HOMEWORK 4

Q1) What is JPA ?

The Java Persistence API (JPA) is the Java standard for mapping Java objects to a relational database.

JPA is one possible approach to ORM. Via JPA, the developer can map, store, update, and retrieve data from relational databases to Java objects.

Q2) What is the naming convention for finder methods in the Spring data repository interface ?

Spring data JPA has its own naming conventions for methods. Following these conventions we can build sophisticated queries. These conventions are called also called as method name strategies. These strategies have defined set of keyword to use in method names. Based on the formed method name, method performs predefined operations.

Q3) What is PagingAndSortingRepository ?

PagingAndSortingRepository is an extension of CrudRepository to provide additional methods to retrieve entities using the pagination and sorting abstraction. It provides two methods :

- Page findAll(Pageable pageable) – returns a Page of entities meeting the paging restriction provided in the Pageable object.
- Iterable findAll(Sort sort) – returns all entities sorted by the given options. No paging is applied here.

Q4) Differentiate between findById() and getOne() ?

findById()	**getOne()**
The findById() method is available in CrudRepository interface.	The getOne() method is available in JpaRepositpry interface.
The findById() method will return null if the record doesn't exist in the database.	The getOne() method throw EntityNotFoundException if the record doesn't exist in the database.
Internally findById() method use EntityManger find() method.	Internally getOne() method use EntityManger getReference() method.
Calling findById() returns an eagerly fetched entity.	Calling getOne() returns a lazily fetched entity.

Q5) What is @Query used for ?

The @Query annotation declares finder queries directly on repository methods. While similar @NamedQuery is used on domain classes, Spring Data JPA @Query annotation is used on Repository interface. This frees the domain classes from persistence specific information, which is a good thing.

Q6) What is lazy loading in hibernate ?

Lazy loading is a fetching technique used for all the entities in Hibernate. It decides whether to load a child class object while loading the parent class object. When we use association mapping in Hibernate, it is required to define the fetching technique. The main purpose of lazy loading is to fetch the needed objects from the database.

For example, we have a parent class, and that parent has a collection of child classes. Now, 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 and reduce memory requirements.

Lazy loading can be used with all types of Hibernate mapping, i.e., one-to-one, one-to-many, many-to-one, and many-to-many.

Q7) What is SQL injection attack ? Is Hibernate open to SQL injection attack ?

SQL Injection: Injecting the value to the SQL statement. SQL injection refers to the act of someone inserting a MySQL statement to be run on your database without your knowledge. Injection usually occurs when you ask a user for input, like their name, and instead of a name they give you a MySQL statement that you will unknowingly run on your database.

Normal: ”SELECT * FROM student WHERE studentName= ‘sweety'”
Injection: “SELECT * FROM student WHERE studentName= ” +studentName

It is a very common misconception that ORM solutions, like hibernate, are SQL Injection proof. Hibernate allows the use of “native SQL” and defines a proprietary query language, named, HQL the former is prone to SQL Injection and the later is prone to HQL injection.

Q8) What is criteria API in hibernate?

If you want to define your query dynamically at runtime, you can use JPA's Criteria API. In the past, Hibernate also offered its own proprietary Criteria API. It has been deprecated in Hibernate 5, and you should avoid it when implementing new use cases.

Hibernate's Criteria query is a little easier to define and execute. As you will see in the migration chapter, it uses a straightforward approach to create the different parts of the query and execute it. JPA's Criteria API, on the other hand, makes heavy use of the CriteriaBuilder interface to create the different parts of your query. It's verbosity often makes it a little hard to read.

Q9) What Is Erlang? Why Is It Required For Rabbitmq ?

Erlang is a functional, general-purpose language oriented towards building scalable, concurrent systems with high availability guarantees.

It was built at the end of the 1980s at Ericsson for handling telephone switches. At the time, telephone switching systems were one of the most complicated systems out there, like the internet is nowadays. For this reason, the language used to program them needed to support high concurrency and zero downtime.

Q10) What is the JPQL ?

JPQL is Java Persistence Query Language defined in JPA specification. It is used to create queries against entities to store in a relational database. JPQL is developed based on SQL syntax. But it won't affect the database directly.

JPQL can retrieve information or data using SELECT clause, can do bulk updates using UPDATE clause and DELETE clause. EntityManager.createQuery() API will support for querying language.

Q11) What are the steps to persist an entity object ?

1. Add the hibernate dependency: Add the hibernate-entitymanager dependency in the pom.xml

2. Add the JDBC driver: I will make use of in-memory h2 database for this course for simplicity.

3. Add JPA and database config in META-INF/persistence.xml: Now, you need to specify the database and ORM specific configuration in persistence.xml file.

4. Create EntityManagerFactory: Next step is to create javax.persistence.EntityManagerFactory from javax.persistence.Persistence.

5. Mapping the Entity class: Annotate the class with @Entity that you want to persist as shown below.

6. Use EntityManager to perform CRUD on Entity: Now, let us design the Dao. As I mentioned before, ideally JPA code should only be present in the DAO layer. Not recommended to use in the higher layers.

Q12) What are the different types of entity mapping?

One-to-One: in DBMS is a relationship between an instance of an entity with another. Example: like Employee-Department, Student-Course, etc.

One-to-Many: in DBMS is a relationship between instances of an entity with more than one instance of another entity.

Many-to-One: in DBMS is a relationship between more than one instances of an entity with one instance of another entity. Example: Student and Project

Many-to-Many: in DBMS is a relationship between more than one instance of an entity with more than one instance of another entity i.e. both the entities can have many relationships between each other.

Q13) What are the properties of an entity?

An entity can be place, person, object, event or a concept, which stores data in the database. The characteristics of entities are must have an attribute, and a unique key. Every entity is made up of some 'attributes' which represent that entity.

Q14) Difference between CrudRepository and JpaRepository in Spring Data JPA?

CrudRepository: It is a base interface and extends Repository Interface. It contains methods for CRUD operations. For example save(), saveAll(), findById(), findAll(), etc. It doesn't provide methods for implementing pagination and sorting. It works as a marker interface. To perform CRUD operations, define repository extending CrudRepository.

JpaRepository: It extends PagingAndSortingRepository that extends CrudRepository. It contains the full API of CrudRepository and PagingAndSortingRepository. For example, it contains flush(), saveAndFlush(), saveAllAndFlush(), deleteInBatch(), etc along with the methods that are available in CrudRepository. It provides all the methods for which are useful for implementing pagination. It extends both CrudRepository and PagingAndSortingRepository. To perform CRUD as well as batch operations, define repository extends JpaRepository.