



NGÔN NGỮ PL / SQL: PROCEDURE, FUNCTION, TRIGGER

Bài thực hành này giúp sinh viên cài đặt PL / SQL phần kiến thức liên quan đến Procedure, Function và Trigger nâng cao.

I. Tóm tắt bài thực hành

1.1. Yêu cầu lý thuyết

Sinh viên đã được trang bị kiến thức:

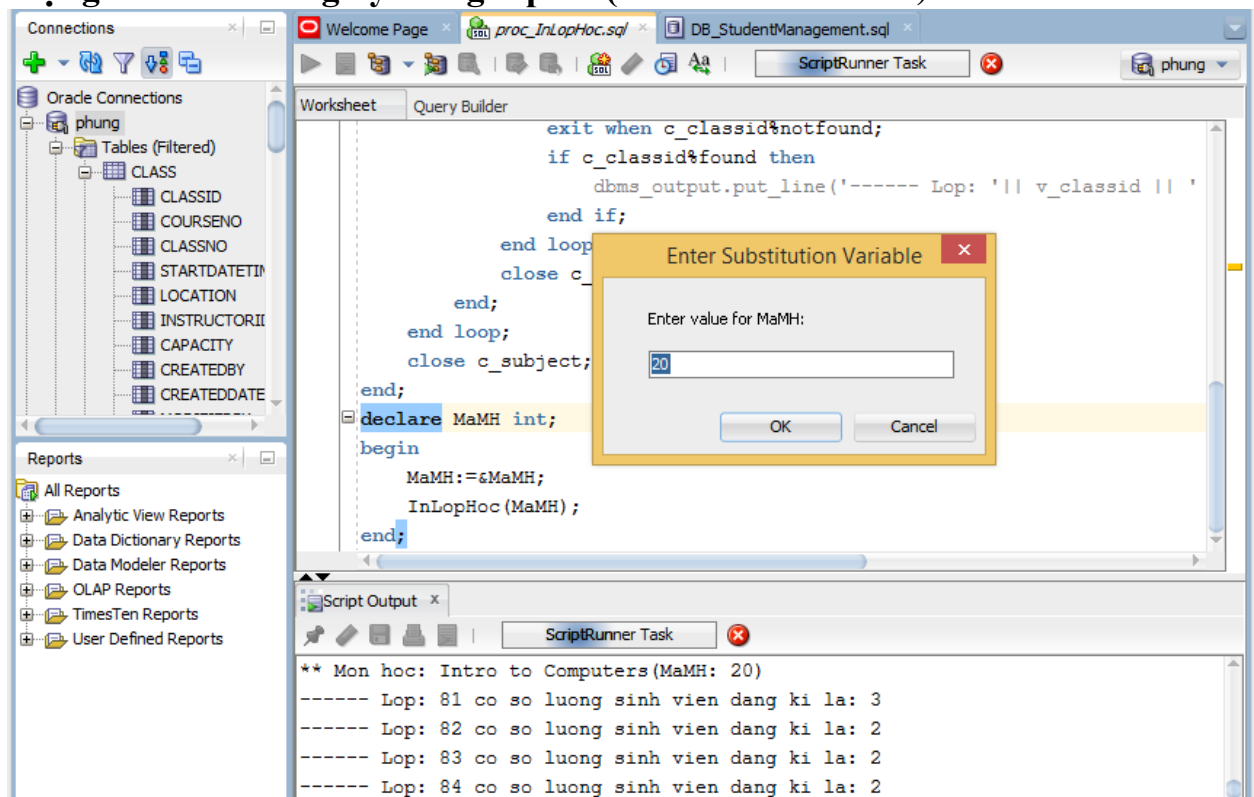
- Các kiến thức cơ bản ngôn ngữ PL / SQL.
- Các khái niệm: Procedure, Function và Trigger trong Oracle.

1.2. Nội dung

- ❖ **Sử dụng công cụ Oracle SQL Developer 4 R2 xây dựng các Procedure, Function và Trigger**
 - Lồng ghép Procedure, Function và Cursor.
 - Cài đặt các trigger, mutating trigger và compound trigger.
- ❖ **Một số lưu ý**
 - Sinh viên cần nắm rõ các vấn đề liên quan đến trigger, mutating trigger và compound trigger.

II. Thao tác từng bước

2.1. Sử dụng CSDL Quản lý sinh viên, xây dựng thủ tục cho phép nhập vào mã số môn học và in ra danh sách các lớp học của môn học này và số lượng sinh viên đăng ký trong lớp đó (xem hình bên dưới):



Nhận xét:

- Yêu cầu bài toán trên có liên quan đến việc xây dựng thủ tục **Procedure**.
- Để in ra kết quả theo yêu cầu của bài toán thì cần phải sử dụng con trỏ **Cursor**.

