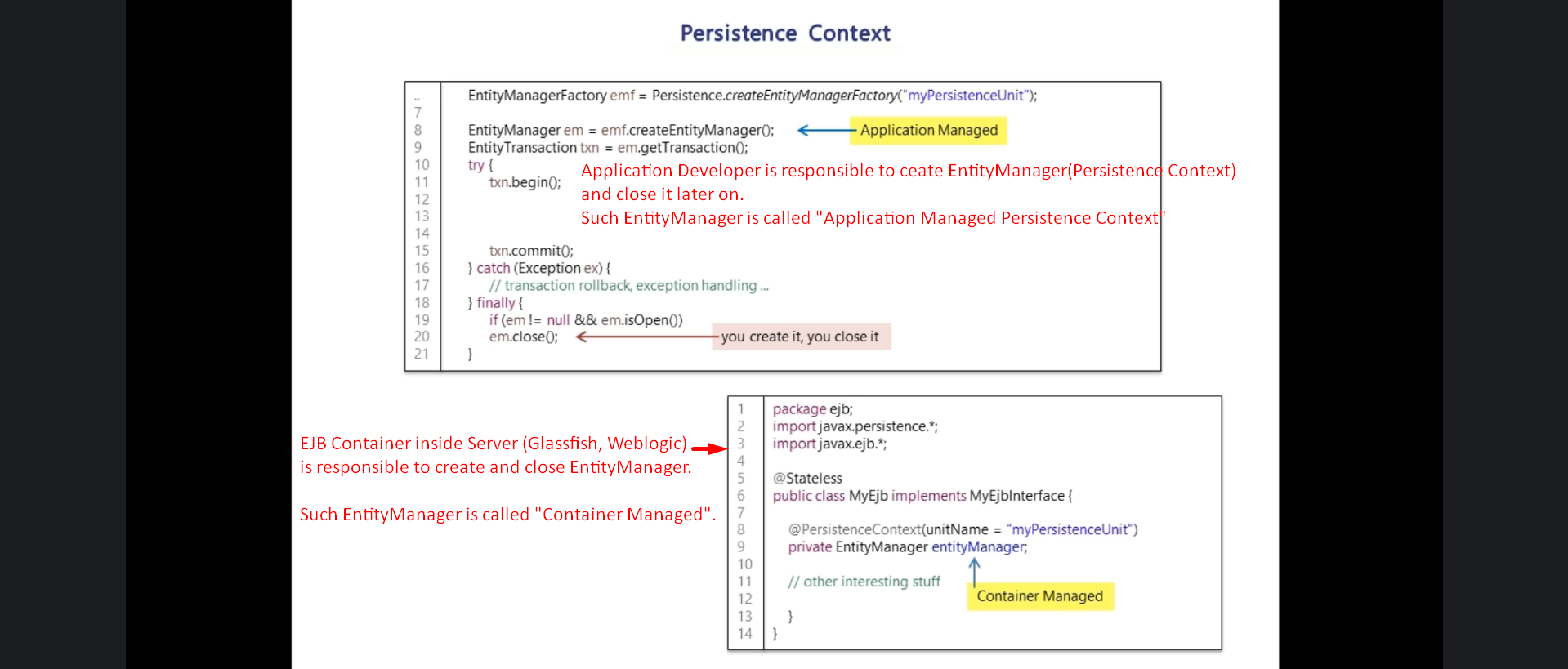
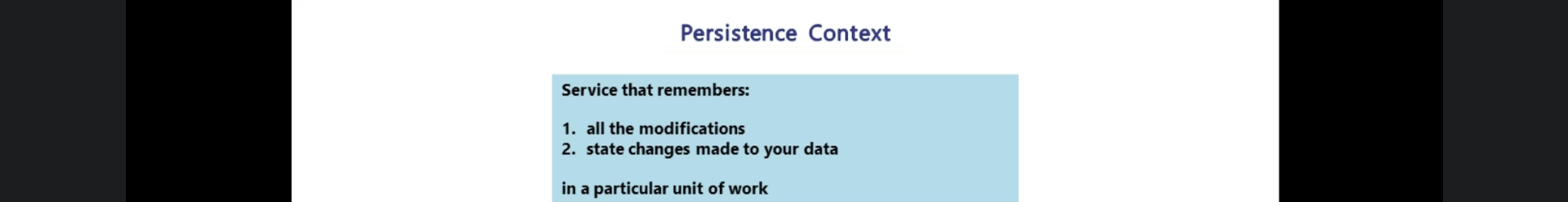
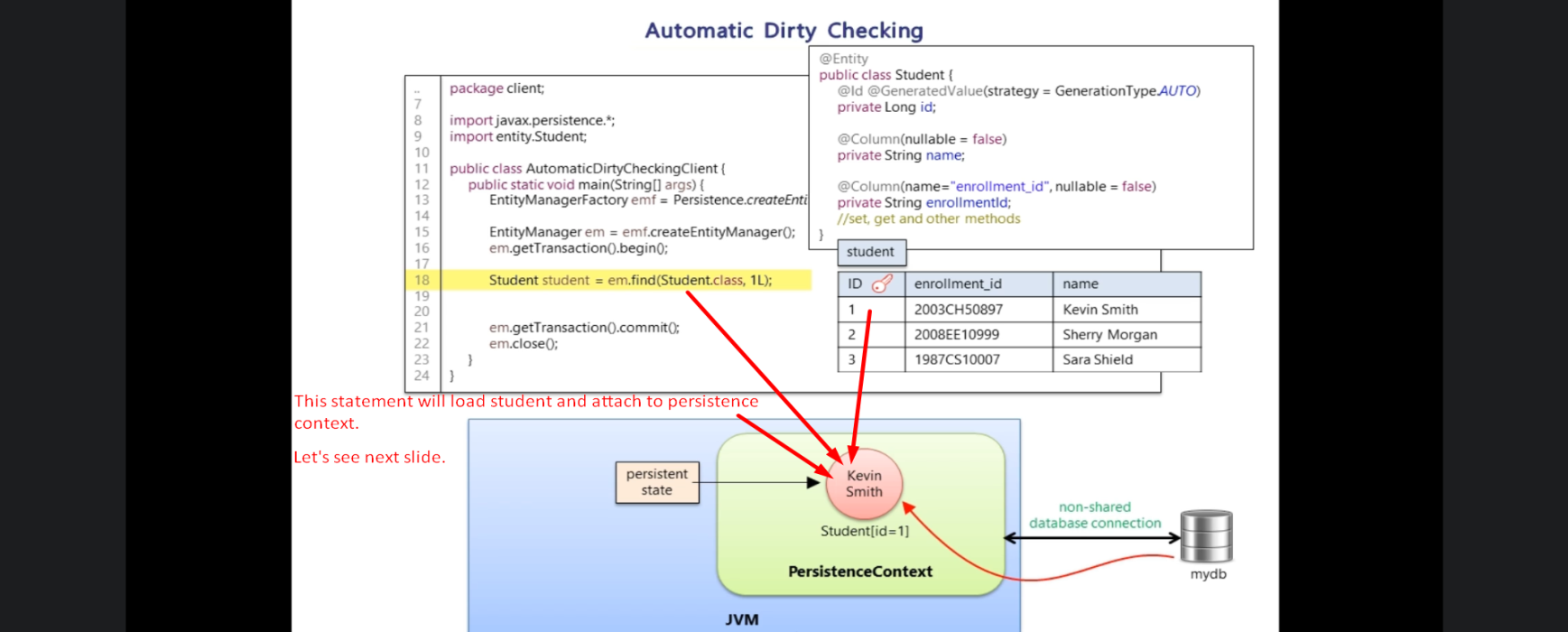
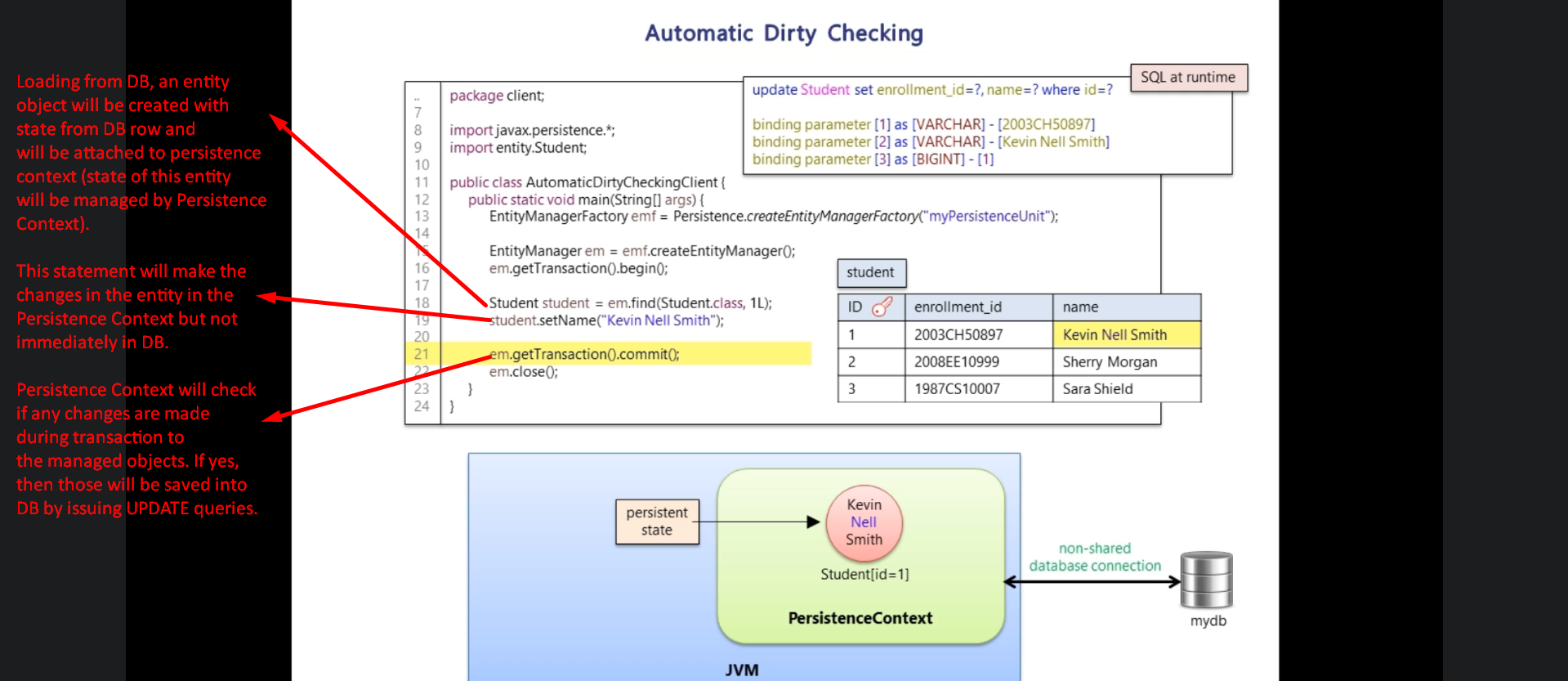
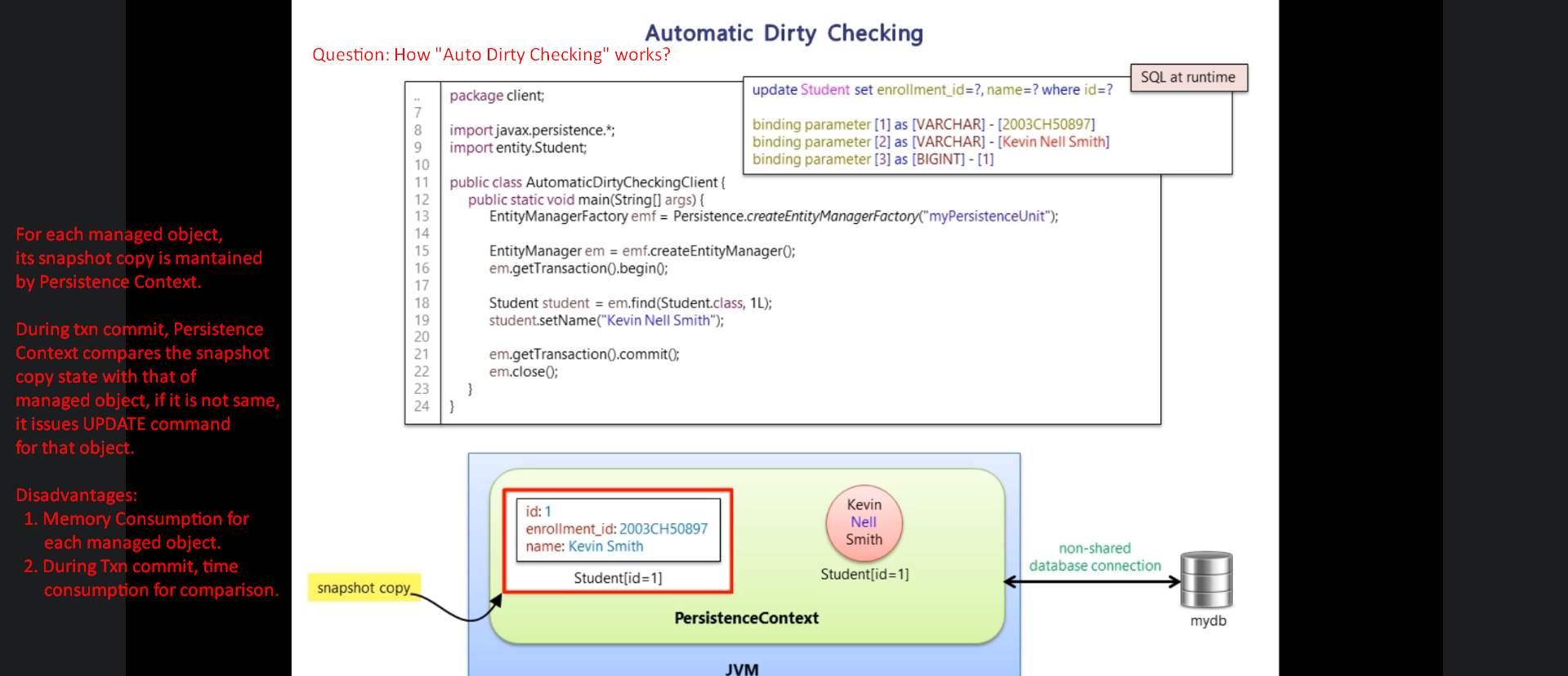
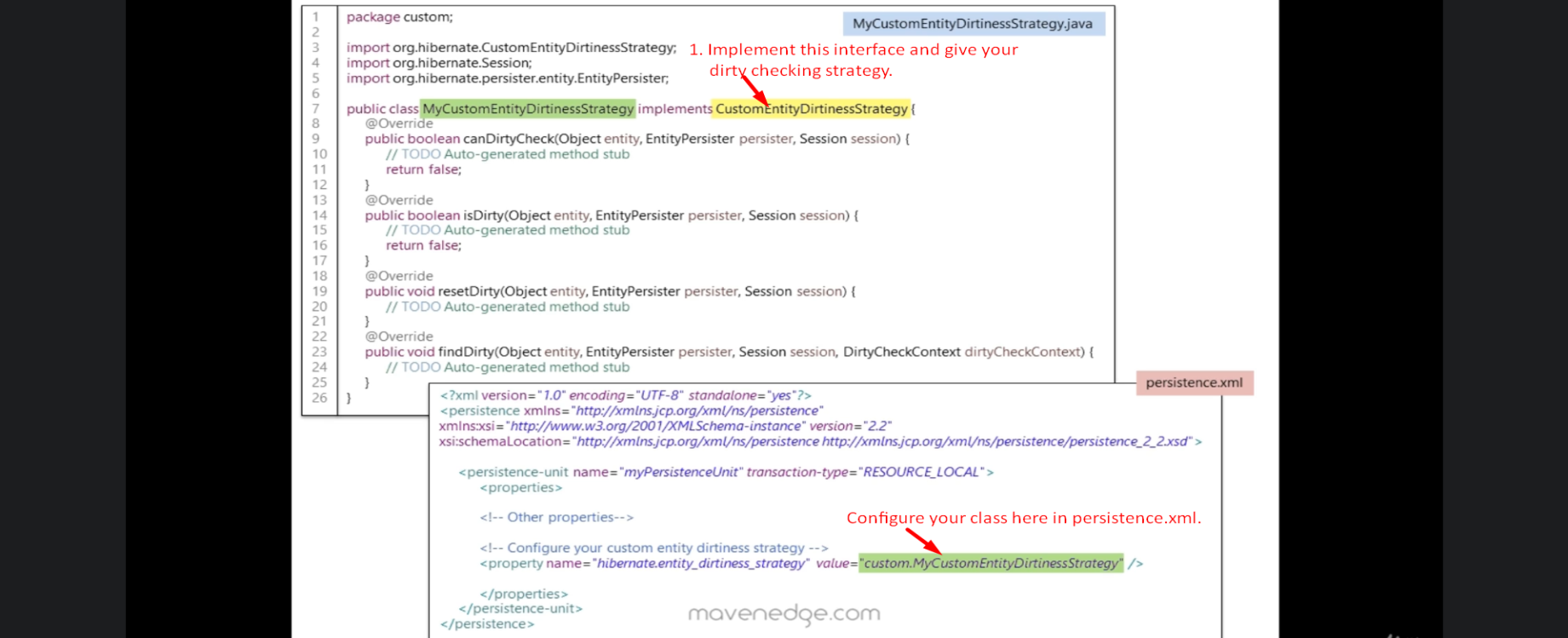
