

Hibernate Many-To-One Övningar

Sammanfattning

I tidigare övningar har vi skapat ett projekt vid namn HibernateProject. I detta projekt har vi en Student-klass.

Med hjälp av Hibernate har vi:

- Skapat en student-tabell i databasen.
- Lagt till, tagit bort, hämtat och uppdaterat data via Java.
- Lärt oss att använda Field Access och Property Access.

Mål med övningen

I denna övning fortsätter vi med samma projekt och fokuserar på att:

- Skapa en Many-To-One-relation mellan Student och Tutor.
- Implementera nödvändiga Hibernate-konfigurationer.

Steg 1: Skapa klassen Tutor

- Skapa en ny klass vid namn **Tutor** med följande attribut:
 - `private String tutorId;`
 - `private String name;`
 - `private int salary;`
- Skapa en konstruktor med alla tre attribut som argumenten och getter-metoder.
- Implementera en `toString()`-metod.
- Lägg till Hibernate-konfigurationer genom att annotera klassen korrekt, och lägga till ett tomt konstruktor- och id attribut.

Steg 2: Modifiera Student-klassen

- Ta bort `@Table(name="TBL_STUDENT")` från Student-klassen.
- Byt ut `tutorName` mot `Tutor tutor` i Student-klassen.

- Uppdatera konstruktorn, getter-metoder och toString().

OBS. Om du får ett kompilatorfel på Tutor, lämna det bara. Vi fixar det mycket snart genom att lägga till anteckningen.

Steg 3: Lägg till Many-To-One-relation

- Annotera tutor fältet i Student-klassen med @ManyToOne.
- Importera jakarta.persistence.ManyToOne
- Skapa metoder i Studentklassen för att:
 - Tilldela en tutor till en student. (*allocateTutor()*)
 - Hämta namnet på studentens tutor. (*getTutorName()*)

Steg 4: Uppdatera Hibernate-konfiguration

- Lägg Tutor i hibernate.cfg.xml.

Steg 5: Testa och spara data

I HibernateTest class:

- Skapa en ny Tutor-instans.
- Skapa en Student och koppla den till en tutor.
- Se till att spara både student och tutor i databasen.

Steg 6: Kontrollera databasen

I ij>:

- Visa och kontrollera att tabellerna har skapats korrekt.
- Finns det någon kolumn tillagd i student tabellen?
- Hur är den här kolumnen relaterad till tutor tabellen?

Steg 7: Hämta data från databasen

- Hämta en student från databasen (med hjälp av `session.get` metoden) och kontrollera att den har en kopplad tutor.
- Hämta tutors namn för en given student.

Steg 8: Uppdatera en students tutor

- Hämta en befintlig tutor från databasen (med hjälp av `session.get()`)
- Koppla en ny student till en befintlig tutor.
- Spara ändringen i databasen.