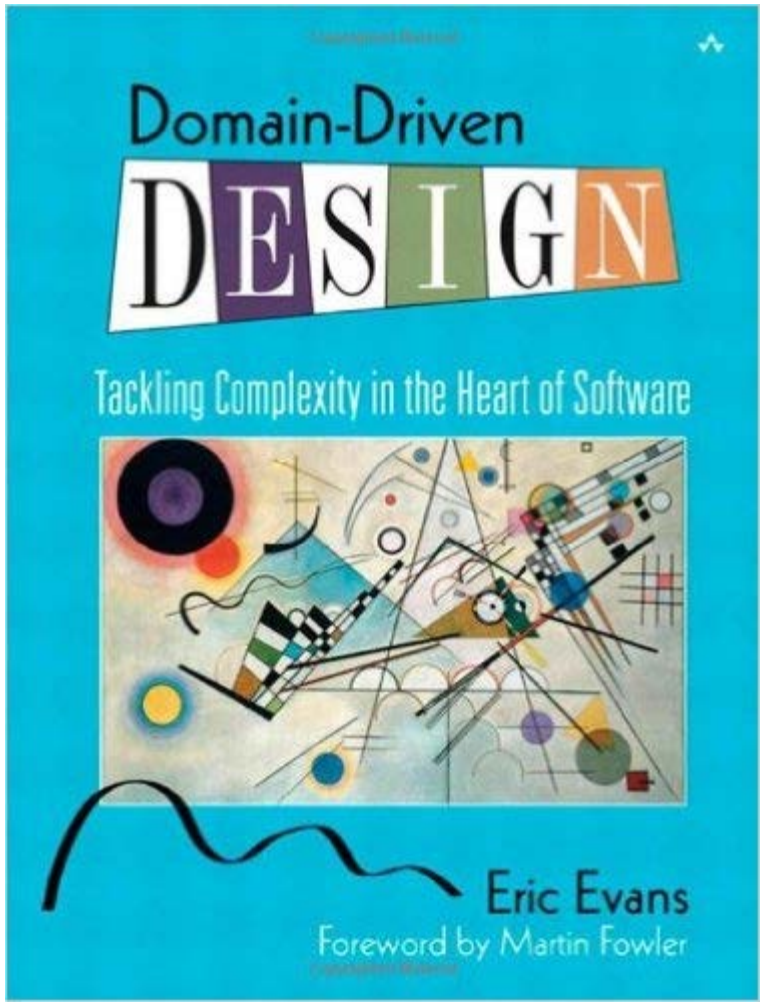
Building Complex Objects with the Specification Pattern

Zoran Horvat

OWNER AT CODING HELMET CONSULTANCY

@zoranh75 www.codinghelmet.com

Specification Design Pattern



”In all kinds of applications,
Boolean test methods appear
that are really parts of little rules.

As long as they are simple,
we handle them
with testing methods.

...

