



ORM overerving

Veerle Ongenae



Object ↔ Tabel ??

- Klasse → tabel
 - Vaak, maar niet altijd!
 - Soms anders o.w.v. performantie

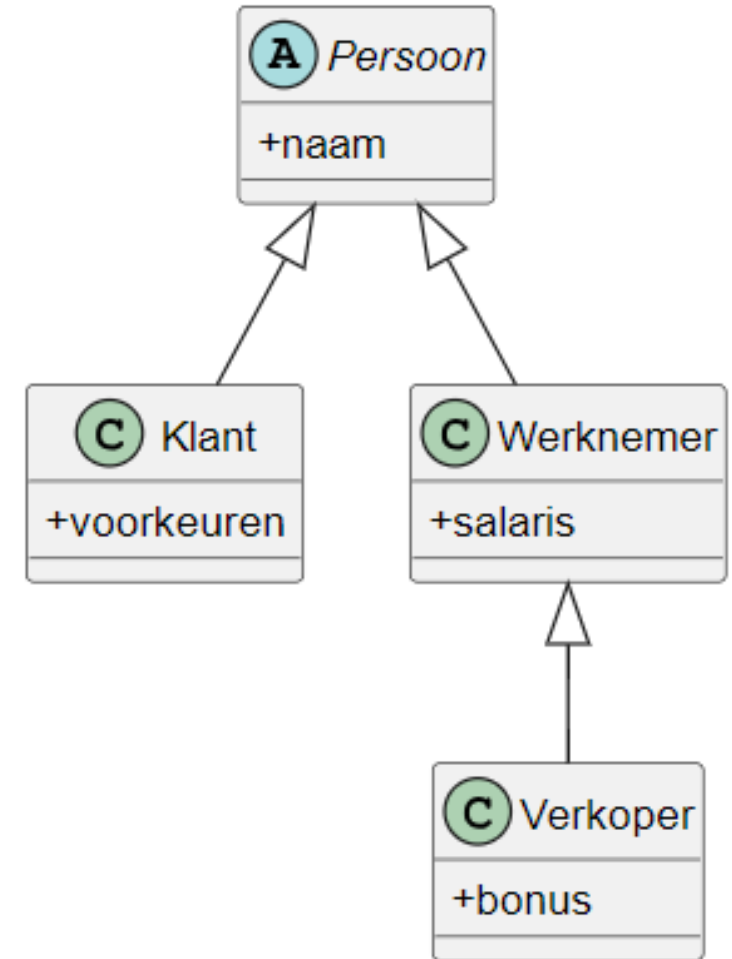


Overerving

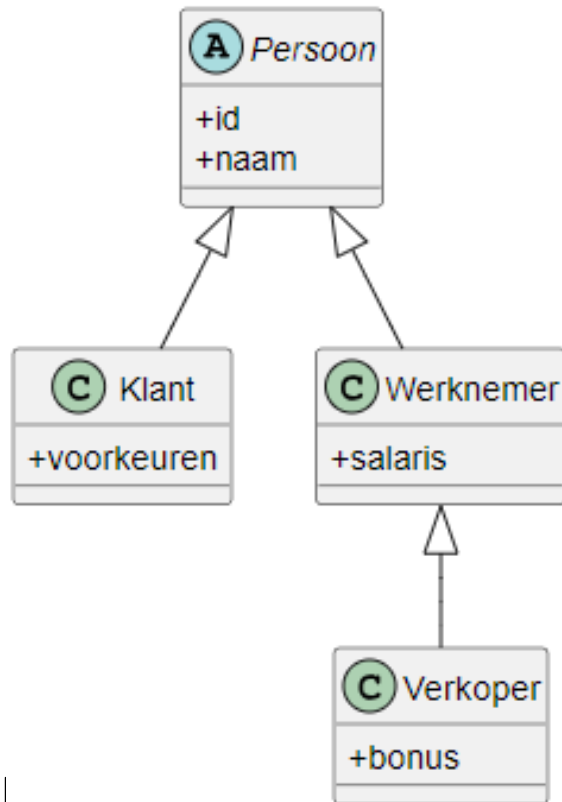
- Hoe vertaal je overerving naar een gegevensbank?

