Q1. Create a Person entity with instance variables Firstname, Lastname, salary, age and Id.

Solution

Person.java

```
package com.demo.springData.entities;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class Person {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  Integer id;
  String firstName;
  String lastName;
  int salary;
  int age;

 public Person() {
}

public Person(String firstName, String lastName, int salary, int age) {
  this.firstName = firstName;
  this.lastName = lastName;
  this.salary = salary;
  this.age = age;
}


  public Integer getId() {
     return id;
  }

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

  public String getFirstName() {
     return firstName;
  }

  public void setFirstName(String firstName) {
     this.firstName = firstName;
  }

  public String getLastName() {
     return lastName;
```

```
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public int getSalary() {
        return salary;
    }

    public void setSalary(int salary) {
        this.salary = salary;
    }

    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    @Override
    public String toString() {
        return "Person{" +
                "id=" + id +
                ", firstName='" + firstName + '\'' +
                ", lastName='" + lastName + '\'' +
                ", salary=" + salary +
                ", age=" + age +
                '}';
    }
}
```

Q2. Implement CrudRepository for it.

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;

import com.demo.springData.Person.entities.Person;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PersonRepository extends CrudRepository<Person,Integer> {
}
```

Q3. Perform all the methods inside CrudRepository for Person Class.

Solution

PersonService.java

```
package com.demo.springData.Person.service;

import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.repositories.PersonRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PersonService {
  @Autowired
  PersonRepository personRepository;

//    Question3-->Crud repository methods

  public Person save(Person person)
  {
    return personRepository.save(person);
  }

  public Iterable<Person>save(Iterable<Person> personIterable)
  {
     return personRepository.save(personIterable);
  }

  public Person findOne(Integer id)
  {
     return personRepository.findOne(id);
  }

  public boolean exists(Integer id)
  {
     return personRepository.exists(id);
  }

  public Iterable<Person> findAll()
  {
     return personRepository.findAll();
  }

  public Iterable<Person> findAll(Iterable<Integer>ids)
  {
     return personRepository.findAll(ids);
```

```
    }

    public long count()
    {
        return personRepository.count();
    }

    public void delete(Integer id)
    {
        personRepository.delete(id);
    }

    public void delete(Person person)
    {
        personRepository.delete(person);
    }
    public void delete(Iterable<Person> persons)
    {
        personRepository.delete(persons);
    }

    public void deleteAll()
    {
        personRepository.deleteAll();
    }

//    CrudRepository methods end here
}
```

Main.java

```
package com.demo.springData;

import com.demo.springData.config.PersistenceContext;
import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.service.PersonService;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main {

    public static void main(String[] args) {
        ConfigurableApplicationContext context=new AnnotationConfigApplicationContext(PersistenceContext.class);

        PersonService personService=context.getBean(PersonService.class);
```

```
    Person person=new Person("Surbhi","Garg",40000,23);

   List<Person> personList= Arrays.asList(new Person("Abhinav","Shrivastav",70000,28),
           new Person("Gagan","Kushwaha",40000,23));

    System.out.println("*********Question3******************");

    System.out.println("Saved person with following details: "+personService.save(person));

    System.out.println("Saved list of following persons: "+personService.save(personList));

    System.out.println("Person with id 1 is: "+personService.findOne(1));

    System.out.println("Person with id 1 exists? "+personService.exists(1));

    System.out.println("All persons in database: "+personService.findAll());

    System.out.println("Count of persons: "+personService.count());

    personService.delete(2);
    System.out.println("After deleting person with id 2"+personService.findAll());

    personService.delete(person);
    System.out.println("After deleting person object: "+personService.findAll());

    personService.delete(personList);
    System.out.println("After deleting personList: "+personService.findAll());

    personService.save(person);
    personService.save(personList);
    personService.deleteAll();
    System.out.println("After saving again and deleteAll(): "+personService.findAll());



