TRƯỜNG ĐẠI HỌC CẦN THƠ

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Mã Học Phần：CT205-02

SV Thực Hiện ：Nguyễn Thị Nhật Thiên Lan

MSSV：B2103427

Giảng Viên Hướng Dẫn：TS. Nguyễn Thái Nghe

Học Kì II, Năm Học: 2023-2024

Hệ Quản Trị Cơ Sở Dữ Liệu

Báo Cáo

****

**MỤC LỤC**

[Phần I. Bài tập 1 Hệ Thống Quản Lý Ô Tô 1](#_Toc23076)

[1. Mô tả 1](#_Toc24916)

[2. Sơ đồ CDM 1](#_Toc27848)

[3. CSDL về xe ô tô như bảng sau: 1](#_Toc19857)

[3.1 Tạo các bảng để lưu các thông tin trên. Lưu ý tất cả các ràng buộc (khóa chính, khóa ngoại, check, not null, default), biểu thức chính quy có thể có. 2](#_Toc12088)

[3.2 Viết thủ tục cho phép nhập vào 1 xe mới 7](#_Toc5611)

[3.3 Gọi thủ tục để Nhập liệu như bảng trên 8](#_Toc14278)

[3.4 Viết thủ tục cho phép cập nhật giá niêm yết của 1 xe nào đó. 8](#_Toc9611)

[3.5 Viết hàm trả về giá của 1 xe nào đó 10](#_Toc8876)

[3.6 Viết hàm trả về loại xe, hãng xe có giá cao nhất 10](#_Toc22187)

[3.7 Viết thủ tục để xóa 1 xe nào đó 11](#_Toc31870)

[3.8 Viết hàm trả về giá lăn bánh ở Cần Thơ của 1 xe nào đó, biết rằng giá lăn bánh được tính theo công thức sau: 12](#_Toc14790)

[3.9 Viết trigger để theo dõi việc sửa giá trong bảng xe 14](#_Toc22806)

[3.10 Viết trigger để theo dõi việc xóa dữ liệu trong bảng xe 15](#_Toc2905)

[3.11 Người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng (quản lý, nhân viên bán xe,..) 15](#_Toc17128)

[Phần II. Bài Tập 2 Hệ Thống Quản Lý Cửa Hàng Bán Bánh 16](#_Toc29798)

[1. Mô tả 16](#_Toc19931)

[2. Sơ đồ CDM 16](#_Toc21597)

[3. CSDL về quản lý bán bánh như bảng sau: 16](#_Toc12836)

[3.1 Hàm Tính tổng số lượng tồn kho của tất cả sản phẩm. 21](#_Toc1823)

[3.2  Hàm kiểm tra sản phẩm có tồn kho hay không. 21](#_Toc13998)

[3.3 Hàm tính tổng lương của nhân viên. 22](#_Toc28718)

[3.4 Thủ tục thêm 1 sản phẩm mới vào bảng sản phẩm. 23](#_Toc12312)

[3.5 Thủ tục xóa nhân viên theo mã nhân viên. 23](#_Toc7242)

[3.6 Thủ tục tăng lương thêm 500000 cho nhân viên nào lương dưới 10 triệu. 24](#_Toc16467)

[3.7  Viết trigger cập nhật thay đổi lương của nhân viên. 25](#_Toc5387)

[3.8  Viết trigger cập nhật nhân viên bị xóa khỏi bảng nhân viên. 26](#_Toc9307)

[3.9 Viết trigger để lưu lại thời gian thêm sản phẩm mới vào. 27](#_Toc7736)

[Phần III. Bài tập 3 28](#_Toc2217)

[1. Tóm tắt câu trả lời cho các vấn đề dưới đây (Không copy & paste): 28](#_Toc4844)

[1.1 HQTCSDL là gì? Các HQTCSDL hiện nay 28](#_Toc25169)

[1.2 Khác nhau giữa HQTCSDL và bảng tính (vd, Excel) là gì? 28](#_Toc2246)

[1.3 DBA là ai? Nhiệm vụ? 28](#_Toc31207)

[1.4 Các phương pháp bảo vệ dữ liệu 28](#_Toc20920)

[1.5 Giao dịch là gì? Tại sao phải cần GD? 29](#_Toc32408)

[1.6 Các trạng thái của GD 29](#_Toc31025)

[1.7 Thuộc tính ACID là gì? 29](#_Toc27430)

[1.8 Các vấn đề cạnh tranh trong môi trường đa người dùng là gì? Giải pháp khắc phục? 29](#_Toc32696)

[1.9 Lịch trình là gì? Lịch trình tuần tự và không tuần tự 30](#_Toc5027)

[1.10 Các kỹ thuật quản lý cạnh tranh: khóa chốt và nhãn thời gian 30](#_Toc28716)

[1.11 Các kỹ thuật phục hồi DL 31](#_Toc10784)

**DANH MỤC HÌNH ẢNH**

Phần I. Bài tập 1 Quản lý Ô Tô

[Hình 3.1. 1: Bảng hãng xe 2](#_Toc17954)

[Hình 3.1. 2: Bảng dòng xe 3](#_Toc30266)

[Hình 3.1. 3: Bảng phiên bản 4](#_Toc15095)

[Hình 3.1. 4: Bảng động cơ 4](#_Toc30181)

[Hình 3.1. 5: Bảng phân khúc 5](#_Toc20769)

[Hình 3.1. 6: Bảng tĩnh thành 6](#_Toc25864)

[Hình 3.1. 7: Bảng xe 7](#_Toc7023)

[Hình 3.2. 1: Tạo thủ tục thành công 8](#_Toc14613)

[Hình 3.3. 1: Bảng xe đã thêm 1 xe mới 8](#_Toc6145)

[Hình 3.4. 1: Tạo thủ tục thành công 9](#_Toc9304)

[Hình 3.4. 2: Bảng phiên bản đã cập nhật thành công giá niêm yết 9](#_Toc20977)

[Hình 3.5. 1: Lấy giá xe của một xe nào đó 10](#_Toc10103)

[Hình 3.6. 1: Lấy loại xe, hãng xe có giá cao nhất 11](#_Toc18787)

[Hình 3.7. 1: Viết thủ tục xóa xe thành công 11](#_Toc982)

[Hình 3.7. 2: bảng xe với mã xe X001 bị xóa 12](#_Toc7370)

[Hình 3.8. 1: Bảng trả về giá lăng bánh của một xe 14](#_Toc15920)

[Hình 3.9. 1: Sửa giá xe thành công 14](#_Toc1452)

[Hình 3.10. 1: Theo dõi việc xóa dữ liệu trong bảng xe 15](#_Toc6884)

Phần II. Bài Tập 2 Quản Lý Cửa Hàng Bán Bánh

[Hình 3.1: Bảng sản phẩm 17](#_Toc1100)

[Hình 3.2: Bảng khách hàng 18](#_Toc14757)

[Hình 3.3: Bảng đơn hàng 18](#_Toc31280)

[Hình 3.4: Bảng chi tiết đơn hàng 19](#_Toc21047)

[Hình 3.5: Bảng Thanh Toán 20](#_Toc2780)

[Hình 3.6: Bảng nhân viên 21](#_Toc13745)

[Hình 3.1.1: Hàm tính tổng tồn kho 21](#_Toc12820)

[Hình 3.2.1: Kiểm tra sản phẩm có tồn kho không 22](#_Toc5305)

[Hình 3.3.1: Hàm tính tổng lương nhân viên 22](#_Toc14866)

[Hình 3.4.1: kết quả thủ tục thêm một sản phẩm 23](#_Toc9247)

[Hình 3.5.1: kết quả thủ tục xóa nhân viên 24](#_Toc20957)

[Hình 3.6.1: Kết quả tăng lương cho nhân viên 24](#_Toc22114)

[Hình 3.7.1: bảng lưu lại kết quả cập nhật lương 25](#_Toc27462)

[Hình 3.8.1: Bảng lưu lại nhân viên bị xóa 26](#_Toc8753)

[Hình 3.9.1: Bảng lưu lại sản phẩm mới được thêm 27](#_Toc17879)

