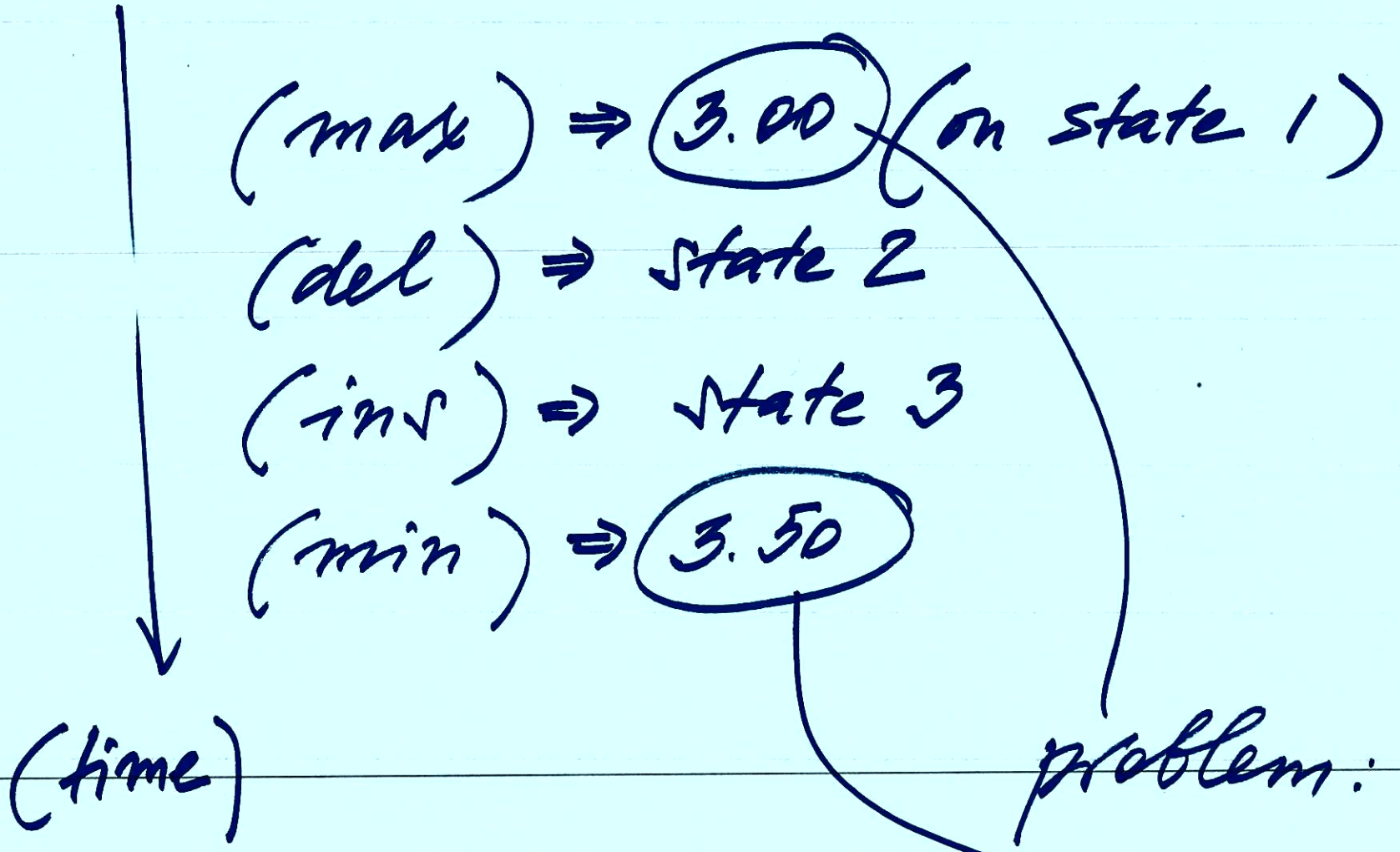
