

# Modeling Entity Relationships



**Cătălin Tudose**

Java Champion, PhD in Computer Science, Java and Web Technologies Expert

[www.catalintudose.com](http://www.catalintudose.com) | [www.linkedin.com/in/catalin-tudose-847667a1](https://www.linkedin.com/in/catalin-tudose-847667a1)

# Overview

**The types of relationships**

**The directions of the relationships**

**Annotations used for relationships**

**Embedded classes**

**Collections of embedded classes**



# Types of Relationships

**One-to-many**

**Many-to-one**

**Many-to-many**

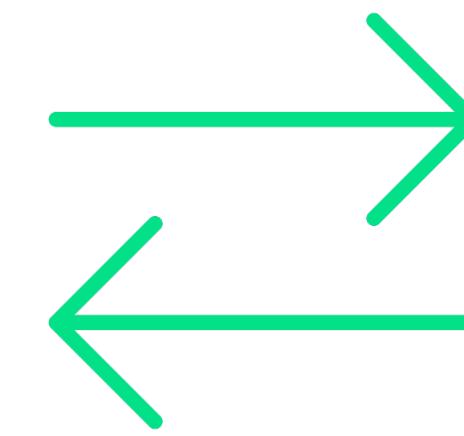
**One-to-one**



# Direction of the Relationship



Unidirectional



Bidirectional



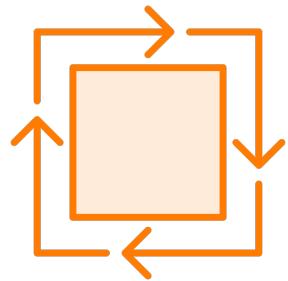
# Owning Side



**Any relationship has an owning side**



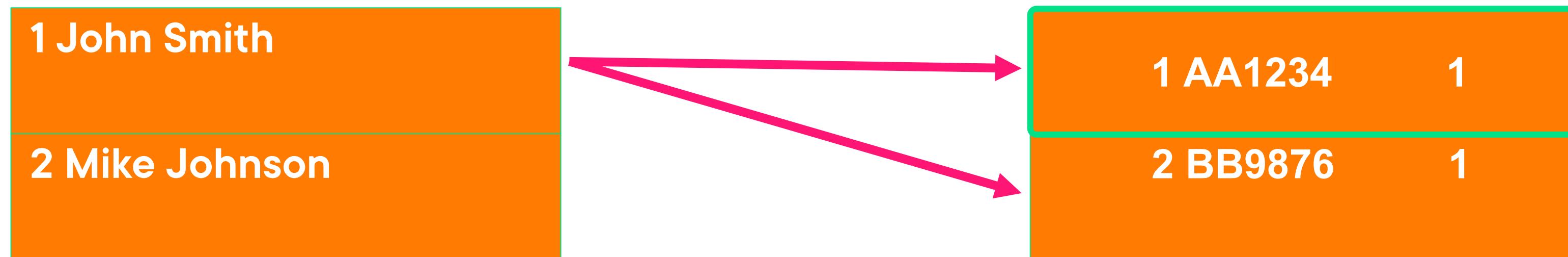
**Owning side drives the updates to relationship in a database**



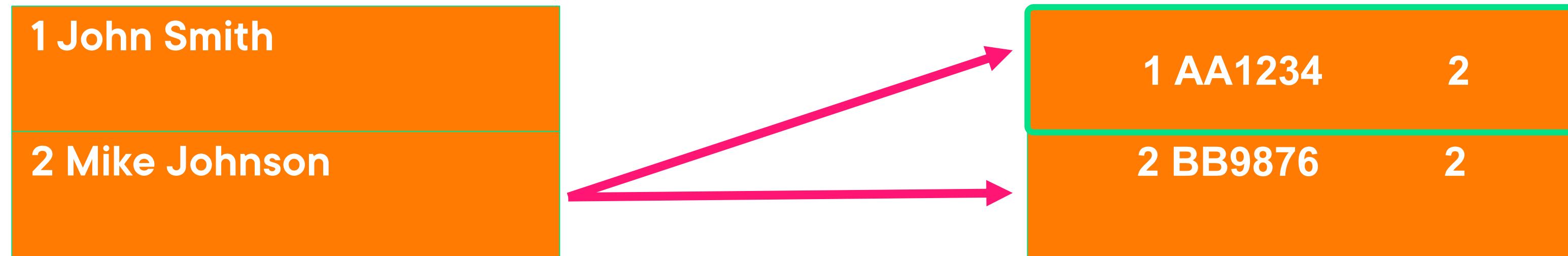
**The many part is always the owning side**



