

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



BÁO CÁO CUỐI KÌ MÔN HỆ CƠ SỞ DỮ LIỆU

BÁO CÁO CUỐI KÌ MÔN HỆ CƠ SỞ DỮ LIỆU

Người hướng dẫn: **TS DOÃN XUÂN THANH**

Người thực hiện: **NGUYỄN DUY ĐÔNG – 52000749**

Lớp : 20050301

Khoá : 24

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2014

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO CUỐI KÌ MÔN HỆ CƠ SỞ DỮ LIỆU

BÁO CÁO CUỐI KÌ MÔN HỆ CƠ SỞ DỮ LIỆU

Người hướng dẫn: **TS DOÃN XUÂN THANH**

Người thực hiện: **NGUYỄN DUY ĐÔNG**

Lớp : **20050301**

Khoá : **24**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2014

LỜI CẢM ƠN

Đầu tiên em xin cảm ơn Trường đại học Tôn Đức Thắng đã đưa bộ môn Hệ cơ sở dữ liệu vào giảng dạy. Bên cạnh đó, em cũng xin cảm ơn thầy Doãn Xuân Thanh đã đồng hành cùng em trong suốt học kỳ vừa qua. Nhờ vào việc giảng dạy và giải đáp tận tình của thầy mà em có thể nắm được kiến thức của môn học này.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi và được sự hướng dẫn của TS Nguyễn Văn A;. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày tháng năm

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Duy Đông

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Báo cáo cuối kì tổng hợp những kiến thức đã học trong học kỳ qua dưới 4 phần. Mỗi phần là một chủ đề quan trọng khác nhau.

MỤC LỤC

LỜI CẢM ƠN	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	iii
TÓM TẮT	iv
MỤC LỤC	1
DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT	3
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ	4
PHẦN 1	6
1.1 Yêu cầu đề bài.....	6
1. Viết các câu lệnh SQL để thực hiện các thao tác sau: (0.5 điểm) ..	6
2. Viết các hàm sau: (1.0 điểm)	7
3. Viết các thủ tục sau: (0.5 điểm)	7
4. Viết 1 trigger sau: (1 điểm).....	7
1.2 Giải quyết yêu cầu	7
1.2.1 Câu 1:	7
1.2.2 Câu 2:	10
1.2.3 Câu 3:	14
1.2.4 Câu 4:	14
PHẦN 2	17
2.1 Yêu cầu đề bài:.....	17
2.2 Giải quyết yêu cầu:	17
PHẦN 3	22
3.1 Yêu cầu đề bài:.....	22
3.2 Giải quyết yêu cầu:	22
PHẦN 4	26
4.1 Yêu cầu đề bài:.....	26
4.2 Giải quyết yêu cầu:	26

TÀI LIỆU THAM KHẢO.....	32
PHỤ LỤC.....	33

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC KÝ HIỆU

CÁC CHỮ VIẾT TẮT

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1.1 Tạo database.....	8
Hình 1.2 Tạo bảng Sân bóng.....	8
Hình 1.3 Tạo bảng dịch vụ.....	8
Hình 1.4 Tạo bảng giá tiền thuê sân.....	8
Hình 1.5 Tạo bảng thuê sân.....	9
Hình 1.6 Tạo bảng chi tiết thuê sân	9
Hình 1.7 Nhập dữ liệu vào bảng sân bóng	9
Hình 1.8 Nhập dữ liệu vào bảng dịch vụ	10
Hình 1.9 Nhập dữ liệu vào bảng giá tiền thuê sân	10
Hình 1.10 Nhập dữ liệu vào bảng thuê sân	10
Hình 1.11 Nhập dữ liệu vào bảng chi tiết thuê sân	10
Hình 1.12 Hàm phát sinh mã số tự động.....	11
Hình 1.13 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc	12
Hình 1.14 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc	12
Hình 1.15 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc	13
Hình 1.16 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc	13
Hình 1.17 Thủ tục thêm vào bảng giá tiền thuê sân.....	14
Hình 1.18 Thủ tục cập nhật đơn giá dịch vụ thuê áo thêm 10.000	14
Hình 1.19 Tạo trigger.....	15
Hình 1.20 Trigger thêm vào bảng chi tiết thuê sân	15
Hình 1.21 Trigger xóa bảng chi tiết thuê sân.....	15
Hình 1.22 Trigger sửa bảng chi tiết thuê sân	16
 Hình 3.1 Sơ đồ ERD	 24
Hình 3.2 Lược đồ quan hệ.....	25

Hình 4.1 Bộ giá trị đầu vào	26
Hình 4.2 Hàm tìm khóa của lược đồ quan hệ.....	27
Hình 4.3 Kết quả sau khi thực hiện hàm tìm khóa.....	27
Hình 4.4 Hàm tìm phủ tối thiểu	28
Hình 4.5 hàm tìm phủ tối thiểu	29
Hình 4.6 Kết quả khi thực hiện hàm tìm phủ tối thiểu.....	29
Hình 4.7 Kiểm tra đạt chuẩn 2NF	30
Hình 4.8 Kiểm tra đạt chuẩn 3NF	30
Hình 4.9 Kiểm tra đạt chuẩn BCNF	31
Hình 4.10 Kết quả sau khi thực hiện tìm dạng chuẩn	31

DANH MỤC BẢNG

Bảng 2.1 Tìm tất cả các khóa	18
------------------------------------	----

PHẦN 1

1.1 Yêu cầu đề bài

Cho lược đồ CSDL quản lý Sân bóng đá mini như sau:

- Sân bóng (mã sân bóng, tên sân bóng, loại sân)

Tân từ: Sân bóng có mã sân bóng là khoá chính, tên sân bóng, loại sân cho biết sân đó được thi đấu bao nhiêu người. Ví dụ: sân 5, sân 7, sân 11, ...

