(1) Create an Employee Entity which contains following fields

Name

Id

Age

Location

<Employee.java>

```
package com.springdatajpawithhibernatepart1.assignment1.entities;

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

@Entity
public class Employee {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private long id;
  private String name;
  private int age;
  private String location;

  public long getId() {
     return id;
  }

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

  public String getName() {
     return name;
  }

  public void setName(String name) {
     this.name = name;
  }
```

```
    public int getAge() {
        return age;
    }

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

    public String getLocation() {
        return location;
    }

    public void setLocation(String location) {
        this.location = location;
    }

    @Override
    public String toString() {
        return "Employee{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", age=" + age +
                ", location='" + location + '\'' +
                '}';
    }
}
```

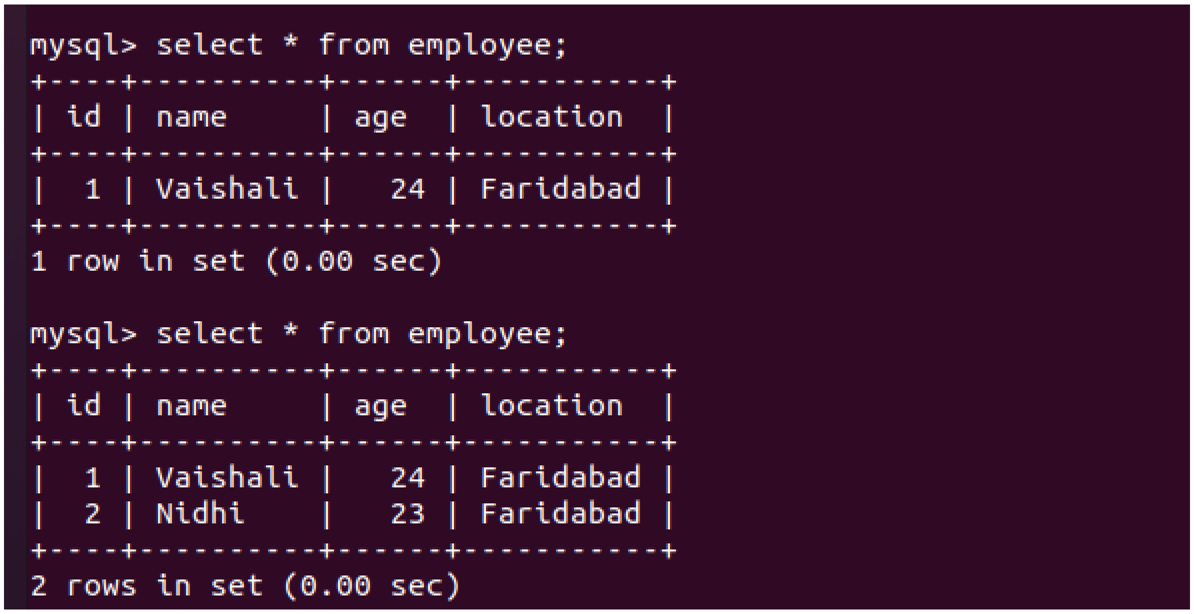
(2) Set up EmployeeRepository with Spring Data JPA

<EmployeeRepository.java>

```
package com.springdatajpawithhibernatepart1.assignment1.repos;

import com.springdatajpawithhibernatepart1.assignment1.entities.Employee;
import org.springframework.data.repository.CrudRepository;
import org.springframework.data.repository.PagingAndSortingRepository;

import java.util.List;

public interface EmployeeRepository extends PagingAndSortingRepository<Employee,Long> {
    List<Employee> findByName(String name);

    List<Employee> findByNameStartingWith(String name);

    List<Employee> findByAgeBetween(int age1, int age2);
}
```

