**ASSIGNMENT**

Student name: Pham Ngoc Hoa | SE05740

Student name: nguyen hai nam | se05123

Teacher: nguyen quynh chi

DBI202 – DATABASE SYSTEM OF DORM FPT UNIVERSITY

March 20, 2018

# TABLE OF CONTENTS

## INTRODUCE THE PROBLEM…………………………………………………………………………………………….

### Describe the problem……………………………………………………………………………………………………….

### Management objectives…………………………………………………………………………………………………..

## entity – relationship – er…………………………………………………………………………………………

### difinITION entity – attributE…………………………………………………………………………………………..

### set-up entity – relationship…………………………………………………………………………………………….

## data dictionary………………………………………………………………………………………………………….

### DEFINITION OF TABLES………………………………………………………………………………………………………….

### SET-UP TRIGGER…………………………………………………………………………………………………………………….

# INTRODUCE THE PROBLEM

## Describe the problem

Nowadays, Students in FPT University almost live in five Dom, there are A, B, C, D, F and have some guard to manager it. But, it is still have not database system for manage dorm. After the actual survey, the result are as follows:

* Student live in room of dom, each room have one or more student, however at one point an student is only live in one room.
* The head of each room is an student who is the head of the department. And every student is managed by an guard.
* Guard records information about students, including: student code, first name, last name, email, phone, gender, course, roles. Character of the student are “M”(Boy) of ”F”(Girl). Phone number can not greater than 11 digits.
* Gender of student have to equal gender of room.
* Guard also needs to manager number penalize to calculate fines by day or month. Calculate the amount of money owed by the student due to violation of regulations.
* Each penalty include: name of student is fined, type of penalize and quantity.
* Student can register live in a room when number of bed is empty, that student have to pay for register. Check Price are “1” (have paid) or “0” (not paid).
* Information for register include: room code, student code, date that student was check – in , day that student was check-out.
* Each room can 6 to 12 people live together, each dom have one or more dom, however at one point an room is only is in one dom.
* Room code is combination by Name Dome and Floor and number of room and number of bed. Example: a2055, it mean dome a, floor 2, room 05, bed 5.
* Information of room include: room code, location, number bed, gender, status bed.
* Location of the room are “L” (Left) or “R” (Right).
* Status bed of room are “1” (This bed is empty, student can register live in this bed) or ”0” (This bed is registered, student can not register in this bed).
* Guard also need to manager facilities in FPT University to call fixer to fix the rooms are broken to student continue to live safe.
* Items of room can need fix example: light bulb, table,… ect. Each item inclue: Item code, name of things need fix, content fix.
* Fix detail include: room code, item code, name of fixer, day reported malfunction, day fix malfunction, status are “1” (fixe) or “0” (not fixed), quantity the same thing is broken, and price for pay fix that each thing.
* Guard manage dom, each dom have one or more guard manage, however at one point an guard is only manage in one room.
* Information of guard include: guard code, first name of guard, last name of guard, email of guard, phone of guard, and guard can manage guard.
* Information of dom include: name of dom, maximum people can live in that room, price have to pay for each semester.

**Request:**

* Daily, guard need to caculate the total sutdent fined, caculate the total money.
* Guard need check day student check-in, check out, day report item broken.
* Monthly, guard need to count and display student not pay money for room.
* Monthly, guard need cacule total amount collected.
* Guard can check information of student.
* Guard can check room is emty for student register.
* Guard can check item broken in dom, in room to call fixer.

## Management objectives

* Manage student and change of students.
* Manage penalize and calculate money fined.
* Manage room for student register.
* Manage items broken to fix.

**Important output**

* Total money for fined of each student.
* Count number room is empty.
* Count number room have items broken.