**DANH MỤC TỪ VIẾT TẮT**

CSDL Cơ Cở Dữ Liệu

RAID Redundant Array of Independent Disks

CDM Conceptual Data Model

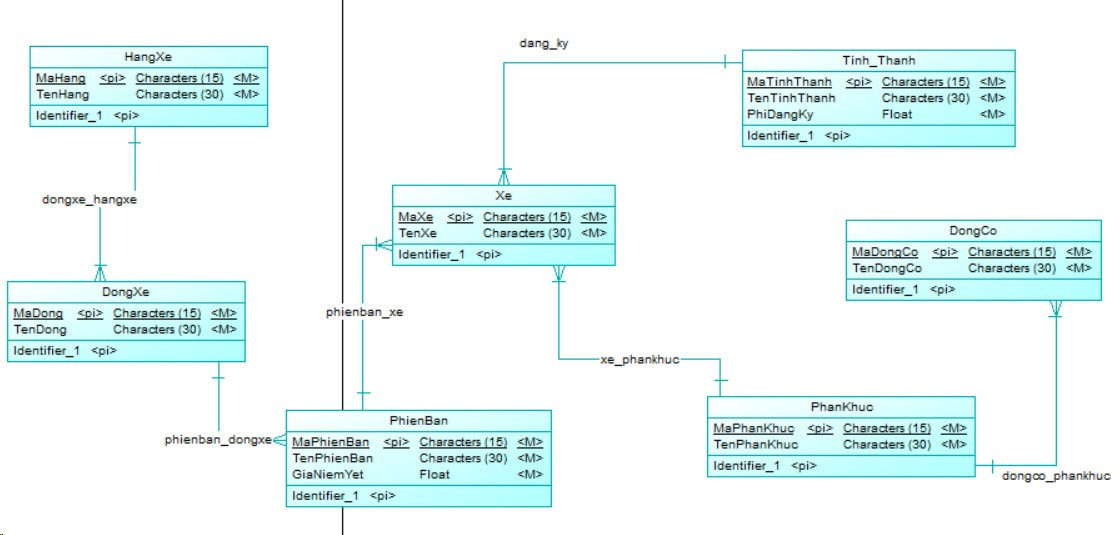
GD Giao Dịch

**Phần I. Bài tập 1 Hệ Thống Quản Lý Ô Tô**

1. **Mô tả**

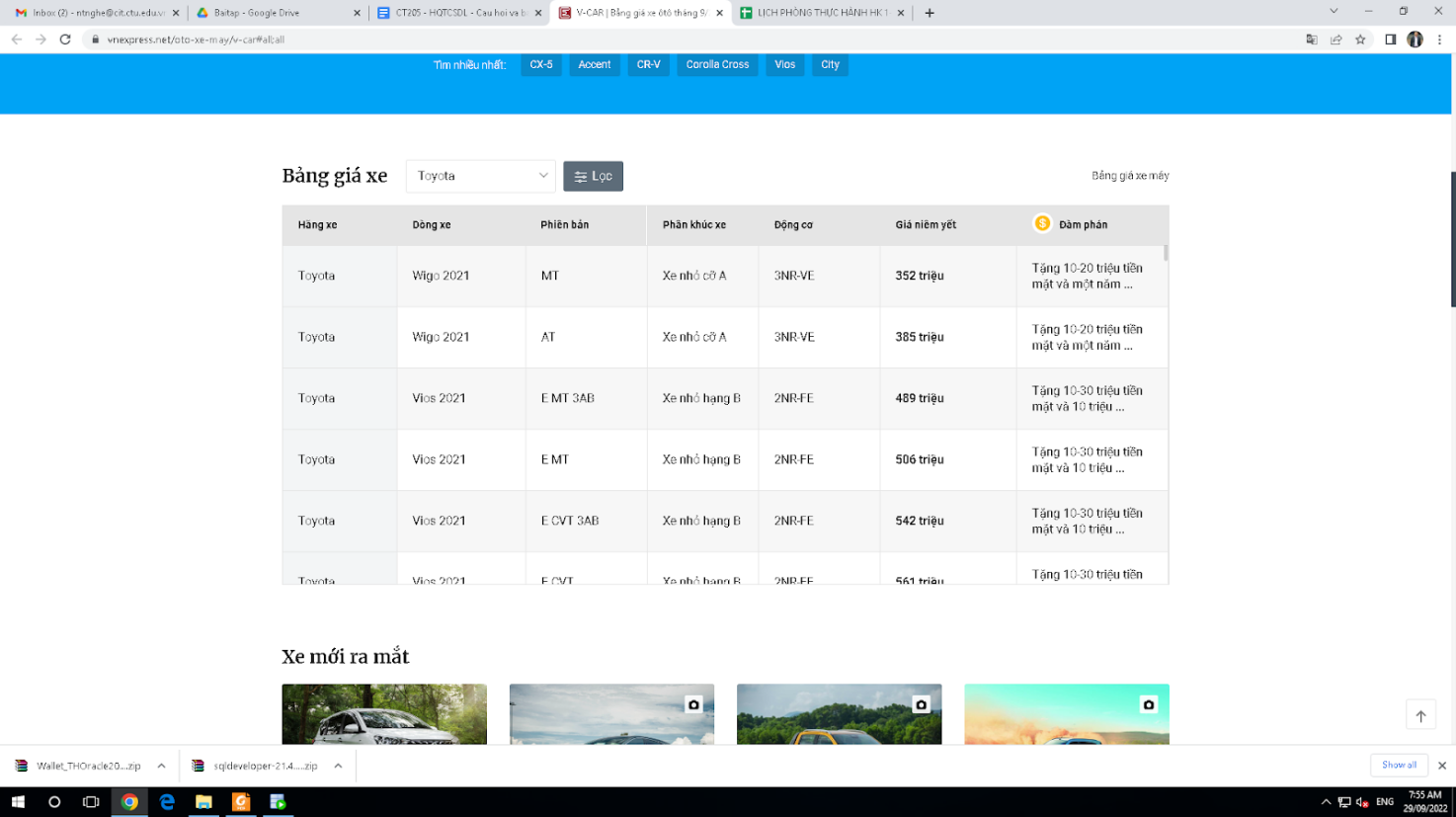
Trong bối cảnh ngành công nghiệp ô tô đang phát triển mạnh mẽ, việc sử dụng một hệ thống quản lý bán ô tô là không thể thiếu. Hệ thống này không chỉ giúp doanh nghiệp quản lý hiệu quả các quy trình kinh doanh mà còn để theo dõi, cập nhật xe mới, giá cả, cung cấp báo cáo và thống kê về doanh số bán và lợi nhuận, và quản lý người dùng và phân quyền truy cập. Đồng thời, nó còn giúp doanh nghiệp thích nghi với những thay đổi trong thị trường và tận dụng cơ hội mới để phát triển.

1. **Sơ đồ CDM**



1. **CSDL về xe ô tô như bảng sau:**

[https://vnexpress.net/oto-xe-may/v-car#all;all](https://vnexpress.net/oto-xe-may/v-car" \l "all;all)



**3.1 Tạo các bảng để lưu các thông tin trên. Lưu ý tất cả các ràng buộc (khóa chính, khóa ngoại, check, not null, default), biểu thức chính quy có thể có.**

CREATE TABLE HangXe (

    MaHang VARCHAR2(15) PRIMARY KEY,

    TenHang VARCHAR2(30) NOT NULL

);

-- Thêm dữ liệu vào bảng HangXe

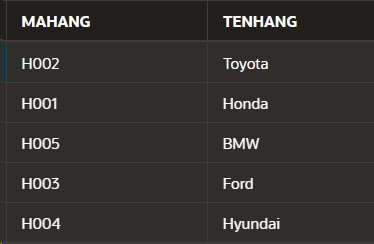
INSERT INTO HangXe (MaHang, TenHang) VALUES ('H001', 'Honda');

INSERT INTO HangXe (MaHang, TenHang) VALUES ('H002', 'Toyota');

INSERT INTO HangXe (MaHang, TenHang) VALUES ('H003', 'Ford');

INSERT INTO HangXe (MaHang, TenHang) VALUES ('H004', 'Hyundai');

INSERT INTO HangXe (MaHang, TenHang) VALUES ('H005', 'BMW');



Hình 3.1. 1: Bảng hãng xe

CREATE TABLE DongXe (

    MaDong VARCHAR2(15) PRIMARY KEY,

    TenDong VARCHAR2(30) NOT NULL,

    MaHang VARCHAR2(15),

    FOREIGN KEY (MaHang) REFERENCES HangXe(MaHang)

);

-- Thêm dữ liệu vào bảng DongXe

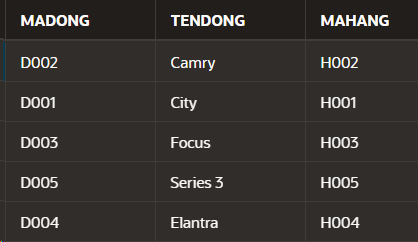
INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES ('D001', 'City', 'H001');

INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES ('D002', 'Camry', 'H002');

INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES ('D003', 'Focus', 'H003');

INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES ('D004', 'Elantra', 'H004');

INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES ('D005', 'Series 3', 'H005');



Hình 3.1. 2: Bảng dòng xe

CREATE TABLE PhienBan (

    MaPhienBan VARCHAR2(15) PRIMARY KEY,

    TenPhienBan VARCHAR2(30) NOT NULL,

    GiaNiemYet FLOAT check(GiaNiemYet > 0) NOT NULL,

    MaDong VARCHAR2(15),

    MaHang VARCHAR2(15),

    FOREIGN KEY (MaDong) REFERENCES DongXe(MaDong),

    FOREIGN KEY (MaHang) REFERENCES HangXe(MaHang)

);

-- Thêm dữ liệu vào bảng PhienBan

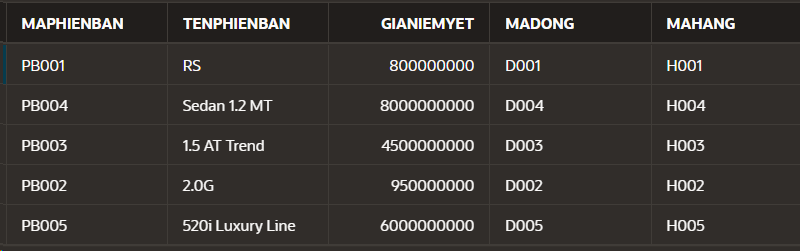
INSERT INTO PhienBan (MaPhienBan, TenPhienBan, GiaNiemYet, MaDong, MaHang) VALUES ('PB001', 'RS', 800000000, 'D001', 'H001');

INSERT INTO PhienBan (MaPhienBan, TenPhienBan, GiaNiemYet, MaDong, MaHang) VALUES ('PB002', '2.0G', 950000000, 'D002', 'H002');

INSERT INTO PhienBan (MaPhienBan, TenPhienBan, GiaNiemYet, MaDong, MaHang) VALUES ('PB003', '1.5 AT Trend', 4500000000, 'D003', 'H003');

INSERT INTO PhienBan (MaPhienBan, TenPhienBan, GiaNiemYet, MaDong, MaHang) VALUES ('PB004', 'Sedan 1.2 MT', 8000000000, 'D004', 'H004');

INSERT INTO PhienBan (MaPhienBan, TenPhienBan, GiaNiemYet, MaDong, MaHang) VALUES ('PB005', '520i Luxury Line', 6000000000, 'D005', 'H005');



Hình 3.1. 3: Bảng phiên bản

CREATE TABLE DongCo (

    MaDongCo VARCHAR2(15) PRIMARY KEY,

    TenDongCo VARCHAR2(30) NOT NULL

);

