**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A picture containing text, clipart

Description automatically generated**

**BÀI TẬP LỚN­**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**CƠ SỞ DỮ LIỆU QUẢN LÝ TRƯỜNG TRUNG HỌC PHỔ THÔNG**

**Giảng viên hướng dẫn: T.S Nguyễn Huy Đức**

**Thực hiện: Nhóm 13 - Lớp 61HT**

**1.Đặng Quang Vinh**

**2.Đỗ Thị Ngọc Ánh**

**3.Trần Thị Phúc**

Hà Nội, 1/2022

**TRƯỜNG ĐẠI HỌC THỦY LỢI**

**KHOA CÔNG NGHỆ THÔNG TIN**

**A picture containing text, computer

Description automatically generated**

**BÀI TẬP LỚN**

**CƠ SỞ DỮ LIỆU**

**ĐỀ TÀI:**

**CƠ SỞ DỮ LIỆU QUẢN LÝ TRƯỜNG TRUNG HỌC PHỔ THÔNG**

**Giảng viên hướng dẫn: T.S Nguyễn Huy Đức**

**Thực hiện: Nhóm 13 - Lớp 61HT**

**1.Đặng Quang Vinh**

**2.Đỗ Thị Ngọc Ánh**

**3.Trần Thị Phúc**

HÀ NỘI, 1/2022

**BÁO CÁO BÀI TẬP LỚN**

**Học phần**: Hệ quản trị cơ sở dữ liệu

**Để tài: Quản lý trường trung học phổ thông**

**Mã nhóm**: 13

**Thành viên**:

1. Đỗ Thị Ngọc Ánh:1951060538

2. Trần Thị Phúc: 1951060923

3. Đặng Quang Vinh: 1951061121(Nhóm trưởng)

# **Phân công công việc và thông tin**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phân công công việc trong nhóm** | | | |
|  | Đỗ Thị Ngọc Ánh | Trần Thị Phúc | Đặng Quang Vinh |
| Công việc |  |  |  |
| Tự đánh giá | 10 | 10 | 10 |
| Ghi chú |  |  |  |

**MỤC LỤC**

[**LỜI NÓI ĐẦU 2**](#_Toc478733600)

[**MỤC LỤC 3**](#_Toc478733601)

[**I. Xác định quy tắc/ràng buộc, xây dựng mô hình thực thể liên kết ER 4**](#_Toc478733602)

[**1.1 Liệt kê, chính xác hóa và lựa chọn thông tin. 4**](#_Toc478733603)

[**1.2 Các thực thể, thuộc tính: 4**](#_Toc478733604)

[**1.3 Các quy tắc/ ràng buộc 4**](#_Toc478733605)

[**1.4 Vẽ mô hình thực thể liên kết chuẩn hóa và thu gọn sơ đồ ER: 5**](#_Toc478733606)

[**II . Chuyển từ mô hình ER sang mô hình quan hệ 6**](#_Toc478733607)

[**III. Xác định khóa 7**](#_Toc478733608)

[**3.1 Xác định phụ thuộc hàm. 7**](#_Toc478733609)

[**3.2 Tìm khóa cho các quan hệ 8**](#_Toc478733610)

[**lV. Chuẩn hóa lược đồ quan hệ thành dạng 3NF hoặc BCNF 9**](#_Toc478733611)

[**V. Câu lệnh truy vấn dữ liệu SQL 11**](#_Toc478733612)

[**5.1 Tạo bảng 11**](#_Toc478733613)

[**5.2 Các câu hỏi truy vấn và ngôn ngữ SQL. 13**](#_Toc478733614)

[**KẾT LUẬN 18**](#_Toc478733615)

[**TÀI LIỆU THAM KHẢO 19**](#_Toc478733616)

**LỜI NÓI ĐẦU**

**T**rong thời đại ngày nay, công nghệ thông tin đang được phát triển mạnh mẽ, nhanh chóng và xâm nhập vào nhiều lĩnh vực khoa học, kỹ thuật, cũng như trong cuộc sống. Nó trở thành công cụ hữu ích để quản lý , xây dựng , thiết kế … đặc biệt trong công tác quản lý nói chung và quản lý nhà hàng nói riêng.

Trước đây, khi máy tính chưa được ứng dụng rộng rãi, các công việc quản lý nhà hàng đều được làm rất thủ công nên mất rất nhiều thời gian, nhân lực cũng như tài chính. Ngày nay, cùng với sự phát triển của CNTT máy tính đã được sử dụng rỗng rãi trong các cơ quan, nhà máy, trường, siêu thị, … giúp cho công việc được tốt hơn. Việc sử dụng máy tính vào công tác quản lý là một yêu cầu cấp thiết được cải tiến ngày nay. Vì vậy, em đã xây dựng một cơ sở dữ liệu “ Quản lý trường THPT” với mong muốn có thể giúp cho việc xây dựng hệ thống quán lý một cách dễ dàng hơn. Để từ đó có thể đưa ra được thực nghiệm ứng dụng vào thực tế.

