Designing database services for the convenience of students living in dormitories and the efficient work of managers

21101926 김소은

ABSTRACT

According to the atmosphere of living with COVID-19, the number of people who accommodate in the dormitory is increasing. In accordance with this trend, this paper designed and implemented a service system that can efficiently support the work of managers as well as functions for student. First, the proposed system for students is an environment where students can receive packages, apply for repairs, check reward points at any times. Second, the proposed system for managers is an environment where managers can update and check packages arrival records, repair completion records, reward points, and good student data for work efficiency.

Introduction

Due to COVID-19, there were many dorms that accommodated fewer people or did not operate at all. However, as the influence of COVID-19 has decreased, many high schools and universities currently operates dormitories, and the number of dormitories accommodated has also increased. For this reason, as the number of students using dormitories increases, it has become difficult to manage students efficiently in the existing way from the perspective of managing them. Along with these management difficulties, students also experience inconvenience. That's why dormitory support services are needed.

The composition of this paper is as follows. Chapter 2 is an extension of ‘Introduction’ so it explains the motivation to create a new service through a simple comparison, and Chapter 3 explains how the data structure and er model were designed in earnest, and how the database and program were connected. Chapter 4 shows the code for program execution and the actual execution, and I will conclude.

Motivation

Now I will explain the design plan of the dormitory service that I want to create by comparing the dormitory service of Seoul National University of Science and Technology with the new service I want to design. The new service can inform the student that the package has arrived, so it can manage the package by preventing it from accumulating or losing it, and students can also receive it comfortably. On the other hand, existing services do not implement any practical convenience services such as delivery services for students. Second, the reward and penalty point system. In the new service design, the manager can directly grant a store or penalty point to the student for efficient management at any time, but the existing service is not. Finally, it is the facility repair part. The service requesting repairs to the facility is the same, but in the case of the existing service, it does not tell whether the repair has been completed. However, the new dormitory service informs students that it was completed, so this system is more convenient for both managers and students. In addition, the basic rule for managing the building is different. In the case of existing service, there is one manager in the entire building, but the newly designed service has managers on each floor considering the larger size of the dormitory.

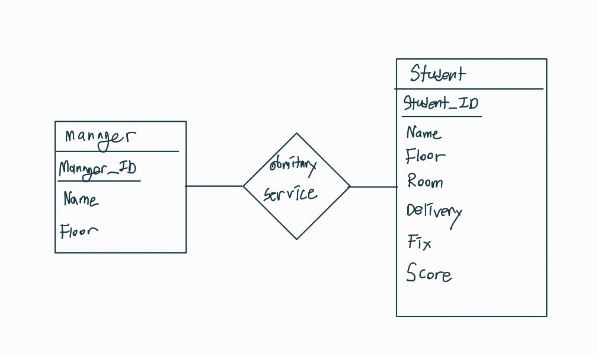
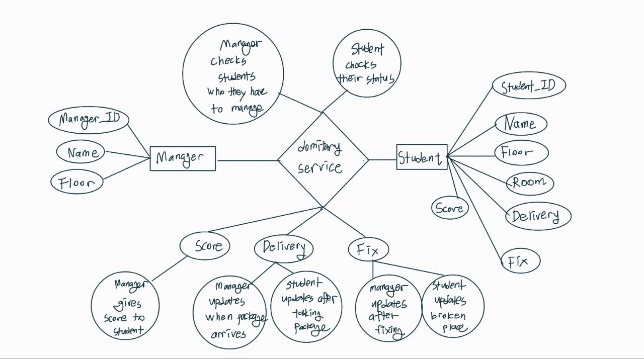
Design and implementation

1) How I designed the data structure

From now on, I will explain how I designed the data structure. First, a total of three attributes were designed to create a table that can contain manager information. The first attribute is set to 'Primary key' to distinguish managers with ‘Manager\_ID’, and the data type is set to ‘interger' because the employee number is numeric data. The second attribute is "Name", which means the administrator's name and the name can be the same, so it can be duplicated, and it is set to "text" type because it is text data. The last attribute is 'Floor', which stores the layers that each administrator needs to manage, and it is also numeric data, so it is set to 'interger' type.

Next, a total of 7 attributes were designed to create a table that can contain student information. The first attribute was set to 'primary key' to distinguish students with 'Student\_ID', and the data type was set to 'interger' because the class number is numerical data. The second attribute is 'Name', which means the student's name and the name can be the same, so it can be duplicated, and it is set to 'text' type because it is text data. The third attribute is ‘Floor’, which means the floor of the room where the student lives, and is also set to the ‘interger’ type. The fourth attribute is 'Room', which is the exact number of the room where the student lives, and the 'interger' type. The fifth attribute is 'Delivery', which is an attribute related to delivery. It is set to "text" type because it is displayed as ‘nothing’ when nothing happens, ‘new\_arrive’ when the package arrives, and ‘take’ when the student receives it. The sixth attribute is ‘Fix’, which is related to repair, and it is also set to ‘text’ type because it is displayed as ‘nothing’ when nothing happens, and as the place name where student wants to repair and as ‘fixed’ when manager completes the repair. The last attribute is 'Score' related to reward and punishment points, and the manager can give scores that are integers between -9 and +9 points at a time, and the student can check his score at any time, and it is set to 'interger' type.

2) ER model of the database

The ER model for the designed database will be described. The 'Manager' table and the 'Student' table are designed to be connected through an attribute called 'Floor'. This is because the manager should be able to know the information of the students living on the floor he manages and change their status. And in the case of large dormitories, there can be several managers on one floor, so it is a 'one to many' or 'many to many' structure. The following image is a simple designed ER model and a more specifically expressed ER model (many to many).  


