W2D5 Review Writing

W1D4: During W1D4, I learned about the introduction of Hibernate. Because we want to map our OO with RDB by following the domain driven design, we need ORM(JPA). To do mapping, there are a lot of things to do. The minor thing I learned first is that we need persistence.xml and we call that xml in our main app to do mapping. The most important thing I learned today is that we have to add annotations to our classes to map our OO with database tables. In that case, I learned many types of annotations for how to map our entities and how to map our ID and data types. (e.g. @Entity, @Id, @GeneratedValue, @Column, @Temporal, @Transient, @Lob).

W1D5: During W1D5, I learned about different states of entity lifecycle(transient, managed, detached and removed). And I studied the persistence API about the different methods for CRUD and methods for cache by changing the states of entities, given by entity manager or persistence context. e.g. em.persist() changes entity state from transient to managed. Then, I also learned about how Hibernate manages entities in cache for each method.

W1D6: During W1D6, I learned about associations among entities first. I studied 7 types of relationships with codes to know which entity can give which table in the database. As there are many important points on each of them(e.g. @OneToMany Unidirectional association default is JoinTable), I learned to take care of the associations. So, we know how database tables are created by Hibernate. As the W1D6 lecture also talked about collections, I learned 4 different types of collections(bag, set, list, map) and how Hibernate works with them.

W2D1: During W2D1, I learned about how to work inheritance with Hibernate(SINGLE\_TABLE, JOINED, TABLE\_PER\_CLASS). And I studied how to make complex mapping for existing table structure via secondary table, embedded class and composite keys. Then, I learned about JPQL, how to write a JPQL to get the data that we want if we don’t know entity ID(because if we know ID, we can easily use em.find() or em.getReference() methods). In the JPQL part, I’ve especially learned the details about query objects, from clause and where clause from this lecture.

W2D2: During W2D2, I learned more about queries, how to join entities in queries and about select clauses in detail. Because of joining entities in queries, I learned about optimization and about the common problems, like Cartesian product, N+1 problem and how to solve them in many ways (e.g. for N+1 problem, we can do our best using Entity Graph, Join Fetch, Batch Size and Sub Select). Another way I learned is to use 2nd level cache. But the important fact I will not forget is “Premature optimization is the root of all evil”, meaning that I don’t have to optimize until I have problems.

W2D3: During W2D3, I learned about the web container, by comparing the summary of Spring container and Hibernate container. As a view technology with a web container, I studied JSP, how to write JSP codes(xml) with JSTL and EL. Secondly, I learned about how to implement hibernate in our application, especially for entity management.We have to create an EMF(Entity Manager Factory) once. Then in the places where we use it, I learned the differences between EntityManager per operation(ani-pattern), Entity Manager per request and Open Entity Manager in View. Lastly, I learned about concurrency and advantages and disadvantages of different isolation levels and their problems and solutions. (e.g. in read-committed isolation level, optimistic concurrency can solve for lost update problems and pessimistic locking can solve Non-repeatable read problems.)

W2D4: During W2D4, I learned about the local transactions and global transactions. Then, I have clearly known the different types of transaction propagations and how to work them in Spring(e.g. Transaction Demarcation). And I studied how to configure Spring in a web container, how to configure Hibernate with Spring, how transaction managers configure in Spring and configuration with OpenEntityManagerInView in Spring.

Science of Consciousness: Based on the above lectures, to work Hibernate with Spring, it has to be configured first in Spring. To build a web application with Hibernate, there are many layers, View, Control, Service, Domain and Persistence. We have to add JPA annotations on the entity classes in the Domain layer. Based on the annotations from the domain layer, Hibernate creates database tables and manages objects, based on the entity associations in the Persistence layer. Therefore, “Life is found in layers”, according to SCI principles.