Chapter 12

Exercise 12.1

Given the relational schema:

EMPLOYEE (Name, Salary, DeptNum)
DEPARTMENT(DeptNum, ManagerName)

define the following active rules in Oracle and DB2

1) A rule that deletes all the employees belonging to a department when that department is deleted.

2) A rule that reacts to the deletion of the employee who is manager in a department by deleting that department and all its employees.

3) A rule that, each time the salary of an employee becomes higher than that of his or her manager, makes that salary equal to that of the manager.

4) A rule that, each time the salaries are modified, verifies that there are no departments in which the average salary increases more that three percent, and in this case cancels the modification.

5) A rule that, each time the salaries are modified, verifies their average and if it is higher than 50 thousand, deletes all the employees whose salary has been modified and are higher than 80 thousand.

Sol:

The first 4 triggers have the same syntax both in Oracle and DB2

```
1)   create trigger T1
     after delete on Department
     for each row
     when (Employee.DeptNum=Old.DeptNum)
     delete from Employee where DeptNum=Old.DeptNum

2)   create trigger T2
     after delete on Employee
     for each row
     when (old.Name in ( select ManagerName
                         from Department )
     begin
          delete from Employee where DeptNum=Old.DeptNum;
          delete from Department where DeptNum=old.DeptNum;
     end
```

```
3)   create trigger T3
     after update of Salary on Employee
     for each row
          declare x number;
          begin
               select Salary into x
               from Employee join Department on
                    Name=ManagerName
               where Department.DeptNum=new.DeptNum
          if new.salary > x then
               update Employee set Salary=x
               where Name=new.Name

4)   create trigger T4
     after update of Salary on Employee
     for each row
     declare x number;
     declare y number;
     declare l number;
     begin
          select avg(salary), count(*) into x,l
          from Employee
          where DeptNum=new.DeptNum;
          y=((x*l)-new.Salary+old.Salary)/l;
          if (x>(y*1.03)) then
               update Employee set Salary=old.Salary
               where DeptNum=new.DeptNum;
     end

5)   (ORACLE)
     create trigger T5
     after update of Salary of Employee
     for each statement
     when ((select avg(Salary) from new_table) > 50000)
     delete from Employee
     where Salary>80000
     and Name in ( Select new_table.Name
                   from new_table an n join old_table as o
                       on n.Name=o.Name
                   where n.Salary<>old.Salary )

     (DB2)
     create trigger T5
     after update of Salary of Employee
     when ((select avg(Salary) from new_table) > 50000)
     delete from Employee
     where Salary>80000
     and Name in ( Select new_table.Name
                   from new_table an n join old_table as o
                       on n.Name=o.Name
                   where n.Salary<>old.Salary )
```

Exercise 12.2

Referring to the active database system in Exercise 12.1, consider a database state with eight employees: Glenna, Mary, Tom, Bob, Andrew, Gary, Sandro, Clara, in which:

- Glenna is manager of department 1;
- Mary is manager of department 2, in which Tom and Andrew work;
- Gary is manager of department 3, in which Sandro and Clara work
- Bob is manager of department 4;

Describe a SQL transaction that deletes the employee Glenna and then modifies some of the employees' salaries, thus activating rules 3-5. Describe the behaviour of triggers after these modifications; describe the state of the database after each statement and rule execution and at the end of the transaction.

Sol:

Initial state of the database:

EMPLOYEE

Name	Salary	DeptNum
Glenna	50000	1
Mary	50000	2
Tom	40000	2
Bob	50000	4
Andrew	40000	2
Gary	50000	3
Sandro	40000	3
Clara	40000	3

DEPARTMENT

DeptNum	ManagerName
1	Glenna
2	Mary
3	Gary
4	Bob

SQL transaction:

```
delete from Employee where Name=”Glenna”
update Employee set Salary=55000 where Name=”Sandro”
update Employee set Salary=85000 where Name=”Mary”
```

The first operation causes the activation of trigger T2, which will delete from table DEPARTMENT the tuple (1,Glenna). :

New state of the database:

EMPLOYEE

Name	Salary	DeptNum
Mary	50000	2
Tom	40000	2
Bob	50000	4
Andrew	40000	2
Gary	50000	3
Sandro	40000	3
Clara	40000	3

