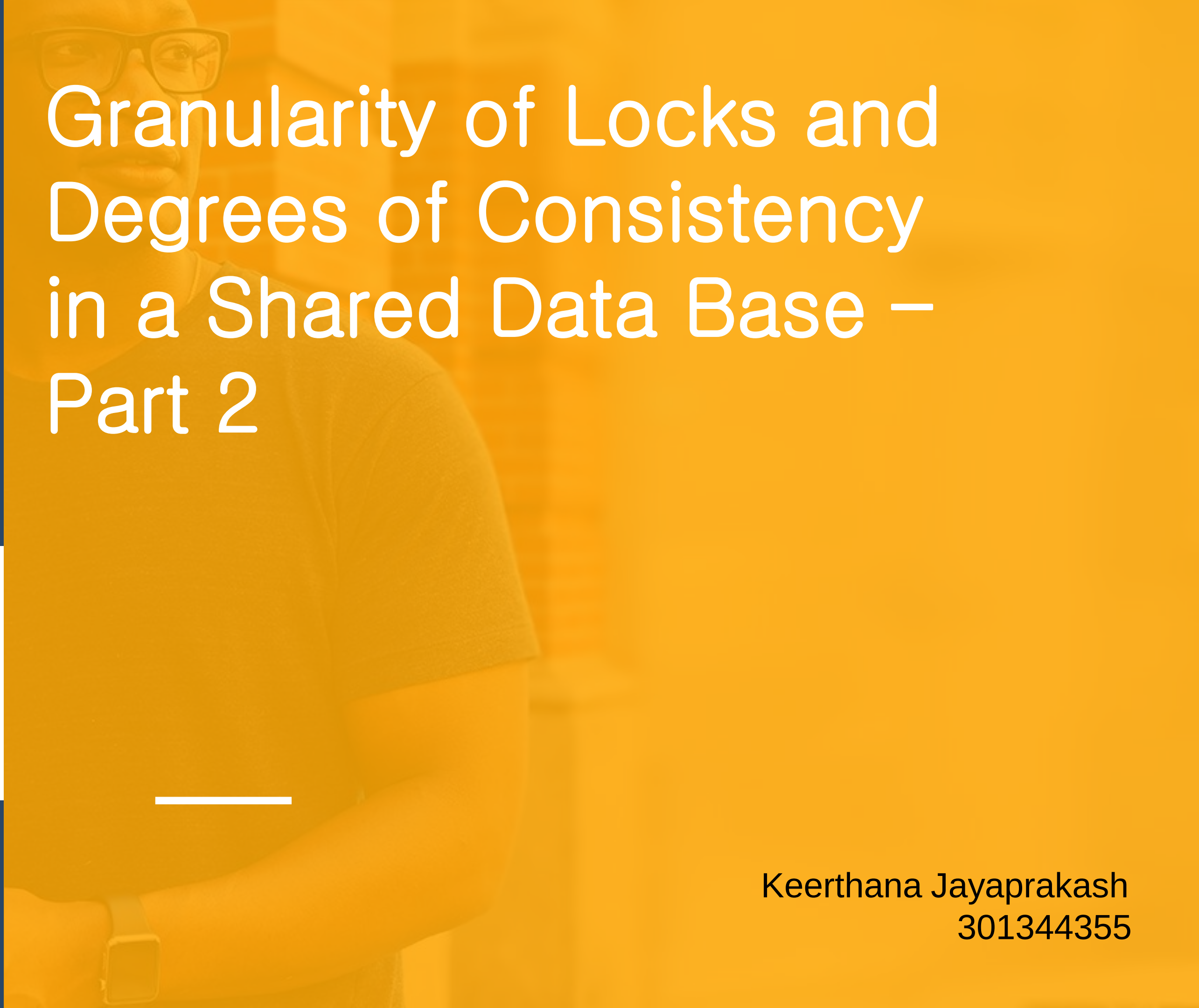
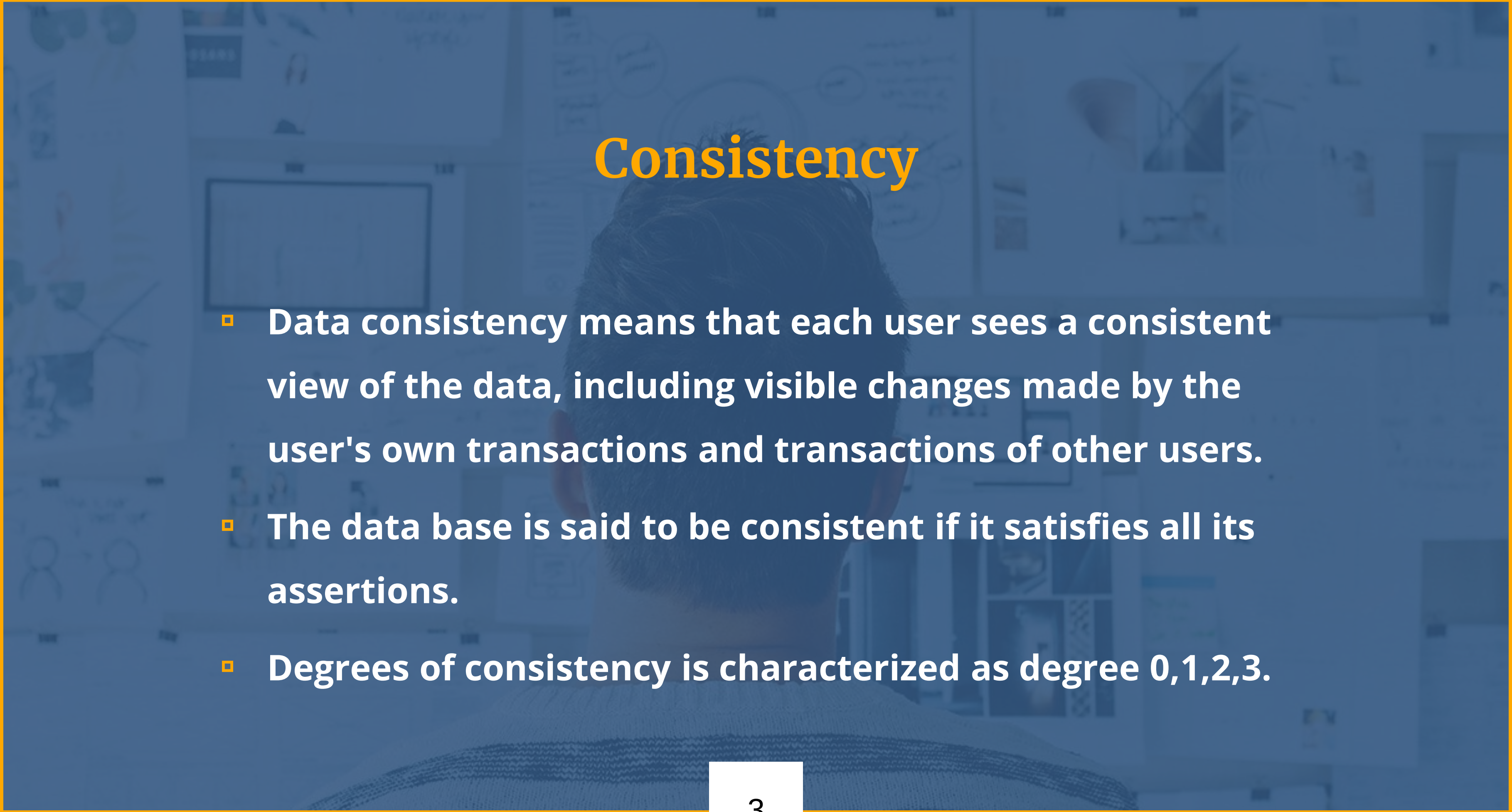
Granularity of Locks and Degrees of Consistency in a Shared Data Base – Part 2