-- Thêm dữ liệu vào bảng DongCo

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES ('DC001', 'VTEC');

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES ('DC002', 'VVT-i');

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES ('DC003', 'EcoBoost');

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES ('DC004', 'Gamma');

INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES ('DC005', 'N20');



Hình 3.1. 4: Bảng động cơ

CREATE TABLE PhanKhuc (

    MaPhanKhuc VARCHAR2(15) PRIMARY KEY,

    TenPhanKhuc VARCHAR2(30) NOT NULL,

    MaDongCo VARCHAR2(15),

    FOREIGN KEY (MaDongCo) REFERENCES DongCo(MaDongCo)

);

-- Thêm dữ liệu vào bảng PhanKhuc

INSERT INTO PhanKhuc (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK001', 'Xe nhỏ hạng B', 'DC001');

INSERT INTO PhanKhuc (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK002', 'Xe cỡ trung hạng D', 'DC002');

INSERT INTO PhanKhuc (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK003', 'Xe nhỏ hạng B', 'DC003');

INSERT INTO PhanKhuc (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK004', 'Xe nhỏ cỡ A', 'DC004');

INSERT INTO PhanKhuc (MaPhanKhuc, TenPhanKhuc, MaDongCo) VALUES ('PK005', 'Xe sang cỡ trung', 'DC005');



Hình 3.1. 5: Bảng phân khúc

CREATE TABLE Tinh\_Thanh (

    MaTinhThanh VARCHAR2(15) PRIMARY KEY,

    TenTinhThanh VARCHAR2(30) NOT NULL,

    PhiDangKi FLOAT check(PhiDangKi > 0) NOT NULL

);

-- Thêm dữ liệu vào bảng Tinh\_Thanh

INSERT INTO Tinh\_Thanh (MaTinhThanh, TenTinhThanh, PhiDangKi) VALUES ('29', 'Hà Nội',900000);

INSERT INTO Tinh\_Thanh (MaTinhThanh, TenTinhThanh, PhiDangKi) VALUES ('51', 'TP.HCM', 1500000);

INSERT INTO Tinh\_Thanh (MaTinhThanh, TenTinhThanh, PhiDangKi) VALUES ('43', 'Đà Nẵng', 1000000);

INSERT INTO Tinh\_Thanh (MaTinhThanh, TenTinhThanh, PhiDangKi) VALUES ('15', 'Hải Phòng', 1200000);

INSERT INTO Tinh\_Thanh (MaTinhThanh, TenTinhThanh, PhiDangKi) VALUES ('65', 'Cần Thơ', 1000000);



Hình 3.1. 6: Bảng tĩnh thành

CREATE TABLE Xe (

    MaXe VARCHAR2(15) PRIMARY KEY,

    TenXe VARCHAR2(30) NOT NULL,

    MaHang VARCHAR2(15),

    MaDong VARCHAR2(15),

    MaPhienBan VARCHAR2(15),

    MaTinhThanh VARCHAR2(15),

    PhanKhuc VARCHAR2(15),

    MaDongCo VARCHAR2(15),

    FOREIGN KEY (MaHang) REFERENCES HangXe(MaHang),

    FOREIGN KEY (MaDong) REFERENCES DongXe(MaDong),

    FOREIGN KEY (MaPhienBan) REFERENCES PhienBan(MaPhienBan),

    FOREIGN KEY (MaTinhThanh) REFERENCES Tinh\_Thanh(MaTinhThanh),

    FOREIGN KEY (PhanKhuc) REFERENCES PhanKhuc(MaPhanKhuc),

    FOREIGN KEY (MaDongCo) REFERENCES DongCo(MaDongCo)

);

-- Thêm dữ liệu vào bảng Xe

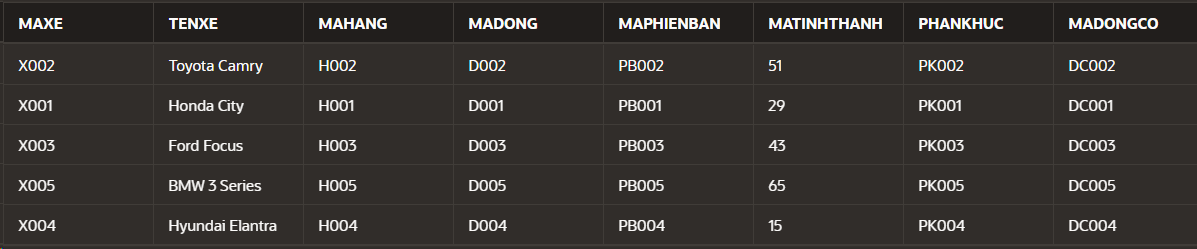
INSERT INTO Xe (MaXe, TenXe, MaHang, MaDong, MaPhienBan, MaTinhThanh, PhanKhuc, MaDongCo) VALUES ('X001', 'Honda City', 'H001', 'D001', 'PB001', '29', 'PK001', 'DC001');

INSERT INTO Xe (MaXe, TenXe, MaHang, MaDong, MaPhienBan, MaTinhThanh, PhanKhuc, MaDongCo) VALUES ('X002', 'Toyota Camry', 'H002', 'D002', 'PB002', '51', 'PK002', 'DC002');

INSERT INTO Xe (MaXe, TenXe, MaHang, MaDong, MaPhienBan, MaTinhThanh, PhanKhuc, MaDongCo) VALUES ('X003', 'Ford Focus', 'H003', 'D003', 'PB003', '43', 'PK003', 'DC003');

INSERT INTO Xe (MaXe, TenXe, MaHang, MaDong, MaPhienBan, MaTinhThanh, PhanKhuc, MaDongCo) VALUES ('X004', ' [Hyundai](https://vnexpress.net/oto-xe-may/v-car/hang-xe/hyundai-9) Elantra', 'H004', 'D004', '15', 'TT004', 'PK004', 'DC004');

INSERT INTO Xe (MaXe, TenXe, MaHang, MaDong, MaPhienBan, MaTinhThanh, PhanKhuc, MaDongCo) VALUES ('X005', 'BMW Series 3', 'H005', 'D005', 'PB005', '65', 'PK005', 'DC005');



Hình 3.1. 7: Bảng xe

**3.2 Viết thủ tục cho phép nhập vào 1 xe mới**

CREATE OR REPLACE PROCEDURE ThemXeMoi (

    p\_MaXe IN VARCHAR2,

    p\_TenXe IN VARCHAR2,

    p\_MaHang IN VARCHAR2,

    p\_MaDong IN VARCHAR2,

    p\_MaPhienBan IN VARCHAR2,

    p\_MaTinhThanh IN VARCHAR2,

    p\_PhanKhuc IN VARCHAR2,

    p\_MaDongCo IN VARCHAR2

)

IS

BEGIN

    INSERT INTO Xe (MaXe, TenXe, MaHang, MaDong, MaPhienBan, MaTinhThanh, PhanKhuc, MaDongCo)

    VALUES (p\_MaXe, p\_TenXe, p\_MaHang, p\_MaDong, p\_MaPhienBan, p\_MaTinhThanh, p\_PhanKhuc, p\_MaDongCo);

    COMMIT;

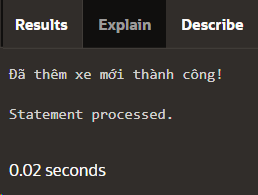
    DBMS\_OUTPUT.PUT\_LINE('Đã thêm xe mới thành công!');

EXCEPTION

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Đã xảy ra lỗi: ' || SQLERRM);

END;



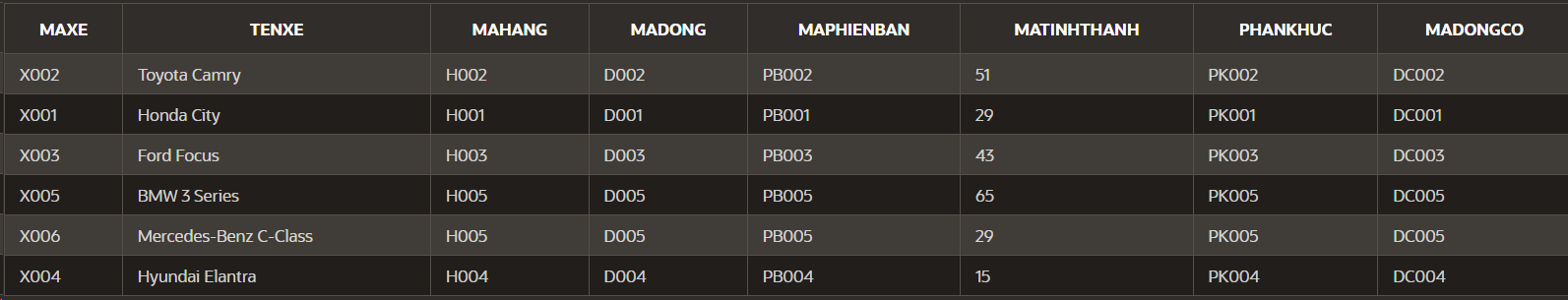
Hình 3.2. 1: Tạo thủ tục thành công

**3.3 Gọi thủ tục để Nhập liệu như bảng trên**

BEGIN

     ThemXeMoi('X006', 'Mercedes-Benz C-Class', 'H005', 'D005', 'PB005', '29', 'PK005', 'DC005');

END;



Hình 3.3. 1: Bảng xe đã thêm 1 xe mới

**3.4 Viết thủ tục cho phép cập nhật giá niêm yết của 1 xe nào đó.**

CREATE OR REPLACE PROCEDURE CapNhatGiaXe (

    p\_MaXe IN VARCHAR2,

    p\_GiaMoi IN FLOAT

)