Là một sinh viên công nghệ thông tin, nhận được tầm quan trọng của tin học hóa và các doanh nghiệp tin học trong cơ chế thị trường như hiện nay ở Việt Nam, sau một thời gian tìm hiểu và học tập, em đã quyết định chọn đề tài **“Quản lý trường THPT”** nhằm ứng dụng những kiến thức đã học được và kết hợp chúng với kiến thức về thực tế.

Em xin chân thành cảm ơn thầy giáo TS. Nguyễn Huy Đức giảng viên đã tận tình hướng dẫn, giảng dạy em môn Hệ quản trị cơ sở dữ liệu để hoàn thành được bài tập này.

# **Xác định quy tắc/ràng buộc, xây dựng mô hình thực thể liên kết ER**

## **Các thực thể, thuộc tính**:

**a. USERS**(UserID, UserName, UserPassword, UserEmail, UserStatus, UserCode, UserRole)

Mỗi người dùng có mã (UserID) duy nhất, tên tài khoản của người dùng (UserName), mật khẩu (UserPassword), Email (UserEmail), trạng thái tài khoản (UserStatus), mã xác thực (UserCode), vai trò của người dùng (UserRole)

**b.PROFILES** (UserID, ProName, ProPhone, ProAddress, ProGender, ProBirth, ProAva)

Mỗi người dùng có mã(UserID) duy nhất, họ và tên (ProNam), số điện thoại (ProPhone), địa chỉ (ProAddress), giới tính (ProGender), ngày sinh (ProBirth), ảnh đại diện (ProAva)

**c.CLASS** (ClassID, ClassName, ClassGrade)

Mỗi lớp có mã (ClassID) duy nhất, tên lớp (ClassName), tên khối (ClassGrade)

**d.SUBJECTS**(SubjectID, SubjectName, SubjectType)

Mỗi môn học có mã (SubjectID) duy nhất, tên môn học (SubjectName), loại môn học (SubjectType)

## **Các quy tắc / ràng buộc**

1. Mỗi người dung (user) có một UserID duy nhất được sử dụng để phân biệt các người dùng (primary key)
2. Mỗi người dùng là học sinh sẽ thuộc duy nhất một lớp
3. Mỗi người dùng là giáo viên (UserRole là Giáo viên) có thể dạy nhiều môn học cho nhiều lớp khác nhau
4. Mỗi người dùng chỉ có thể sở hữu một sơ yếu lý lịch (profile)
5. Mỗi người dùng là học sinh có thể học nhiều môn học
6. Mỗi người dùng có thể gửi tin nhắn cho nhiều người dùng khác
7. Mỗi tên tài khoản (UserName) của người dùng đều là duy nhất
8. Mỗi sơ yếu lý lịch (profile) chỉ thuộc về duy nhất một người dùng
9. Mỗi sơ yếu lý lịch có một UserID duy nhất được sử dụng để phân biệt các sơ yếu lý lịch (primary key)
10. Mỗi sơ yếu lý lịch có một ProGender để xác định giới tính của người dùng và chỉ nhận 1 trong 2 giá trị là ‘Nam’ hoặc ‘Nữ’
11. Mỗi lớp (class) được theo học bỏi nhiểu học sinh
12. Mỗi lớp có dạy nhiều môn học do các giáo viên khác nhau giảng dạy
13. Mỗi lớp có một ClassID duy nhất sử dụng để phân biệt giữa các lớp khác nhau (primary key)
14. Mỗi môn học (subject) được giảng dạy cho nhiều lớp bởi nhiều giáo viên khác nhau
15. Mỗi môn học được theo học bởi nhiều học sinh
16. Mỗi môn học có một SubjectID duy nhất sử dụng để phân biệt giữa các môn học khác nhau (primary key)
17. Mỗi học sinh khi theo học một môn học sẽ sinh ra điểm của học sinh cho môn đó được lưu vào bảng học (study)
18. Mỗi người dùng khi nhắn tin cho người dùng khác sẽ được lưu nội dung tin nhắn và thời gian gửi vào bảng tin nhắn (messenger)
19. Hồ sơ lý lịch (profiles) của mỗi người có một cột evaluate để lưu đánh giá của người đó. Quy tắc đánh giá như sau:

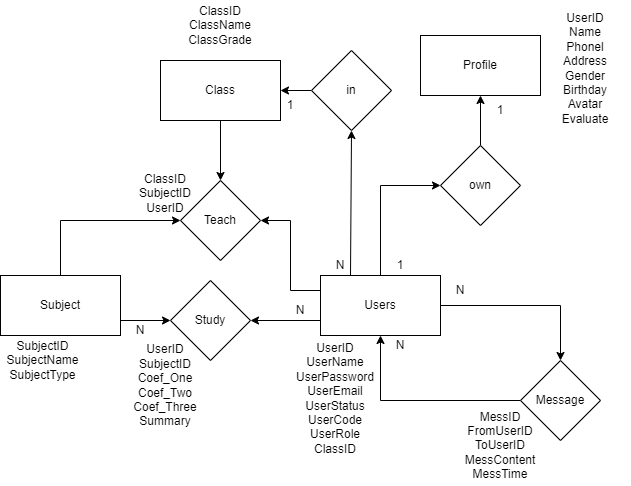
Với học sinh:

* Điểm tổng kết lớn hơn 9.5 và không có môn nào dưới 9 điểm tổng kết thì đạt học sinh xuất sắc
* Điểm tổng kết từ 8.0 và không có môn nào dưới 7 điểm tổng kết thì đạt học sinh xuất sắc
* Điểm tổng kết từ 6.5 tới 8.0 và không có môn nào dưới 5 điểm tổng kết thì đạt học sinh xuất sắc
* Điểm tổng kết từ 3.5 tới 6.5 và không có môn nào dưới 0 điểm tổng kết thì đạt học sinh xuất sắc
* Điểm tổng kếttừ 0 tới 3.5 thì đạt học sinh xuất sắc

Với giáo viên:

* Số học sinh do giáo viên đó dạy có điểm tổng kết môn lớn hơn 6.5 là lớn hơn ¾ tổng số học sinh giáo viên đó dạy và không có học sinh nào dưới 3.5 điểm tổng kết môn thì giáo viên đạt giáo viên dạy giỏi
* Nếu không giáo viên đạt giáo viên tiên tiến

## **Vẽ mô hình thực thể liên kết chuẩn hóa và thu gọn sơ đồ**



Hình 1: Mô hình thực thể liên kết ER

# **Chuyển từ mô hình ER sang mô hình quan hệ**

Trong bài bao gồm các bước:

Bước 1: Các thực thể (trừ thực thể yếu) 🡪 Các bảng các quan hệ.

Bước 2: Thực thể yếu 🡪 Các bảng các quan hệ (Khóa của thực thể chủ sẽ thành khóa ngoại).

Bước 3: Liên kết 1:1 🡪Xác định khóa của quan hệ T\_S. Đưa khóa chính của T sang làm khóa ngoại của S. Thuộc tính của T thành thuộc tính của S.

Bước 4: Liên kết 1:N🡪 Khóa của bảng bên một trở thành khóa ngoại của bảng bên nhiều.

* Mô hình quan hệ.

Diagram, schematic

Description automatically generated

Hình 2: Mô hình quan hệ

# **Chức năng của hàm, thủ tục, trigger, view trong phần mềm**

## **View**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên view | Cột có trong view | Từ các bảng | Tác dụng |
| v\_studentsOfClass | CLassID, ClassName, UserID, Name, Birth, Gender, Address, Avatar | class,  users,  profiles | Hiển thị danh sách học sinh theo lớp học |
| v\_scoresOfStudent | UserID, UserName, ClassID, ClassName, SubjectID, SubjectName, SubjectType, Coef\_one, Coef\_two, Coef\_three, Summary | users, class, subjects, study | Hiển thị và quản lý bảng điểm theo từng học sinh |
| v\_users\_profiles | UserID, UserName, UserPassword, UserEmail,  UserStatus, UserCode, UserRole, ClassID, UserID\_, ProName, ProPhone, ProAddress, ProGender, ProBirth, ProAva, Evaluate | users,  profiles | Hiển thị và quản lý tài khoản và thông tin của tài khoản đó |
| v\_subjectsOfClass | ClassID, subjects.SubjectID, SubjectName, SubjectType | subjects, teach | Hiển thị và quản lý các môn học đang được giảng dạy trong lớp |
| v\_studentsOfTeachers | teacherID, studentID, SubjectID, summary | teach, users, study | Hiển thị và quản lý các học sinh đang được giảng dạy bởi mỗi giáo viên |
| v\_users\_messeger | UserID, UserName, UserPassword, UserEmail,  UserStatus, UserCode, UserRole, ClassID, MessID, FromID, ToID, MessContent, MessTime | Users, messenger | Hiển thị và quản lí tin nhắn của người dùng |
| v\_user\_teach\_subject | UserID, UserName, UserPassword, UserEmail,  UserStatus, UserCode, UserRole, ClassID, UserIDD, SubjectID, ClassIDD, SubjectIDD, SubjectName, SubjectType | Users, teach, subjects | Hiển thị và quản lí môn học của người dùng |

## **function**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên function | Đầu vào | Trả về | Tác dụng |
| f\_DiemTrungBinh | @UserID varchar(4), @SubjectID varchar(10) | float | Tính điểm tổng kết cho môn học nào đó của học sinh với công thức:  @DiemTrungBinh = ((Coef\_one \* 1 + Coef\_two \* 2 + Coef\_three \* 3)/6) |
| f\_Diem | @UserID nvarchar(4) | table | Hiển thị và quản lý điểm số của tất cả các môn mà học sinh đó đã và đang học. |
| f\_SiSoHocSinh | @ClassID varchar(10) | int | Đém sĩ số của một lớp bất kì. Hỗ trọ cho việc quản lý các lớp học |
| get\_avgAllScore | @UserID varchar(4) | float | Tính điểm trung bình cả năm của một học sinh theo công thức:  (tổng điểm trung bình) / (số môn) |
| f\_evaluate | @UserID varchar(4) | nvarchar(50) | Đánh giá một học sinh hoặc một giáo viên dựa trên các điều kiện đánh giá |
| Tachten | @ProName nvarchar(60) | nvarchar(30) | Tách tên từ chuỗi Họ tên người dùng |
| Dockhoinguyen | @ClassGrade int | nvarchar(20) | Đọc tên Khối ra thành chữ tương ứng |

