

# Aggregates in DDD

A Critical Pattern for Managing Complexity in the Domain

Julie Lerman  
TheDataFarm.com  
@julielerman



Steve Smith  
Ardalis.com  
@ardalis

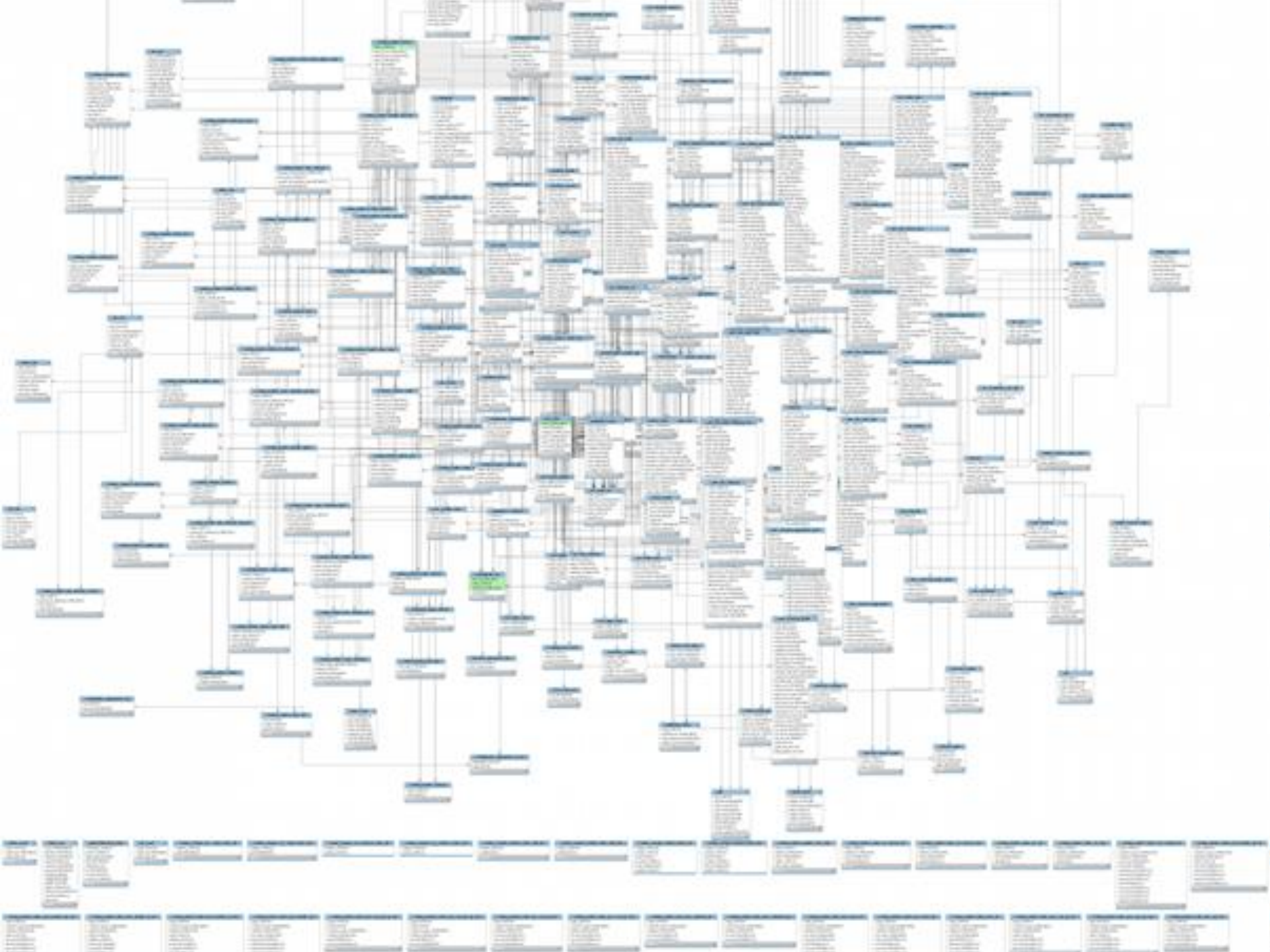


