

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG



Báo Cáo

Hệ Quản Trị Cơ Sở Dữ Liệu

Mã Học Phần: CT205-02

SV Thực Hiện : Nguyễn Thị Nhật Thiên Lan

MSSV: B2103427

Giảng Viên Hướng Dẫn: TS. Nguyễn Thái Nghe

Học Kỳ II, Năm Học: 2023-2024

MỤC LỤC

Phần I. Bài tập 1	3
1. Mô tả	3
2. CSDL về xe ô tô như bảng sau:	3
2.1 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.	3
2.2 Viết thủ tục cho phép nhập vào 1 xe mới	8
2.3 Gọi thủ tục để Nhập liệu như bảng trên	8
2.4 Viết thủ tục cho phép cập nhật giá niêm yết của 1 xe nào đó.	9
2.5 Viết hàm trả về giá của 1 xe nào đó	9
2.6 Viết hàm trả về loại xe, hãng xe có giá cao nhất	10
2.7 Viết thủ tục để xóa 1 xe nào đó	11
2.8 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:	11
2.9 Viết trigger để theo dõi việc sửa giá trong bảng xe	13
2.10 Viết trigger để theo dõi việc xóa dữ liệu trong bảng xe	14
2.11 Tạo các người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng (quản lý, nhân viên bán xe,..)	15
Phần II. Bài Tập 2	15
1. Mô tả	15
2. Xây dựng cơ sở dữ liệu	15
2.1 Hàm Tính tổng số lượng tồn kho của tất cả sản phẩm.	19
2.2 Hàm kiểm tra sản phẩm có tồn kho hay không.	19
2.3 Hàm tính tổng lương của nhân viên.	20
2.4 Thủ tục thêm 1 sản phẩm mới vào bảng sản phẩm.	20
2.5 Thủ tục xóa nhân viên theo mã nhân viên.	20
2.6 Thủ tục tăng lương thêm 500000 cho nhân viên nào lương dưới 10 triệu.	21
2.7 Viết trigger cập nhật thay đổi lương của nhân viên.	21
2.8 Viết trigger cập nhật nhân viên bị xóa khỏi bảng nhân viên.	22
2.9 Viết trigger để lưu lại thời gian thêm sản phẩm mới vào.	23
Phần III. Bài tập 3	23
1. Tóm tắt câu trả lời cho các vấn đề dưới đây (Không copy & paste):	24
1.1 HQTCSDDL là gì? Các HQTCSDDL hiện nay	24
1.2 Khác nhau giữa HQTCSDDL và bảng tính (vd, Excel) là gì? ..	24
1.3 DBA là ai? Nhiệm vụ?	24
1.4 Các phương pháp bảo vệ dữ liệu	24
1.5 Giao dịch là gì? Tại sao phải cần GD?	24
1.6 Các trạng thái của GD	25
1.7 Thuộc tính ACID là gì?	25
1.8 Các vấn đề cạnh tranh trong môi trường đa người dùng là gì? Giải pháp khắc phục?	25

1.9 Lịch trình là gì? Lịch trình tuần tự và không tuần tự	25
1.10 Các kỹ thuật quản lý cạnh tranh: khóa chốt và nhấn thời gian	26

