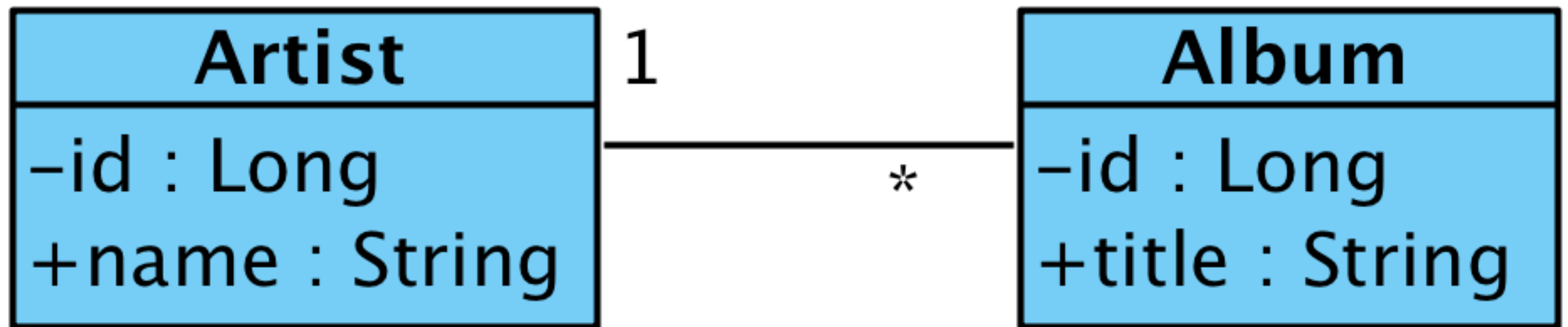




Persistence - Part 2

Philippe Collet, contains 78,3% of slides from
Sébastien Mosser

Problem: Representing associations



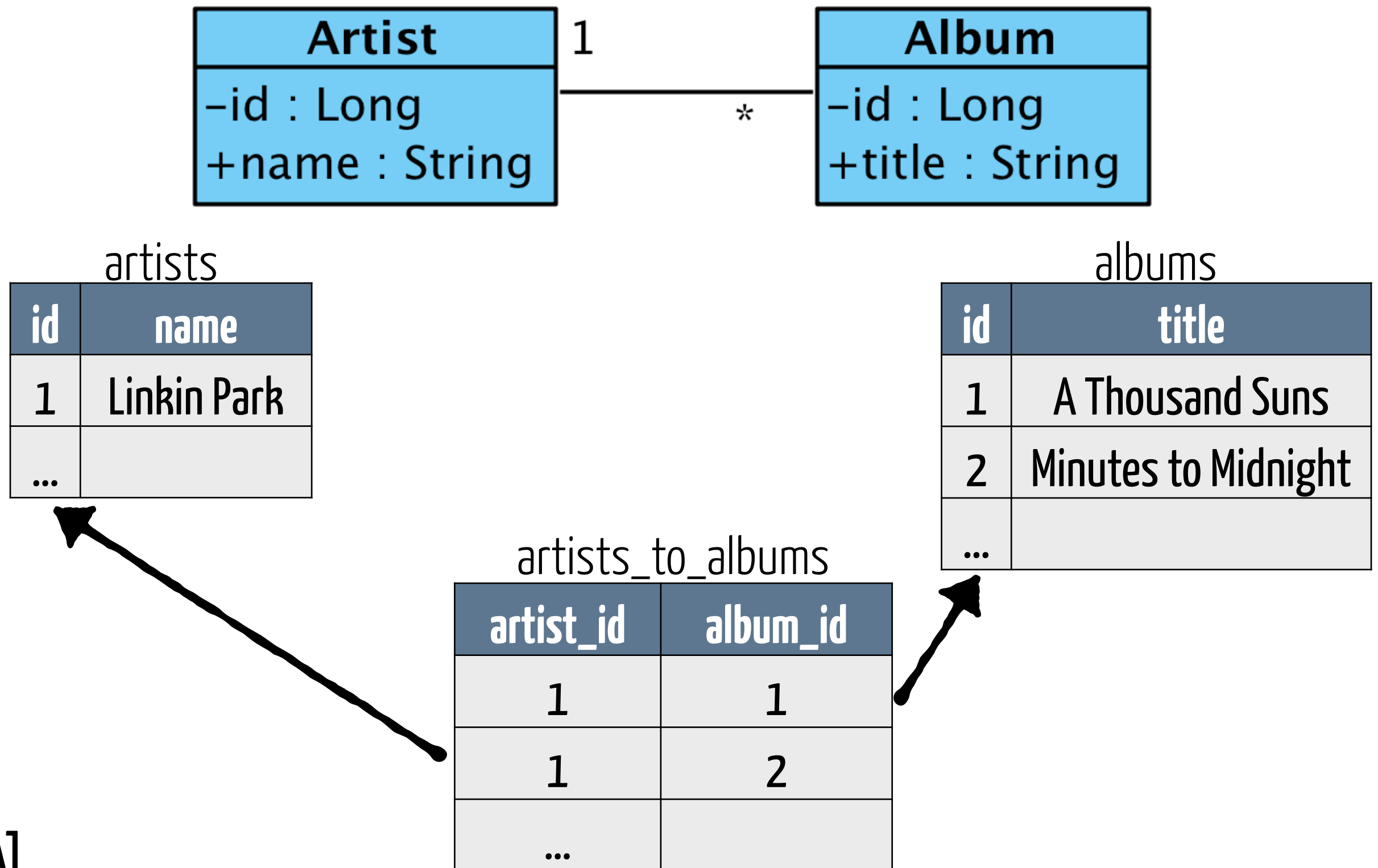
artists

id	name
