

(set transaction isolation  
level to

SERIALIZABLE

[ OR: REPEATABLE READ

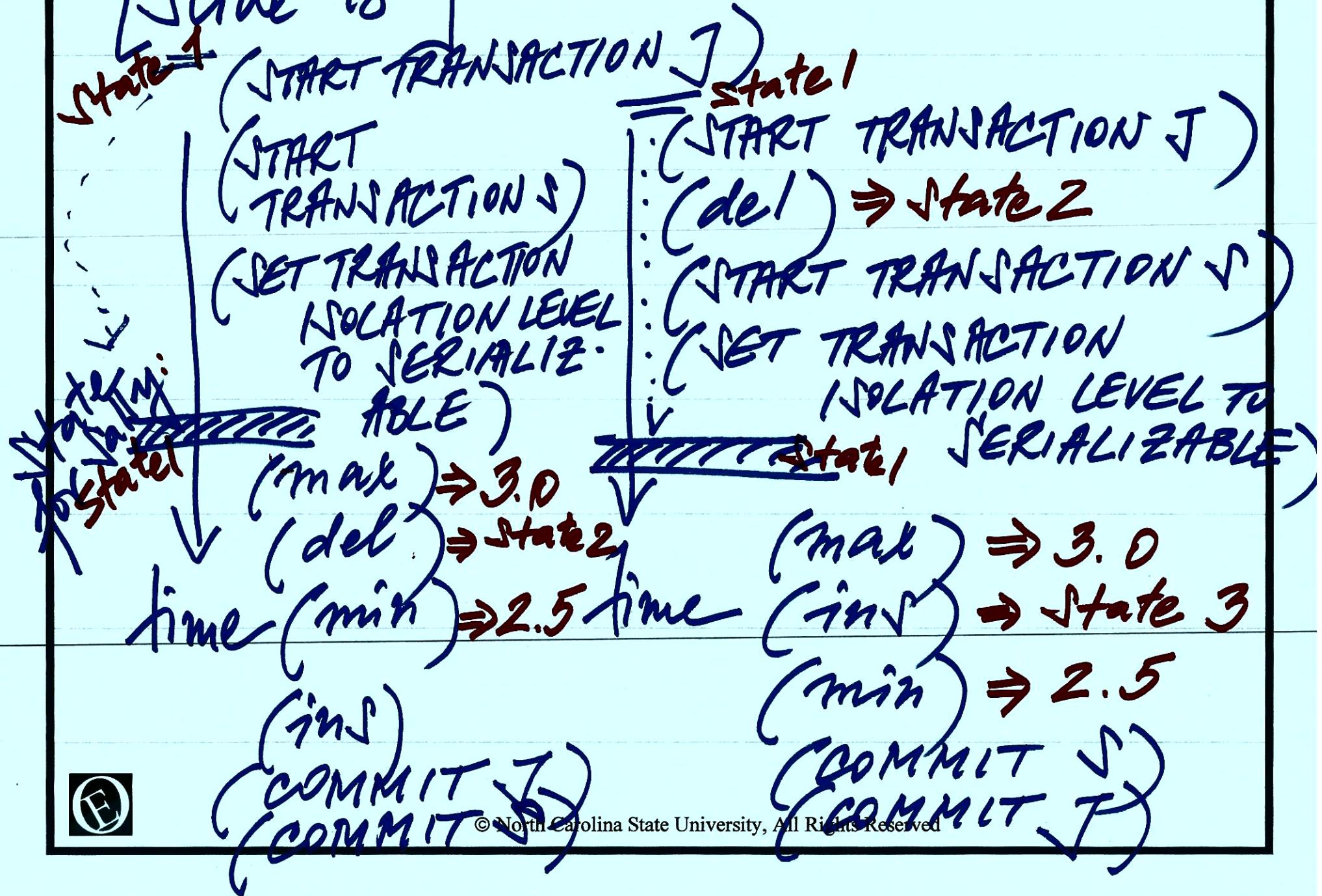
OR: READ COMMITTED

OR: READ UNCOMMITTED ]

These  
define  
what  
your  
read  
actions  
see.



