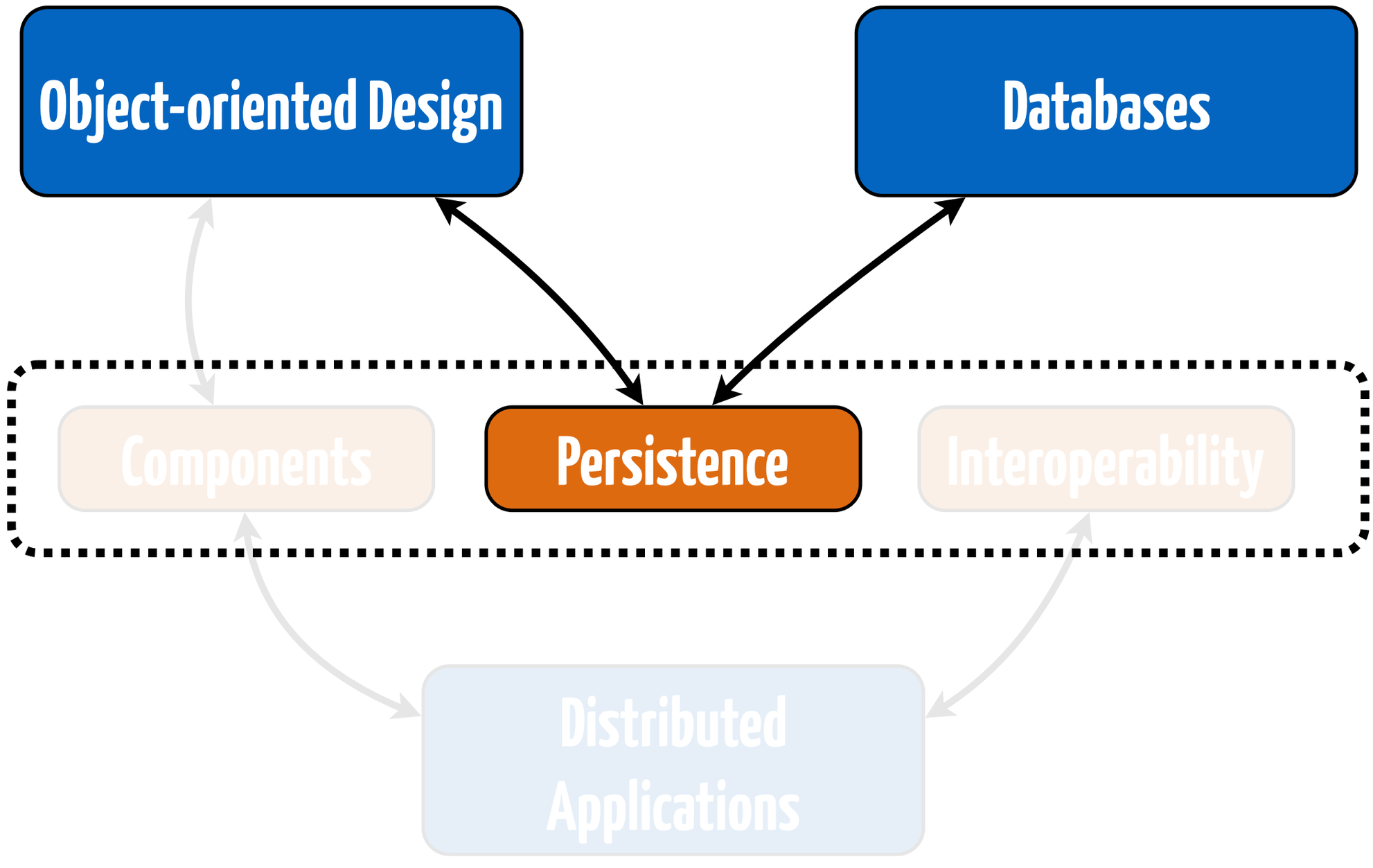
Object-Relational Mapping

Sébastien Mosser

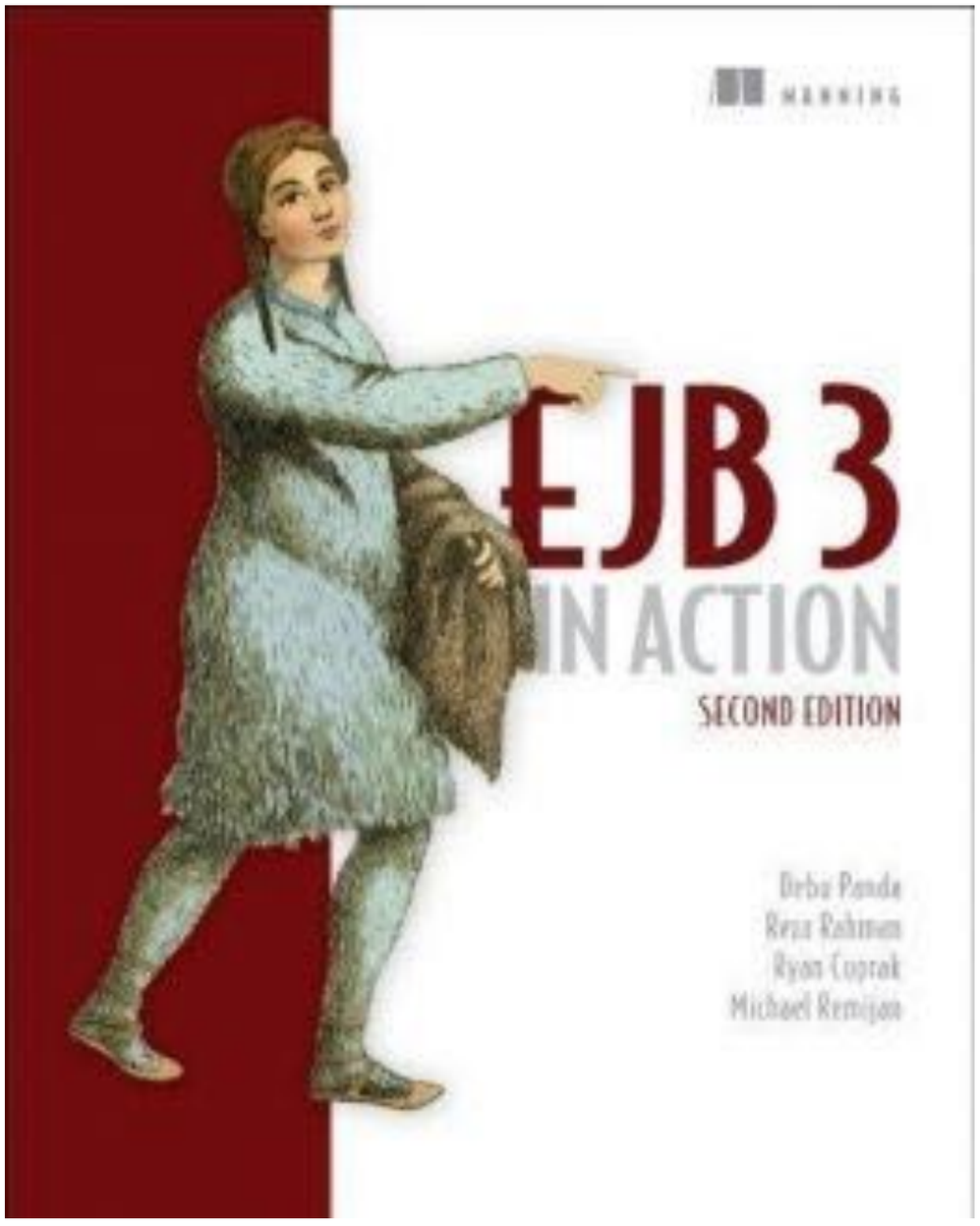
Lecture #2, 06.02.2017

pictures: sxc.hu

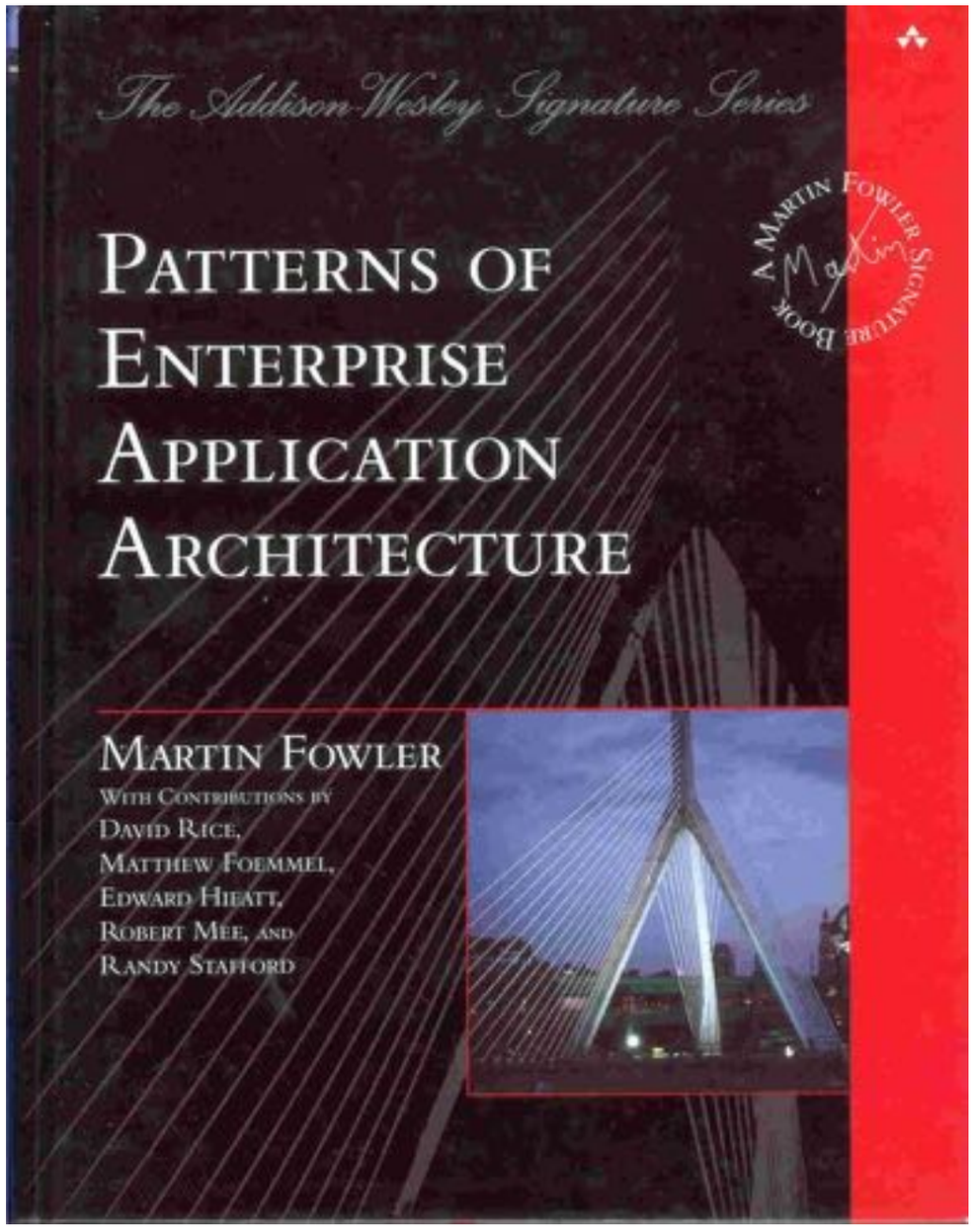
Applications Server: Dependencies



Bibliography



[EJB3 in Action, 2nd ed., 2014]



[PoEAA, 2002]

1 Structural **Mappings**

2 **Architectural** patterns

3 **Behavioral** patterns

4 **Conclusions**

Structural

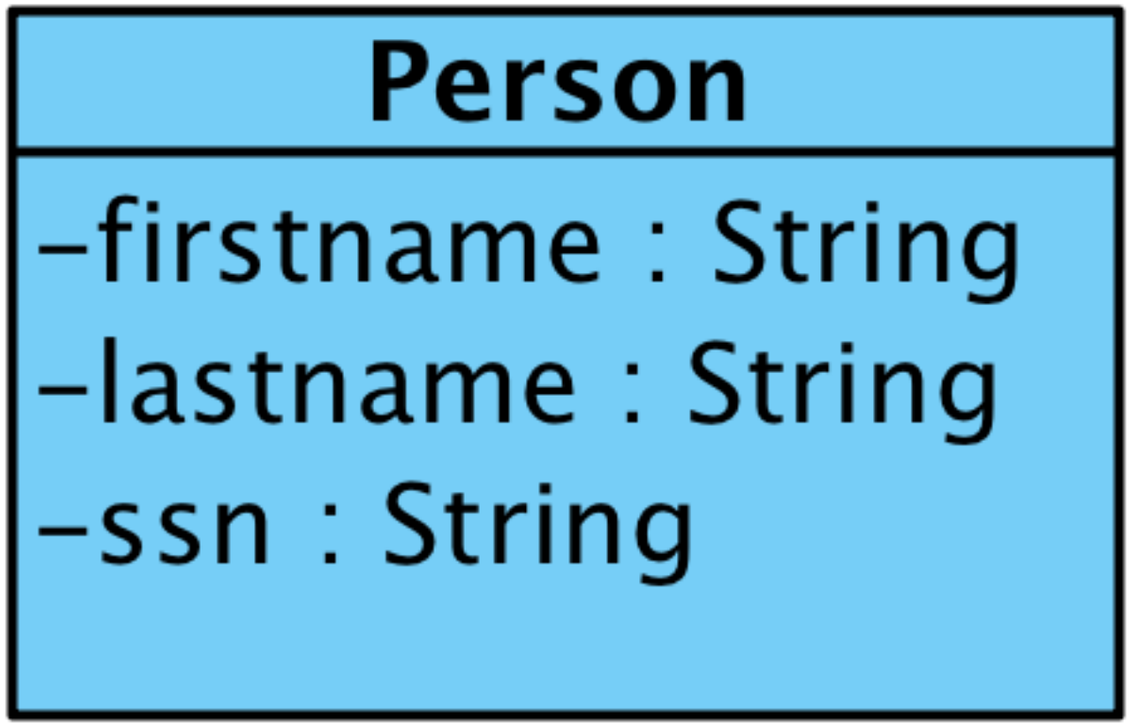
Mappings

Impedance Mismatch

Object-Oriented	Relational
Classes	Relation (table)
Object	Tuple (row)
Attribute	Attribute (column)
Identity	Primary Key
Reference	Foreign Key
Inheritance	N/A
Methods	~ Stored Procedure

[EJB3 in Action]

Example of **Domain Model**



first_name	last_name	ssn
Sébastien	MOSSER	16118325358
...		