But not all rules are so simple.“

Eric Evans

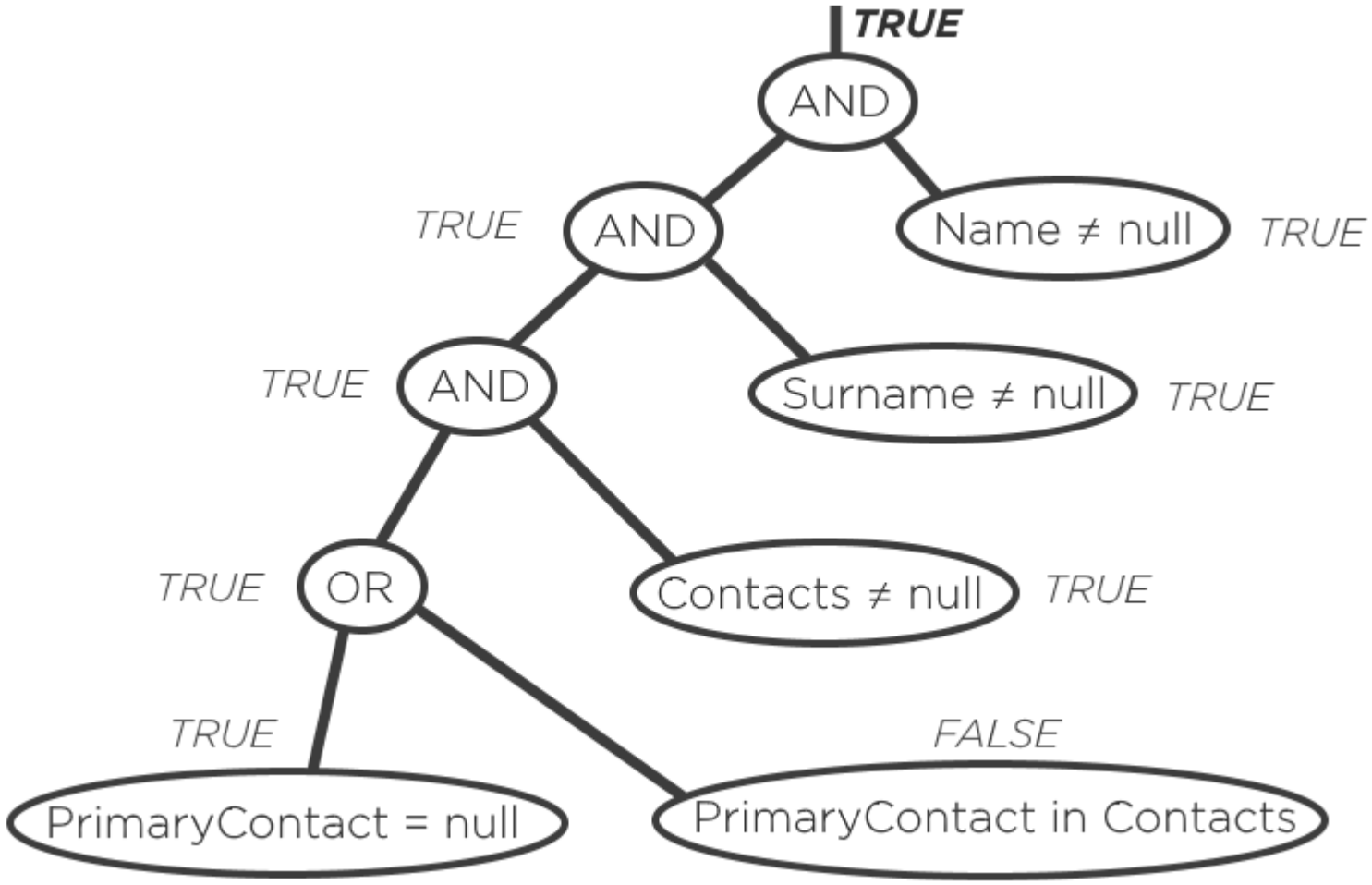
Motivation Behind Specification Pattern

```
class Invoice
{

    internal DateTime DueDate { get; set; }

    public bool IsOverdue()
    {
        DateTime currentDate = DateTime.UtcNow;
        return currentDate > this.DueDate;
    }

    public bool IsDelinquent()
    {
        ...
    }
}
```

