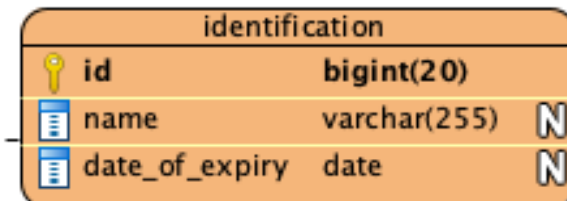
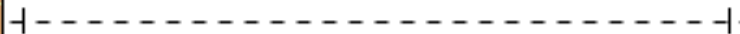
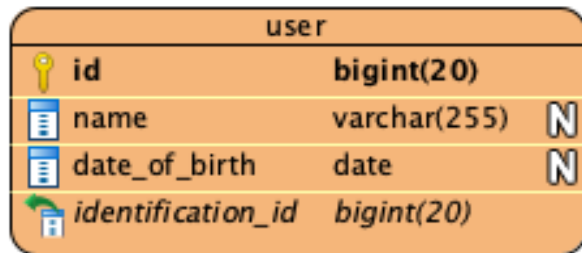


# Session 04: Spring Data JPA (cont)

# Learning goals

- OneToOne
- OneToMany
- ManyToMany
- CascadeType
- FetchType

# OneToOne



- Using @ManyToOne and @JoinColumn

```
@Entity
@Table(name = "user")
public class User {

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

    //..

    @ManyToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "identification_id", referencedColumnName = "id")
    private Identification identification;

    //getters/setters

}
```

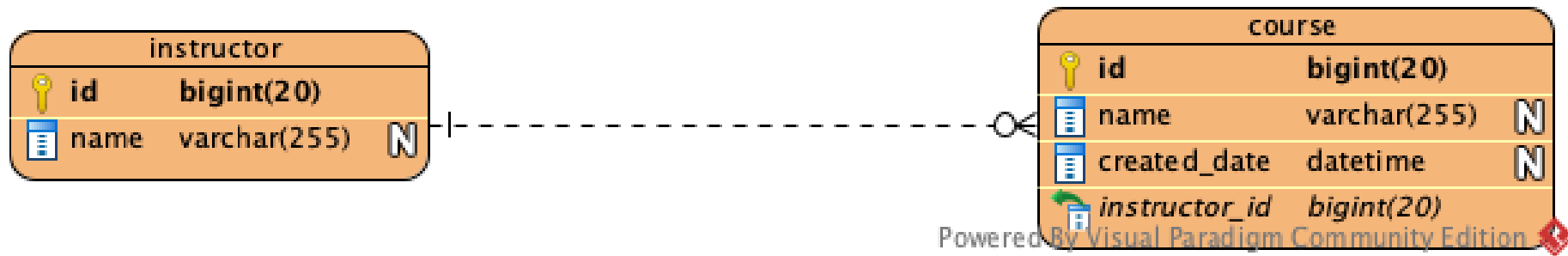
- Using `@OneToOne` with `mappedBy`

```
@Entity
@Table(name = "identification")
public class Identification {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;

    //...
    @OneToOne(mappedBy = "identification")
    private User user;

    //getters/setters
}
```

# OneToMany



# OneToMany

- Using **@OneToMany** annotation

```
@Entity
@Table(name = "instructor")
public class Instructor {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;

    //
    @OneToMany(mappedBy = "instructor")
    private Set<Course> courses;
    //getters/setters
}
```

