**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO BÀI TẬP LỚN**

**ĐỀ TÀI:**

**Nghiên cứu và XÂY DỰNG CƠ SỞ DỮ LIỆU**

**QUẢN LÝ XE BUÝT TRONG 1 THÀNH PHỐ**

Giảng viên hướng dẫn: Phạm Thị Miên

Nhóm Sinh viên thực hiện:

1. Lê Huỳnh Minh 5951071059
2. Khuất Lê Thành Luân 5951071054\
3. Đỗ Văn Sáng 5951071091

Lớp: CQ.59.CNTT Khoá: 59

TP. Hồ Chí Minh, năm 2020

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO BÀI TẬP LỚN**

**ĐỀ TÀI:**

**Nghiên cứu và XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ XE BUÝT TRONG 1 THÀNH PHỐ**

Giảng viên hướng dẫn: Phạm Thị Miên

Nhóm Sinh viên thực hiện:

1. Lê Huỳnh Minh 5951071059
2. Khuất Lê Thành Luân 5951071054\
3. Đỗ Văn Sáng 5951071091

Lớp: CQ.59.CNTT Khoá: 59

TP. Hồ Chí Minh, năm 2020

**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH Độc lập – Tự do – Hạnh phúc**

# **NHIỆM VỤ**

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Khóa:** 59 **Lớp:** CQ.59.CNTT

**Tên thành viên** **Mã sinh viên**

1. Lê Huỳnh Minh 5951071059
2. Khuất Lê Thành Luân 5951071054
3. Đỗ Văn Sáng 5951071091
4. **Tên đề tài**

NGHIÊN CỨU VÀ XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ XE BUÝT TRONG 1 THÀNH PHỐ

1. **Mục đích, yêu cầu**
   1. **Mục đích:**

Nhằm phục vụ cho nhu cầu tra cứu các tuyến xe buýt lưu thông trong thành phố.

* 1. **Yêu cầu:**
* **Yêu cầu công nghệ**
  + Sử dụng công cụ Microsoft SQL Server Management.
* **Yêu cầu chức năng**
  + Cho phép tra cứu thông tin xe buýt.

1. **Nội dung và phạm vi đề tài**
   1. **Nội dung:**
      * Mô tả bài toán.
      * Tổng quan về công nghệ đang sử dụng.
      * Phân tích và thiết kế hệ thống.
      * Thử nghiệm hệ thống cơ sở dữ liệu.
      * Kết quả thu được
   2. **Phạm vi:**
      * Nghiên cứu công cụ Microsoft SQL Server Management.
2. **Công nghệ, công cụ và ngôn ngữ lập trình**
   * + Công cụ Microsoft SQL Server Management.
3. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng**
   * + Xây dựng thành công hệ thống cơ sở dữ liệu quản lý xe buýt.
4. **Giảng viên và cán bộ hướng dẫn**

Họ tên: Trần Thị Dung

Đơn vị công tác: Phân hiệu Trường Đại học Giao thông Vận tải.

**Ngày ...... tháng ...... năm 2018 Đã giao nhiệm vụ TKTN**

**Giảng viên hướng dẫn**

**Phạm Thị Miên**

Đã nhận nhiệm vụ TKTN Ký tên:

Lê Huỳnh Minh

Khuất Lê Thành Luân

Đỗ Văn Sáng

# **NHẬN XÉT CỦA GIẢNG VIÊN**

*Tp. Hồ Chí Minh, ngày … tháng … năm …*

Giảng viên hướng dẫn

MỤC LỤC

[**NHIỆM VỤ** i](#_Toc58267422)

[**NHẬN XÉT CỦA GIẢNG VIÊN** iii](#_Toc58267423)

[**MỞ ĐẦU** 1](#_Toc58267424)

[**1.** **Lý do chọn đề tài.** 1](#_Toc58267425)

[**2.** **Hướng tiếp cận của đề tài.** 1](#_Toc58267426)

[**3.** **Mục tiêu nghiên cứu.** 1](#_Toc58267427)

[**4.** **Đối tượng và phạm vi nghiên cứu.** 1](#_Toc58267428)

[**5.** **Phương pháp nghiên cứu.** 1](#_Toc58267429)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT** 2](#_Toc58267430)

[**1.1** **Khái niệm cơ sở dữ liệu** 2](#_Toc58267431)

[**1.2** **Tổng quan về SQL** 2](#_Toc58267432)

[**CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ** 3](#_Toc58267433)

[**2.1** **Mô tả bài toán** 3](#_Toc58267434)

[**2.2** **Chuẩn bị** 3](#_Toc58267435)

[**2.2.1** **Sơ đồ ER** 3](#_Toc58267436)

[4](#_Toc58267437)

[**2.2.2** **Mô hình quan hệ** 5](#_Toc58267438)

[**2.2.3** **Những khó khăn gặp phải** 5](#_Toc58267439)

[**CHƯƠNG 3: THỬ NGHIỆM ỨNG DỤNG** 6](#_Toc58267440)

[**3.1.** **Môi trường cài đặt** 6](#_Toc58267441)

[**3.2.** **Kết quả khởi chạy và đánh giá** 9](#_Toc58267442)

[Kết quả khởi chạy 9](#_Toc58267443)

[Đánh giá 11](#_Toc58267444)

[**CHƯƠNG 4: RÀNG BUỘC TOÀN VẸN (RBTV)** 11](#_Toc58267445)

[**CHƯƠNG 5: DẠNG CHUẨN VÀ CHUẨN HÓA** 12](#_Toc58267446)

[**CHƯƠNG 6: THỦ TỤC STORED PROCEDURE VÀ TRIGGER** 17](#_Toc58267447)

[**1.** **Thủ tục Stored Procedure** 17](#_Toc58267448)

[**2.** **Trigger** 19](#_Toc58267449)

[**CHƯƠNG 7: PHÂN QUYỀN TRONG CSDL** 21](#_Toc58267450)

[**1.** **Tổng quan về phân quyền và người dùng trong CSDL** 21](#_Toc58267451)

[**2.** **Phân quyền và người dùng trong CSDL ‘Quản lý xe BUS trong 1 thành phố’** 21](#_Toc58267452)

# **MỞ ĐẦU**

1. **Lý do chọn đề tài.**

Ùn tắc giao thông ở Việt Nam không còn xa lạ gì đối với chúng ta. Ai cũng có xe riêng, mật độ dân số thì đông, đường xá thì không thể chu cấp đủ không gian cho từng người. Phương tiện công cộng là 1 giải pháp được nhà nước khuyến khích sử dụng để tránh ùn tắc giao thông và còn bảo vệ môi trường. Không phương tiện giao thông nào có thể sánh được với xe buýt về phần chi phí lưu thông rẻ nhất, ít nhất là đối với hành khách. Và xe buýt cũng là phương tiện giao thông hàng đầu của sinh viên, ít nhất là đối với sinh viên nghèo và không có xe riêng hoặc không biết lái xe, khi mà xe buýt lúc nào cũng tràn ngập sinh viên và khi sinh viên vắng thì xe buýt cũng vắng. Do nhu cầu đó, 1 hệ CSDL quản lý xe buýt phải được ra đời để lưu trữ các thông tin cần thiết nhằm giúp quản lý xe buýt tốt hơn.

1. **Hướng tiếp cận của đề tài.**

* Nghiên cứu và hiểu được cách thức hoạt động của Microsoft SQL Server Management.
* Thiết kế được 1 cơ sở dữ liệu hoàn chỉnh.
* Nắm vững kiến thức nền tảng của môn học cơ sở dữ liệu.
* Tiến hành kiểm tra và chạy thử chương trình.

1. **Mục tiêu nghiên cứu.**

* Thiết kế thành công 1 hệ thống cơ sở dữ liệu quản lý xe buýt.
* Hiểu được cách thức hoạt động của cơ sở dữ liệu.
* Khiến ứng dụng trở nên có ích hơn cho xã hội.

1. **Đối tượng và phạm vi nghiên cứu.**

* Đối tượng: Các công ty Xe buýt
* Phạm vi ứng dụng: Quản lý xe buýt

1. **Phương pháp nghiên cứu.**

* Tài liệu: Tìm hiểu và khai thác các tài liệu liên quan đến cơ sở dữ liệu, các nền tảng kiến thức về công cụ Microsoft SQL Server Management.
* Thực hành: Vẽ sơ đồ ER, Chuyển sơ đồ ER thành mô hình quan hệ, Tiến hành kết nối cơ sở dữ liệu với Microsoft SQL Server Management. Tiến hành thao tác với dữ liệu.

# **CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

* 1. **Khái niệm cơ sở dữ liệu**

Cơ sở dữ liệu là tập hợp dữ liệu được tổ chức theo một cấu trúc chặt chẽ nhằm phục vụ (chia sẻ) cho nhiều mục tiêu khác nhau một cách có chọn lọc.

Ví dụ: Cơ sở dữ liệu nhân viên, cơ sở dữ liệu hàng hóa, cơ sở dữ liệu khách hàng,….v.v.