## **trigger**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên trigger | Bảng kích hoạt | Loại trigger | Tác dụng |
| add\_user | v\_users\_profiles | instead of insert | Khi tạo một tài khoản mới sẽ thêm cho tài khoản đó một sơ yếu lý lịch. Nếu tài khoản đó là của học sinh sẽ tự động thêm bảng điểm cho học sinh đó |
| del\_user | v\_users\_profiles | instead of delete | Khi xóa đi một tài khoản khỏi hệ thống thì những thông tin liên quan tới tài khoản đó sẽ bị xóa đi như sơ yếu lý lịch, tin nhắn, bảng điểm đối với học sinh, các môn đang giảng dạy đối với giáo viên |
| update\_scores | study | after update | Sau khi sửa điểm vào bảng điểm của học sinh thì tự động cập nhật lại điểm trung bình cho học sinh đó |
| update\_users | users | for update | Khi sửa thông tin của tài khoản không cho phép sửa tên tài khoản |
| InsertTeach | Teach | Insert Teach | Insert bảng teach, nếu người dùng có mã userrole là giáo viên thì thông báo insert thành công còn không thì thông báo lỗi. |
| InsertStudy | Study | Insert Study | Insert bảng study, nếu người dùng có mã userrole là học sinh thì thông báo insert thành công còn không thì thông báo lỗi. |

## **Stored procedures**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên stored procedures | Đầu vào | Trả về | Tác dụng |
| sp\_SiSoNhieu | @SiSo int | @dsss cursor varying | Lấy ra các lớp với sĩ số nhỏ hơn số nhập vào |
| sp\_SoLuongTinNhan | @UserID nvarchar(4), @NgayDau date, @NgayCuoi date, | @TongTinNhan int | Lấy ra số lượng tin nhắn mà một người đã sử dụng trong một khoảng thời gian nào đó |
| sp\_updateEvaluate\_class | @ClassID varchar(10) |  | Cập nhật đánh giá cho học sinh trong một lớp |
| sp\_updateEvaluate\_user | @UserID varchar(4) |  | Cập nhật đánh giá cho một người dùng |
| sp\_searchMatchMess | @search Nvarchar(50) | @count int, @matchedList cursor varying | Tìm kiếm các tin nhán trùng khớp với chuỗi nhập vào |
| INSERT\_users |  |  | Bổ sung dữ liệu cho bảng users |
| DiemTBcaonhat | @tenmon nvarchar(10) | @diem float, @ten nvarchar(30) | Lấy ra tên học sinh có điểm trung bình cao nhất với tham số truyền vào là môn học |

# **Câu lệnh truy vấn dữ liệu SQL**

## **Tạo bảng**

-- Tạo cơ sở dũ liệu, file dữ liệu + file nhật ký

create database thpt\_vap

on primary (

Name=THPT\_data,

Filename= 'D:\Hệ quản trị CSDL\btl\THPT\_data.mdf',

size=10MB,

maxsize=50MB,

filegrowth =2MB)

log on (

Name=THPT\_log,

Filename= 'D:\Hệ quản trị CSDL\btl\THPT\_log.ldf',

size=5MB,

maxsize=20MB,

filegrowth =1MB);

use thpt\_vap

--Tạo bảng class

CREATE TABLE class (

ClassID varchar(10) NOT NULL,

ClassName Nvarchar(10) NOT NULL,

ClassGrade int NOT NULL,

PRIMARY KEY (ClassID),

UNIQUE (ClassName)

)

--Tạo bảng users

CREATE TABLE users (

UserID varchar(4) NOT NULL,

UserName Nvarchar(60) NOT NULL,

UserPassword varchar(255) NOT NULL,

UserEmail varchar(100) DEFAULT NULL,

UserStatus int NOT NULL DEFAULT 0,

UserCode varchar(8) DEFAULT 00000000,

UserRole Nvarchar(20) NOT NULL,

ClassID varchar(10) DEFAULT NULL,

PRIMARY KEY (UserID),

UNIQUE (UserName),

UNIQUE (UserEmail),

FOREIGN KEY (ClassID) REFERENCES class (ClassID)

)

--Tạo bảng profiles

create table profiles(

UserID varchar(4) NOT NULL,

ProName Nvarchar(60) NOT NULL,

ProPhone varchar(20),

ProAddress Nvarchar(150),

ProGender Nvarchar(20),

ProBirth Date,

ProAva varchar(60) DEFAULT 'ava.png',

evaluate nvarchar(50) DEFAULT N'Chưa có đánh giá',

primary key (UserID),

foreign key (UserID) references users(UserID),

)