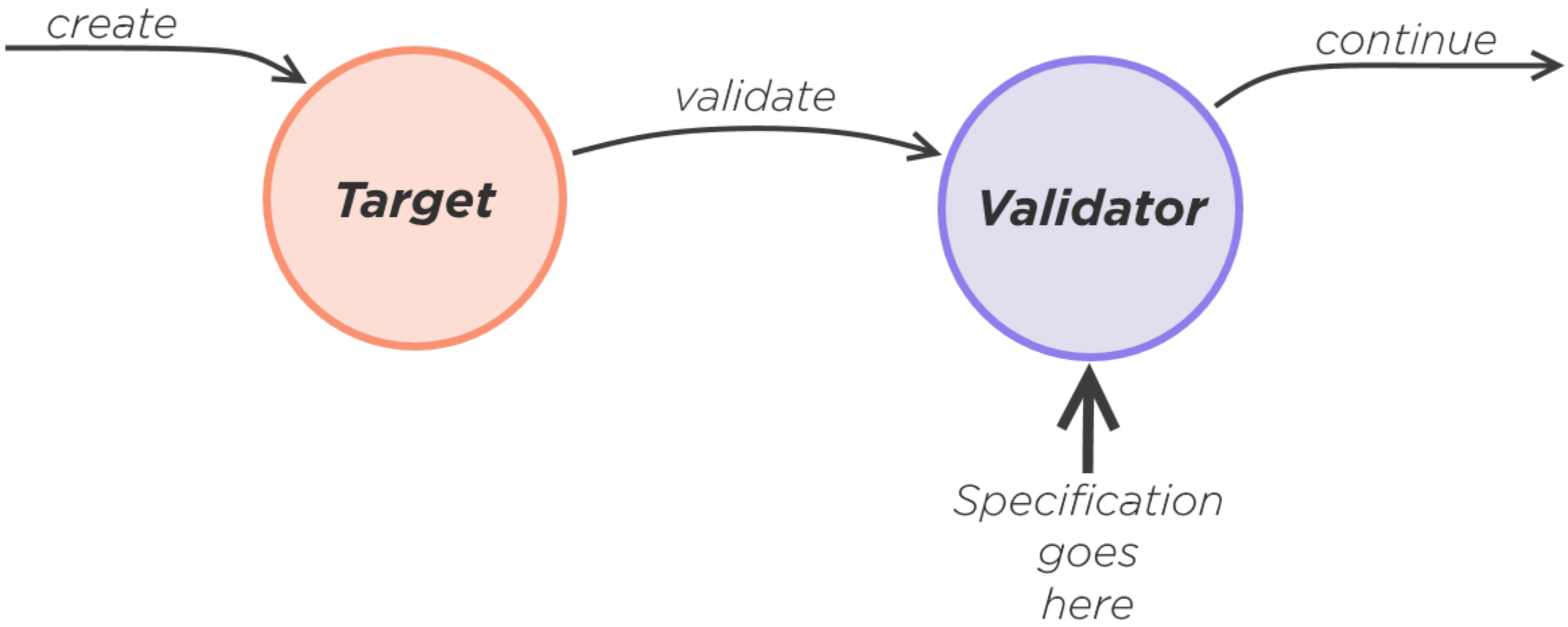
Would have to depend on account, payment history, company policy, etc.