1	Linkin Park
	...

albums

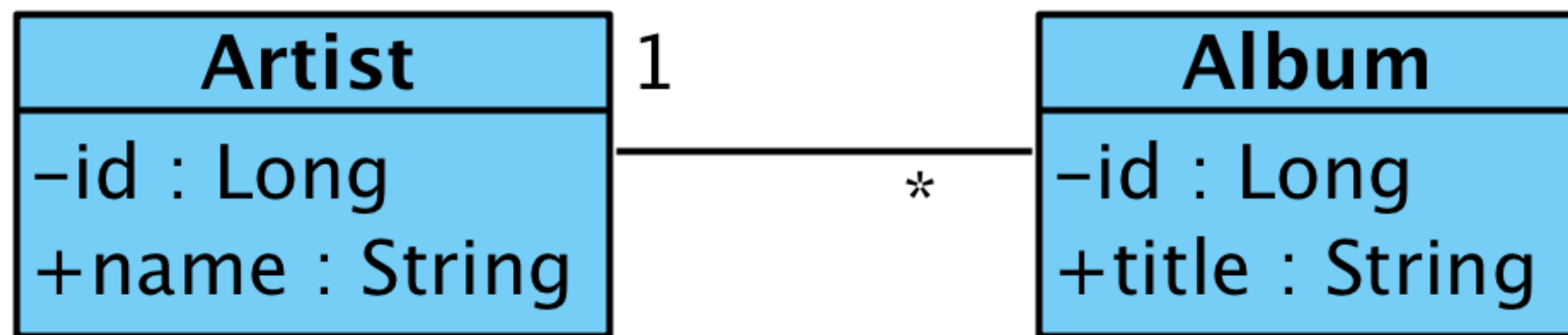
id	title
1	A Thousand Suns
2	Minutes to Midnight
	...

Solution #1: Association Table [M-N]



Solution #2: Foreign Key

[1-N]



artists

id	name
1	Linkin Park
...	

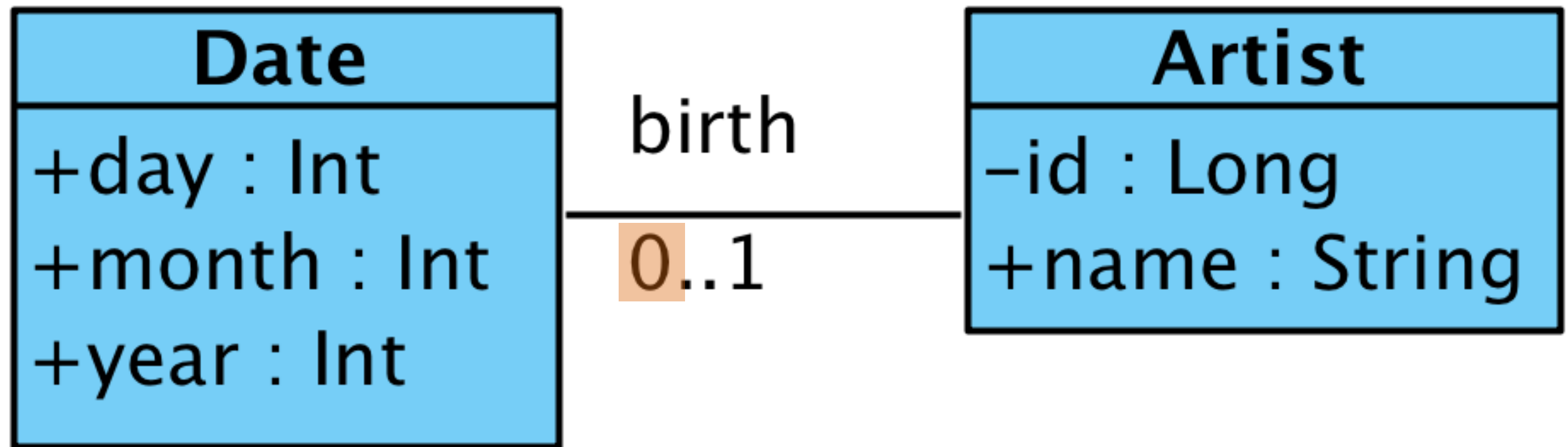
albums

id	title	artist_id
1	A Thousand Suns	1
2	Minutes to Midnight	1
...		

or **[1-N]** \equiv **[M-N]** when $N = 1$

Solution #3: Relation Merge

[1-1]



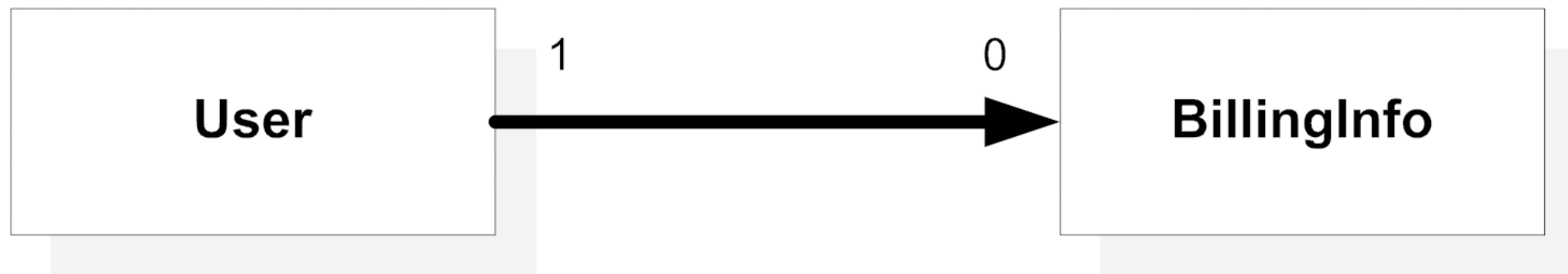
artists

id	name	birth_day	birth_month	birth_year
1	Linkin Park	-1	-1	-1
...				

or $[1-1] \equiv [1-N]$ when $N = 1$

or $[1-1] \equiv [M-N]$ when $M = 1$ and $N = 1$

Type of Relationship	Annotation
1-1	@OneToOne
1-n	@OneToMany
n-1	@ManyToOne
n-m	@ManyToMany



@Entity

```
public class User {
```

```
    @Id
```

```
    protected String userId;
```

```
    protected String email;
```

```
    @OneToOne
```

```
    protected BillingInfo billingInfo;
```

```
}
```

**Unidirectional
1-1 mapping**

[EiA]

For property-based beans, annotate the getter.

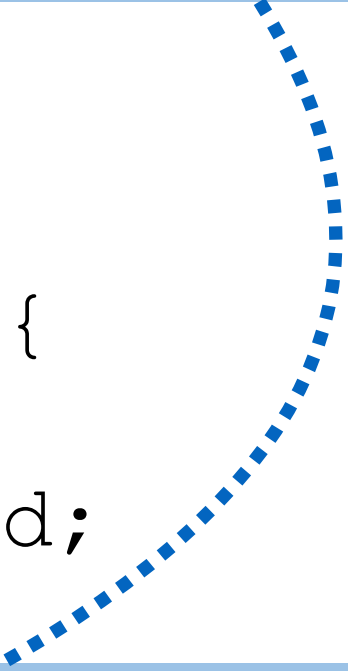
Bidirectional 1-1 mapping

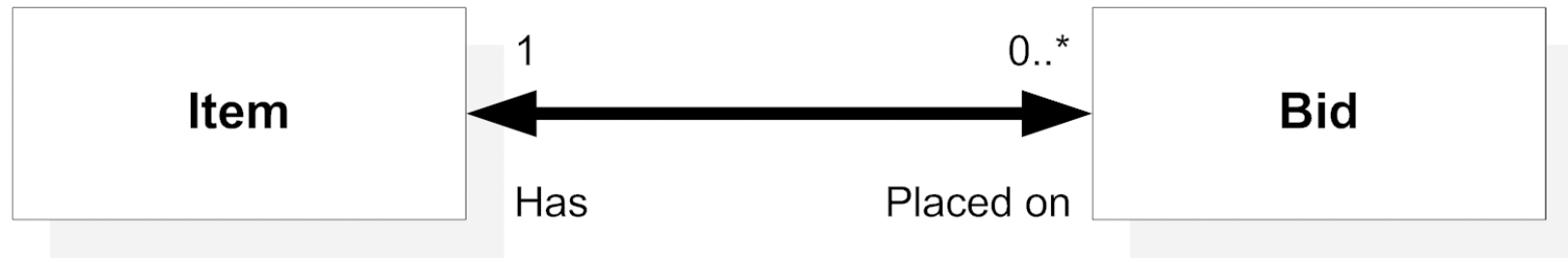
```
@Entity
public class User {
    @Id
    protected String userId;
    protected String email;

    @OneToOne
    protected BillingInfo billingInfo;
}
```

```
@Entity
public class BillingInfo {
    @Id
    protected Long billingId;

    @OneToOne (mappedBy="billingInfo", optional=false)
    protected User user;
}
```





@Entity

```
public class Bid {
```

```
    @Id
```

```
    protected String bidId;
```

```
    @ManyToOne
```

```
    protected Item item;
```

```
}
```

Owner

@Entity

```
public class Item {
```

```
    @Id
```

```
    protected String itemId;
```

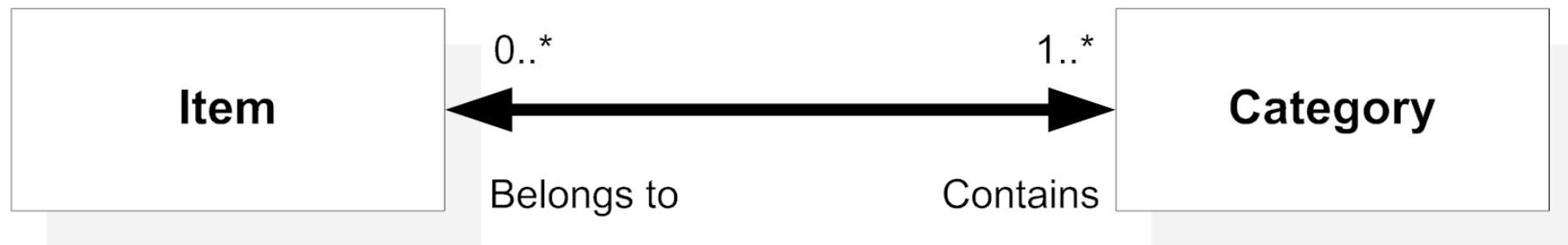
```
    @OneToMany(mappedBy="item")
```

```
    protected Set<Bid> bids;
```

```
}
```

[EiA]

