**Spring Data Story**

* **Theory:**

Spring Framework - a platform for developing software applications, may include code libraries, a compiler, and other programs used in the software development process

* Spring Boot
* Spring Data
* Spring Data Repositories
* Spring Data Query Creation
* Spring Data Services

Spring Annotations:

* @Component - tells Spring to manage this class for us
* @Configuration
* @Bean
* @Service
* @Repository
* @Autowired - сладаме само на този конструктор, за който искаме да ни се инстанцират елементите в него. Ако имаме само един конструктор, тази анотация не ни е нужна
* @MappedSuperclass - to be inherited correctly from other classes
* @Entity
* @Table

Relationships in Spring:

* @OneToOne
* @OneToMany
* @ManyToMany

JPA/Hibernate Cascade Types(in *OneToMany* associations, we've mentioned cascade type in the annotation)

Gson

Validation/Package javax.validation.constraints

IoC container

**Practical Task:**

1. **Setup Spring Project:**

* Using Intellij idea community version: <https://start.spring.io/>

Add dependencies:

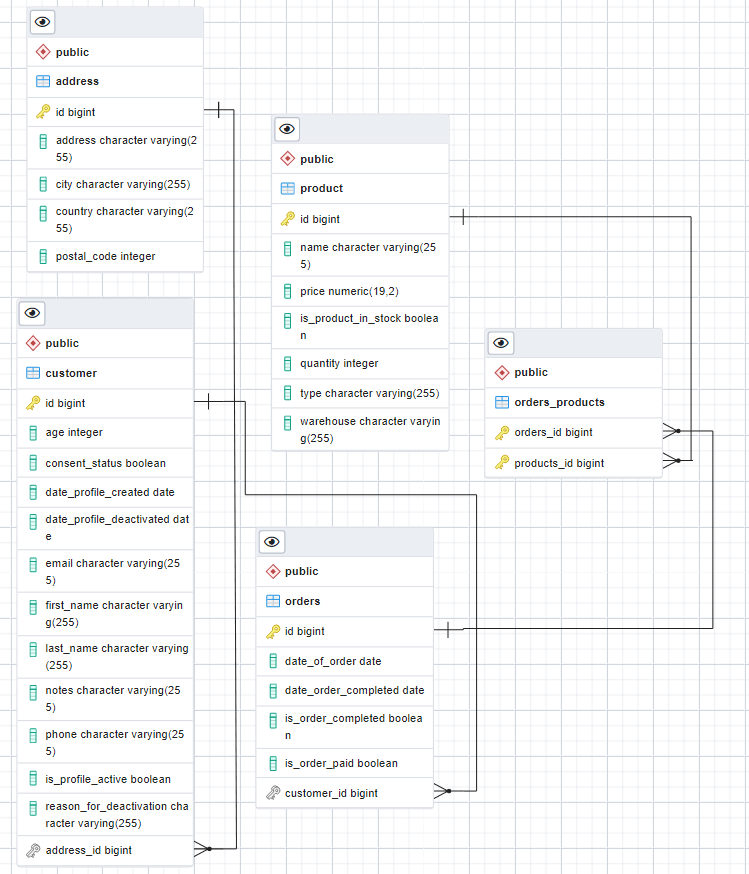
-> Spring Data JPA

-> PostgreSQL Driver

Export .jip and open project

Hint: <https://www.geeksforgeeks.org/spring-data-jpa-column-annotation/>

1. **Create Database by Code First Approach with Spring Framework:**



* **Customer <-> Address (Many to One relationship)**
* **Customer - > Orders (One to Many relationship)**
* **Orders <-> Products (Many to Many relationship)**
* **Customer Table:**
* Id Long, Not Null, Auto Increment
* First name, Not Null
* Last name, Not Null
* Age, Integer (1-100)
* Emai, Unique, Not Null
* Phone, Not Null,
* Consent\_status, Boolean, Not Null
* Is\_profile\_active Boolean, Not Null
* Date\_profile\_created, LocalDate, Not Null
* Date\_profile\_deactivated, LocalDate
* Reason\_for\_deactivation, String
* notes TEXT, max 200 characters
* Address\_id (FOREIGN KEY)
* Set<Orders> orders
* **Address Table:**
* Id Long, Not Null, Auto Increment
* Postal Code, Integer, positive number
* Address, String
* City, String Not Null
* Country, String, Not Null
* Set<Customer>
* **Product Table:**
* Id Long, Not Null, Auto Increment
* Name, String, Not Null
* Quantity, Integer, Not Null, Positive number
* Type, String, Not Null
* Price, BigDecimal, Not Nul, Positivel
* Is the product in Stock, Boolean, Not Null
* Warehouse, String, Not Null
* **Orders Table:**
* Id Long, Not Null, Auto Increment
* Is Order Completed, Boolean, Not Null
* Is Order Paid, Boolean, Not Null
* Date of Order, LocalDate, Not Null
* Date Order Completed, LocalDate
* Customer\_id, Not Null (FOREIGN KEY)
* Set<Product> products

1. **Fill the information in the tables with the information from the attached JSON files**

* **Useful Resources**

<https://www.baeldung.com/jpa-one-to-one>

<https://www.baeldung.com/hibernate-one-to-many>

<https://www.baeldung.com/jpa-many-to-many>

<https://www.baeldung.com/jpa-cascade-types>

<https://www.objectdb.com/api/java/jpa/Column>

<https://docs.jboss.org/hibernate/jpa/2.1/api/javax/persistence/Column.html>

<https://docs.oracle.com/javaee/6/api/javax/validation/Validator.html>

<https://docs.oracle.com/javaee/7/api/javax/validation/constraints/package-summary.html#:~:text=constraints%20Description,constraints%20on%20basic%20JDK%20types>.