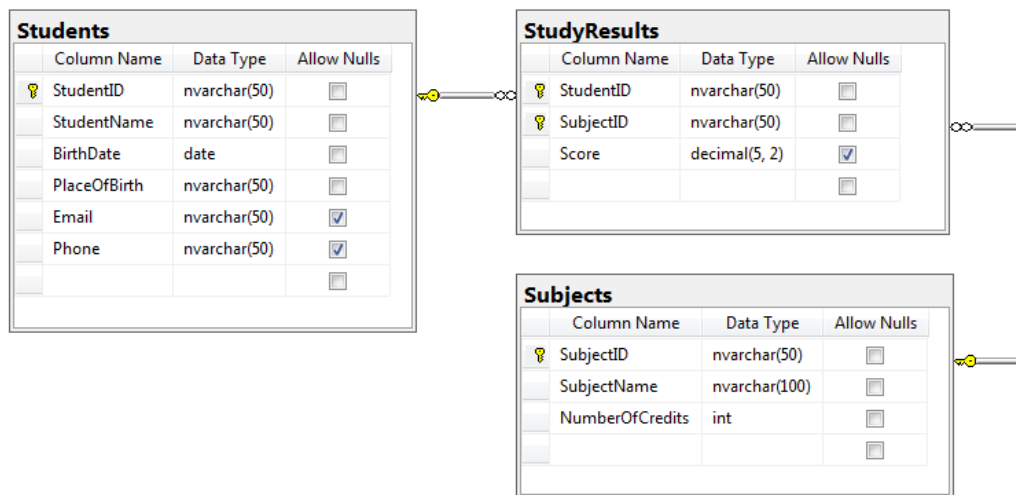


- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Tạo cơ sở dữ liệu có tên **EXAM_<Mã_sinh_viên>** (ví dụ: **EXAM_19T1080001**) và cài đặt các bảng dữ liệu cho cơ sở dữ liệu này theo sơ đồ dưới đây:



(Lưu ý: Sinh viên tự nhập dữ liệu cho các bảng trên)

Câu 1:

- (0,5 đ) Tạo khung nhìn **view_Students_2000** có chức năng lấy được thông tin của các sinh viên sinh năm 2000 tại Huế.
- (0,5 đ) Thông qua khung nhìn trên, cập nhật nơi sinh của các sinh viên sinh năm 2000 tại Huế thành Thừa Thiên Huế.

Câu 2:

- (1,5 đ) Tạo thủ tục

```
proc_AddStudyResult
    @StudentID nvarchar(50),
    @SubjectID nvarchar(50),
```

@Score decimal(5,2),

@Result nvarchar(255) OUTPUT

có chức năng bổ sung kết quả học tập một môn học của một sinh viên. Thủ tục phải kiểm tra được tính hợp lệ của dữ liệu đầu vào, trong đó điểm thi phải là giá trị từ 0 đến 10. Tham số đầu ra **@Result** trả về chuỗi rỗng nếu bổ sung thành công, và trả về chuỗi khác rỗng cho biết nguyên nhân trong trường hợp bổ sung không thành công.

b. (1 đ) Tạo thủ tục

proc_DeleteStudent @StudentID nvarchar(50)

có chức năng xóa một sinh viên dựa vào mã của sinh viên. Lưu ý không được phép xóa sinh viên nếu như sinh viên đã có điểm thi.

c. (1,5 đ) Tạo thủ tục

proc_ListStudents

@Page int,

@PageSize int,

@RowCount int output

có chức năng hiển thị danh sách sinh viên dưới dạng phân trang. Trong đó, **@Page** là trang cần hiển thị và **@PageSize** là số dòng dữ liệu được hiển thị trên mỗi trang. Tham số đầu ra **@RowCount** cho biết tổng số sinh viên hiện có.

d. (1 đ) Tạo thủ tục

proc_CountStudentsByBirthMonth @Year int

có chức năng thống kê số lượng sinh viên sinh trong mỗi tháng của năm **@Year**. Yêu cầu kết quả thống kê phải hiển thị đủ 12 tháng của năm.

Câu 3: (1,5 đ) Viết trigger **trg_Subjects_Update** để xử lý trường hợp khi cập nhật dữ liệu của cột **NumberOfCredits** trong bảng **Subjects**, trong đó chỉ cho phép cập nhật nếu số tín chỉ (**NumberOfCredits**) của môn học phải là giá trị từ 2 đến 4.

Câu 4:

a. (1đ) Viết hàm **func_CountStudents(@PlaceOfBirth nvarchar(50))** có chức năng trả về giá trị cho biết số lượng sinh viên được sinh tại **@PlaceOfBirth**

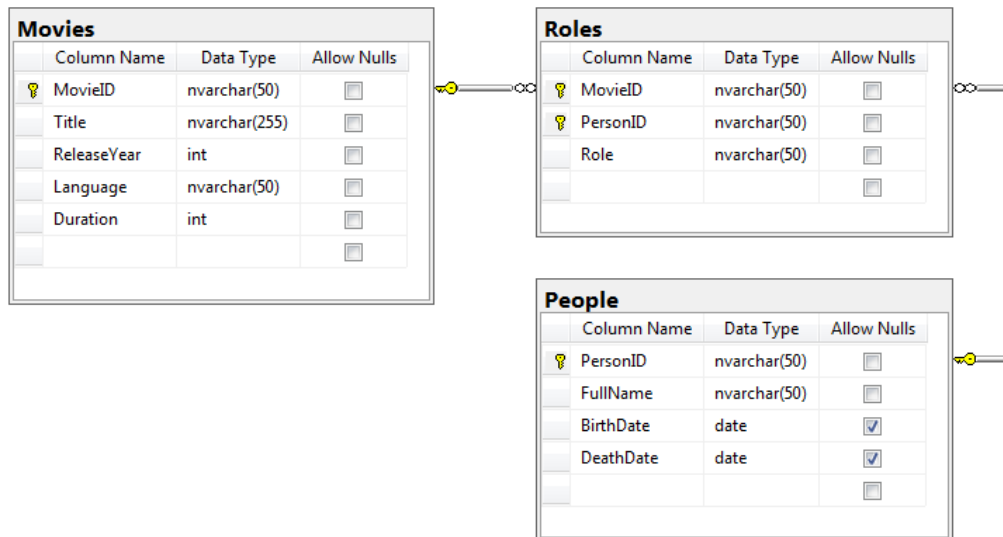
b. (1,5 đ) Viết hàm **func_GetStudyResults(@StudentID nvarchar(50))** có chức năng trả về một bảng cho biết kết quả học các môn học của sinh viên có mã là **@StudentID**.

- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Tạo cơ sở dữ liệu có tên **EXAM_<Mã_sinh_viên>** (ví dụ: **EXAM_19T1080001**) và cài đặt các bảng dữ liệu cho cơ sở dữ liệu này theo sơ đồ dưới đây:



(Lưu ý: Sinh viên tự nhập dữ liệu cho các bảng trên)

Câu 1:

- (0.5 đ) Tạo khung nhìn **view_Movies_2010** có chức năng lấy thông tin của các phim phát hành năm 2010 có ngôn ngữ là *Chinese*.
- (0.5 đ) Thông qua khung nhìn trên, cập nhật ngôn ngữ của các phim phát hành năm 2010 có ngôn ngữ là *Chinese* thành *Japanese*.

