

***Assignment:***

***Database Design and Development***

***Name: Nguyen Duc Thang***

***ID: GCH17549***

***Class: GCH0707***

***Gmail: thangndgch17549@fpt.edu.vn***

***Leature: Pham Thuy Duong***

Contents

[***I.* *Introduction* 3**](#_Toc11135722)

[**II. System Requirements 3**](#_Toc11135723)

[***III.* *ERD Design* 4**](#_Toc11135724)

[**1. ERD 4**](#_Toc11135725)

[**2. The role of each table 4**](#_Toc11135726)

[***IV.* *Relationship between entities* 7**](#_Toc11135727)

[**1. Contact between Account table and Student table 7**](#_Toc11135728)

[**2. Contact between Student table and Attendance table 8**](#_Toc11135729)

[**3. Contact between Attendance table and Course table 8**](#_Toc11135730)

[**4. Contact between Class table and Schedule 9**](#_Toc11135731)

[**5. Contact between Class table and Tutor table 9**](#_Toc11135732)

[**6. Contact between Class table and Student table 10**](#_Toc11135733)

[***V.* *Physical design* 11**](#_Toc11135734)

[***VI.* *Summary* 15**](#_Toc11135735)

# ***Introduction***

I was recruited as a database developer for a large IT consulting company. The company has been approached by FPT University is expanding due to the growth in the number of students. FPT University is currently facing difficulties in dealing with university management. It decided to develop some academic systems for easier university management, including: Online library system, Student grading system, Attendance system, CMS system, Planning system Planning, Admissions System, etc.

A database is a software that is very important for managing a system in a company, or a business. So the report below will describe how to design a database system for school about "Attendance". The following will describes the relationships between tables together suitable for a database system by tool draw.io. The database is a systematic data library with the main purpose of storing and analyzing data. Database makes it easier for users to find and manage information and data. Moreover, the database is managed by a system to avoid applications that are detrimental to it.

# **System Requirements**

To determine the requirements of the system, we need to clearly define the role of the system in place. Those roles are required objects and requirements set for that object. The requirements must be appropriate and reasonable with the system. Below are requirements for attendance system for students in class:

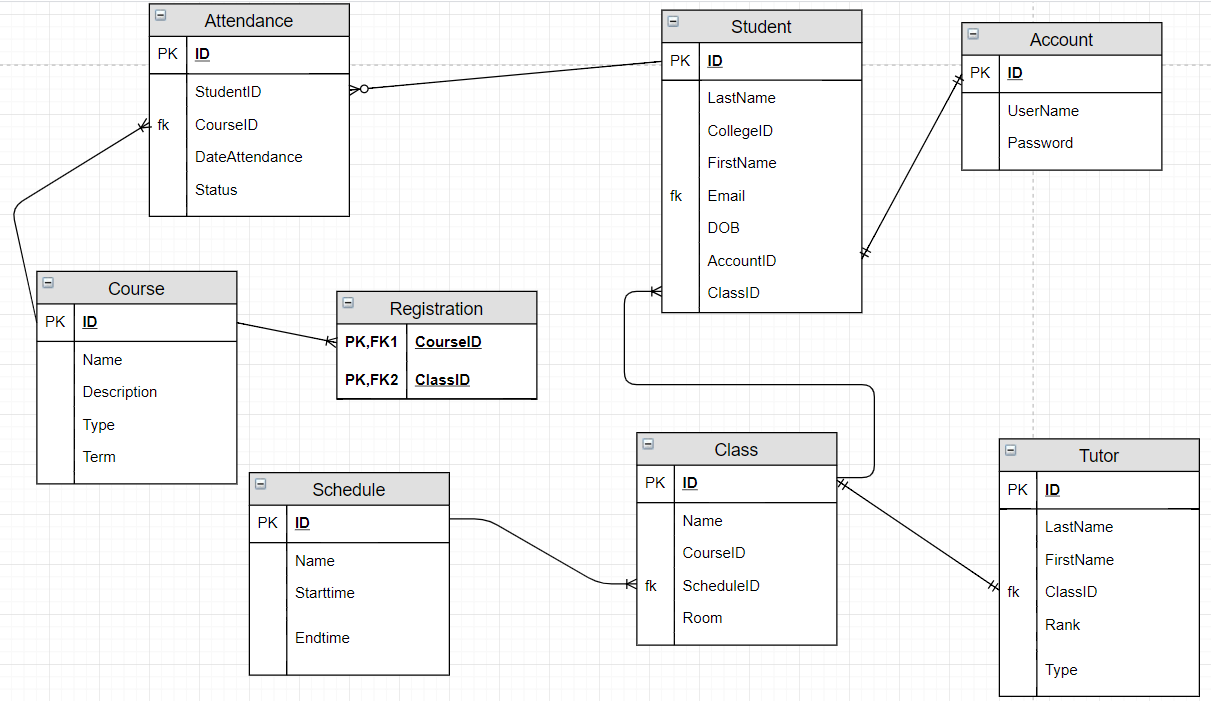
|  |  |
| --- | --- |
| **Object** | **Requirements** |
| Account | Each student or teacher only has a single account that cannot be duplicated |
| Student | Students can view class schedules, classes and attendance records |
| Tutor | Tutor can access the attendance system and update it by adding, deleting, editing |
| Admin | People can set up classes, update students by adding or deleting |