- Mogelijkheden

- Volledige hiërarchie naar één tabel
- Elke klasse naar een tabel
- Elke concrete klasse naar eigen tabel



Tabel per klassenhiërarchie



Persoon	
PERSOON_ID	→ primaire sleutel
PERSOON_TYPE	→ type object
NAAM	
VOORKEUREN	
SALARIS	
BONUS	

- Nadeel
 - verplichte velden niet mogelijk
 - moeilijk uitbreidbaar



Annotaties

DEFAULT

```
@Entity
```

```
@Inheritance(strategy = InheritanceType.SINGLE_TABLE)
```

```
@DiscriminatorColumn(name = "PERSOON_TYPE",  
    discriminatorType= DiscriminatorType.STRING)
```

```
public abstract class Persoon implements Serializable { ...}
```

→ Type opslag overerving

→ Kolom objecttype

```
@Entity
```

```
@DiscriminatorValue("K")
```

```
public class Klant extends Persoon {...}
```

→ Waarde ~ objecttype

```
@Entity
```

```
@DiscriminatorValue("W")
```

```
public class Werknemer extends Persoon {...}
```

```
@Entity
```

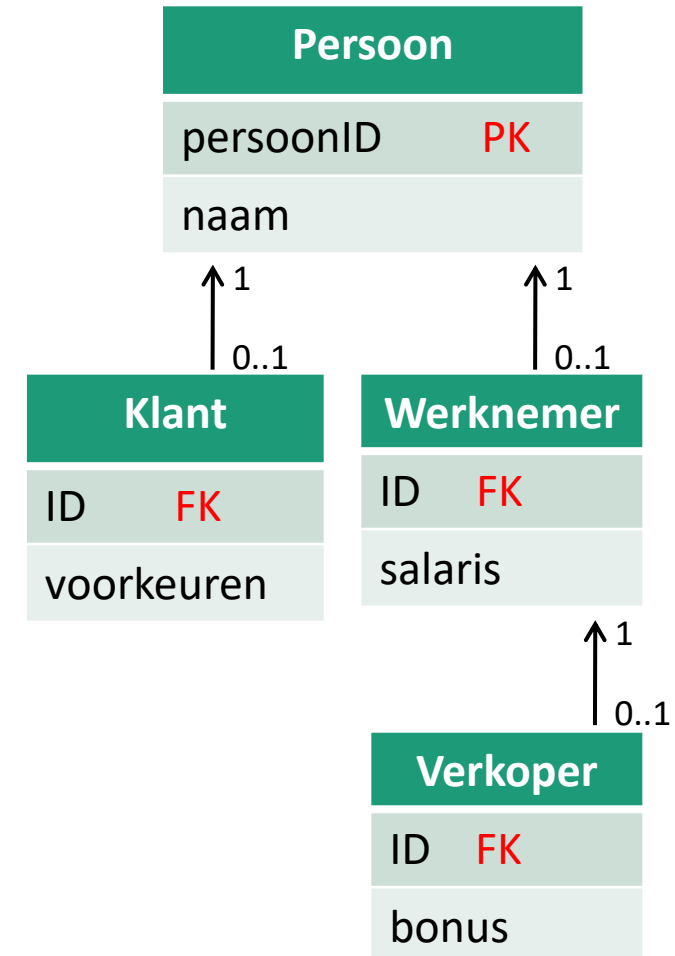
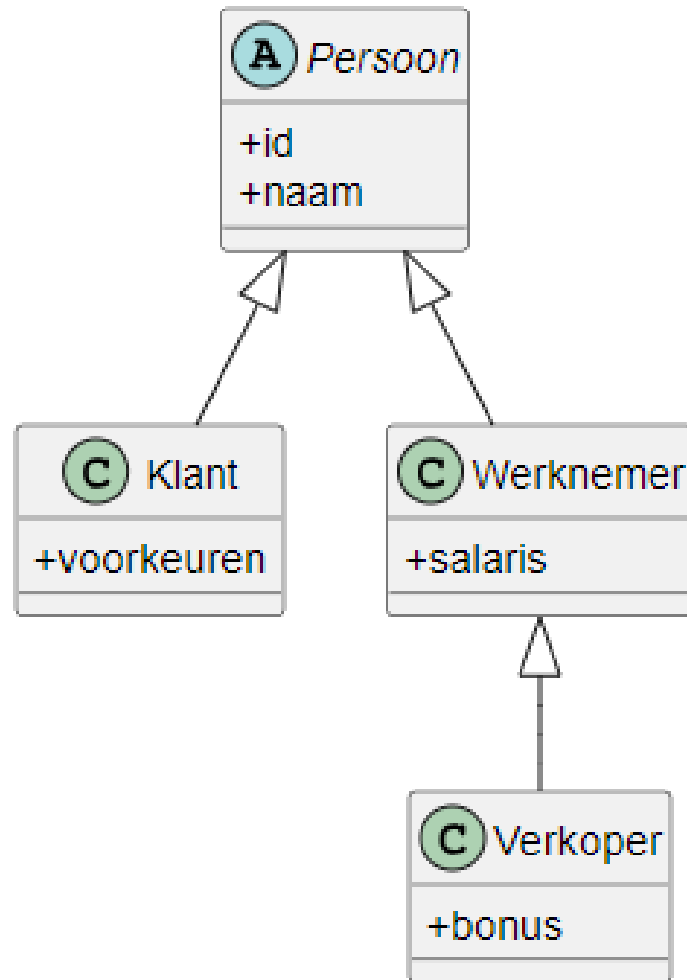
```
@DiscriminatorValue("V")
```

```
public class Verkoper extends Werknemer {...}
```