* 1. **Tổng quan về SQL**

SQL được xem là yếu tố chính đóng góp vào sự thành công của CSDL quan hệ khi áp dụng trong thực tế. Là ngôn ngữ mức cao, người dùng chỉ cần xác định kết quả của truy vấn là gì, phần còn lại là tính toán và tối ưu hoá câu lệnh được DBMS đảm nhiệm.

1970: SQL (Structured Query Language) bắt nguồn từ ngôn ngữ SEQUEL (Structured English QUEry Language), ngôn ngữ được thiết kế tại tập đoàn IBM nhằm khi đưa ra hệ quản trị CSDL có tên là SYSTEM-R.

DDL – Data Definition Language

DML – Data Manipulation Language

SQL – Structured Query Language

DCL – Data Control Language 

SQL gồm 2 nhóm lệnh

DDL: Tạo cấu trúc CSDL

DML: Thao tác trên CDSDL

• CREATE

• SELECT

• INSERT, UPDATE, DELETE…

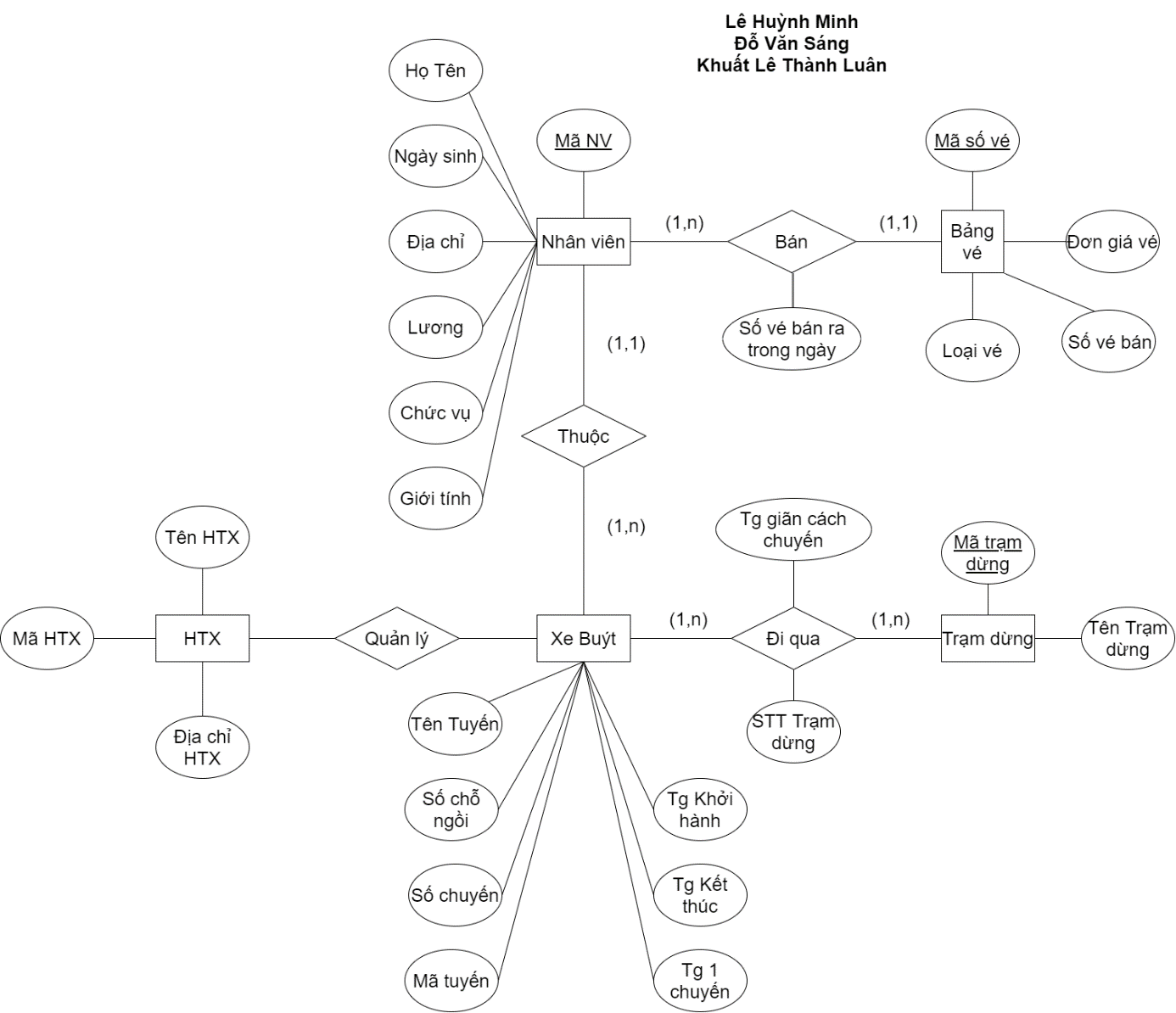
# **CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ**

1. **Mô tả bài toán**

* Các bến xe có hệ thống lưu thông bằng xe buýt kết nối các trạm với nhau. Mỗi bến có nhiều xe buýt lưu thông theo tuyến được chỉ định bằng mã tuyến (số xe buýt), Tên tuyến, Thời gian một chuyến, Số chỗ ngồi, Số chuyến, Thời gian khởi hành, Thời gian kết thúc. Xe buýt cùng số xe trực thuộc 1 Hợp tác xã.
* Hợp tác xã có mã hợp tác xã, tên hợp tác xã và địa chỉ hợp tác xã. Hợp tác xã quản lý nhiều số xe buýt.
* Với mỗi xe buýt có 1 hoặc 2 nhân viên tương ứng với 1 tài xế và 1 tiếp viên hoặc 1 nhân viên tức là chỉ có tài xế không có tiếp viên nào cả. Nhân viên gồm có các thông tin như sau: Mã nhân viên, Họ tên, Giới tính, Ngày sinh, Lương, Địa chỉ, Chức vụ (Tài xế hoặc Tiếp viên). Tiếp viên bán vé xe. Quan hệ bán vé phát sinh thêm thuộc tính số vé bán ra trong ngày.
* Thông tin về vé gồm mã tập vé, đơn giá vé, loại vé và số vé bán. Vé có 2 loại cơ bản là vé sinh viên và vé thường. 1 Nhân viên có thể bán nhiều tập vé cho nhiều hành khách và mỗi tập vé chỉ được bán 1 lần.
* Xe buýt đi qua nhiều trạm với trạm có mã trạm, tên trạm và khi đi qua các trạm thì phát sinh thêm thứ tự trạm mà xe buýt đi qua và thời gian giãn cách chuyến.
* Cần có 1 hệ CSDL quản lý xe buýt để lưu trữ các thông tin cần thiết và đặc biệt là số vé bán ra trong ngày khi nhân viên bán vé.

1. **Chuẩn bị**
   * 1. **Sơ đồ ER**

Sơ đồ ER cơ sở dữ liệu quản lý xe buýt



* + 1. **Mô hình quan hệ**

**STAFF**(MaNV, MaTuyen, HoTen, NgaySinh, Luong, DiaChi, GioiTinh, Chucvu, So\_Ve\_Ban\_Ra\_Trong\_Ngay)

**TICKET\_PACK**(MaTapVe, MaNV, DonGiaVe, LoaiVe, SoVeBan)

**CROSS\_STATION**(MaTuyen, MaTramDung, GianCachChuyen, STT\_TramDung)

**BUS**(MaTuyen, MaHTX, TenTuyen, ThoiGian\_MotChuyen, SoChoNgoi, SoChuyen, KhoiHanh, KetThuc)

**HTX**(MaHTX, TenHTX, DCHTX)

**STATION**(MaTramDung, TenTramDung)

**Những khó khăn gặp phải**

* Chủ đề quản lý xe buýt tương đối xa lạ với sinh viên vì nó không gần gũi như những bài thường làm như quản lý học sinh sinh viên, quản lý nhà hàng, quản lý công ty. Cơ sở dữ liệu quản lý xe buýt phải quản lý nhiều xe buýt và trong xe buýt chỉ có 2 hoặc 1 nhân viên. Xe buýt còn có nhiều lớp khác nhau như xe buýt cùng số xe và tuyến, xe buýt cùng hợp tác xã.
* Nhưng không thể nhập hết 200 tuyến vì nên không đủ nguồn lực và thời gian làm. Chúng em chỉ nhập 10 tuyến vì có quá nhiều trạm dừng không theo thứ tự khó kiểm soát và trạm đi qua cũng rất nhiều.