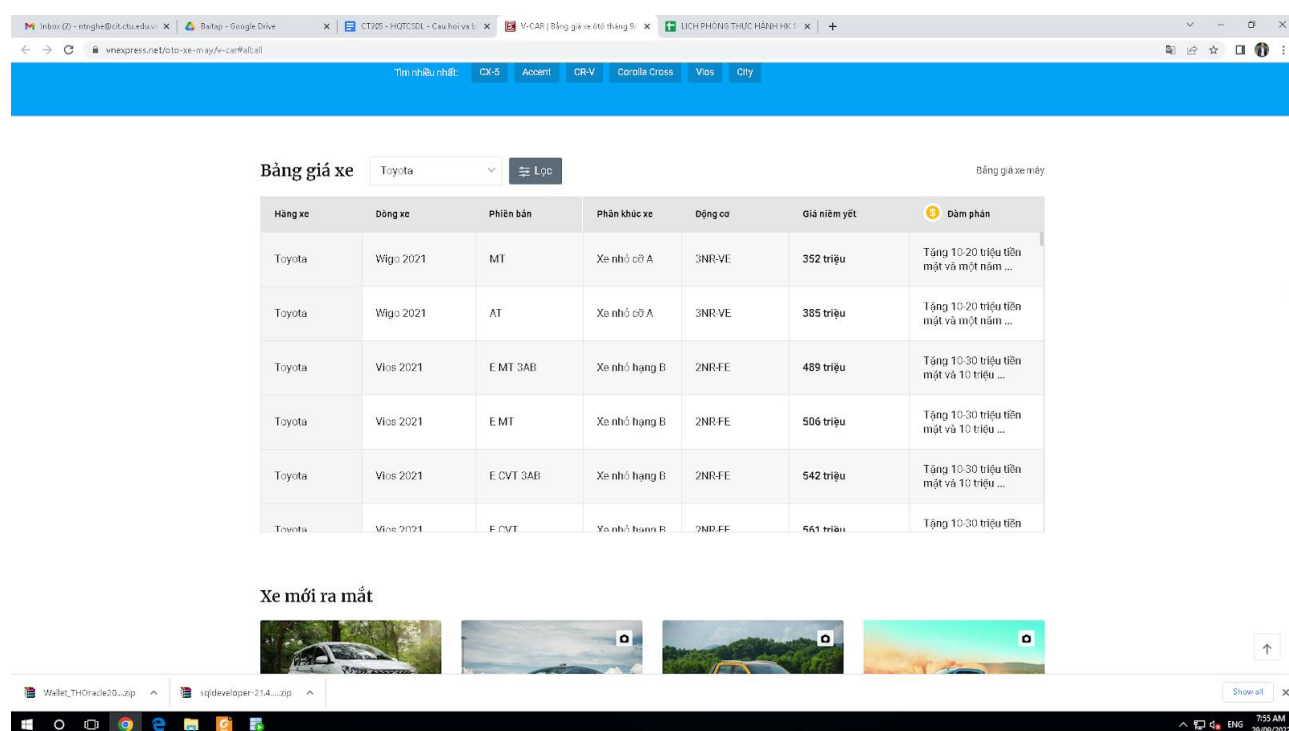
Phần I. Bài tập 1

1. Mô tả

Trong một thị trường đa dạng và phát triển như ngày nay, sự cần thiết của giá xe không thể phủ nhận. Đối với người tiêu dùng, giá xe không chỉ đơn thuần là một bản danh sách giá cả, mà còn là một công cụ hỗ trợ quan trọng trong quá trình quyết định mua xe. Việc có được một bảng giá xe chi tiết và chính xác giúp họ tiết kiệm thời gian và công sức trong việc tìm kiếm thông tin về các dòng xe, từ đó giúp họ đưa ra quyết định mua xe có tính chất thông minh và đáng tin cậy. Hơn nữa, bảng giá xe còn là một công cụ đối kháng quan trọng, giúp người tiêu dùng so sánh giữa các dòng xe và lựa chọn một mẫu xe phản ánh tốt nhất nhu cầu và ưu tiên của họ.