  }
}
```

Output

Q4. For class Person find person declare methods in repository to find person by firstname, lastname and Id.

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;

import com.demo.springData.Person.entities.Person;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PersonRepository extends CrudRepository<Person,Integer> {

  //Question4
  Person findById(Integer id);
  List<Person>findByFirstName(String firstName);
  List<Person>findByLastName(String lastName);
}
```

Q5. Use the methods declared above to fetch the person

Solution

PersonService.java

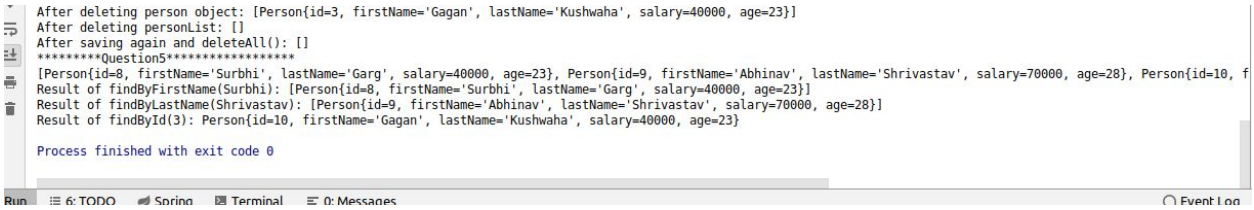
```
package com.demo.springData.Person.service;

import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.repositories.PersonRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PersonService {
  @Autowired
  PersonRepository personRepository;
  //Question5-->findByMethods
  public List<Person> findByFirstName(String firstName)
  {
     return personRepository.findByFirstName(firstName);
  }

  public List<Person>findByLastName(String lastName)
  {
     return personRepository.findByLastName(lastName);
  }
  public Person findById(Integer id)
  {
     return personRepository.findById(id);
  }
  }
```

Main.java

```
package com.demo.springData;

import com.demo.springData.config.PersistenceContext;
import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.service.PersonService;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main {
```

```
  public static void main(String[] args) {
     ConfigurableApplicationContext context=new AnnotationConfigApplicationContext(PersistenceContext.class);

     PersonService personService=context.getBean(PersonService.class);

     Person person=new Person("Surbhi","Garg",40000,23);

    List<Person> personList= Arrays.asList(new Person("Abhinav","Shrivastav",70000,28),
          new Person("Gagan","Kushwaha",40000,23));

     System.out.println("*********Question5******************");
     personService.save(person);
     personService.save(personList);
     System.out.println(personService.findAll());

     System.out.println("Result of findByFirstName(Surbhi): "+personService.findByFirstName("Surbhi"));
     System.out.println("Result of findByLastName(Shrivastav): "+personService.findByLastName("Shrivastav"));
     System.out.println("Result of findById(3): "+personService.findById(10));


  }
}
```

Output



Q6. Use @Query to fetch firstname of the Person whose age is 25.

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;

import com.demo.springData.Person.entities.Person;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PersonRepository extends CrudRepository<Person,Integer> {
  //Question6
  @Query("select firstName from Person where  age=:age")
  List<String> getFirstNamesByAge(@Param("age") Integer age);
}
```

PersonService.java

```
package com.demo.springData.Person.service;

import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.repositories.PersonRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PersonService {
  @Autowired
  PersonRepository personRepository;
  //Question6
  public List<String> getFirstNamesByAge(Integer age)
  {
     return personRepository.getFirstNamesByAge(age);
  }

}
```

Main.java

```
package com.demo.springData;

import com.demo.springData.config.PersistenceContext;
```

```
import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.service.PersonService;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main {

  public static void main(String[] args) {
     ConfigurableApplicationContext context=new
AnnotationConfigApplicationContext(PersistenceContext.class);

     PersonService personService=context.getBean(PersonService.class);

     Person person=new Person("Surbhi","Garg",40000,23);

    List<Person> personList= Arrays.asList(new Person("Abhinav","Shrivastav",70000,28),
         new Person("Gagan","Kushwaha",40000,23));