- Dịch vụ (mã dịch vụ, tên dịch vụ, số lượng, đơn giá)

Tân từ: Bảng dịch vụ thể hiện các dịch vụ mà sân bóng cung cấp, như: thuê áo đấu, giày thi đấu, nước uống,

- Giá tiền thuê sân (Mã giá tiền, từ giờ, đến giờ, số tiền)

Tân từ: Tùy thuộc vào khung giờ thi đấu mà có giá tiền khác nhau. Ví dụ: Từ 6:00 AM đến 8:00 AM thì mỗi giờ tốn 300.000, từ 10:00 AM đến 11:59 AM thì mỗi giờ tốn 100.000.

- Thuê sân (Mã thuê sân, điện thoại khách hàng, mã sân, ngày thuê sân, giờ bắt đầu, giờ kết thúc, tổng tiền)

Tân từ: Khi khách hàng đến thuê sân, hệ thống lưu lại các thông tin như: mã thuê sân, số điện thoại khách hàng thuê sân, mỗi lần thuê sân chỉ được thuê 1 sân (mã sân), ngày thuê sân, giờ bắt đầu và giờ kết thúc trận đấu; tổng tiền cho biết tổng tiền thuê sân thi đấu và những dịch vụ đã sử dụng.

- Chi tiết thuê sân (Mã thuê sân, mã dịch vụ, số lượng, đơn giá)

Tân từ: Mỗi lần thuê sân thi đấu khách hàng có thể sử dụng một số dịch vụ mà sân bóng cung cấp, như: uống nước, thuê áo, ...

1. Viết các câu lệnh SQL để thực hiện các thao tác sau: (0.5 điểm)

- a. Tạo Cơ sở dữ liệu, tạo bảng bao gồm khoá chính, khoá ngoại. (0.25 điểm)
- b. Thêm dữ liệu cho các bảng theo đúng trình tự bên 1 trước bên nhiều sau. Mỗi bảng ít nhất 5 dòng dữ liệu. (0.25 điểm).

2. Viết các hàm sau: (1.0 điểm)

- a. Viết hàm phát sinh mã số tự động cho bảng thuê sân theo nguyên tắc:
Ngày thuê sân + số thứ tự gồm 3 chữ số tăng dần. Ví dụ: Ngày thuê sân là ngày 20/06/2022 và hiện đã có lần thuê sân thứ 3 (Bảng Thuê sân đang có mã số là: “20062022003” thì mã số tự động tiếp theo sẽ là “20062022004”) (0.5 điểm)
- b. Viết hàm trả về số tiền thuê sân khi biết được giờ vào sân và giờ ra khỏi sân (0.5 điểm)

3. Viết các thủ tục sau: (0.5 điểm)

- a. Thêm vào bảng Giá tiền thuê sân có kiểm tra khoá chính, giờ vào phải nhỏ hơn giờ ra. (0.25 điểm)
- b. Cập nhật đơn giá cho dịch vụ thuê áo thêm 10.000 (0.25 điểm).

4. Viết 1 trigger sau: (1 điểm)

Khi thêm, xoá, sửa dữ liệu trong bảng Chi tiết thuê sân:

- a. Cập nhật lại Tổng tiền trong bảng thuê sân. Trong đó, tổng tiền = tiền thuê sân + tiền sử dụng dịch vụ của lần thuê sân đó (0.5 điểm)
- b. Đơn giá trong bảng Chi tiết thuê sân phải bằng với đơn giá trong bảng dịch vụ (0.5 điểm)

1.2 Giải quyết yêu cầu

1.2.1 Câu 1:

- a. Tạo Cơ sở dữ liệu, tạo bảng bao gồm khoá chính, khoá ngoại.

- Tạo cơ sở dữ liệu:

```
use master
go
if exists(select * from sysdatabases where NAME = 'QUANLYSANBONG')
    DROP DATABASE QUANLYSANBONG
go
create database QUANLYSANBONG
go

use QUANLYSANBONG
go
```

Hình 1.1 Tạo database

- Tạo bảng sân bóng:

```
create table SanBong (
    MaSB char(10) not null,
    TenSB nvarchar(50) not null,
    LoaiSan nvarchar(20) not null,
    primary key (MaSB)
);
```

Hình 1.2 Tạo bảng Sân bóng

- Tạo bảng dịch vụ:

```
create table DichVu (
    MaDV char(10) not null,
    TenDichVu nvarchar(50),
    SoLuong int not null,
    DonGia int not null,
    primary key (MaDV)
);
```

Hình 1.3 Tạo bảng dịch vụ

- Tạo bảng giá tiền thuê sân:

```
create table GiaTienThueSan (
    MaGT char(10) not null,
    startAt time(0) not null,
    endAt time(0) not null,
    SoTien int not null,
    primary key (MaGT)
);
```

Hình 1.4 Tạo bảng giá tiền thuê sân

- Tạo bảng thuê sân:

```
create table ThueSan (
    MaThueSan char(50) not null,
    SDTKH char(10) not null,
    MaSB char(10) not null,
    NgayThue date not null,
    GioBD time(0) not null,
    GioKetThuc time(0) not null,
    TongTien int not null,
    primary key (MaThueSan)
);
```

Hình 1.5 Tạo bảng thuê sân

- Tạo bảng chi tiết thuê sân:

```
create table ChiTietThueSan (
    MaThueSan char(50) not null,
    MaDV char(10) not null,
    SoLuong int not null,
    DonGia int not null,
    primary key (MaThueSan, MaDV),
    foreign key (MaThueSan) references ThueSan(MaThueSan),
    foreign key (MaDV) references DichVu(MaDV)
);
```

Hình 1.6 Tạo bảng chi tiết thuê sân

- Thêm dữ liệu cho các bảng theo đúng trình tự bên 1 trước bên nhiều sau.
Mỗi bảng ít nhất 5 dòng dữ liệu.

