TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIẾN THÁM**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**O**

**Q**

**HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

## ĐỀ TÀI:

**ỨNG DỤNG HỆ QUẢN TRỊ CSDL ĐỂ XÂY DỰNG HỆ THỐNG**

**ĐẶT VÉ XE KHÁCH**

Giảng viên hướng dẫn: **Th.S Phạm Trọng Huynh**

Sinh viên thực hiện: Lớp:

Khóa:

**Nhóm 16**

**CNPM 2**

**09**



#### TP. Hồ Chí Minh, tháng 04 năm 2023

1

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM

**KHOA: HỆ THỐNG THÔNG TIN VÀ VIẾN THÁM**



**BÁO CÁO ĐỒ ÁN MÔN HỌC HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU**

## ĐỀ TÀI:

**ỨNG DỤNG HỆ QUẢN TRỊ CSDL ĐỂ XÂY DỤNG HỆ THỐNG**

**ĐẶT VÉ XE KHÁCH**

Giảng viên hướng dẫn: **Th.S Phạm Trọng Huynh**

Sinh viên thực hiện:

Lớp: Khóa:

**ĐỖ PHƯỚC KHẢI**

**LÝ HUỲNH THANH HẬU**

**NGUYỄN TRUNG KIÊN**

**CNPM 2**

**09**

#### TP. Hồ Chí Minh, tháng 04 năm 2023

**Nhận xét của giảng viên**

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

………………………………………………………………………………………………

……….., ngày….tháng…. năm 2023

Giảng viên

**Phạm Trọng Huynh**

MỤC LỤC

[CHƯƠNG 1: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 8](#_TOC_250015)

* 1. [Lời giới thiệu 8](#_TOC_250014)
  2. [Phân tích hệ thống quản lý 8](#_TOC_250013)
     1. [Thực trạng hiện nay của hệ thống quản lý đặt vé xe khách 8](#_TOC_250012)
     2. [Biểu đồ phân cấp chức năng: 10](#_TOC_250011)
     3. [Biểu đồ luồng dữ liệu: 11](#_TOC_250010)
     4. [Phân loại các thuộc tính vào một tập thực thể 11](#_TOC_250009)

[CHƯƠNG 2: ỨNG DỤNG LÝ THUYẾT 14](#_TOC_250008)

* 1. [Xây dựng CSDL trên HQT CSDL 14](#_TOC_250007)

2.1.1 Câu lệnh tạo bảng, tạo khóa 14

2.1.2 Cấu trúc từng bảng 16

* 1. Thao tác trên CSDL trên HQT CSDL 20
  2. Store Procedure, Function, Trigger 23
     1. [Store Procedure 23](#_TOC_250006)
     2. [Function 23](#_TOC_250005)
     3. [Trigger 23](#_TOC_250004)
  3. Bảo mật, phân quyền 24

[CHƯƠNG 3: KẾT LUẬN 25](#_TOC_250003)

* 1. [Những kết quả đạt được của đồ án 35](#_TOC_250002)
  2. [Nhược điểm của đồ án 35](#_TOC_250001)

TÀI LIỆU THAM KHẢO 36

DANH MỤC HÌNH ẢNH

[Hình 1: biểu đồ phân cấp chức năng 10](#_TOC_250015)

[Hình 2: Biểu đồ phân luồng dữ liệu 11](#_TOC_250013)

[Hình 3: bảng khách hàng 11](#_TOC_250012)

[Hình 4: bảng ticket 12](#_TOC_250011)

[Hình 5: bảng XeBus 14](#_TOC_250010)

[Hình 6: bảng BookingBus 15](#_TOC_250009)

Hình 7: bảng Seat 15

[Hình 8-12: Các bảng nhập thông tin 15-18](#_TOC_250000)

[Hình 13: liên kết diagrams 19](#_TOC_250000)

[Hình 14: tạo login 20](#_TOC_250000)

[Hình 15: danh sách người dùng 21](#_TOC_250000)

[Hình 16: Xóa người dùng 22](#_TOC_250000)

[Hình 17: Danh sách người dùng dau khi xóa 23](#_TOC_250000)

[Hình 18: store procedure 28](#_TOC_250000)

Hình 19: Trigger31

# LỜI MỞ ĐẦU

### Đặt vấn đề

Hiện nay, hệ thống quản lý đặt vé xe khách đã có sự phát triển và cải tiến rất nhiều. Các công ty xe khách thường sử dụng phần mềm quản lý đặt vé để đáp ứng nhu cầu của khách hàng.

Tuy nhiên, trong một số trường hợp còn tồn tại một số khó khăn như:

* Hệ thống quản lý đặt vé vẫn còn thủ công nhiều, gây khó khăn cho nhân viên quản lý
* Các công ty xe khách chưa sử dụng được các công nghệ mới như trí tuệ nhân tạo, big data để cải tiến quy trình quản lý và tăng khả năng phục vụ khách hàng
* Một số công ty xe khách chưa đầu tư đầy đủ vào hệ thống quản lý đặt vé, dẫn đến khó khăn trong việc quản lý và giám sát sản phẩm

Vì vậy, việc cải thiện hệ thống quản lý đặt vé xe khách là rất cần thiết để đáp ứng nhu cầu ngày càng tăng của khách hàng.

### Mục tiêu nghiên cứu

Mục tiêu của bài báo cáo là ứng dụng hệ quản trị CSDL hệ thống đặt vé xe khách dựa trên SQL Server Management Studio có thể bao gồm:

Thiết kế và triển khai cơ sở dữ liệu cho hệ thống đặt vé xe khách, đảm bảo tính bảo mật, an toàn và bền vững của dữ liệu.

Tối ưu hoá truy vấn dữ liệu để đảm bảo hiệu suất của hệ thống đặt vé.

Xây dựng các chức năng quản lý đặt vé, bao gồm: quản lý thông tin khách hàng, quản lý vé, quản lý chuyến đi và các báo cáo định kỳ.

Phân tích dữ liệu về việc đặt vé để đưa ra các dự đoán về nhu cầu của khách hàng và phân bổ tài nguyên xe khách hiệu quả hơn.

Hỗ trợ cho việc phát triển ứng dụng di động để đặt vé và thanh toán cho khách hàng.

1. **Đối tượng sử dụng:**

Nhân viên quản lý đặt vé: họ sẽ sử dụng hệ thống để quản lý thông tin về khách hàng, vé xe, lịch trình và doanh thu.

Nhân viên bán vé: họ sẽ sử dụng hệ thống để bán vé xe cho khách hàng.

Khách hàng: họ sẽ sử dụng hệ thống để đặt vé xe, thanh toán và quản lý thông tin tài khoản của mình.

Nhà quản lý doanh nghiệp: họ sẽ sử dụng hệ thống để theo dõi hoạt động kinh doanh của doanh nghiệp, đưa ra các quyết định về đầu tư và phát triển.

Nhà phát triển ứng dụng: họ sẽ sử dụng hệ thống để xây dựng các ứng dụng liên quan đến việc đặt vé, thanh toán và quản lý thông tin khách hàng.

1. **Yêu cầu chức năng hệ thống:**
2. Đăng nhập và quản lý tài khoản người dùng.
3. Quản lý thông tin khách hàng và vé xe, bao gồm thông tin cá nhân, lịch sử đặt vé và thanh toán, thông tin về loại vé và giá cả, các lịch trình xe khách.
4. Hỗ trợ đặt vé và thanh toán trực tuyến.
5. Cập nhật thông tin và đồng bộ hóa dữ liệu giữa các hệ thống khác nhau.
6. Quản lý doanh thu và lợi nhuận, bao gồm báo cáo thống kê về số lượt vé bán được, doanh thu, chi phí và lợi nhuận của từng chuyến xe.
7. Bảo mật và bảo vệ dữ liệu cá nhân của khách hàng.

## 

## CHƯƠNG 1: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### Lời giới thiệu

[Hệ thống đặt vé xe khách dựa trên SQL Server là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) được sử dụng để quản lý thông tin về các chuyến xe khách và đặt vé cho khách hàng](https://sqladvice.com/tong-quan-ve-sql-server/" \t "_blank). Hệ thống này được thiết kế để giúp các công ty vận tải quản lý thông tin về các chuyến xe khách và đặt vé cho khách hàng một cách dễ dàng và hiệu quả.

[SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được sử dụng để lưu trữ và truy xuất dữ liệu trong các ứng dụng phần mềm](https://timoday.edu.vn/bai-1-tong-quan-ve-sql-server/). SQL Server cung cấp các tính năng như bảo mật, sao lưu và phục hồi dữ liệu, xử lý đồng thời và tối ưu hóa truy vấn để giúp các nhà phát triển phát triển các ứng dụng phần mềm chất lượng cao.

[Hệ thống đặt vé xe khách dựa trên SQL Server có thể được tích hợp với các ứng dụng web hoặc di động để cho phép khách hàng đặt vé trực tuyến](https://text.123docz.net/document/5040857-btl-qlda-he-thong-ban-ve-xe-khach.htm). Hệ thống này cũng có thể được tích hợp với các hệ thống thanh toán trực tuyến để cho phép khách hàng thanh toán vé của mình trực tuyến.