     System.out.println("*********Question6******************");
     Person person2=new Person("Aakash","Garg",65000,25);
     personService.save(person2);

     Person person3=new Person("Srishti","Singhal",65000,25);
     personService.save(person3);

     System.out.println("FirstName of person with age 25: "+personService.getFirstNamesByAge(25));


  }
}
```

Output

```
Result of findById(3): Person{id=10, firstName='Gagan',
*********Question6******************
FirstName of person with age 25: [Aakash, Srishti]

Process finished with exit code 0
```

Run 6: TODO Spring Terminal 0: Messages

Q7. Use @Query to fetch firstname and lastname of the Person whose age is 25.

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;

import com.demo.springData.Person.entities.Person;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PersonRepository extends CrudRepository<Person,Integer> {
  //Question7
  @Query("select firstName,lastName from Person  where age=:age")
  List<Object[]>getFirstNamesAndLastNamesByAge(@Param("age")Integer age);
}
```

PersonService.java

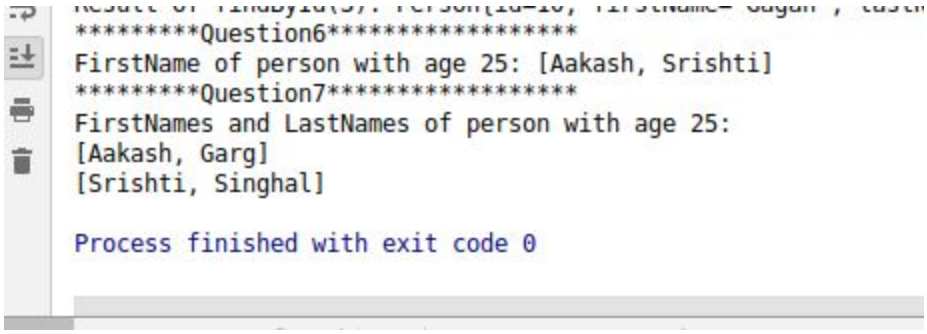
```
package com.demo.springData.Person.service;

import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.repositories.PersonRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PersonService {
  @Autowired
  PersonRepository personRepository;
  //Question7
  public List<Object[]>getFirstNamesAndLastNamesByAge(Integer age)
  {
     return personRepository.getFirstNamesAndLastNamesByAge(age);
  }
}
```

Main.java

```
package com.demo.springData;

import com.demo.springData.config.PersistenceContext;
import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.service.PersonService;
import org.springframework.context.ConfigurableApplicationContext;
```

```
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main {

  public static void main(String[] args) {
    ConfigurableApplicationContext context=new
AnnotationConfigApplicationContext(PersistenceContext.class);

    PersonService personService=context.getBean(PersonService.class);

    Person person=new Person("Surbhi","Garg",40000,23);

   List<Person> personList= Arrays.asList(new Person("Abhinav","Shrivastav",70000,28),
        new Person("Gagan","Kushwaha",40000,23));

     System.out.println("*********Question7*****************");
    System.out.println("FirstNames and LastNames of person with age 25: ");
    personService.getFirstNamesAndLastNamesByAge(25).forEach(ele->
System.out.println(Arrays.toString(ele)));

  }
}
```

Output



Q8. Get complete information of the Employee whose age is 25 using @Query.

Solution

PersonRepositoy.java

```
package com.demo.springData.Person.repositories;

import com.demo.springData.Person.entities.Person;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PersonRepository extends CrudRepository<Person,Integer> {
        //Question8
        @Query("select p from Person p where age=:age")
        List<Person>getPersonsByAge(@Param("age")Integer age);
}
```

