CSE 4508 – RDBMS Programming Lab

Lab 8

Group 1B

Instructor: Shahriar Ivan

Materials: Prof. Dr. Abu Raihan Mostafa Kamal

Suppose you have an entity called “Student” which has the attributes ID, Name and Grade. Grade can be integer values from 0 (Kindergarten) to 12 (HSC). Also, you have a relationship called “Friends” where a Student may be a friend of another Student. Now, complete the following tasks:

A. Write a trigger that automatically deletes a student when they graduate, i.e. when their grade is updated to a value greater than 12. Write another trigger so that when a student is moved ahead one grade, all their friends are also moved ahead one grade.

B. Write one or more triggers to maintain symmetry in friend relationships. Specifically, if (A, B) is inserted into the Friends table, then (B, A) should be inserted too. Also, if (A, B) is deleted from Friends table, then (B, A) should be deleted too.

TASK 1

— —Done in the lab

```
set serveroutput on;


create user ork identified by ork;
grant all privileges to ork;


create table Student(
    id int,
    name varchar(20),
    grade int, --check(grade>=0 and grade <=12),
    constraint id_pk primary key(id)
);
create table Friends(
    id1 int,
    id2 int,
    constraint rel primary key(id1,id2)
);



insert into Friends values(1,2);
insert into Friends values(1,3);
insert into Friends values(1,4);
insert into Friends values(1,5);
insert into Friends values(1,6);
insert into Friends values(7,8);
```

```
insert into Friends values(7,9);

insert into Friends values(7,10);



insert into Student values (1,'ash',10);

insert into Student values (2,'asha',10);

insert into Student values (3,'ashar',10);

insert into Student values (4,'ashari',10);

insert into Student values (5,'asharin',10);

insert into Student values (6,'asharina',10);

insert into Student values (7,'asma',8);

insert into Student values (8,'asman',8);

insert into Student values (9,'asmani',8);

insert into Student values (10,'asmanik',8);


create or replace trigger removeGraduate

after update

on Student

begin

    delete from Student where grade>12;

end;

/


update Student set grade = 13 where id= 2;

select * from Student;
```

```
create or replace trigger updateFriends

after update

on Student

for each row

declare

    PRAGMA AUTONOMOUS_TRANSACTION;

    cid int;

begin

    cid:= :new.id;

    update Student set grade=grade+1 WHERE id in (SELECT id2 from Friends
where id1=cid);

    commit;

end;

/


update Student set grade=grade+1 where id=1;

select * from Student;
```

TASK 2

— —Done later

```
-- task2


create or replace trigger F1_insert
after insert on Friends
for each row
declare
pragma autonomous_transaction;
begin
   insert into Friends values (:new.id2 , :new:id1);
end;
/


insert into Friends values(5,7)
select * from Friends;



create or replace trigger F1_Del
after delete on Friends
for each row
declare
pragma autonomous_transaction;
begin
   delete from Friends
   where (id1 = :old.id2 and id2 =:old.id1);
   commit;
end;
/
```

```
delete from Friends where id1=5 and id2=7;

select * from Friends;
```