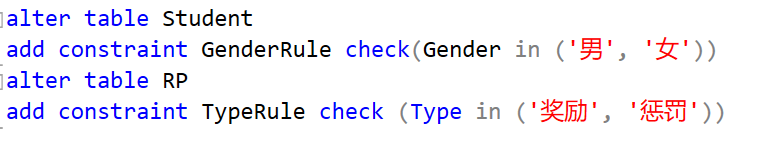


Fig.11 Rule-Setting SQL

Thirdly, since basic information contains too much, a view called SeachView is used to simplify the search process, which contains main information like IDs, names, academies, majors and classes. The design and performance of the view are shown in Fig.12 and Fig.13.

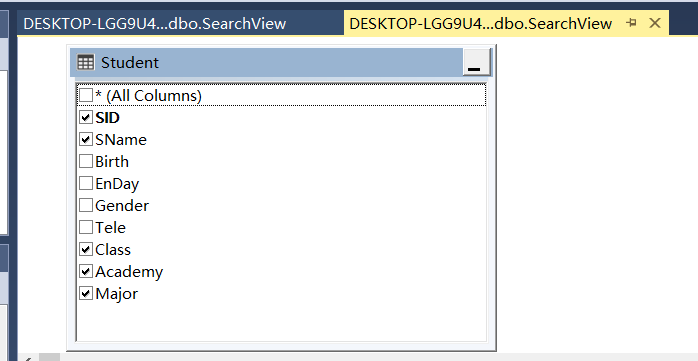


Fig.12 The Design of SearchView

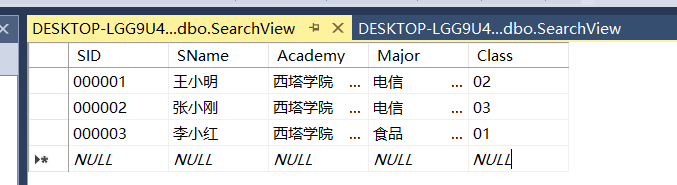


Fig.13 The Performance of SearchView

It is also required that triggers are set to enable the number of students in each class can change automatically according to the change of students’ class information. To satisfy this, since class information of students can only be changed in table Student, triggers can be created on Student after updating, inserting and deleting its information. Fig.14 to Fig.16 illustrate the creation SQL of the triggers for different situations.

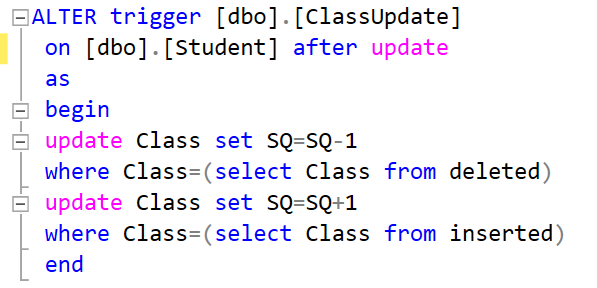


Fig.14 Creation SQL of Update Trigger

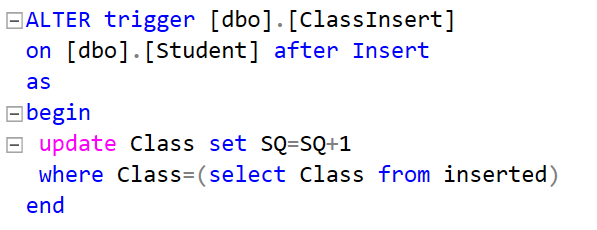


Fig.15 Creation SQL of Insertion Trigger

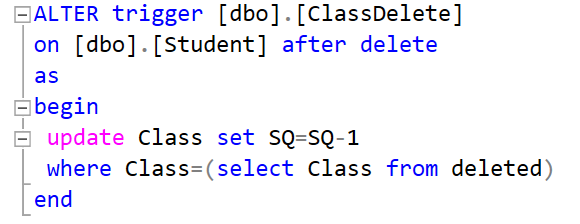


Fig.16 Creation SQL of Deletion Trigger

The last demand requires referential integrity constraints for all the tables. In this database, for example, a student not in table Student, which means he or she does not exist, cannot have grades or reward and punishment records. So, constraints that connect their students’ IDs are set respectively like Fig.17 and Fig.18. Specifically, SID is main key for table Student but external key for table Grade and RP and they are attached by the constraints.

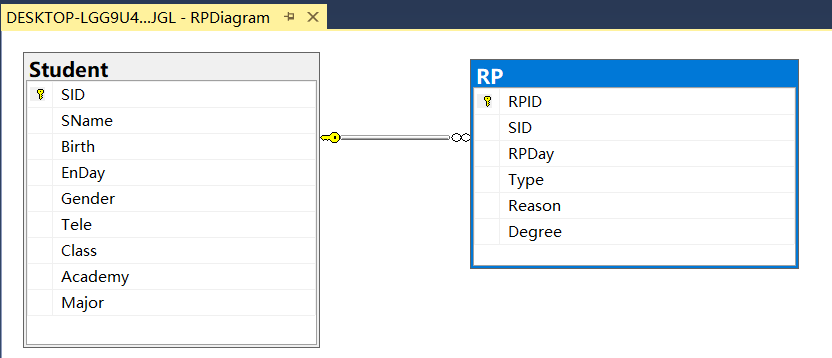


Fig.17 Referential Integrity Constraint for RP

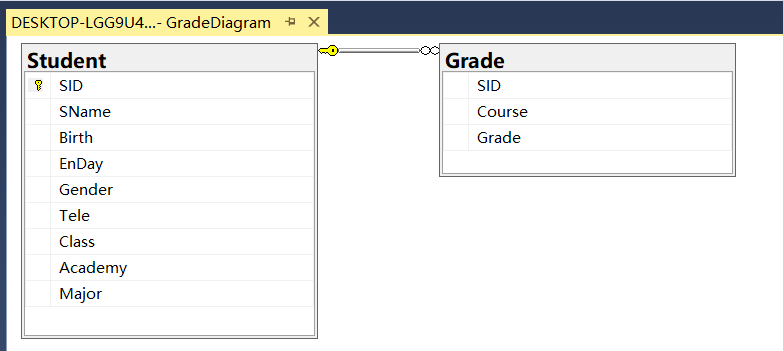


Fig.18 Referential Integrity Constraint for Grade

### Java Design

Java operates with classes and objects and here I create 7 classes for the whole project. The most basic one is called JdbcUtils, which is used to make connections between java file and database. Meanwhile, it also defines some simple operations to manipulate database. The other classes are all UI, containing methods to deal with data in database.

The first UI is LOG. It receives username and password input by users and compares them with the ones in the database to judge whether the users have the accessibility to the database. If so, the second UI TYPE appears which reminds users to select what operations to do with data. Accessible operations are adding, altering, searching and deleting different kinds of information. After selection, there will be corresponding UI to guide users. ADD enables users to append basic information, students’ grades, and reward and punishment records to database. ALTER offers choices for changing basic information, students’ grades, reward and punishment records and courses’ grades. SEARCH provides with basic information, students’ grades, reward and punishment records, courses’ grades and number of students in classes. DELETE gives deletion for basic information, students’ grades, and reward and punishment records. The detailed operation procedures and principles will be given in System Implementation part.

### Development Environment

Database is created by SQL Server 2018. Java file is built by Eclipse, JDK13, and executable file is generated by exe4j. Importantly, the Initialization.sql file can only be executed on SQL Server 2008 or higher and win10 64bit (Details are in Instructions.docx).

## System Implementation and Users Guide

This part illustrates in detail the principles and code in the program and at the same time, shows the procedure of using this program.

In class JdbcUtils, the username and password of the server are defined as sa and 123456789 to ensure security. So only after the password of sa in current computer is set to 123456789 can server be accessed. Besides, this class also register sql driver and get connection with server automatically when main file is run.

In LOG, as it mentioned above, when the program receives username typed by users, it looks for its corresponding password in the database by using “select AdPw from Admin where AdID=?”. If it’s not accordant with the one users input, the program will reject the login request and give message. If login succeed, TYPE shows.

TYPE is the simplest UI and has no relationship with server. There is a combo box, giving options for operation selection. After selection, the connection with database is made again and ADD, ALTER, DELETE or SEARCH will appear.