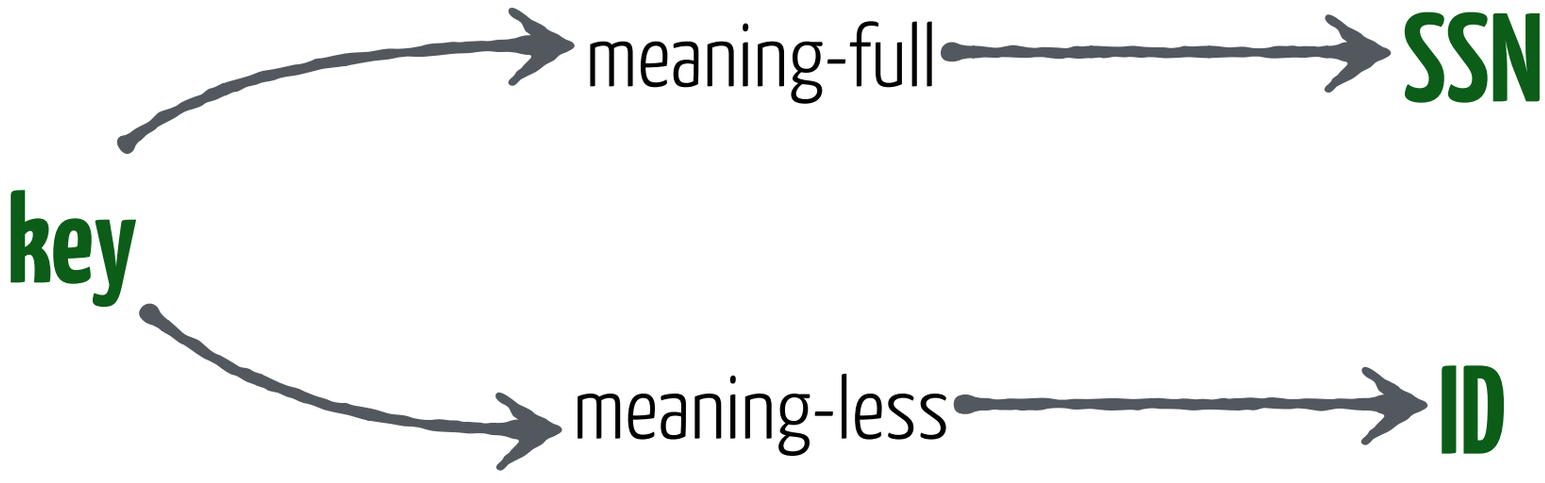
[PoEAA]

Problem: Domain Object Identity

```
Person p1 = new Person("X","Y","DDMMYYXXXXX")
Person p2 = new Person("X","Y","DDMMYYXXXXX")
```

How to support persons **uniqueness**?

Candidates for **Key** role



- Necessary condition:
  - **Key must be unique**
- Nice-to-have condition:
  - **Key should be immutable**
- **Compound** versus **Simple**

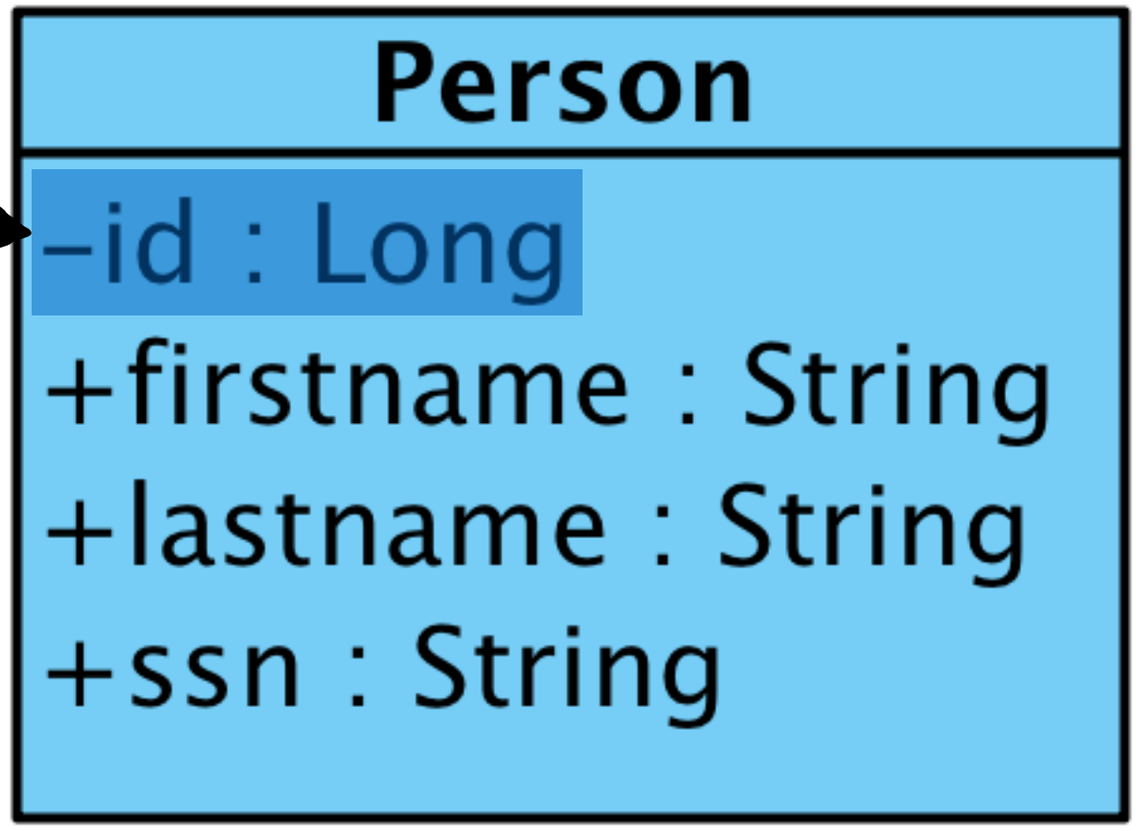
[PoEAA]

Pattern: **Identity Field**

Not visible at
the business level

auto-generated

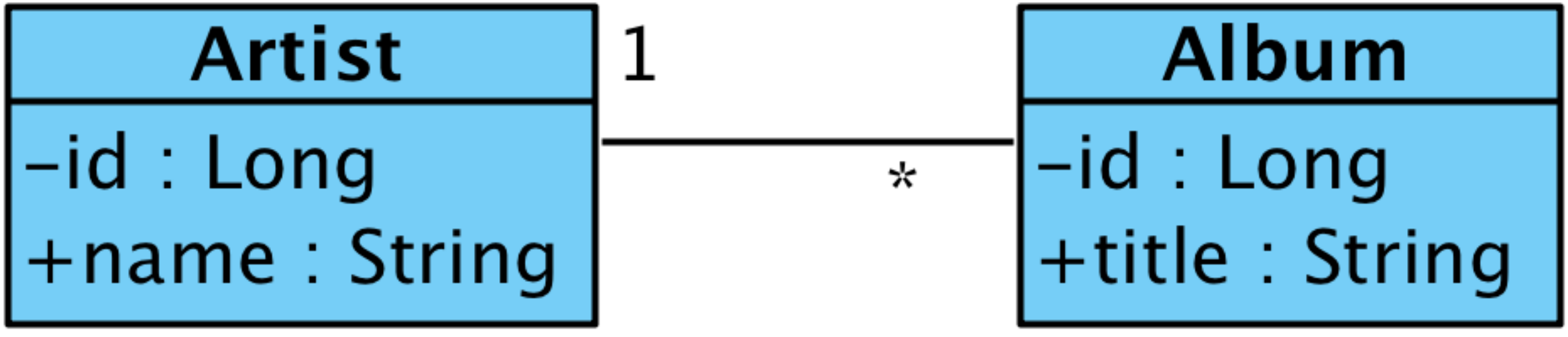
GUID DIY



[PoEAA]

When are 2 customers equals?