Would have to contacts other objects

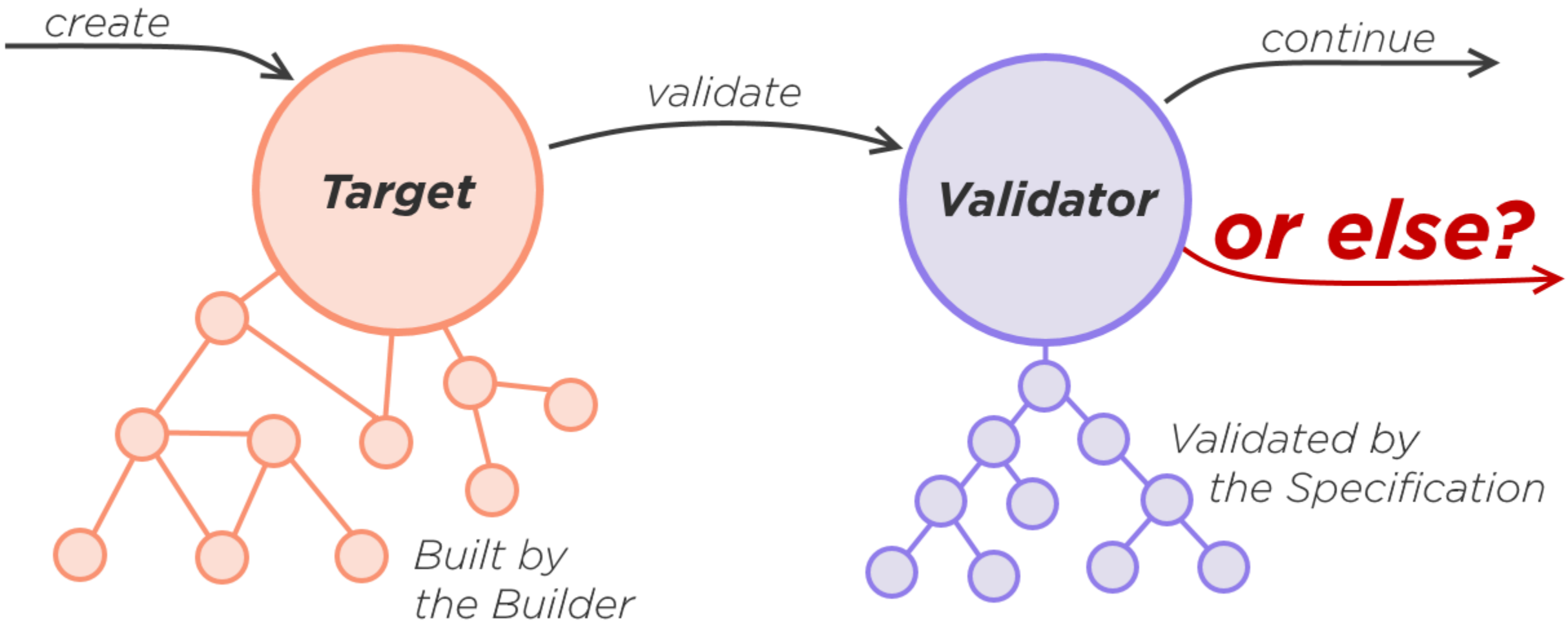
Specification as Expression Tree



The Building Specification Idea



The Building Specification Idea



Persisting Incomplete Objects

Complex object may be the result of a long business transaction

Business transaction may depend on external events

Waiting for the next user's input

Waiting for an outer system

Incomplete object must be persisted before continuing

SELECT SUM(...) ... ***WHERE DoB IS NULL or EmailAddress IS NULL***
SELECT AVERAGE(...) ... ***WHERE DoB IS NULL or EmailAddress IS NULL***

