- All T's commit only after all T's whose changes they read have committed

- A T is strict if T is recoverable & when T is writing an object no other T can read/write until T is committed

T_1	T_2
R(A)	
R(B)	
W(A)	
	R(A)
C	
	R(B)
	C

(2) 1) If T_2 reads init value in S_1, it does so also in S_2.

2) If T_2 writes O last in S_2, it does so also in S_1

3) If T_2 reads a value written by T_j in S_1 it does so also in S_2.

T_1	T_2	T_3
R(A)		
	W(A)	
	C	
W(A)		
C		
		W(A)
		C

3. For $S = (R\_1(A), R\_2(A), W\_2(A), W\_1(A))$.

T1	T2
R	
	R
	W
W	

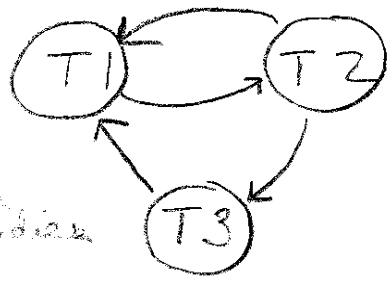
allowed $RTS(A) = TS(T1)$

Abort — as $TS(T1) < RTS(A)$

$RTS(A) = TS(T2)$ (allowed)
$WTS(A) = TS(T2)$ (allowed)

4.)

T1	T2	T3
W(A)		
	W(B)	
W(B)		
		W(A) younger than T1 - therefore dies
	W(A) T2 younger - therefore dies	



Wait or die: If T has higher priority it is allowed to wait otherwise it is aborted.

5

Problems:

Checkpoints:

① begin-check point record is written to the log file

② Construct an end-check point record by copying transaction & dirty page tables to buffer, then force writing to stable storage.
Must write all log file to up to end-checkpoint to disk.

③ Master record of LSN of the begin-check point to some known position on the disk.

[redacted]

Practice Final

6. ARIES uses a steal, no-force approach to page writing. In regards to steal, dirty pages in the buffer pool can be written to disk even if the transaction that made the changes have not committed. By no-force, it means that it isn't forced to write dirty pages of a committed transaction to disk.
A Compensation Log Record (CLR) describes the action taken to undo the actions recorded in the corresponding update log record and is appended to the log tail.

7. Aries would follow the following steps

① Analysis

T2, T3 need to be undone T1 aborts so doesn't need to be undone aga

P3, P4

② Redo - The algorithm would redo all the items in the log tail up to the last checkpoint.

③ Undo - The algorithm would undo all Transactions that didn't complete

LSN	LOG
00, 05	begin-checkpoint, end-checkpoint
10	update: T1 writes P2
20	update: T3 writes P4
50	update: T2 writes P3
55	T1 aborts
60	CLR: undo T1 LSN 10
65	T1 End
70	T2 commit
75 X	crash, restart
80	CLR: undo T2 LSN 50
90	CLR: T2 End
100	CLR: undo T3 LSN 20
190	CLR: T3 End

8 (a) One might leave a table in 3NF rather than put it in BCNF if a dependency might be lost by the stronger normal form or if have a query that would be done "all in one table" if didn't do split.
i.e. Suppose had R(a, b, c)
and FD's ab → c, b → b
then in 3NF but not BCNF.
a BCNF decomposition would lose the dependency ab → c
If also had query
select b, c
from R
where a > 10 and it was a frequent query, might also want to denormalize

(b) Horizontal partitioning is the splitting of a table into two or more tables based on attributes of a row value.
Ex ~~Car Model~~
Have Car(~~Model~~ model, options)
and split based on model to get tables like
CarCivic(model, options)
CarAccord(model, options)
(might do if queries tended to lie within one such car model.

9

Discretionary access control: - each object has a list of privileges and there is a method for granting priviledges on objects to users.

Mandatory access control - each object has a security class & each user is assigned clearance to see some classes.

REVOKE SELECT ON Sailors FROM Bob;

10)

CREATE TYPE CSET AS OBJECT