Câu 2:

- (1,5 đ) Tạo thủ tục:

```
proc_Roles_Insert
    @MovieID nvarchar(50),
```

```
@PersonID nvarchar(50),  
@Role nvarchar(50),  
@Result nvarchar(255) OUTPUT
```

Thực hiện chức năng bổ sung thêm một người tham gia trong phim. Thủ tục phải kiểm tra được tính hợp lệ của dữ liệu đầu vào, trong đó vai trò (Role) của người tham gia trong phim phải là một trong số các chuỗi giá trị *Director, Actor, Actress*. Tham số đầu ra @Result trả về chuỗi rỗng nếu bổ sung thành công, và trả về chuỗi khác rỗng cho biết lý do nếu bổ sung không thành công.

b. (1 đ) Tạo thủ tục

```
proc_People_UpdateDeathDate  
@PersonID nvarchar(50),  
@DeathDate date
```

có chức năng cập nhật ngày qua đời (DeathDate) của một người. Lưu ý, chỉ cho phép cập nhật nếu ngày qua đời phải sau ngày sinh.

c. (1,5 đ) Tạo thủ tục:

```
proc_ListMovies  
@Page int,  
@PageSize int,  
@RowCount int output
```

có chức năng hiển thị danh sách các phim dưới dạng phân trang dữ liệu. Trong đó, @Page là trang cần hiển thị và @PageSize là số dòng dữ liệu được hiển thị trên mỗi trang. Tham số đầu ra @RowCount cho biết tổng số lượng phim hiện có.

d. (1 đ) Viết thủ tục

```
proc_SummaryByYears  
@FromYear int,  
@ToYear int
```

có chức năng thống kê số lượng phim được phát hành trong mỗi năm từ năm @FromYear đến năm @ToYear. Yêu cầu kết quả thống kê phải hiển thị đủ các năm trong khoảng thời gian cần thống kê.

Câu 3: (1,5 đ) Viết trigger **trg_Role_Update** để xử lý trường hợp khi cập nhật cột **Role** trong bảng **Roles**. Nếu vai trò (Role) của người tham gia trong phim không phải là trong số các giá trị *Director, Actor, Actress* thì không cho phép cập nhật.

Câu 4:

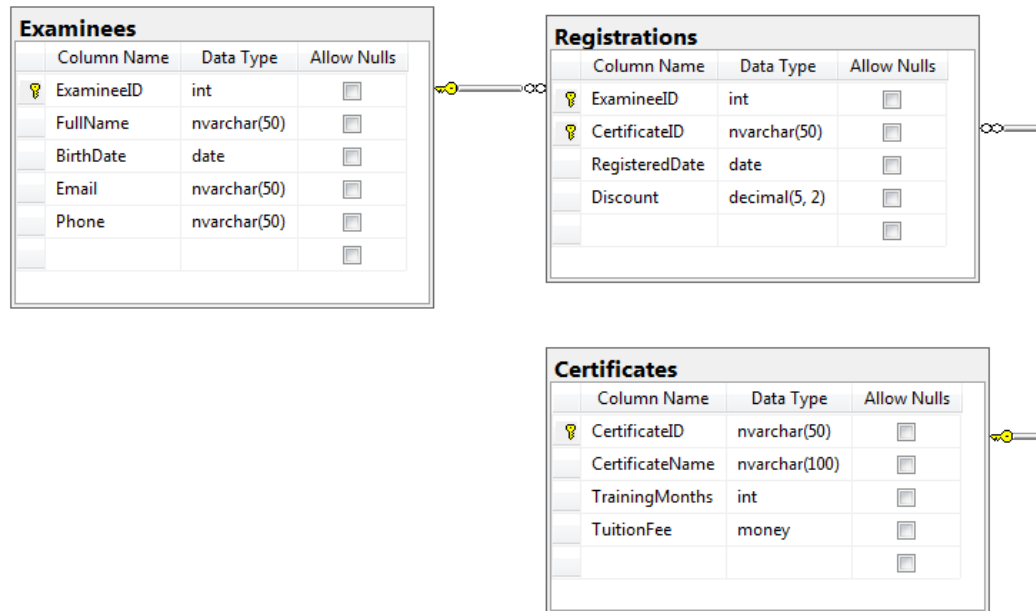
- a. (1 đ) Viết hàm **func_CountMoviesByLanguage(@Language nvarchar(50))** có chức năng trả về giá trị cho biết số lượng phim có ngôn ngữ là **@Language**.
- b. (1,5 đ) Viết hàm **func_GetRoles(@MovieID nvarchar(50))** có chức năng trả về một bảng cho biết họ tên và vai trò của những người có tham gia trong phim có mã là **@MovieID**.

- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Tạo cơ sở dữ liệu có tên **EXAM_<Mã_sinh_viên>** (ví dụ: **EXAM_19T1080001**) và cài đặt các bảng dữ liệu cho cơ sở dữ liệu này theo sơ đồ dưới đây:



(Lưu ý: Sinh viên tự nhập dữ liệu cho các bảng trên)

Câu 1:

- (0,5 đ) Tạo khung nhìn **view_Certificates_InAMonth** có chức năng lấy thông tin của các chứng chỉ có thời gian đào tạo trong vòng 1 tháng và có mức phí nhỏ hơn 1500000.
- (0,5 đ) Thông qua khung nhìn trên, cập nhật phí đào tạo của các chứng chỉ có thời gian đào tạo trong 1 tháng và có mức phí nhỏ hơn 1500000 thành 1500000.

Câu 2:

a. (1,5 đ) Viết thủ tục

```
proc_AddRegistration  
    @ExamineeID int,  
    @CertificateID nvarchar(50),  
    @RegisteredDate date,  
    @Discount decimal(5,2),  
    @Result nvarchar(255) OUTPUT
```

có chức năng bổ sung một bản ghi cho bảng **Registrations**. Thủ tục phải kiểm tra được tính hợp lệ của dữ liệu đầu vào, trong đó mức miễn giảm (**Discount**) phải là giá trị lớn hơn hoặc bằng 0 và nhỏ hơn hoặc bằng 1. Tham số đầu ra **@Result** trả về chuỗi rỗng nếu bổ sung thành công, và trả về chuỗi khác rỗng cho biết nguyên nhân nếu bổ sung không thành công.

b. (1 đ) Tạo thủ tục **proc_Examinees_Delete @ExamineeID** có chức năng xóa người dự thi có mã là **@ExamineeID**. Lưu ý không được xóa nếu người này đã có đăng ký dự thi ít nhất một chứng chỉ.

c. (1,5 đ) Tạo thủ tục:

```
proc_ListExaminees  
    @Page int,  
    @PageSize int,  
    @RowCount int output
```

có chức năng hiển thị danh sách người dự thi dưới dạng phân trang dữ liệu. Trong đó, **@Page** là trang cần hiển thị và **@PageSize** là số dòng dữ liệu được hiển thị trên mỗi trang. Tham số đầu ra **@RowCount** cho biết tổng số người dự thi hiện có.

d. (1 đ) Viết thủ tục

```
proc_SummaryByMonth @Year int
```

có chức năng thống kê số lượng người đăng ký dự thi trong mỗi tháng của năm **@Year**. Yêu cầu kết quả thống kê phải hiển thị đủ 12 tháng của năm.