(3) Perform Create Operation on Entity using Spring Data JPA

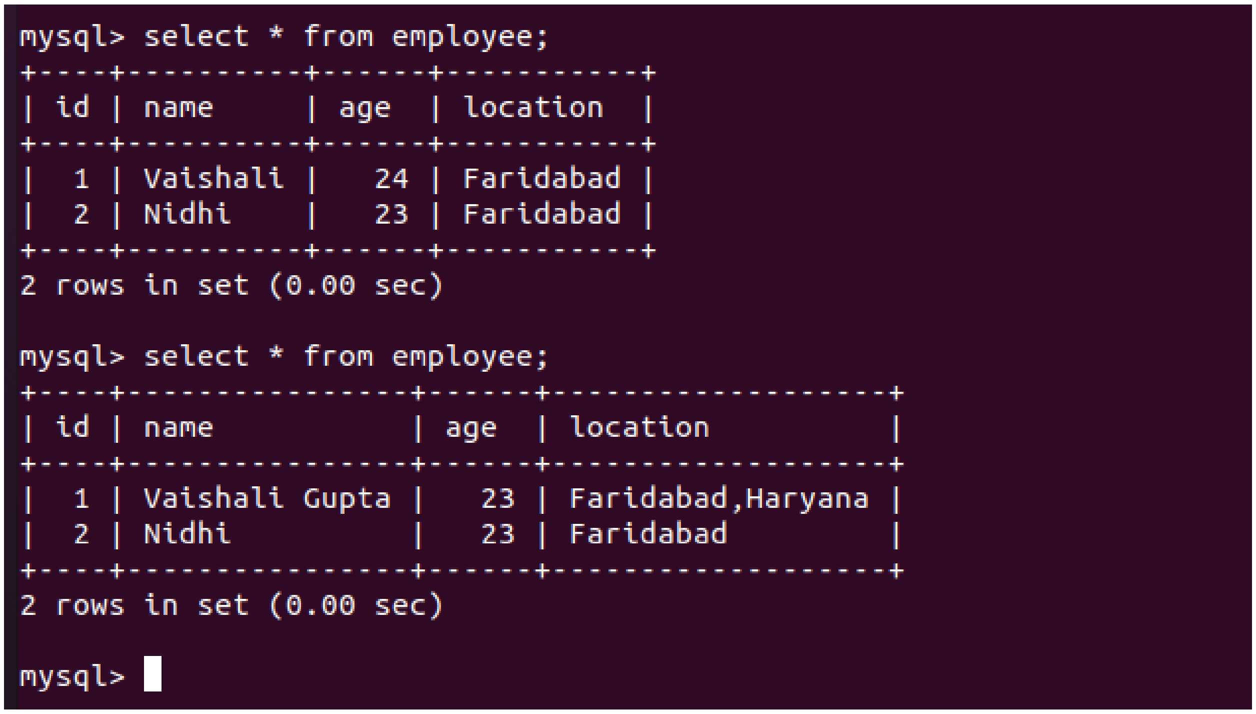
```
/* Perform Create Operation on Entity using Spring Data JPA */
@Test
public void testCreateEmployee(){
  Employee employee = new Employee();
  employee.setAge(26);
  employee.setName("Veena");
  employee.setLocation("UP");
  repository.save(employee);
}
```



```
mysql> select * from employee;
+----+----------+------+-----------+
| id | name     | age  | location  |
+----+----------+------+-----------+
|  1 | Vaishali |   24 | Faridabad |
+----+----------+------+-----------+
1 row in set (0.00 sec)

mysql> select * from employee;
+----+----------+------+-----------+
| id | name     | age  | location  |
+----+----------+------+-----------+
|  1 | Vaishali |   24 | Faridabad |
|  2 | Nidhi    |   23 | Faridabad |
+----+----------+------+-----------+
2 rows in set (0.00 sec)
```

