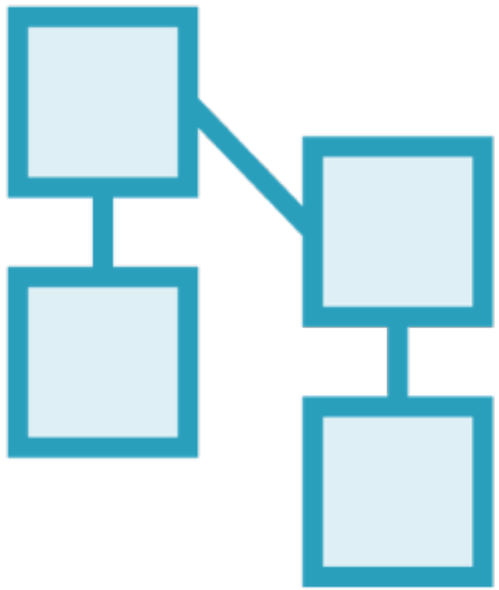
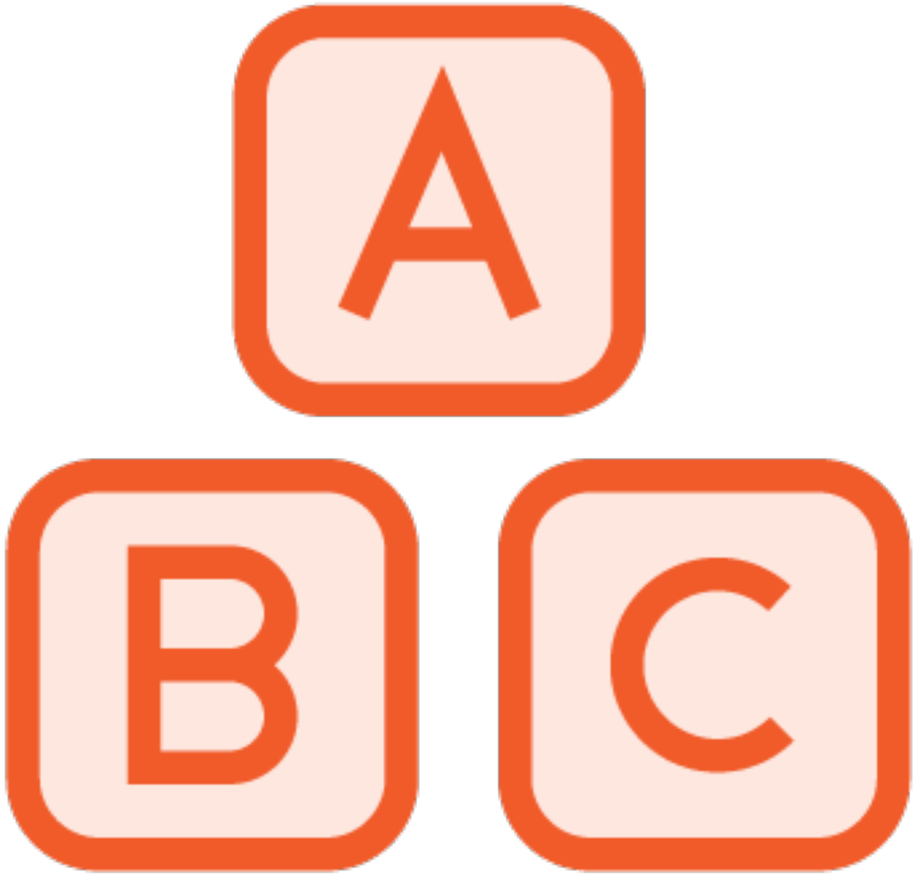
Understanding Data Model Terminology



Core Data stores structured data

Schema

- Entities
- Attributes
- Relationships


A
B
C

Data Model Building Blocks

Entities

Nouns of the data model

- People
- Places
- Things
- Ideas

Entities for ShoutOut App

ShoutOut

Represents the concept of a digital “job well-done” message

Employee

Represents the concept of an employee at our company

Attributes

Entities are composed of Attributes

Describe an Entity

Give Entities their meaning

ShoutOut

- message
- from
- sentDate
- shoutCategory

Good Work!

Awesome Job!

Well Done!

Attributes and Data Types

Integer 16

Integer 32

Integer 64

Decimal

Double

Float

String

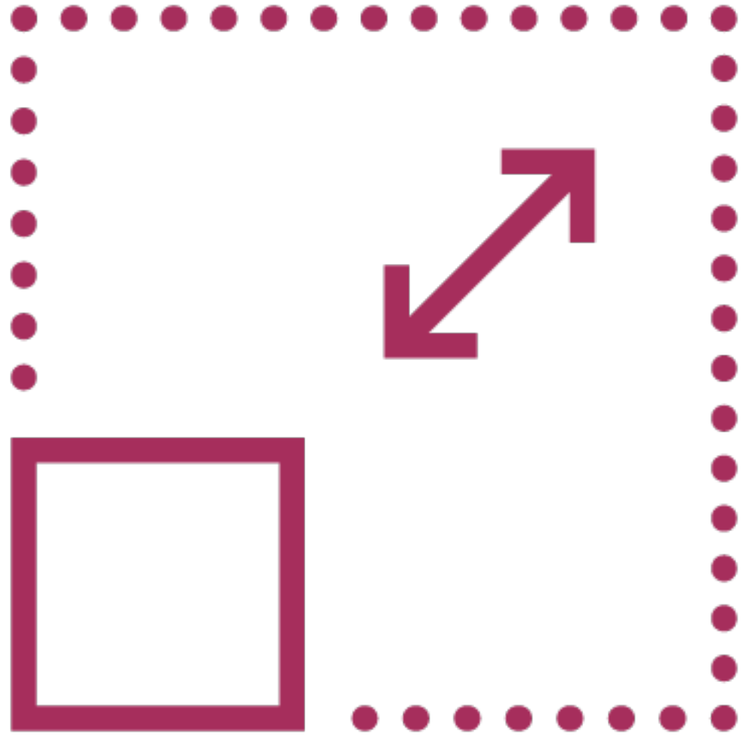
Boolean

Date

Binary

Transformable

NSCoding



Value Transformer

ShoutOut Attributes

ShoutOut Entity

message

Type	Optional	Indexed
string	yes	no

from

Type	Optional	Indexed
string	yes	no

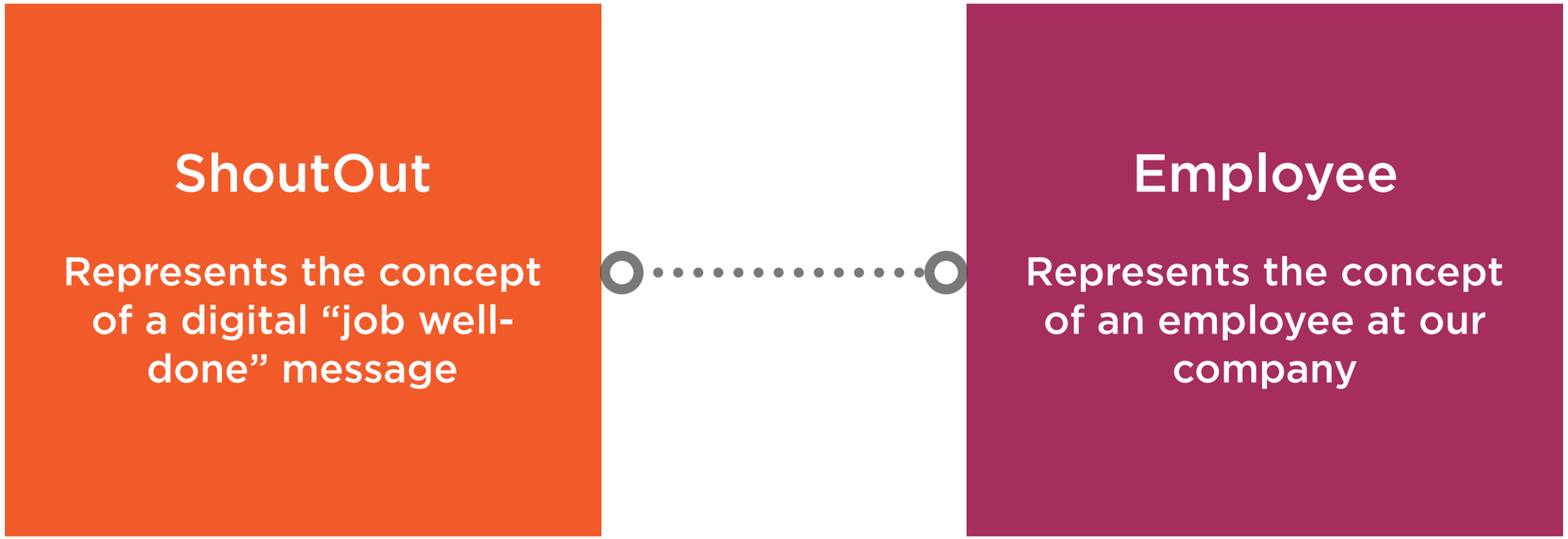