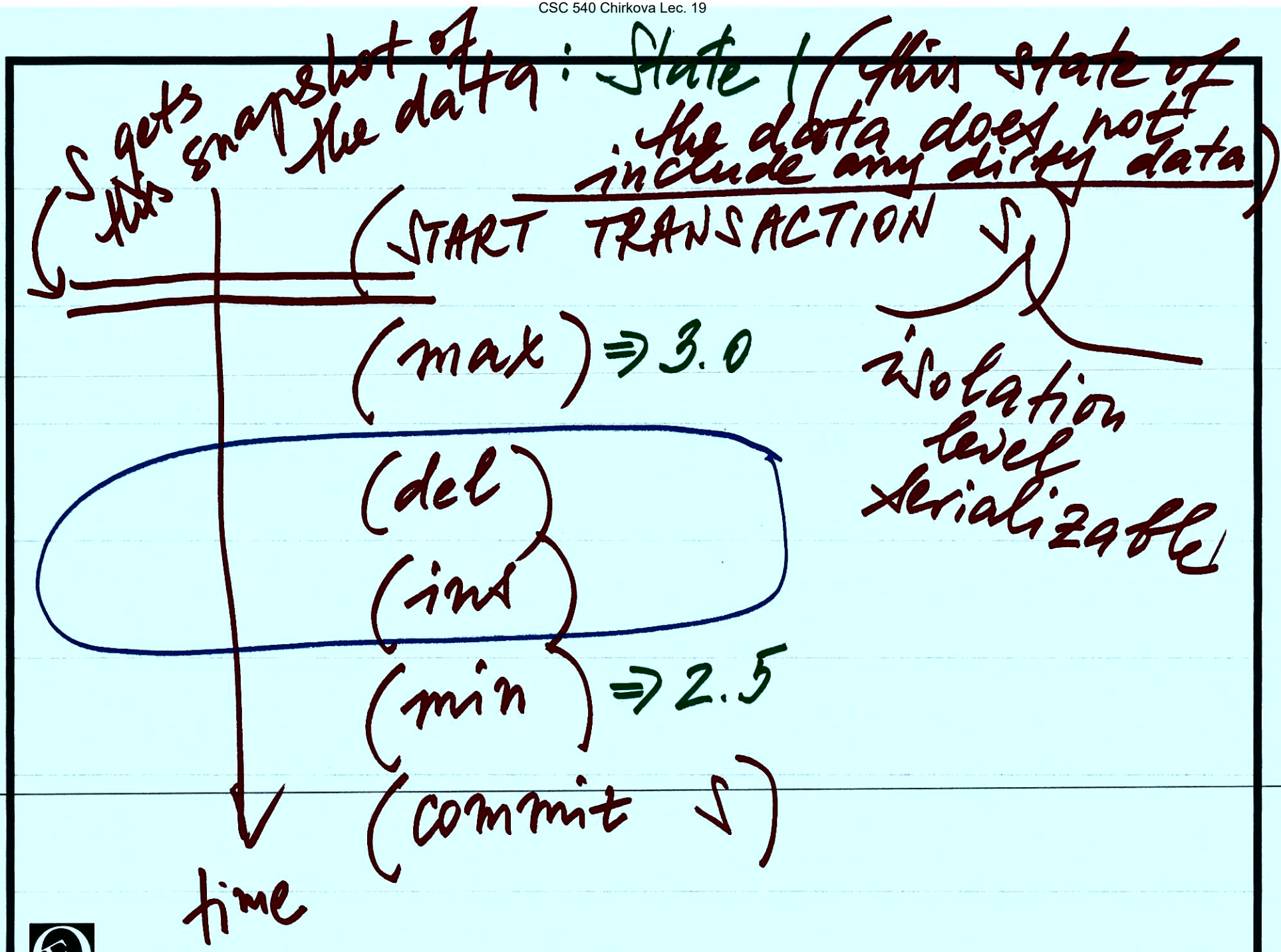