- Thêm dữ liệu vào bảng sân bóng:

```
insert into SanBong values ('SB001', N'Sân Bóng 1', N'Sân 5 người'),
('SB002', N'Sân Bóng 2', N'Sân 7 người'),
('SB003', N'Sân Bóng 3', N'Sân 11 người'),
('SB004', N'Sân Bóng 4', N'Sân 5 người'),
('SB005', N'Sân Bóng 5', N'Sân 11 người')
```

Hình 1.7 Nhập dữ liệu vào bảng sân bóng

- Thêm dữ liệu vào bảng dịch vụ:

```
insert into DichVu values ('DV001', N'Thuê áo đấu', 100, 40000),
('DV002', N'Thuê giày đấu', 100, 50000),
('DV003', N'Nước uống', 100, 15000),
('DV004', N'Thức ăn', 100, 30000),
('DV005', N'Thuê tất', 100, 5000)
```

Hình 1.8 Nhập dữ liệu vào bảng dịch vụ

- Thêm dữ liệu vào bảng giá tiền thuê sân:

```
insert into GiaTienThueSan values ('GT001', '7:00:00', '9:00:00', 20000),
('GT002', '9:00:00', '11:00:00', 25000),
('GT003', '13:00:00', '15:00:00', 30000),
('GT004', '15:00:00', '17:00:00', 35000),
('GT005', '19:00:00', '21:00:00', 40000)
```

Hình 1.9 Nhập dữ liệu vào bảng giá tiền thuê sân

- Thêm dữ liệu vào bảng thuê sân:

```
set dateformat dmy
insert into ThueSan values ('02032022001', '0369573202', 'SB001', '2/3/2022', '7:00:00', '9:00:00', 40000),
('04032022001', '0357340098', 'SB002', '4/3/2022', '9:00:00', '11:00:00', 50000),
('05032022001', '0357320943', 'SB003', '5/3/2022', '13:00:00', '15:00:00', 60000),
('06032022001', '0369343284', 'SB004', '6/3/2022', '15:00:00', '17:00:00', 70000),
('07032022001', '0369432498', 'SB005', '7/3/2022', '19:00:00', '21:00:00', 80000),
('07032022002', '0369432498', 'SB005', '7/3/2022', '7:00:00', '10:30:00', 90000),
('09032022001', '0357314832', 'SB005', '9/3/2022', '13:00:00', '16:30:00', 112500)
```

Hình 1.10 Nhập dữ liệu vào bảng thuê sân

- Thêm dữ liệu vào bảng chi tiết thuê sân:

```
insert into ChiTietThueSan values ('02032022001', 'DV002', 2, 50000),
('04032022001', 'DV005', 10, 5000),
('05032022001', 'DV001', 5, 40000),
('06032022001', 'DV003', 20, 15000),
('07032022001', 'DV004', 20, 30000),
('09032022001', 'DV002', 5, 50000)
```

Hình 1.11 Nhập dữ liệu vào bảng chi tiết thuê sân

1.2.2 Câu 2:

- Viết hàm phát sinh mã số tự động cho bảng thuê sân theo nguyên tắc: Ngày thuê sân + số thứ tự gồm 3 chữ số tăng dần. Ví dụ: Ngày thuê sân là ngày 20/06/2022

và hiện đã có lần thuê sân thứ 3 (Bảng Thuê sân đang có mã số là: “20062022003” thì mã số tự động tiếp theo sẽ là “20062022004”).

```

go
]create function auto_id(@ngay date)
returns char(20)
as
begin
    declare @t varchar(20) = replace(CONVERT(varchar, @ngay, 112), '/', '')
    declare @temp varchar(20) = substring(@t,7,2) + substring(@t,5,2) + substring(@t,1,4)
    declare @count int = (select count(NgayThue) from ThueSan where NgayThue = @ngay)
    declare @result char(20)
    if @count = 0
        set @result = @temp + '001'
    else
        if @count > 0 and @count < 9
            set @result = @temp + '00' + convert(char, @count + 1)
        else if @count >= 9
            set @result = @temp + '0' + convert(char, @count + 1)
        else if @count >= 99
            set @result = @temp + convert(char, @count + 1)
    return @result
end
go

```

Hình 1.12 Hàm phát sinh mã số tự động

- b. Viết hàm trả về số tiền thuê sân khi biết được giờ vào sân và giờ ra khỏi sân.

```

create function getMoneyFromHour(@startAt time, @endAt time)
returns float
as
begin
    declare @result float
    declare @a time
    declare @b time
    declare @a1 float
    if @startAt < '9:00:00'
    begin
        if @endAt <= '9:00:00'    --7h đến 9h
        begin
            set @a = cast((cast(@endAt as datetime) - cast(@startAt as datetime)) as time)
            set @a1 = datepart(hour, @a) + datepart(minute, @a)/60.0
            set @result = @a1*20000
            return @result
        end
    else
        --7h đến 11h
        begin
            set @a = cast((cast('9:00:00' as datetime) - cast(@startAt as datetime)) as time)
            set @b = cast((cast(@endAt as datetime) - cast('9:00:00' as datetime)) as time)
            declare @c float = datepart(hour, @a) + datepart(minute, @a)/60.0
            declare @c1 float = datepart(hour, @b) + datepart(minute, @b)/60.0
            set @result = @c*20000 + @c1*25000
            return @result
        end
    end
end

```

Hình 1.13 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc

```

else
begin
    if @endAt <= '11:00:00'    -- 9h đến 11h
    begin
        set @a = cast((cast(@endAt as datetime) - cast(@startAt as datetime)) as time)
        set @a1 = datepart(hour, @a) + datepart(minute, @a)/60.0
        set @result = @a1*25000
        return @result
    end
end

```

Hình 1.14 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc

```

if @startAt < '15:00:00'
begin
    if @endAt <= '15:00:00'    -- 13h đến 15h
    begin
        set @a = cast((cast(@endAt as datetime) - cast(@startAt as datetime)) as time)
        set @a1 = datepart(hour, @a) + datepart(minute, @a)/60.0
        set @result = @a1*30000
        return @result
    end
else
    -- 13h đến 17h
    begin
        set @a = cast((cast('15:00:00' as datetime) - cast(@startAt as datetime)) as time)
        set @b = cast((cast(@endAt as datetime) - cast('15:00:00' as datetime)) as time)
        declare @d float = datepart(hour, @a) + datepart(minute, @a)/60.0
        declare @d1 float = datepart(hour, @b) + datepart(minute, @b)/60.0
        set @result = @d*30000 + @d1*35000
        return @result
    end
end
else
begin
    if @endAt <= '17:00:00'    -- 15h đến 17h
    begin
        set @a = cast((cast(@endAt as datetime) - cast(@startAt as datetime)) as time)
        set @a1 = datepart(hour, @a) + datepart(minute, @a)/60.0
        set @result = @a1*35000
        return @result
    end
end
end