2. CSDL về xe ô tô như bảng sau:

<https://vnexpress.net/oto-xe-may/v-car#all;all>




Bảng giá xe Toyota

Đánh giá xe này

Hãng xe	Dòng xe	Phiên bản	Phân khúc xe	Động cơ	Giá niêm yết	Đảm bảo
Toyota	Wigo 2021	MT	Xe nhỏ cỡ A	3NR-VE	352 triệu	Tặng 10-20 triệu tiền mặt và một năm ...
Toyota	Wigo 2021	AT	Xe nhỏ cỡ A	3NR-VE	385 triệu	Tặng 10-20 triệu tiền mặt và một năm ...
Toyota	Vios 2021	E MT 3AB	Xe nhỏ hạng B	2NR-FE	489 triệu	Tặng 10-30 triệu tiền mặt và 10 triệu ...
Toyota	Vios 2021	E MT	Xe nhỏ hạng B	2NR-FE	506 triệu	Tặng 10-30 triệu tiền mặt và 10 triệu ...
Toyota	Vios 2021	E CVT 3AB	Xe nhỏ hạng B	2NR-FE	542 triệu	Tặng 10-30 triệu tiền mặt và 10 triệu ...
Toyota	Vios 2021	E CVT	Xe nhỏ hạng B	2NR-FE	561 triệu	Tặng 10-30 triệu tiền mặt và 10 triệu ...

Xe mới ra mắt



2.1 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.

- Bảng hãng xe

```
CREATE TABLE HangXe (  
    MaHang VARCHAR(15) PRIMARY KEY,
```

```
TenHangXe VARCHAR(50) NOT NULL  
);
```

- Bảng dòng xe

```
CREATE TABLE DongXe (  
    MaDong VARCHAR(15) PRIMARY KEY,  
    TenDong VARCHAR(50) NOT NULL,  
    MaHang VARCHAR(15) ,  
    FOREIGN KEY (MaHang) REFERENCES HangXe(MaHang)  
);
```

- Bảng động cơ

```
CREATE TABLE DongCo (  
    MaDongCo VARCHAR(15) PRIMARY KEY,  
    TenDongCo VARCHAR(50) NOT NULL  
);
```

- Bảng phân khúc xe

```
CREATE TABLE PhanKhucXe (  
    MaPhanKhuc VARCHAR(15) PRIMARY KEY,  
    TenPhanKhuc VARCHAR(50) NOT NULL,  
    MaDongCo VARCHAR(15),  
    FOREIGN KEY (MaDongCo) REFERENCES DongCo(MaDongCo)  
);
```

- Bảng phiên bản

```
CREATE TABLE PhienBan (  
    MaPhienBan VARCHAR(15) PRIMARY KEY,  
    TenPhienBan VARCHAR(50) NOT NULL,  
    MaPhanKhuc VARCHAR(15),  
    FOREIGN KEY (MaPhanKhuc) REFERENCES  
PhanKhucXe(MaPhanKhuc)  
);
```

- bảng giá phiên bản

```
CREATE TABLE GiaPhienBan (  
    MaGiaPhienBan INT PRIMARY KEY,  
    MaPhienBan VARCHAR(15) NOT NULL,  
    GiaNiemYet FLOAT check(GiaNiemYet >0),  
    FOREIGN KEY (MaPhienBan) REFERENCES  
PhienBan(MaPhienBan)
```

);

- Bảng nơi đăng kí

```
CREATE TABLE NoiDangKy (  
    MaTinhThanh VARCHAR(15) PRIMARY KEY,  
    TenTinhThanh VARCHAR(50) NOT NULL,  
    PhiDangKy FLOAT check(PhiDangKy >0)  
);
```

- Bảng Thông số kỹ thuật

```
CREATE TABLE ThongSoKyThuat (  
    TSKT_Ma VARCHAR(15) PRIMARY KEY,  
    MaDongCo VARCHAR(15),  
    CongSuat INT NOT NULL,  
    SoCho INT NOT NULL,  
    FOREIGN KEY (MaDongCo) REFERENCES DongCo(MaDongCo)  
);
```

- Bảng xe

```
CREATE TABLE Xe (  
    MaXe VARCHAR(15) PRIMARY KEY,  
    HangXe VARCHAR(15),  
    DongXe VARCHAR(15),  
    PhanKhucXe_Ma VARCHAR(15),  
    TSKT_Ma VARCHAR(15),  
    MaTinhThanh VARCHAR(15),  
    FOREIGN KEY (HangXe) REFERENCES HangXe(MaHang),  
    FOREIGN KEY (DongXe) REFERENCES DongXe(MaDong),  
    FOREIGN KEY (PhanKhucXe_Ma) REFERENCES  
PhanKhucXe(MaPhanKhuc),  
    FOREIGN KEY (TSKT_Ma) REFERENCES  
ThongSoKyThuat(TSKT_Ma),  
    FOREIGN KEY (MaTinhThanh) REFERENCES  
NoiDangKy(MaTinhThanh)  
);
```

