Demonstrate implementation of Query Methods feature of Spring Data JPA

```
package com.example.querymethods_demo.entity;

import jakarta.persistence.*;

@Entity
public class Product {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Long id;

  private String name;
  private Double price;
  private String category;

  public Product() {}

  public Product(String name, Double price, String category) {
    this.name = name;
    this.price = price;
    this.category = category;
  }


  public Long getId() { return id; }
  public String getName() { return name; }
```

```
    public Double getPrice() { return price; }
    public String getCategory() { return category; }

    public void setId(Long id) { this.id = id; }
    public void setName(String name) { this.name = name; }
    public void setPrice(Double price) { this.price = price; }
    public void setCategory(String category) { this.category = category; }

    @Override
    public String toString() {
        return "Product{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", price=" + price +
                ", category='" + category + '\'' +
                '}';
    }
}
package com.example.querymethods_demo.repository;

import com.example.querymethods_demo.entity.Product;
import org.springframework.data.jpa.repository.JpaRepository;

import java.util.List;

public interface ProductRepository extends JpaRepository<Product, Long> {
    List<Product> findByCategory(String category);
    List<Product> findByPriceLessThan(Double price);
```

```
    Product findByName(String name);
    List<Product> findByNameContaining(String keyword);
}
package com.example.querymethods_demo;

import com.example.querymethods_demo.entity.Product;
import com.example.querymethods_demo.repository.ProductRepository;
import org.springframework.boot.CommandLineRunner;
import org.springframework.stereotype.Component;

@Component
public class DataLoader implements CommandLineRunner {

    private final ProductRepository repo;

    public DataLoader(ProductRepository repo) {
        this.repo = repo;
    }

    @Override
    public void run(String... args) {
        repo.save(new Product("Laptop", 80000.0, "Electronics"));
        repo.save(new Product("Keyboard", 1500.0, "Electronics"));
        repo.save(new Product("Notebook", 50.0, "Stationery"));
        repo.save(new Product("Pen", 10.0, "Stationery"));
        repo.save(new Product("Refrigerator", 40000.0, "Appliances"));

        System.out.println("Products in Electronics:");
```

```
        repo.findByCategory("Electronics").forEach(System.out::println);

        System.out.println("\nProducts cheaper than 1000:");
        repo.findByPriceLessThan(1000.0).forEach(System.out::println);

        System.out.println("\nProduct named 'Laptop':");
        System.out.println(repo.findByName("Laptop"));

        System.out.println("\nProducts with name containing 'note':");
        repo.findByNameContaining("note").forEach(System.out::println);
    }
}
```

OUTPUT: