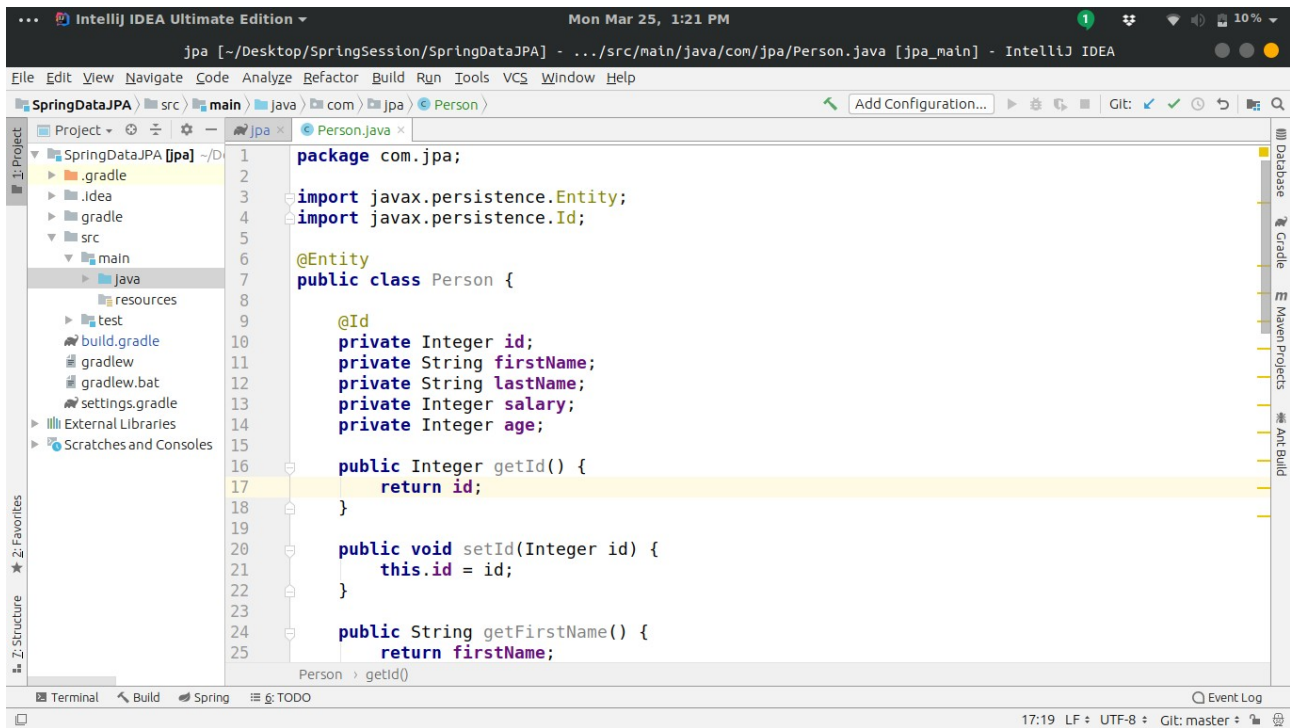


1. Create a Person entity with instance variables Firstname, Lastname, salary, age and Id.

Ans.



```
package com.jpa;

import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class Person {

    @Id
    private Integer id;
    private String firstName;
    private String lastName;
    private Integer salary;
    private Integer age;

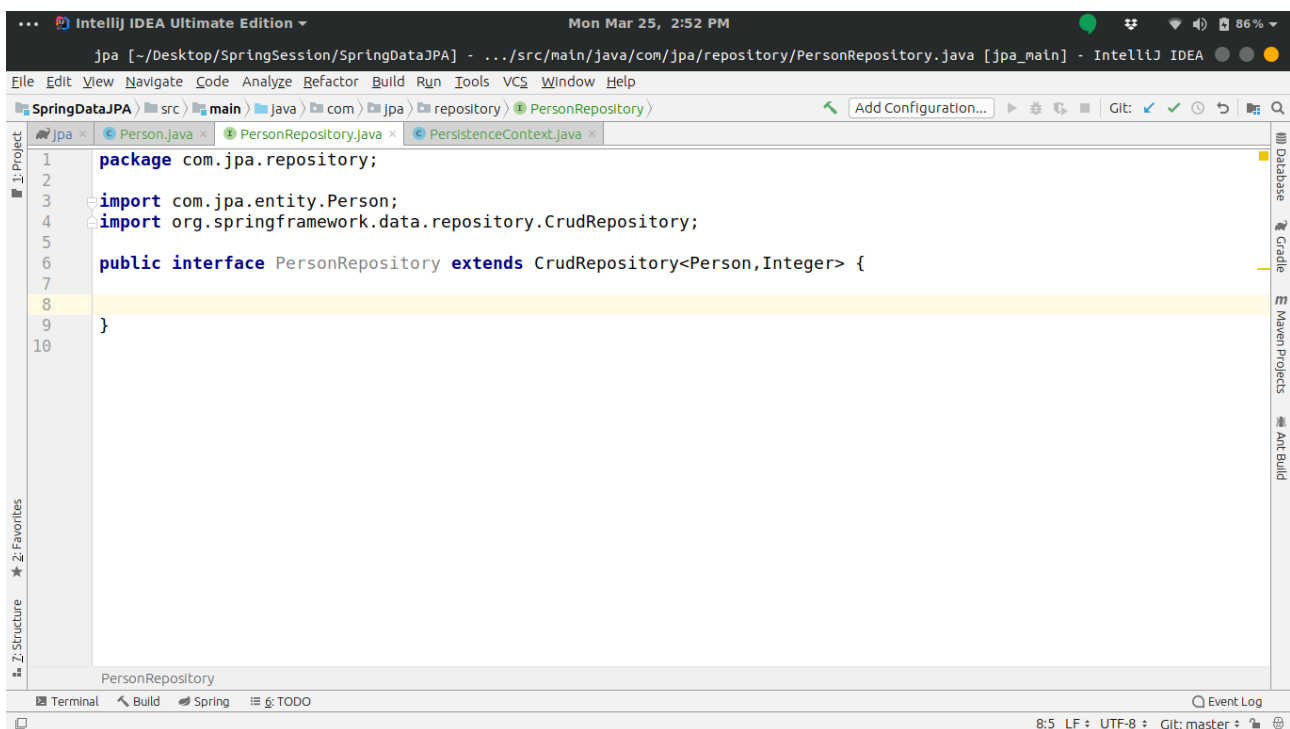
    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
        this.id = id;
    }

    public String getFirstName() {
        return firstName;
    }
}
```

