**Bài tập tổng hợp – CT205**

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

* **HQTCSDL là gì? Các HQTCSDL hiện nay**

Hệ quản trị cơ sở dữ liệu là một hệ thống phần mềm cho phép người dùng định nghĩa, tạo và duy trì CSDL, cung cấp dịch vụ truy cập đến CSDL một cách có quản lý

Các hệ quản trị cơ sở dữ liệu: Oracle, MS Sql Server, MS Access, MySQL

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

Người dùng có thể lưu trữ, truy xuất và cập nhật dữ liệu bằng cách sử dụng các ngôn ngữ định nghĩa, ngôn ngữ thao tác dữ liệu, bao gồm ngôn ngữ truy xuất dữ liệu. HQTCSDL phải hỗ trợ thực hiện các chức năng này từ một máy tính từ xa truy cập vào csdl qua mạng

* **DBA là ai? Nhiệm vụ?**

Nhà quản trị csdl. Chịu trách nhiệm đảm bảo sự hoạt động của các chức năng, tính hiệu quả của csdl và ứng dụng truy cập vào các csdl của tổ chức.

* **Các phương pháp bảo vệ dữ liệu?**

Cấp quyền, khung nhìn, sao lưu và phục hồi, toàn vẹn dữ liệu, mật hó

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

Là một hành động hay một chuỗi các hành động được thực hiện bởi một người hay một ứng dụng, trong đó có truy cập hoặc thay đổi nội dung CSDL.

Một giao dịch sẽ luôn luôn chuyển CSDL từ một trạng thái nhất quán này sang một trạng thái nhất quán khác, mặc dù chúng ta chấp nhận sự nhất quán có thể bị phá vỡ trong khi GD đang được thực hiện.

* **Các trạng thái của GD?**

Hoạt động(active), hoàn tất một phần (partially committed), thất bại (failed), hủy bỏ (Aborted), hoàn tất (committed).

* **Thuộc tính ACID là gì?**
* Tính nguyên tử (Atomicity)
* Tính nhất quán (Consistency)
* Tính cô lập (Isolation)
* Tính bền vững (Duration)
* **Các vấn đề cạnh tranh trong môi trường đa người dùng là gì?** **Giải pháp khắc phục?**

Khi hai hay nhiều người dùng cùng truy cập vào một dữ liệu và có ít nhất một người đang cập nhật dữ liệu thì có khả năng xảy ra sự can thiệp dẫn đến không nhất quán dữ liệu.

Một giao dịch có vài bước liên quan đến hoạt động I/O, các bước khác liên quan đến hoạt động CPU. Ta có thể tận dụng sự hoạt động song song của hệ thống CPU và I/O để chạy nhiều giao dịch song song nhờ tính đa chương trình.

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

Là chuỗi các thao tác được thực hiện bởi một tập hợp các giao dịch cạnh tranh mà vẫn đảm bảo thứ tự của các thao tác trong từng giao dịch đơn lẻ.

Lịch trình tuần tự: là lịch trình trong trong đó các thao tác của một giao dịch được thực hiện liên tiếp nhau, không có bất kỳ thao tác nào của một giao dịch khác xen vào giữa.