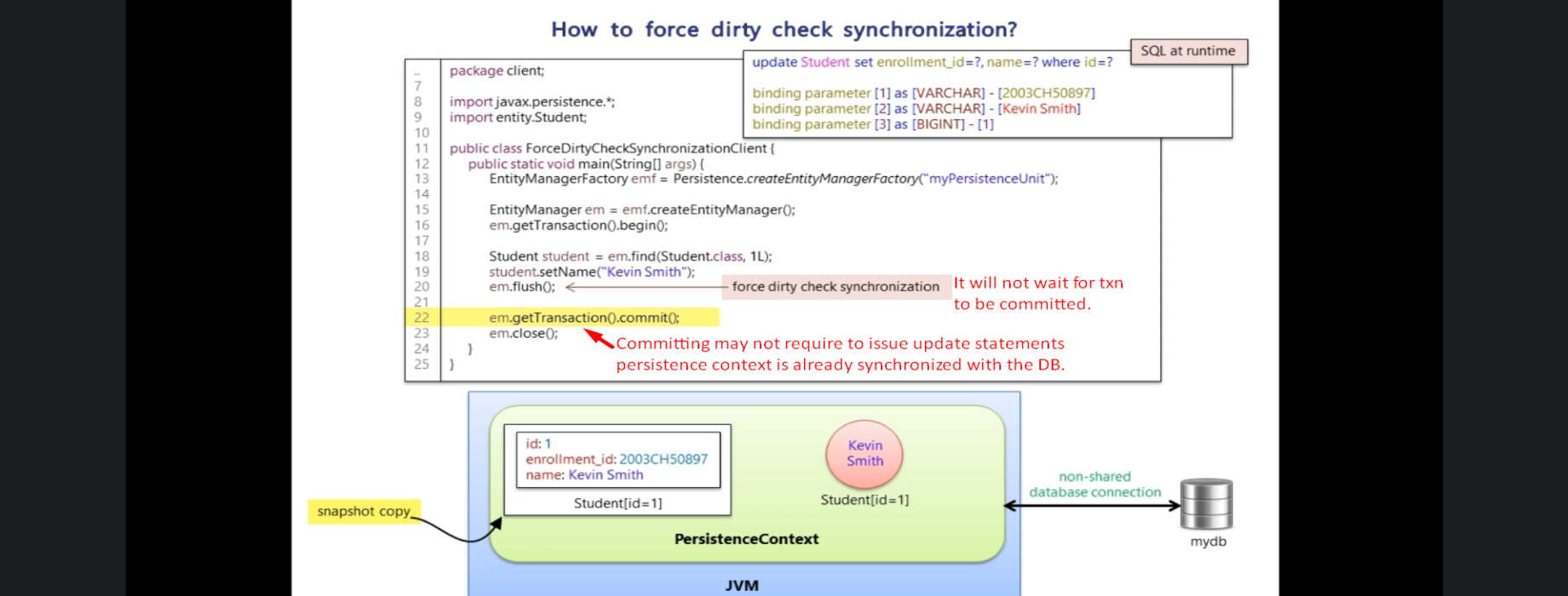
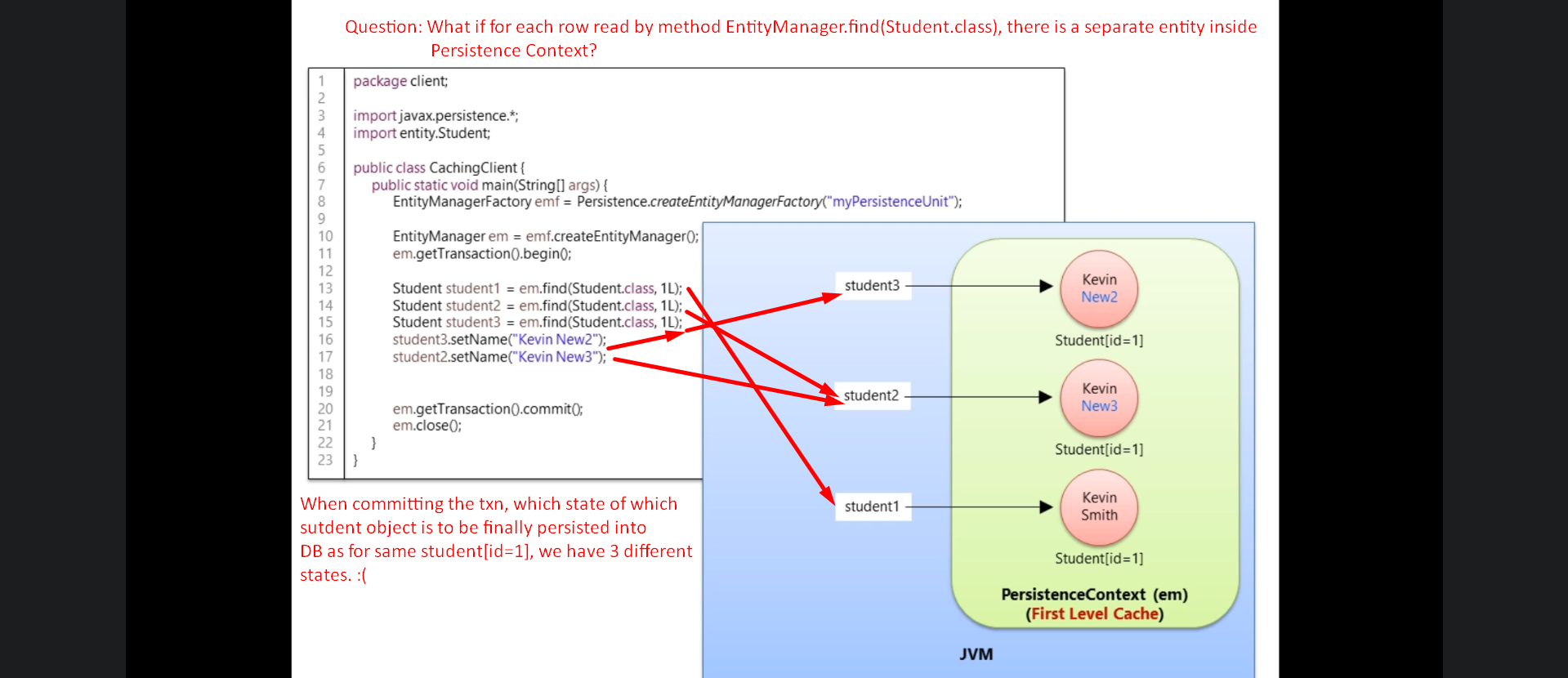
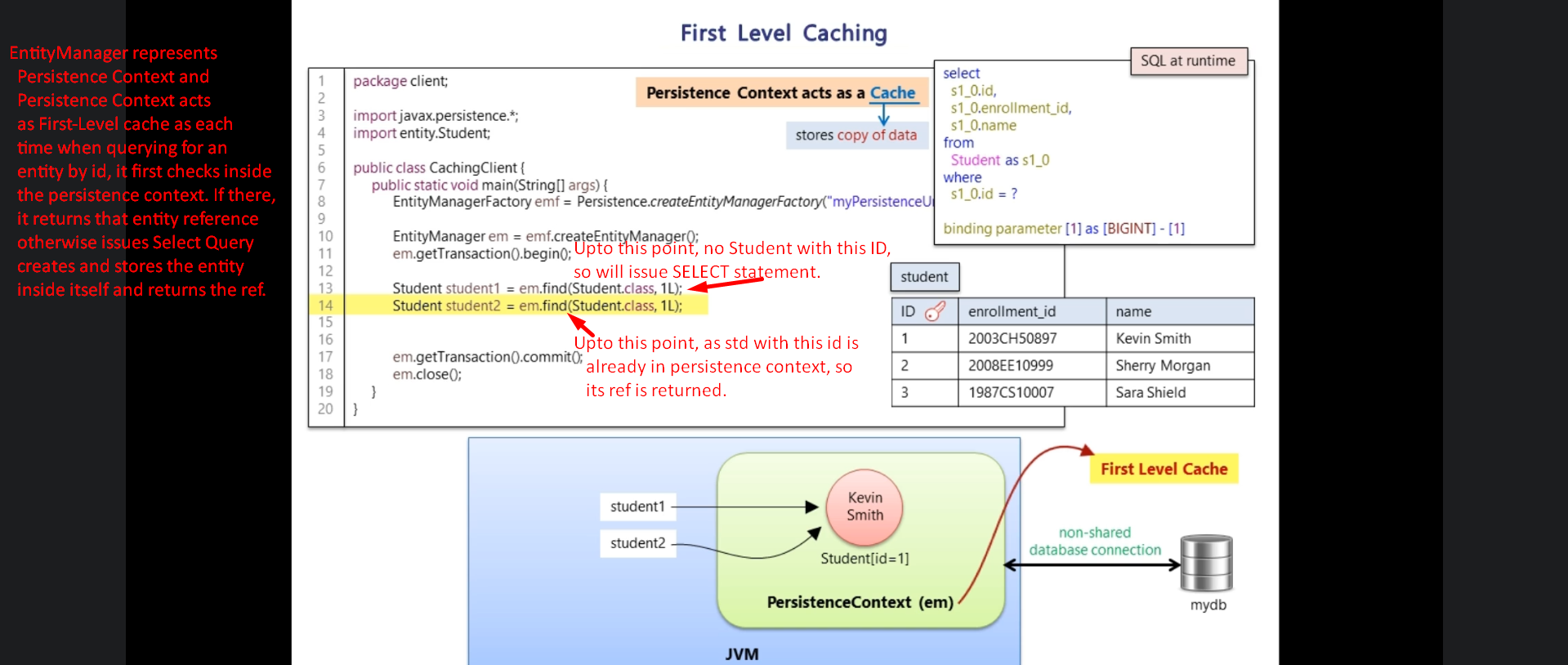
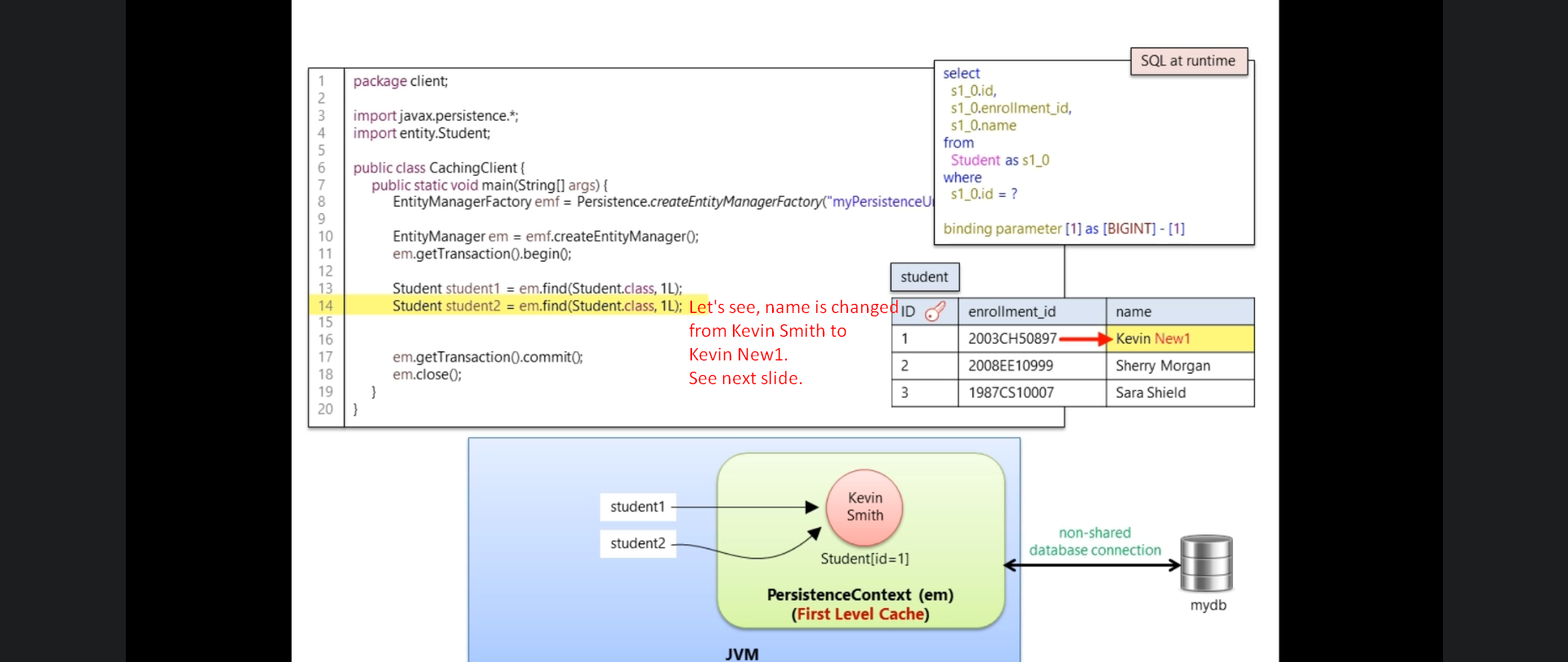
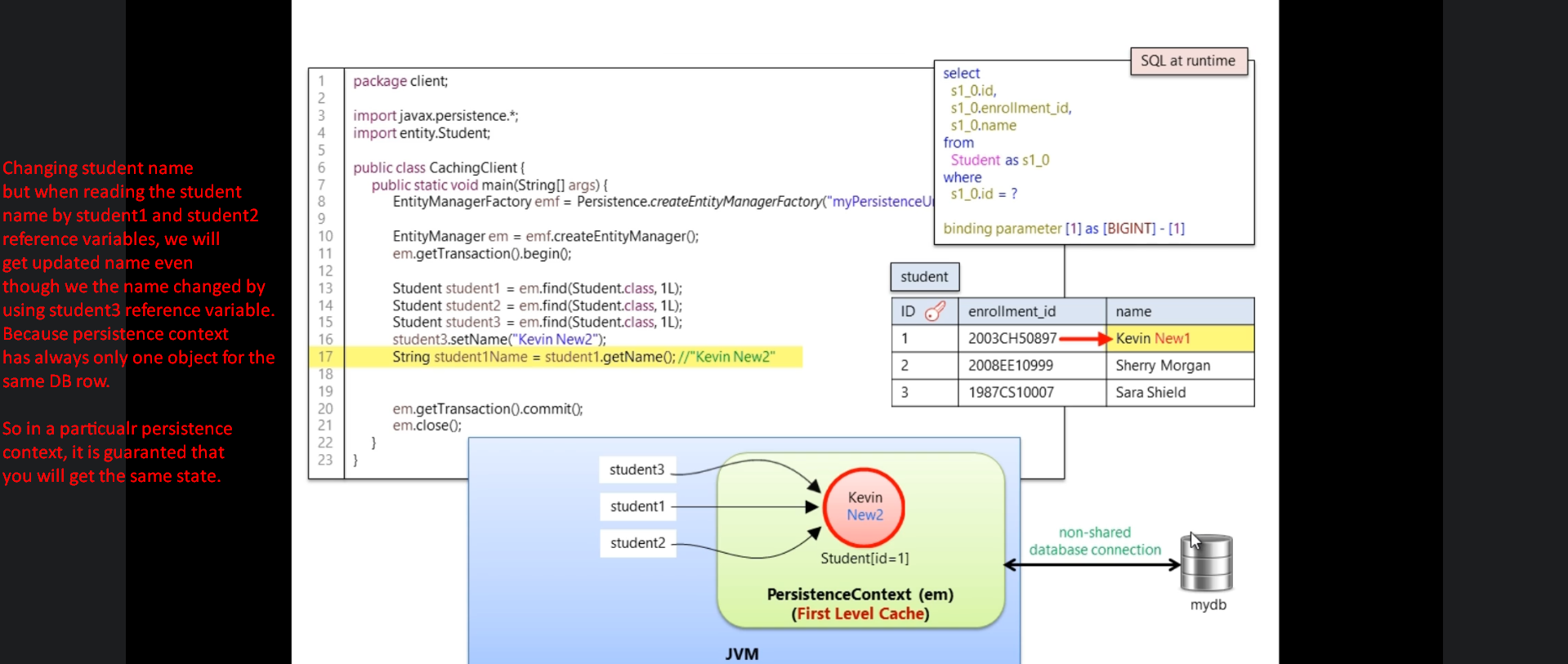