In ADD, first, select the kind of information to be appended and press Confirm Selection button, a table with one blank row and columns of the corresponding kind of information will be shown below. In this way, users are supposed to fill the blank row with the data to be added and press Confirm Addition button. Once the second button is pressed, sql “insert into ? values (?,?,?)” is triggered and data is sent to database. However, if the data input does not conform to the rules of the database, it will be rejected. For example, most data cannot be null and gender can only be male or female. Finally, interface returns to TYPE for other more operations.

In ALTER, similarly, choose the type of information to be altered, type the keyword to be searched in the text field and press Confirm button, and a table including the objected data will be shown below by “selecting \* from ? where ?=?”. Users can directly change information of the table. After operation, press Confirm Change button and every piece of the data will be compared with its original contents. If they are different, it means changes happened on that piece of data and “update ? set ?=? where ?=?” will be triggered to do the same alteration to the database. A message will tell users whether or not the alternation is successful. Finally, interface returns to TYPE for other more operations.

As for SEARCH, its interface is almost the same as ALTER, but uses view SearchView instead of table Student, and does not offer users the right to change the data shown (The data after change will not be sent back to database). Besides, Confirm Change button is replaced with Back button, by pressing which users can return to TYPE.

The last interface is DELETE. Choose which variety of information to be deleted and press Confirm Search button, and then all the information of that variety will be displayed by “selecting \* from ?”. Here users can look through all statistics and type the key whose information is about to be removed into text field below. Push down the Confirm Deletion button and “delete from ? where ?=?” will be used to accomplish the deletion. A message will tell users whether or not the removal is successful. Finally, interface returns to TYPE for other more operations.

## Program Testing and Evaluation

### Program Testing

Uncompress Program.zip, run Management System.exe and the login screen is displayed as Fig.19.

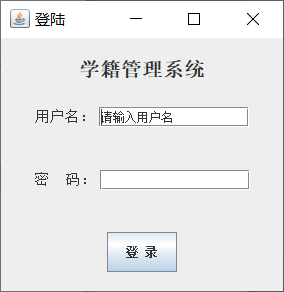


Fig.19 Login Screen

Enter a correct set of username and password to log in and users can select operation types as Fig.20.

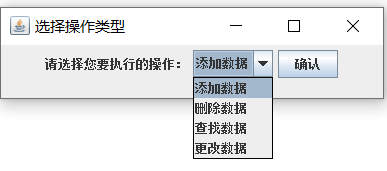


Fig.20 Operation Selection

From Fig.21 to Fig.24 respectively show addition, alternation, search and deletion operation.

