

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN**

**HỆ QUẢN TRỊ CỞ SỞ DỮ LIỆU**

**Đề tài: XÂY DỰNG PHẦN MỀM QUẢN LÝ CÔ NHI VIỆN**

Giảng viên hướng dẫn: Thầy Nguyễn Hồ Duy Tri

Lớp: IS210.J12

**NHÓM TYQ**

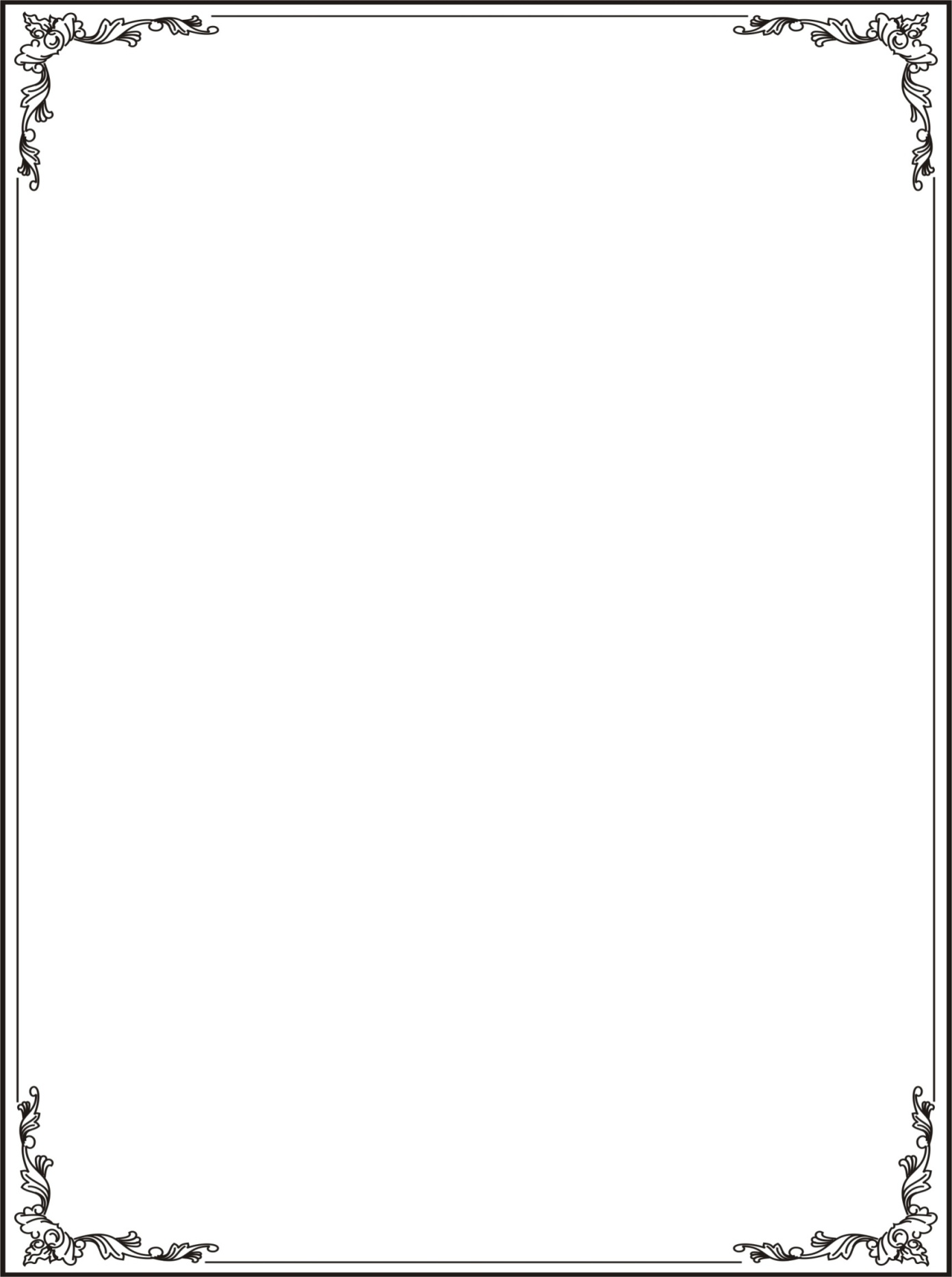
Sinh viên thực hiện:

1) Bùi Thị Huyền Trân. MSSV: 16521275

2) Nguyễn Thị Kim Yến. MSSV: 16521485

3) Dương Thị Xuân Quỳnh. MSSV: 16521016

TP HCM, tháng năm 2018



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN**

**HỆ QUẢN TRỊ CỞ SỞ DỮ LIỆU**

**Đề tài: XÂY DỰNG PHẦN MỀM QUẢN LÝ CÔ NHI VIỆN**

Giảng viên hướng dẫn: Thầy Nguyễn Hồ Duy Tri

Lớp: IS210.J12

**NHÓM TYQ**

Sinh viên thực hiện:

1.Bùi Thị Huyền Trân -16521275

2.Nguyễn Thị Kim Yến -16521485

3.Dương Thị Xuân Quỳnh -16521016

TP HCM, tháng năm 2018

**LỜI CÁM ƠN**

Nhóm TYQ xin gửi lời cảm ơn chân thành đến các giảng viên khoa Hệ thống thông tin, trường Đại học Công nghệ Thông tin đã giúp cho nhóm có những kiến thức nền tảng cơ bản để thực hiện đề tài này. Đặc biệt, nhóm gửi lời cảm ơn sâu sắc tới thầy Nguyễn Hồ Duy Tri - giảng viên thực hành và thầy Trần Thanh Hùng – giảng viên lý thuyết môn Hệ quản trị Cơ sở dữ liệu. Các thầy đã tận tình giúp đỡ, trực tiếp chỉ bảo, góp ý cho nhóm trong suốt quá trình làm đồ án môn học. Nhờ đó, chúng em đã tiếp thu được nhiều kiến thức bổ ích và vận dụng vào đồ án. Ngoài ra, để đồ án được hoàn thành thì không quên cám ơn những người đã làm ra, cám ơn các thành viên trong nhóm đã chăm chỉ hoàn thành nhiệm vụ đúng tiến độ.

Từ những sự cố gắng của nhóm và các phần góp ý của các thầy, cuối cùng nhóm đã hoàn thành đồ án có tên: “Quản lý cô nhi viện Hoa Hướng Dương”. Trải qua thời gian một học kỳ thực hiện đề tài, nhóm đã vận dụng những kiến thức nền tảng đã tích lũy, kết hợp sự tìm tòi, học hỏi, cùng với sự hướng dẫn tận tình cùng những đóng góp quý báu của các Thầy và các bạn đã giúp nhóm chúng em hoàn thành tốt hơn báo cáo môn học của mình. Tuy nhiên, trong quá trình thực hiện, không tránh khỏi những sai sót. Do đó, rất mong nhận được những sự góp ý từ phía Thầy để giúp nhóm hoàn thiện hơn. Từ đó nhóm rút kinh nghiệm để thực hiện các đề tài khác được tốt hơn, cũng như tích lũy kiến thức cho công việc sau này. Xin chân thành cảm ơn Thầy và các bạn! Trân trọng cảm ơn.

Tp. Hồ Chí Minh Tháng 01/2019

**NHẬN XÉT CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**Quản lý tài liệu**

|  |  |
| --- | --- |
| Ngày tạo: 20/11/2018 | Thời gian lưu: 20/11/2018 |
| Nhóm: TYQ | Lưu bởi: Bùi Thị Huyền Trân |

**Lịch sử thay đổi**

|  |  |  |  |
| --- | --- | --- | --- |
| Người thực hiện | Ngày thực hiện | Nội dung | Phiên bản |
| Bùi Thị Huyền Trân | 24/11 | Tạo báo cáo | {A.B.x} |
| Bùi Thị Huyền Trân | 26/11 | Chương 1, Chương 2, Chương 3 |  |
| Nguyễn Thị Kim Yến | 07/12 | Chương 5: 3, 1, 4  Chương 4: 1 |  |
| Nguyễn Thị Kim Yến | 10/12 | Chương 2: 1 |  |
| Dương Thị Xuân Quỳnh | 10/12 | Chương 2: 2 |  |
| Bùi Thị Huyền Trân | 12/12 | Chương 4: 1,2  Chương 6 |  |
| Nguyễn Thị Kim Yến | 12/12 | Chương 2: 2 |  |
| Bùi Thị Huyền Trân | 4/1/2019 | Chương 5: 2,4 |  |
| Nguyễn Thị Kim Yến | 7/1/2019 | Chương 5: 2 |  |
| Dương Thị Xuân Quỳnh | 8/1/2019 | Chương 5: 2 |  |

**Lịch sử kiểm tra**

|  |  |  |  |
| --- | --- | --- | --- |
| Người kiểm tra | Ngày kiểm tra | Nhận xét/đánh giá | Phiên bản |
| {Người kiểm tra} | {dd/MM/yyyy} | {Nội dung nhận xét/đánh giá về phiên bản được kiểm tra} | {A.B.x} |

Mục Lục

[CHƯƠNG 1: GIỚI THIỆU 7](#_Toc534787268)

[1.Từ ngữ viết tắt và thuật ngữ 7](#_Toc534787269)

[2. Tham khảo 7](#_Toc534787270)

[3. Phát biểu bài toán 7](#_Toc534787271)

[a. Đặt vấn đề 8](#_Toc534787272)

[b. Mục tiêu 8](#_Toc534787273)

[c. Công cụ sử dụng 8](#_Toc534787274)

[CHƯƠNG 2: XÁC ĐỊNH VÀ PHÂN TÍCH YÊU CẦU 10](#_Toc534787275)

[1. Đặc tả bài toán 10](#_Toc534787276)

[2. Khảo sát yêu cầu 11](#_Toc534787277)

[a. Đối tượng khảo sát 11](#_Toc534787278)

[b. Nội dung khảo sát 11](#_Toc534787279)

[c. Phương pháp khảo sát 12](#_Toc534787280)

[d. Kết quả khảo sát 16](#_Toc534787281)

[3. Yêu cầu chức năng 20](#_Toc534787282)

[4. Yêu cầu phi chức năng 21](#_Toc534787283)

[CHƯƠNG 3: THIẾT KẾ DỮ LIỆU 22](#_Toc534787284)

[1. Mô hình quan hệ 22](#_Toc534787285)

[2. Bảng thuyết minh và thuộc tính 23](#_Toc534787286)

[3. Mô tả ràng buộc toàn vẹn 25](#_Toc534787287)

[c.1Ràng buộc khóa chính 25](#_Toc534787288)

[c.2 Ràng buộc khóa ngoại 26](#_Toc534787289)

[c.3 Ràng buộc thuộc tính tổng hợp 27](#_Toc534787290)

[c.4 Ràng buộc miền giá trị 28](#_Toc534787291)

[c.5 Ràng buộc liên thuộc tính 30](#_Toc534787292)

[c.6 Ràng buộc do thuộc tính tổng hợp 31](#_Toc534787293)

[CHƯƠNG 4: XÂY DỰNG STORE PROCEDURE VÀ TRIGGER 32](#_Toc534787294)

[1. Store Procedure 32](#_Toc534787295)

[a. Store Procedure trong SQL Server 32](#_Toc534787296)

[b. Danh sách các store procedure 32](#_Toc534787297)

[c. Mô tả store procedure 41](#_Toc534787298)

[2. Trigger 51](#_Toc534787299)

[a. Trigger trong SQL Server 51](#_Toc534787300)

[b. Danh sách các trigger 51](#_Toc534787301)

[c. Mô tả các trigger 51](#_Toc534787302)

[CHƯƠNG 5: XỬ LÝ ĐỒNG THỜI 52](#_Toc534787303)

[1. Các mức cô lập trong SQL Server 52](#_Toc534787304)

[a. Read uncommited 52](#_Toc534787305)

[b. Read commited 52](#_Toc534787306)

[c. Repeatable read 52](#_Toc534787307)

[d. Serializable 53](#_Toc534787308)

[2. Transaction 53](#_Toc534787309)

[2.1 Lost update 54](#_Toc534787310)

[2.2 Dirty read 58](#_Toc534787311)

[2.3 Non - repeatable Read 62](#_Toc534787312)

[2.4 Phantom 65](#_Toc534787313)

[3. Lock 69](#_Toc534787314)

[4. Deadlock 70](#_Toc534787315)

[CHƯƠNG 6 : THIẾT KẾ GIAO DIỆN 77](#_Toc534787316)

[1. Form chính 77](#_Toc534787317)

[2. Form quản lý thông tin trẻ 77](#_Toc534787318)

[3. Form quản lý thông tin loại nhân viên 78](#_Toc534787319)

[4. Form quản lý thông tin nhân viên 79](#_Toc534787320)

[5. Form quản lý thông tin nhà tài trợ 79](#_Toc534787321)

[6. Form quản lý chương trình tài trợ 80](#_Toc534787322)

[7. Form quản lý người nhận trẻ 81](#_Toc534787323)

[8. Form quản lý quá trình nhận trẻ 81](#_Toc534787324)

[9. Form quản lý chi tiêu 82](#_Toc534787325)

[10. Form quản lý chi tiết chi tiêu 83](#_Toc534787326)

[11. Form báo cáo thông tin trẻ 83](#_Toc534787327)

[12. Form báo cáo tài chính 85](#_Toc534787328)

[13. Form login 86](#_Toc534787329)

[6. Kết luận 87](#_Toc534787330)

[6.1 Kết quả đạt được 87](#_Toc534787331)

[6.2 Hạn chế 87](#_Toc534787332)

[6.3 Hướng phát triển 87](#_Toc534787333)

[7. Bảng phân công công việc 88](#_Toc534787334)

**CHƯƠNG 1: GIỚI THIỆU**

* Giới thiệu: xây dựng phần mềm quản lý cô nhi viên.
* Mục đích:
* Về công nghệ:
* Nắm được quy trình xây dựng và phát triển phần mềm.
* Tiếp xúc và sử dụng một số phần mềm, công cụ hỗ trợ như sau:

+ Xây dựng phần mềm: NetBeans IDE 8.2

+ Xây dựng cơ sở dữ liệu: SQL Server 2014 Management Studio.

* Về hiệu quả:
* Xây dựng được phần mềm quản lý việc nhập và bán sách gần gũi, thiết thực, chính xác, dễ dàng sử dụng cho nhân viên trong nhà sách.
* Cải tiến phần mềm với giao diện thân thiện hiệu quả cao hơn so với việc sử dụng Excel, Word.
* Ngăn chặn tốt các vấn đề, không để xảy ra trình trạng không nhất quán trong dữ liệu.
* Tạo thuận tiện và giảm tối đa thời gian chờ cho người sử dụng hệ thống.
* Về kết quả đạt được (đối với sinh viên xây dựng đồ án):
* Nắm vững kiến thức để bổ trợ cho các môn học tiếp theo, cũng như cách ngăn chặn tốt các vấn đề, không để xảy ra trình trạng không nhất quán trong dữ liệu, tạo thuận tiện và giảm tối đa thời gian chờ cho người sử dụng hệ thống.

## Từ ngữ viết tắt và thuật ngữ

|  |  |
| --- | --- |
| Từ viết tắt | Diễn giải |
|  |  |
|  |  |

## Tham khảo

|  |  |  |
| --- | --- | --- |
| STT | Tên tài liệu/ Nguồn tham khảo | Mô tả chi tiết |
|  | Đồ án Hệ quản trị cơ sở dữ liệu, đề tài: Quản lý nhân sự - tiền lương | Tác giả: nhóm sinh viên K9 |
|  | Đồ án Hệ quản trị cơ sở dữ liệu, đề tài: Quản lý rạp chiếu phim | Tác giả: Nguyễn Minh Phát |

1. **Phát biểu bài toán**
   1. **Đặt vấn đề**

* Việc tin học hóa hệ thống quản lý đang trở thành mối quan tâm của nhiều doanh  
  nghiệp, cơ quan, tổ chức. Lợi ích của các hệ thống quản lý này đem lại là vô cùng to  
  lớn. Nó khắc phục được những yếu điểm của phương pháp quản lý truyền thống bằng  
  thủ công và mở ra một môi trường làm việc hiện đại, tự động cho doanh nghiệp, tổ chức. Từ đó, phương pháp này đem lại nhiều hiệu quả về mặt thời gian, chi phí và nhân lực cho doanh nghiệp và các tổ chức.
* Cô nhi viện Hướng Dương ở số 184 Phan Văn Khỏe, Phường 5, Quận 1, thành phố Hồ Chí Minh là nơi cưu mang, nuôi dưỡng những đứa trẻ có hoàn cảnh bất hạnh. Hằng ngày cô nhi viện phải quản lý một lượng lớn thông tin về các trẻ, cũng như nguồn tài trợ và chi tiêu của ở đây, việc giải quyết các công việc với một lượng dữ liệu khá lớn, đòi hỏi tính chính xác cao, cũng như vấn đề truy xuất dữ liệu nhanh, thì phương pháp truyền thống ghi sổ sách, và các phần mềm Word, Excel không còn đáp ứng được yêu cầu bài toán ở đây.
* Giải pháp: cô nhi viện cần thay đổi hình thức quản lý từ thủ công truyền thống sang hình thức quản lý bằng ứng dụng công nghệ thông tin. Hệ thống được xây  
  dựng phải giải quyết các vấn đề về lưu trữ dữ liệu, xử lý dữ liệu và hỗ trợ các thao tác, nghiệp vụ cho nhân viên ở cô nhi viện. Dựa trên tình hình hiện tại ở cô nhi viện nhóm quyết định đề ra giải pháp xây dựng phần mềm quản lý để công việc quản lý trở nên đơn giản, dễ dàng, nhanh chóng, hiệu quả, chính xác.
  1. **Mục tiêu**
* Quản lý thông tin trẻ.
* Quản lý thông tin nhân viên.
* Quản lý nhà tài trợ.
* Quản lý tài trợ của các nhà tài trợ.
* Quản lý chi tiêu.
* Quản lý người nhận trẻ.
* Báo cáo: báo cáo chi tiêu, báo cáo thông tin trẻ.
  1. **Công cụ sử dụng**
* Trong quá trình thực hiện, nhóm đã sử dụng một số phần mềm, ứng dụng phục vụ cho việc xây dựng và quản lý đề tài:  
  + Microsoft SQLServer  
  + Netbeans IDE

**+** Github

**+** Word Online

**CHƯƠNG 2: XÁC ĐỊNH VÀ PHÂN TÍCH YÊU CẦU**

**Đặc tả bài toán**

* Nhân viên phải đăng nhập mới sử dụng được phần mềm quản lý cô nhi viện (nhằm đảm bảo tính báo mật cho thông tin tại cô nhi viện). Sau khi đăng nhập phần mềm sẽ chuyển đến trang chủ với các chức năng sau:
* Quản lý gồm các chức năng: thêm, cập nhật, xóa, tra cứu. Với các mục quản lý:

+ Quản lý trẻ: quản lý các thông tin: Mã trẻ, tên trẻ, ngày sinh, giới tính, ngày vào, hoàn cảnh, người đưa trẻ vào, trạng thái, tên bảo mẫu.

+ Quản lý nhân viên:

* Quản lý loại nhân viên: quản lý các thông tin: mã loại nhân viên, tên nhân viên
* Quản lý nhân viên: quản lý các thông tin: loại nhân viên, mã nhân viên, tên nhân viên, giới tính, ngày sinh, địa chỉ, điện thoại, ngày vào làm.

+ Quản lý tài trợ:

* Quản lý nhà tài trợ: quản lý các thông tin: mã nhà tài trợ, tên nhà tài trợ, số điện thoại, địa chỉ
* Quản lý chương trình tài trợ: quản lý các thông tin: mã tài trợ, tên nhà tài trợ, ngày tài trợ, hình thức tài trợ, số tiền tài trợ.

+ Quản lý chi tiêu:

* Quản lý chi tiêu: quản lý các thông tin: mã chi tiêu, tên chi tiêu, ngày chi, tổng tiền
* Quản lý chi tiết chi tiêu: quản lý các thông tin: mã chi tiêu, mã chi tiết chi tiêu, tên chi tiết chi tiêu, số tiền

+ Quản lý nhận nuôi trẻ:

* Quản lý người nhận nuôi trẻ: quản lý các thông tin: mã người nhận, tên người nhận, địa chỉ, số điện thoại.
* Quản lý việc nhận nuôi trẻ: quản lý các thông tin: tên người nhận nuôi, tên trẻ, ngày nhận.