(4) Perform Update Operation on Entity using Spring Data JPA

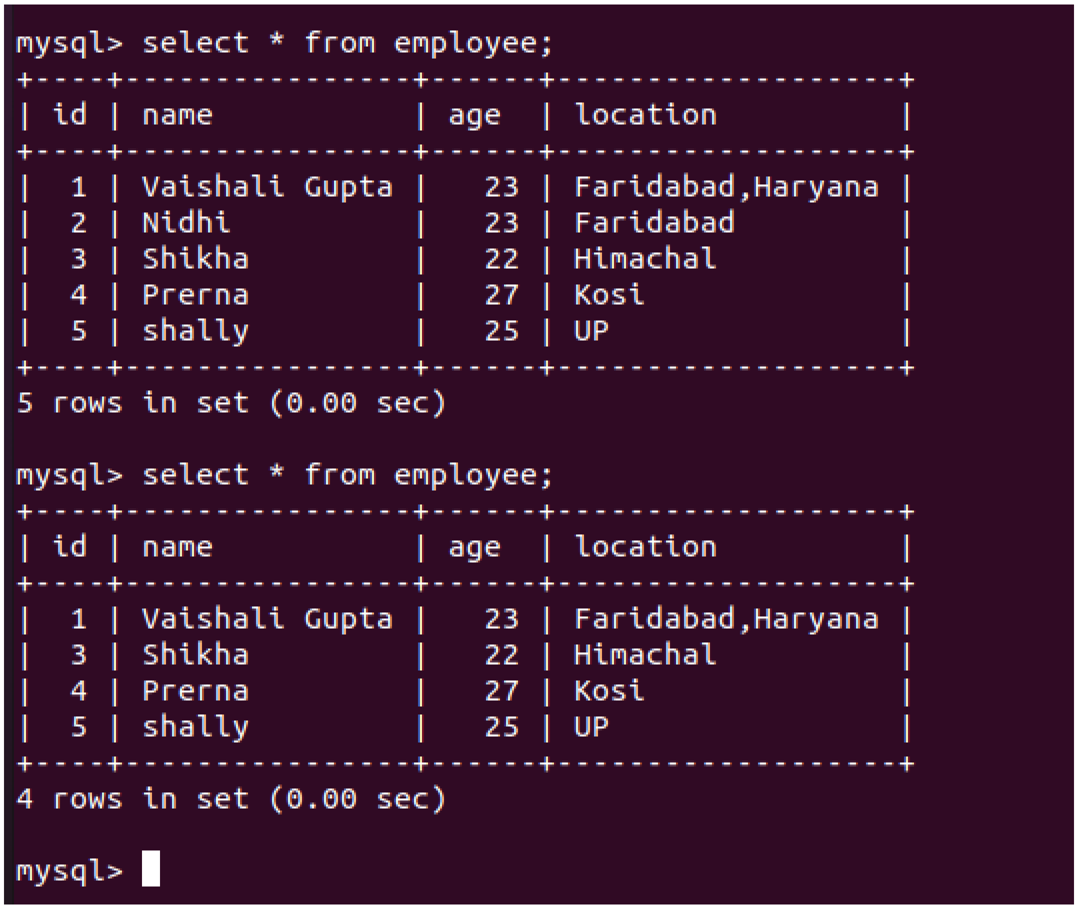
```
/* Perform Update Operation on Entity using Spring Data JPA */
@Test
public void testUpdateEmployee(){
  Employee employee = repository.findById(1l).get();
  employee.setName("Vaishali Gupta");
  employee.setAge(23);
  employee.setLocation("Faridabad,Haryana");
  repository.save(employee);
}
```



```
mysql> select * from employee;
+----+----------+------+-----------+
| id | name     | age  | location  |
+----+----------+------+-----------+
|  1 | Vaishali |   24 | Faridabad |
|  2 | Nidhi    |   23 | Faridabad |
+----+----------+------+-----------+
2 rows in set (0.00 sec)

mysql> select * from employee;
+----+----------------+------+-------------------+
| id | name           | age  | location          |
+----+----------------+------+-------------------+
|  1 | Vaishali Gupta |   23 | Faridabad,Haryana |
|  2 | Nidhi          |   23 | Faridabad         |
+----+----------------+------+-------------------+
2 rows in set (0.00 sec)

mysql> 
```

