МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНЙ УНІВЕРСИТЕТ

Кафедра програмного забезпечення автоматизованих систем

ЗВІТ

про виконання

Лабораторної роботи №4

з дисципліни "Розробка Web-додатків на Java з допомогою Spring фреймворку"

Перевірив роботу: Виконав роботу:

"\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ студент гр. ПЗС-2244

викладач\_\_\_\_\_\_\_\_\_\_\_\_\_ Брус В.М.

Черкаси 2023

Тема: Spring

Мета: Використати бд, реалізувати crud

Завдання:

Завдання до л.р. №4

1. Встановити MySQL або іншу б.д.

2. Підключити java додаток розроблений у попередніх л.р. до бд.

3. Створити таблиці до ваших класів.

4. Створити звязок між вашими класами і таблицями у бд.

5. Реалізувати збереження, отримання, зміну і видалення (CRUD) java обєктів у бд.

6. Реалізувати у вашому додатку відношення між таблицями: один до одного, один до багатьох, багато до багатьох.

7. Реалізувати тип загрузки eager і lazy.

Хід роботи:

1.Створення таблиць в PostgreSQL:

Таблиця bean\_a:

CREATE TABLE IF NOT EXISTS public.bean\_a

(

id bigint NOT NULL DEFAULT nextval('bean\_a\_id\_seq'::regclass),

name character varying(255) COLLATE pg\_catalog."default",

CONSTRAINT bean\_a\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.bean\_a

OWNER to postgres;

Таблиця bean\_b:

CREATE TABLE IF NOT EXISTS public.bean\_b

(

id bigint NOT NULL DEFAULT nextval('bean\_b\_id\_seq'::regclass),

bean\_a\_id bigint,

age integer,

CONSTRAINT bean\_b\_pkey PRIMARY KEY (id),

CONSTRAINT bean\_b\_bean\_a\_id\_fkey FOREIGN KEY (bean\_a\_id)

REFERENCES public.bean\_a (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.bean\_b

OWNER to postgres;

Таблиця bean\_c:

CREATE TABLE IF NOT EXISTS public.bean\_c

(

id bigint NOT NULL DEFAULT nextval('bean\_c\_id\_seq'::regclass),

bean\_b\_id bigint,

code character varying(255) COLLATE pg\_catalog."default",

CONSTRAINT bean\_c\_pkey PRIMARY KEY (id),

CONSTRAINT bean\_c\_bean\_b\_id\_fkey FOREIGN KEY (bean\_b\_id)

REFERENCES public.bean\_b (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.bean\_c

OWNER to postgres;

Таблиця bean\_d:

CREATE TABLE IF NOT EXISTS public.bean\_d

(

id integer NOT NULL DEFAULT nextval('bean\_d\_id\_seq'::regclass),

bean\_a\_id bigint,

CONSTRAINT bean\_d\_pkey PRIMARY KEY (id),

CONSTRAINT bean\_d\_bean\_a\_id\_fkey FOREIGN KEY (bean\_a\_id)

REFERENCES public.bean\_a (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.bean\_d

OWNER to postgres;

2. Додавання залежностей, зміна application.proporties:

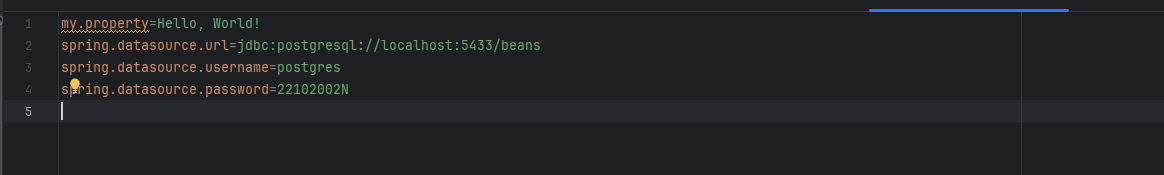


Рисунок 1 - Підключення до бд

3. Створення BeanD, створення репозиторіїв:

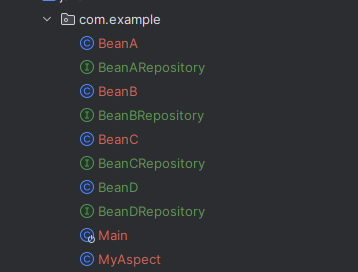


Рисунок 2 - Структура файлів

4. Зміна Main для наявного маніпулювання даними в бд.