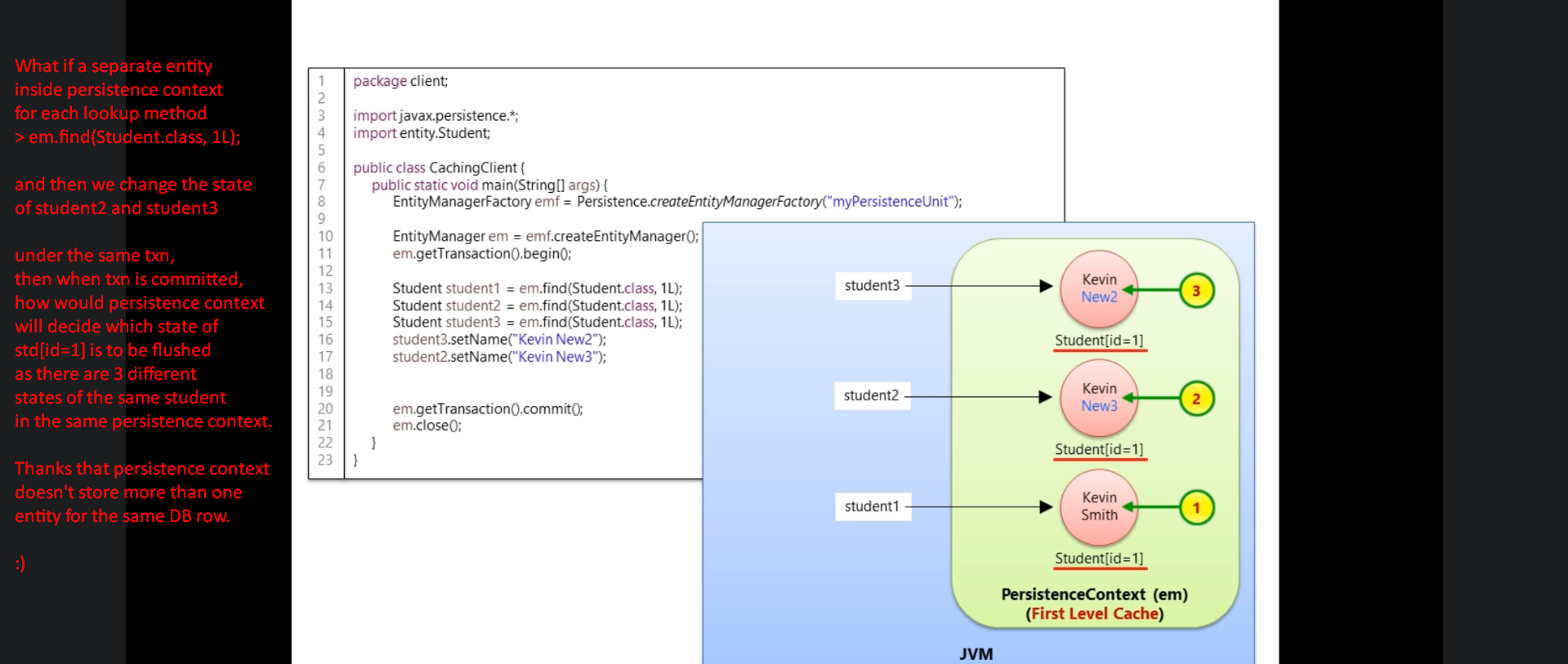
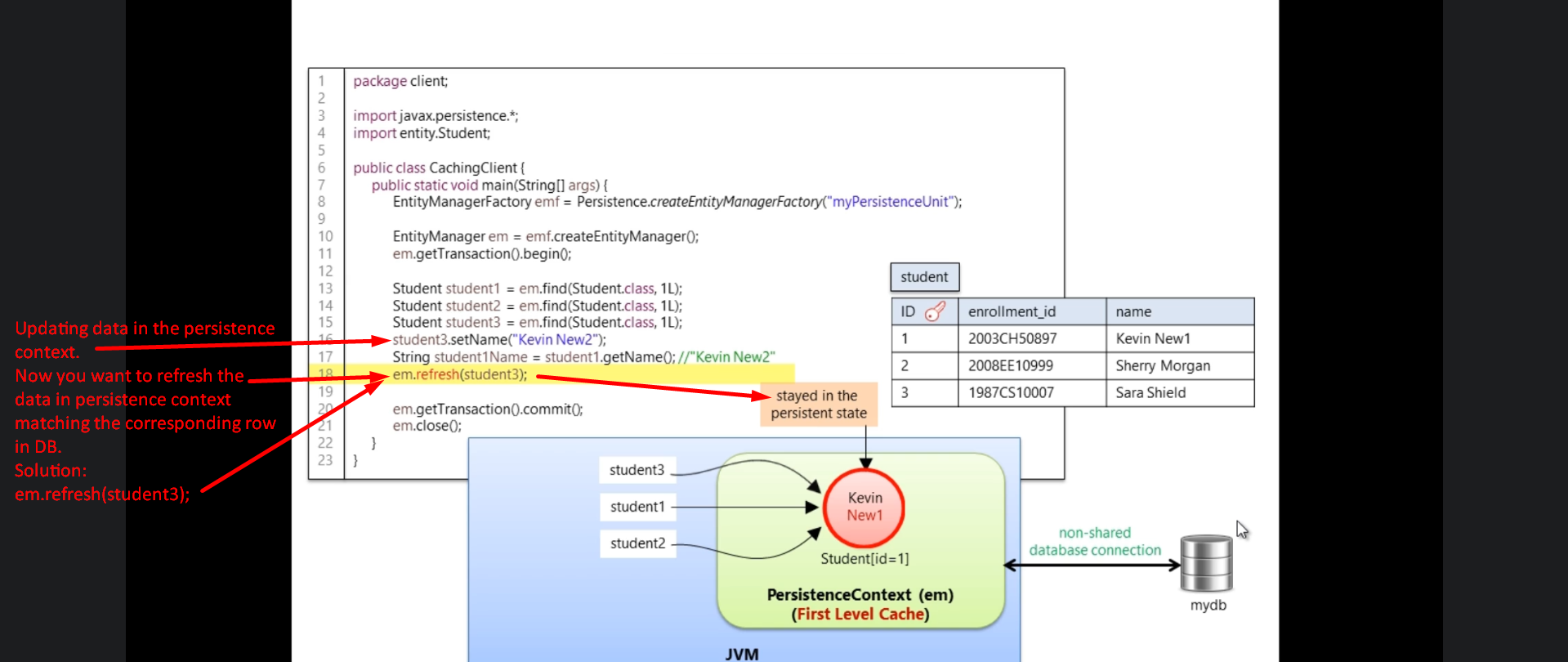
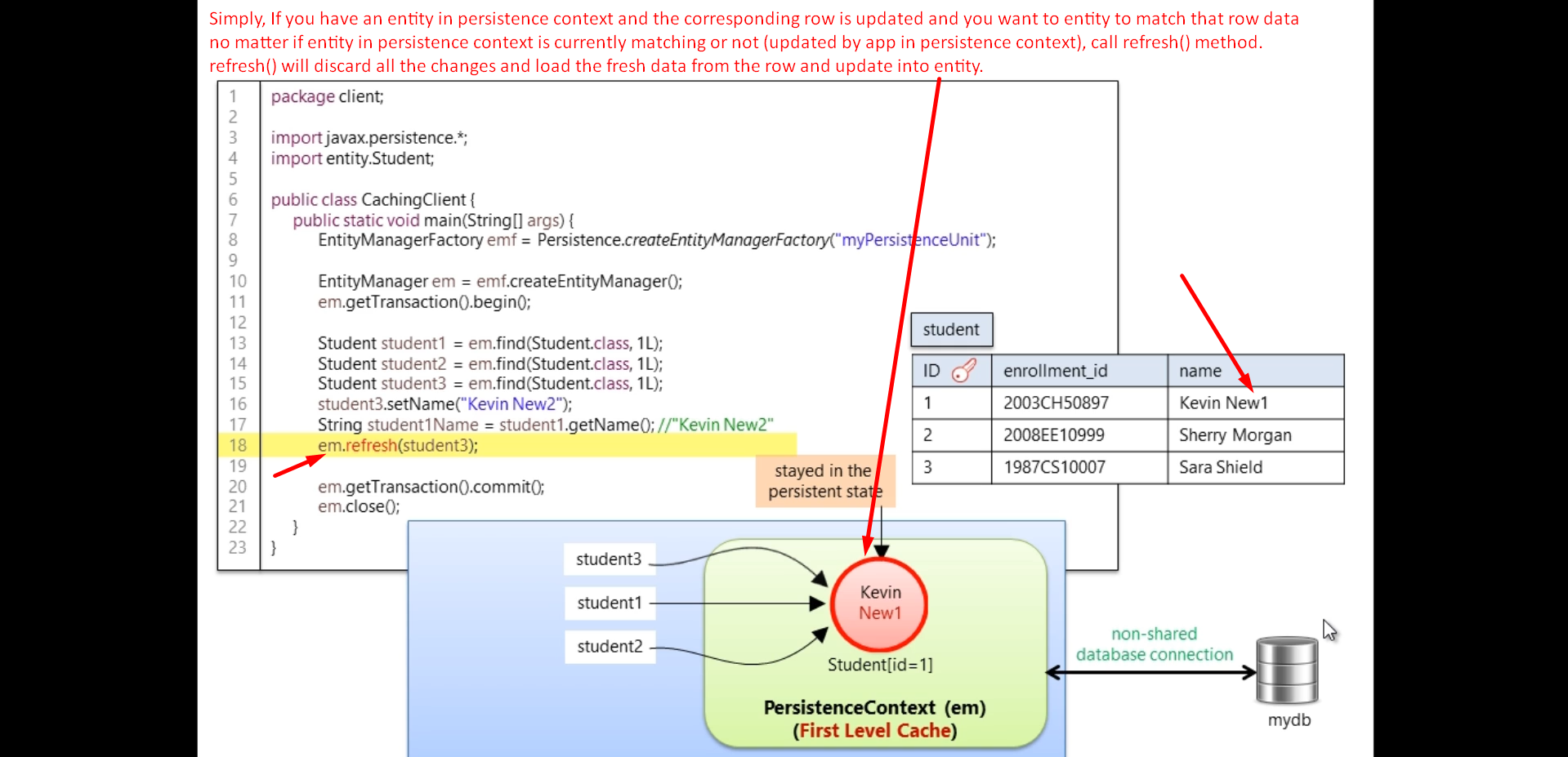
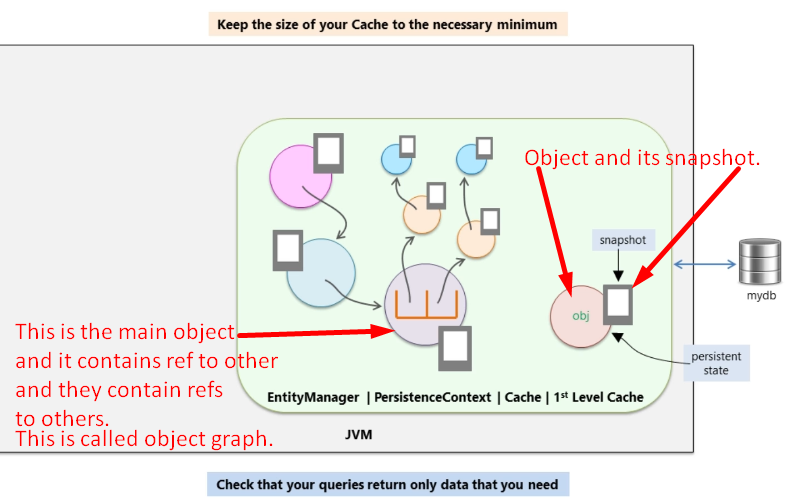
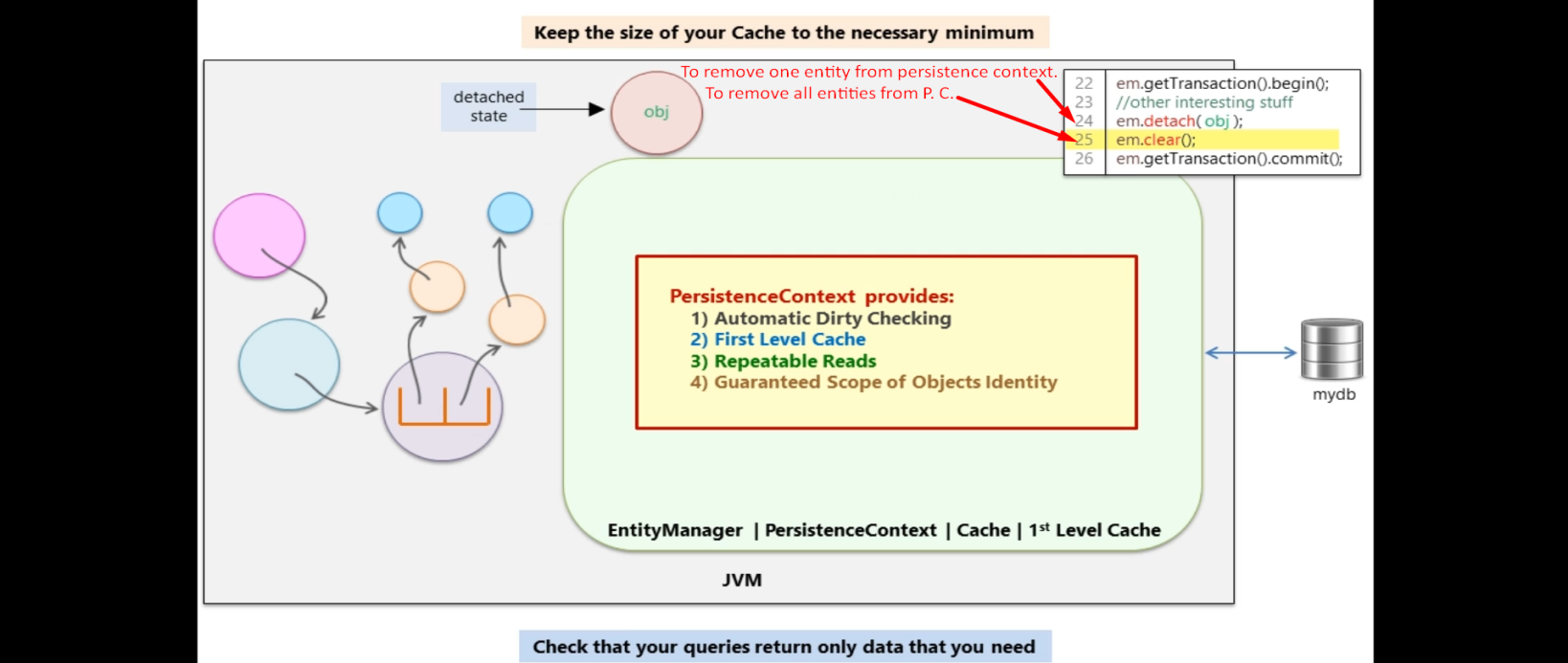
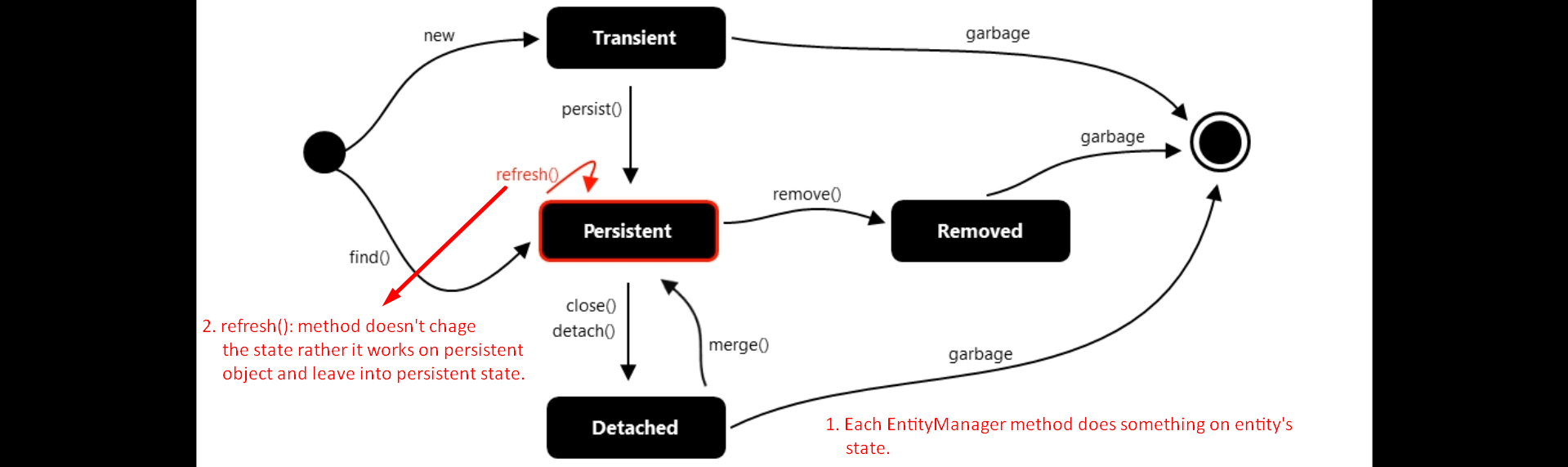
1. 
2. 
3. One of the things that Persistence Context does for us is **“Automatic Dirty Checking”**.
4. ****
5.   
   **NOTE**: Transaction commit will do “Auto Dirty Checking” means it keeps track changes made to managed entities and generates update statements for those entities whose states have been changed during the transaction.
6. 
7. Hibernate gives you extension point, where you can customize the way hibernate checks for “Dirty Check”.
8. CustomEntityDirtynessStrategy interface:
9. EntityManager.flush()
10. 
11. 
12. **First Level Cache**:
    1. Persistence Context acts as First-Level Cache.
    2. 
    3. 
    4. 
    5.   
       
13. refresh():
    1. 
14. Cache can improvement your app performance if you use it properly.
    1. Keep the size of your cache to the necessary minimum:
       1. It doesn’t shrink automatically. For each entity, it creates a snapshot and thus takes extra memory for each entity.  
          What if you have a very large object graph and you’re not using it?
    2. 
    3. To evict unnecessary objects from Persistence Context.  
       
    4. 
    5. f