

Assignment 4: Triggers, Stored Procedures, Transactions

Course ISE 322: Database Systems

Due January 24, 2010

For the following questions, consider the following relational schema. An employee can work in more than one department; the `pctTime` field of the `Works` relation shows the percentage of time that a given employee works in a given department.

`Emp`(*eid*:integer, *ename*:string, *age*:integer, *salary*:real)
`Works`(*eid*:integer, *did*:integer, *pcttime*: integer)
`Dept`(*did*:integer, *dname*:string, *budget*:real, *managerid*: integer)

1 Assertions and Triggers (43 points)

Write SQL-92 integrity constraints (domain, key, foreign key, or CHECK constraints; or assertions) or MS SQL Server triggers to ensure each of the following requirements, considered independently.

1. (3 points) Employees must make a minimum salary of \$1000.
2. (5 points) Every manager must be listed as working in the Department she manages (must appear in `Works` for that *did*).
3. (10 points) When an employee's `pcttime` is updated for any department, ensure the change doesn't cause his total appointment to be over 100%. If the change will cause the total appointments of the Employee to be over 100%, reject the update. Assume only one Employee is updated at a time.
4. (10 points) A manager must always have a higher salary than any employee in any of her departments.
5. (15 points) Whenever an employee is given a raise, all of her managers' salaries must be increased to be at least as much as her new salary (*e.g.* if Sally receives a raise to \$1000, if her boss Clara has a salary of \$900, it must be increased to \$1000. Sally's other boss, Berenice, who has a salary of \$1100 does not need to be changed).

Similarly, when an employee is added to work in a department, the boss of the department must receive a raise (if needed) to ensure that her salary is at least as big as the new employee's.

(You may find it easier to write this with two different triggers). Assume only one employee is inserted/modified at a time.

2 Stored Procedures (22 points)

Write MS SQL Server stored procedures to perform the following actions.

1. (5 points) Give a raise of value *raiseAmt* to a single Employee *empId*
2. (5 points) Give a raise to of value *raiseAmt* to all employees who work in Department *deptId*
3. (5 points) Give a raise of value *raiseAmt* to all managers who are at least *a* years old

4. (7 points) Give a raise of value *raiseAmt* (default \$100) to all Employees for whom the sum of all of their *pctTime*'s is at least *p*

3 Transactions (10 points)

Consider the following sequence of reads (R) and writes (W) by three concurrent transactions *T1*, *T2*, *T3*. For each schedule, write whether it is:

- Non-serializable and forbidden by Strict 2PL - Explain why it is forbidden by Strict 2PL (include all conflicts)
- Serializable and forbidden by Strict 2PL - Write down (a) the serial ordering it is equivalent to and (b) why it is forbidden by Strict 2PL (include all conflicts)
- Serializable and permitted by Strict 2PL - Write down the serial ordering it is equivalent to

1. (5 points) Schedule 1:

T1	T2	T3
	W(X)	
R(X) W(X)		
	W(Y) Commit	
		R(X) R(Y)
R(Y) Commit		
		W(Y) Commit

2. (5 points) Schedule 2

T4	T5	T6
	R(X)	
W(Y)		
		R(X)
R(X) Commit		
		W(X)
	R(Y)	
		R(Y) Commit
	W(Y) Commit	

4 What to turn in by January 24, 2010 at 11:59pm

You may work in groups of two or three students.

You must turn in the SQL for the Triggers and Stored Procedures and answers for the Transactions in digital format via email ([ise322@gmail](mailto:ise322@gmail.com)), Telem, or in person. In exceptional circumstances you may turn them in on paper to Bili or by fax to the School of Engineering.