2. Implement CrudRepository for it.

Ans.



```
package com.jpa.repository;

import com.jpa.entity.Person;
import org.springframework.data.repository.CrudRepository;

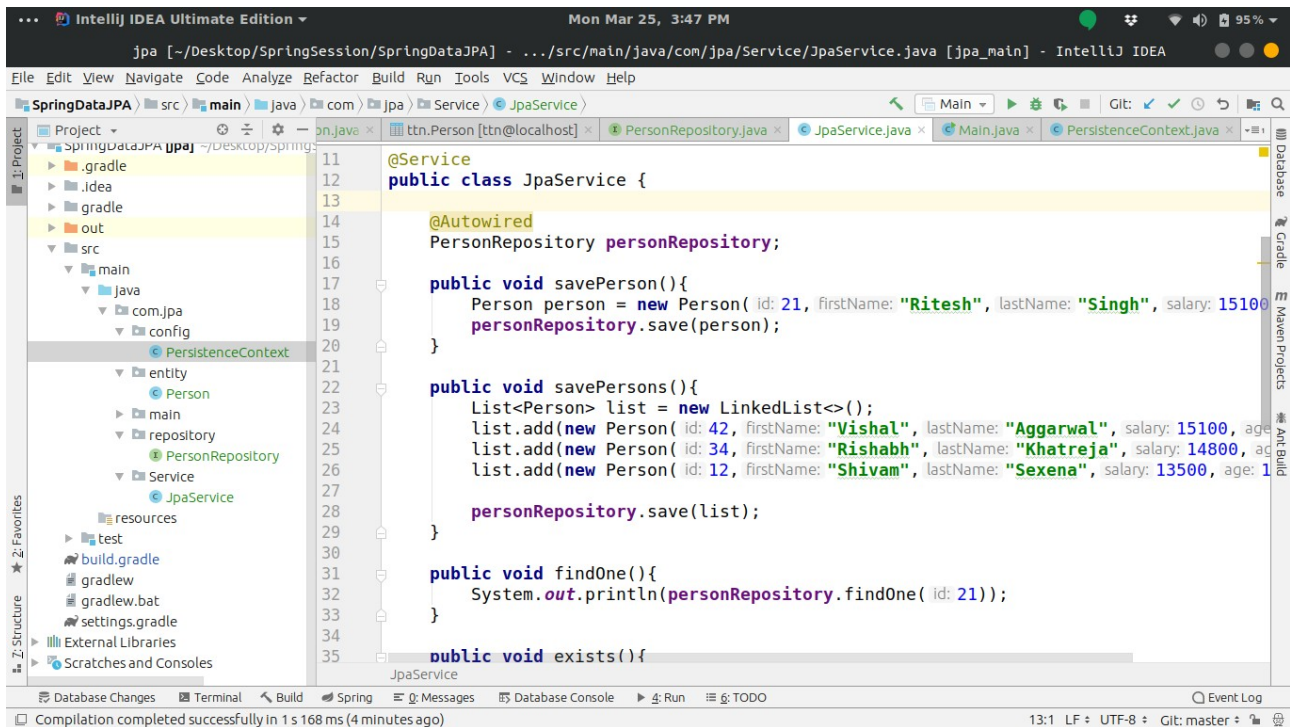
public interface PersonRepository extends CrudRepository<Person,Integer> {

}

PersonRepository
```

3. Perform all the methods inside CrudRepository for Person Class.

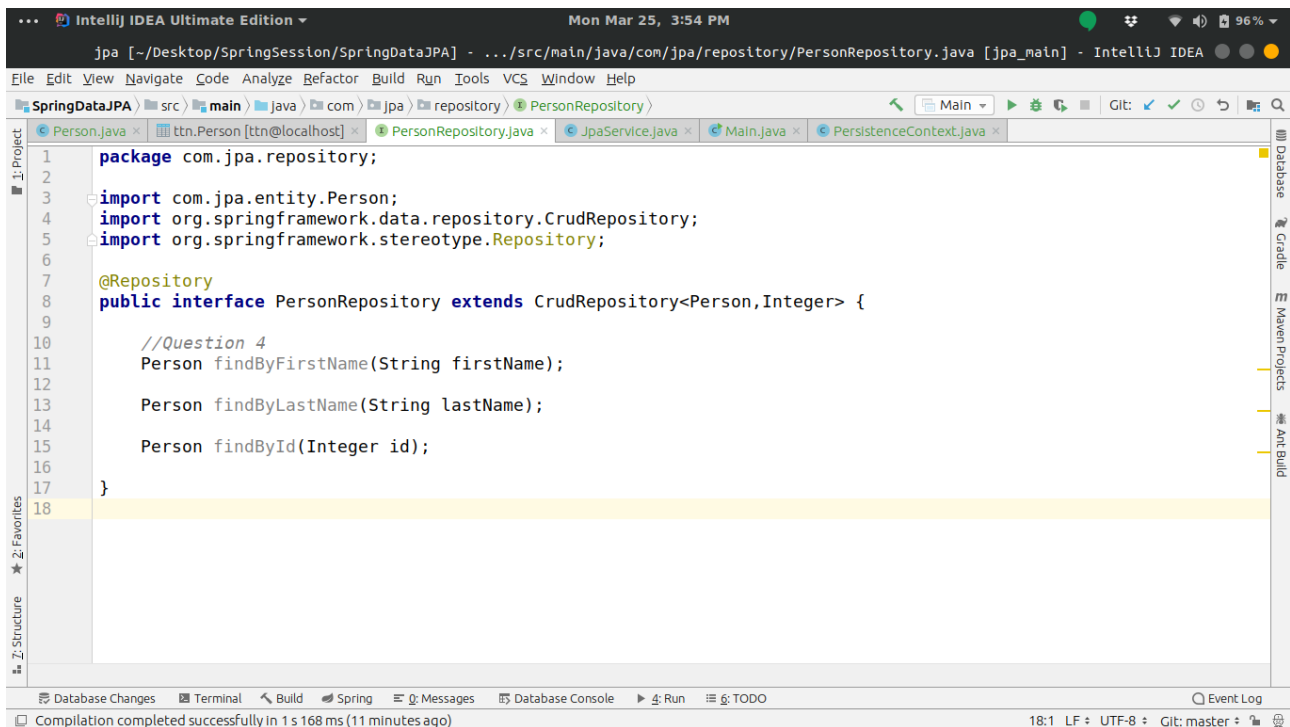
Ans.



```
11 @Service
12 public class JpaService {
13
14     @Autowired
15     PersonRepository personRepository;
16
17     public void savePerson(){
18         Person person = new Person( id: 21, firstName: "Ritesh", lastName: "Singh", salary: 15100);
19         personRepository.save(person);
20     }
21
22     public void savePersons(){
23         List<Person> list = new LinkedList<>();
24         list.add(new Person( id: 42, firstName: "Vishal", lastName: "Aggarwal", salary: 15100, age: 21));
25         list.add(new Person( id: 34, firstName: "Rishabh", lastName: "Khatreja", salary: 14800, age: 22));
26         list.add(new Person( id: 12, firstName: "Shivam", lastName: "Sexena", salary: 13500, age: 18));
27         personRepository.save(list);
28     }
29
30     public void findOne(){
31         System.out.println(personRepository.findOne( id: 21));
32     }
33
34     public void exists(){
35
36     }
```