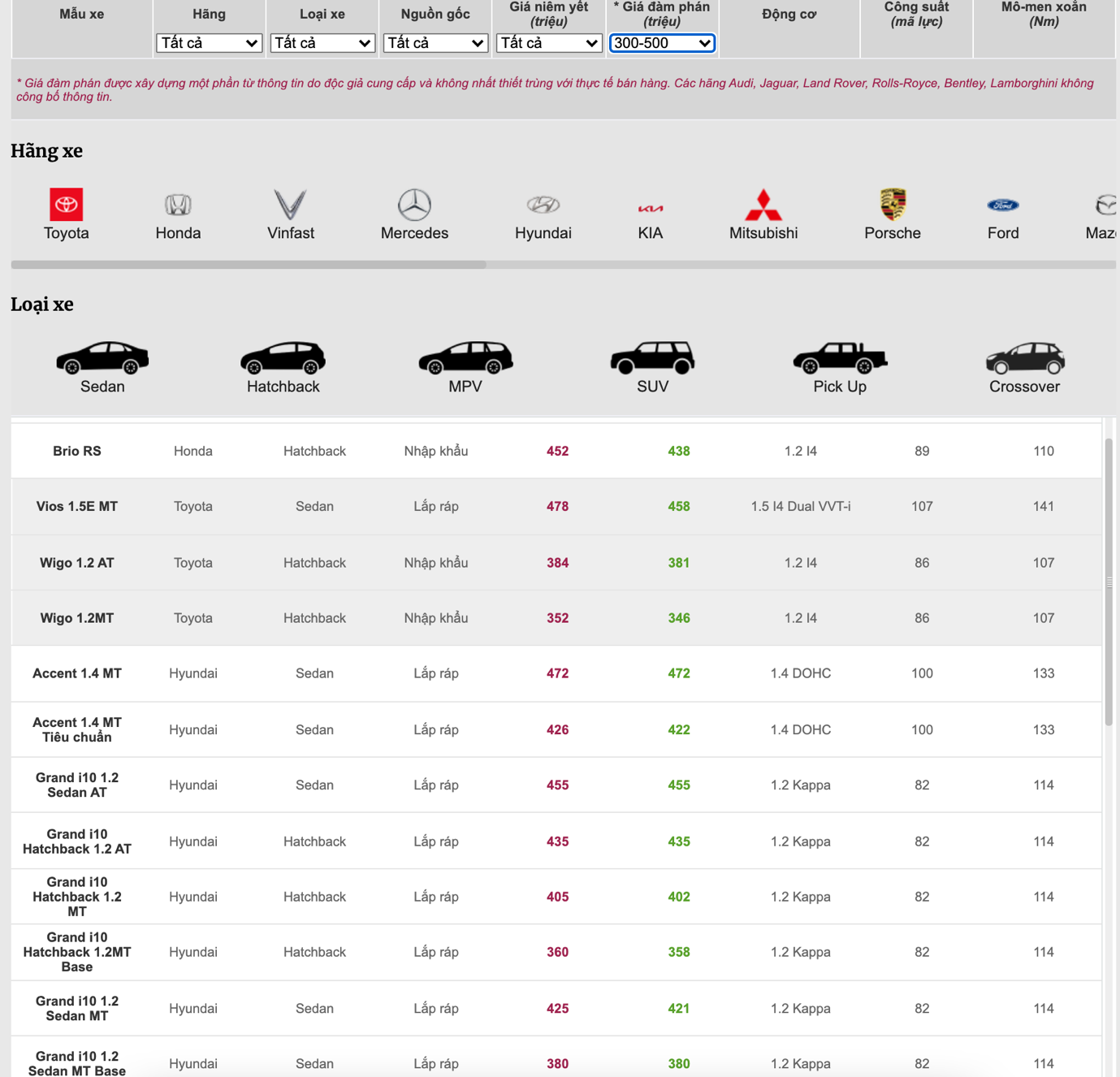
Lịch trình không tuần tự: là lịch trình trong đó các thao tác từ một tập hợp các giao dịch cạnh tranh đan xen lẫn nhau.

* **Các kỹ thuật quản lý cạnh tranh?**
* Bi quan (lock)
* Sử dụng Giao thức khóa 2 kỳ để giải quyết cạnh tranh
* Các vấn đề khi sd 2PL
* ..
* Deadlock:
  + - * + Là tình huống bế tắc khi hai hay nhiều giao dịch đang chờ lẫn nhau để có được các khóa đang giữ bởi đối phương.
        + Xử lý deadlock:
* Ngăn chặn khóa chết: ra lệnh cho các GD sử dụng các nhãn thời gian
* Phát hiện \_ phục hồi khóa chết: giải thuật phát hiện khóa chết sẽ sinh ra đồ thị vào những khoản thời gian định kỳ và kiểm tra xem có chu trình không.
* Sử dụng nhãn thời gian
* Các phương pháp khóa chốt nói chung ngăn chặn xung đột bằng cách buộc các GD phải đợi. Với các phương pháp nhãn thời gian, hoàn toàn không có chờ đợi; các GD có xung đột chỉ đơn giản bị cuộn lại và được khởi động lại
* Lạc quan: 3 kỳ (đọc, kiểm tra, ghi): Một giao dịch có 3 kỳ nếu là giao dịch cập nhật
* Kỳ đọc: kỳ này kéo dài từ lúc bắt đầu đến ngay trước khi commit.
* Kỳ kiểm tra: các kiểm tra được thực hiện để đảm bảo tính khả tuần tự không bị vi phạm khi các cập nhật GD được đưa vào csdl
* Kỳ ghi: trong kỳ này, các cập nhật đã được thực hiện trên biến cục bộ sẽ được chép vào csdl.
* Độ mịn của mục dữ liệu: Là kích cỡ của mục dữ liệu được chọn như là một đơn vị bảo vệ bởi giao thức điều khiển cạnh tranh
* **Phục hồi dữ liệu**
* Tại sao cần phục hồi DL
* Việc lưu trữ dữ liệu thông thường bao gồm bốn loại thiết bị là: bộ nhớ trong, đĩa từ, băng từ và đĩa quang. Bộ nhớ trong là bộ lưu trữ không ổn định (volatile), dữ liệu sẽ bị mất khi hệ thống có sự cố, các thiết bị còn lại là các thiết bị ổn định (non-volatile). Có rất nhiều loại lỗi có thể ảnh hưởng đến việc xử lý của CSDL, mỗi loại phải được xử lý khác nhau. Một vài lỗi chỉ ảnh hưởng đến bộ nhớ trong, trong khi các loại khác liên quan đến bộ lưu trữ ổn định. Có hai ảnh hưởng chủ yếu mà chúng ta cần quan tâm là mất dữ liệu trong bộ nhớ trong, bao gồm các vùng đệm CSDL; và mất bản sao chép của CSDL trong đĩa.
  + Điểm kiểm tra là gì (check point)
  + Là một điểm mà tại đó sự đồng bộ giữa CSDL và tập tin nhật ký GD được ghi nhận.
  + Các kỹ thuật phục hồi DL
  + Cập nhật trì hoãn: trì hoãn các cập nhật thực sự lên CSDL cho đến khi GD kết thúc thành công và đạt đến điểm hoàn tất
  + Cập nhật tức thì: CSDL có thể được cập nhật ngay bởi các thao tác của một GD, trước khi GD tiến đến điểm hoàn tất.
  + Tạo trang bóng: Cơ chế này duy trì các bảng hai-trang trong suốt quá trình thực hiện của GD: một bảng trang hiện tại và một bảng trang bóng

