

Hibernate Many to Many Övningar

Sammanfattning

Vi har tidigare lärt oss hur man skapar en Many-To-One och One-To-Many-relation med Hibernate. Nu ska vi skapa en Many-To-Many-relation mellan två klasser. Detta kommer att generera tre tabeller:

- En tabell för varje klass
- En tredje tabell som fungerar som en länk mellan de två klasserna

Starta Derby-databasen

Om du har glömt hur du ansluter till Derby-databasen kan du titta på följande länk:

<https://docs.google.com/document/d/1HAzDTrvUANrktA4JHrppm-8HOalLxTHNONL4SssAA/edit?usp=sharing>

Many-To-Many relation i Hibernate

Vi kommer att använda Set istället för List och Map. För att skapa en Many-To-Many relation:

- Lägg till en samling (Set) i båda klasserna.
- Annotera varje samling med @ManyToMany.
- Använd mappedBy på ena sidan av relationen.
- Hibernate skapar automatiskt en länk-tabell för att hantera relationen.

Innehåll i denna övning

- Skapa en Subject-klass.
- Lägga till flera Subject-objekt i databasen.
- Skapa en Many-To-Many-relation mellan Subject och Tutor.
- Implementera metoder för att hantera relationen.
- Testa relationen i huvudmetoden.
- Optimera kod genom att kombinera metoder.

Skapa en Subject klass

- Skapa en ny klass Subject med följande privata attribut:
 - id: int
 - subjectName: String
 - numberOfSemesters: int
- Lägg till nödvändiga Hibernate annoteringar.
- Skapa en konstruktor som tar emot subjectName och numberOfSemesters som parametrar.
- Implementera getter metoder för att göra attributen åtkomliga.
- Lägg till Subject-klassen i hibernate.cfg.xml

Steg 2: Skapa Many-To-Many relationen

- I Subject-klassen:
 - Lägg till en samling Set<Tutor>.
 - Annotera samlingen med @ManyToMany.
 - Initiera samlingen i konstruktorn.
 - Skapa en metod (addTutorToSubject) för att lägga till en tutor till subject.
- I Tutor-klassen:
 - Lägg till en samling Set<Subject>.
 - Annotera samlingen med @ManyToMany(mappedBy="tutors").
 - Initiera samlingen i konstruktorn.
 - Skapa en metod (addSubjectsToTeach) för att lägga till ett subject till tutors lista.

Steg 3: Uppdatera huvudmetoden

- Skapa en Tutor och flera Student-objekt.
- Skapa två Subject-objekt och spara dem i databasen.
- Koppla subject till en tutor och koppla även tutorn till subject.
- Lägg till en ny tutor och koppla ett subject till den, samt koppla subject tillbaka till tutorn.
- Spara alla förändringar i databasen.

Testa och kontrollera databasen

- Öppna ij och kör:
show tables;
*select * from subject_tutor;*
- Kontrollera att rätt relationer har skapats.

Steg 4: Optimera relationen

- Ändra metoden *addSubjectsToTeach* i Tutor klassen så att den automatiskt uppdaterar Subject-klassen.
- Implementera *getTutors()* i Subject klassen för att returnera alla tutors kopplade till ett ämne.
- Ta bort onödig kod från huvudmetoden.
- Kör om koden och verifiera resultaten i databasen.