Elke klasse naar een tabel

VERTICALE MAPPING



Annotaties

```
@Entity
@Inheritance(strategy = InheritanceType.JOINED)
@Table(name="PERSONEN")
public abstract class Persoon implements Serializable { ...}
```

—————→ Type opslag overerving
—————→ Tabel

```
@Entity
@Table(name="KLANTEN")
@PrimaryKeyJoinColumn(name="ID")
public class Klant extends Persoon { ...}
```

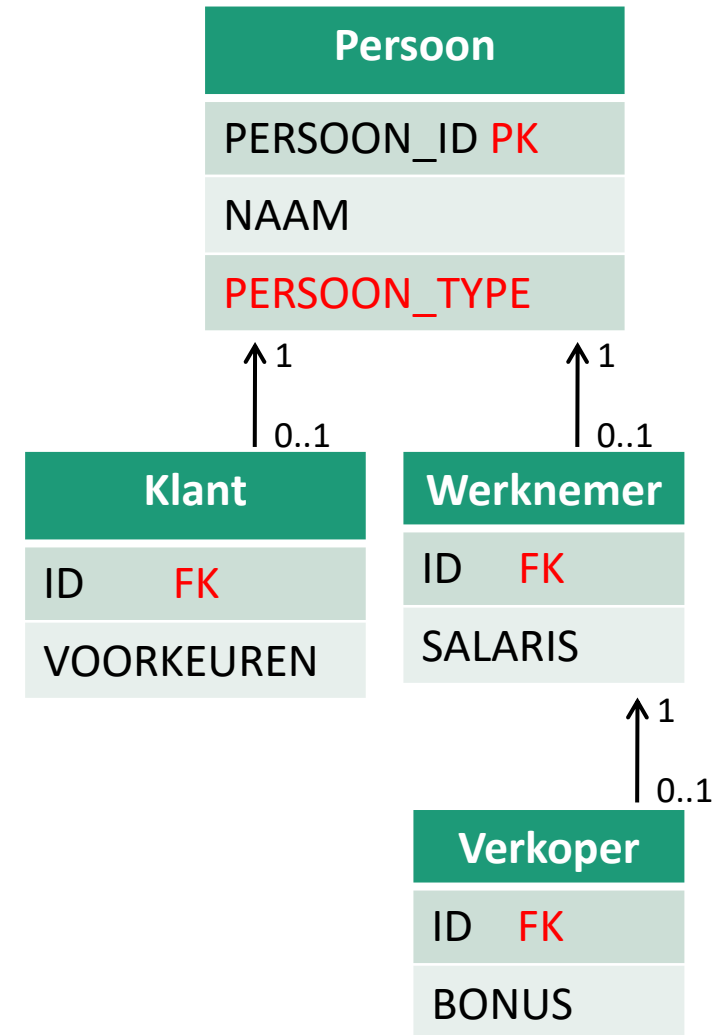
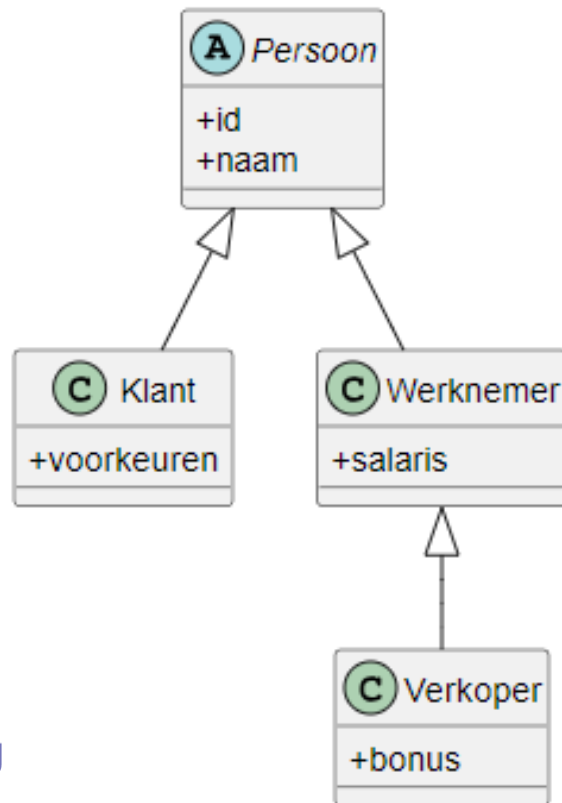
—————→ Kolom foreign key
~ primary key

```
@Entity
@Table(name="VERKOPERS")
@PrimaryKeyJoinColumn(name="ID")
public class Verkoper extends Werknemer { ...}
```



Efficiënter?

- In basistabel
 - Extra kolom ~ subtype



Annotaties

```
@Entity
```

```
@Inheritance(strategy = InheritanceType.JOINED)
```

```
@Table(name="PERSONEN_EFFICIENT")
```

```
@DiscriminatorColumn(name = "PERSON_TYPE", discriminatorType=  
DiscriminatorType.STRING)
```

```
public abstract class Persoon implements Serializable { ...}
```

Type opslag overerving

Tabel

Kolom objecttype

```
@Entity
```

```
@DiscriminatorValue("K")
```

```
@Table(name="KLANTEN_EFFICIENT")
```

```
@PrimaryKeyJoinColumn(name="ID")
```