4. For class Person find person declare methods in repository to find person by firstname, lastname and Id.

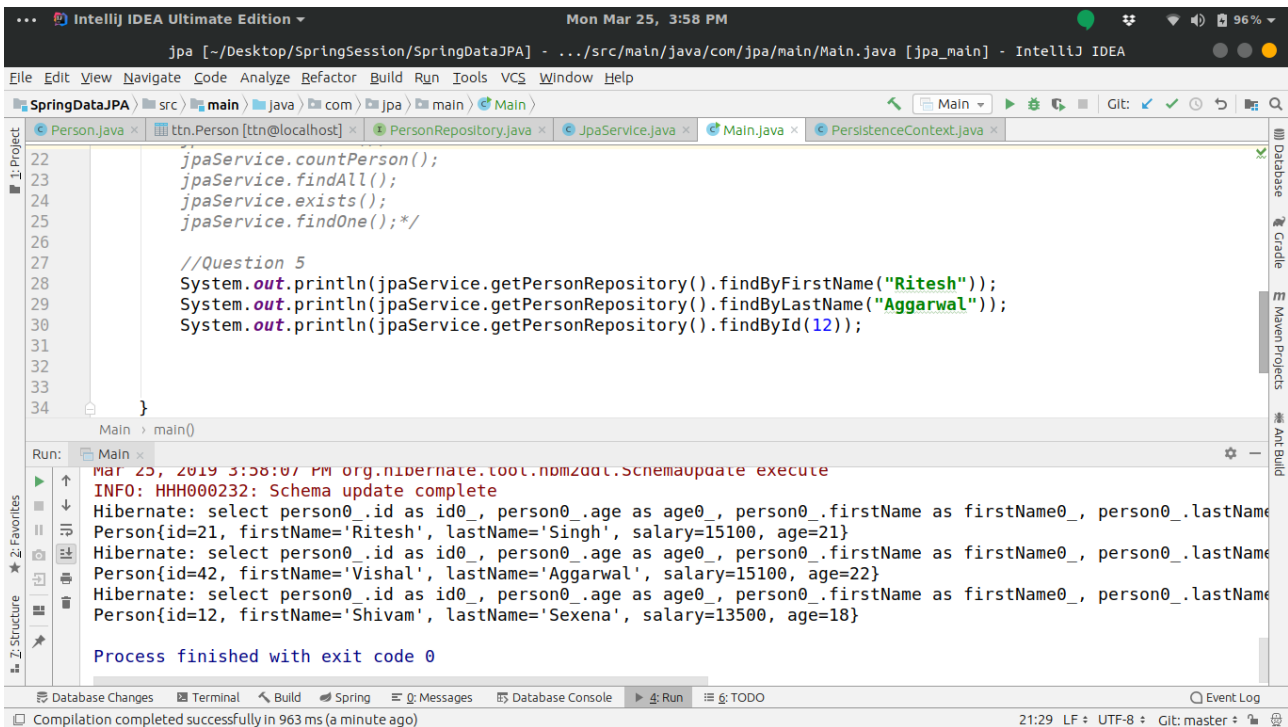
Ans.



```
1 package com.jpa.repository;
2
3 import com.jpa.entity.Person;
4 import org.springframework.data.repository.CrudRepository;
5 import org.springframework.stereotype.Repository;
6
7 @Repository
8 public interface PersonRepository extends CrudRepository<Person,Integer> {
9
10     //Question 4
11     Person findByFirstName(String firstName);
12
13     Person findByLastName(String lastName);
14
15     Person findById(Integer id);
16
17 }
18
```

5. Use the methods declared above to fetch the person.

Ans.



```
... IntelliJ IDEA Ultimate Edition
jpa [-/Desktop/SpringSession/SpringDataJPA] - .../src/main/java/com/jpa/main/Main.java [jpa_main] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

SpringDataJPA | src | main | java | com | jpa | main | Main
Person.java x ttn.Person [ttn@localhost] x PersonRepository.java x JpaService.java x Main.java x PersistenceContext.java x
22 jpaService.countPerson();
23 jpaService.findAll();
24 jpaService.exists();
25 jpaService.findOne();/*
26
27 //Question 5
28 System.out.println(jpaService.getPersonRepository().findByFirstName("Ritesh"));
29 System.out.println(jpaService.getPersonRepository().findByLastName("Aggarwal"));
30 System.out.println(jpaService.getPersonRepository().findById(12));
31
32
33
34 }

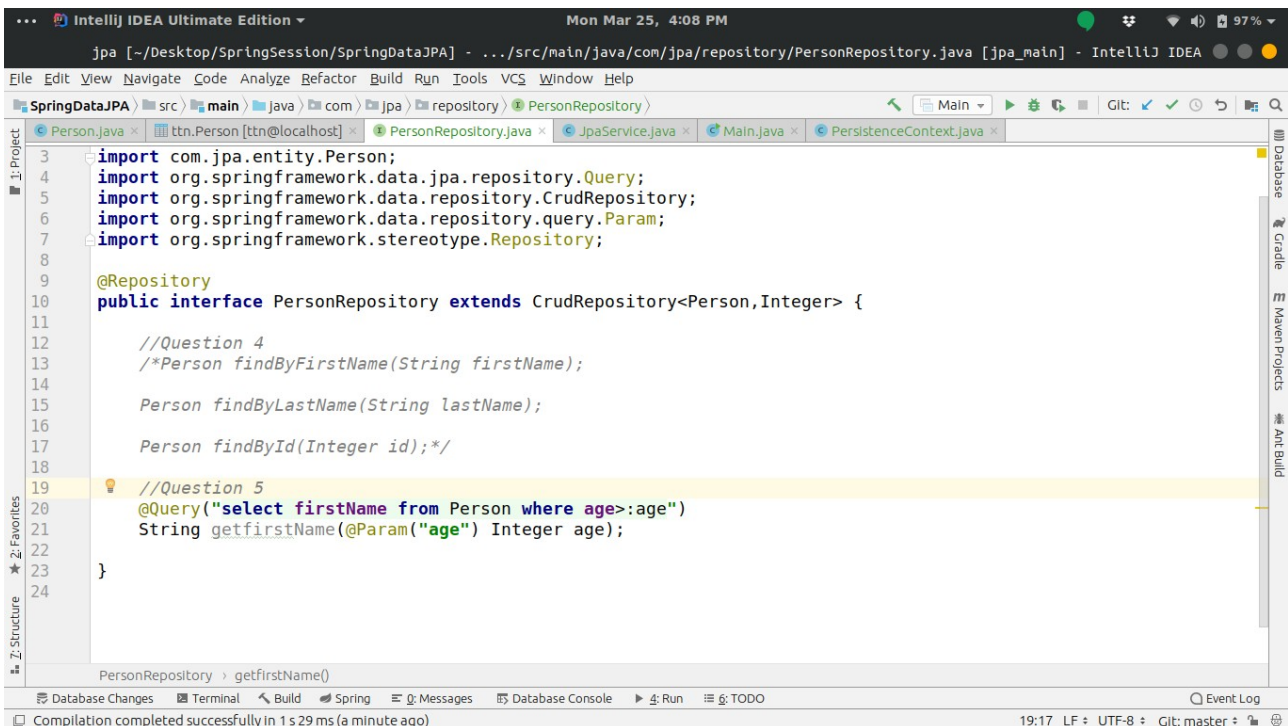
Main | main()
Run: Main
Mar 25, 2019 3:58:07 PM org.hibernate.tool.internal.schema.SchemaUpdate.execute
INFO: HHH000232: Schema update complete
Hibernate: select person0_.id as id0_, person0_.age as age0_, person0_.firstName as firstName0_, person0_.lastName
Person{id=21, firstName='Ritesh', lastName='Singh', salary=15100, age=21}
Hibernate: select person0_.id as id0_, person0_.age as age0_, person0_.firstName as firstName0_, person0_.lastName
Person{id=42, firstName='Vishal', lastName='Aggarwal', salary=15100, age=22}
Hibernate: select person0_.id as id0_, person0_.age as age0_, person0_.firstName as firstName0_, person0_.lastName
Person{id=12, firstName='Shivam', lastName='Sexena', salary=13500, age=18}

Process finished with exit code 0

Database Changes Terminal Build Spring Messages Database Console Run TODO
Compilation completed successfully in 963 ms (a minute ago) 21:29 LF UTF-8 Git: master
```