**CT205/CT269 – Bài tập 1**

Cho CSDL về xe ô tô như bảng sau:

<https://vnexpress.net/interactive/2016/bang-gia-xe#all;all>



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.**

--tạo bảng Hảng xe--

CREATE TABLE HangXe

(

ID\_Hang VARCHAR2(20) not null PRIMARY key,

TenHang NVARCHAR2(30) not null

);

--tạo bảng Loại xe--

CREATE TABLE LoaiXe

(

ID\_Loai VARCHAR2(20) not null PRIMARY key,

TenLoai NVARCHAR2(30) not null

);

--tạo bảng Mẫu Xe va đặt foreign key cho các thuộc tính ID\_Hang và ID\_Loai--

CREATE TABLE MauXe (

ID\_Xe VARCHAR2(20) not null PRIMARY key,

TenXe NVARCHAR2(50) not null,

ID\_Hang VARCHAR2(20) not null,

ID\_Loai VARCHAR2(20) not null,

CONSTRAINT fk\_hang

FOREIGN KEY (ID\_Hang)

REFERENCES HangXe (ID\_Hang),

CONSTRAINT fk\_loai

FOREIGN KEY (ID\_Loai)

REFERENCES LoaiXe (ID\_Loai)

);

-- tạo bảng Gía và đặt foreign key cho thuộc tính ID\_Xe--

CREATE TABLE Gia

(

ID\_Xe VARCHAR2(20) not null,

GiaNiemYet NUMBER not null,

GiaDamPhan NUMBER not null,

CONSTRAINT fk\_xe

FOREIGN KEY (ID\_Xe)

REFERENCES MauXe (ID\_Xe)

ON DELETE CASCADE

);

--tạo bảng Chi tiết xe và đặt foreign key cho thuộc tính ID\_Xe --

CREATE TABLE ChiTietXe

(

ID\_Xe VARCHAR(20) not null,

NguonGoc NVARCHAR2(50) not null,

DongCo NUMBER not null,

HopSo NVARCHAR2(30) not null,

CongSuat NUMBER not null,

MomenXoan NUMBER not null,

CONSTRAINT fk\_MauXe

FOREIGN KEY (ID\_Xe)

REFERENCES MauXe (ID\_Xe)

ON DELETE CASCADE

);

1. **Viết thủ tục cho phép nhập vào 1 xe mới:**

--thủ tục thêm Hảng xe mới

CREATE OR REPLACE PROCEDURE "INSERTNEWHANGXE"

(ID\_Hang in VARCHAR2,

TenHang in NVARCHAR2)

IS

BEGIN

INSERT INTO HangXe VALUES (ID\_Hang, TenHang);

END;

--thủ tục thêm mới Loại xe--

CREATE OR REPLACE PROCEDURE "INSERTNEWLOAIXE"

(ID\_Loai in VARCHAR2,

TenLoai in NVARCHAR2)

IS

BEGIN

INSERT INTO LoaiXe VALUES (ID\_Loai, TenLoai);

END;

-- thủ tục thêm mới Mẫu xe, Gía và Chi tiết xe--

create or replace procedure "INSERTNEWXE"

( ID\_xe in VARCHAR2,

TenXe in NVARCHAR2,

ID\_Hang in VARCHAR2,

ID\_Loai in VARCHAR2,

NguonGoc in NVARCHAR2,

DongCo in NUMBER,

HopSo in NVARCHAR2,

CongSuat in NUMBER,

MomenXoan in NUMBER,

GiaNiemYet in NUMBER,

GiaDamPhan in NUMBER )

is

BEGIN

INSERT INTO MauXe VALUES (ID\_xe, TenXe, ID\_hang, ID\_loai);

INSERT INTO Gia VALUES (ID\_xe, GiaNiemYet, GiaDamPhan);

INSERT INTO ChiTietXe VALUES(ID\_xe,NguonGoc,DongCo,HopSo,CongSuat,MomenXoan);

END;

--xoá thủ tục--

DROP PROCEDURE INSERTNEWLOAIXE;

1. **Gọi thủ tục để Nhập liệu như bảng trên**

--goi thut tuc nhap mot Hang xe moi--

BEGIN

INSERTNEWHANGXE('suzuki','SUZUKI');

INSERTNEWHANGXE('sym','SYM');

INSERTNEWHANGXE('piaggio','PIAGGIO');

INSERTNEWHANGXE('vespa','VESPA');

INSERTNEWHANGXE('yamaha','YAMAHA');

INSERTNEWHANGXE('honda','HONDA');

dbms\_output.put\_line('record inserted successfully');

END;

SELECT \* FROM HangXe;