Problem: Representing associations



artists

id	name
1	Linkin Park
	…

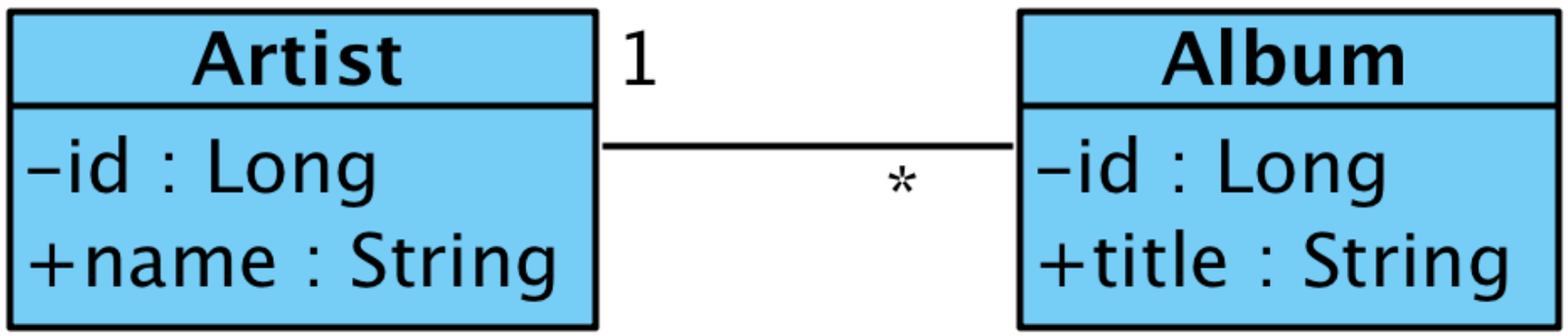
albums

id	title
1	A Thousand Suns
2	Minutes to Midnight
	…

[PoEAA]

How to bind customers to orders?

Solution #1: Association Table [M-N]



artists

id	name
1	Linkin Park
...	

albums

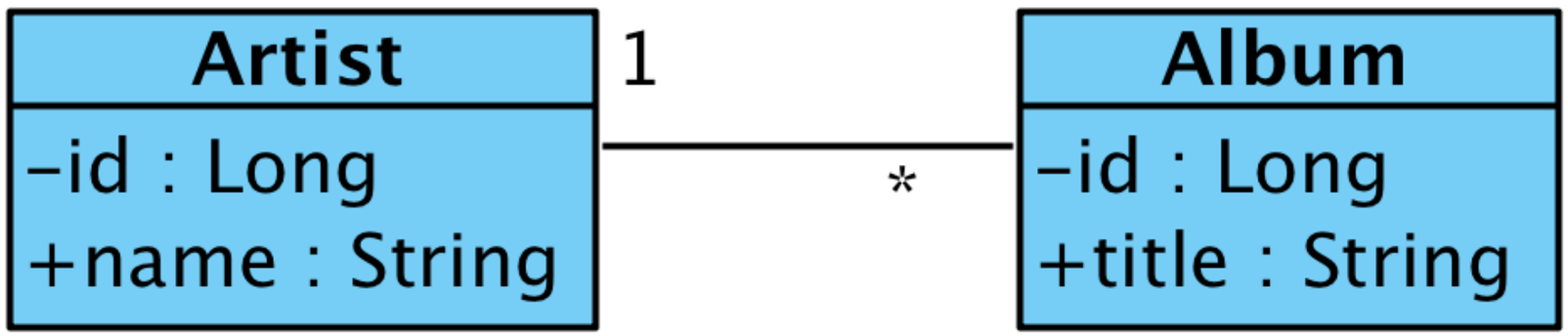
id	title
1	A Thousand Suns
2	Minutes to Midnight
...	

artists_to_albums

artist_id	album_id
1	1
1	2
...	

[PoEAA]

Solution #2: **Foreign Key** [1-N]



artists

id	name
1	Linkin Park
...	

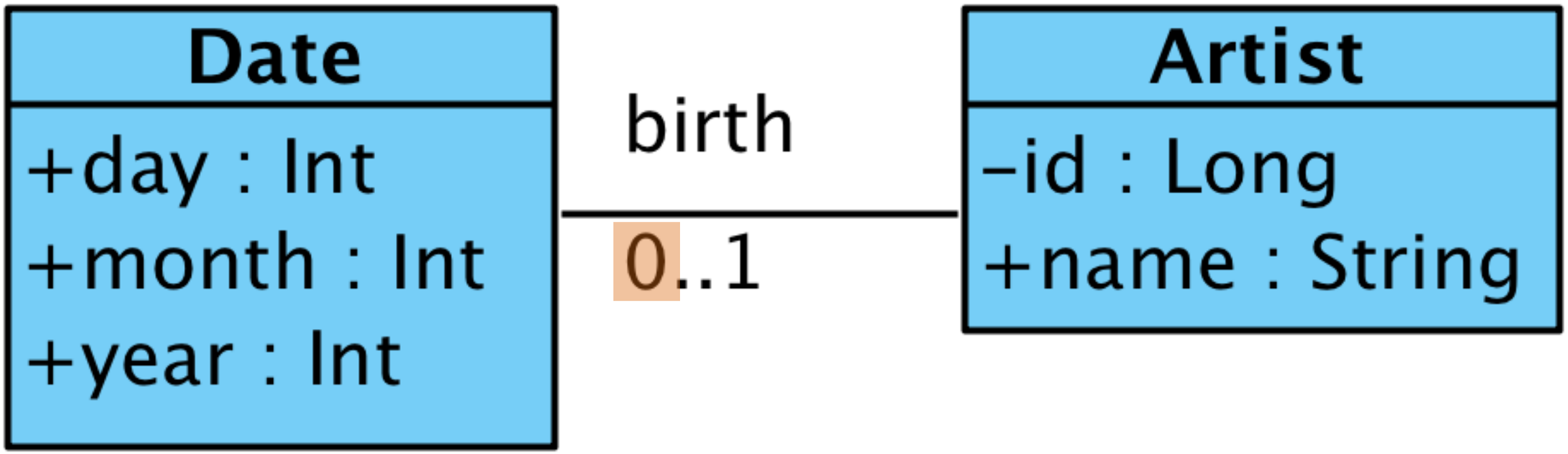
albums

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

or **[1-N]** ≡ **[M-N]** when N =1

[PoEAA]

Solution #3: **Relation Merge** [1-1]