- Bảng giá xe

```
CREATE TABLE GiaXe (  
    MaGiaXe INT PRIMARY KEY,  
    MaXe VARCHAR(15),  
    PhienBan_Ma VARCHAR(15),  
    DongCo_Ma VARCHAR(15),  
    GiaNiemYet FLOAT,
```

```

FOREIGN KEY (MaXe) REFERENCES Xe(MaXe),
FOREIGN KEY (PhienBan_Ma) REFERENCES
PhienBan(MaPhienBan),
FOREIGN KEY (DongCo_Ma) REFERENCES DongCo(MaDongCo)
);

-- Thêm dữ liệu cho bảng HangXe
INSERT INTO HangXe (MaHang, TenHangXe) VALUES('01', 'Toyota');
INSERT INTO HangXe (MaHang, TenHangXe) VALUES('02', 'Honda');
INSERT INTO HangXe (MaHang, TenHangXe) VALUES('03', 'Ford');

-- Thêm dữ liệu cho bảng DongXe
INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES('D001',
'Corolla', '01');
INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES('D002',
'Civic', '02');
INSERT INTO DongXe (MaDong, TenDong, MaHang) VALUES('D003',
'Fiesta', '03');

-- Thêm dữ liệu cho bảng DongCo
INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES('DC001',
'1.5L VTEC');
INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES('DC002',
'1.8L i-VTEC');
INSERT INTO DongCo (MaDongCo, TenDongCo) VALUES('DC003',
'1.6L EcoBoost');

-- Thêm dữ liệu cho bảng PhanKhucXe
INSERT INTO PhanKhucXe (MaPhanKhuc, TenPhanKhuc, MaDongCo)
VALUES('PK001', 'Sedan', 'DC001');
INSERT INTO PhanKhucXe (MaPhanKhuc, TenPhanKhuc, MaDongCo)
VALUES('PK002', 'Hatchback', 'DC002');
INSERT INTO PhanKhucXe (MaPhanKhuc, TenPhanKhuc, MaDongCo)
VALUES('PK003', 'SUV', 'DC003');

-- Thêm dữ liệu cho bảng PhienBan
INSERT INTO PhienBan (MaPhienBan, TenPhienBan, MaPhanKhuc)
VALUES ('PB001', 'Corolla 2022', 'PK001');
INSERT INTO PhienBan (MaPhienBan, TenPhienBan, MaPhanKhuc)
VALUES ('PB002', 'Civic 2022', 'PK002');

```

```
INSERT INTO PhienBan (MaPhienBan, TenPhienBan, MaPhanKhuc)
VALUES ('PB003', 'Fiesta 2022', 'PK003');
```

-- Thêm dữ liệu cho bảng GiaPhienBan

```
INSERT INTO GiaPhienBan (MaGiaPhienBan, MaPhienBan,
GiaNiemYet) VALUES (1, 'PB001', 500000000);
INSERT INTO GiaPhienBan (MaGiaPhienBan, MaPhienBan,
GiaNiemYet) VALUES (2, 'PB002', 550000000);
INSERT INTO GiaPhienBan (MaGiaPhienBan, MaPhienBan,
GiaNiemYet) VALUES (3, 'PB003', 450000000);
```