(5) Perform Delete Operation on Entity using Spring Data JPA

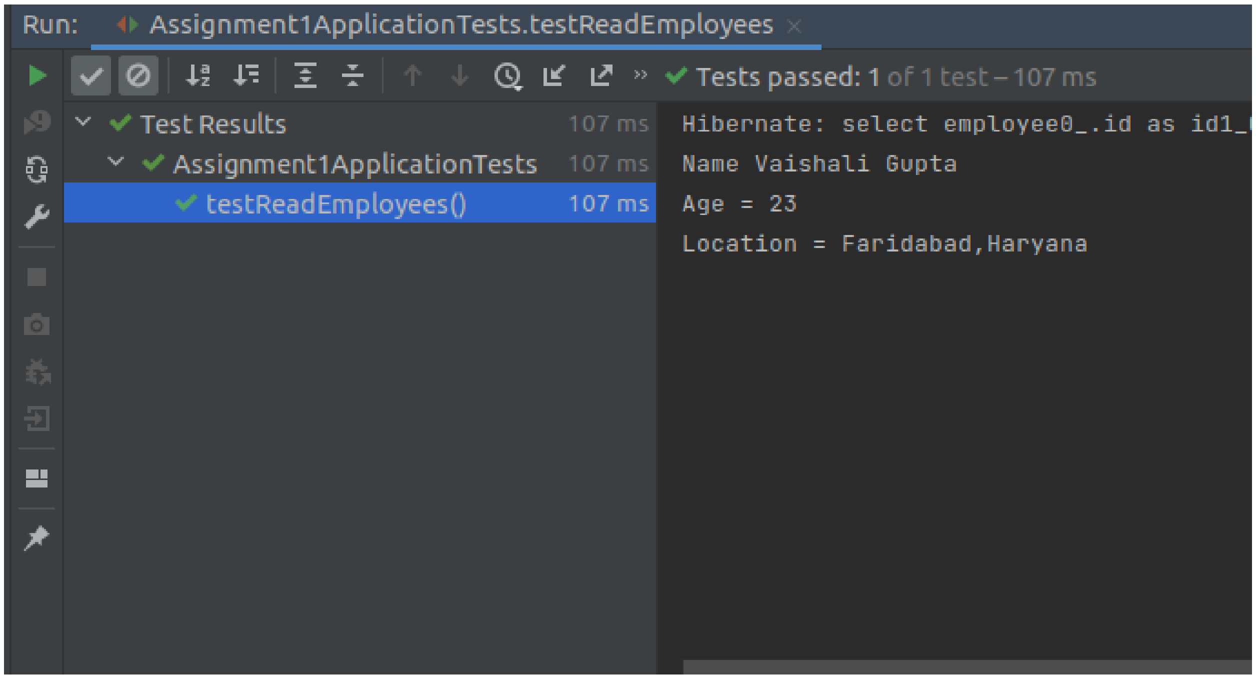
```
/* Perform Delete Operation on Entity using Spring Data JPA */
@Test
public void testDeleteEmployee(){
  Employee employee = repository.findById(2l).get();
  repository.delete(employee);
}
```



```
mysql> select * from employee;
+----+----------------+------+-------------------+
| id | name           | age  | location          |
+----+----------------+------+-------------------+
|  1 | Vaishali Gupta |   23 | Faridabad,Haryana |
|  2 | Nidhi          |   23 | Faridabad         |
|  3 | Shikha         |   22 | Himachal          |
|  4 | Prerna         |   27 | Kosi              |
|  5 | shally         |   25 | UP                |
+----+----------------+------+-------------------+
5 rows in set (0.00 sec)

mysql> select * from employee;
+----+----------------+------+-------------------+
| id | name           | age  | location          |
+----+----------------+------+-------------------+
|  1 | Vaishali Gupta |   23 | Faridabad,Haryana |
|  3 | Shikha         |   22 | Himachal          |
|  4 | Prerna         |   27 | Kosi              |
|  5 | shally         |   25 | UP                |
+----+----------------+------+-------------------+
4 rows in set (0.00 sec)

mysql> 
```

(5) Perform Read Operation on Entity using Spring Data JPA

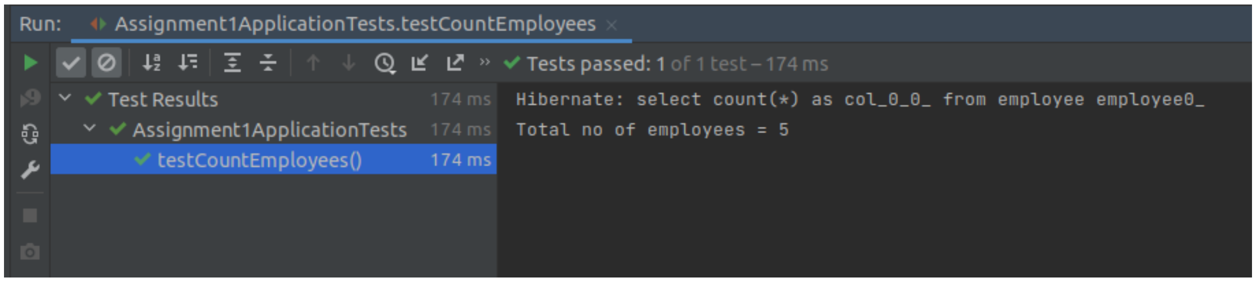
```
/* Perform Read Operation on Entity using Spring Data JPA */
@Test
public void testReadEmployees(){
  Employee employee = repository.findById(1l).get();
  assertEquals("Faridabad,Haryana",employee.getLocation());
  System.out.println("Name " + employee.getName());
  System.out.println("Age = " + employee.getAge());
  System.out.println("Location = " + employee.getLocation());
}
```