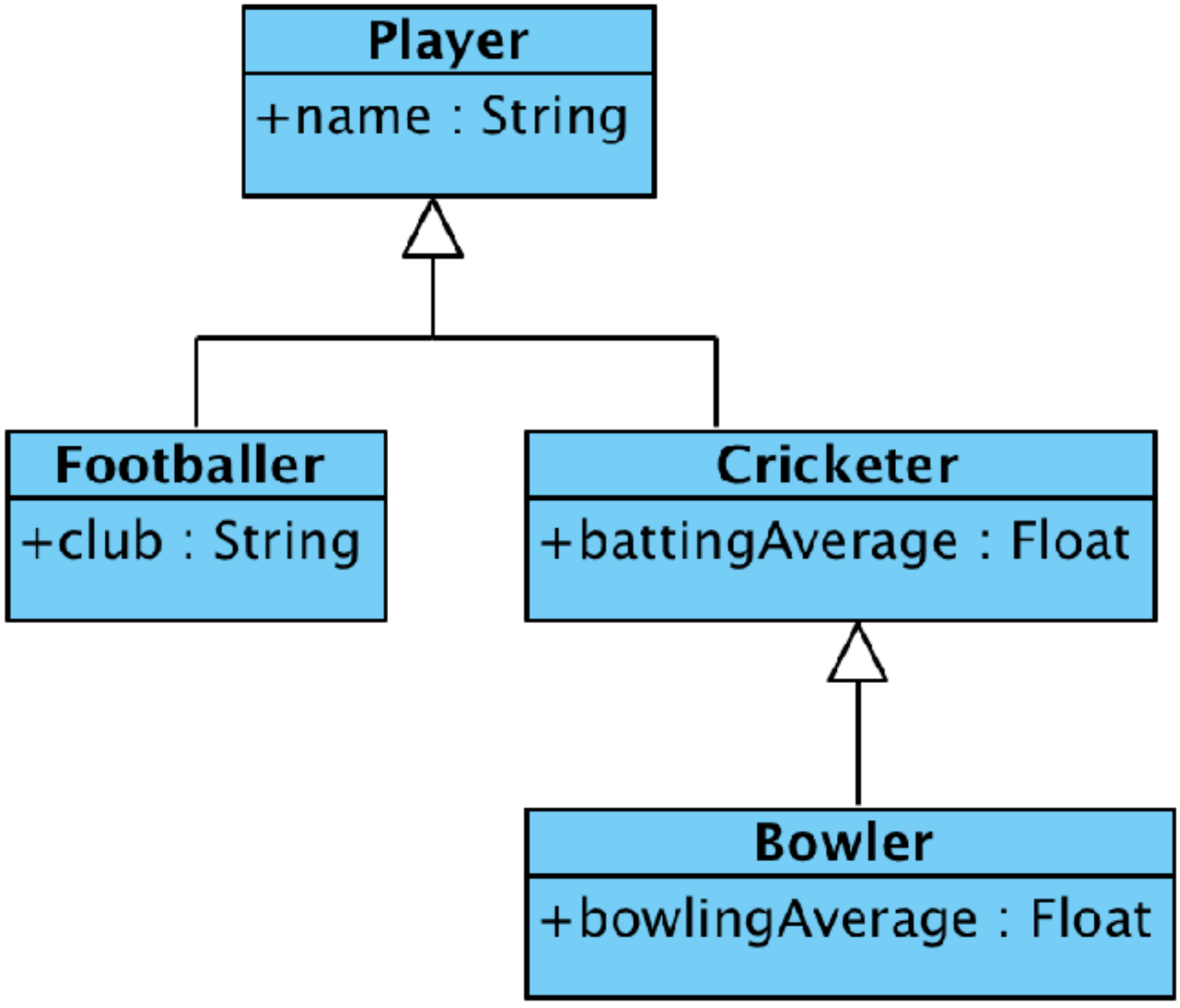
artists

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

or **[1-1]** ≡ **[1-N]** when N =1

[PoEAA] or **[1-1]** ≡ **[M-N]** when M = 1 and N =1

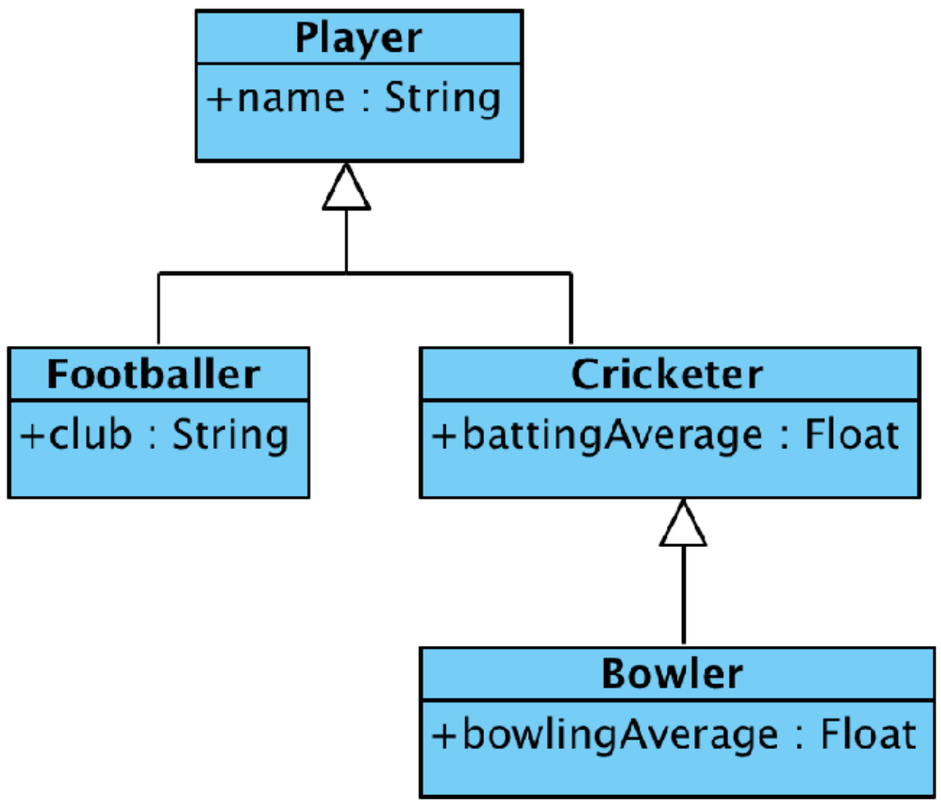
Problem: Implementing Inheritance



[PoEAA]

How to model multiple kind of customers?

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

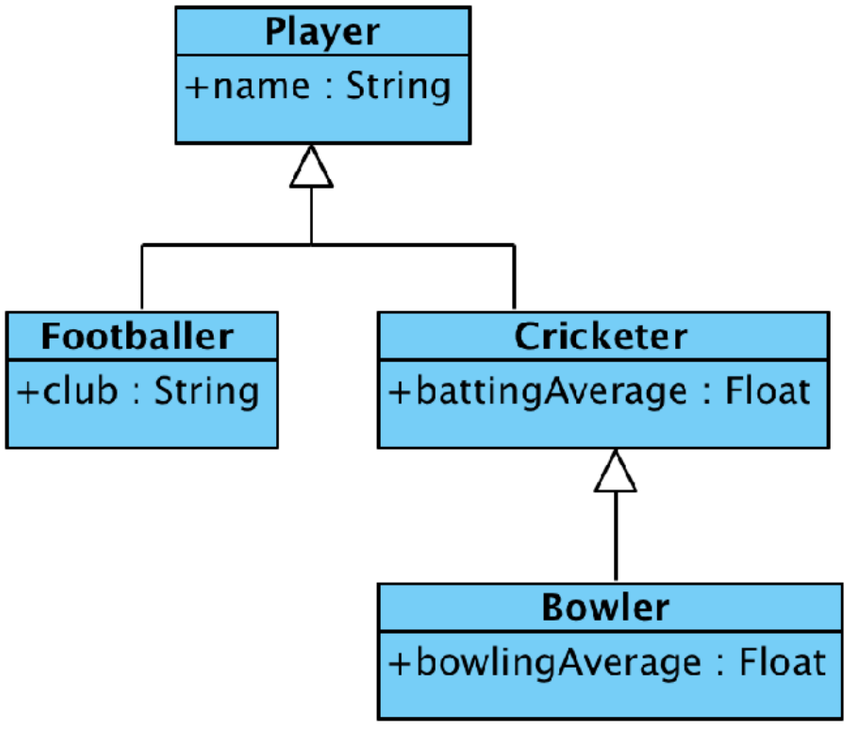


players

name	club	batting_avg	bowling_avg	type

[PoEAA]