# **CHƯƠNG 3: THỬ NGHIỆM ỨNG DỤNG**

* 1. **Môi trường cài đặt**

CREATE DATABASE BusManager

use BusManager

CREATE TABLE HTX

(

MaHTX varchar(10) primary key not null,

TenHTX nvarchar(100) not null,

DCHTX nvarchar(50) not null

)

CREATE TABLE BUS

(

MaTuyen varchar(10) primary key not null,

MaHTX varchar(10) constraint FK\_BUS\_MHTX foreign key references HTX(MaHTX) not null,

TenTuyen nvarchar(100) not null,

ThoiGian\_MotChuyen int not null,

SoChoNgoi int not null,

SoChuyen int not null,

KhoiHanh time not null,

KetThuc time not null,

constraint check\_BUS check (KhoiHanh != KetThuc)

)

CREATE TABLE STAFF

(

MaNV varchar(20) primary key not null,

MaTuyen varchar(10) constraint FK\_ST\_MT foreign key references BUS(MaTuyen) not null,

HoTen nvarchar(50) not null,

NgaySinh smalldatetime not null,

Luong money not null,

DiaChi nvarchar(100) not null,

GioiTinh nvarchar(5) not null,

Chucvu nvarchar(50) not null,

--Doanh\_Thu\_Trong\_Ngay int not null,

So\_Ve\_Ban\_Ra\_Trong\_Ngay int not null,

TenTuyen nvarchar(50) not null

)

CREATE TABLE TICKET\_PACK

(

MaTapVe varchar(10) primary key not null,

MaNV varchar(20) constraint FK\_TK\_MNV foreign key references STAFF(MaNV)not null,

DonGiaVe int not null,

LoaiVe nvarchar(50) not null,

SoVeBan int not null

)

CREATE TABLE STATION

(

MaTramDung varchar(50) primary key not null,

TenTramDung nvarchar(50) not null

)

CREATE TABLE CROSS\_STATION

(

MaTuyen varchar(10) constraint FK\_CST\_MT foreign key references BUS(MaTuyen) not null,

MaTramDung varchar(50) constraint FK\_CST\_MTD foreign key references STATION(MaTramDung) not null,

constraint PK\_ST\_MTD\_MT primary key (MaTuyen, MaTramDung),

GianCachChuyen int not null,

STT\_TramDung int not null

)

--nhap Hop Tac Xa

INSERT INTO HTX VALUES ('0301410987',N'Hợp Tác Xã Vận Tải Xe Buýt Quyết Thắng',N'1129 Lạc Long Quân P.11 Q.Tân Bình TP.HCM')

--nhap Xe Buyt

INSERT INTO BUS VALUES ('08','0301410987',N'Bến Xe Buýt Quận 8- Đại học Quốc Gia',90,80,15,'4:40','20:30')

--nhap Nhan Vien

INSERT INTO STAFF VALUES ('08A107','08',N'Nguyễn Văn Cường','07/17/1970',11000000,N'789 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tài xế',80,N'Bến Xe Buýt Quận 8- Đại học Quốc Gia')

--nhap Ve

INSERT INTO TICKET\_PACK VALUES ('2951314','08A108',3000,N'Học Sinh Sinh Viên', 50)

--NHẬP TRẠM DỪNG CHO XE (NẾU TRẠM ĐÃ CÓ THÌ KHÔNG NHẬP NỮA)

-- xe so 8

INSERT INTO STATION VALUES ('BX87',N'Bến xe buýt A Khu ĐH Quốc Gia TP.HCM')

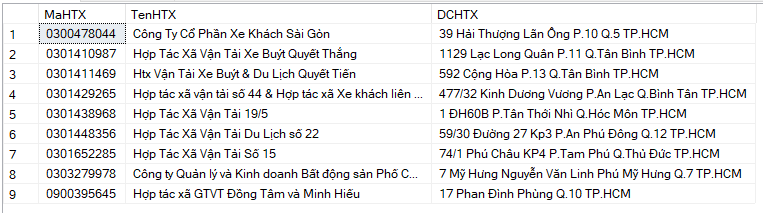
--nhap Tram Dung Di Qua

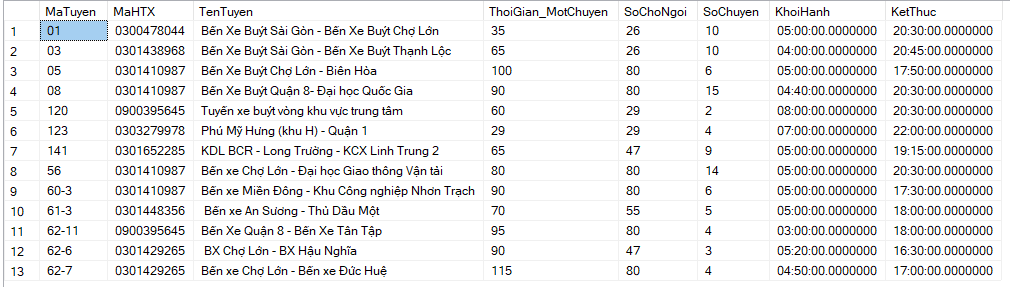
-- xe số 8

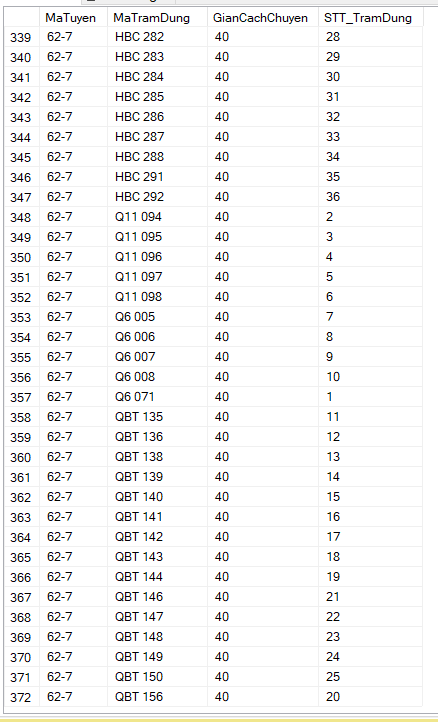
INSERT INTO CROSS\_STATION VALUES (N'08','BX87',3,0)

* 1. **Kết quả khởi chạy và đánh giá**

### Kết quả khởi chạy

*dbo.HTX*

*dbo.BUS*

 *dbo.STATION dbo.* *CROSS\_STATION*

**

*dbo.* *TICKET\_PACK*

**

### Đánh giá

Kết quả hoàn thành 90% như yêu cầu đề ra.

### 

# **CHƯƠNG 4: RÀNG BUỘC TOÀN VẸN (RBTV)**

* RBTV là một quy tắc định nghĩa trên một hoặc nhiều quan hệ do môi trường ứng dụng quy định . Đó chính là quy tắc để đảm bảo tính nhất quán của dữ liệu.
* Mỗi RBTV được định nghĩa bằng 1 thuật toán trong CSDL.
* Khóa nội, Khóa ngoại, NOT NULL là những RBTV về miền giá trị của thuộc tính trong quan hệ.

CREATE TABLE STAFF

(

MaNV varchar(20) primary key not null,

MaTuyen varchar(10) constraint FK\_ST\_MT foreign keyreferences BUS(MaTuyen) not null,

HoTen nvarchar(50) not null,

NgaySinh smalldatetime not null,

Luong money not null,

DiaChi nvarchar(100) not null,

GioiTinh nvarchar(5) not null,

Chucvu nvarchar(50) not null,

So\_Ve\_Ban\_Ra\_Trong\_Ngay int not null

)

# **CHƯƠNG 5: DẠNG CHUẨN VÀ CHUẨN HÓA**

**Chuẩn hóa là gì**

* Chuẩn hóa là kỹ thuật dùng để tạo ra một tập các quan hệ có các đặc điểm mong muốn dựa vào các yêu cầu về dữ liệu của 1 enterprise.
* Chuẩn hóa là 1 cách tiếp cận từ dưới lên (bottom-up approach) để thiết kế CSDL, bắt đầu từ các mối liên hệ giữa các thuộc tính.

**Mục đích**

* Loại bỏ các bất thường của 1 quan hệ để có được các quan hệ có cấu trúc tốt hơn, nhỏ hơn.

**Quan hệ có cấu trúc tốt (well-structured relation):**

* Là quan hệ có sự dư thừa dữ liệu là tối thiểu và cho phép người dùng thêm, sửa, xóa mà không gây ra mâu thuẫn dữ liệu.

**Sự dư thừa dữ liệu**

* Sự phụ thuộc giữa các thuộc tính gây ra sự dư thừa.
* Sự dư thừa sang dị thường.
* Các giá trị không xác định.
* Các bộ giả.

**Một số quy tắc khi thiết kế CSDL quan hệ**

* NT1: Rõ ràng về mặt ngữ nghĩa, tránh các sự phụ thuộc giữa các thuộc tính với nhau
* NT2: Tránh sự trùng lặp về nội dung đảm bảo tránh được các dị thường khi thao tác cập nhật dữ liệu .
* Phải có một số thao tác khi thêm mới và cập nhật vào lược đồ quan hệ, cũng như có thể gây sai hỏng trong trường hợp xóa bỏ các bộ.
* NT3: Tránh sử dụng các thuộc tính có nhiều giá trị Null
* Khó thực hiện các phép nối và kết hợp.
* NT4: Thiết kế các lược đồ quan hệ sao cho chúng có thể được nối với điều kiện bằng trên các thuộc tính là khoá chính hoặc khoá ngoài theo cách đảm bảo không sinh ra các bộ “giả”.