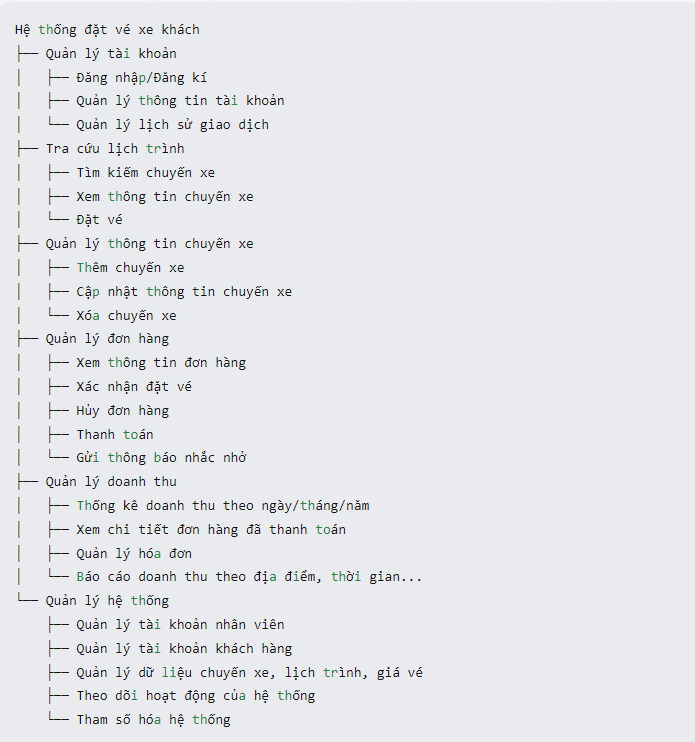
### Phân tích hệ thống quản lý

### Thực trạng hiện nay của hệ thống quản lý đặt vé xe khách:

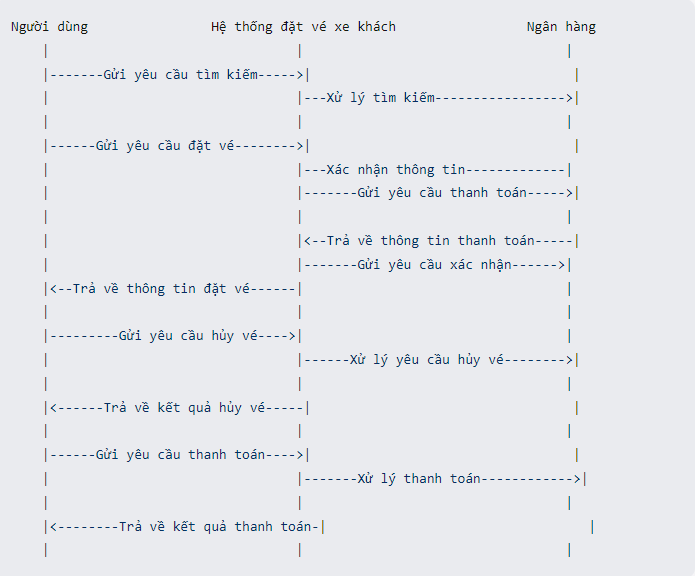
Quá tải: Hệ thống đặt vé thường xuyên xảy ra tình trạng quá tải khi có nhiều người truy cập cùng một lúc. Điều này khiến cho trang web chậm và tồn đọng nhiều thời gian chờ đợi để đặt vé.

Chất lượng dịch vụ kém: Một số hệ thống đặt vé vẫn chưa đảm bảo chất lượng dịch vụ tốt, khiến cho khách hàng cảm thấy không hài lòng. Ví dụ như thời gian xử lý đơn hàng chậm, không cập nhật thông tin đúng lúc...

Thiếu tính năng tiện lợi: Một số tính năng mà khách hàng muốn có như đặt chỗ theo tên, gửi thông báo nhắc nhở trước giờ lên xe... chưa được tích hợp vào hệ thống, dẫn đến mất điểm trước mắt khách hàng.

**1.2.2 Biểu đồ phân cấp chức năng**

**Hình 1: biểu đồ chức năng**

**1.2.3 biểu đồ luồng dữ liệu**

**Hình 2: biểu đồ luồng dữ liệu**

**1.2.4 Phân loại các thuộc tính vào một tập thực thể**

• Thông tin khách hàng: Tên, địa chỉ, số điện thoại, email, mật khẩu, token (mã xác nhận).

• Thông tin Xe bus: mã đặt vé, mã khách hàng, ngày đặt vé, ngày khởi hành, ngày đến, địa điểm khởi hành, địa điểm đến.

• Thông tin ghế: Mã xe bus, số ghế, ngày khởi hành

• Thông tin vé: mã vé, mã khách hàng, ngày đặt vé, ngày khởi hành, ngày đến, địa điểm khởi hành, địa điểm đến, số thứ tự khách hàng, tổng tiền.

1. **Bảng khách hàng**

KHACHHANG(MAKH, TENKH, DIACHI, SDT) – thể hiện các khách hàng của hệ thống đặt vé. Các khách hàng được phân biệt với nhau bằng mã khách hàng và tên khách hàng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Khoá chính | Khoá ngoại | Diễn giải |
| 1 | MAKH | Varchar(15) | X |  | Mã khách hàng |
| 2 | HO | Nvarchar(50) |  |  | Họ khách hàng |
| 3 | TEN | Nvarchar(50) |  |  | Tên khách hàng |
| 3 | Email | Nvarchar(50) |  |  | Địa chỉ khách hàng |
| 4 | SDT | Varchar(10) |  |  | Số điện thoại khách hàng |

1. **Bảng BookingBus**

BookingBus(MaDatVe,MAKH,NgayDatVe,NgayKhoiHanh,NgayDen, DDKhoiHanh, NgayDen, DDKhoiHanh, DDDen, STTKH) – thể hiện chuyến xe bus. Các chuyến xe được phân biệt với nhau bằng mã đặt vé và mã khách hàng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Khoá chính | Khoá ngoại | Diễn giải |
| 1 | MaDatVe | int | X |  | Mã đặt vé |
| 2 | MaKH | int |  |  | Mã khách hàng |
| 3 | NgayDatVe | datetime |  |  | Ngày đặt vé |
| 4 | NgayKhoiHanh | datetime |  |  | Ngày khởi hành |
| 5 | NgayDen | datetime |  |  | Ngày đến |
| 6 | DDKhoiHanh | Varchar(100) |  |  | Địa điểm khởi hành |
| 7 | DDDen | Varchar(100) |  |  | Địa điểm đến |
| 8 | STTKH | int |  |  | Số thứ tự khách hàng |
| 9 | TongTien | Decimal(18,2) |  |  | Tổng tiền |

1. **Bảng Seat**

Seat(MaXeBus, SoGhe, NgayKhoiHanh,) – thể hiện số ghế của hệ thống. Các ghế phân biệt với nhau bằng mã xe bus và.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Khoá chính | Khoá ngoại | Diễn giải |
| 1 | MaXeBus | int | X |  | Mã xe bus |
| 2 | SoGhe | int | X |  | Số ghế |
| 3 | NgayKhoiHanh | datetime | X |  | Ngày khởi hành |

1. **Bảng Ticket**

Ticket(MaVe, MaKH, NgayDatVe, NgayKhoiHanh, NgayDen, DDKhoiHanh, DDDen, STTKH, TongTien) – thể hiện vé của hệ thống. Các vé được phân biệt với nhau bằng mã vé và mã khách hàng.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Khoá chính | Khoá ngoại | Diễn giải |
| 1 | MaVe | int | X |  | Mã đặt vé |
| 2 | MaKH | int |  | X | Mã khách hàng |
| 3 | NgayDatVe | datetime |  |  | Ngày đặt vé |
| 4 | NgayKhoiHanh | datetime |  |  | Ngày khởi hành |
| 5 | NgayDen | datetime |  |  | Ngày đến |
| 6 | DDKhoiHanh | Varchar(100) |  |  | Địa điểm khởi hành |
| 7 | DDDen | Varchar(100) |  |  | Địa điểm đến |
| 8 | STTKH | int |  |  | Số thứ tự khách hàng |
| 9 | TongTien | decimal(18, 2) |  |  | Tổng tiền |

1. **Bảng XeBus**

XeBus(MaXeBus, MauXe, SoDangKy, SoGhe, TienNghi) – thể hiện loại xe bus của hệ thống. Các loại xe được phân biệt với nhau bằng mã xe bus và mẫu xe.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Tên thuộc tính | Kiểu dữ liệu | Khoá chính | Khoá ngoại | Diễn giải |
| 1 | MaXeBus | int | X |  | Mã nhân viên |
| 2 | MauXe | varchar(50) |  |  | Tên nhân viên |
| 3 | SoDangKy | varchar(50) |  |  | Giới tính |
| 4 | SoGhe | int |  |  | Ngày sinh |
| 5 | TienNghi | Varchar(200) |  |  | Địa chỉ |

