TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN MÔN HỆ CƠ SỞ DỮ LIỆU**

*Người hướng dẫn*: **GV. DOÃN XUÂN THANH**

*Người thực hiện*: **TRẦN THỊ VẸN**

Lớp **: 21050301**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO BÀI TẬP LỚN MÔN HỆ CƠ SỞ DỮ LIỆU**

*Người hướng dẫn*: **GV. DOÃN XUÂN THANH**

*Người thực hiện*: **TRẦN THỊ VẸN**

Lớp **: 21050301**

Khoá  **: 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2022**

LỜI CẢM ƠN

Xin gửi lời cảm ơn đến thầy vì đã cung cấp cho em đủ kiến thức trong quá trình học tập nhằm thực hiện được báo cáo này một cách tốt nhất.

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của thầy Doãn Xuân Thanh. 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 luận văn 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 luận văn 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)*

TÓM TẮT

Trong báo cáo này em sẽ dùng kiến thức mình đã học được trong học kỳ vừa rồi nhằm giải quyết các vẫn đề liên quan đến môn học hệ cơ sở dữ liệu.

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc103686801)

[TÓM TẮT iii](#_Toc103686802)

[MỤC LỤC 1](#_Toc103686803)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 2](#_Toc103686804)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 3](#_Toc103686805)

[PHẦN 1: 4](#_Toc103686806)

[1.1 phần câu hỏi: 4](#_Toc103686807)

[1.1 phần trả lời: 5](#_Toc103686808)

[PHẦN 2: 22](#_Toc103686809)

[2.1 phần câu hỏi: 22](#_Toc103686810)

[2.2 phần trả lời: 23](#_Toc103686811)

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

Hình 1.1 Mô hình ERD 5

Hình 1.2 Mô hình quan hệ 6

Hình 1.3 Kết quả kiểm thử procedure với thông tin hợp lệ 18

Hình 1.4 Kết quả kiểm thử procedure với thông tin không hợp lệ 19

Hình 1.5 Dữ liệu của bảng PHIEUNHAPHANG\_MATHANG 20

Hình 1.6 Kết quả kiểm thử của function tính tiền của 1 hóa đơn 21

Hình 1.7 Kết quả kiểm thử của trigger cập nhật số lượng và đơn giá khi nhập hàng 22

PHẦN 1:

* 1. phần câu hỏi:

Cho đặc tả như sau:

Một siêu thị cần xây dựng một hệ thống quản lý hàng hóa đang bày bán tại siêu thị. Siêu thị hiện đang có những hoạt động như sau:

Siêu thị nhập hàng từ nhà cung cấp. Các nhà cung cấp cung cấp nhiều mặt hàng với số lượng khác nhau. Mỗi mặt hàng cần ghi nhận thông tin mã hàng, tên hàng, đơn vị tính, số lượng, đơn giá. Mỗi mặt hàng chỉ do một nhà sản xuất làm ra. Những thông tin về nhà sản xuất là: mã nhà sản xuất, tên nhà sản xuất, quốc tịch. Mỗi lần nhập hàng cần lưu lại phiếu nhập hàng gồm các thông tin như: mã phiếu nhập, các mặt hàng nhập, nhập từ nhà cung cấp nào, số lượng, đơn giá nhập, ngày nhập. Các thông tin về nhà cung cấp là mã nhà cung cấp, họ tên, địa chỉ. Siêu thị có rất nhiều quầy hàng bán các mặt hàng, tuy nhiên mỗi mặt hàng chỉ được bày bán ở một quầy nhất định nào đó. Thông tin về quầy hàng bao gồm số quầy, tên quầy, vị trí. Các khách hàng sẽ đến siêu thị mua hàng. Mỗi lần khách mua hàng, hệ thống sẽ in ra phiếu bán hàng gồm các thông tin mã phiếu bán, khách hàng, các mặt hàng, số lượng, đơn giá bán. Thông tin của khách hàng bao gồm mã khách hàng, tên khách hàng, địa chỉ.

1. Vẽ mô hình ERD (1 điểm)

2. Chuyển đổi mô hình ERD sau sang mô hình quan hệ (1 điểm)

3. Sử dựng ngôn ngữ SQL để thực hiện các yêu cầu sau cho mô hình quan hệ trên: (1 điểm)

a. Tạo Database

b. Tạo các bảng dữ liệu

c. Thêm ít nhất 10 dòng dữ liệu cho mỗi bảng.