# Changing the Relationship



# Changing the Relationship



# Annotations to Relationships

**@OneToMany**

**@ManyToOne**

**@OneToOne**

**@ManyToMany**



# Demo

**Define one-to-many relationships**

**Define many-to-one relationships**



# Demo

**Define many-to-many relationships**



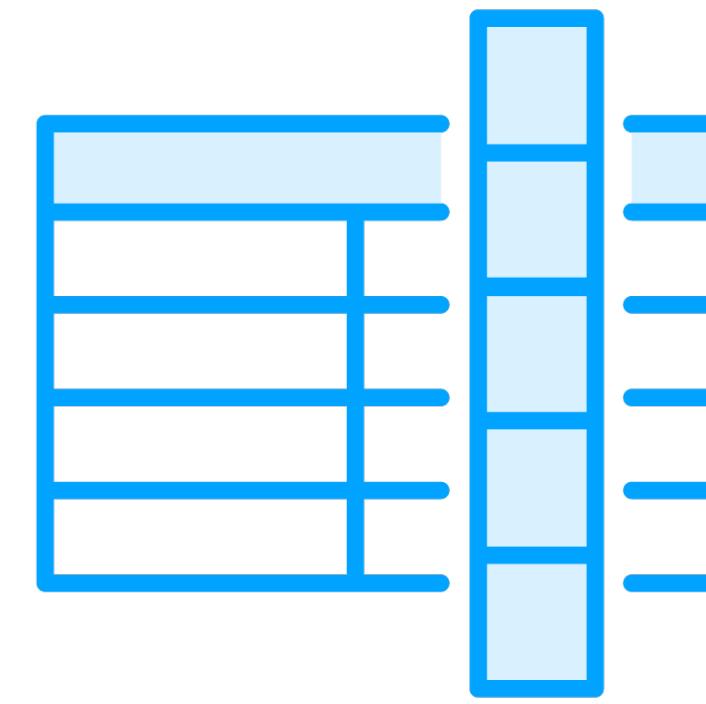
# Annotations for Relationship Definition



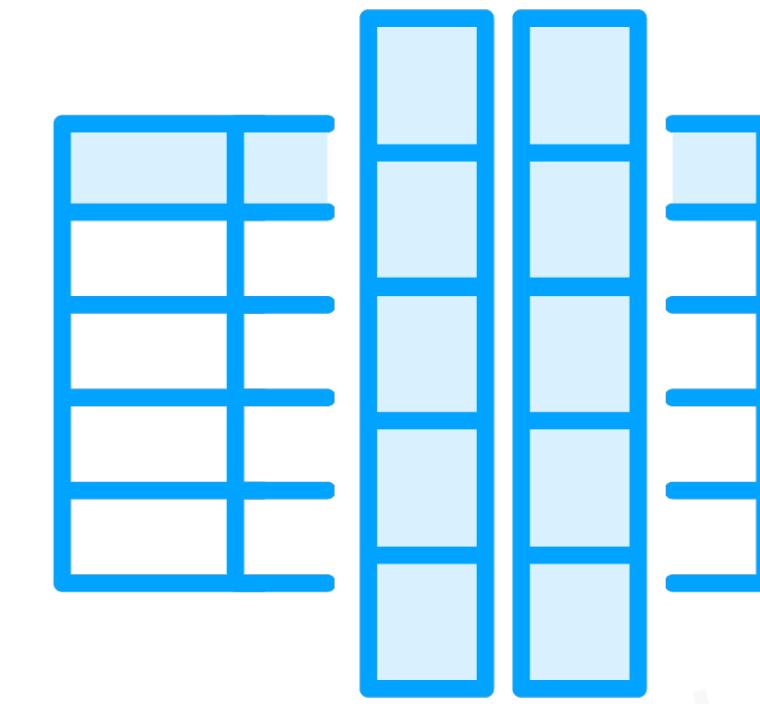
# Annotations for Relationship Definition