Person			
Name	Surname	DateOfBirth	EmailAddress

Nullable columns

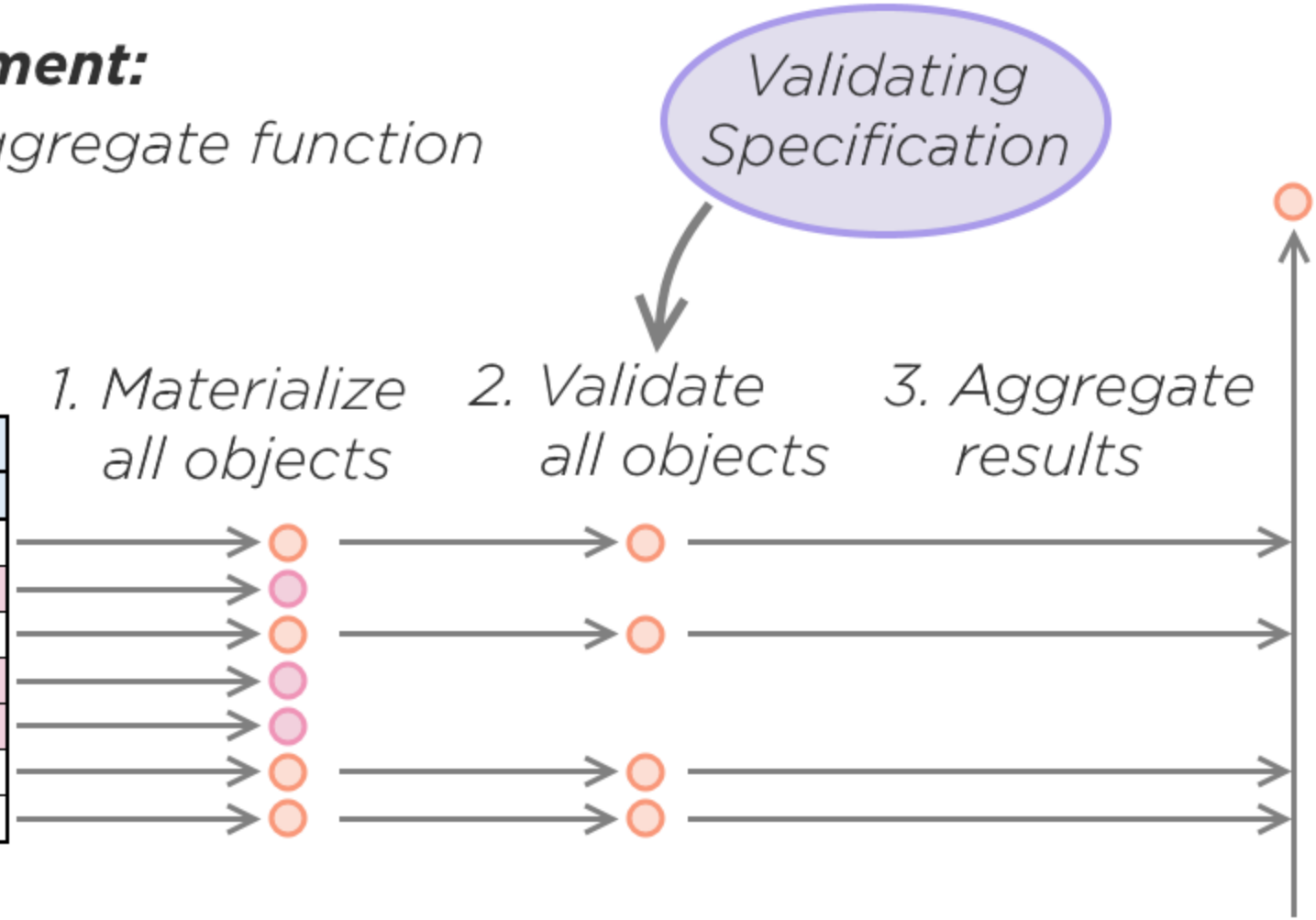
Validating Specification

Person			