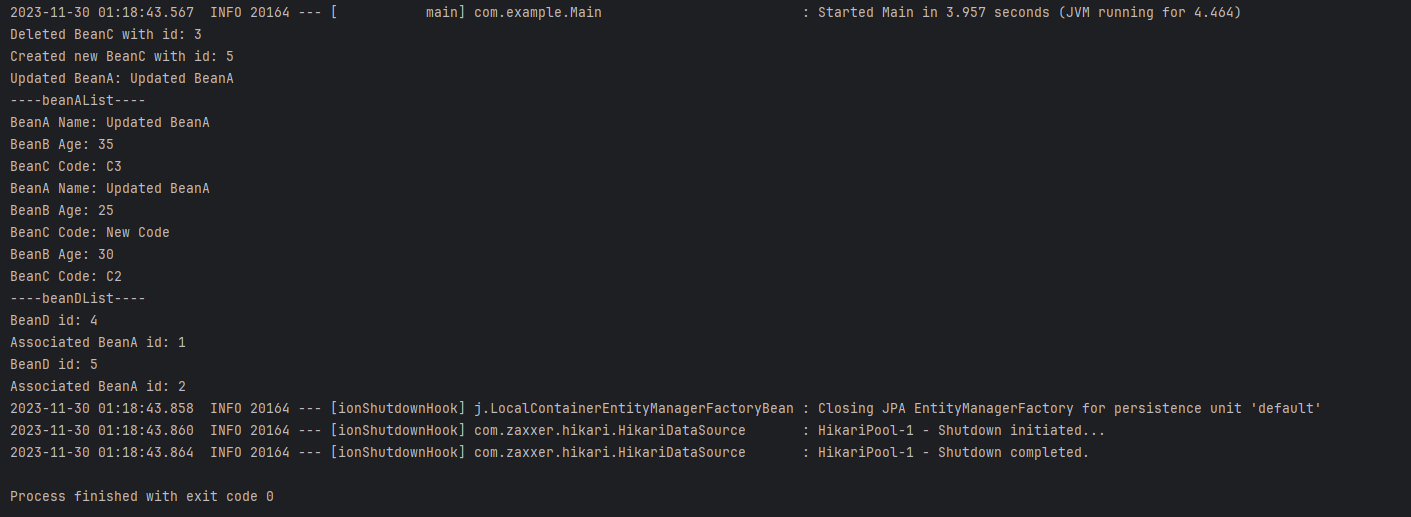


Рисунок 3 - Результат роботи програми

Лістинг

BeanARepository:

package com.example;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface BeanARepository extends JpaRepository<BeanA, Long> {  
//Оскільки репозиторії успадковують JpaRepository, то вони вже мають базові CRUD методи  
}

BeanBRepository:

package com.example;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
@Repository  
public interface BeanBRepository extends JpaRepository<BeanB, Long> {  
 // додаткові методи  
}

BeanCRepository:

package com.example;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface BeanCRepository extends JpaRepository<BeanC, Long> {  
  
}

BeanDRepository:

package com.example;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface BeanDRepository extends JpaRepository<BeanD, Long> {  
}

Application.properties:

my.property=Hello, World!  
spring.datasource.url=jdbc:postgresql://localhost:5433/beans  
spring.datasource.username=postgres  
spring.datasource.password=22102002N

Pom.xml:

<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
  
 <groupId>com.example</groupId>  
 <artifactId>spring-annotation-config-example</artifactId>  
 <version>1.0-SNAPSHOT</version>  
  
 <parent>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-parent</artifactId>  
 <version>2.5.4</version>  
 </parent>  
  
 <dependencies>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.aspectj</groupId>  
 <artifactId>aspectjweaver</artifactId>  
 <version>1.9.7</version>  
 </dependency>  
 <dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-data-jpa</artifactId>  
 </dependency>  
 <dependency>  
 <groupId>org.postgresql</groupId>  
 <artifactId>postgresql</artifactId>  
 </dependency>  
 </dependencies>  
  
 <build>  
 <plugins>  
 <plugin>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-maven-plugin</artifactId>  
 </plugin>  
 </plugins>  
 </build>  
  
</project>

Висновки

В ході цієї роботи було виконано наступне - встановлення базу даних і налаштування з’єднання з нею, використано JPA для створення сутностей та репозиторіїв для роботи з базою даних. Виконано операції створення, читання, оновлення та видалення (CRUD) для сутностей. Створено відносини між сутностями і використано стратегії завантаження EAGER і LAZY