Câu 3: (1,5 đ) Viết trigger **trg_Registrations_Update** để xử lý trong trường hợp cập nhật giá trị của cột **Discount** trong của bảng **Registrations**. Nếu mức miễn giảm nhỏ hơn 0 hoặc lớn hơn 1 thì không cho phép cập nhật dữ liệu.

Câu 4:

a. (1 đ) Viết hàm **func_CountRegistrations(@Month int)** có chức năng trả về giá trị cho biết số lượng đăng ký thi trong tháng **@Month**.

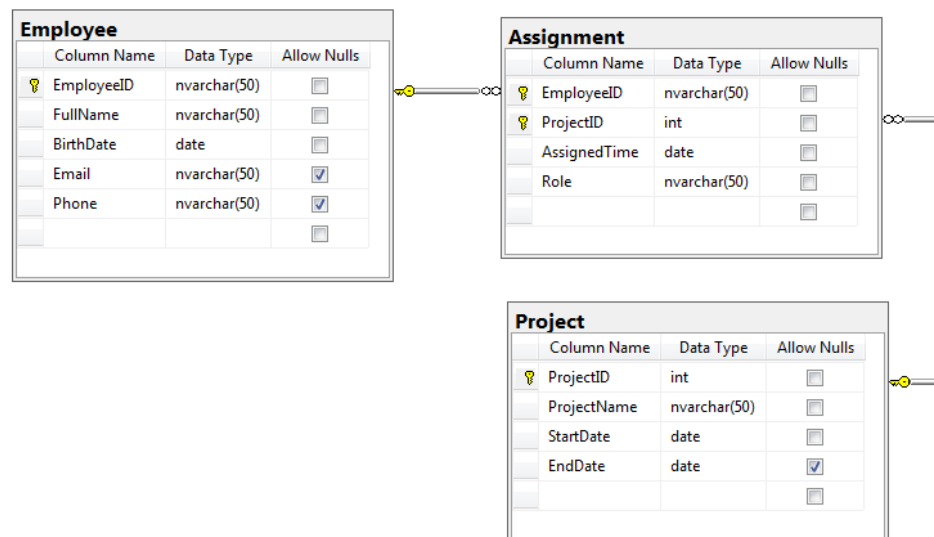
b. (1,5 đ) Viết hàm **func_GetExaminees(@CertificateID nvarchar(50))** có chức năng trả về bảng cho biết danh sách những người đăng ký thi chứng chỉ có mã là **@CertificateID**.

- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Tạo cơ sở dữ liệu có tên **EXAM_<Mã_sinh_viên>** (ví dụ: **EXAM_19T1080001**) và cài đặt các bảng dữ liệu cho cơ sở dữ liệu này theo sơ đồ dưới đây:



(Lưu ý: Sinh viên tự nhập dữ liệu cho các bảng trên)

Câu 1:

- (0,5 đ) Tạo khung nhìn **view_Project_2021** có chức năng lấy thông tin các dự án bắt đầu trong năm 2021 và thời điểm kết thúc khác **NULL**.
- (0,5 đ) Thông qua khung nhìn trên, hãy cập nhật thời điểm kết thúc của các dự án bắt đầu trong năm 2021 và thời điểm kết thúc khác **NULL** thành **NULL**.

Câu 2:

- (1,5 đ) Viết thủ tục
proc_Project_Insert
@ProjectID int,

```

@ProjectName nvarchar(50),
@StartDate date,
@EndDate date = NULL,
@Result nvarchar(255) OUTPUT

```

có chức năng bổ sung thêm một dự án mới vào bảng Project. Thủ tục phải kiểm tra được tính hợp lệ của dữ liệu đầu vào, trong đó nếu như dự án đã có thời điểm kết thúc thì thời điểm kết thúc phải sau thời điểm bắt đầu của dự án. Tham số đầu ra **@Result** trả về chuỗi rỗng nếu bổ sung thành công, và trả về chuỗi khác rỗng cho biết lý do trong trường hợp bổ sung không thành công.

b. (1 đ) Viết thủ tục

```

proc_Project_UpdateEndDate
    @ProjectID int,
    @EndDate date

```

có chức năng cập nhật thời điểm kết thúc của dự án có mã là **@ProjectID**. Lưu ý chỉ được cập nhật nếu như thời điểm kết thúc phải sau thời điểm bắt đầu của dự án.

c. (1,5 đ) Tạo thủ tục:

```

proc_ListEmployees
    @Page int,
    @PageSize int,
    @RowCount int output

```

có chức năng hiển thị danh sách nhân viên dưới dạng phân trang dữ liệu. Trong đó, **@Page** là trang cần hiển thị và **@PageSize** là số dòng dữ liệu được hiển thị trên mỗi trang. Tham số đầu ra **@RowCount** cho biết tổng số lượng nhân viên hiện có.

d. (1 đ) Viết thủ tục

```

proc_SummaryByMonth @Year int

```

có chức năng thống kê số lượng dự án bắt đầu trong mỗi tháng của năm **@Year**. Yêu cầu kết quả thống kê phải hiển thị đủ 12 tháng của năm.

Câu 3: (1đ) Viết trigger trg_Assignment_Insert để xử lý trường hợp bổ sung dữ liệu vào bảng Assignment theo yêu cầu: Nếu thời điểm giao việc của dự án cho nhân viên nằm sau thời điểm kết thúc của dự án thì không cho phép bổ sung dữ liệu.

Câu 4:

a. (1đ) Viết hàm **func_CountAssignment(@ProjectID int)** có chức năng trả về giá trị cho biết số lượng nhân viên được phân công thực hiện dự án có mã là **@ProjectID**.

b. (1,5 đ)Viết hàm **func_GetAssignments(@ProjectID int)** có chức năng trả về bảng cho biết họ tên, ngày sinh, email, điện thoại, thời điểm giao việc và vai trò của các nhân viên được phân công thực hiện dự án có mã là **@ProjectID**.