4. Tạo procedure cho một trong các nghiệp vụ sau: (1 điểm)

a. Tạo mới một phiếu nhập

b. Thêm dữ liệu vào hoá đơn

c. Thêm dữ liệu vào mặt hàng

5. Tạo function cho một trong các yêu cầu sau: (1 điểm)

a. Cho biết tổng tiền ứng với một phiếu nhập nào đó

b. Cho biết tổng tiền ứng với một hoá đơn nào đó

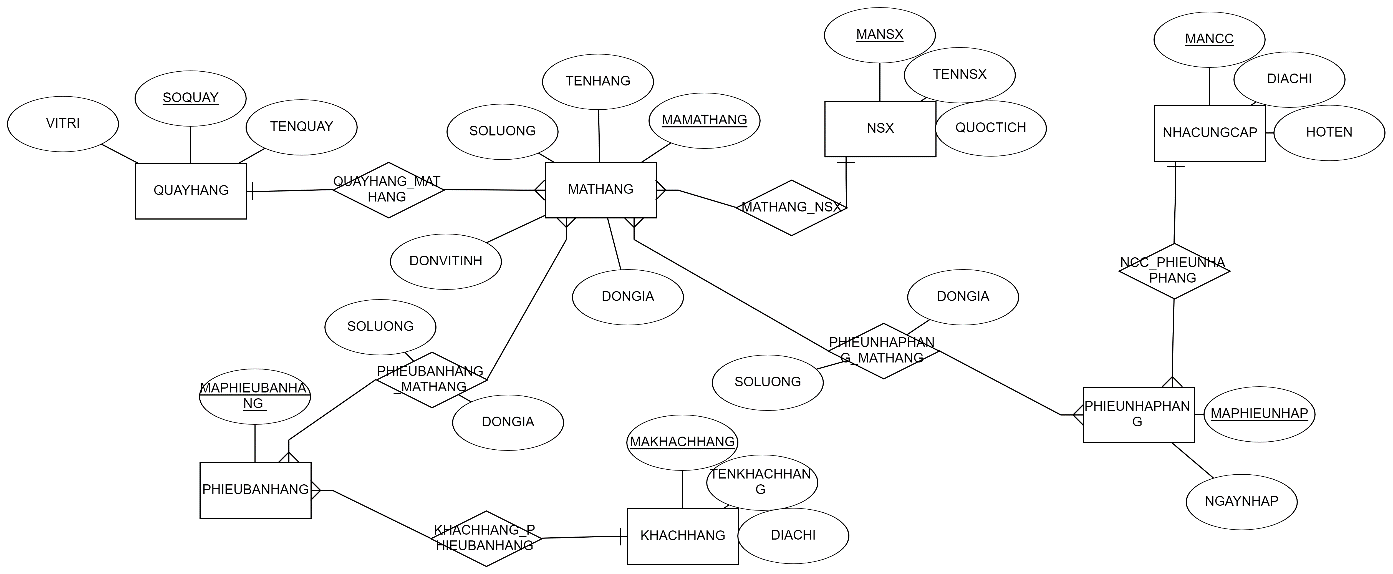
6. Tạo trigger cho một trong các yêu cầu sau: (1 điểm)

a. Cập nhật số lượng và đơn giá cho mặt hàng mỗi khi nhập hàng. Ví dụ, ở phiếu nhập PN01 nhập mặt hàng PEPSI với số lượng là 10 và đơn giá 10.000 thì cộng thêm số lượng PEPSI trong bảng Mặt hàng và cập nhật đơn giá mới là 10.000

b. Cập nhật số lượng cho mặt hàng mỗi khi bán hàng. Ví dụ, ở phiếu bán hàng PB01 bán mặt hàng PEPSI với số lượng là 10 thì trừ đi số lượng PEPSI này trong bảng Mặt hàng.

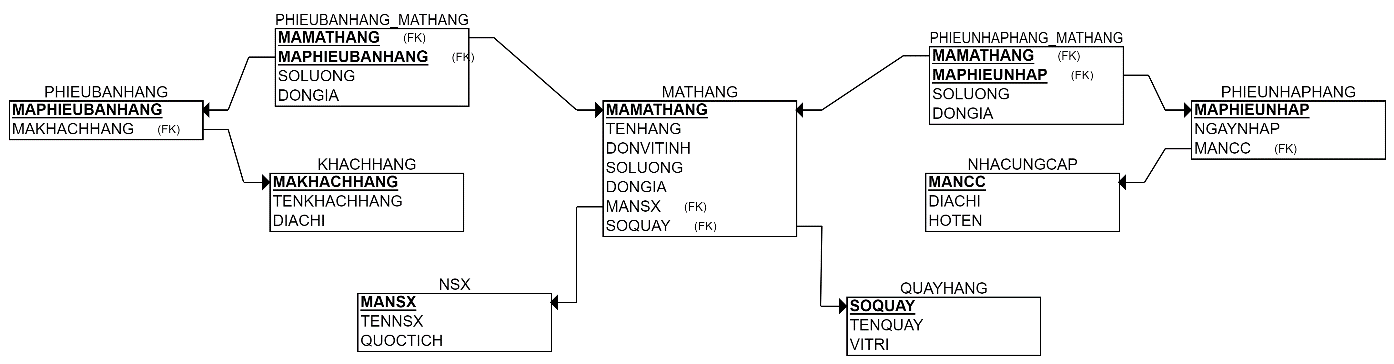
* 1. phần trả lời:

Câu 1: Vẽ mô hình ERD:



Hình 1.1 Mô hình ERD

Câu 2: Chuyển mô hình ERD sang mô hình quan hệ:



