**Câu 1:** Cho cơ sở dữ liệu như sau: (3 điểm)

KHACHHANG(MAKH, TENKH, DIACHI, DIENTHOAI)

DONDATHANG(MADDH, NGAYDH, MAKH, TINHTRANG)

TINHTRANG: ĐÃ GIAO, CHƯA GIAO, ĐÃ HỦY

CHITIETDATHANG(MADDH, MASP, SL)

SANPHAM(MASP, TENSP, DVT, SLCON, DONGIA)

a) Tạo View cho biết các khách hàng có đơn đặt hàng trong năm 2023.(1.5 điểm)

b) Tạo Store Procedure (dạng Transaction) cho công việc: Lập một đơn hàng (thêm đơn hàng mới và chi tiết của đơn hàng đó). (1.5 điểm)

a/

create view kh2023 as

select A.MAKH, A.TENKH, A.DIACHI, A.DIENTHOAI

from KHACHHANG A, DONDATHANG B

where A.MAKH = B.MAKH

and DATEPART(YEAR, NGAYDH) = ‘2023’

b/ Tạo Store Procedure (dạng Transaction) cho công việc: Lập một đơn hàng (thêm đơn hàng mới và chi tiết

của đơn hàng đó).

create proc sp\_Lapdonhang

@MADDH VARCHAR(10),

@NGAYDH DATE,

@MAKH VARCHAR(10),

@TINHTRANG VARCHAR(20),

**@CHITIETDONHANG AS TABLE (MASP VARCHAR(10), SL INT) READONLY**

AS

BEGIN TRANSACTION

BEGIN TRY

INSERT INTO DONDATHANG(MADDH, NGAYDH, MAKH, TINHTRANG)

VALUES (@MADDH, @NGAYDH, @MAKH, @TINHTRANG)

INSERT INTO CHITIETDATHANG(MADDH, MASP, SL)

**VALUES (@MADDH, SELECT MASP, SL FROM CHITIETDONHANG)**

-- Cập nhật số lượng còn lại trong bảng SANPHAM

UPDATE SP

SET SLCON = SP.SLCON - CT.SL

FROM SANPHAM SP

JOIN @ChiTietDonHang CT ON SP.MASP = CT.MASP

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF (@@error <> 0)

BEGIN

ROLLBACK TRANSACTION

RETURN

END

END CATCH

**Câu 2:** Cho CSDL như sau: (3 điểm)

TAIKHOAN (MATK, SODU, SODUCO, SODUNO, MAKH)

KHACHANG (MAKH, TENKH, ĐCKH, ĐTKH)

GIAODICH (MAGD, NGAYGD, TKNO, TKCO, SOTIEN)

Trong đó: SODU cho biết giá trị của tài khoản, được tính từ SODUCO và SODUNO. Mỗi khi có một giao dịch diễn ra (gửi tiền, rút tiền hoặc chuyển tiền) thì **có 1 dòng dữ liệu thêm vào bảng GIAODICH.** **TKNO hoặc TKCO có thể mang giá trị NULL.**

Hãy viết các giao tác sau:

a) Gửi tiền

b) Rút tiền

c) Chuyển tiền

**1. Bảng TAIKHOAN**

| **Thuộc tính** | **Ý nghĩa** |
| --- | --- |
| MATK | Mã tài khoản, là khóa chính duy nhất để xác định từng tài khoản. |
| SODU | Số dư hiện tại của tài khoản, tính từ số dư có (SODUCO), số dư nợ (SODUNO), và giao dịch. |
| SODUCO | Số dư có, là số tiền khách hàng có sẵn trong tài khoản từ trước. |
| SODUNO | Số dư nợ, là số tiền khách hàng đang nợ (nếu có). |
| MAKH | Mã khách hàng, liên kết với bảng KHACHHANG để xác định chủ sở hữu tài khoản. |