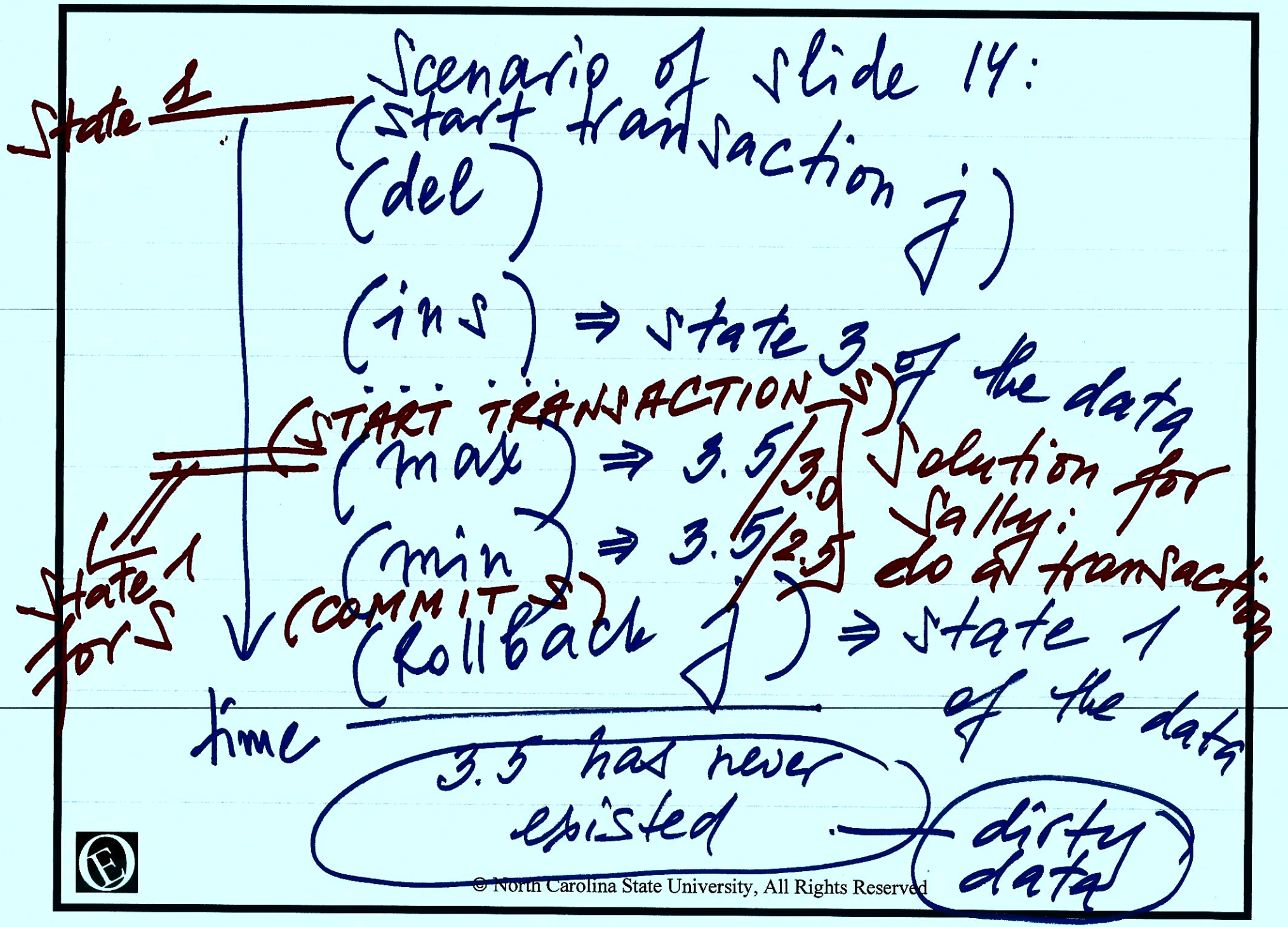