+ Quản lý tài khoản: quản lý các thông tin: tên đăng nhập, mật khẩu, tên hiển thị

* Báo cáo gồm các chức năng: thống kê, in báo cáo. Các mục báo cáo trong khoảng thời gian:

+ Báo cáo sỉ số trẻ:

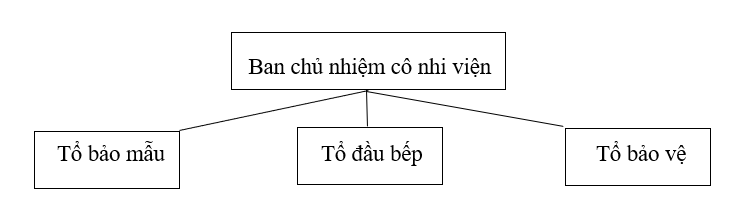
* + Trẻ thêm mới
  + Trẻ được nhận nuôi
  + Trẻ đang ở cô nhi viện

+ Báo cáo tài chính

1. **Khảo sát yêu cầu**
   1. **Đối tượng khảo sát**



* 1. **Nội dung khảo sát**
* Hiện trạng tổ chức
  + Đối nội:

Đối ngoại:

* Cô nhi viện nhận ủng hộ, quyên góp từ các nhà tài trợ.
* Nhận trẻ từ những người đưa trẻ vào.
* Gửi trẻ cho những người nhận nuôi trẻ.
* Hiện trạng nghiệp vụ
  + Ban chủ nhiệm cô nhi viện: quản lý, giám sát chung các hoạt động trong cô nhi viện. Là nơi đưa ra quyết định cuối cùng.
  + Tổ bảo mẫu: quản lý các hoạt động trẻ, chi tiêu, tài trợ, nhận nuôi trẻ.
  + Tổ đầu bếp: quản lý việc ăn uống trong cô nhi viện.
  + Tổ bảo vệ: quản lý an ninh trong cô nhi viện.
* Hiện trạng tin học
  + Phần cứng:
    - Các thiết bị hiện tại: máy tính để bàn
    - Số lượng: 3
    - Cấu hình: Intel ® Core ™ i5
    - Vị trí (vật lý) : phòng ban chủ nhiệm
    - Tình hình kết nối mạng: có kết nối mạng, đường truyền ổn định
    - Loại kết nối: LAN
  + Phần mềm:
    - Hệ điều hành: Windows 10
    - Các phần mềm tiện ích khác: Microsoft Office
  + Con người:
    - Trình độ chuyên môn Tin học: có khả năng sử dụng máy tính và các phần mềm với giao diện đơn giản, gần gũi.
  1. **Phương pháp khảo sát**

Phương pháp khảo sát: phỏng vấn

|  |  |
| --- | --- |
| **Người được phỏng vấn: Ban chủ nhiệm cô nhi viện** | **Ngày phỏng vấn: 17/11/2018** |
| Câu hỏi: | Ghi nhận: |
| Câu 1:  Thưa cô hiện nay cô nhi viện có ứng dụng CNTT trong việc quản lý học sinh không? | Trả lời:  Hiện nay cô nhi viện có dùng Word và Excel để lưu các thông tin cũng như tính toán, cơ bản đáp ứng được các yêu cầu đơn giản. Nhưng chưa đem lại hiệu quả lắm. |
| Câu 2:  Không biết là hiện tại cô nhi viện có những nhóm nhân viên nào thưa cô? | Trả lời:  Hiện tại ở đây có 15 nhân viên với 4 nhóm: ban chủ nhiệm, bảo mẫu, bếp và bảo vệ |
| Câu 3:  Trước đây cô nhi viện có từng sử dụng phần mềm chuyên dụng nào để quản lý chưa ạ? | Trả lời:  Ngoài Word và Excel thì ở đây chưa từng sử dụng phần mềm chuyên dụng nào khác. |
| Câu 4:  Cô nhi viện mình hiện tại có bao nhiêu nghiệp vụ và đó là nhưng gì thưa cô? | Trả lời:  Cô nhi viện có các nghiệp vụ như sau:   * Tiếp nhận trẻ và quản lý các thông tin của các em. * Quản lý nhân viên. * Tiếp nhận tài trợ. * Giao trẻ cho nơi nhận nuôi trẻ. * Quản lý việc chi tiêu * Quản lý các bữa ăn, chế độ dinh dưỡng của các em. * Đảm bảo môi trường an ninh trật tự. * Tạo các hoạt động ngoại khóa cho các em. * Báo cáo định kì về việc thu chi và sỉ số trẻ. |
| Câu 5:  Theo cô mong muốn những nghiệp vụ nào sẽ có trong phần mềm quản lý cô nhi viện? | Trả lời:  Trước mắt cần có các nghiệp vụ cơ bản sau:   * Quản lý trẻ * Quản lý nhân viên * Quản lý tài trợ * Quản lý chi tiêu * Quản lý việc nhận nuôi trẻ * Báo cáo thu chi và sỉ số trẻ |
| Câu 6:  Ứng với từng nghiệp vụ trong phần mềm là nhóm nhân viên nào đảm nhiệm thưa cô? | Trả lời:  Các nghiệp vụ đó đa số là do bảo mẫu quản lý, ban chủ nhiệm quản lý chung, dựa vào các báo cáo để nắm tình hình, nếu có quyết định quan trọng thì bảo mẫu sẽ trao đổi với ban chủ nhiệm. Ban chủ nhiệm là nơi đưa ra quyết định cuối cùng. |
| Câu 7:  Báo cáo tài chính cụ thể là báo cáo những gì ạ? | Trả lời:  Báo cáo tài chính bao gồm tổng thu (tiền tài trợ), tổng chi (tiền chi tiêu) và tiền quỹ hiện tại trong cô nhi viện |
| Câu 8:  Về phần báo cáo sỉ số trẻ cụ thể như thế nào thưa cô? | Trả lời:  Bao gồm các hoạt động:   * Sỉ số trẻ thêm mới * Sỉ số trẻ được nhận nuôi * Sỉ số trẻ đang ở cô nhi viện |

|  |  |
| --- | --- |
| **Người được phỏng vấn: Tổ bảo mẫu** | **Ngày phỏng vấn: 17/11/2018** |
| Câu hỏi: | Ghi nhận: |
| Câu 9:  Thưa các cô, hiện tại tổ bảo mẫu của mình gồm bao nhiêu người? | Trả lời:  Hiện tại tổ bảo mẫu gồm 10 người. |
| **QUẢN LÝ TRẺ** | |
| Câu 10:  Nghiệp vụ quản lý trẻ là mình quản lý các thông tin cá nhân của các em và tra cứu thông tin khi cần thiết đúng không ạ? | Trả lời:  Đúng |
| Câu 11:  Những thông tin mình cần quản lý là những gì thưa các cô? | Trả lời:  Cần quản lý các thông tin: tên trẻ, ngày sinh, giới tính, ngày vào, hoàn cảnh, người đưa trẻ vào, trạng thái, tên bảo mẫu. |
| Câu 12:  Về tìm kiếm thì cần tìm kiếm theo gì để thuận tiện nhất ạ? | Trả lời:  Cô thấy nên tìm theo tên của trẻ là thuận tiện nhất. |
| **QUẢN LÝ NHÂN VIÊN** | |
| Câu 13:  Về phần quản lý nhân viên thì em thấy nên chia ra quản lý loại nhân viên và quản lý thông tin cá nhân của từng nhân viên, các cô thấy như thế nào? | Trả lời:  Hợp lý |
| Câu 14:  Những thông tin nào mình cần cho 1 nhân viên ạ? | Trả lời:  Các thông tin cá nhân cần thiết là: loại nhân viên, tên nhân viên, giới tính, ngày sinh, địa chỉ, điện thoại, ngày vào làm. Với lại cũng cần thêm mục tra cứu nhân viên theo tên. |
| **QUẢN LÝ TÀI TRỢ** | |
| Câu 15:  Hiện tại cô nhi viện mình có thường xuyên nhận các nguồn tài trợ không thưa các cô? | Trả lời:  Nguồn tài trợ hiện tại của cô nhi viện khá ổn định. Ngoài những doanh nghiệp địa phương thì thỉnh thoảng cũng có các đoàn Việt kiều ủng hộ. |
| Câu 16:  Các thông tin về tài trợ mà cô nhi viện mình quan tâm là gì ạ? | Trả lời:  Các thông tin mà cô nhi viện cần lưu trữ lại là:   * Thông tin nhà tài trợ: tên nhà tài trợ, số điện thoại, địa chỉ. * Thông tin về việc tài trợ: tên nhà tài trợ, ngày tài trợ, hình thức tài trợ, số tiền tài trợ. |
| **QUẢN LÝ CHI TIÊU** | |
| Câu 17:  Quản lý chi tiêu bao gồm các thông tin nào ạ? | Trả lời:  Hiện tại các cô cần lưu các thông tin: tên chi tiêu, ngày chi, tổng tiền với cụ thể bao gồm các khoản chi tiết chi tiêu: tên chi tiết chi tiêu, số tiền, thuộc chi tiêu nào. |
| **QUẢN LÝ VIỆC NHẬN NUÔI TRẺ** | |
| Câu 18:  Có nhiều người đến xin nhận nuôi trẻ không ạ? | Trả lời:  Cũng ít thôi. Trung bình mỗi năm khoảng 6 - 8 trẻ được nhận nuôi. |
| Câu 19:  Khi tiếp nhận và đồng ý cho việc nhận nuôi trẻ. Các thông tin mà cô nhi viện mình lưu trữ là gì ạ? | Trả lời:   * Thông tin người nhận nuôi trẻ: quản lý các thông tin: tên người nhận, địa chỉ, số điện thoại. * Việc nhận nuôi trẻ: quản lý các thông tin: tên người nhận nuôi, tên trẻ, ngày nhận. |
| Câu 20:  Các cô có nghĩ ứng với mỗi nghiệp vụ quản lý chúng ta nên có mã cho từng dòng thông tin, giống như số thứ tự của trẻ để quản lý thông tin chặt chẽ hơn? | Trả lời:  Các cô nghĩ nên phải có. |
| Câu 21:  Cô nhi viện mình có cần khung tìm kiếm ứng với mỗi nghiệp vụ quản lý không ạ. | Trả lời:  Cô nghĩ việc đó khá cần để khi tìm nhanh. |

* 1. **Kết quả khảo sát**

|  |  |  |
| --- | --- | --- |
| Dự án: Quản lý cô nhi viện | Công việc/xử lý: Tiếp nhận trẻ | Số thứ tự: XX  Ngày lập: 20/11/2018 |
| 1.Mô tả công việc:  2.Điều kiện bắt đầu:  3.Thông tin đầu vào:  4. Kết quả đầu ra:  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm: | * Thêm thông tin trẻ * Khi có trẻ mới vào cô nhi viện, nhập sai thông tin trẻ chỉnh sửa lại hay cần tra cứu thông tin trẻ. * MaTre, TenTre, GioiTinh, NgaySinh, NgayVao, HoanCanh, NguoiDuaTreVao, TrangThai, MaNV. * Thông tin trẻ đã được thêm mới, cập nhật hy cần tìm kiếm. * Tổ bảo mẫu | |

|  |  |  |
| --- | --- | --- |
| Dự án: Quản lý cô nhi viện | Công việc/xử lý: Quản lý nhân viên | Số thứ tự: XX  Ngày lập: 20/11/2018 |
| 1.Mô tả công việc:  2.Điều kiện bắt đầu:  3.Thông tin đầu vào:  4. Kết quả đầu ra:  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm: | * Thêm, cập nhật, lưu trữ, tìm kiếm thông tin nhân viên. * Khi có thông tin nhân viên cần thêm mới, cập nhật hay cần tra cứu thông tin nhân viên. * MaNV, TenNV, GioiTinh, NgaySinh, DiaChi, SoDT, NgayVaoLam, MaLoaiNV, TrangThai. * Thông tin nhân viên đã được thêm mới, cập nhật hay cần tra cứu. * Ban chủ nhiệm cô nhi viện. * Ngày sinh nhân viên phải nhỏ hơn ngày vào làm, nhân viên phải trên 18 tuổi. | |

|  |  |  |
| --- | --- | --- |
| Dự án: Quản lý cô nhi viện | Công việc/xử lý: Quản lý tài trợ | Số thứ tự: XX  Ngày lập: 20/11/2018 |
| 1.Mô tả công việc:  2.Điều kiện bắt đầu:  3.Thông tin đầu vào:  4. Kết quả đầu ra:  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm: | * Thêm, cập nhật, xóa, lưu trữ, tìm kiếm thông tin tài trợ và nhà tài trợ. * Khi có thông tin tài trợ mới cần thêm, sửa, xóa hay cần tra cứu. * MaNhaTaiTro, TenNhaTaiTro, SoDT, Diachi, MaTaiTro, TenTaiTro, NgayTaitro, HinhThucTaiTro, SoTien. * Thông tin tài trợ, nhà tài trợ đã được thêm mới, cập nhật hay cần tra cứu. * Ban chủ nhiệm cô nhi viện. | |

|  |  |  |
| --- | --- | --- |
| Dự án: Quản lý cô nhi viện | Công việc/xử lý: Quản lý chi tiêu | Số thứ tự: XX  Ngày lập: 20/11/2018 |
| 1.Mô tả công việc:  2.Điều kiện bắt đầu:  3.Thông tin đầu vào:  4. Kết quả đầu ra:  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm: | * Thêm, sửa, xóa, tra cứu thông tin chi tiêu. * Khi có phát sinh phí phục vụ cho mọi hoạt động ở cô nhi viện. * MaChiTieu, TenChiTieu, NgayChi, SoTien, ChiTiet\_ChiTieu * Thông tin chi tiêu đã được thêm mới, cập nhật hay cần tra cứu. * Ban chủ nhiệm cô nhi viện, tổ bảo mẫu. | |

|  |  |  |
| --- | --- | --- |
| Dự án: Quản lý cô nhi viện | Công việc/xử lý: Quản lý người nhận trẻ | Số thứ tự: XX  Ngày lập: 20/11/2018 |
| 1.Mô tả công việc:  2.Điều kiện bắt đầu:  3.Thông tin đầu vào:  4. Kết quả đầu ra:  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm: | * Thêm, sửa, xóa, lưu trữ, tra cứu thông tin người nhận nuôi trẻ. * Khi có người có lòng hảo tâm muốn nhận nuôi, chăm sóc trẻ, cần tra cứu thông tin người nhận nuôi trẻ. * MaNguoiNhan, TenNguoiNhan, DiaChi, SoDT, MaTre, NgayNhan. * Thông tin người nhận nuôi trẻ đã được thêm mới, cập nhật hay cần tra cứu. * Ban chủ nhiệm cô nhi viện. | |

|  |  |  |
| --- | --- | --- |
| Dự án: Quản lý cô nhi viện | Công việc/xử lý: Tiếp nhận trẻ | Số thứ tự: XX  Ngày lập: 20/11/2018 |
| 1.Mô tả công việc:  2.Điều kiện bắt đầu:  3.Thông tin đầu vào:  4. Kết quả đầu ra:  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm: | * Thêm, cập nhật, xóa thông tin trẻ * Khi nhận nuôi trẻ ta sẽ thêm trẻ, khi cần chỉnh sửa thông tin thì cập nhật, hoặc xóa trẻ * Họ tên trẻ, Ngày sinh, Giới tính, Ngày vào, Hoàn cảnh, Người đưa trẻ vào, Trạng thái, Tên bảo mẫu. * Thông tin mới nhất của trẻ và mã trẻ. * Tổ bảo mẫu | |

|  |  |  |
| --- | --- | --- |
| Dự án: Quản lý cô nhi viện | Công việc/xử lý: Báo cáo thống kê | Số thứ tự: XX  Ngày lập: 20/11/2018 |
| 1.Mô tả công việc:  2.Điều kiện bắt đầu:  3.Thông tin đầu vào:  4. Kết quả đầu ra:  5. Nơi sử dụng:  6. Những quy tắc điều kiện đi kèm: | * Thống kê thu chi, sĩ số trẻ định kì hàng tuần, hàng tháng, hàng năm. * Khi cần báo cáo cho ban chủ nhiệm cô nhi viện, khi cần kê khai với nhà nước. * Khoảng thời gian cần báo cáo thống kê. * Bản báo cáo theo yêu cầu. * Ban chủ nhiệm cô nhi viện, tổ bảo mẫu. * Thực hiện theo đúng mẫu báo cáo, thông tin chính xác , khách quan. | |

1. **Yêu cầu chức năng**

Hệ thống cần đáp ứng các chức năng sau:

* Chức năng lưu trữ, đây có thể xem là chức năng quan trọng nhất của hệ thống. Hệ thông cho phép chúng ta lưu trữ:
  + Thông tin trẻ:
  + Thông tin nhân viên:
  + Thông tin nhà tài trợ:
  + Thông tin chương trình tài trợ:
  + Thông tin người nhận trẻ:
  + Thông tin chi tiêu:
  + Thông tin tài khoản:
* Ngoài chức năng lưu trữ thì hệ thống cần đáp ứng các chức năng sau:
* Chức năng tra cứu: tra cứu thông tin trẻ, thông tin nhà tài trợ, thông tin chương trình tài trợ, thông tin người nhận trẻ, thông tin chi tiêu, tài khoản.
* Chức năng tính toán: tính toán chi tiêu,...
* Chức năng thống kê: trẻ, nhà tài trợ, chương trình tài trợ, người nhận trẻ, chi tiêu, tài khoản,...
* Chức năng tự động: tự động sinh mã trẻ, nhân viên, nhà tài trợ,...
* Chức năng bảo mật: nhân viên cần có tài khoản để sử dụng phần mềm.

1. **Yêu cầu phi chức năng**

* Yêu cầu về giao diện: giao diện gần gũi, sinh động, thân thiện.

- Yêu cầu về chất lượng:

* Tính tiến hóa
* Tính tiện dụng
* Tính hiệu quả
* Tính tương thích

# CHƯƠNG 3: THIẾT KẾ DỮ LIỆU

1. **Mô hình quan hệ**



TaiKhoan( **TenDangNhap**, MatKhau, TenHienThi)

Tre( **MaTre**, TenTre, GioiTinh, NgaySinh, NgayVao, HoanCanh, NguoiDuaTreVao, TrangThai, MaNV)

NhanVien( **MaNV**, TenNV, GioiTinh, NgaySinh, DiaChi, SoDT, NgayVL, MaLoaiNV, TrangThai)

LoaiNV( **MaLoaiNV**, TenLoaiNV)

TaiTro( **MaTaiTro, MaNhaTaiTro**, NgayTaiTro, HinhThucTaiTro, SoTien)

NhaTaiTro( **MaNhaTaiTro**, TenNhaTaiTro, SoDT, DiaChi)

NguoiNhanTre( **MaNguoiNhan**, TenNguoiNhan, DiaChi, SoDT)

CT\_NguoiNhanTre\_Tre(**MaNguoiNhan, MaTre**, NgayNhan)

ChiTieu( **MaChiTieu**, TenChiTieu, NgayChi, TongSoTien)

CT\_ChiTieu( **MaCT\_ChiTieu, MaChiTieu**, TenCT\_ChiTieu, SoTien)

1. **Bảng thuyết minh và thuộc tính**

|  |  |  |  |
| --- | --- | --- | --- |
| Thực thể/ mô hình kết hợp | Thuộc tính | Kiểu dữ liệu | Ghi chú |
| Tre  (Trẻ) | MaTre | Int | Mã trẻ |
| TenTre | Nvarchar(100) | Tên trẻ |
| GioiTinh | Navrchar(100) | Giới tính |
| NgaySinh | Smalldatetime | Ngày sinh |
| NgayVao | Smalldatetime | Ngày trẻ vào cô nhi viên |
| HoanCanh | Navarchar(4000) | Hoàn cảnh của trẻ |
| NguoiDuaVao | Nvarchar(100) | Người đưa trẻ vào |
| TrangThai | Int | Trạng thái của trẻ( còn ở cô nhi viện hay không) |
| MaNV | int | Mã nhân viên bảo mẫu của trẻ |
| NhanVien  (Nhân viên) | MaNV | int | Mã nhân viên |
| TenNV | nvarchar(100) | Tên nhân viên |
| GioiTinh | Nvarchar(100) | Giới tính |
| NgaySinh | Smalldatime | Ngày sinh |
| DiaChi | Navarchar(100) | Địa chỉ |
| SoDT | Navarchar(10) | Số điện thoại |
| NgayVL | Smalldatime | Ngày vào làm |
| MaLoaiNV | Int | Mã loại nhân viên |
| TrangThai | int | Trạng thái của nhân viên( 0 là đã nghỉ việc, 1 là đang làm việc) |
| LoaiNhanVien  (Loại nhân viên) | MaLoaiNV | int | Mã loại nhân viên |
| TenLoaiNV | Nvarchar(100) | Tên loại nhân viên |
| NhaTaiTro  (Nhà tài trợ) | MaNhaTaiTro | Int | Mã nhà tài trợ |
| TenNhaTaiTro | Nvarchar(100) | Tên nhà tài trợ |
| SoDT | Nvarchar(10) | Số điện thoại |
| DiaChi | Nvarchar(100) | Địa chỉ |
| TaiTro  (Tài trợ) | MaTaiTro | Int | Mã tài trợ |
| MaNhaTaiTro | Int | Mã nhà tài trợ |
| NgayTaiTro | Smalldatetime | Ngày tài trợ |
| HinhThucTaiTro | Nvarchar(100) | Hình thức tài trợ |
| SoTien | money | Số tiền |
| ChiTieu  (Chi tiêu) | MaChiTieu | Int | Mã chi tiêu |
| TenChiTieu | Nvarhar(100) | Tên chi tiêu |
| NgayChi | Smalldatetime | Ngày chi |
| SoTien | Money | Số tiền |
| CT\_ChiTieu  (Chi tiết chi tiêu) | MaCT\_ChiTieu | Int | Mã chi tiết chi tiêu |
| TenCT\_ChiTieu | Nvarchar(100) | Tên chi tiết chi tiêu |
| SoTien | Money | Số tiền |
| MaChiTieu | int | Mã chi tiêu |
| NguoiNhanTre  (Người nhận trẻ) | MaNguoiNhan | Int | Mã người nhận |
| TenNguoiNhan | Nvarchar(100) | Tên người nhận |
| DiaChi | Nvarchar(100) | Địa chỉ |
| SoDT | Nvarchar(10) | Số điện thoại |
| CT\_NguoiNhanTre\_Tre  (Chi tiết người nhận trẻ trẻ) | MaNguoiNhan | Int | Mã người nhận |
| MaTre | Int | Mã trẻ |
| NgayNhan | Smalldatetime | Ngày nhận trẻ |
| TaiKhoan  (Tài khoản) | TenDangNhap | Nvarchar(100) | Tên đăng nhập |
| MatKhau | Nvarchar(100) | Mật khẩu |
| TenHienThi | Nvarchar(100) | Tên hiện thị |

## 3. Mô tả ràng buộc toàn vẹn

### c.1Ràng buộc khóa chính

**Ràng buộc 1**: Mã trẻ trong quan hệ Trẻ là duy nhất

* Thuật toán: ∀ T1, T2 ∈ Tre , T1. MaTre≠ T2. MaTre
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R1 | Thêm | Sửa | Xóa |
| Tre | + | + | -(\*) |

**Ràng buộc 2**: Tên đăng nhập trong quan hệ Tài khoản là duy nhất

* Thuật toán: ∀ T1, T2 ∈ TaiKhoan , T1. TenDangNhap≠ T2. TenDangNhap
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R2 | Thêm | Sửa | Xóa |
| TaiKhoan | + | + | -(\*) |

**Ràng buộc 3**: Mã nhân viên trong quan hệ Nhân viên là duy nhất

* Thuật toán: ∀ T1, T2 ∈ NhanVien , T1. MaNV≠ T2. MaNV
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R3 | Thêm | Sửa | Xóa |
| NhanVien | + | + | -(\*) |

**Ràng buộc 3**: Mã loại nhân viên trong quan hệ Loại nhân viên là duy nhất

* Thuật toán: ∀ T1, T2 ∈ LoaiNhanVien , T1. MaLoaiNV≠ T2. MaLoaiNV
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R4 | Thêm | Sửa | Xóa |
| LoaiNV | + | + | -(\*) |

**Ràng buộc 5**: Mã nhà tài trợ trong quan hệ tài trợ là duy nhất

* Thuật toán: ∀ T1, T2 ∈ TaiTro , T1. MaTaiTro≠ T2. MaTaiTro
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R5 | Thêm | Sửa | Xóa |
| TaiTro | + | + | -(\*) |

**Ràng buộc 6**: Mã chi tiết tài trợ trong bảng chi tiết tài trợ là duy nhất

* Thuật toán: ∀ T1, T2 ∈ CT\_TaiTro , T1. MaCT\_TaiTro≠ T2. MaCT\_TaiTro
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R6 | Thêm | Sửa | Xóa |
| CT\_TaiTro | + | + | -(\*) |

**Ràng buộc 7**: Mã người nhận trẻ trong bảng người nhận trẻ là duy nhất

* Thuật toán: ∀ T1, T2 ∈ NguoiNhanTre , T1. MaNguoiNhan≠ T2. MaNguoiNhan
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R7 | Thêm | Sửa | Xóa |
| NguoiNhanTre | + | + | -(\*) |

**Ràng buộc 8**: Mã chi tiêu trong bảng Chi tiêu là duy nhất

* Thuật toán: ∀ T1, T2 ∈ Tre , T1. MaChiTieu≠ T2. MaChiTieu
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R8 | Thêm | Sửa | Xóa |
| ChiTieu | + | + | -(\*) |

**Ràng buộc 9**: Mã chi tiết chi tiêu trong bảng Chi tiết chi tiêu là duy nhất

* Thuật toán: ∀ T1, T2 ∈ CT\_ChiTieu , T1. MaCT\_ChiTieu≠ T2. MaCT\_ChiTieu
* Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R9 | Thêm | Sửa | Xóa |
| CT\_ChiTieu | + | + | -(\*) |

### c.2 Ràng buộc khóa ngoại

**Ràng buộc 10**: Mỗi tài trợ phải thuộc một nhà tài trợ củ thể

* Thuật toán: ∀ a ∈ TaiTro, ∋ b ∈ NhaTaiTro: a. MaNhaTaiTro =  
  b.MaNhaTaiTro
* Bảng tầm ảnh hưởng:
* Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R10 | Thêm | Sửa | Xóa |
| TaiTro | + | **+** (MaNhaTaiTro) | − (\*) |
| NhaTaiTro | - | **-** | − (\*) |

**Ràng buộc 11**: Mỗi nhân viên phải có một cụ thể một loại nhân viên

* Thuật toán: ∀ a ∈ NhanVien, ∋ b ∈ LoaiNhanVien: a. MaLoaiNV =  
  b.MaLoaiNV
* Bảng tầm ảnh hưởng:
* Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R10 | Thêm | Sửa | Xóa |
| NhanVien | + | **+** (MaLoaiNV) | − (\*) |
| LoaiNhanVien | - | **-** | − (\*) |