shoutCategory

Type	Optional	Indexed
string	no	yes

sentDate

Type	Optional	Indexed
date	yes	no

Relationships Between Entities



Demo

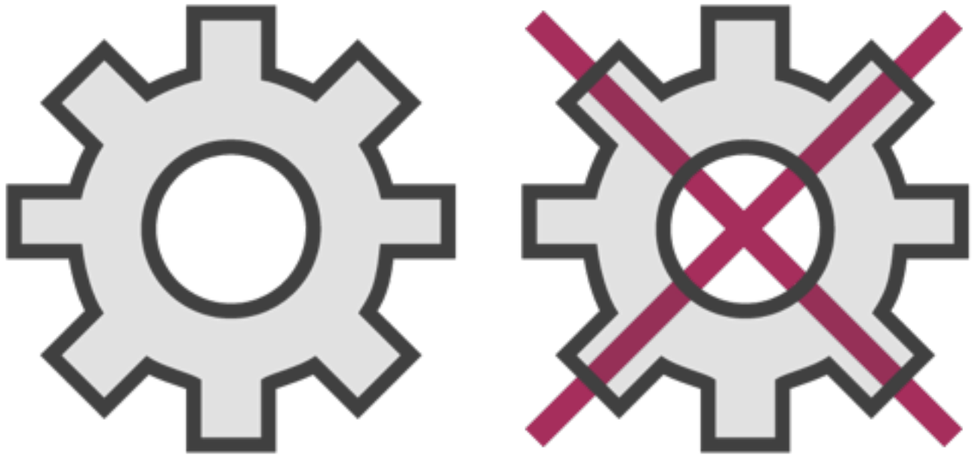
Create ShoutOut Entity

Add Attributes

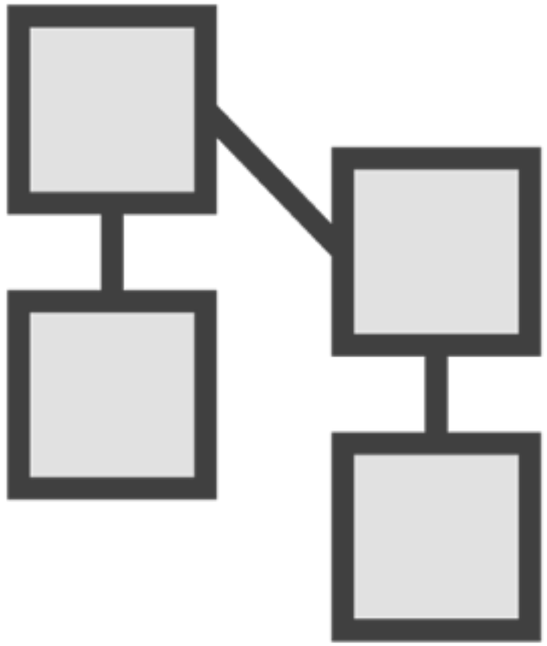
Tour Xcode's data model designer

```
let fetchRequest = NSFetchRequest<ShoutOut>(. . .)
```