```

Hình 1.15 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc

```

if @startAt between '19:00:00' and '21:00:00' and @endAt between '19:00:00' and '21:00:00'
begin
    set @a = cast((cast(@endAt as datetime) - cast(@startAt as datetime)) as time)
    set @a1 = datepart(hour, @a) + datepart(minute, @a)/60.0
    set @result = @a1*40000
end
return @result
end

```

Hình 1.16 Hàm trả về tổng tiền khi biết giờ bắt đầu và giờ kết thúc

1.2.3 Câu 3:

- a. Thêm vào bảng Giá tiền thuê sân có kiểm tra khoá chính, giờ vào phải nhỏ hơn giờ ra.

```
go
create proc addGiaTienThueSan @magt char(10), @startAt time(0), @endAt time(0), @sotien int
as
begin
    if @magt is null
        print N'Khóa chính không được phép rỗng, vui lòng nhập lại'
    else if @magt is not null and exists (select MaGT from GiaTienThueSan where MaGT = @magt)
        print N'Khóa chính đã tồn tại, vui lòng nhập lại'
    else if @startAt > @endAt
        print N'Giờ vào phải nhỏ hơn giờ ra'
    else
        insert into GiaTienThueSan values (@magt, @startAt, @endAt, @sotien)
end
go
```

Hình 1.17 Thủ tục thêm vào bảng giá tiền thuê sân

- b. Cập nhật đơn giá cho dịch vụ thuê áo thêm 10.000

```
]create proc updateDVThueAo
as
]begin
]    update DichVu
    set DonGia += 10000
    where MaDV = 'DV001'
]end
]go
]exec dbo.updateDVThueAo
]drop proc updateDVThueAo
]go
```

Hình 1.18 Thủ tục cập nhật đơn giá dịch vụ thuê áo thêm 10.000

1.2.4 Câu 4:

- Tạo trigger:

```

go
|create trigger cau4
on ChiTietThueSan
for insert, update, delete
as
|begin

```

Hình 1.19 Tạo trigger

- Khi thêm vào bảng chi tiết thuê sân:

```

if (exists (select * from inserted) and not exists (select * from deleted)) -- them
begin
    --- update thue san
    update ThueSan
    set TongTien = dbo.getMoneyFromHour(GioBD, GioKetThuc)
                    + (select sum(ct.Soluong * dv.DonGia) from DichVu dv, ChiTietThueSan ct, inserted
                        where ct.MaDV = dv.MaDV and ct.MaThueSan = inserted.MaThueSan)
    from ThueSan
    join inserted on ThueSan.MaThueSan = inserted.MaThueSan
    -- update chi tiet thue san
    update ChiTietThueSan
    set DonGia = DichVu.DonGia
    from DichVu, inserted
    where ChiTietThueSan.MaDV = inserted.MaDV
    print N'Dã thêm vào chi tiết thuê sân mới'
end

```

Hình 1.20 Trigger thêm vào bảng chi tiết thuê sân

- Khi xóa bảng chi tiết thuê sân:

```

else if (not exists (select * from inserted) and exists (select * from deleted)) -- xoa
begin
    declare @mats char(20) = (select MaThueSan from deleted)
    declare @madv char(10) = (select MaDV from deleted)
    -- update thue san
    update ThueSan
    set TongTien = dbo.getMoneyFromHour(GioBD, GioKetThuc)
                    - (select sum(ct.Soluong * dv.DonGia) from DichVu dv, ChiTietThueSan ct, deleted
                        where ct.MaDV = dv.MaDV and ct.MaThueSan = deleted.MaThueSan)
    from ThueSan
    join deleted on ThueSan.MaThueSan = deleted.MaThueSan
end

```

Hình 1.21 Trigger xóa bảng chi tiết thuê sân

- Khi sửa bảng chi tiết thuê sân:

```

else -- sửa
begin
    declare @mts_new char(20) = (select MaThueSan from inserted)
    declare @madv_new char(10) = (select MaDV from inserted)
    declare @sl_new int = (select SoLuong from inserted)
    declare @mts_old char(20) = (select MaThueSan from deleted)
    declare @madv_old char(10) = (select MaDV from deleted)
    declare @sl_old int = (select SoLuong from deleted)

    if update(MaThueSan) or update(MaDV) or update(SoLuong)
    begin
        update ChiTietThueSan
        set MaThueSan = @mts_new, MaDV = @madv_new, SoLuong = @sl_new
        where MaThueSan = @madv_old and MaDV = @madv_old and SoLuong = @sl_old

        --- update thue san
        update ThueSan
        set TongTien = dbo.getMoneyFromHour(GioBD, GioKetThuc)
            + (select sum(@sl_new * dv.DonGia) from DichVu dv, ChiTietThueSan ct
                where @madv_new = dv.MaDV and ct.MaThueSan = @mts_new)
        from ThueSan
        join inserted on ThueSan.MaThueSan = inserted.MaThueSan

        -- update chi tiet thue san
        update ChiTietThueSan
        set DonGia = DichVu.DonGia
        from DichVu
        where DichVu.MaDV = @madv_new and MaThueSan = @mts_new and ChiTietThueSan.MaDV = @madv_new
    end
end

```

Hình 1.22 Trigger sửa bảng chi tiết thuê sân

PHẦN 2

2.1 Yêu cầu đề bài:

Cho lược đồ quan hệ $R(U, F)$, $U = MNOPQRS$

$F = \{S \rightarrow MR; NS \rightarrow QM; PQ \rightarrow RS; MO \rightarrow NR; N \rightarrow R\}$

