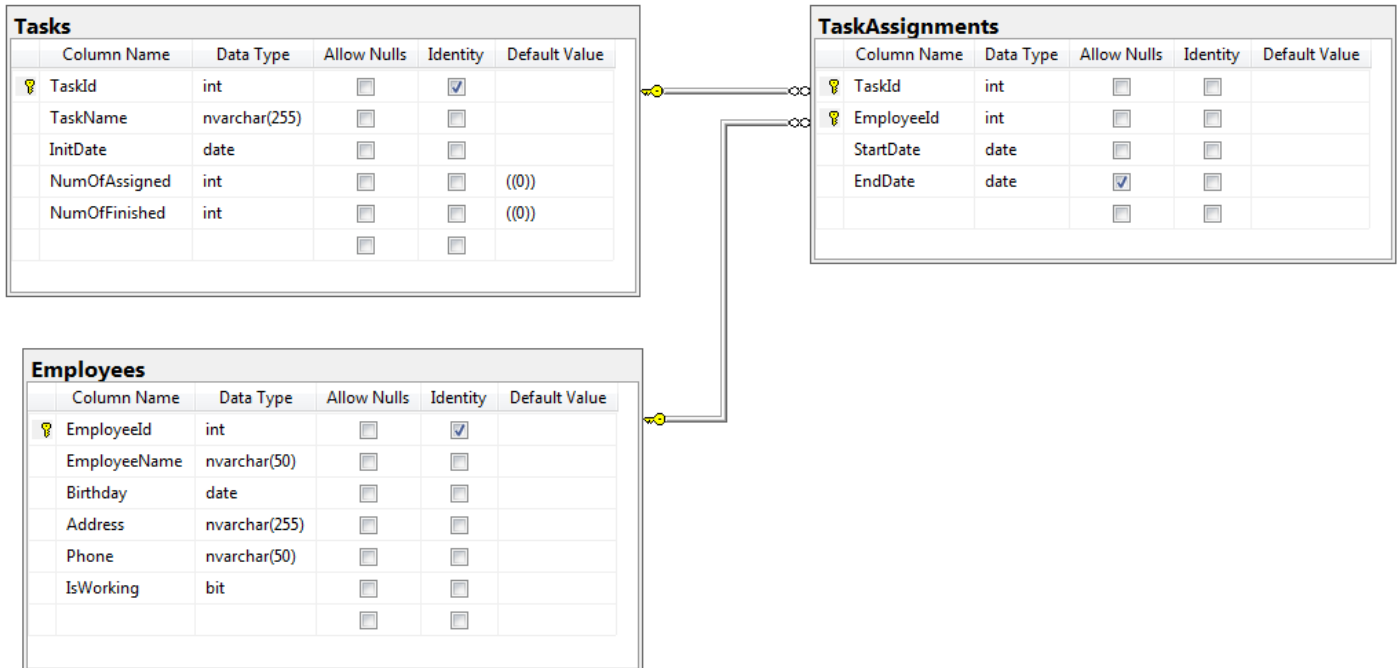


- Tên học phần: **Các hệ quản trị cơ sở dữ liệu**
- Mã học phần: **TIN3053**
- Thời gian làm bài: **120 phút**
- Số tín chỉ: **3**  
(Không kể thời gian phát đề)

Tạo cơ sở dữ liệu có tên BAITHI\_MãSinhViên (Ví dụ: BAITHI\_18T1020001) và cài đặt các bảng dữ liệu theo sơ đồ sau:



**Lưu ý:**

- Sinh viên tự nhập dữ liệu cho các bảng.
- Bài nộp của sinh viên phải đầy đủ 03 file, bao gồm: 01 file script bài làm và 02 file của cơ sở dữ liệu đã được tạo ở trên.

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

- (1 điểm) 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**)
- (1,5 điểm) 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**)

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:

- (1 điểm) **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. (1 điểm) **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. (1,5 điểm) **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. (1,5 điểm) **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. (1 điểm) **func\_CountNotEndTasks(@EmployeeId int)** có chức năng đếm số lượng công việc mà nhân viên có mã **@EmployeeId** chưa hoàn thành.

b. (1,5 điểm) **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)