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



players

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

footballers

id	club
42	...

cricketers

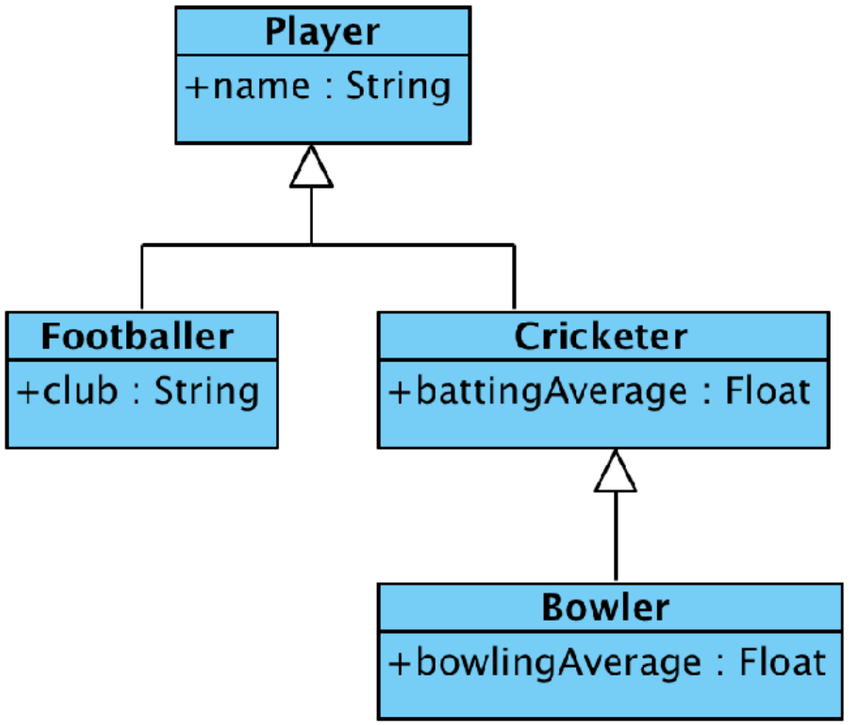
id	batting_avg
74	...
96	...

bowlers

id	bowling_avg
96	...

[PoEAA]

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



footballers

id	name	club
42	...	...

cricketers

id	name	batting_avg
74	...	...

bowlers

id	name	batting_avg	batting_avg
96	...	...	

[PoEAA]