- Tìm tất cả các khóa (khóa dự tuyển) của lược đồ quan hệ (0.5 điểm)
- Tìm phủ tối thiểu của tập phụ thuộc hàm (0.5 điểm)
- Xác định dạng chuẩn cao nhất của lược đồ quan hệ (0.5 điểm)
- Nếu lược đồ quan hệ chưa đạt dạng chuẩn BC, đưa lược đồ về dạng chuẩn BC (0.5 điểm)

2.2 Giải quyết yêu cầu:

a) Tập $N = \{O, P\}$

Tập $D = \{R\}$

Tập $L = \{M, N, Q, S\}$

N		L				X	X_{+F}	Superkey
O	P	M	N	Q	S			
1	1	0	0	0	0	OP	OP	
1	1	1	0	0	0	MOP	MNOPR	
1	1	0	1	0	0	NOP	NOPR	
1	1	1	1	0	0	MNOP	MNOPR	
1	1	0	0	1	0	OPQ	MNOPQRS	OPQ
1	1	1	0	1	0	MOPQ	MNOPQRS	MOPQ
1	1	0	1	1	0	NOPQ	MNOPRQS	NOPQ

1	1	1	1	1	0	MNOPQ	MNOPQRS	MNOPQ
1	1	0	0	0	1	OPS	MNOPQRS	OPS
1	1	1	0	0	1	MOPS	MNOPQRS	MOPS
1	1	0	1	0	1	NOPS	MNOPQRS	NOPS
1	1	1	1	0	1	MNOPS	MNOPQRS	MNOPS
1	1	0	0	1	1	OPQS	MNOPQRS	OPQS
1	1	1	0	1	1	MOPQS	MNOPQRS	MOPQS
1	1	0	1	1	1	NOPQS	MNOPQRS	NOPQS
1	1	1	1	1	1	MNOPQS	MNOPQRS	MNOPQS

Bảng 2.1 Tìm tất cả các khóa

Vậy tập khóa là: {OPQ, OPS}

b) Bước 1: Phân rã về phải

$F1 = \{S \rightarrow M; S \rightarrow R; NS \rightarrow Q; NS \rightarrow M; PQ \rightarrow R; PQ \rightarrow S; MO \rightarrow N; MO \rightarrow R; N \rightarrow R\}$

Bước 2: Loại bỏ thuộc tính dư thừa ở về trái

+Xét $NS \rightarrow Q$ có thuộc tính N và S không dư thừa vì

$N^+ = NR$ không chứa Q

$S^+ = SMR$ không chứa Q

+Xét $NS \rightarrow M$ có thuộc tính N dư thừa vì

$N^+ = NR$ không chứa M

$S^+ = SMR$ chứa M

+Xét $PQ \rightarrow R$ có thuộc tính P và Q không dư thừa vì

$P = P$ không chứa R

$Q = Q$ không chứa R

+ Xét $PQ \rightarrow S$ có thuộc tính P và Q không dư thừa vì

$P = P$ không chứa S

$Q = Q$ không chứa S

+ Xét $MO \rightarrow N$ có thuộc tính M và O không dư thừa vì

$M = M$ không chứa N

$O = O$ không chứa N

+ Xét $MO \rightarrow R$ có thuộc tính M và O không dư thừa vì

$M = M$ không chứa R

$O = O$ không chứa R

$\Rightarrow F2 = \{S \rightarrow M; S \rightarrow R; NS \rightarrow Q; PQ \rightarrow R; PQ \rightarrow S; MO \rightarrow N; MO \rightarrow R; N \rightarrow R\}$

Bước 3: Loại bỏ phụ thuộc hàm dư thừa

+ Xét $S \rightarrow M$ là không dư thừa vì

$S^+_{(F2 - \{S \rightarrow M\})} = SR$ không chứa M

+ Xét $S \rightarrow R$ là không dư thừa vì

$S^+_{(F2 - \{S \rightarrow R\})} = SM$

+ Xét $NS \rightarrow Q$ là không dư thừa vì

$NS^+_{(F2 - \{NS \rightarrow Q\})} = NSM$ không chứa Q

+ Xét $PQ \rightarrow R$ là dư thừa vì

$PQ^+_{(F2 - \{PQ \rightarrow R\})} = PQSMR$ chứa R

+ Xét $PQ \rightarrow S$ là không dư thừa vì

$PQ^+_{(F2 - \{PQ \rightarrow S\})} = PQR$ không chứa S

+ Xét $MO \rightarrow N$ là không dư thừa vì

$MO^+_{(F2 - \{MO \rightarrow N\})} = MOR$ không chứa N

+ Xét $MO \rightarrow R$ là dư thừa vì

$MO^+_{(F2 - \{MO \rightarrow R\})} = MONR$ chứa R

+ Xét $N \rightarrow R$ là không dư thừa vì

$$N^+_{(F2- \{N \rightarrow R\})} = N \text{ không chứa } R$$

Vậy $F3 = \{S \rightarrow M; S \rightarrow R; NS \rightarrow Q; PQ \rightarrow S; MO \rightarrow N; N \rightarrow R\}$ là phủ tối thiểu

c) Thuộc tính khóa : $\{OPQS\}$

Thuộc tính không khóa : $\{MNR\}$

Xét 2NF :

Xét $OPQS \rightarrow M$ là phụ thuộc hàm không đầy đủ vì :

$$O^+_F = O \text{ không chứa } M$$

$$P^+_F = P \text{ không chứa } M$$

$$Q^+_F = Q \text{ không chứa } M$$

$$S^+_F = SMR \text{ chứa } M$$

➔ Thuộc tính không khóa M không phụ thuộc đầy đủ vào thuộc tính khóa nên lược đồ không thuộc dạng chuẩn 2NF

Xét 3NF:

Đặt S lưu trữ những phụ thuộc hàm mà không có vế trái là siêu khóa

$$\text{Vậy } S = \{S \rightarrow MR; NS \rightarrow QM; PQ \rightarrow RS; MO \rightarrow NR; N \rightarrow R\}$$

Ta kiểm tra điều kiện : để lược đồ quan hệ đạt 3NF thì với mọi phụ thuộc hàm $X \rightarrow A$ trong lược đồ quan hệ thì A (với A khác X) là thuộc tính nguyên tố.