--Tạo bảng subjects

create table subjects(

SubjectID varchar(10) NOT NULL,

SubjectName Nvarchar(60) NOT NULL,

SubjectType Nvarchar(60),

primary key (SubjectID),

)

--Tạo bảng teach

create table teach(

UserID varchar(4) Not Null,

SubjectID varchar(10) Not Null,

ClassID varchar(10) Not Null,

primary key (UserID, SubjectID, ClassID),

foreign key (UserID) references users(UserID),

foreign key (SubjectID) references subjects(SubjectID),

foreign key (ClassID) references class(ClassID)

)

--Tạo bảng study

create table study(

UserID varchar(4) Not Null,

SubjectID varchar(10) Not Null,

Coef\_one float,

Coef\_two float,

Coef\_three float,

Summary float,

primary key (UserID, SubjectID),

foreign key (UserID) references users(UserID),

foreign key (SubjectID) references subjects(SubjectID),

)

--Tạo bảng messenger

CREATE TABLE messenger (

MessID int NOT NULL IDENTITY(1,1),

FromID varchar(4) NOT NULL,

ToID varchar(4) NOT NULL,

MessContent Nvarchar(255) NOT NULL,

MessTime datetime NOT NULL DEFAULT current\_timestamp,

PRIMARY KEY (MessID,FromID,ToID),

FOREIGN KEY (ToID) REFERENCES users(UserID),

FOREIGN KEY (FromID) REFERENCES users(UserID)

)

## **Tạo View, Function, Trigger, Stored procedures**

### **Tạo View**

--View học sinh của lớp

create view v\_studentsOfClass(CLassID, ClassName, UserID, Name, Birth, Gender, Address, Avatar)

as select c.CLassID, c.ClassName, u.UserID, p.ProName, p.ProBirth, p.ProGender, p.ProAddress, p.ProAva

from class as c, users as u, profiles as p

where c.ClassID = u.ClassID and u.UserID = p.UserID

--2.Tạo view v\_scoresOfStudent để hiển thị thông tin của học sinh và môn học gồm UserID, UserName, ClassName, SubjectName, SubjectType,

-- cofe\_one, Coef\_two, Cofe\_three, Summary, Conduct

create view v\_scoresOfStudent(UserID, UserName, ClassID, ClassName, SubjectID, SubjectName, SubjectType, Coef\_one, Coef\_two, Coef\_three, Summary)

as

select u.UserID, u.UserName, c.ClassID, ClassName, sb.SubjectID, sb.SubjectName, sb.SubjectType, st.Coef\_one, st.Coef\_two,

st.Coef\_three, st.Summary

from users as u, class as c, subjects as sb, study as st

where u.ClassID=c.ClassID and st.UserID=u.UserID and st.SubjectID=sb.SubjectID

-- View users x profiles

create view v\_users\_profiles(UserID, UserName, UserPassword, UserEmail,

UserStatus, UserCode, UserRole, ClassID, UserID\_, ProName, ProPhone, ProAddress, ProGender, ProBirth, ProAva, Evaluate)

as select \* from users, profiles where users.UserID = profiles.UserID

-- View lấy ra môn học của một lớp

create view v\_subjectsOfClass as

select ClassID, subjects.SubjectID, SubjectName, SubjectType from subjects, teach

where teach.SubjectID = subjects.subjectID

-- View học sinh giáo viên

create view v\_studentsOfTeachers(teacherID, studentID, SubjectID, summary) as

select distinct teach.UserID as teacherID, users.UserID as studentID, study.SubjectID, study.Summary

from teach, users, study where teach.ClassID = users.ClassID and users.UserID = study.UserID;

--tạo view user\_messenger--

create view v\_users\_messeger(UserID, UserName, UserPassword, UserEmail,

UserStatus, UserCode, UserRole, ClassID, MessID, FromID, ToID, MessContent, MessTime)

AS

select \* from users, messenger where users.UserID = messenger.FromID ;

SELECT \* FROM v\_users\_messeger

--tạo view v\_user\_teach\_subject--

create view v\_user\_teach\_subject(UserID, UserName, UserPassword, UserEmail,

UserStatus, UserCode, UserRole, ClassID, UserIDD, SubjectID, ClassIDD, SubjectIDD, SubjectName, SubjectType)

AS

SELECT \* FROM users, teach, subjects where users.UserID = Teach.UserID and Teach.SubjectID = Subjects.SubjectID;

### **Tạo Function**

--1.Viết một hàm f\_DiemTrungBinh trả về một bảng chi tiết họ tên, lớp, số điểm trung bình của 1 học sinh

--với DiemTrungBinh = ((Coef\_one \* 1 + Coef\_two \* 2 + Coef\_three \* 3)/6)