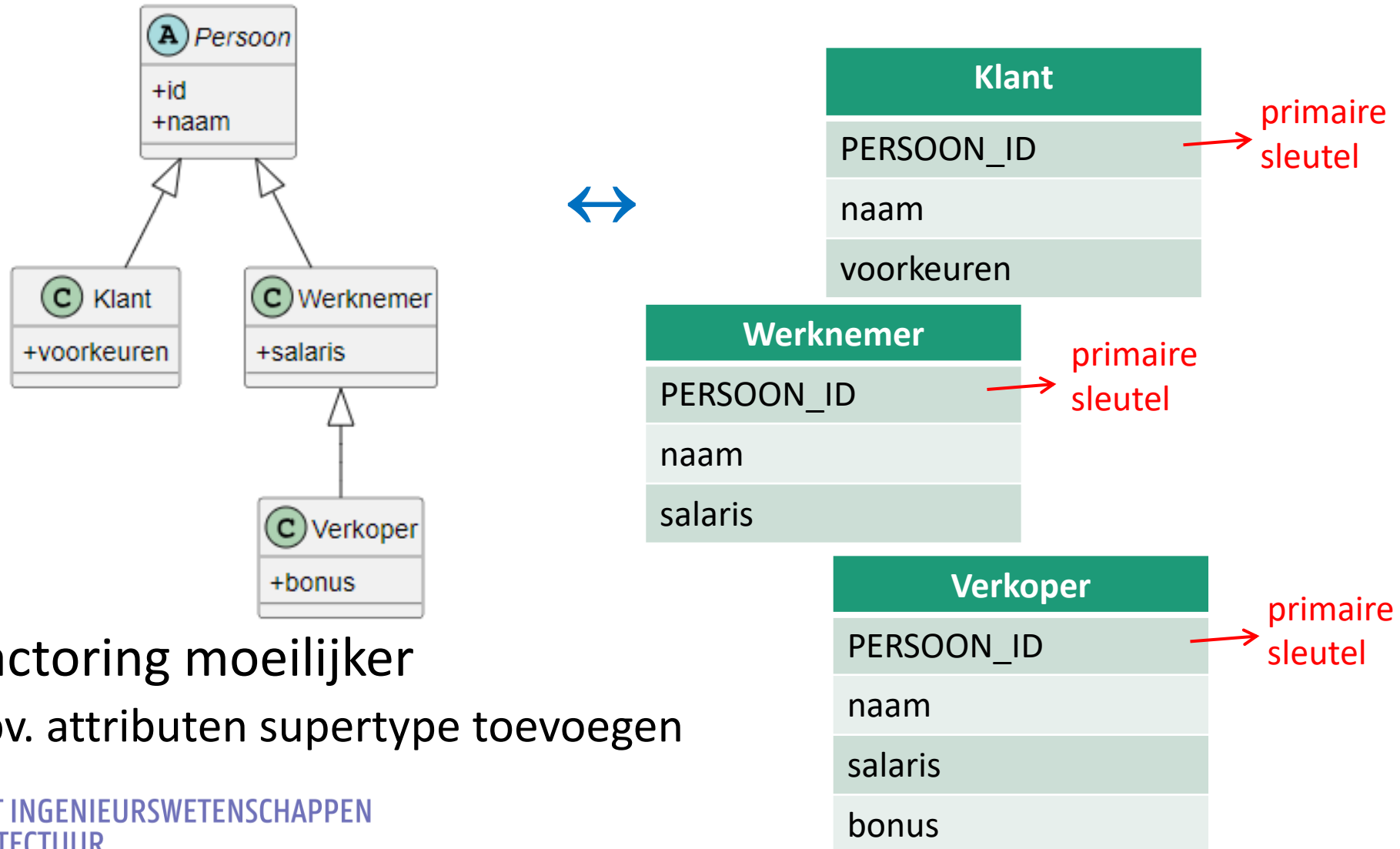
```
public class Klant extends Persoon {...}
```

Waarde objecttype

Kolom foreign key
~ primary key



Elke concrete klasse naar eigen tabel



- Refactoring moeilijker
 - bv. attributen supertype toevoegen



Annotaties

```
@Entity
@Inheritance(strategy = InheritanceType.TABLE_PER_CLASS)
public abstract class Persoon implements Serializable {
    @Id
    @Column(name="PERSOON_ID")
    @GenericGenerator(name="generator", strategy="increment")
    @GeneratedValue(generator="generator")
    public int getId() { return id; }
    ... }
```

Type opslag overerving

Primaire sleutel gedeeld
over verschillende
tabellen

```
@Entity
public class Klant extends Persoon {...}
```