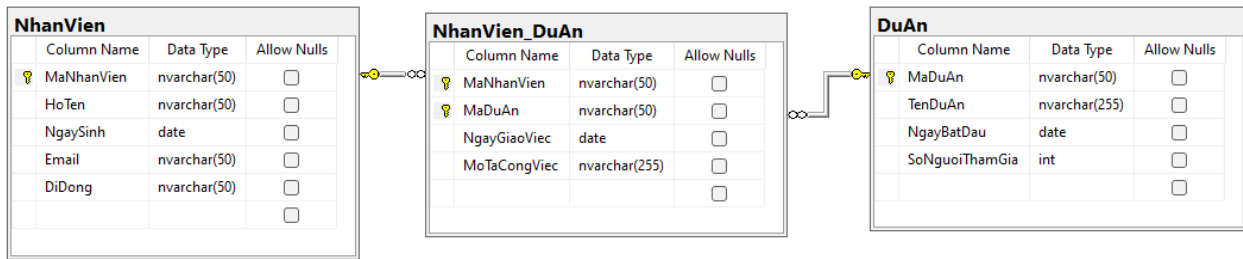
- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Câu 1 (1.0 điểm):

a. Tạo cơ sở dữ liệu có tên là **MãSinhViên** (ví dụ: **20T1020001**). Cài đặt các bảng cho cơ sở dữ liệu theo sơ đồ dưới đây:



b. Nhập dữ liệu cho các bảng như sau:

Bảng NhanVien

MaNhanVien	HoTen	NgaySinh	Email	DiDong
NV001	Nguyễn Thanh An	1980-12-01	thanhan@gmail.com	0914422578
NV002	Trần Chí Hiếu	1985-05-17	hieu85@gmail.com	0987454125
NV003	Vũ Thành Chung	1986-11-20	chungvt@gmail.com	0935254771
NV005	Lê Thị Hải Yến	1986-08-14	lthyen@gmail.com	0983120547

Bảng DuAn

MaDuAn	TenDuAn	NgayBatDau	SoNguoiThamGia
DA001	SmartUni	2022-01-01	0
DA002	E-Shop	2022-05-01	0
DA003	LiteCMS	2022-09-01	0

Câu 2 (1.0 điểm): Tạo trigger có tên **trg_NhanVien_DuAn_Insert** bắt lệnh **INSERT** trên bảng **NhanVien_DuAn** sao cho mỗi lần bổ sung thêm dữ liệu cho bảng **NhanVien_DuAn** (tức là

giao cho nhân viên thực hiện dự án) thì cập nhật lại cột **SoNguoiThamGia** của bảng **DuAn** bằng đúng với số lượng nhân viên đã được giao thực hiện dự án.

Câu 3: Tạo các thủ tục sau đây

a. (1.0 điểm) **proc_NhanVien_DuAn_Insert**
 @MaNhanVien nvarchar(50),
 @MaDuAn nvarchar(50),
 @MoTaCongViec nvarchar(255),
 @KetQua nvarchar(255) output

Có chức năng bổ sung dữ liệu cho bảng **NhanVien_DuAn** nhằm giao việc cho nhân viên có mã **@MaNhanVien** thực hiện dự án có mã **@MaDuAn**. Ngày giao việc được tính là thời điểm hiện tại. Tham số đầu ra **@KetQua** trả về chuỗi rỗng nếu bổ sung thành công, ngược lại tham số này trả về chuỗi cho biết lý do vì sao không bổ sung được dữ liệu.

b. (1.0 điểm) **proc_DuAn_DanhSachNhanVien**
 @TenDuAn nvarchar(255),
 @NgayGiaoViec date

Có chức năng hiển thị danh sách các nhân viên được giao thực hiện dự án có tên **@TenDuAn** trước ngày **@NgayGiaoViec**. Thông tin hiển thị bao gồm: Mã nhân viên, Họ tên, Email, Di động, Ngày giao việc và Mô tả công việc được giao.

c. (1.5 điểm) **proc_NhanVien_TimKiem**
 @HoTen nvarchar(50) = N'',
 @Tuoi int,
 @SoLuong int output

Có chức năng tìm kiếm các nhân viên mà trong Họ tên có chứa **@HoTen** và có Tuổi lớn hơn hoặc bằng **@Tuoi**. Lưu ý, nếu tham số **@HoTen** bằng rỗng thì chỉ tìm kiếm các nhân viên có Tuổi lớn hơn hoặc bằng **@Tuoi**. Thông tin cần hiển thị bao gồm: Mã nhân viên, Họ tên, Ngày sinh, Tuổi, Email và Di động. Tham số đầu ra **@SoLuong** cho biết số lượng nhân viên được tìm thấy.

d. (1.5 điểm) **proc_ThongKeGiaoViec**
 @MaDuAn nvarchar(50),
 @TuNgay date,
 @DenNgay date

Có chức năng thống kê số lượng nhân viên được giao thực hiện dự án có mã **@MaDuAn** theo từng ngày giao việc trong khoảng thời gian **@TuNgay** đến **@DenNgay**. Yêu cầu kết quả thống kê phải hiển thị đầy đủ tất cả các ngày trong khoảng thời gian cần thống kê, những ngày không

có nhân viên được giao việc thì hiển thị với số lượng là 0. Thông tin cần hiển thị bao gồm: Ngày giao việc và Số nhân viên được giao việc.

Câu 4: Tạo các hàm sau đây:

a. (1 điểm) **func_TKeDuAn**
 @TuNam int,
 @DenNam int

Có chức năng trả về một bảng thống kê số lượng dự án được thực hiện trong mỗi năm trong khoảng thời gian từ năm **@TuNam** đến năm **@DenNam** (năm thực hiện dự án được xác định dựa vào Ngày bắt đầu của dự án). Thông tin cần hiển thị bao gồm: Năm thực hiện và Số lượng dự án.

b. (1.5 điểm) **func_TKeDuAn_DayDuCacNam**
 @TuNam int,
 @DenNam int

Có chức năng trả về một bảng thống kê số lượng dự án được thực hiện trong mỗi năm trong khoảng thời gian từ năm **@TuNam** đến năm **@DenNam** (năm thực hiện dự án được xác định dựa vào Ngày bắt đầu của dự án). Thông tin cần hiển thị bao gồm: Năm thực hiện và Số lượng dự án. Yêu cầu kết quả phải thể hiện đầy đủ tất cả các năm trong khoảng thời gian cần thống kê (tức là những năm không có dự án nào được thực hiện thì cũng hiển thị với số lượng dự án là 0).

Câu 5 (1.0 điểm) Viết các lệnh thực hiện các yêu cầu sau đây

- Tạo tài khoản có tên là **user_MãSinhViên** (ví dụ: **user_2T1020001**) với mật khẩu là **123456**
- Cho phép tài khoản trên được phép truy cập vào cơ sở dữ liệu đã tạo.
- Cấp phát cho tài khoản trên các quyền sau đây:
 - o Được phép thực hiện lệnh **SELECT** và **INSERT** trên bảng **NhanVien**
 - o Được phép sử dụng các thủ tục và hàm đã tạo ở trên

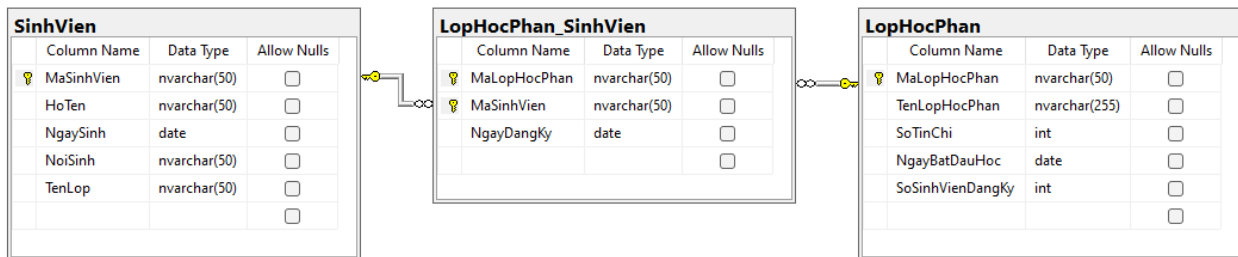
- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (Không kể thời gian phát đề)

Lưu ý:

- Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.
- Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.