**2. Bảng KHACHHANG**

| **Thuộc tính** | **Ý nghĩa** |
| --- | --- |
| MAKH | Mã khách hàng, là khóa chính duy nhất để xác định mỗi khách hàng. |
| TENKH | Tên khách hàng. |
| ĐCKH | Địa chỉ khách hàng, dùng để liên hệ hoặc xác minh thông tin khách hàng. |
| ĐTKH | Số điện thoại khách hàng, để liên hệ trực tiếp hoặc xác nhận giao dịch. |

**3. Bảng GIAODICH**

| **Thuộc tính** | **Ý nghĩa** |
| --- | --- |
| MAGD | Mã giao dịch, là khóa chính duy nhất để xác định từng giao dịch. |
| NGAYGD | Ngày thực hiện giao dịch, dùng để theo dõi lịch sử giao dịch. |
| TKNO | Tài khoản ghi nợ, là tài khoản bị trừ tiền trong giao dịch. |
| TKCO | Tài khoản ghi có, là tài khoản được cộng tiền trong giao dịch. |
| SOTIEN | Số tiền giao dịch, là số tiền thực hiện trong giao dịch (rút, gửi, hoặc chuyển tiền). |

**-- a) Gửi tiền**

CREATE PROCEDURE GuiTien (

@MATK VARCHAR(20),

@SOTIEN FLOAT

)

AS

BEGIN TRANSACTION

BEGIN TRY

-- Cập nhật số dư tài khoản

UPDATE TAIKHOAN

SET SODU = SODU + @SOTIEN, SODUCO = SODUCO + @SOTIEN

WHERE MATK = @MATK

-- Thêm giao dịch vào bảng GIAODICH

INSERT INTO GIAODICH (NGAYGD, TKCO, SOTIEN)

VALUES (GETDATE(), @MATK, @SOTIEN)

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF (@@error <> 0)

begin

ROLLBACK TRANSACTION

RETURN

end

END CATCH

**-- b) Rút tiền**

CREATE PROCEDURE RutTien (

@MATK VARCHAR(10),

@SOTIEN FLOAT

)

AS

BEGIN TRANSACTION

BEGIN TRY

-- Kiểm tra số dư trước khi rút

IF (SELECT SODU FROM TAIKHOAN WHERE MATK = @MATK) >= @SOTIEN

BEGIN

-- Cập nhật số dư tài khoản

UPDATE TAIKHOAN

SET SODU = SODU - @SOTIEN, SODUNO = SODUNO + @SOTIEN

WHERE MATK = @MATK

-- Thêm giao dịch vào bảng GIAODICH

INSERT INTO GIAODICH (NGAYGD, TKNO, SOTIEN)

VALUES (GETDATE(), @MATK, @SOTIEN)

COMMIT TRANSACTION

END

ELSE

BEGIN

-- Nếu số dư không đủ, rollback transaction và thông báo lỗi

ROLLBACK TRANSACTION;

THROW 50001, 'Số dư không đủ để thực hiện giao dịch', 1;

END

END TRY

BEGIN CATCH

IF (@@error <> 0)

Begin

ROLLBACK TRANSACTION

RETURN

end

END CATCH

**-- c) Chuyển tiền**

CREATE PROCEDURE ChuyenTien (

@MATK\_NO VARCHAR(10),

@MATK\_CO VARCHAR(10),

@SOTIEN DECIMAL(18,2)

)

AS

BEGIN TRANSACTION

BEGIN TRY

-- Kiểm tra số dư tài khoản nguồn

IF (SELECT SODU FROM TAIKHOAN WHERE MATK = @MATK\_NO) >= @SOTIEN

BEGIN

-- Cập nhật số dư tài khoản nguồn

UPDATE TAIKHOAN

SET SODU = SODU - @SOTIEN, SODUNO = SODUNO + @SOTIEN

WHERE MATK = @MATK\_NO

-- Cập nhật số dư tài khoản đích

UPDATE TAIKHOAN

SET SODU = SODU + @SOTIEN, SODUCO = SODUCO + @SOTIEN

WHERE MATK = @MATK\_CO

-- Thêm giao dịch vào bảng GIAODICH

INSERT INTO GIAODICH (NGAYGD, TKNO, TKCO, SOTIEN)

