**ASSIGNMENT 1 FRONT SHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification** | **TEC Level 5 HND Diploma in Computing** | | |
| **Unit number and title** | **Unit 04: Database Design & Development** | | |
| **Submission date** |  | **Date Received 1st submission** |  |
| **Re-submission Date** |  | **Date Received 2nd submission** |  |
| **Student Name** | Nguyen Ha Nguyen | **Student ID** | GCH17452 |
| **Class** | GCH0802 | **Assessor name** | Han Minh Phuong |
| **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** | Nguyen  Nguyen Ha Nguyen |

**Grading grid**

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

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

Contents

[I. Introduction 5](#_Toc48910581)

[II. Scenario of the Student Grading system 5](#_Toc48910582)

[III. SRS (System Requirement Specification) 5](#_Toc48910583)

[User requirement 5](#_Toc48910584)

[Student: 5](#_Toc48910585)

[Teacher 5](#_Toc48910586)

[Staff: 6](#_Toc48910587)

[System requirements 6](#_Toc48910588)

[Business rules 6](#_Toc48910589)

[IV. Logical design 7](#_Toc48910590)

[Entity Relational Diagram (ERD) 7](#_Toc48910591)

[V. Explanation 9](#_Toc48910592)

[VI. Physical Design 15](#_Toc48910593)

[Acccout 18](#_Toc48910594)

[Class 18](#_Toc48910595)

[Course 18](#_Toc48910596)

[Result 19](#_Toc48910597)

[Student 19](#_Toc48910598)

[Teacher 20](#_Toc48910599)

[Staff 20](#_Toc48910600)

[VII. Evaluate 20](#_Toc48910601)

# **I.** Introduction

Nowadays, databases are important because organizations are increasingly developing their information processing systems. FPT is currently having difficulty dealing with university management. This report is used to describe the requirements and logical design of the database of the Student Grading System.

# II. Scenario of the Student Grading system

Currently, FPT University does not have an online scoring system. Students must see scores from the report. Teachers send paper grades to the training room, so it's easy to make mistakes and hard to save.

I have to design an online grading system for students to see scores in time. Help Teachers check easily. Store information as easy and undamaged as paper storage systems.

# III. SRS (System Requirement Specification)

## User requirement

### Student:

* + Students can check student scores.
  + Students must register for the class and must attend that class.

### Teacher

* + Input grade of their classes
  + View grades

### Staff:

* + Create courses
  + Create accounts for teacher & students
  + Update student information quickly.
  + Edit grade

### System requirements

* + Store, retrieve and update data.
  + Database recovery facility.
  + Support access from remote locations.
  + Enforce constraints to ensure data in the database complies with certain rules.
  + Student grades will be updated automatically.
  + Students may cancel a course.
  + The repeat score will be replaced with the old grade.

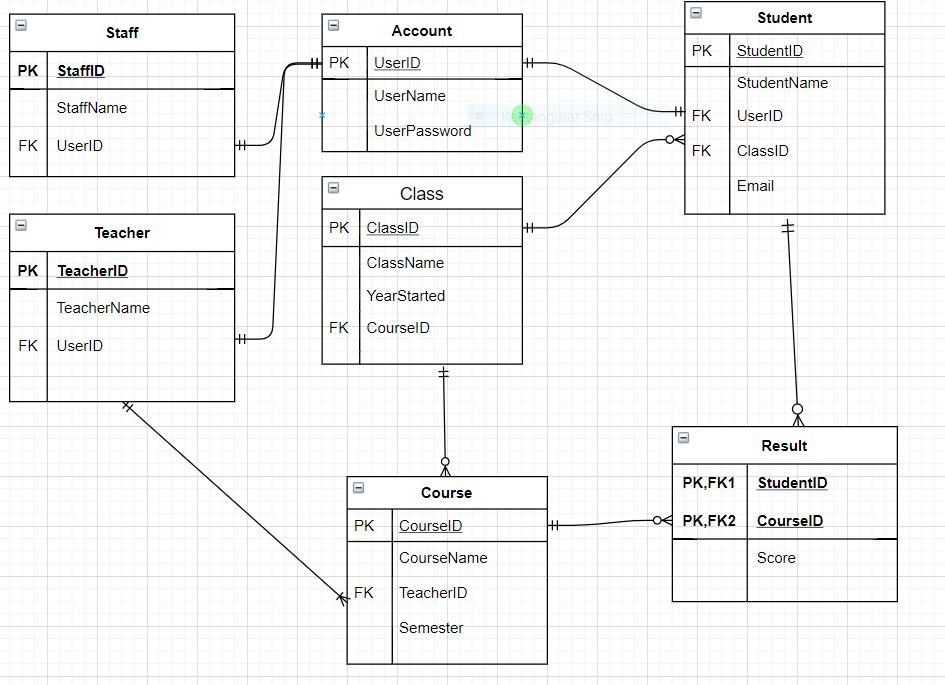
## Business rules

* Before the semester, staff create information about the course such as: course, teacher, class.
* At the end of the course, the teacher enters the score.
* Staff allows teachers to enter scores within 1 week after the end of the course
* Then students see the scores and feedback
* If there is feedback, the teacher will review the lesson and ask the staff to correct the score if any.
* Teachers do not have the right to edit grades after the end of the course 1 week. Only staff can edit scores.
* Score is in the form of a natural number (1-100).