# entity – relationship – er

## difinITION entity – attributE

Base on the problem description and management objectives, we can present several entities and attributes of the entity as follow:

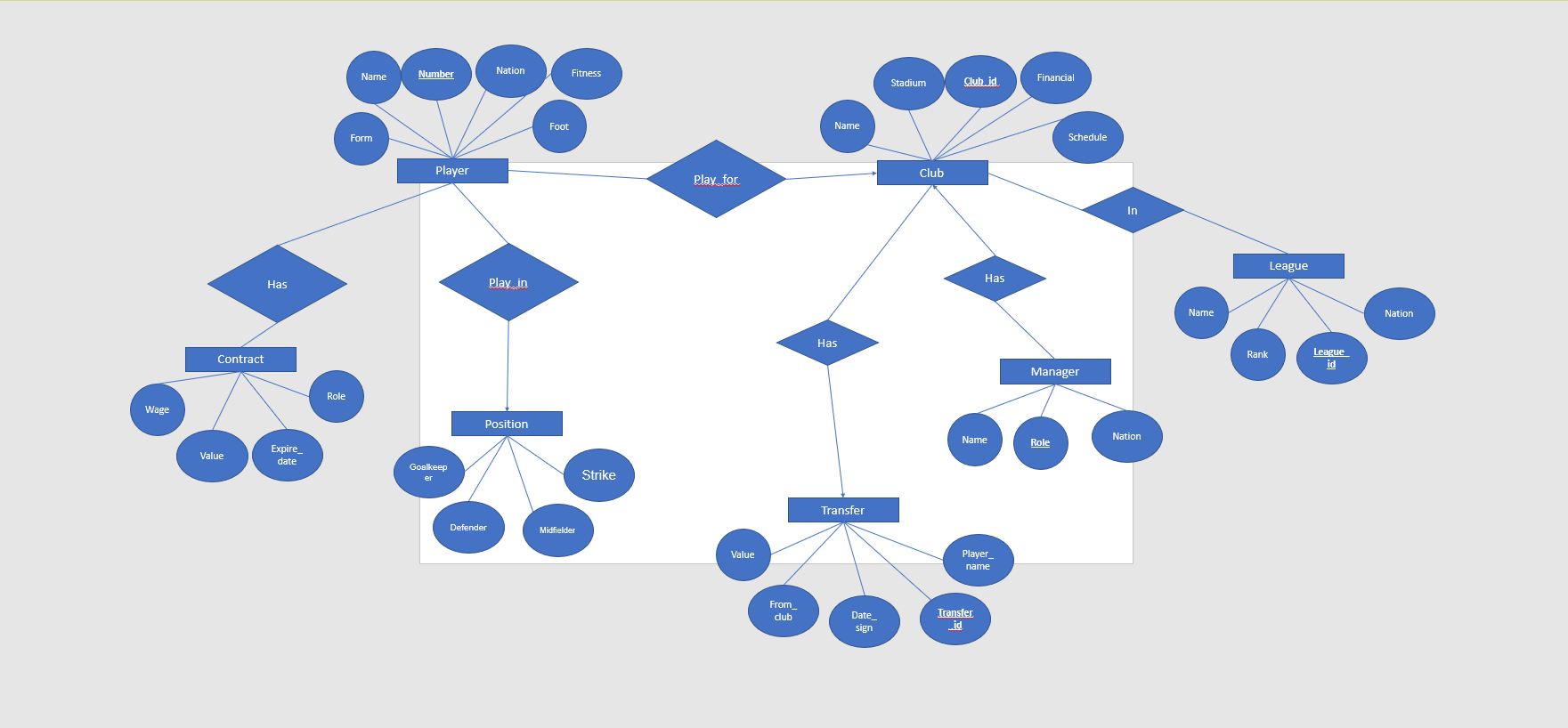
* Student: **StudentID, ManagerID,** First Name, Last Name, Email, Phone, Gender, Course, Role, Country.
* Register: **RoomID, StudentID,** Check-In, Check-Out, CheckPrice.
* Room: **RoomID** , **NameDom,** Name, Floor, Location, Number Bed, Gender, Status.
* Fix Detail: **RoomID, ItemID,** Fixer, Date Report, Date Dix, Quantity, Price.
* Items: **ItemID**, Name, Content.
* Student Penalize: **StudentID, PenalizeID,** Date, Quantity.
* Penalize: **PenalizeID**, Name, Price.
* Guard: **GuardID*,*** First Name, Last Name, Email, Phone.
* Dome: **NameDome**, **GuardID,** People Number, Price.

