CS186 Discussion 9

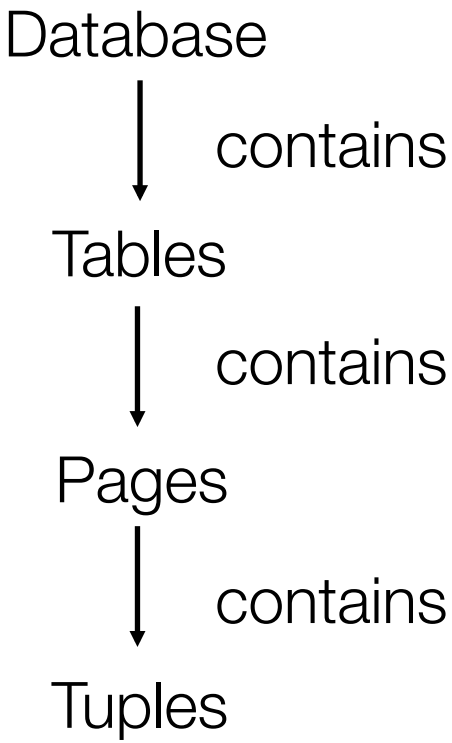
(Lock Granularity, Multiversion Concurrency Control, 2PC)

Matthew Deng

Lock Granularity

Lock Granularity

- Select amount to lock
- Used with 2PL to guarantee serializability
- Steps:
  1. Start at root
  2. Get locks top-down
  3. Release locks bottom-up



Intent Locks

- S: shared lock for reading
- X: exclusive lock for writing (and reading)
- IS: intent to get S lock(s) at finer granularity
- IX: intent to get X lock(s) at finer granularity
- SIX: shared lock with intent to get X lock(s) at finer granularity

Lock Compatibility Matrix

	S	X
S	√	–
X	–	–

Lock Compatibility Matrix

	IS	IX	SIX	S	X
IS					
IX					
SIX					
S				√	–
X				–	–

Lock Compatibility Matrix

	IS	IX	SIX	S	X
IS	√	√	√	√	–
IX	√	√	–	–	–
SIX	√	–	–	–	–
S	√	–	–	√	–
X	–	–	–	–	–

Lock Granularity Exercises

Lock Granularity Exercises

1. Suppose a transaction, T1, wants to scan a table R and update a few of its tuples. What kind of locks should T1 have on R, its pages, and the tuples that are updated?

Lock Granularity Exercises

1. Suppose a transaction, T1, wants to scan a table R and update a few of its tuples. What kind of locks should T1 have on R, its pages, and the tuples that are updated?