PersonService.java

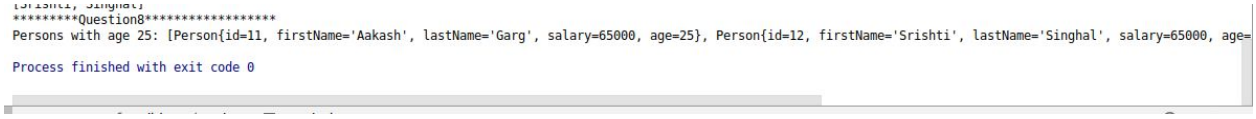
```
package com.demo.springData.Person.service;

import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.repositories.PersonRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PersonService {
  @Autowired
  PersonRepository personRepository;
 //Question8
 public List<Person>getPersonsByAge(Integer age)
 {
   return personRepository.getPersonsByAge(age);
 }

}
```

Main.java

```
package com.demo.springData;

import com.demo.springData.config.PersistenceContext;
import com.demo.springData.Person.entities.Person;
```

```
import com.demo.springData.Person.service.PersonService;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main {

  public static void main(String[] args) {
    ConfigurableApplicationContext context=new
AnnotationConfigApplicationContext(PersistenceContext.class);

    PersonService personService=context.getBean(PersonService.class);

    Person person=new Person("Surbhi","Garg",40000,23);

   List<Person> personList= Arrays.asList(new Person("Abhinav","Shrivastav",70000,28),
        new Person("Gagan","Kushwaha",40000,23));

System.out.println("*********Question8*****************");
System.out.println("Persons with age 25: "+personService.getPersonsByAge(25));


  }
}
```

Output



Q9. Count Person where name is "Peter" using @Query.

Solution

PersonRepositoy.java

```
package com.demo.springData.Person.repositories;

import com.demo.springData.Person.entities.Person;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PersonRepository extends CrudRepository<Person,Integer> {
        //Question9
        @Query("select count(p) from Person p where firstName=:firstName")
        Integer countPersonByFirstName(@Param("firstName")String firstName);

}
```

PersonService.java

```
package com.demo.springData.Person.service;

import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.repositories.PersonRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PersonService {
  @Autowired
  PersonRepository personRepository;
 //Question9
public Integer countPersonByFirstName(String firstName)
{
  return personRepository.countPersonByFirstName(firstName);
}


}
```

Main.java

```
package com.demo.springData;
```

```
import com.demo.springData.config.PersistenceContext;
import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.service.PersonService;
import org.springframework.context.ConfigurableApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.PageRequest;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Main {

  public static void main(String[] args) {
    ConfigurableApplicationContext context=new
AnnotationConfigApplicationContext(PersistenceContext.class);

    PersonService personService=context.getBean(PersonService.class);

    Person person=new Person("Surbhi","Garg",40000,23);

   List<Person> personList= Arrays.asList(new Person("Abhinav","Shrivastav",70000,28),
        new Person("Gagan","Kushwaha",40000,23));

System.out.println("*********Question9******************");
Person person1=new Person("Peter","Parker",50000,22);
personService.save(person1);
System.out.println("Count of persons with firtName=Peter:
"+personService.countPersonByFirstName("Peter"));



  }
}
```

Output

Q10. Implement following methods for Person repository:

- **count**
- **distinct**
- **or**
- **and**
- **between**
- **LessThan**
- **GreaterThan**
- **Like**
- **Not**
- **In**
- **IgnoreCase**

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;

import com.demo.springData.Person.entities.Person;

import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.query.Param;

import java.util.List;

public interface PersonRepository extends CrudRepository<Person,Integer> {
	//Question10
	Integer countByAge(int age);
	Integer countDistinctByFirstName(String firstName);
	List<Person>findByLastNameOrAge(String lastName,Integer age);
	List<Person>findByLastNameAndAge(String lastName,Integer age);
	List<Person>findByAgeBetween(Integer minimum,Integer maximum);
	List<Person>findByAgeLessThan(Integer bound);
	List<Person>findByAgeGreaterThan(Integer bound);
	List<Person>findByFirstNameLike(String pattern);
	List<Person>findBySalaryNot(Integer salary);
	List<Person>findBySalaryIn(List<Integer>salaries);
	List<Person>findByFirstNameIgnoreCase(String firstName);


