Hibernate zadania

Zadanie 1.

Zainicjalizuj bazę danych H2, którą dołączyłeś do projektu Spring-Boot. Utwórz w tym celu plik schema.sql, który będzie tworzył strukturę bazy (tabelę STUDENTS) oraz plik data.sql, który będzie zawierał instrukcje inicjalizujące tabele w bazie (inserty wstawiające rekordy do tabeli). Umieść pliki w katalogu resources projektu a następnie uruchom aplikację. W przypadku gdy skrypty będą zawierały błąd aplikacja się nie uruchomi.

Zadanie 2.

Zweryfikuj czy tabela i rekordy zostały poprawnie dodane. Skorzystaj w tym celu z H2 console.

a) Dodaj dependency do pom.xml:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-devtools</artifactId>
    <optional>true</optional>
</dependency>
```

b) Wykonaj mvn install w konsoli lub przy pomocy Intellij
c) Zrestartuj aplikację
d) Wejdź na http://localhost:8081/h2-console/ i zaloguj się używając domyślnych namiarów: url: jdbc:h2:mem:testdb, user: sa
e) Zweryfikuj czy tabela została dodana i czy znajdują się w niej odpowiednie wpisy

Zadanie 3.

Zmodyfikuj kod aplikacji tak aby korzystał z bazy danych zamiast z listy studentów przechowywanych w pamięci:

a) Przerób klasę Student tak aby była encją bazodanową
b) Stwórz nowy interfejs StudentRepository, który będzie rozszerzał JpaRepository
c) Zmodyfikuj klasę serwisu tak aby korzystała z nowego interfejsu StudentRepository zamiast sztucznego StudentDao, które bazowało na liście przechowywanej w obiekcie.
d) Dodaj do propertiesów parametr: **spring.jpa.hibernate.ddl-auto=none**
e) Uruchom aplikację i przetestuj jej działanie, użyj do tego konsoli H2 oraz klienta REST'oweg np. Postman pod Chrome.

Zadanie 4.

Dodaj tabelę z danymi adresowymi ADDRESSES i powiąż ją relacją jeden do jednego z tabelą STUDENTS. Następnie odwzoruj to powiązanie po stronie encji Student w Javie. Wykorzystaj adnotację: @OneToOne. Przetestuj czy powiązanie działa poprawnie.

Zadanie 5.

Dodaj tabelę przechowującą dane o książkach dostępnych w bibliotece: BOOKS. Powiąż ją relacją wiele do jeden z tabelą STUDENTS. Następnie odwzoruj to powiązanie po stronie obydwu encji w Javie. Wykorzystaj adnotacje: @OneToMany oraz @ManyToOne Przetestuj czy powiązanie działa poprawnie.

Zadanie 6.

Dodaj tabelę przechowującą dane o dostępnych kursach: COURSES. Powiąż ją relacją wiele do wiele z tabelą STUDENTS. Następnie odwzoruj to powiązanie po stronie encji Student w Javie. Wykorzystaj adnotację: @ManyToMany. Przetestuj czy powiązanie działa poprawnie.

Zadanie 7.

Rozszerz tabelę STUDENTS o dane kontaktowe: numer telefonu i adres email. Następnie zmodyfikuj encję tak aby mapowała dodatkowe pola ale w taki sposób aby dane kontaktowe znajdowały się w osobnej klasie. Będzie wtedy można jej użyć w wielu encjach zamiast powielać w każdej te same pola. Należy użyć adnotacji @Embeddable oraz @Embedded.

Zadanie 8.

Zmień nazwę tabeli STUDENTS na PERSONS, rozszerz ją o dane specyficzne dla nauczyciela oraz dodaj kolumnę ROLE. Następnie po stronie Javy zrealizuj dziedziczenie **Table per class hierarchy** dodając nową encję bazową Person oraz podklasę Teacher. Encja Student również powinna dziedziczyć z encji Person.