**pluralsight**   
hardcore dev and IT training

# **In This Module**

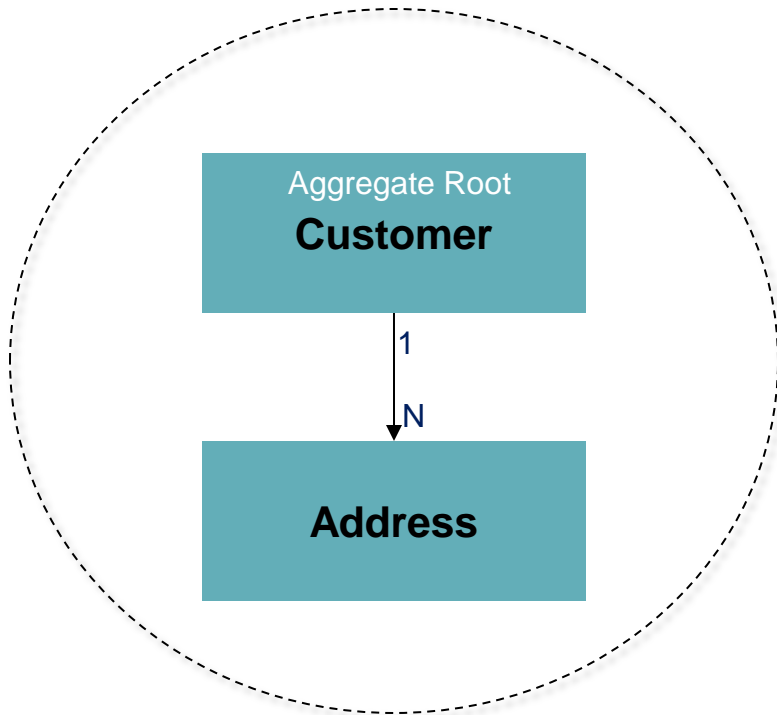
- **Aggregates**
- **Aggregate Roots**
- **Invariants**
- **Shifting our Design to a Smarter Aggregate**
- **Implementing Aggregates in Code**