**Ràng buộc 12**: Mỗi chi tiết chi tiêu phải có một chi tiêu cụ thể củ thể

* Thuật toán: ∀ a ∈ CT\_ChiTieu, ∋ b ∈ ChiTieu: a. MaChiTieu =  
  b.MaNhaChiTieu
* Bảng tầm ảnh hưởng:
* Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R12 | Thêm | Sửa | Xóa |
| CT\_ChiTieu | + | **+** (MaChiTieu) | − (\*) |
| ChiTieu | - | **-** | − (\*) |

**Ràng buộc 13**: Mỗi chi tiết người nhận trẻ phải có một trẻ cụ thể

* Thuật toán: ∀ a ∈ CT\_NguoiNhanTre\_Tre, ∋ b ∈ Tre: a. MaTre =  
  b.MaTre
* Bảng tầm ảnh hưởng:
* Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R13 | Thêm | Sửa | Xóa |
| CT\_NguoiNhanTre\_Tre | + | **+** (MaTre) | − (\*) |
| Tre | - | **-** | − (\*) |

**Ràng buộc 14**: Mỗi chi tiết người nhận trẻ phải có một người nhận cụ thể

* Thuật toán: ∀ a ∈ CT\_NguoiNhanTre\_Tre, ∋ b ∈ NguoiNhanTre: a. MaNguoiNhan = MaNguoiNhan
* Bảng tầm ảnh hưởng:
* Bảng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R14 | Thêm | Sửa | Xóa |
| CT\_NguoiNhanTre\_Tre | + | **+** (MaNguoiNhan) | − (\*) |
| NguoiNhan | - | **-** | − (\*) |

### c.3 Ràng buộc thuộc tính tổng hợp

Ràng buộc 15:

### c.4 Ràng buộc miền giá trị

**Ràng buộc 16**: Trạng thái trẻ là 1 hoặc 0 ( 1 là trẻ đã rời khỏi cô nhi viên, 0 là trẻ còn ở cô nhi viên)

- Thuật toán: ∀ tre ∈ Tre: tre. TinhTrang ∈ {0, 1 }

- Bàng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R16 | Thêm | Sửa | Xóa |
| Tre | + | **+** (TinhTrang) | − |

**Ràng buộc 17**: Giới tính trẻ là ‘Nam’ hoặc ‘Nữ’

- Thuật toán: ∀ tre ∈ Tre: tre. GioiTinh ∈ {‘Nam’, ‘Nữ’ }

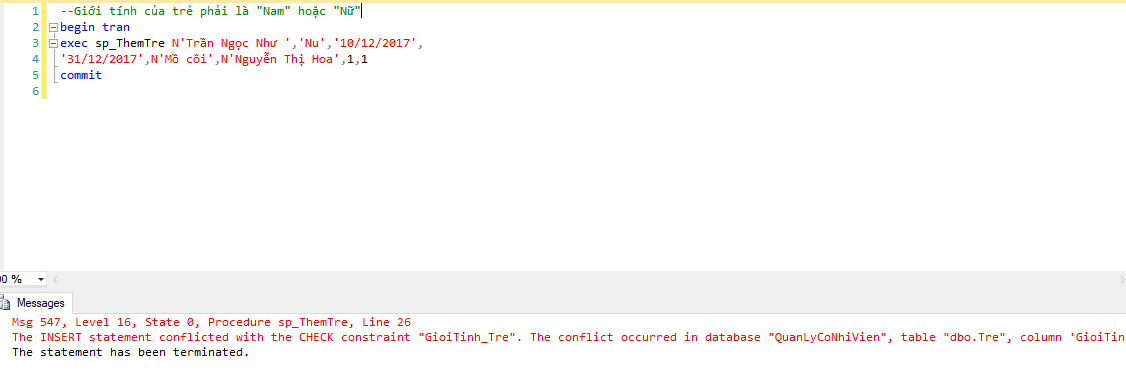
- Bàng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R17 | Thêm | Sửa | Xóa |
| Tre | + | **+** (GioiTinh) | − |

--Giới tính của trẻ là 'Nam' hoặc 'Nữ'

alter table Tre add constraint GioiTinh\_Tre check (GioiTinh in('Nam',N'Nữ'))

go



**Ràng buộc 18**: Giới tính nhân viên là ‘Nam’ hoặc ‘Nữ’

- Thuật toán: ∀ nhavien∈ NhanVien: nhanvien. GioiTinh ∈ {‘Nam’, ‘Nữ’ }

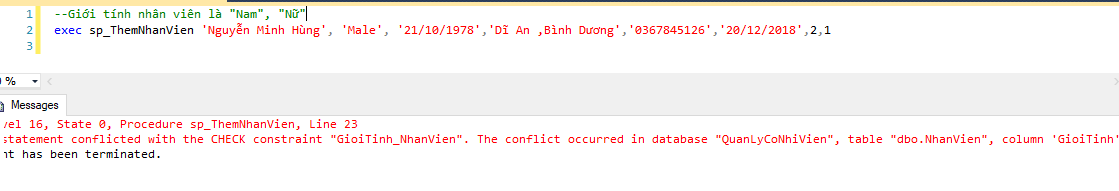
- Bàng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R18 | Thêm | Sửa | Xóa |
| NhanVien | + | **+** (GioiTinh) | − |

--Giới tính của nhân viên là 'Nam' hoặc 'Nữ'

alter table NhanVien add constraint GioiTinh\_NhanVien check (GioiTinh in('Nam',N'Nữ'))

go



**Ràng buộc 19**: Tình trạng trẻ là 0 hoặc 1 }( 0 là trẻ không còn ở cô nhi viện, 1 là trẻ còn ở cô nhi viện)

- Thuật toán: ∀ tre∈ Tre: tre. TinhTrang ∈ {0, 1 }

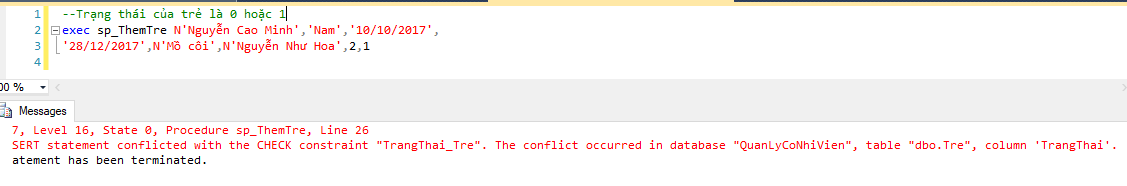
- Bàng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R18 | Thêm | Sửa | Xóa |
| Tre | + | **+** (TinhTrang) | − |

--Trạng thái của trẻ là 0 hoặc 1 với 0 là trẻ không còn ở cô nhi viện, 1 là trẻ đang ở cô nhi viện

alter table Tre add constraint TrangThai\_Tre check (TrangThai in(0,1))

go



**Ràng buộc 20**: Tình trạng nhân viên là 0 hoặc 1 (0 là nhân viên đã thôi việc, 1 là nhân viên đang làm việc)

- Thuật toán: ∀ nhavien∈ NhanVien: nhanvien. TinhTrang ∈ {0, 1 }

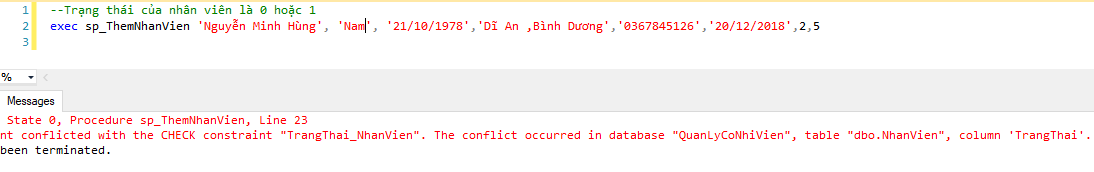
- Bàng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R20 | Thêm | Sửa | Xóa |
| NhanVien | + | **+** (TinhTrang) | − |

--Trạng thái của nhân viên là 0 hoặc 1 với 0 là đã nghỉ việc, 1 là đang làm việc

alter table NhanVien add constraint TrangThai\_NhanVien check(TrangThai in(0,1))

go

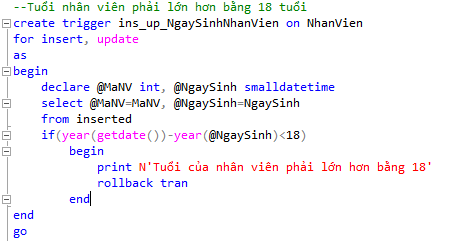


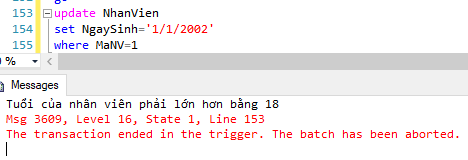
**Ràng buộc 21**: Nhân viên phải lớn hơn bằng 18 tuổi

- Thuật toán: ∀ nhavien∈ NhanVien: getDate()-year(nhanvien. NgaySinh )>=18

- Bàng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R21 | Thêm | Sửa | Xóa |
| NhanVien | + | **+** (NgaySinh) | − |





**Ràng buộc 22**: Hình thức tài trợ gồm có : tiền mặt, chuyển khoản, hiện vật.

- Thuật toán: ∀ tatro∈ TaiTro: taitro. HinhThucTaiTro ∈ {‘Tiền mặt’, ‘Chuyển khoản’,’Hiện vật’ }

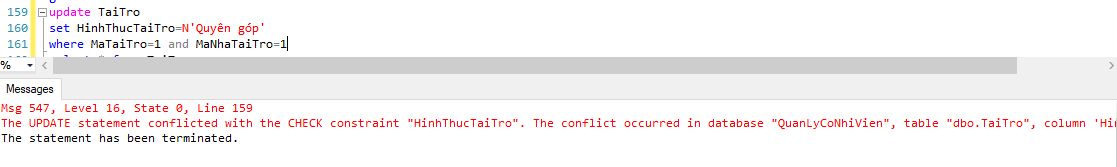
- Bàng tầm ảnh hưởng:

|  |  |  |  |
| --- | --- | --- | --- |
| R22 | Thêm | Sửa | Xóa |
| TaiTro | + | **+** (HinhThucTaiTro) | − |

--Hình thức tài trợ trong bảng tài trợ là 'Tiền mặt', 'Chuyển khoản','Hiện vật'

alter table TaiTro add constraint HinhThucTaiTro check (HinhThucTaiTro in(N'Tiền mặt',N'Chuyển khoản',N'Hiện vật'))

go



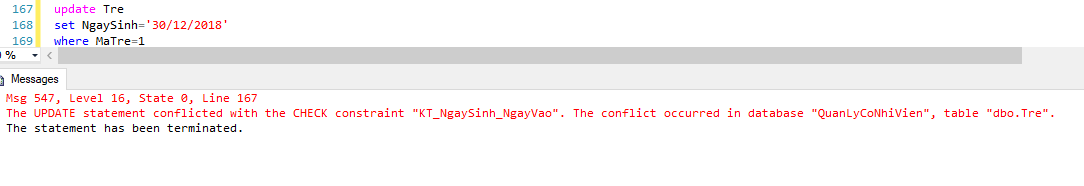
**c.5 Ràng buộc liên thuộc tính**

**Ràng buộc 23**: Ngày sinh của trẻ phải nhỏ hơn hoặc bằng ngày trẻ vào  
Thuật toán:  
∀ tre ∈ Tre: tre. NgaySinh < tre. NgayVao  
Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R23 | Thêm | Sửa | Xóa |
| Tre | + | **+** (NgaySinh, NgayVao) | − |

alter table Tre add constraint KT\_NgaySinh\_NgayVao check (NgaySinh<=NgayVao)

go

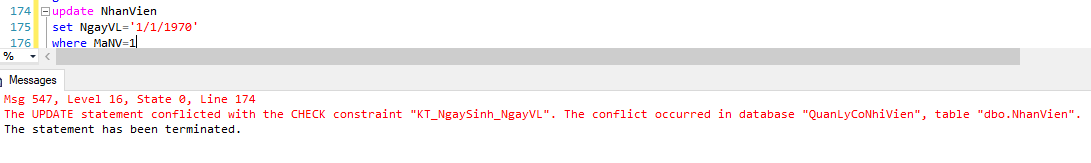


**Ràng buộc 24**: Ngày sinh của nhân viên phải nhỏ hơn ngày vào làm  
Thuật toán:  
∀ nhanvien ∈ NhanVien: nhanvien. NgaySinh < nhanvien. NgayVL  
Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R24 | Thêm | Sửa | Xóa |
| NhanVien | + | **+** (NgaySinh, NgayVL) | − |

alter table NhanVien add constraint KT\_NgaySinh\_NgayVL check(NgaySinh<NgayVL)

go



### c.6 Ràng buộc do thuộc tính tổng hợp

**Ràng buộc 25**: Số tiền trong bảng chi tiêu bằng tổng số tiền trong chi tiết chi tiết của chi tiêu đó  
Thuật toán: ∀h ∈ ChiTieu h.SoTien = Σ (ct ∈ CT\_ChiTieu : ct.MaChiTieu = h.MaChiTieu) (ct.SoTien)

Bảng tầm ảnh hưởng

|  |  |  |  |
| --- | --- | --- | --- |
| R25 | Thêm | Sửa | Xóa |
| ChiTieu | - | **+** (SoTien) | − |
| CT\_ChiTieu | + | +(MaCT,SoTien) | + |

**CHƯƠNG 4: XÂY DỰNG STORE PROCEDURE VÀ TRIGGER**

1. **Store Procedure**
2. **Store Procedure trong SQL Server**

* Khái niệm: Stored procedure là một tập các lệnh Transact SQL được đặt tên và lưu trữ trong database server. Có thể nhận tham số vào và tham số trả giá trị về. Trả về trạng thái thực thi của procedure là thành công hay không.
* Ưu điểm:

+ Cho phép lập trình theo hướng hướng modular.

+ Thực thi nhanh hơn , giảm được được việc chiếm dụng đường truyền mạng

+ Bảo mật.

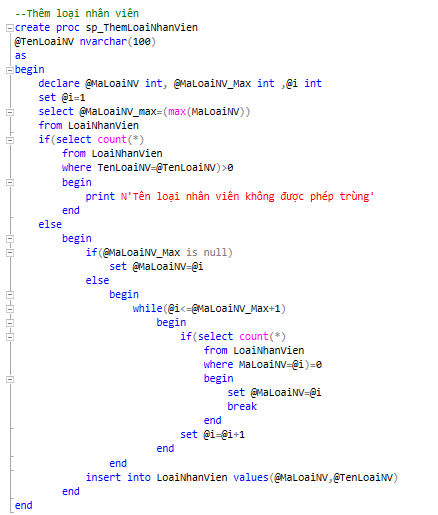
+ Xử lý các chức năng và chia sẻ với các ứng dụng khác.

1. **Danh sách các store procedure**

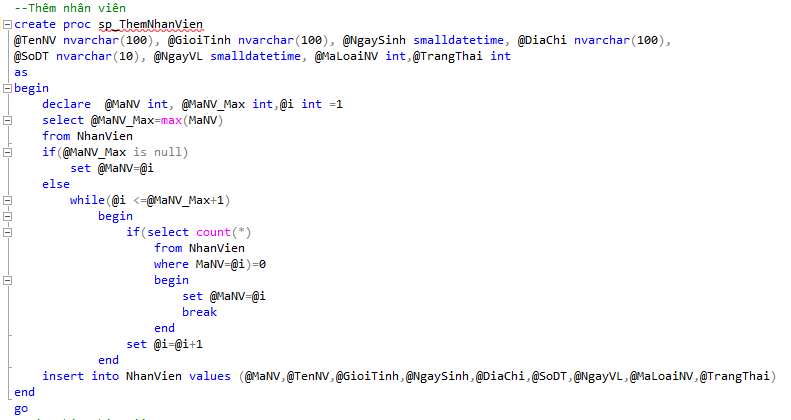
|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên store procedure** | **Tham số vào** | **Ý nghĩa** |
|  | sp\_ThemLoaiNhanVien | @TenLoaiNV nvarchar(100) | Thêm loại nhân viên |
|  | sp\_ThemNhanVien | @TenNV nvarchar(100), @GioiTinh nvarchar(100), @NgaySinh smalldatetime, @DiaChi nvarchar(100),  @SoDT nvarchar(10), @NgayVL smalldatetime, @MaLoaiNV int,  @TrangThai int | Thêm nhân viên |
|  | sp\_CapNhatNhanVien | @MaNV int,  @TenNV nvarchar(100), @GioiTinh nvarchar(100), @NgaySinh smalldatetime, @DiaChi nvarchar(100),  @SoDT nvarchar(10),  @NgayVL smalldatetime, @MaLoaiNV int,@TrangThai int | Cập nhật thông tin nhân viên. |
|  | sp\_HienThiDanhSachNhanVien |  | Hiển thị thông tin danh sách nhân viên và thông tin loại nhân viên. |
|  | sp\_TimNhanVien | @key nvarchar(100) | Tìm nhân viên theo tên |
|  | sp\_HienThiDanhSachTre |  | Hiển thông tin danh sách trẻ và nhân viên quản lý trẻ |
|  | sp\_HienThiDanhSachTreHienOCoNhiVien |  | Hiển thị danh sách trẻ hiện ở cô nhi viện |
|  | sp\_ThemTre | @TenTre nvarchar(100), @GioiTinh nvarchar(100), @NgaySinh smalldatetime,  @NgayVao smalldatetime, @HoanCanh nvarchar(4000),  @NguoiDuaTreVao nvarchar(100),  @TrangThai int,  @MaNV int | Thêm trẻ |
|  | sp\_CapNhatTre | @MaTre int,  @TenTre nvarchar(100), @GioiTinh nvarchar(100), @NgaySinh smalldatetime,  @NgayVao smalldatetime, @HoanCanh nvarchar(4000),  @NguoiDuaTreVao nvarchar(100),  @TrangThai int,  @MaNV int | Cập nhật thông tin trẻ |
|  | sp\_TimTre | @key nvarchar(100) | Tìm trẻ theo tên |
|  | sp\_HienThiDanhSachBaoMau |  | Hiển thị danh sách nhân viên là bảo mẫu hiện đang làm việc |
|  | SP\_ThemChiTieu | @TenChiTieu nvarchar(100), @NgayChi nvarchar(19) | Thêm chi tiêu |
|  | SP\_CapNhatChiTieu | @MaChiTieu int, @TenChiTieu nvarchar(100), @NgayChi nvarchar(19) | Cập nhật chi tiêu |
|  | SP\_ThemCTCT | @MaChiTieu int, @TenCTChiTieu nvarchar(100),  @SoTien money | Thêm chi tiết chi tiêu |
|  | SP\_TimKiemCTCT | @Day nvarchar(19) | Tìm chi tiết chi tiêu theo ngày |
|  | sp\_ThemNhaTaiTro | @TenNhaTaiTro nvarchar(100),  @SoDT nvarchar(10),  @DiaChi nvarchar(100) | Thêm nhà tài trợ |
|  | sp\_HienThiDanhSachTaiTro |  | Hiển thị danh sách tài trợ |
|  | sp\_ThemTaiTro | @MaNhaTaiTro int,  @NgayTaiTro smalldatetime,  @HinhThuc nvarchar(100), @SoTien money | Thêm tài trợ |
|  | sp\_CapNhatTaiTro | @MaTaiTro int,  @MaNhaTaiTro int,  @NgayTaiTro smalldatetime,  @HinhThuc nvarchar(100), @SoTien money | Cập nhật thông tin tài trợ |
|  | sp\_TimTaiTro | @key nvarchar(100) |  |
|  | sp\_ThemNguoiNhanTre | @TenNguoiNhan nvarchar(100),  @DiaChi nvarchar(100),  @SoDT nvarchar(10) | Thêm người nhận nuôi trẻ |
|  | sp\_HienThiChiTietNguoiNhanTre |  | Hiển thị chi tiết việc nhận trẻ |
|  | sp\_TraCuuChiTietNguoiNhanTreTheoTenTre | @key nvarchar(100) | Tra cứu chi tiết việc nhận trẻ theo tên trẻ |
|  | sp\_TimChiTietNguoiNhanTreTheoTenNguoiNhan | @key nvarchar(100) | Tra cứu chi tiết việc nhận nuôi trẻ theo tên người nhận |
|  | SP\_BaoCaoTreThemMoi | @TuNgay nvarchar(19),  @DenNgay nvarchar(19) | Thống kê thông tin trẻ thêm mới trong khoảng thời gian |
|  | SP\_BaoCaoTreDuocNhanNuoi | @TuNgay nvarchar(19), @DenNgay nvarchar(19) | Thống kê thông tin trẻ được nhận nuôi trong khoảng thời gian |
|  | SP\_BaoCaoTreDangOCoNhiVien | @TuTuoi int,  @DenTuoi int | Thông kê thông tin trẻ đang ở cô nhi viện trong độ tuổi cần thống kê. |
|  | SP\_ThongKeTaiTro | @TuNgay nvarchar(19), @DenNgay nvarchar(19) | Thống kê tổng tiền tài trợ trong khoảng thời gian. |
|  | SP\_ThongKeChiTieu | @TuNgay nvarchar(19), @DenNgay nvarchar(19) | Thống kê tổng tiền chi tiêu trong khoảng thời gian. |
|  | SP\_ThongKeQuy |  | Thống kê tổng tiền còn lại trong cô nhi viện |