Câu 1 (1.0 điểm):

a. Tạo cơ sở dữ liệu có tên là **MãSinhViên** (ví dụ: **20T1020001**). Cài đặt các bảng cho cơ sở dữ liệu theo sơ đồ dưới đây:



b. Nhập dữ liệu cho các bảng như sau:

Bảng SinhVien

MaSinhVien	HoTen	NgaySinh	NoiSinh	TenLop
SV001	Trần Thị Hoài Diệp	2001-12-05	Thừa Thiên Huế	Tin K44A
SV002	Vũ Thanh Toàn	2002-06-11	Quảng Nam	Tin K44B
SV003	Vương Đình Hòa	2021-05-12	Thừa Thiên Huế	Tin K44A
SV004	Nguyễn Văn Trung	2022-01-19	Quảng Trị	Tin K44A
SV005	Hoàng Thị Thanh Hà	2021-04-18	Thừa Thiên Huế	Tin K44B
SV006	Lê Thị Hải Yến	2022-03-24	Quảng Trị	Tin K44A

Bảng LopHocPhan

MaLopHocPhan	TenLopHocPhan	SoTinChi	NgayBatDauHoc	SoSinhVienDangKy
L0001	Hệ Quản trị CSDL - Nhóm 1	3	2022-09-01	0
L0002	Nhập môn lập trình - Nhóm 3	3	2022-09-15	0
L003	Cơ sở toán - Nhóm 1	2	2022-09-01	0
L004	Môi trường đại cương - Nhóm 1	2	2022-09-15	0
L005	Kiến trúc máy tính - Nhóm 5	3	2022-09-15	0

Câu 2 (1.0 điểm) Tạo trigger có tên **trg_LopHocPhan_SinhVien_Insert** có chức năng bắt lệnh **INSERT** trên bảng **LopHocPhan_SinhVien** sao cho mỗi lần bổ sung thêm dữ liệu cho bảng **LopHocPhan_SinhVien** (tức là thêm sinh viên đăng ký học ở lớp học phần) thì cập nhật lại giá trị của cột **SoSinhVienDangKy** trong bảng **LopHocPhan** bằng đúng với số lượng sinh viên đã đăng ký học.

Câu 3: Tạo các thủ tục sau đây

a. (1.0 điểm): **proc_LopHocPhan_SinhVien_Insert**
 @MaLopHocPhan nvarchar(50),
 @MaSinhVien nvarchar(50),
 @KetQua nvarchar(255) output

Có chức năng bổ sung thêm dữ liệu cho bảng **LopHocPhan_SinhVien** để đăng ký thêm sinh viên có mã là **@MaSinhVien** vào lớp học phần có mã là **@MaLopHocPhan**. Ngày đăng ký được tính là thời điểm hiện tại. Tham số đầu ra **@KetQua** trả về chuỗi rỗng nếu việc bổ sung là thành công, ngược lại tham số này trả về chuỗi cho biết lý do vì sao không bổ sung được dữ liệu.

b. (1.0 điểm) : **proc_LopHocPhan_SinhVien_SelectByLop**
 @MaLopHocPhan nvarchar(50),
 @TenLop nvarchar(50)

Có chức năng hiển thị danh sách sinh viên thuộc lớp **@TenLop** đã đăng ký học lớp học phần có mã **@MaLopHocPhan**. Thông tin hiển thị bao gồm Mã sinh viên, Họ tên, Ngày sinh, Nơi sinh và được sắp xếp tăng dần theo Họ tên.

c. (1.5 điểm) **proc_SinhVien_TimKiem**
 @HoTen nvarchar(50) = N'',
 @Tuoi int,
 @SoLuong int output

Có chức năng tìm kiếm các sinh viên mà trong Họ tên có chứa **@HoTen** và có Tuổi lớn hơn hoặc bằng **@Tuoi**. Lưu ý, nếu tham số **@HoTen** bằng rỗng thì chỉ tìm kiếm các sinh viên có Tuổi lớn hơn hoặc bằng **@Tuoi**. Thông tin cần hiển thị bao gồm: Mã sinh viên, Họ tên, Ngày sinh, Nơi sinh và Tên lớp. Tham số đầu ra **@SoLuong** cho biết số lượng sinh viên tìm được.

d. (1.5 điểm) **proc_ThongKeDangKyHoc**
 @MaLopHocPhan nvarchar(50),
 @TuNgay date,
 @DenNgay date

Có chức năng thống kê số lượng sinh viên đăng ký học lớp học phần có mã **@MaLopHocPhan** theo từng ngày đăng ký trong khoảng thời gian từ **@TuNgay** đến **@DenNgay**. Yêu cầu kết quả

thống kê phải hiển thị đầy đủ tất cả các ngày trong khoảng thời gian cần thống kê, những ngày không có sinh viên đăng ký thì hiển thị với số lượng đăng ký là 0. Thông cần tin hiển thị bao gồm: Ngày đăng ký và Số lượng sinh viên đăng ký.

Câu 4: Tạo các hàm sau đây

a. (1.0 điểm): **func_TkeKhoiLuongDangKyHoc**
 @MaSinhVien nvarchar(50)
 @TuNam int,
 @DenNam int

Có chức năng trả về một bảng thống kê tổng số tín chỉ mà sinh viên có mã **@MaSinhVien** đã đăng ký học trong từng năm trong khoảng thời gian từ năm **@TuNam** đến năm **@DenNam** (năm được xác định dựa vào ngày đăng ký học). Thông tin cần hiển thị bao gồm Năm và Tổng số tín chỉ.

b. (1.5 điểm): **func_TkeKhoiLuongDangKyHoc_DayDuNam**
 @MaSinhVien nvarchar(50)
 @TuNam int,
 @DenNam int

Có chức năng trả về một bảng thống kê tổng số tín chỉ mà sinh viên có mã **@MaSinhVien** đã đăng ký học trong từng năm trong khoảng thời gian từ năm **@TuNam** đến năm **@DenNam** (năm được xác định dựa vào ngày đăng ký học). Thông tin cần hiển thị bao gồm Năm và Tổng số tín chỉ. Yêu cầu kết quả thống kê phải thể hiện được đầy đủ các năm trong khoảng thời gian cần thống kê (tức là những năm mà sinh viên không đăng ký thì cũng phải hiển thị với tổng số tín chỉ đăng ký là 0).