## CHƯƠNG 2: ỨNG DỤNG LÝ THUYẾT

### Xây dựng CSDL trên HQT CSDL

### Câu lệnh tạo bảng, tạo khóa

### Tạo bảng, tạo khóa:

### Bảng khách hàng

CREATE TABLE KhachHang (

MaKH INT PRIMARY KEY,

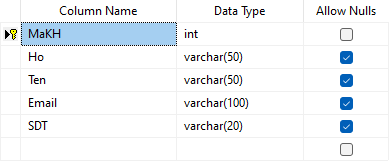
Ho VARCHAR(50),

Ten VARCHAR(50),

Email VARCHAR(100),

SDT VARCHAR(20)

);

****

**Hình 3: bảng khách hàng**

Bảng ticket

CREATE TABLE Ticket (

MaVe INT PRIMARY KEY,

MaKH INT FOREIGN KEY REFERENCES KhachHang(MaKH),

NgayDatVe DATETIME,

NgayKhoiHanh DATETIME,

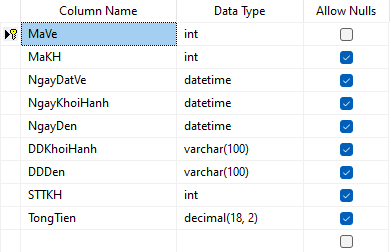
NgayDen DATETIME,

DDKhoiHanh VARCHAR(100),

DDDen VARCHAR(100),

STTKH INT,

TongTien DECIMAL(18,2)

); 

**Hình 4: bảng ticket**

Bảng XeBus

CREATE TABLE XeBus (

MaXeBus INT PRIMARY KEY,

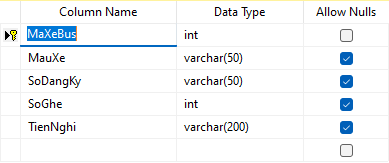
MauXe VARCHAR(50),

SoDangKy VARCHAR(50),

SoGhe INT,

TienNghi VARCHAR(200)

);



**Hình 5: bảng XeBus**

Bảng BookingBus

CREATE TABLE BookingBus (

MaDatVe INT PRIMARY KEY not null,

MaXeBus INT FOREIGN KEY REFERENCES XeBus(MaXeBus) not null,

MaKH INT FOREIGN KEY REFERENCES KhachHang(MaKH),

NgayDatVe DATETIME,

NgayKhoiHanh DATETIME,

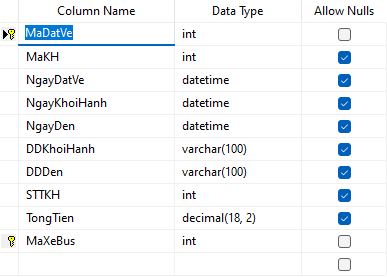
NgayDen DATETIME,

DDKhoiHanh VARCHAR(100),

DDDen VARCHAR(100),

STTKH INT,

TongTien DECIMAL(18,2) );



**Hình 6: BookingBus**

Bảng Seat

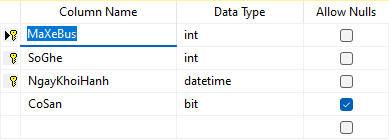
CREATE TABLE Seat (

MaXeBus INT FOREIGN KEY REFERENCES XeBus(MaXeBus) Not null,

SoGhe INT not null,

NgayKhoiHanh DATETIME not null,

CoSan bit );



**Hình 7: bảng Seat**

**2.2.2 Cấu trúc từng bảng:**

* **Bảng khách hàng:**

INSERT INTO KhachHang (MaKH, Ho, Ten, Email, SDT)

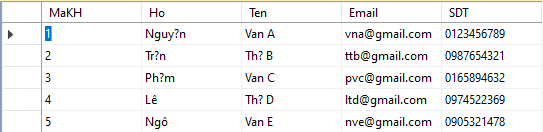
VALUES (1, 'Nguyễn', 'Văn A', 'vna@gmail.com', '0123456789'),

(2, 'Trần', 'Thị B', 'ttb@gmail.com', '0987654321'),

(3, 'Phạm', 'Văn C', 'pvc@gmail.com', '0165894632'),

(4, 'Lê', 'Thị D', 'ltd@gmail.com', '0974522369'),

(5, 'Ngô', 'Văn E', 'nve@gmail.com', '0905321478');



**Hình 8: bảng nhập thông tin khách hàng**

* **Bảng Ticket:**

INSERT INTO Ticket (MaVe, MaKH, NgayDatVe, NgayKhoiHanh, NgayDen, DDKhoiHanh, DDDen, STTKH, TongTien)

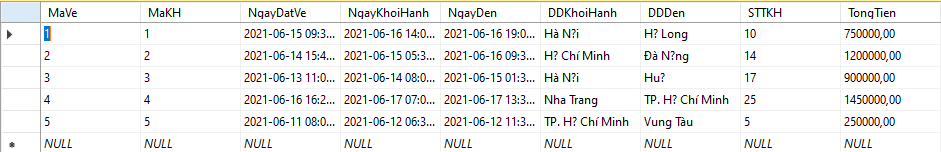
VALUES (1, 1, '2021-06-15 09:30:00', '2021-06-16 14:00:00', '2021-06-16 19:00:00', 'Hà Nội', 'Hạ Long', 10, 750000),

(2, 2, '2021-06-14 15:45:00', '2021-06-15 05:30:00', '2021-06-16 09:30:00', 'Hồ Chí Minh', 'Đà Nẵng', 14, 1200000),

(3, 3, '2021-06-13 11:00:00', '2021-06-14 08:00:00', '2021-06-15 01:30:00', 'Hà Nội', 'Huế', 17, 900000),

(4, 4, '2021-06-16 16:20:00', '2021-06-17 07:00:00', '2021-06-17 13:30:00', 'Nha Trang', 'TP. Hồ Chí Minh', 25, 1450000),

(5, 5, '2021-06-11 08:00:00', '2021-06-12 06:30:00', '2021-06-12 11:30:00', 'TP. Hồ Chí Minh', 'Vũng Tàu', 5, 250000);



**Hình 9: bảng nhập thông tin vé**

* + **Bảng XeBus:**

INSERT INTO XeBus (MaXeBus, MauXe, SoDangKy, SoGhe, TienNghi)

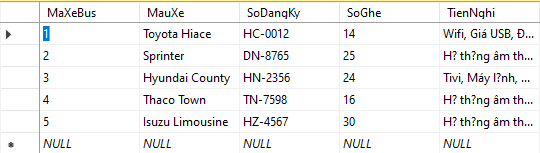
VALUES (1, 'Toyota Hiace', 'HC-0012', 14, 'Wifi, Giá USB, Đèn đọc sách'),

(2, 'Sprinter', 'DN-8765', 25, 'Hệ thống âm thanh 3D, Máy lạnh'),

(3, 'Hyundai County', 'HN-2356', 24, 'Tivi, Máy lạnh, Thú rối'),

(4, 'Thaco Town', 'TN-7598', 16, 'Hệ thống âm thanh, Máy lạnh, Két đựng tiền'),

(5, 'Isuzu Limousine', 'HZ-4567', 30, 'Hệ thống âm thanh, Máy lạnh, Tủ bảo quản thức ăn');



**Hình 10: bảng nhập thông tin loại xe**

* + **Bảng BookingBus:**

INSERT INTO BookingBus (MaDatVe, MaXeBus, MaKH, NgayDatVe, NgayKhoiHanh, NgayDen, DDKhoiHanh, DDDen, STTKH, TongTien)

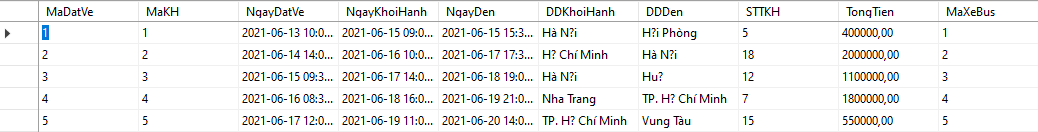
VALUES (1, 1, 1, '2021-06-13 10:00:00', '2021-06-15 09:00:00', '2021-06-15 15:30:00', 'Hà Nội', 'Hải Phòng', 5, 400000),

(2, 2, 2, '2021-06-14 14:00:00', '2021-06-16 10:00:00', '2021-06-17 17:30:00', 'Hồ Chí Minh', 'Hà Nội', 18, 2000000),

(3, 3, 3, '2021-06-15 09:30:00', '2021-06-17 14:00:00', '2021-06-18 19:00:00', 'Hà Nội', 'Huế', 12, 1100000),

(4, 4, 4, '2021-06-16 08:30:00', '2021-06-18 16:00:00', '2021-06-19 21:00:00', 'Nha Trang', 'TP. Hồ Chí Minh', 7, 1800000),

(5, 5, 5, '2021-06-17 12:00:00', '2021-06-19 11:00:00', '2021-06-20 14:00:00', 'TP. Hồ Chí Minh', 'Vũng Tàu', 15, 550000);



**Hình 11: bảng nhập thông tin chuyến xe**

* + **Bảng Seat:**

INSERT INTO Seat (MaXeBus, SoGhe, NgayKhoiHanh, CoSan)

