Assignment 4 Solutions

CS338 Fall 2013

1 Problem 1

Consider four different transaction execution schedules (include **read/write** opearations)

$H_1 = r_1[x]r_1[y]w_1[y]r_2[y]r_2[x]w_2[y]w_2[x]r_2[z]$

$H_2 = r_2[y]r_2[x]w_2[y]w_2[x]r_2[z]r_1[x]r_1[y]w_1[y]$

$H_3 = r_3[y]w_2[x]r_1[x]r_1[z]r_3[z]w_1[z]w_3[z]r_1[y]r_2[y]$

$H_4 = w_2[x]w_3[z]r_3[x]r_4[y]r_3[z]w_1[y]w_4[x]r_1[x]r_1[z]r_4[z]$

Answer the following questions:

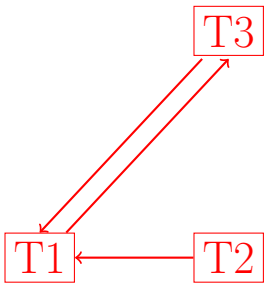
1. List all the conflicting pairs for H_1 and H_2.
   $H_1 = \{(r_1[x], w_2[x]), (r_1[y], w_2[y]), (w_1[y], r_2[y]), (w_1[y], w_2[y])\}$
   Transaction Order: $T_1 \rightarrow T_2$

   $H_2 = \{(r_2[x], w_2[x]), (w_2[x], r_1[x]), (r_2[y], w_2[y]), (r_2[y], w_1[y]), (w_2[y], r_1[y]), (w_2[y], w_1[y])\}$
   Transaction Order: $T_2 \rightarrow T_1$

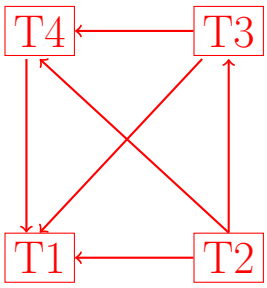
2. Are H_1 and H_2 conflict equivalent and why?
   No, H_1 Transaction Order: $T_1 \rightarrow T_2 \neq$ H_2 Transaction Order: $T_2 \rightarrow T_1$

3. For H_3 and H_4,
   - Give the serialization graph.
   - Determine whether or not the schedule is serializable, and justify your answer.
   - If the schedule is serializable, *specify a serial order of transaction execution to which it is equivalent.*

   1. H_3 is not serializable since a cycle is inside the graph ($T_1 \leftrightarrow T_3$)



2. H_4 is serializable since no cycle in the graph (Transaction Order: $T_2 \rightarrow T_3 \rightarrow T_1 \rightarrow T_4$)



Problem 2

Suppose user *Bob* has privileges to read a secret table T. User *Mallory* wants to see the data in T (but does not have the privileges to do so). If the system is using **Discretionary AC (Access Control)**, *Mallory* may have the chance to conduct a *Trojan Horse Attack* by performing the following steps:

1. *Mallory* creates a table $T^{'}$ and gives INSERT privileges to *Bob*.
2. *Mallory* tricks *Bob* into copying data from T to $T^{'}$ (e.g. by extending the "functionality" of a program used by *Bob*).
3. *Mallory* can then see the data that comes from T

Mandatory AC could stop this kind of attack. For example, if we are using the *Bell-LaPadula Model*, where four different *Security Clearances* are provided: Top Secret(TS), Secret(S), Confidential(C), unclassified(U). Order of the privilege level is

$$TS > S > C > U \tag{1}$$

Suppose user *Bob* still has privileges to read a secret table T, which means

$$clearance(Bob) := S \tag{2}$$

And User *Mallory* still wants to see the data in T (but does not have the privileges to do so).

$$clearance(Mallory) < S \tag{3}$$

Explain: why user *Mallory* can not see the content of secret table T, if he tries to use the same strategy as described above, under *Bell-LaPadula Model*.