Name	Surname	DateOfBirth	EmailAddress

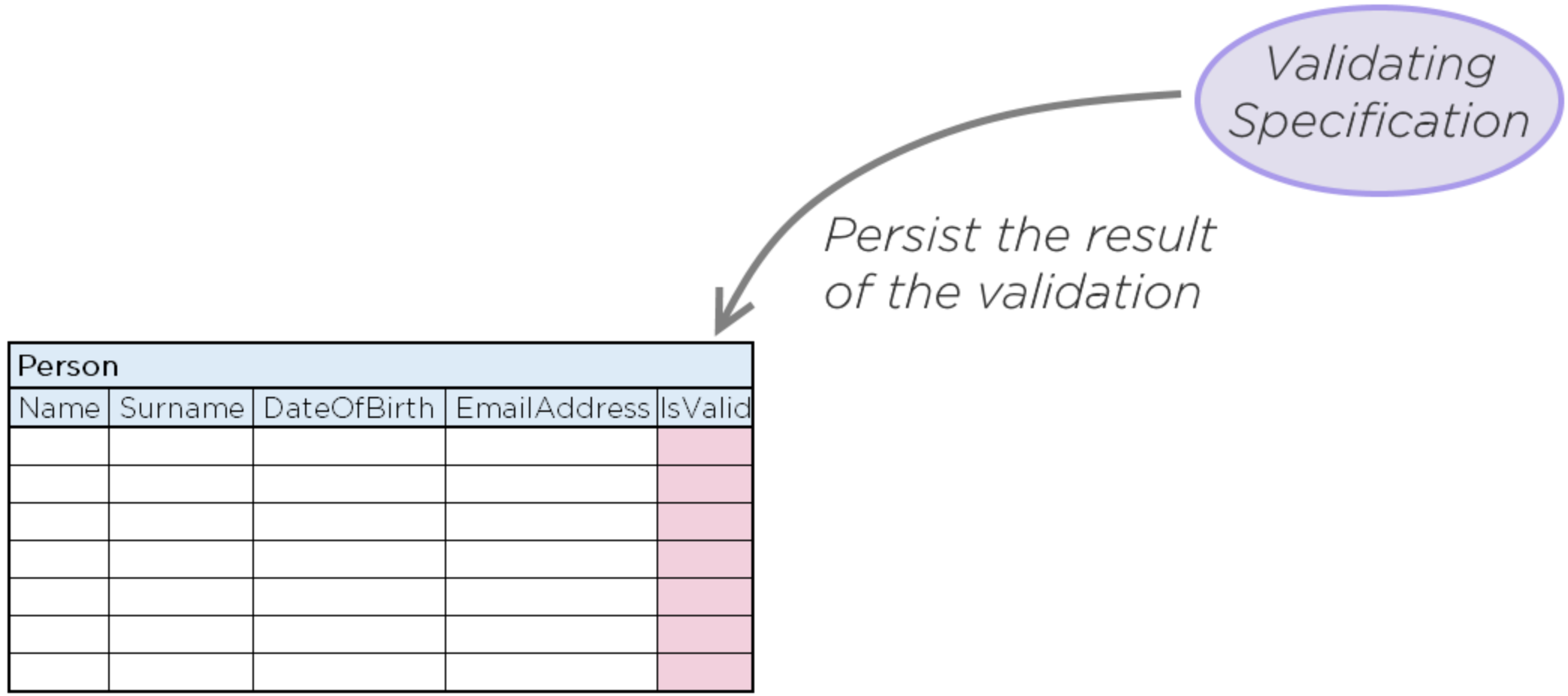
Incomplete objects

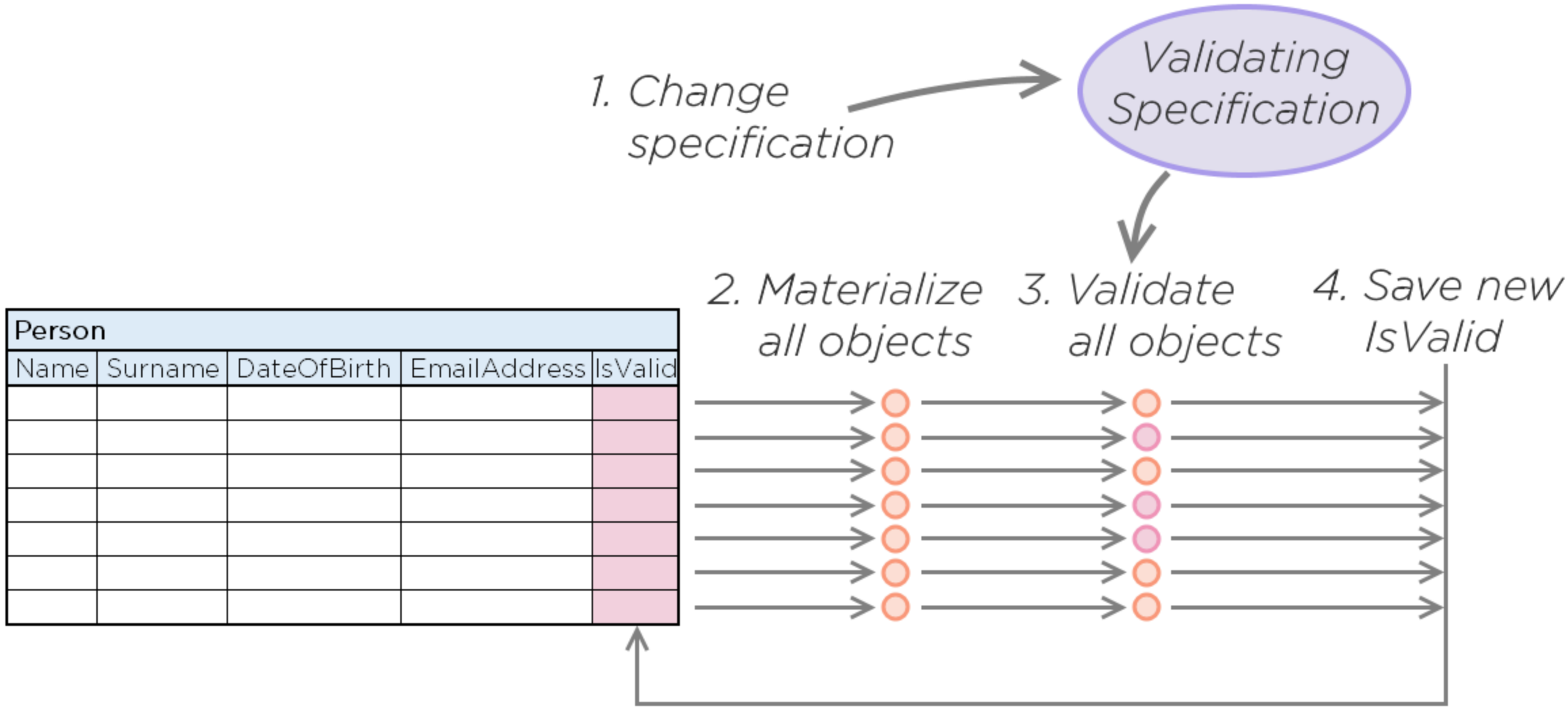
Business requirement:

Calculate some aggregate function

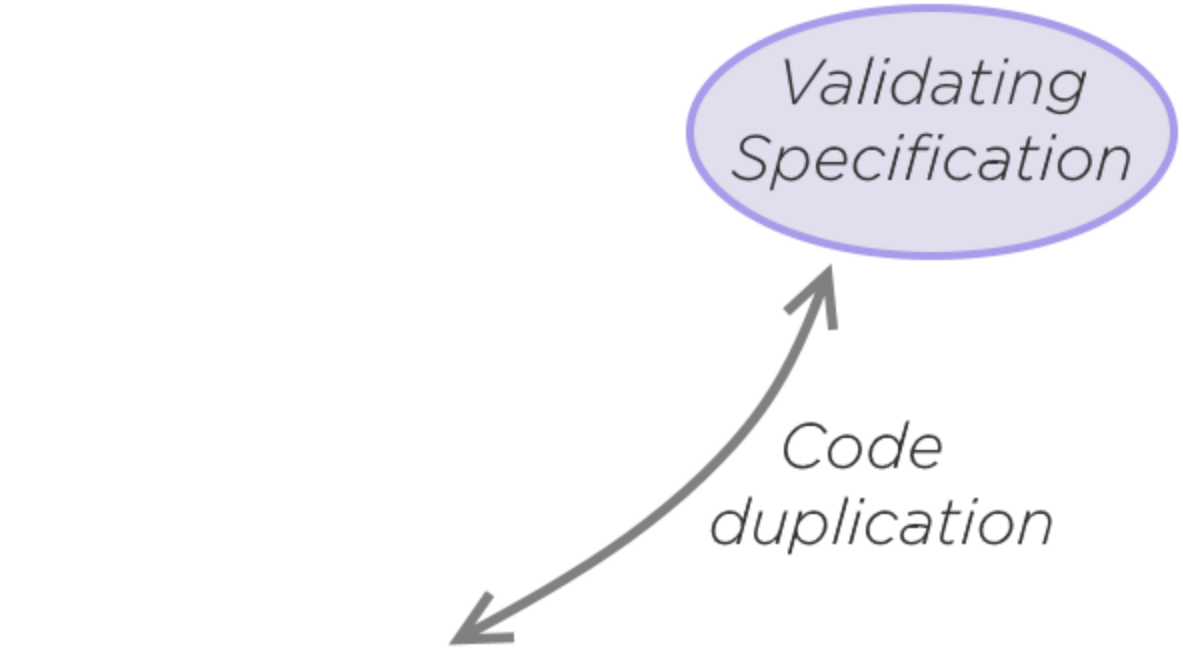


Person			
Name	Surname	DateOfBirth	EmailAddress

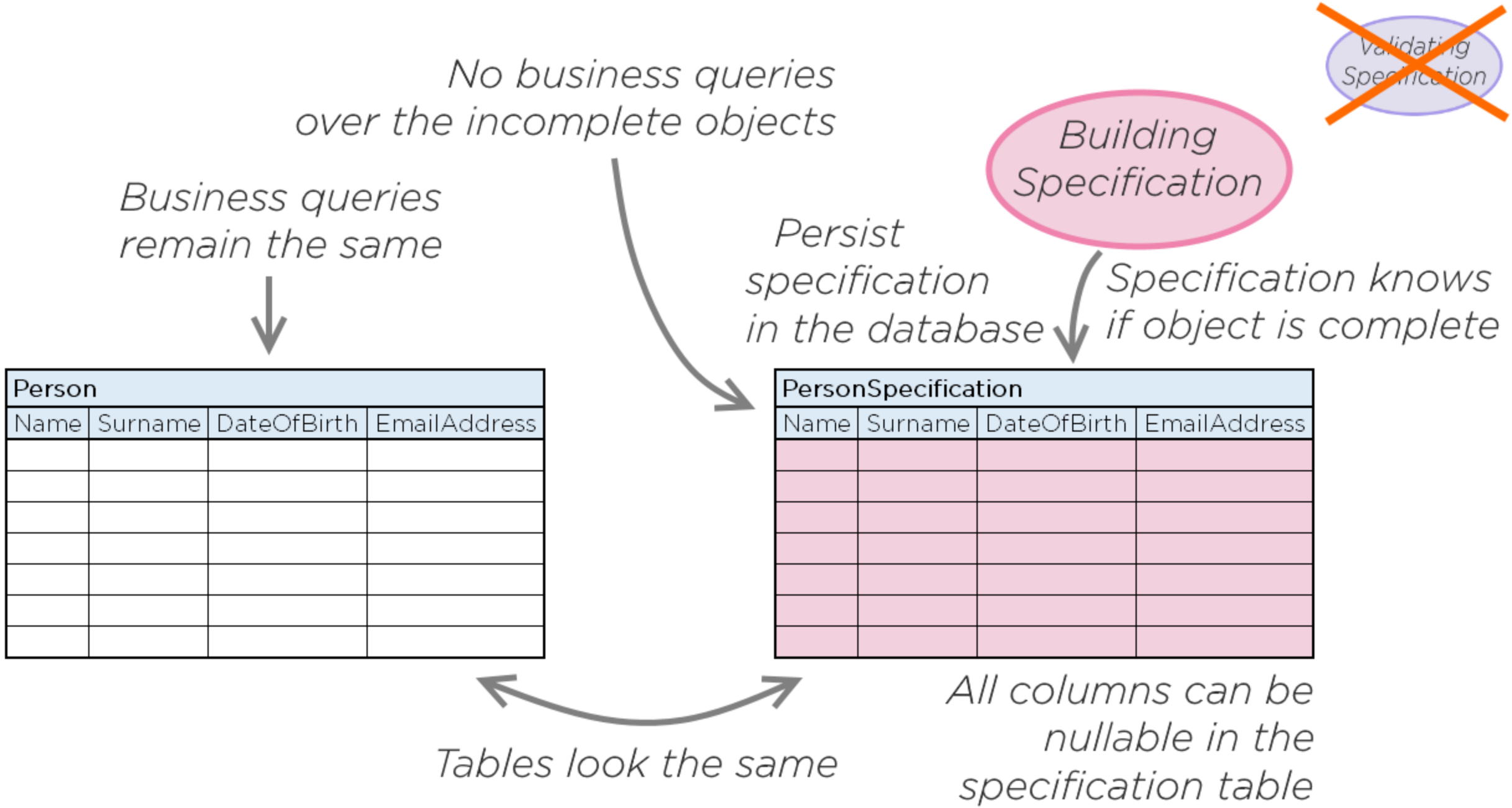

Validating Specification
Persist the result of the validation
Person
Name
Surname
DateOfBirth
EmailAddress
IsValid


1. Change specification
Validating Specification
2. Materialize all objects
3. Validate all objects
4. Save new IsValid
Person
Name
Surname
DateOfBirth
EmailAddress
IsValid

Person			
Name	Surname	DateOfBirth	EmailAddress



+ Validation stored procedures


Validating Specification
No business queries over the incomplete objects
Building Specification
Business queries remain the same
Persist specification in the database
Specification knows if object is complete
Person
Name
Surname
DateOfBirth
EmailAddress
PersonSpecification
Name
Surname
DateOfBirth
EmailAddress
Tables look the same
All columns can be nullable in the specification table

Summary

Specification design pattern

- Wraps Boolean test
- Tells whether an object is valid or not

Object validation problem

- One object can be valid in one context
- Same object can be invalid in another context
- Boolean specifications deal well with validation

Summary

An attempt to unite two concepts

- Constructing a complex object
- Validating a complex object

Building Specification pattern

- Organized like a multi-level Builder
- Builder receives parts of the product
- Building Specification receives specifications of future parts
- Building Specification forms a tree
- Construction process is recursive

Next module –
Building Object Graphs with the Specification Pattern