Run: Assignment1ApplicationTests.testReadEmployees
Tests passed: 1 of 1 test – 107 ms
Test Results 107 ms
Assignment1ApplicationTests 107 ms
testReadEmployees() 107 ms

```
Hibernate: select employee0_.id as id1_
Name Vaishali Gupta
Age = 23
Location = Faridabad,Haryana
```

(6) Get the total count of the number of Employees

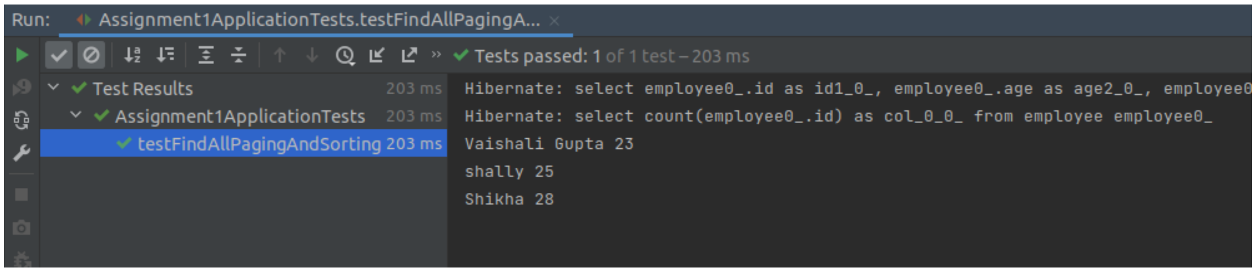
```
/* Get the total count of the number of Employees */
@Test
public void testCountEmployees(){
  System.out.println("Total no of employees = " + repository.count());
}
```


Run: Assignment1ApplicationTests.testCountEmployees
Tests passed: 1 of 1 test – 174 ms
Test Results 174 ms
Assignment1ApplicationTests 174 ms
testCountEmployees() 174 ms

```
Hibernate: select count(*) as col_0_0_ from employee employee0_
Total no of employees = 5
```

(7) Implement Pagination and Sorting on the bases of Employee Age

```
/* Implement Pagination and Sorting on the bases of Employee Age */
@Test
public void testFindAllPagingAndSorting(){
  Pageable pageable = PageRequest.of(0,3, Sort.Direction.ASC,"age");
  repository.findAll(pageable).forEach(p -> System.out.println(p.getName() + " " + p.getAge()));
}
```



Run: Assignment1ApplicationTests.testFindAllPagingA...

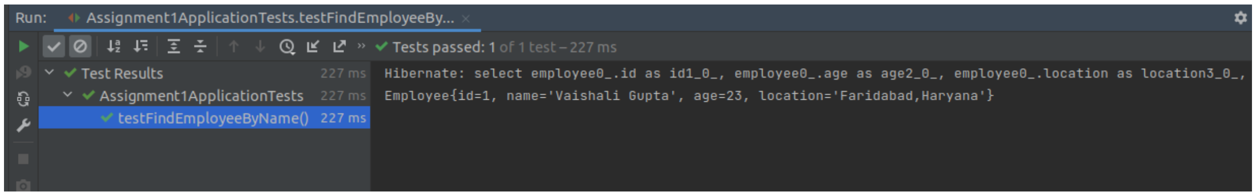
Tests passed: 1 of 1 test – 203 ms

```
Test Results                           203 ms
  Assignment1ApplicationTests          203 ms
    testFindAllPagingAndSorting        203 ms
```

```
Hibernate: select employee0_.id as id1_0_, employee0_.age as age2_0_, employee0
Hibernate: select count(employee0_.id) as col_0_0_ from employee employee0_
Vaishali Gupta 23
shally 25
Shikha 28
```

(8) Create and use finder to find Employee by Name

```
/* Create and use finder to find Employee by Name */
@Test
public void testFindEmployeeByName(){
  List<Employee> employeeList = repository.findByName("Vaishali Gupta");
  employeeList.forEach(employee -> System.out.println(employee));
}
```



Run: Assignment1ApplicationTests.testFindEmployeeBy...