   SIX lock on R

   IX lock on pages

   X lock on updated tuples

Lock Granularity Exercises

2. Is an S lock compatible with an IX lock? Explain why or why not. Make your description as simple as possible.

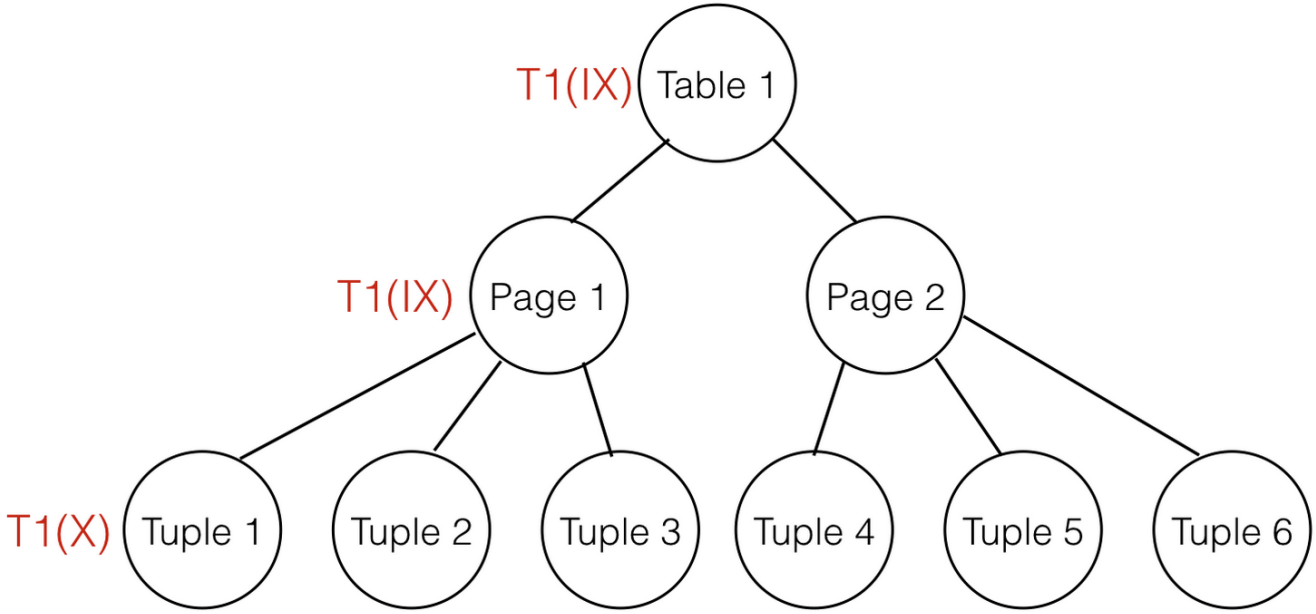
Lock Granularity Exercises

2. Is an S lock compatible with an IX lock? Explain why or why not. Make your description as simple as possible.

   Suppose T1 wants an S lock on an object, O, and T2 wants an IX lock on the same object O. An S lock implies that T1 will read the entire object (all of its sub-objects). An IX lock implies that T2 will write some of the sub-objects of the object. This means that there is some sub-object of O that T1 will read and T2 will write. This is not valid, so the S and IX locks must be incompatible.

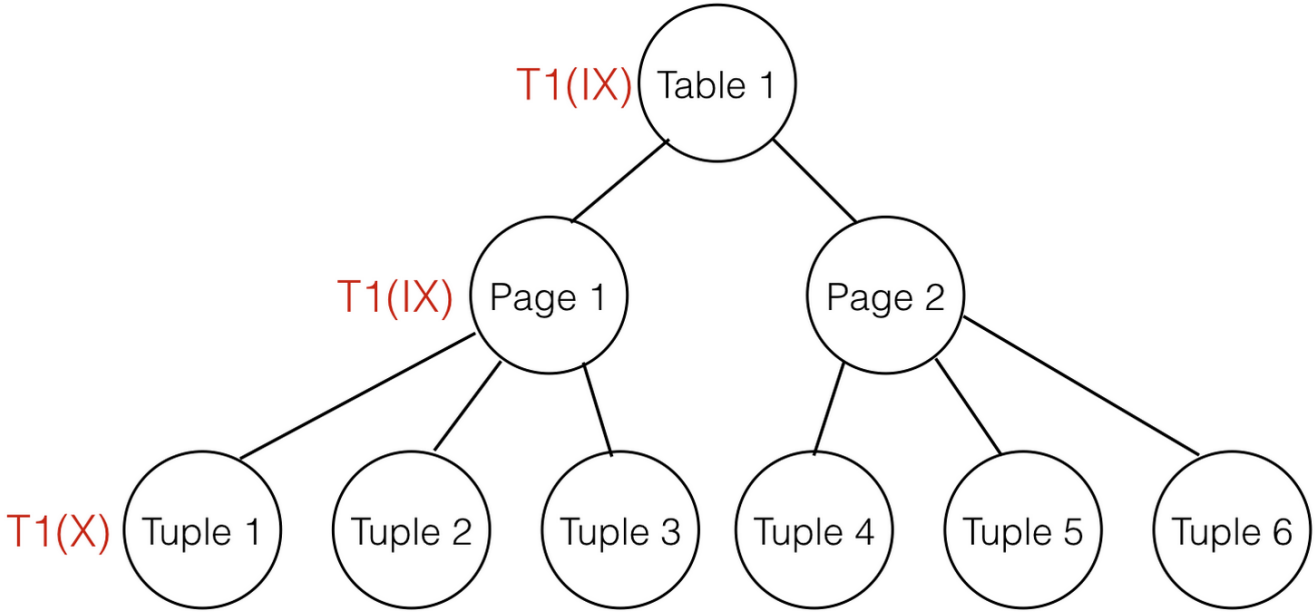
Lock Granularity Exercises

3. Given that transaction T1 has an IX lock on table 1, an IX lock on page 1, and an X lock on tuple 1, which locks could be granted to transaction T2 for tuple 2?