Hình 1.2 Mô hình quan hệ

Câu 3: Sử dụng ngôn ngữ SQL để thực hiện các yêu cầu sau cho mô hình quan hệ trên

tạo cơ sở dữ liệu:

*CREATE* *DATABASE* QUANLYHANGHOA

*GO*

*USE* QUANLYHANGHOA

*GO*

tạo các bảng:

*CREATE* *TABLE* NHACUNGCAP

(

  MANCC *VARCHAR*(*10*) *NOT* *NULL*,

  DIACHI *VARCHAR*(*50*) *NOT* *NULL*,

  HOTEN *VARCHAR*(*50*) *NOT* *NULL*,

*PRIMARY* *KEY* (MANCC)

);

*CREATE* *TABLE* NSX

(

  MANSX *VARCHAR*(*10*) *NOT* *NULL*,

  TENNSX *VARCHAR*(*50*) *NOT* *NULL*,

  QUOCTICH *VARCHAR*(*50*) *NOT* *NULL*,

*PRIMARY* *KEY* (MANSX)

);

*CREATE* *TABLE* QUAYHANG

(

  SOQUAY *VARCHAR*(*10*) *NOT* *NULL*,

  TENQUAY *VARCHAR*(*50*) *NOT* *NULL*,

  VITRI *VARCHAR*(*50*) *NOT* *NULL*,

*PRIMARY* *KEY* (SOQUAY)

);

*CREATE* *TABLE* KHACHHANG

(

  MAKHACHHANG *VARCHAR*(*10*) *NOT* *NULL*,

  TENKHACHHANG *VARCHAR*(*50*) *NOT* *NULL*,

  DIACHI *VARCHAR*(*50*) *NOT* *NULL*,

*PRIMARY* *KEY* (MAKHACHHANG)

);

*CREATE* *TABLE* PHIEUNHAPHANG

(

  MAPHIEUNHAP *VARCHAR*(*10*) *NOT* *NULL*,

  NGAYNHAP *DATE* *NOT* *NULL*,

  MANCC *VARCHAR*(*10*) *NOT* *NULL*,

*PRIMARY* *KEY* (MAPHIEUNHAP),

*FOREIGN KEY* (MANCC) *REFERENCES* NHACUNGCAP(MANCC)

);

*CREATE* *TABLE* PHIEUBANHANG

(

  MAPHIEUBANHANG *VARCHAR*(*10*) *NOT* *NULL*,

  MAKHACHHANG *VARCHAR*(*10*) *NOT* *NULL*,

*PRIMARY* *KEY* (MAPHIEUBANHANG),

*FOREIGN KEY* (MAKHACHHANG) *REFERENCES* KHACHHANG(MAKHACHHANG)

);

*CREATE* *TABLE* MATHANG

(

  MAMATHANG *VARCHAR*(*10*) *NOT* *NULL*,

  TENHANG *VARCHAR*(*50*) *NOT* *NULL*,

  DONVITINH *VARCHAR*(*20*) *NOT* *NULL*,

  SOLUONG *INT* *NOT* *NULL*,

  DONGIA *FLOAT* *NOT* *NULL*,

  MANSX *VARCHAR*(*10*) *NOT* *NULL*,

  SOQUAY *VARCHAR*(*10*) *NOT* *NULL*,

*PRIMARY* *KEY* (MAMATHANG),

*FOREIGN KEY* (MANSX) *REFERENCES* NSX(MANSX),

*FOREIGN KEY* (SOQUAY) *REFERENCES* QUAYHANG(SOQUAY)

);

*CREATE* *TABLE* PHIEUBANHANG\_MATHANG

(

  SOLUONG *INT* *NOT* *NULL*,

  DONGIA *FLOAT* *NOT* *NULL*,

  MAMATHANG *VARCHAR*(*10*) *NOT* *NULL*,

  MAPHIEUBANHANG *VARCHAR*(*10*) *NOT* *NULL*,

*PRIMARY* *KEY* (MAMATHANG, MAPHIEUBANHANG),

*FOREIGN KEY* (MAMATHANG) *REFERENCES* MATHANG(MAMATHANG),

*FOREIGN KEY* (MAPHIEUBANHANG) *REFERENCES* PHIEUBANHANG(MAPHIEUBANHANG)

);

*CREATE* *TABLE* PHIEUNHAPHANG\_MATHANG

(

  SOLUONG *INT* *NOT* *NULL*,

  DONGIA *FLOAT* *NOT* *NULL*,

  MAMATHANG *VARCHAR*(*10*) *NOT* *NULL*,

  MAPHIEUNHAP *VARCHAR*(*10*) *NOT* *NULL*,

*PRIMARY* *KEY* (MAMATHANG, MAPHIEUNHAP),

*FOREIGN KEY* (MAMATHANG) *REFERENCES* MATHANG(MAMATHANG),

*FOREIGN KEY* (MAPHIEUNHAP) *REFERENCES* PHIEUNHAPHANG(MAPHIEUNHAP)

