With commit bits; Only one version for each data item

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false
r3(X)	denied; make wait	TS(T3) >= WTS(X) but c(X) == false

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false
r3(X)	denied; make wait	TS(T3) >= WTS(X) but c(X) == false
c2	allowed	c(X) = true

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false
r3(X)	denied; make wait	TS(T3) >= WTS(X) but c(X) == false
c2	allowed	c(X) = true
r3(X)	allowed!	RTS(X) = 300

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false
r3(X)	denied; make wait	TS(T3) >= WTS(X) but c(X) == false
c2	allowed	c(X) = true
r3(X)	allowed!	RTS(X) = 300
c3	allowed	c(Y) = true

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false
r3(X)	denied; make wait	TS(T3) >= WTS(X) but c(X) == false
c2	allowed	c(X) = true
r3(X)	allowed!	RTS(X) = 300
c3	allowed	c(Y) = true
w1(Y)	ignored	TS(T1) >= RTS(Y) but TS(T1) < WTS(Y) and c(Y) == true

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false
r3(X)	denied; make wait	TS(T3) >= WTS(X) but c(X) == false
c2	allowed	c(X) = true
r3(X)	allowed!	RTS(X) = 300
c3	allowed	c(Y) = true
w1(Y)	ignored	TS(T1) >= RTS(Y) but TS(T1) < WTS(Y)
		and c(Y) == true
r1(X)	denied; roll back	TS(T1) < WTS(X); RTS(Y) = 0

With commit bits; Only one version for each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; w1(Y); r1(X);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed	RTS(Y) = 100
w2(Y)	allowed	WTS(Y) = 200, c(Y) = false
w2(X)	allowed	WTS(X) = 200, c(X) = false
r3(Z)	allowed	RTS(Z) = 300
r2(Z)	allowed	RTS is not changed (see below)
w3(Y)	allowed	WTS(Y) = 300, C(Y) remains false
r3(X)	denied; make wait	TS(T3) >= WTS(X) but c(X) == false
c2	allowed	c(X) = true
r3(X)	allowed!	RTS(X) = 300
c3	allowed	c(Y) = true
w1(Y)	ignored	TS(T1) >= RTS(Y) but TS(T1) < WTS(Y)
		and c(Y) == true
r1(X)	denied; roll back	TS(T1) < WTS(X); RTS(Y) = 0
c1	doesn't happen	T1 has already been rolled back

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0
w2(X)	allowed	create X(200) with RTS = 0

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0
w2(X)	allowed	create X(200) with RTS = 0
r3(Z)	allowed to read Z(0)	RTS(Z(0)) = 300

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0
w2(X)	allowed	create X(200) with RTS = 0
r3(Z)	allowed to read Z(0)	RTS(Z(0)) = 300
r2(Z)	allowed to read Z(0)	RTS(Z(0)) is unchanged

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0
w2(X)	allowed	create X(200) with RTS = 0
r3(Z)	allowed to read Z(0)	RTS(Z(0)) = 300
r2(Z)	allowed to read Z(0)	RTS(Z(0)) is unchanged
w3(Y)	allowed	create Y(300) with RTS = 0

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0
w2(X)	allowed	create X(200) with RTS = 0
r3(Z)	allowed to read Z(0)	RTS(Z(0)) = 300
r2(Z)	allowed to read Z(0)	RTS(Z(0)) is unchanged
w3(Y)	allowed	create Y(300) with RTS = 0
r3(X)	allowed to read X(200)	RTS(X(200)) = 300

