Review

- Consider the following sequence of lock requests:
 l₁(B); l₂(A); l₃(C); l₁(C); l₂(B); l₃(A)
- Assume that upon start, transactions T_1 , T_2 , T_3 were assigned timestamps 10, 20, 30, respectively. What is the order in which transactions commit in a wait-die scheme?
- What is the order in which transactions commit in a wound-wait scheme?

CS5208 - Crash Recovery

Review

```
l_1(B); l_2(A); l_3(C); l_1(C); l_2(B); l_3(A)
```

- wait-die scheme T_3, T_1, T_2
- wound-wait scheme T₁, T₂, T₃

CS5208 - Crash Recovery

Review Four transactions T1, T2, T3, T4 used 2PL for concurrency control. Since 2PL ensures conflict-serializability, the schedule (say S) of actions of the four transactions has to be conflict equivalent to some serial schedule. We do not have access to the entire schedule S but only a part of it, which Log-Based Recovery Schemes looks as follows: $S = :::; u_4(A); l_1(B); ::::; u_1(B); l3(A); l_2(B); ::::; u_3(A); l_2(A); ::::$ If you are going to be in the logging business, one of the things that you Which of the following schedules are possible serial schedules that are have to do is to learn about heavy conflict-equivalent to S. (a) T_1, T_3, T_2, T_4 equipment. Robert VanNatta, (b) T₁, T₄, T₃, T₂ Logging History of (c) T_4 , T_1 , T_3 , T_2 (Correct) Columbia County CS5208 - Crash Recovery CS5208 - Crash Recovery

Integrity or consistency constraints

- Predicates data must satisfy, e.g.
 - x is key of relation R
 - $x \rightarrow y$ holds in R
 - Domain(x) = {Red, Blue, Green}
 - no employee should make more than twice the average
 - salary
- Definitions
 - <u>Consistent state:</u> satisfies all constraints
 - <u>Consistent DB:</u> DB in consistent state

CS5208 - Crash Recovery

Observation:

DB cannot always be consistent!







































































Redo logging (deferred modification)

- In UNDO logging, we remember the "old" values.
- How about remembering the "new" (updated) values instead?
- What does this mean?

Redo logging (deferred modification)

- In UNDO logging, we remember the "old" value.
- How about remembering the "new" (updated) values instead?
- What does this mean?

CS5208 - Crash Recovery

• NO updates must be written to disk until a transaction commits! So?

Redo logging (deferred modification)

- In UNDO logging, we remember the "old" value.
- How about remembering the "new" (updated) values instead?
- What does this mean?
 - NO updates must be written to disk until a transaction commits!
 - All updates have to be buffered in memory!

CS5208 - Crash Recovery





















- Undo logging: increase the number of disk I/Os
- *Redo logging:* need to keep all modified blocks in memory until commit

CS5208 - Crash Recovery





