-- Thêm dữ liệu cho bảng NoiDangKy

```
INSERT INTO NoiDangKy (MaTinhThanh, TenTinhThanh, PhiDangKy)
VALUES ('29', 'Hà Nội', 1200000);
INSERT INTO NoiDangKy (MaTinhThanh, TenTinhThanh, PhiDangKy)
VALUES ('51', 'Hồ Chí Minh', 1200000);
INSERT INTO NoiDangKy (MaTinhThanh, TenTinhThanh, PhiDangKy)
VALUES ('65', 'Cần Thơ', 1000000);
```

-- Thêm dữ liệu cho bảng ThôngSốKỹThuật

```
INSERT INTO ThôngSốKỹThuật (TSKT_Ma, MaDongCo, CongSuat,
SoCho) VALUES ('TSKT001', 'DC001', 120, 5);
INSERT INTO ThôngSốKỹThuật (TSKT_Ma, MaDongCo, CongSuat,
SoCho) VALUES ('TSKT002',
INSERT INTO ThôngSốKỹThuật (TSKT_Ma, MaDongCo, CongSuat,
SoCho) VALUES ('TSKT003', 'DC003', 150, 5);
```

-- Thêm dữ liệu cho bảng Xe

```
INSERT INTO Xe (MaXe, HangXe_Ma, DongXe_Ma,
PhanKhucXe_Ma, TSKT_Ma, MaTinhThanh) VALUES('X001', '01',
'D001', 'PK001', 'TSKT001', '29');
INSERT INTO Xe (MaXe, HangXe_Ma, DongXe_Ma,
PhanKhucXe_Ma, TSKT_Ma, MaTinhThanh) VALUES('X002', '02',
'D002', 'PK002', 'TSKT002', '51');
INSERT INTO Xe (MaXe, HangXe_Ma, DongXe_Ma,
PhanKhucXe_Ma, TSKT_Ma, MaTinhThanh) VALUES('X003', '03',
'D003', 'PK003', 'TSKT003', '65');
```

-- Thêm dữ liệu cho bảng GiaXe

```
INSERT INTO GiaXe (MaGiaXe, MaXe, PhienBan_Ma, DongCo_Ma,
GiaNiemYet) VALUES (1, 'X001', 'PB001', 'DC001', 500000000);
```



```

INSERT INTO GiaXe (MaGiaXe, MaXe, PhienBan_Ma, DongCo_Ma,
GiaNiemYet) VALUES (2, 'X002', 'PB002', 'DC002', 550000000);
INSERT INTO GiaXe (MaGiaXe, MaXe, PhienBan_Ma, DongCo_Ma,
GiaNiemYet) VALUES (3, 'X003', 'PB003', 'DC003', 450000000);

```

2.2 Viết thủ tục cho phép nhập vào 1 xe mới

```

CREATE OR REPLACE PROCEDURE ThemXe(
    p_MaXe VARCHAR2,
    p_HangXe_Ma VARCHAR2,
    p_DongXe_Ma VARCHAR2,
    p_PhanKhucXe_Ma VARCHAR2,
    p_TSKT_Ma VARCHAR2,
    p_MaTinhThanh VARCHAR2
)
IS
BEGIN
    INSERT INTO Xe (MaXe, HangXe_Ma, DongXe_Ma,
PhanKhucXe_Ma, TSKT_Ma, MaTinhThanh)
    VALUES (p_MaXe, p_HangXe_Ma, p_DongXe_Ma,
p_PhanKhucXe_Ma, p_TSKT_Ma, p_MaTinhThanh);

    COMMIT;

END ThemXe;

EXECUTE ThemXe('X004', '01', 'D001', 'PK001', 'TSKT001', '29');

-- xem kết quả
SELECT * FROM Xe WHERE MaXe = 'X004';

```

2.3 Gọi thủ tục để Nhập liệu như bảng trên

```

CREATE OR REPLACE PROCEDURE ThemGiaXe(
    p_MaGiaXe INT,
    p_MaXe VARCHAR2,
    p_PhienBan_Ma VARCHAR2,
    p_DongCo_Ma VARCHAR2,
    p_GiaNiemYet FLOAT
)
IS
BEGIN
    INSERT INTO GiaXe (MaGiaXe, MaXe, PhienBan_Ma, DongCo_Ma,
GiaNiemYet)