**@JoinTable**



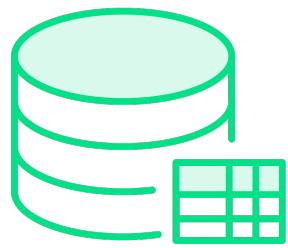
**@JoinColumn**



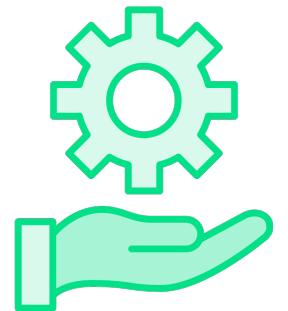
**@JoinColumns**



# @JoinTable



**Cross-reference table for the mapping of relationship**



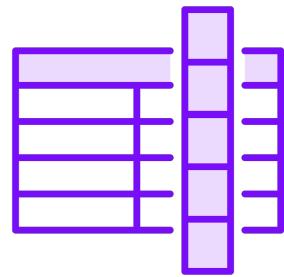
**On the owning side of relationship**



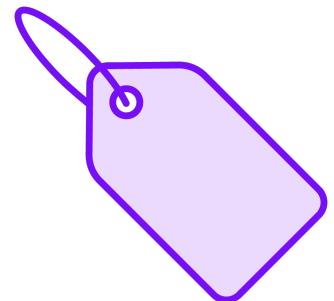
**Parameters**



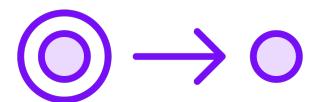
# @JoinColumn



**Specifies a column for joining an entity association**



**Name parameter**



**referencedColumnName parameter**



# @JoinColumns



**Mapping for composite foreign keys**



**Groups @JoinColumn annotations**

K	V

