Types of Locks

Shared/Exclusive Locks

- **Problem**: Simple locks don't not allow two readers of a DB element X at the same time.

- Multiple readers not a problem.

- Shared lock $\mathbf{sl_i(X)}$ Exclusive lock $\mathbf{xl_i(X)}$

	S	X
S	yes	no
X	no	no

Exercise

r1(A); r2(B); r3(C); r1(B); r2(C); r3(D); w1(A); w2(B); w3(C);

T1	T2	T3
xl(A); r1(A)		
	xl(B); r2(B)	
		xl(C); r3(C)
sl(B) **denied**		
	sl(C) **denied**	
		sl(D); r3(D); u(D)
		w3(C); u(C)
	sl(C); r2(C);	
	w2(B); u(B); u(C)	
sl(B); r1(B);		
w1(A); u(A); u(B)		

Upgrading Locks

- Instead of taking an exclusive lock immediately, a transaction can take a *shared* lock on X, read X, and then upgrade the lock to *exclusive* so that it can write X.

T_1	T_2
$sl_1(A);\ r_1(A);$	
	$sl_2(A);\ r_2(A);$
	$sl_2(B);\ r_2(B);$
$sl_1(B);\ r_1(B);$	
$xl_1(B)$ Denied	
	$u_2(A);\ u_2(B)$
$xl_1(B);\ w_1(B);$	
$u_1(A);\ u_2(B);$	

Upgrading Locks allows more concurrent operation:

Had T1 asked for an exclusive lock on B before reading B, the request would have been denied, because T2 already has a shared lock on B.

Exercise

r1(A); r2(B); r3(C); r1(B); r2(C); r3(D); w1(A); w2(B); w3(C);

T1	T2	T3
sl(A); r1(A)		
	sl(B); r2(B)	
		sl(C); r3(C)
sl(B); r1(B)		
	sl(C); r2(C)	
		sl(D); r3(D)
xl(A); w1(A);		
u(A); u(B)		
	xl(B); w2(B);	
	u(B); u(C)	
		xl(C);w3(C);
		u(C); u(D)

Possibility for Deadlocks

Example:T1 and T2 each reads X and later writes X.

T_1	T_2
$sl_1(X)$	
	$sl_2(X)$
$xl_1(X)$ Denied	
	$xl_2(X)$ Denied

Problem: when we allow upgrades, it is easy to get into a deadlock situation.

Solution: Update Locks

- Update lock **$ul_i(X)$**.
  - Only an update lock (not shared lock) can be upgraded to exclusive lock (if there are no shared locks anymore).
  - A transaction that will read and later on write some element A, asks initially for an update lock on A, and then asks for an exclusive lock on A. Such a transaction doesn't ask for a shared lock on A.
- Legal schedules:
  - read action permitted when there is either a shared or update lock.
  - An update lock can be granted while there is a shared lock, but the scheduler will not grant a shared lock when there is an update lock.

- **2PL condition:** No transaction may have an **sl(X), ul(X)** or **xl(X)** after a **u(Y)**.

	S	X	U
S	yes	no	yes
X	no	no	no
U	no	no	no

Example

T1	T2	T3
sl(A); r(A)		
	ul(A); r(A)	
		sl(A) **Denied**
	xl(A) **Denied**	
u(A)		
	xl(A); w(A)	
	u(A)	
		sl(A); r(A)
		u(A)

(No) Deadlock Example

T_1 and T_2 each read X and later write X.

T_1	T_2
$sl_1(X)$	
	$sl_2(X)$
$xl_1(X)$ Denied	
	$xl_2(X)$ Denied

Deadlock when using sl and xl locks only.

Fine when using update locks.

T_1	T_2
$ul_1(X);\ r_1(X);$	
	$ul_2(X)$ Denied
$xl_1(X);\ w_1(X);\ u_1(X);$	
	$ul_2(X);\ r_2(X);$
	$xl_2(X);\ w_2(X);\ u_2(X);$

Benefits of Upgrade Locks

T1	T2	T3	T4	T5	T6	T7	T8	T9
s(A);r(A)								
	s(A);r(A)							
		s(A);r(A)						
			s(A);r(A)					
				ul(A);r(A)				
					s(A);denied			
						s(A);denied		
							s(A);denied	
								s(A);denied
u(A)								
	u(A)							
		u(A)						
			u(A)					
				xl(A);w(A)				
				u(A)				
					s(A);r(A)			
						s(A);r(A)		
							s(A);r(A)	
								s(A);r(A)