

Normal forms cont. & intro to transactions

Remy Wang 04/24/2025

superkey

$$X \rightarrow \{A_1, \dots, A_5\}$$

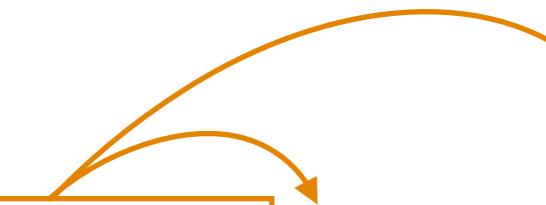
A1	A2	A3	A4	A5

PK

name	location	salary
remy	LA	\$30
vincent	LA	\$20

A diagram illustrating a query operation on a table. The table has four columns: **first n.**, **last n.**, **location**, and **salary**. The first two columns are highlighted with an orange border. Two orange curved arrows point from the **last n.** column towards the **location** and **salary** columns, indicating a projection or selection operation.

first n.	last n.	location	salary
remy	w	LA	\$30
dan	s	seattle	\$50
dan	o	zurich	\$50



first n.	last n.	location	salary	course
remy	w	LA	\$30	143
remy	w	LA	\$30	240
remy	w	LA	\$30	249
dan	s	seattle	\$50	344
dan	s	seattle	\$50	444
dan	o	zurich	\$50	101
dan	o	zurich	\$50	113

name	job	location	salary	tax %
remy	prof	LA	\$30	20
dan	prof	seattle	\$50	15
vincent	TA	LA	\$20	10

$\text{job} \rightarrow \text{salary}$

$\text{name} \rightarrow \text{location} \quad \Rightarrow \quad \text{name, job} \rightarrow \text{tax \%}$

$\text{location, salary} \rightarrow \text{tax \%}$

BNCF

$$X \rightarrow Y : \begin{cases} Y \subseteq X \text{ (trivial FD)} \\ X \text{ is a superkey} \end{cases}$$

PK

name	location	salary
remy	LA	\$30
vincent	LA	\$20

A diagram illustrating a query operation on a table. The table has four columns: **first n.**, **last n.**, **location**, and **salary**. The first two columns are highlighted with an orange border. Two orange curved arrows point from the **last n.** column towards the **location** and **salary** columns, indicating a projection or selection operation.

first n.	last n.	location	salary
remy	w	LA	\$30
dan	s	seattle	\$50
dan	o	zurich	\$50

A diagram illustrating a relationship between the first and last names of individuals and their location. A curved orange arrow originates from the 'first n.' and 'last n.' columns and points to the 'location' column.

first n.	last n.	location	salary	course
remy	w	LA	\$30	143
remy	w	LA	\$30	240
remy	w	LA	\$30	249
dan	s	seattle	\$50	344
dan	s	seattle	\$50	444
dan	o	zurich	\$50	101
dan	o	zurich	\$50	113

name	job	location	salary	tax %
remy	prof	LA	\$30	20
dan	prof	seattle	\$50	15
vincent	TA	LA	\$20	10

$\text{job} \rightarrow \text{salary}$

$\text{name} \rightarrow \text{location} \quad \Rightarrow \quad \text{name, job} \rightarrow \text{tax \%}$

$\text{location, salary} \rightarrow \text{tax \%}$

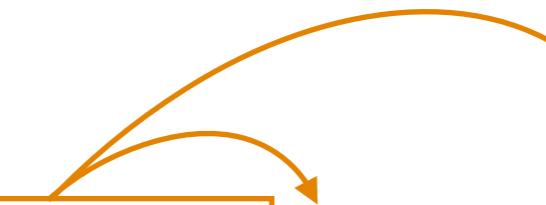
Decomposition

Factor out violating FDs

$$X \rightarrow Y : \begin{cases} Y \subseteq X \text{ (trivial FD)} \\ X \text{ is a superkey} \end{cases}$$

Make new table over $X \cup Y$

Drop Y from old table (keep X)