**Các dạng chuẩn**

**Dạng chuẩn 1 (1NF – first normal form)**

* Quan hệ R được gọi là ở dạng 1NF nếu miền giá trị của một thuộc tính (bất kỳ) chỉ chứa giá trị nguyên tố đơn (đơn trị, không phân chia được) và giá trị của mỗi thuộc tính cũng là một giá trị đơn lấy từ miền giá trị của nó.

**Dạng chuẩn 2 (2NF – second normal form)**

* Một quan hệ được gọi là ở dạng chuẩn 2 (2NF) nếu:
* Thỏa mãn dạng chuẩn 1NF .
* Các thuộc tính không khóa đều phụ thuộc hàm đầy đủ vào khóa chính.

**Dạng chuẩn 3 (3NF – third normal form)**

* Một lược quan hệ R được coi đạt dạng chuẩn 3 (3NF) nếu nó:
* Thỏa mãn là 2NF.
* Không có thuộc tính không khoá nào của R phụ thuộc bắc cầu vào khoá chính.

CREATE TABLE STAFF

(

MaNV varchar(20) primary key not null,

MaTuyen varchar(10) constraint FK\_ST\_MT foreign key references BUS(MaTuyen) not null,

HoTen nvarchar(50) not null,

NgaySinh smalldatetime not null,

Luong money not null,

DiaChi nvarchar(100) not null,

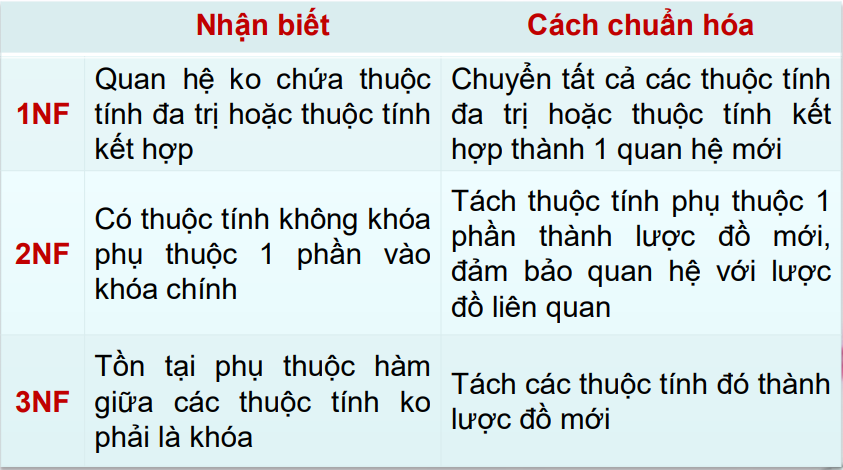
GioiTinh nvarchar(5) not null,

Chucvu nvarchar(50) not null,

So\_Ve\_Ban\_Ra\_Trong\_Ngay int not null

)

**Nhân viên** (MãNV, Mã tuyến, Họ tên, Địa chỉ, Lương, Giới tính, Ngày sinh, Chức vụ, Số vé bán ra trong ngày, Tên Tuyến)



**Dạng chuẩn BCNF (Boyce-Codd normal form)**

* Một lược đồ quan hệ R được gọi là ở dạng chuẩn Boyce-Codd (BCNF) nếu nó.
* Thỏa mãn dạng 3NF .
*  X→Y  F+ thì X là siêu khóa (chứa khóa của quan hệ) hoặc Y X. Nói cách khác, quan hệ đạt R sẽ không đạt BCNF nếu tồn tại phụ thuộc hàm mà vế trái không phải là khóa.

Nếu một lược đồ quan hệ không thoả mãn điều kiện BCNF, thủ tục chuẩn hóa bao gồm:

* Loại bỏ các thuộc tính khóa phụ thuộc hàm vào thuộc tính không khóa ra khỏi quan hệ.
* Tách chúng thành một quan hệ riêng có khoá chính là thuộc tính không khóa gây ra phụ thuộc.

**Tính chất của giải thuật phân rã BCNF**

Không mất mát thông tin.

Nhưng có thể không bảo toàn phụ thuộc hàm.

Là giải thuật không xác định (nondeterministic), phụ thuộc vào thứ tự các PTH được chọn để xét phân ra.

**Phân rã lược đồ quan hê**

Phân rã lược đồ quan hệ là việc tách lược đồ quan hệ kém chất lượng ban đầu (chưa đạt chuẩn) cùng với tập phụ thuộc hàm của nó thành những lược đồ quan hệ chất lượng hơn.

Sau phân rã, CSDL không còn lược đồ quan hệ R mà chỉ lưu lại các lược đồ quan hệ chiếu của nó R1 , R2, ..,Rn.

Hai vấn đề cần quan tâm:

Phân rã bảo toàn thông tin (khôi phục được thông tin ban đầu từ các lược đồ đã tách?) .

Phân rã bảo toàn Phụ thuộc hàm (Bảm đảm khôi phục được các PTH gốc).

**Phân rã bảo toàn thông tin**

Phân rã tạo ra nhiều bộ hơn nhưng nếu những bộ đó không đúng thì sẽ mất mát thông tin.

**Chuẩn hóa CSDL**

Quá trình chuẩn hóa được thực hiện qua nhiều bước. Mỗi bước tương ứng một dạng chuẩn.

Bước chuẩn hóa

– Bước 1: Đưa về dạng 1NF, loại bỏ các thuộc tính đa trị .

– Bước 2: Đưa về dạng 2NF, loại bỏ phụ thuộc hàm bộ phận vào khóa .

– Bước 3: Đưa về dạng 3NF, loại bỏ phụ thuộc bắc cầu vào khóa .

– Bước 4: Đưa về dạng BCNF: Mọi phụ thuộc hàm phải có vế trái là siêu khóa

# **CHƯƠNG 6: THỦ TỤC STORED PROCEDURE VÀ TRIGGER**

* + - 1. **Thủ tục Stored Procedure**

**Thủ tục là gì?**

* **Procedure (Thủ tục)** là một chương trình trong cơ sở dữ liệu gồm nhiều câu lệnh mà bạn lưu lại cho những lần sử dụng sau.
* Stored Procedure là một tập hợp các câu lệnh SQL dùng để thực thi một nhiệm vụ nhất định. Nó hoạt động giống như một hàm trong các ngôn ngữ lập trình khác.
* 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.

**Các Loại Stored Procedure:** Stored procedure có thể được chia thành 5 nhóm sau:

* **System Stored Prcedure :** Là những stored procedure chứa trong Master database và thường bắt đầu bằng tiếp đầu ngữ **sp\_ .** Các stored procedure này thuộc loại built-in và chủ yếu dùng trong việc quản lý database (administration) và security
* **Local Stored Procedure :** Ðây là loại thường dùng nhất. Chúng được chứa trong user database và thường được viết để thực hiện một công việc nào đó. Thông thường người ta nói đến stored procedure là nói đến loại này.
* **Temporary Stored Procedure** : Là những stored procedure tương tự như local stored procedure nhưng chỉ tồn tại cho đến khi connection đã tạo ra chúng bị đóng lại hoặc SQL Server shutdown.
* **Extended Stored Procedure** : Ðây là một loại stored procedure sử dụng một chương trình ngoại vi (external program) vốn được compiled thành một DLL để mở rộng chức năng hoạt động của SQL Server. Thường bắt đầu bằng tiếp đầu ngữ **xp\_**
* **Remote Stored Procedure** : Những stored procedure gọi stored procedure ở server khác.

**Khi nào bạn cần sử dụng Stored Procedure?**

Khi phải thực hiện một project lớn, đòi hỏi nghiêm ngặt về tốc độ thực thi thì Stored Procedure. Stored Procedure cho tốc độ thực thi nhanh hơn so với các câu lệnh SQL Server thông thường, Stored Procedure đã được lưu sẵn tại SQL server, do vậy chương trình không cần gởi hàng ngàn dòng lệnh đến SQL Server. Nó chỉ cần gọi một dòng lệnh để thực thi Stored Procedure, điều này giúp tăng tốc độ thực thi.