# IV. Logical design

## Entity Relational Diagram (ERD)

The detailed attributes for each entity are described about: Account, Student, Teacher, Class, Staff, Object, Result.

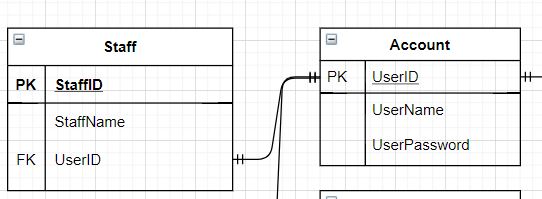


# V. Explanation

Table Staff store information of employees of Training Department:

* StaffID: ID of staff, example: STAFFxxxx

Staff and Account is 1-1 because each employee has only 1 account to log in the system.

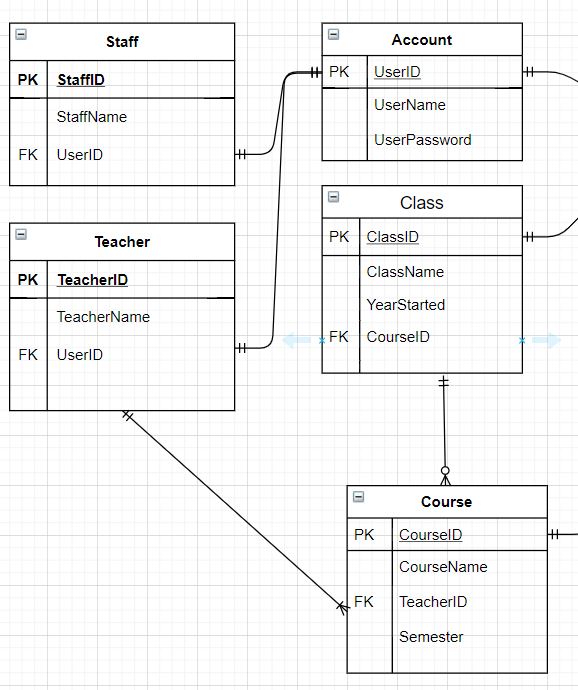


Teacher can teach many course. Teachers also have their own id.

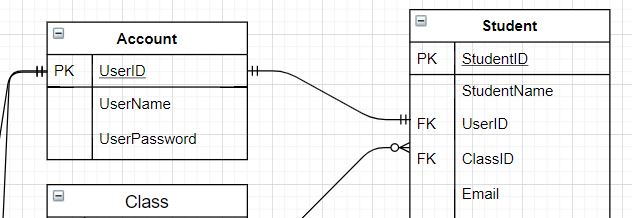
Example: TeacherXXXX

Teacher and Account is 1-1 because each employee has only 1 account to log in the system.

Teacher and Course is 1-Many because thay can teach many course.

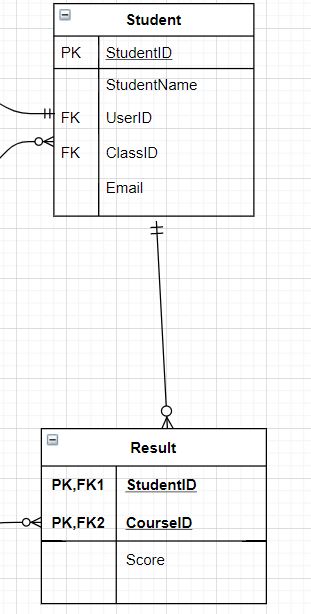


Student and Account is 1-1 because each employee has only 1 account to log in the system



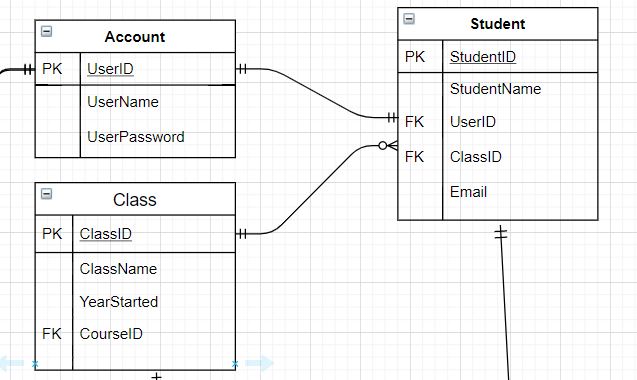
Each student takes many courses, so they can have many results.

Student and Result is 1-Many.