IS

BEGIN

    UPDATE PhienBan

    SET GiaNiemYet = p\_GiaMoi

    WHERE MaPhienBan = (

        SELECT MaPhienBan

        FROM Xe

        WHERE MaXe = p\_MaXe

    );

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE('Đã cập nhật giá niêm yết của xe thành công!');

EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        DBMS\_OUTPUT.PUT\_LINE('Không tìm thấy thông tin về xe có mã ' || p\_MaXe);

    WHEN OTHERS THEN

        DBMS\_OUTPUT.PUT\_LINE('Đã xảy ra lỗi: ' || SQLERRM);

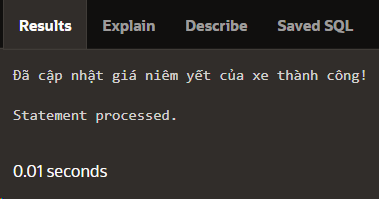
END;

--gọi thủ tục

BEGIN

CapNhatGiaXe('X001', 850000000);

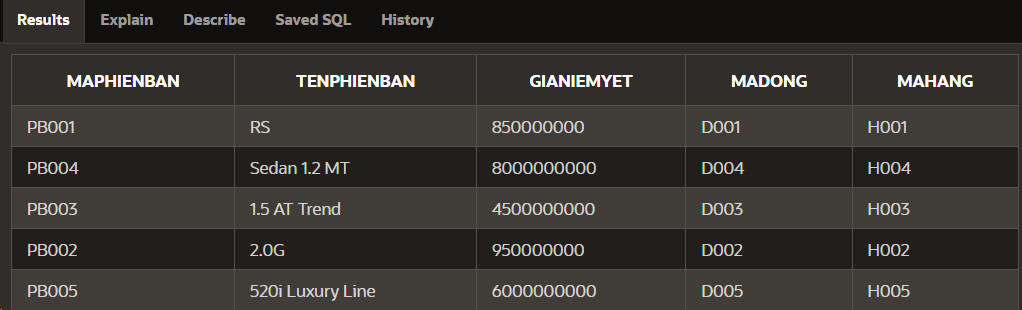
END;



Hình 3.4. 1: Tạo thủ tục thành công

SELECT \* FROM PhienBan;

-- Cập nhật giá niêm yết của xe có mã 'X001' thành 850,000,000 VND



Hình 3.4. 2: Bảng phiên bản đã cập nhật thành công giá niêm yết

**3.5 Viết hàm trả về giá của 1 xe nào đó**

CREATE OR REPLACE FUNCTION LayGiaXe(

    p\_MaXe IN VARCHAR2

) RETURN FLOAT

IS

    v\_Gia FLOAT;

BEGIN

    SELECT PhienBan.GiaNiemYet INTO v\_Gia

    FROM Xe

    INNER JOIN PhienBan ON Xe.MaPhienBan = PhienBan.MaPhienBan

    WHERE Xe.MaXe = p\_MaXe;

    RETURN v\_Gia;

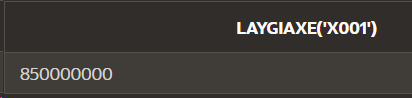
EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RETURN NULL;

END;

SELECT LayGiaXe('X001') FROM dual;



Hình 3.5. 1: Lấy giá xe của một xe nào đó

**3.6 Viết hàm trả về loại xe, hãng xe có giá cao nhất**

CREATE OR REPLACE FUNCTION TimXeGiaCaoNhat RETURN VARCHAR2

IS

    v\_LoaiXe VARCHAR2(100);

BEGIN

    SELECT DX.TenDong || ' ' || HX.TenHang INTO v\_LoaiXe

    FROM Xe X

    INNER JOIN DongXe DX ON X.MaDong = DX.MaDong

    INNER JOIN HangXe HX ON X.MaHang = HX.MaHang

    INNER JOIN PhienBan PB ON X.MaPhienBan = PB.MaPhienBan

    WHERE PB.GiaNiemYet = (SELECT MAX(GiaNiemYet) FROM PhienBan);

    RETURN v\_LoaiXe;

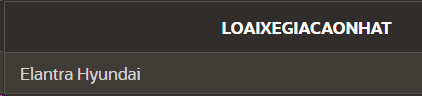
EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RETURN NULL;

END;

SELECT TimXeGiaCaoNhat() AS LoaiXeGiaCaoNhat FROM dual;



Hình 3.6. 1: Lấy loại xe, hãng xe có giá cao nhất

**3.7 Viết thủ tục để xóa 1 xe nào đó**

CREATE OR REPLACE PROCEDURE XoaXe (

    p\_MaXe IN VARCHAR2

)

IS

BEGIN

    DELETE FROM Xe

    WHERE MaXe = p\_MaXe;

    COMMIT;

    DBMS\_OUTPUT.PUT\_LINE('Đã xóa xe thành công!');

EXCEPTION

    WHEN OTHERS THEN

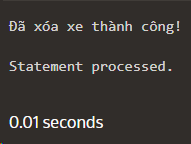
        DBMS\_OUTPUT.PUT\_LINE('Đã xảy ra lỗi: ' || SQLERRM);

END;

BEGIN

    XoaXe('X001');

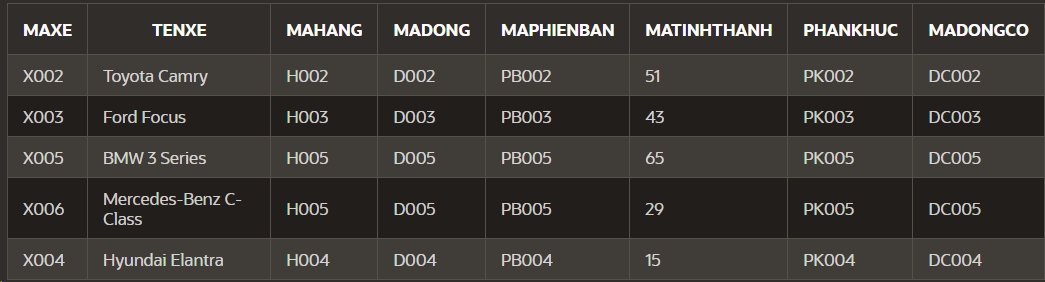
END;



Hình 3.7. 1: Viết thủ tục xóa xe thành công

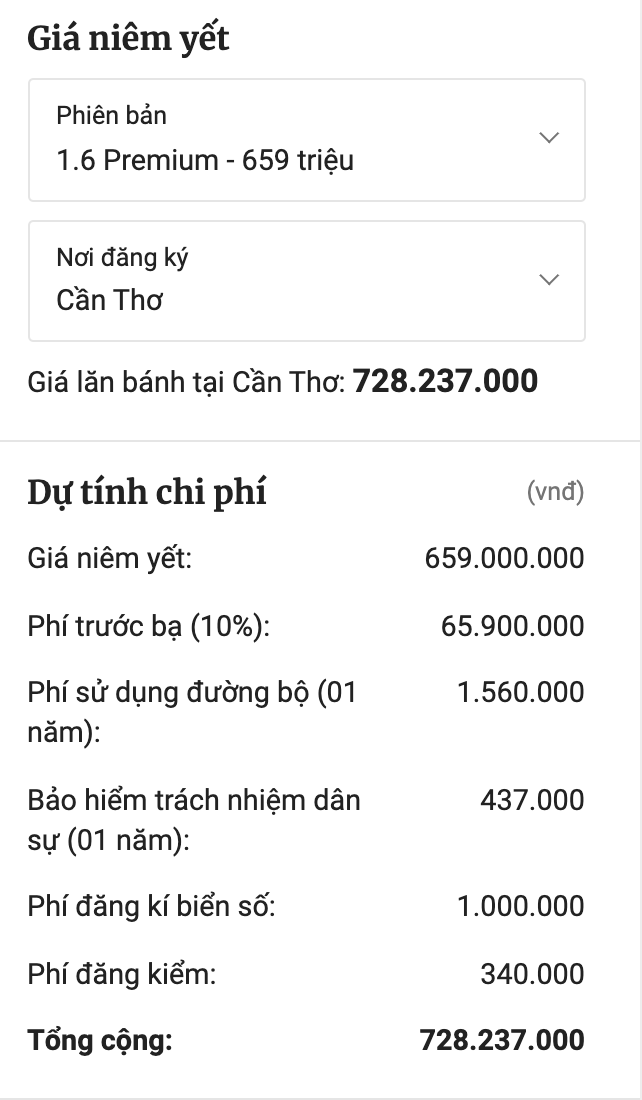
--MaXe X001 đã bị xóa

SELECT \* FROM Xe;



Hình 3.7. 2: bảng xe với mã xe X001 bị xóa

**3.8 Viết hàm trả về giá lăn bánh ở Cần Thơ của 1 xe nào đó, biết rằng giá lăn bánh được tính theo công thức sau:**



CREATE OR REPLACE FUNCTION CALCULATE\_TOTAL\_PRICE (

    p\_maphienban IN VARCHAR2,

    p\_ma\_tinhthanh IN VARCHAR2

) RETURN FLOAT

IS

    v\_gia\_niem\_yet FLOAT;

    v\_phi\_cuuoc\_ban FLOAT;

    v\_phi\_sudung\_duongbo FLOAT := 1560000;

v\_baohiem\_trachnhiemdansu FLOAT := 437000;

    v\_phi\_dangky\_bienso FLOAT;

    v\_phi\_dangkiem FLOAT := 340000;

    v\_gia\_lanbanh\_cantho FLOAT;

BEGIN

    -- Lấy giá niêm yết của xe

    SELECT GiaNiemYet INTO v\_gia\_niem\_yet

    FROM PhienBan

    WHERE MaPhienBan = p\_maphienban;

    -- Tính phí cước bạn (10%)

    v\_phi\_cuuoc\_ban := v\_gia\_niem\_yet \* 0.1;