# OneToMany

- Using `@ManyToOne` and `@JoinColumn`

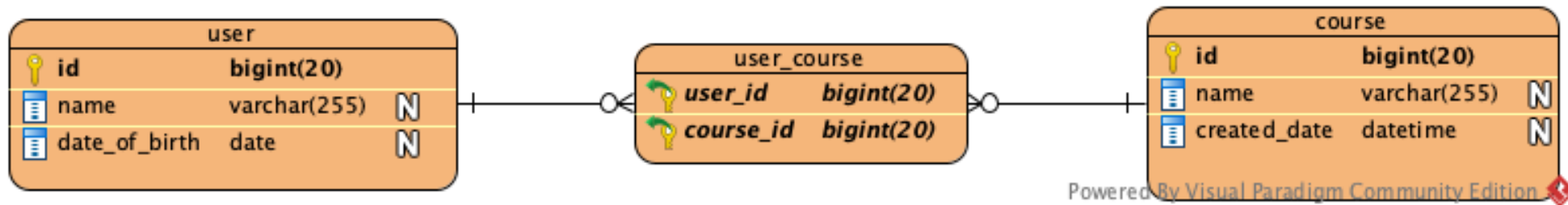
```
@Entity
@Table(name = "course")
public class Course {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;

    //
    @ManyToOne
    @JoinColumn(name = "instructor_id", nullable = false)
    private Instructor instructor;

    //getters/setters
}
```



# ManyToMany



# ManyToMany

- Using **@ManyToMany** and **@JoinTable**

```
@Entity
@Table(name = "course")
public class Course {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private Long id;

    //
    @ManyToMany
    @JoinTable(name = "enrollment",
        joinColumns = @JoinColumn(name = "course_id"),
        inverseJoinColumns = @JoinColumn(name = "user_id"))
    private Set<User> users = new HashSet<>();

    //getters/setters
}
```

# ManyToMany

- Using **@ManyToMany** with **mappedBy**

```
@Entity
@Table(name = "user")
public class User {

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

    //..

    @ManyToMany(mappedBy = "users")
    private Set<Course> courses = new HashSet<>();

    //getters/setters
}
```

CASCADE	MÔT TẢ
ALL	Tương ứng với tất cả các loại cascade. <code>cascade={DETACH, MERGE, PERSIST, REFRESH, REMOVE}</code>
DETACH	Nếu đối tượng cha bị detached khỏi persistence context thì các đối tượng tham chiếu tới nó cũng bị detached.
MERGE	Nếu đối tượng cha được merged vào persistence context, thì các đối tượng tham chiếu tới nó cũng được merged.
PERSIST	Nếu đối tượng cha được persisted vào persistence context, thì các đối tượng tham chiếu tới nó cũng được persisted.
REFRESH	Nếu đối tượng cha được refreshed ở persistence context hiện tại, thì các đối tượng tham chiếu tới nó cũng được refreshed.
REMOVE	Nếu đối tượng cha bị removed khỏi persistence context, thì các đối tượng tham chiếu tới nó cũng được removed.

# CascadeType

```
@Entity
@Table(name = "user")
public class User {

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

    //..

    @OneToOne(cascade = CascadeType.ALL)
    private Identification identification;

}
```

- The **EAGER** strategy is a requirement on the persistence provider runtime that data must be eagerly fetched
- The **LAZY** strategy is a hint to the persistence provider runtime that data should be fetched lazily when it is first accessed
- Default FetchType
  - OneToMany: LAZY
  - ManyToOne: EAGER
  - ManyToMany: LAZY
  - OneToOne: EAGER

# FetchType

```
@Entity
@Table(name = "user")
public class User {

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

    //..

    @OneToOne(cascade = CascadeType.ALL, fetch = FetchType.LAZY)
    private Identification identification;

}
```

*Keeping up those **inspiration** and the **enthusiasm** in the **learning path**.  
Let confidence to bring it into **your career path** for getting gain the **success**  
as your expectation.*

**Thank you**

#### **Contact**

- Name: R2S Academy
- Email: [daotao@r2s.edu.vn](mailto:daotao@r2s.edu.vn)
- Hotline/Zalo: 0919 365 363
- Website: <https://r2s.edu.vn>
- Fanpage: <https://www.facebook.com/r2s.tuyendung>

#### **Questions and Answers**