3) Connection to database

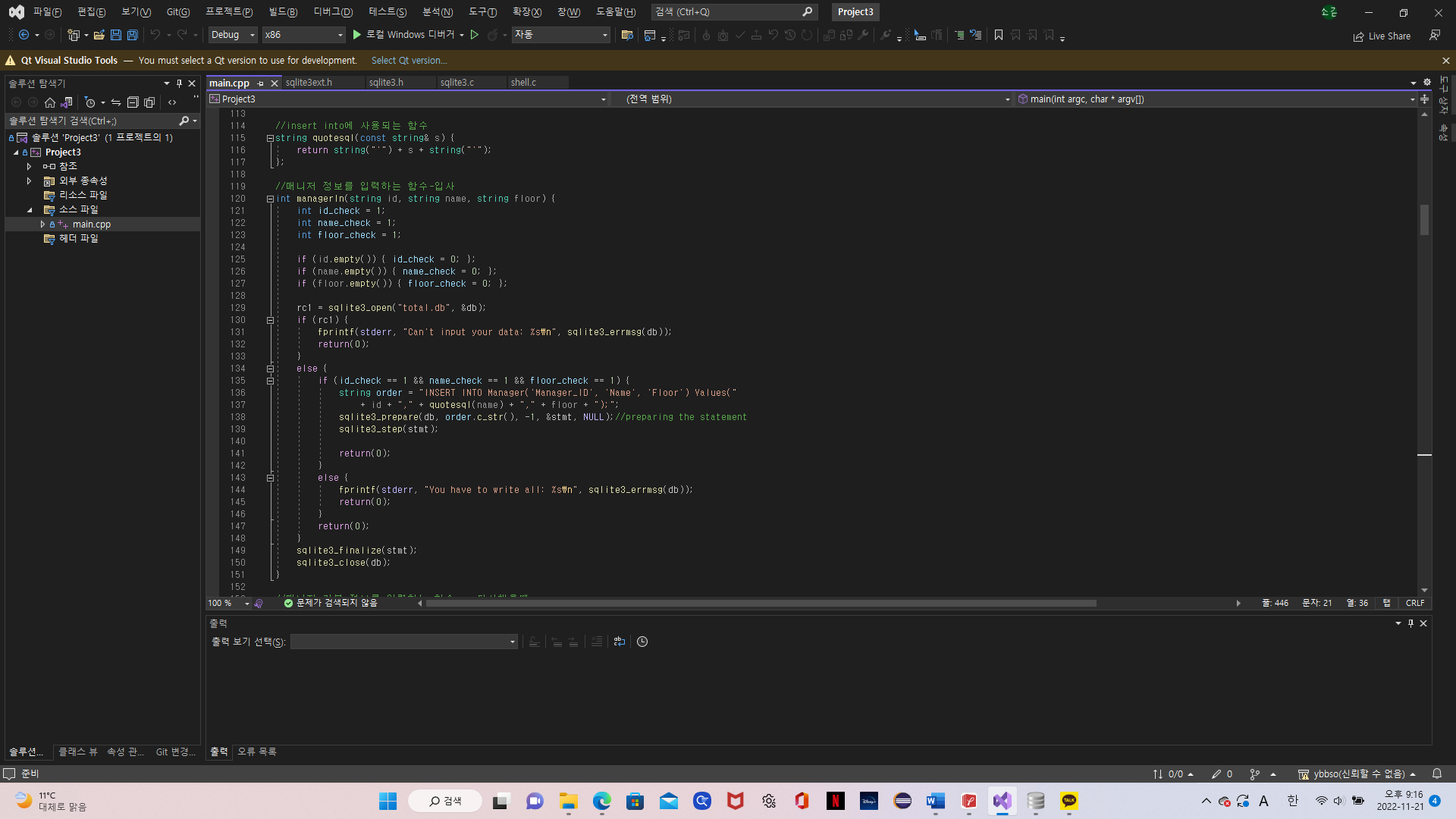
I will explain how the database was implemented. The languages used in the implementation are c++ and sqlite3. Largely, table generation and basic setting, table modification function, and information retrieval function were created and implemented. First, 'create table' was used for creating ‘Manager’, ‘Student’ table, and when a new column was created in the Student table, a function was created to add the basic values described above using 'update'.

텍스트, 스크린샷, 모니터, 전자기기이(가) 표시된 사진

자동 생성된 설명텍스트, 스크린샷, 모니터, 컴퓨터이(가) 표시된 사진

자동 생성된 설명텍스트, 모니터, 스크린샷, 검은색이(가) 표시된 사진

자동 생성된 설명

Next, there are four manager functions and three student functions in terms of table modification function. Looking at the manager function first, the first is the manager registration and manager deletion functions. If you newly join the company, register basic information (manager\_ID, name, floor) using 'insert', and if you leave the company, delete the information in the manager database using 'delete'. Second, it is a package arrival input function. When the package arrives, the administrator uses ‘update’ to change the status of the student's ‘Delivery’ to ‘new\_arrive’. Third, it is a repair completion input function. When the repair requested by the student is completed, the administrator uses 'update' to change the status of the student's 'Fix' to 'fixed'. Finally, it is a reward point input function. The administrator uses 'update' to add or subtract numbers within the range of -9 to +9 from the student's 'Score' in the Student table. Now, looking at the student function, first, the student registration function and the student deletion function. If you newly enter the room, register basic information (student\_ID, name, room) using ‘insert’, and if you leave the room, delete the information in the Student database using ‘delete’. Second, it is a package receipt input function. After receiving the package, the student changes his or her 'Delivery' status to 'take' using 'update'. Finally, it is a repair application function. The student uses 'update' to change the status of his 'Fix' on the Student table to what he wants to request for repair. (The pictures below are the codes for the functions, and the names of the functions for each code are included in the pictures.)텍스트, 모니터, 스크린샷, 화면이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 모니터, 전자기기이(가) 표시된 사진