    -- Lấy phí đăng ký biển số từ bảng Tinh\_Thanh

    SELECT PhiDangKi INTO v\_phi\_dangky\_bienso

    FROM Tinh\_Thanh

    WHERE MaTinhThanh = p\_ma\_tinhthanh;

    -- Tính tổng giá lăn bánh tại Cần Thơ

    v\_gia\_lanbanh\_cantho := v\_gia\_niem\_yet + v\_phi\_cuuoc\_ban + v\_phi\_sudung\_duongbo + v\_baohiem\_trachnhiemdansu + v\_phi\_dangky\_bienso + v\_phi\_dangkiem;

    RETURN v\_gia\_lanbanh\_cantho;

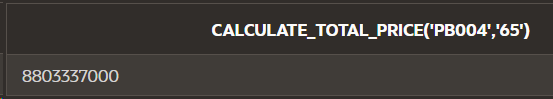
EXCEPTION

    WHEN NO\_DATA\_FOUND THEN

        RETURN NULL;

END;

SELECT CALCULATE\_TOTAL\_PRICE('PB004', '65') FROM dual;

**

Hình 3.8. 1: Bảng trả về giá lăng bánh của một xe

**3.9 Viết trigger để theo dõi việc sửa giá trong bảng xe**

CREATE TABLE CapNhatGiaXe(

    MaPhienBan VARCHAR(10),

    GiaNiemYetCu FLOAT,

    GiaNiemYetMoi FLOAT,

    ThoiGianCapNhat TIMESTAMP

);

CREATE OR REPLACE TRIGGER TheoDoiCapNhatGiaXe

AFTER UPDATE OF GiaNiemYet ON PhienBan

FOR EACH ROW

BEGIN

    IF UPDATING('GiaNiemYet') THEN

        INSERT INTO CapNhatGiaXe (MaPhienBan, GiaNiemYetCu, GiaNiemYetMoi, ThoiGianCapNhat)

        VALUES (:OLD.MaPhienBan, :OLD.GiaNiemYet, :NEW.GiaNiemYet, SYSTIMESTAMP);

    END IF;

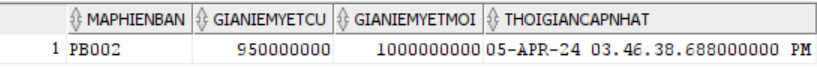
END;

-- Cập nhật giá cho một phiên bản xe

UPDATE PhienBan SET GiaNiemYet = 1000000000 WHERE MaPhienBan = 'PB002';

-- Xem kết quả

SELECT \* FROM CapNhatGiaXe;



Hình 3.9. 1: Sửa giá xe thành công

**3.10 Viết trigger để theo dõi việc xóa dữ liệu trong bảng xe**

CREATE TABLE LichSuXoaXe (

    MaXe VARCHAR2(15),

    TenXe VARCHAR2(30),

    ThoiGianXoa DATE

);

CREATE OR REPLACE TRIGGER TheoDoiXoaXe

AFTER DELETE ON Xe

FOR EACH ROW

BEGIN

    INSERT INTO LichSuXoaXe (MaXe, TenXe, ThoiGianXoa)

    VALUES (:OLD.MaXe, :OLD.TenXe, SYSDATE);

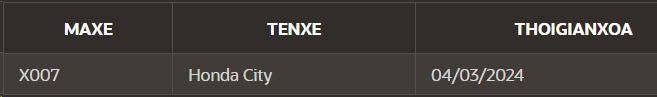
END;

DELETE FROM Xe

WHERE MaXe = 'X007';

SELECT \* FROM LichSuXoaXe;

INSERT INTO Xe (MaXe, TenXe, MaHang, MaDong, MaPhienBan, MaTinhThanh, PhanKhuc, MaDongCo) VALUES ('X007', 'Honda City', 'H001', 'D001', 'PB001', '29', 'PK001', 'DC001');



Hình 3.10. 1: Theo dõi việc xóa dữ liệu trong bảng xe

**3.11 Người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng (quản lý, nhân viên bán xe,..)**

-- quanlyoto có tất cả các quyền

CREATE USER quanlyoto IDENTIFIED BY quanlyoto123;

GRANT ALL PRIVILEGES TO quanlyoto;

-- nhân viên có quyền xem và cập nhật

CREATE USER nhanvienbanxe IDENTIFIED BY nhanvien123;

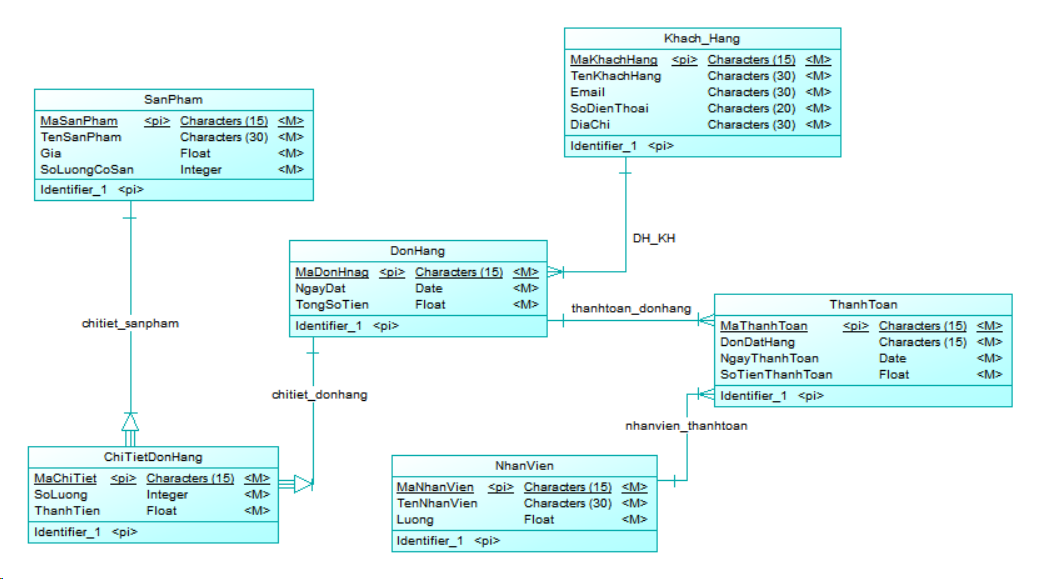
GRANT SELECT, UPDATE ON GiaXe TO nhanvienbanxe;

**Phần II. Bài Tập 2 Hệ Thống Quản Lý Cửa Hàng Bán Bánh**

1. **Mô tả**

Hệ thống quản lý bán bánh là một ứng dụng máy tính được thiết kế để hỗ trợ các cửa hàng bán bánh trong việc quản lý các hoạt động kinh doanh hàng ngày một cách hiệu quả. Hệ thống này cho phép quản lý sản phẩm bánh, quản lý đơn hàng và khách hàng, theo dõi tồn kho, cung cấp báo cáo và thống kê về doanh số bán hàng và lợi nhuận, và quản lý người dùng và phân quyền truy cập. Với các tính năng này, hệ thống giúp cửa hàng tối ưu hóa quá trình quản lý, tăng cường hiệu suất làm việc và cung cấp trải nghiệm mua sắm tốt hơn cho khách hàng.

1. **Sơ đồ CDM**



1. **CSDL về quản lý bán bánh như bảng sau:**

--Tạo bảng sản phẩm

CREATE TABLE  SanPham (

    MaSanPham  VARCHAR (15) PRIMARY KEY,

    TenSanPham  VARCHAR(30) NOT NULL,

    Gia FLOAT check(Gia >0),

    SoLuongCoSan INT NOT NULL

);

-- Nhập dữ liệu cho bảng Sản phẩm

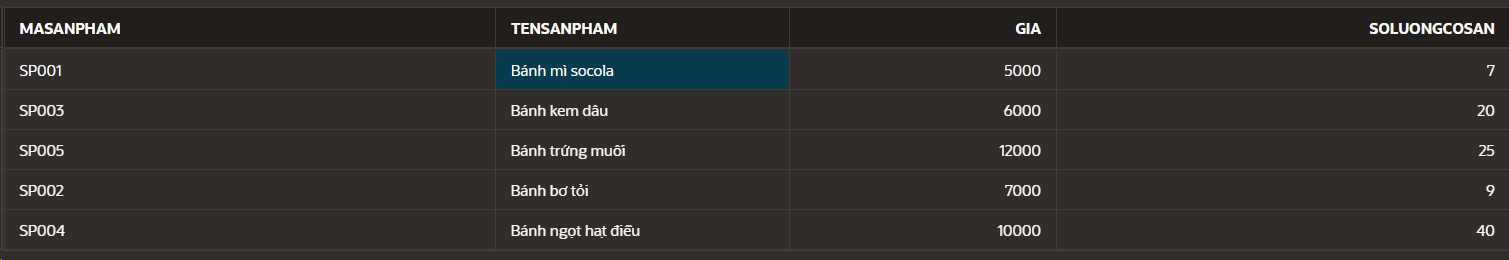
INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP001', 'Bánh mì socola', 5000, 7);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP002', 'Bánh bơ tỏi', 7000, 9);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP003', 'Bánh kem dâu', 6000, 20);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP004', 'Bánh ngọt hạt điều', 10000, 40);

INSERT INTO SanPham(MaSanPham, TenSanPham, Gia, SoLuongCoSan) VALUES ('SP005', 'Bánh trứng muối', 12000, 25);



Hình 3.1: Bảng sản phẩm

--Tạo bảng khách hàng

CREATE TABLE  Khach\_Hang (

    MaKhachHang  VARCHAR (15)PRIMARY KEY,

    TenKhachHang VARCHAR(30) NOT NULL,

    Email VARCHAR(30) NOT NULL,

    SoDienThoai VARCHAR(20) NOT NULL,

    DiaChi VARCHAR(30) NOT NULL

);