## set-up entity – relationship

\* Some symbols used in the model

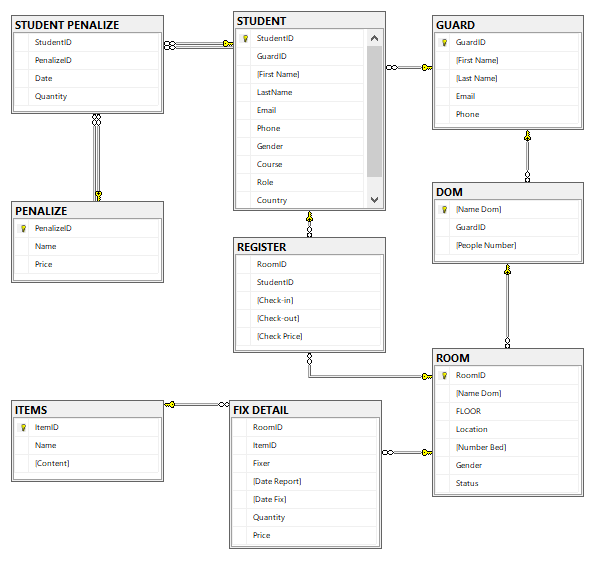
**Attibute**

|  |  |
| --- | --- |
| * Key / identifier attribute |  |
| * Attribute description / description | **ENTITY**  Attribute |
| * Entity | **WEAK ENTIRY** |
| * Weak entity | *Relationship* |
| * Relationship |  |
| * Connectivity (force) = 1 |  |
| * Connectivity = N |  |



\

### We have a model



# data dictionary

Just for example on some tables (other table are similar, you have to define all the tables in your database). Note: to run the query you have to define the table 1 first then go to the side tables much

## DEFINITION OF TABLES

### **Table Student**

| Column Name | Data Type | Default | Check | Key/ Index/ Constraint |
| --- | --- | --- | --- | --- |
| StudentID | Narchar(7) |  | SE|SBxxxxx | PK, Not null |
| GuardID | Int |  |  | FK reference GUARD(GuardID) |
| First Name | Nvarchar(45) |  |  | Not null |
| Last Name | Nvarchar(45) |  |  | Not null |
| Email | Nvarchar(255) |  |  |  |
| Phone | Char(11) |  | 0 to 9 | Unique |
| Gender | Char(1) | M | ‘F’ or ‘M’ | Not null |
| Course | Char(3) |  | Kxx |  |
| Role | Nvarchar(30) | Member | ‘Member’ or ‘Leader’ | Not null |
| Country | Nvarchar(50) | VietNam |  |  |

***Example:***

| StudentID | GuardID | First Name | Last Name | Email | Phone | Gender | Course | Role | Contry |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SE05740 | 2 | Hòa | Phạm Ngọc | [Thaycacac@gmail.com](mailto:Thaycacac@gmail.com) | 0968038714 | M | K12 | Member | VietNam |
| SE05451 | 1 | Nam | Nguyễn Hải | [Hainam98@gmail.com](mailto:Hainam98@gmail.com) | 0965125744 | M | K12 | Leader | VietNam |
| SE03751 | 1 | Vân | Lương Hồng | [louhoang@gmail.com](mailto:louhoang@gmail.com) | 0984481347 | F | K13 | Member | VietNam |
| SB04754 | 3 | Abbey | Dove | [Dovesb04754@fpt.edu.vn](mailto:Dovesb04754@fpt.edu.vn) | 0267554687 | M | K9 | Member | England |

***Code:***

--create table student

CREATE TABLE STUDENT(

StudentID VARCHAR(7) PRIMARY KEY NOT NULL CHECK(StudentID LIKE 'SE[0-9][0-9][0-9][0-9][0-9]' OR StudentID LIKE 'SB[0-9][0-9][0-9][0-9][0-9]'),

GuardID INT FOREIGN KEY REFERENCES dbo.GUARD(GuardID),

[First Name] NVARCHAR(45) NOT NULL,

[LastName] NVARCHAR(45) NOT NULL,

Email NVARCHAR(255),

Phone CHAR(11) UNIQUE CHECK(Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