VALUES (1, 1, '2021-06-15 09:00:00', 0),

(1, 2, '2021-06-15 09:00:00', 1),

(1, 3, '2021-06-15 09:00:00', 1),

(1, 4, '2021-06-15 09:00:00', 0),

(1, 5, '2021-06-15 09:00:00', 1),

(2, 6, '2021-06-16 10:00:00', 1),

(2, 7, '2021-06-16 10:00:00', 1),

(2, 8, '2021-06-16 10:00:00', 0),

(2, 9, '2021-06-16 10:00:00', 1),

(2, 10, '2021-06-16 10:00:00', 1),

(3, 11, '2021-06-17 14:00:00', 0),

(3, 12, '2021-06-17 14:00:00', 1),

(3, 13, '2021-06-17 14:00:00', 1),

(3, 14, '2021-06-17 14:00:00', 0),

(3, 15, '2021-06-17 14:00:00', 1),

(4, 16, '2021-06-18 16:00:00', 1),

(4, 17, '2021-06-18 16:00:00', 0),

(4, 18, '2021-06-18 16:00:00', 1),

(4, 19, '2021-06-18 16:00:00', 1),

(4, 20, '2021-06-18 16:00:00', 0),

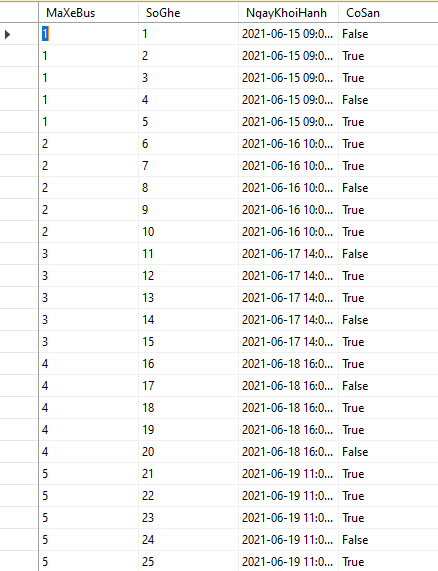
(5, 21, '2021-06-19 11:00:00', 1),

(5, 22, '2021-06-19 11:00:00', 1),

(5, 23, '2021-06-19 11:00:00', 1),

(5, 24, '2021-06-19 11:00:00', 0),

(5, 25, '2021-06-19 11:00:00', 1);



**Hình 12: bảng nhập thông tin ghế ngồi**

### Diagrams liên kết các bảng

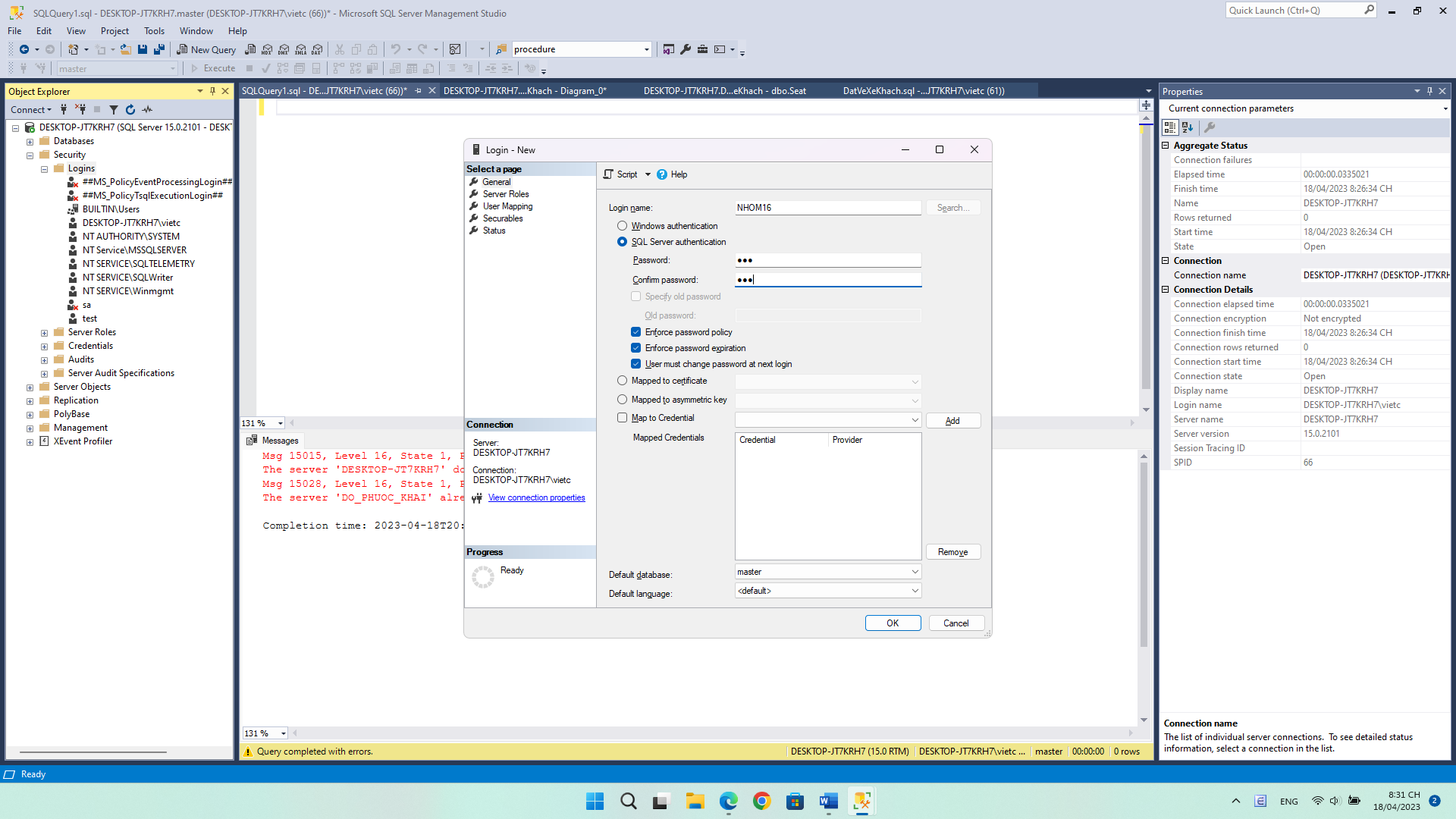
### 

### Hình 13: liên kết các Diagrams

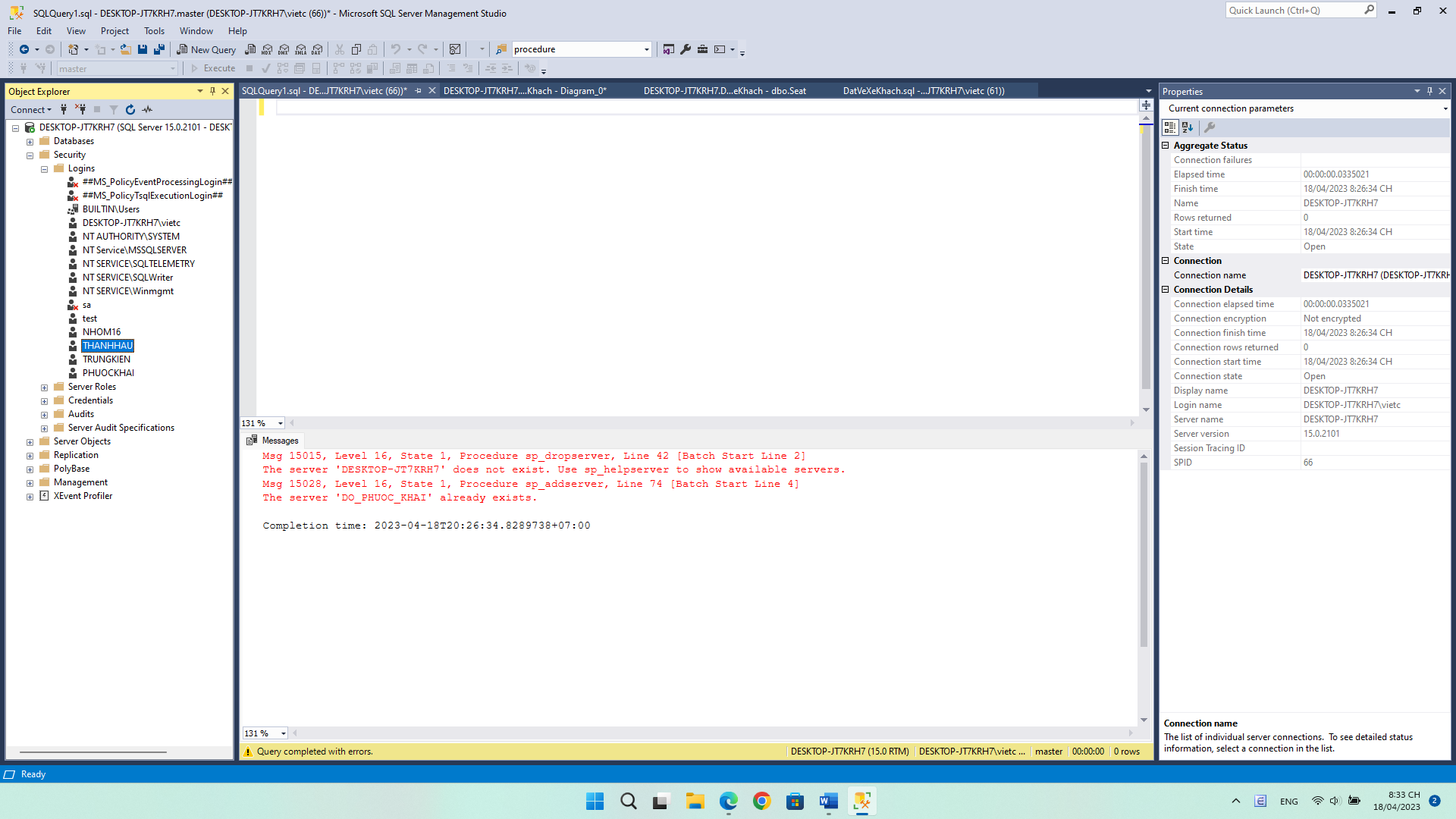
### Thao tác trên CSDL trên HQT CSDL

**Tạo và xoá tài khoản người dùng đăng nhập quyền sql server bằng công cụ**

* + Tạo người sử dụng (user)