Based on the table above we can see the object and the requirements they need to do. With Admin, they can login to the account and update the students. For tutors, they have a duty to log into the system each time they teach in class to view and update data in the system. For students, they need to see their class ID to avoid confusion about the class, they must see their attendance to avoid tutors from forgetting to attendance.

# ***ERD Design***

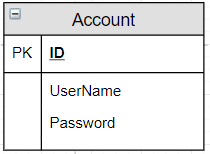
## **ERD**

With this mission, I have designed a student Attendance System. The system consists of 8 tables: Account, Student, Attendance, Course, Registration, Class, Tutor and Schedule.



## **The role of each table**

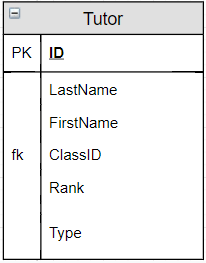
***Account table:***

1. 

Account table with PK is “Account ID”

Role of Account table: Help users can to access to system, to view and update data in the system

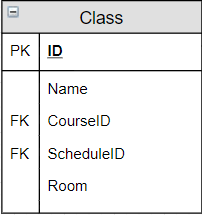
***Tutor table:***



Tutor table with PK is “Tutor ID” and FK is “Class ID”

Role of Tutor table: Tutor has the role of managing and updating data for the class

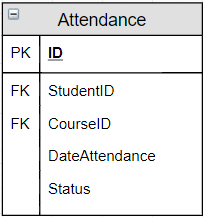
***Class table:***



Class table with PK is “Class ID” and FK is “Course ID”, “Schedule ID”

Role of Class table: Provide class information for students, such as starting year, class name, class ID

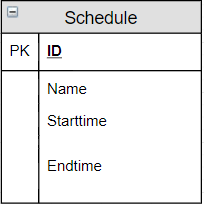
***Attendance table:***



Attendance table with FK is “Student ID”, “Course ID”

Role of Attendance table: Update and attendance for students

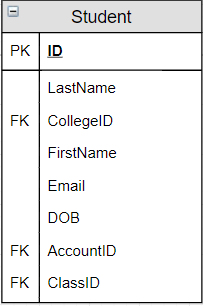
***Schedule table:***



Schedule table with PK is “Schedule ID”

Role of Schedule table: Provide study schedules such as subjects and classes.

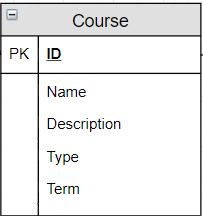
***Student table:***



Student table with PK is ”Student ID” and FK is “College ID”, “Account ID”, “Class ID”

Role of Student table: Provide personal information about yourself and receive class.

***Couse table:***



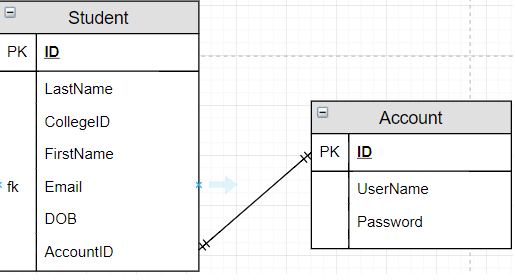
Course table with PK is ”Course ID”

Role of Course table: Provide subjects, study time for each subject, Description, Term for students and Tutor