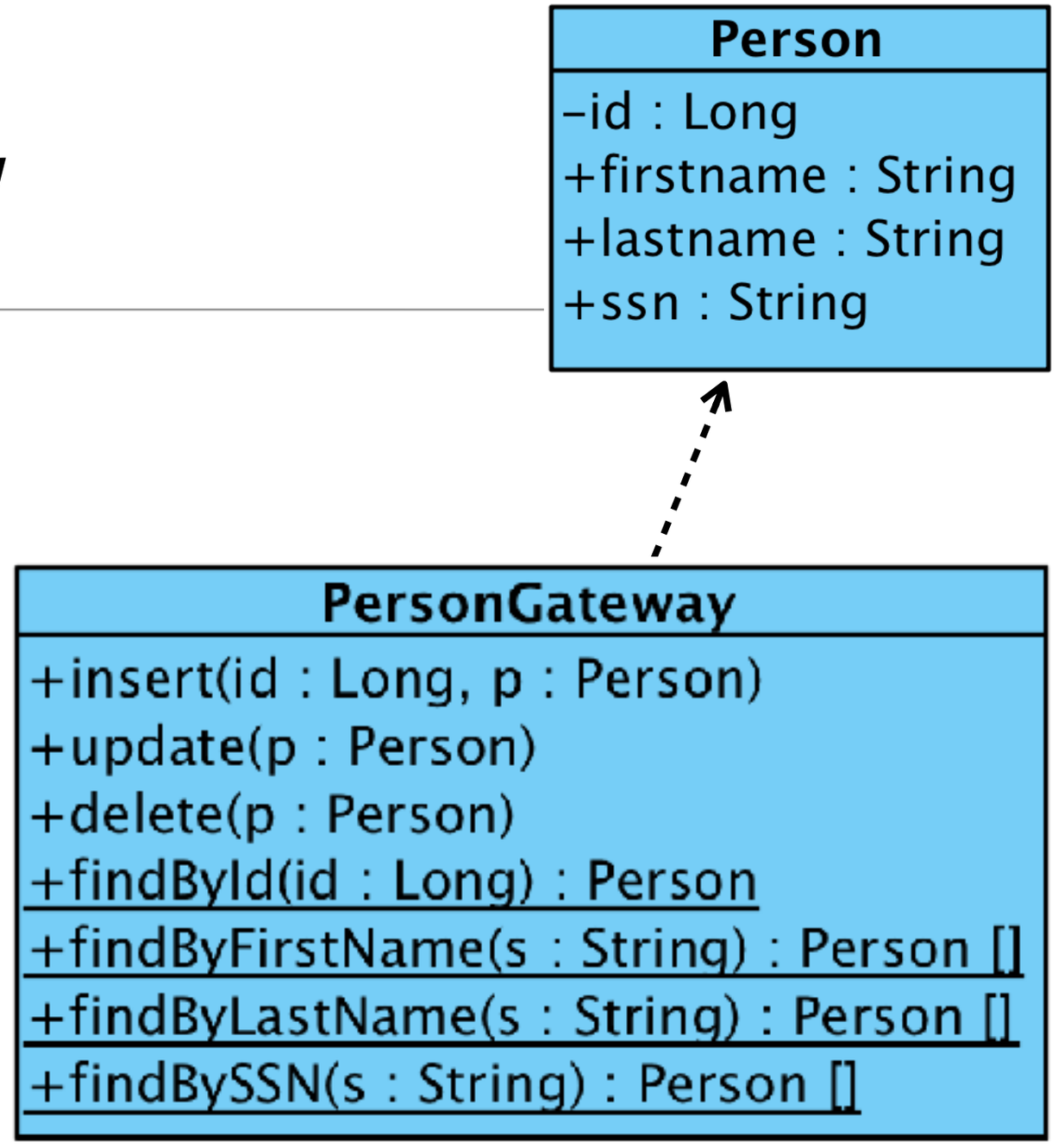
Architectural

patterns

2

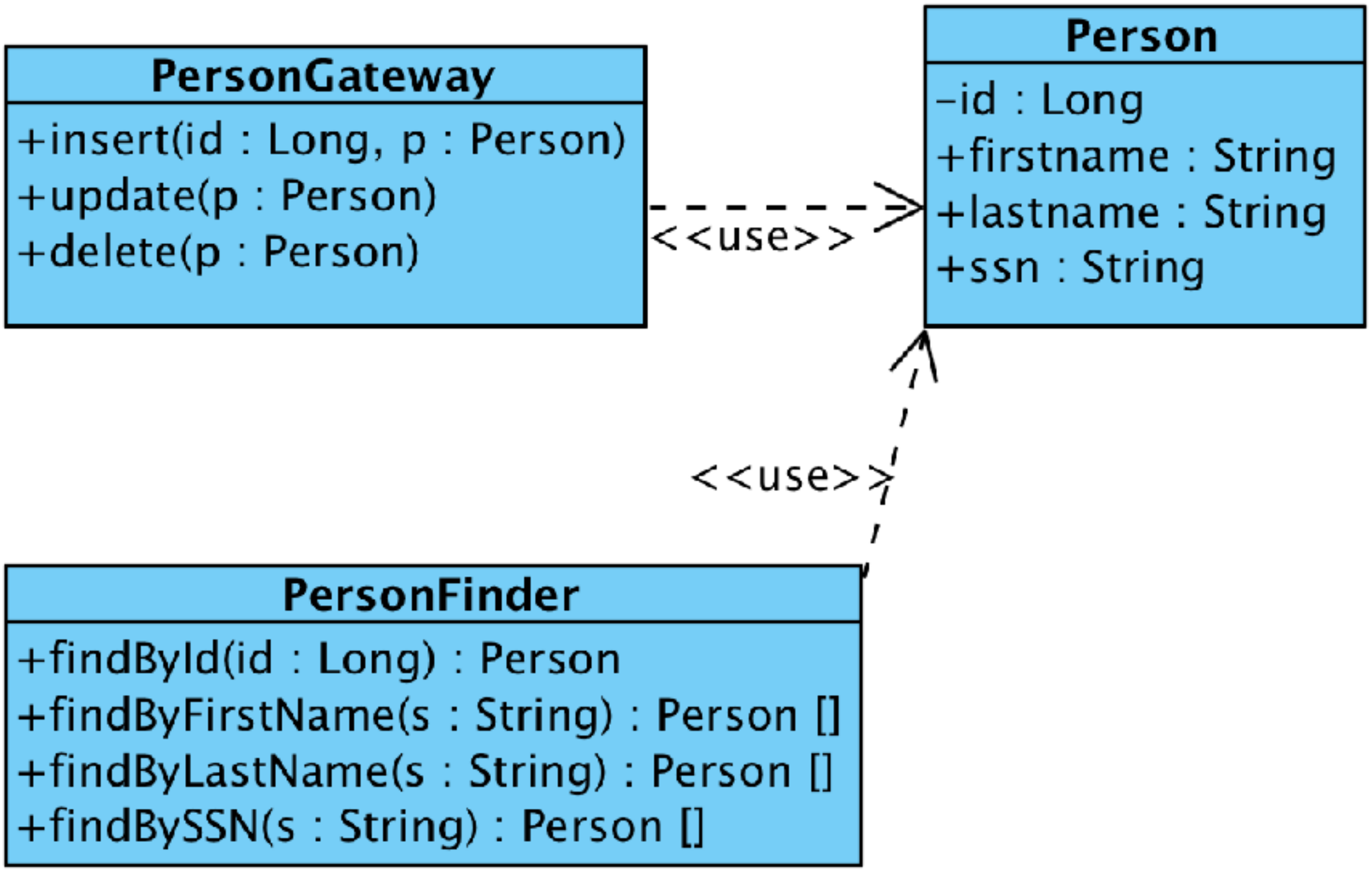
How one can interact with the persistent objects?

Table-Data Gateway

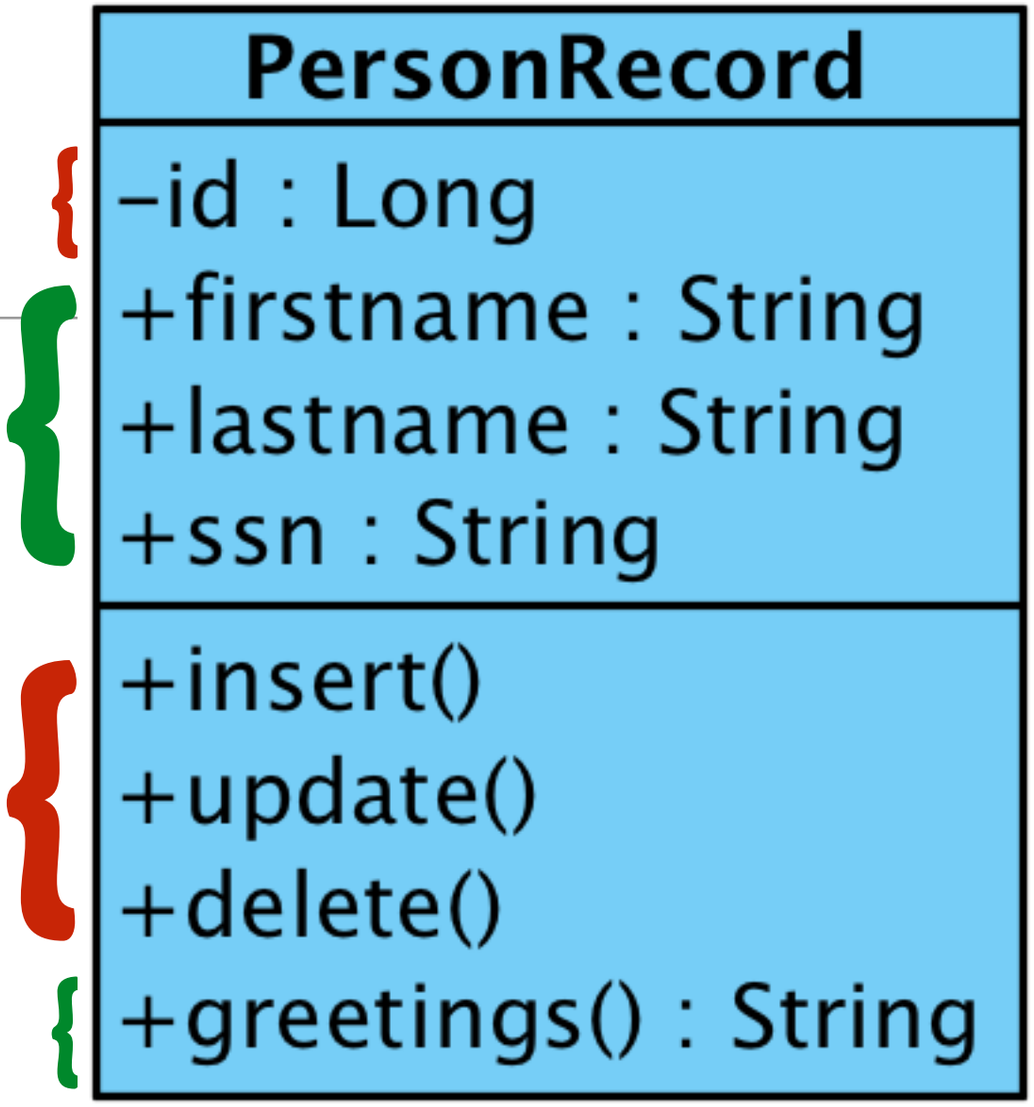


[PoEAA]