Lock Granularity Exercises

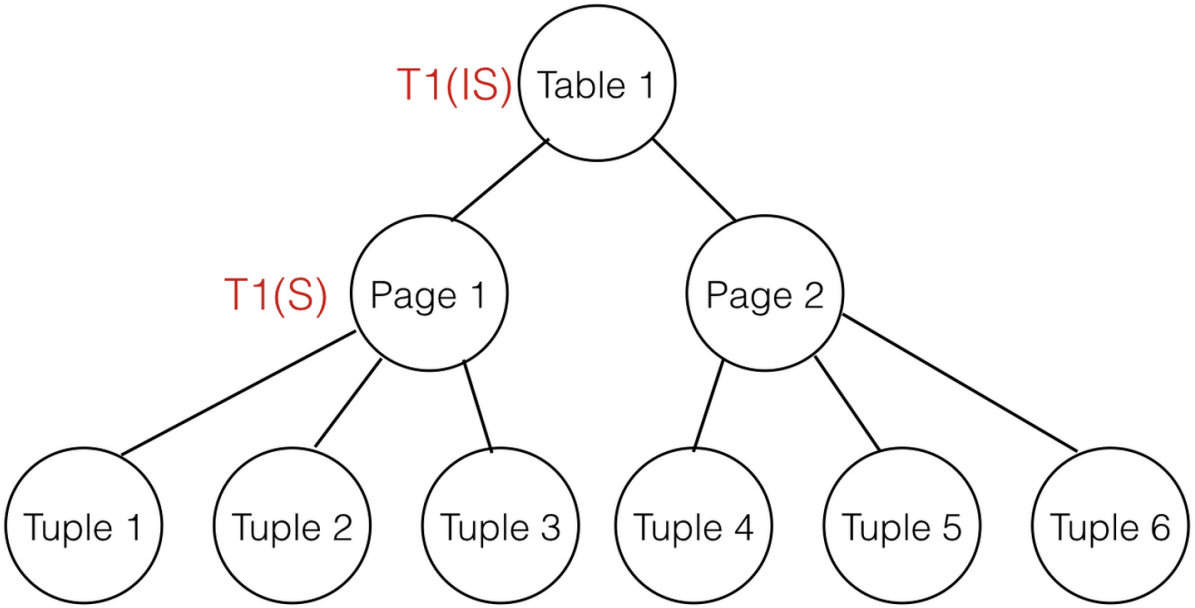
3. Given that transaction T1 has an IX lock on table 1, an IX lock on page 1, and an X lock on tuple 1, which locks could be granted to transaction T2 for tuple 2?



X, S

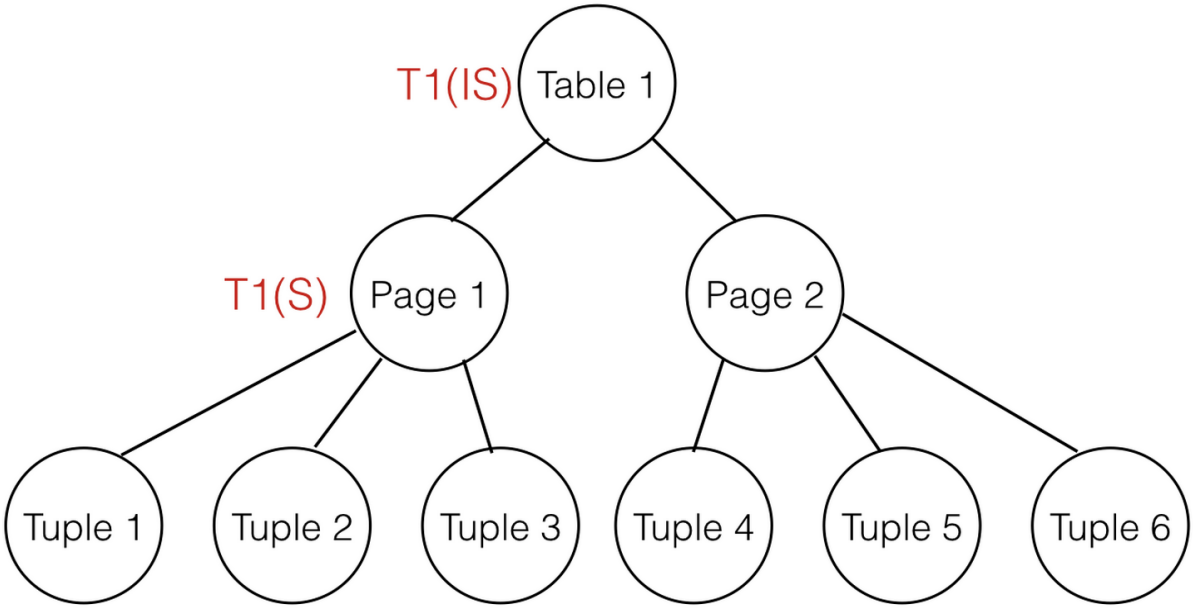
Lock Granularity Exercises

4. Given that T1 has an IS lock on table 1 and an S lock on page 1, what locks could be granted to T2 for page 1?



Lock Granularity Exercises

4. Given that T1 has an IS lock on table 1 and an S lock on page 1, what locks could be granted to T2 for page 1?



S, IS

Multiversion Concurrency Control

Multiversion Concurrency Control

- Timestamp Ordered Multiversion Concurrency Control
- Alternative to 2PL

Multiversion Concurrency Control

- For each object, keep track of its “timeline”
  - Reads: {R-ts}
  - Versions: {<W-ts, value>}

- $Read_{ts}(x)$:

  Read value of version with with highest W-ts < ts

  Add ts to Reads

- $Write_{ts}(x, v)$:

  If lowest R-ts > ts comes before lowest W-ts > ts:

  abort

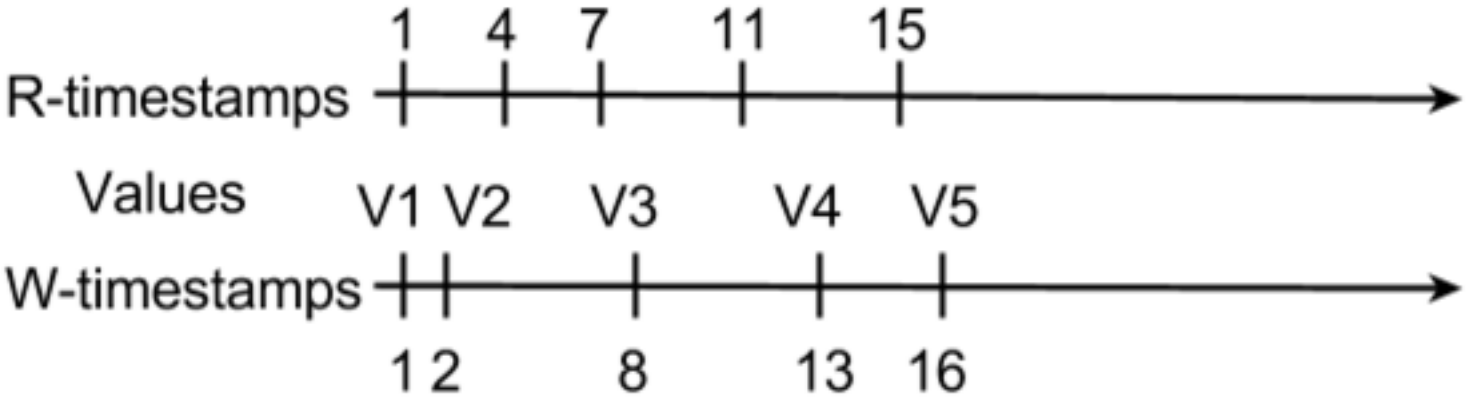
  Else:

  add <ts, v> to Versions

Multiversion Concurrency Control Exercises

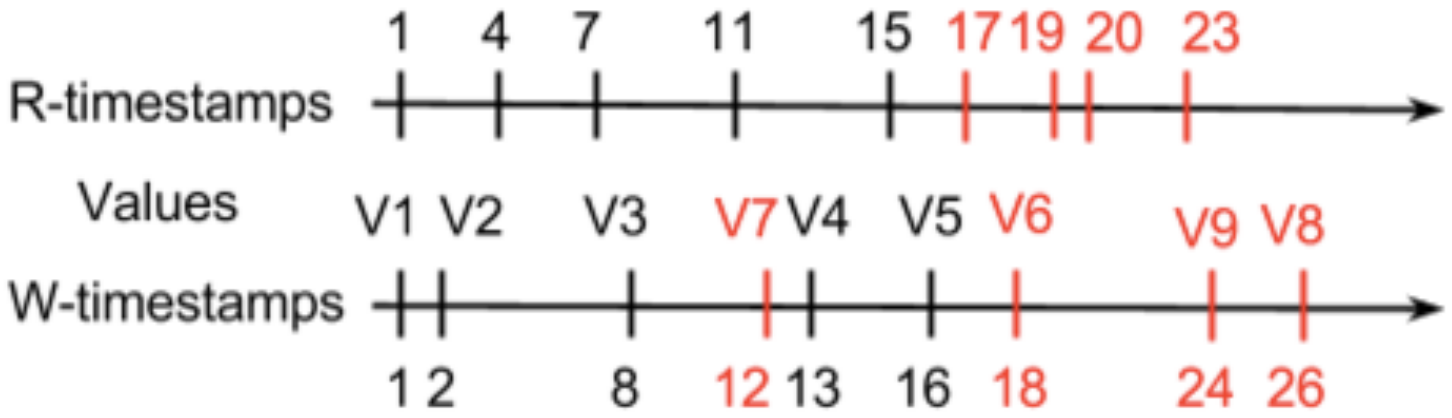
MVCC Exercises

1. For the given timeline of timestamps for a value X, fill out the rest of the timeline for the following reads and writes: R(X)@17, W(X)@18, W(X)@14, W(X)@12, R(X)@20, R(X)@19, R(X)@23, W(X)@26, W(X)@24



MVCC Exercises

1. For the given timeline of timestamps for a value X, fill out the rest of the timeline for the following reads and writes: R(X)@17, W(X)@18, W(X)@14, W(X)@12, R(X)@20, R(X)@19, R(X)@23, W(X)@26, W(X)@24



Two Phase Commit

Two Phase Commit

- Distributed database concurrency control
- Phase 1 (voting phase):
  - Coordinator asks participants
  - Participants vote yes or no
- Phase 2 (commit phase):
  - If all participants voted yes, commit
  - Else, abort

Two Phase Commit Exercises

Two Phase Commit Exercises

1. What ACID properties does TPC ensure? How does TPC ensure these properties?

Two Phase Commit Exercises

1. What ACID properties does TPC ensure? How does TPC ensure these properties?

   TPC ensures atomicity and durability by ensuring that a write happens across ALL replicas or NONE of them.

Two Phase Commit Exercises

2. What command does the MASTER need to issue in order to persist the changes of a transaction? What needs to happen in order for this command to be issued?

Two Phase Commit Exercises

2. What command does the MASTER need to issue in order to persist the changes of a transaction? What needs to happen in order for this command to be issued?

   Must issue a GLOBAL-COMMIT. This can only happen when all of the slaves have voted to commit.

Two Phase Commit Exercises

3. Preview: What role does logging play in TPC?

Two Phase Commit Exercises

3. Preview: What role does logging play in TPC?

   Failure handling