-- goi thu tuc nhap vao mot loai xe --

BEGIN

INSERTNEWLOAIXE('contay','Xe tay con');

INSERTNEWLOAIXE('tayga','Xe tay ga');

INSERTNEWLOAIXE('xeso','Xe so');

dbms\_output.put\_line('record inserted successfully');

END;

--goi thu tuc them moi mot xe

BEGIN

INSERTNEWXE ('winner150','WINER 155 phan khoi nam 2015', 'honda','contay','Nhap khau',150,'6 so',9.9,9.9,55.5,52);

INSERTNEWXE ('ex150', 'Exciter 150 nam ', 'yamaha', 'contay', 'Nhap khau', 150, '5 so', 10.8, 9.9, 55.5, 52.5);

INSERTNEWXE ('waveA', 'Wave Apha', 'honda', 'xeso', 'Lap rap', 100, '4 so', 6.8, 7, 17.8, 17.8);

INSERTNEWXE ('waveRSX', 'Wave RSX phanh dia vanh nan hoa', 'honda', 'xeso', 'Lap rap', 110, '4 so', 8.8, 8.7, 21.5, 21);

INSERTNEWXE ('bladeSPORT', 'Blade the thao, phanh dia, vanh duc', 'honda', 'xeso', 'Lap rap', 110, '4 so', 11.6, 8.6, 21.1, 20.9);

INSERTNEWXE ('winner150RX', 'WINER 155 RX', 'suzuki', 'contay', 'Nhap khau', 150, '6 so', 9.9, 9.9, 55.5, 52);

INSERTNEWXE ('ex150RP', 'Exciter RP ', 'yamaha', 'contay', 'Nhap khau', 150, '5 so', 10.8, 9.9, 55.5, 52.5);

INSERTNEWXE ('waveAX', 'Wave AphaX', 'yamaha', 'xeso', 'Lap rap', 100, '4 so', 6.8, 7, 17.8, 17.8);

INSERTNEWXE ('waveRS', 'Wave RS phanh dia, vanh nan hoa ', 'suzuki', 'xeso', 'Lap rap', 110, '4 so', 8.8, 8.7, 21.5, 21);

INSERTNEWXE ('bladeSTANDART', 'Blade tieu chuan, phanh dia, vanh duc', 'sym', 'xeso', 'Lap rap', 110, '4 so', 11.6, 8.6, 21.1, 20.9);

dbms\_output.put\_line ('record inserted successfully');

END;

1. **Viết thủ tục cho phép cập nhật giá niêm yết và giá đàm phán của 1 xe nào đó. ​​Tất cả hàm, thủ tục đều phải có lệnh test thử.**

CREATE OR REPLACE PROCEDURE updateGia

(P\_IDxe IN Gia.ID\_Xe%TYPE,

p\_gianiemyet IN Gia.GiaNiemYet%TYPE,

p\_giamdamphan IN Gia.GiaDamPhan%TYPE)

IS

BEGIN

UPDATE Gia SET GiaNiemYet = p\_gianiemyet, GiaDamPhan = p\_giamdamphan where ID\_Xe = P\_IDxe;

COMMIT;

END;

BEGIN

updateGia('winner150',66,66.6);

END;

1. **Viết hàm trả về giá đàm phán của 1 xe nào đó**

CREATE OR REPLACE FUNCTION Get\_GiaDamPhan

( p\_IDxe VARCHAR2 )

RETURN NUMBER

is p\_GiaDamPhan NUMBER;

BEGIN

BEGIN

SELECT Gia.GiaDamPhan INTO p\_GiaDamPhan

FROM Gia

WHERE Gia.ID\_Xe = p\_IDxe;

END;

RETURN p\_GiaDamPhan;

END;

-- use function--

set serveroutput on

declare

result number;

begin

result := Get\_GiaDamPhan('winner150');

dbms\_output.put\_line(result);

end;

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

-- create function find the HangXe with the highest price --

CREATE FUNCTION GETHANGXE

RETURN NVARCHAR2

IS

HangXeMAX NVARCHAR2(30);

BEGIN

select TenHang into HangXeMAX

from MauXe join HangXe

on MauXe.ID\_Hang = HangXe.ID\_Hang

where MauXe.ID\_Xe = (select ID\_Xe from Gia

where GiaNiemYet = (select max(Gia.GiaNiemYet) from Gia)

);

RETURN HangXeMAX;

END;

--gọi hàm và sử dụng--

set serveroutput on

declare

result NVARCHAR2(30);

begin

result := GETHANGXE();

dbms\_output.put\_line(result);

end;

-- create function find the LoaiXe with the highest price --

CREATE FUNCTION GETLOAIXE

RETURN NVARCHAR2

IS

LoaiXeMAX NVARCHAR2(30);

BEGIN

select TenLoai INTO LoaiXeMAX

from MauXe join LoaiXe

on MauXe.ID\_Loai = LoaiXe.ID\_Loai

where MauXe.ID\_Xe = ( select ID\_Xe from Gia

where GiaNiemYet = (select max(Gia.GiaNiemYet) from Gia)

);

RETURN LoaiXeMAX;

END;