--với UserID là tham số truyền vào

create function f\_DiemTrungBinh (@UserID varchar(4), @SubjectID varchar(10))

returns float

as

begin

declare @DiemTrungBinh float

select @DiemTrungBinh = ((Coef\_one \* 1 + Coef\_two \* 2 + Coef\_three \* 3)/6)

from study where UserID=@UserID and SubjectID = @SubjectID

return @DiemTrungBinh

end

--2.Viết một hàm f\_DiemTrungBinh trả về một bảng chi tiết họ tên, lớp, điểm số 1, điểm số 2, điểm số 3 và số điểm trung bình của 1 học sinh

--với UserID là tham số truyền vào

create function f\_Diem(@UserID nvarchar(4))

returns @Diem table(HoVaTen nvarchar(60), Mon nvarchar(20), Lop nvarchar(10), Coef\_one float, Coef\_tow float, Coef\_three float, DiemTrungBinh float)

as

begin

insert into @Diem

select UserName, SubjectID, ClassName, Coef\_one,

Coef\_two, Coef\_three, dbo.f\_DiemTrungBinh(UserID, SubjectID)

from v\_scoresOfStudent where UserID=@UserID;

return

end

-- Hàm đếm sĩ số của lớp tham số truyền vào là 1 ID lớp

create function f\_SiSoHocSinh(@ClassID varchar(10))

returns int

begin

declare @SiSo int;

select @SiSo = count(\*) from users where ClassID = @ClassID;

return @SiSo

end

-- Hàm tính điểm tổng kết đầu vào 1 UserID

create function get\_avgAllScore(@UserID varchar(4))

returns float as

begin

declare @x float;

select @x = (sum(Summary)/count(\*)) from study where UserID = @UserID;

return @x;

end

-- Hàm Đánh giá

create function f\_evaluate(@UserID varchar(4))

returns nvarchar(50) as

begin

declare @eva nvarchar(50), @UserRole nvarchar(20);

select @UserRole = UserRole from users where UserID = @UserID;

declare @score float;

if (@UserRole = N'Học sinh')

begin

set @score = dbo.get\_avgAllScore(@UserID);

select @eva = (case

when (@score > 9.5 and not EXISTS(select \* from study where UserID = @UserID and Summary < 9)) Then N'Học sinh xuất sắc'

when (@score > 8.0 and not EXISTS(select \* from study where UserID = @UserID and Summary < 7)) Then N'Học sinh giỏi'

when (@score > 6.5 and not EXISTS(select \* from study where UserID = @UserID and Summary < 5)) Then N'Học sinh tiên tiến'

when (@score > 3.5 and not EXISTS(select \* from study where UserID = @UserID and Summary < 0)) Then N'Học sinh trung bình'

when (@score > 0) Then N'Học sinh yếu'

else N'không đủ điều kiện đánh giá'

end);

end

else if (@UserRole = N'Giáo viên')

begin

select @score =

(select count(\*) from v\_studentsOfTeachers

where teacherID = @UserID and summary is not null and summary > 6.5) / count(\*)

from v\_studentsOfTeachers where teacherID = @UserID;

if(@score > 0.75 and

not EXISTS(select \* from v\_studentsOfTeachers

where teacherID = @UserID and summary < 3.5))

begin

set @eva = N'Giáo viên giỏi';

end

else set @eva = N'Giáo viên tiên tiến'

end

return @eva;

end

--Viết hàm tách tên từ chuỗi Họ tên người dùng --

CREATE FUNCTION TACHTEN(@ProName nvarchar(60))

RETURNS nvarchar(30)

AS

BEGIN

DECLARE @ten varchar(10), @L int, @i int,@j int,@kt varchar(10)

SET @L=LEN(@ProName)

SET @i=1

WHILE @i<=@L

BEGIN

SET @kt=SUBSTRING(@ProName,@i,1)

IF @kt='' SET @j=@i

SET @i=@i+1

END

SET @ten=SUBSTRING(@ProName,@j+1,10)

RETURN @ten

END

select dbo.TACHTEN(N'TRAN DUC) as táchten

drop function tachten

--:Viết hàm đọc tên Khối ra thành chữ tương ứng

CREATE FUNCTION DOCKHOINGUYEN(@ClassGrade int)

RETURNS nvarchar(20)

AS

BEGIN

DECLARE @khoichu nvarchar(20)

SET @khoichu=CASE

WHEN @ClassGrade=10 THEN N'Khối mười'

WHEN @ClassGrade=11 THEN N'Khối mười một'

WHEN @ClassGrade=12 THEN N'Khối mười hai'

