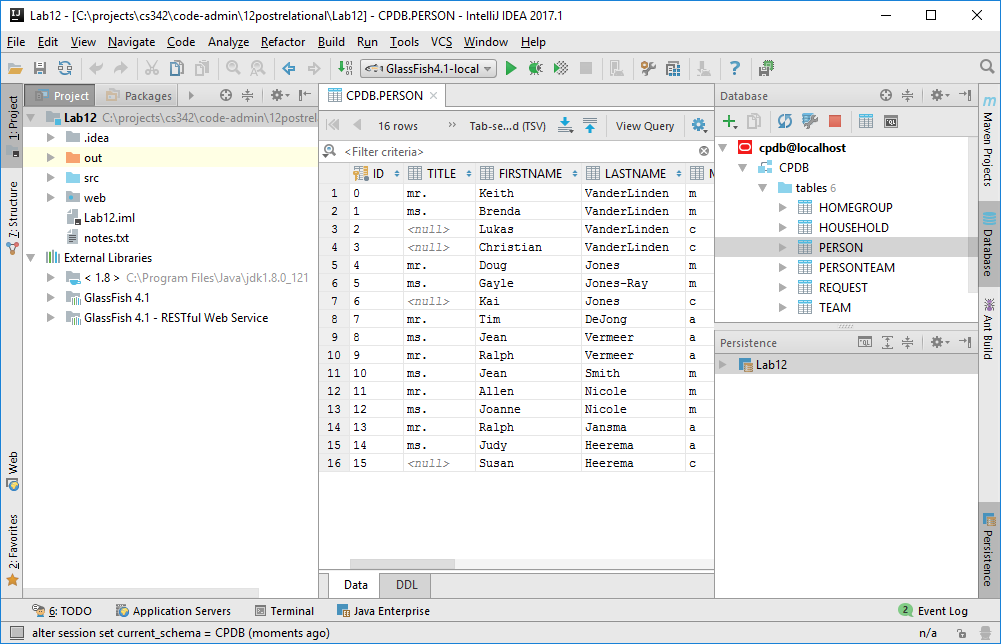
1. **Read specified sections of** [**Java Persistence**](https://en.wikibooks.org/wiki/Java_Persistence) **and explain the given concepts.**
   1. **Chapter 3, “What is Java persistence?”**
      1. ***Persistence* in Java and how it relates to the *impedance mismatch* discussed in unit 8**Managing accessing, persisting data between Java objects.
      2. ***POJO*s**Plain Old Java Object. Describe things as the “Normal Class”
   2. **Chapter 4, “Persistence Products, Which to Use?” — Name 2–3 of the major JPA implementations and indicate which one we are using in this class.**Open JPA, EclipseLink, TopLink
   3. **Chapter 5, “Mapping, Round Pegs into Square Holes” — Annotate this Java class for object-relational mapping using *Java Annotations* with the property *access type*.**@Entity

**public class Solder {** @transient  
 @ManyToMany **private String name;**

@transient  
 @ManyToOne **private String rank;** @id **private long serialNumber;  
  
 public String getName() { return name; }  
 public void setName(String name) { this.name = name; }  
  
 public String getRank() { return rank; }  
 public void setRank(String rank) { this.rank = rank; }  
  
 public String getSerialNumber() { return serialNumber; }  
 public void setSerialNumber(String serialNumber) { this.serialNumber = serialNumber; }  
}**

**In particular, add annotations for the following.**

* + 1. ***Persistence Entity* — Annotate the class as a persistence entity (Chapter 5 overview).**
    2. ***OID* — Annotate the object ID (Section 5.2).**
    3. ***Basic* fields — Annotate the non-ID fields (Section 5.6).**
    4. ***Relationships* — Add new fields/accessors/annotations that exemplify the following JPA relationships (Section 5.7 overview & Sections 5.7.2–5.7.4).**
       - **A *many-to-one* relationship to a commander — The commander is another Solder object.**
       - **A *many-to-many* relationship to a Battle class — You don’t need to add the Battle class.**
  1. **You don’t need to run this code, but be familiar with the concepts and the annotations that implement them.**
  2. **Chapter 6, “Runtime, Doing the Hokey Pokey (EntityManager)”**
     1. ***Entity Manager* (Chapter 6 overview)**The EntityManager provides API for creating queries, accessing transactions, and finding, persisting, merging and deleting objects. The JPA API can be used in any Java environment including JSE and JEE.
     2. ***Injection* — We’ll use J2EE dependency injection to access to a container-managed EntityManager (Chapter 6 overview).**An EntityManager can be created through an EntityManagerFactory, or can be injected into an instance variable in an EJB SessionBean, or can be looked up in JNDI in a JEE server.
     3. ***Persistence operations* — Name the four basic persistence operations (Section 6.2).**Create, read, update, and delete.
     4. 

1. **Review Intellij’s support to database connections and JPA.**
   1. **Create a new J2EE application as you created the JDK/J2EE application in the previous lab, activating the following “Java Enterprise” libraries/frameworks.**
      1. **Web Application 3.1**
      2. **J2EE Persistence 2.1; Provider: EclipseLink; setting up the library later**
      3. **RESTful Web Service; using the library from the GlassFish installation**
   2. **Store it in your repo under cs342/12postrelational/Lab12.**
   3. [**Databases and SQL**](https://www.jetbrains.com/help/idea/2016.3/databases-and-sql.html) **— Note that Oracle must be running with the CPDB database loaded for this data connection to work.**
      1. **Open the Intellij database tool pane by choosing *View→Tool Windows→Database*.**
      2. **Create a new data source for your existing Oracle CPDB database, name itcpdb&localhost and make sure that “Test Connection” works properly. You may need to download a required library (to make IDEA happy; GlassFish already has all the required libraries).**
      3. **Fetch the records in the Person table by double-clicking on the table name in the Database pane.**
   4. [**Java Persistence API (JPA)**](https://www.jetbrains.com/help/idea/2016.3/java-persistence-api-jpa.html)
      1. **Open the Intellij “Persistence” tool pane by choosing *View→Tool Windows→Persistence*.**