**Câu lệnh tạo thủ tục:**

CREATE PROCEDURE | PROC procedure\_name

(@parameter datatype ) [ OUT | OUTPUT | READONLY ]

AS

BEGIN

[declaration\_section]

Executable\_section

END;

**Ví dụ về store procedure trong CSDL ‘Quản Lý xe buýt trong thành phố’:**

--1. Xuất ra số xe buýt đi qua trạm được chỉ định

CREATE PROCEDURE sp\_TramDung(@TramDung nvarchar(50))

AS

BEGIN

select MaTuyen, TenTuyen

from bus

where matuyen in(

select matuyen

from cross\_station

where matramdung in(

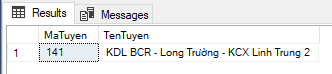
select matramdung

from station

where Tentramdung=@TramDung))

END

EXECUTE sp\_TramDung N'Đại Học GTVT'



**Các thủ tục khác:**

--2. Xuất ra tổng doanh thu từng nhân viên

CREATE PROCEDURE sp\_DoanhThuNV

AS

BEGIN

SELECT STAFF.HoTen AS N'Tên nhân viên', SUM((SoVeBan\*DonGiaVe)) AS N'Doanh thu trong ngày (đồng)'

FROM STAFF, TICKET\_PACK

WHERE TICKET\_PACK.MaNV=STAFF.MaNV

GROUP BY STAFF.HoTen

ORDER BY SUM((SoVeBan\*DonGiaVe))

END

EXECUTE sp\_DoanhThuNV

--3. Xuất ra tổng doanh thu từng số xe

CREATE PROCEDURE sp\_DoanhThuTungSoXe

AS

BEGIN

SELECT BUS.MaTuyen AS N'Mã tuyến', SUM((SoVeBan\*DonGiaVe)) AS N'Doanh thu trong ngày (đồng)'

FROM BUS, STAFF, TICKET\_PACK

WHERE BUS.MaTuyen=STAFF.MaTuyen AND TICKET\_PACK.MaNV=STAFF.MaNV

GROUP BY BUS.MaTuyen

ORDER BY SUM((SoVeBan\*DonGiaVe))

END

EXECUTE sp\_DoanhThuTungSoXe

--4. Xuất ra tổng doanh thu từng HTX

CREATE PROCEDURE sp\_DoanhThuHTX

AS

BEGIN

SELECT HTX.TenHTX AS N'Tên HTX', SUM((SoVeBan\*DonGiaVe)) AS N'Doanh thu trong ngày (đồng)'

FROM HTX, BUS, STAFF, TICKET\_PACK

WHERE HTX.MaHTX=BUS.MaHTX AND BUS.MaTuyen=STAFF.MaTuyen AND TICKET\_PACK.MaNV=STAFF.MaNV

GROUP BY HTX.TenHTX

ORDER BY SUM((SoVeBan\*DonGiaVe))

END

EXECUTE sp\_DoanhThuHTX

--5 Xuất ra số xe có nhiều chuyến nhất

CREATE PROCEDURE sp\_SoXeNhieuChuyenNhat

AS

BEGIN

SELECT Distinct MaTuyen, TenTuyen, SoChuyen

FROM BUS

WHERE SoChuyen = (SELECT MAX(SoChuyen) FROM BUS)

END

EXEC sp\_SoXeNhieuChuyenNhat

--6 Xuất ra nhân viên có lương lớn hơn lương của ít nhất 1 nhân viên xe số được chỉ định

CREATE PROCEDURE sp\_LuongLonHonItNhat1NhanVienXe(@MaTuyen nvarchar(5))

AS

BEGIN

select \*

from staff

where Luong > ANY(

select Luong

from staff

where MaTuyen=@MaTuyen

)

END

EXEC sp\_LuongLonHonItNhat1NhanVienXe N'05'

--7 Xuất ra số xe có khởi hành với giờ quy định

CREATE PROCEDURE sp\_XeKhoiHanhLuc(@Gio time)

AS

BEGIN

SELECT MaTuyen, TenTuyen, KhoiHanh

FROM BUS

WHERE KhoiHanh = @Gio

END

EXEC sp\_XeKhoiHanhLuc '5:00'

* + - 1. **Trigger**

**Trigger là gì?**

* **Trigger** là một thủ tục SQL được thực thi ở phía server khi có một sự kiện như **Insert**, **Delete**, hay **Update**.
* **Trigger** là một loại **stored procedure** **đặc biệt** **(không có tham số)** được thực thi (**execute**) một cách tự động khi có một sự kiện thay đổi dữ liệu (data modification).
* 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.

**Trigger được sử dụng khi nào?**

* Trigger thường được sử dụng để kiểm tra ràng buộc (check constraints) trên nhiều quan hệ (nhiều bảng/table) hoặc trên nhiều dòng (nhiều record) của bảng.
* Ngoài ra việc sử dụng Trigger để chương trình có những hàm chạy ngầm nhằm phục vụ nhưng trường hợp hữu hạn và thường không sử dụng cho mục đích kinh doanh hoặc giao dịch.
* Ngăn chặn việc xóa những dữ liệu quan trọng. (có thể dùng back up các dữ liệu quan trọng sang table khác phòng khi …bị xóa ngoài ý muốn).

**Ưu và nhược điểm của Trigger**

* **Ưu điểm của Trigger**
* Trigger có thể bắt lỗi business logic ở mức csdl.
* Có thể dùng trigger là một cách khác để thay thế việc thực hiện những công việc hẹn giờ theo lịch.
* Trigger rất hiệu quả khi được sử dụng để kiểm soát những thay đổi của dữ liệu trong bảng.
* **Nhược điểm của Trigger**
* Trigger chỉ là một phần mở rộng của việc kiểm tra tính hợp lệ của dữ liệu chứ không thay thế được hoàn toàn công việc này.
* Trigger hoạt động ngầm ở trong csdl, không hiển thị ở tầng giao diện. Do đó, khó chỉ ra được điều gì xảy ra ở tầng csdl.
* Trigger thực hiện các update lên bảng dữ liệu vì thế nó làm gia tăng lượng công việc lên csdl và làm cho hệ thống chạy chậm lại.

**Khai báo trigger như thế nào?**

CREATE TRIGGER trigger\_name ON table\_name

FOR INSERT | UPDATE | DELETE

BEGIN

[declaration\_section]

Executable\_section

END;

**Ví dụ về trigger trong CSDL ‘Quản Lý xe buýt trong thành phố’:**

/\* 1.1. Tạo Trigger để kiểm tra khi nhập dữ liệu vào bảng STAFF:

Nếu giới tính nằm ngoài Nam hoặc Nữ, Chức vụ nằm ngoài Tài xế hoặc tiếp viên, số vé nhỏ hơn 0, lương nhỏ hơn 6.900.000, tuổi nhỏ hơn 18 thì KHÔNG cho nhập. Ngược lại thì cho nhập. \*/

--1. Kiểm tra input của Staff (Nhân viên)

--1. Kiểm tra input của Staff (Nhân viên)

CREATE TRIGGER tr\_StaffInput

ON STAFF

FOR INSERT

AS

BEGIN

IF((SELECT GioiTinh FROM inserted) NOT IN (N'Nam', N'Nữ'))

BEGIN

PRINT N'Giới tính không hợp lệ.'

ROLLBACK TRAN

END

IF((SELECT Chucvu FROM inserted)

NOT IN (N'Tài xế', N'Tiếp Viên'))

BEGIN

PRINT N'Chức vụ không hợp lệ.'

ROLLBACK TRAN

END

IF((SELECT So\_Ve\_Ban\_Ra\_Trong\_Ngay FROM inserted) < 0))

BEGIN

PRINT N'Số vé phải lớn hơn hoặc bằng 0.'

ROLLBACK TRAN

END

IF((SELECT Luong FROM inserted)<6900000)

BEGIN

PRINT N'Lương phải lớn hơn hoặc bằng 6.900.000'

ROLLBACK TRAN

END

IF((SELECT DATEDIFF(YEAR, NgaySinh, getDate()) FROM inserted)<18)

BEGIN

PRINT N'Tuổi phải lớn hơn hoặc bằng 18'

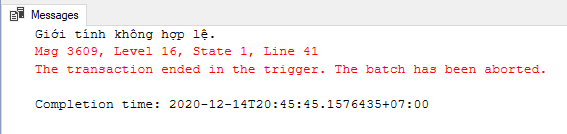
ROLLBACK TRAN

END

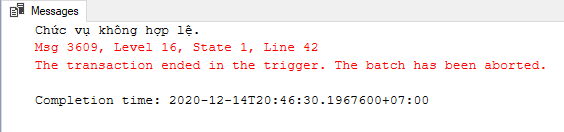
END

-- Test Trigger

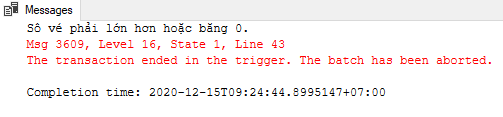
INSERT INTO STAFF VALUES ('08A420','08',N'Nguyễn Nguyên Nguyện','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Cis',N'Tiếp Viên',0)



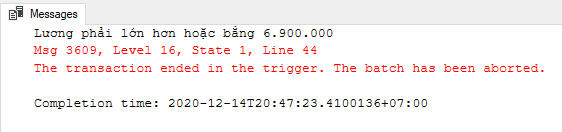
INSERT INTO STAFF VALUES ('08A420','08',N'Nguyễn Nguyên Nguyện','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Lơ xe',0)



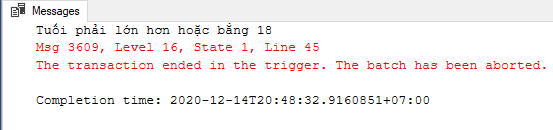
INSERT INTO STAFF VALUES ('08A420','08',N'Nguyễn Nguyên Nguyện','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',-10)



INSERT INTO STAFF VALUES ('08A420','08',N'Nguyễn Nguyên Nguyện','9/6/1970',300000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',0)



INSERT INTO STAFF VALUES ('08A420','08',N'Nguyễn Nguyên Nguyện','9/6/2005',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',0)



Các trigger khác:

--1.2. Kiểm tra sửa của Staff (Nhân viên)

CREATE TRIGGER tr\_StaffUpdate

ON STAFF

FOR UPDATE

AS

BEGIN

IF((SELECT GioiTinh FROM inserted) NOT IN (N'Nam', N'Nữ'))

BEGIN

PRINT N'Giới tính không hợp lệ.'