no commit bits; **multiple versions** of each data item

```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0
w2(X)	allowed	create X(200) with RTS = 0
r3(Z)	allowed to read Z(0)	RTS(Z(0)) = 300
r2(Z)	allowed to read Z(0)	RTS(Z(0)) is unchanged
w3(Y)	allowed	create Y(300) with RTS = 0
r3(X)	allowed to read X(200)	RTS(X(200)) = 300
r1(X)	allowed to read X(0)	RTS(X(0)) = 100

no commit bits; **multiple versions** of each data item

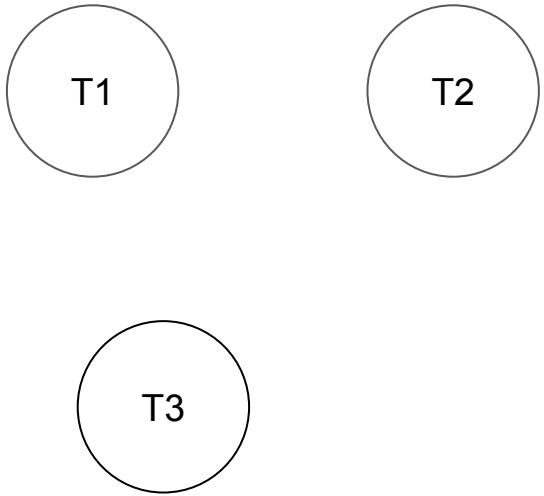
```
s1; r1(Y); s2;
w2(Y); w2(X); s3;
r3(Z); r2(Z);
w3(Y); r3(X); c2;
c3; r1(X); w1(Z);
c1
```

request	response	state changes and/or explanation
r1(Y)	allowed to read Y(0)	RTS(Y(0)) = 100
w2(Y)	allowed	create Y(200) with RTS = 0
w2(X)	allowed	create X(200) with RTS = 0
r3(Z)	allowed to read Z(0)	RTS(Z(0)) = 300
r2(Z)	allowed to read Z(0)	RTS(Z(0)) is unchanged
w3(Y)	allowed	create Y(300) with RTS = 0
r3(X)	allowed to read X(200)	RTS(X(200)) = 300
r1(X)	allowed to read X(0)	RTS(X(0)) = 100
w1(Z)	denied; rollback	RTS(X(0)) = 0; RTS(Y(0)) = 0

Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

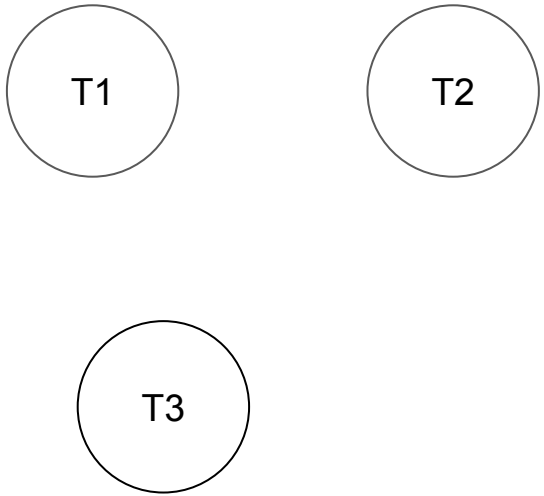
T1	T2	T3
sl(X); r(X)		



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

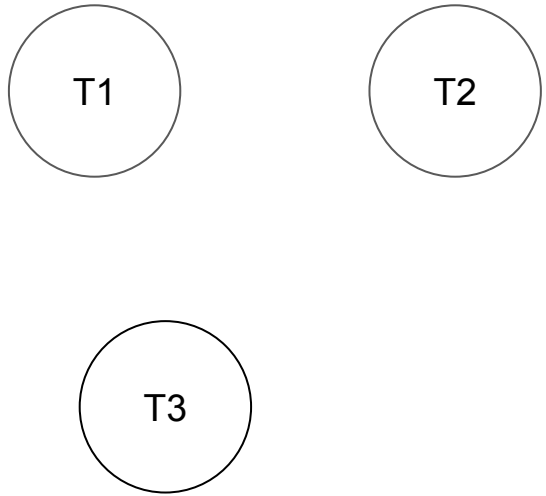
T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

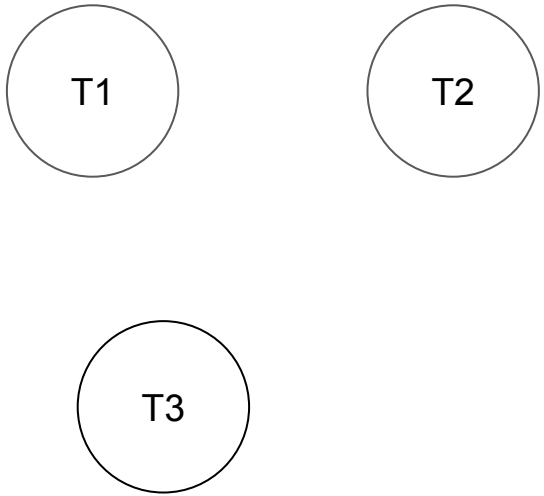