DEPARTMENT

DeptNum	ManagerName
2	Mary
3	Gary
4	Bob

The second operation causes the activation of trigger T3, and so the tuple (Sandro, 50000, 3) becomes (Sandro, 50000, 3)

New state of the database (table DEPARTMENT does not change)

EMPLOYEE

Name	Salary	DeptNum
Mary	50000	2
Tom	40000	2
Bob	50000	4
Andrew	40000	2
Gary	50000	3
Sandro	50000	3
Clara	40000	3

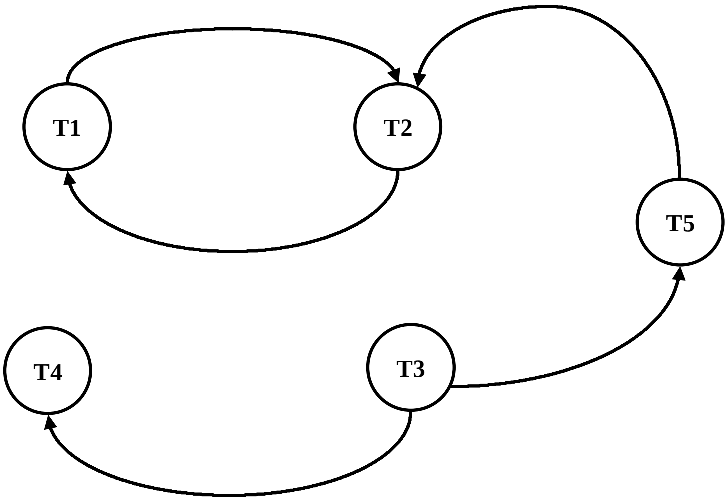
The last operation can cause the activation of triggers T4 and T5. The trigger T4 is activated first, because T5 is an after-statement trigger, while T4 is an after-row trigger. The modification is cancelled, and the final state of the database does not change.

Exercise 12.3

Discuss the properties of termination, confluence and observable determination for the rules of Exercise 12.1.

Sol:

To discuss the properties of termination for the rules of Exercise 12.1, first of all we have to examine the activation graph:



The graph is cyclic, but however the set of rules guarantees termination because T1 and T2 can activate each other only once; the second activation does not produce any effect in the database because all the employees have already been removed, and so does not activate other triggers.

The set of rules does not guarantee confluence, because the same event can activate more than one trigger (for example an increase of salary for an employee can activate trigger T3, T4 and T5).
The final result is different if the trigger T5 is activated before or after the other triggers (it deletes an employee, while T4 and T5 update the salary).

Of course the set of rules does not guarantee observable determination because it does not guarantee confluence.

Exercise 12.4

Given the relational schema:

STUDENT(Name, Subject, SuperVisor)
PROFESSOR(Name, Subject)
COURSE(Title, Professor)
EXAM(StudentName, CourseTitle)

describe the triggers that manager the following integrity constraints (business rules);

1) Each student must work in the same area of his or her supervisor.
2) Each student must have taken at least three courses in the subject of his or her supervisor.
3) Each student must have taken the exam for the course taught by his or her supervisor.

Sol:

```
1)   create trigger T1
     after update of Subject on Student
     for each row
     when Subject <> select Subject
                     from Professor
                     where Professor.Name=new.SuperVisor
     then signal SQLSTATE 70005 ( “Wrong Subject” )
```

```
2)   create trigger T2
     after update of Subject on Student
     for each row
     when 3 < ( select count(*)
                from Exam join Course on CourseTitle=Title
                    join Professor on Professor=Professor.Name
                    join Student on StudentName=Student.Name
                    join Professor as P2 on SuperVisor=P2.Name
                where Professor.Subject=P2.Subject
                    and student.Name=new.Name )
     then signal SQLSTATE 70006 (“Two few Courses”)
```

```
3)   create trigger T3
     after update of SuperVisor on Student
     for each row
     when not exist ( select *
                      from Exam join Course on CourseTitle=Title
                      where StudentName=new.Name and
                         Professor=new.Supervisor )
     then signal SQLSTATE 70007 (“Exam loss”)
```