Keerthana Jayaprakash
301344355

Agenda

- Consistency
- Transactions
- Dirty Data
- Transaction Recovery
- Lock Protocols
- Consistency of Schedules
- Transaction Backup and System Recovery
- Summary



Consistency

- Data consistency means that each user sees a consistent view of the data, including visible changes made by the user's own transactions and transactions of other users.
- The data base is said to be consistent if it satisfies all its assertions.
- Degrees of consistency is characterized as degree 0,1,2,3.

Transactions

- Sequences of atomic actions are grouped to form transactions.
- Transactions are the units of consistency.
- Transactions are also the units of recovery.

Dirty Data

- **Dirty data : A transaction reads/write data that has been written by another transaction and has not been committed yet.**
- **Crux of concurrency is preventing the reading or writing of other transactions' dirty data.**

Dirty Data

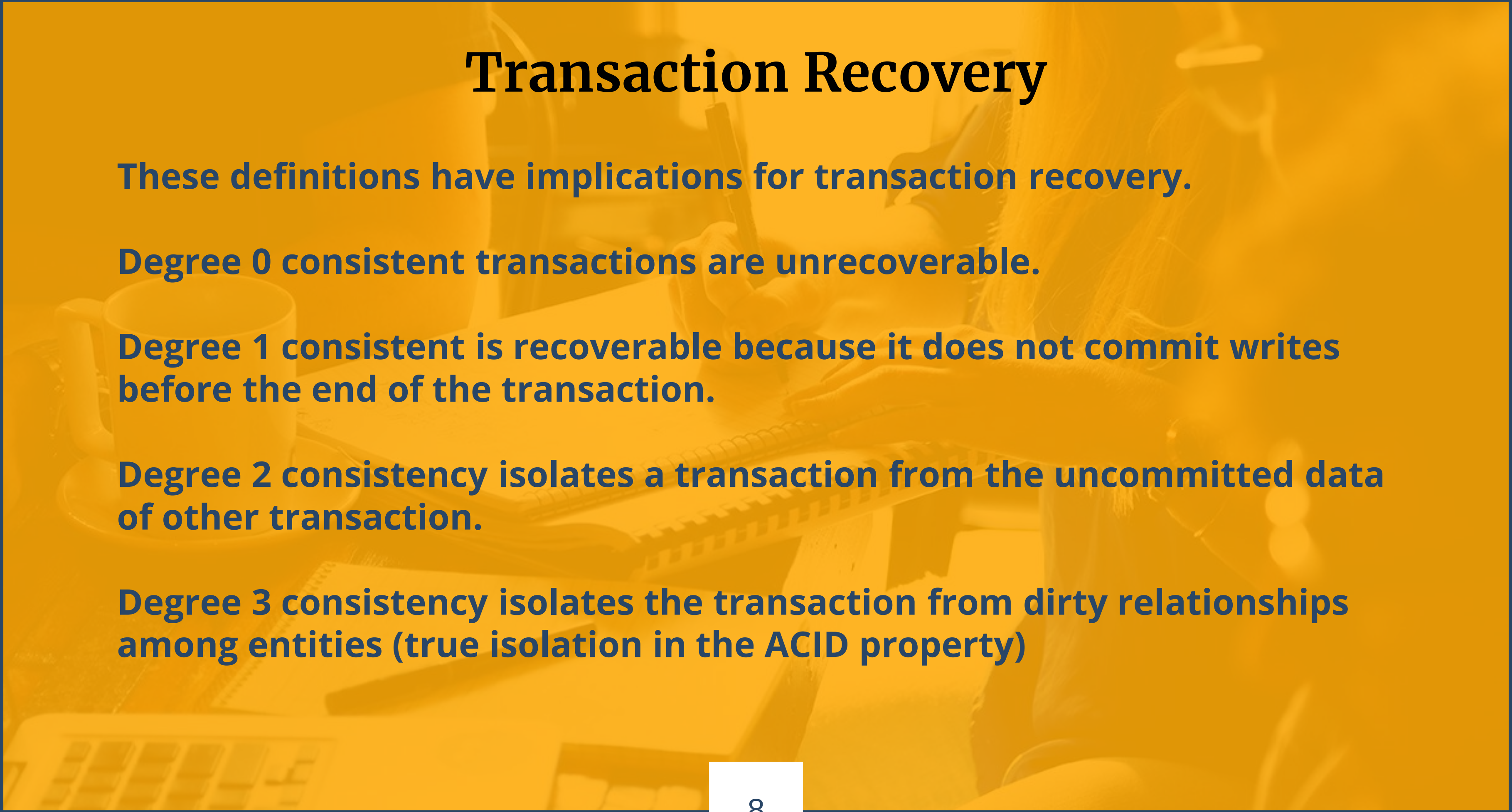
- **Degrees of consistency:** given the possible constraints on behavior of transaction T:

   1. **T does not overwrite dirty data of other transactions.**
   2. **T does not commit any writes until it completes all its writes (i.e. until the end of transaction (EOT)) .**
   3. **T does not read dirty data from other transactions.**
   4. **Other transactions do not dirty any data read by T before T completes.**

Dirty Data

- Degree 3: T satisfies 1,2,3,4
- Degree 2: T satisfies 1,2,3
- Degree 1: T satisfies 1,2
- Degree 0: T satisfies 1

Transaction Recovery



These definitions have implications for transaction recovery.

Degree 0 consistent transactions are unrecoverable.

Degree 1 consistent is recoverable because it does not commit writes before the end of the transaction.

Degree 2 consistency isolates a transaction from the uncommitted data of other transaction.

Degree 3 consistency isolates the transaction from dirty relationships among entities (true isolation in the ACID property)

Lock Protocols

- A lock is simply a mechanism to control concurrent access to data item.
- A locking protocol is a set of rules which are followed by all transactions while requesting and releasing locks.
- Locks are of 2 types - share mode and exclusive mode locks.
- By duration – short duration locks and long duration locks.

Lock Protocols

- **Degrees of consistency:** given the possible constraints on behavior of transaction T:
- T observes degree 3 lock protocol if :
  - T sets a long exclusive lock on any data it dirties.
  - T sets a long share lock on any data it reads.

Lock Protocols

- T observes degree 2 lock protocol if :
  - T sets a long exclusive lock on any data it dirties.
  - T sets a short share lock on any data it reads.
- T observes degree 1 lock protocol if :
  - T sets a long exclusive lock on any data it dirties.
- T observes degree 0 lock protocol if :
  - T sets a short exclusive lock on any data it dirties.

Lock Protocol – Redefined

- A transaction is well formed with respect to writes(reads) if it always locks an entity in exclusive(shared) mode before writing(reading) it.
- A transaction is well formed if it is well formed with respect to read and writes.
- A two phase transaction is adequate to insure consistency.

Lock Protocol – Redefined

Degree 3 : T is well formed and T is two phase.

Degree 2 : T is well formed and T is two phase with respect to writes.

Degree 1 : T is well formed with respect to writes and T is two phase with respect to writes.

Degree 0: T is well formed with respect to writes.

Consistency of Schedules

- A transaction is any sequence of actions beginning with a begin action and ending with an end action and not containing other begin or end actions.
- Any (sequence preserving) merging of the actions of a set of transactions into a single sequence is called a schedule for the set of transactions.

Consistency of Schedules

- One transaction at a time schedules are called serial schedules because they have no concurrency among transactions.
- A schedule is legal only if it does not schedule a lock action on an entity for one transaction when that entity is already locked by some other transaction in a conflicting mode.

Consistency of Schedules

- A transaction runs at degree 0(1,2,3) consistency in schedule S if T sees degree 0(1,2,3) consistency in schedule S.
- If all transactions run at degree 0(1,2,3) consistency in schedule S then S is said to be a degree 0(1,2,3) consistent schedule.

Assertion 1

A) If each transaction observes the degree 0 (1, 2 or 3) lock protocol then any legal schedule is degree 0 (1, 2 or 3) consistent.

B) Unless transaction T observes the degree 1 (2 or 3) lock protocol then it is possible to define another transaction T' which does observe the degree I (2 or 3) lock protocol such that T and T' have a legal schedule S but T does not run at degree 1 (2 or 3) consistency in S.

Assertion 2

If each transaction in a set of transactions at least observes the degree 0 lock protocol and if transaction T observes the degree I (2 or 3) lock protocol then T runs at degree 1 (2 or 3) consistency (Definitions 1, 3) in any legal schedule for the set of transactions. .