6. Use @Query to fetch firstname of the Person whose age is 25.

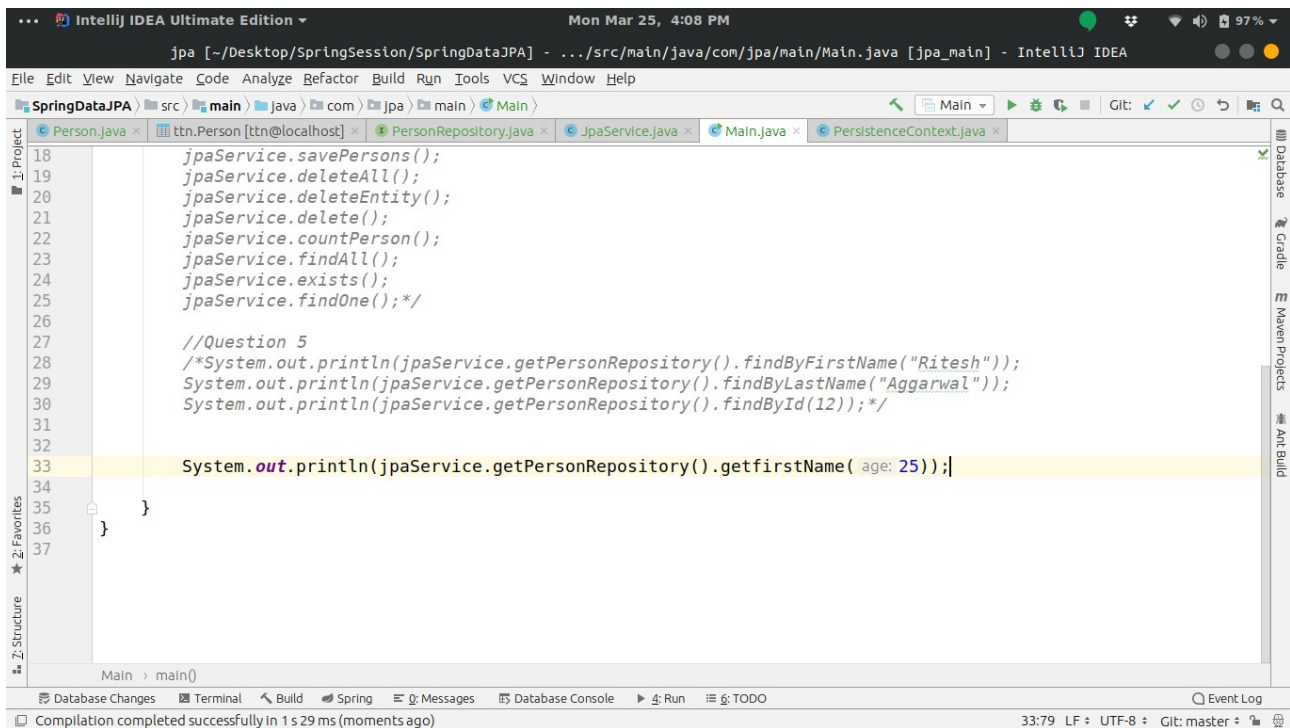
Ans.



```
... IntelliJ IDEA Ultimate Edition
jpa [-/Desktop/SpringSession/SpringDataJPA] - .../src/main/java/com/jpa/repository/PersonRepository.java [jpa_main] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

SpringDataJPA | src | main | java | com | jpa | repository | PersonRepository
Person.java x ttn.Person [ttn@localhost] x PersonRepository.java x JpaService.java x Main.java x PersistenceContext.java x
3 import com.jpa.entity.Person;
4 import org.springframework.data.jpa.repository.Query;
5 import org.springframework.data.repository.CrudRepository;
6 import org.springframework.data.repository.query.Param;
7 import org.springframework.stereotype.Repository;
8
9 @Repository
10 public interface PersonRepository extends CrudRepository<Person,Integer> {
11
12     //Question 4
13     /*Person findByFirstName(String firstName);
14
15     Person findByLastName(String lastName);
16
17     Person findById(Integer id);*/
18
19     //Question 5
20     @Query("select firstName from Person where age>:age")
21     String getFirstName(@Param("age") Integer age);
22
23
24 }

PersonRepository | getFirstName()
Database Changes Terminal Build Spring Messages Database Console Run TODO
Compilation completed successfully in 1 s 29 ms (a minute ago) 19:17 LF UTF-8 Git: master
```

