

## **Домашнее задание по теме «JDBC, JPA»**

### **Формулировка задания:**

1. Реализовать проект для структуры данных из прошлого задания. Использовать JPA, Hibernate только в качестве JPA-провайдера. Сгенерировать таблицы в схеме shop базы данных PostgreSQL.

#### **Покупатель**

- id
- имя/фамилия

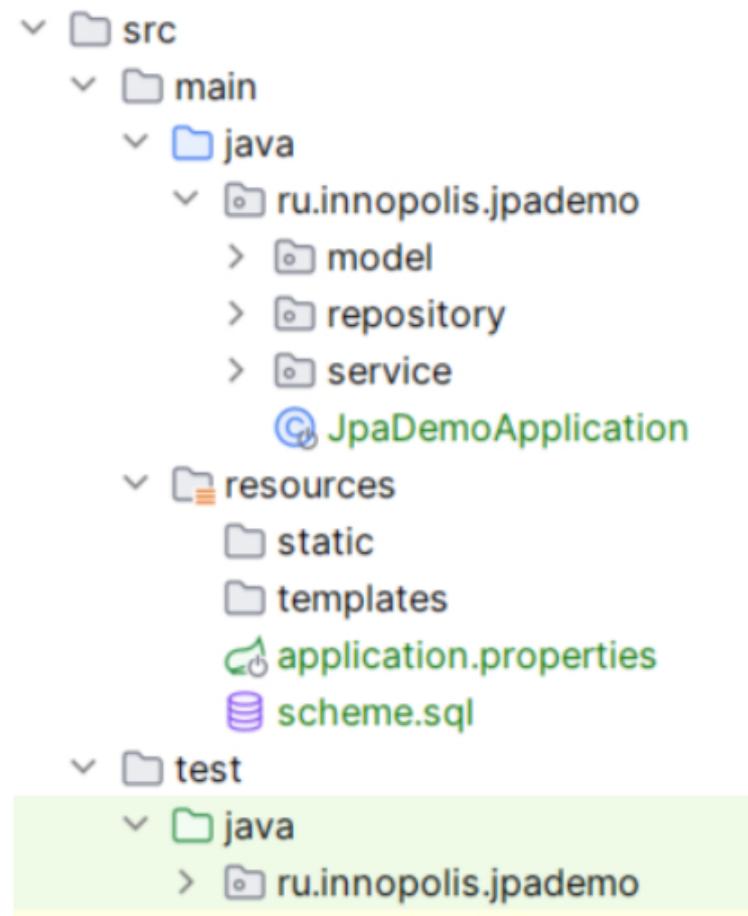
#### **Заказ**

- id-заказа
- id-покупателя (внешний ключ)
- дата заказа
- общее число заказов
- скидка покупателя

2. Использовать методы JpaRepository для следующих действий:

- получение всех данных таблицы
- удаление всех данных таблицы
- получение данных по id
- Получение данных по дате

3. Дополнительно. Реализовать следующую структуру проекта:



В файле schema.sql реализовать заполнение данных в базу.

4. Дополнительно. Настроить файл миграций базы с помощью flyway.

```
homework18
  .mvn
    wrapper
  src
    main
      java
        ru.klgd.javadev
          entity
            Order
            User
          repository
            OrderRepository
            UserRepository
          service
            OrderService
            UserService
            App
        resources
          static
          templates
          application.properties
          schema.sql
    test
  target
  .gitattributes
  .gitignore
  mvnw
  mvnw.cmd
  pom.xml
```

application.properties

```
1 spring.application.name=homework18
2
3 spring.datasource.driver-class-name=org.postgresql.Driver
4 spring.datasource.url=${DATASOURCE_URL:jdbc:postgresql://localhost:6543/postgres}
5 spring.datasource.username=${DATASOURCE_USERNAME:postgres}
6 spring.datasource.password=${DATASOURCE_PASSWORD:0001}
```

```
1 package ru.klgd.javadev;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.SpringBootApplication;
5
6  @SpringBootApplication & programakc *
7   public class App {
8
9  public static void main(String[] args) { & programakc
10           SpringApplication.run(App.class, args);
11
12       }
13
14 }
```

```
"C:\Program Files\Java\jdk-21.0.2\bin\java.exe" ...
```

```
.   ---  
/\ / ___'_ _ _ _(_)_ -- _ _ _ \ \ \ \  
( )\___ | '_| '_| '_\ \_` | \ \ \ \  
\ \_ )| |_)| | | | | | ( | | ) ) )  
' | ____| .__|_|_|_|_\_, | / / / /  
=====|_|=====|_|/_=/_|/_/_/
```

```
:: Spring Boot ::          (v3.5.7)
```

```
2025-10-29T23:51:12.709+02:00  INFO 20856 --- [homework18] [main] ru.klgd.javadev.App : Starting App using Java 21.0.2 with PID  
20856 (D:\JavaProject\JavaDevHomeworks\homework18\target\classes started by Dannelion in D:\JavaProject\JavaDevHomeworks)  
2025-10-29T23:51:12.712+02:00  INFO 20856 --- [homework18] [main] ru.klgd.javadev.App : No active profile set, falling back to 1  
default profile: "default"  
2025-10-29T23:51:13.464+02:00  INFO 20856 --- [homework18] [main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA  
repositories in DEFAULT mode.  
2025-10-29T23:51:13.528+02:00  INFO 20856 --- [homework18] [main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning  
in 54 ms. Found 2 JPA repository interfaces.  
2025-10-29T23:51:13.994+02:00  INFO 20856 --- [homework18] [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)  
2025-10-29T23:51:14.011+02:00  INFO 20856 --- [homework18] [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]  
2025-10-29T23:51:14.011+02:00  INFO 20856 --- [homework18] [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache  
Tomcat/10.1.48]  
2025-10-29T23:51:14.074+02:00  INFO 20856 --- [homework18] [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded  
WebApplicationContext  
2025-10-29T23:51:14.074+02:00  INFO 20856 --- [homework18] [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:  
initialization completed in 1292 ms  
2025-10-29T23:51:14.164+02:00  INFO 20856 --- [homework18] [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...  
2025-10-29T23:51:14.359+02:00  INFO 20856 --- [homework18] [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection
```

```
org.postgresql.jdbc.PgConnection@61615b4a
2025-10-29T23:51:14.360+02:00  INFO 20856 --- [homework18] [
2025-10-29T23:51:14.443+02:00  INFO 20856 --- [homework18] [
  PersistenceUnitInfo [name: default]
2025-10-29T23:51:14.497+02:00  INFO 20856 --- [homework18] [
  6.6.33.Final
2025-10-29T23:51:14.534+02:00  INFO 20856 --- [homework18] [
2025-10-29T23:51:14.872+02:00  INFO 20856 --- [homework18] [
  class transformer
2025-10-29T23:51:14.966+02:00  INFO 20856 --- [homework18] [
  Database JDBC URL [Connecting through datasource 'HikariDataSource (HikariPool-1)']
  Database driver: undefined/unknown
  Database version: 18.0
  Autocommit mode: undefined/unknown
  Isolation level: undefined/unknown
  Minimum pool size: undefined/unknown
  Maximum pool size: undefined/unknown
2025-10-29T23:51:15.825+02:00  INFO 20856 --- [homework18] [
  (set 'hibernate.transaction.jta.platform' to enable JTA platform integration)
2025-10-29T23:51:15.827+02:00  INFO 20856 --- [homework18] [
  main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for
  persistence unit 'default'
2025-10-29T23:51:16.147+02:00  WARN 20856 --- [homework18] [
  main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by
  default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2025-10-29T23:51:16.521+02:00  INFO 20856 --- [homework18] [
  main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with