```

```
VALUES (p_MaGiaXe, p_MaXe, p_PhienBan_Ma, p_DongCo_Ma,  
p_GiaNiemYet);
```

```
COMMIT;  
END ThemGiaXe;
```

```
EXECUTE ThemGiaXe(4, 'X004', 'PB001', 'DC001', 500000000);
```

```
-- xem kết quả  
SELECT * FROM GiaXe WHERE MaGiaXe = 4;
```

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

```
SET SERVEROUTPUT ON;  
CREATE OR REPLACE PROCEDURE CapNhatGiaNiemYet(  
    p_MaXe VARCHAR2,  
    p_GiaMoi FLOAT  
)  
IS  
BEGIN  
    UPDATE GiaXe  
    SET GiaNiemYet = p_GiaMoi  
    WHERE MaXe = p_MaXe;  
  
    COMMIT;  
    DBMS_OUTPUT.PUT_LINE('Đã cập nhật giá niêm yết của xe có mã '  
|| p_MaXe || ' thành ' || p_GiaMoi);  
EXCEPTION  
    WHEN OTHERS THEN  
        ROLLBACK;  
        DBMS_OUTPUT.PUT_LINE('Đã xảy ra lỗi khi cập nhật giá niêm  
yết của xe.');
```

```
END;  
  
EXECUTE CapNhatGiaNiemYet('X001', 800000000);
```

```
--xem kết quả  
SELECT * FROM GiaXe;
```

2.5 Viết hàm trả về giá của 1 xe nào đó

```
SET SERVEROUTPUT ON;
```

```

CREATE OR REPLACE FUNCTION LayGiaXe(
    p_MaXe VARCHAR
)
RETURN FLOAT
IS
    v_GiaNiemYet FLOAT;
BEGIN
    SELECT GiaNiemYet INTO v_GiaNiemYet
    FROM GiaXe
    WHERE MaXe = p_MaXe;

    RETURN v_GiaNiemYet;
END LayGiaXe;

BEGIN
    DBMS_OUTPUT.PUT_LINE('Giá niêm yết của xe có MaXe = X001
là: ' || LayGiaXe('X001'));
END;

```

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

```

CREATE OR REPLACE FUNCTION LayXeCoGiaCaoNhat
RETURN VARCHAR2
IS
    v_LoiXe VARCHAR2(100);
BEGIN
    SELECT DISTINCT dx.TenDong || ' - ' || hx.TenHangXe
    INTO v_LoiXe
    FROM GiaXe gx
    JOIN Xe x ON gx.MaXe = x.MaXe
        AND gx.DongXe_Ma = x.DongXe_Ma
    JOIN DongXe dx ON gx.MaDong = x.MaDong
    JOIN HangXe hx ON gx.MaHang = hx.MaHang
    WHERE gx.GiaNiemYet = (SELECT MAX(GiaNiemYet) FROM
GiaXe);

    RETURN v_LoiXe;
END LayXeCoGiaCaoNhat;

-- xem kết quả
SET SERVEROUTPUT ON;

DECLARE
    v_Xe VARCHAR2(100);

```

```

BEGIN
    v_Xe := LayXeCoGiaCaoNhat;
    DBMS_OUTPUT.PUT_LINE('Loại xe, hãng xe có giá cao nhất: ' ||
v_Xe);
END;

```

2.7 Viết thủ tục để xóa 1 xe nào đó

```

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE XoaXe(
    p_MaXe VARCHAR2
)
IS
BEGIN
    DELETE FROM Xe
    WHERE MaXe = p_MaXe;

    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Đã xóa xe có mã ' || p_MaXe);
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        DBMS_OUTPUT.PUT_LINE('Đã xảy ra lỗi khi xóa xe.');
```

```

END;

INSERT INTO Xe (MaXe, HangXe_Ma, DongXe_Ma,
PhanKhucXe_Ma, TSKT_Ma, MaTinhThanh) VALUES('X005', '03',
'D003', 'PK003', 'TSKT003', '65');

EXECUTE XoaXe('X005');

-- xem kết quả
SELECT * FROM Xe

