**Exercise C:** Choose **two** of your procedural **constraints**. One which has no problems with non-repeatable read or phantoms, in a multi-user environment under the default isolation level (READ COMMITTED) and one which does.

Explain your choices by giving scenarios with two transactions that illustrate why it can or can’t go wrong. Add a success scenario with an isolation level that solves the problematic one.

For every scenario describe what kind of locks are acquired (e.g. s-locks and x-locks), when, why and for how long.

**No problems with non-repeatable reads/phantoms**

|  |  |
| --- | --- |
| **Connection 1 – Read commited** | **Connection 2 – Read commited** |
| BEGIN  declare @tr\_name varchar(10) = 'none'  BEGIN TRY  if @@trancount > 0  begin  set @tr\_name = 'yep'  save tran @tr\_name  end  else  begin  begin tran  end  if((select e.job from emp e where e.empno = 1) = 'PRESIDENT')  Connection selects the job from employee number 1. It sees that that job is ‘PRESIDENT’ so the value in the if statement is true. S-lock was held while the connection was reading the data, but it is now removed after this statement finished. |  |
|  | BEGIN  declare @tr\_name varchar(10) = 'none'  BEGIN TRY  if @@trancount > 0  begin  set @tr\_name = 'yep'  save tran @tr\_name  end  else  begin  begin tran  end  if((select e.job from emp e where e.empno = 1) = 'PRESIDENT')  Connection selects the job from employee number 1. It sees that that job is ‘PRESIDENT’. Connection 1 has not changed this value so there is no nonrepeatable read. Also there are no records in the result of this select statement that weren’t there before the begin transaction of connection 2 was called, so there are no phantom records.The value in the if statement is true. S-lock was held while the connection was reading the data, but it is now removed after this statement finished. |
| Begin  if(1000000 >= 10000)  begin  update emp  set msal = @newMsal  where empno = @empno  Connection 1 puts an S-lock on this data. There is no S-lock on it because connection 2 has already dropped it’s lock because the isolation level is ‘Read commited’.  end  else  THROW 50001, 'President msal can not be less than 10000', 1;  end  else  begin  update emp  set msal = @newMsal  where empno = @empno  end  if @tr\_name = 'none'  COMMIT TRAN  The transaction is commited by connection 1 so all s-locks and x-locks placed by connection 1 are dropped.  END TRY  BEGIN CATCH  if @tr\_name = 'none'  ROLLBACK TRAN  else  rollback tran @tr\_name  DECLARE @Message nvarchar(2048) = ERROR\_MESSAGE()  raiserror (@Message, 16, 1)  END CATCH  END |  |
|  | Begin  if(1000000 >= 10000)  begin  update emp  set msal = @newMsal  where empno = @empno  Connection 2 puts an S-lock on this data. There were no locks on it when this statement was executed.  end  else  THROW 50001, 'President msal can not be less than 10000', 1;  end  else  begin  update emp  set msal = @newMsal  where empno = @empno  end  if @tr\_name = 'none'  COMMIT TRAN  The transaction is commited by connection 2.  END TRY  BEGIN CATCH  if @tr\_name = 'none'  ROLLBACK TRAN  else  rollback tran @tr\_name  DECLARE @Message nvarchar(2048) = ERROR\_MESSAGE()  raiserror (@Message, 16, 1)  END CATCH  END |

**Problem with non-repeatable reads/phantoms**

|  |  |
| --- | --- |
| **Connection 1 – Read commited** | **Connection 2 – Read commited** |
| go  CREATE TRIGGER utr\_deleteEmp  on emp  after delete  AS  BEGIN TRY  if ((select job from deleted) = 'ADMIN')  If statement is true  begin  if (not exists (select \* from emp where deptno = (select deptno from deleted) and (job = 'ADMIN')))  This if statement is true. Which means there are no more admins in the dept of the deleted admin. S-locks are placed and immediately removed after the select is done. |  |
|  | update emp  set job = 'MANAGER'  where empno = 1  Connection 2 puts an x-lock on empno 1 (which is the same empno as in connection 1) and releases it due to autocommit. |
| begin  if (exists (select \* from emp where deptno = (select deptno from deleted) and (job = 'PRESIDENT' or job = 'MANAGER')))  This if statement is true. So there is a manager or president in this dept. S-locks are placed and immediately removed.  THROW 50002, 'You cant delete an admin from a department with no more admins and a president/manager', 1;  Connection one throws this error that says you can’t delete this admin because it is in a dept with no more admins and it has a president/admin in it. This is false because the emp that is being deleted (empno = 1) is not an admin anymore. Connection 2 changed it to a Manager which means this error is false.  end  end  END TRY  BEGIN CATCH  ;THROW  END CATCH  go |  |

**Solution**

|  |  |
| --- | --- |
| **Connection 1 – Repeatable read** | **Connection 2 – Read commited** |
| go  CREATE TRIGGER utr\_deleteEmp  on emp  after delete  AS  BEGIN TRY  if ((select job from deleted) = 'ADMIN')  If statement is true  begin  if (not exists (select \* from emp where deptno = (select deptno from deleted) and (job = 'ADMIN')))  This if statement is true. Which means there are no more admins in the dept of the deleted admin. S-locks are placed and held until transaction commit due to isolation level repeatable read. |  |
|  | update emp  set job = 'MANAGER'  where empno = 1  Connection 2 tries to put an x-lock the row with empno = 1 but can’t because connection 1 has put on s-lock on it, this statement can’t be executed. |
| begin  if (exists (select \* from emp where deptno = (select deptno from deleted) and (job = 'PRESIDENT' or job = 'MANAGER')))  This if statement is true. So there is a manager or president in this dept. S-locks are placed and held until transaction commit  THROW 50002, 'You cant delete an admin from a department with no more admins and a president/manager', 1;  Connection one throws this error that says you can’t delete this admin because it is in a dept with no more admins and it has a president/admin in it. This is correct.  end  end  END TRY  BEGIN CATCH  ;THROW  END CATCH  Go After rollback transaction/tansaction commit, S-locks are removed. |  |