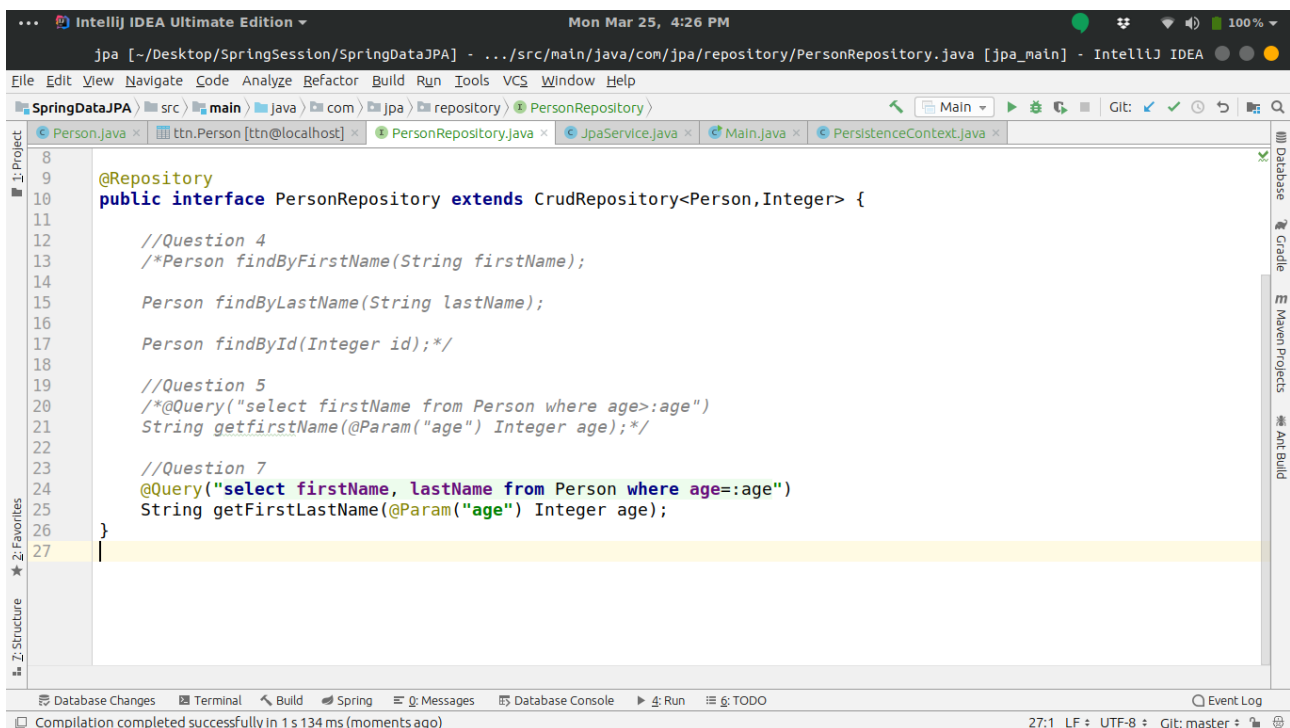


The screenshot shows the IntelliJ IDEA interface with the file `Main.java` open. The code defines a `JpaService` class with various methods for interacting with a JPA repository. The methods include `savePersons()`, `deleteAll()`, `deleteEntity()`, `delete()`, `countPerson()`, `findAll()`, `exists()`, and `findOne()`. There are also test-like calls to `findByFirstName`, `findByLastName`, and `findById`. A specific line is highlighted: `System.out.println(jpaService.getPersonRepository().getfirstName(age: 25));`. The status bar at the bottom indicates that compilation completed successfully.

```
18 jpaService.savePersons();
19 jpaService.deleteAll();
20 jpaService.deleteEntity();
21 jpaService.delete();
22 jpaService.countPerson();
23 jpaService.findAll();
24 jpaService.exists();
25 jpaService.findOne();*/
26
27 //Question 5
28 /*System.out.println(jpaService.getPersonRepository().findByFirstName("Ritesh"));
29 System.out.println(jpaService.getPersonRepository().findByLastName("Aggarwal"));
30 System.out.println(jpaService.getPersonRepository().findById(12));*/
31
32
33 System.out.println(jpaService.getPersonRepository().getfirstName( age: 25));|
34
35 }
36
37
```

7. Use `@Query` to fetch firstname and lastname of the Person whose age is 25.

Ans.

