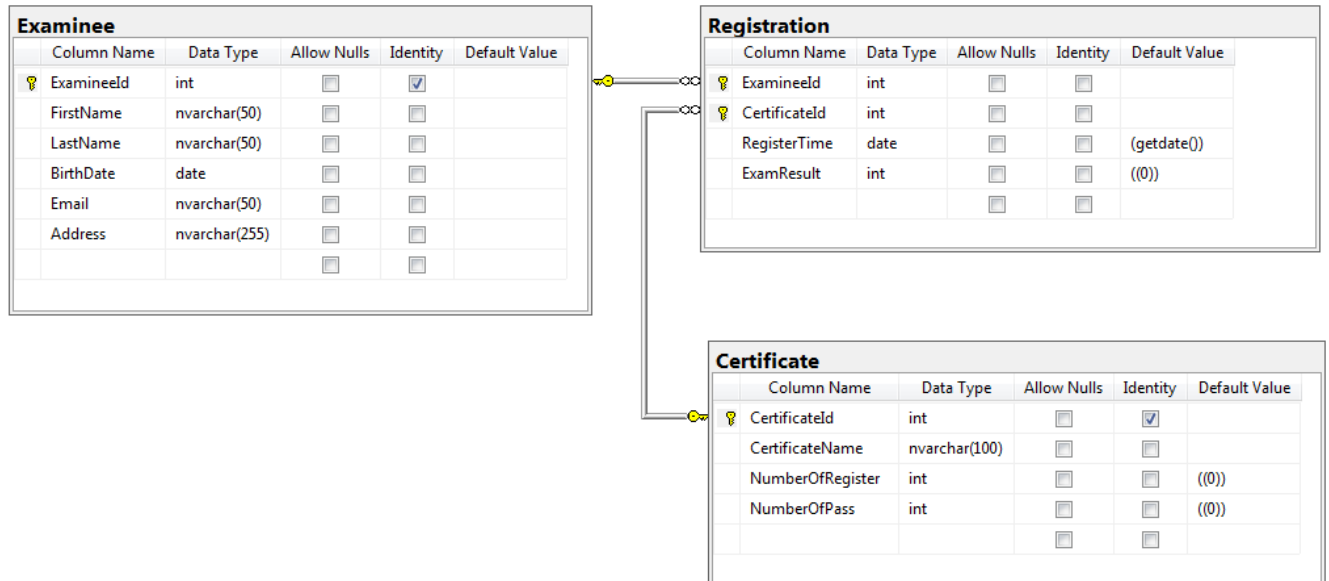


BÀI THỰC HÀNH 4

Bài IV.01:

Tạo CSDL có tên **BTH0401_MãSinhViên** (Ví dụ: **BTH0401_18T1020001**) và cài đặt các bảng dữ liệu theo sơ đồ sau:



Nhập dữ liệu cho các bảng Examinee và Certificate

Câu 1: Viết các trigger sau đây (giả thiết mỗi lần bổ sung hoặc cập nhật dữ liệu chỉ tác động trên một dòng):

- Trigger **trg_Registration_Insert** bắt lệnh **INSERT** trên bảng **Registration** sao cho mỗi khi bổ sung thêm một dòng dữ liệu trong bảng này thì cập nhật lại số lượng người đăng ký dự thi chứng chỉ (cột **NumberOfRegister**) trong bảng **Certificate**.
- Trigger **trg_Registration_Update** bắt lệnh **UPDATE** trên bảng **Registration** sao cho khi cập nhật giá trị cột **ExamResult** của một dòng trong bảng này thì đồng thời cập nhật lại số lượng người đã thi đạt chứng chỉ (cột **NumberOfPass**) trong bảng **Certificate**.
- Trigger **trg_Registration_Delete** bắt lệnh **UPDATE** trên bảng **Registration** sao cho khi xóa một dòng dữ liệu trong bảng này thì cập nhật lại số lượng người đăng ký (cột **NumberOfRegister**) và số người thi đạt chứng chỉ (cột **NumberOfPass**) trong bảng **Certificate**.

Lưu ý: Kết quả thi chứng chỉ là đạt nếu điểm thi lớn hơn hoặc bằng 5.

Câu 2: Viết các thủ tục sau đây:

- proc_Registration_Add**
 @ExamineeId int,
 @CertificateId int,

@Result nvarchar(255) output

Có chức năng bổ sung một hồ sơ đăng ký dự thi chứng chỉ. Nếu bổ sung thành công, tham số **@Result** trả về chuỗi rỗng, ngược lại tham số này trả về chuỗi cho biết lý do không bổ sung được đăng ký.

b. **proc_SaveExamResult**

@ExamineeId int,

@CertificateId int,

@ExamResult int,

@Result nvarchar(255) output

Có chức năng cập nhật điểm thi chứng chỉ. Trong đó lưu ý điểm thi phải là giá trị từ 0 đến 10. Nếu cập nhật thành công, tham số **@Result** trả về chuỗi rỗng, ngược lại tham số này trả về chuỗi cho biết lý do không cập nhật được điểm thi.

c. **proc_Examinee_Select**

@SearchValue nvarchar(255) = 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 dữ liệu. Trong đó, **@SearchValue** là giá trị cần tìm (tìm kiếm tương đối theo họ tên, nếu tham số này là chuỗi rỗng thì không tìm kiếm), **@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 và tham số đầu ra **@PageCount** cho biết tổng số trang.

d. **proc_CountRegisteringByDate**

@From date,

@To date

Có chức năng thống kê số lượng đăng ký dự thi của mỗi ngày trong khoảng thời gian từ ngày **@From** đến ngày **@To**. 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).

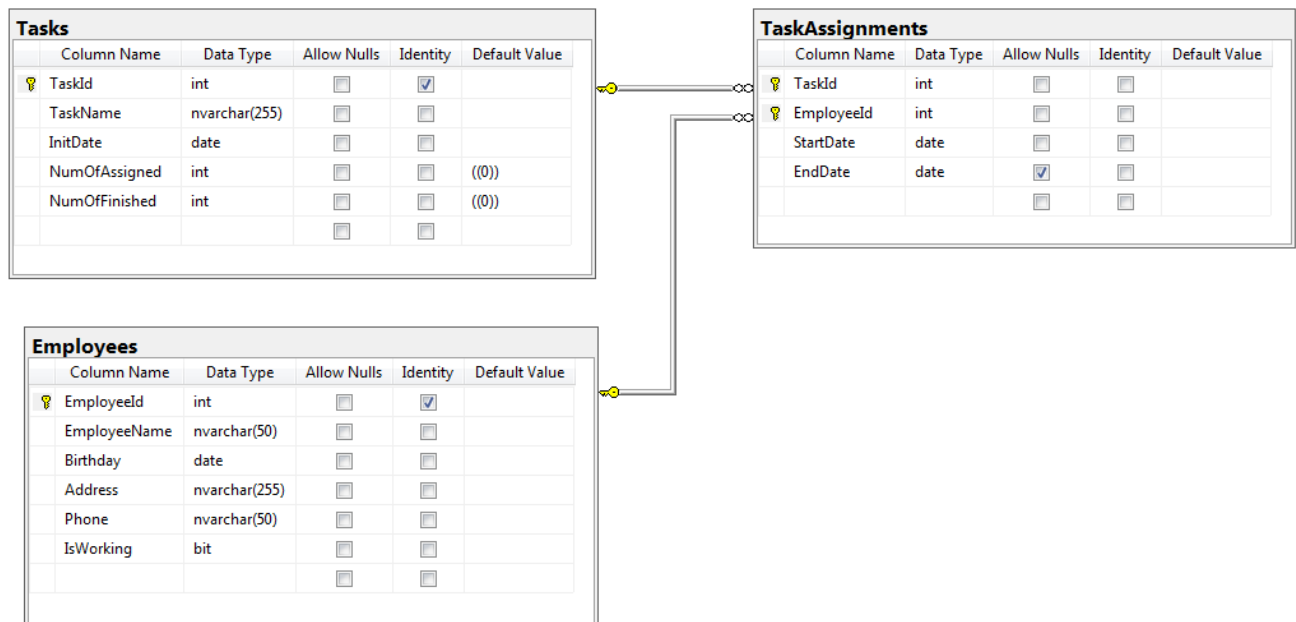
Câu 3: Viết các hàm sau đây

a. **func_CountPassed(@ExamineeId int)** có chức năng tính số lượng chứng chỉ mà người dự thi có mã **@ExamineeId** đã thi đạt.

b. **func_TotalByDate(@From date, @To date)** có chức năng trả về bảng thống kê số lượng đăng ký dự thi của mỗi ngày trong khoảng thời gian từ ngày **@From** đến ngày **@To**. 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).