- Kiểm tra: $S \rightarrow MR$
 - M không là thuộc tính nguyên tố
 - R không là thuộc tính nguyên tố
- Kiểm tra: $NS \rightarrow QM$
 - M không là thuộc tính nguyên tố
- Kiểm tra: $PQ \rightarrow RS$
 - R không là thuộc tính nguyên tố
- Kiểm tra: $MO \rightarrow NR$:
 - N không là thuộc tính nguyên tố
 - R không là thuộc tính nguyên tố

- Kiểm tra: $N \rightarrow R$:
 - R không phải là thuộc tính nguyên tố

Kết luận: lược đồ đạt dạng chuẩn 1NF

d) $F = \{S \rightarrow MR; NS \rightarrow QM; PQ \rightarrow RS; MO \rightarrow NR; N \rightarrow R\}$

Chia lược đồ quan hệ thành:

- $R1 = \{MRS\}$ với $F1 = \{S \rightarrow MR\}$
- $R2 = \{NOPQS\}$ với $F2 = (PQ \rightarrow S, NS \rightarrow Q, OS \rightarrow N)$

Phân rã $R2 = \{NOPQS\}$ thành:

- $R21 = \{PQS\}$ với $F21 = \{PQ \rightarrow S\}$
- $R22 = \{NOPQ\}$ với $F22 = \{OPQ \rightarrow N\}$

Vậy dạng chuẩn hóa BC của lược đồ là:

$R1 = \{MRS\}, F1 = \{S \rightarrow MR\}$

$R21 = \{PQS\}, F21 = \{PQ \rightarrow S\}$

$R22 = \{NOPQ\}, F22 = \{OPQ \rightarrow N\}$

PHẦN 3

3.1 Yêu cầu đề bài:

Sinh viên tự viết đặc tả về một nghiệp vụ quản lý nào đó. Đặc tả phải thể hiện được có các mối quan hệ 1-1, 1-n, n-n, kế thừa, thực thể mạnh, thực thể yếu, ... Sau đó, sinh viên vẽ mô hình ERD cho đặc tả đó, chuyển mô hình ERD sang mô hình quan hệ.

- a. Viết đặc tả (1.5 điểm)
- b. Vẽ ERD (0.75 điểm)
- c. Chuyển ERD sang mô hình quan hệ (0.75 điểm)

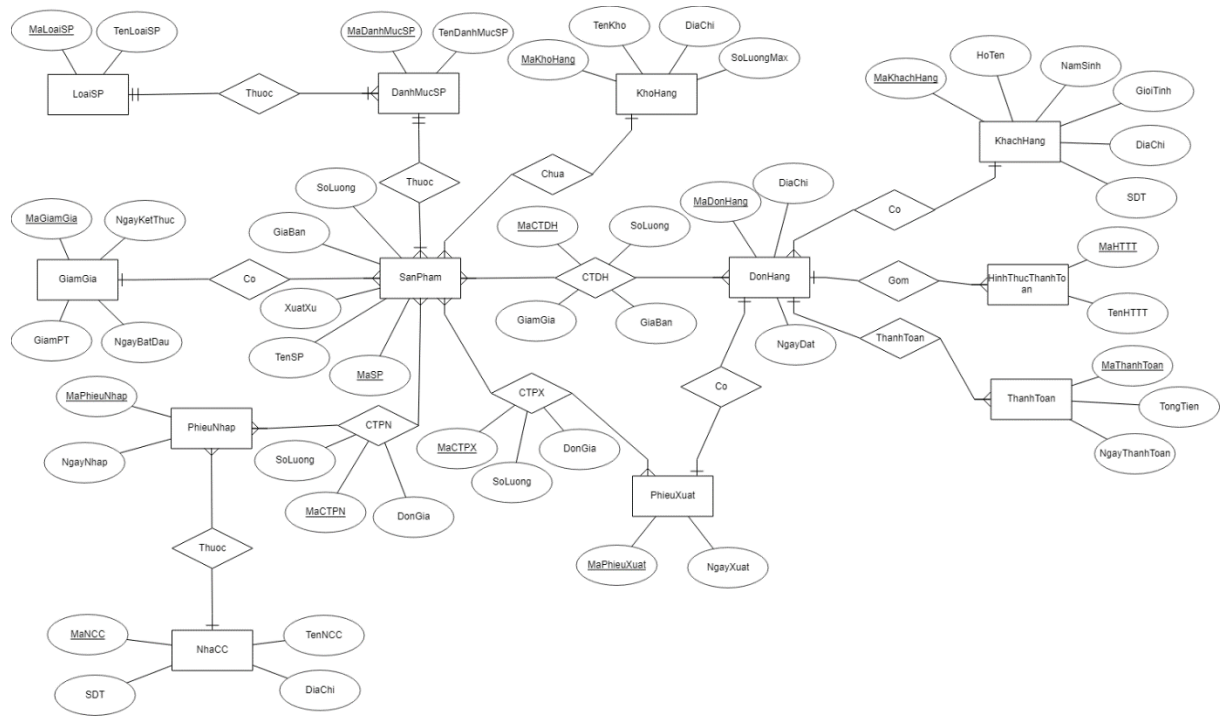
3.2 Giải quyết yêu cầu:

Một cửa hàng điện tử buôn bán nhiều loại sản phẩm khác nhau từ điện thoại thông minh cho đến máy tính bảng hay các loại laptop và PC. Do đó cửa hàng cần xây dựng cơ sở dữ liệu để quản lý công việc của cửa hàng. Mỗi sản phẩm có các thông tin như mã sản phẩm, tên sản phẩm, xuất xứ, giá bán, số lượng. Các sản phẩm sẽ thuộc một danh mục sản phẩm khác nhau. Thông về một danh mục sản phẩm bao gồm mã danh mục sản phẩm và tên của danh mục. Bên cạnh đó, các danh mục sản phẩm sẽ thuộc một loại sản phẩm. Loại sản phẩm sẽ bao gồm các thông tin như mã loại sản phẩm và tên loại sản phẩm. Các sản phẩm sẽ được nhập từ những nhà cung cấp khác nhau. Mỗi nhà cung cấp sẽ có một mã nhà cung cấp riêng để phân biệt với các nhà cung cấp khác, cũng như các thông tin về số điện thoại, tên nhà cung cấp và địa chỉ. Khi các nhà cung cấp cung cấp sản phẩm cho cửa hàng sẽ xuất ra phiếu nhập và các nhà cung cấp có thể nhập cùng một mặt hàng nhiều lần. Phiếu nhập sẽ bao gồm các thuộc tính như mã phiếu nhập và ngày nhập. Và các thông tin chi tiết của phiếu nhập như mã chi tiết phiếu nhập, số lượng và đơn giá. Ngoài ra, cửa hàng còn có các chương trình giảm giá và các sản phẩm sẽ có một

loại giảm giá khác nhau. Thông tin về chương trình giảm giá bao gồm mã giảm giá, ngày bắt đầu giảm giá, ngày kết thúc giảm giá và sẽ giảm bao nhiêu phần trăm.

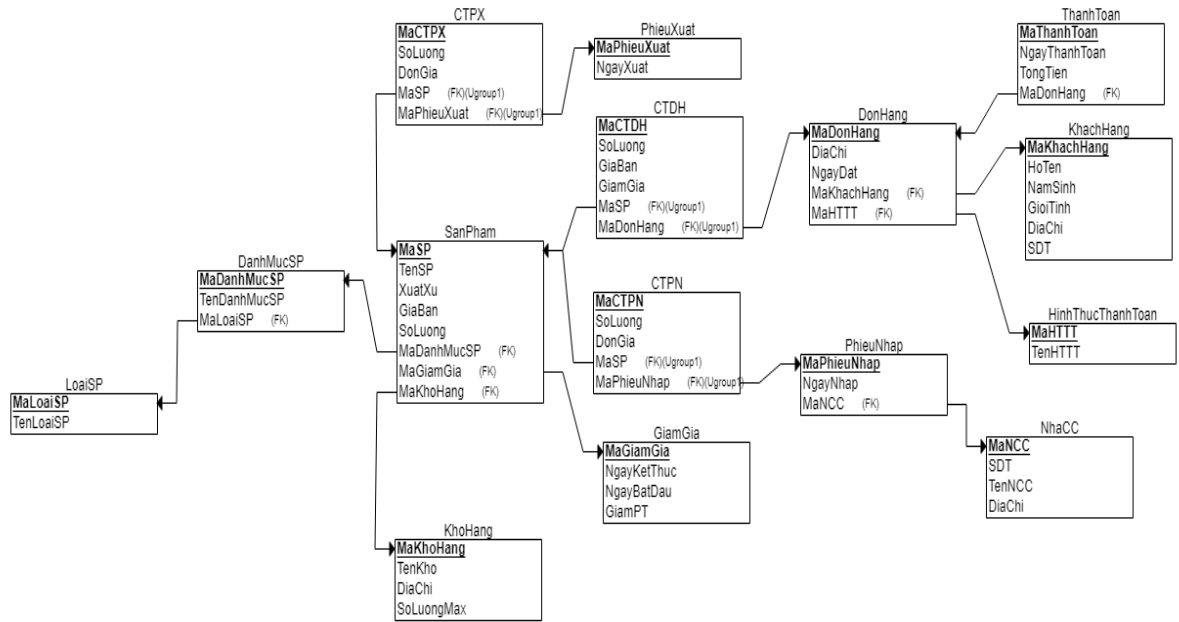
Các khách hàng sẽ đến mua sản phẩm tại cửa hàng. Mỗi khách hàng sẽ có các thông tin như mã khách hàng, họ tên, năm sinh, giới tính, địa chỉ và số điện thoại. Mỗi lần mua hàng tại cửa hàng, khách hàng sẽ nhận được một đơn hàng. Thông tin về đơn hàng gồm mã đơn hàng, địa chỉ giao hàng và ngày đặt. Sau khi đã chọn được sản phẩm muốn mua, khách hàng có thể chọn mình thức thanh toán mà mình muốn. Mỗi đơn hàng mà khách hàng đã mua có thể có nhiều hình thức thanh toán khác nhau. Hình thức thanh toán sẽ bao gồm các thuộc tính như mã hình thức thanh toán và tên hình thức thanh toán. Sau khi đã chọn được hình thức thanh toán, khách hàng sẽ được yêu cầu thanh toán đơn hàng. Các thông tin về thanh toán như mã thanh toán, ngày thanh toán tổng tiền. Ở đây khách hàng có thể thanh toán nhiều lần. Cuối cùng, sản phẩm sau khi được mua sẽ được xuất ra một phiếu xuất có các thuộc tính như mã phiếu xuất và ngày xuất. Chi tiết phiếu xuất có các thông tin như mã chi tiết phiếu xuất, số lượng và đơn giá.