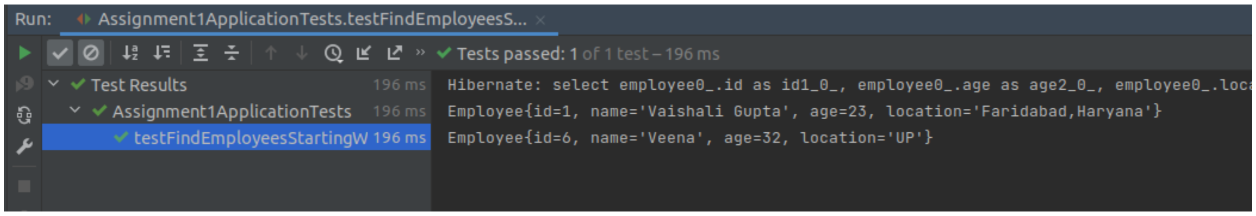
Tests passed: 1 of 1 test – 227 ms

```
Test Results                           227 ms
  Assignment1ApplicationTests          227 ms
    testFindEmployeeByName()           227 ms
```

```
Hibernate: select employee0_.id as id1_0_, employee0_.age as age2_0_, employee0_.location as location3_0_,
Employee{id=1, name='Vaishali Gupta', age=23, location='Faridabad,Haryana'}
```

(9) Create and use finder to find Employees starting with A character

```
/* Create and use finder to find Employees starting with V character */
@Test
public void testFindEmployeesStartingWithVChar(){
  List<Employee> employeeList = repository.findByNameStartingWith("V");
  employeeList.forEach(employee -> System.out.println(employee));
}
```

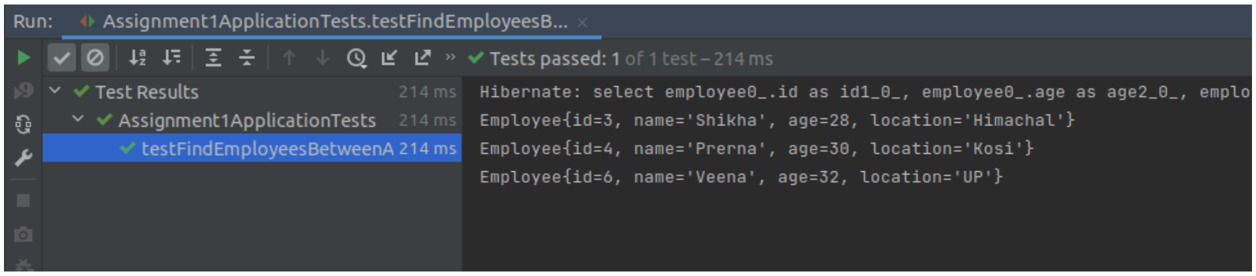


```
Hibernate: select employee0_.id as id1_0_, employee0_.age as age2_0_, employee0_.loca
Employee{id=1, name='Vaishali Gupta', age=23, location='Faridabad,Haryana'}
Employee{id=6, name='Veena', age=32, location='UP'}
```

Run:
Assignment1ApplicationTests.testFindEmployeesS...
Tests passed: 1 of 1 test – 196 ms
Test Results 196 ms
Assignment1ApplicationTests 196 ms
testFindEmployeesStartingW 196 ms

(10) Create and use finder to find Employees Between the age of 28 to 32

```
/* Create and use finder to find Employees Between the age of 28 to 32 */
@Test
public void testFindEmployeesBetweenAge28And32(){
  List<Employee> employeeList = repository.findByAgeBetween(28,32);
  employeeList.forEach(employee -> System.out.println(employee));
}
```



```
Hibernate: select employee0_.id as id1_0_, employee0_.age as age2_0_, emplo
Employee{id=3, name='Shikha', age=28, location='Himachal'}
Employee{id=4, name='Prerna', age=30, location='Kosi'}
Employee{id=6, name='Veena', age=32, location='UP'}
```

Run:
Assignment1ApplicationTests.testFindEmployeesB...
Tests passed: 1 of 1 test – 214 ms
Test Results 214 ms
Assignment1ApplicationTests 214 ms
testFindEmployeesBetweenA 214 ms