[Slide 18]



# Isolation levels (for $\downarrow$ ):

1. Serializable:

( $\downarrow$ START S)

db state (no dirty reads)

2. Repeatable read

- same as 3, except all the "previous data" stay around

3. Read committed

( $\downarrow$ START  $\downarrow$ )

db state (no dirty reads)

(commit J)

another state (no dirty reads)

(commit T)

(commit F)

(commit H)

(commit K)

(commit L)

(commit M)

(commit N)

(commit O)

(commit P)

(commit Q)

(commit R)

(commit S)

(commit T)

(commit U)

(commit V)

(commit W)

(commit X)

(commit Y)

(commit Z)

(commit AA)

(commit BB)

(commit CC)

(commit DD)

(commit EE)

(commit FF)

(commit GG)

(commit HH)

(commit II)

(commit JJ)

(commit KK)

(commit LL)

(commit MM)

(commit NN)

(commit OO)

(commit PP)

(commit QQ)

(commit RR)

(commit SS)

(commit TT)

(commit UU)

(commit VV)

(commit WW)

(commit XX)

(commit YY)

(commit ZZ)

(commit AAA)

(commit BBB)

(commit CCC)

(commit DDD)

(commit EEE)

(commit FFF)

(commit GGG)

(commit HHH)

(commit III)

(commit JJJ)

(commit KKK)

(commit LLL)

(commit MMM)

(commit NNN)

(commit OOO)

(commit PPP)

(commit QQQ)

(commit RRR)

(commit SSS)

(commit TTT)

(commit UUU)

(commit VVV)

(commit WWW)

(commit XXX)

(commit YYY)

(commit ZZZ)

(commit AAAA)

(commit BBBB)

(commit CCCC)

(commit DDDD)

(commit EEEE)

(commit FFFF)

(commit GGGG)

(commit HHHH)

(commit IIII)

(commit JJJJ)

(commit KKKK)

(commit LLLL)

(commit MMM)

(commit NNNN)

(commit OOOO)

(commit PPPP)

(commit QQQQ)

(commit RRRR)

(commit SSSS)

(commit TTTT)

(commit UUUU)

(commit VVVV)

(commit WWWW)

(commit XXXX)

(commit YYYY)

(commit ZZZZ)

4. Read uncommitted: S sees all dirty reads



[Slide 8]

State 1

Sells: bar beer price

'J'	'B'	2.50
'J'	'M'	3.00

State 2  
(after (del))

Sells: bar beer price  
[empty relation]

State 3  
(after (ins))

Sells: bar beer price

'J'	'H'	3.50
-----	-----	------



Set of transactions for  
2nd example:

~~State 1~~

$\Sigma_1$ : (max) and (min)

- same as before

~~J1~~: (del)  $\leftarrow$  del of (J' 'M' 3.0)

~~State A~~

~~J2~~: (ins)  $\leftarrow$  ins (J' 'H' 3.5)

~~State B~~

(del)  $\leftarrow$  del of (J' 'B' 2.5)

~~State C~~



2nd example:

State T

Sells: bar beer price

'J'	'B'	2.5
'J'	'M'	3.0

State A

(after  $\sqrt{J1}$  (del))

Sells: bar beer price

'J'	'B'	2.5
-----	-----	-----

State B

(after  $\sqrt{J2}$  (ins))