# ***Relationship between entities***

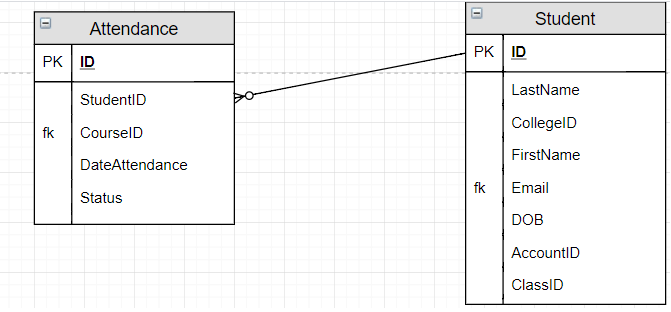
## **Contact between Account table and Student table**

The "Account ID" will be the FK of the Student table and it has a "1 to 1" relationship because each student has only one account.



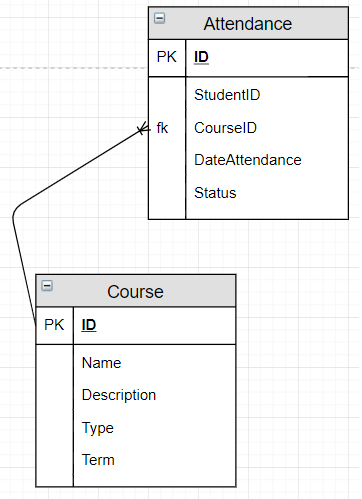
## **Contact between Student table and Attendance table**

The "Student ID" will be the FK of the Attendance table and it has a "1 to Many" relationship because each student will get a multiple attendance record.



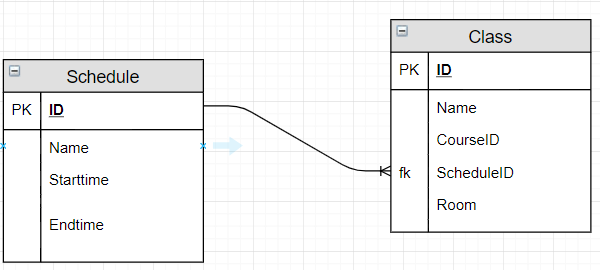
## **Contact between Attendance table and Course table**

The "Course ID" will be the FK of the attendance table and it has a "1 to Many" relationship because each course will have multiple attendance.



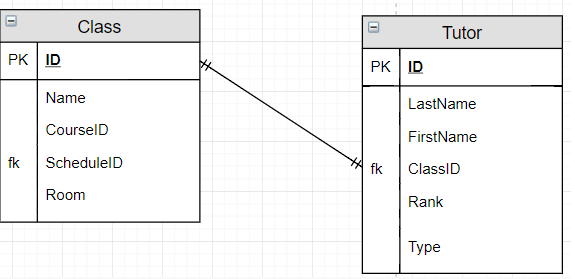
## **Contact between Class table and Schedule**

The "schedule ID" will be FK of the Class table and it has a "1 to Many" relationship because each class has a separate schedule and a schedule can create multiple classes.



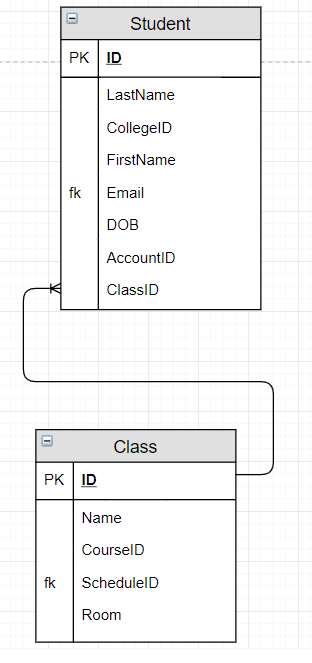
## **Contact between Class table and Tutor table**

The "class ID" will be the FK of the Tutor table and it has a "1 to 1" relationship because each class has only one teacher



## **Contact between Class table and Student table**

The "class ID" will be the FK of the Student table and it has a "1 to Many" relationship because each class has a lot of students.



# ***Physical design***

*Physical design of Account:* The Account table allows us to log in to an account, each student has a separate account and only one account.