Sơ đồ ERD



Hình 3.1 Sơ đồ ERD

Lược đồ quan hệ



Hình 3.2 Lược đồ quan hệ

PHẦN 4

4.1 Yêu cầu đề bài:

Sử dụng ngôn ngữ C/Java/Python để cài đặt các chức năng sau:

1. Cho lược đồ cơ sở dữ liệu (LĐCSDL), cho biết các khoá của lược đồ LĐCSDL (0.5 điểm)
2. Cho LĐCSDL, cho biết phủ tối thiểu của LĐCSDL (0.75 điểm)
3. Cho LĐCSDL, cho biết dạng chuẩn cao nhất của LĐCSDL. Giải thích tại sao. (0.75 điểm)

4.2 Giải quyết yêu cầu:

- Đầu vào:

```
LDQH = {'M', 'N', 'O', 'P', 'R', 'Q', 'S'}
PTH = [({'S'}, {'R', 'M'}),
        ({'N', 'S'}, {'Q', 'M'}),
        ({'P', 'Q'}, {'R', 'S'}),
        ({'O', 'M'}, {'R', 'N'}),
        ({'N'}, {'R'})]
```

Hình 4.1 Bộ giá trị đầu vào

- Hàm tìm khóa của lược đồ quan hệ:


```

from itertools import combinations

def KTBaoDong(attr, PTH):
    closure = set(attr)
    last = set()
    while closure != last:
        last = closure.copy()
        for dep in PTH:
            if dep[0].issubset(closure):
                closure.update(dep[1])
    return closure

def timKhoa(LDQH, PTH):
    result = []
    for i in range(1, len(LDQH)+1):
        for keys in combinations(LDQH, i):
            if KTBaoDong(keys, PTH) == LDQH:
                k = set(keys)
                if not any([x.issubset(k) for x in result]):
                    result.append(k)
    return result

```

Hình 4.2 Hàm tìm khóa của lược đồ quan hệ

→ Kết quả:

```

Khóa của LDQH: [{'O', 'P', 'Q'}, {'O', 'P', 'S'}]

```

Hình 4.3 Kết quả sau khi thực hiện hàm tìm khóa

- Hàm tìm phủ tối thiểu:

```

def unique_list(l_2d):
    check = []
    return [x for x in l_2d if x not in check and not check.append(x)]

def phuToiThieu(PTH):
    G = []
    #phan ra cac thuoc tinh ben ve phai
    for dep in PTH:
        if len(dep[1]) == 1:
            G.append((set(dep[0]), set(dep[1])))
            continue
        for e in dep[1]:
            G.append((set(dep[0]), set(e)))
    #loai bo thuoc tinh du thua ve trai
    for dep in G:
        save = []
        if len(dep[0]) > 1:
            for e in dep[0]:
                temp = G.copy()
                tempSet = [(set(e), dep[1])]
                if tempSet in G:
                    temp.remove(tempSet)
                if dep[1].issubset(KTBaoDong(e, temp)) is False:
                    save.append(e)
        if len(save) != len(dep[0]):

```

Hình 4.4 Hàm tìm phủ tối thiểu

```

        for x in save:
            dep[0].discard(x)
    G = unique_list(G)
    # loại bỏ phụ thuộc hàm dư thừa
    GTemp = G.copy()
    for dep in G:
        attr = tuple(dep[0])
        temp = G.copy()
        temp.remove(dep)
        if dep[1].issubset(KTBaoDong(attr, temp)):
            GTemp.remove(dep)
    G = GTemp
    return G

```

Hình 4.5 hàm tìm phủ tối thiểu

→ Kết quả:

```

Phủ tối thiểu của lược đồ quan hệ:
S  ->  R
S  ->  M
SN ->  Q
PQ ->  S
OM ->  N
N  ->  R

```

Hình 4.6 Kết quả khi thực hiện hàm tìm phủ tối thiểu

- Kiểm tra dạng chuẩn cao nhất:

```

def ktDangChuan2NF(PTH, keys, notKeys):
    # X -> A
    print('----- Kiem tra 2NF-----')
    for dep in PTH:
        print('Kiem tra', ''.join(dep[0]), ' -> ', ''.join(dep[1]))
        if dep[0] in keys: #T.hop A thuoc X hoac X la sieu khoa
            print(''.join(dep[0]), 'la sieu khoa')
            continue
        if len(dep[0] & notKeys):
            print('co thuoc tinh khon khoa', ''.join(dep[0] & notKeys), 'thuoc', ''.join(dep[1]))
            continue
        for attr in notKeys:
            if attr in dep[1]: # A khong la thuoc tinh khoa trong R
                print(attr, 'Khong la thuoc tinh nguyen to')
                print('LDQH khong dat 2NF')
                return False
    return True

```

Hình 4.7 Kiểm tra đạt chuẩn 2NF

```

def ktDangChuan3NF(PTH, keys, notKeys):
    # X -> Y. Dat dang chuan 3 khi :
    # X la sieu khoa
    # Y la thuoc tinh nguyen to
    print('----- Kiem tra 3NF-----')
    for dep in PTH:
        print('Kiem tra', ''.join(dep[0]), ' -> ', ''.join(dep[1]))
        if dep[0] in keys: # X la sieu khoa
            print(dep[0], 'la sieu khoa')
            continue
        for attr in notKeys:
            if attr in dep[1]: # Y la thuoc tinh nguyen to
                print(attr, 'Khong phai la thuoc tinh nguyen to')
                print('LDQH khong dat 3NF')
                return False
    return True

```

Hình 4.8 Kiểm tra đạt chuẩn 3NF

```
def ktDangChuanBCNF(PTH, keys):
    print('----- Kiem tra BCNF-----')
    for dep in PTH:
        print('Kiem tra', ''.join(dep[0]), ' -> ', ''.join(dep[1]))
        if dep[0] not in keys:
            print(''.join(dep[0]), 'khong la sieu khoa')
            print('LDQH khong dat BCNF')
            return False
    return True
```

Hình 4.9 Kiểm tra đạt chuẩn BCNF

→ Kết quả:

```
Thuoc tinh nguyen to: {'O', 'S', 'P', 'Q'}
Thuoc tinh khong khoa: {'N', 'R', 'M'}
----- Kiem tra 2NF-----
Kiem tra S -> RM
R Khong la thuoc tinh nguyen to
LDQH khong dat 2NF
----- Kiem tra 3NF-----
Kiem tra S -> RM
R Khong phai la thuoc tinh nguyen to
LDQH khong dat 3NF
----- Kiem tra BCNF-----
Kiem tra S -> RM
S khong la sieu khoa
LDQH khong dat BCNF
-----Ket Luan-----
Vay luoc do quan he dat chuan 1NF
```

Hình 4.10 Kết quả sau khi thực hiện tìm dạng chuẩn

TÀI LIỆU THAM KHẢO

Tiếng Việt

Tiếng Anh

PHỤ LỤC