(set transaction isolation level to

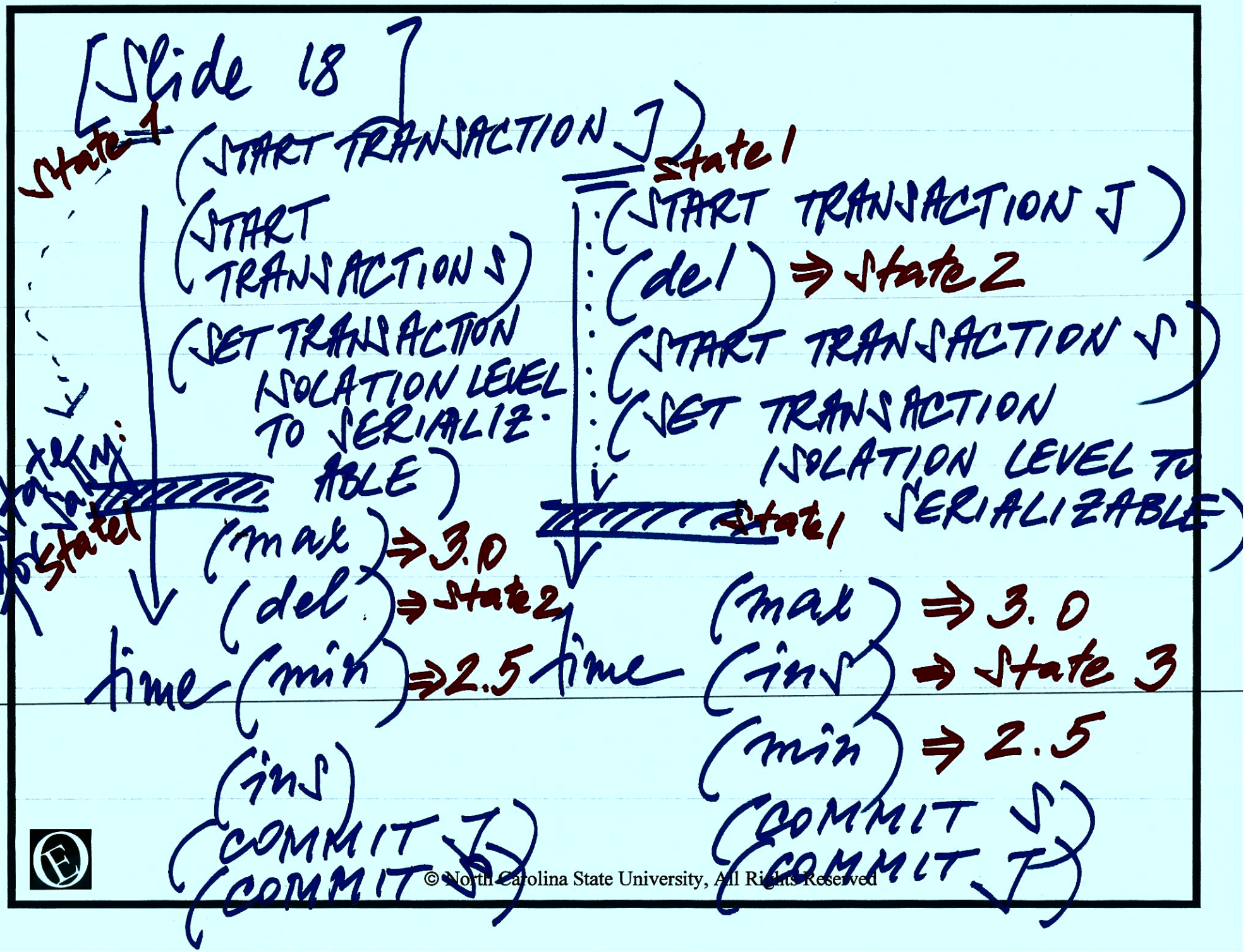
SERIALISABLE

OR: REPEATABLE READ

OR: READ COMMITTED

OR: READ UNCOMMITTED)

these define what your transaction sees



Isolation levels (for √):

1. Serializable:
   (START S) —— db state (no dirty reads)
2. Repeatable read
   — same as 3, except all the "previous data" "√tay around"
3. Read committed
   (START √) —— db state (no dirty reads)
   (COMMIT T) —— another state (no dirty reads)
   (COMMIT) —— . . . .
4. Read uncommitted: √ see all dirty read√

[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

S1: (max) and (min) — same as before

T1: (del) ← del of ('J' 'M' 3.0)

State A

T2: (ins) ← ins ('J' 'H' "3.5")

State B

(del) ← del of ('J' 'B' 2.5)

State C

2nd Example.

State 1

Sells:

bar	beer	price
'J'	'B'	2.5
'J'	'M'	3.0

State A
(after J1 (del))

Sells:

bar	beer	price
'J'	'B'	2.5

State B
(after J2 (ins))