○ Viết code

```

set serveroutput on;
create or replace procedure InLopHoc(MaMH int) is
  cursor c_subject is select courseno,description from course where courseno=MaMH;
  v_courseno course.courseno%type;
  v_description course.description%type;
begin
  open c_subject;
  loop
    fetch c_subject into v_courseno,v_description;
    exit when c_subject%notfound;
    dbms_output.put_line('** Mon hoc: ' || v_description || '(MaMH: ' ||v_courseno||')');
  declare
    cursor c_classid is select c.classid,count(*)
      from class c,enrollment e
      where c.classid=e.classid
      and c.courseno=v_courseno
      group by c.classid;
    v_classid class.classid%type;
    v_total number(10);
  begin
    open c_classid;
    loop
      fetch c_classid into v_classid,v_total;
      exit when c_classid%notfound;
      if c_classid%found then
        dbms_output.put_line('-- Lop ' || v_classid || ': co so luong sinh vien dang ky la: '|| v_total);
      end if;
    end loop;
    close c_classid;
  end;
end loop;
close c_subject;
end;

```

○ Thực thi thủ tục

```

declare
  MaMH int;
begin
  MaMH:=&MaMH;
  InLopHoc (MaMH) ;
end;

```

2.2. Sử dụng CSDL Quản lý giáo vụ, xây dựng ràng buộc trigger sao cho mỗi sinh viên đăng ký không được quá 4 lớp học:

The screenshot displays the Oracle SQL Developer interface. The top pane shows the following SQL code:

```

insert into ENROLLMENT (STUDENTID, CLASSID) values (214, 101);
-- Them khong bi loi: do chua vuot qua 5 lop
insert into ENROLLMENT (STUDENTID, CLASSID) values (250, 101);

-- Kiem tra lai cac gia tri thu nghiem
select * from ENROLLMENT
where STUDENTID=214;
select * from ENROLLMENT
where STUDENTID=250;
-- Xoa cac bo gia tri da thu nghiem
delete from ENROLLMENT

```

The bottom pane shows two query results. The first result, for STUDENTID=214, shows 4 rows of enrollment data, which are highlighted with a red box. The second result, for STUDENTID=250, shows 3 rows of enrollment data, which are also highlighted with a red box.

Query Result 1: STUDENTID=214

STUDENTID	CLASSID	ENROLLDATE	FINALGRADE	CREATEDBY	CREATEDDATE	MODIFIEDBY	MODIFIEDDATE
214	123	13-FEB-99	(null)	DSCHERER	14-DEC-99	BROSENZW	05-JAN-00
214	135	13-FEB-99	(null)	DSCHERER	14-DEC-99	BROSENZW	05-JAN-00
214	146	13-FEB-99	(null)	DSCHERER	14-DEC-99	BROSENZW	05-JAN-00
214	156	13-FEB-99	(null)	DSCHERER	14-DEC-99	BROSENZW	05-JAN-00

Query Result 2: STUDENTID=250

STUDENTID	CLASSID	ENROLLDATE	FINALGRADE	CREATEDBY	CREATEDDATE	MODIFIEDBY	MODIFIEDDATE
250	126	19-FEB-99	(null)	DSCHERER	14-DEC-99	BROSENZW	05-JAN-00
250	146	19-FEB-99	(null)	DSCHERER	14-DEC-99	BROSENZW	05-JAN-00
250	154	19-FEB-99	(null)	DSCHERER	14-DEC-99	BROSENZW	05-JAN-00

Annotations:

- Sinh viên có mã **214** đã đăng ký đủ 4 lớp nên sẽ không được đăng ký thêm lớp thứ 5.
- Sinh viên có mã **250** chỉ mới đăng ký 3 lớp nên sẽ được đăng ký thêm lớp thứ 4.

○ Viết code

```
set serveroutput on;
create or replace trigger trg_EnrollMent_Insert
before insert on ENROLLMENT
for each row
declare
    totalClass int;
begin
    select count(CLASSID) into totalClass
    from ENROLLMENT
    where STUDENTID=:new.STUDENTID;
    if (totalClass>=4) then
        RAISE_APPLICATION_ERROR(-20000, 'Tong so lop cua sinh vien da qua 5');
    else
        DBMS_OUTPUT.PUT_LINE('Da them thanh cong');
    end if;
end;
```

Đếm số lớp mà sinh viên này đã đăng ký

Lệnh **RAISE_APPLICATION_ERROR()** dùng để huỷ thao tác thêm khi điều kiện không thoả.