ROLLBACK TRAN

END

IF((SELECT Chucvu FROM inserted) NOT IN (N'Tài xế', N'Tiếp Viên'))

BEGIN

PRINT N'Chức vụ không hợp lệ.'

ROLLBACK TRAN

END

IF((SELECT So\_Ve\_Ban\_Ra\_Trong\_Ngay FROM inserted) < 0)

BEGIN

PRINT N'Số vé phải lớn hơn hoặc bằng 0.'

ROLLBACK TRAN

END

IF((SELECT Luong FROM inserted)<6900000)

BEGIN

PRINT N'Lương phải lớn hơn hoặc bằng 6.900.000'

ROLLBACK TRAN

END

IF((SELECT DATEDIFF(YEAR, NgaySinh, getDate()) FROM inserted)<18)

BEGIN

PRINT N'Tuổi phải lớn hơn hoặc bằng 18'

ROLLBACK TRAN

END

END

DROP TRIGGER tr\_StaffUpdate

--Test Trigger tr\_StaffUpdate

INSERT INTO STAFF VALUES ('08A421','08',N'Nguyễn Thử Nghiệm','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',0)

UPDATE STAFF SET MaTuyen='08',HoTen=N'Nguyễn Nguyên Nguyện', NgaySinh='9/6/1970', Luong=11000000, DiaChi=N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM', GioiTinh= N'Cis', Chucvu=N'Tiếp Viên',So\_ve\_ban\_ra\_trong\_ngay=0 WHERE MaNV='08A421'

UPDATE STAFF SET MaTuyen='08',HoTen=N'Nguyễn Nguyên Nguyện', NgaySinh='9/6/1970', Luong=11000000, DiaChi=N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM', GioiTinh= N'Nam', Chucvu=N'Lơ xe',So\_ve\_ban\_ra\_trong\_ngay=0 WHERE MaNV='08A421'

UPDATE STAFF SET MaTuyen='08',HoTen=N'Nguyễn Nguyên Nguyện', NgaySinh='9/6/1970', Luong=11000000, DiaChi=N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM', GioiTinh= N'Nam', Chucvu=N'Tiếp Viên',So\_ve\_ban\_ra\_trong\_ngay=-10 WHERE MaNV='08A421'

UPDATE STAFF SET MaTuyen='08',HoTen=N'Nguyễn Nguyên Nguyện', NgaySinh='9/6/1970', Luong=110000, DiaChi=N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM', GioiTinh= N'Nam', Chucvu=N'Tiếp Viên',So\_ve\_ban\_ra\_trong\_ngay=0 WHERE MaNV='08A421'

UPDATE STAFF SET MaTuyen='08',HoTen=N'Nguyễn Nguyên Nguyện', NgaySinh='9/6/2005', Luong=11000000, DiaChi=N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM', GioiTinh= N'Nam', Chucvu=N'Tiếp Viên',So\_ve\_ban\_ra\_trong\_ngay=0 WHERE MaNV='08A421'

--1.3 Trigger xóa staff

CREATE TRIGGER tr\_StaffRemove

ON STAFF

INSTEAD OF DELETE

AS

BEGIN

DELETE TICKET\_PACK WHERE MaNV=(SELECT MaNV FROM deleted)

DELETE STAFF WHERE MaNV=(SELECT MaNV FROM deleted)

END

DROP TRIGGER tr\_StaffRemove

INSERT INTO STAFF VALUES ('69','08',N'Nguyễn Ngôn','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',0)

DELETE STAFF WHERE MaNV='69'

--2.1. Kiểm tra nhập của BUS (Xe buýt)

CREATE TRIGGER tr\_BusInput

ON BUS

FOR INSERT

AS

BEGIN

IF((SELECT SoChoNgoi FROM inserted)<20)

BEGIN

PRINT N'Số chỗ ngồi phải lớn hơn hoặc bằng 20.'

ROLLBACK TRAN

END

IF((SELECT SoChuyen FROM inserted)<20)

BEGIN

PRINT N'Số chuyến phải lớn hơn hoặc bằng 20.'

ROLLBACK TRAN

END

IF((SELECT ThoiGian\_MotChuyen FROM inserted)<20)

BEGIN

PRINT N'Thời gian 1 chuyến phải lớn hơn hoặc bằng 20 phút.'

ROLLBACK TRAN

END

IF((SELECT CAST(KhoiHanh AS Time) FROM inserted)>(SELECT CAST(KetThuc AS Time) FROM inserted))

BEGIN

PRINT N'Thời gian khởi hành phải nhỏ hơn thời gian kết thúc.'

ROLLBACK TRAN

END

END

--DROP TRIGGER tr\_BusInput

--TEST TRIGGER NHẬP XE BUÝT

INSERT INTO BUS VALUES ('420','0301410987',N'Bến xe buýt Weeds',90,80,348,'14:40','2:30')

INSERT INTO BUS VALUES ('420','0301410987',N'Bến xe buýt Weeds',19,80,348,'4:40','20:30')

INSERT INTO BUS VALUES ('420','0301410987',N'Bến xe buýt Weeds',90,19,348,'4:40','20:30')

INSERT INTO BUS VALUES ('420','0301410987',N'Bến xe buýt Weeds',90,80,19,'4:40','20:30')

--2.2. Kiểm tra sửa của BUS (Xe buýt)

CREATE TRIGGER tr\_BusUpdate

ON BUS

FOR UPDATE

AS

BEGIN

IF((SELECT SoChoNgoi FROM inserted)<20)

BEGIN

PRINT N'Số chỗ ngồi phải lớn hơn hoặc bằng 20.'

ROLLBACK TRAN

END

IF((SELECT SoChuyen FROM inserted)<20)

BEGIN

PRINT N'Số chuyến phải lớn hơn hoặc bằng 20.'

ROLLBACK TRAN

END

IF((SELECT ThoiGian\_MotChuyen FROM inserted)<20)

BEGIN

PRINT N'Thời gian 1 chuyến phải lớn hơn hoặc bằng 20.'

ROLLBACK TRAN

END

IF((SELECT CAST(KhoiHanh AS Time) FROM inserted)>(SELECT CAST(KetThuc AS Time) FROM inserted))

BEGIN

PRINT N'Thời gian khởi hành phải nhỏ hơn thời gian kết thúc.'

ROLLBACK TRAN

END

END

DROP TRIGGER tr\_BusUpdate

INSERT INTO BUS VALUES ('421','0301410987',N'Bến xe buýt Weeds+1',90,80,29,'4:40','20:30')

--TEST TRIGGER CẬP NHẬT XE BUS

UPDATE BUS SET MaHTX='0301410987', TenTuyen=N'Bến xe buýt Grass+1',ThoiGian\_MotChuyen=90,SoChoNgoi=80,SoChuyen=348,KhoiHanh='14:40',KetThuc='2:30' WHERE MaTuyen='421'

UPDATE BUS SET MaHTX='0301410987', TenTuyen=N'Bến xe buýt Grass+1',ThoiGian\_MotChuyen=19,SoChoNgoi=80,SoChuyen=348,KhoiHanh='2:30',KetThuc='14:40' WHERE MaTuyen='421'

UPDATE BUS SET MaHTX='0301410987', TenTuyen=N'Bến xe buýt Grass+1',ThoiGian\_MotChuyen=90,SoChoNgoi=19,SoChuyen=348,KhoiHanh='2:30',KetThuc='14:40' WHERE MaTuyen='421'

UPDATE BUS SET MaHTX='0301410987', TenTuyen=N'Bến xe buýt Grass+1',ThoiGian\_MotChuyen=90,SoChoNgoi=80,SoChuyen=19,KhoiHanh='2:30',KetThuc='14:40' WHERE MaTuyen='421'

--UPDATE BUS SET MaHTX='0301410987', TenTuyen=N'Bến xe buýt Grass+1',ThoiGian\_MotChuyen=90,SoChoNgoi=80,SoChuyen=348,KhoiHanh='2:30',KetThuc='14:40' WHERE MaTuyen='421'

--2.3. Trigger xóa của BUS (Xe buýt)

CREATE TRIGGER tr\_BusRemove

ON BUS

INSTEAD OF DELETE

AS

BEGIN

DELETE TICKET\_PACK WHERE MaNV=(SELECT MaNV FROM STAFF INNER JOIN deleted ON STAFF.MaTuyen=deleted.MaTuyen)

DELETE STAFF WHERE MaNV=(SELECT MaNV FROM STAFF INNER JOIN deleted ON STAFF.MaTuyen=deleted.MaTuyen)

DELETE CROSS\_STATION WHERE MaTuyen=(SELECT MaTuyen FROM deleted)

DELETE BUS WHERE MaTuyen=(SELECT MaTuyen FROM deleted)

PRINT N'Đã xóa xe buýt cùng với nhân viên và các quan hệ trạm của xe đó.'