VALUES (GETDATE(), @MATK\_NO, @MATK\_CO, @SOTIEN)

COMMIT TRANSACTION

END

ELSE

BEGIN

-- Nếu số dư không đủ, rollback transaction và thông báo lỗi

ROLLBACK TRANSACTION;

THROW 50001, 'Số dư không đủ để thực hiện giao dịch', 1;

END

END TRY

BEGIN CATCH

IF (@@error <> 0)

begin

ROLLBACK TRANSACTION

RETURN

end

END CATCH

Cho lược đồ CSDL **“ Quản lý sản phẩm tiếp thị ”** như sau:

NHANVIEN (**MANV**, TENNV) SANPHAM(**TENSP**, DVT, DONGIA) TIEPTHI(**MANV, TENSP**) HOADON(**SOHD, MANV**, NGAY, KHUVUC) CTHD(**SOHD, MANV, TENSP**, SOLUONG)

***(Sinh viên sử dụng MS SQL SERVER tạo Database ở trên, nhập dữ liệu mẫu để Test phần yêu cầu bên dưới)***

**YEÂU CAÀU:**

**PHẦN 1: (Dùng ngôn ngữ DML để trả lời các câu hỏi sau) (3 đ)**

Câu 1) Tên các nhân viên bán được tất cả các sản phẩm mà họ được phân công tiếp thị

Câu 2) Tên sản phẩm của từng nhân viên bán được tổng lượng nhiều nhất (tên nhân viên, tên

sản phẩm, tổng số lượng)

Câu 3) Tên sản phẩm và tên nhân viên được phân công tiếp thị sản phẩm đó nhưng nhân viên

này chưa bán được sản phẩm đó (tên nhân viên, tên sản phẩm)

**PHẦN 2: (STORED PROCEDURE) (3 đ)**

Câu 1) Tạo 1 thủ tục nội với tên **Sp\_Min** cho biết tên sản phẩm có giá thấp nhất.

Câu 2) Tạo 1 thủ tục nội với tên **TienTheoNgay** có tham số truyền vào là 1 ngày và tham số

ra là Tổng tiền bán được trong ngày đó.

Câu 3) Tạo 1 thủ tục nội với tên **Spham\_of\_Nvien** liệt kê tên các sản phẩm (TENSP) đã

được phân công cho nhân viên có mã số X (X là tham số được truyền vào)

**PHẦN 3: (TRIGGER) (4 đ)** Tạo các Trigger sau**:**

Câu 1) Mỗi nhân viên chỉ được tiếp thị tối đa 5 sản phẩm. (R1)

Câu 2) Khu vực ghi trên hóa đơn chỉ nhận các giá trị “Q1”, “Q3”, “Q10”. (R2)

Câu 3) Giá trị TENSP trong bảng TIEPTHI phải thuộc tập hợp các giá trị TENSP trong bảng

SANPHAM. (R3)

Câu 4) DONGIA của sản phẩm phải là số dương. (R4)

Dưới đây là các đoạn mã SQL tương ứng với các yêu cầu trong bài.

**PHẦN 1: DML**

**Câu 1: Tên các nhân viên bán được tất cả các sản phẩm mà họ được phân công tiếp thị**

SELECT NV.TENNV

FROM NHANVIEN NV

WHERE NOT EXISTS (

SELECT TT.TENSP

FROM TIEPTHI TT

WHERE TT.MANV = NV.MANV

AND NOT EXISTS (

SELECT 1

FROM CTHD

WHERE CTHD.MANV = TT.MANV

AND CTHD.TENSP = TT.TENSP

)

);