# Tackling Data Complexity

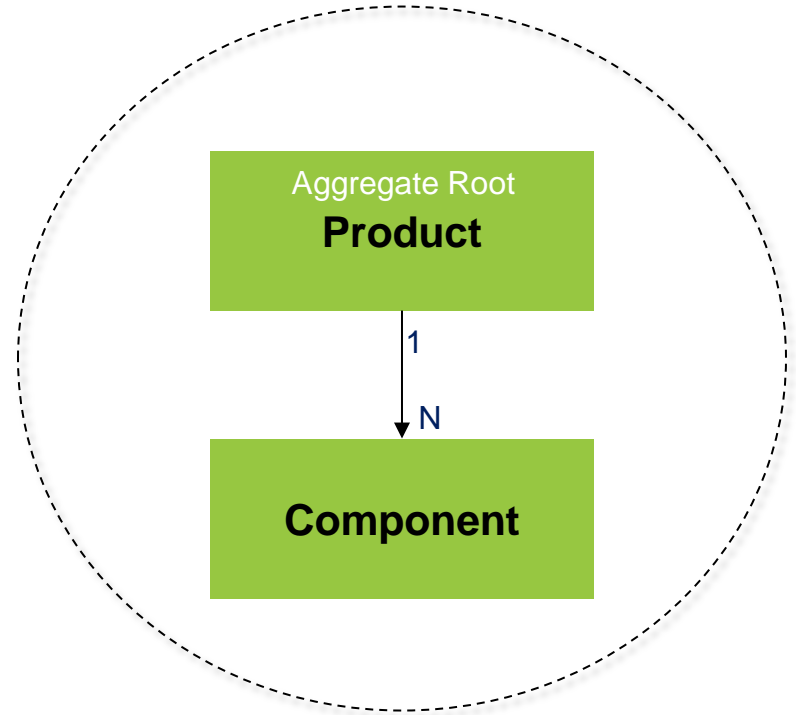


# Aggregates

Customer Aggregate

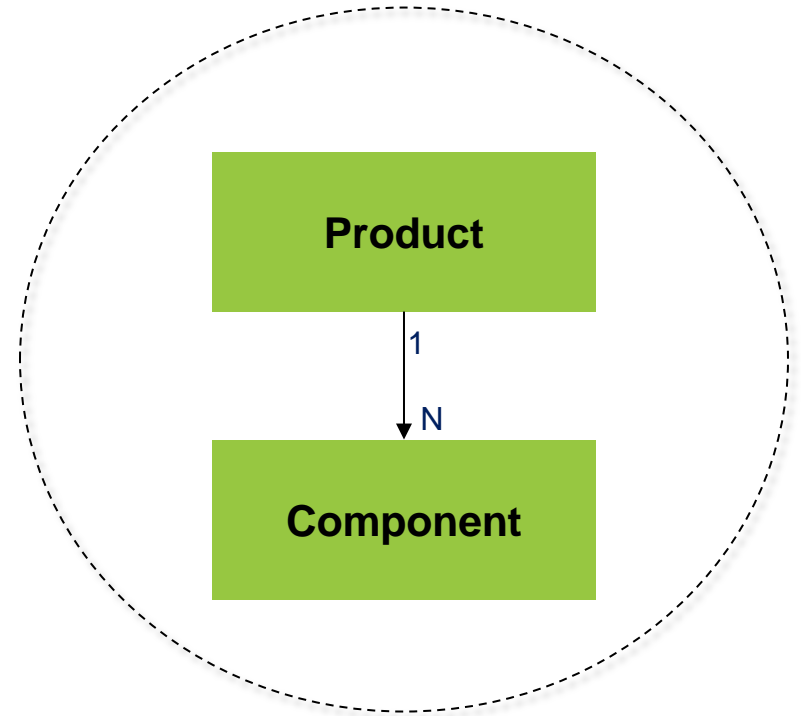
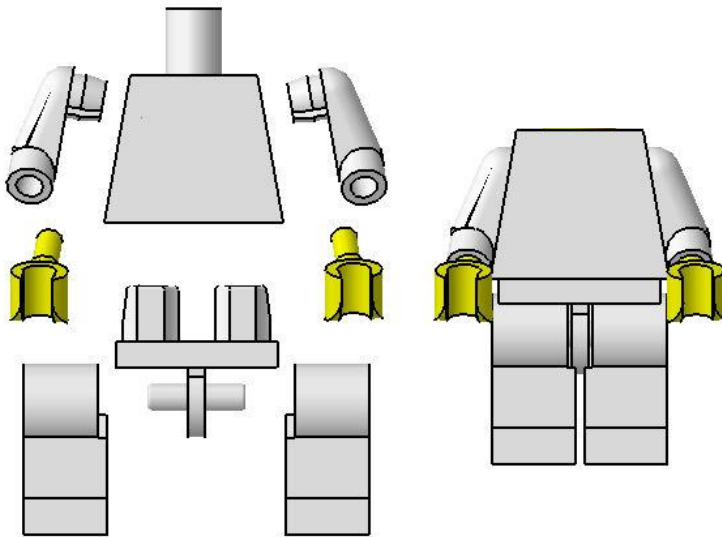