END

DROP TRIGGER tr\_BusRemove

--Test Trigger Xóa Xe buýt

INSERT INTO STAFF VALUES ('421A69','421',N'Trần Thử Nghiệm','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',50)

INSERT INTO TICKET\_PACK VALUES ('421HS01','421A69',3000,N'Học Sinh Sinh Viên', 50)

INSERT INTO TICKET\_PACK VALUES ('421HS02','421A69',6000,N'Vé Thường', 50)

INSERT INTO CROSS\_STATION VALUES (N'421','BX87',3,0)

INSERT INTO CROSS\_STATION VALUES (N'421','QTD 177',3,1)

DELETE BUS WHERE MaTuyen='421'

--3.1. Trigger xóa HTX

CREATE TRIGGER tr\_HTXRemove

ON HTX

INSTEAD OF DELETE

AS

BEGIN

DELETE TICKET\_PACK WHERE MaNV=(SELECT MaNV FROM BUS INNER JOIN STAFF ON BUS.MaTuyen=STAFF.MaTuyen INNER JOIN deleted ON BUS.MaHTX=deleted.MaHTX)

DELETE STAFF WHERE MaNV=(SELECT MaNV FROM BUS INNER JOIN STAFF ON BUS.MaTuyen=STAFF.MaTuyen INNER JOIN deleted ON BUS.MaHTX=deleted.MaHTX)

DELETE CROSS\_STATION WHERE MaTuyen=(SELECT MaTuyen FROM deleted INNER JOIN BUS ON deleted.MaHTX=BUS.MaHTX)

DELETE BUS WHERE MaTuyen=(SELECT MaTuyen FROM deleted INNER JOIN BUS ON deleted.MaHTX=BUS.MaHTX)

DELETE HTX WHERE MaHTX=(SELECT MaHTX FROM deleted)

PRINT N'Đã xóa HTX, xe buýt cùng với nhân viên và các quan hệ trạm của xe đó.'

END

DROP TRIGGER tr\_HTXRemove

INSERT INTO HTX VALUES ('420420420',N'Hợp Tác Xã Vận Tải Xe Buýt Weeds',N'1129 Lạc Long Quân P.11 Q.Tân Bình TP.HCM')

INSERT INTO BUS VALUES ('42','420420420',N'Bến xe buýt Weeds+1',90,80,29,'4:40','20:30')

INSERT INTO STAFF VALUES ('42A69','42',N'Trần Thử Nghiệm','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',50)

INSERT INTO TICKET\_PACK VALUES ('42HS01','42A69',3000,N'Học Sinh Sinh Viên', 50)

INSERT INTO TICKET\_PACK VALUES ('42HS02','42A69',3000,N'Vé Thường', 50)

INSERT INTO CROSS\_STATION VALUES (N'42','BX87',3,0)

INSERT INTO CROSS\_STATION VALUES (N'42','QTD 177',3,1)

DELETE HTX WHERE MaHTX='420420420'

--4.1. Kiểm tra nhập của TICKET\_PACK (Vé Xe buýt)

CREATE TRIGGER tr\_TicketInput

ON TICKET\_PACK

AFTER INSERT

AS

BEGIN

UPDATE STAFF SET So\_Ve\_Ban\_Ra\_Trong\_Ngay+=INSERTED.SoVeBan

FROM INSERTED

WHERE INSERTED.MaNV=STAFF.MaNV

END

DROP TRIGGER tr\_TicketInput

INSERT INTO HTX VALUES ('420420420',N'Hợp Tác Xã Vận Tải Xe Buýt Weeds',N'1129 Lạc Long Quân P.11 Q.Tân Bình TP.HCM')

INSERT INTO BUS VALUES ('42','420420420',N'Bến xe buýt Weeds+1',90,80,29,'4:40','20:30')

INSERT INTO STAFF VALUES ('42A69','42',N'Trần Thử Nghiệm','9/6/1970',11000000,N'10 Lạc Long Quân P.10 Q.Tân Bình, TP.HCM',N'Nam',N'Tiếp Viên',0)

INSERT INTO TICKET\_PACK VALUES ('42HS01','42A69',3000,N'Học Sinh Sinh Viên', 50)

INSERT INTO TICKET\_PACK VALUES ('42PB02','42A69',7000,N'Vé Thường', 50)

--4.2. Kiểm tra XÓA của TICKET\_PACK (Vé Xe buýt)

CREATE TRIGGER tr\_TicketRemove

ON TICKET\_PACK

AFTER DELETE

AS

BEGIN

UPDATE STAFF SET So\_Ve\_Ban\_Ra\_Trong\_Ngay-=deleted.SoVeBan

FROM DELETED

WHERE deleted.MaNV=STAFF.MaNV

END

DELETE TICKET\_PACK WHERE MaTapVe='42HS01'

--4.3. Kiểm tra SỬA của TICKET\_PACK (Vé Xe buýt)

CREATE TRIGGER tr\_TicketUpdate

ON TICKET\_PACK

AFTER UPDATE

AS

BEGIN

UPDATE STAFF SET So\_Ve\_Ban\_Ra\_Trong\_Ngay=So\_Ve\_Ban\_Ra\_Trong\_Ngay-deleted.SoVeBan+inserted.SoVeBan

FROM DELETED, inserted

WHERE deleted.MaNV=STAFF.MaNV AND inserted.MaNV=STAFF.MaNV

END

DROP TRIGGER tr\_TicketUpdate

UPDATE TICKET\_PACK SET SoVeBan=175 WHERE MaTapVe='42PB02'

**CHƯƠNG 7: PHÂN QUYỀN TRONG CSDL**

* + - 1. **Tổng quan về phân quyền và người dùng trong CSDL**

**Phân quyền trong CSDL là gì?**

Để phát huy được tính hiệu quả và phối hợp làm việc giữa các cá nhân và bộ phận, người dùng hệ thống cần phải được phân quyền tương đương vơi chức vụ và quyền hạn của người đó. Phân quyền là một trong những chức năng quan trọng để bảo mật thông tin.

Phân quyền giúp người quản lý có thể chỉ định ai được quyền truy cập, được quyền sử dụng chức năng nào.

**Người dùng CSDL là gì?**

Mỗi CSDL có một danh sách người dung được xác thức để truy cập đến CSDL đó.

Mỗi người dung luôn gắn với một Login ID ở mức độ Server.

Khi tạo mới một database use:

* User chỉ có quyền chọn ngữ cảnh CSDL, không có quyền thực thi các thao tác trên CSDL và trên các đối tượng của CSDL đó.
* Để có thể thực hiện các thao tác trên CSDL và trên các đối tượng của CSDL thì user phải được cấp quyền thao tác trên CSDL và trên các đối tượng của CSDL
  + - 1. **Phân quyền và người dùng trong CSDL ‘Quản lý xe BUS trong 1 thành phố’**

Database BusManager phân quyền thành 2 user. Các user đó là:

* BusManagerNomal: là người được trao 1 phần quyền điều hành đối với CSDL
* BusManagerAdmin; là người được trao quyền điều hành toàn bộ CSDL

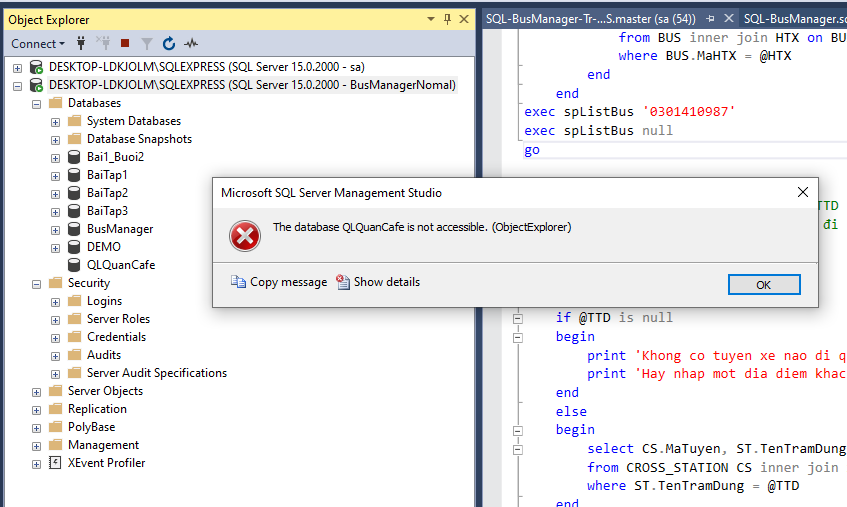
Phân quyền user trong CSDL ‘Quản lý xe BUS trong 1 thành phố’