Bài IV.02:

Tạo CSDL có tên **BTH0402_MãSinhViên** (Ví dụ: **BTH0402_18T1020001**) và cài đặt các bảng dữ liệu theo sơ đồ sau:



Nhập dữ liệu cho các bảng **Employees** và **Tasks**

Câu 1: Viết các trigger sau đây (giả thiết mỗi lần bổ sung hoặc cập nhật dữ liệu chỉ tác động trên một dòng):

- Trigger **trg_TaskAssignments_Insert** bắt lệnh **INSERT** trên bảng **TaskAssignments** sao cho mỗi khi bổ sung một dòng dữ liệu trong bảng này thì tính lại số lượng nhân viên đã được giao việc thực hiện công việc (cột **NumOfAssigned** của bảng **Tasks**)
- Trigger **trg_TaskAssignments_Update** bắt lệnh **UPDATE** trên bảng **TaskAssignments** sao cho mỗi khi cập nhật giá trị cột **EndDate** của một dòng trong bảng này thì tính lại số lượng nhân viên đã hoàn thành công việc được giao (cột **NumOfFinished** của bảng **Tasks**)
- Trigger **trg_TaskAssignments_Delete** bắt lệnh **DELETE** trên bảng **TaskAssignments** sao cho mỗi khi xóa một dòng trong bảng **TaskAssignments** thì cập nhật lại số lượng nhân viên được giao thực hiện công việc (cột **NumOfAssigned**) và số lượng nhân viên hoàn thành công việc được giao (Cột **NumOfFinished**) của bảng **Tasks**.

*Lưu ý: Một công việc được giao là hoàn thành nếu giá trị của cột **EndDate** khác NULL.*

Câu 2: Viết các thủ tục sau đây:

- proc_TaskAssignments_Create**
 @TaskId int,
 @EmployeeId int,
 @StartDate date
 @Result nvarchar(255) output

Có chức năng giao việc có mã **@TaskId** cho nhân viên có mã **@EmployeeId**. Tham số đầu ra **@Result** trả về chuỗi rỗng trong trường hợp giao việc thành công; Trong trường hợp ngược lại, tham số **@Result** trả về chuỗi cho biết lý do không giao được việc.

b. **proc_TaskAssignments_Update**

@TaskId int,

@EmployeeId int,

@EndDate date,

@Result nvarchar(255) output

Có chức năng cập nhật ngày hoàn thành công việc (cột **EndDate** của bảng **TaskAssignments**). Tham số đầu ra **@Result** trả về chuỗi rỗng nếu việc cập nhật thành công, ngược lại tham số này trả về chuỗi cho biết lý do không cập nhật được dữ liệu.

c. **proc_Employees_Select**

@SearchName nvarchar(255) = 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 dữ liệu. Trong đó, **@SearchName** là giá trị cần tìm (tìm kiếm tương đối theo họ tên, nếu tham số này là chuỗi rỗng thì không tìm kiếm), **@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 và tham số đầu ra **@PageCount** cho biết tổng số trang.

d. **proc_SummaryEndedTaskByDate**

@From date,

@To date

Có chức năng thống kê số lượt công việc đã được ghi nhận hoàn thành của mỗi ngày trong khoảng thời gian từ ngày **@From** đến ngày **@To**. Yêu cầu kết quả thống kê phải hiển thị đầy đủ tất cả các ngày trong khoảng thời gian trên (những ngày không có công việc được ghi nhận hoàn thành thì hiển thị với số lượng là 0)

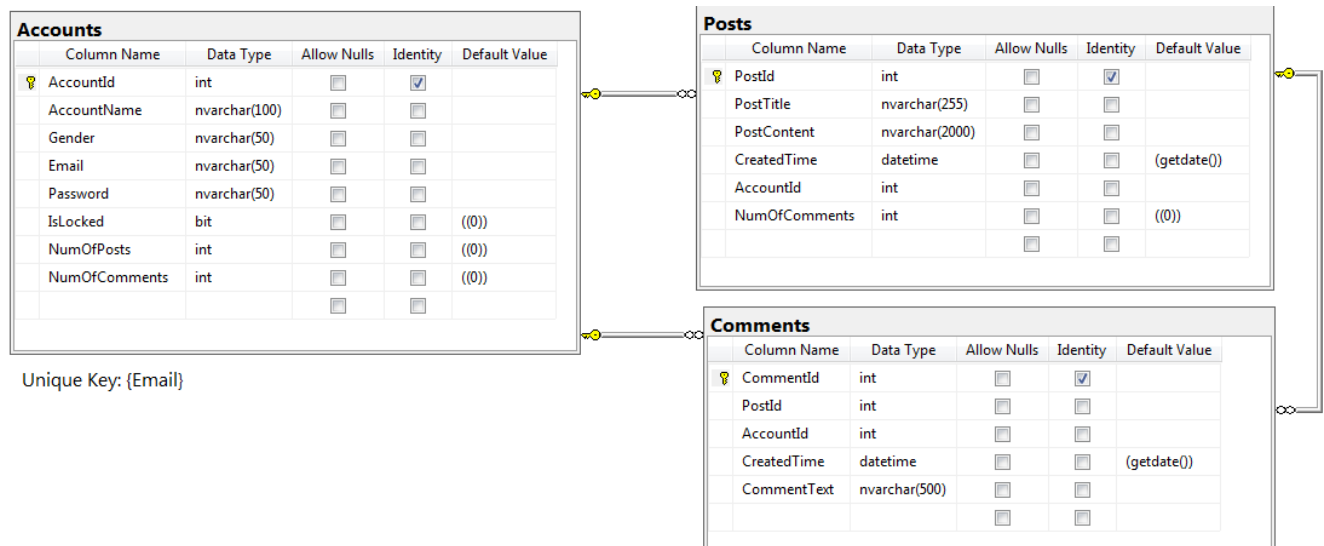
Câu 3: Viết các hàm sau đây

a. **func_CountNotEndTasks(@EmployeeId int)** có chức năng đếm số lượng công việc mà nhân viên có mã **@EmployeeId** chưa hoàn thành.