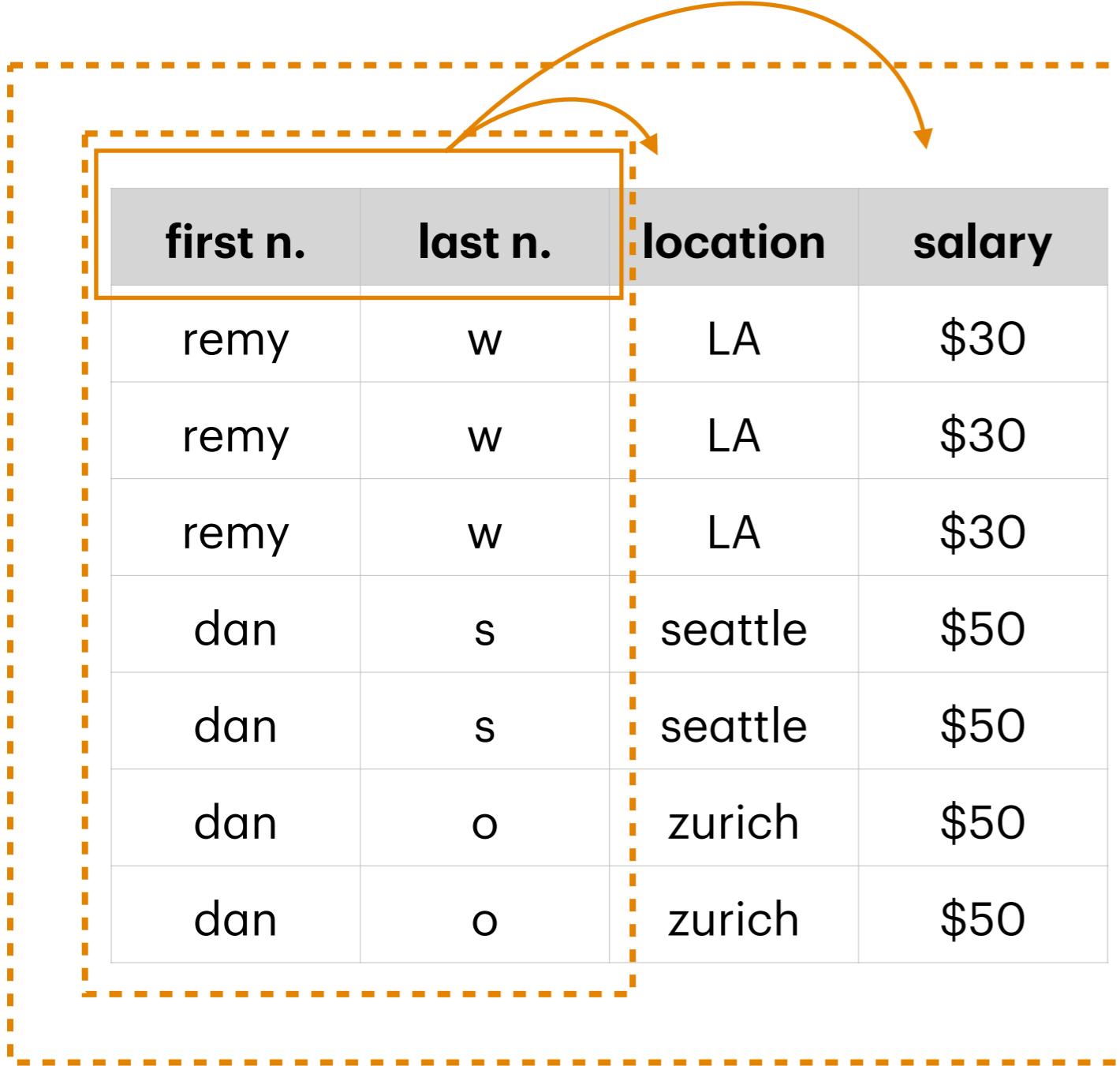


first n.	last n.	location	salary	course
remy	w	LA	\$30	143
remy	w	LA	\$30	240
remy	w	LA	\$30	249
dan	s	seattle	\$50	344
dan	s	seattle	\$50	444
dan	o	zurich	\$50	101
dan	o	zurich	\$50	113

The diagram illustrates a table with a dashed orange border. A curved arrow originates from the top of the 'first n.' column and points to the top of the 'last n.' column, indicating a relationship or dependency between these two fields.

first n.	last n.	location	salary	course
remy	w	LA	\$30	143
remy	w	LA	\$30	240
remy	w	LA	\$30	249
dan	s	seattle	\$50	344
dan	s	seattle	\$50	444
dan	o	zurich	\$50	101
dan	o	zurich	\$50	113

first n.	last n.	location	salary
remy	w	LA	\$30
remy	w	LA	\$30
remy	w	LA	\$30
dan	s	seattle	\$50
dan	s	seattle	\$50
dan	o	zurich	\$50
dan	o	zurich	\$50



The diagram illustrates a query mapping from the 'location' column of the first table to the 'course' column of the second table. The first table has a dashed orange border around its first two columns ('first n.' and 'last n.'), and the second table has a dashed orange border around its first two columns ('first n.' and 'last n.').

first n.	last n.	course
remy	w	143
remy	w	240
remy	w	249
dan	s	344
dan	s	444
dan	o	101
dan	o	113

job → salary

name → location

location, salary → tax %

name	job	location	salary	tax %
remy	prof	LA	\$30	20
dan	prof	seattle	\$50	15
vincent	TA	LA	\$20	10

3rd Normal Form

$$X \rightarrow Y : \begin{cases} Y \subseteq X \text{ (trivial FD)} \\ X \text{ is a superkey} \\ \forall y \in Y : y \in \text{key} \end{cases}$$

3rd Normal Form

$$X \rightarrow Y : \begin{cases} Y \subseteq X \text{ (trivial FD)} \\ X \text{ is a superkey} \\ \forall y \in Y : y \in \text{key} \end{cases}$$

never loses FDs

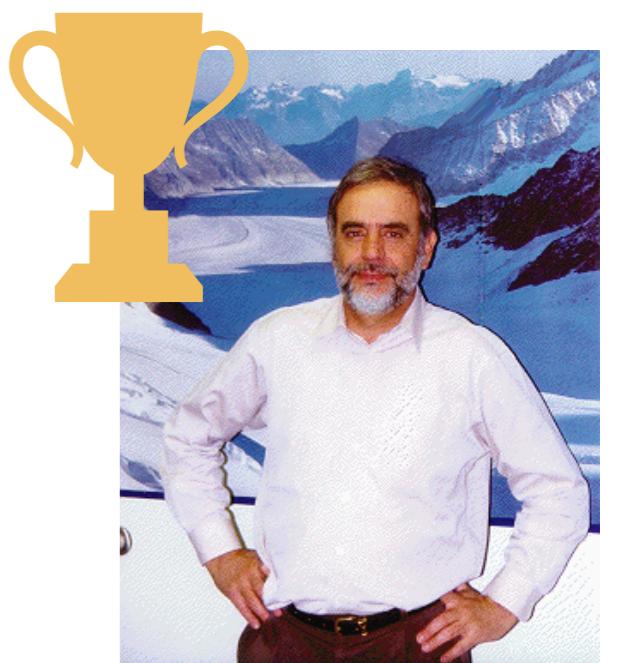
decomposition in P time

3rd Normal Form

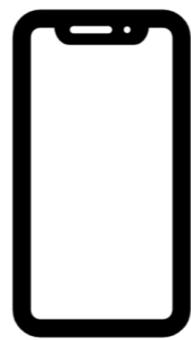
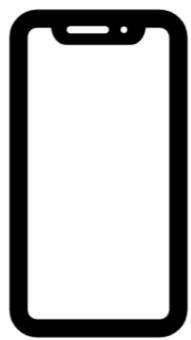
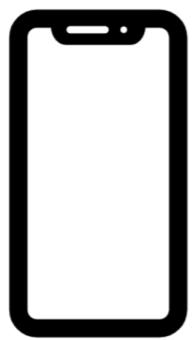
$$X \rightarrow Y : \begin{cases} Y \subseteq X \text{ (trivial FD)} \\ X \text{ is a superkey} \\ \forall y \in Y : y \in \text{key} \end{cases}$$

never loses FDs

decomposition in P time



Transactions: it's a wild world out there



exercise 1: free points for all!



atomicity

a TX either completes

or leaves no trace

exercise 2: playing favorites!



consistency

a TX should leave the DB

in a consistent state

exercise 3: armageddon!



isolation

multiple concurrent TX

should not interfere

exercise 4: blackout!

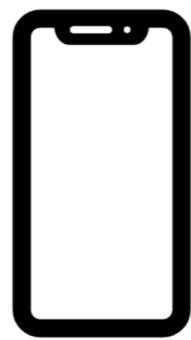
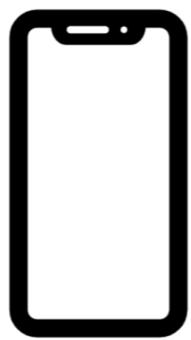
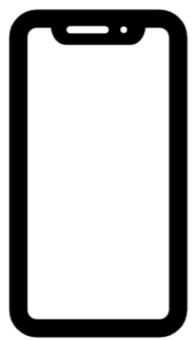


durability



completed TX are forever

SQLite demo



```
import sqlite3 as sql

con = sql.connect("bank.db")
cur = con.cursor()

res = cur.execute("SELECT * FROM acc")
```

transaction

a set of read & write ops

T1

READ(A, t)

$t := t + 100$

WRITE(A, t)

READ(B, t)

$t := t + 100$

WRITE(B,t)

A,B are
elements
in the DB

t is a local
variable
in the app

T1

READ(A, t)

$t := t + 100$

WRITE(A, t)

READ(B, t)

$t := t + 100$

WRITE(B,t)

A,B are elements in the DB

t is a local variable in the app

T2

READ(A, s)

$s := s^*2$

WRITE(A,s)

READ(B,s)

