

Course ISE 322: Database Systems

Recitation 14: Triggers and Transactions

Michael J. May

January 20, 2009

1 Instead Of Trigger Example

Let's consider another example trigger to better understand how "Instead of" triggers work.

Consider the schema:

```
Emp (eid INT, ename STRING, city STRING)
HourlyEmp (eid INT, hoursWorked INT, hourlyRate INT)
ContractEmp (eid INT, monthlySal INT)
```

Consider the following view:

```
CREATE VIEW MonthlyPay (eid, ename, city, sal) AS
  SELECT E1.eid, E1.ename, E1.city, H.hoursWorked * H.hourlyRate
  FROM Emp E1, HourlyEmp H
  WHERE E1.eid = H.eid

UNION

  SELECT E2.eid, E2.ename, E2.city, C.monthlySal
  FROM Emp E2, ContractEmp C
  WHERE E2.eid = C.eid
```

If somebody would want to indicate that employee 107 moved to Tiberias using the following update:

```
UPDATE MonthlyPay SET city = 'Tiberias' WHERE eid = 107
```

We would get an immediate error message. It's not possible to edit a field in a view such as this. So how can we allow updates to the city field using just the view?

1.1 Instead of Trigger

Let's write a trigger which will capture the intended action and actually do it on Emp. A first cut would be a row-level trigger. This one only allow updating one employee at a time.

```
CREATE TRIGGER updateCity
ON MonthlyPay INSTEAD OF UPDATE AS
IF UPDATE (city)
BEGIN
  DECLARE @newcit CHAR(20)
  DECLARE @eid INT
```

```

SELECT @eid = (SELECT eid FROM inserted)
SELECT @newcit = (SELECT city FROM inserted)

UPDATE Emp SET city = @newcit WHERE eid = @eid
END

```

As a second cut we can change the trigger to work where we update multiple employees with the same city. Note that it won't allow us to modify multiple employees with multiple cities.

```

CREATE TRIGGER updateCity
ON MonthlyPay INSTEAD OF UPDATE AS
IF UPDATE(city)
BEGIN
    UPDATE Emp SET city = (SELECT DISTINCT city FROM inserted) WHERE eid IN (SELECT eid FROM inserted)
END

```

Now attempting to run the above update command, we find that it works and that the table Emp is updated with the new city information. Try it also with the command:

```

UPDATE MonthlyPay SET city = 'Golan' WHERE eid = 101 OR eid = 102

```

1.2 What to do

Now it's your turn to write INSTEAD OF triggers.

1. (Easy) Adapt the above trigger updateCity to allow the changing of names instead of cities. Call the new trigger updateName.
2. (Harder) Adapt the above trigger updateCity to allow the updating of salaries instead of cities. For contract employees, update the monthlySal value. For hourly employees, updating the hourlyRate value (if the employee worked 10 hours and is paid 100 per hour = 1000 and the raise is to 2000, change the hourlyRate to 200). Call the new trigger updateSal.

Note: In order to define the above triggers you will need to drop the first one. You can not define more than one INSTEAD OF Update trigger on a single relation. (Why?)

2 Transactions

Consider the following sequence of reads (R) and writes (W) by three concurrent transactions T_1 , T_2 , T_3 . 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. Schedule 1:

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

2. Schedule 2

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