Lecture 15: One-to-Many and Many-to-One Mapping

Theory

When you want to avoid creating junction tables for certain relationships, you can use OneToMany and ManyToOne annotations directly.

Problem Statement

If we don't want the third table (alien_laptop), we can establish direct relationships.

Implementation

Alien Entity (One Side):

```java
@Entity
public class Alien {
    @Id
    private int aid;
    private String aname;
    private String tech;

    @OneToMany(mappedBy = "alien")
    private List<Laptop> laptops;

}
```

Laptop Entity (Many Side):

```java
@Entity
public class Laptop {
    @Id
    private int lid;
    private String brand;
    private String model;
    private int ram;

    @ManyToOne
    private Alien alien;

}
```

Key Concepts

- **@OneToMany**: One alien can have multiple laptops
- **@ManyToOne**: Many laptops can belong to one alien
- **mappedBy**: Specifies the owning side of the relationship
- **No Junction Table**: Direct foreign key relationship

Important Note

The `get()` method has been **deprecated in Hibernate 7.1.0** but not removed.