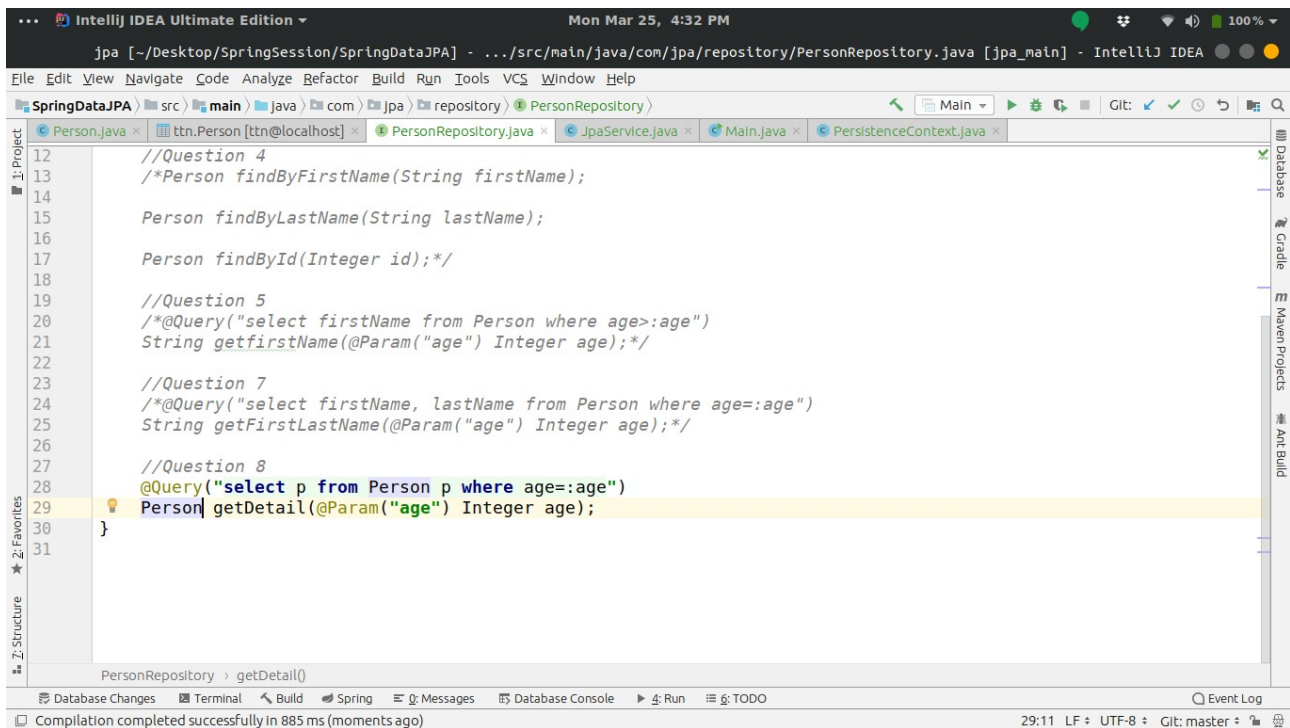


The screenshot shows the IntelliJ IDEA interface with the file `PersonRepository.java` open. The code defines a `PersonRepository` interface that extends `CrudRepository<Person, Integer>`. It includes several methods with `@Query` annotations for specific queries. The highlighted line is: `@Query("select firstName, lastName from Person where age=:age") String getFirstLastName(@Param("age") Integer age);`. The status bar at the bottom indicates that compilation completed successfully.

```
8
9 @Repository
10 public interface PersonRepository extends CrudRepository<Person,Integer> {
11
12     //Question 4
13     /*Person findByFirstName(String firstName);
14
15     Person findByLastName(String lastName);
16
17     Person findById(Integer id);*/
18
19     //Question 5
20     /*@Query("select firstName from Person where age>:age")
21     String getfirstName(@Param("age") Integer age);*/
22
23     //Question 7
24     @Query("select firstName, lastName from Person where age=:age")
25     String getFirstLastName(@Param("age") Integer age);
26
27 }
```

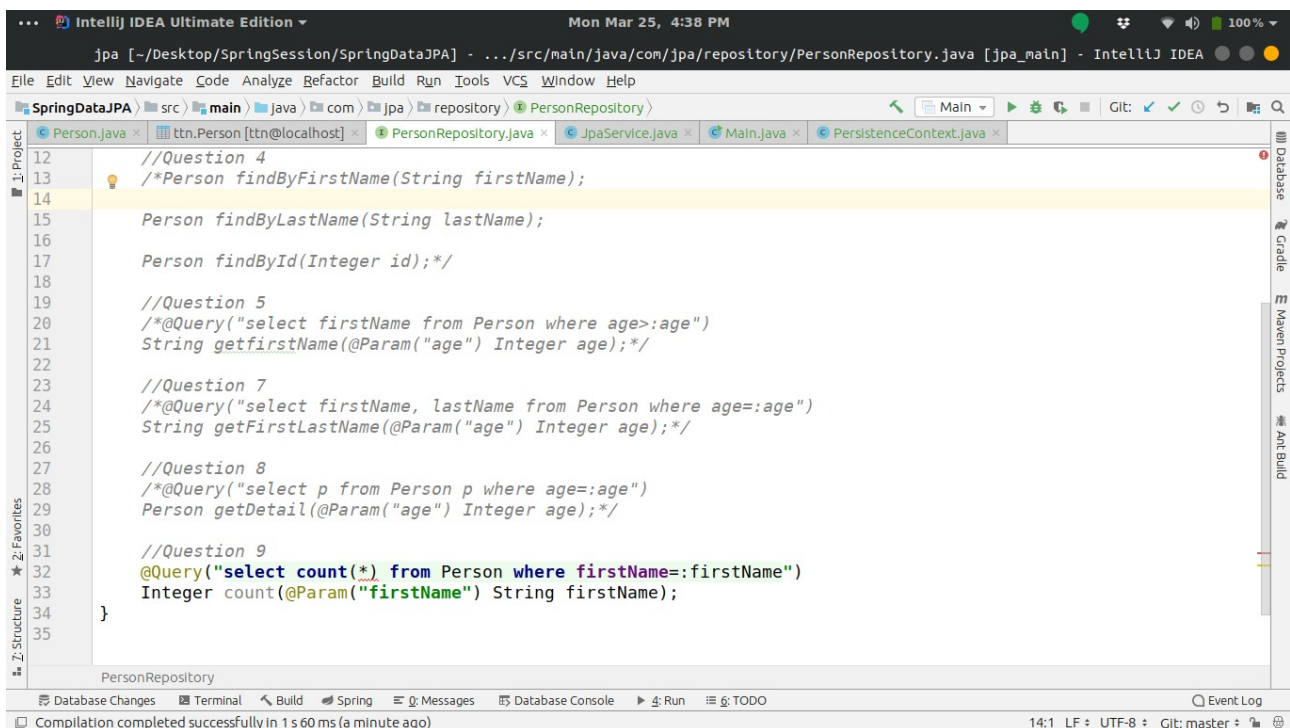
8. Get complete information of the Employee whose age is 25 using `@Query`.

Ans.



9. Count Person where name is "Peter" using @Query.

Ans.

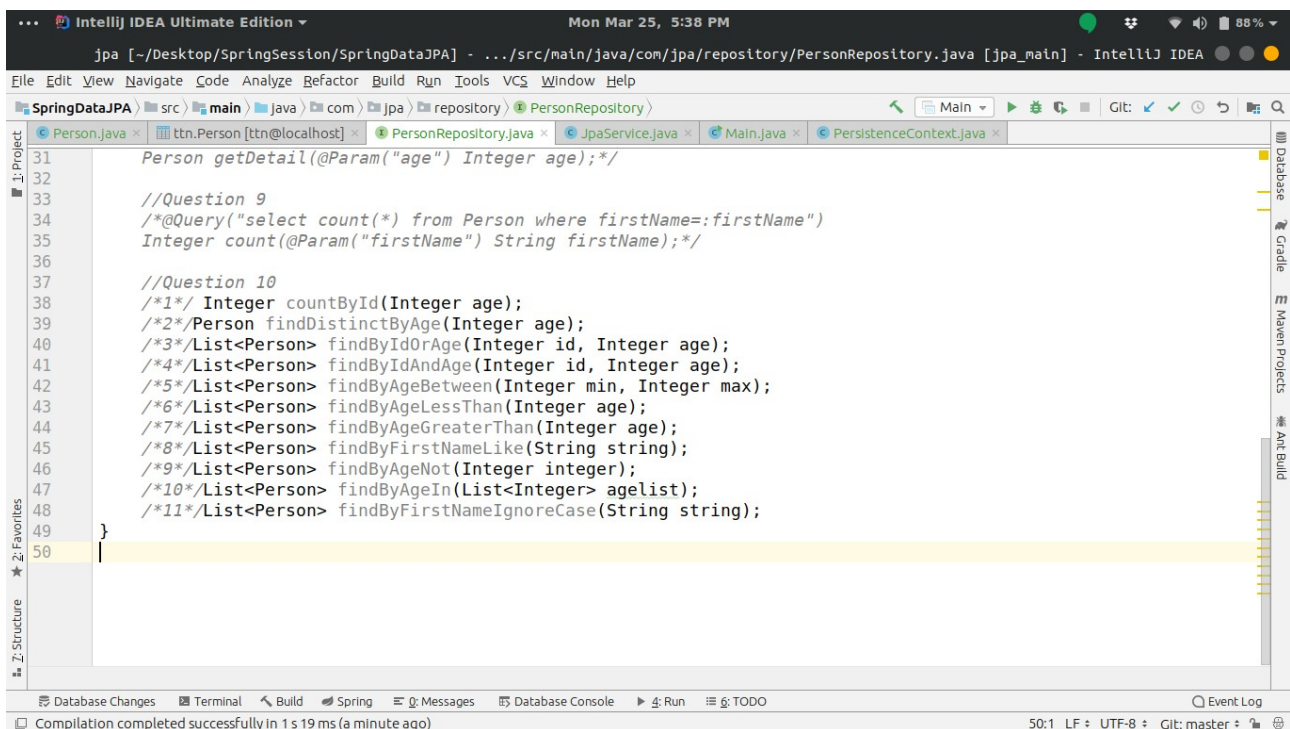


10. Implement following methods for Person repository:

1. count
2. distinct

3. or
4. and
5. between
6. LessThan
7. GreaterThan
8. Like
9. Not
10. In
11. IgnoreCase

