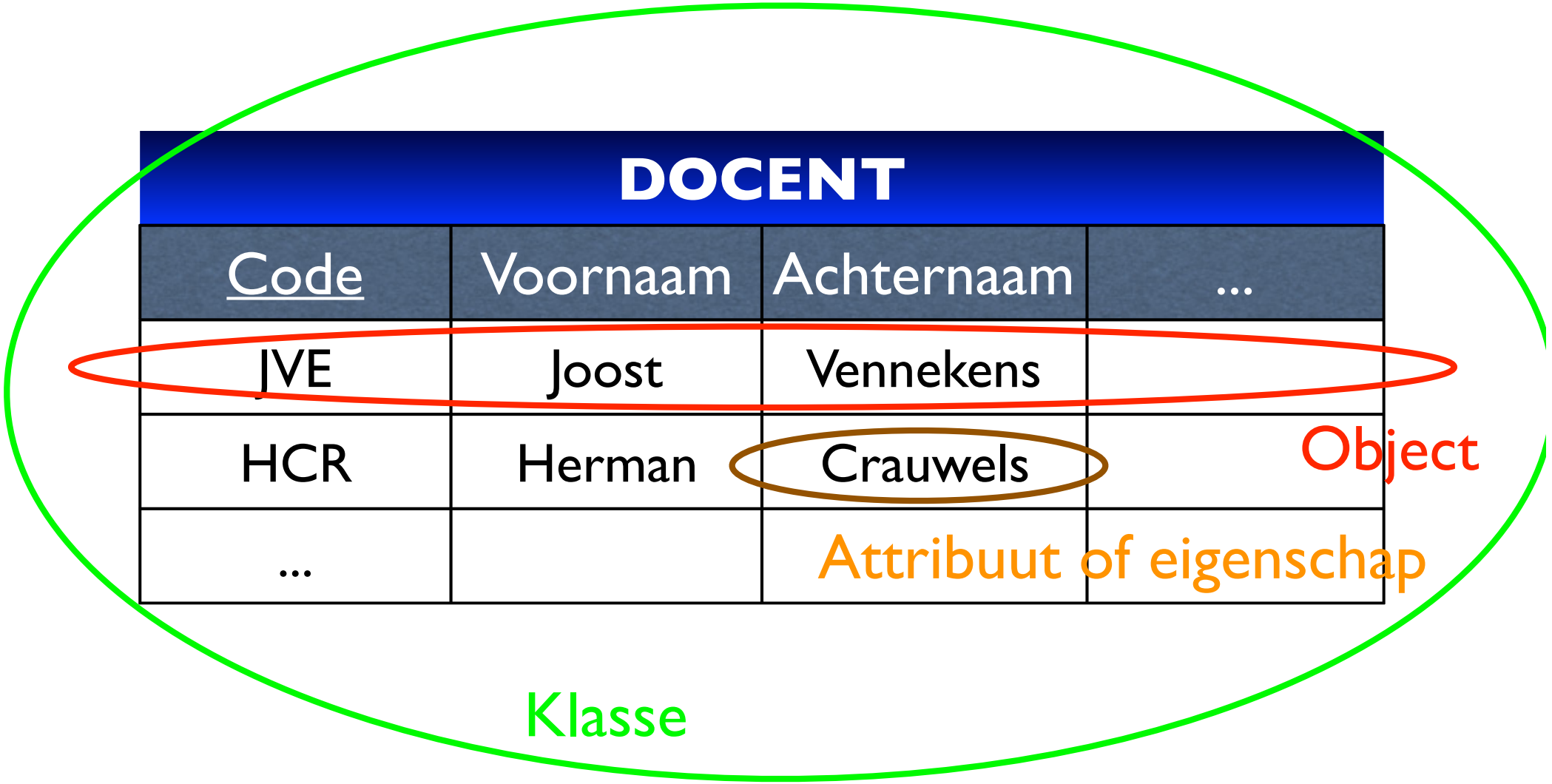
Java Persistence API

Relationele DB

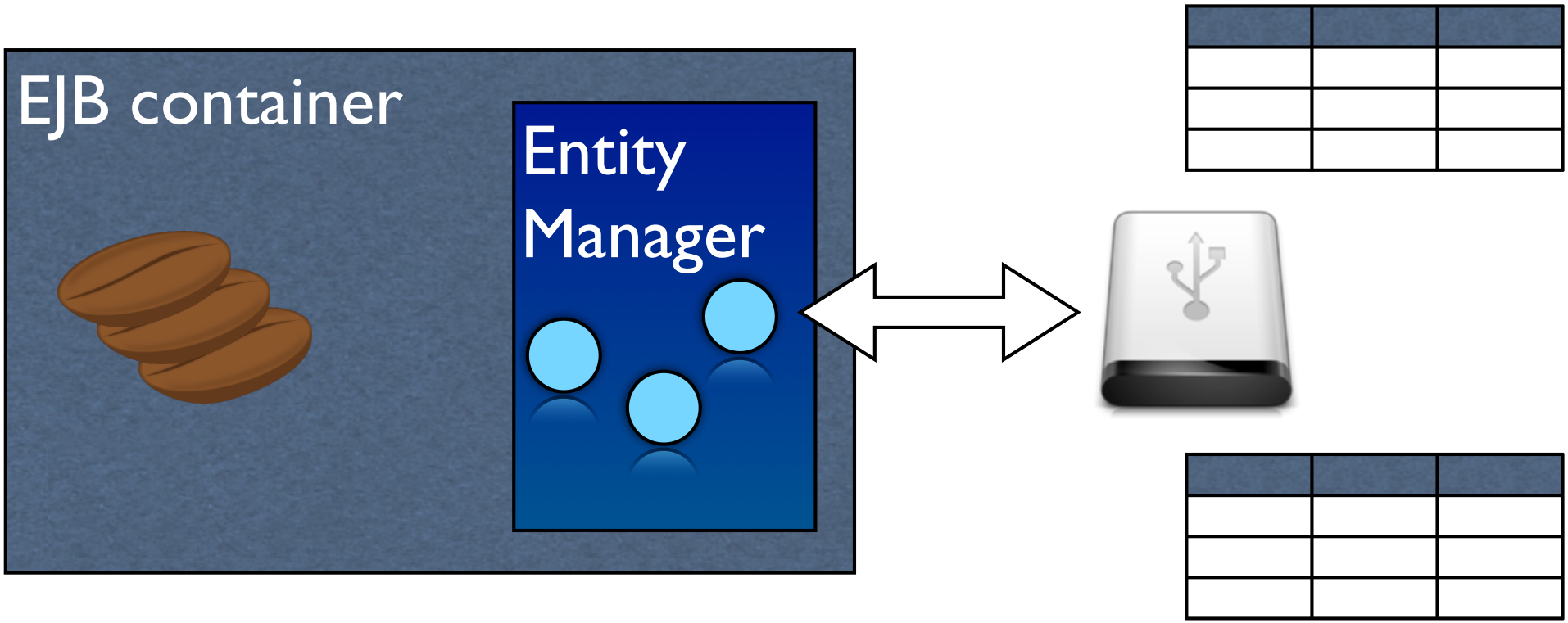
DOCENT			
Code	Voornaam	Achternaam	...
JVE	Joost	Vennekens	
HCR	Herman	Crauwels	
...			

Relationale DB



```
public class Docent {
    private String code;
    private String voornaam;
    private String achternaam;
    public String getVoornaam() { ... }
    public void setVoornaam(String v) { ... }
    public Docent(String code) { ... }
... }

Docent joost = new Docent("JVE");
```