}
```

PersonService.java

```
package com.demo.springData.Person.service;

import com.demo.springData.Person.entities.Person;
import com.demo.springData.Person.repositories.PersonRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Page;
import org.springframework.data.domain.Pageable;
import org.springframework.stereotype.Service;

import java.util.List;

@Service
public class PersonService {
  @Autowired
  PersonRepository personRepository;
//Question10
public Integer countByAge(int age)
{
   return personRepository.countByAge(age);
}
public Integer countDistinctByFirstName(String firstName)
{
   return personRepository.countDistinctByFirstName(firstName);
}
public List<Person>findByLastNameOrAge(String lastName,Integer age)
{
   return personRepository.findByLastNameOrAge(lastName,age);
}
public List<Person>findByLastNameAndAge(String lastName,Integer age)
{
   return personRepository.findByLastNameAndAge(lastName,age);
}
public List<Person>findByAgeBetween(Integer minimum,Integer maximum)
{
   return personRepository.findByAgeBetween(minimum,maximum);
}
public List<Person>findByAgeLessThan(Integer bound)
{
   return personRepository.findByAgeLessThan(bound);
}
public List<Person>findByAgeGreaterThan(Integer bound)
{
   return personRepository.findByAgeGreaterThan(bound);
}
public List<Person>findByFirstNameLike(String pattern)
{
   return personRepository.findByFirstNameLike(pattern);
}
```

```
public List<Person>findBySalaryNot(Integer salary)
{
   return personRepository.findBySalaryNot(salary);
}
public List<Person>findBySalaryIn(List<Integer>salaries)
{
   return personRepository.findBySalaryIn(salaries);
}
public List<Person>findByFirstNameIgnoreCase(String firstName)
{
   return personRepository.findByFirstNameIgnoreCase(firstName);
}

}
```

Main.java

```
package com.demo.springData;

import com.demo.springData.config.PersistenceContext;

import com.demo.springData.Person.entities.Person;

import com.demo.springData.Person.service.PersonService;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.PageRequest;



import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;



public class Main {

  public static void main(String[] args) {

    ConfigurableApplicationContext context = new
AnnotationConfigApplicationContext(PersistenceContext.class);

    PersonService personService = context.getBean(PersonService.class);
```

```
        Person person = new Person("Surbhi", "Garg", 40000, 23);


        List<Person> personList = Arrays.asList(new Person("Abhinav", "Shrivastav", 70000, 28),new
Person("Gagan", "Kushwaha", 40000, 23));            System.out.println("*********Question10******************");

        Person person4=new Person("Peter","William",50000,35);

        personService.save(person4);


        System.out.println("Current Users");

        personService.findAll().forEach(System.out::println);


        System.out.println("countByAge(25); "+personService.countByAge(25));


        System.out.println("countDistinctByFirstName()"+personService.countDistinctByFirstName("Peter"));


        System.out.println("findByLastNameOrAge(Garg,25):
"+personService.findByLastNameOrAge("Garg",25));


        System.out.println("findByLastNameAndAge(Garg,25):
"+personService.findByLastNameAndAge("Garg",25));


        System.out.println("findByAgeBetween(20,22): "+personService.findByAgeBetween(20,22));


        System.out.println("findByAgeLessThan(23): "+personService.findByAgeLessThan(23));


        System.out.println("findByAgeGreaterThan(23): "+personService.findByAgeGreaterThan(23));


        System.out.println("findByFirstNameLike(__r%): "+personService.findByFirstNameLike("__r%"));
```

```
        System.out.println("findBySalaryNot(50000): "+personService.findBySalaryNot(50000));


        System.out.println("findBySalaryIn(Arrays.asList(40000,50000)):
"+personService.findBySalaryIn(Arrays.asList(40000,50000)));



        System.out.println("findByFirstNameIgnoreCase(Peter):
"+personService.findByFirstNameIgnoreCase("peter"))