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **ID** | **Char(1)** |
| **UserName** | **Varchar(30)** |
| **Password** | **Int** |

*Example:*

|  |  |  |
| --- | --- | --- |
| **ID** | **UserName** | **Password** |
| **1** | **NguyenThang** | **12345678** |
| **2** | **BuiKha** | **87654321** |

*Physical design of Student:* With the Student table, we can clearly see each student ID, with each ID corresponding to a student associated with an account.

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **ID** | **Char(8)** |
| **LastName** | **Varchar(30)** |
| **CollegeID** | **Varchar(20)** |
| **FirstName** | **Varchar(30)** |
| **Email** | **Varchar(30)** |
| **DOB** | **Date** |
| **AccountID** | **Char(1)** |
| **ClassID** | **Char(7)** |

*Example:*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | **LastName** | **CollegeID** | **FirstName** | **Email** | **DOB** | **AccountID** | **ClassID** |
| **1** | **Thang** | **1** | **Nguyen** | **nguyenthang@fpt.edu.com** | **1999-08-04** | **1** | **GCH0707** |
| **2** | **Kha** | **2** | **Bui** | **buikha@fpt.edu.vn** | **1999-05-03** | **2** | **GCH0706** |
| **3** | **Dat** | **1** | **Pham** | **phamdat@fpt.edu.vn** | **2000-11-02** | **3** | **GCH0707** |

*Physical design of Attendance:* The Attendance table takes the Student ID of the Student table.

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **ID** | **Char(2)** |
| **StudentID** | **Char(8)** |
| **CourseID** | **Char(2)** |
| **DataAttendance** | **Date** |
| **Status** | **Varchar(30)** |

*Example:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **StudentID** | **CourseID** | **DataAttendance** | **Status** |
| **10** | **GCH17549** | **D1** | **2019-04-19** | **Absent** |
| **15** | **GCH17465** | **C1** | **2019-04-17** | **Attended** |
| **20** | **GCH17532** | **J2** | **2019-04-17** | **Attended** |

*Physical design of Course:*

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **ID** | **Char(2)** |
| **Name** | **Varchar(30)** |
| **Description** | **Varchar(30)** |
| **Type** | **Varchar(30)** |
| **Term** | **Varchar(30)** |

*Example:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Type** | **Term** |
| **D1** | **Database** | **Good** | **IT** | **Summer** |
| **C2** | **C#** | **Great** | **IT** | **Spring** |
| **J1** | **Java** | **Bad** | **IT** | **Spring** |

*Physical design of Registration*:

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **CourseID** | **Char(2)** |
| **ClassID** | **Char(7)** |

*Example:*

|  |  |
| --- | --- |
| **CourseID** | **ClassID** |
| **D1** | **GCH0707** |
| **C2** | **GCH0706** |
| **J1** | **GCH0707** |

*Physical design of Class:* Each Class has a private Tutor and has a separate schedule. For each Tutor and a schedule will create a class.

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **ID** | **Char(7)** |
| **Name** | **Varchar(30)** |
| **CourseID** | **Char(2)** |
| **ScheduleID** | **Char(1)** |
| **Room** | **Int** |

*Example:*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Name** | **CourseID** | **ScheduleID** | **Room** |
| **GCH0707** | **IT** | **D1** | **1** | **102** |
| **GCH0706** | **Maketing** | **C2** | **3** | **203** |

*Physical design of Tutor:*

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **ID** | **Char(7)** |
| **LastName** | **Varchar(30)** |
| **FirstName** | **Varchar(30)** |
| **ClassID** | **Char(7)** |
| **Rank** | **Varchar(20)** |
| **Type** | **Varchar(20)** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **LastName** | **FirstName** | **ClassID** | **Rank** | **Type** |
| **GCH1852** | **Duong** | **Pham** | **GCH0707** | **Professor** | **Teacher** |
| **GCh1923** | **Hoang** | **Tran** | **GCH0706** | **Master** | **Staff** |

*Physical design of Schedule:* Each schedule is tied to each class, each class has a different schedule.

|  |  |
| --- | --- |
| **Column Name** | **Data type** |
| **ID** | **Char(1)** |
| **Name** | **Varchar(30)** |
| **Starttime** | **Date** |
| **Endtime** | **Date** |

*Example:*

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Starttime** | **Endtime** |
| **1** | **Spring** | **2019-02-20** | **2019-06-15** |
| **3** | **Summer** | **2019-06-20** | **2019-10-08** |

# ***Summary***

As mentioned above, I designed an Attendance system for FPT University, helping the school to manage better. Thank you!!!