Ans



```
31 Person getDetail(@Param("age") Integer age);*/
32
33 //Question 9
34 /*@Query("select count(*) from Person where firstName=:firstName")
35 Integer count(@Param("firstName") String firstName);*/
36
37 //Question 10
38 /*1*/ Integer countById(Integer age);
39 /*2*/ Person findDistinctByAge(Integer age);
40 /*3*/ List<Person> findByIdOrAge(Integer id, Integer age);
41 /*4*/ List<Person> findByIdAndAge(Integer id, Integer age);
42 /*5*/ List<Person> findByAgeBetween(Integer min, Integer max);
43 /*6*/ List<Person> findByAgeLessThan(Integer age);
44 /*7*/ List<Person> findByAgeGreaterThan(Integer age);
45 /*8*/ List<Person> findByFirstNameLike(String string);
46 /*9*/ List<Person> findByAgeNot(Integer integer);
47 /*10*/ List<Person> findByAgeIn(List<Integer> agelist);
48 /*11*/ List<Person> findByFirstNameIgnoreCase(String string);
49 }
50
```

11. Get the persons greater than age 25 and sort them in descending order according to id by method query.

Ans.

```
Integer count(@Param("firstName") String firstName);*/

//Question 10
/*1/** Integer countById(Integer age);
2/** Person findDistinctByAge(Integer age);
3/** List<Person> findByIdOrAge(Integer id, Integer age);
4/** List<Person> findByIdAndAge(Integer id, Integer age);
5/** List<Person> findByAgeBetween(Integer min, Integer max);
6/** List<Person> findByAgeLessThan(Integer age);
7/** List<Person> findByAgeGreaterThan(Integer age);
8/** List<Person> findByFirstNameLike(String string);
9/** List<Person> findByAgeNot(Integer integer);
10/** List<Person> findByAgeIn(List<Integer> agelist);
11/** List<Person> findByFirstNameIgnoreCase(String string);*/

//Question 11
List<Person> findByAgeGreaterThanOrderByIdDesc(Integer integer);
}
```

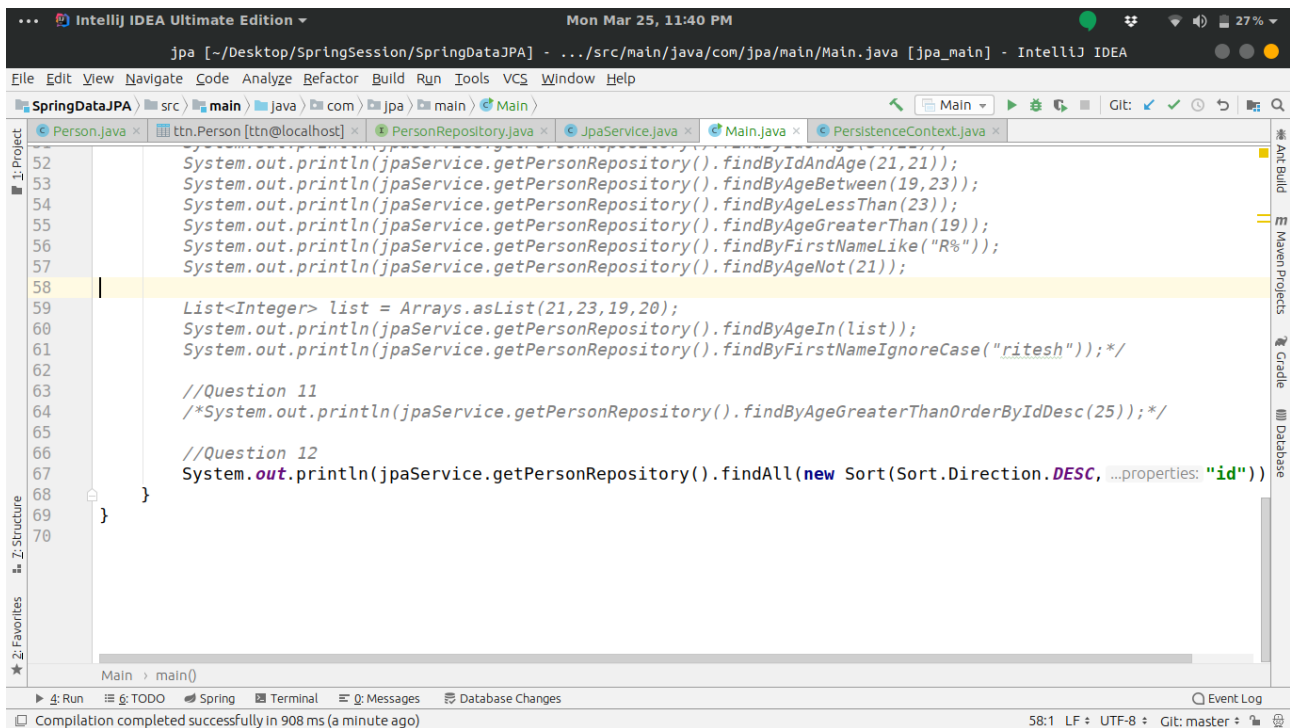
12. Do the question above using the Sort class.