--gọi hàm và sử dụng--

set serveroutput on

declare

result NVARCHAR2(30);

begin

result := GETLOAIXE();

dbms\_output.put\_line(result);

end;

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

CREATE OR REPLACE PROCEDURE "DELETEXE"

(IDxe in VARCHAR2)

IS

BEGIN

DELETE from Gia where ID\_Xe = IDxe;

DELETE FROM ChiTietXe where ID\_Xe = IDxe;

DELETE from MauXe where ID\_Xe = IDxe;

END;

drop procedure DELETEXE;

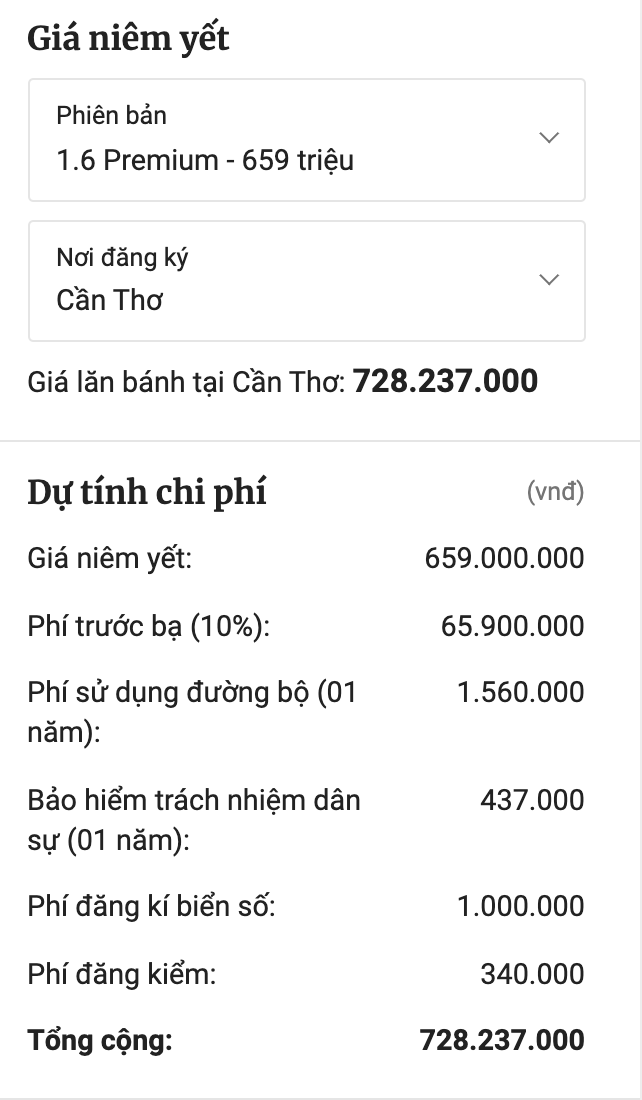
BEGIN

DELETEXE('ex150');

END;

SELECT \* FROM MauXe;

1. **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 getGiaLanBanh

(IDxe varchar2)

return number

is

all\_number number;

gia number;

begin

all\_number := 0;

gia := 0;

select GiaNiemYet into gia from Gia where ID\_Xe = IDXe and GiaNiemYet is not null;

all\_number := (gia + (gia\*0.1) + 1.560 +0.437+1+0.340)\*1000000 ;

return all\_number;

end getGiaLanBanh;

--using function

set serveroutput on

declare

result number;

begin

result := getGiaLanBanh('waveA');

dbms\_output.put\_line(result);

end;

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

-- Tao bang moi de luu thong tin--

CREATE TABLE change\_giaxe(

ID\_Xe varchar2(20),

modify\_time date,

old\_GiaNiemYet number,

new\_GiaNiemYet number,

old\_GiaDamPhan number,

new\_GiaDamPhan number

);

-- Tao trigger

create or replace trigger store\_change\_GiaXe

after update on Gia

for each row

begin

insert into change\_giaxe(ID\_Xe, MODIFY\_TIME, old\_GiaNiemYet, new\_GiaNiemYet ,old\_GiaDamPhan , new\_GiaDamPhan)

values(:old.ID\_Xe, Sysdate, :old.GiaNiemYet, :new.GiaNiemYet , :old.GiaDamPhan, :new.GiaDamPhan);

end;

-- cap nhat du lieu va xem ket qua

select \* from Gia;

update Gia

set GiaNiemYet = 100,

GiaDamPhan = 95

where ID\_Xe='waveRSX';

select \* from change\_giaxe;

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

create table delete\_mauxe(

ID\_Xe VARCHAR2(20),

TenXe NVARCHAR2(50),

modify\_time date,

ID\_Hang VARCHAR2(20),

ID\_Loai VARCHAR2(20)

)

-- tao trigger

create or replace trigger store\_delete\_mauxe

before delete on MauXe

for each row

begin

insert into delete\_mauxe(ID\_Xe, TenXe, modify\_time, ID\_Hang, ID\_Loai)

values (:old.ID\_Xe, :old.TenXe ,Sysdate,:old.ID\_Hang , :old.ID\_Loai );

end;

-- cap nhat du lieu va xem ket qua

delete MauXe

where ID\_Xe='waveRSX';

select \* from delete\_mauxe;

