Lecture 14: One-to-One Mapping Implementation

Theory

OneToOne mapping in Hibernate uses JPA annotations to establish bidirectional relationships between entities. This eliminates the need for manual foreign key management.

Code Implementation

Entity Classes

Laptop Entity:

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

    // Getters and setters
}
```

Alien Entity with OneToOne:

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

    @OneToOne
    private Laptop laptop;

    // Getters and setters
}
```

Session Management Example:

```
public class Main {
    public static void main(String[] args) {
        // Create entities
        Laptop l1 = new Laptop();
        l1.setLid(1);
        l1.setBrand("Asus");
        l1.setModel("Rog");
        l1.setRam(16);

        Alien a1 = new Alien();
        a1.setAid(101);
        a1.setAname("Navin");
        a1.setTech("Java");
        a1.setLaptop(l1);

        // Session operations
        Session session = sf.openSession();
        Transaction transaction = session.beginTransaction();

        session.persist(a1);
        session.persist(l1);

        transaction.commit();
        session.close();
    }
}
```

Configuration Setup:

```
SessionFactory sf = new Configuration()
    .addAnnotatedClass(com.telusko.Alien.class)
    .addAnnotatedClass(com.telusko.Laptop.class)
    .configure()
    .buildSessionFactory();
```

Key Features

- **@OneToOne**: Establishes one-to-one relationship
- **Automatic FK Management**: Hibernate manages foreign keys
- **Bidirectional Support**: Can navigate from both entities