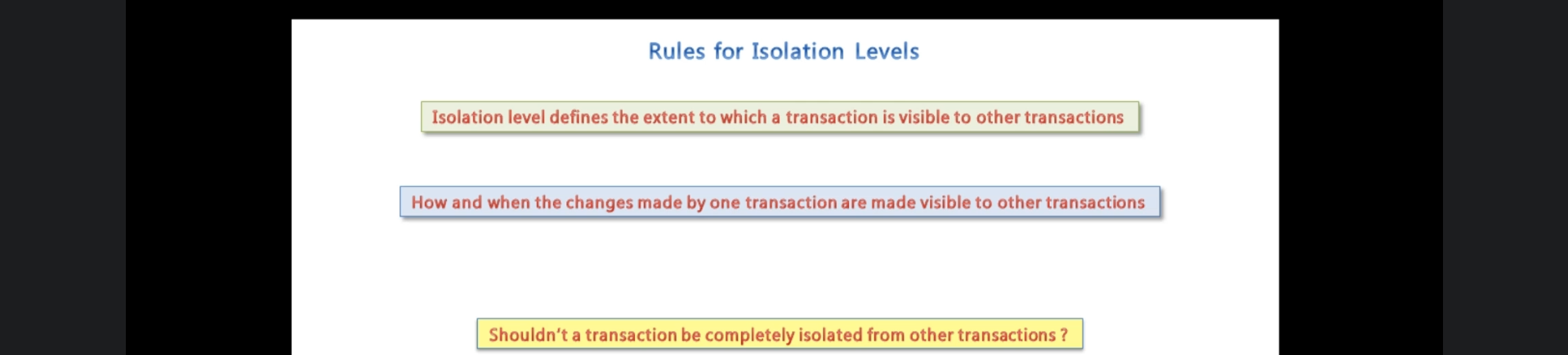
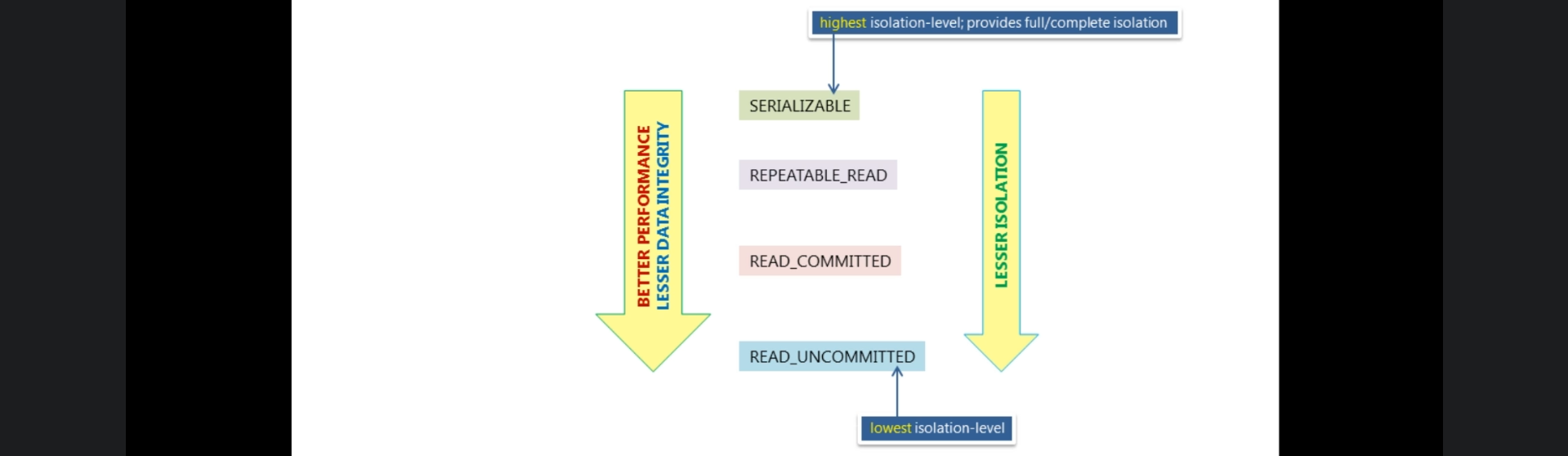
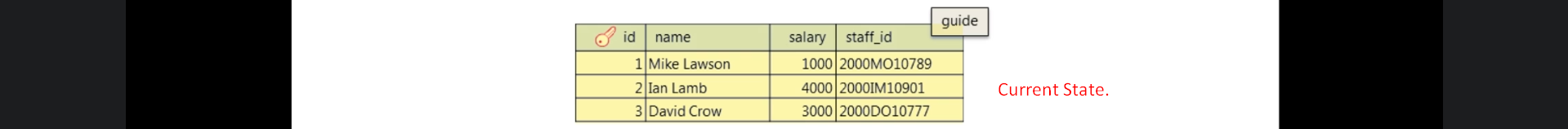
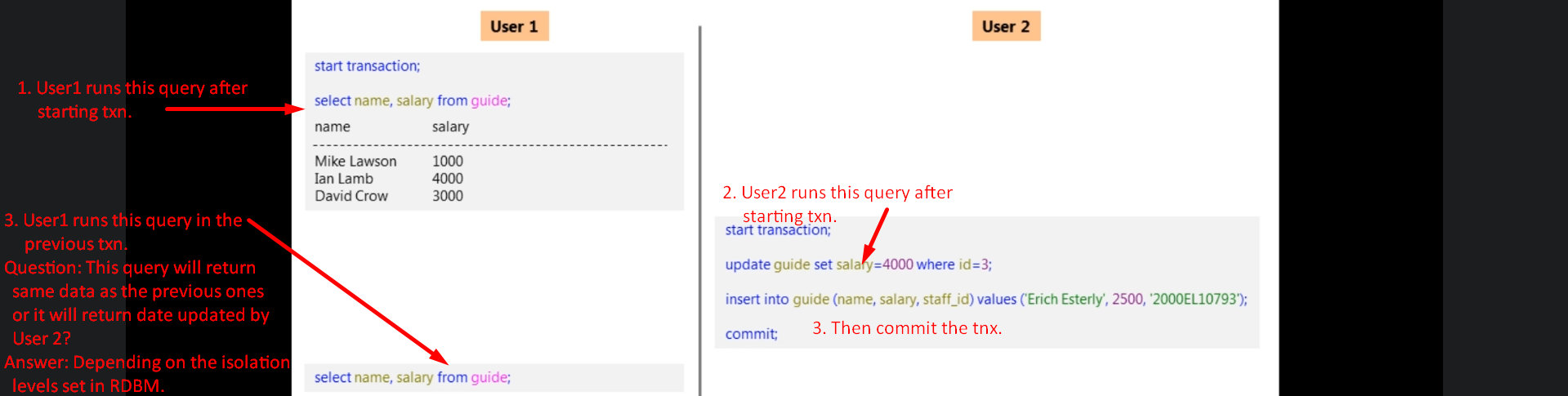
1. **Agenda**:
   1. Various isolation levels.
2. Isolation Level?
   1. It defines the extent to which a transaction is visible to other transaction.  
      It defines how and when changes made by one transaction are visible to other transactions.

