IT344 DBMS

Assignment No. : 01

Late Date for Submission: ***28 November 2015***

Total Marks: 10
Total weightage: 5

NOTE: Write your answers on this word document and submit it in the appropriate submission folder. Copying & late submission will result in ZERO.

Q1: Consider the following database. (5 marks)

EMPLOYEE

Emp_No	**Emp_Name**	**Emp_Address**	**Emp_Salary**	**Dept_No**

DEPARTMENT

Dept_No	**Dept_Name**	**Dept_Location**	**Total_Salary**

Note that **Dept_No** in **EMPLOYEE** table is foreign key. **Total_Salary** in **DEPARTMENT** table is a derived attribute and it keeps the total salary of all the employees working in a specific department. On the basis of Active Database Concepts, answer the following questions:

A. **Identify all possible *events* which may change the value of Total_Salary attribute.**
B. **Write the active rules (triggers) for any two events you have identified in part A of this question. Use syntax of creating triggers SQL.**

Solution:

A. There are four possible events given under:

a) Inserting (one or more) new employee records in employee table
b) Changing the salary of (one or more) existing employees in employee table
c) Changing the assignment of existing employees from one department to another department in employee table
d) Deleting (one or more) employee records from employee table

B. Active rules for above events are given under:
a)

```
CREATE TRIGGER Total_Salary_01
AFTER INSERT ON EMPLOYEE
FOR EACH ROW
WHEN ( NEW.Dept_No IS NOT NULL )
UPDATE DEPARTMENT
```

```
SET Total_Salary = Total_Salary + NEW.Emp_Salary
WHERE Dept_No = NEW.Dept_No;
```

b)

```
CREATE TRIGGER Total_Salary_02
AFTER UPDATE OF Emp_Salary ON EMPLOYEE
FOR EACH ROW
WHEN ( NEW.Dept_No IS NOT NULL )
UPDATE DEPARTMENT
SET Total_Salary = Total_Salary + NEW.Emp_Salary – OLD.Emp_Salary
WHERE Dept_No = NEW.Dept_No;
```

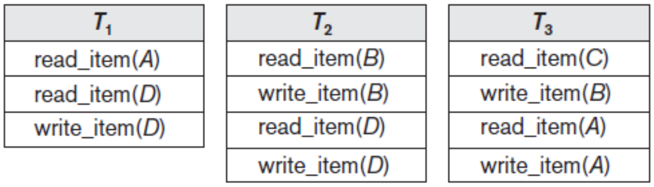
c)

```
CREATE TRIGGER Total_Salary_03
AFTER UPDATE OF Dept_No ON EMPLOYEE
FOR EACH ROW
BEGIN
UPDATE DEPARTMENT
SET Total_Salary = Total_Salary + NEW.Emp_Salary
WHERE Dept_No = NEW.Dept_No;
UPDATE DEPARTMENT
SET Total_Salary = Total_Salary – OLD.Emp_Salary
WHERE Dept_No = OLD.Dept_No;
END;
```

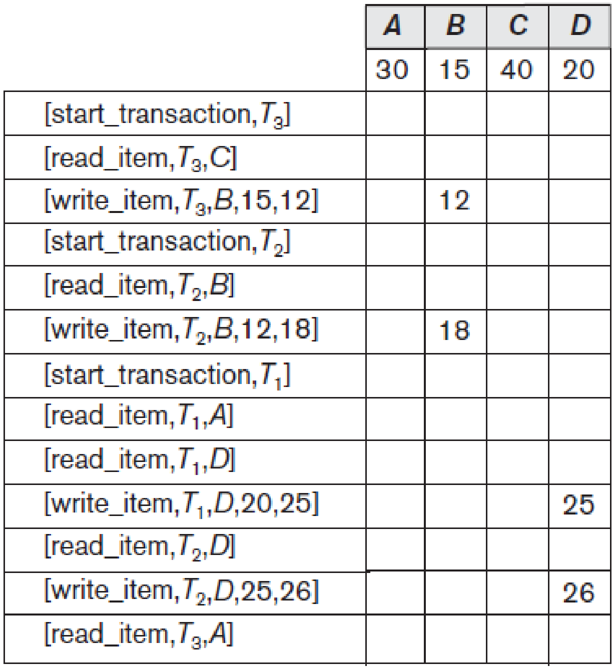
d)

```
CREATE TRIGGER Total_Salary_04
AFTER DELETE ON EMPLOYEE
FOR EACH ROW
WHEN ( OLD.Dept_No IS NOT NULL )
UPDATE DEPARTMENT
SET Total_Salary = Total_Salary – OLD.Emp_Salary
WHERE Dept_No = OLD.Dept_No;
```

Q2. Suppose that the system crashes before the [write_item,T2,D,25,26] entry is written to the log in the following figure; Which transactions have to be rolled back? Why? (1.5 marks)



T_1	T_2	T_3
read_item(*A*)	read_item(*B*)	read_item(*C*)
read_item(*D*)	write_item(*B*)	write_item(*B*)
write_item(*D*)	read_item(*D*)	read_item(*A*)
	write_item(*D*)	write_item(*A*)



	A	B	C	D
	30	15	40	20
[start_transaction,T_3]				
[read_item,T_3,C]				
[write_item,T_3,B,15,12]		12		
[start_transaction,T_2]				
[read_item,T_2,B]				
[write_item,T_2,B,12,18]		18		
[start_transaction,T_1]				
[read_item,T_1,A]				
[read_item,T_1,D]				
[write_item,T_1,D,20,25]				25
[read_item,T_2,D]				
[write_item,T_2,D,25,26]				26
[read_item,T_3,A]				

System crash

Since both transactions T2 and T3 are not yet committed, they have to be rolled back during the recovery process.

Q3. What is the idea behind distributed query processing using the *semijoin* operation? Write your answer with an explanation (1.5 marks)

Answer:

The idea behind distributed query processing using the semijoin operation is to reduce the number of tuples in a relation before transferring it to another site. The idea is to send the joining column of one relation R to the site where the other relation S is located; this column is then joined with S. Following that, the join attributes, along with the attributes required in the result, are projected out and shipped back to the original site and joined with R. Hence, only the joining column of R is transferred in one direction, and a subset of S with no extraneous tuples or attributes is transferred in the other direction. If only a small fraction of the tuples in S participate in the join, this can be quite an efficient solution to minimizing data transfer.

Q4. What is mandatory access control? Provide advantages and disadvantages of Bell-LaPadula Confidentiality Model. (2 marks)

Mandatory Access Control

In (MAC), the system (and not the users) specifies which subjects can access specific data objects.

The MAC model is based on security labels. Subjects are given a security clearance (secret, top secret, confidential, etc.), and data objects are given a security classification (secret, top secret, confidential, etc.). The clearance and classification data are stored in the security labels, which are bound to the specific subjects and objects.

Advantages of Bell-LaPadula in Multi-Layer Secure (MLS) systems,

- making MLS systems immune to Trojan Horse attacks. In perfect implementations, MLS systems implementing Bell-LaPadula MAC are not susceptible Trojan Horse forced security violations because users do not have the ability to declassify information.
- Additionally, MAC is relatively straightforward and is considered a good model for commercial systems that operate in hostile environments (web servers and financial institutions) where the risk of attack is very high, confidentiality is a primary access control concern, or the objects being protected are valuable

Limitations.

- Only addresses confidentiality, control of writing (one form of integrity), ★-property and discretionary access control
- Covert channels are mentioned but are not addressed comprehensively
- The tranquility principle limits its applicability to systems where security levels do not change dynamically. It allows controlled copying from high to low via trusted subjects.
- The state-transition model does not contain any state invariants.