자동 생성된 설명텍스트, 모니터, 스크린샷, 전자기기이(가) 표시된 사진

자동 생성된 설명

텍스트, 모니터, 스크린샷, 전자기기이(가) 표시된 사진

자동 생성된 설명텍스트, 모니터, 스크린샷, 전자기기이(가) 표시된 사진

자동 생성된 설명

텍스트, 모니터, 스크린샷, 전자기기이(가) 표시된 사진

자동 생성된 설명

Finally, information retrieval function will be described. A total of four functions were designed with two functions in terms of managers, one in terms of students, and one function that brings up the entire data. If we first look at the manager function, first, we implemented a function that retrieves the student information (student\_ID, Name, Room) that the manager needs to manage using the command "SELECT Student\_ID, Student.Name, Room FROM Student, Manager WHERE Student. Floor=Manager.Manager\_ID=Input value". Second, "SELECT Student\_ID, Student.Name, Room, Score FROM Student WHERE Score>= Input value ORDER BY Score DESC;" is used to implement a function that allows Score to recall student information with a specific score or higher. Next, looking at the student function, we implemented a function that allows students to check their "Delivery", "Fix", and "Score" status at any time using "SELECT \* FROM Student WHERE Student\_ID=Input value". Finally, a function to retrieve all data in the database was implemented "SELECT Manager\_ID, Manager.Name As Manager\_Name, Manager.Floor, Student\_ID, Student.Name As Student\_Name, Room, Delivery, Fix, Score From Student, Manager WHERE Student=Floor ORDER BY Manager;Floor. (The pictures below are the codes for the functions, and the names of the functions for each code are included in the pictures.)

텍스트, 스크린샷, 모니터, 컴퓨터이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 모니터, 검은색이(가) 표시된 사진

자동 생성된 설명

Evaluation

1) Executing query

The database and the program were connected using the functions described above. And we implemented the program code for real users to use. When a simple input is received to run the program, a table can be created, and then user can enter manager mode or student mode. A total of six functions are available when entering administrator mode, and a total of four functions are available when entering student mode. You can also newly register or delete administrator or student information. The specific code implementing this and what functions it has will be shown through the following image.  
텍스트, 모니터, 스크린샷, 전자기기이(가) 표시된 사진

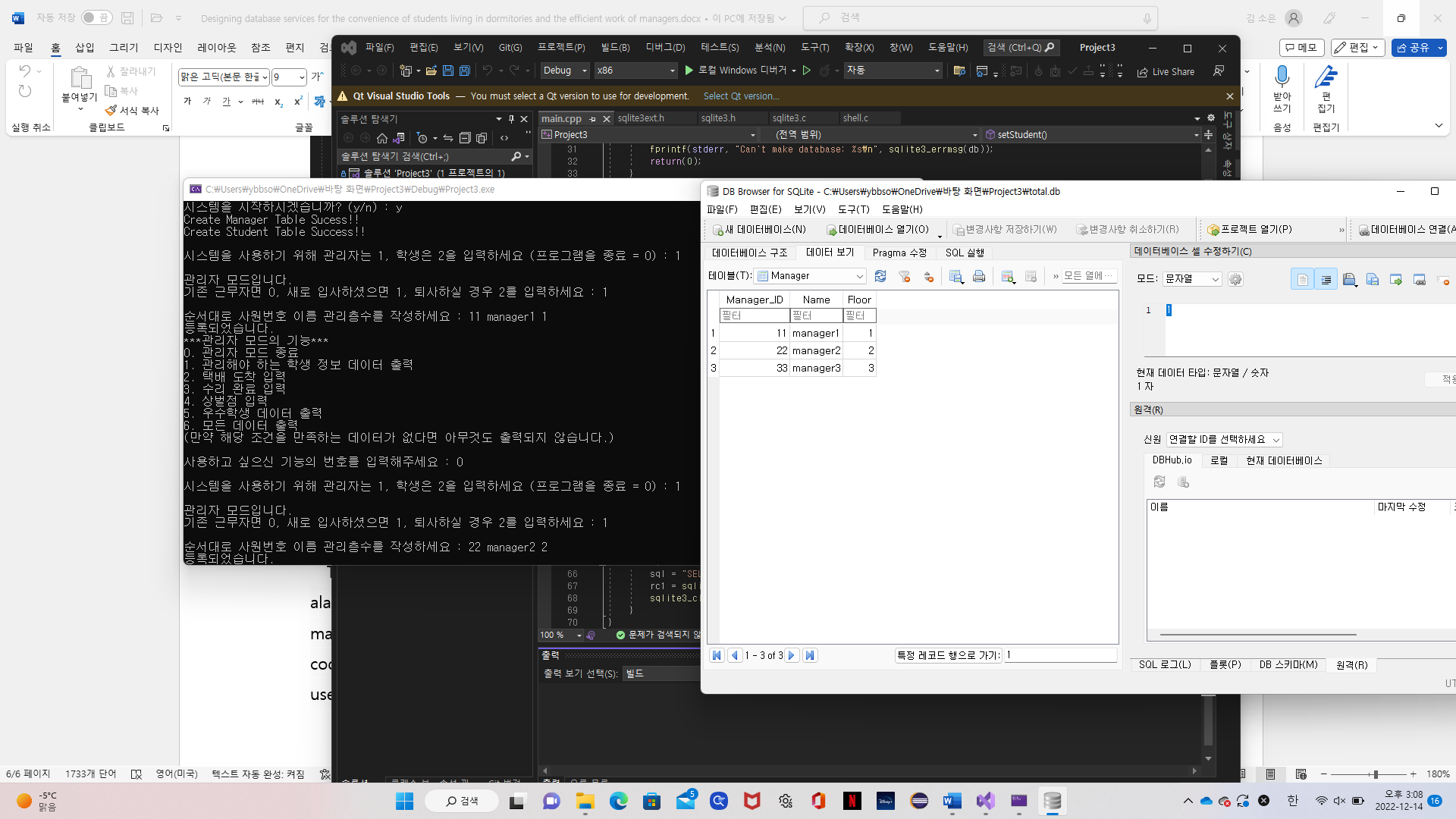
자동 생성된 설명텍스트, 모니터, 스크린샷, 전자기기이(가) 표시된 사진