ShoutOut
Employee

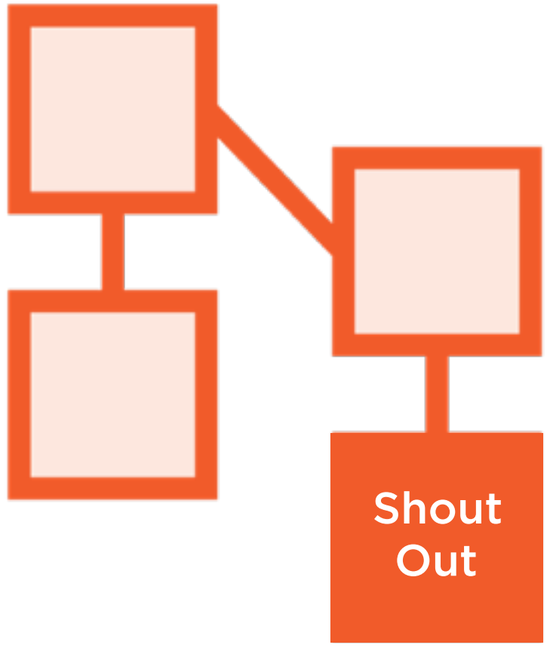


Multiple
Configurations



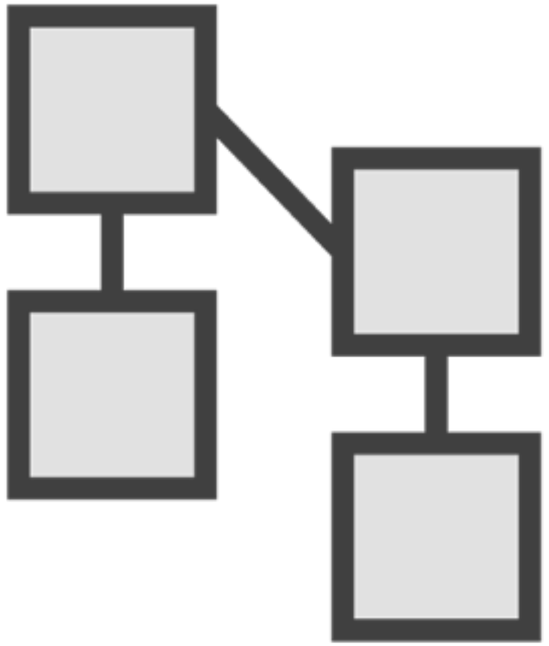
Empty Data Model

(Version 1)

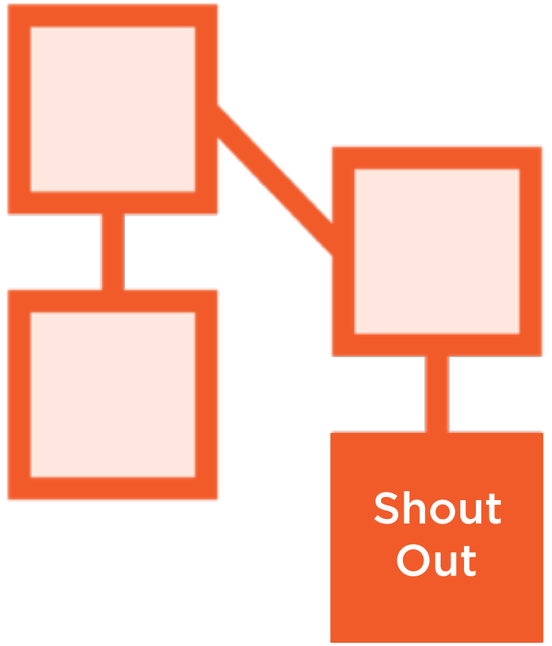


Data Model w/ ShoutOut

(Version 2)



Migration



Empty Data Model

(Version 1)

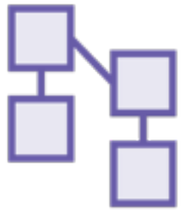
Data Model w/ ShoutOut

(Version 2)

Representing Entities in Code

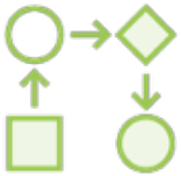
The Core Data Stack

Model



NSManagedObjectModel

Manage



NSManagedObject

NSManagedObjectContext

Persist

NSPersistentStoreCoordinator

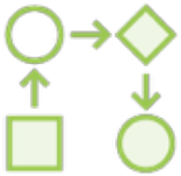
NSPersistentStore

The Core Data Stack

Model

NSManagedObjectModel

Manage



NSManagedObject

NSManagedObjectContext

Persist

NSPersistentStoreCoordinator

NSPersistentStore

NSManagedObject

Implements basic behavior required of a Core Data model object

- Change tracking
- Validation
- Lifecycle events

Only NSManagedObject and its subclasses can be used with NSManagedObjectContext

Custom subclasses not required, but helpful

Advantages of NSManagedObject Subclasses

Type safety

Convenience

Advantages of NSManagedObject Subclasses

Type safety

Convenience

Custom logic

- **entityName property**