**Work through the following materials this week, making sure that you can do the given activities.**

1. **Sections 20.1-20.4.1 & 20.6 — Explain the following.**
   1. ***interleaving* vs. *parallelism*Interleaving:** process switch task when another process is waiting for something.  
      **Parallelism:** multiple processes at once.
   2. **the *lost update* and *dirty read* problems  
      Lost update:** This problem occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database items incorrect  
      **Dirty read:** This problem occurs when one transaction updates a database item and then the transaction fails for some reason. … called dirty data because it has been created by a transaction that has not completed and committed yet; hence, this problem is also known as the dirty read problem.
   3. ***transactions*, including *commits* and *rollbacks*Transactions:** is an executing program that forms a logical unit of database processing. A transaction includes one or more database access operations—these can include insertion, deletion, modification (update), or retrieval operations.  
      **Commits:** Transaction complete.  
      **Aborted:** Transaction Failed.
   4. **the *ACID* properties  
      Atomicity.** A transaction is an atomic unit of processing; it should either be performed in its entirety or not performed at all.  
      **Consistency preservation.** A transaction should be consistency preserving, meaning that if it is completely executed from beginning to end without interference from other transactions, it should take the database from one consistent state to another.  
      **Isolation.** A transaction should appear as though it is being executed in isolation from other transactions, even though many transactions are executing concurrently. That is, the execution of a transaction should not be interfered with by any other transactions executing concurrently.  
      **Durability or permanency.** The changes applied to the database by a committed transaction must persist in the database. These changes must not be lost because of any failure.
   5. ***Schedule:*** is an ordering of the operations of the transactions. Operations from different transactions can be interleaved in the schedule S.
2. **Chapter 21 introduction and Section 21.1 — Explain the following.**
   1. ***Lock:*** s a variable associated with a data item that describes the status of the item with respect to possible operations that can be applied to it. Generally, there is one lock for each data item in the database. Locks are used as a means of synchronizing the access by concurrent transactions to the database items.
   2. ***two-phase locking:*** if all locking operations (read\_lock, write\_lock) precede the first unlock operation in the transaction.
3. **Review Oracle’s** [**Overview of Transaction Processing in PL/SQL**](http://docs.oracle.com/cd/B19306_01/appdev.102/b14261/sqloperations.htm#i7105) **(only; you can skip the other sections on this page) — Compare and contrast the following.**
   1. **SAVEPOINT, COMMIT vs ROLLBACK [TO]  
      SavePoint:** Saves data for rollback **Commit:** At end. Commits changes.  
      **RollBack:** Rolls back changes to SavePoint
   2. **LOCK TABLE vs. FOR UPDATE  
      Lock Table:** explicity lock entire tables.  
      **For Update:** can lock rows when updating them.