Sells: bar beer price

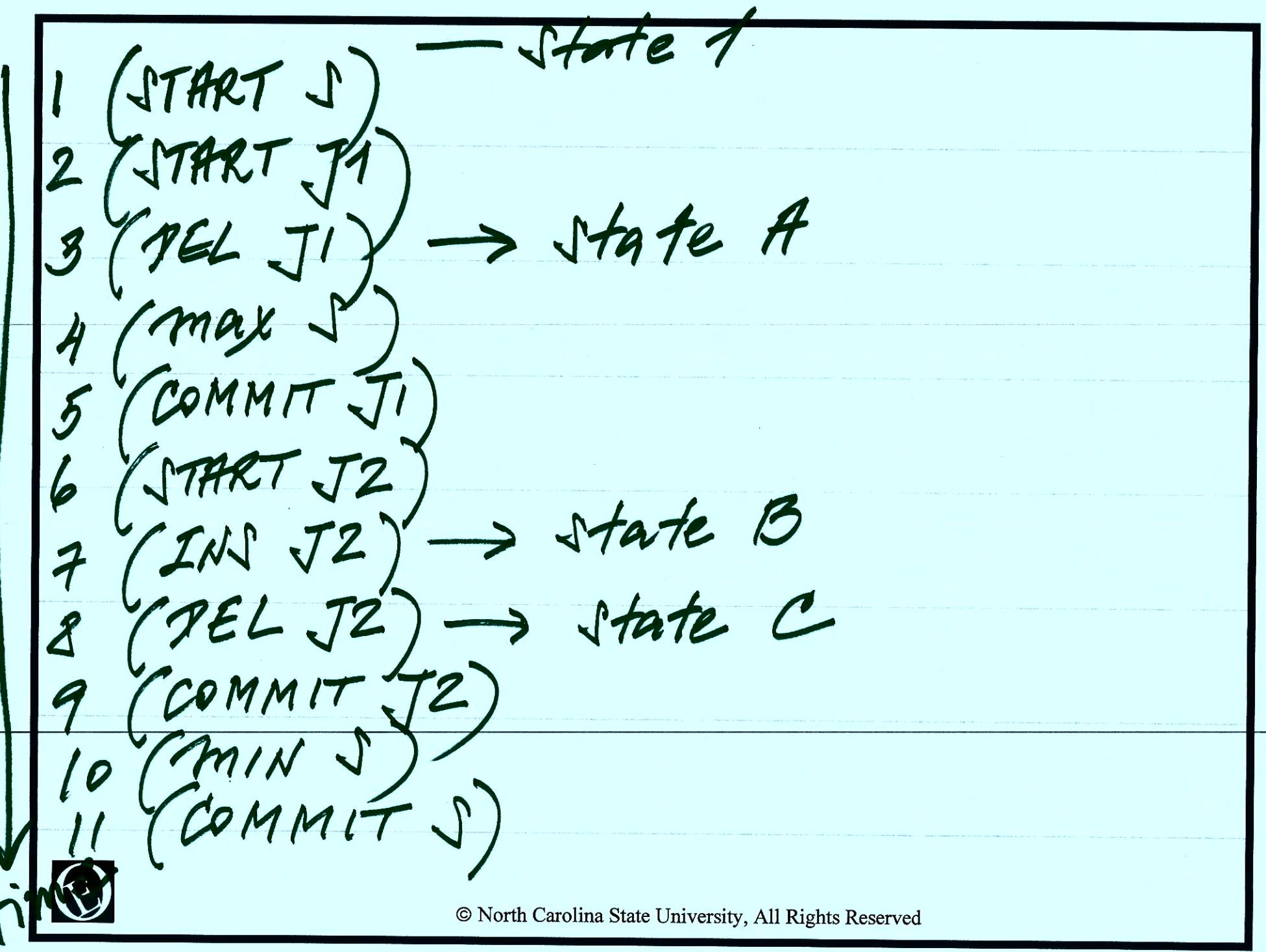
'J'	'B'	2.5
'J'	'H'	3.5

State C

(after  $\sqrt{J^2}$  (del))

Sells: bar beer price

'J'	'H'	3.5
-----	-----	-----



= SERIALIZABLE  $\Rightarrow$  state 1

4  $\Rightarrow$  3.0

10  $\Rightarrow$  2.5



~~=~~ READ UNCOMMITTED - State 1

4  $\Rightarrow$  2.5

10  $\Rightarrow$  3.5



# == READ COMMITTED - State 1

4  $\Rightarrow 3.0$  (from state 1)  
5.  $\Rightarrow$  STATE A for S'

9  $\Rightarrow$  STATE C for A  
10  $\Rightarrow 3.5$  (from state C)



## ~~REPEATABLE READ~~ — State 1

4  $\Rightarrow$  3.0 (from state 1)

5  $\Rightarrow$  STATE A + whatever S saw before  
 $(2.5)$   $(2.5, 3.0)$

9  $\Rightarrow$  STATE C  $(3.5)$  + whatever S saw before  
 $(2.5, 3.0)$

10  $\Rightarrow$  2.5 (from  $(3.5, 2.5, 3.0)$ )