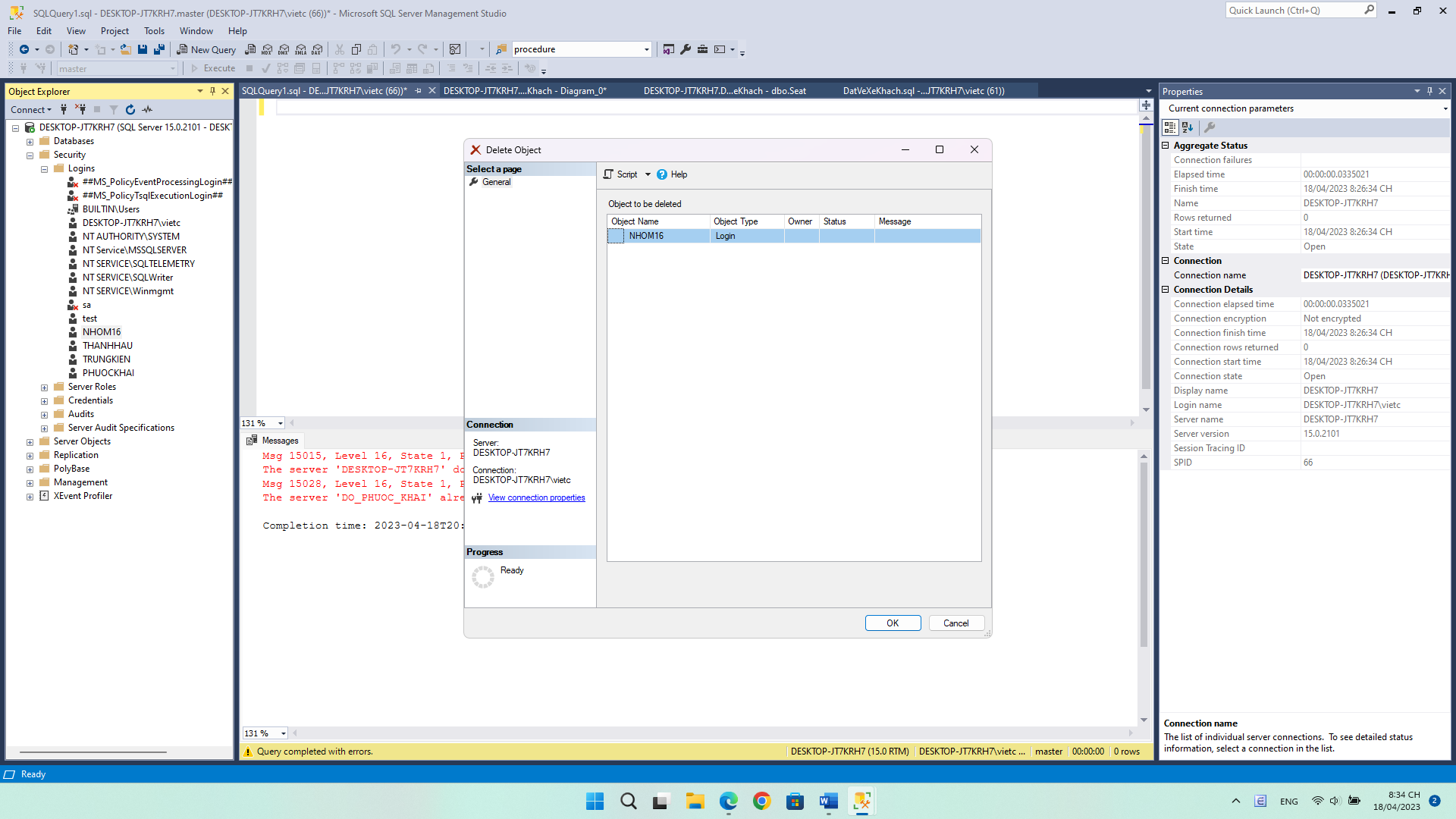


**Hình 14 : Tạo tài khoản login**

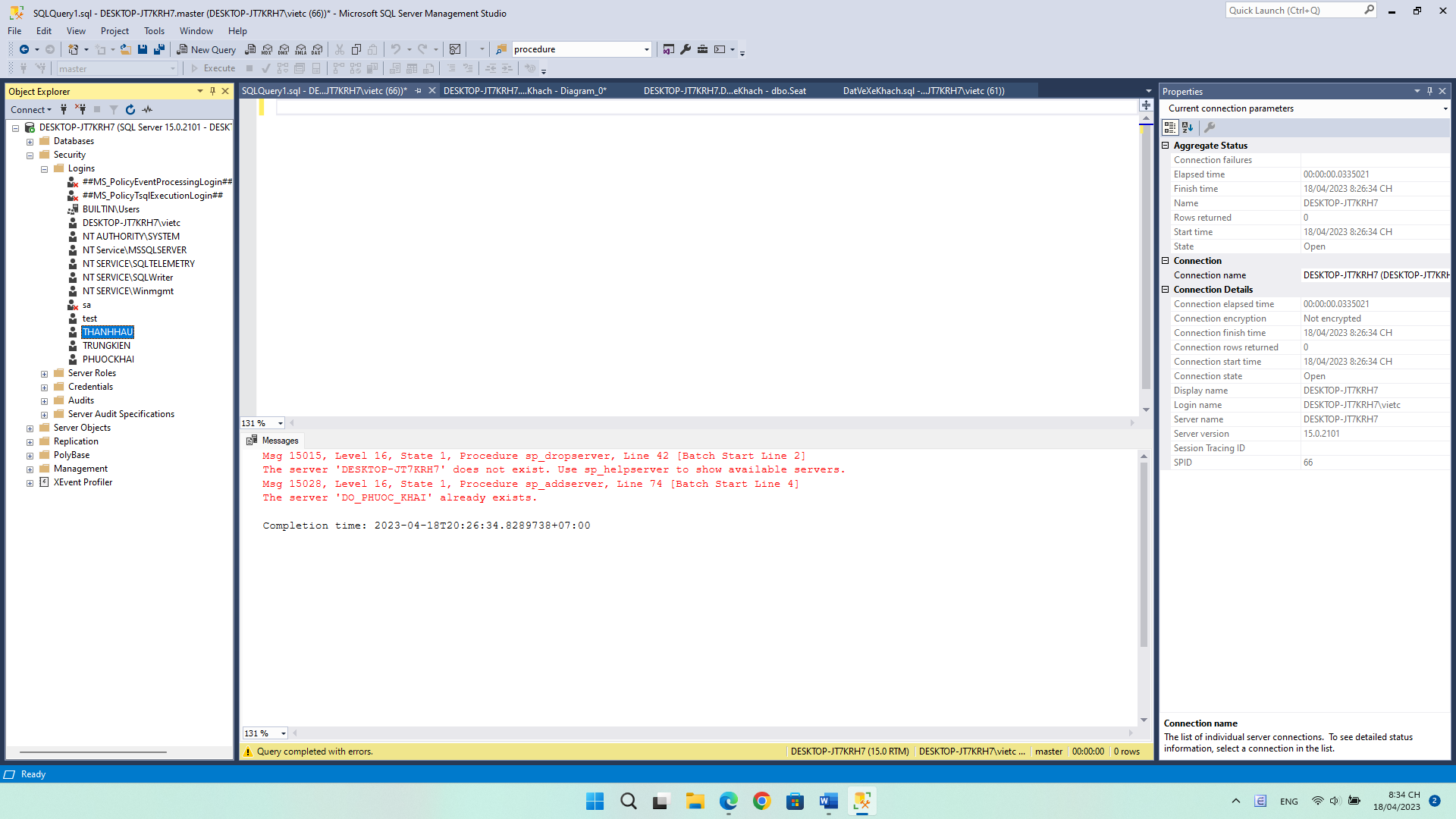


**Hình 15 : danh sách tài khoản user**

* + Xoá người dùng



**Hình 16: Xoá người dùng**



**Hình 17: Danh sách người dùng sau khi xoá**

Tạo và xoá tài khoản người dùng đăng nhập quyền sql server bằng câu lệnh

* Tạo user :

CREATE USER NHOM13 FOR LOGIN PHUOCKHAI

* Xoá user :

DROP LOGIN PHUOCKHAI

* 1. **Store Procedure, Function, Trigger**

### 2.3.1 Store Procedure

* **Stored procedure là gì?**

Là một tập hợp các câu lệnh SQL dùng để thực thi một nhiệm vụ nhất định. Nó có thể xem như một hàm trong các ngôn ngữ lập trình.