1. **Xóa tất cả các người dùng Test1, Test2, Test3, ... mà bạn đã tạo (Lưu ý KHÔNG XÓA NHẦM của người khác)**

DROP USER test1;

DROP USER test2;

Drop USER test3;

1. **Tạo người dùng mới để quản lý CSDL Ô tô trên, cấp quyền tương ứng**

**CREATE USER B1805825\_user1 IDENTIFIED BY "THOracle@2022";**

--cap quyen

GRANT CREATE SESSION TO B1805825\_user1

GRANT CREATE TABLE, CREATE USER, CREATE SESSION TO

B1805825\_user1 WITH ADMIN OPTION;

CT205/CT269 – Bai tap 2

1. Tao bang va nhap lieu

* Mô tả và thiết kế CSDL về một lĩnh vực nào đó (bán hàng, ql sinh viên, quản lý quán cafe, du lich, ...) gồm ít nhất 5 bảng.
* Mỗi cột trong bảng phải có các ràng buộc (Khóa chính, khóa ngoài, RB duy nhất, RB kiểm tra, not null, mặc định, biểu thức chính quy - vd Mã hàng: MH01, …)

--database quan ly ban hang--

CREATE TABLE BT\_NHANVIEN(

MaNV NVARCHAR2(10) NOT NULL CONSTRAINT PK\_NV PRIMARY KEY CHECK (REGEXP\_LIKE (MaNV,'[NV]\d\*')),

HoTen NVARCHAR2(50) NOT NULL,

NgaySinh DATE NOT NULL,

GioiTinh VARCHAR2(5)NOT NULL,

soDT VARCHAR2(15) NOT NULL

);

CREATE TABLE BT\_SANPHAM(

MaSP NVARCHAR2(10) NOT NULL CONSTRAINT PK\_SP PRIMARY KEY CHECK (REGEXP\_LIKE (MaSP,'[SP]\d\*')),

TenSP NVARCHAR2(100) NOT NULL,

NuocSX NVARCHAR2(50) NOT NULL,

Gia NUMBER NOT NULL

);

CREATE TABLE BT\_KHACHHANG(

MaKH NVARCHAR2(10) NOT NULL CONSTRAINT PK\_KHACHHANG PRIMARY KEY CHECK (REGEXP\_LIKE (MaKH,'[KH]\d\*')),

TenKH NVARCHAR2(50) NOT NULL,

SDT VARCHAR2(15) NOT NULL

);

CREATE TABLE BT\_HOADON(

MaHD NVARCHAR2(10) NOT NULL CONSTRAINT PK\_HD PRIMARY KEY CHECK (REGEXP\_LIKE (MaHD,'[HD]\d\*')),

NgayLap DATE NOT NULL,

MaKH NVARCHAR2(10) NOT NULL,

MaNV NVARCHAR2(10) NOT NULL,

TRIGIA NUMBER NOT NULL,

CONSTRAINT fk\_NHANVIEN

FOREIGN KEY (MaNV)

REFERENCES BT\_NHANVIEN (MaNV),

CONSTRAINT fk\_KHACHHANG

FOREIGN KEY (MaKH)

REFERENCES BT\_KHACHHANG (MaKH)

);

CREATE TABLE BT\_CHITIETHOADON(

MaHD NVARCHAR2(10) NOT NULL,

MaSP NVARCHAR2(10) NOT NULL,

SoLuong NUMBER NOT NULL,

CONSTRAINT fk\_HOADON

FOREIGN KEY (MaHD)

REFERENCES BT\_HOADON (MaHD) ON DELETE CASCADE

);