Ans.

```
/*1/** Integer countById(Integer age);
2/** Person findDistinctByAge(Integer age);
3/** List<Person> findByIdOrAge(Integer id, Integer age);
4/** List<Person> findByIdAndAge(Integer id, Integer age);
5/** List<Person> findByAgeBetween(Integer min, Integer max);
6/** List<Person> findByAgeLessThan(Integer age);
7/** List<Person> findByAgeGreaterThan(Integer age);
8/** List<Person> findByFirstNameLike(String string);
9/** List<Person> findByAgeNot(Integer integer);
10/** List<Person> findByAgeIn(List<Integer> agelist);
11/** List<Person> findByFirstNameIgnoreCase(String string);*/

//Question 11
/*List<Person> findByAgeGreaterThanOrderByIdDesc(Integer integer);*/

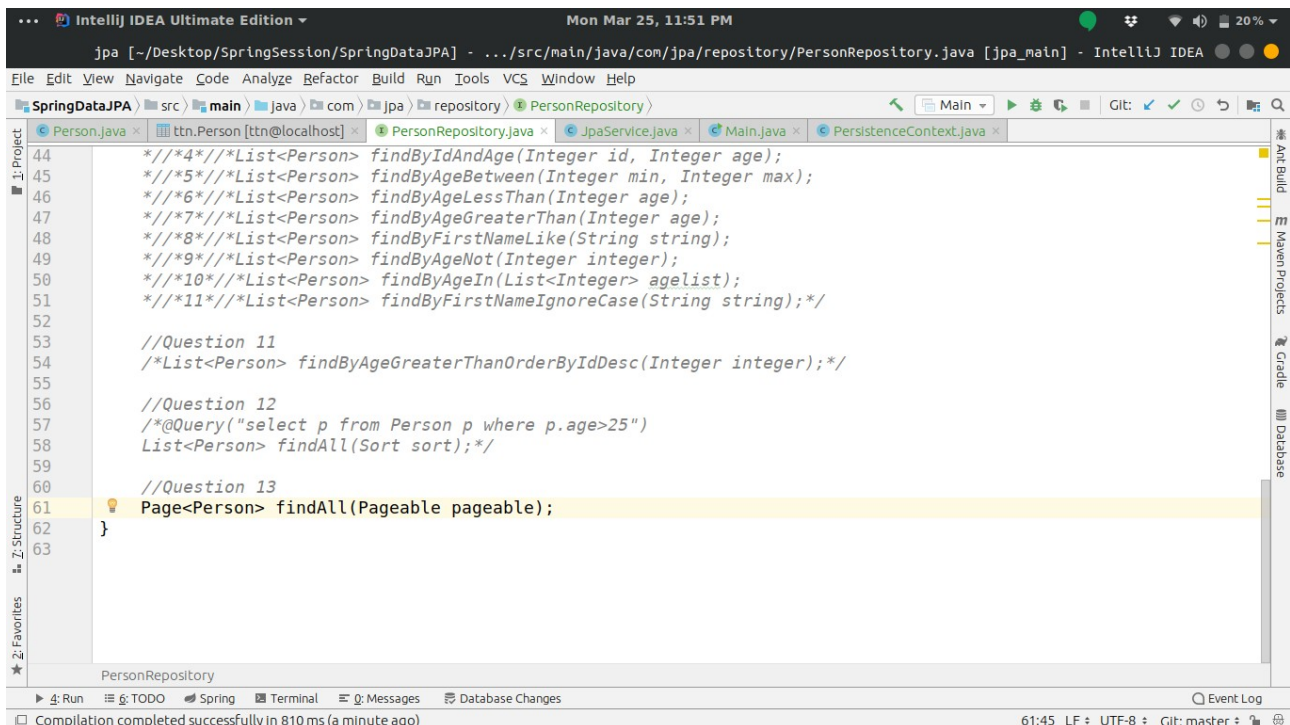
//Question 12
@Query("select p from Person p where p.age>25")
List<Person> findAll(Sort sort);
}
```



```
52 System.out.println(jpaService.getPersonRepository().findByIdAndAge(21,21));
53 System.out.println(jpaService.getPersonRepository().findByAgeBetween(19,23));
54 System.out.println(jpaService.getPersonRepository().findByAgeLessThan(23));
55 System.out.println(jpaService.getPersonRepository().findByAgeGreaterThan(19));
56 System.out.println(jpaService.getPersonRepository().findByFirstNameLike("R%"));
57 System.out.println(jpaService.getPersonRepository().findByAgeNot(21));
58
59 List<Integer> list = Arrays.asList(21,23,19,20);
60 System.out.println(jpaService.getPersonRepository().findByAgeIn(list));
61 System.out.println(jpaService.getPersonRepository().findByFirstNameIgnoreCase("ritesh"));*/
62
63 //Question 11
64 /*System.out.println(jpaService.getPersonRepository().findByAgeGreaterThanOrderByIdDesc(25));*/
65
66 //Question 12
67 System.out.println(jpaService.getPersonRepository().findAll(new Sort(Sort.Direction.DESC, ...properties: "id"))
68 }
69 }
70
```

13. Apply Pagination on Person entities.

Ans.



```
44 /**4/**List<Person> findByIdAndAge(Integer id, Integer age);
45 /**5/**List<Person> findByAgeBetween(Integer min, Integer max);
46 /**6/**List<Person> findByAgeLessThan(Integer age);
47 /**7/**List<Person> findByAgeGreaterThan(Integer age);
48 /**8/**List<Person> findByFirstNameLike(String string);
49 /**9/**List<Person> findByAgeNot(Integer integer);
50 /**10/**List<Person> findByAgeIn(List<Integer> agelist);
51 /**11/**List<Person> findByFirstNameIgnoreCase(String string);*/
52
53 //Question 11
54 /*List<Person> findByAgeGreaterThanOrderByIdDesc(Integer integer);*/
55
56 //Question 12
57 /*@Query("select p from Person p where p.age>25")
58 List<Person> findAll(Sort sort);*/
59
60 //Question 13
61 Page<Person> findAll(Pageable pageable);
62 }
63
```


IntelliJ IDEA Ultimate Edition Mon Mar 25, 11:51 PM

jpa [~/Desktop/SpringSession/SpringDataJPA] - .../src/main/java/com/jpa/main/Main.java [jpa_main] - IntelliJ IDEA

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

SpringDataJPA > src > main > java > com > jpa > main > Main

Main

```
17 public static void main(String[] args) {
59     System.out.println(jpaService.getPersonRepository().findByFirstNameLike("R%"));
60     System.out.println(jpaService.getPersonRepository().findByAgeNot(21));
61
62     List<Integer> list = Arrays.asList(21,23,19,20);
63     System.out.println(jpaService.getPersonRepository().findByAgeIn(list));
64     System.out.println(jpaService.getPersonRepository().findByFirstNameIgnoreCase("ritesh"));*/
65
66     //Question 11
67     /*System.out.println(jpaService.getPersonRepository().findByAgeGreaterThanOrderByIdDesc(25));*/
68
69     //Question 12
70     /*System.out.println(jpaService.getPersonRepository().findAll(new Sort(Sort.Direction.DISC,"id")));*/
71
72     //Question 13
73     Page<Person> personPage = jpaService.getPersonRepository()
74         .findAll(new PageRequest( page: 0, size: 3,new Sort(Sort.Direction.ASC, ...properties: "id")));
75     List<Person> personList = personPage.getContent();
76     System.out.println(personList);
77 }
78
79
```

Main

Run TODO Spring Terminal Messages Database Changes Event Log

Compilation completed successfully in 810 ms (2 minutes ago) 77:6 LF UTF-8 Git: master