1. **Mô tả store procedure**

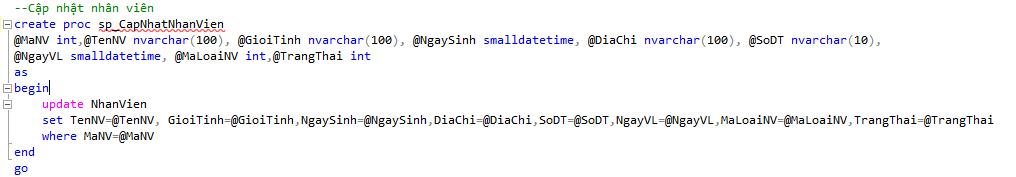
c.1



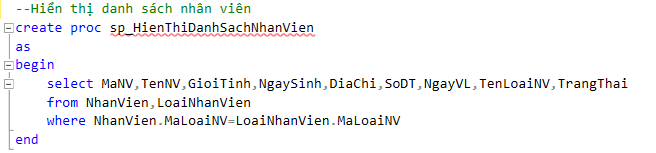
c.2

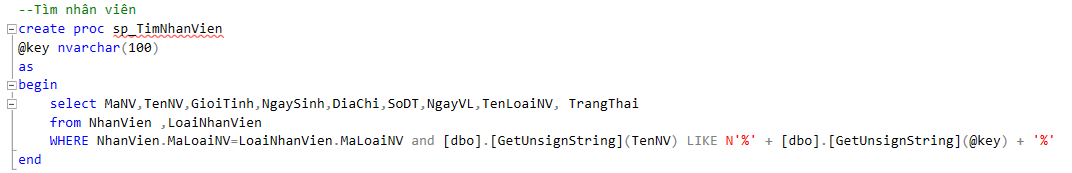


c.3

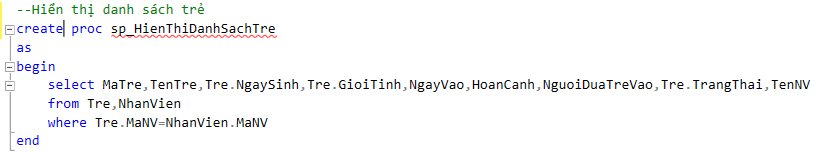


c.4

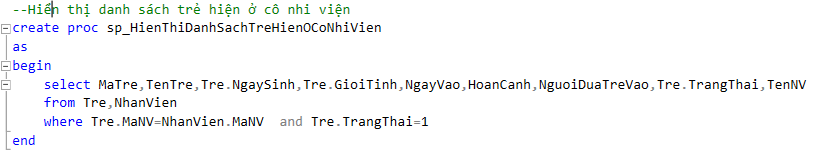


c.5

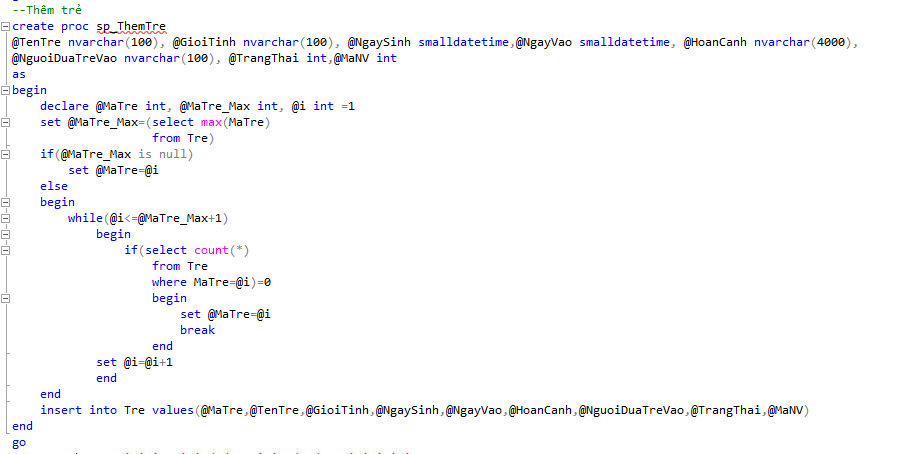
c.6



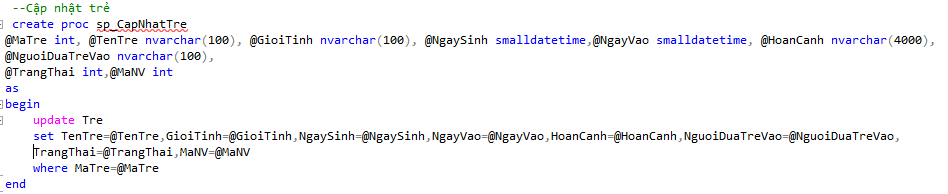
c.7



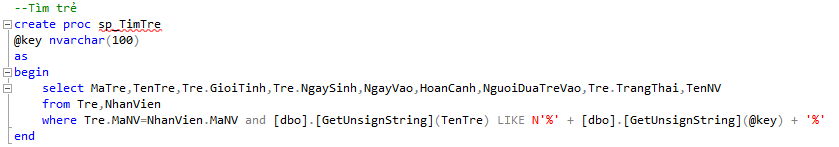
c.8



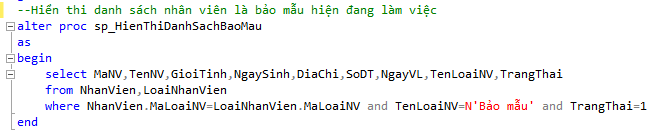
c.9



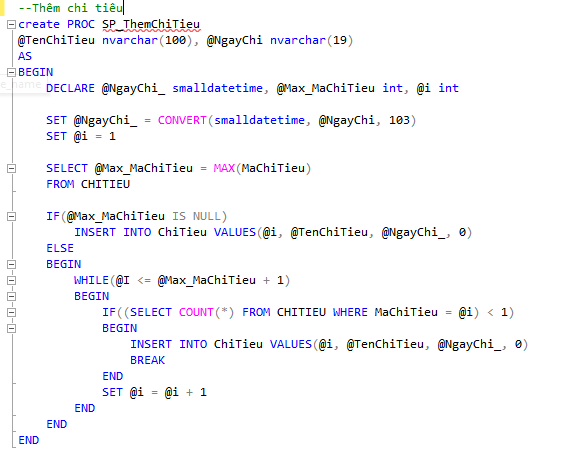
c.10



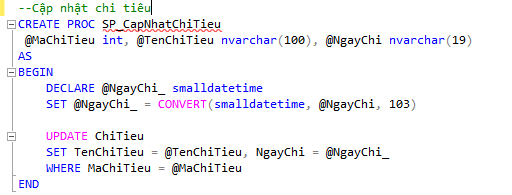
c.11



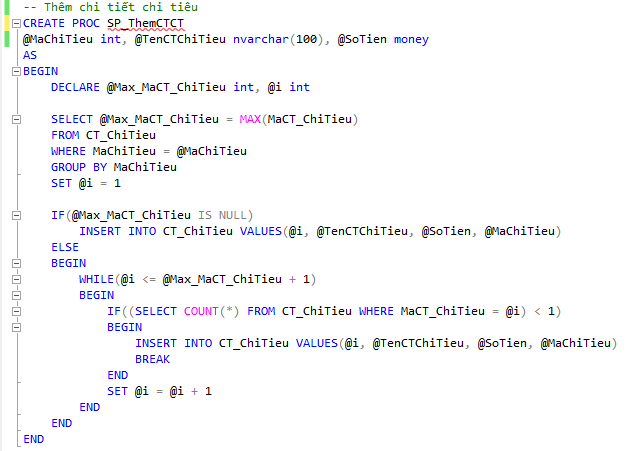
c.12



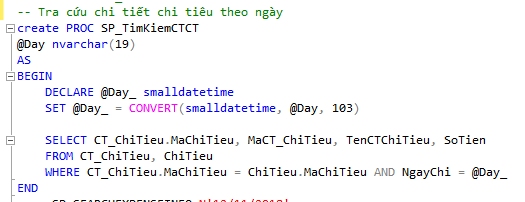
c.13



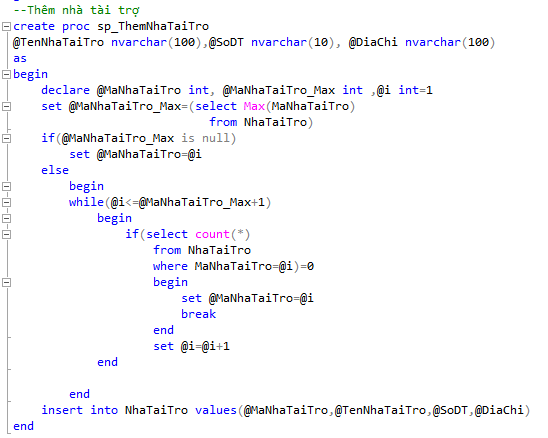
c.14



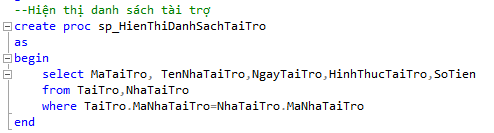
c.15



c.16



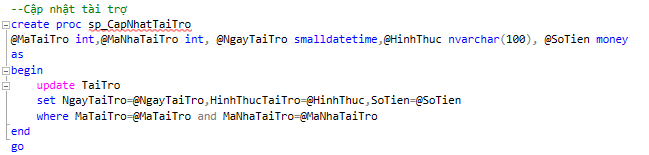
c.17



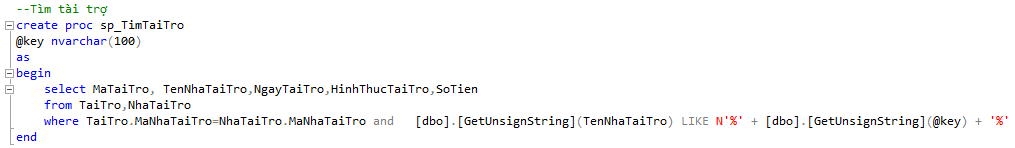
c.18



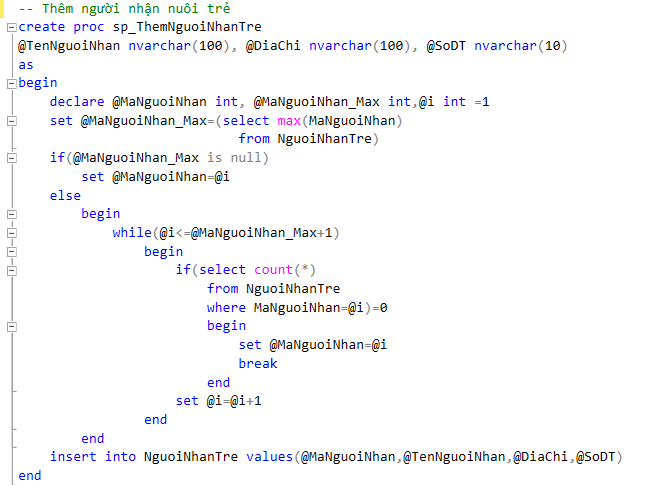
c.19



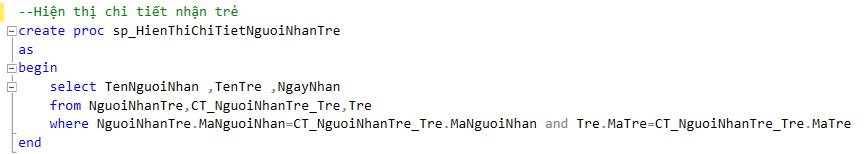
c.20



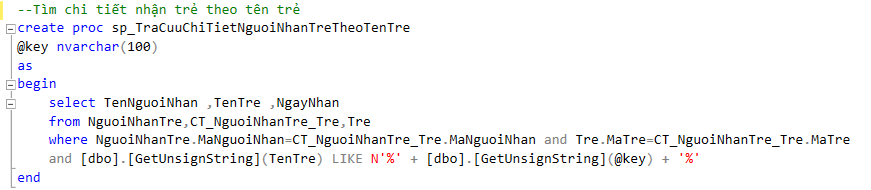
c.21



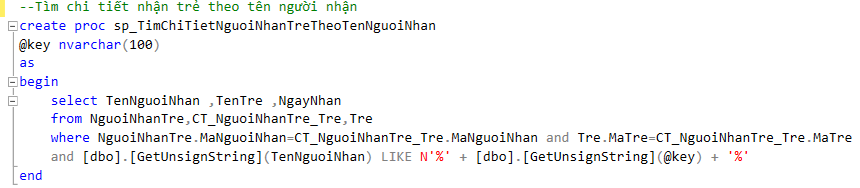
c.22



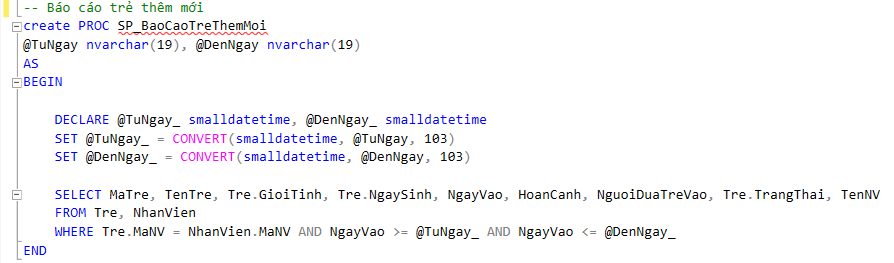
c.23



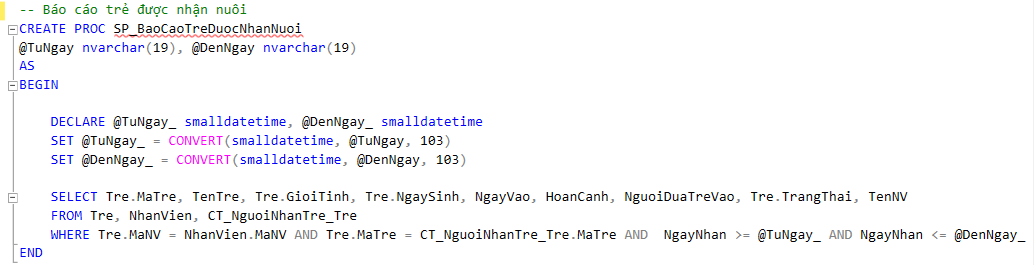
c.24



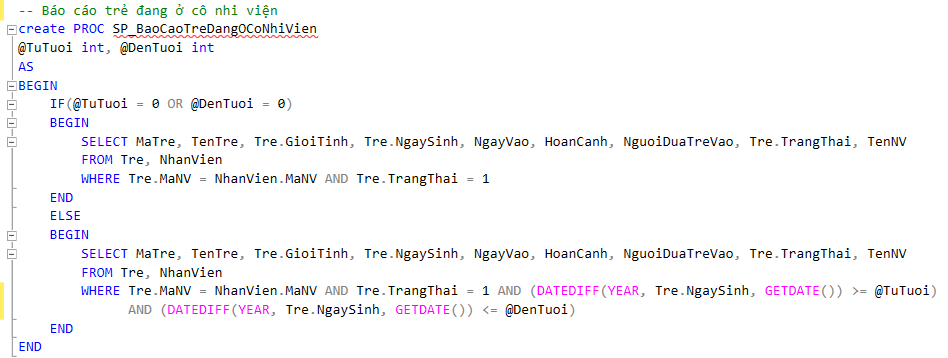
2.25



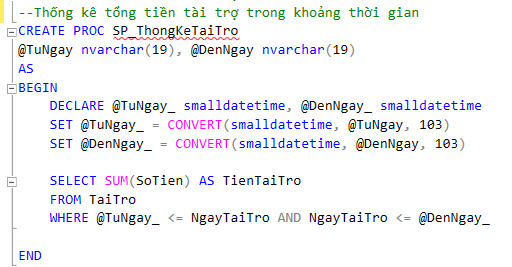
2.26



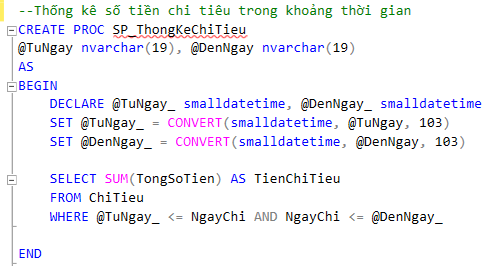
c.27



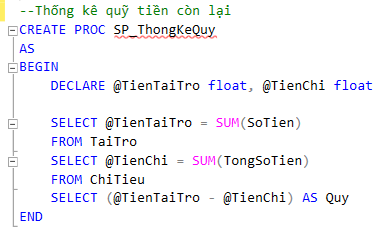
c.28



c.29

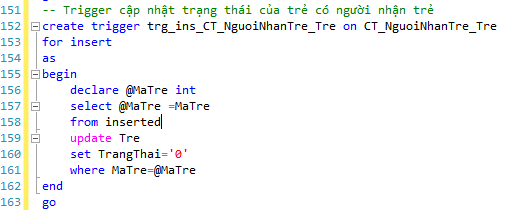


c.30

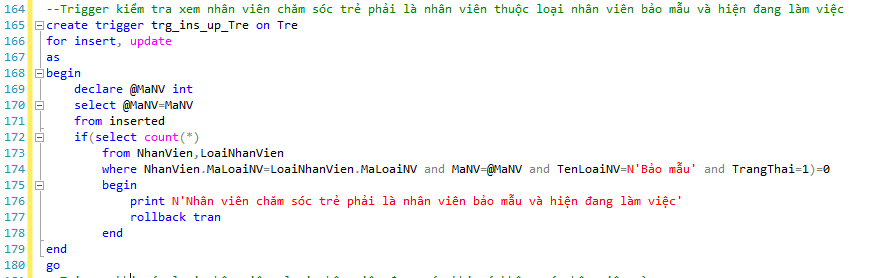


1. **Trigger**
   1. **Trigger trong SQL Server**
   2. **Danh sách các trigger**
   3. **Mô tả các trigger**

**b.1 Khi có người nhận trẻ thì ta sẽ cập nhật lại trạng thái của trẻ**



**b.2 Trigger kiểm tra xem người chăm sóc cho trẻ phải là nhân viên bảo mẫu và hiện đang làm việc hay không**



# CHƯƠNG 5: XỬ LÝ ĐỒNG THỜI

1. **Các mức cô lập trong SQL Server**
2. **Read uncommited**

* Đặc điểm: Không thiết lập Shared Lock trên những đơn vị dữ liệu cần đọc. Do đó không phải chờ khi đọc dữ liệu (kể cả khi dữ liệu đang bị lock bởi giao tác khác. Vẫn tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác)
* Ưu điểm: Tốc độ xử lý rất nhanh. Không cản trở những giao tác khác thực hiện việc cập nhật dữ liệu
* Khuyết điểm: Có khả năng xảy ra mọi vấn đề khi xử lý đồng thời : Dirty Reads, Unrepeatable Reads, Phantoms, Lost Updates

1. **Read commited**

* Đặc điểm: Đây là mức độ cô lập mặc định của SQL Server. Tạo Shared Lock trên đơn vị dữ liệu được đọc, Shared Lock được giải phóng ngay sau khi đọc xong dữ liệu. Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác.
* Ưu điểm: Giải quyết vấn đề Dirty Reads. Shared Lock được giải phóng ngay, không cần phải giữ cho đến hết giao tác nên không cản trở nhiều đến thao tác cập nhật của các giao tác khác.
* Khuyết điểm: Chưa giải quyết được vấn đề Unrepeatable Reads, Phantoms, Lost Updates. Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock)

1. **Repeatable read**

* Đặc điểm: Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác. Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này. (Repeatable Read = Read Committed + Giải quyết Unrepeatable Reads). Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác.
* Ưu điểm: Giải quyết vấn đề Dirty Reads và Unrepeatable Reads .
* Khuyết điểm: Chưa giải quyết được vấn đề Phantoms, do vẫn cho phép insert những dòng dữ liệu thỏa điều kiện thiết lập shared lock. Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock). Shared lock được giữ đến hết giao tác cản trở việc cập nhật dữ liệu của các giao tác khác.

1. **Serializable**

* Đặc điểm: Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ shared lock này đến hết giao tác. Các giao tác khác phải chờ đến khi giao tác này kết thúc nếu muốn cập nhật, thay đổi giá trị trên đơn vị dữ liệu này. Không cho phép Insert những dòng dữ liệu thỏa mãn điều kiện thiết lập Shared Lock (sử dụng Key Range Lock). Serializable = Repeatable Read + Giải quyết Phantoms. Tạo Exclusive Lock trên đơn vị dữ liệu được ghi, Exclusive Lock được giữ cho đến hết giao tác.
* Ưu điểm: Giải quyết thêm được vấn đề Phantoms.
* Khuyết điểm: Phải chờ nếu đơn vị dữ liệu cần đọc đang được giữ khoá ghi (xlock). Cản trở nhiều đến việc cập nhật dữ liệu của các giao tác khác

1. **Transaction**

* Khái niệm: Giao tác (transaction) là 1 tập hợp có thứ tự các thao tác (statement) truy xuất dữ liệu trên CSDL thành 1 đơn vị công việc logic (xem là 1 thao tác nguyên tố), chuyển CSDL từ trạng thái nhất quán này sang trạng thái nhất quán khác.
* Tính ACID: Một giao dịch là một đơn vị rất nhỏ của một chương trình và nó có thể chứa một số nhiệm vụ cấp thấp. Một giao dịch trong hệ thống cơ sở dữ liệu phải duy trì tính nguyên tố, nhất quán, cô lập, và bền vững - thường được gọi là thuộc tính ACID - để đảm bảo độ chính xác, đầy đủ và toàn vẹn dữ liệu.
* Tính nguyên tố: Thuộc tính này đảm bảo mỗi transaction được xử lý như một nguyên tử (một khối duy nhất), được thực hiện trọn vẹn hoặc hoàn toàn không được thực hiện. Nếu có một lỗi nào đó xảy ra trong transaction, nó sẽ được quay trở lại (rollback) trạng thái ban đầu.
* Nhất quán: Thuộc tính này đảm bảo ở mọi thời điểm dữ liệu luôn luôn phải nhất quán, khi transaction được thực hiện hoàn tất, dữ liệu sau khi cập nhật cũng phải ở trạng thái nhất quán. Nếu transaction gây ra những vi phạm về tính nhất quán của cơ sở dữ liệu, hệ thống sẽ không cho phép thực hiện tiếp và hủy bỏ toàn bộ transaction.
* Cô lập: Cũng như các hệ thống server khác, SQL Server có thể đáp ứng nhiều yêu cầu xảy ra đồng thời. Mỗi transaction được đảm bảo thực hiện trong một ngữ cảnh riêng biệt của nó và không bị ảnh hưởng bởi các transaction khác. Khi hai transaction cùng cập nhật một dữ liệu, SQL Server đảm bảo chúng được thực hiện tuần tự không đè lên nhau.
* Bền vững: Khi transaction thực hiện xong (đã commit), những cập nhật trở nên cố định và dữ liệu sẽ luôn luôn là như vậy. Khi hệ thống gặp sự cố bất ngờ, trong quá trình khôi phục lại nó sẽ đảm bảo khôi phục lại dữ liệu cho những transaction đã được commit.
* Các trạng thái:
  + **Active**: Trong trạng thái này, các giao dịch đang được thực hiện. Đây là trạng thái ban đầu của mỗi giao dịch.
  + **Partially Committed:** Sau khi một giao dịch thực hiện hoạt động cuối cùng,  
    nó được cho là trong tình trạng partially committed.
  + **Failed:** Một giao dịch được cho là trong thái failed state nếu có bất kì kiểm  
    tra được thực hiện bởi hệ thống phục hồi cơ sở dữ liệu bị lỗi. Một giao dịch failed không thể tiếp tục thực hiện được nữa.
  + **Aborted:** Sau khi giao dịch đã quay lui và CSDL đã phục hồi lại trạng thái như trước khi bắt đầu giao dịch. Có thể chọn một trong hai cách sau để thực hiện tiếp khi giao dịch trong trạng thái Aborted:  
    • Bắt đầu lại giao dịch  
    • Hủy bỏ các giao dịch
  + **Committed:** Nếu giao dịch đã thực hiện thành công tất cả các hoạt động nó được cho là committed. Tất cả các tác động của giao dịch giờ đây vĩnh viễn được thành lập trên hệ thống cơ sở dữ liệu.
* Trong môi trường truy xuất đồng thời, có thể xảy ra một số vấn đề như sau:
  1. **Lost update**

**2.1.1**

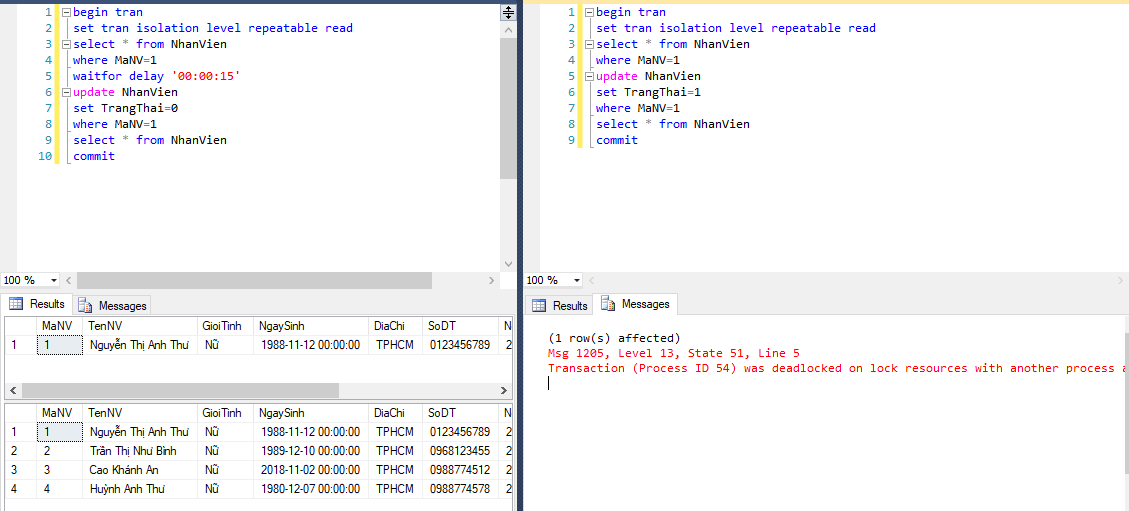


Đây là trường hợp Lost Update, xảy ra khi 2 transaction đọc và cập nhật trên cùng  
một đơn vị dữ liệu.  
− Transaction 1 (T1) cập nhật trạng thái của nhân viên 1 là 0 (“Đã nghỉ việc”)  
− Transaction 2 (T2) cập nhật lại trạng thái của nhân viên 1 là 1 (“Đang làm việc”)

* + Trạng thái của T1 là 0
  + Trạng thái của T2 là 1

=> Như vậy T1 bị mất dữ liệu do T2 cập nhật.