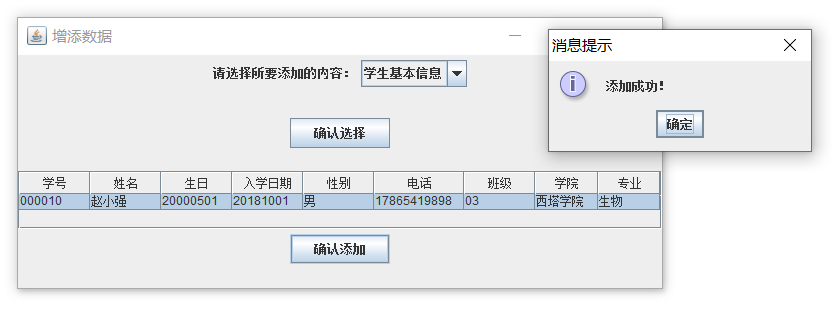


Fig.21 Addition Operation

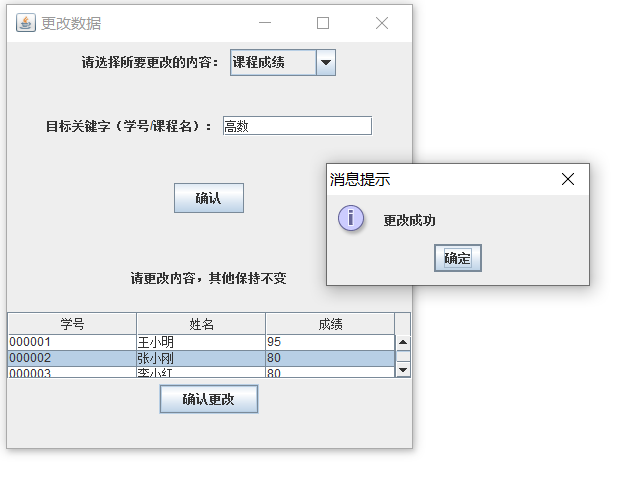


Fig.22 Alternation Operation



Fig.23 Search Operation

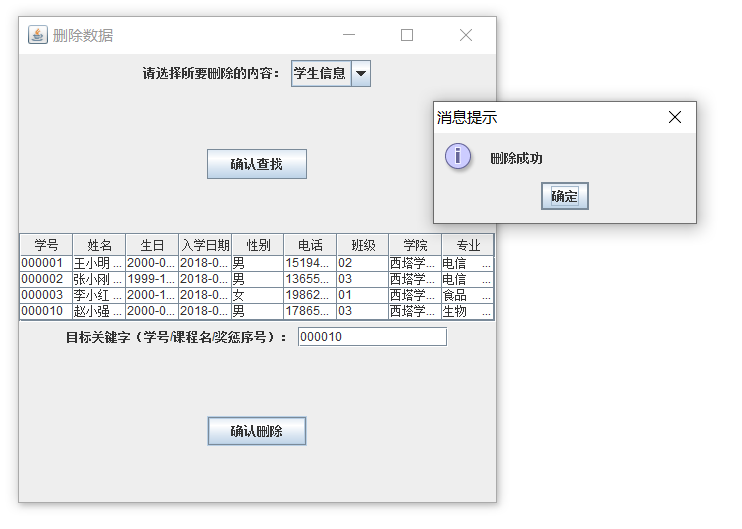


Fig.24 Deletion Operation

### Program Evaluation

Check the database after each step and find that it was changed rightly based on the main file. Details can be seen in Instructions.mkv.

Generally, the program achieves all requirements given, all functions needed and offers clear guidance for users. However, to get the latest information in database, main file is configured to disconnect and reconnect with database after every operation. As a result, it will occupy some space and slow down other applications in a tiny degree.

Besides, since main file is compiled with its assisting files, which contains some unnecessary jre, it needs around 200MB for storage.

## Conclusion

By doing this project, I not only fully reviewed the theoretical contents learned in this semester's database, such as referential integrity constraints, rules, views and so on, but also learned how to apply them in practice. At the same time, I also learned a lot of new knowledge, including triggers and others. More importantly, although it was a simple program, combining Java with database enabled me to master the most basic implementation principles of the local database management system and lay a solid foundation for possible more advanced operations in the future.

However, there are still many deficiencies in the program and the process of this project, mainly including the following problems. 1. The main program interface is not nice enough. When the interface is small, less data can be displayed at the same time, while when the interface is large, there will be too much blank space, which will affect impressions. 2. Operation is limited. For example, in delete and add operations, only one message can be deleted or added at one time. If you want to add more than one piece of information, you can only do it multiple times. 3. Narrow application space. Because it is a simple operation of the local database, to run the program, prepared SQL statement file to create a test database must first be executed on the local computer with SQL Server and modify specific permission settings. Therefore, only program using network database can make the management system have practical application value. 4. Not trying new software. For databases, there are many programs that allow users to manipulate them, such as Visual C++, python, net and so on. But because of their high learning costs, I chose the simple Java I had learned before and didn't get out of my comfort zone. 5. Still not familiar with the writing of an experimental report. Can't divide parts well, can't utilize resources on the Internet efficiently, can't elaborate and accurately state my ideas.

Where there is a will there is a way. Although there is still much trouble in coding, theoretical understanding and report writing at this stage, the difficulties that can be overcome are not problems. For database courses, the most important thing is to do more comprehensive experiments, even small projects, and seriously write experiment reports every time, summarize the problems and improve methods, and then put efforts into the next practice. Over time, integrative ability will be greatly improved, and the utilization of database can be mastered thoroughly.

## Appendix

Main file: Management System.exe

Assisting file: jre (needn’t be run)

Preparation file: Initialization.sql (run before main file)

Explanation file: Instructions.docx and Instructions.mkv

JAVA file: CJGL (source code, needn’t be run)