```

2.8 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:

Giá niêm yết

Phiên bản

1.6 Premium - 659 triệu



Nơi đăng ký

Cần Thơ



Giá lăn bánh tại Cần Thơ: **728.237.000**

Dự tính chi phí

(vnd)

Giá niêm yết: 659.000.000

Phí trước bạ (10%): 65.900.000

Phí sử dụng đường bộ (01 năm): 1.560.000

Bảo hiểm trách nhiệm dân sự (01 năm): 437.000

Phí đăng kí biển số: 1.000.000

Phí đăng kiểm: 340.000

Tổng cộng: 728.237.000

SET SERVEROUTPUT ON;

```
CREATE OR REPLACE FUNCTION GiaLanBanhCanTho(  
    p_MaXe IN VARCHAR2  
)
```

```
RETURN NUMBER
```

```
IS
```

```
    v_GiaNiemYet NUMBER; -- Giá niêm yết  
    v_PhiCuocBac NUMBER; -- Phí cước bạ  
    v_PhiSuDungDuongBo NUMBER := 1560000; -- Phí sử dụng đường  
    bộ  
    v_BaoHiemTN NUMBER := 437000; -- Bảo hiểm trách nhiệm dân sự  
    v_PhiDangKy NUMBER; -- Phí đăng ký  
    v_PhiDangKiem NUMBER := 340000; -- Phí đăng kiểm  
    v_GiaLanBanh NUMBER; -- Giá lăn bánh
```

```
BEGIN
```

```
-- Lấy giá niêm yết từ bảng GiaXe
```

```
SELECT GiaNiemYet INTO v_GiaNiemYet
```

```

FROM GiaXe
WHERE MaXe = p_MaXe;

-- Tính phí cược bạ (10% giá niêm yết)
v_PhiCuocBac := v_GiaNiemYet * 0.1;

-- Lấy phí đăng ký từ bảng NoiDangKy dựa trên mã tỉnh thành '65'
SELECT PhiDangKy INTO v_PhiDangKy
FROM NoiDangKy
WHERE MaTinhThanh = '65';

-- Tính tổng các phí
v_GiaLanBanh := v_GiaNiemYet + v_PhiCuocBac +
v_PhiSuDungDuongBo + v_BaoHiemTN + v_PhiDangKy +
v_PhiDangKiem;

-- Trả về giá lăn bánh
RETURN v_GiaLanBanh;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Không tìm thấy thông tin cho xe có
mã ' || p_MaXe);
        RETURN NULL;
END;

DECLARE
    v_GiaLanBanh NUMBER;
BEGIN
    v_GiaLanBanh := GiaLanBanhCanTho('X001');
    DBMS_OUTPUT.PUT_LINE('Giá lăn bánh của xe X001 tại Cần Thơ
là: ' || v_GiaLanBanh);
END;

```

2.9 Viết trigger để theo dõi việc sửa giá trong bảng xe

```

-- Tạo bảng SuaGiaXe
CREATE TABLE SuaGiaXe (
    MaXe VARCHAR2(15),
    GiaCu NUMBER,
    GiaMoi NUMBER,
    ThoiGian TIMESTAMP
);

```

```
-- Tạo trigger Sua_Gia_Xe
CREATE OR REPLACE TRIGGER Sua_Gia_Xe
AFTER UPDATE OF GiaNiemYet ON GiaXe
FOR EACH ROW
BEGIN
    INSERT INTO SuaGiaXe ( MaXe, GiaCu, GiaMoi, ThoiGian)
    VALUES (:OLD.MaXe, :OLD.GiaNiemYet, :NEW.GiaNiemYet,
    SYSTIMESTAMP);
END;
-- Sửa giá của xe có mã là 'X001' từ 5000000000 thành 5500000000
UPDATE GiaXe SET GiaNiemYet = 5500000000 WHERE MaXe =
'X001';

SELECT * FROM SuaGiaXe
```

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

```
-- Tạo bảng XoaXe để lưu trữ thông tin về các bản ghi bị xóa
CREATE TABLE XoaXe (
    MaXe VARCHAR2(15),
    ThoiGian TIMESTAMP
);

-- Tạo trigger Xoa_Xe để theo dõi việc xóa dữ liệu trong bảng Xe
CREATE OR REPLACE TRIGGER Xoa_Xe
BEFORE DELETE ON Xe
FOR EACH ROW
BEGIN
    INSERT INTO XoaXe ( MaXe, ThoiGian)
    VALUES (:OLD.MaXe, TIMESTAMP);
END;

INSERT INTO Xe (MaXe, HangXe_Ma, DongXe_Ma,
PhanKhucXe_Ma, TSKT_Ma, MaTinhThanh) VALUES('X004', '03',
'D001', 'PK002', 'TSKT003', '65');

DELETE FROM Xe WHERE MaXe = 'X004';

--xem kết quả
SELECT * FROM XoaXe
```

2.11 Tạo các người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng (quản lý, nhân viên bán xe,..)

-- quanlyoto có quyền quản trị

```
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

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.

2. Xây dựng cơ sở dữ liệu

--Quản Lý Tiệm bán Bánh

-- 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  
);
```

-- 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  
);
```

-- 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)
);

```

-- 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 )
);

```

--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 )
);

```

-- 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 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);

```

-- Nhập dữ liệu cho bảng Khách_Hang

```

INSERT INTO Khách_Hang (MaKhachHang, TenKhachHang, Email,
SoDienThoai, DiaChi) VALUES('KH001', 'Nguyễn Văn A',
'nguyenvana@gmail.com', '0987654321', 'TP.HCM');
INSERT INTO Khách_Hang (MaKhachHang, TenKhachHang, Email,
SoDienThoai, DiaChi) VALUES('KH002', 'Trần Thị B',
'tranthib@gmail.com', '0912345678', 'TP.Cần Thơ');
INSERT INTO Khách_Hang (MaKhachHang, TenKhachHang, Email,
SoDienThoai, DiaChi) VALUES('KH003', 'Lê Thị C', 'lethic@gmail.com',
'0978123456', 'Hà Nội');
INSERT INTO Khách_Hang (MaKhachHang, TenKhachHang, Email,
SoDienThoai, DiaChi) VALUES('KH004', 'Phạm Văn D',
'phamvand@gmail.com', '0909123456', 'Huế');
INSERT INTO Khách_Hang (MaKhachHang, TenKhachHang, Email,
SoDienThoai, DiaChi) VALUES('KH005', 'Hoàng Văn E',
'hoangvane@gmail.com', '0988777666', 'Vĩnh Long');

```

-- 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);
```

-- 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);
```

-- 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);
```

-- 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);

```

2.1 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;

```

2.2 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;
```

2.3 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;
```

2.4 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
EXECUTE ThemSanPhamMoi('SP006', 'bánh macaron', 13000, 30);

-- gọi lại bảng để xem kết quả
SELECT * FROM SanPham;
```

2.5 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
EXECUTE Xoanhanvien('NV002');

-- xem kết quả
SELECT * FROM NhanVien;

```

2.6 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;

```

2.7 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

```

2.8 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
```

2.9 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.SoLuongC
oSan, 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;
```

Phần III. Bài tập 3

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

1.1 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,...

1.2 Khác nhau giữa HQTCSDL và bảng tính (vd, Excel) là gì?

HQTCSDL	Excel
<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- 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.

1.3 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.

1.4 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.

1.5 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.

1.6 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.

1.7 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.

1.8 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.

1.9 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.

1.10 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.

Các kỹ thuật phục hồi DL

- Cập nhật tri hoãn.
- Cập nhật tức thì.
- Tạo trang bóng.