T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		
	xl(Y); w(Y)	



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

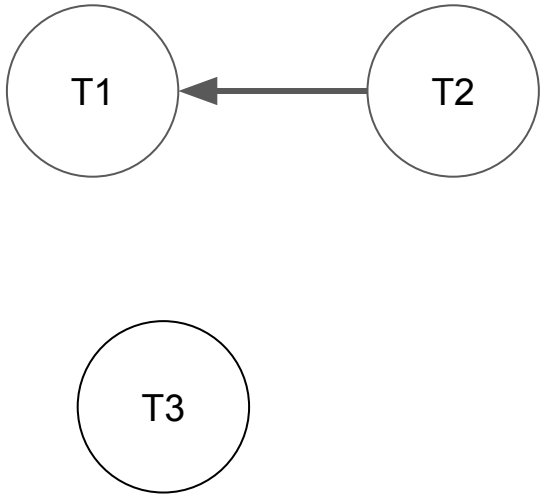
T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		
	xl(Y); w(Y)	
	xl(X)	



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

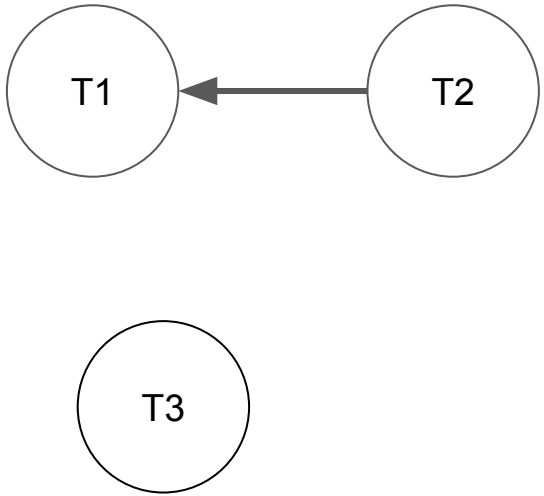
T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		
	xl(Y); w(Y)	
	xl(X)	
	denied	
	wait for T1	



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

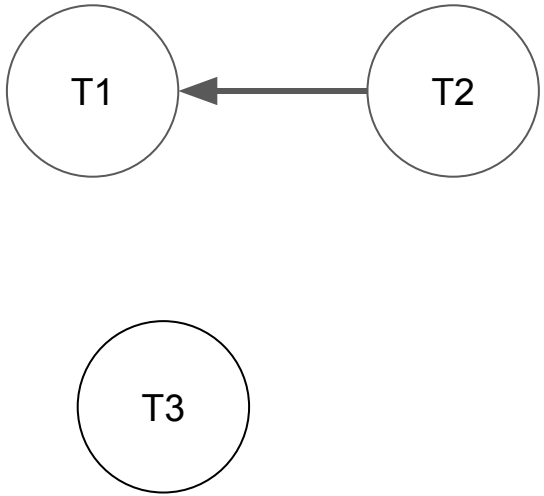
T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		
	xl(Y); w(Y)	
	xl(X)	
	denied	
	wait for T1	
		sl(Z); r(Z)



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

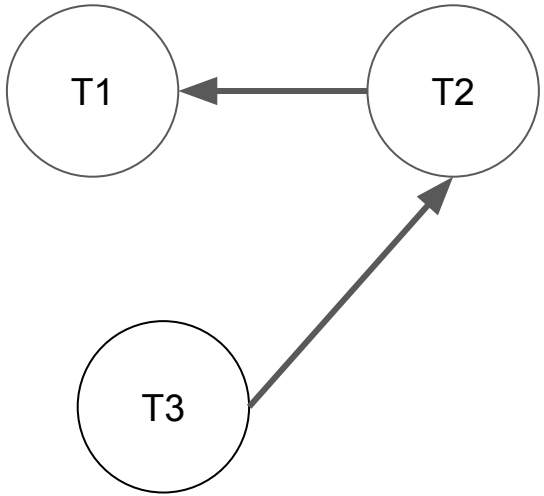
T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		
	xl(Y); w(Y)	
	xl(X)	
	denied	
	wait for T1	
		sl(Z); r(Z)
		xl(Y)



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		
	xl(Y); w(Y)	
	xl(X)	
	denied	
	wait for T1	
		sl(Z); r(Z)
		xl(Y)
		denied
		wait for T2



Rigorous 2PL
No update locks (upgrade directly from shared to exclusive)

```
r1(X); r1(Z); w2(Y); w2(X); r3(Z); w3(Y); w1(Z)
```

T1	T2	T3
sl(X); r(X)		
sl(Z); r(Z)		
	xl(Y); w(Y)	
	xl(X)	
	denied	
	wait for T1	
		sl(Z); r(Z)
		xl(Y)
		denied
		wait for T2
xl(Z)		
denied		
wait for T3