Entity instances

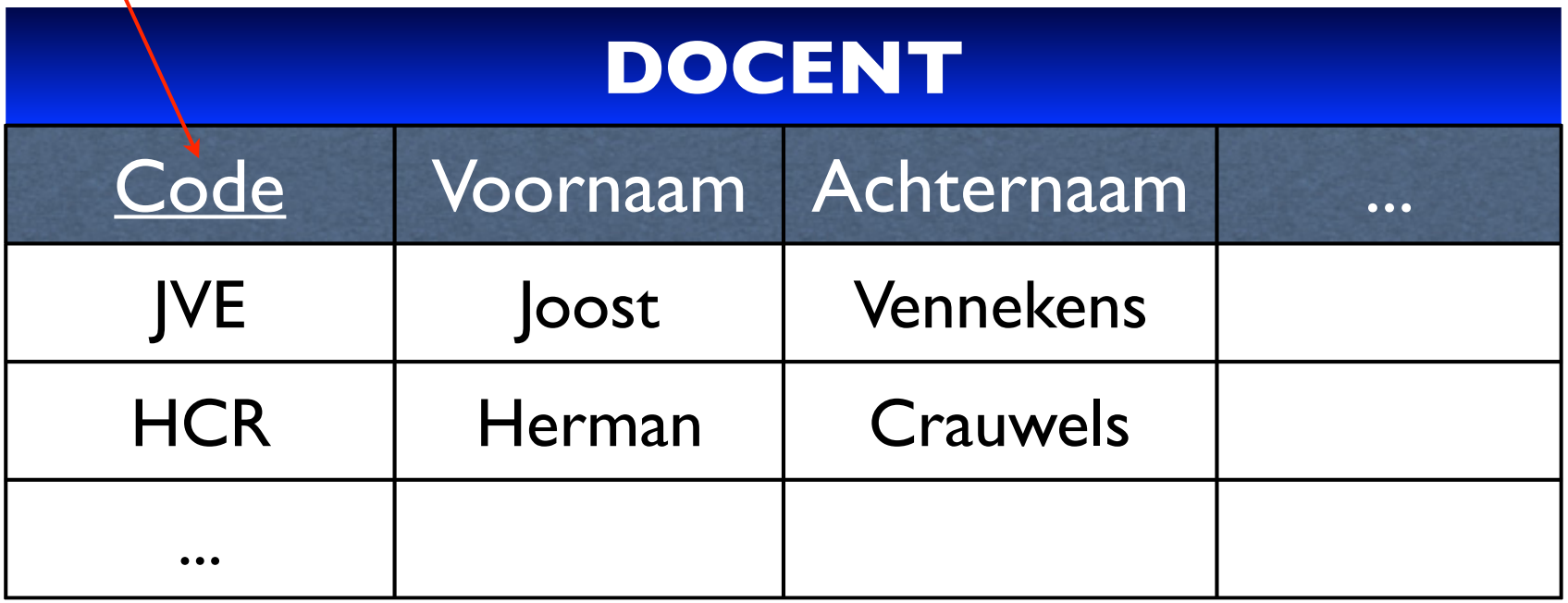
@Entity

public class Docent {

private String code;