* Nhập dữ liệu cho bảng (ít nhất 5 dòng)
* INSERT INTO BT\_NHANVIEN
* VALUES('NV01','NGUYEN TRUNG TOAN', TO\_DATE('01/01/2000', 'DD/MM/YYYY' ), 'NAM', '0707800224');
* INSERT INTO BT\_NHANVIEN
* VALUES('NV02','NGUYEN MINH NHAN', TO\_DATE('18/01/1990','DD/MM/YYYY'), 'NAM', '0978001123');
* INSERT INTO BT\_NHANVIEN
* VALUES('NV03', 'NGUYEN LINH DANG', TO\_DATE('19/05/1985','DD/MM/YYYY'), 'NU','0899011230');
* INSERT INTO BT\_NHANVIEN
* VALUES('NV04', 'NGUYEN NGOC HOA',TO\_DATE('15/01/2000','DD/MM/YYYY'), 'NU','0900801123');
* INSERT INTO BT\_NHANVIEN
* VALUES('NV05','NGUYEN MINH NGHIA', TO\_DATE('01/06/1997','DD/MM/YYYY'), 'NAM','0988001198');
* SELECT \* FROM BT\_NHANVIEN;
* --
* INSERT INTO BT\_SANPHAM
* VALUES('SP01','IPHONE 11 PROMAX 256GB','KOREA','28500000');
* INSERT INTO BT\_SANPHAM
* VALUES('SP02','IPHONE 12 MINI 256GB','KOREA','25000000');
* INSERT INTO BT\_SANPHAM
* VALUES('SP03','IPHONE XSMAX 128GB','KOREA','15500000');
* INSERT INTO BT\_SANPHAM
* VALUES('SP04','IPHONE 13 PROMAX GOLD 1TBG','KOREA','39500000');
* INSERT INTO BT\_SANPHAM
* VALUES('SP05','IPHONE 13 PROMAX Graphite Black 256GB','KOREA','35500000');
* SELECT \* FROM BT\_SANPHAM;
* --
* INSERT INTO BT\_KHACHHANG
* VALUES('KH01','NGUYEN VAN VINH','0901234321');
* INSERT INTO BT\_KHACHHANG
* VALUES('KH02','NGUYEN THUY LINH','0901234888');
* INSERT INTO BT\_KHACHHANG
* VALUES('KH03','NGUYEN THU HONG','0989834322');
* INSERT INTO BT\_KHACHHANG
* VALUES('KH04','NGUYEN TUAN ANH','0901239891');
* INSERT INTO BT\_KHACHHANG
* VALUES('KH05','NGUYEN THI HOA','0901239876');
* SELECT \* FROM BT\_KHACHHANG;
* --
* INSERT INTO BT\_HOADON
* VALUES('HD01', TO\_DATE('16/02/2022','DD/MM/YYYY'), 'KH01', 'NV03', '85500000');
* INSERT INTO BT\_HOADON
* VALUES('HD02', TO\_DATE('10/01/2022','DD/MM/YYYY'),'KH03','NV01','50000000');
* INSERT INTO BT\_HOADON
* VALUES('HD03', TO\_DATE('01/03/2022','DD/MM/YYYY'),'KH02','NV03','45000000');
* INSERT INTO BT\_HOADON
* VALUES('HD04', TO\_DATE('11/03/2022','DD/MM/YYYY'),'KH04','NV01','39500000');
* INSERT INTO BT\_HOADON
* VALUES('HD05', TO\_DATE('01/04/2022','DD/MM/YYYY'),'KH05','NV02','142000000');
* INSERT INTO BT\_HOADON
* VALUES('HD06', TO\_DATE('02/04/2022','DD/MM/YYYY'),'KH01','NV03','35500000');
* SELECT \* FROM BT\_HOADON;
* --
* INSERT INTO BT\_CHITIETHOADON VALUES('HD01','SP01','3');
* INSERT INTO BT\_CHITIETHOADON VALUES('HD02','SP02','2');
* INSERT INTO BT\_CHITIETHOADON VALUES('HD03','SP03','3');
* INSERT INTO BT\_CHITIETHOADON VALUES('HD04','SP04','1');
* INSERT INTO BT\_CHITIETHOADON VALUES('HD05','SP05','4');
* INSERT INTO BT\_CHITIETHOADON VALUES('HD06','SP05','1');
* SELECT \* FROM BT\_CHITIETHOADON;
* Viết ít nhất 3 hàm, 3 thủ tục, 3 trigger để quản lý dữ liệu trên

--ham tra ve ten nhan vien lap nhieu hoa don nhat

CREATE OR REPLACE FUNCTION GETMAXNHANVIEN

RETURN NVARCHAR2

IS

nhanvienMAX NVARCHAR2(100);

BEGIN

SELECT Hoten INTO nhanvienMAX FROM BT\_NHANVIEN

WHERE MaNV = (

SELECT B.MaNV FROM (SELECT MaNV, COUNT(MaHD) AS SoHD FROM BT\_HOADON GROUP BY Manv) B

WHERE B.SoHD = (SELECT MAX(A.SoHD) FROM (SELECT MaNV, COUNT(MaHD) AS SoHD FROM BT\_HOADON GROUP BY Manv) A));

RETURN nhanvienMAX;

END;

set serveroutput on

declare

result NVARCHAR2(100);

begin

result := GETMAXNHANVIEN();

dbms\_output.put\_line(result);

end;

--Ham tra ve khach hang có hoa don cao nhat

CREATE OR REPLACE FUNCTION GETMAXKHACHHANG

RETURN NVARCHAR2

IS

khachhangMAX NVARCHAR2(50);

BEGIN

SELECT TenKH INTO khachhangMAX FROM BT\_KHACHHANG WHERE MaKH =(SELECT MaKH FROM BT\_HOADON WHERE TRIGIA =(SELECT MAX(TRIGIA) FROM BT\_HOADON));

RETURN khachhangMAX;

END;

set serveroutput on

declare

result NVARCHAR2(50);

begin

result := GETMAXKHACHHANG();

dbms\_output.put\_line(result);

end;

--ham tra ve san pham ban chay nhat

CREATE OR REPLACE FUNCTION GETMAXSP

RETURN NVARCHAR2

IS

sanphamMAX NVARCHAR2(100);

BEGIN

SELECT TenSP INTO sanphamMAX FROM BT\_SANPHAM WHERE BT\_SANPHAM.MaSP = (SELECT B.MaSP FROM

(SELECT MaSP,SUM(SoLuong) as tongso FROM BT\_CHITIETHOADON GROUP BY MaSP) B

WHERE B.tongso =(SELECT MAX(A.tongso) FROM

(SELECT MaSP,SUM(SoLuong) as tongso FROM BT\_CHITIETHOADON GROUP BY MaSP) A));