Stored procedure là một khái niệm khá phổ biến và được hầu hết các hệ quản trị cơ sở dữ liệu (DBMS) hỗ trợ, tuy nhiên không phải tất cả đều hỗ trợ Stored Procedure.

* **Ưu điểm của Stored Procedure**
* Thân thiện với nhà phát triển ( khả năng tái sử dụng)
* Cải thiện hiệu suất: giảm bớt gánh nặng không cần thiết cho các máy chủ làm việc
* Tốc độ: chỉ thực hiện được một lần trong quá trình tạo và được lưu trữ ở bộ nhớ cache trong máy chủ. Do đó, không cần biên dịch lặp lại trong quá trình thực hiện thủ tục, làm tăng độ thực thi. Tuy nhiên, nếu có bất kỳ cập nhật nào được thực hiện trong quy trình thì nó sẽ được biên dịch lặp lại.
* Bảo mật: vì không có quyền truy cập trực tiếp vào các bảng nên nó ngăn chặn việc thao túng trái phép các bảng ghi bên trong các bảng. một thủ tục được viết đúng cách hoạt động như một biện pháp ngăn chặn vững chắc đối với SQL injections.
* Xử lý dựa trên tập hợp: thường đề cập đến việc xử lý dữ liệu hàng loạt cùng một lúc và nhanh hơn nhiều. đây là một lợi thế khác của việc sử dụng thủ tục được lưu trữ vì nó xử lý phần lớn dữ liệu. các ứng dụng phân tán trên toàn mạng, phụ thuộc rất nhiều vào các thủ tục này vì nó làm giảm các chuyến đi khứ hồi.
* **Nhược điểm của Stored Procedure**
* Gỡ lỗi: gỡ lỗi không bao giờ dễ dàng, vì vậy không nên viết và thực thi logic nghiệp vụ phức tạp bằng cách sử dụng các thủ tục lưu trữ.
* Phụ thuộc
* Đắt tiền
* Nhà cung cấp cụ thể: Store Procedure thường được viết trên một nền tảng có thể không được chạy trên nền tảng khác. Các thủ tục được viết bằng Oracle phức tạp hơn và bạn có thể phải viết lại toàn bộ thủ tục cho SQL Server.
* Hạn chế ngôn ngữ lập trình: lập trình SP không phong phú như các nền tảng phát triển khác như java hay php.
* My SQL không có tool hỗ trợ Debug Store
* Để phát triển ứng dụng thì đòi hỏi bạn phải có một kỹ năng nhất định mới có thể maintenance được.
* **Các kiểu Store Procedure**

Chủ yếu có 2 loại:

1. System stored procedures : Thủ tục mà những người sử dụng chỉ có quyền thực hiện, không được phép thay đổi.Các tác vụ quản trị bao gồm : liệt kê, thêm, cập nhập, xóa.

Các ưu điểm của System stored procedures là:

* Tiện lợi: System stored procedures được cài đặt sẵn trong hệ thống quản lý cơ sở dữ liệu và có thể được sử dụng mà không cần phải tạo các stored procedure riêng.
* Tính ổn định: System stored procedures được thiết kế và kiểm tra kỹ lưỡng bởi các nhà phát triển của hệ thống quản lý cơ sở dữ liệu. Vì vậy, chúng rất ổn định và luôn hoạt động đúng mức.
* Tính bảo mật: System stored procedures có thể được sử dụng để quản lý các hoạt động bảo mật trong hệ thống quản lý cơ sở dữ liệu, như phân quyền và kiểm tra tính toàn vẹn dữ liệu. Vì vậy, chúng giúp tăng cường bảo mật cho cơ sở dữ liệu.
* Tính tường minh: System stored procedures có thể được sử dụng để thực hiện các tác vụ phức tạp trong cơ sở dữ liệu. Chúng cung cấp cho người dùng một cách thực hiện tường minh và dễ hiểu các tác vụ phức tạp như tối ưu hóa truy vấn và quản lý việc sao lưu và phục hồi dữ liệu.
* Tính nhân rộng: System stored procedures có thể được sử dụng để thực hiện các tác vụ trên các bảng lớn và cơ sở dữ liệu phức tạp. Chúng có tính nhân rộng tốt và có thể được sử dụng trong các hệ thống có quy mô lớn mà không gặp vấn đề về hiệu suất.

System stored procedures có thể có một số nhược điểm, bao gồm:

* Rủi ro bảo mật: Vì các thủ tục được lưu trữ trong hệ thống có đặc quyền cao và thường được chạy trong ngữ cảnh bảo mật của tài khoản dịch vụ SQL Server, chúng có thể tiềm ẩn nguy cơ bảo mật nếu không được bảo mật đúng cách.
* Điều quan trọng là đảm bảo rằng chỉ những người dùng được ủy quyền mới có quyền truy cập để thực hiện các thủ tục được lưu trữ trong hệ thống.
* Độ phức tạp: Các thủ tục được lưu trữ trong hệ thống thường phức tạp và có thể khó hiểu và khó khắc phục sự cố. Chúng cũng có thể có nhiều tham số cần được cấu hình chính xác để chúng hoạt động bình thường.
* Các vấn đề về khả năng tương thích: Một số quy trình được lưu trữ trong hệ thống có thể không tương thích với các phiên bản SQL Server cũ hơn hoặc với các phiên bản SQL Server khác, điều này có thể gây ra sự cố nếu bạn đang nâng cấp hoặc di chuyển sang một máy chủ mới.
* Hiệu suất: Tùy thuộc vào độ phức tạp và khối lượng công việc của hệ thống của bạn, một số quy trình được lưu trữ trong hệ thống có thể ảnh hưởng đến hiệu suất và làm chậm SQL Server. Điều quan trọng là phải giám sát các số liệu hiệu suất và mức sử dụng để đảm bảo rằng các quy trình được lưu trữ trong hệ thống không ảnh hưởng đến hiệu suất cơ sở dữ liệu tổng thể.
* Bảo trì: Như với bất kỳ mã nào, các thủ tục được lưu trữ trong hệ thống yêu cầu bảo trì và cập nhật liên tục để đảm bảo rằng chúng tiếp tục hoạt động bình thường và vẫn tương thích với các thay đổi đối với môi trường SQL Server. Điều này có thể đòi hỏi nỗ lực và nguồn lực đáng kể.

Kí hiệu : sp\_..., xp\_...

* Để thực thi 1 SP, ta sử dụng lệnh EXEC. VD: EXEC sp\_databases (liệt kê danh sách các database có trong SQL server)
* Được chia thành các nhóm sau: o SP dùng để liệt kê thông tin (liệt kê danh sách database ,...)
* SP dùng để trình bày thông tin (trình bày thông tin table, ...)
* SP dùng để thêm, xóa, cập nhập thông tin (đổi mật khẩu, ....)

1. User - Defined Stored Procedures: Thủ tục do người dùng tạo và thực hiện.

* Là các SP do người dùng tạo.
* Thủ tục khi chạy sẽ theo 3 bước :
* Resolution: xử lý truy vấn, đọc và kiểm tra lỗi chính tả.
* Optimization : tối ưu hóa truy vấn.
* Compilation
* Các ưu điểm của User-Defined Stored Procedures (UDSPs) bao gồm:
* Tính sử dụng lại: Một khi bạn đã định nghĩa một UDSP, nó có thể được tái sử dụng trong nhiều ứng dụng khác nhau mà không cần phải viết lại mã.
* Bảo mật: Với UDSP, bạn có thể cung cấp quyền truy cập riêng tư vào cơ sở dữ liệu của mình. UDSP chỉ được phép truy cập vào các bảng hoặc các quy trình không phải là công khai, giúp bảo vệ dữ liệu của bạn.
* Hiệu suất cao hơn: Khi sử dụng UDSP, cơ sở dữ liệu chỉ phải tạo và biên dịch các câu lệnh một lần. Sau đó, chúng có thể được lưu trữ và thực thi nhiều lần mà không cần phải tạo lại chúng nữa, giúp tăng hiệu suất và giảm tài nguyên hệ thống.
* Dễ bảo trì: Khi có sự thay đổi về yêu cầu hoặc cấu trúc cơ sở dữ liệu, chỉ cần chỉnh sửa UDSP và tất cả các ứng dụng sử dụng nó sẽ tự động được cập nhật.
* Quản lý dữ liệu dễ dàng hơn: Bạn có thể sử dụng UDSP để quản lý các thao tác dữ liệu phức tạp như lập kế hoạch và lựa chọn dữ liệu, cho phép các quản trị viên cơ sở dữ liệu dễ dàng kiểm soát các thao tác dữ liệu.
* Tính tái sử dụng: Một khi bạn đã định nghĩa một UDSP, nó có thể được sử dụng lại trong các ứng dụng khác mà không cần viết lại mã. Điều này giúp tiết kiệm thời gian và chi phí phát triển.
* Một số nhược điểm của User-Defined Stored Procedures bao gồm:
* Tốn thời gian và công sức: Viết các stored procedure cho một ứng dụng sẽ tốn nhiều thời gian và công sức so với việc viết code trực tiếp trong ứng dụng.
* Khó quản lý: Khi có nhiều stored procedure trong ứng dụng, việc quản lý chúng có thể trở nên khó khăn, đặc biệt là khi trình quản lý cơ sở dữ liệu không hỗ trợ công cụ tìm kiếm nhanh chóng và hiệu quả.
* Phiên bản lưu trữ: Nếu bạn muốn thay đổi stored procedure, bạn phải cập nhật phiên bản lưu trữ. Nếu ứng dụng đang chạy, việc thay đổi này có thể gây ra sự cố và làm gián đoạn hoạt động của ứng dụng.
* Sự cố khó khăn: Khi xảy ra sự cố trong trình quản lý cơ sở dữ liệu, việc xác định lỗi trong stored procedure có thể khó khăn và tốn nhiều thời gian.
* Quá phụ thuộc: Sử dụng quá nhiều stored procedure có thể dẫn đến việc ứng dụng của bạn phụ thuộc quá nhiều vào trình quản lý cơ sở dữ liệu, và điều này có thể làm cho ứng dụng của bạn trở nên chậm chạp và không ổn định.