@Id

@Table



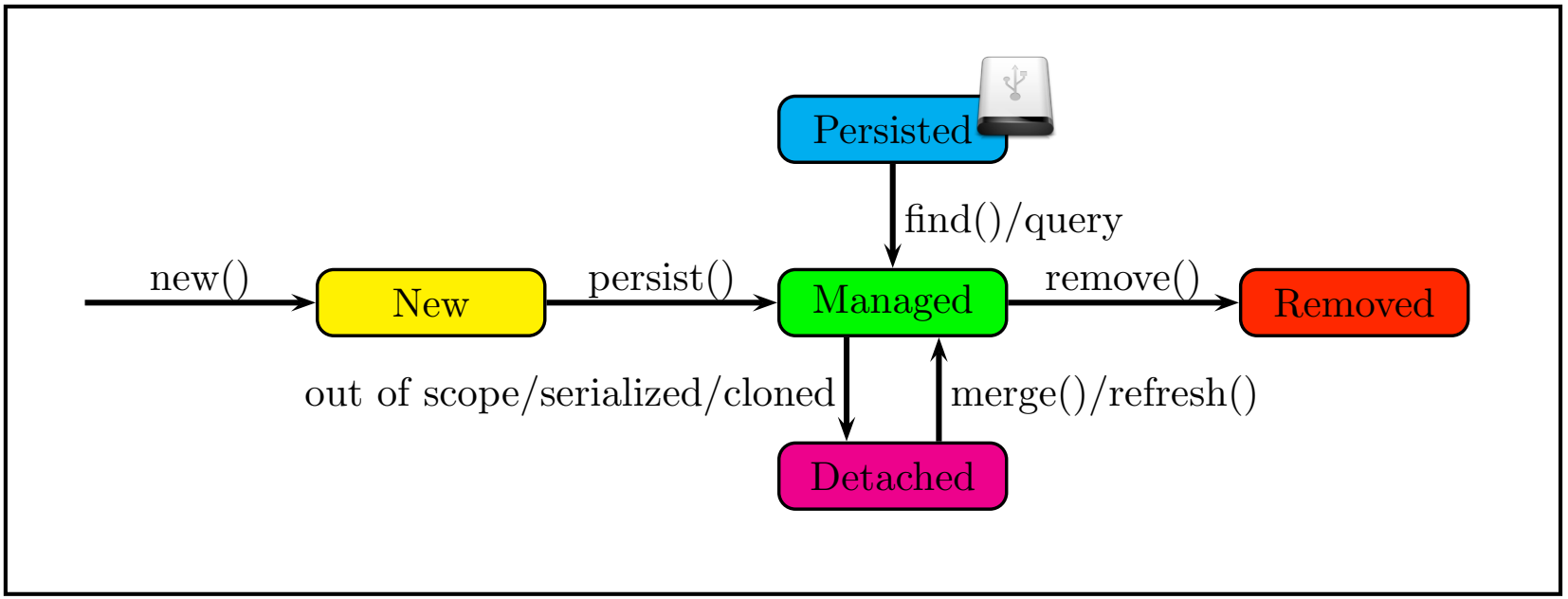
@Column

private String achternaam;

