**ASSIGNMENT 1 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **TEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | **Unit 04: Database Design & Development** | | |
| **Submission date** | February 18, 2022 | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Le Minh Duc | **Student ID** | GCD210433 |
| **Class** | GCD1001 | **Assessor name** | Do Duy Thao |
| **Student declaration**  I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice. | | | |
|  |  | **Student’s signature** | Duc |

**Grading grid**

|  |  |  |
| --- | --- | --- |
| P1 | M1 | D1 |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **❒ Summative Feedback: ❒ Resubmission Feedback:** | | |
| **Grade:** | **Assessor Signature:** | **Date:** |
| **Signature & Date:** | | |

Contents

[Chapter 1 - Statements of user and system requirements 4](#_Toc96207712)

[1.1 Overview about the problem 4](#_Toc96207713)

[1.2 Requirements of the application 4](#_Toc96207714)

[Chapter 2 – Design the relational database system 5](#_Toc96207715)

[2.1. Database definition 5](#_Toc96207716)

[2.2. Requirement analysis 7](#_Toc96207717)

[2.3. Database design with explanations 7](#_Toc96207718)

[Bill table 11](#_Toc96207719)

[Data type: 15](#_Toc96207720)

[2.4. Review whether the database is normalized 15](#_Toc96207721)

[1. Normalization definition (Thi, n.d.) 15](#_Toc96207722)

[2. My normalized database 16](#_Toc96207723)

[References 18](#_Toc96207724)

[Figure 1. Database 6](#_Toc96207725)

[Figure 2. Logical design of database 8](#_Toc96207726)

[Figure 3. Student table 9](#_Toc96207727)

[Figure 4. Room table 9](#_Toc96207728)

[Figure 5. RoomOrder table 10](#_Toc96207729)

[Figure 6. Service table 10](#_Toc96207730)

[Figure 7. Bill table 11](#_Toc96207731)

[Figure 8. Student-RoomOrder relationship 12](#_Toc96207732)

[Figure 9. Room-RoomOrder relationship 13](#_Toc96207733)

[Figure 10. Room-Service relationship 14](#_Toc96207734)

[Figure 11. Room-Bill relationship 15](#_Toc96207735)

# Chapter 1 - Statements of user and system requirements

## Overview about the problem

In this assignment, I am supposed to be an employment as a Database Developer for a large IT consultancy company. Due to the unexpected problem in managing the outcome of students in the rooms the dormitory, I am assigned a tasked that designing a database allowing manager to take control of the expenses of every room in dormitory.

## Requirements of the application

The dormitory has two user roles named manager to take control the spends and students to register, which means most of functional commands are dependent on the manager of that dormitory and some of them are dependent on the others.

As a manager of a dormitory:

1. I want to make sure that the database allows me to know the personal information of students who register for the rooms so that I can easily manage the students’ status.
2. I want the rooms also have some features for the management to have a quick check the status of every room such as empty room or damaged facilities, …
3. I want to know the expenses of students in the room, especially water and electric spend based on their figure.
4. I want to know the total money that students spent in each room so that I can moderate.

As a student in a dormitory:

1. I want to register in a suitable room so that I can have a good living environment.
2. I want to know the expenses of the room used by me and other students so that an eligible moderation can be executed.

As for the abilities of the database, they are Create/ Read/ Update/ Delete (CRUD) which are the basic element for the database. The function of each command can be displayed below:

* Create: Create students register form leading to the room order.
* Read: Initialize and read all the student information or the expenses of every room.
* Update: The student information as well as their order and room’s attributes have to be casually checked.
* Delete: All extra information should be removed like the monthly bills of every room.

# Chapter 2 – Design the relational database system

Before starting with the requirements, the definition of database system should be made clearly and specifically.

## Database definition

First of all, Data can refer to elements about anything under examination. Your name, age, gender, occupation, and other personal information are examples of data about you. Data may also be a picture, a song and so on.



Figure . Database

Therefore, a database is a collection of data that has been organized in a systematic manner. They allow data to be contained and executed. Data handling is simplified thanks to databases.

The databases have so many types in fact. In my design, the relational database is considered using due to its unique attributes. The relational database uses interrelated tables to determine the database relationship, which is also known as Relational DBMS. Examples of that systems is Microsoft SQL Server, which is used to create a perfect database. (Peterson, 2022)

## Requirement analysis

The requirements are displayed based on the demand of users. There are 2 user roles as I mentioned before so I mainly focus on two user roles.