);

Thêm ít nhất 10 dòng dữ liệu cho mỗi bảng:

*INSERT* *INTO* NHACUNGCAP (MANCC, DIACHI, HOTEN)

*VALUES*

    ('NCC01', 'Nha Trang', 'NCCA'),

    ('NCC02', 'Da Nang', 'NCCB'),

    ('NCC03', 'Hue', 'NCCC'),

    ('NCC04', 'Quang Nam', 'NCCD'),

    ('NCC05', 'Quang Ngai', 'NCCE'),

    ('NCC06', 'Quang Ninh', 'NCCF'),

    ('NCC07', 'Quang Tri', 'NCCG'),

    ('NCC08', 'Tay Ninh', 'NCCH'),

    ('NCC09', 'Tay Ninh', 'NCCI'),

    ('NCC10', 'Tay Ninh', 'NCCJ')

*GO*

*INSERT* *INTO* NSX (MANSX, TENNSX, QUOCTICH)

*VALUES*

    ('NSX01', 'NSXA', 'Viet Nam'),

    ('NSX02', 'NSXB', 'Viet Nam'),

    ('NSX03', 'NSXC', 'Viet Nam'),

    ('NSX04', 'NSXD', 'Viet Nam'),

    ('NSX05', 'NSXE', 'Viet Nam'),

    ('NSX06', 'NSXF', 'Viet Nam'),

    ('NSX07', 'NSXG', 'Viet Nam'),

    ('NSX08', 'NSXH', 'Viet Nam'),

    ('NSX09', 'NSXI', 'Viet Nam'),

    ('NSX10', 'NSXJ', 'Viet Nam')

*GO*

*INSERT* *INTO* QUAYHANG (SOQUAY, TENQUAY, VITRI)

*VALUES*

    ('QUYH01', 'QUAYHANGA', 'VT1'),

    ('QUYH02', 'QUAYHANGB', 'VT2'),

    ('QUYH03', 'QUAYHANGC', 'VT3'),

    ('QUYH04', 'QUAYHANGD', 'VT4'),

    ('QUYH05', 'QUAYHANGE', 'VT5'),

    ('QUYH06', 'QUAYHANGF', 'VT6'),

    ('QUYH07', 'QUAYHANGG', 'VT7'),

    ('QUYH08', 'QUAYHANGH', 'VT8'),

    ('QUYH09', 'QUAYHANGI', 'VT9'),

    ('QUYH10', 'QUAYHANGJ', 'VT10')

*GO*

*INSERT* *INTO* KHACHHANG (MAKHACHHANG, TENKHACHHANG, DIACHI)

*VALUES*

    ('KH01', 'Thomas', 'Nha Trang'),

    ('KH02', 'James', 'Da Nang'),

    ('KH03', 'John', 'Hue'),

    ('KH04', 'Robert', 'Quang Nam'),

    ('KH05', 'Michael', 'Quang Ngai'),

    ('KH06', 'William', 'Quang Ninh'),

    ('KH07', 'David', 'Quang Tri'),

    ('KH08', 'Richard', 'Tay Ninh'),

    ('KH09', 'Joseph', 'Tay Ninh'),

    ('KH10', 'Thomas', 'Tay Ninh')

*GO*

*INSERT* *INTO* PHIEUNHAPHANG (MAPHIEUNHAP, NGAYNHAP, MANCC)

*VALUES*

    ('PN01', '01/01/2019', 'NCC01'),

    ('PN02', '02/01/2019', 'NCC02'),

    ('PN03', '03/01/2019', 'NCC03'),

    ('PN04', '04/01/2019', 'NCC04'),

    ('PN05', '05/01/2019', 'NCC05'),

    ('PN06', '06/01/2019', 'NCC06'),

    ('PN07', '07/01/2019', 'NCC07'),

    ('PN08', '08/01/2019', 'NCC08'),

    ('PN09', '09/01/2019', 'NCC09'),

    ('PN10', '10/01/2019', 'NCC10')

*GO*

*INSERT* *INTO* PHIEUBANHANG (MAPHIEUBANHANG, MAKHACHHANG)

*VALUES*

    ('PB01', 'KH01'),

    ('PB02', 'KH02'),

    ('PB03', 'KH03'),

    ('PB04', 'KH04'),

    ('PB05', 'KH05'),

    ('PB06', 'KH06'),

    ('PB07', 'KH07'),

    ('PB08', 'KH08'),

    ('PB09', 'KH09'),

    ('PB10', 'KH10')

*GO*

*INSERT* *INTO* MATHANG (MAMATHANG, TENHANG, DONVITINH, SOLUONG, DONGIA, MANSX, SOQUAY)