자동 생성된 설명  
  
텍스트, 스크린샷, 모니터, 전자기기이(가) 표시된 사진

자동 생성된 설명텍스트, 모니터, 스크린샷, 전자기기이(가) 표시된 사진

자동 생성된 설명

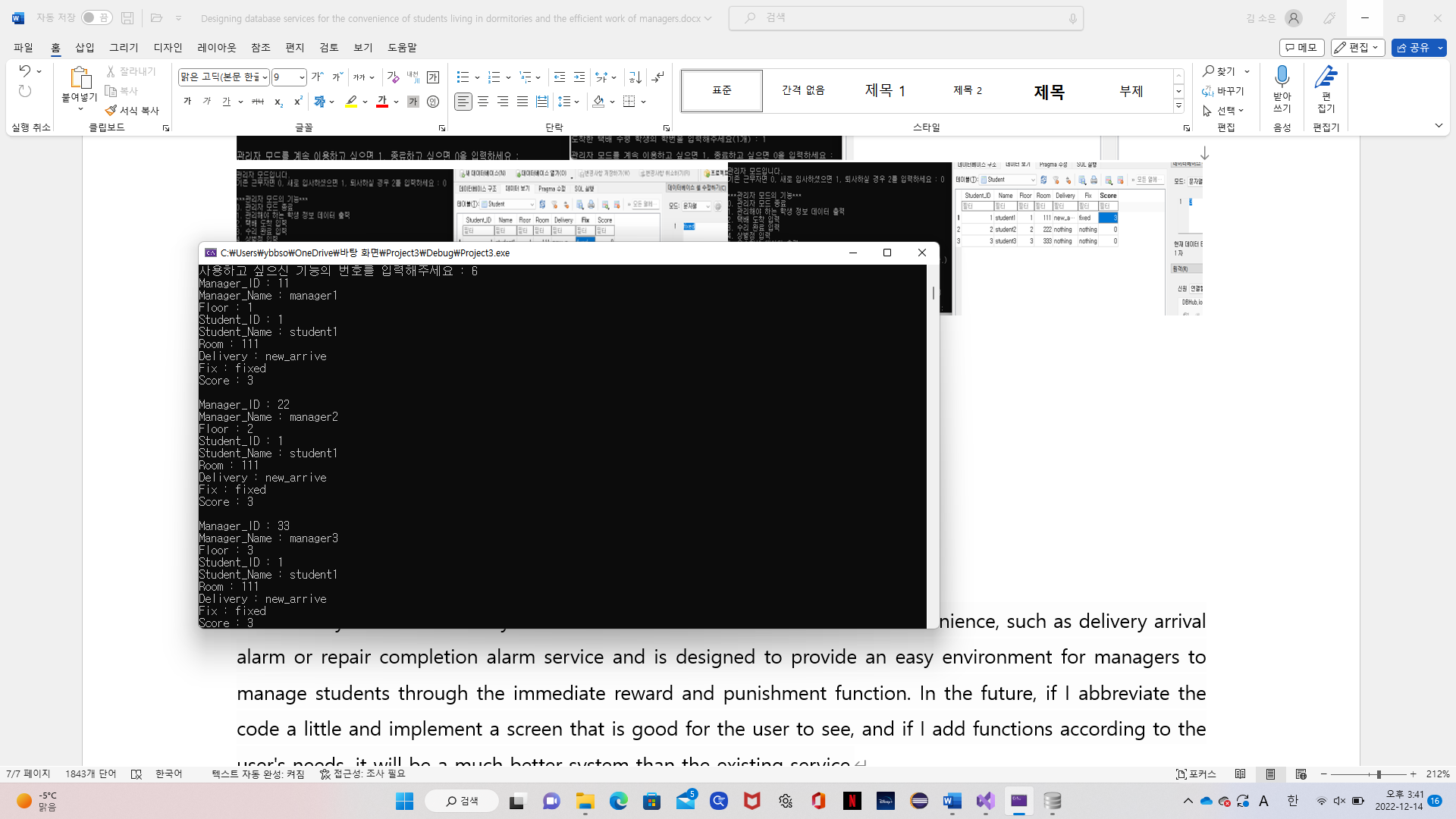
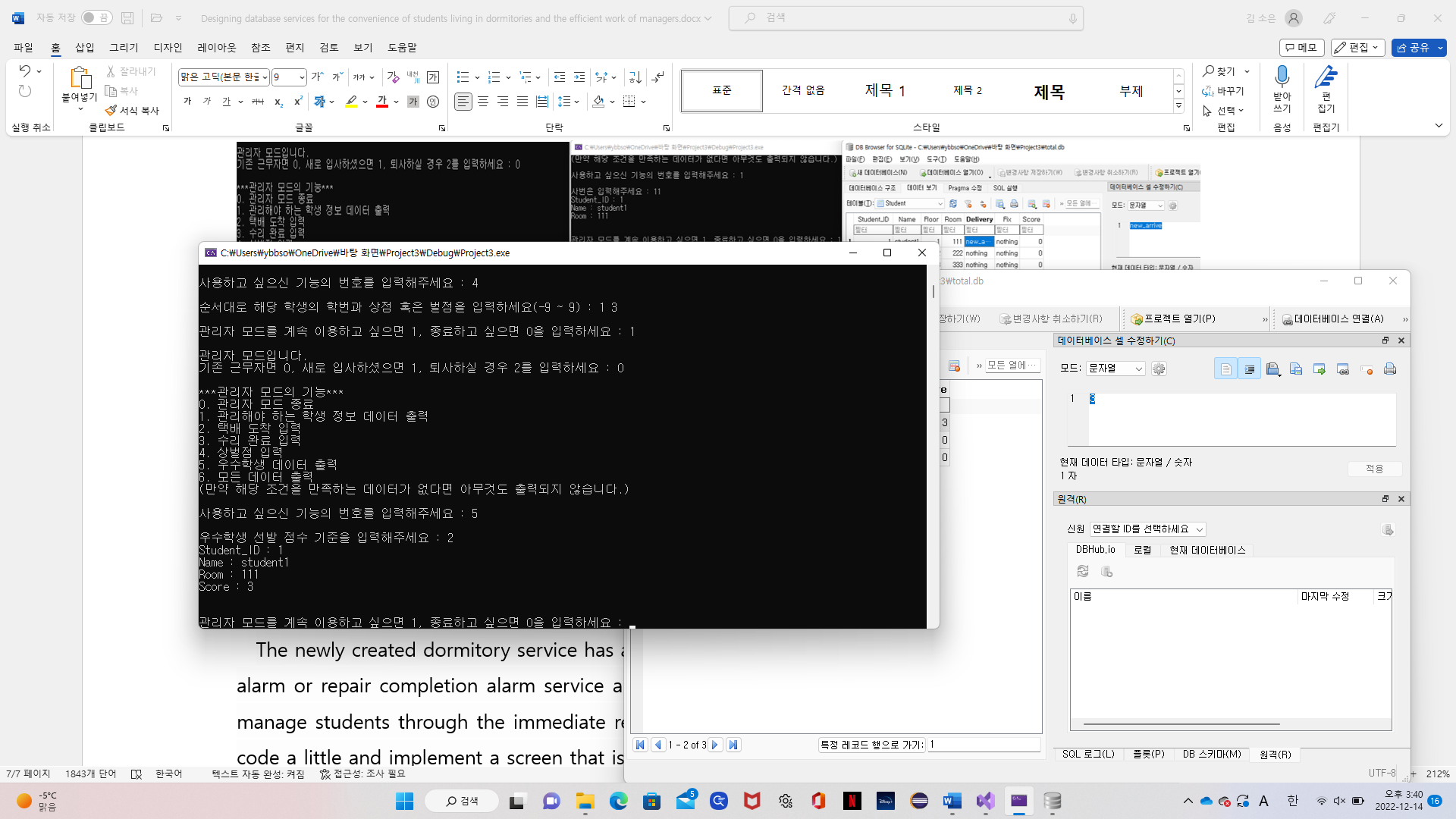