Regarding to the manager:

* The student’s personal information should be clear and can not be duplicated in all kind of situation.
* The room should be regularly well-checked, which means an update can be done casually.
* The monthly outcome named water and electric expenses in the room based on student’s consumption should be displayed and so does the cost
* Finally, the total money on bill should be made clearly so that the manager probably has a monthly statistic.

Regarding to the students:

* The priority in selecting room is allowed.
* The expenses are also be clear and so does the bill.

According to the research, there are some problematic circumstances which can arise during the process:

1. Students have registered the wanted room but changed their mind and switched to another room.
2. The room booking of every student is now dependent on the attributes of the room.
3. The order date of student can be incorrect with the real time they lived in dormitory.
4. The total money foreach room can be issued such as swap ore information lost.

Therefore, I need to consider all elements to have a overall look to cover all cases that can happened during the statistic process in order to improve the quality of managing database of the dormitory.

## 2.3. Database design with explanations

To apply all the requirements, my database is designed below:

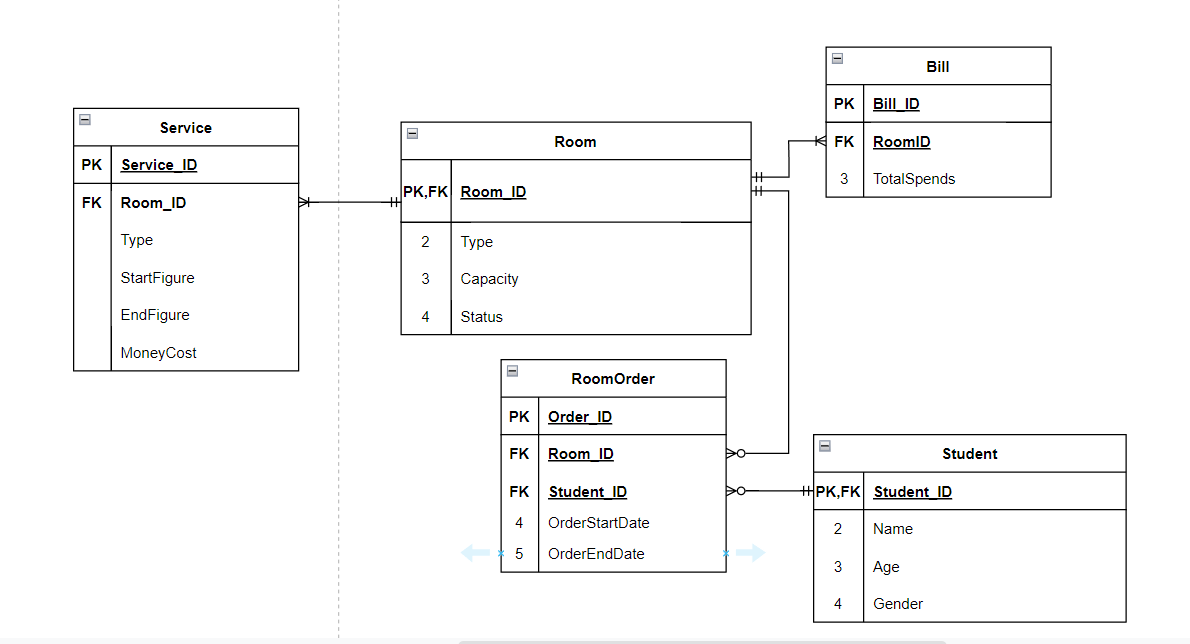


Figure . Logical design of database

Have a quick look, my design has total 5 relationship tables connected together. The Room tables is located in the middle to connect another 3 tables. Each table has their own attributes and storage all needed information.

* Student table:

In this table, the personal information is store based on 4 different variables with the Student\_ID as a primary key. Following that, the name, age and gender of student also are listed in the tables so that the manager can have a basic look at the student who want to register in the dormitory.

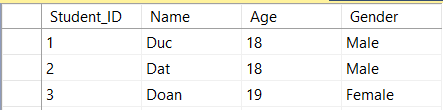


Figure . Student table

* Room table

Like student table, room table also contain the properties of the room including Room\_ID as a primary key, type and capacity and status as candidate keys. That properties are displayed for both the manager and students can have a look so that the room can be checked and updated.

