



CS544 EA

Hibernate

Complex: Composite Keys

Composite Keys

- Composite keys are **multi-column primary keys**
 - By definition natural keys, set by the app
 - Also create **multi-column foreign keys**
 - Generally found in legacy systems



Composite Id

- @Embeddable can create a composite PK
 - Using @EmbeddedId annotation

```
@Embeddable
public class Name {
    private String firstName;
    private String lastName;
    ...
}
```

Requires
hashCode() and equals()
Java check for uniqueness

```
@Entity
public class Employee {
    @EmbeddedId
    private Name name;
    @Temporal(TemporalType.DATE)
    private Date startDate;
    ...
}
```

@EmbeddedID



.hashCode() & .equals()

```
@Override
public int hashCode() {
    final int prime = 31;
    int result = 1;
    result = prime * result + ((firstName == null) ? 0 : firstName.hashCode());
    result = prime * result + ((lastName == null) ? 0 : lastName.hashCode());
    return result;
}
@Override
public boolean equals(Object obj) {
    if (this == obj)
        return true;
    if (obj == null)
        return false;
    if (getClass() != obj.getClass())
        return false;
    Name other = (Name) obj;
    if (firstName == null) {
        if (other.firstName != null)
            return false;
    } else if (!firstName.equals(other.firstName))
        return false;
    if (lastName == null) {
        if (other.lastName != null)
            return false;
    } else if (!lastName.equals(other.lastName))
        return false;
    return true;
}
```

Critical for checking
uniqueness

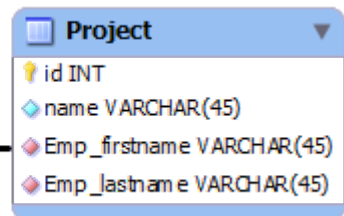
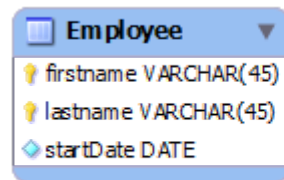
IDE can
generate these
for you

Composite FK

- **Multiple columns** to refer to composite PK

```
@Entity
public class Employee {
    @EmbeddedId
    private Name name;
    @Temporal(TemporalType.DATE)
    private Date startDate;
    @OneToMany(mappedBy="owner")
    private List<Project> projects
        = new ArrayList<>();
    ...
}
```

Normal mappedBy



```
@Entity
public class Project {
    @Id
    @GeneratedValue
    private Long id;
    private String name;
    @ManyToOne
    @JoinColumns({
        @JoinColumn(name = "Emp_firstname", referencedColumnName = "firstname"),
        @JoinColumn(name = "Emp_lastname", referencedColumnName = "lastname")
    })
    private Employee owner;
    ...
}
```

Optional @JoinColumns

Defaults to:
owner_firstname
owner_lastname