Hibernate Advanced Mapping

In DB

1 Multiple Tables

2 Relationship btw tables

Need to model this with Hibernate

Types

1 one to one

Ex : An instructor can have instructor details , Similar instrctur profile

Instructor(T)-> Instructor Details(T)

2 one to many , many to one

Ex: One to many

instructor can have many courses inverse of this is many to one relation many courses can have single Instrctor

3 many to many

Ex : A course an have many student and student can have many courses

Database concepts

1 primary key and Foreign key

-primary key : identify unique row in a table

-Foreign key : Link table together -> a field in one table refer to primary key in another table

2 cascade

Apply same operation to related entites

If I save Instrctor it will cascade and save instructor\_details also

If we save Instructor it performs same operation to Instructor\_details

If we delete Instructor should delete their instructor details also bcz they no longer have record known as “**CASCADE DELETE**”

We have to be carefull with cascade delete in terms Many to many we should not delete check use case

Fetch Types: Eager VS Lazy Loading => “should we retrieve everything”

Eager will retrieve everything

Lazy will retrieve on request

Uni – Directional Relationship

One way relation using Instrctor get instructor details

Instructor-> Instrctor\_details

Bi –directional

Both ways

Instructor<--> Instrctor\_details

ONE TO ONE

1st uni – directional example

Deveolpment process one to one

1 Define database tables

2 create instructor\_details class

3 create instructor class

4 create main App

Table : instructor\_details

create table instructor\_detail(id serial PRIMARY KEY , youtube\_channel VARCHAR(200) DEFAULT NULL,hobby VARCHAR(100) DEFAULT NULL);

Table : instructor

CREATE TABLE instructor (id serial PRIMARY KEY,first\_name varchar(45) DEFAULT NULL,last\_name varchar(45) DEFAULT NULL,email varchar(45) DEFAULT NULL,

instructor\_detail\_id int DEFAULT null, CONSTRAINT fk\_detail FOREIGN KEY (instructor\_detail\_id) REFERENCES instructor\_detail(id)

)

Need to set relation to the table instructor\_detail\_id

Link the tbale using FK

Definf FK

CONSTRAINT fk\_detail FOREIGN KEY (instructor\_detail\_id) REFERENCES instructor\_detail(id)

Forieng key

Preserve relatiosnship btween tbles

Referential Integrity

Prevents the operations that would destroy relationship

Ensure only valid data is inserted into the foreign key column

Can contain only valid reference to the primary key another table

Refactor and Exception handling

javax.persistenceNullPoniterException: Null Pointer

Bi instructorDetail null

java.lang.NullPointerException

at com.mapping.onetoonedemo.DemoExceptionHandle.main(DemoExceptionHandle.java:29)

javax.persistence.NoResultException: No entity found for query

Cascade Delete

Delete InstrutorDetail along with Instructor

Delete only InstrutorDetail keep Instructor

-Modify Cascade on InstructorDetails Entity

In InstructorDetails in cascade type select Excpet REMOVE

@OneToOne(mappedBy="instructorDetail",cascade= {CascadeType.***DETACH***,CascadeType.***MERGE***,CascadeType.***PERSIST***,CascadeType.***REFRESH***,CascadeType.***REMOVE***})

Perform same delete operation of previous one and c changes

javax.persistence.EntityNotFoundException: deleted object would be re-saved by cascade (remove deleted object from associations): [com.mapping.entity.InstructorDetail#3]

need to remove bi directional in Main app code

// remove the assoicated object reference

// break bi directionl link

instructorDetail.getInstructor().setInstructorDetail(**null**);

session.delete(instructorDetail);

OneToMany

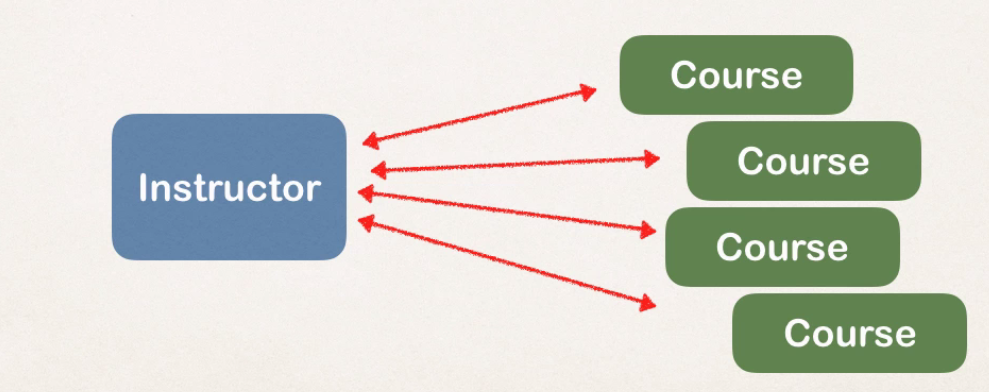
Instructor can have many cource

* Bi directional

Many to One

Many cource can have one intrsrctor

Inverse/ oppsite of one to many



Real project

If delete instructor don’t delete course

If you delete course don’t delete instructor

Means Do not apply cascade DELETE

Development Process

1 define tables

2 create cource class

3 update instructor class

4 main app

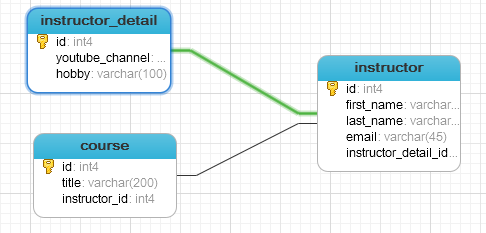
Create Couse table

create table course (id serial PRIMARY KEY , title VARCHAR(200) ,instructor\_id int,

CONSTRAINT FK\_INSTRUCTOR\_MANY FOREIGN KEY (instructor\_id) REFERENCES instructormany(id)

)

Add unique key to title to prevent duplicate



Create Course Class

@Id

@GeneratedValue(strategy = GenerationType.***IDENTITY***)

@Column(name="id")

**private** **int** id;

@Column(name="title")

**private** String title;

// many course has one instructor

// cascade is imp dnt delete so check cascade delete

@ManyToOne(cascade = {CascadeType.***PERSIST***,CascadeType.***DETACH***,CascadeType.***MERGE***,CascadeType.***REFRESH***})

@JoinColumn(name="instructor\_id") // instructor\_id is kay in Course table which points if of Instructor table

**private** Instructor instructor;

Update instructor class