**Giải quyết vấn đề**

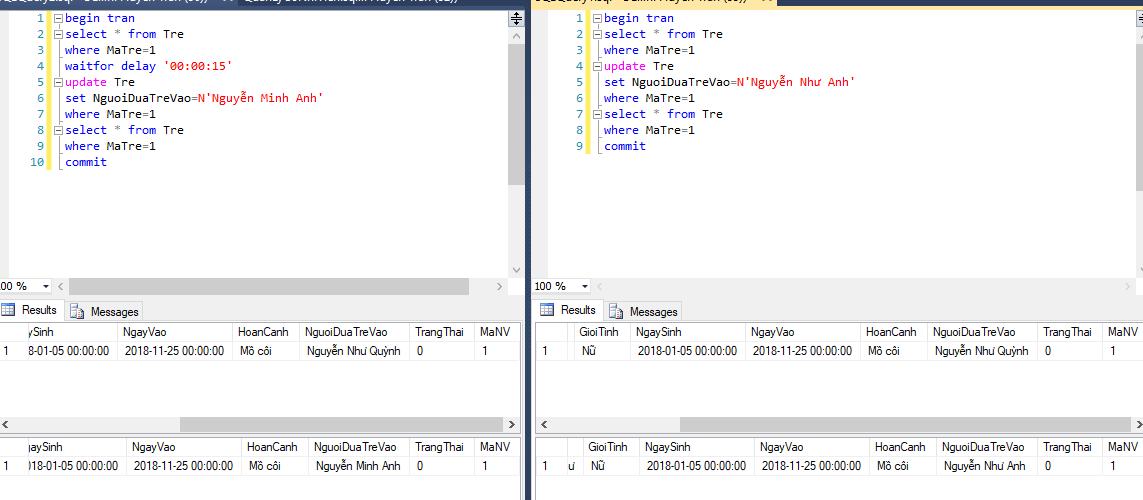


Các mức cô lập Repeatable Read, SnapShot hoặc Serializable sẽ không gây ra trường hợp này. Do đó ta có để đặt mức cô lập cho Tran này từ mức Repeatable Read trở lên.  
− SET TRANSACTION ISOLATION LEVEL REPEATABLE READcho T1 sẽ đảm bảo dữ liệu mà T1 cập nhật, ngăn chặn việc update hay delete ở T2  
− Sau khi T1 commit thì T2 mới có thể update hay delete.  
− Khi chạy T1 rồi chạy T2:

* T2 bị chọn là nạn nhân của deadlock và bị rollback
* T1 thực hiện thành công( vì T1 chiếm lock trên đơn vị dữ liệu này)

**2.1.2**

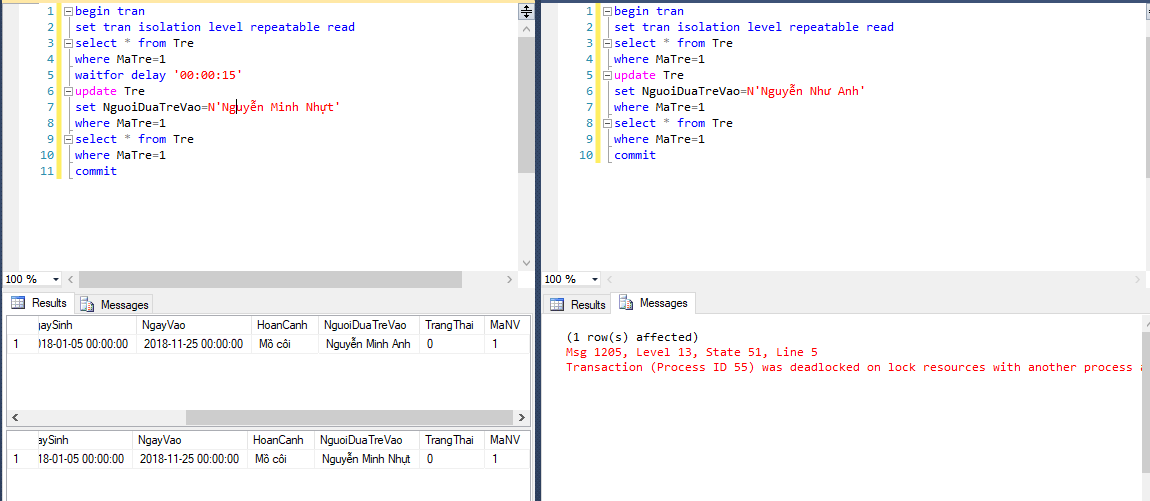
T1 và T2 cùng phát hiện dữ liệu của trẻ thứ nhất bị sai thông tin người đưa trẻ vào, và tiến hành cập nhật.



Đây là trường hợp Lost Update, xảy ra khi 2 transaction đọc và cập nhật trên cùng  
một đơn vị dữ liệu.  
− Transaction 1 (T1) cập nhật người đưa trẻ vào là Nguyễn Minh Anh  
− Transaction 2 (T2) cập nhật người đưa trẻ vào là Nguyễn Như Anh

=> Như vậy T1 bị mất dữ liệu do T2 cập nhật.

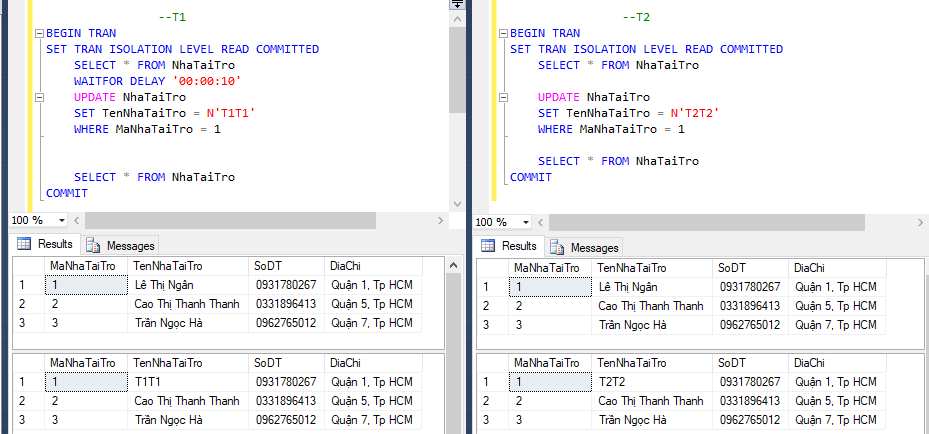
- Giải quyết vấn đề



Các mức cô lập Repeatable Read, SnapShot hoặc Serializable sẽ không gây ra trường hợp này. Do đó ta có để đặt mức cô lập cho Tran này từ mức Repeatable Read trở lên.  
− SET TRANSACTION ISOLATION LEVEL REPEATABLE READcho T1 sẽ đảm bảo dữ liệu mà T1 cập nhật, ngăn chặn việc update hay delete ở T2  
− Sau khi T1 commit thì T2 mới có thể update hay delete.  
− Khi chạy T1 rồi chạy T2:

* T2 bị chọn là nạn nhân của deadlock và bị rollback
* T1 thực hiện thành công( vì T1 chiếm lock trên đơn vị dữ liệu này)

2.1.3



**Đặt vấn đề**

* T1 và T2 cùng cập nhật trên bảng NhaTaiTro.
* Ở mã nhà tài trợ 1, T1 cập nhật tên nhà tài trợ thành ‘T1T1’, T2 cập nhật tên nhà tài trợ thành ‘T2T2’.
* T1 đợi 10s trước khi cập nhật, khi đó tác dụng cập nhật của T1 sẽ ghi đè lên tác dụng cập nhật của T2, làm cho dữ liệu cập nhật của T2 bị mất.

**Giải quyết vấn đề**

Ta đặt lại mức cô lập thành REPEATABLE READ hoặc SERIALIZABLE



**2.2 Dirty read**

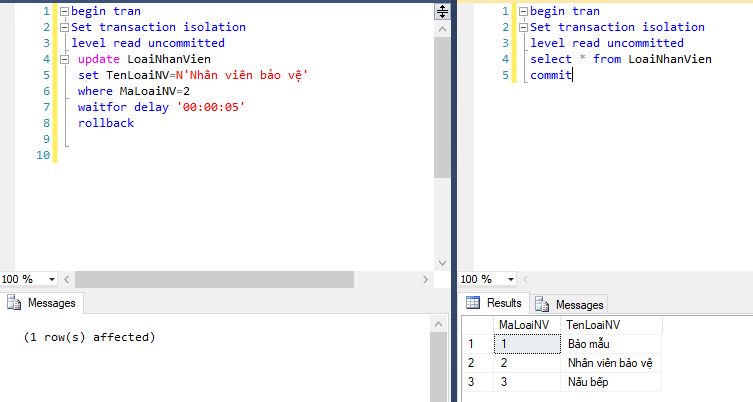
2.2.1

**Đặt vấn đề:**

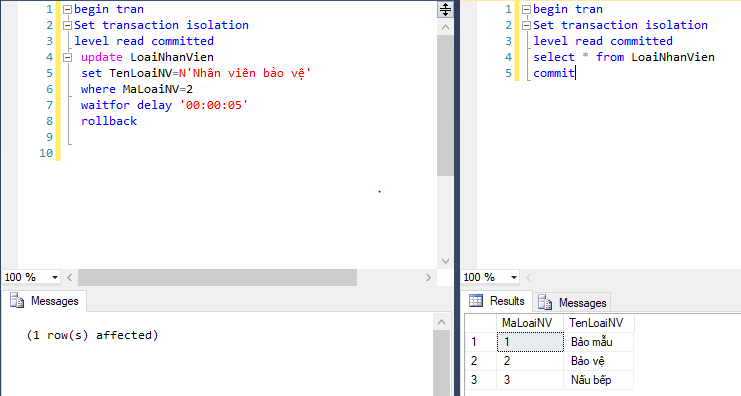
**-** T1 thực hiện update TenLoaiNV=’Nhân viên bảo vệ’ khi MaLoaiNV=2 ở bảng LoaiNhanVien . Nhưng sau đó rollback.

- T2 thực hiện đọc dữ liệu ở bảng LoaiNhanVien, nhưng do ở mức cô lập read uncommitted dẫn đến hiện tượng đọc dữ liệu rác.

- Dirty Read xảy ra khi một tran được phép đọc dữ liệu mà tran khác đang cập nhật  
mà thao tác này chưa commit. Do đó khi T1 rollback sau một vài giây, thì T2 đọc  
được dữ liệu rác là dữ liệu này không tồn tại.



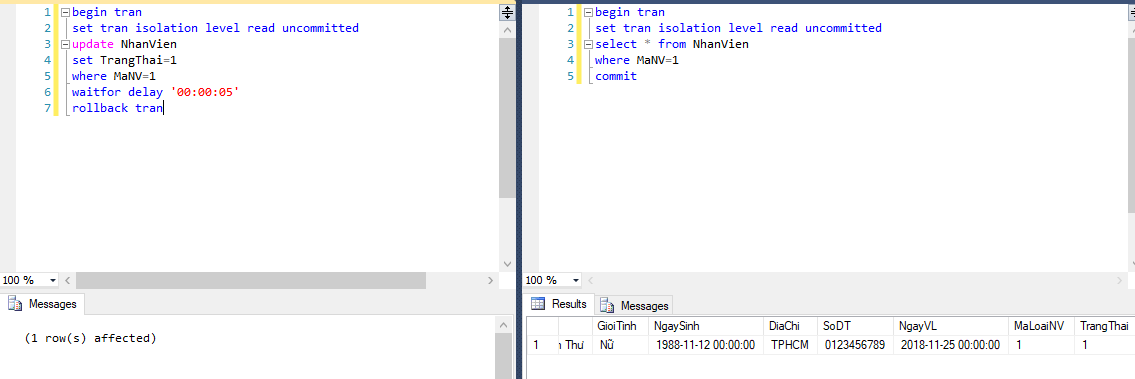
**- Giải quyết vấn đề:** ta sẽ đặt mức cô lập là read commited.



2.2.2

**Đặt vấn đề:**

* T1 thực hiện update TrangThai của nhân viên có MaNV=1 , nhưng sau đó T1 hủy cập nhật
* T2 thực hiện đọc dữ liệu của nhân viên có MaNV=1 nhưng do ở mức cô lập read uncommitted nên dẫn đến hiện tượng dirty read

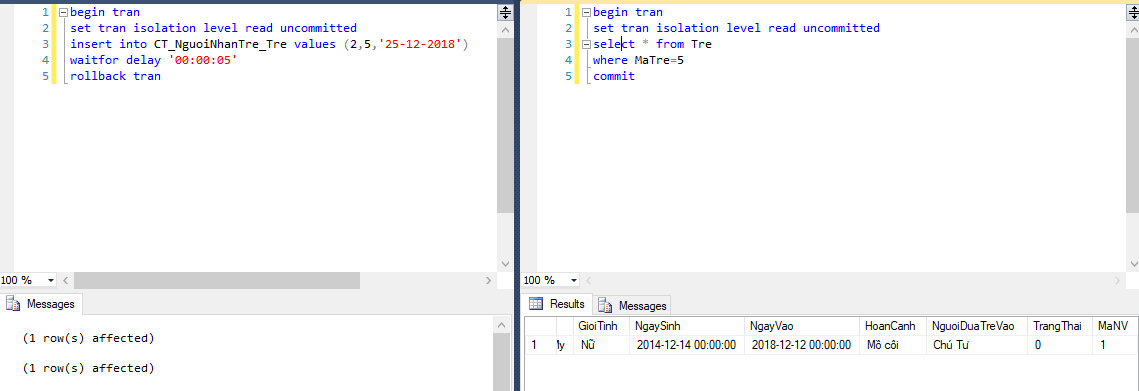


**Giải quyết vấn đề**: ta đặt mức cô lập là read commited nghĩa là T2 chờ cho tới khi T1 cập nhật dữ liệu xong thì T2 mới tiến hành đọc dữ liệu.

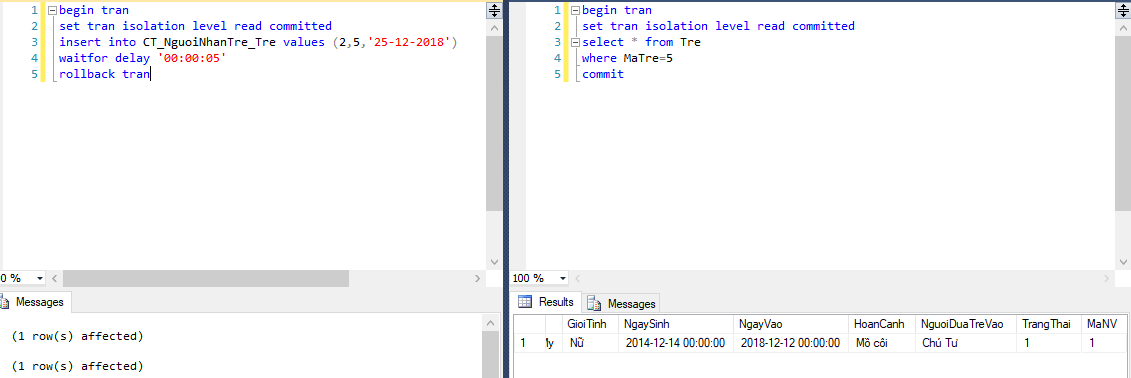
2.2.3

**Đặt vấn đề**

* T1 tiến hành thêm 1 chi tiết nhận trẻ nhưng sau đó T1 hủy, trạng thái của trẻ vẫn là 1 (trẻ vẫn còn ở cô nhi viện)
* T2 đọc dữ liệu trên bảng trẻ nhưng do ở mức cô lập read uncommited nên T2 đọc dữ liệu rác , trạng thái của trẻ là 0 (trẻ không còn ở cô nhi viện).



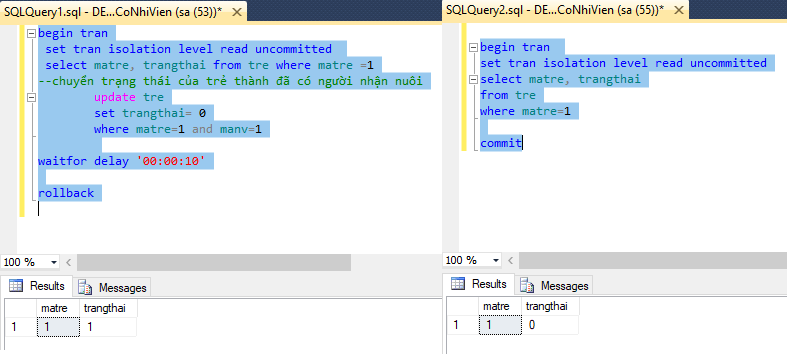
**Giải quyết vấn đề**: ta đặt mức cô lập read committed



2.2.4

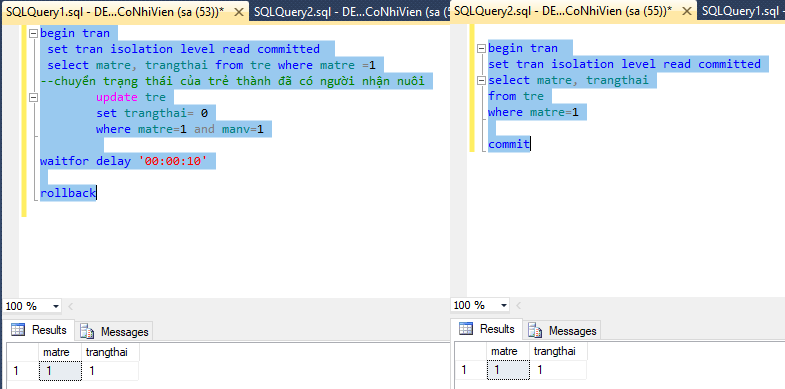
**Đặt vấn đề:**

* + Một nhân viên đang cập nhật trạng thái của trẻ từ đang ở cô nhi viện( trạng thái= 1) thành trẻ đã được nhận nuôi không còn ở cô nhi viện( trạng thái= 0) sau đó tiến hành hủy giao tác.
  + Một nhân viên khác vào xem trạng thái của trẻ thì trẻ không còn ở cô nhi viện (trạng thái= 0)



**Nguyên nhân:** Do T2 thiết lập mức cô lập là read uncommitted nên khi T1 đang cập nhật trạng thái nhưng chưa kịp rollback mà T2 đã được gọi thực hiện thì T2 vẫn sẽ lấy TrangThai của T1 đã update.

**Giải pháp:** Chuyển mức cô lập của T2 thành read committed vì mức cô lập này chỉ cho phép transaction đọc dữ liệu của các transaction đã commit.

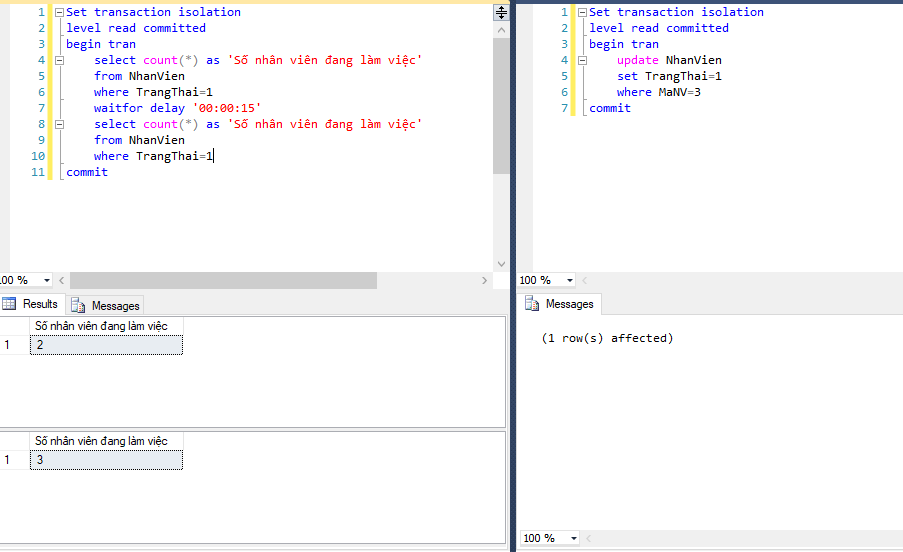


* 1. **Non - repeatable Read**

2.3.1

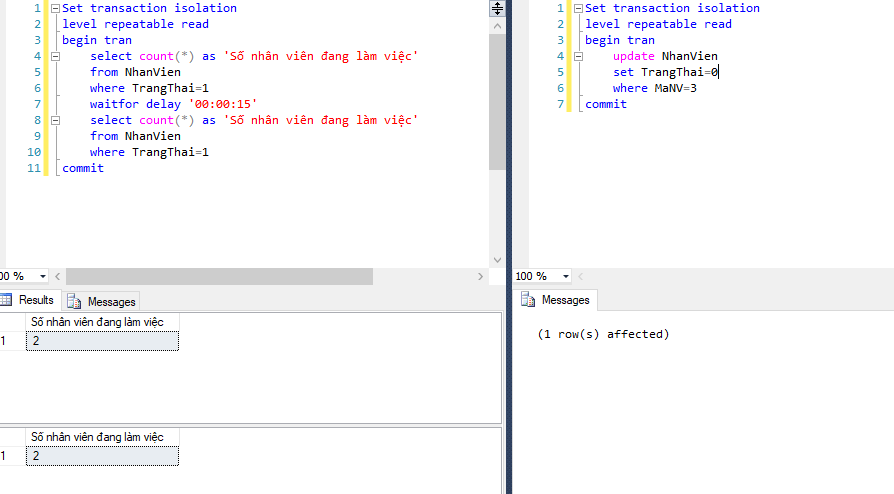
**Đặt vấn đề:**

- Read Committed: Unrepeatable Read (nhận 2 kết quả khác nhau giữa 2 lần đọc mặc dù phiên làm việc vẫn chưa kết thúc)



Số nhân viên đang làm việc ở câu lệnh select count(\*) thứ 2 trả về giá trị là 3 do nhân viên có MaNV=3 đã được cập nhật trạng thái là 1 ở T2. Chính vì thế gây ra tình trạng dữ liệu ở 2 lần đọc trong cùng một giao tác không đồng nhất.

- **Giải quyết vấn đề** ta sẽ sử dụng mức cô lập repeatable read.

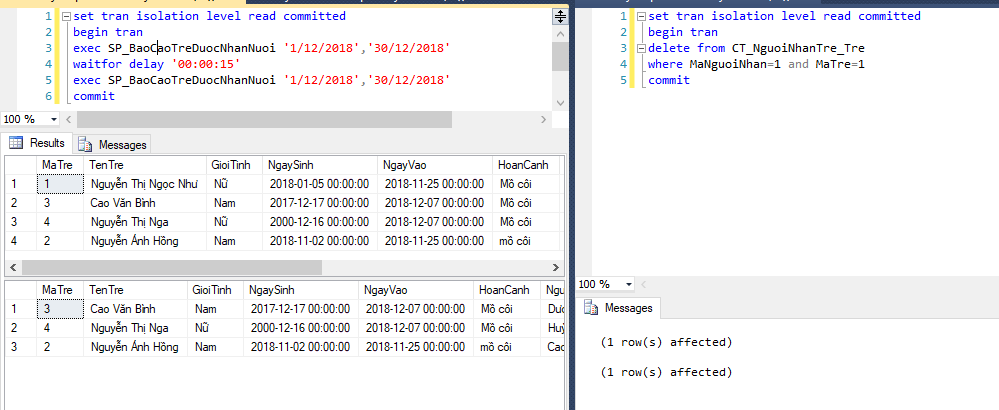
 Do ở mức cô lập repeatable read nên T2 chờ T1 thực thi xong thì mới bắt đầu, nghĩa là dữ liệu đọc sẽ được bảo vệ khỏi các thao tác cập nhật khác. Đảm bảo thao tác đọc trong cùng 1 transaction sẽ cho cùng một kết quả.