Transaction Backup and System Recovery

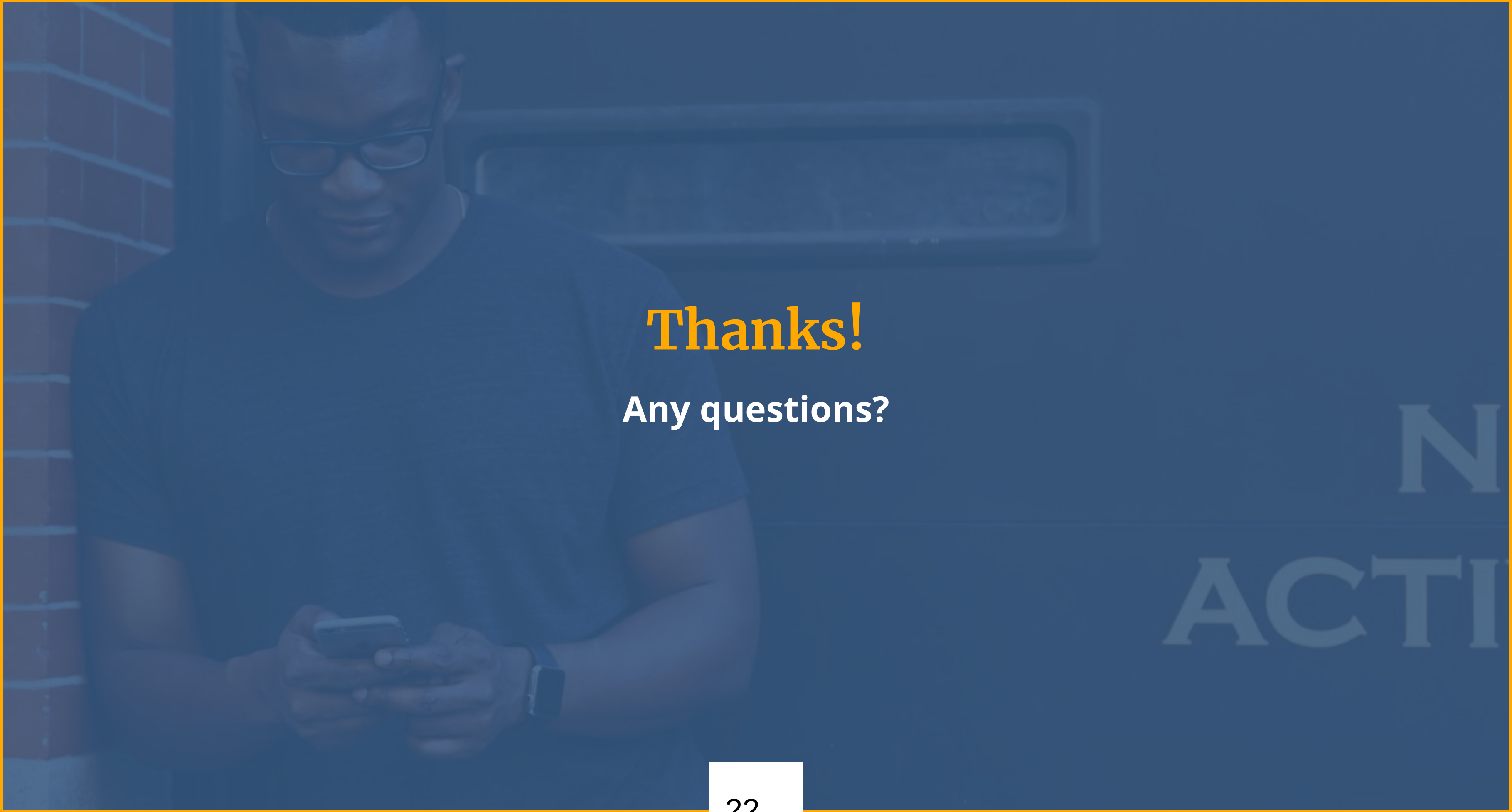
- Degree 1 transactions may read uncommitted (dirty) data, and transaction and system recovery may undo uncommitted updates. Therefore they may produce different results when re-run.

Transaction Backup and System Recovery

- Degree 2 transactions re-run in the order specified in the log will always result in the same consistent state.

Summary

- Degrees of consistency in notion of dirty data
- Degrees of consistency - Lock Protocols
- Degrees of consistency - Schedules
- Assertions
- Implications of transaction back up and system recovery



Thanks!

Any questions?