HOMEWORK4
FATMA BETÜL UYAR

1 – What is JPA ?

JPA is a technology that provides **reading** and **managing relational data** in Java. It allows us to access data between **Java class** and relational **database tables**.
In JPA, data is represented by objects and classes , so we don't have to deal Sql. It uses a **platform-independent query** language called **JPQL**.
It has brought ORMs such as JPA, Hibernate, Toplinkk under **one roof** to make it easier to use. Implementation differences of Hibernate or Toplink, differences in syntax disappeared with JPA.

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

Spring Data JPA supports *find*, *read*, *query*, *count* and *get*. So, we could have done *queryByName*, and Spring Data would behave the same.
The finder method should use a special keyword such as **find** or **get** followed by the name of the variable.
For example,

```
List<User> findByName(String name);
findByNameIsNull(),
findByActiveTrue(),
findByNameStartingWith(String prefix), List<User>
findByAgeLessThan(Integer age);
List<User> findByNameOrBirthDate(String name, ZonedDateTime birthDate);
List<User> findByNameOrderByNameDesc(String name);
```

3 - What is PagingAndSortingRepository ?

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

Page<T>	**findAll(Pageable pageable)**	=> Returns a **Page** of entities
Iterable<T>	**findAll(Sort sort)**	=> Returns all entities sorted by the given options.

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

- **getOne()** returns a **reference to the entity** with the given identifier.
- **findById()** returns **actual objects** the value from the database.

- **getOne** returns a **lazily fetched** entity.I mean, this method will always return a proxy **without** hitting the **database**
- **findById()** returns an **eagerly fetched** entity.

- **getOne()** method throws an **EntityNotFoundException** if the record doesn't exist in the database.
- **findById()** method will return **null** if the record doesn't exist in the database.

- **getOne()** method uses **getReference()** method.
- **findById()** method uses **find()** method.

- **getOne()** method is available in the **JpaRepository** interface.
- **findById()** method is available in the **CrudRepository** interface.

5 - What is @Query used for ?

The @Query is an annotation that is used to **execute both JPQL** and native **SQL** queries.

```
@Query("SELECT u FROM User u WHERE u.status = 1")
Collection<User> findAllActiveUsers();
```

6 - What is lazy loading in hibernate ?

Lazy and Eager are two types of **data loading strategies** that we use when one entity class is having references to other Entities.
Lazy Loading: Associated data loads only when we explicitly call getter or size method.
Egare Loading: Data loading happens at the time their parent is fetched.
In summary, when we **call an entity**, we should use **lazy** if we **do not want** the **related entities.**

```
fetch = FetchType.LAZY
fetch = FetchType.EAGER
```

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

Attackers can capture a SQL database and change or delete it.It is done **by injecting SQL query input from the client** end of the application. If an attacker finds an entry point that has a security bug on the web page or web application, performs a SQL injection attack.

```
query = "SELECT * FROM users WHERE name =' " + name + " ';"
```

The attacker can write or 1=1 instead of name.Comment line characters are used to block the rest of the query.

```
query = "SELECT * FROM users WHERE name =' " + name ' or '1' = '1 + " ';"
```

The query result will **always be positive**, since the **'1'='1'** condition will be satisfied regardless of the username data, and the operation in between is OR. All characters that come after the comment line characters will become comments and they will no longer matter.

As far as I've researched, **hibernate doesn't provide** full protection for sql injection.There is nothing special about HQL (Hibernates subset of SQL) that makes it any more or less susceptible.
To prevent sql injection attacks, we should not write database queries containing user-entered input.

8 - What is criteria API in hibernate ?

Criteria API provides to build up a criteria query object programmatically where you can apply filtration rules and logical conditions.The Hibernate Session interface contains several overloaded **createCriteria()** methods.
For example; I want to get records which salary=3000 ;

```
Criteria cr = session.createCriteria(Employee.class);
cr.add(Restrictions.eq("salary", 3000));
List results = cr.list();
```

Or I want to get records having firstName starting with bet;

```
cr.add(Restrictions.like("firstName", "bet%"));
```

In summary,We can make some Restrictions such as isNull, equal, or sorting.

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

Erlang is a **programming language** that is Functional and Modular.
It is designed for programming large industrial and real-time systems.It is concurrent and functional.It has a very powerful distributed architecture
RabbitMQ runs in Erlang virtual runtime. For this, first of all, Erlang is required to be installed.

10 – What is the JPQL ?

JPQL is a platform-independent **query language**.It is used to perform database operations on persistent entities.Instead of database tables, JPQL uses entities to operate the SQL queries. JPA , transform JPQL into SQL.
It can be used with any type of database, It can perform join operations.

```
Query query = em.createQuery("Select s.s_name from StudentEntity s");
```

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

1. Creating an entity manager factory object
2. Obtaining an entity manager from a factory.
3. Initializing an entity manager.
4. Persisting data into a relational database.
5. Closing the transaction
6. Releasing the factory resources.

12 – What are the different types of entity mapping ?

- **One-to-One:** One entity bean relates only to one other entity bean.
- **One-to-Many Many-to-One:** 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:** In a many-to-many relationship, many objects can reference many objects.

13 - What are the properties of an entity ?

These are the properties of an entity ;

- **Persistability -** An object stored in the database is persistent and can be accessed anytime.
- **Persistent Identity -** Each entity is unique and represented as an object ID.
- **Transactionality -** Entity can perform various operations such as create, delete, update. Each operation makes some changes in the database. It ensures that whatever changes made in the database either be succeed or failed atomically.
- **Granuality -** Entities should not be primitives, primitive wrappers or built-in objects with single dimensional state.

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

JpaRepository will have all the functions of **CrudRepository** and **PagingAndSortingRepository.**
I mean, **CrudRepository only** has **CRUD operations** while **JpaRepository has some extra methods** like flushing data directly to a database.