HW#4

1 - What is JPA?

JPA is an API and it stands for “Java Persistence API”. It is a Java specification that gives some functionality and standard to ORM tools. It defines the management of relational data in a Java application. JPA acts as a bridge between object-oriented domain models and relational database systems.

JPA is just a set of interfaces and it requires an implementation to be used because it is not a stand-alone product. Popular implementations include Hibernate, EclipseLink, OpenJPA and some others. These implementations improve the functionality of JPA with some advanced features.

2 - What is the naming convention for finder methods in the Spring data repository interface?

Finder methods are the methods which starts with “*findBy…*” word. The complete names of these methods depend on the coder. It can finish with a property which is present in a class. For example, “findById” and “findByName” are the most common finder methods. They are like first finder methods which is written by coders.

The exact name of the finder methods should be written according to the naming convention of Spring. According to this naming convention, first letter of finder method, which is “f”, must be lowercase. After the first word, every first letter of every word must be uppercase. And the other letters of the words must be lowercase. By the way, between the words there must be no whitespace characters or any underscore.

3 - What is PagingAndSortingRepository?

Pagination is a technique for splitting a list of multiple records into sublists. Sorting is the processing of arranging the data in ascending and descending order.

PagingAndSortingRepository is an extension of CrudRepository to provide additional methods to retrieve entities using the sorting abstraction. With this extension we can add a special Sort parameter to our query method. It provides two methods. The first method is “*findAll(Sort sort)*” method. “*findAll(Sort Sort)*” method returns a Iterable of entities which meets the sorting condition provided by Sort object. No paging is applied here. If we want to do both sorting and paging the data we can use the second method, “*Page findAll(Pageable pageable)*”. So when we want to make paging and sorting in the results, we just add “*Pageable*” to the definition of the method as a parameter.

4 - Difference between findById() and getOne()?

The "*findById"* method is available in *CrudRepository* interface while "*getOne"* method is available in *JpaRepository* interface. "*findById*" method accesses the database and return to the entity object associated with the given ID. It is EAGER loaded operation that returns null if no record exists in database. "*getOne"* method returns a reference to the entity with the given identifier. It returns a proxy without hitting the database (lazily fetched). This method throws *EntityNotFoundException* at the time of actual access if the requested entity does not exist in the database.

5 - What is @Query used for?

When we want to write a complex query with multiple conditions to filter the data we use *@Query* annotation. *@Query* annotation gives us flexibility over the executed statement and our method name doesn't need to follow any conventions. The only thing we need to do is to define a method in our repository interface, annotate it with *@Query* and provide the statement that we want to execute. *@Query* annotation supports both JPQL and native SQL queries.

6 - What is lazy loading in hibernate?

Lazy loading is the practice of delaying load or initialization of all resources or all objects until they're actually needed. The main purpose of lazy loading is to fetch the needed objects from the database. It decides whether to load a child class object while loading the parent class object. Hibernate can use lazy loading, which means it will load only the required classes, not all classes. It prevents a huge load since the entity is loaded only once when necessary. Lazy loading improves performance by avoiding unnecessary computation, reduces memory requirements and save system resources.

7 - What is SQL injection attack? Is Hibernate open to SQL injection attack?

SQL injection is one of the most common web hacking techniques. SQL injection attack consists of injection of malicious SQL code for backend database manipulation to access information that was not intended to be displayed. This information may include any number of items, including sensitive data or private details.

All systems using SQL database may be a victim of SQL injection attack. Therefore Hibernate does not grant immunity to SQL Injection. There are some methods to improve security against SQL injection attacks but a perfect protection does not exist.

8 - What is criteria API in hibernate?

Criteria is a simplified API for retrieving entities by composing *Criterion* objects. It is a predefined API and an alternative way of defining a JPQL query. These queries are type-safe, and portable and easy to modify by changing the syntax. Criteria API lets us build nested, structured query expressions in Java, providing a compile-time syntax checking that is not possible with a query language like HQL or SQL. It includes query by example (QBE) functionality. The major advantage of the criteria API is that errors can be detected earlier during compile time.

9 - What is Erlang? Why is it required for Rabbitmq?

Erlang is a functional programming language. It is used to build massively scalable soft real-time systems with requirements on high availability. It was designed with the aim of improving the development of telephony applications.

RabbitMQ is an AMQP messaging protocol implementation. It is written in Erlang language. RabbitMQ server is built on the *Open Telecom Platform* framework which is an Erlang framework. That is why we need Erlang for to use RabbitMQ because it uses Erlang frameworks.

10 - What is the JPQL?

The JPQL (Java Persistence Query Language) is a powerful object-oriented query language which allows us to define database queries based on our entity model. Its structure and syntax are very similar to SQL. The main difference between SQL and JPQL is that SQL works with relational database tables, records and fields, whereas JPQL works with Java classes and objects. Instead of database table, JPQL uses entity object model to operate the SQL queries. At this point the role of JPA is to transform JPQL into SQL.

The most important features of JPQL are; it is a platform-independent, simple, powerful query language and it can be used with any type of database such as MySQL, Oracle.

11 - What are the steps to persist an entity object?

Steps to persist an entity object are:

- Adding the "*hibernate-entitymanager"* dependency in the *pom.xml* file
- Adding the JDBC driver as another dependency in the *pom.xml* file
- Adding JPA and database config in *"src/main/resources/META-INF/persistence.xml"* file
- Creating "*javax.persistence.EntityManagerFactory"* from *"javax.persistence.Persistence"*
- Mapping the Entity class with *@Entity* annotation that we want to persist
- Using EntityManager to perform CRUD on Entity
- Testing the code about functionalities

12 - What are the different types of entity mapping?

A mapping constraint is a data constraint that expresses the number of entities to which another entity can be related via a relationship set. There are four possible mapping cardinalities:

- One-to-one Mapping
- One-to-many Mapping
- Many-to-one Mapping
- Many-to-many Mapping

13 - What are the properties of an entity?

Properties are the variables in a class. They are the adjectives describing an entity. Methods are not properties. Properties defines some features of the entity they belong. We may use these properties or we may change them.

What the properties of an entity are depends on the entity and developer. We can create different properties for an entity. But generally "*id*" and "*name*" properties are very common almost for every entity.

14 - Difference between CrudRepository and JpaRepository in Spring Data JPA?

CrudRepository and *JPA repository* both are the interfaces of the spring data repository library. *CrudRepository* is the base interface and extends the *Repository* interface. It provides methods for CRUD operations which are *create, read, update* and *delete*.

JPA repository extends *crudRepository* and *PagingAndSorting repository*. So *JPA repository* contains all methods of these interfaces. Besides them *JPA repository* contains more additional methods which is specific to JPA.