    }

}
```

Output

Q11. Get the persons greater than age 25 and sort them in descending order according to id by method query.

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;


import com.demo.springData.Person.entities.Person;


import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.CrudRepository;

import org.springframework.data.repository.query.Param;


import java.util.List;


public interface PersonRepository extends CrudRepository<Person,Integer> {

  //Question11

  List<Person>findByAgeGreaterThanOrderByIdDesc(Integer bound);

}
```

PersonService.java

```
package com.demo.springData.Person.service;


import com.demo.springData.Person.entities.Person;

import com.demo.springData.Person.repositories.PersonRepository;
```

```
import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.stereotype.Service;


import java.util.List;


@Service

public class PersonService {

  @Autowired

  PersonRepository personRepository;

    //Question11

  public List<Person>findByAgeGreaterThanOrderByIdDesc(Integer bound)

  {

     return personRepository.findByAgeGreaterThanOrderByIdDesc(bound);

  }

}
```

Main.java

```
package com.demo.springData;


import com.demo.springData.config.PersistenceContext;

import com.demo.springData.Person.entities.Person;

import com.demo.springData.Person.service.PersonService;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.data.domain.Page;
```

```
import org.springframework.data.domain.PageRequest;


import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;


public class Main {


  public static void main(String[] args) {

    ConfigurableApplicationContext context = new
AnnotationConfigApplicationContext(PersistenceContext.class);



    PersonService personService = context.getBean(PersonService.class);



    Person person = new Person("Surbhi", "Garg", 40000, 23);



    List<Person> personList = Arrays.asList(new Person("Abhinav", "Shrivastav", 70000, 28),

        new Person("Gagan", "Kushwaha", 40000, 23));

   System.out.println("*********Question11*****************");

    System.out.println("findByAgeGreaterThanOrderByIdDesc(23):
"+personService.findByAgeGreaterThanOrderByIdDesc(23));

  }

}
```

Output

```
findByFirstNameIgnoreCase(Peter): [Person{id=13, firstName='Peter', lastName='Parker', salary=50000, age=22}, Person{id=14, firstName='Peter', lastName='William', salary
*********Question11*****************
findByAgeGreaterThanOrderByIdDesc(23): [Person{id=14, firstName='Peter', lastName='William', salary=50000, age=35}, Person{id=12, firstName='Srishti', lastName='Singhal'

Process finished with exit code 0
```

Q12. Do the question above using the Sort class.

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;


import com.demo.springData.Person.entities.Person;


import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.CrudRepository;

import org.springframework.data.repository.query.Param;


import java.util.List;


public interface PersonRepository extends CrudRepository<Person,Integer> {

  //Question12

List<Person>findByAgeGreaterThan(Integer bound, Sort sort);

}
```

PersonService.java

```
package com.demo.springData.Person.service;


import com.demo.springData.Person.entities.Person;

import com.demo.springData.Person.repositories.PersonRepository;

import org.springframework.beans.factory.annotation.Autowired;
```

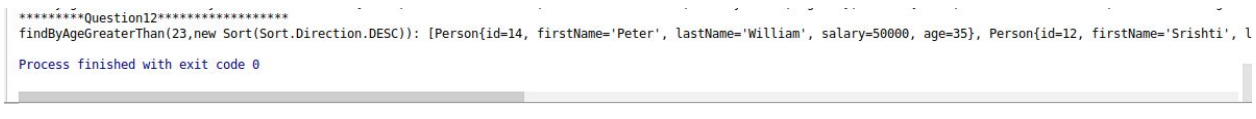
```
import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.stereotype.Service;


import java.util.List;


@Service

public class PersonService {

  @Autowired

  PersonRepository personRepository;

//Question12

public List<Person>findByAgeGreaterThan(Integer bound, Sort sort)

{

   return personRepository.findByAgeGreaterThan(bound,sort);

}


}
```

Main.java

```
package com.demo.springData;


import com.demo.springData.config.PersistenceContext;

import com.demo.springData.Person.entities.Person;

import com.demo.springData.Person.service.PersonService;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.data.domain.Page;
```