**Value parameter**



# Demo

**Join tables on one column**



# Demo

**Join tables on multiple columns**



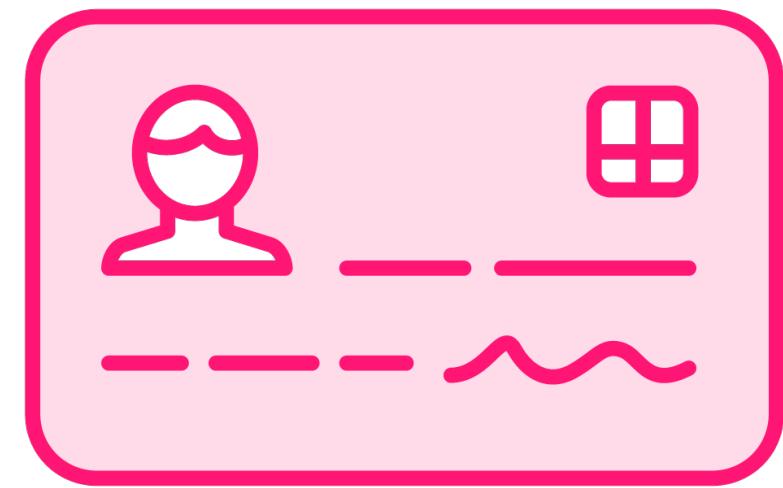
# Embeddable Classes



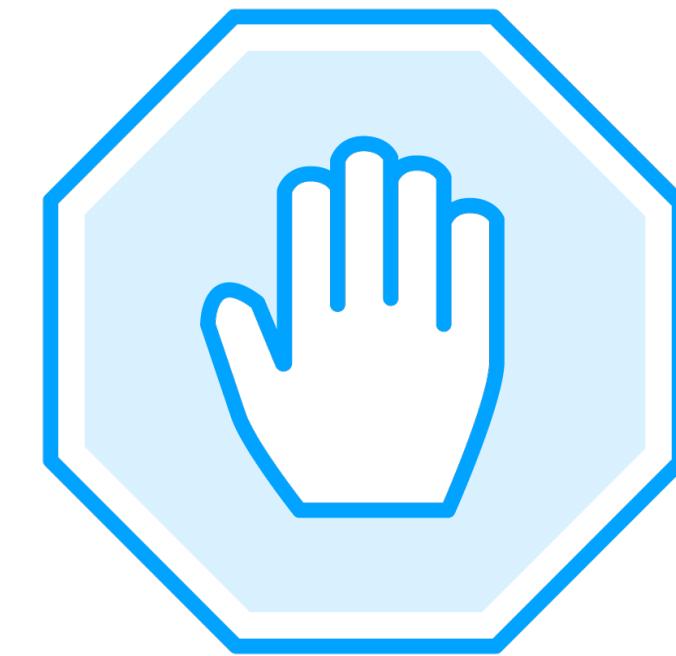
# Embeddable Classes



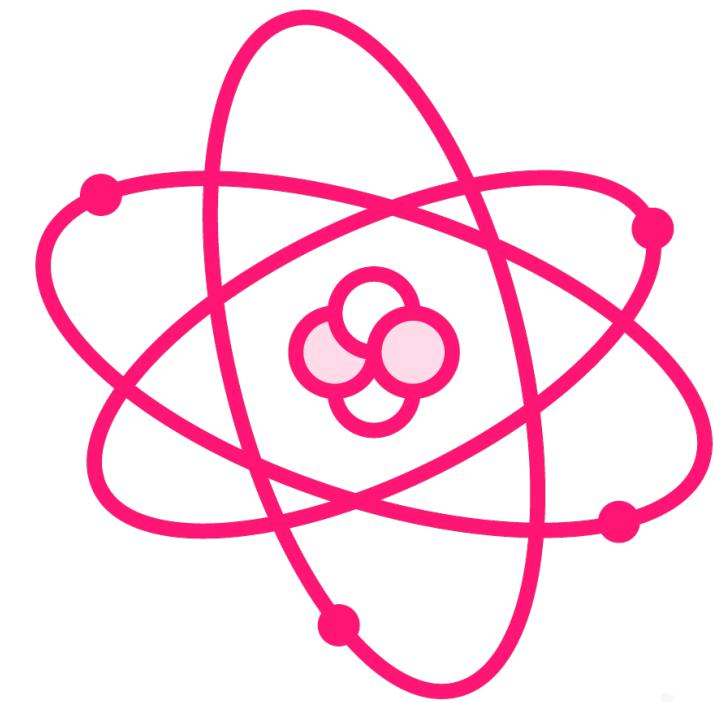
**Fine-grained  
classes  
representing  
entity state**



**Do not have  
persistent  
identity of their  
own**



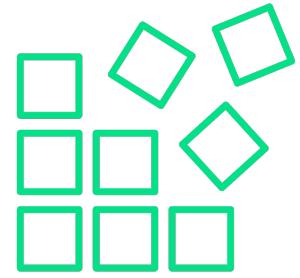
**Cannot be  
shared across  
persistent  
entities**



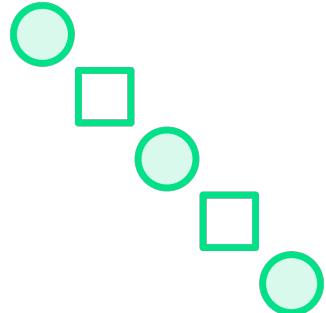
**Only as part of  
the state of the  
entity to which  
they belong**



