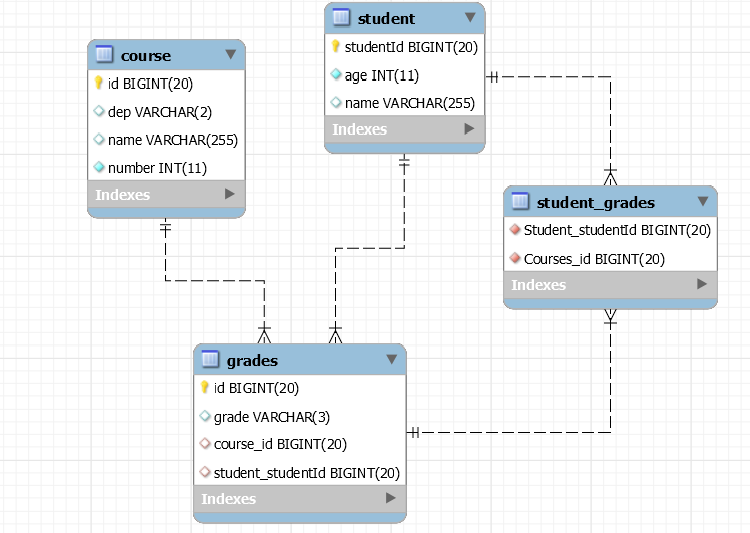
Quiz4: Quiz

CS544 Enterprise Architecture

|  |  |  |
| --- | --- | --- |
| @Entity  public class Student {      @Id      @GeneratedValue      @Column(name="studentId")      private Long id;      private String name;      private int age;      @OneToMany      private List<Enrollment> Courses;  } | @Entity  @Table(name="Grades")  public class Enrollment {      @Id      @GeneratedValue      private Long id;      @Column(length = 3)      private String grade;      @ManyToOne      private Student student;      @ManyToOne      private Course course;  } | @Entity  public class Course {      @Id      @GeneratedValue      private Long id;      private String name;      @Column(name="dep", length = 2)      private String department;      private int number;      @OneToMany(mappedBy = "course")      private List<Enrollment> students;  } |

1. Give the table names, column names and data types that the following annotated classes map to (answer as best as you can). Please also answer the two theory questions on the back of this page.



1. Describe what mappedBy does on @OneToMany, @OneToOne, and @ManyToMany

mappedBy gives up control of the bi-directional association to the other side Says the the other side is the owner of the association

1. Describe what kind of problems you may have if you try to create a bi-directional association with a real list (using @OrderColumn) on both sides of the bi-directional association:

Race condition. The reason for this is that both side needs to be the owning side in order to insert its @OrderColumn... But both sides cannot be the owner at the same time. Either 2 uni-directional associations. Or if you force them to use the same join table and columns then we won't know who wins (race condition).s