RETURN sanphamMAX;

END;

SET SERVEROUTPUT ON

DECLARE

RESULT NVARCHAR2(100);

BEGIN

RESULT := GETMAXSP();

dbms\_output.put\_line(result);

END;

**--thủ tục--**

-- thu tuc them san pham moi

CREATE OR REPLACE PROCEDURE INSERTNEWSP

( MaSP IN NVARCHAR2 ,

TenSP IN NVARCHAR2,

NuocSX IN NVARCHAR2,

Gia IN NUMBER)

IS

BEGIN

INSERT INTO BT\_SANPHAM VALUES (MaSP, TenSP, NuocSX, Gia);

END;

--goi thu tuc--

BEGIN

INSERTNEWSP('SP06','SAMSUNG GALAXY ZLIP3','CHINA',32000000);

INSERTNEWSP('SP07','SAMSUNG GALAXY A53 128GB','CHINA',9800000);

INSERTNEWSP('SP08','OPPO A95','CHINA',6290000);

INSERTNEWSP('SP09','OPPO RENO7 Z 5G','CHINA',9290000);

INSERTNEWSP('SP10','SAMSUNG GALAXY S22 ULTRA 128GB','CHINA',23900000);

dbms\_output.put\_line('record inserted successfully');

END;

--xoa thu tuc--

DROP PROCEDURE INSERTNEWSP;

select \* from BT\_SANPHAM;

--thu tuc cap nhat gia cua mot san pham

CREATE PROCEDURE UPDATE\_GIA\_SP

( p\_MaSP IN BT\_SANPHAM.MaSP%TYPE,

p\_Gia IN BT\_SANPHAM.Gia%TYPE

)

IS

BEGIN

UPDATE BT\_SANPHAM SET Gia = p\_Gia WHERE MaSP = p\_MaSP;

END;

--goi thu tuc

BEGIN

UPDATE\_GIA\_SP('SP01',12000000);

END;

--thu tuc xoa mot hoa don

CREATE OR REPLACE PROCEDURE DELETE\_HOADON

(p\_MaHD IN NVARCHAR2)

IS

BEGIN

DELETE FROM BT\_CHITIETHOADON WHERE MaHD = p\_MaHD;

DELETE FROM BT\_HOADON WHERE MaHD = p\_MaHD;

END;

--GOI THU TUC

BEGIN

DELETE\_HOADON('HD01');

END;

SELECT \* FROM BT\_HOADON;

SELECT \* FROM BT\_CHITIETHOADON;

**--trigger--**

CREATE TABLE CHANGE\_GIASP(

MaSP NVARCHAR2(10),

modify\_time date,

OLD\_Gia number,

NEW\_Gia number

);

CREATE OR REPLACE TRIGGER change\_GiaSP

AFTER UPDATE ON BT\_SANPHAM

FOR EACH ROW

BEGIN

INSERT INTO change\_GIASP(MaSP, MODIFY\_TIME, OLD\_Gia, NEW\_Gia)

VALUES(:old.MaSP, TO\_DATE(Sysdate,'DD/MM/YYYY'), :old.Gia, :new.Gia);

END;

DROP TRIGGER change\_GiaSP;

UPDATE BT\_SANPHAM SET Gia = 8090000 WHERE MaSP = 'SP02';

SELECT \* FROM change\_GIASP;

--TRIGGER delete sanpham

CREATE TABLE DELETE\_SANPHAM(

MaSP NVARCHAR2(10),

TenSP NVARCHAR2(100),

MODIFY\_TIME DATE

);

CREATE OR REPLACE TRIGGER DELETE\_SP

AFTER DELETE ON BT\_SANPHAM

FOR EACH ROW

BEGIN

INSERT INTO DELETE\_SANPHAM(MaSP, TenSP, MODIFY\_TIME)

VALUES(:old.MaSP, :old.TenSP,TO\_DATE(Sysdate,'DD/MM/YYYY'));

END;

DELETE FROM BT\_SANPHAM WHERE MaSP = 'SP04';

SELECT \* FROM DELETE\_SANPHAM;

--TRIGGRT CAP NHAT CHI TIET HOA DON

CREATE TABLE change\_HOADON(

MaHD NVARCHAR2(10),

OLD\_MaKH NVARCHAR2(50),

NEW\_MaKH NVARCHAR2(50),

MODIFY\_TIME DATE

);

CREATE OR REPLACE TRIGGER UPDATE\_HOADON

AFTER UPDATE ON BT\_HOADON

FOR EACH ROW

BEGIN

INSERT INTO change\_HOADON(MaHD,OLD\_MaKH,NEW\_MaKH, MODIFY\_TIME)

VALUES(:old.MaHD, :old.MaKH, :new.MaKH, TO\_DATE(Sysdate,'DD/MM/YYYY'));

END;

UPDATE BT\_HOADON SET MaKH = 'KH02' WHERE MaHD = 'HD02';

Tất cả lý thuyết tóm tắt trên và code đưa vào 1 file word/pdf (tên file là BaitapCK) nộp lên Drive trước tuần **14/04/2022**