1-n mapping



@Entity

```
public class Category {
```

```
    @Id
```

```
    protected String categoryId;
```

```
    @ManyToMany
```

```
    protected Set<Item> items;
```

```
}
```

Owner

@Entity

```
public class Item {
```

```
    @Id
```

```
    protected String itemId;
```

```
    @ManyToMany(mappedBy="items")
```

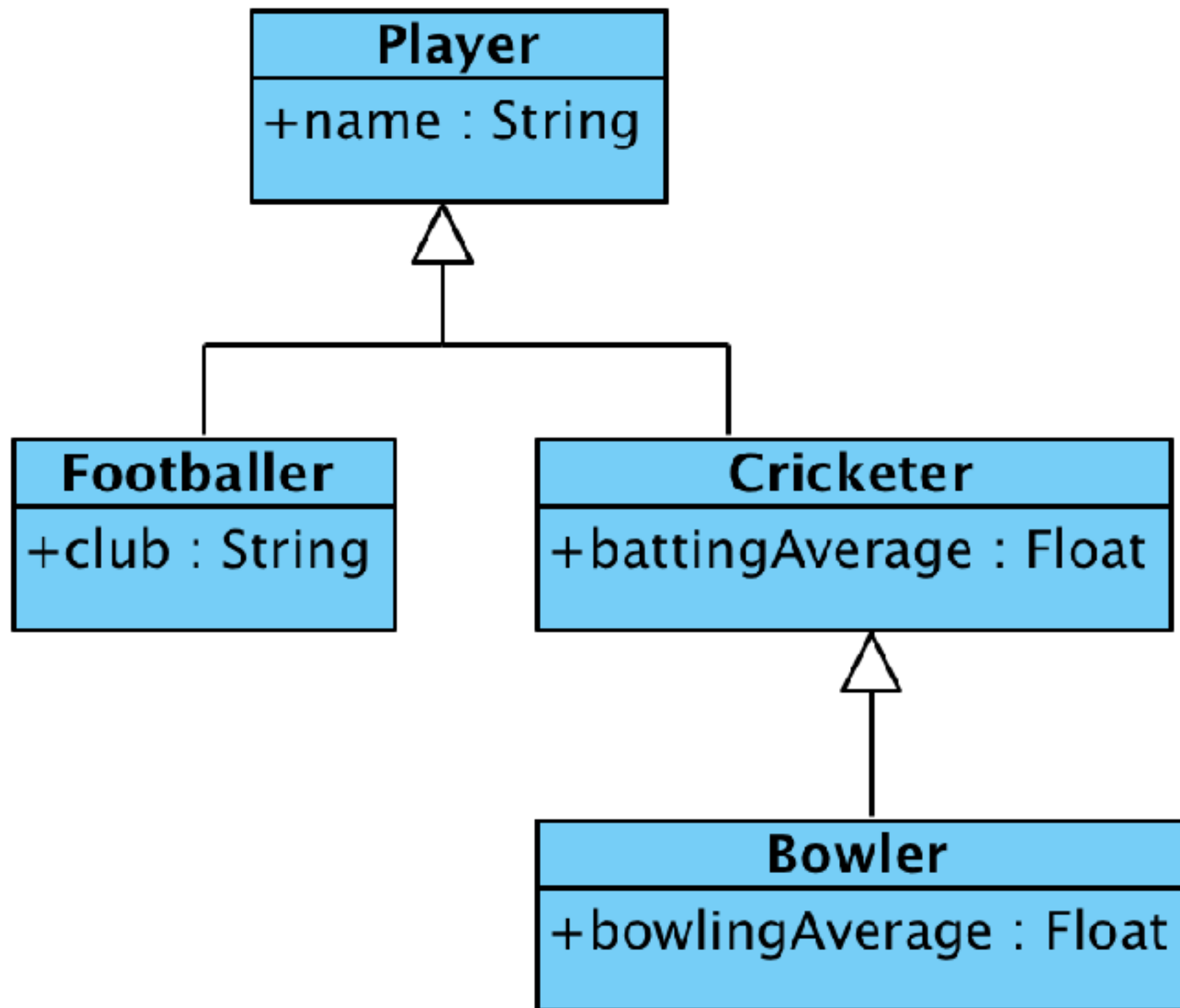
```
    protected Set<Category> categories;
```

```
}
```