1. *Mallory* creates a table $T^{'}$ and gives INSERT privileges to *Bob*.
   - $class(T^{'}) := clearance(Mallory)$
   - i.e. $class(T^{'}) < S$
2. *Mallory* tricks *Bob* into copying data from T to $T^{'}$.
   - writing to $T^{'}$ ***fails*** for *Bob* because $clearance(Bob) \not\leq class(T^{'})$
3. *Mallory* cannot steal the data from T

Problem 3

Consider the following relational schema:
EMPLOYEE(Fname, Lname, Ssn, Bdate, Address, Salary, Dno)
PROJECT(Pname, Pnumber, Plocation, Dnum)
WORKS_ON(Essn,Pno, Hours)
where WORKS_ON.Essn is a foreign key to EMPLOYEE.Ssn, and WORKS_ON.Pno is a foreign key to PROJECT.Pnumber.

Consider the following SQL query:

```
SELECT Pnumber, Pname, COUNT(*)
FROM PROJECT, WORKS_ON, EMPLOYEE
WHERE Pnumber = Pno AND Ssn=Essn AND Dno = 5
GROUP BY Pnumber, Pname
```

Draw two query trees that can represent this query. Argue why these are equivalent (i.e., which rules you applied to get one from the other).

One tree is given in Figure 1 and the second one is given in Figure 2. Figure 2 is obtained from the first by

- Distributing selection over join;

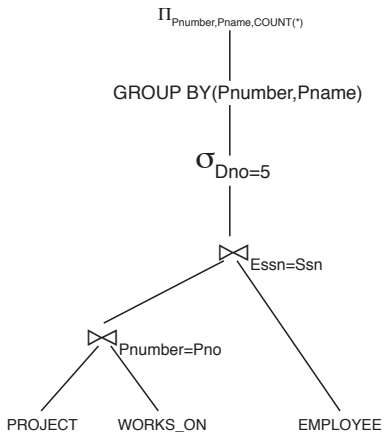


Figure 1 First Tree for Question 3

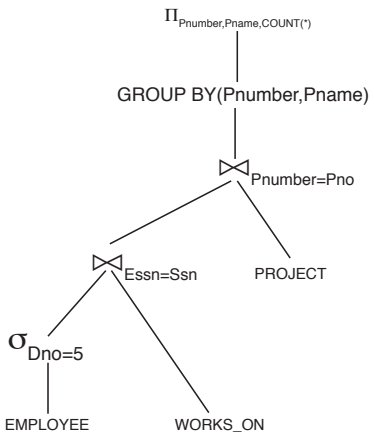


Figure 2 Second Tree for Question 3

- Changing the join order.

Note: it is generally preferred to give a name to the attribute generated by aggregation operators so that the resulting column has a meaningful name. In this case we should have said SELECT Pname, Pnumber, COUNT(*) AS TotalProjects or something like that. If we did that, we could put the COUNT(*) as the next operator after GROUP BY followed by Projection over Pname, Pnumber, TotalProjects.

Problem 4

Let V be a view created over relation R (create view V as $SELECT \ldots FROM \ldots$). Assume that initially *Bob* has all permissions on R (including permission to grant permissions to others), nobody else has permissions on R, and that *Alice* and *Clara* have select permission on V.
Now consider the sequence of commands executed by the specified users to grant and revoke permissions as showed in *Table 1*:

Table 1

Order	Command	Executed by
1	Grant Select on R To Alice with Grant Option	Bob
2	Grant Select on R To Clara	Alice
3	Grant Select on R To Donald	Alice
4	Grant Select on R To Clara	Bob
5	Revoke Select on R From Alice	Bob

Question: Which of *Bob*, *Alice*, *Clara*, *Donald* are authorized to execute each of the commands as showed in *Table 2* at the conclusion of this sequence.

Table 2

Command	Bob	Alice	Clara	Donald
SELECT X FROM R WHERE $Y < 100$	✓		✓	
UPDATE R SET $Y = Y * 3$	✓			
SELECT A FROM V WHERE $C = 10$		✓	✓	
CREATE VIEW $View2$ AS SELECT * FROM R	✓		✓	