**Câu 2: Tên sản phẩm của từng nhân viên bán được tổng lượng nhiều nhất**

WITH TotalSales AS (

SELECT MANV, TENSP, SUM(SOLUONG) AS TotalQuantity

FROM CTHD

GROUP BY MANV, TENSP

), MaxSales AS (

SELECT MANV, MAX(TotalQuantity) AS MaxQuantity

FROM TotalSales

GROUP BY MANV

)

SELECT NV.TENNV, TS.TENSP, TS.TotalQuantity

FROM TotalSales TS

JOIN MaxSales MS ON TS.MANV = MS.MANV AND TS.TotalQuantity = MS.MaxQuantity

JOIN NHANVIEN NV ON TS.MANV = NV.MANV;

**Câu 3: Tên sản phẩm và tên nhân viên chưa bán được sản phẩm họ được phân công tiếp thị**

SELECT NV.TENNV, TT.TENSP

FROM TIEPTHI TT

JOIN NHANVIEN NV ON TT.MANV = NV.MANV

WHERE NOT EXISTS (

SELECT 1

FROM CTHD

WHERE CTHD.MANV = TT.MANV

AND CTHD.TENSP = TT.TENSP

);

**PHẦN 2: STORED PROCEDURE**

**Câu 1: Sản phẩm có giá thấp nhất**

CREATE PROCEDURE Sp\_Min

AS

BEGIN

SELECT TOP 1 TENSP

FROM SANPHAM

ORDER BY DONGIA ASC;

END;

**Câu 2: Tổng tiền bán được trong ngày**

CREATE PROCEDURE TienTheoNgay

@Ngay DATE,

@TongTien MONEY OUTPUT

AS

BEGIN

SELECT @TongTien = SUM(SP.DONGIA \* CTHD.SOLUONG)

FROM CTHD

JOIN SANPHAM SP ON CTHD.TENSP = SP.TENSP

WHERE CTHD.SOHD IN (

SELECT SOHD FROM HOADON WHERE NGAY = @Ngay

);

END;

**Câu 3: Liệt kê sản phẩm của nhân viên**

CREATE PROCEDURE Spham\_of\_Nvien

@Manv INT

AS

BEGIN

SELECT TT.TENSP

FROM TIEPTHI TT

WHERE TT.MANV = @Manv;

END;

**PHẦN 3: TRIGGER**

**Câu 1: Giới hạn mỗi nhân viên tiếp thị tối đa 5 sản phẩm**

CREATE TRIGGER trg\_LimitProducts

ON TIEPTHI

AFTER INSERT

AS

BEGIN

IF EXISTS (

SELECT MANV

FROM TIEPTHI

GROUP BY MANV

HAVING COUNT(TENSP) > 5

)

BEGIN

RAISERROR ('Một nhân viên chỉ được tiếp thị tối đa 5 sản phẩm.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**Câu 2: Kiểm tra khu vực trong hóa đơn**

CREATE TRIGGER trg\_ValidateKhuVuc

ON HOADON

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted

WHERE KHUVUC NOT IN ('Q1', 'Q3', 'Q10')

)

BEGIN

RAISERROR ('Khu vực trên hóa đơn chỉ nhận giá trị "Q1", "Q3", "Q10".', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**Câu 3: Kiểm tra giá trị TENSP trong TIEPTHI**

CREATE TRIGGER trg\_ValidateTensp

ON TIEPTHI

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted i

WHERE NOT EXISTS (

SELECT 1

FROM SANPHAM

WHERE SANPHAM.TENSP = i.TENSP

)

)

BEGIN

RAISERROR ('TENSP trong TIEPTHI phải thuộc tập giá trị trong SANPHAM.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

**Câu 4: DONGIA phải là số dương**

CREATE TRIGGER trg\_ValidateDongia

ON SANPHAM

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted

WHERE DONGIA <= 0

)

BEGIN

RAISERROR ('DONGIA phải là số dương.', 16, 1);

ROLLBACK TRANSACTION;

END

END;