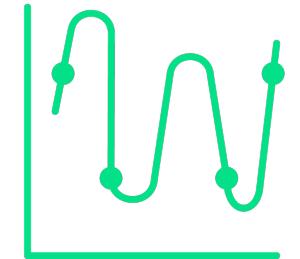
# Embeddable Classes Annotations



**@Embeddable**



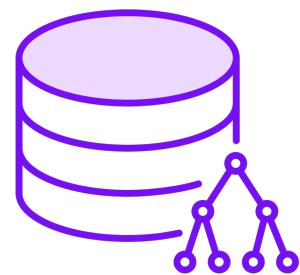
**@AttributeOverride**



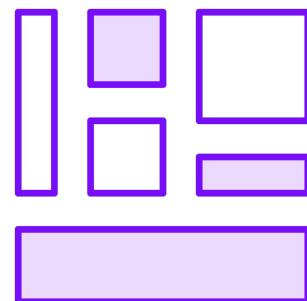
**@AttributeOverrides**



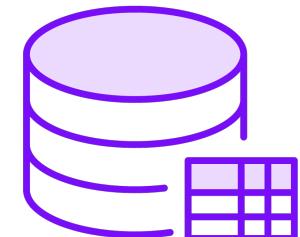
# Collections of Embeddable Classes



**Collections of basic types or embeddable classes**



**@ElementCollection**



**@CollectionTable**



# Maps of Embeddable Classes

K	V

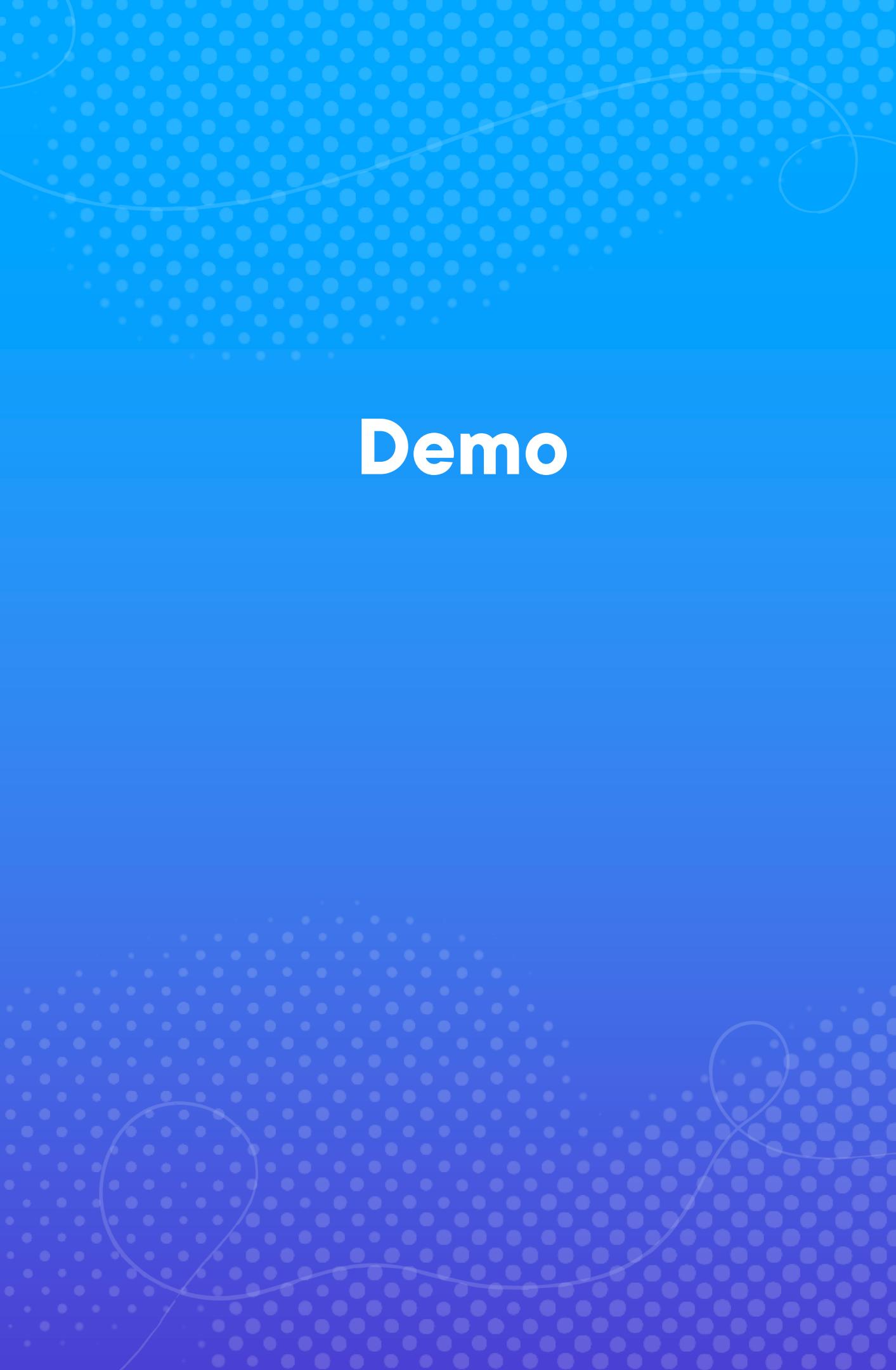
**Key and value: either a basic type or an embeddable class**