Gender CHAR(1) NOT NULL DEFAULT 'M' CHECK(Gender IN('F','M')),

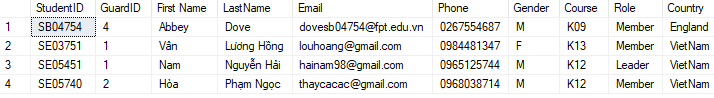
Course CHAR(3) CHECK(Course LIKE 'K[0-9][0-9]'),

[Role] NVARCHAR(30) NOT NULL DEFAULT 'Member',

Country NVARCHAR(50) DEFAULT 'VietNam'

)

***Result:***



### TABLE **GUARD**

| Column Name | Data Type | Default | Check | Key/ Index/ Constraint |
| --- | --- | --- | --- | --- |
| GuardID | Int |  |  | unique |
| First Name | Nvarchar(45) |  |  | Not null |
| Last name | Nvarchar(45) |  |  | Not null |
| Email | Nvarchar(255) |  |  | Not null |
| Phone | Char(11) |  | 0 to 9 | Unique |

***Example:***

| GuardID | First Name | Last Name | Email | Phone |
| --- | --- | --- | --- | --- |
| 1 | Nhật | Trần Quang | [Rutexp@gmail.com](mailto:Rutexp@gmail.com) | 0968745121 |
| 2 | Đức | Phan Văn | [Ducpvsecurity@gmail.com](mailto:Ducpvsecurity@gmail.com) | 0984512365 |
| 4 | Cường | Nguyễn Xuân | [Cuongnx98@gmail.com](mailto:Cuongnx98@gmail.com) | 0956484512 |

***Code:***

--create table guard

CREATE TABLE GUARD(

GuardID INT PRIMARY KEY,

[First Name] NVARCHAR(45) NOT NULL,

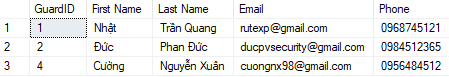
[Last Name] NVARCHAR(45) NOT NULL,

Email NVARCHAR(255),

Phone CHAR(10) UNIQUE CHECK(Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]' OR Phone LIKE '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')

)

***Result:***

******

### **table ROOM**

| Column Name | Data Type | Default | Check | Key/ Index/ Constraint |
| --- | --- | --- | --- | --- |
| RoomID | Char(5) |  |  | Primary key |
| Name Dom | Char(1) |  | A|B|C|D|F | Not null |
| Floor | Smallint |  | 1 to 5 | Not null |
| Location | Nvarchar(4) |  | ‘Right’ or ‘Left’ | Not null |
| Number Bed | Smallint |  | 1 to 12 | Not null |
| Gender | Char(1) | M | ‘F’ or ‘M’ | Not null |
| Status | Char(1) | 1 | ‘1’ or ‘0’ | Not null |

***Example:***

| RoomID | Name Dom | Floor | Location | Number Bed | Gender | Status |
| --- | --- | --- | --- | --- | --- | --- |
| A2056 | A | 2 | Left | 6 | M | 1 |
| D2064 | D | 2 | Left | 4 | M | 1 |
| C3015 | C | 3 | Right | 5 | F | 1 |
| A2058 | A | 2 | Right | 8 | M | 0 |

***Code:***

--create room

CREATE TABLE ROOM(

RoomID CHAR(5) PRIMARY KEY CHECK(RoomID LIKE '[A-F][1-5][0-9][0-9][0-9]' OR RoomID LIKE '[A-F][1-5][0-9][0-9][0-9][0-9]'),

[Name Dom] CHAR(1) FOREIGN KEY REFERENCES DOM([Name Dom]),

[FLOOR] SMALLINT NOT NULL CHECK([FLOOR] BETWEEN 1 AND 5),

[Location] NVARCHAR(5) NOT NULL CHECK([Location] IN ('Left', 'Right')),

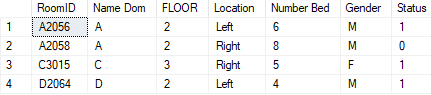
[Number Bed] SMALLINT NOT NULL CHECK([Number Bed] BETWEEN 1 AND 12),

Gender CHAR(1) NOT NULL DEFAULT 'M' CHECK(Gender IN('F','M')),

[Status] SMALLINT NOT NULL DEFAULT 1 CHECK(Status BETWEEN 0 AND 1)

)

***Result:***

******

## SET-UP TRIGGER

….