Product Aggregate



# Aggregates

## Product Aggregate



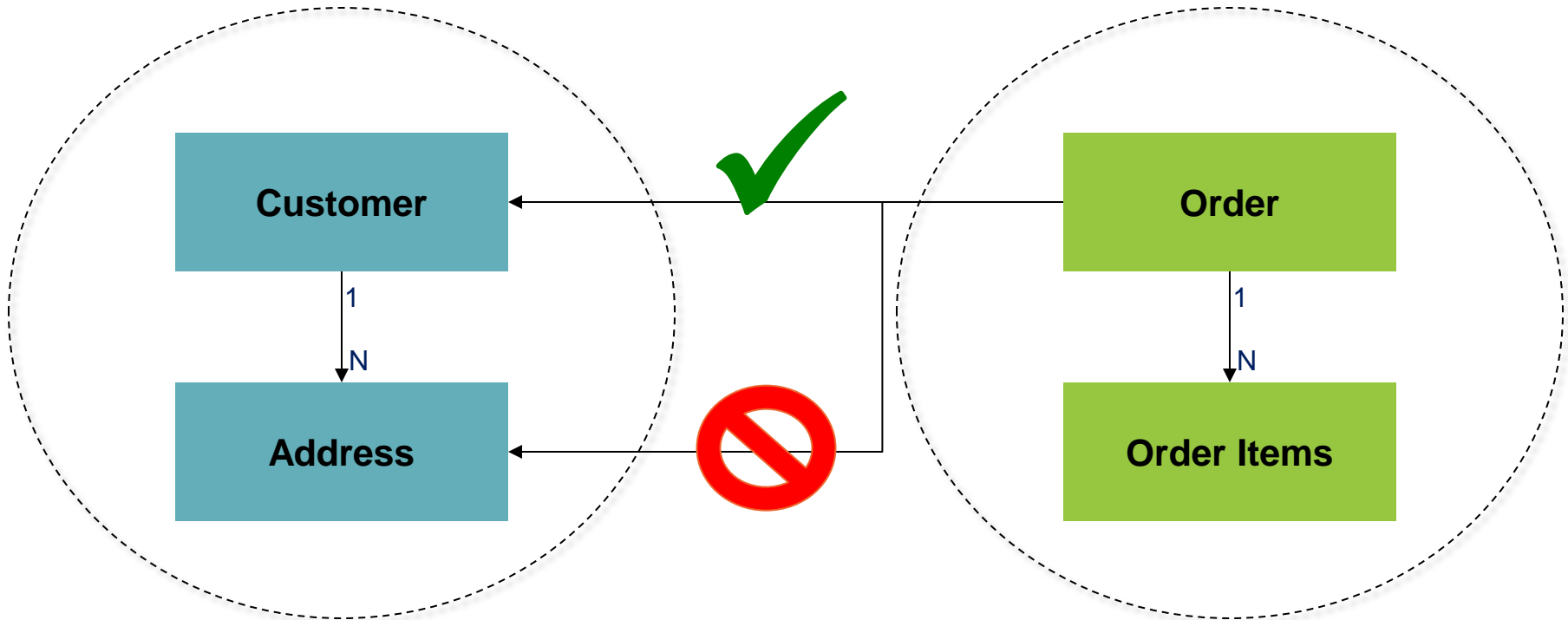
“An **aggregate** is a cluster of associated objects that we treat as a unit for the purpose of data changes.”