-- Nhập dữ liệu cho bảng Khach\_Hang

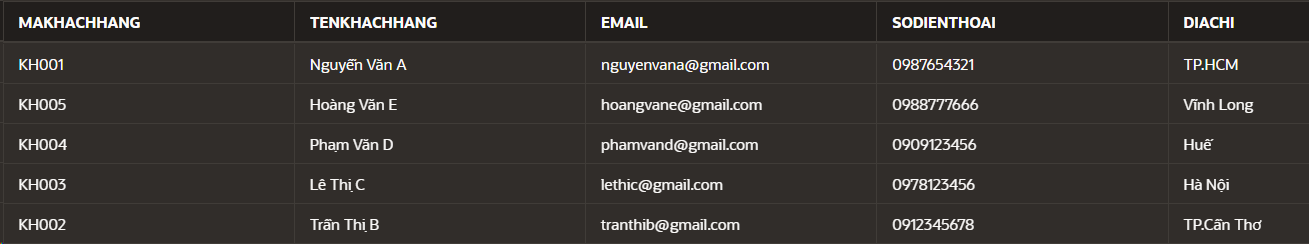
INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH001', 'Nguyễn Văn A', 'nguyenvana@gmail.com', '0987654321', 'TP.HCM');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH002', 'Trần Thị B', 'tranthib@gmail.com', '0912345678', 'TP.Cần Thơ');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH003', 'Lê Thị C', 'lethic@gmail.com', '0978123456', 'Hà Nội');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH004', 'Phạm Văn D', 'phamvand@gmail.com', '0909123456', 'Huế');

INSERT INTO Khach\_Hang (MaKhachHang, TenKhachHang, Email, SoDienThoai, DiaChi) VALUES('KH005', 'Hoàng Văn E', 'hoangvane@gmail.com', '0988777666', 'Vĩnh Long');



Hình 3.2: Bảng khách hàng

--Tạo bảng đơn hàng

CREATE TABLE  DonHang(

   MaDonHang VARCHAR (15) PRIMARY KEY,

    MaKhachHang VARCHAR (15),

    NgayDatHang DATE NOT NULL,

    TongSoTien FLOAT check(TongSoTien >0),

    FOREIGN KEY (MaKhachHang) REFERENCES Khach\_Hang (MaKhachHang)

);

-- Nhập dữ liệu cho bảng DonHang

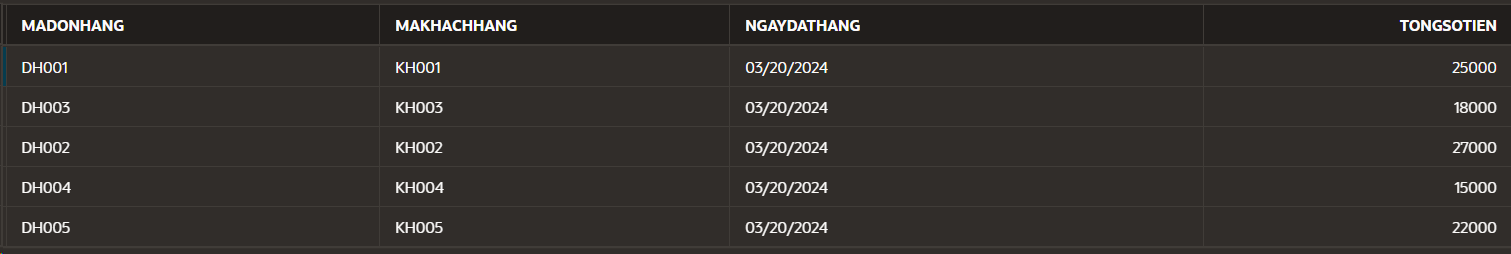
INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH001', 'KH001', to\_date('2024-03-20','yyyy-mm-dd'), 25000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH002', 'KH002', to\_date('2024-03-20','yyyy-mm-dd'), 27000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH003', 'KH003', to\_date('2024-03-20','yyyy-mm-dd'), 18000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH004', 'KH004', to\_date('2024-03-20','yyyy-mm-dd'), 15000);

INSERT INTO DonHang (MaDonHang, MaKhachHang, NgayDatHang, TongSoTien) VALUES('DH005', 'KH005', to\_date('2024-03-20','yyyy-mm-dd'), 22000);



Hình 3.3: Bảng đơn hàng

--Bảng chi tiết đơn hàng

CREATE TABLE ChiTietDonHang (

    MaChiTiet  VARCHAR(15) PRIMARY KEY,

    MaDonHang VARCHAR(15),

    MaSanPham VARCHAR(15),

    SoLuong INT NOT NULL check(SoLuong >0),

    ThanhTien FLOAT check(ThanhTien >0),

    FOREIGN KEY (MaDonHang ) REFERENCES DonHang(MaDonHang ),

    FOREIGN KEY (MaSanPham ) REFERENCES SanPham (MaSanPham )

);

-- Nhập dữ liệu cho bảng ChiTietDonHang

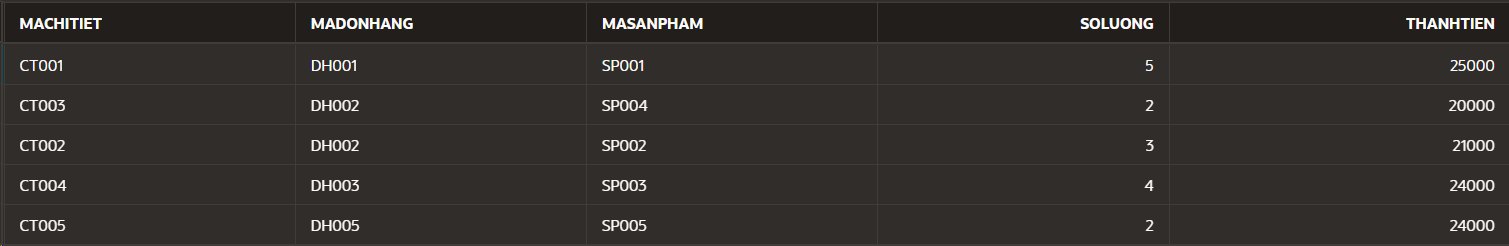
INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT001', 'DH001', 'SP001', 5, 25000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT002', 'DH002', 'SP002', 3, 21000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT003', 'DH002', 'SP004', 2, 20000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT004', 'DH003', 'SP003', 4, 24000);

INSERT INTO ChiTietDonHang (MaChiTiet, MaDonHang, MaSanPham, SoLuong, ThanhTien) VALUES ('CT005', 'DH005', 'SP005', 2, 24000);



Hình 3.4: Bảng chi tiết đơn hàng

- Tạo Bảng thanh toán

CREATE TABLE Thanh\_Toan (

    MaThanhToan VARCHAR(15) PRIMARY KEY,

    DonDatHang VARCHAR(15) NOT NULL,

    NgayThanhToan DATE NOT NULL,

    MaDonHang VARCHAR(15) ,

    SoTienThanhToan FLOAT check(SoTienThanhToan  >0),

    FOREIGN KEY (MaDonHang ) REFERENCES DonHang(MaDonHang )

);

-- Nhập dữ liệu cho bảng Thanh\_Toan

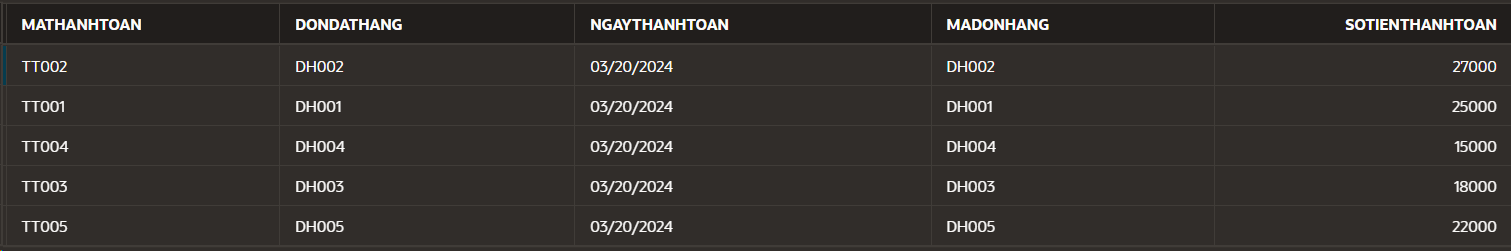
INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT001', 'DH001', to\_date('2024-03-20','yyyy-mm-dd'), 'DH001', 25000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT002', 'DH002', to\_date('2024-03-20','yyyy-mm-dd'), 'DH002', 27000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT003', 'DH003', to\_date('2024-03-20','yyyy-mm-dd'), 'DH003', 18000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT004', 'DH004', to\_date('2024-03-20','yyyy-mm-dd'), 'DH004', 15000);

INSERT INTO Thanh\_Toan (MaThanhToan, DonDatHang, NgayThanhToan, MaDonHang, SoTienThanhToan) VALUES ('TT005', 'DH005', to\_date('2024-03-20','yyyy-mm-dd'), 'DH005', 22000);



Hình 3.5: Bảng Thanh Toán

--Tạo bảng nhân viên

CREATE TABLE NhanVien(

    MaNhanVien VARCHAR(15) PRIMARY KEY,

    TenNhanVien VARCHAR(30) NOT NULL,

    Luong FLOAT check(Luong >0)

):

-- Nhập dữ liệu cho bảng NhanVien

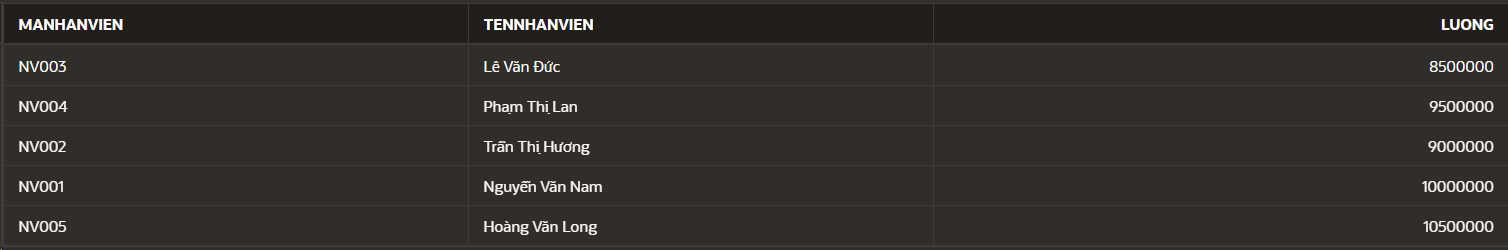
INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV001', 'Nguyễn Văn Nam', 10000000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV002', 'Trần Thị Hương', 9000000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV003', 'Lê Văn Đức', 8500000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV004', 'Phạm Thị Lan', 9500000);

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV005', 'Hoàng Văn Long', 10500000);



Hình 3.6: Bảng nhân viên

**3.1 Hàm Tính tổng số lượng tồn kho của tất cả sản phẩm.**

CREATE OR REPLACE FUNCTION TinhTongTonKho RETURN NUMBER

IS

    TonKho NUMBER := 0;

BEGIN

    FOR SanPham\_L IN (SELECT SoLuongCoSan FROM SanPham) LOOP

        TonKho := TonKho + SanPham\_L.SoLuongCoSan;

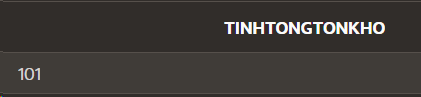
    END LOOP;

    RETURN TonKho ;

END;

-- xem kết quả

SELECT TinhTongTonKho from dual;



Hình 3.1.1: Hàm tính tổng tồn kho

**3.2  Hàm kiểm tra sản phẩm có tồn kho hay không.**

CREATE OR REPLACE FUNCTION KiemTraTonKho(maSP IN VARCHAR) RETURN VARCHAR

IS

    soLuong NUMBER;

    result VARCHAR(20);

BEGIN

    -- Lấy số lượng tồn kho của sản phẩm với mã sản phẩm được cung cấp

    SELECT SoLuongCoSan INTO soLuong

    FROM SanPham

    WHERE MaSanPham = maSP;

    -- Kiểm tra nếu số lượng tồn kho lớn hơn 0, trả về tồn tại, ngược lại trả về không tồn tại

    IF soLuong > 0 THEN

        result := 'Tồn tại';

    ELSE

        result := 'Không tồn tại';

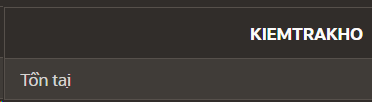
    END IF;

    RETURN result;

END ;

-- xem kết quả

SELECT KiemTraTonKho('SP001') as kiemtrakho FROM dual;



Hình 3.2.1: Kiểm tra sản phẩm có tồn kho không

**3.3 Hàm tính tổng lương của nhân viên.**

CREATE OR REPLACE FUNCTION TongLuongNhanVien  RETURN NUMBER

 IS

    TongTienLuong NUMBER := 0;

BEGIN

      FOR NhanVien\_L IN (SELECT Luong FROM NhanVien)

      LOOP

        TongTienLuong := TongTienLuong + NhanVien\_L.Luong;

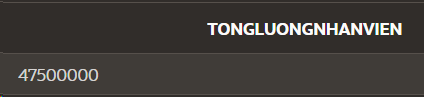
    END LOOP;

    RETURN TongTienLuong;

END;

-- chạy đoạn này để xem kết quả

SELECT TongLuongNhanVien FROM DUAL;



Hình 3.3.1: Hàm tính tổng lương nhân viên

**3.4 Thủ tục thêm 1 sản phẩm mới vào bảng sản phẩm.**

CREATE OR REPLACE PROCEDURE ThemSanPhamMoi(

    Masp VARCHAR ,

    Tensp VARCHAR,

    SP\_Gia FLOAT,

    SP\_SoLuongCoSan INT

) AS

BEGIN

    INSERT INTO SanPham (MaSanPham, TenSanPham, Gia, SoLuongCoSan)

    VALUES (Masp , Tensp , SP\_Gia , SP\_SoLuongCoSan);

    COMMIT;

END;

-- thêm sp mới

begin

ThemSanPhamMoi('SP006', 'bánh macaron', 13000, 30);

end;

-- gọi lại bảng để xem kết quả

SELECT \* FROM SanPham;



Hình 3.4.1: kết quả thủ tục thêm một sản phẩm

**3.5 Thủ tục xóa nhân viên theo mã nhân viên.**

CREATE OR REPLACE PROCEDURE Xoanhanvien(Ma\_NhanVien Varchar)

IS

BEGIN

     DELETE FROM NhanVien

     WHERE MaNhanVien = Ma\_NhanVien;

     COMMIT;

END Xoanhanvien;

-- gọi thủ tục xóa

BEGIN

Xoanhanvien('NV002');

END;

-- xem kết quả

SELECT \* FROM NhanVien;



Hình 3.5.1: kết quả thủ tục xóa nhân viên

**3.6 Thủ tục tăng lương thêm 500000 cho nhân viên nào lương dưới 10 triệu.**

CREATE OR REPLACE PROCEDURE TangLuongChoNhanVien(tang number) AS

BEGIN

    -- Thêm 500000 cho nhân viên có mức lương dưới 10 triệu

    UPDATE NhanVien

    SET Luong = Luong + tang

    WHERE Luong < 10000000;

END TangLuongChoNhanVien;

-- gọi thủ tục xóa

EXECUTE TangLuongChoNhanVien (500000);

-- xem kết quả

SELECT \*FROM NhanVien;



Hình 3.6.1: Kết quả tăng lương cho nhân viên

**3.7  Viết trigger cập nhật thay đổi lương của nhân viên.**

CREATE TABLE TheoDoiCapNhatLuong (

    MaNhanVien VARCHAR(10),

    LuongCu FLOAT,

    LuongMoi FLOAT,

    ThoiGianCapNhat TIMESTAMP

);

CREATE OR REPLACE TRIGGER TheoDoiCapNhatLuongNhanVien

AFTER UPDATE OF Luong ON NhanVien

FOR EACH ROW

BEGIN

    IF UPDATING('Luong') THEN

        INSERT INTO TheoDoiCapNhatLuong (MaNhanVien, LuongCu, LuongMoi, ThoiGianCapNhat)

        VALUES (:OLD.MaNhanVien, :OLD.Luong, :NEW.Luong, SYSTIMESTAMP);

    END IF;

END;

-- Thêm một nhân viên mới vào bảng NhanVien

INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong)

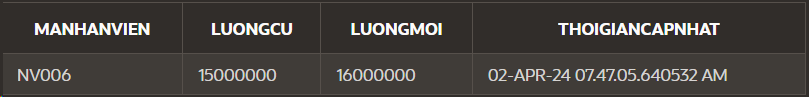
VALUES ('NV007', 'Trần Văn A', 15000000);

-- Cập nhật lương cho nhân viên vừa thêm để kích hoạt trigger

UPDATE NhanVien SET Luong = 16000000 WHERE MaNhanVien = 'NV007';

-- xem kết quả

SELECT \* FROM TheoDoiCapNhatLuong



Hình 3.7.1: bảng lưu lại kết quả cập nhật lương

**3.8  Viết trigger cập nhật nhân viên bị xóa khỏi bảng nhân viên.**

CREATE TABLE Xoa\_NhanVien(

    MaNhanVien VARCHAR(15),

    HoTen VARCHAR(30),

    ThoiGianXoa TIMESTAMP

);

CREATE OR REPLACE TRIGGER TheoDoiXoaNhanVien

AFTER DELETE ON NhanVien

FOR EACH ROW

DECLARE

    v\_ThoiGianXoa TIMESTAMP;

BEGIN

    -- Lấy thời gian hiện tại

    v\_ThoiGianXoa := SYSTIMESTAMP;

    -- Ghi nhận thông tin về nhân viên bị xóa vào bảng Xoa\_NhanVien

    INSERT INTO Xoa\_NhanVien (MaNhanVien, HoTen, ThoiGianXoa)

    VALUES (:OLD.MaNhanVien, :OLD.TenNhanVien, v\_ThoiGianXoa);

END;

-- thêm nhân viên vào

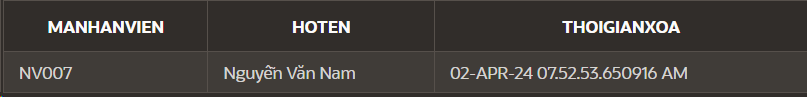
INSERT INTO NhanVien (MaNhanVien, TenNhanVien, Luong) VALUES ('NV008', 'Nguyễn Văn Nam', 12000000);

-- xóa nhân viên

DELETE FROM NhanVien WHERE MaNhanVien = 'NV008';

-- xem kết quả nhân viên bị xóa

SELECT \* FROM Xoa\_NhanVien



Hình 3.8.1: Bảng lưu lại nhân viên bị xóa

**3.9 Viết trigger để lưu lại thời gian thêm sản phẩm mới vào.**

CREATE TABLE them\_banhmoi (

    Mabanh VARCHAR(15),

    Tenbanh VARCHAR(30),

    Giabanh FLOAT,

    SoLuongbanh INT,

    ThoiGianthem TIMESTAMP

);

CREATE OR REPLACE TRIGGER TheoDoiThemBanh

AFTER INSERT ON SanPham

FOR EACH ROW

DECLARE

    v\_ThoiGianThem TIMESTAMP;

BEGIN

    -- Lấy thời gian hiện tại

    v\_ThoiGianThem := SYSTIMESTAMP;

    -- Ghi nhận thông tin về bánh mới được thêm vào

    INSERT INTO them\_banhmoi (Mabanh, Tenbanh, Giabanh, SoLuongbanh, ThoiGianthem)

    VALUES (:NEW.MaSanPham, :NEW.TenSanPham, :NEW.Gia, :NEW.SoLuongCoSan, v\_ThoiGianThem);

END;

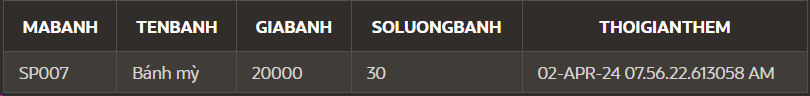
-- thêm sản phẩm mới vào

INSERT INTO SanPham (MaSanPham, TenSanPham, Gia, SoLuongCoSan)

VALUES ('SP007', 'Bánh mỳ', 20000, 30);

-- Xem kết quả

SELECT \* FROM them\_banhmoi;



Hình 3.9.1: Bảng lưu lại sản phẩm mới được thêm

**Phần III. Bài tập 3**

1. Tóm tắt câu trả lời cho các vấn đề dưới đây (Không copy & paste):
   1. **HQTCSDL là gì? Các HQTCSDL hiện nay**

- HQTCSDL là một hệ thống phần mềm cho phép người dùng định nghĩa, tạo và duy trì CSDL đồng thời cung cấp dịch vụ truy cập đến CSDL này một cách có quản lý.

- Các HQTCSDL hiện nay bao gồm: Oracle, DB2, MS SQL Server, MS Access và các phần mềm mở như MySQL, PostGreSQL,...

* 1. **Khác nhau giữa HQTCSDL và bảng tính (vd, Excel) là gì?**

|  |  |
| --- | --- |
| **HQTCSDL** | **Excel** |
| - Bảo vệ CSDL khỏi những đe dọa có chủ ý hay vô tình thông qua các biện pháp có sử dụng máy tính hoặc không có sử dụng máy tính.  - Có thể phục hồi từ toàn bộ cơ sở dữ liệu hoặc các bảng cụ thể, tùy thuộc vào nhu cầu và cấu trúc backup | - Thường phục hồi từ toàn bộ tập tin hoặc một phần của nó, không có khả năng phục hồi từng phần riêng lẻ của dữ liệu.  - Excel không có tính năng kiểm soát đồng thời như HQTCSDL. Nếu nhiều người truy cập cùng một tệp Excel, có thể xảy ra xung đột dữ liệu hoặc mất tính nhất quán. |

* 1. **DBA là ai? Nhiệm vụ?**

- DBA là viết tắt của Database Administrator, đó là một vị trí quan trọng trong các tổ chức có cơ sở dữ liệu. Một DBA có trách nhiệm quản lý, duy trì và bảo vệ.

- Nhiệm vụ: Quản lý CSDL, bảo mật dữ liệu, sao lưu và phục hồi, hỗ trợ và giám sát hiệu suất và điều chỉnh hệ thống.

* 1. **Các phương pháp bảo vệ dữ liệu**

- An toàn trong CSDL là bảo vệ CSDL khỏi những đe dọa có chủ ý hay vô tình.

- Cấp quyền là sự gán quyền cho một người dùng hay chương trình để có thể truy cập vào một hệ thống hay một đối tượng của hệ thống.

- Khung nhìn thay vì cấp quyền cho người dùng trên nhiều bảng, ta có thể gán các quyền thích hợp trên một khung nhìn định nghĩa trên bảng này.

- Sao lưu và phục hồi là quá trình chép CSDL và các tập tin nhật ký vào các thiết bị lưu trữ dự phòng một cách định kỳ.

- Toàn vẹn dữ liệu là đảm bảo rằng dữ liệu không bị thay đổi, biến đổi hoặc mất mát một cách không được ủy quyền hay không được mong muốn.

- Mật hóa dữ liệu là sự mã hóa dữ liệu bằng một giải thuật đặc biệt làm cho dữ liệu không thể đọc được nếu không có khóa giải mã.

- RAID là một công nghệ được sử dụng để tăng cường hiệu suất và độ tin cậy của hệ thống lưu trữ dữ liệu bằng cách kết hợp nhiều ổ đĩa vật lý thành một đơn vị lưu trữ logic.

* 1. **Giao dịch là gì? Tại sao phải cần GD?**

**-** Giao dịch là một hành động hay một chuỗi các hành động được thực hiện bởi một người dùng hoặc một chương trình ứng dụng.

- Tại sao cần giao dịch? vì nó giúp đảm bảo tính nhất quán và độ tin cậy của hệ thống thông tin.

* 1. **Các trạng thái của GD**

- Hoạt động (Active) Trạng thái khởi đầu; giao dịch giữ trong trạng thái

- Hoàn tất một phần (Partially Committed) Sau khi lệnh cuối cùng được thực hiện.

- Thất bại (Failed) Sau khi phát hiện rằng sự thực hiện không thể tiếp tục được nữa.

- Hủy bỏ (Aborted) Sau khi giao dịch đã bị cuộn lại (rolled back) và CSDL đã phục hồi lại trạng thái của nó trước khi khởi động giao dịch này trong khi nó đang thực hiện.

* 1. **Thuộc tính ACID là gì?**

- Thuộc tính ACID là một tập hợp các đặc điểm quan trọng để đảm bảo tính nhất quán và độ tin cậy của giao dịch trong cơ sở dữ liệu.

* 1. **Các vấn đề cạnh tranh trong môi trường đa người dùng là gì? Giải pháp khắc phục?**

- Xung đột dữ liệu: Khi nhiều người dùng cố gắng truy cập hoặc chỉnh sửa cùng một dữ liệu đồng thời, có thể xảy ra xung đột dữ liệu khi dữ liệu được thay đổi một cách không đồng nhất, gây ra sự không nhất quán hoặc mất mát dữ liệu.

- Thất bại của giao dịch: Trong một môi trường đa người dùng, giao dịch của một người dùng có thể bị ảnh hưởng bởi các giao dịch khác đang diễn ra, dẫn đến thất bại hoặc không hoàn thành giao dịch một cách thành công.

- Hiệu suất giảm sút: Sự cạnh tranh giữa các người dùng có thể dẫn đến tình trạng mất hiệu suất khi nhiều người cùng truy cập và thao tác trên cùng một hệ thống hoặc tài nguyên cùng một lúc.

Giải pháp:

- Kiểm soát đồng thời (Concurrency Control): Sử dụng các kỹ thuật kiểm soát đồng thời như khóa, giao thức đọc ghi, và kiểm soát đồng thời để đảm bảo rằng các thay đổi dữ liệu được thực hiện một cách an toàn và nhất quán.

- Quản lý giao dịch (Transaction Management): Sử dụng các cơ chế quản lý giao dịch để đảm bảo tính toàn vẹn và đồng nhất của giao dịch, bao gồm các nguyên tắc ACID.

- Phân chia tài nguyên (Resource Partitioning): Phân chia tài nguyên như cơ sở dữ liệu hoặc máy chủ thành các phần nhỏ hơn để giảm thiểu sự cạnh tranh và tăng cường hiệu suất.

* 1. **Lịch trình là gì? Lịch trình tuần tự và không tuần tự**

- Lịch trình là một chuỗi các thao tác thực hiện bởi một tập hợp các GD cạnh tranh mà vẫn đảm bảo thứ tự của các thao tác trong từng GD đơn lẻ.

- Tuần tự (serial schedule) là một lịch trình trong đó các thao tác của một GD được thực hiện liên tiếp nhau, không có bất kỳ thao tác nào của các GD khác xen vào giữa.

- Trình không tuần tự (nonserial schedule) là một lịch trình trong đó các thao tác từ một tập hợp các giao dịch cạnh tranh đan xen lẫn nhau.

* 1. **Các kỹ thuật quản lý cạnh tranh:** khóa chốt và nhãn thời gian

- Bi quan (lock)**:** là các tiếp cận thận trọng vì các phương pháp này làm cho các GD bị ngưng trệ khi chúng xảy ra xung đột với các giao dịch khác tại một thời điểm nào đó trong tương lai.

- Sử dụng Giao thức khóa 2 kỳ để giải quyết cạnh tranh

Các vấn đề khi sử dụng 2PL

Deadlock

- Khóa chết là một tình huống bế tắc khi hai hay nhiều GD đang chờ lẫn nhau để có được các khóa đang giữ bởi đối phương.

Xử lý deadlock

- Chỉ có một cách để phá bỏ tình trạng khóa chết là hủy bỏ 1 hoặc nhiều GD

Sử dụng nhãn thời gian

- Các nhãn thời gian có thể được tạo ra một cách đơn giản bằng cách sử dụng đồng hồ của hệ thống ở thời điểm GD bắt đầu hoặc bằng cách tăng một con số đếm luận lý mỗi khi một GD mới bắt đầu.

Lạc quan: 3 kỳ (đọc, kiểm tra, ghi )

Độ mịn của mục dữ liệu

- Độ mịn là kích cỡ của mục dữ liệu được chọn như làm một đơn vị bảo vệ bởi giao thức điều khiển cạnh tranh.

Phục hồi dữ liệu

Tại sao cần phục hồi DL

- Phục hồi dữ liệu là quá trình khôi phục lại thông tin từ các hệ thống lưu trữ khi dữ liệu gặp sự cố hoặc bị mất. Đây là một khía cạnh quan trọng của quản lý dữ liệu và bảo mật thông tin, vì nó đảm bảo rằng thông tin quan trọng không bị mất hoặc bị hỏng.

Điểm kiểm tra là gì (check point)

- check point là một thao tác khá rẻ, và nó thường được thực hiện khoảng 4 lần trong 1 giờ, Bằng cách này, ta sẽ không phải phục hồi nhiều hơn lượng công việc trong khoảng 15-20 phút.

**1.11 Các kỹ thuật phục hồi DL**

- Cập nhật trì hoãn.

- Cập nhật tức thì.

- Tạo trang bóng.