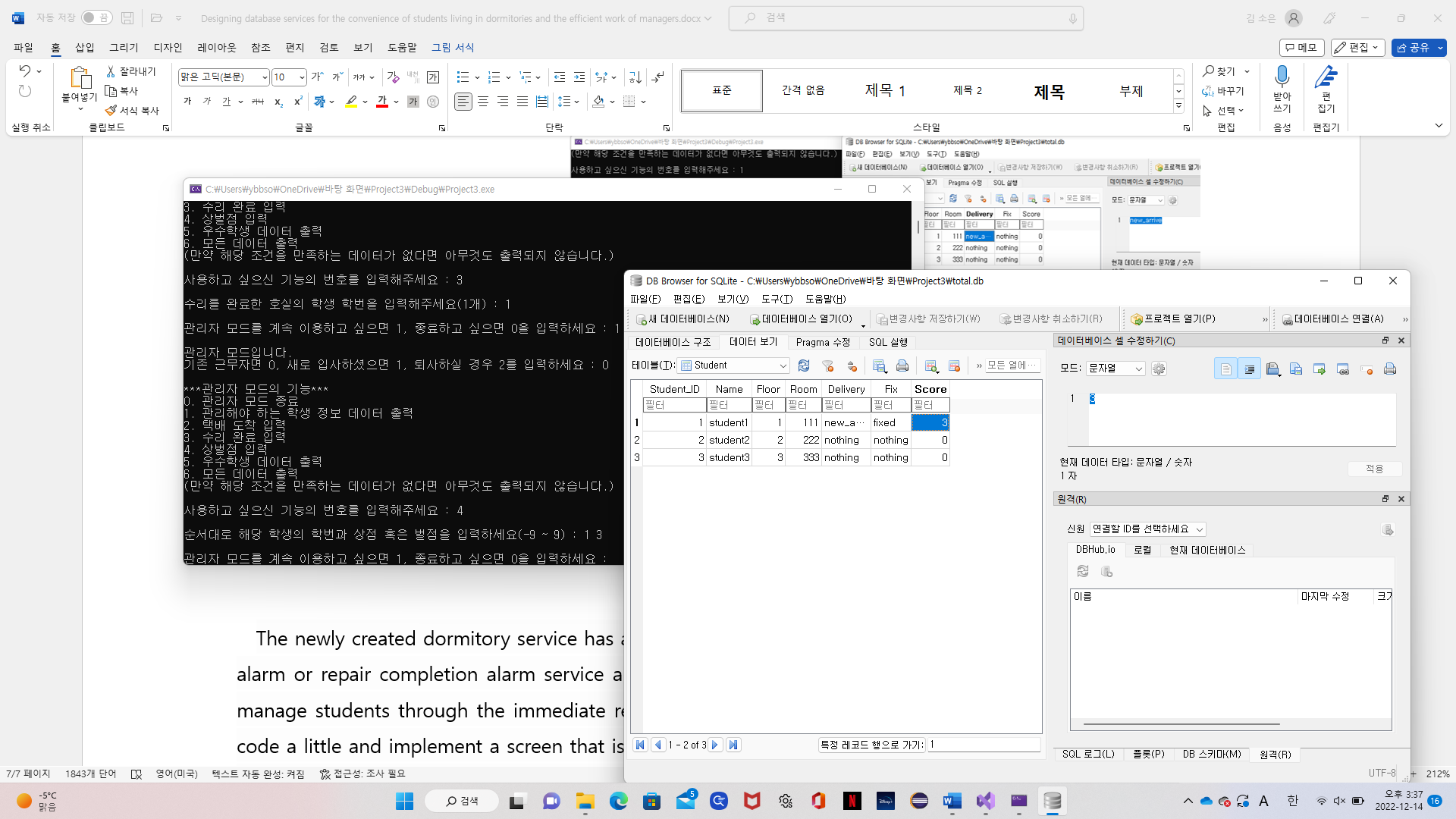
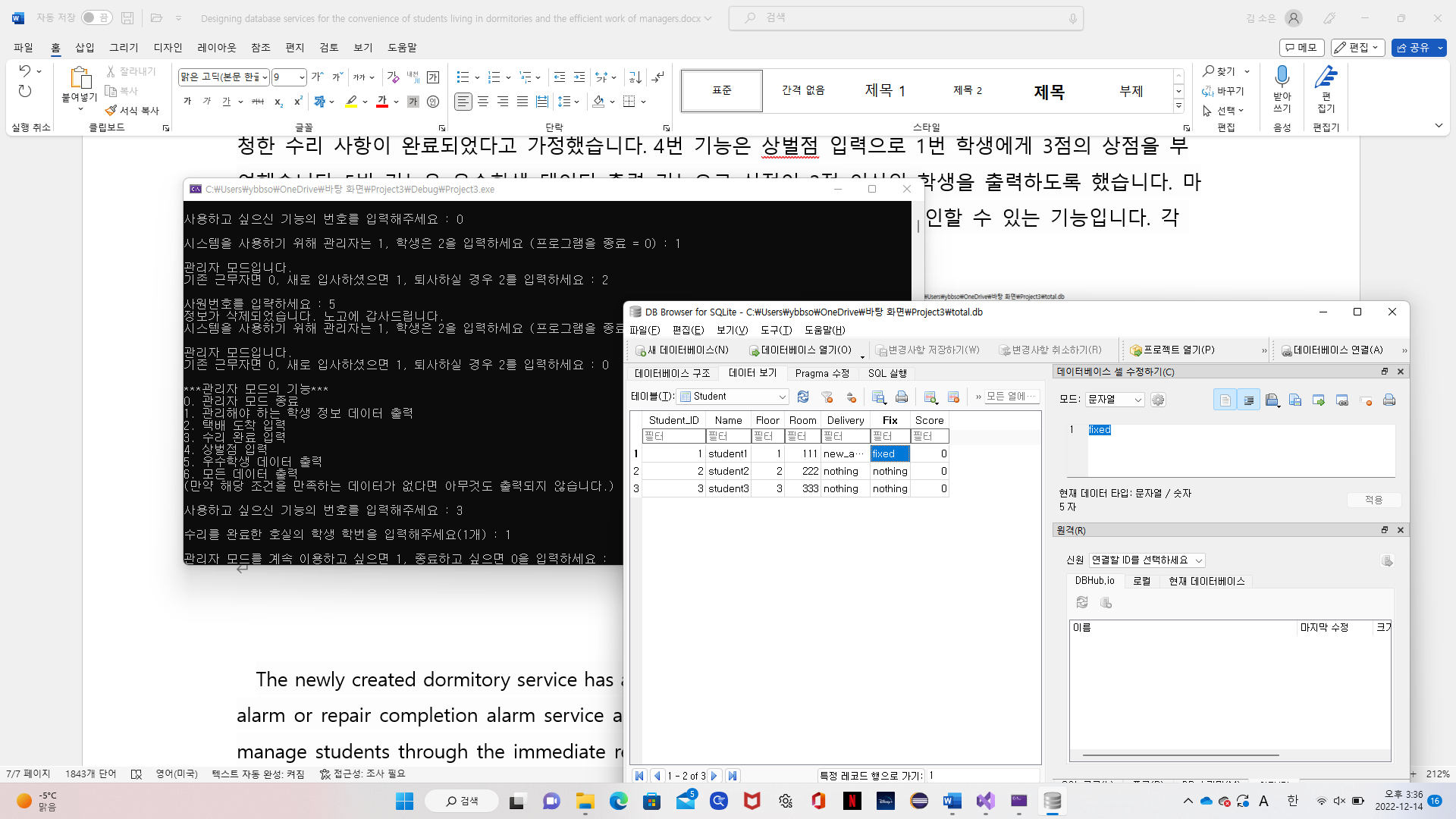
2) Result of executing query  
 I first started the system with tables, creating administrator and student tables, and I added employees with ID 11, 22, and 33 to the administrator. Next, I added a student with ID 1, 2, and 3 to the student. Specific information can be found in the table in the picture.