**Những lợi ích khi sử dụng Stored procedure**

* Một stored procedure cung cấp một lớp bảo mật quan trọng giữa user và database. Những người dùng cuối không thể thực thi được một số tác vụ quan trọng trong database, bằng cách cung cấp các stored procedure, người dùng cuối có thể thoải moái sử dụng để thực thi các tác vụ này mà không cần quan tâm bên dưới đang làm gì.
* Stored Procedure được biên dịch và lưu vào bộ nhớ khi khởi tạo. Điều đó có nghĩa rằng nó sẽ thực thi nhanh hơn so với việc gửi từng đoạn lệnh SQL tới SQL Server. Vì nếu bạn gửi từng đoạn lệnh nhiều lần thì SQL Server cũng sẽ phải biên dịch lại nhiều lần, rất mất thời gian so với việc biên dịch sẵn.
* Chúng ta cũng có thể sử dụng lại các stored procedure trong ứng dụng, khi mà chúng ta chỉ cần viết một lần, và các mã lệnh có thể gọi đến nó bất kỳ lúc nào.
* **Cú pháp:**

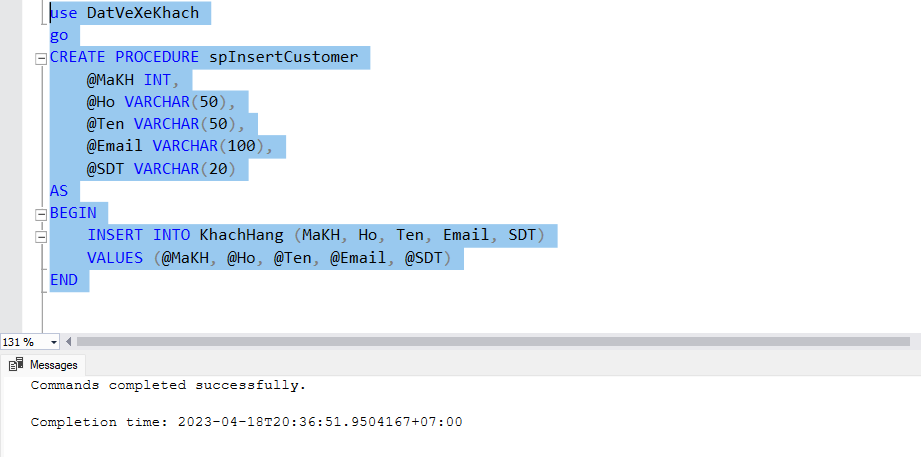
Create procedure <procedure\_Name>

As

Begin

<SQL Statement>

END;

****

**Hình 18: Store Procedure**

* + 1. **Function là gì?**

Function (Hàm) là một đối tượng trong cơ sở dữ liệu bao gồm một tập nhiều câu lệnh được nhóm lại với nhau và được tạo ra với mục đích sử dụng lại. Trong SQL Server, hàm được lưu trữ và bạn có thể truyền các tham số vào cũng như trả về các giá trị.

**➢ Cú pháp**

CREATE FUNCTION function\_name (parameter\_list)

RETURN data\_type

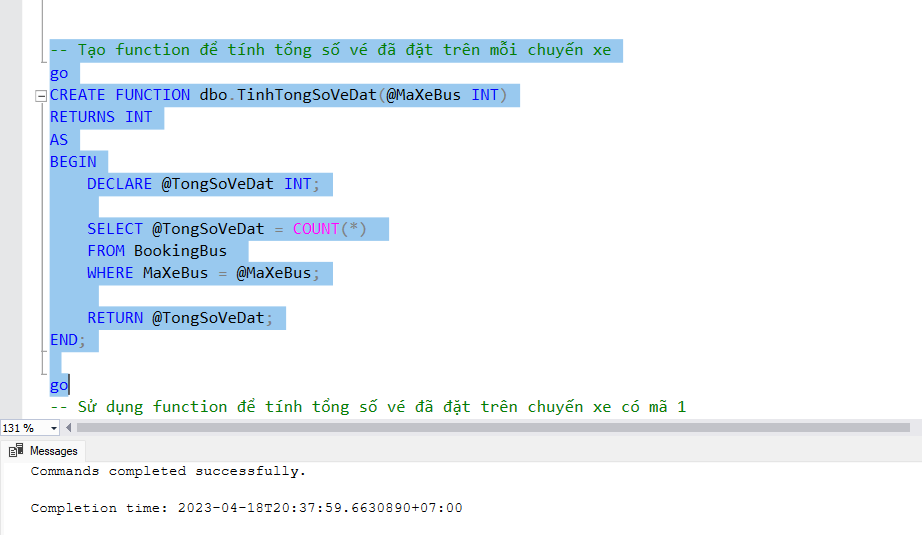
AS

BEGIN

statements

RETURN value

END



**Hình 33: Function**

* **Các loại Function**
* Scalar-valued: Trả về giá trị vô hướng của các kiểu dữ liệu T-SQL.
* Table-valued: Trả về bảng, là kết quả của một hoặc nhiều lệnh.
* **Ưu điểm:**
* Tái sử dụng mã: Function trong SQL Server cho phép tái sử dụng mã giữa các truy vấn và được sử dụng nhiều lần trong cùng một truy vấn hoặc trong các truy vấn khác.
* Giảm thiểu lỗi: Tích hợp function trong SQL Server giúp giảm thiểu lỗi trong các câu lệnh truy vấn bằng cách loại bỏ phần mã trùng lặp.
* Tiết kiệm thời gian: Function trong SQL Server được thiết kế để thực hiện một số nhiệm vụ định sẵn và giúp tiết kiệm thời gian trong thiết kế câu lệnh truy vấn.
* Quản lý dữ liệu hiệu quả: Function có thể được sử dụng để thao tác với dữ liệu và mang lại hiệu quả thông qua việc thiết lập và sử dụng các hàm tùy chỉnh.
* **Nhược điểm:**
* Hiệu suất kém: Sử dụng function trong SQL Server có thể làm giảm hiệu suất truy vấn nếu chúng ta gọi các function phức tạp hoặc được sử dụng nhiều trong một truy vấn.
* Không biết trước map execution: SQL Server không biết trước cách function sẽ được thực thi - trước hoặc sau khi truy vấn gốc được thực thi.
* Cần thời gian để tạo và thử nghiệm: Cấu trúc phức tạp của function trong SQL Server có thể yêu cầu thời gian để tạo và thử nghiệm các hàm tùy chỉnh để đảm bảo rằng chúng hoạt động như mong muốn.
* Không thể thay đổi thông tin: Khi function trong SQL Server được sử dụng trong các câu lệnh truy vấn, chúng là cố định và không thể thay đổi bằng cách sử dụng các giá trị khác.
  + 1. **Trigger**
* Khái niệm: là một mã thủ tục được thực thi tự động để phản hồi lại các sự kiện nhất định trên một bảng đã chỉ định. Các ứng dụng có thể bao gồm: lưu lại thay đổi hoặc cập nhật dữ liệu các bảng khác. Trigger được lưu trữ và quản lý trong Server DB, được dùng trong trường hợp ta muốn kiểm tra các ràng buộc toàn vẹn trong DB.
* Ưu điểm:
* Dễ dàng viết code
* Cho phép người dùng có thể tạo audit cơ bản thông qua việc sử dụng bảng dữ liệu đã xóa trong Trigger.
* Hỗ trợ người dùng gọi các store procedure, hàm lưu trữ trong một Trigger.
* Được sử dụng để thực thi tính toàn vẹn tham chiếu dữ liệu.
* Sử dụng Trigger để thay thế các công việc hẹn giờ theo lịch.
* Nhược điểm:
* Trigger đóng vai trò là một phần của quá trình kiểm tra tính hợp lệ dữ liệu. Nó không thay thế cho toàn bộ hoạt động này.
* Chỉ hoạt động ngầm trong cơ sở dữ liệu và không hiển thị ở giao diện nên khó chỉ ra các vấn đề xảy ra ở tầng cơ sở dữ liệu.