**@MapKeyColumn**


**@Column**





# Demo

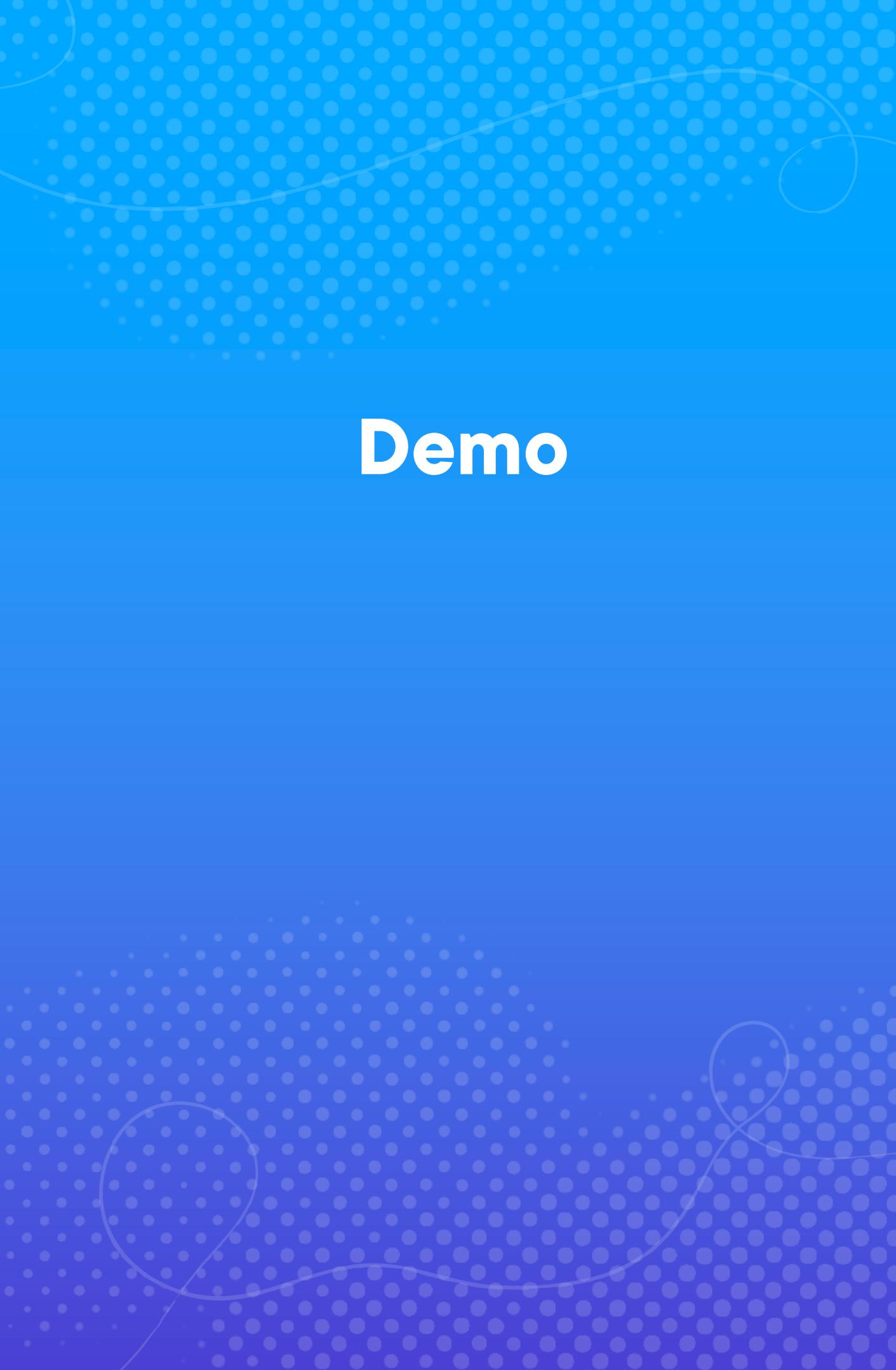
## Embedding classes in entities



# Demo

**Embedding collections of classes in entities**





# Demo

## Embedding maps of classes in entities



# Summary

**Types and directions of relationships**

**Annotations to define relationships**

**Embedded classes**

**Collections of embedded classes**

**Maps of embedded classes**