Decoupling **Finder** from Gateway



Active Record



Record = Domain ⊕ DB

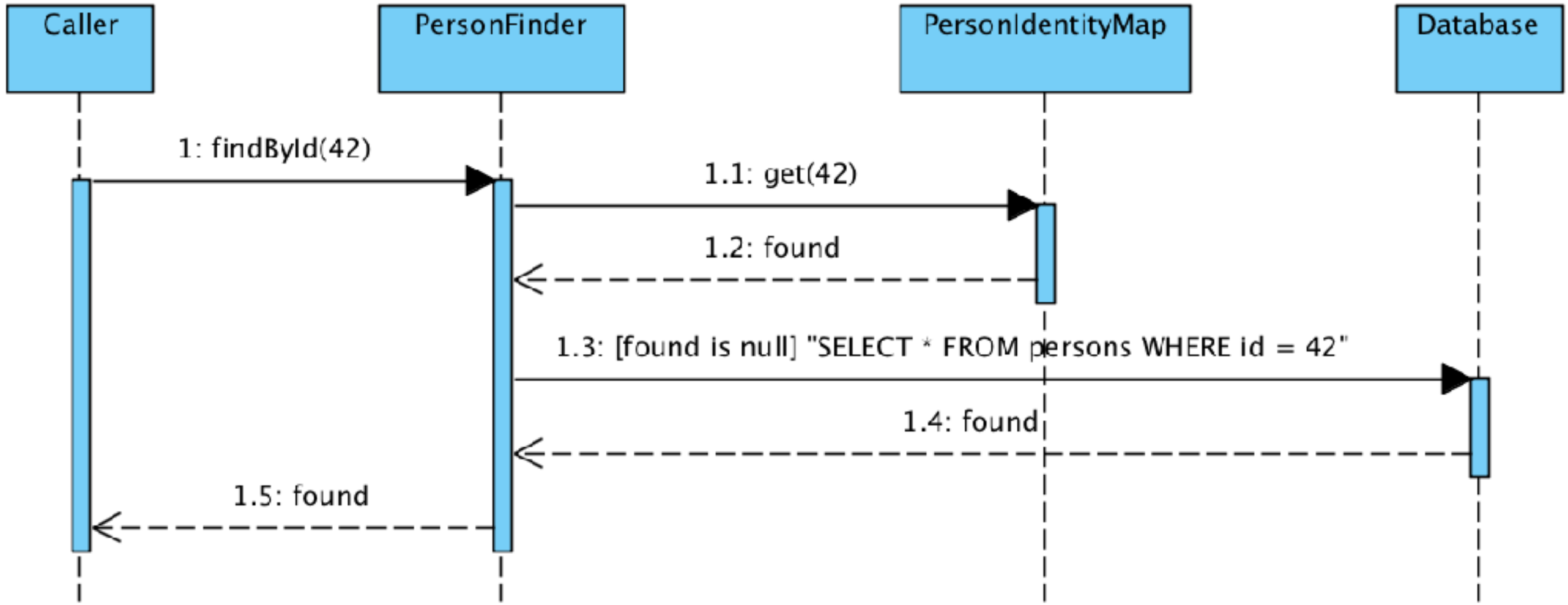
[PoEAA]

Behavioral patterns

3

How to accelerate the access to the persistent layer?

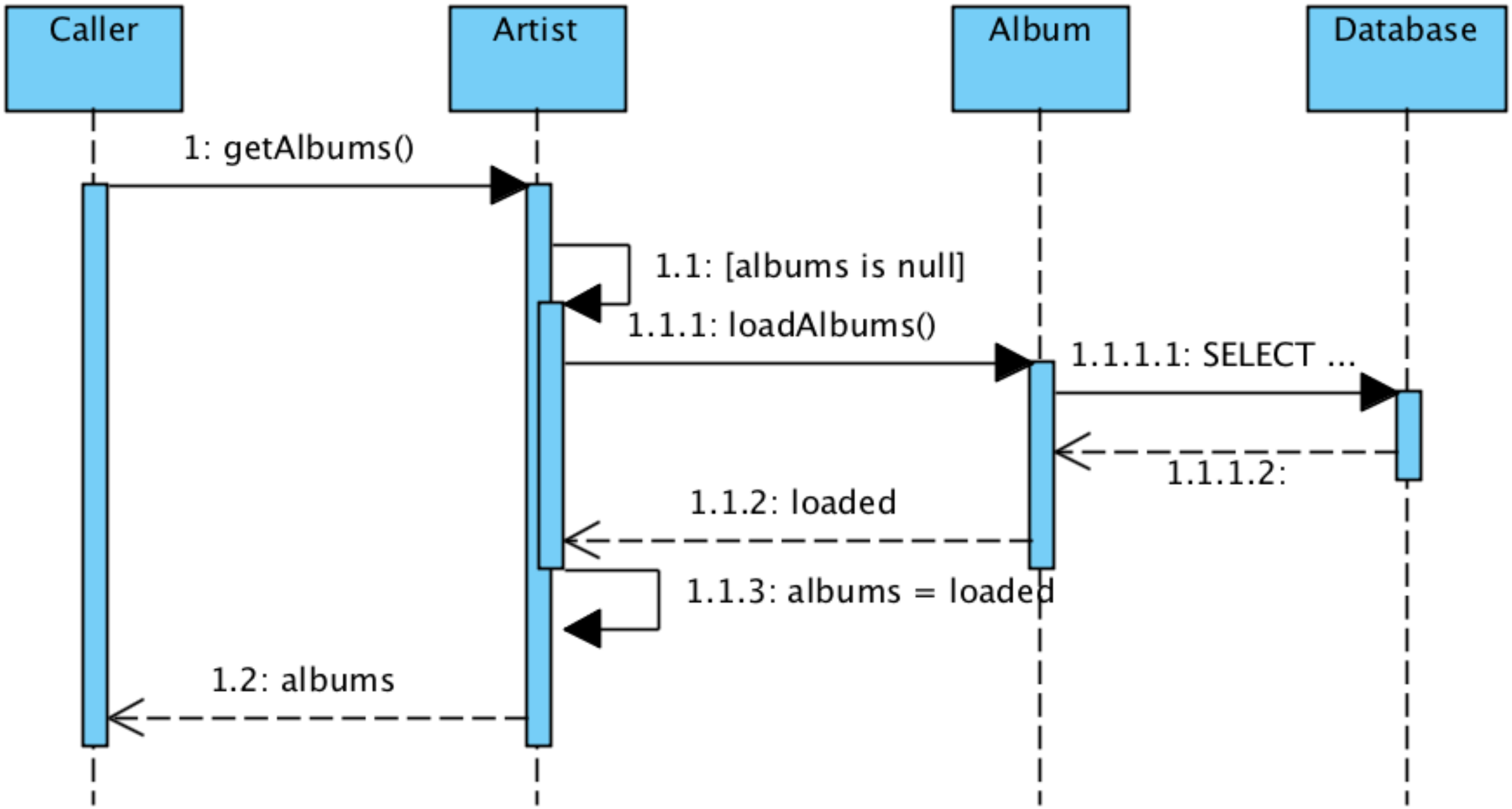
Identity Map



[PoEAA]

How to ease the burden of data loading?

Lazy Loading



[PoEAA]

Conclusions

4

Architecture is a creative process

The same choice will not win each time

Tradeoffs are the key

“If you only have an **hammer**, everything looks like a **nail**