  context path '/'
2025-10-29T23:51:16.528+02:00  INFO 20856 --- [homework18] [
  main] ru.klgd.javadev.App : Started App in 4.418 seconds (process
  running for 5.297)
|
```

```
1 package ru.klgd.javadev.entity;  
2  
3 import jakarta.persistence.*;  
4 import lombok.AllArgsConstructor;  
5 import lombok.Getter;  
6 import lombok.NoArgsConstructor;  
7 import lombok.Setter;  
8  
9 @Entity new *  
10 @Getter  
11 @Setter  
12 @NoArgsConstructor  
13 @AllArgsConstructor  
14 @Table(name = "users", schema = "public")  
15  public class User {  
16  
17     @Id  
18     @GeneratedValue(strategy = GenerationType.IDENTITY)  
19     @Column(name = "id")  
20      private Long id;  
21  
22     @Column(name = "name")  
23      private String name;  
24 }
```

```
1 package ru.klgd.javadev.entity;
2
3 import jakarta.persistence.*;
4 import lombok.AllArgsConstructor;
5 import lombok.Getter;
6 import lombok.NoArgsConstructor;
7 import lombok.Setter;
8
9 import java.util.Date;
10
11 @Entity new *
12 @Getter
13 @Setter
14 @NoArgsConstructor
15 @AllArgsConstructor
16 @Table(name = "orders", schema = "public")
17 public class Order {
18
19     @Id
20     @GeneratedValue(strategy = GenerationType.IDENTITY)
21     @Column(name = "id")
22     private Long id;
23
24     @ManyToOne
25     @JoinColumn(name = "buyer_id")
26     private User user;
27
28     @Column(name = "order_date")
29     private Date date;
```

```
30  
31     @Column(name = "count_orders")  
32 ④     private Integer countOrders;  
33  
34     @Column(name = "discount")  
35 ④     private Integer discount;  
36  
37 }
```

```
1 package ru.klgd.javadev.repository;  
2  
3 import org.springframework.data.jpa.repository.JpaRepository;  
4 import ru.klgd.javadev.entity.User;  
5  
6  public interface UserRepository extends JpaRepository<User, Long> { 2 usages new *  
7 }
```

```
1 package ru.klgd.javadev.repository;  
2  
3 import org.springframework.data.jpa.repository.JpaRepository;  
4 import ru.klgd.javadev.entity.Order;  
5  
6 import java.util.Date;  
7 import java.util.List;  
8  
9  public interface OrderRepository extends JpaRepository<Order, Long> { 2 usages new *  
10  
11      List<Order> findAllByDate(Date date); 1 usage new *  
12  
13 }
```

```
1 package ru.klgd.javadev.service;
2
3 import lombok.RequiredArgsConstructor;
4 import org.springframework.stereotype.Service;
5 import ru.klgd.javadev.entity.User;
6 import ru.klgd.javadev.repository.UserRepository;
7
8 import java.util.List;
9 import java.util.NoSuchElementException;
10
11 @Service no usages new *
12 @RequiredArgsConstructor
13 public class UserService {
14
15     private final UserRepository repository;
16
17     > public List<User> getUsers() { return repository.findAll(); }
18
19     @ public User updateUser(User user) { no usages new *
20         User entity = repository.findById(user.getId())
21             .orElseThrow(() -> new NoSuchElementException("Пользователь " +
22                 "не найден"));
23
24         return repository.saveAndFlush(entity);
25     }
26
27 }
28
29 }
```

```
1 package ru.klgd.javadev.service;
2
3 import lombok.RequiredArgsConstructor;
4 import org.springframework.stereotype.Service;
5 import ru.klgd.javadev.entity.Order;
6 import ru.klgd.javadev.repository.OrderRepository;
7
8 import java.util.Date;
9 import java.util.List;
10
11 @Service no usages new *
12 @RequiredArgsConstructor
13 public class OrderService {
14
15     private final OrderRepository repository;
16
17     public List<Order> getOrders() { no usages new *
18         return repository.findAll();
19     }
20
21     public Order getOrderById(Long id) { no usages new *
22         return repository.findById(id).orElseThrow(RuntimeException::new);
23     }
24
25     public void deleteAllOrders() { no usages new *
26         repository.deleteAll();
27     }
28 }
```

```
29     public List<Order> getOrdersByDate(Date date) { no usages new *
30         |     return repository.findAllByDate(date);
31     }
32 }
```