Câu 5 (1.0 điểm) Viết các lệnh thực hiện các yêu cầu sau đây

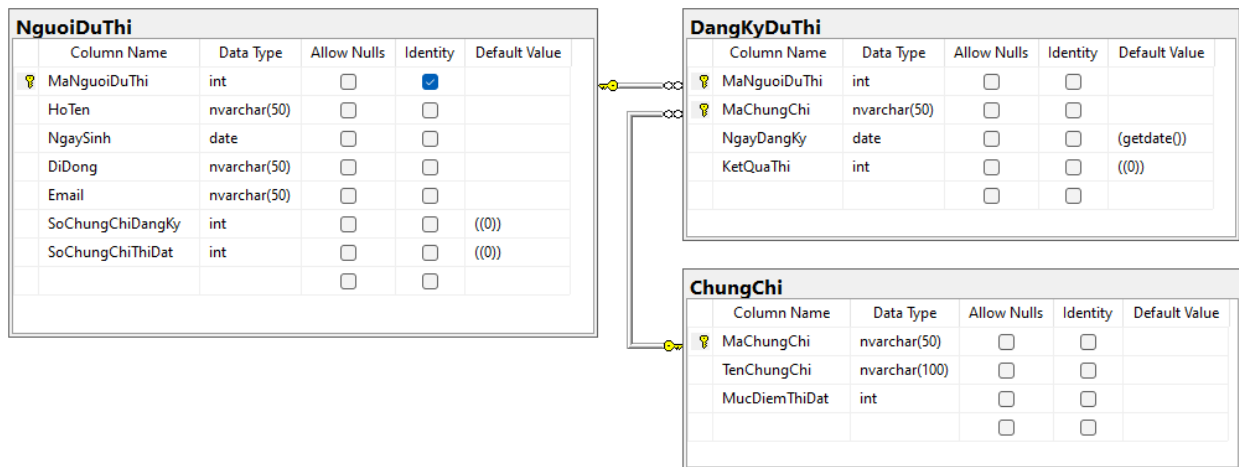
- Tạo tài khoản có tên là **user_MãSinhViên** (ví dụ: **user_2T1020001**) với mật khẩu là **123456**
- Cho phép tài khoản trên được phép truy cập vào cơ sở dữ liệu đã tạo.
- Cấp phát cho tài khoản trên các quyền sau đây:
 - o Được phép thực hiện lệnh **SELECT** và **UPDATE** trên bảng **SinhVien**
 - o Được phép sử dụng các thủ tục và hàm đã tạo ở trên

- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Câu 1: (1.0 điểm) Tạo cơ sở dữ liệu **BAITHI_MãSinhViên** (Ví dụ: **BAITHI_21T1020001**) và cài đặt các bảng dữ liệu theo sơ đồ dưới đây:



Sinh viên tự nhập dữ liệu cho 2 bảng **NgdoiDuThi** và **ChungChi** (lưu ý không nhập dữ liệu cho bảng **DangKyDuThi**)

Câu 2: Viết các trigger sau đây (giả thiết mỗi lần bổ sung hoặc cập nhật dữ liệu chỉ tác động đến tối đa 01 dòng)

- (1.0 điểm)** Trigger **trg_DangKyDuThi_Insert** bắt lệnh **INSERT** trên bảng **DangKyDuThi** sao cho mỗi khi bổ sung thêm một dòng dữ liệu trong bảng này thì cập nhật số lượng chứng chỉ đã đăng ký của người dự thi (cột **SoChungChiDangKy** trong bảng **NgdoiDuThi**).
- (1.0 điểm)** Trigger **trg_DangKyDuThi_Update** bắt lệnh **UPDATE** trên bảng **DangKyDuThi** sao cho khi cập nhật giá trị của cột **KetQuaThi** của một dòng trong bảng này thì đồng thời cập nhật số lượng chứng chỉ đã thi đạt của người dự thi (cột **SoChungChiThiDat** trong bảng **NgdoiDuThi**)

Lưu ý: Kết quả thi một chứng chỉ được coi là đạt nếu kết quả thi lớn hơn hoặc bằng mức điểm thi đạt của chứng chỉ đó (cột **MucDiemThiDat** trong bảng **ChungChi**)

Câu 3: Viết các thủ tục sau đây

a. (1.0 điểm) proc_DangKyDuThi_BoSung

```
@MaNguoiDuThi int,  
@MaChungChi nvarchar(50),  
@ThongBao nvarchar(255) output
```

Có chức năng bổ sung thêm một đăng ký dự thi vào bảng **DangKyDuThi**. Nếu bổ sung thành công, tham số **@ThongBao** trả về chuỗi rỗng, ngược lại tham số này trả về chuỗi cho biết lý do vì sao không bổ sung được dữ liệu.

b. (1.0 điểm) proc_DangKyDuThi_CapNhatKetQuaThi

```
@MaNguoiDuThi int,  
@MaChungChi nvarchar(50),  
@KetQuaThi int,  
@ThongBao nvarchar(255) output
```

Có chức năng cập nhật kết quả thi chứng chỉ của người dự thi. Trong đó lưu ý kết quả thi phải là giá trị lớn hơn hoặc bằng 0. Nếu cập nhật thành công, tham số **@ThongBao** trả về chuỗi rỗng, ngược lại tham số này trả về chuỗi cho biết lý do vì sao không cập nhật được dữ liệu.

c. (1.0 điểm) proc_NguoiDuThi_Select

```
@HoTen nvarchar(50) = N'',  
@Page int = 1,  
@PageSize int = 20,  
@RowCount int output,  
@PageCount int output
```

Có chức năng tìm kiếm và hiển thị danh sách người dự thi dưới dạng phân trang với thông tin cần hiển thị bao gồm: Họ tên, ngày sinh, di động, email, tên chứng chỉ đăng ký dự thi, ngày đăng ký dự thi và kết quả thi. Trong đó:

- Nếu **@HoTen** là rỗng thì phạm vi tìm kiếm là tất cả người dự thi, ngược lại thì tìm kiếm người dự thi theo **@HoTen** (tìm kiếm tương đối).
- **@Page** là trang cần hiển thị, **@PageSize** là số dòng dữ liệu được hiển thị trên mỗi trang
- Tham số đầu ra **@RowCount** cho biết tổng số dòng dữ liệu truy vấn được, **@PageCount** cho biết tổng số trang.

d. (1,5 điểm) proc_ThongKeSoLuongDangKyTheoNgay

```
@TuNgay date,
```

@DenNgay date

Có chức năng thống kê số lượng đăng ký dự thi của mỗi ngày trong khoảng thời gian từ ngày **@TuNgay** đến ngày **@DenNgay**. Yêu cầu kết quả thống kê phải hiển thị đầy đủ tất cả các ngày trong khoảng thời gian trên (những ngày không có người đăng ký dự thi thì hiển thị với số lượng là 0).

Câu 4: Viết các hàm sau đây

a. (1.0 điểm) func_DemSoLuongThiDat(@MaChungChi nvarchar(50)) có chức năng tính số lượng người dự thi đã thi đạt chứng chỉ có mã là **@MaChungChi**.

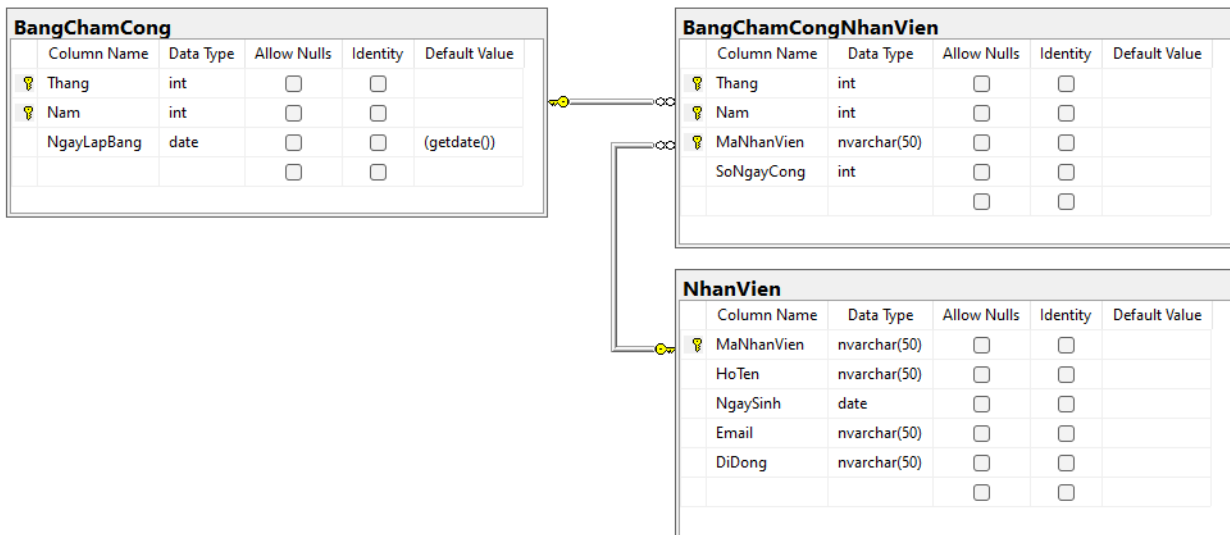
b. (1.5 điểm) func_ThongKeSoLuongDangKyTheoNgay(@TuNgay date, @DenNgay date) có chức năng trả về bảng thống kê số lượng đăng ký dự thi của mỗi ngày trong khoảng thời gian từ ngày **@TuNgay** đến ngày **@DenNgay**. Yêu cầu kết quả thống kê phải hiển thị đầy đủ tất cả các ngày trong khoảng thời gian trên (những ngày không có người đăng ký dự thi thì hiển thị với số lượng là 0).

- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Câu 1: (1.0 điểm) Tạo cơ sở dữ liệu **BAITHI_MãSinhViên** (Ví dụ: **BAITHI_21T1020001**) và cài đặt các bảng dữ liệu theo sơ đồ dưới đây:



Sinh viên tự nhập dữ liệu cho bảng **NhanVien** (lưu ý không nhập dữ liệu cho bảng **BangChamCong** và **BangChamCongNhanVien**)

Câu 2: Viết các trigger sau đây (giả thiết mỗi lần bổ sung hoặc cập nhật dữ liệu chỉ tác động đến tối đa 01 dòng)

- (1.0 điểm)** Trigger **trg_BangChamCong_Insert** bắt lệnh **INSERT** trên bảng **BangChamCong** sao cho mỗi khi bổ sung một bảng chấm công của tháng/năm thì tự động bổ sung danh sách các nhân viên được chấm công trong tháng/năm vào trong bảng **BangChamCongNhanVien** với số ngày công chính là số ngày của tháng lập bảng chấm công.
- (1.0 điểm)** Trigger **trg_BangChamCongNhanVien_Update** bắt lệnh **UPDATE** trên bảng **BangChamCongNhanVien** sao cho mỗi khi cập nhật giá trị của cột **SoNgayCong** thì kiểm tra

xem số ngày công có hợp lệ hay không. Nếu số ngày công không hợp lệ thì hủy bỏ thao tác cập nhật.

Lưu ý: Số ngày công hợp lệ nếu lớn hơn hoặc bằng 0 và nhỏ hơn hoặc bằng số ngày của tháng.

Câu 3: Viết các thủ tục sau đây

a. (1.0 điểm) proc_BangChamCong_Insert

```
@Thang int,  
@Nam int,  
@NgayLapBang date,  
@ThongBao nvarchar(255) output
```

Có chức năng bổ sung một dòng dữ liệu vào bảng **BangChamCong**. Tham số đầu ra **@ThongBao** trả về chuỗi rỗng nếu bổ sung thành công, ngược lại tham số này trả về chuỗi khác rỗng cho biết lý do vì sao không bổ sung được dữ liệu.

b. (1.0 điểm) proc_BangChamCongNhanVien_Update

```
@Thang int,  
@Nam int,  
@MaNhanVien nvarchar(50),  
@SoNgayCong int,  
@ThongBao nvarchar(255) output
```

Có chức năng cập nhật số ngày công của nhân viên có mã **@MaNhanVien** trong tháng **@Thang** và năm **@Nam**. Tham số đầu ra **@ThongBao** trả về chuỗi rỗng nếu cập nhật thành công, ngược lại tham số này trả về chuỗi khác rỗng cho biết lý do vì sao không cập nhật được dữ liệu.

c. (1.0 điểm) proc_NhanVien_Select

```
@HoTen nvarchar(50) = N'',  
@Page int = 1,  
@PageSize int = 20,  
@RowCount int output,  
@PageCount int output
```

Có chức năng tìm kiếm và hiển thị danh sách nhân viên dưới dạng phân trang. Trong đó **@HoTen** là họ tên của nhân viên cần tìm (tìm kiếm tương đối theo họ tên của nhân viên, nếu tham số này là chuỗi rỗng thì lấy tất cả nhân viên), **@Page** là trang cần hiển thị, **@PageSize** là số dòng dữ liệu được hiển thị trên mỗi trang, tham số đầu ra **@RowCount** cho biết tổng số dòng dữ liệu tìm được, tham số đầu ra **@PageCount** cho biết tổng số trang.

d. (1,5 điểm) proc_ThongKeTongNgayCong

@TuNam int,
@DenNam int

Có chức năng thống kê tổng số ngày công của từng năm trong khoảng thời gian từ năm **@TuNam** đến năm **@DenNam**. Yêu cầu kết quả phải hiển thị đủ tất cả các năm trong khoảng thời gian cần thống kê (những năm không có dữ liệu thì hiển thị với tổng số ngày công là 0)

Câu 4: Viết các hàm sau đây

a. (1.0 điểm) func_TongNgayCongCuaThang(@Thang int, @Nam int) có chức năng tính tổng số ngày công của các nhân viên trong tháng **@Thang** năm **@Nam**.