* Hoạt động của Trigger là cập nhật lên bảng dữ liệu nên làm tăng khối lượng công việc cho cơ sở dữ liệu. Vì thế, hệ thống sẽ vận hành chậm hơn.
* **Cú pháp**

CREATE TRIGGER [Tên Trigger]

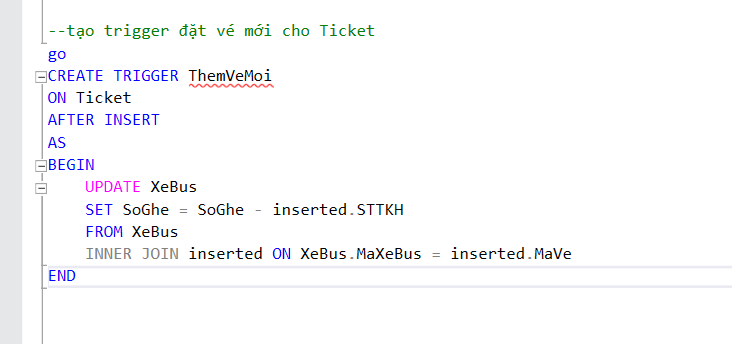
ON [Tên Bảng]

[FOR/AFTER/INSTEAD OF] [INSERT/UPDATE/DELETE]

AS

BEGIN

[Thân trigger]

****END

**Hình 19: Trigger**

* **Các loại Trigger**
* Trigger cho DML (ngôn ngữ thao tác dữ liệu): bao gồm 3 kiểu là INSERT, UPDATE và DELETE. Ưu điểm của Trigger cho DML gồm có:
* Bảo vệ chống lại các hoạt động INSERT, UPDATE và DELETE độc hại hoặc không chính xác và thực thi các hạn chế khác phức tạp hơn các hạn chế được xác định bằng các ràng buộc CHECK.
* Có thể đánh giá trạng thái của bảng trước và sau khi sửa đổi dữ liệu và thực hiện hành động dựa trên sự khác biệt đó.
* Nhiều trigger DML cùng loại (CHÈN, CẬP NHẬT hoặc XÓA) trên một bảng cho phép nhiều hành động khác nhau diễn ra để phản hồi lại cùng một câu lệnh sửa đổi.
* Nếu các ràng buộc tồn tại trên bảng kích hoạt, chúng sẽ được kiểm tra sau khi thực thi kích hoạt INSTEAD OF nhưng trước khi thực thi kích hoạt AFTER. Nếu các ràng

buộc bị vi phạm, các hành động kích hoạt INSTEAD OF sẽ được khôi phục và kích hoạt AFTER không được thực thi.

Tuy nhiên, Trigger cũng có một số nhược điểm như sau:

* Hiệu suất thấp: Nếu có nhiều trigger cho một bảng hoặc view, thì hiệu suất của cơ sở dữ liệu có thể giảm do việc thực hiện nhiều trigger.
* Khó quản lý: Khi cơ sở dữ liệu có nhiều trigger, việc quản lý và bảo trì chúng trở nên khó khăn và tốn thời gian.
* Lỗi dữ liệu: Trigger có thể gây ra các lỗi không mong muốn nếu không được xây dựng hoặc cấu hình đúng cách, hoặc nếu dữ liệu đầu vào không đúng với các ràng buộc được thiết lập bởi trigger.
* Trigger cho DDL (ngôn ngữ định nghĩa dữ liệu): sử dụng cho các lệnh CREATE, ALTER và DROP.
* Ưu điểm:
* Nhanh chóng phát triển khai thác: Kích hoạt cho DDL là dễ dàng phát triển nhanh chóng và sử dụng. Người quản trị cơ sở dữ liệu có thể định nghĩa nhiều Trình kích hoạt cho DDL để đáp ứng nhu cầu quản lý dữ liệu.
* Tự động hóa công việc: Kích hoạt cho DDL giúp tự động hóa công việc quản trị cơ sở dữ liệu. Ví dụ: khi một bảng được tạo, Trigger có thể tự động tạo các bảng phụ hoặc các quy tắc kiểm tra.
* Giảm thiểu sai sót: Kích hoạt cho DDL giúp giảm thiểu sai sót trong quá trình quản lý cơ sở dữ liệu. Trình kích hoạt có thể kiểm tra tính hợp lệ của dữ liệu và đảm bảo rằng các ràng buộc được áp dụng đúng cách.
* Tăng tính linh hoạt: Kích hoạt cho DDL cung cấp tính linh hoạt và cho phép người quản trị cơ sở dữ liệu thực hiện nhiều nhiệm vụ phức tạp hơn.
* Nhược điểm:
* Khó khăn trong quá trình gỡ lỗi: Một số Trình kích hoạt cho DDL có thể gây ra các vấn đề chưa được xác định hoặc khác với kết quả mong muốn khi được triển khai. Điều này làm cho việc gỡ lỗi trở nên khó khăn hơn.
* Ảnh hưởng đến hiệu suất: Kích hoạt cho DDL có thể ảnh hưởng đến hiệu suất của cơ sở dữ liệu khi được khai thác. Kích hoạt có thể cần phải thực hiện các tính toán phức tạp hoặc kiểm tra dữ liệu, có thể làm chậm hệ thống.
* Trigger cho LOGON: sử dụng để kiểm soát các lệnh khi đăng nhập vào máy chủ.
* Ưu điểm:
* Tăng tính an toàn của hệ thống: Kích hoạt cho LOGON có thể được sử dụng để giám sát và kiểm soát quyền truy cập vào hệ thống. Nó có thể cung cấp các thông tin về người dùng để quản lý thành viên có thể kiểm tra xem các tài khoản có được sử dụng đúng cách hay không.
* Giảm sự mất mát danh tính: Trình kích hoạt cho ĐĂNG NHẬP có thể được sử dụng để ngăn chặn các hành động không hợp lệ có thể xảy ra đối với tài khoản người dùng. Nó có thể giúp giảm thiểu tình trạng mất mát danh tính và các trường hợp lừa đảo.
* Được sử dụng để xử lý các lỗi trong quá trình đăng nhập: Kích hoạt cho LOGON có thể được sử dụng trong trường hợp xảy ra lỗi đăng nhập để giải quyết các lỗi này xử lý một cách tự động và đảm bảo rằng người dùng không bị đoạn gián đoạn trong quá trình sử dụng hệ thống.
* Nhược điểm:
* Tốn nhiều tài nguyên: Kích hoạt cho LOGON có thể chiếm nhiều tài nguyên của hệ thống khi được kích hoạt. Nếu không được kiểm tra cẩn thận, nó có thể gây ra sự chậm chạp và giảm hiệu suất của hệ thống.
* Có thể bị tấn công: Trigger for LOGON có thể trở thành một điểm yếu trong hệ thống và có thể bị tấn công bởi tin tặc hoặc tin tặc. Nếu không được cấu hình chặt chẽ, nó có thể làm cho hệ thống trở nên dễ bị tấn công và khó khăn trong việc bảo vệ dữ liệu.
* Đòi hỏi kỹ năng cao để thiết lập: Kích hoạt để LOGON yêu cầu người quản trị có kỹ năng cao để cấu hình và thiết lập cho phù hợp với yêu cầu của hệ thống. Cấu hình sai có thể gây ra các vấn đề bảo mật và dễ dàng bị tấn công.
  1. **bảo mật và phân quyền:**
* Bảo mật dữ liệu: Oracle Database cung cấp các tính năng bảo mật như mã hóa dữ liệu, kiểm soát truy cập dữ liệu và kiểm soát quyền truy cập dữ liệu.
* Bảo mật hệ thống: Oracle Database cung cấp các tính năng bảo mật như kiểm soát truy cập hệ thống, kiểm soát quyền truy cập hệ thống và giám sát hệ thống. Ngoài ra, Oracle Database còn cung cấp các giải pháp phân quyền như:
* Quản lý người dùng: Oracle Database cho phép bạn tạo và quản lý người dùng với các quyền truy cập khác nhau vào cơ sở dữ liệu của bạn.
* Quản lý vai trò: Oracle Database cho phép bạn tạo và quản lý vai trò với các quyền truy cập khác nhau vào cơ sở dữ liệu của bạn.

**CHƯƠNG 3: KẾT LUẬN**

### Những kết quả đạt được của đồ án

+ Chúng em trau dồi thêm kiến thức về sql

+ Tự học và thử trên một số phương diện mới nhưu winform và C#

+ Nhớ các câu lệnh cũng như những cú pháp cơ bản trong sql

+ Thêm năng lực tự học hỏi

### Nhược điểm của đồ án

+Vì thời gian nhóm phân chia chưa hợp lý nên dữ liệu còn ít

+Bảng cũng không nhiều chưa thể thành một dự án lớn

+Bài còn sơ sài và thiếu nhiều điểm

+ Những câu lệnh còn đơn giản

# TÀI LIỆU THAM KHẢO