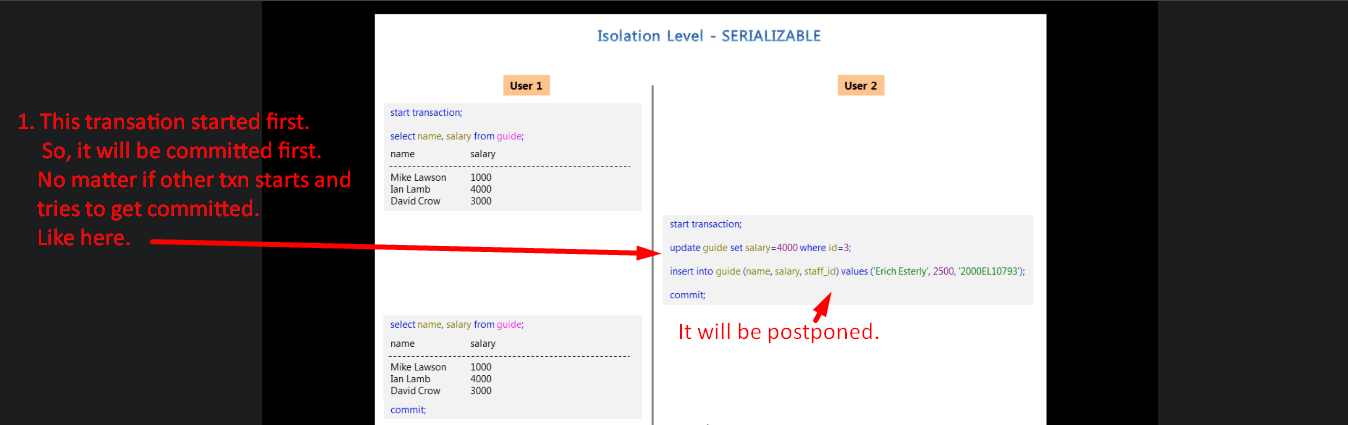
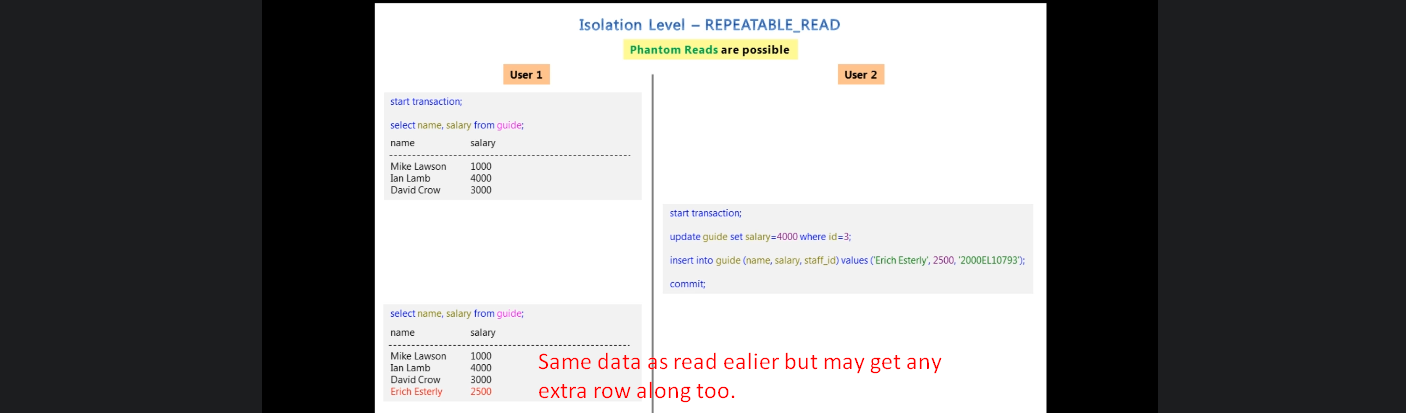


1. Question: You may ask that should not one transaction be completely isolated from other transactions?  
   Should not one transaction be completely invisible to other transactions?
2. Complete Isolation:
   1. **Disadvantage**: Database transaction isolation comes at a high price as in this case, the performance and scalability of a multi-user application might not actually be as good as you might like.
   2. **Advantage**: The more you isolate one transaction from other transactions; the better will be the data integrity of that system.



1. Let’s see a transaction is affected by other transactions running at the same time.
2. 
3. d
   1. User1 and User2 running at the same time.
   2. Down line is timeline.



1. We got 4 levels of transaction isolation.
   1. **Serializable Isolation Level:**
      1. You will get the same result as in the previous query as this isolation level provides highest level of isolation b/w transaction. (Complete isolation called true isolation)
      2. Until, the previous transaction is not committed, the next will be postponed.  
         So, transactions get committed in sequence.
   2. 
   3. 
   4. **Disadvantages**:
      1. Performance gets slow.
      2. Other issues too.
2. **Repeatable Read Isolation Level**:
   1. This isolation level is called **Softer Isolation Level (one of softer isolation levels) as Softer than Serializable Isolation Level.**
   2. 
3. **Read Committed Isolation Level**:
   1. 
4. **Read Uncommitted Isolation Level**:
   * 1. 