Hinh ảnh sau khi phân quyền và đăng nhập vào từng user

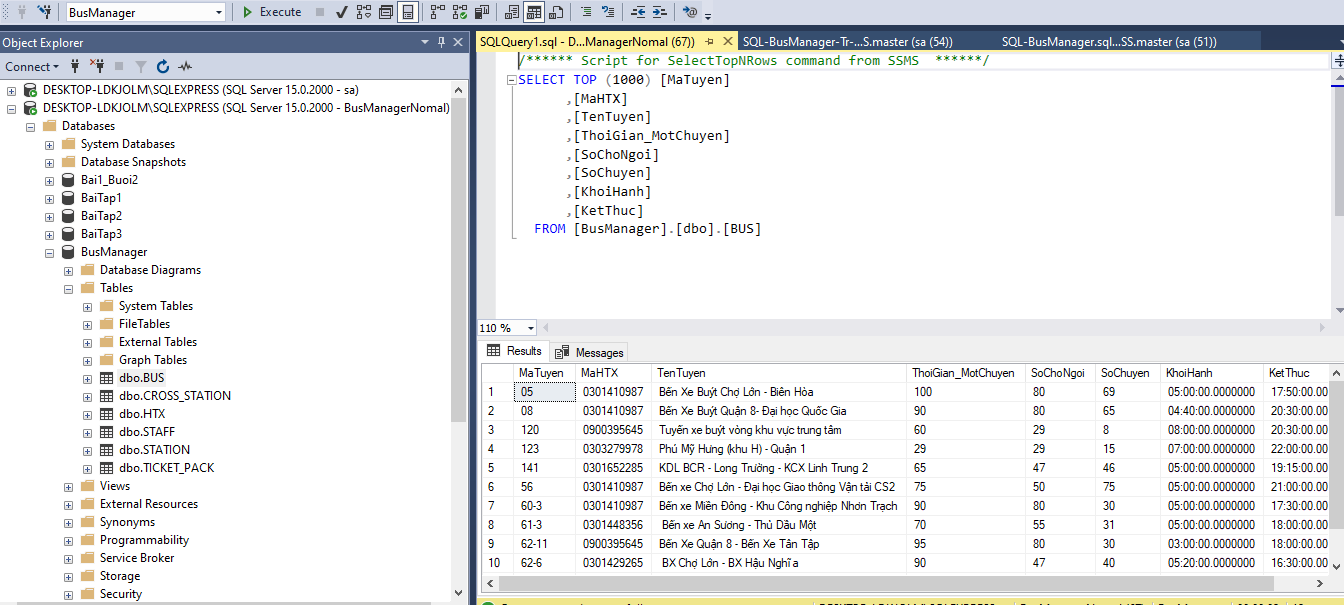
**Phân quyền với user BusManagerNamal**

User BusManagerNomal phân quyền chỉ cho phép truy cập vào Database BusManage nhưng không được phép truy cập vào các Database khác trong

Database Snapshots

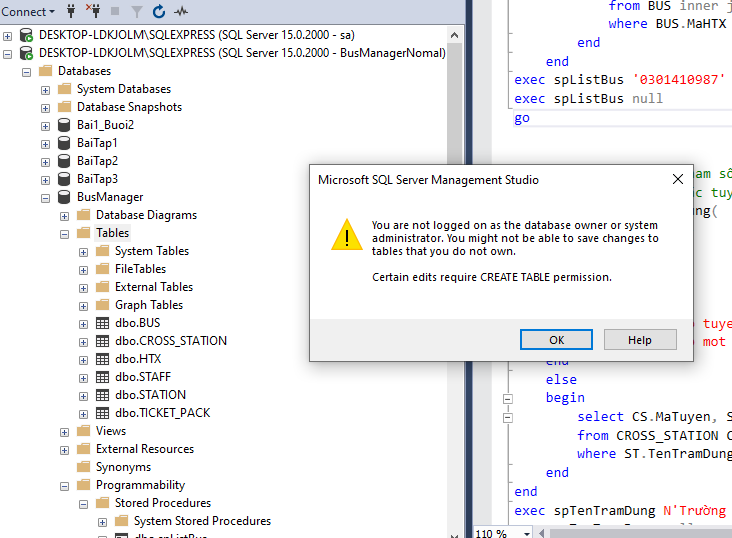


Kết quả thực hiện truy vấn từ bảng dbo.BUS là:



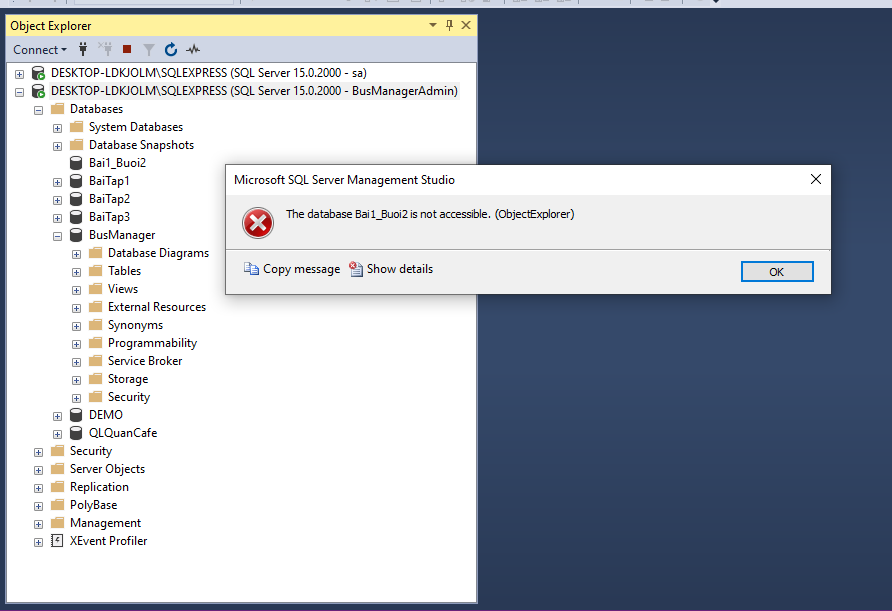
BusManagerNamal không có quyền thực hiện các Stored Procedures, Function.

BusManagerNamal chỉ được cấp một vài quyền. Trong đó không được cấp quyền tạo mới bảng. Qua đó BusManagerNamal không thể thay đổi cấu trúc của cơ sở dữ liệu

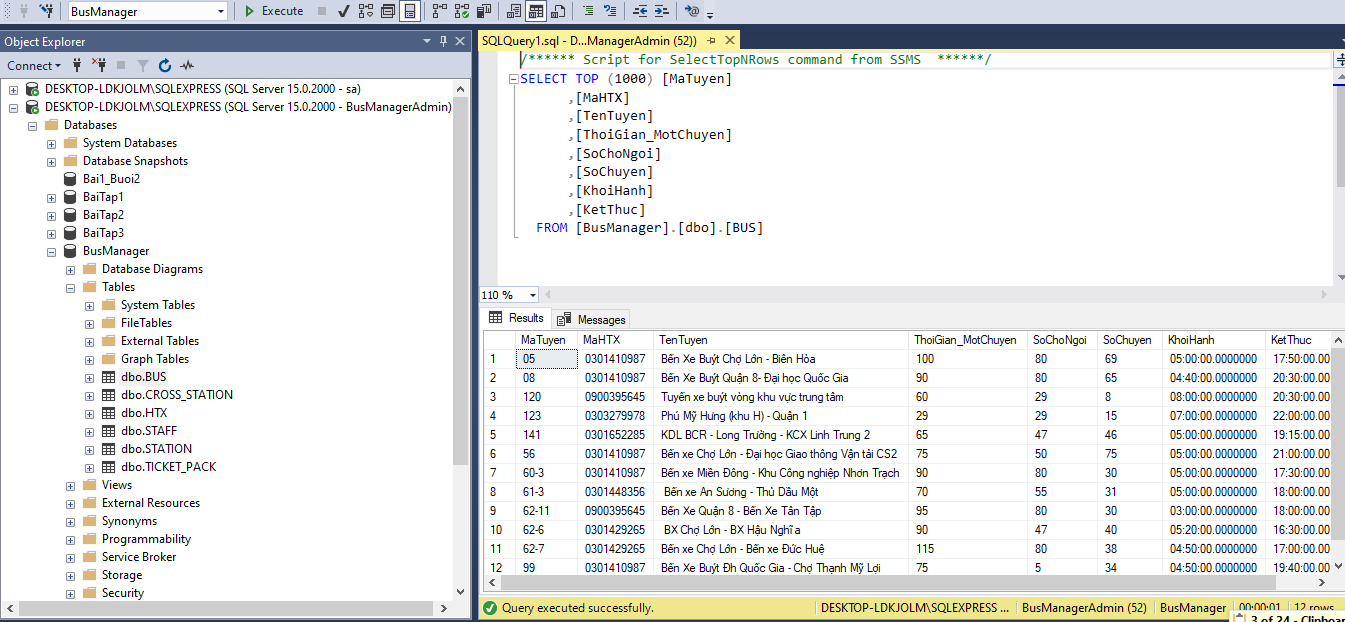


**Phân quyền với user BusManagerAdmin**

Giống như BusManagerNomal, BusManagerAdmin cũng không thể truy cập vào dữ liệu khác trong Database Snapshots ngoài database Busmanager.



Kết quả khi truy vấn bảng dbo.BUS của user BusManagerAdmin :



BusManagerAdmin được cấp toàn quyền sử dụng trên CSDL thực hiện các Stored Procedures, Function , tạo bảng, …

Ví dụ : BusManagerAdmin được cấp quyền tạo mới bảng. Qua đó BusManagerNamal có thể thay đổi cấu trúc của cơ sở dữ liệu.

