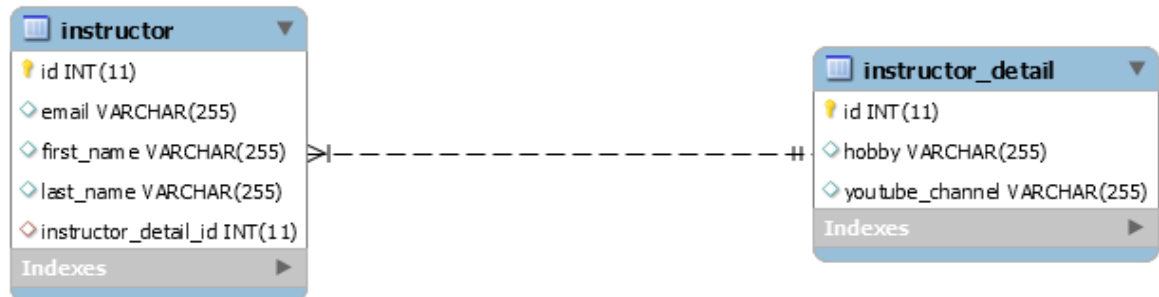


JPA/Hibernate One to One Mapping Annotation Example



```
public class Instructor {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "first_name")  
    private String firstName;  
  
    @Column(name = "last_name")  
    private String lastName;  
  
    @Column(name = "email")  
    private String email;  
  
    @OneToOne(cascade = CascadeType.ALL)  
    @JoinColumn(name = "instructor_detail_id")  
    private InstructorDetail instructorDetail;  
}
```

```
public class InstructorDetail {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "youtube_channel")  
    private String youtubeChannel;  
  
    @Column(name = "hobby")  
    private String hobby;  
}
```

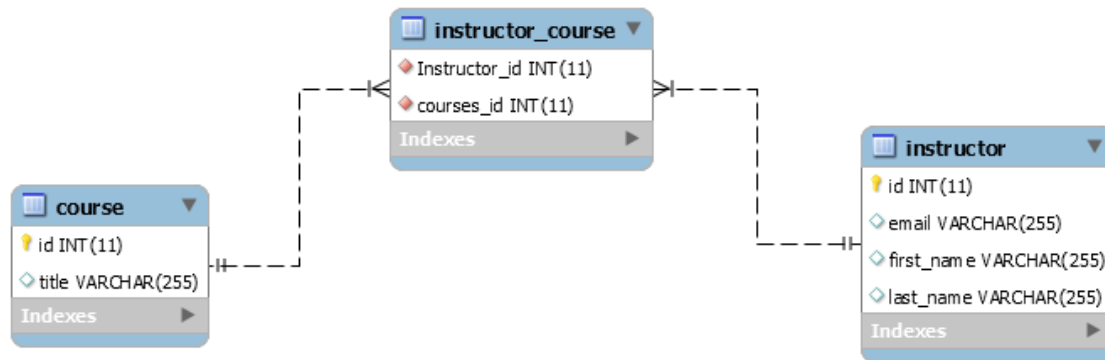
JPA/Hibernate One to One Bidirectional Mapping Annotation Example



```
public class Instructor {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "first_name")  
    private String firstName;  
  
    @Column(name = "last_name")  
    private String lastName;  
  
    @Column(name = "email")  
    private String email;  
  
    @OneToOne(cascade = CascadeType.ALL, mappedBy = "instructor", fetch =  
FetchType.LAZY)  
    private InstructorDetail instructorDetail
```

```
public class InstructorDetail {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "youtube_channel")  
    private String youtubeChannel;  
  
    @Column(name = "hobby")  
    private String hobby;  
  
    @OneToOne(fetch = FetchType.LAZY)  
    @JoinColumn(name = "instructor_id")  
    private Instructor instructor;
```

JPA / Hibernate One to Many Unidirectional Mapping Example



```
public class Instructor {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "first_name")  
    private String firstName;  
  
    @Column(name = "last_name")  
    private String lastName;  
  
    @Column(name = "email")  
    private String email;  
  
    @OneToMany(cascade = CascadeType.ALL)  
    private List < Course > courses = new ArrayList < Course > ();  
}
```

```
public class Course {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "title")  
    private String title;  
}
```

JPA/Hibernate One to Many Bidirectional Mapping Example



```
public class Instructor {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "first_name")  
    private String firstName;  
  
    @Column(name = "last_name")  
    private String lastName;  
  
    @Column(name = "email")  
    private String email;  
  
    @OneToMany(mappedBy = "instructor", cascade = { CascadeType.PERSIST,  
        CascadeType.MERGE, CascadeType.DETACH,  
        CascadeType.REFRESH })  
    private List<Course> courses;  
}
```

```
public class Course {  
  
    @Id  
    @GeneratedValue(strategy = GenerationType.IDENTITY)  
    @Column(name = "id")  
    private int id;  
  
    @Column(name = "title")  
    private String title;  
  
    @ManyToOne(cascade = CascadeType.ALL)  
    @JoinColumn(name = "instructor_id")  
    private Instructor instructor;  
}
```

Hibernate Many to Many Annotation Mapping Example



```
public class Employee {

    private static final long serialVersionUID = 1L;

    @Id
    @Column(name = "employee_id")
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long employeeId;

    @Column(name = "first_name")
    private String firstName;

    @Column(name = "last_name")
    private String lastName;

    @ManyToMany(cascade = {
        CascadeType.ALL
    })
    @JoinTable(
        name = "employees_projects",
        joinColumns = {
            @JoinColumn(name = "employee_id")
        },
        inverseJoinColumns = {
            @JoinColumn(name = "project_id")
        }
    )
    Set < Project > projects = new HashSet < Project > ();
}
```

```
public class Project {  
  
    private static final long serialVersionUID = 1L;  
  
    @Id  
    @Column(name = "project_id")  
    @GeneratedValue  
    private Long projectId;  
  
    @Column(name = "title")  
    private String title;  
  
    @ManyToMany(mappedBy = "projects", cascade = { CascadeType.ALL })  
    private Set<Employee> employees = new HashSet<Employee>();  
}
```