2.3.2 **Đặt vấn đề**

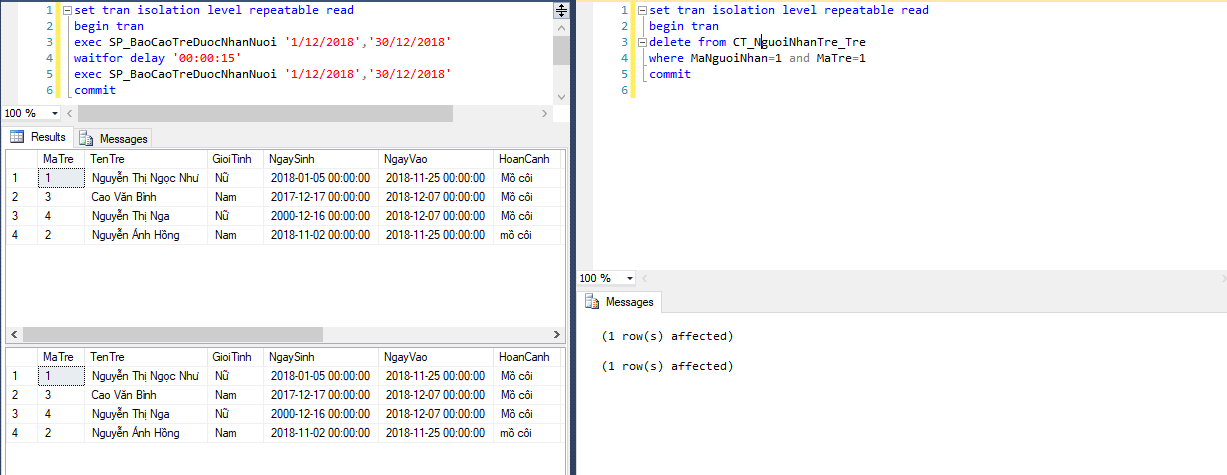
- T1 thực hiện 2 câu lệnh để xem danh sách trẻ đang được nhận nuôi từ ngày 1/12/2018 đến ngày 20/12/2018.

- T2 thực hiện xóa chi tiết nhận trẻ có mã người nhận là 1 và mã trẻ là 4.

- 2 câu lệnh ở T1 cho 2 kết quả khác nhau, điều này có nghĩa là dữ liệu đọc ở T1 không được bảo vệ khỏi thao tác cập nhật khi có hiện tượng thêm, xóa chi tiết nhận trẻ. Gây ra hiện tượng không đồng nhất dữ liệu trong một giao tác.



**Giải quyết vấn đề:** ta sẽ sử dụng mức cô lập repeatable read.



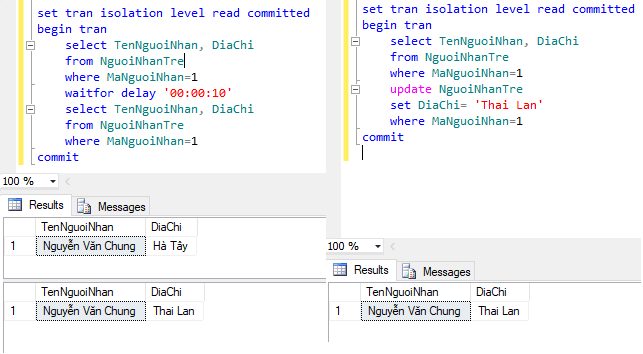
Do ở mức cô lập repeatable read nên T2 chờ T1 thực thi xong thì mới bắt đầu, nghĩa là dữ liệu đọc sẽ được bảo vệ khỏi các thao tác cập nhật khác. Đảm bảo thao tác đọc trong cùng 1 transaction sẽ cho cùng một kết quả.

2.3.4

**Đặt vấn đề:**

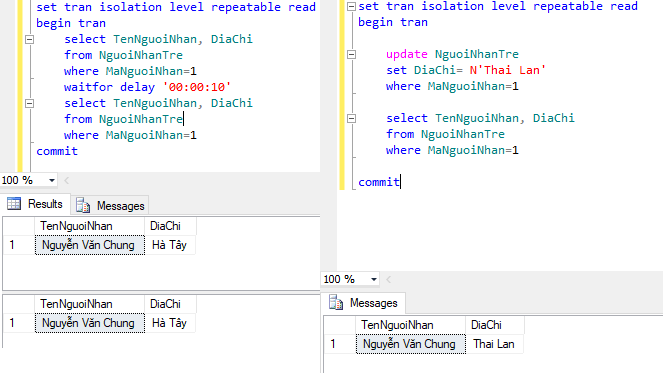
Khi thay đổi địa chỉ của người nhận nuôi trẻ. Ban đầu T1 đọc thông tin của người nhận nuôi trẻ có MaNguoiNhan = 1. T2 vào thay đổi thông tin địa chỉ của người nhận trẻ , sau đó commit lên CSDL. T1 trở lại đọc dữ liệu nhưng không thể đọc dữ liệu ban đầu.

Người nhận nuôi trẻ 1 ban đầu có địa chỉ là Hà Tây



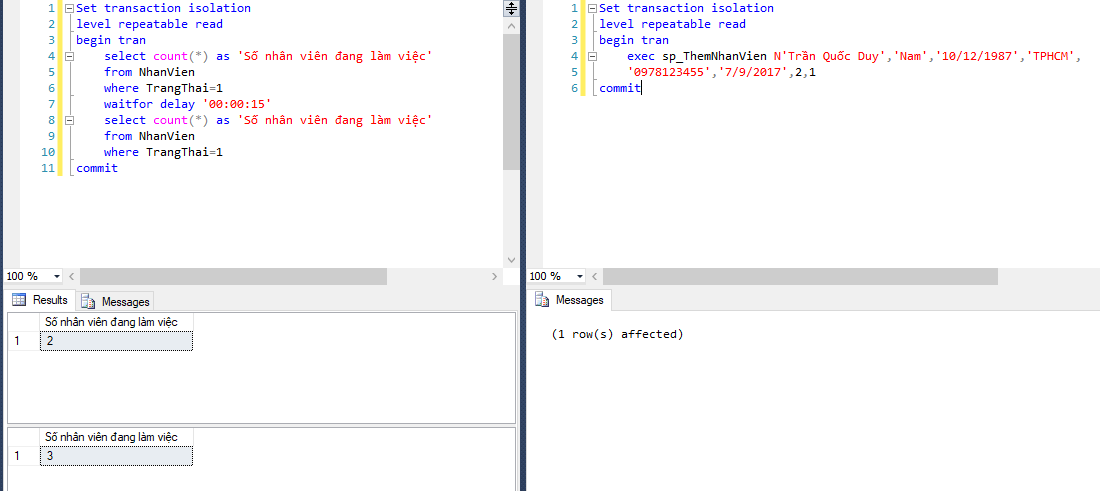
**Nguyên nhân:** Do ở mức độ cô lập Read Commited nên Shared Lock được giải phóng ngay, không cần phải giữ cho đến hết giao tác nên không cản trở nhiều đến thao tác cập nhật của giao tác T2 => Unrepeatable Read.

**Hướng giải quyết:** Dùng mức cô lập Repeatable Read. Khi chạy dòng lệnh select sẽ tạo Share lock và giữ Share lock này đến hết giao tác. T2 phải đợi T1 thực hiện xong mới update.



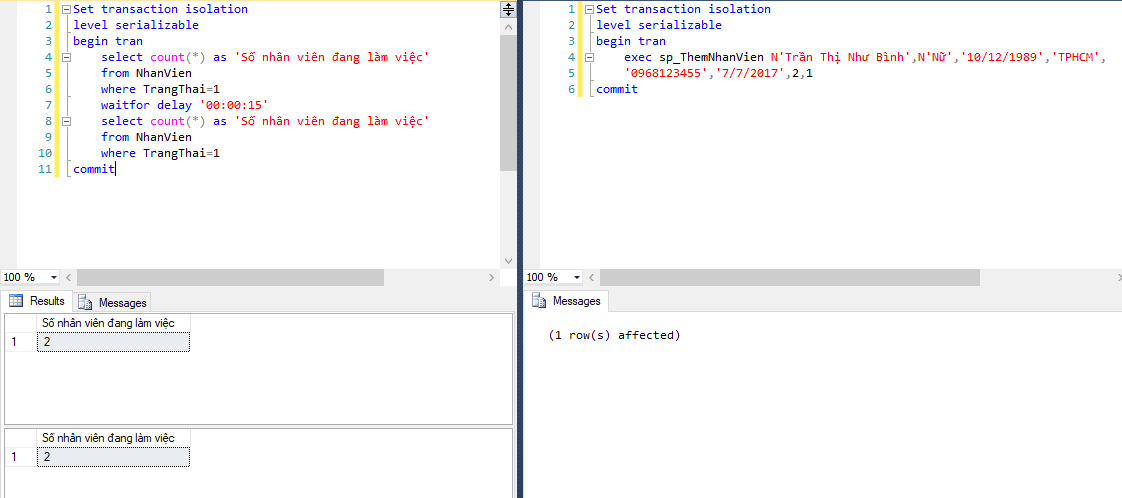
* 1. **Phantom**

**2.4.1**

* Reapeatable Read: không ngăn được Insert dữ liệu từ giao tác khác chèn vào (phantom).
* Đặt vấn đề: T1 có 2 câu lệnh đếm số nhân viên đang làm việc, T2 thực hiện thêm mới nhân viên.

Kết quả trả về ở câu lệnh select count(\*) ở T1 có giá trị là 3 trong đó trả về 3 bộ dữ liệu thì có 1 bộ dữ liệu là phantom từ kết quả thực hiện T2.

* **Giải quyết vấn đề**: đặt mức cô lập Serializable để giải quyết Phantom (phát sinh share lock).

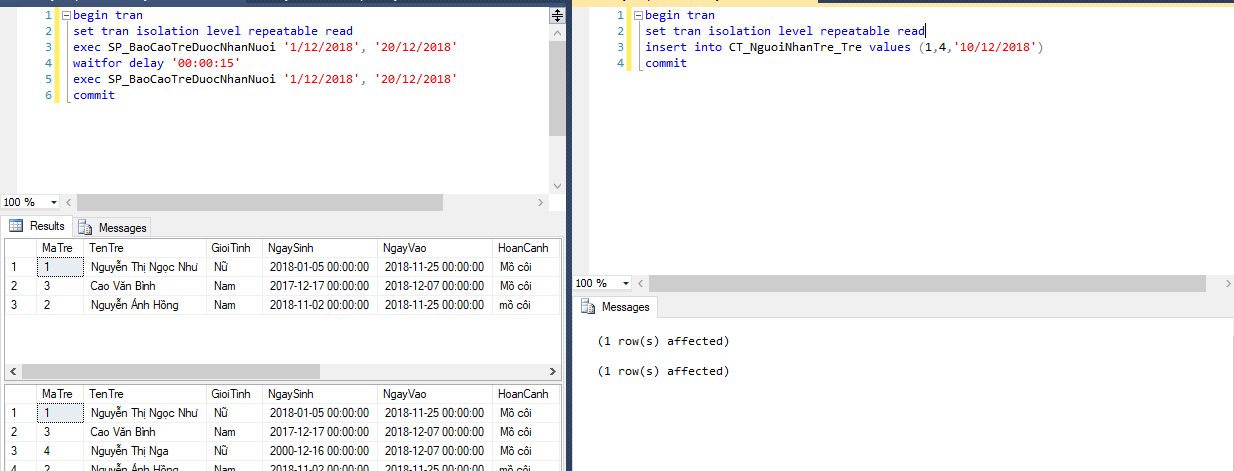


Ở câu lệnh select đầu tiên trong T1 trả về số lượng nhân viên đang làm việc là 2. Do chúng ta đặt mức cô lập là serializable nên dữ liệu read ở T1 sẽ được bảo vệ khỏi dữ liệu write ở T2 điều này có nghĩa là T2 sẽ không thực thi cho đến khi T1 thực thi xong, thì T2 mới được thực thi.

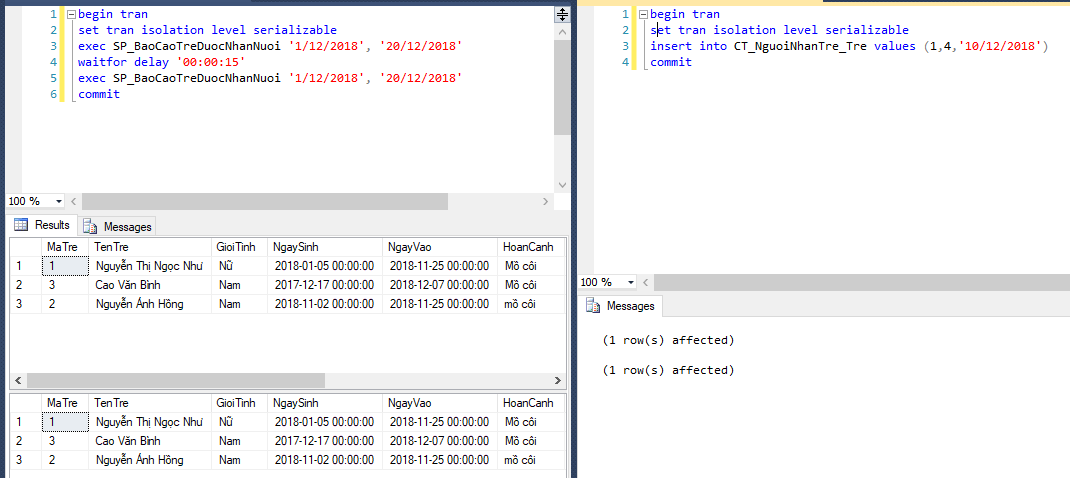
2.4.2

**Đặt vấn đề**

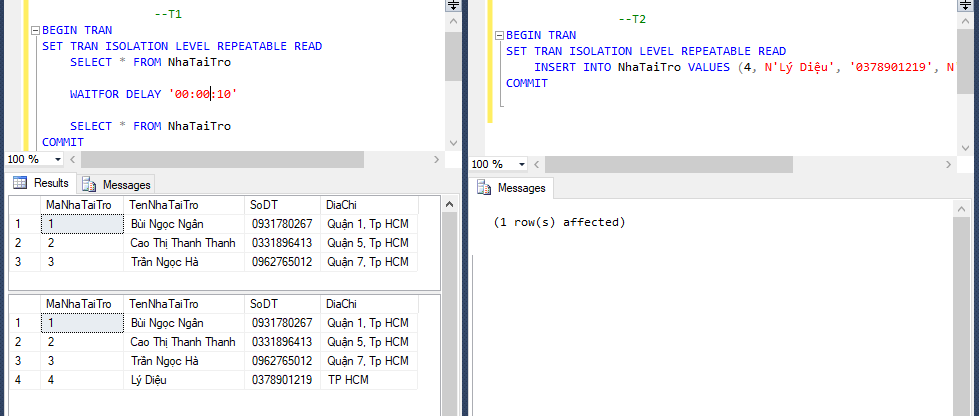
* T1 thực hiện 2 câu lệnh để xem danh sách trẻ đang được nhận nuôi từ ngày 1/12/2018 đến ngày 20/12/2018.
* T2 thực hiện thêm chi tiết nhận trẻ có mã người nhận là 1 và mã trẻ là 4.
* 2 câu lệnh ở T1 cho 2 kết quả khác nhau, điều này có nghĩa là dữ liệu đọc ở T1 không được bảo vệ khỏi thao tác cập nhật khi có hiện tượng thêm, xóa chi tiết nhận trẻ. Gây ra hiện tượng không đồng nhất dữ liệu trong một giao tác.



**Giải quyết vấn đề:** ta đặt mức cô lập là serializable, lúc đó dữ liệu đọc ở T1 sẽ được bảo vệ khỏi các thao tác thêm, xóa, sửa. Nghĩa là T1 thực thi xong thì T2 mới được phép thực thi.



2.4.3

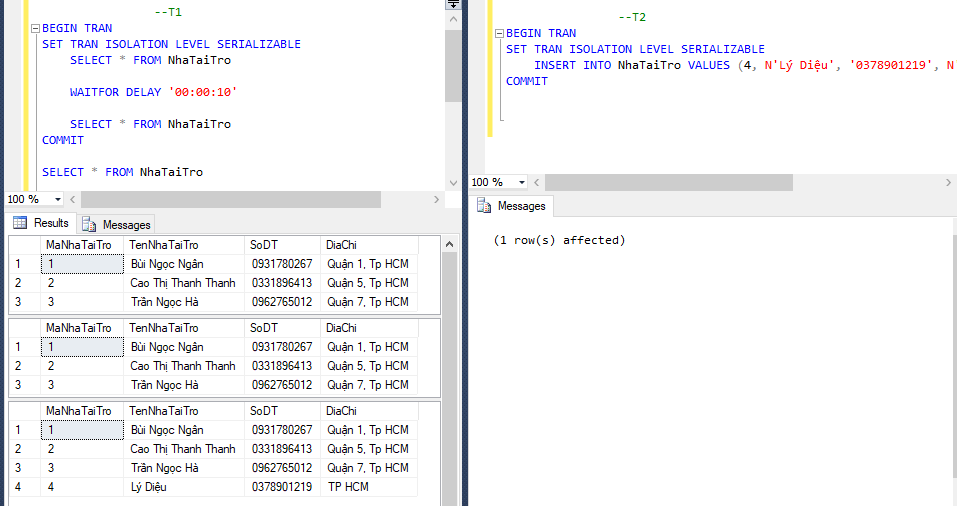


**Đặt vấn đề**

* T1 và T2 đều thực hiện các thao tác trên bảng NhaTaiTro
* Lần đọc thứ nhất ở T1 cho danh sách có 3 nhà tài trợ, khi T1 chưa kết thúc thì T2 thêm 1 nhà tài trợ có MaNhaTaiTro = 4
* Sau khi T2 thực hiện xong, T1 đọc lần thứ 2 cho danh sách có 4 nhà tài trợ
* Kết quả ở 2 lần đọc của T1 khác nhau => Phantoms

**Các giải quyết**

* Ta đặt mức cô lập thành SERIALIZABLE
* T1 thực hiện xong và kết thúc rồi mới tới T2 thực hiện, do đó 2 lần đọc của T1 có cùng kết quả



1. **Lock**

Một số chế độ khóa SQL Server cung cấp :

* 1. Readuncomitted/ Nolock: Không thiết lập shared lock khi đọc (tương tự mức cô lập read uncommitted)
  2. Readcommitted
* Đây là chế độ mặc định (tương tự mức cô lập read committed)
* Chỉ đọc những dữ liệu đã được commit
* Thiết lập shared lock trên đơn vị dữ liệu cần đọc và mở lock ra ngay sau khi đọc xong
  1. Repeatableread: Thiết lập shared lock khi select và giữ shared lock đến hết giao tác (tương tự mức cô lập repeatable read)
  2. Serializable/ Holdlock
* Thiết lập shared lock khi đọc và giữ đến hết giao tác
* Tương tự như sử dụng Isolation Level là Serializable

1. Updlock: Sử dụng Updatelock thay vì Shared lock.
2. Xlock: Khoá độc quyền
3. Readpast: Chỉ có thể sử dụng trong lệnh Select và chỉ áp dụng trên khóa của dòng dữ liệu (row- lock). Những dòng bị khóa sẽ được bỏ qua.
4. Rowlock: Khóa chỉ những dòng cần thao tác
5. Tablock

* Khóa toàn bộ bảng trong CSDL.
* Các thao tác cập nhật (insert/ delete/ update) của những giao tác khác không thể thực hiện trên bảng này trong khi khóa vẫn đang được giữ.

1. Tablockx: Xlock + tablock
2. **Deadlock**

* Khái niệm: Là tình trạng 2 hay nhiều giao tác đang tranh chấp tài nguyên và phải chờ các giao tác còn lại hoàn tất, kết quả là không 1 giao tác nào thực hiện được.
* Phân loại: có 2 loại deadlock:

+ Cycle deadlock



+ Conversion deadlock



* Phương pháp ngăn ngừa deadlock:

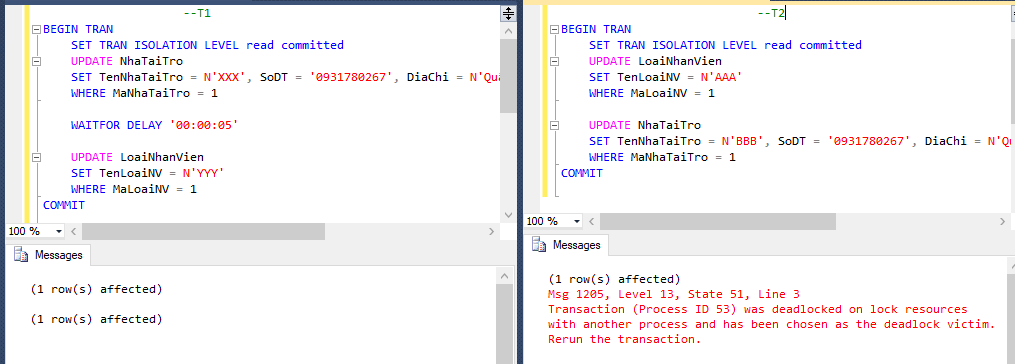
+ Nguyên lý: Trước khi hệ thống chấp nhận 1 yêu cầu (thao tác) từ 1 giao tác lên cơ sở dữ liệu (CSDL), hệ thống sẽ kiểm tra xem CSDL này có đang bị tranh chấp không đồng thời dự báo xem nếu chấp nhận yêu cầu này có thể đưa hệ thống vào tình trạng Deadlock không. Nếu có thì hệ thống tạm ngưng hoặc hủy bỏ yêu cầu.