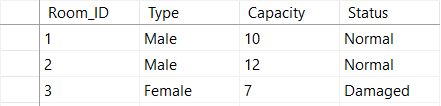


Figure . Room table

* RoomOrder table

The purpose of creating RoomOrder table is allowing the two table above are connected together in a right way. It has a variable name Order\_ID as a primary key while the other two variables named Student\_ID and Room\_ID as foreign key which create relationship in logical design. Besides, the last two variables named OrderStartDate and OrderEndDate are also stored to make sure the register-finish day of each student is noted.

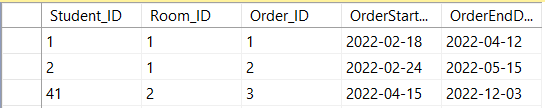


Figure . RoomOrder table

* Service table

The service table should contain the expenses of all students in the specific room. While the Service\_ID is considered as a primary key in this table, other variables which are seen as candidate keys are the useful information to check. Specially, this table is connected with the Room table so the variable named Room\_ID is a foreign key of this table.

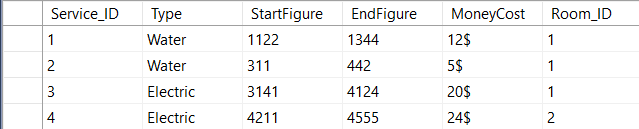


Figure . Service table

* Bill table

Finally, the bill table allows users to know the total cost of the services students used in a month. Hence, the bill is connected to each room, which mean the Room\_ID located in this table is a foreign key. Also, Bill\_ID is a primary key in this table and the Totalspend element should display the total money spent.

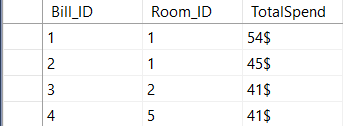


Figure . Bill table

### Relationship explanation:

1. Relationship between Student table and RoomOrder table:

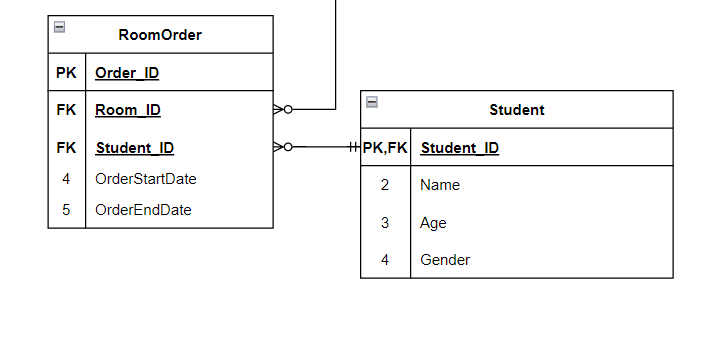


Figure . Student-RoomOrder relationship

Have a quick look to the screenshot, the Student table connects with RoomOrder table based on one-to-many relationship. Because each student has right to choose a room they want to live so that there are many RoomOrder can be called by a student. Finally, the most suitable order is written in the database so one-to-many relationship is the best choice for this relationship.

1. Relationship between Room table and RoomOrder table:

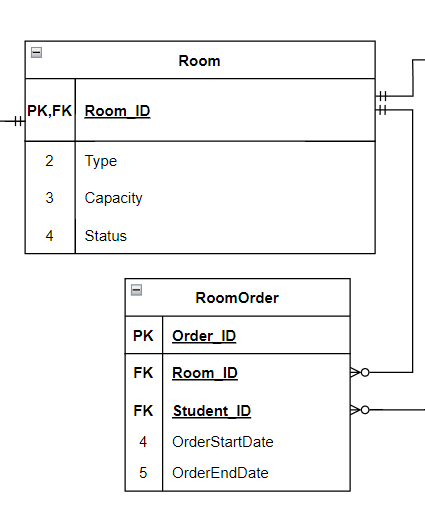


Figure . Room-RoomOrder relationship

Similar to Student table, the Room table also have one-to-many relationship with the RoomOrder table. The reason for this can be explained like this: a room can have many orders and in return, only one order is suitable for one room.

1. Relationship between Room table and Service table:

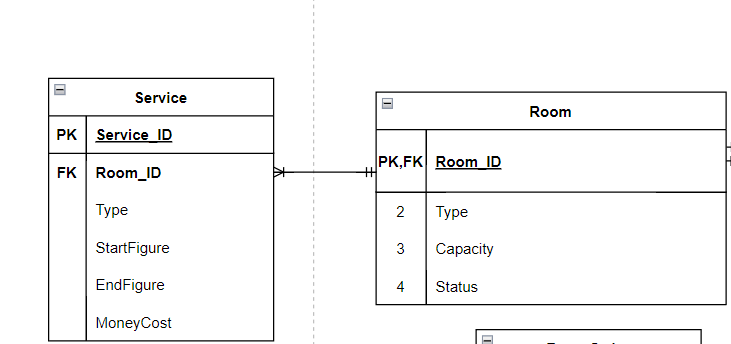


Figure . Room-Service relationship

It is obvious that a room can have many services and each room service is applied in just a room. More clearly, we can not have the expenses of this room refer to another room. Thus, one-to-many is settled between two tables.

1. Relationship between Room table and Bill table:

Same explanation with service table, because the bill is charged monthly so the room can have many bills, 12 bills in a year for example. In conclusion, a one-to-many relationship is established between tables.

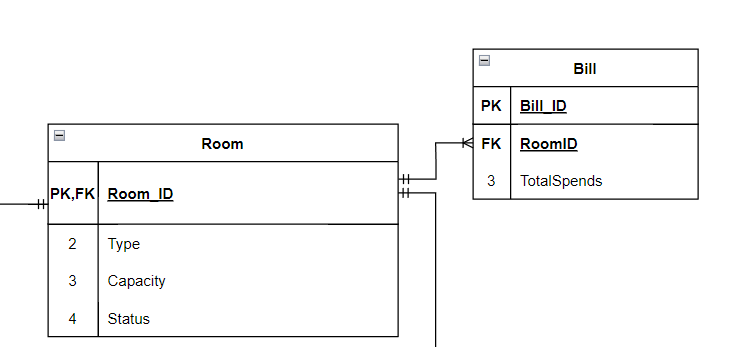


Figure . Room-Bill relationship

### Data type:

* All the IDs should be declared as integer type.
* All the date variables should be declared as date type.
* The other variables should be declared as varchar(50).

## 2.4. Review whether the database is normalized

### 1. Normalization definition (Thi, n.d.)

Normalization is a database design approach that minimizes undesired aspects including Insertion, Update, and Deletion Anomalies and lowers data redundancy. Normalization rules break down huge tables into smaller ones and use relationships to link them. Normalization in SQL is used to remove unnecessary data (repetition) and guarantee that data is stored correctly.

Normalization list:

1. 1NF (First Normal Form)
2. 2NF (Second Normal Form)
3. 3NF (Third Normal Form)
4. BCNF (Boyce-Codd Normal Form)
5. 4NF (Fourth Normal Form)
6. 5NF (Fifth Normal Form)
7. 6NF (Sixth Normal Form)

In SQL Server, the Data Normalization philosophy (hence referred to as Data Normalization) is currently being developed. In the 6th Normal Form, for example, there are still debates. Normalization in 3rd Normal Form, on the other hand, is best in most practical situations.

### My normalized database

According to my evaluation, my database is at the 2NF normal form. (Thi, n.d.)

According to 1NF, each table cell should contain only single value and each line should need to be sole. That mean no extra information is allowed in my database design. My design has no unnecessary information so it can pass the 1NF.

According to 2NF, one primary key is just for one variable. To be more detailed, we can’t improve our database if we don’t know how to settle the table based on the one or many single primary keys. Take a look at my design, the table is clearly separated and each record has own primary key in each table. Therefore, my database is normalized at second form.

According to 3NF, there are no connections in functional dependencies. For example, when data in a cell is deleted, the data in other cell should not be deleted due to previous deletion. I realized that my database still has duplicated information. Thus, I can’t pass the 3NF.

## 

# References

Peterson, R., 2022. *Guru99.* [Online]   
Available at: https://www.guru99.com/introduction-to-database-sql.html  
[Accessed 27 January 2022].

Thi, D. K., n.d. [Online]   
Available at: https://www.codelean.vn/2019/12/binh-thuong-hoa-la-gi-1nf-2nf-3nf-bcnf.html