*VALUES*

    ('MH01', 'HANGHOA1', 'THUNG', *20*, *2*.*2*, 'NSX01', 'QUYH01'),

    ('MH02', 'HANGHOA2', 'THUNG', *20*, *2*.*2*, 'NSX02', 'QUYH02'),

    ('MH03', 'HANGHOA3', 'THUNG', *20*, *2*.*2*, 'NSX03', 'QUYH03'),

    ('MH04', 'HANGHOA4', 'THUNG', *20*, *2*.*2*, 'NSX04', 'QUYH04'),

    ('MH05', 'HANGHOA5', 'THUNG', *20*, *2*.*2*, 'NSX05', 'QUYH05'),

    ('MH06', 'HANGHOA6', 'THUNG', *20*, *2*.*2*, 'NSX06', 'QUYH06'),

    ('MH07', 'HANGHOA7', 'THUNG', *20*, *2*.*2*, 'NSX07', 'QUYH07'),

    ('MH08', 'HANGHOA8', 'THUNG', *20*, *2*.*2*, 'NSX08', 'QUYH08'),

    ('MH09', 'HANGHOA9', 'THUNG', *20*, *2*.*2*, 'NSX09', 'QUYH09'),

    ('MH10', 'HANGHOA10', 'THUNG', *20*, *2*.*2*, 'NSX10', 'QUYH10')

*GO*

*INSERT* *INTO* PHIEUNHAPHANG\_MATHANG (SOLUONG, DONGIA, MAMATHANG, MAPHIEUNHAP)

*VALUES*

    (*10*, *2*.*2*, 'MH01', 'PN01'),

    (*10*, *2*.*2*, 'MH02', 'PN02'),

    (*10*, *2*.*2*, 'MH03', 'PN03'),

    (*10*, *2*.*2*, 'MH04', 'PN04'),

    (*10*, *2*.*2*, 'MH05', 'PN05'),

    (*10*, *2*.*2*, 'MH06', 'PN06'),

    (*10*, *2*.*2*, 'MH07', 'PN07'),

    (*10*, *2*.*2*, 'MH08', 'PN08'),

    (*10*, *2*.*2*, 'MH09', 'PN09'),

    (*10*, *2*.*2*, 'MH10', 'PN10')

*GO*

*INSERT* *INTO* PHIEUBANHANG\_MATHANG (SOLUONG, DONGIA, MAMATHANG, MAPHIEUBANHANG)

*VALUES*

    (*10*, *2*.*2*, 'MH01', 'PB01'),

    (*10*, *2*.*2*, 'MH02', 'PB02'),

    (*10*, *2*.*2*, 'MH03', 'PB01'),

    (*10*, *2*.*2*, 'MH04', 'PB04'),

    (*10*, *2*.*2*, 'MH05', 'PB01'),

    (*10*, *2*.*2*, 'MH06', 'PB06'),

    (*10*, *2*.*2*, 'MH07', 'PB01'),

    (*10*, *2*.*2*, 'MH08', 'PB08'),

    (*10*, *2*.*2*, 'MH09', 'PB01'),

    (*10*, *2*.*2*, 'MH10', 'PB10')

*GO*

Câu 4: tạo procedure cho một trong các nghiệp vụ sau:

Trong câu này em chọn làm câu c thêm dữ liệu vào bảng mặt hàng

*CREATE* *PROC* [DBO].[SP\_THEM\_MATHANG]

    @MAMATHANG *NVARCHAR*(*50*),

    @TENHANG *NVARCHAR*(*50*),

    @DONVITINH *NVARCHAR*(*50*),

    @SOLUONG *INT*,

    @DONGIA *INT*,

    @MANSX *NVARCHAR*(*50*),

    @SOQUAY *NVARCHAR*(*50*)

*AS*

*BEGIN*

*IF* (*SELECT* COUNT(\*) *FROM* MATHANG *WHERE* MAMATHANG = @MAMATHANG) <> *0*

*BEGIN*

*RAISERROR* ('Mã mặt hàng đã tồn tại', *16*, *1*)

*RETURN*

*END*

*ELSE* *IF* (@SOLUONG < *0*)

*BEGIN*

*RAISERROR* ('Số lượng không hợp lệ', *16*, *1*)

*RETURN*

*END*

*ELSE* *IF* (@DONGIA < *0*)

*BEGIN*

*RAISERROR* ('Đơn giá không hợp lệ', *16*, *1*)

*RETURN*

*END*

*ELSE* *IF* (*SELECT* COUNT(\*) *FROM* NSX *WHERE* MANSX = @MANSX) <> *1*

*BEGIN*

*RAISERROR* ('Mã nhà sản xuất không tồn tại', *16*, *1*)

*RETURN*

*END*

*ELSE* *IF* (*SELECT* COUNT(\*) *FROM* QUAYHANG *WHERE* SOQUAY = @SOQUAY) <> *1*

*BEGIN*

*RAISERROR* ('Số quầy hàng không tồn tại', *16*, *1*)

*RETURN*

*END*

*ELSE*

*BEGIN*

*INSERT* *INTO* MATHANG (MAMATHANG, TENHANG, DONVITINH, SOLUONG, DONGIA, MANSX, SOQUAY)

*VALUES* (@MAMATHANG, @TENHANG, @DONVITINH, @SOLUONG, @DONGIA, @MANSX, @SOQUAY)