+ Yêu cầu: Hệ thống cần biết các tài nguyên đang được sử dụng, đang tranh chấp,…

4.1 Các tình huống xảy ra deadlock và cách giải quyết

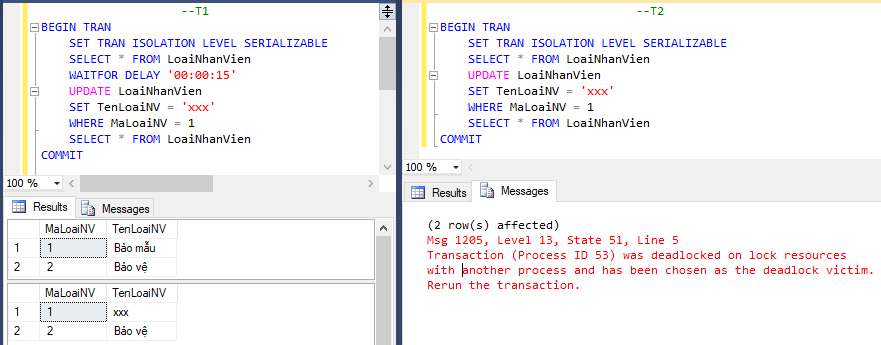
4.1.1

a) Cycle deadlock

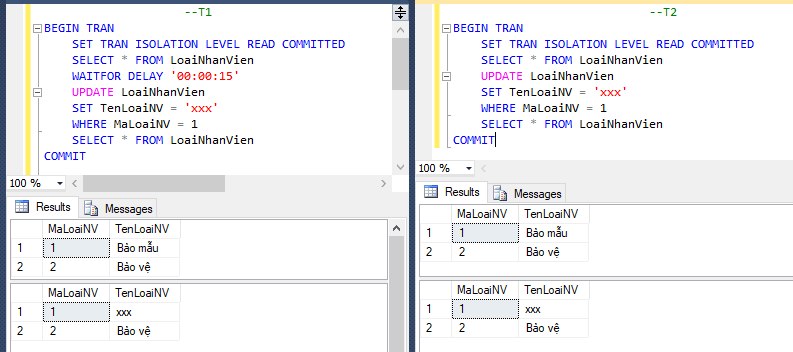


* Vấn đề: Ta xét mức cô lập READ COMMITTED: tạo Exclusive Lock trên đơn vị dữ liệu được ghi và giữ đến hết giao tác. T1 tạo Exclusive Lock trên bảng NhaTaiTro và cập nhật trên bảng này. T2 tạo Exclusive Lock trên bảng LoaiNhanVien và cập nhật trên bảng này. T1 xin lock trên bảng LoaiNhanVien nhưng bảng LoaiNhanVien đang được Exclusive Lock bởi T2. T2 xin lock trên bảng NhaTaiTro nhưng bảng NhaTaiTro đang được Exclusive Lock bởi T1. Hay nói cách khác T1 chờ T2 trả khóa ghi trên LoaiNhanVien, T2 chờ T1 trả khóa ghi trên NhaTaiTro -> Hệ thống bị treo.
* Giải pháp: Để giải quyết deadlock, hệ quản trị sẽ tự động rollback transaction có kích cỡ nhỏ hơn. Kích cỡ của 1 transaction được xác định bởi số dòng insert, update or delete. Trong trường hợp này, 2 transaction có kích cỡ bằng nhau nên hệ quản trị sẽ rollback transaction T2 thực hiện sau.

b) Conversion deadlock

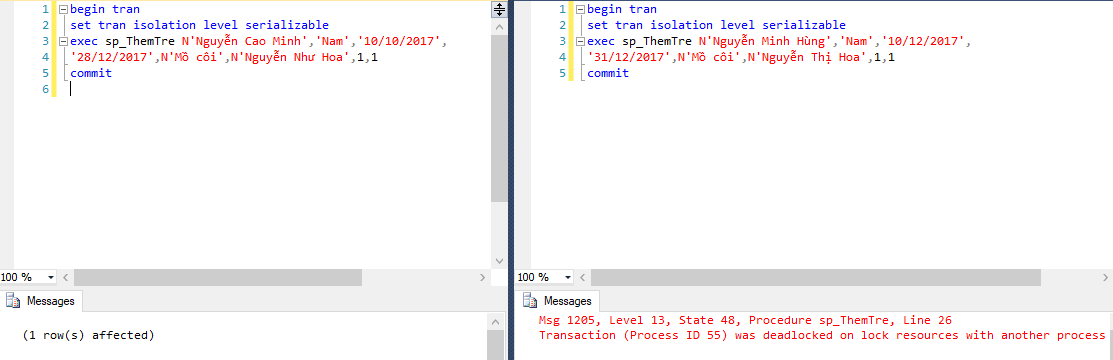


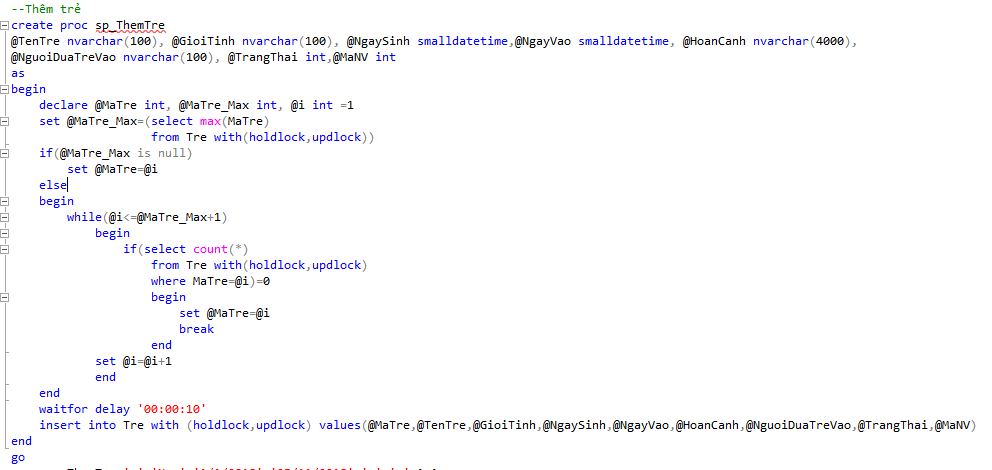
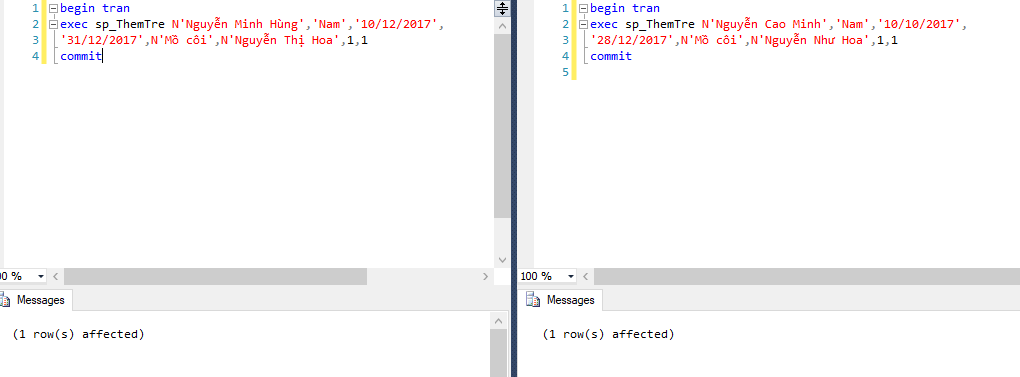
* Vấn đề: ta xét mức cô lập SERIALIZABLE: Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ Shared Lock này đến hết giao tác. Tạo Exclusive Lock trên đơn vị dữ liệu được ghi và giữ Exclusive Lock cho đến hết giao tác. T1 chờ T2 trả khóa đọc trên bảng LoaiNhanVien và T2 chờ T1 trả khóa đọc trên bảng LoaiNhanVien. => Gây ra hiện tượng deadlock.
* Giải pháp: Để giải quyết deadlock, hệ quản trị sẽ tự động rollback transaction có kích cỡ nhỏ hơn. Kích cỡ của 1 transaction được xác định bởi số dòng insert, update or delete. Trong trường hợp này, 2 transaction có kích cỡ bằng nhau nên hệ quản trị sẽ rollback transaction T2 thực hiện sau.
* Cách khắc phục: ta đặt lại mức cô lập thành READ COMMITTED, sau khi đọc xong transaction sẽ giải phóng khóa đọc do đó không gây ra hiện tượng chờ transaction khác trả khóa đọc => Không deadlock.



4.1.2

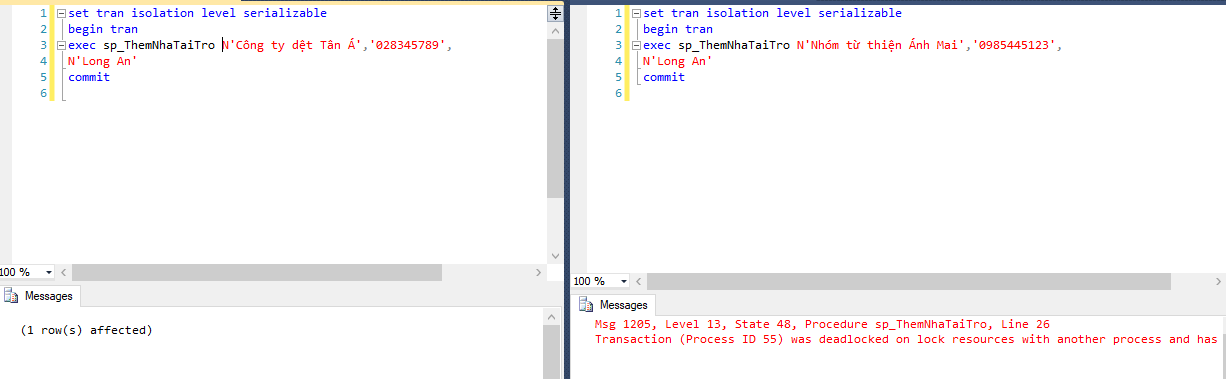
* Đặt vấn đề: nhân viên thứ nhất tiến hành thêm thông tin cho trẻ, nhân viên thứ hai lại tiến hành thêm thông tin cho một trẻ khác. Ta xét mức cô lập SERIALIZABLE: Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ Shared Lock này đến hết giao tác. Tạo Exclusive Lock trên đơn vị dữ liệu được ghi và giữ Exclusive Lock cho đến hết giao tác. T1 chờ T2 trả khóa đọc trên bảng Tre và T2 chờ T1 trả khóa đọc trên bảng Tre. => Gây ra hiện tượng deadlock



* Giải pháp: Để giải quyết deadlock, hệ quản trị sẽ tự động rollback transaction có kích cỡ nhỏ hơn. Kích cỡ của 1 transaction được xác định bởi số dòng insert, update or delete. Trong trường hợp này, 2 transaction có kích cỡ bằng nhau nên hệ quản trị sẽ rollback transaction T2 thực hiện sau.
* Cách khắc phục: ta đặt khóa updlock, holdlock vào các câu lệnh insert và select trong proc sp\_ThemTre. HoldLock thiết lập shared lock khi đọc và giữ đến hết giao tác tương tự như sử dụng Isolation Level là Serializable. Cỏn Updlock tự như Sharedlock.
* Kết quả thu được:

4.1.3

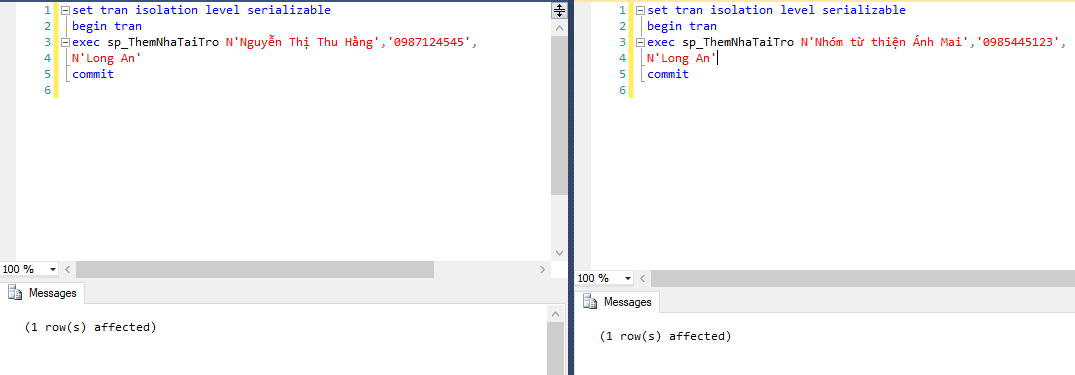
* Đặt vấn đề: nhân viên thứ nhất tiến hành thêm thông tin cho nhà tài trợ, nhân viên thứ hai lại tiến hành thêm thông tin cho một nhà tài trợ khác. Ta xét mức cô lập SERIALIZABLE: Tạo Shared Lock trên đơn vị dữ liệu được đọc và giữ Shared Lock này đến hết giao tác. Tạo Exclusive Lock trên đơn vị dữ liệu được ghi và giữ Exclusive Lock cho đến hết giao tác. T1 chờ T2 trả khóa đọc trên bảng NhaTaiTro và T2 chờ T1 trả khóa đọc trên bảng NhaTaiTro. => Gây ra hiện tượng deadlock



* Giải pháp: Để giải quyết deadlock, hệ quản trị sẽ tự động rollback transaction có kích cỡ nhỏ hơn. Kích cỡ của 1 transaction được xác định bởi số dòng insert, update or delete. Trong trường hợp này, 2 transaction có kích cỡ bằng nhau nên hệ quản trị sẽ rollback transaction T2 thực hiện sau.
* Cách khắc phục: ta đặt khóa updlock, holdlock vào các câu lệnh insert và select trong proc sp\_ThemTre. HoldLock thiết lập shared lock khi đọc và giữ đến hết giao tác tương tự như sử dụng Isolation Level là Serializable. Cỏn Updlock tương tự như Sharedlock.



* Kết quả thu được:

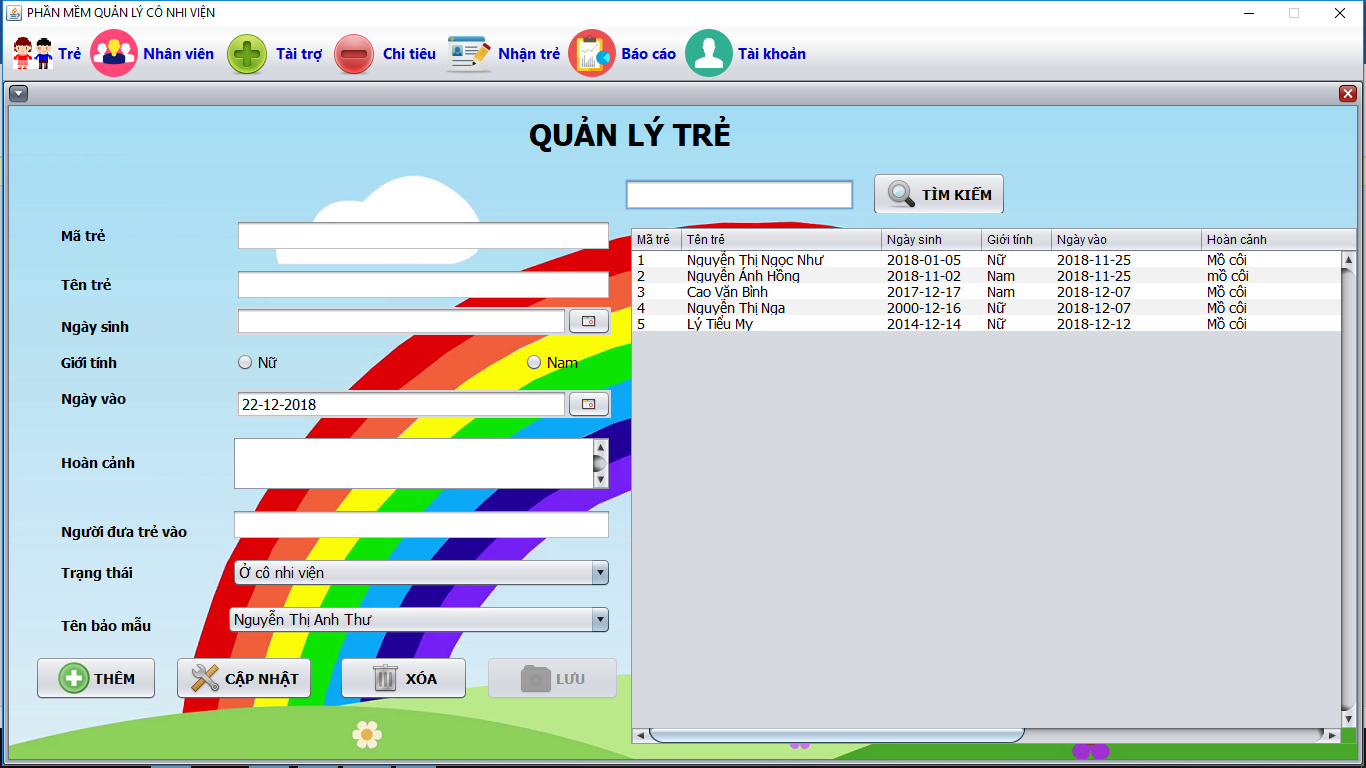


# CHƯƠNG 6 : THIẾT KẾ GIAO DIỆN

1. **Form chính**

* Đây là màn hình khi người dùng đăng nhập thành công vào phần mềm.
* Khi người dùng muốn quản lý thông tin trẻ thì nhấn vào mục “Trẻ”. Ở mục này người dùng có thể thêm, xóa, sửa trẻ.
* Tương tự khi muốn quản lý nhân viên thì nhấn vào mục “Nhân viên”. Trong quản lý nhân viên sẽ có loại nhân viên và thông tin nhân viên, người dùng có thể chọn thêm, xóa sửa loại nhân viên cũng như nhân viên.
* Quản lý tài trợ thì chọn mục “Tài trợ”. Trong đó sẽ có quản lý nhà tài trợ và chương trình tài trợ.
* Quản lý chi tiêu thì chọn mục “Chi tiêu”.
* Nếu muốn xem báo cáo thì chọn mục “Báo cáo”,..... Ở mỗi mục như vậy sẽ có các mục con người dùng chọn các mục con tương ứng rồi thực hiện.

1. **Form quản lý thông tin trẻ**

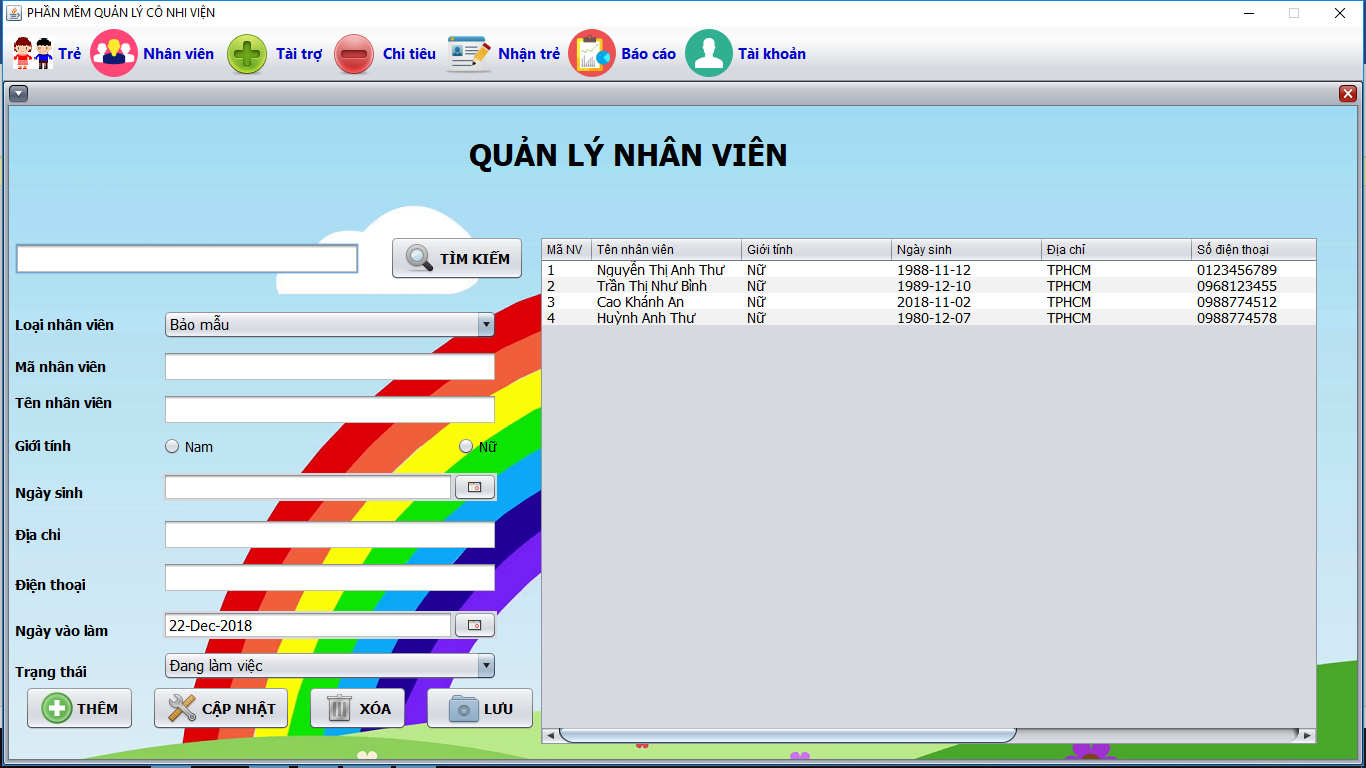


* Ở form này người dùng có thể chọn các chức năng như thêm mới trẻ, cập nhật thông tin trẻ, xóa trẻ, khi có sự thay đổi gì thì sẽ được thay đổi trên bảng Trẻ và luôn xuống cơ sở dử liệu.

1. **Form quản lý thông tin loại nhân viên**



* Tại form này người dùng có thêm, xóa, sửa loại nhân viên. Đồng thời việc thêm xóa sửa này sẽ được cập nhật xuống cở sở dữ liệu và hiện thị lên bảng bên cạnh. Người dùng muốn thêm loại nhân viên thì chỉ cần nhập tên loại nhân viên, mã loại nhân viên sẽ tự phát sinh.

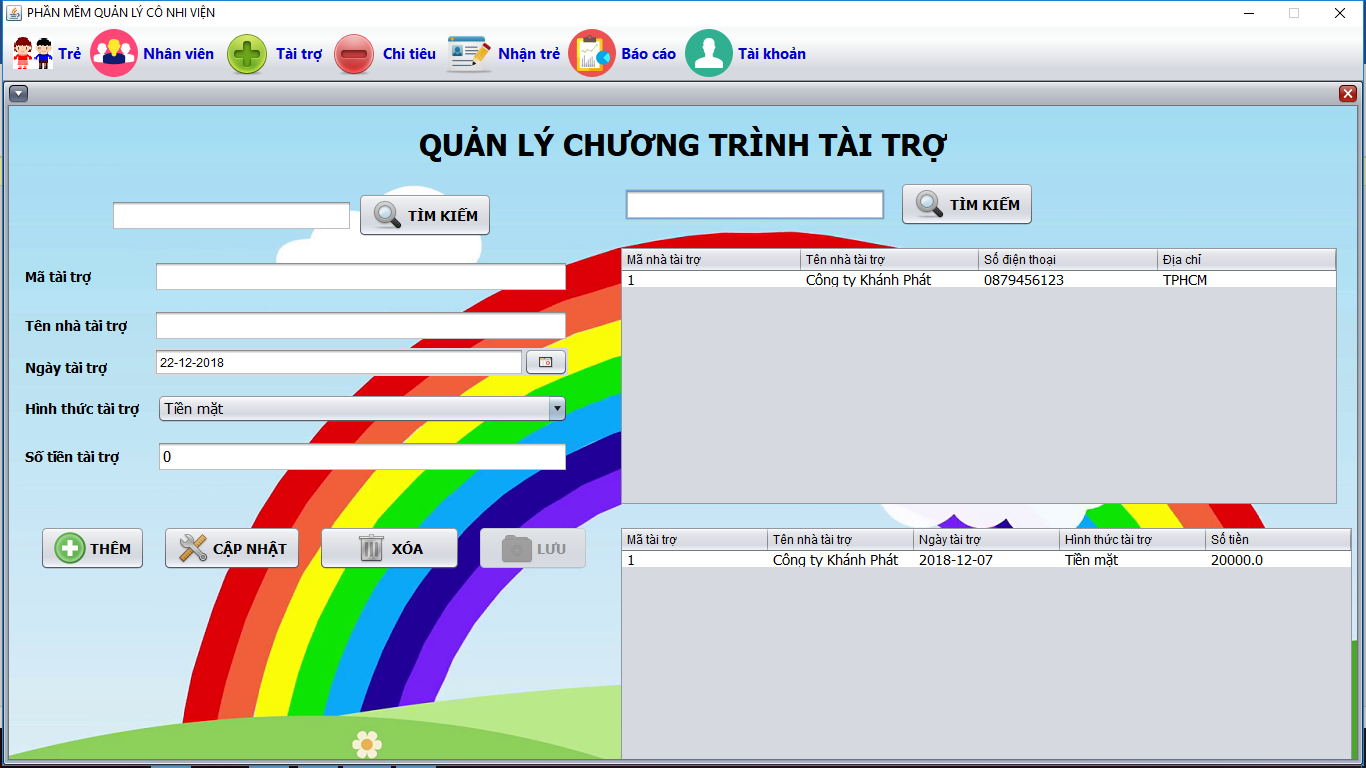
1. **Form quản lý thông tin nhân viên**

* Tương tư như các form trên, form này cho phép người dùng thêm, xóa, sửa thông tin nhân viên.