— **Eric Evans**  
Domain-Driven Design

# Relationships Between Aggregates

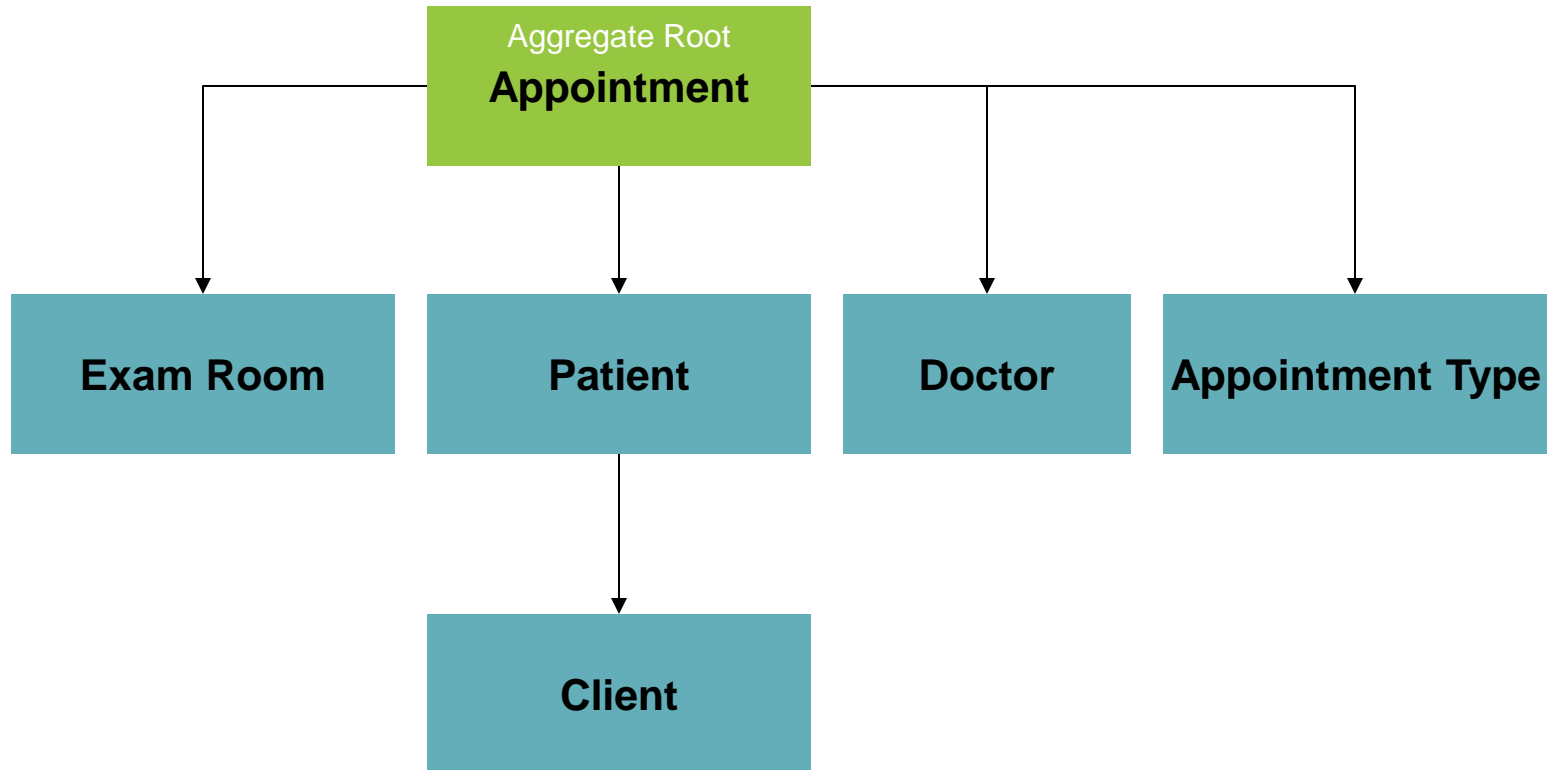
Customer Aggregate

Order Aggregate





# Our Design (draft)



# Our Design (revised)

## Appointment



### Properties



AppointmentTypeId



ClientId



DateTimeConfirmed



DoctorId



IsPotentiallyConflicting



PatientId



RoomId

**Client**

**Exam Room**

**Patient**

**Doctor**

**Appointment Type**

## Invariants

The Speed of Light

*1079252849 km/h*

*670,616,629 mph*

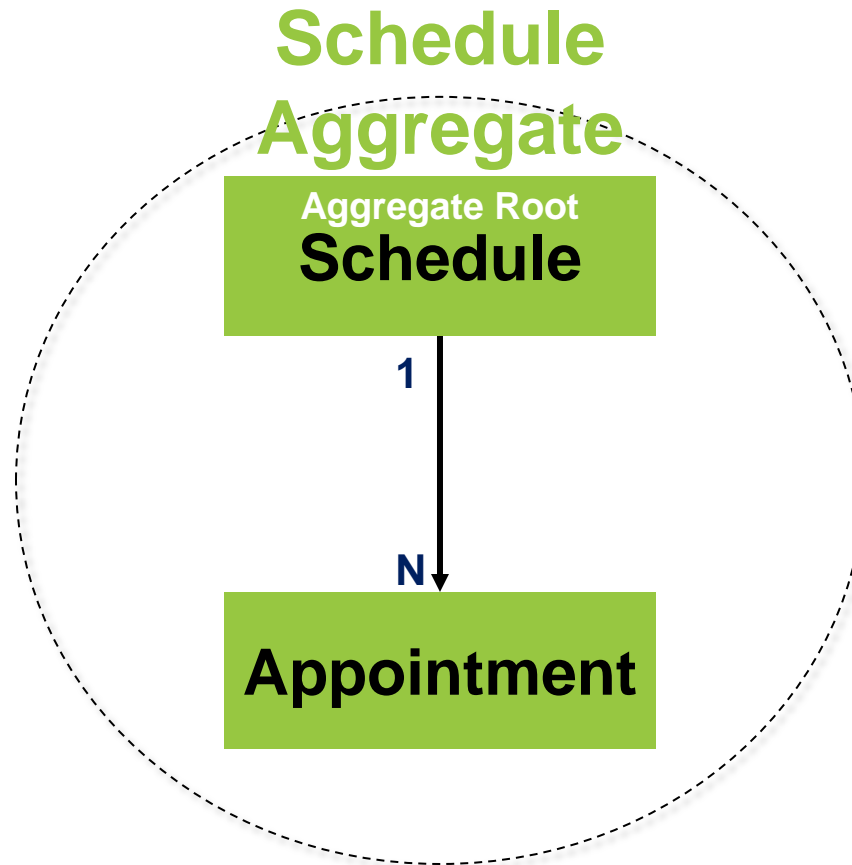
# Examples of Aggregate **Invariants**

Total items on  
purchase order  
**do not exceed limit**

Two appointments  
**do not overlap**  
one another

End date  
**follows**  
Begin date

# Our Design (revised)



## **Aggregates in our Application**

# Aggregate Tips

Aggregates are **not always** the answer!

Aggregates can **connect only by the root**

Don't overlook using **FKs for non-root entities**

**Too many FKs** to non-root entities may suggest a problem

**"Aggregates of one"** are acceptable

**"Rule of Cascading Deletes"**

# Glossary of **Terms** from this Module

## **Aggregate**

A transactional graph of objects

## **Aggregate Root**

The entry point of an **aggregate** which ensures the integrity of the entire graph

## **Invariant**

A condition that should always be true for the system to be in a consistent state

## **Persistence Ignorant Classes**

Classes that have no knowledge about how they are persisted



# References

## Books

Domain-Driven Design <http://amzn.to/1kstiRg>

Implementing Domain-Driven Design <http://amzn.to/1dgYRY3>

## Web

Eric Evan's website [DomainLanguage.com](http://DomainLanguage.com)

DDD Community website [DDDCommunity.org](http://DDDCommunity.org)

## On Pluralsight:

Entity Framework in the Enterprise – [bit.ly/PS-EFEnterprise](http://bit.ly/PS-EFEnterprise)

SOLID Principles of OO Design - [bit.ly/solid-smith](http://bit.ly/solid-smith)

# Thanks!

**Julie Lerman**

**TheDataFarm.com**

**Twitter: @julielerman**

**Steve Smith**

**Ardalis.com**

**Twitter: @ardalis**

***To Teach Is To Learn Twice***

**pluralsight**  
hardcore developer training