n-m mapping

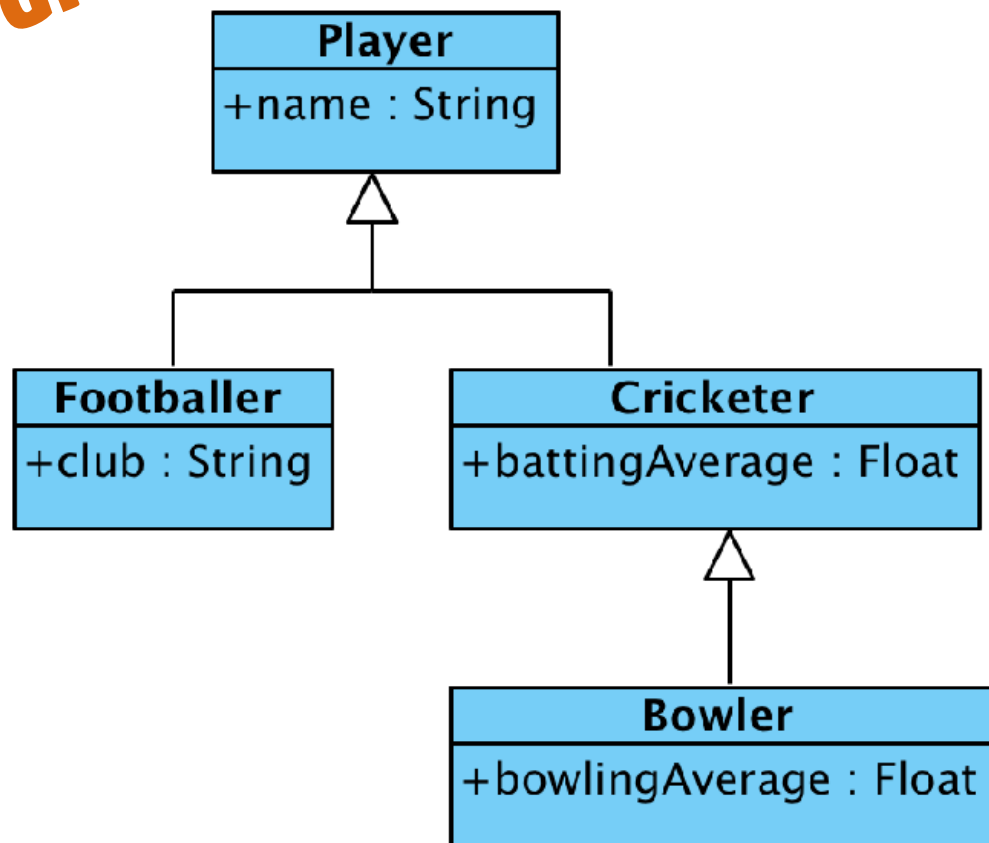
[EiA]