*END*

*END*

*GO*

Tiến hành kiểm thử procedure:

Đoạn code SQL phục vụ cho việc kiểm thử:

*EXEC* SP\_THEM\_MATHANG 'MH01', 'HANGHOA11', 'THUNG', *20*, *2*.*2*, 'NSX11', 'QUYH11'

*EXEC* SP\_THEM\_MATHANG 'MH11', 'HANGHOA11', 'THUNG', -*20*, *2*.*2*, 'NSX11', 'QUYH11'

*EXEC* SP\_THEM\_MATHANG 'MH12', 'HANGHOA11', 'THUNG', *20*, -*2*.*2*, 'NSX11', 'QUYH11'

*EXEC* SP\_THEM\_MATHANG 'MH13', 'HANGHOA11', 'THUNG', *20*, *2*.*2*, 'NSX11', 'QUYH11'

*EXEC* SP\_THEM\_MATHANG 'MH14', 'HANGHOA11', 'THUNG', *20*, *2*.*2*, 'NSX01', 'QUYH11'

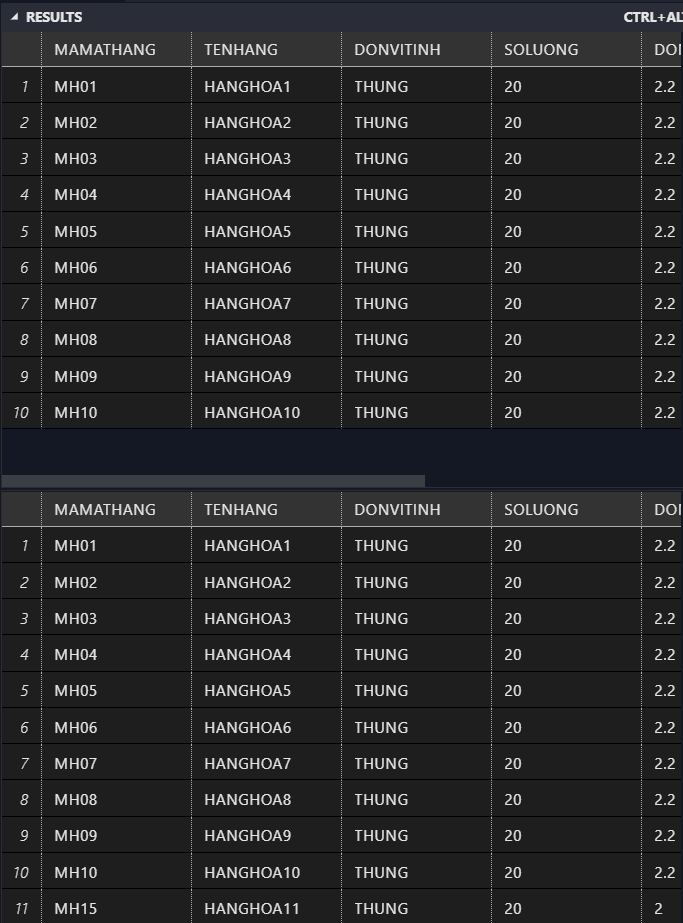
*SELECT* \* *FROM* MATHANG

*EXEC* SP\_THEM\_MATHANG 'MH15', 'HANGHOA11', 'THUNG', *20*, *2*.*2*, 'NSX01', 'QUYH01'

*SELECT* \* *FROM* MATHANG

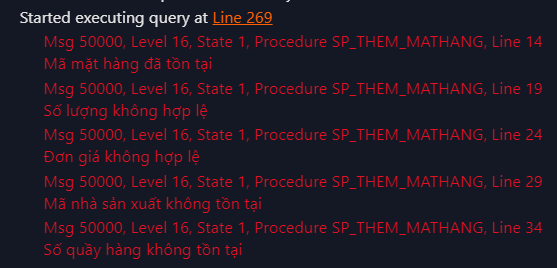
Kết quả sau khi tiến hành kiểm thử:

Trường hợp các thông tin nhập vào proc là hợp lệ



Hình 1.3 Kết quả kiểm thử procedure với thông tin hợp lệ

Trương hợp các thông tin đưa vào proc là không hợp lệ:



Hình 1.4 Kết quả kiểm thử procedure với thông tin không hợp lệ

Câu 5: Tạo function cho một trong các yêu cầu sau:

Trong câu này em chọn là câu b cho biết tổng tiền của một hóa đơn nào đó:

*CREATE* *FUNCTION* [DBO].[TINHTIEN] (

    @MAPHIEUBAN *NVARCHAR*(*10*)

)

*RETURNS* *INT*

*AS*

*BEGIN*

*RETURN* (

*SELECT* SUM(SOLUONG \* DONGIA)