텍스트이(가) 표시된 사진

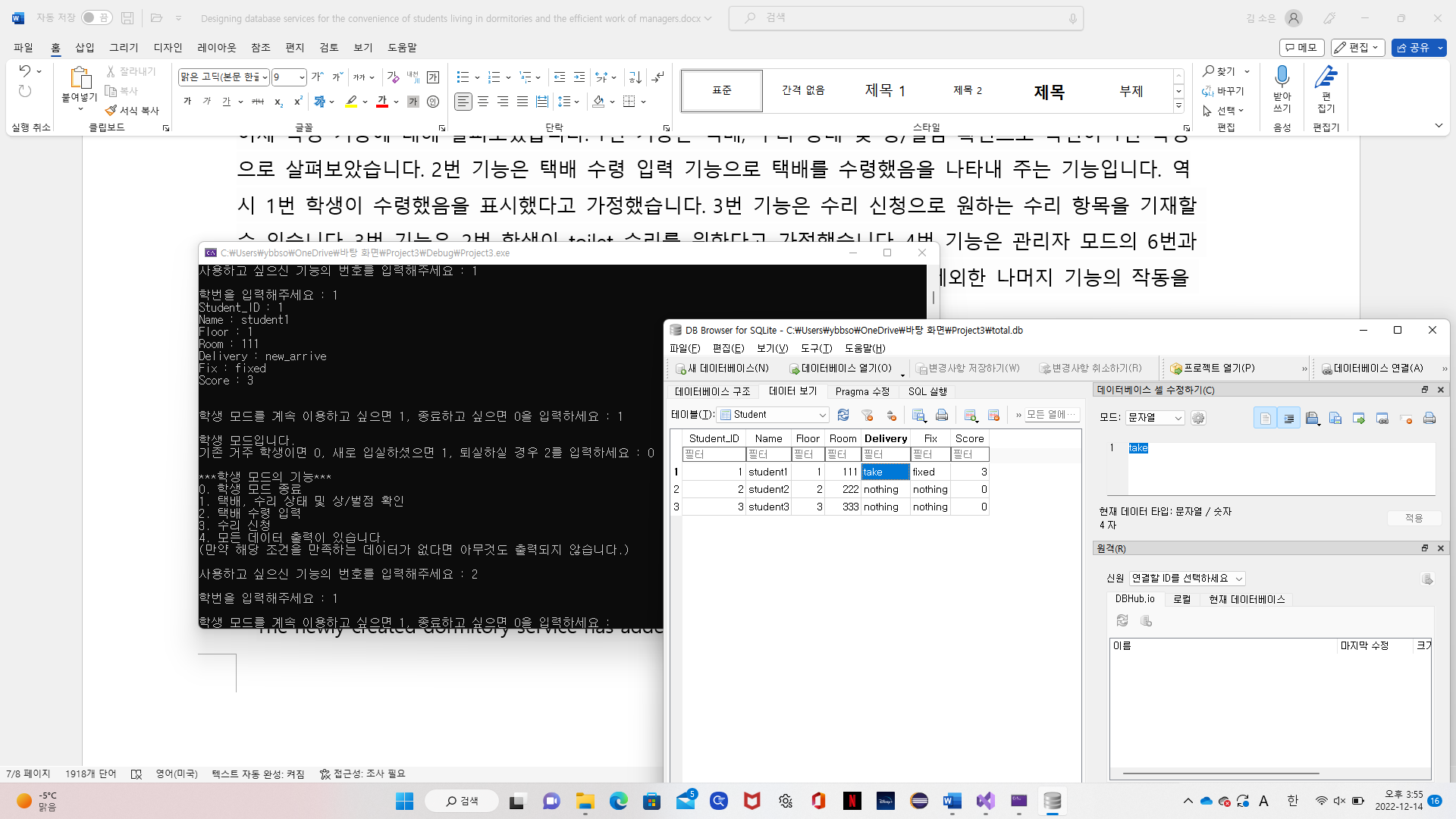
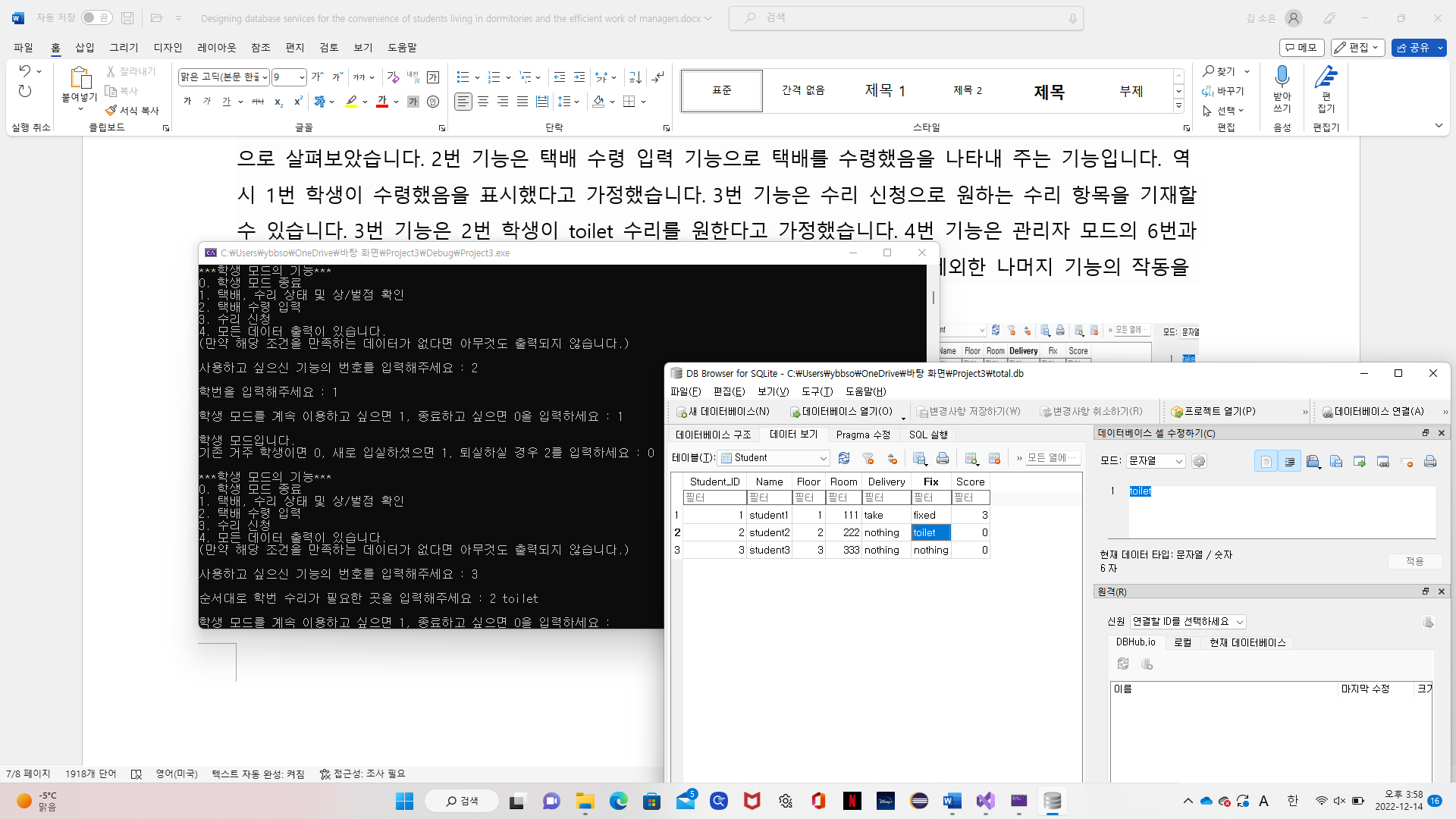
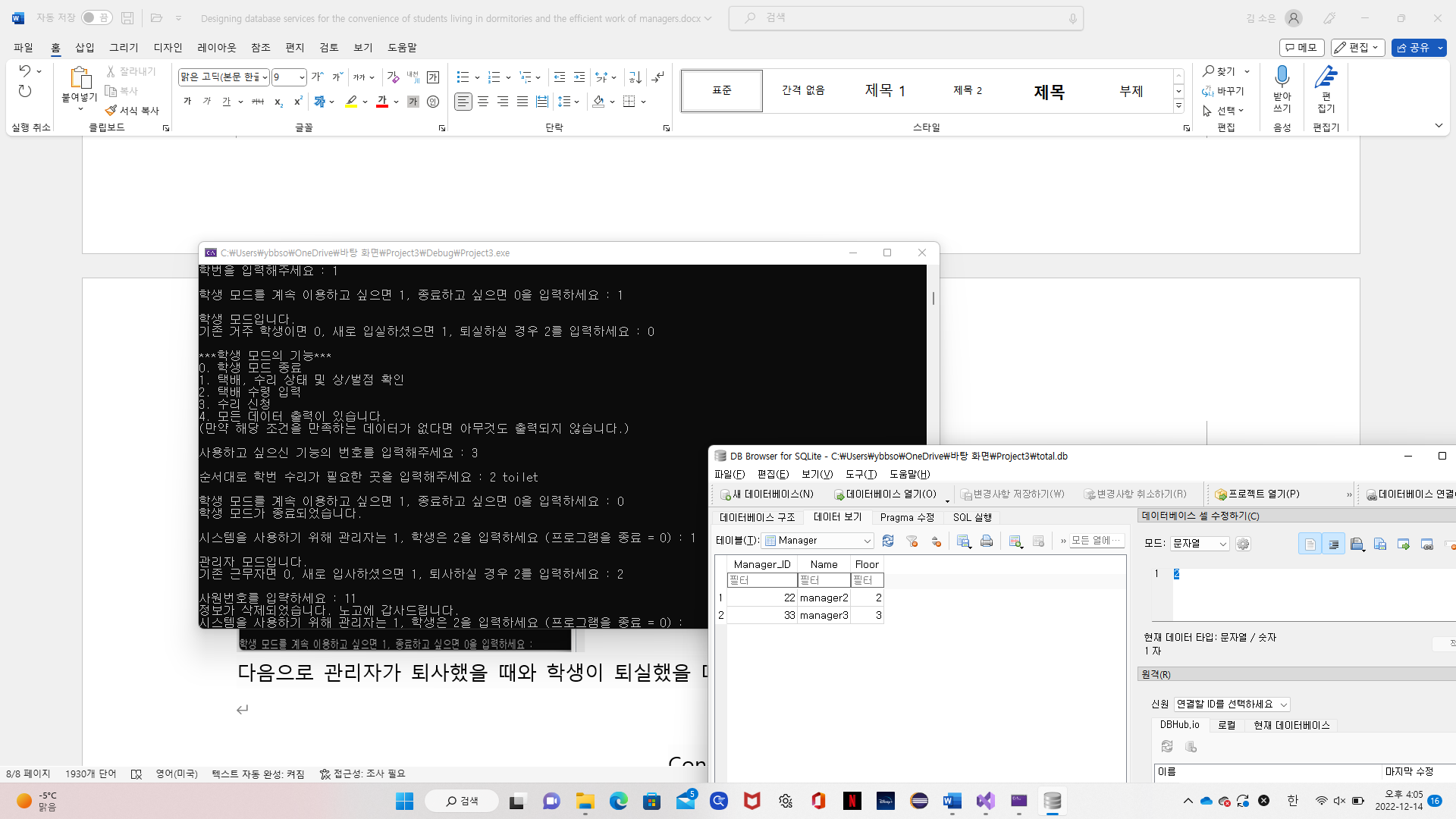
자동 생성된 설명  
 Next, let's look at the administrator functions. Function 1 is a function of outputting student information data that needs to be managed, and has been outputting student information that needs to be managed by manager1 with a company number of 11. Function 2 assumed that the delivery of a student with a single class number arrived by entering the delivery arrival. Function 3 is the input of repair completion, and it is assumed that the repair applied by a single student in the class has been completed. Function No. 4 gave 3 points to student1 by entering reward and penalty points. Function 5 is a data output function for excellent students and requires students with 2 or more points to print out. Finally, number 6 is the ability to see all the information in the database with all the data output. Here is the implementation view for each feature.

텍스트, 스크린샷, 컴퓨터이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명  


Now let's look at student functions. Function 1 was examined as a single student in the class by checking the delivery, repair status, and prize/penalty points. Function #2 is a package receipt input function that indicates that the package has been received. It is assumed that student1 indicated that he had received package. Function 3 allows you to enter the desired repair item by requesting a repair. Function 3 assumes student2 wants toilet repair. Function 4 outputs all data the same as number 6 in Administrator mode. The operation of the remaining functions except for function 4 shown above will be shown in the following picture in order.  
텍스트이(가) 표시된 사진

자동 생성된 설명   
  
Next is the action of erasing your information from the database when the administrator leaves the office and when the student leaves the dormitary.  
  


Conclusion

The newly created dormitory service has added functions for students' convenience, such as delivery arrival alarm or repair completion alarm service and is designed to provide an easy environment for managers to manage students through the immediate reward and punishment function. In the future, if I abbreviate the code a little and implement a screen that is good for the user to see, and if I add functions according to the user's needs, it will be a much better system than the existing service.