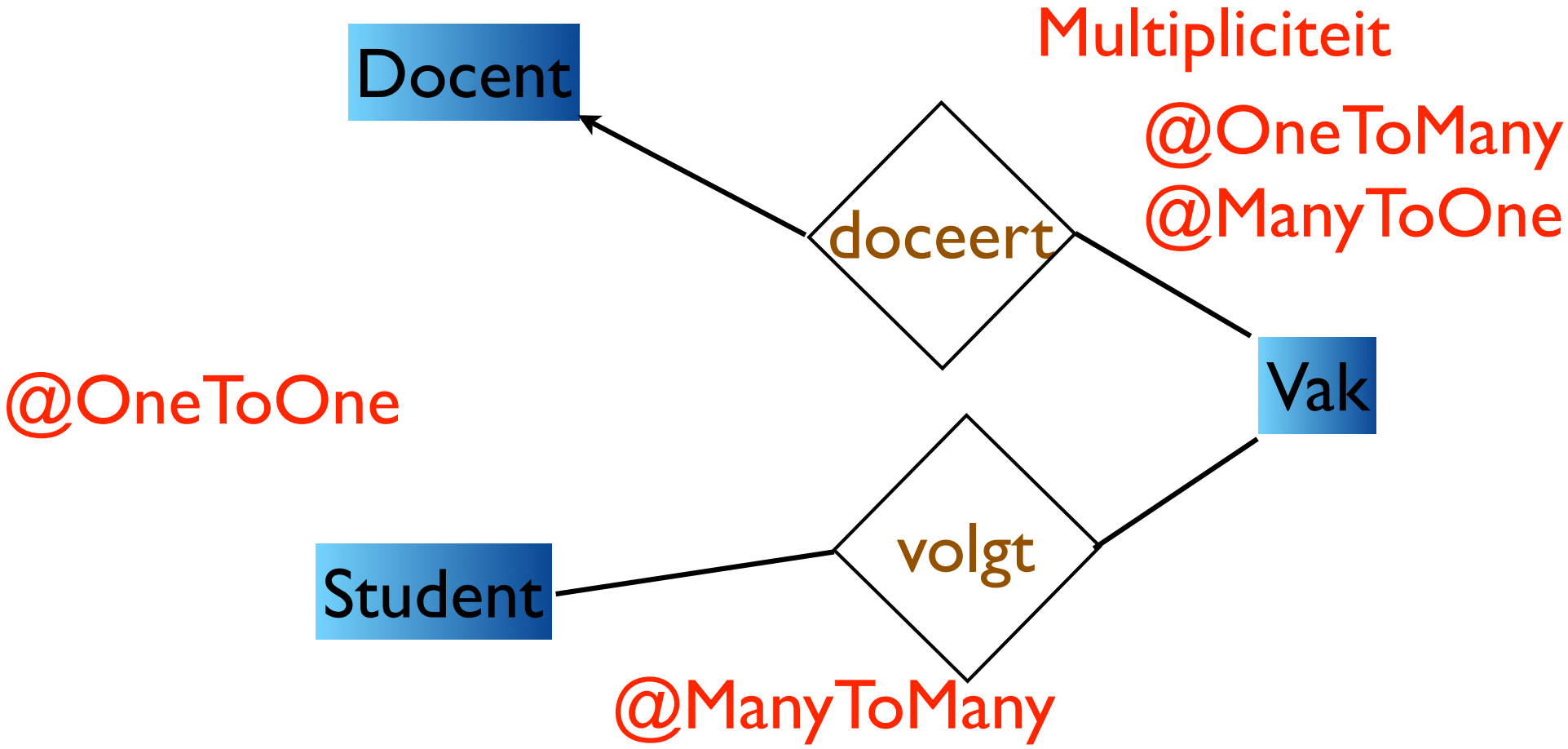
Entity klasse

- Geannoteerd met @Entity
- Default constructor
- Niet final
- Geen public variabelen
- Implementeert Serializable