*FROM* PHIEUBANHANG

*RIGHT JOIN* PHIEUBANHANG\_MATHANG

*ON* *PHIEUBANHANG*.*MAPHIEUBANHANG* = *PHIEUBANHANG\_MATHANG*.*MAPHIEUBANHANG*

*WHERE* *PHIEUBANHANG*.*MAPHIEUBANHANG* = @MAPHIEUBAN

*GROUP BY* SOLUONG, DONGIA, *PHIEUBANHANG*.*MAPHIEUBANHANG*

    )

*END*

*GO*

Tiền hành kiểm thử function:

Đoạn code phục vụ việc kiểm thử:

*SELECT* [DBO].[TINHTIEN]('PB01') *AS* TONGTIEN

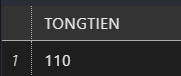
*GO*

Dữ liệu về số lượng và giá tiền hiện tại của các phiếu nhập hàng:



Hình 1.5 Dữ liệu của bảng PHIEUNHAPHANG\_MATHANG

Kết quả sau khi tiến hành kiểm thử:



Hình 1.6 Kết quả kiểm thử của function tính tiền của 1 hóa đơn

Câu 6: Tạo trigger cho một trong các yêu cầu sau:

Trong câu này em chọn làm câu a Cập nhật số lượng và đơn giá cho mặt hàng mỗi khi nhập hàng. Ví dụ, ở phiếu nhập PN01 nhập mặt hàng PEPSI với số lượng là 10 và đơn giá 10.000 thì cộng thêm số lượng PEPSI trong bảng Mặt hàng và cập nhật đơn giá mới là 10.000

*CREATE* TRIGGER [DBO].[CAPNHATMATHANG]

*ON* [DBO].[PHIEUNHAPHANG\_MATHANG]

*AFTER* *INSERT*

*AS*

*BEGIN*

*UPDATE* MATHANG

*SET* SOLUONG = SOLUONG + (*SELECT* SOLUONG *FROM* INSERTED),

        DONGIA = DONGIA + (*SELECT* DONGIA *FROM* INSERTED)

*WHERE* *MATHANG*.*MAMATHANG* = (*SELECT* MAMATHANG *FROM* INSERTED)

*END*

Tiến hành kiểm thử trigger;

Đoạn mã SQL phục vụ cho việt kiểm thử:

*SELECT* MAMATHANG, SOLUONG

*FROM* MATHANG

*WHERE* MAMATHANG = 'MH01'

*INSERT* *INTO* PHIEUNHAPHANG\_MATHANG (SOLUONG, DONGIA, MAMATHANG, MAPHIEUNHAP)

*VALUES* (*10*, *3*.*3*, 'MH01', 'PN02')

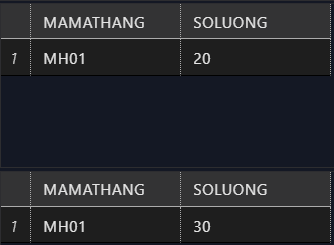
*SELECT* MAMATHANG, SOLUONG

*FROM* MATHANG

*WHERE* MAMATHANG = 'MH01'

Kết quả kiếm thử:

Trước và sau khi trigger chạy:



Hình 1.7 Kết quả kiểm thử của trigger cập nhật số lượng và đơn giá khi nhập hàng

PHẦN 2:

* 1. phần câu hỏi:

Sinh viên sử dụng ngôn ngữ Java hoặc Python để cài đặt chức năng sau:

1. Input: File .txt chứa các thực thể và mối quan hệ giữa các thực thể của mô hình ERD. Output: File .txt chứa các bảng dữ liệu và mối quan hệ giữa các bảng. Định dạng của file Input.txt do sinh viên tự thiết lập. (2 điểm)

2. Input: File .txt chứa lược đồ CSDL bao gồm các bảng, các thuộc tính và các phụ thuộc hàm. Output: File .txt chứa bao đóng của tập thuộc tính X (Tập thuộc tính X do người dùng truyền vào); các khoá của lược đồ quan hệ. (2 điểm)

* 1. phần trả lời:

Câu 1:

*# open input.txt, output.txt*

rfile = open("input.txt", "r")

wfile = open("output.txt", "w")

*for* i in rfile.readlines():

*if* '-' in i:

*# split line into table name and relational data*

        temp = i.strip('*\n*').split('-')

*# create junction table if relation is many to many, linked junction table to 2 related table and write to output.txt*

*if* temp[*1*] == 'MM':

            wfile.write(temp[*0*] + '\_' + temp[*2*] + ' -- ' +

                        temp[*0*] + '*\n*')

            wfile.write(temp[*0*] + '\_' + temp[*2*] + ' -- ' +

                        temp[*2*] + '*\n*')

*# linked related tables and write to output.txt*

*elif* temp[*1*] == 'OM' or temp[*1*] == 'MO' or temp[*1*] == 'OO':

            wfile.write(temp[*0*] + ' -- ' + temp[*2*] + '*\n*')

*else*:

        wfile.write(i)

nội dung file input.txt của câu 1:

QUAYHANG-OO-MATHANG

PHIEUBANHANG-MM-MATHANG

KHACHHANG-OM-PHIEUBANHANG

PHIEUNHAPHANG-MM-MATHANG

NSX-OM-MATHANG

NCC-OM-PHIEUNHAPHANG

TEST

Giải thích nội dung file input, file này được chia thành từng dòng với mỗi dòng là hai thực thể trong mô hình ERD cùng với mối quan hệ giữa hai thực thể đó nằm giữa hai thực thể và được cách nhau bởi ký tự ‘-‘

Nội dung file output.txt của câu 1:

QUAYHANG -- MATHANG

PHIEUBANHANG\_MATHANG -- PHIEUBANHANG

PHIEUBANHANG\_MATHANG -- MATHANG

KHACHHANG -- PHIEUBANHANG

PHIEUNHAPHANG\_MATHANG -- PHIEUNHAPHANG

PHIEUNHAPHANG\_MATHANG -- MATHANG

NSX -- MATHANG

NCC -- PHIEUNHAPHANG

TEST

Giải thích nội dung file output.txt, file này chứa dữ liệu chia thành từng dòng với từng dòng bao gồm các bảng dữ liệu và mối liên hệ giữa chúng(nếu có)

Câu 2:

rfile = open("input.txt", "r")

wfile = open("output.txt", "w")

tables = {}

*# read input.txt and add data to tables dictionary*

*for* line in rfile.readlines():

    fd = line[line.index(')') + *2*:].strip('*\n*').split(',')

    fd\_temp = []

*for* j in fd:

        src, des = j.split('->')

*if* '+' not in src:

*if* '+' not in des:

                fd\_temp.extend([[src, des]])

*else*:

                fd\_temp.extend([[src, des.split('+')]])

*else*:

*if* '+' not in des:

                fd\_temp.extend([[src.split('+'), des]])

*else*:

                fd\_temp.extend([[src.split('+'), des.split('+')]])

    tables[line[*0*:line.index('(')]] = {

        'attrs': line.strip('*\n*')[line.index('(')+*1*:line.index(')')].split(','),

        'fds': fd\_temp

    }

*# function to find closure from list of attributes*

*def* find\_closure(*table\_name*, *X*):

*while*(*True*):

        isChanged = *False*

*for* i in tables[table\_name]['fds']:

*if* type(i[*0*]) is not list and type(i[*1*]) is not list and i[*0*] in X and i[*1*] not in X:

                X.append(i[*1*])

                isChanged = *True*

*if* type(i[*0*]) is list and type(i[*1*]) is not list and all(x in X *for* x in i[*0*]) and i[*1*] not in X:

                X.append(i[*1*])

                isChanged = *True*

*if* type(i[*0*]) is not list and type(i[*1*]) is list and i[*0*] in X and any(x not in X *for* x in i[*1*]):

                X.extend(i[*1*])

                isChanged = *True*

*if* type(i[*0*]) is list and type(i[*1*]) is list and all(x in X *for* x in i[*0*]) and any(x not in X *for* x in i[*1*]):

                X.extend(i[*1*])

                isChanged = *True*

*if* isChanged == *False*:

*break*

*return* list(set(X))

X = ['B', 'G', 'H']

wfile.write('X = ' + str(X) + '*\n*')

wfile.write('closure of X = ' +

            str(find\_closure('TEST', ['B', 'G', 'H'])) + '*\n*')

*# function to find keys of tables*

*def* find\_key():

    keys = {}

*for* table\_name in tables.keys():

        res = []

        temp1 = tables[table\_name]['attrs'].copy()

*for* i in tables[table\_name]['attrs']:

            temp = temp1.copy()

            temp.remove(i)

*if* all(i in find\_closure(table\_name, temp) *for* i in tables[table\_name]['attrs']):

                temp1.remove(i)

                res.append(i)

        keys[table\_name] = [i *for* i in tables[table\_name]['attrs'] *if* i not in res]

*return* keys

wfile.write('keys = ' + str(find\_key()) + '*\n*')

nội dung file input.txt của câu 2:

TEST(A,B,C,D,E,G,H) B->A,D+A->C+E,D->H,G+H->C,A+C->D

TEST2(A,B,C,D,E) B->A,D+A->C+E,D->H,A+C->D

Giải thích nội dung file input.txt, nội dung file này là một lược đồ cơ sở dữ liệu, được chia thành từng dòng với mỗi dòng một lược đồ quan hệ đi kèm với các phụ thuộc hàm tương ứng

Nội dung file output.txt của câu 2:

X = ['B', 'G', 'H']

table = TEST

closure of X = ['C', 'E', 'D', 'G', 'A', 'H', 'B']

keys = {'TEST': ['B', 'G', 'H'], 'TEST2': ['B', 'D']}

giải thích nội dung của file output.txt, nội dung file này bao gồm X là tập thuộc tính mà người dùng nhập vào để tìm bao đóng, table là bảng chứa tập thuộc tính cần tìm bao đóng, key là một từ điển chứa tên và các khóa của từng lược đồ quan hệ bên trong lược đồ cơ sở dữ liệu.

**TÀI LIỆU THAM KHẢO**

**Tiếng Anh**