$s := s^*2$

WRITE(B,s)

transaction

a set of read & write ops

A either execute all, or nothing

C

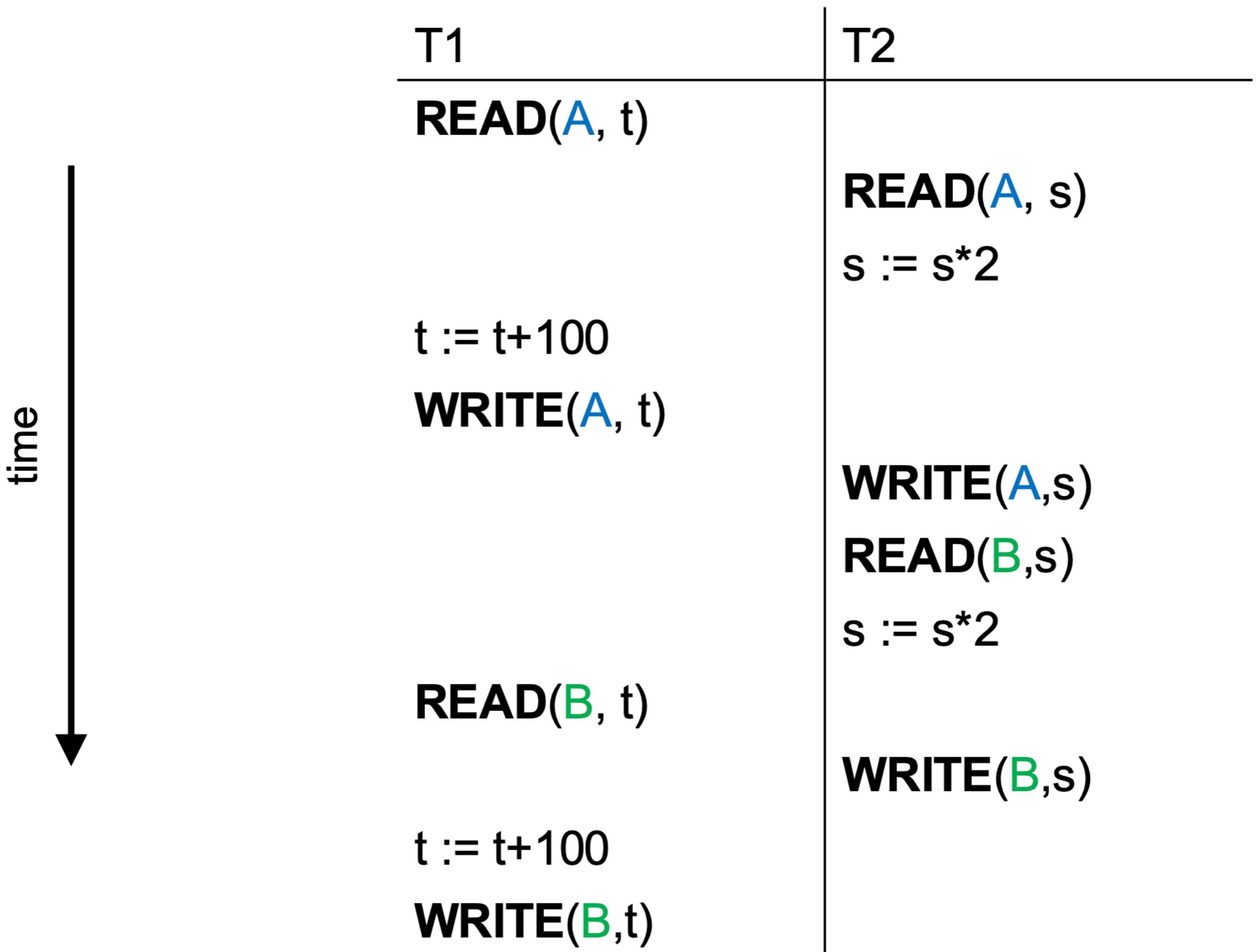
I 2 transactions do not interfere

D

schedule

an interleaving & read/writes

from different TXs



serial schedule

"one at a time!"

time

A = 2
B = 2

T1

READ(A, t)

$t := t + 100$

WRITE(A, t)

READ(B, t)

$t := t + 100$

WRITE(B,t)

T2

READ(A, s)

$s := s^*2$

WRITE(A,s)

READ(B,s)

$s := s^*2$

WRITE(B,s)

T1	T2
A = 2 B = 2	READ(A, s) $s := s^*2$
A = 4 B = 4	WRITE(A,s) READ(B,s) $s := s^*2$ WRITE(B,s)
READ(A, t) $t := t+100$ WRITE(A, t) READ(B, t) $t := t+100$ WRITE(B,t)	
A = 104 B = 104	

time



serializable schedule

equivalent to "one at a time!"

T1

READ(A, t)

$t := t + 100$

WRITE(A, t)

READ(B, t)

$t := t + 100$

WRITE(B,t)

T2

A = 2
B = 2

A = 102
B = 2

A = 204
B = 2

A = 204
B = 102

A = 204
B = 204

serial → isolation

Serializable = serial

Serializable → isolation