```
import org.springframework.data.domain.PageRequest;


import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;


public class Main {


  public static void main(String[] args) {

    ConfigurableApplicationContext context = new
AnnotationConfigApplicationContext(PersistenceContext.class);


    PersonService personService = context.getBean(PersonService.class);


    Person person = new Person("Surbhi", "Garg", 40000, 23);


    List<Person> personList = Arrays.asList(new Person("Abhinav", "Shrivastav", 70000, 28),

        new Person("Gagan", "Kushwaha", 40000, 23));

   System.out.println("*********Question12*****************");

 System.out.println("findByAgeGreaterThan(23,new Sort(Sort.Direction.DESC)):
"+personService.findByAgeGreaterThan(23,new Sort(Sort.Direction.DESC,"id")));

  }

}
```

Output



Q13. Apply Pagination on Person entities.

Solution

PersonRepository.java

```
package com.demo.springData.Person.repositories;


import com.demo.springData.Person.entities.Person;


import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.data.jpa.repository.Query;

import org.springframework.data.repository.CrudRepository;

import org.springframework.data.repository.query.Param;


import java.util.List;


public interface PersonRepository extends CrudRepository<Person,Integer> {

 //Question13

Page<Person> findAll(Pageable pageable);

}
```

PersonService.java

```
package com.demo.springData.Person.service;


import com.demo.springData.Person.entities.Person;

import com.demo.springData.Person.repositories.PersonRepository;

import org.springframework.beans.factory.annotation.Autowired;
```

```
import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.stereotype.Service;


import java.util.List;


@Service

public class PersonService {

  @Autowired

  PersonRepository personRepository;

//Question13

public Page<Person>findAll(Pageable pageable)

{

  return personRepository.findAll(pageable);

}

}
```

Main.java

```
package com.demo.springData;


import com.demo.springData.config.PersistenceContext;

import com.demo.springData.Person.entities.Person;

import com.demo.springData.Person.service.PersonService;

import org.springframework.context.ConfigurableApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.PageRequest;
```

```
import java.util.ArrayList;

import java.util.Arrays;

import java.util.List;


public class Main {


  public static void main(String[] args) {

    ConfigurableApplicationContext context = new
AnnotationConfigApplicationContext(PersistenceContext.class);



    PersonService personService = context.getBean(PersonService.class);


    Person person = new Person("Surbhi", "Garg", 40000, 23);


    List<Person> personList = Arrays.asList(new Person("Abhinav", "Shrivastav", 70000, 28),

        new Person("Gagan", "Kushwaha", 40000, 23));

   System.out.println("*********Question13******************");

        System.out.println("Current Users");

        personService.findAll().forEach(System.out::println);

        System.out.println("Pagination (0th page, 3 records)");

        personService.findAll(new PageRequest(0,3)).getContent().forEach(System.out::println);


  }

}
```

Output


*********Question13******************
Current Users
Person{id=8, firstName='Surbhi', lastName='Garg', salary=40000, age=23}
Person{id=9, firstName='Abhinav', lastName='Shrivastav', salary=70000, age=28}
Person{id=10, firstName='Gagan', lastName='Kushwaha', salary=40000, age=23}
Person{id=11, firstName='Aakash', lastName='Garg', salary=65000, age=25}
Person{id=12, firstName='Srishti', lastName='Singhal', salary=65000, age=25}
Person{id=13, firstName='Peter', lastName='Parker', salary=50000, age=22}
Person{id=14, firstName='Peter', lastName='William', salary=50000, age=35}
Pagination (0th page, 3 records)
Person{id=8, firstName='Surbhi', lastName='Garg', salary=40000, age=23}
Person{id=9, firstName='Abhinav', lastName='Shrivastav', salary=70000, age=28}
Person{id=10, firstName='Gagan', lastName='Kushwaha', salary=40000, age=23}
Process finished with exit code 0
4: Run
6: TODO
Build
Spring
Terminal
0: Messages