○ Thực thi kiểm tra trigger

```
-- Them bi loi: do vuot qua 5 lop
insert into ENROLLMENT(STUDENTID,CLASSID) values (214,101);
-- Them khong bi loi: do chua vuot qua 5 lop
insert into ENROLLMENT(STUDENTID,CLASSID) values (250,101);
```

○ Nhận xét

- Đoạn lệnh trên chỉ mới ràng buộc cho câu lệnh Insert.
- Cần phải viết ràng buộc trên câu lệnh Update.

2.3. Tìm hiểu và cài đặt mutating trigger và compound trigger:

- Xây dựng quan hệ Employee như sau:

```
create table employee
(
  id int primary key,
  name nvarchar2(50),
  salary int,
  id_manager int references employee(id)
);
delete from employee;
insert into employee values(1,'Phong',500,null);
insert into employee values(2,'Tri',800,1);
insert into employee values(3,'Nguyen',800,1);
insert into employee values(4,'Thanh',900,2);
select * from employee;
```

- Lần lượt cài đặt và nhận xét kết quả thực thi của các trigger cho từng trường hợp sau:
 - ✓ Trường hợp 1:

```
set serveroutput on;
CREATE OR REPLACE TRIGGER TRG_TEST
AFTER UPDATE OF salary ON employee
FOR EACH ROW
DECLARE
  v_Count NUMBER;
BEGIN
  SELECT count(*)
  INTO v_count
  FROM employee
  WHERE salary =1000;
  dbms_output.put_line('Total employee are ' || v_count);
END;

update employee
set salary=1000
where salary=800;
```

Lưu ý: đoạn lệnh này có sử dụng câu lệnh **FOR EACH ROW**

✓ Trường hợp 2:

```
set serveroutput on;
CREATE OR REPLACE TRIGGER TRG_TEST
AFTER UPDATE OF salary ON employee
DECLARE
    v_Count NUMBER;
BEGIN
    SELECT count(*)
    INTO v_count
    FROM employee
    WHERE salary =1000;
    dbms_output.put_line('Total employee are ' || v_count);
END;

update employee
set salary=1000
where salary=800;
```

✓ Trường hợp 3:

```
CREATE OR REPLACE TRIGGER TRG_TEST
AFTER UPDATE OF salary ON employee
FOR EACH ROW
DECLARE
    PRAGMA AUTONOMOUS_TRANSACTION;
    v_Count NUMBER;
BEGIN
    SELECT count(*) INTO v_count
    FROM employee
    WHERE salary = 1000;
    dbms_output.put_line('Total employee are ' || v_count);
END;

update employee
set salary=1000
where salary =800;
```

Lưu ý: đoạn lệnh này có sử dụng câu lệnh
PRAGMA AUTONOMOUS_TRANSACTION;

✓ Trường hợp 4:

```

CREATE OR REPLACE TRIGGER TRG_TEST
FOR UPDATE ON employee
COMPOUND TRIGGER
/* Declaration Section*/
v_count NUMBER;
AFTER EACH ROW IS
BEGIN
    dbms_output.put_line('Update is done');
END AFTER EACH ROW;

AFTER STATEMENT IS
BEGIN
    SELECT count(*) INTO v_count
    FROM employee
    WHERE salary = 1000;
    dbms_output.put_line('Total employee are ' || v_count);
END AFTER STATEMENT;
END TRG_TEST;

update employee
set salary=1000
where salary =800;

```

Lưu ý: đoạn lệnh này có sử dụng câu lệnh **COMPOUND TRIGGER**;

III. Bài tập yêu cầu

Sử dụng CSDL Quản lý bán hàng (tập tin đính kèm trên hệ thống), xây dựng các câu trigger sau:

- 1) Ngày mua hàng (NGHD) của một khách hàng thành viên sẽ lớn hơn hoặc bằng ngày khách hàng đó đăng ký thành viên (NGDK).
- 2) Ngày bán hàng (NGHD) của một nhân viên phải lớn hơn hoặc bằng ngày nhân viên đó vào làm.
- 3) Mỗi một hóa đơn phải có ít nhất một chi tiết hóa đơn.
- 4) Trị giá của một hóa đơn là tổng thành tiền (số lượng*đơn giá) của các chi tiết thuộc hóa đơn đó.
- 5) Doanh số của một khách hàng là tổng trị giá các hóa đơn mà khách hàng thành viên đó đã mua.

IV. Câu hỏi thường gặp

- Trường hợp nào sử dụng Mutating Trigger và Compound Trigger?