Đề tài:

**Đăng ký thi chứng chỉ ngoại ngữ**

Sinh viên thực hiện:

1542228- Phạm Trung Hiếu

1542242- Hồ Hải Nam

1542244- Vũ Hoàng Nam

**MỤC LỤC**

[**I.** **Yêu cầu chức năng**](#_30j0zll)

[**II.** **Thiết kế CSDL**](#_1fob9te)

[**III.** **Cài đặt Stored Procedure**](#_3znysh7)

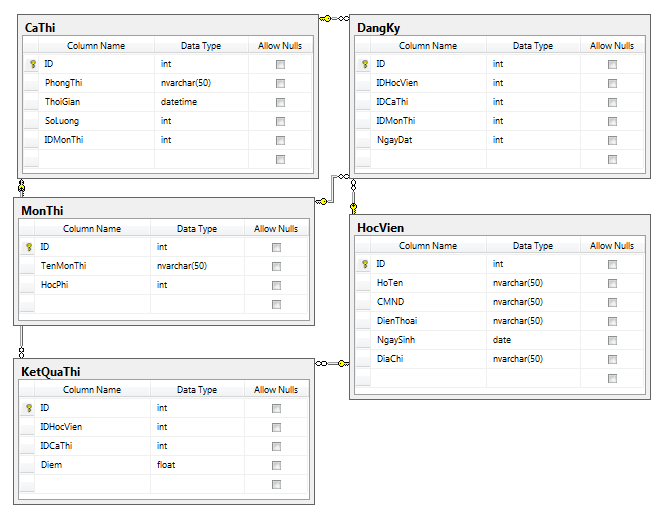
[**IV.** **Mô tả các tình huống tranh chấp**](#_2et92p0)

[**V.** **Giải quyết tranh chấp**](#_tyjcwt)

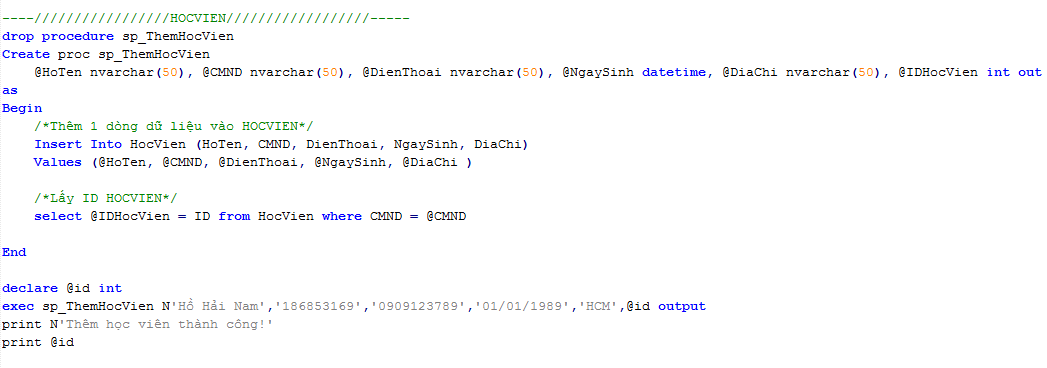
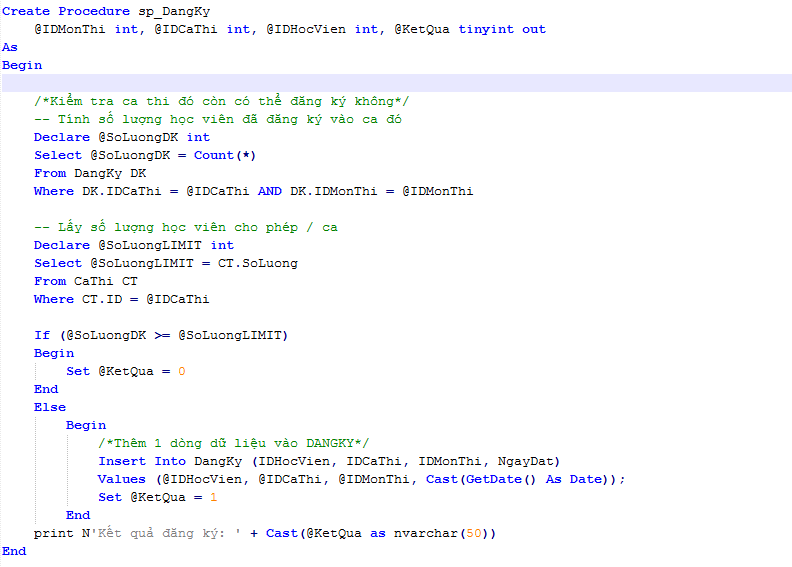
1. **Yêu cầu chức năng**

* Lấy thông tin học viên
* Đăng ký thi

1. **Thiết kế CSDL**



1. **Cài đặt Stored Procedure**

* Lấy thông tin học viên  
    
  
* Đăng ký thi  
    
  

1. **Mô tả các tình huống tranh chấp**

* **Tình huống: lấy thông tin học viên**

|  |  |
| --- | --- |
| **T1** | **T2** |
| Thêm dữ liệu học viên |  |
|  | Lấy danh sách học viên |
| Nếu lỗi thì rollback |  |