1. **Form quản lý thông tin nhà tài trợ**



* Đây là form dùng để quản lý các nhà tài trợ hay nói cách khác là lưu trữ thông tin của các nhà tài trợ để có thể tìm nguồn tài trợ cho cô nhi viện.

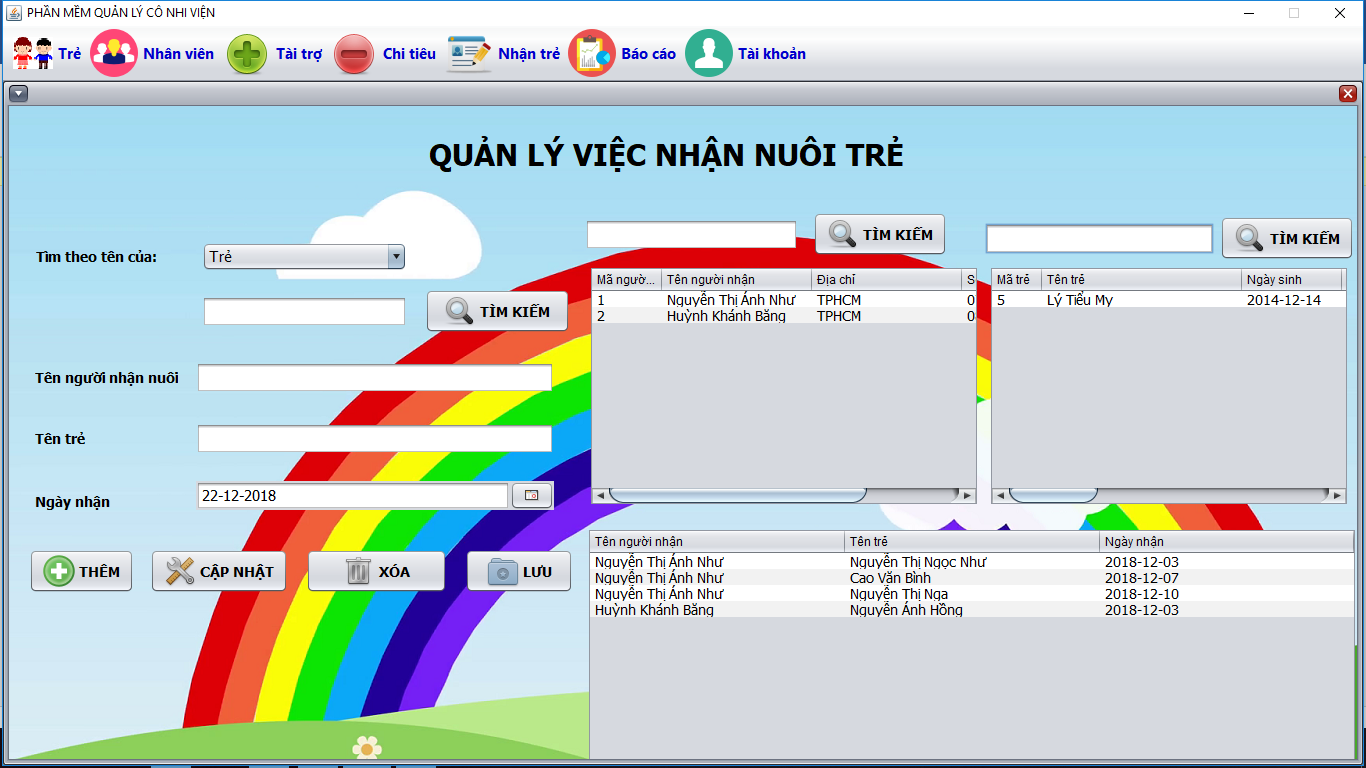
1. **Form quản lý chương trình tài trợ**

* Ở form này người dùng có thể thêm, xóa, sửa một chi tiết tài trợ. Khi muốn thêm chi tiết tài trợ, người dùng sẽ chọn từ bảng nhà tài trợ để lấy thông tin nhà tài trợ tương ứng sau đó tiến hành nhập các thông tin các còn lại như ngày tài trợ, hình thức tài trợ số tiền. Sau khi tiến hành thêm, xóa, sửa thành công dữ liệu sẽ được hiện thị lên bảng và lưu xuống cơ sở dữ liệu.

1. **Form quản lý người nhận trẻ**

* Người dùng có thể thêm, xóa, sửa người nhận trẻ. Dữ liệu được cập nhật sẽ được cập nhật lên bảng và đồng thời lưu xuống cơ sở dữ liệu.

1. **Form quản lý quá trình nhận trẻ**



* Người dùng sẽ chọn người nhận trẻ từ bảng người nhận trẻ và chọn trẻ được nhận từ bảng trẻ để hiện thị thông tin lên các text field. Sau đó, tiến hành điền các dữ liệu còn lại và thêm chi tiết nhận trẻ.
* Đối với cập nhật, và xóa người dùng sẽ chọn dữ liệu cần sửa hoặc xóa từ bảng chi tiết nhận trẻ sau đó tiến hành các thao tác tương ứng.

1. **Form quản lý chi tiêu**

* Form này cho phép người dùng thêm, xóa, sửa chi tiêu.

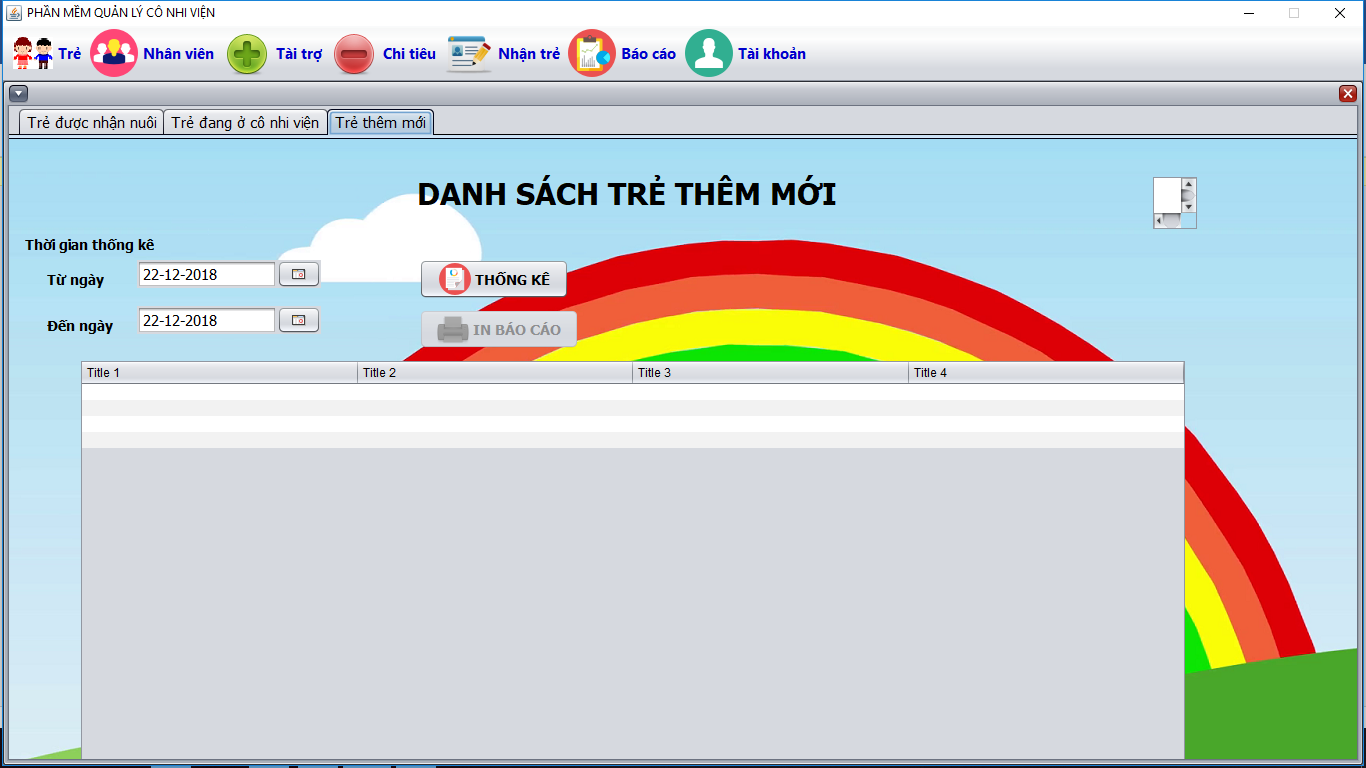


1. **Form quản lý chi tiết chi tiêu**

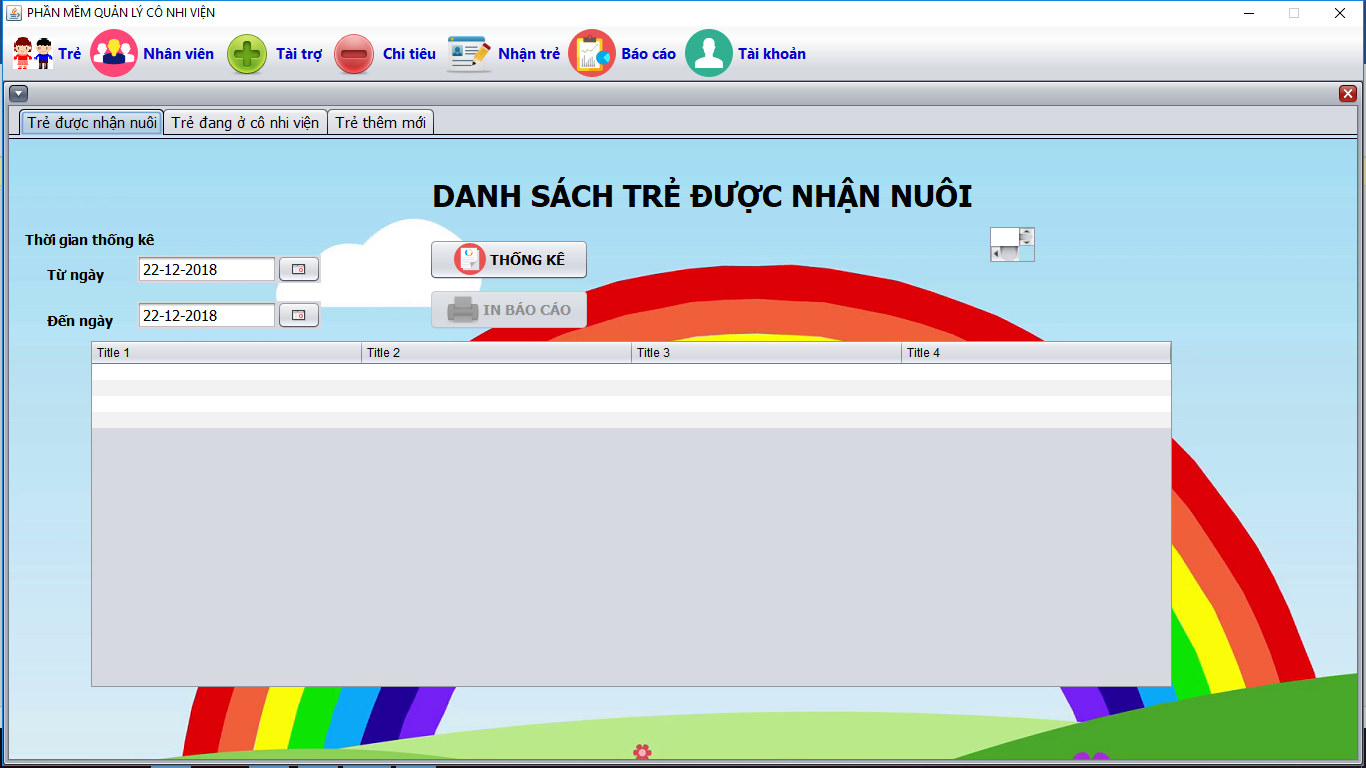
* Form cho phép người dùng thêm, xóa, sửa chi tiết chi tiêu.

1. **Form báo cáo thông tin trẻ**

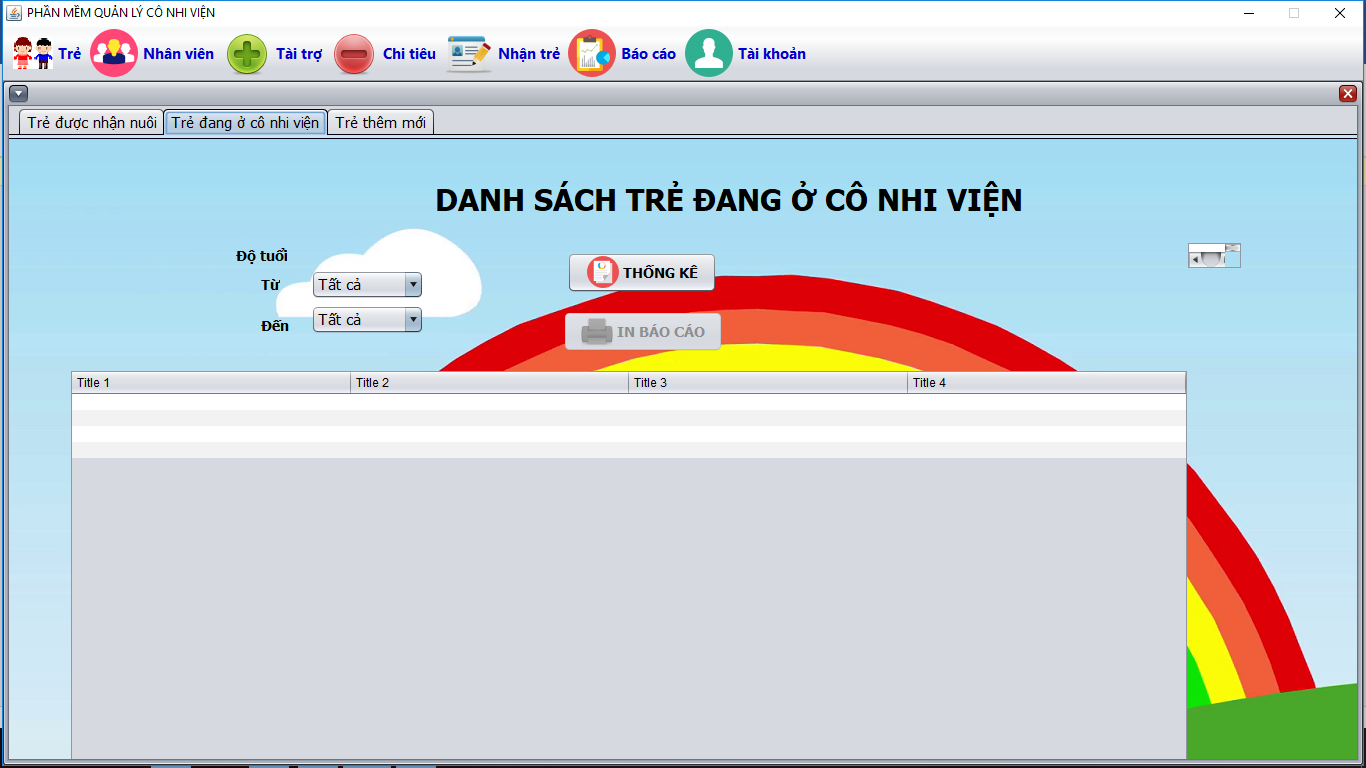
* Thống kê trẻ được thêm mới trong thời gian mà người dùng chọn sau đó có thể tiến hành in báo cáo.

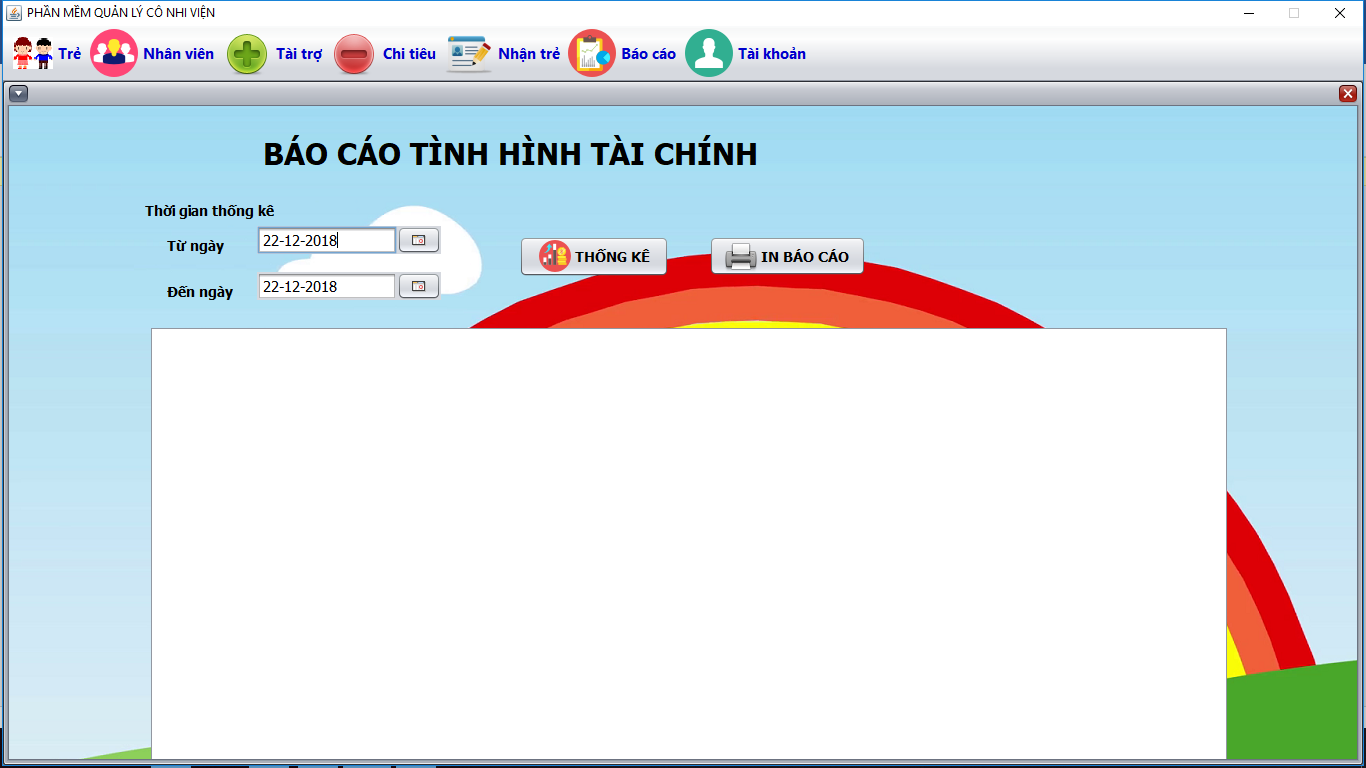


* Thống kê trẻ được nhận nuôi trong thời gian mà người dùng chọn sau đó có thể tiến hành in báo cáo.

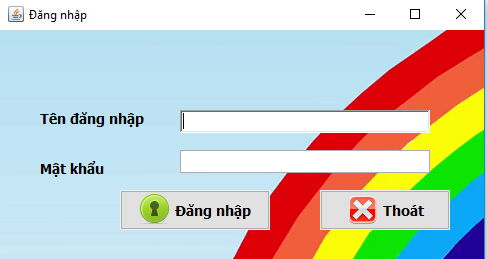


* Thống kê trẻ trong độ tuổi mà người dùng chọn đang ở cô nhi viện sau đó có thể tiến hành in báo cáo.



1. **Form báo cáo tài chính**

* Thống kê tình hình tài chính trong khoảng thời gian người dùng chọn, từ đó có thể tiến hành in báo cáo.

1. **Form login**

# CHƯƠNG 7: KẾT LUẬN

## 1.Kết quả đạt được

* Quản lý cô nhi viện sử dụng hệ thống thông tin không phải là một hướng đi mới. Tuy nhiên, hiện tại vẫn còn rất nhiều cô nhi viện chỉ quản lý bằng cách thủ công. Nhận thấy đây là một đề tài hay và thiết thực có thể ứng dụng rộng rãi. Chính vì vậy nhóm đã quyết định chọn đề tài nhằm tạo ra hệ thống quản lý cô nhi viện nhằm giúp ích cho việc quản lý đơn giản hơn, tiện ích hơn và giảm tải lượng công việc thủ công, …. Dưới đây là những chức năng cơ bản mà phần mềm quản lý cô nhi viện đạt được:
  + Quản lý thông tin trẻ.
  + Quản lý thông tin nhân viên.
  + Quản lý nhà tài trợ.
  + Quản lý tài trợ của các nhà tài trợ.
  + Quản lý chi tiêu.
  + Quản lý người nhận trẻ.
  + Báo cáo: báo cáo chi tiêu, báo cáo thông tin trẻ.
* Bên cạnh đó phần mềm phát triển thêm các xử lý đồng thời.

## 2.Hạn chế

* Vì thời gian thực hiện và nghiên cứu đề tài còn giới hạn, do đó các chức năng của phần mềm quản lý cô nhi viện còn 1 số hạn chế, cụ thể như sau:
  + Chưa kết nối được với mạng xã hội thông dụng (Facebook, zalo, ….)
  + Hiển thị dữ liệu bảng chưa hoàn thiện
  + Các kiểu định dạng dữ liệu chưa chuẩn
  + Chưa binding toàn bộ dữ liệu từ EntityManager
  + Mọi thao tác với database chỉ mới thao tác bằng store procedure (sử dụng transaction
  + Chưa triển khai trên nhiều laptop (database chỉ mới ở mức local)
* Do đó, đề tài chỉ dừng lại ở mức đồ án môn học và chưa thể hoàn thiện cũng như đặt được 1 số mục tiêu ban đầu đã đề ra.

**3.Hướng phát triển**

* Như đã trình bày ở các phần trước, nhóm nhận định còn rất nhiều việc cần phải làm để hoàn thiện hệ thống quản lý cô nhi viện. Chính vì vậy nhóm đặt ra mục tiêu để phát triển đề tài trong tương lai như sau:
  + Trang bị thêm nhiều ngôn ngữ, trước mắt là Tiếng Anh.
  + Quản lý chấm công và tính lương cho nhân viên
  + Import file (ví dụ: excel) thay vì nhập từng dòng dữ liệu.
  + Mở rộng quy mô phần mềm
  + Giao diện trực quan
* **Bảng phân công công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| Công việc | Bùi Thị Huyền Trân | Nguyễn Thị Kim Yến | Dương Thị Xuân Quỳnh |
| Phát biểu bài toán | x |  |  |
| Đặc tả bài toán |  | x |  |
| Khảo sát yêu cầu |  | x |  |
| Yêu cầu chức năng |  |  | x |
| Yêu cầu phi chức năng |  |  | x |
| Mô hình quan hệ | x |  |  |
| Bảng thuyết minh mô hình quan hệ | x |  |  |
| Mô tả các ràng buộc toàn vẹn | x |  |  |
| Xây dựng các store proccedure | x | x |  |
| Trigger | x | x |  |
| Các mức cô lập |  | x |  |
| Lost Update | x (2.1.1, 2.1.2) | x (2.1.3) |  |
| Dirty Read | x (2.2.1, 2.2.2) |  | x(2.2.4) |
| Non- repeatable read | x(2.3.1, 2.3.2, 2.3.3) |  | x(2.3.4) |
| Phantom | x(2.4.1, 2.4.2) | x (2.4.3) |  |
| Lock |  | x |  |
| Deadlock | x(4.1.2, 4.1.3) | x (4.1.1) |  |
| Thiết kế giao diện | x(2.1.1, 2.1.2, 2.1.3) |  |  |
| Xây dựng App | x | x |  |