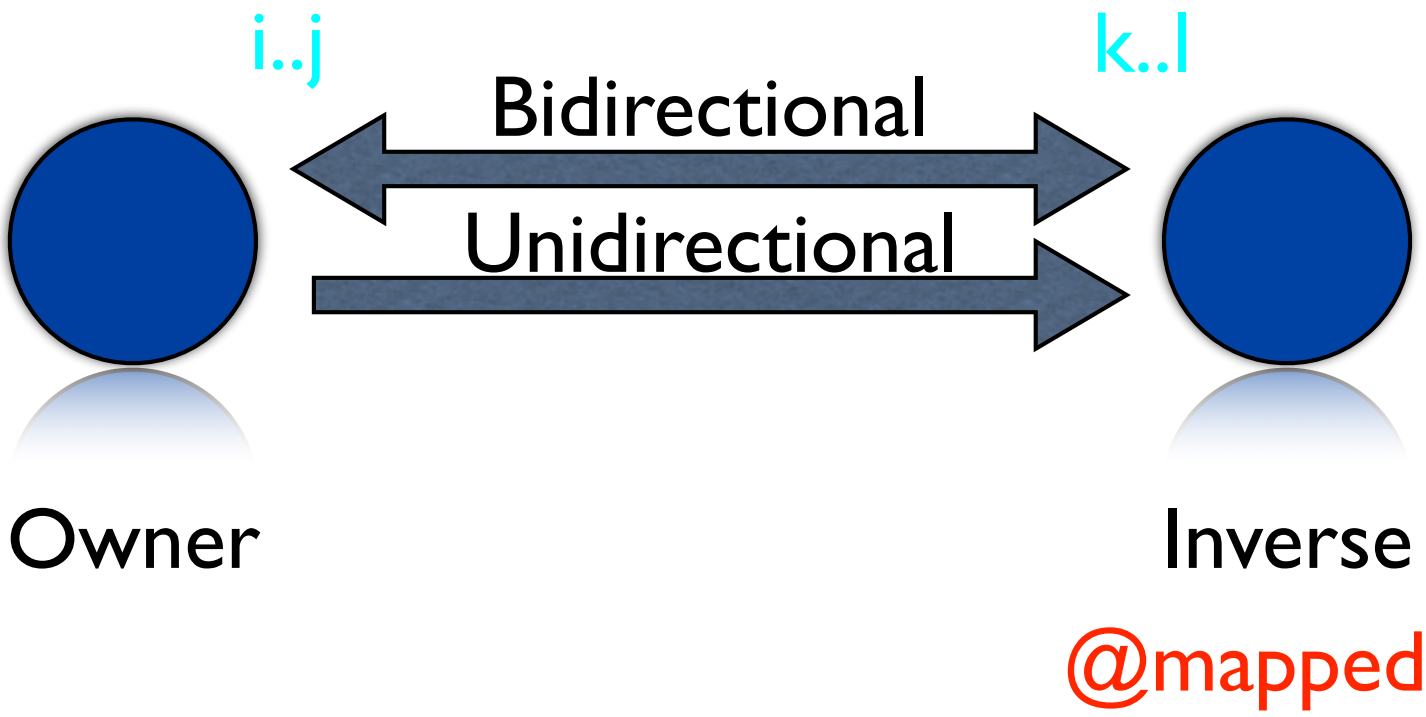
Levenscyclus



ER model


Multipliciteit
Docent
@OneToMany
@ManyToOne
doceert
Vak
@OneToOne
volgt
Student
@ManyToMany

In Java

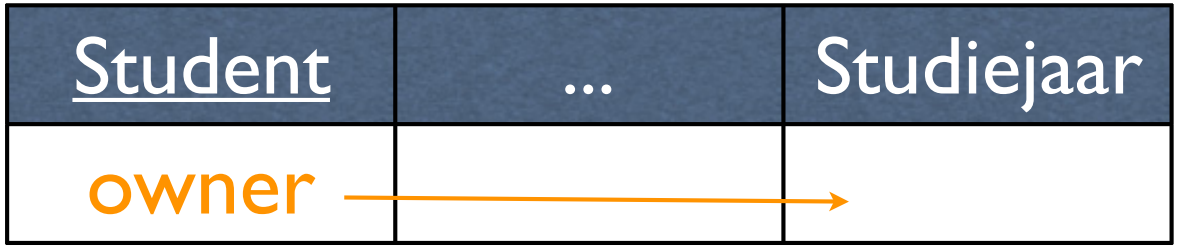

i..j
k..l
Bidirectional
Unidirectional
Owner
Inverse
@mappedBy

In de DB

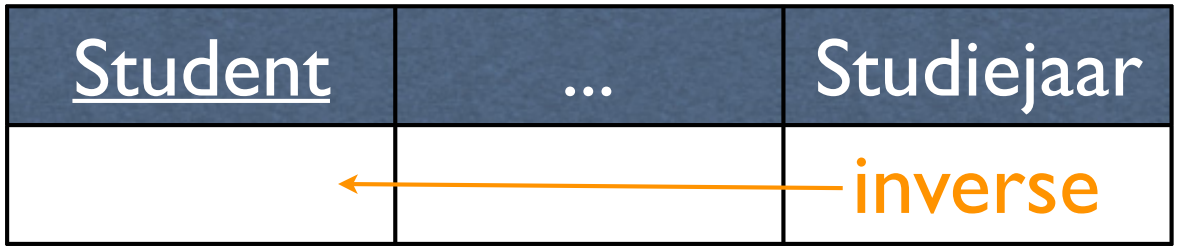
- OneToOne
  @JoinColumn



- ManyToOne
  @JoinColumn



- OneToMany
  @JoinColumn



- ManyToMany
  @JoinTable



Transacties

= ondeelbare verandering aan DB

```
@Resource
private javax.transaction. UserTransaction trans;

public void doStuff(...) {
  ...
  trans.begin();
  ...
  trans.commit();
  ...
  }
```

} Transactie

trans.rollback()

Andere topics

- EJB Query Language
- Caching (persistence.xml)
- Connection Pools en Resources
- Auto-generated entities