HOMEWORK 4

RÜMEYSA TÜRKER

1 – What is JPA?

The Java Persistence API (JPA) is a specification of Java. It is used to persist data between Java object and relational database. JPA acts as a bridge between object-oriented domain models and relational database systems.

Java Persistence API is a collection of classes and methods to persistently store the vast amounts of data into a database which is provided by the Oracle Corporation.

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

Finder methods are the most powerful methods, we can create finder methods to select the records from the database without writing SQL queries. Behind the scenes, Data JPA will create SQL queries based on the finder method and execute the query for us.

To create finder methods in Data JPA, we need to follow a certain naming convention. To create finder methods for the entity class field name, we need to create a method starting with findBy followed by field name.

3 - 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.

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

getOne()	**findById()**
Lazily loaded reference to target entity	Actually loads the entity for the given id
Useful only when access to properties of object is not required	Object is eagerly loaded so all attributes can be accessed
Throws EntityNotFoundException if actual object does not exist at the time of access invocation	Returns null if actual object corresponding to given Id does not exist
Better performance	An additional round-trip to database is required

5 - 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.

6 - 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.

Lazy loading in hibernate improves the performance. It loads the child objects on demand. Since Hibernate 3, lazy loading is enabled by default, you don't need to do lazy="true". It means not to load the child objects when parent is loaded.

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

SQL Injection (SQLi) is a type of an injection attack that makes it possible to execute malicious SQL statements. These statements control a database server behind a web application. Attackers can use SQL Injection vulnerabilities to bypass application security measures. They can go around authentication and authorization of a web page or web application and retrieve the content of the entire SQL database. They can also use SQL Injection to add, modify, and delete records in the database.

Hibernate does not grant immunity to SQL Injection, one can misuse the api as they please.

There is nothing special about HQL (Hibernates subset of SQL) that makes it any more or less susceptible.

Functions such as createQuery(String query) and createSQLQuery(String query) create a Query object that will be executed when the call to commit() is made. If the query string is tainted you have sql injection. The details of these functions are covered later.

8 - What is criteria API in hibernate?

The criteria query API lets you 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.

The Criteria API also includes query by example (QBE) functionality. This lets you supply example objects that contain the properties you would like to retrieve instead of having to step-by-step spell out the components of the query. It also includes projection and aggregation methods, including count(). Let’s explore it’s different features in detail.

9 - What Is Erlang? Why Is It Required For Rabbitmq?

Erlang is a programming language used to build massively scalable soft real-time systems with requirements on high availability.

If we need robustness in the form of crash resilience, knowing whether or not a job was started or completed, being able to let a back-end process go down and not worry too much about losing work, etc. then we need rabbitmq.

10 – 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.

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

1. Creating an entity manager factory object. The EntityManagerFactory interface present in java.
2. Obtaining an entity manager from factory.
3. Intializing an entity manager.
4. Persisting a data into relational database.
5. Closing the transaction.
6. Releasing the factory resources.

12 – What are the different types of entity mapping?

- **one-to-one:** A one-to-one relationship is the simplest relationship between two beans. One entity bean relates only to one other entity bean.
- **one-to-many or many-to-one (dependent on the direction):** In a one-to-many relationship, one object can reference several instances of another. A many-to-one relationship is when many objects reference a single object.
- **many-to-many:** many objects can reference many objects. This cardinality is the most difficult to manage.

13 - What are the properties of an entity?

An entity is a lightweight persistence domain object. Typically, an entity represents a table in a relational database, and each entity instance corresponds to a row in that table. The primary programming artifact of an entity is the entity class, although entities can use helper classes.

Entities are represented by means of their properties, called attributes. All attributes have values. For example, a student entity may have name, class, and age as attributes.

An attribute is a characteristic or property of an entity. Yes, if person is an entity then same attributes can be used to describe a person in different databases that store medical, student, and fitness club data.

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

CrudRepository	JpaRepository
It is a base interface and extends Repository Interface.	It extends PagingAndSortingRepository that extends CrudRepository.
It contains methods for CRUD operations. For example save(), saveAll(), findById(), findAll(), etc.	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 doesn't provide methods for implementing pagination and sorting	It provides all the methods for which are useful for implementing pagination.
It works as a marker interface.	It extends both CrudRepository and PagingAndSortingRepository.
To perform CRUD operations, define repository extending CrudRepository.	To perform CRUD as well as batch operations, define repository extends JpaRepository.
Syntax: public interface CrudRepository<T, ID> extends Repository<T, ID>	Syntax: public interface JpaRepository<T,ID> extends PagingAndSortingRepository<T,ID>, QueryByExampleExecutor<T>