END

RETURN @khoichu

end

### **Tạo Trigger**

-- Thêm account mới -> thêm 1 profile mới

-- Thêm Account học sinh mới -> thêm tất cả bảng điểm

create trigger add\_user

on v\_users\_profiles

instead of insert as

begin

declare @UserID varchar(4), @ClassID varchar(10), @UserRole Nvarchar(20);

select @UserID = UserID, @ClassID = ClassID, @UserRole = UserRole from inserted;

insert into users(UserID, UserName, UserPassword, UserEmail, UserCode, UserRole, ClassID)

select UserID, UserName, UserPassword, UserEmail, UserCode, UserRole, ClassID from inserted;

insert into profiles (UserID, ProName, ProPhone, ProAddress, ProGender, ProBirth, ProAva)

select UserID, ProName, ProPhone, ProAddress, ProGender, ProBirth, ProAva from inserted;

if(@UserRole = N'Học sinh')

begin

insert into study(UserID, SubjectID) select UserID, SubjectID

from v\_subjectsOfClass as sc, inserted where inserted.ClassID = sc.ClassID;

end

end

-- Xóa user -> xóa profile, xóa message

-- Học sinh -> xóa điểm

-- Giáo viên -> xóa teach

create trigger del\_user

on users

instead of delete as

begin

declare @UserID varchar(4), @Role Nvarchar(20);

select @UserID = UserID, @Role = UserRole from deleted;

delete from profiles where UserID in (select UserID from deleted);

delete from messenger where FromID in (select UserID from deleted) or ToID in (select UserID from deleted);

if @Role = N'Học sinh'

begin

delete from study where UserID in (select UserID from deleted);

end

else if @Role = N'Giáo viên'

begin

delete from teach where UserID in (select UserID from deleted);

end

delete from users where UserID in (select UserID from deleted);

end

-- Sua diem -> sua lai tong diem

create trigger update\_scores

on study

after update as

begin

if (update(Coef\_one) or update(Coef\_two) or update(Coef\_three) or update(summary))

update study set summary = dbo.f\_DiemTrungBinh(inserted.UserID, inserted.SubjectID)

from inserted where study.UserID = inserted.UserID and study.SubjectID = inserted.SubjectID;

end

-- Xóa Thông tin cá nhân -> xóa người dùng

create trigger update\_users

on users

for update as

begin

if update(UserName)

begin

print N'Không thể thay đổi tên tài khoản';

rollback transaction

end

end

--: Viết trigger insert bảng teach,

--nếu người dùng có mã userrole là giáo viên thì thông báo insert thành công

--còn không thì thông báo lỗi.

create or alter trigger InsertTeach

on teach

for insert

as

begin

declare @userrole nvarchar(10)

select @userrole = users.userrole from users,inserted

where users.userid = inserted.userid

if(@userrole like N'Giáo viên')

begin

print N'thêm giáo viên thành công'

end

else

begin

print N'Người dùng không phải giáo viên'

rollback transaction

end

end

select\* from users

select\* from profiles

insert into teach values ('0002', 'van11', '2021A2')

, ('1002', 'van11', '2021A2')

delete from teach

select\* from teach

--: Viết trigger insert bảng teach,

--nếu người dùng có mã userrole là giáo viên thì thông báo insert thành công

--còn không thì thông báo lỗi.

create or alter trigger InsertStudy

on study

for insert

as

begin

if(not exists(select \* from users where UserID in (select UserID from inserted) and UserRole = N'Học sinh'))

begin

print N'Người dùng không phải là học sinh';

rollback transaction

end

else

begin

print N'Thêm học sinh thành công';

end

end

### **Tạo Stored procedures**

--1. Thủ tục lấy ra sĩ số của các lớp có sĩ số > x(với x là đầu vào) dưới dạng con trỏ

Create procedure sp\_SiSoNhieu

@SiSo int, @dsss cursor varying output

as begin

set @dsss=cursor for

select ClassID, ClassName, ClassGrade, SiSo = dbo.f\_SiSoHocSinh(ClassID) from Class where dbo.f\_SiSoHocSinh(ClassID) >= @SiSo order by SiSo asc

open @dsss

end

--2.Viết 1 thủ tục số lượng tin nhắn đã đc gửi đi & nhận về từ ngày nào đó -> ngày

Create proc sp\_SoLuongTinNhan

@UserID nvarchar(4), @NgayDau date, @NgayCuoi date, @TongTinNhan int out

as

begin

Select @TongTinNhan = Count(\*) from messenger

where FromID = @UserID and MessTime>=@ngaydau and MessTime<=@ngaycuoi

end

-- Thủ tục cập nhật đánh giá cho Học sinh theo lớp

create proc sp\_updateEvaluate\_class @ClassID varchar(10) as

begin

update profiles set evaluate = dbo.f\_evaluate(profiles.UserID) from users

where (users.ClassID = @ClassID) and profiles.UserID = users.UserID

end

-- Thủ tục cập nhật đánh giá cho Học sinh theo lớp

create proc sp\_updateEvaluate\_user @UserID varchar(4) as

begin

update profiles set evaluate = dbo.f\_evaluate(profiles.UserID)

where (profiles.UserID = @UserID)

end

-- Thủ tục lấy ra tin nhắn với đầu vào là một chuỗi

create proc sp\_searchMatchMess @search Nvarchar(50), @count int output, @matchedList cursor varying output

as begin

set @matchedList = cursor for

select \* from messenger where MessContent like '%'+@search+'%';

select @count = count(\*) from messenger where MessContent like '%'+@search+'%';

open @matchedList;

end

--Tạo thủ tục bổ sung dữ liệu cho bảng users--

select\*from Users

CREATE PROCEDURE INSERT\_users

@UserID varchar(4), @UserName Nvarchar(60), @UserPassword varchar(255), @UserEmail varchar(100), @UserStatus int,

@UserCode varchar(8), @UserRole Nvarchar(20), @ClassID varchar(10)

AS

BEGIN

IF EXISTS(SELECT \* FROM users WHERE UserID = @UserID )

BEGIN

PRINT N'users này đã có, nhập mã khác'

RETURN -1

END

IF NOT EXISTS (SELECT \* FROM users

WHERE ClassID=@ClassID)

BEGIN

PRINT N'Mã lớp này không tồn tại'

RETURN -1

END

INSERT INTO users

VALUES (@UserID, @UserName, @UserPassword, @UserEmail, @UserStatus, @UserCode, @UserRole, @ClassID)

END

exec INSERT\_users '1000', N'bacaiten', N'TRẦN THỊ PHÚC', 'Tranphucnm@gmail.com', 1, NULL, N' học sinh', '2020A2'

select \* from users where UserID= '1000'

-- Tạo thủ tục của học sinh có điểm trung bình cao nhất với tham số truyền vào là môn học

create proc DiemTBcaonhat @tenmon nvarchar(10) , @diem float output, @ten nvarchar(30) output

as

begin

select @diem = max(summary) from study,subjects

where study.SubjectID=subjects.subjectid

and subjects.subjectname = @tenmon

select @ten = (select proname from profiles,study

where profiles.userid = study.userid

and study.Summary=@diem)

end

declare @DTB float

declare @hoten nvarchar(30)

exec DiemTBcaonhat N'anh 10' , @dtb out , @hoten out

print N'Học sinh ' + cast(@hoten as nvarchar(30)) + N' có DTB cao nhất môn là: ' + cast(@dtb as char(4))

select\*from subjects

## **Chèn dữ liệu**

**KẾT LUẬN**

Sau khi hoàn thành bài tập lớn này, em đã thu được những kiến thức về môn học cơ sở dữ liệu cũng như áp dụng kiến thức đó vào phân tích các bài toán thực tế.

**Bài tập đã làm được các nội dung sau đây:**

Bài toán 1: Xác định được các mối liên kết và ràng buộc dữ liệu. Xây dựng được mô hình thực thể liên kết ER. Sau đó từ mô hình này thực hiển chuyển sang mô hình quan hệ.

Bài toán 2: Từ mô hình quan hệ thu được từ bài toán 1. Tiến hành tìm các phụ thuộc hàm, tìm các khóa. Sau đó chuẩn hóa mô hình về dang chuẩn 3NF hoặc BCNF.

Bài toán 3: Từ các dữ liệu đã có. Truy vấn cơ sở dữ liệu bằng ngôn ngữ SQL qua các câu lệnh

**Từ đó kết luận được:**

Cơ sở dữ liệu là một hệ thống các thống tin có cấu trúc, được lưu trữ trên các thiết bị nhằm thỏa mãn nhu cầu khai thác thông tin từ thực tiễn đồng thời của nhiều người sử dụng hay nhiều chương trình ứng dụng chạy cùng một lúc với những mục đích khác nhau. Vì cậy để đảm bảo cơ sở dữ liệu được chuẩn xác cần phải đi thực nghiệm thực thế các hệ thống quản lý tại nơi đó rồi từ đó phân tích và thiết kế ra được cơ sở dữ liệu phù hợp. Thực thể càng nhiều cơ sở dữ liệu càng được thể hiện hệ thống chặt chẽ. Xác định các phụ thuộc hàm phải đảm bảm các yêu cầu về dàng buộc dữ liệu để tránh bị vi phạm ràng buộc toàn vẹn trong thiết kế cơ dữ liệu.

Việc sử dụng thông thạo các câu truy vấn SQL giúp cho việc truy vấn dữ liệu theo yêu cầu sẽ rất dễ dàng.

**Kiến nghị:**

Trên đây là những kết quả đầu tiên mà em học tập và làm dựa trên yêu cầu bài tập lớn môn Cơ sở dữ liệu. Trong tương lai, từ môn học này kết hợp với môn học liên quan em sẽ phân tích rồi thiết kế được một số cơ sở dữ liệu hữu dụng cho thực tế.

# TÀI LIỆU THAM KHẢO

[1] Bài giảng, Cơ sở dữ liệu, Bộ môn Hệ thống thông tin, Trường Đại học Thủy Lợi.

[2] Phần I, Phần II, Cơ sở dữ liệu, Bộ môn Công nghệ phần mềm, Trường Đại học Thủy Lợi

[3] Bài giảng, Hệ quản trị cơ sở dữ liệu, Bộ môn Hệ thống thông tin, Trường Đại học Thủy Lợi.

[4] Marcilina S. Garcia, Jamie Reding, Edward Whalen, Steve Adrien DeLuca, SQL Server 2000 Administrator’s Companion, Microsoft Press, 2000.

