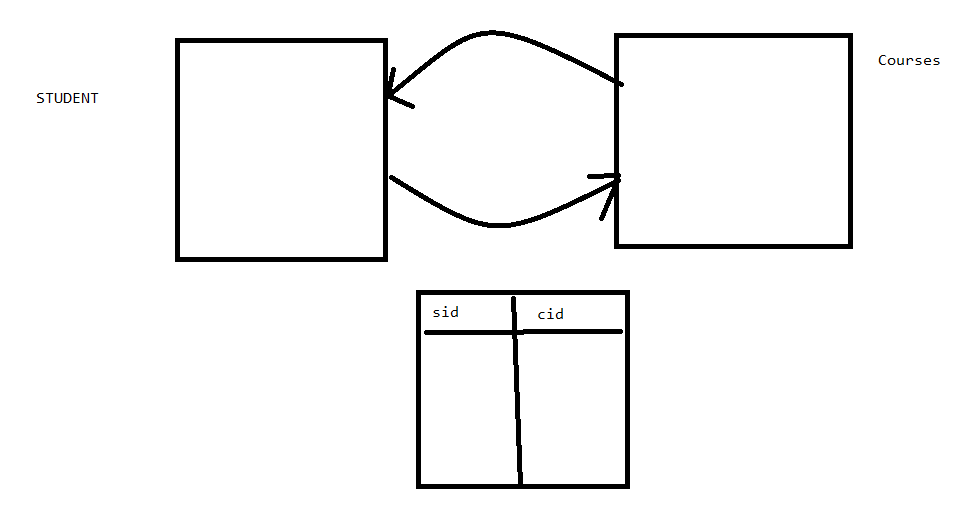
MANY TO MANY:

EX: A STUDENT COULD HAVE MANY COURSES.

ONE COURSE COULD HAVE MANY STUDENTS.

So a third table is created with s\_id and c\_id



COURSE CLASS:

@Data

@Table(name = "course\_info")

@Entity

public class Course {

@Id

@Column

private int cid;

@Column

private String name;

@Exclude

@ManyToMany(cascade = CascadeType.ALL , mappedBy = "courses")

private List<Student> students;

}

STUDENT CLASS:

@Data

@Table(name = "student\_info")

@Entity

public class Student {

@Id

@Column

private int sid;

@Column

private String name;

@Exclude

@ManyToMany(cascade = CascadeType.ALL)

@JoinTable( name = "student\_course\_info" , joinColumns = @JoinColumn(name="sid") ,

inverseJoinColumns = @JoinColumn(name="cid"))

private List<Course> courses;

}

MANYTOMANY TEST:

public class ManyToManyTest {

public static void main(String[] args) {

EntityManagerFactory entityManagerFactory = null;

EntityManager entityManager = null;

EntityTransaction entityTransaction = null;

Student student = new Student();

student.setSid(10);

student.setName("ABC");

Student student2 = new Student();

student2.setSid(20);

student2.setName("ZBC");

Course course = new Course();

course.setCid(60);

course.setName("JAVA");

Course course2 = new Course();

course2.setCid(70);

course2.setName("Hibernate");

ArrayList<Student> students = new ArrayList<Student>();

students.add(student);

students.add(student2);

course.setStudents(students);

course2.setStudents(students);

ArrayList<Course> courses = new ArrayList<Course>();

courses.add(course2);

courses.add(course);

student.setCourses(courses);

student2.setCourses(courses);

try {

entityManagerFactory = Persistence.createEntityManagerFactory("mapping");

entityManager = entityManagerFactory.createEntityManager();

entityTransaction = entityManager.getTransaction();

entityTransaction.begin();

entityManager.persist(student);

entityTransaction.commit();

//TO GET THE COURSE DETAILS BY FINDING STUDENT

// Student student3= entityManager.find(Student.class, 10);

// System.out.println(student3.getCourses().get(0).getName());

// System.out.println(student3.getCourses().get(1));

//TO GET THE STUDENT DETAILS BY FINDING COURSE

// Course course3 = entityManager.find(Course.class, 70);

// System.out.println(course3.getStudents().get(0));

// System.out.println(course3.getStudents().get(1));

} catch (Exception e) {

e.printStackTrace();

}

}

}