b. **func_SummaryEndedTasksByDate(@From date, @To date)** có chức năng trả về bảng thống kê số lượng công việc hoàn thành của mỗi ngày trong khoảng thời gian từ ngày **@From** đến ngày **@To**. Yêu cầu kết quả thống kê phải hiển thị đầy đủ tất cả các ngày trong khoảng thời gian trên (những ngày không có công việc được ghi nhận hoàn thành thì hiển thị với số lượng là 0)

Bài IV.03:

Tạo cơ sở dữ liệu có tên **BHT0403_MãSinhViên** (Ví dụ: **BTH0403_18T1020001**) và cài đặt các bảng dữ liệu theo sơ đồ sau:



Nhập dữ liệu cho bảng **Accounts**

Câu 1: Viết các trigger sau đây:

a. Các trigger cho bảng **Posts**

Tên Trigger

trg_Posts_Insert

trg_Post_Delete

Lệnh xử lý

INSERT

DELETE

Sao cho khi bổ sung hoặc xóa dữ liệu trong bảng **Posts** thì cập nhật lại số lượng bài viết của tài khoản (cột **NumOfPosts**) trong bảng **Accounts**

b. Các trigger cho bảng **Comments**

Tên Trigger

trg_Comments_Insert

trg_Comments_Delete

Lệnh xử lý

INSERT

DELETE

Sao cho khi bổ sung hoặc xóa dữ liệu trong bảng **Comments** thì cập nhật lại số lượng bài bình luận của tài khoản (cột **NumOfComments** của bảng **Accounts**) và số lượng bài bình luận của bài viết (cột **NumOfComments** của bảng **Posts**)

Câu 2: Viết các thủ tục sau đây:

a. **proc_Posts_Insert**

@PostTitle nvarchar(255),

@PostContent nvarchar(2000),

@AccountId int,

@PostId int output

Có chức năng tạo mới một bài viết. Tham số đầu ra **@PostId** trả về mã của bài viết được tạo mới trong trường hợp thành công; Ngược lại, tham số này trả về giá trị nhỏ hơn hoặc bằng 0 nhằm cho biết lý do không tạo được bài viết.

b. **proc_Accounts_Update**

@AccountId int,

@AccountName nvarchar(100),

@Gender nvarchar(50),

@Email nvarchar(50),

@Result nvarchar(255) output

Có chức năng cập nhật thông tin của tài khoản có mã **@AccountId**. Nếu việc cập nhật là thành công, tham số đầu ra **@Result** trả về chuỗi rỗng; Ngược lại, tham số này trả về chuỗi cho biết lý do tại sao không cập nhật được dữ liệu.

c. **proc_Posts_Select**

@SearchValue nvarchar(255) = N'',

@Page int = 1,

@PageSize int = 20,

@RowCount int output,

@PageCount int output

Có chức năng tìm kiếm và hiển thị danh sách các bài viết dưới dạng phân trang. Trong đó, tham số **@SearchValue** là tiêu đề hoặc nội dung của bài viết cần tìm (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 và tham số đầu ra **@PageCount** cho biết tổng số trang.

d. **proc_CountPostByYear**

@FromYear int,

@ToYear int

Có chức năng thống kê số lượng bài viết và số lượng bài thảo luận của từng năm trong khoảng thời gian từ năm **@FromYear** đến năm **@ToYear**. Yêu cầu kết quả thống kê phải hiển thị đủ tất cả các năm trong khoảng thời gian trên (kể cả những năm không có bài viết hay bài thảo luận).

Câu 3: Viết các hàm sau đây

a. **func_CountPost(@From date, @To date)** có chức năng tính tổng số lượng bài được viết trong khoảng thời gian từ ngày **@From** đến ngày **@To**.

b. **func_CountPostByYear(@FromYear int, @ToYear int)** trả về bảng thống kê số lượng bài viết và số lượng bài thảo luận của từng năm trong khoảng thời gian từ năm **@FromYear** đến năm **@ToYear**. Yêu cầu kết quả thống kê phải hiển thị đủ tất cả các năm trong khoảng thời gian trên (kể cả những năm không có bài viết hay bài thảo luận).