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

1. **Read Chapter 14 (focus primarily on Sections 14.1-2, 14.3.1-3, 14.5-6) — It’s ok to submit rather fuzzy or even empty answers on the material specifically marked for Wednesday’s discussion.**
   1. **Explain the four informal guidelines discussed in Section 14.1.  
      1.** Design a relation schema so that it is easy to explain its meaning.  
      **2.**  Design the base relation schemas so that no insertion, deletion, or modification anomalies are present in the relations.  
      **3.** As far as possible, avoid placing attributes in a base relation whose values may frequently be NULL.  
      **4.** Design relation schemas so that they can be joined with equality conditions on attributes that are appropriately related (primary key, foreign key) pairs in a way that guarantees that no spurious tuples are generated.
   2. **Explain the terms *functional dependency* (for Monday, see Section 14.2) and *multi-valued dependency* (for Wednesday, see Section 14.6) by giving an example of each and showing how it satisfies the term’s formal definition.  
      Functional dependency** is a relationship that exists when one attribute uniquely determines another attribute. If R is a relation with attributes X and Y, a functional dependency between the attributes is represented as X->Y, which specifies Y is functionally dependent on X.  
      **multivalued dependency** is a full constraint between two sets of attributes in a relation.
   3. **Review the formal definitions of these terms: *superkey*; *key* (see Section 14.3.3).**A superkey of a relation schema R = {A1, A2, … , An} is a set of attributes S ⊆ R with the property that no two tuples t1 and t2 in any legal relation state r of R will have t1[S] = t2[S]. A key K is a superkey with the additional property that removal of any attribute from K will cause K not to be a superkey anymore.
   4. **Explain *BCNF* (for Monday, see Section 14.5) and *4NF* (for Wednesday, see Section 14.6) by giving an example relation and proving whether its schema conforms (or fails to conform) to the definitions of each normal form.**Proposed as a simpler form of 3NF, but was found to be stricter than 3NF.
   5. **Explain when one would one want to *normalize* a relational schema and when one might want to *de-normalize* one.**Normalize would get rid of redundancies. No good reason to de-normalize.
2. **Review Section 15.1.1 well enough to be able to define *Armstrong’s axioms* (IR1–3) and *closures*.**

IR1 (reflexive rule)2: If X ⊇ Y, then X →Y.

2The reflexive rule can also be stated as X → X; that is, any set of attributes functionally determines itself.

IR2 (augmentation rule)3: {X → Y} |=XZ → YZ.

3The augmentation rule can also be stated as X → Y |= XZ → Y; that is, augmenting the left-hand-side attributes of an FD produces another valid FD.

IR3 (transitive rule): {X → Y, Y → Z} |=X → Z