1:Giao tác T1 tiến hành thêm học viên, trong quá trình thêm thì xảy ra lỗi => rollback toàn bộ. Trong khi đó T2 tiến hành lấy danh sách học viên trước khi T1 rollback =>đọc dữ liệu rác.

|  |  |
| --- | --- |
| **T1** | **T2** |
| Lấy danh sách học viên |  |
|  | Thêm dữ liệu học viên |
| Lấy danh sách học viên |  |

2:Danh sách học viên của T1 trong hai lần đọc lên là khác nhau, bởi vì trước khi T1 đọc dữ liêu lên lần hai thì T2 đã tiến hành thêm vào học viên mới =>lỗi bóng ma.

|  |  |
| --- | --- |
| **T1** | **T2** |
| Lấy danh sách học viên |  |
|  | Sửa dữ liệu học viên |
| Lấy danh sách học viên |  |

3:Danh sách học viên của T1 trong hai lần đọc lên là khác nhau, bởi vì trước khi T1 đọc dữ liêu lên lần hai thì T2 đã tiến hành sửa dữ liệu học viên =>lỗi không đọc lại được dữ liệu.

* **Tình huống: đăng ký thi**

|  |  |
| --- | --- |
| **T1** | **T2** |
| Thêm dữ liệu đăng ký thi |  |
|  | Lấy danh sách đăng ký thi |
| Nếu lỗi thì rollback |  |

-Giao tác T1 tiến hành thêm dữ liệu đăng kí thi, trong quá trình thêm thì xảy ra lỗi => rollback toàn bộ. Trong khi đó T2 tiến hành lấy danh sách đăng ký thi trước khi T1 rollback =>đọc dữ liệu rác.

|  |  |
| --- | --- |
| **T1** | **T2** |
| Lấy danh sách đăng ký thi |  |
|  | Thêm dữ liệu đăng ký thi |
| Lấy danh sách đăng ký thi |  |

-Danh sách đăng ký thi của T1 trong hai lần đọc lên là khác nhau, bởi vì trước khi T1 đọc dữ liêu lên lần hai thì T2 đã tiến hành thêm dữ liệu đăng ký thi mới =>lỗi bóng ma.

|  |  |
| --- | --- |
| **T1** | **T2** |
| Lấy danh sách đăng ký thi |  |
|  | Sửa dữ liệu đăng ký thi (xác nhận đăng ký) |
| Lấy danh sách đăng ký thi |  |

-Danh sách đăng ký thi của T1 trong hai lần đọc lên là khác nhau, bởi vì trước khi T1 đọc dữ liêu lên lần hai thì T2 đã tiến hành sửa dữ liệu đăng ký thi =>lỗi không đọc lại được dữ liệu.

|  |  |
| --- | --- |
| **T1** | **T2** |
| Lấy danh sách đăng ký thi |  |
|  | Lấy danh sách đăng ký thi |
| Cập nhật xác nhận đăng ký thi |  |
|  | Cập nhật xác nhận đăng kí thi |

-Dữ liệu ở T1 bị mất do T2 ghi đè => lỗi mất dữ liệu cập nhật.

1. **Giải quyết tranh chấp**

* **Hướng giải quyết**
  1. Tình huống lấy thông tin học viên
     + 1. Trường hợp này ta dùng mức cô lập Read Commited ở transaction lấy danh sách học viên, vì với mức cô lập này transaction lấy danh sách học viên sẽ không đọc được dữ liệu học viên đang ở trạng thái cập nhật,mà nó phải chờ cho đến khi việc cập nhật danh học viên chấm dứt.
       2. Dùng mức cô lập Serializable ở Transaction 1. Vì mức cô lập này nó sẽ khóa toàn bộ bảng danh sách học viên ngăn không cho một transaction khác tác động vào việc thêm hoặc cập nhật dữ liệu vào bảng danh sách học viên. Do đó transaction thứ 2 sẽ chỉ insert được dữ liệu mới khi mà transaction 1 đã chạy xong,kết quả transaction 1 sẽ cho hai lần đọc dữ liệu như nhau.
       3. Dùng mức cô lập Repeatable Read ở transaction T2. Mức cô lập này sẽ làm cho T2 sẽ không ghi dữ liệu mới vào trong bảng học viên khi mà T1 đang đọc, do đó sẽ ngăn được tình trạng không đọc lại được dữ liệu
  2. Tình huống đăng ký thi:
     + 1. Trường hợp này ta dùng mức cô lập Read Commited ở transaction T2. Vì khi dùng mức cô lập này T2 sẽ bị treo cho đến khi T1 thực hiện xong việc cập nhật.
       2. Trường hợp này ta dùng mức cô lập Serializable Read ở Transaction 1. Với mức cô lập này Transactions 1 sẽ khóa bảng đăng ký lại không cho bất kì transaction nào tiến hành ghi vào cho đến khi nó thực hiện xong.
       3. Trường hợp này ta dùng mức cô lập Repeatable Read ở transaction T2.Mức cô lập này sẽ giúp cho T2 không insert vào bảng đăng ký khi mà đang có transaction nào đang đọc trước đó.
       4. Trường hợp này ta sử dụng update lock trên T1, nó sẽ tiến hành khóa bảng đăng ký lại ngăn không cho giao tác nào đọc và ghi trên bảng đăng ký cho đến khi nó thực hiện xong, do đó việc cập nhật dữ liệu của T1 sẽ có ý nghĩa và không gây ra hiện tượng lost update.