

Spring Data Jpa Testing:

```
package integration.tests.databases;
```

```
import java.util.List;
```

```
import javax.validation.ConstraintViolationException;
```

```
import org.assertj.core.api.Assertions;
```

```
import org.junit.jupiter.api.AfterEach;
```

```
import org.junit.jupiter.api.Test;
```

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

```
import org.springframework.boot.test.context.SpringBootTest;
```

```
import org.springframework.test.context.jdbc.Sql;
```

```
import integration.tests.databases.product.Product;
```

```
import integration.tests.databases.product.ProductRepository;
```

```
@SpringBootTest
```

```
class JpaIntegrationDataJpaTest {
```

```
    @Autowired
```

```
    private ProductRepository productRepository;
```

```
    @AfterEach
```

```
    public void cleanUp() {
```

```
        productRepository.deleteAll();
```

```
    }
```

```
    @Test
```

```
    @Sql(statements = "insert into product (name, price) values('Mouse',15)")
```

```
    public void findByNameIgnoreCaseReturnAProduct() {
```

```
        Product product =
```

```
productRepository.findByNameIgnoreCase("mouse");
```

```
        Assertions.assertThat(product.getName()).isEqualTo("Mouse");
```

```
        Assertions.assertThat(product.getPrice()).isEqualTo(15);
```

```
    }
```

```
    @Test
```

```
    public void saveAProductWithNameNullThrowsAnException() {
```

```
        Product product = new Product("", 60);
```

```
        Assertions.assertThatThrownBy(()->{
```

```
            productRepository.save(product);
```

```

        }).isInstanceOf(ConstraintViolationException.class)
        .hasMessageContaining("Name is mandatory")
        .hasMessageContaining("Price must be lower than fifty");
    }

    @Test
    @Sql(statements = "insert into product (name,price)
values('Scanner',40)")
    @Sql(statements = "insert into product (name,price) values('Mouse',15)")
    public void
findAllProductsOrderedAscendentByPriceReturnAnOrderedListOfProducts() {

        List<Product> products =
productRepository.findAllOrderedByPriceAsc();

        Assertions.assertThat(products).extracting(Product::getName).containsExactly
("Mouse", "Scanner");

    }
}

```