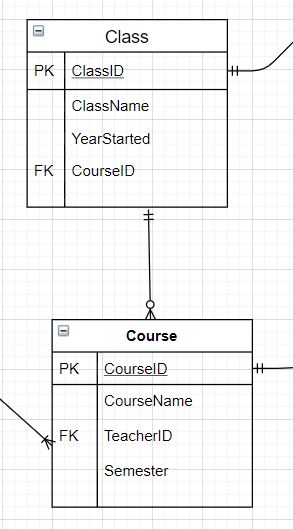


Each class has many students and many course so:

Class and Student is 1-Many.



Class and Course is 1-Many.



# VI. Physical Design

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Table 1. Account | |  | |  | |
|  |  |  |  |  | |  | |
|  |  | Column Name | Data Type | | Constrain | |
|  |  | UserID | int | | PK | |
|  |  | UserName | nvarchar(30) | |  | |
|  |  | UserPassword | varchar(20) | |  | |
| 2. | Table 2. Course | |  | |  | |
|  |  |  |  |  | |  | |
|  |  | Column Name | Data Type | | Constrain | |
|  |  | CourseID | int | | PK | |
|  |  | CourseName | nvarchar(100) | |  | |
|  |  | TeacherID | varchar(10) | |  | |
|  |  | Semester | nvarchar(100) | | FK | |
| 3. | Table 3. Class | |  | |  | |
|  |  |  |  |  | |  | |
|  |  | Column Name |  | Data Type | | Constrain | |
|  |  | ClassID |  | varchar(10) | | PK | |
|  |  | ClassName |  | varchar(10) | |  | |
|  |  | YearStarted |  | date | |  | |
|  |  | CourseID |  | varchar(10) | | FK | |
| 4. | Table 4. Student | |  | |  | |
|  |  |  |  | |  | |
|  |  | Column Name | Data Type | | Constrain | |
|  |  | StudentID | varchar(10) | | PK | |
|  |  | ClassID | varchar(10) | |  | |
|  |  | StudentName | varchar(20) | |  | |
|  | | Email | varchar(300) | | FK | |
|  | | UserID | varchar(10) | | FK | |
| 5. Table 5. Result | | |  | |  | |
|  | |  |  | |  | |
|  | | Column Name | Data Type | | Constrain | |
|  | | StudentID | varchar(10) | | PK,FK1 | |
|  | | CourseID | varchar(10) | | PK,FK2 | |
|  | | Score | int | |  | |

6. Table 6. Teacher

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Column Name | Data Type | Constrain |
|  | TeacherID | nvarchar(20) | PK |
|  | UserID | int | FK |
|  | TeacherName | varchar(20) |  |

7. Table 7. Staff

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | Column Name | Data Type | Constrain |
|  | StaffID | nvarchar(20) | PK |
|  | UserID | int | FK |
|  | StaffName | varchar(20) |  |

As show above, the designed database is conducted by 7 table

### Acccout

|  |  |  |
| --- | --- | --- |
| UserID | UserName | Userpassword |
| 123 | Nguyen Ha Nguyen | 123456 |

### Class

|  |  |  |  |
| --- | --- | --- | --- |
| ClassID | ClassName | YearStarted | CourseID |
| R1 | Gch17432 | 08-06-2020 | 1618 |

### Course

|  |  |  |  |
| --- | --- | --- | --- |
| CourseID | CourseName | TeacherID | Semester |
| 1618 | Programing | TEACHER1 | Fall20 |

### Result

|  |  |  |
| --- | --- | --- |
| StudentID | CourseID | Mark |
| 17452 | 1618 | 65 |
| 17468 | 1619 | 70 |
| 17457 | 1620 | 75 |
| 17678 | 1622 | 60 |

### Student

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| StudentID | ClassID | StudentName | UserID | Email |
| 17432 | R1 | John Cena | 147 | [Cena17432@gmail.com](mailto:Cena17432@gmail.com) |
| 17414 | R2 | David Wakada | 148 | [Wakada17414@gmail.com](mailto:Wakada17414@gmail.com) |
| 17452 | R1 | John Wick | 149 | [Wick17452@gmail.com](mailto:Wick17452@gmail.com) |
| 17324 | R2 | Ha Nguyen | 157 | [Nguyen17324@gmail.com](mailto:Nguyen17324@gmail.com) |
| 17476 | R1 | David Jones | 158 | [Jones17476@gmail.com](mailto:Jones17476@gmail.com) |
| 17674 | R2 | Nguyen Du | 159 | [Du17674@gmail.com](mailto:Du17674@gmail.com) |

### Teacher

|  |  |  |
| --- | --- | --- |
| StaffID | StaffName | UserID |
| 1 | Nguyen Ngoc Han | 123 |

### Staff

|  |  |  |
| --- | --- | --- |
| TeacherID | StaffName | UserID |
| 159 | Vuong Thi Nhung | 888 |

# VII. Evaluate

The benefit of managing data with a database is that it restricts data duplication, leading to redundancy of data that the manager cannot control manually. Stored data formats ensure synchronization and maintain the integrity of data. At the same time, using the database will create many advantages for sharing to other team members, anyone can easily access and access the data to serve the work, help solve. solve problems that arise quickly.