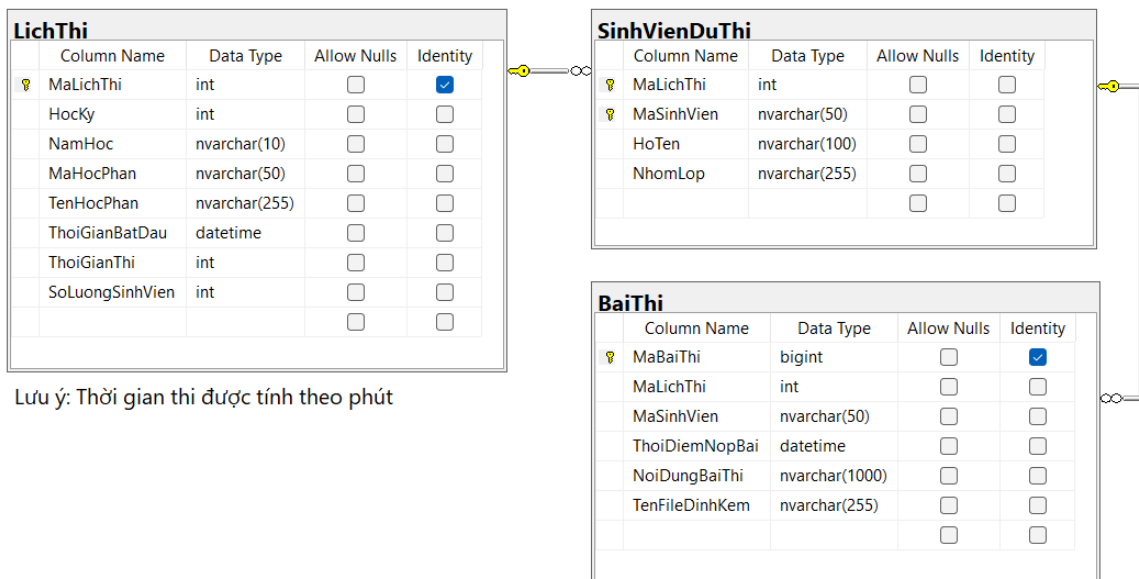
b. (1.5 điểm) func_ThongKeTongNgayCong(@TuNam int, @DenNam int) có chức năng trả về bảng thống kê tổng số ngày công của từng năm trong khoảng thời gian từ năm **@TuNam** đến năm **@DenNam**. Kết quả thống kê phải đầy đủ tất cả các năm trong khoảng thời gian yêu cầu (những năm không có dữ liệu thì hiển thị với tổng số ngày công là 0).

- Tên học phần: Các hệ quản trị cơ sở dữ liệu
- Mã học phần: TIN3053 Số tín chỉ: 3
- Thời gian làm bài: 120 phút (*Không kể thời gian phát đề*)

Lưu ý:

- *Không được sử dụng: Internet, điện thoại, các thiết bị nhớ ngoài.*
- *Kết thúc làm bài và nộp bài: Sao lưu cơ sở dữ liệu với tên file sao lưu là **BAITHI_MãSinhViên.bak** (ví dụ: **BAITHI_21T1020001.bak**), nộp file sao lưu và file script bài làm.*

Câu 1: (1.0 điểm) Tạo cơ sở dữ liệu **BAITHI_MãSinhViên** (Ví dụ: **BAITHI_22T1020001**) và cài đặt các bảng theo sơ đồ dưới đây:



Lưu ý: Sinh viên tự nhập dữ liệu cho bảng

Câu 2 (1.0 điểm): Tạo trigger có tên **trg_SinhVienDuThi_Insert** bắt lệnh **INSERT** trên bảng **SinhVienDuThi** sao cho mỗi lần bổ sung thêm dữ liệu cho bảng này thì cập nhập lại cột **SoLuongSinhVien** của bảng **LichThi** bằng đúng với số lượng sinh viên đã bố trí dự thi theo lịch thi.

Câu 3: Tạo các thủ tục sau đây:

a. (1.0 điểm) **proc_BaiThi_Insert**
 @MaLichThi int,
 @MaSinhVien nvarchar(50),


```

@NoiDungBaiThi nvarchar(1000),
@TenFileDinhKem nvarchar(255),
@KetQua bigint output

```

Có chức năng bổ sung thêm dữ liệu cho bảng **BaiThi** (tức là ghi nhận sinh viên nộp bài thi). Tham số đầu ra **@KetQua** trả về giá trị theo qui ước sau:

- Nếu bổ sung thành công, giá trị của tham số chính là mã của bài thi được bổ sung.
- Nếu bổ sung không thành công, giá trị của tham số trả về là giá trị âm cho biết lý do tại sao không bổ sung được bài thi (giá trị của mỗi trường hợp do sinh viên tự qui định).

Lưu ý, không thể nộp bài thi nếu thời điểm nộp bài thi không nằm trong khoảng thời gian thi.

b. (1.0 điểm) **proc_LichThi_SelectBySinhVien**
 @MaSinhVien nvarchar(50),
 @HocKy int,
 @NamHoc nvarchar(10)

Có chức năng hiển thị danh sách các lịch thi trong học kỳ **@HocKy**, năm học **@NamHoc** mà sinh viên có mã **@MaSinhVien** có thể tham gia thi ở thời điểm hiện tại.

c. (1.5 điểm) **proc_LichThi_SelectBaiThi**
 @MaLichThi bigint

Có chức năng hiển thị danh sách các bài thi của sinh viên dự thi thuộc lịch thi có mã là **@MaLichThi**, thông tin cần hiển thị bao gồm: *Mã sinh viên, Họ tên, Nhóm lớp, Số thứ tự, Thời điểm nộp bài, Nội dung bài thi và Tên file đính kèm*. Trong đó *Số thứ tự* được đánh số theo từng sinh viên dựa trên thứ tự tăng dần của thời điểm nộp bài. Lưu ý những sinh viên chưa nộp bài thi cũng phải được hiển thị trong danh sách (với thông tin *Thời điểm nộp bài, Nội dung bài thi và Tên file đính kèm* được hiển thị là NULL).

d. (1.5 điểm) **proc_ThongKeLichThi**
 @TuNgay date,
 @DenNgay date

Có chức năng thống kê số lượng lịch thi của mỗi ngày trong khoảng thời gian từ ngày **@TuNgay** đến ngày **@DenNgay**. Yêu cầu những ngày nào không có lịch thi cũng phải hiển thị trong kết quả thống kê với số lượng lịch thi được thể hiện là 0.

Câu 4: Tạo các hàm sau đây:

a. (1 điểm) **func_SoLuongSinhVienDaNopBai(@MaLichThi int)**

Có chức năng thống kê số lượng sinh viên đã có nộp bài thi cho lịch thi có mã là **@MaLichThi**.

b. (1.0 điểm) **func_ThongKeSinhVienDuThi(@TuNgay date, @DenNgay date)**

Có chức năng trả về một bảng cho biết số lượng sinh viên được bố trí dự thi mỗi ngày trong khoảng thời gian từ ngày **@TuNgay** đến ngày **@DenNgay**. Yêu cầu những ngày nào không có

sinh viên được bố trí dự thi cũng phải hiển thị trong kết quả với số lượng sinh viên được bố trí dự thi là 0.

Câu 5 (1.0 điểm) Viết các lệnh thực hiện các yêu cầu sau đây:

- Tạo tài khoản có tên là **user_MãSinhViên** (ví dụ: **user_22T1020001**) với mật khẩu là **123456**
- Cho phép tài khoản trên được phép truy cập vào cơ sở dữ liệu đã tạo.
- Cấp phát cho tài khoản trên các quyền sau đây:
 - Được phép thực hiện lệnh **SELECT** trên các bảng của cơ sở dữ liệu.
 - Được phép sử dụng các thủ tục và hàm đã tạo ở trên.