Problem: Implementing Inheritance



Solution #1: **Single-Table** Inheritance

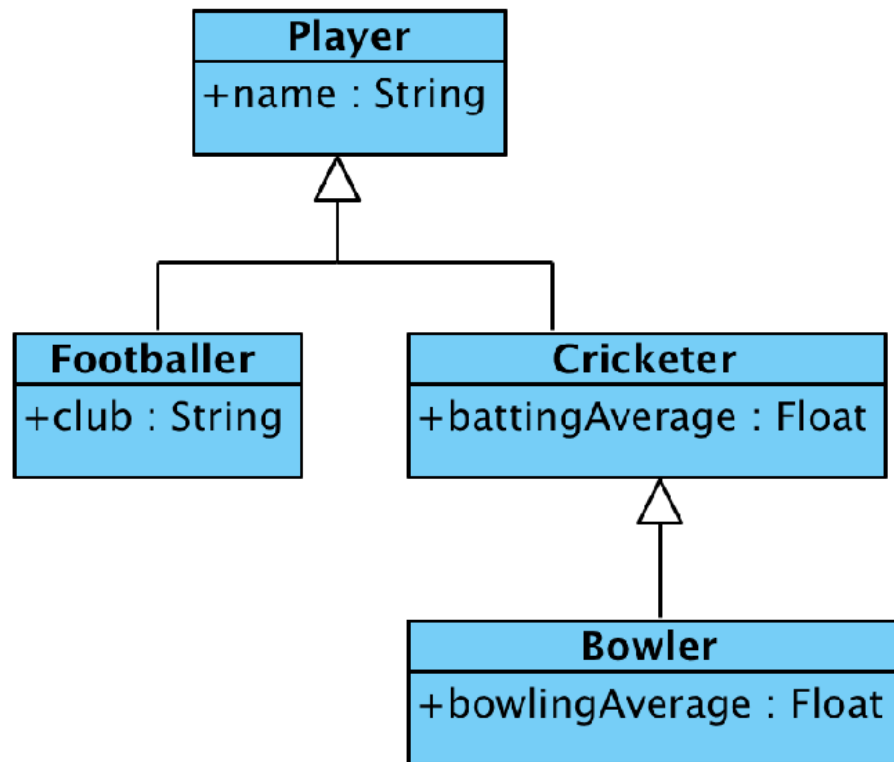
Par défaut



players

name	club	batting_avg	bowling_avg	type

Solution #2: **Class-Table** Inheritance



players

id	name
42	...
74	...
96	...

footballers

id	club
42	...

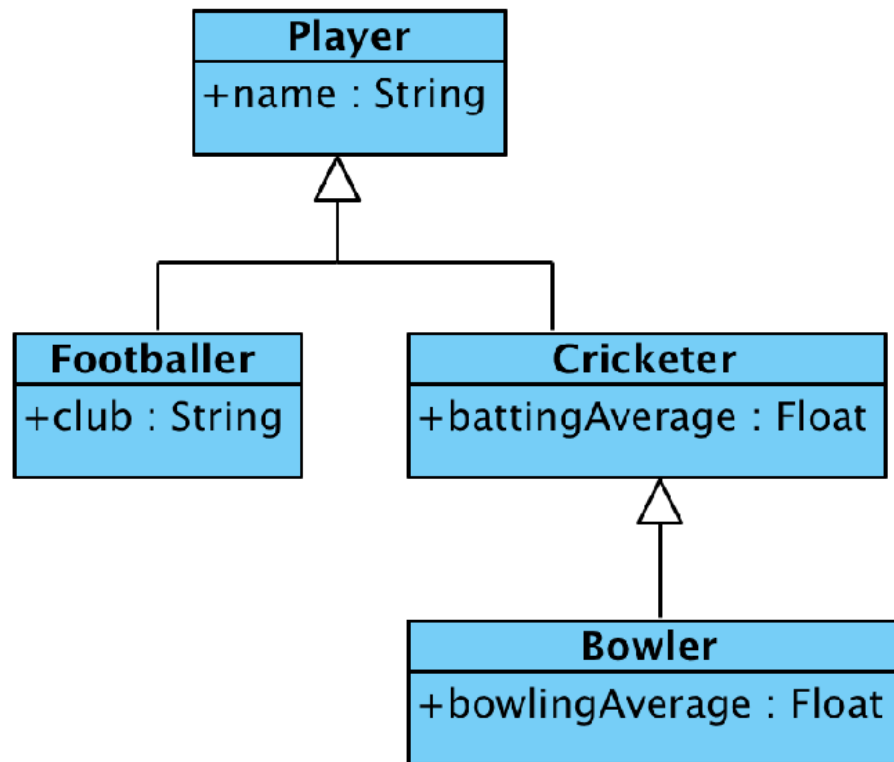
cricketers

id	batting_avg
74	...
96	...

bowlers

id	bowling_avg
96	...

Solution #3: **Concrete-Table** Inheritance



footballers

id	name	club
42

cricketers

id	name	batting_avg
74

bowlers

id	name	batting_avg	bowling_avg
96

Controlling Inheritance

```
@Entity
@Table(name="USERS")
@Inheritance(strategy=InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name="USER_TYPE", ...)
public class User {
    // ...
}
```

```
@Entity
@DiscriminatorValue(value="S")
public class Seller extends User { ... }

// ...
```

See [EiA], chapter 9