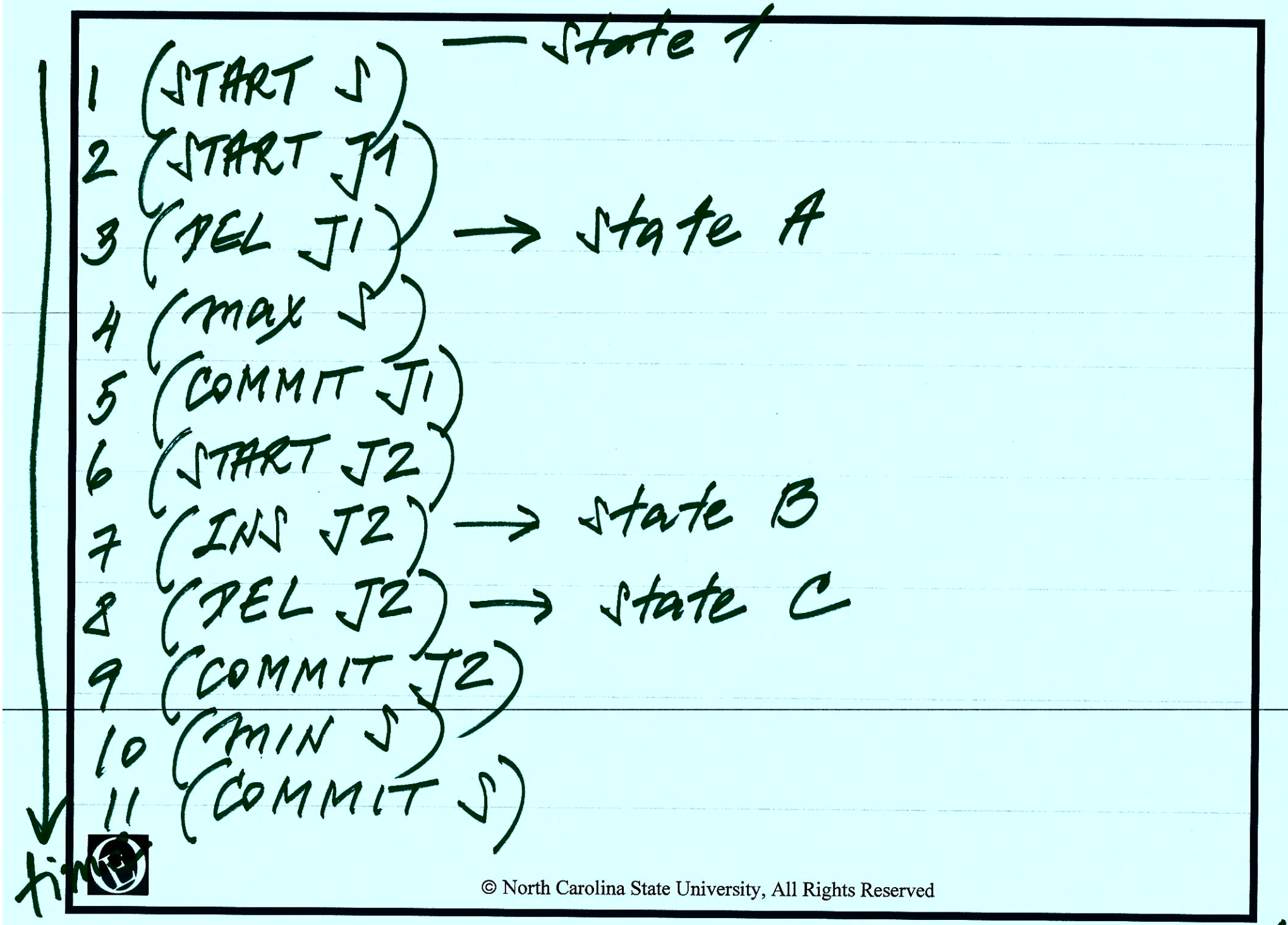
Sells:

bar	beer	price
'J'	'B'	2.5
'J'	'H'	3.5

State C
(after J2 (del))

Sells:

bar	beer	price
'J'	'H'	3.5



SERIALIZABLE ⇒ State 1

4 ⇒ 3.0

10 ⇒ 2.5

READ UNCOMMITTED – State 1

4 ⇒ 2.5

10 ⇒ 3.5

READ COMMITTED — State 1

4 ⇒ 3.0 (from State 1)

5 ⇒ STATE A for S

...

9 ⇒ STATE C for A

10 ⇒ 3.5 (from State C)

REPEATABLE READ — State 1

4 ⇒ 3.0 (from State 1)

5 ⇒ STATE A + whatever I saw before
(2.5) (2.5, 3.0)

9 ⇒ STATE C (3.5) + whatever I saw before
(2.5, 3.0)

10 ⇒ 2.5 (from (3.5, 2.5, 3.0))

E