

Creating a Data Model



Andrew Bancroft

@andrewcbancroft www.andrewcbancroft.com

Overview

Create a data model with Xcode

Build an Entity

Configure an Entity's Attributes

First look at Relationships

**Represent an Entity in code as an
NSManagedObject subclass**

Understanding Data Model Terminology



Core Data stores structured data

Schema

- Entities
- Attributes
- Relationships

A

B

C

Data Model Building Blocks



The diagram consists of three adjacent rectangular blocks arranged horizontally. The first block on the left is orange and contains the word 'Entity'. The middle block is a dark purple color and contains the word 'Attribute'. The third block on the right is a light green color and contains the word 'Relationship'. All three blocks are of equal height and width, and they are separated by small gaps.

Entity

Attribute

Relationship

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

Decimal

Integer 32

Double

Integer 64

Float

.....

String

Binary

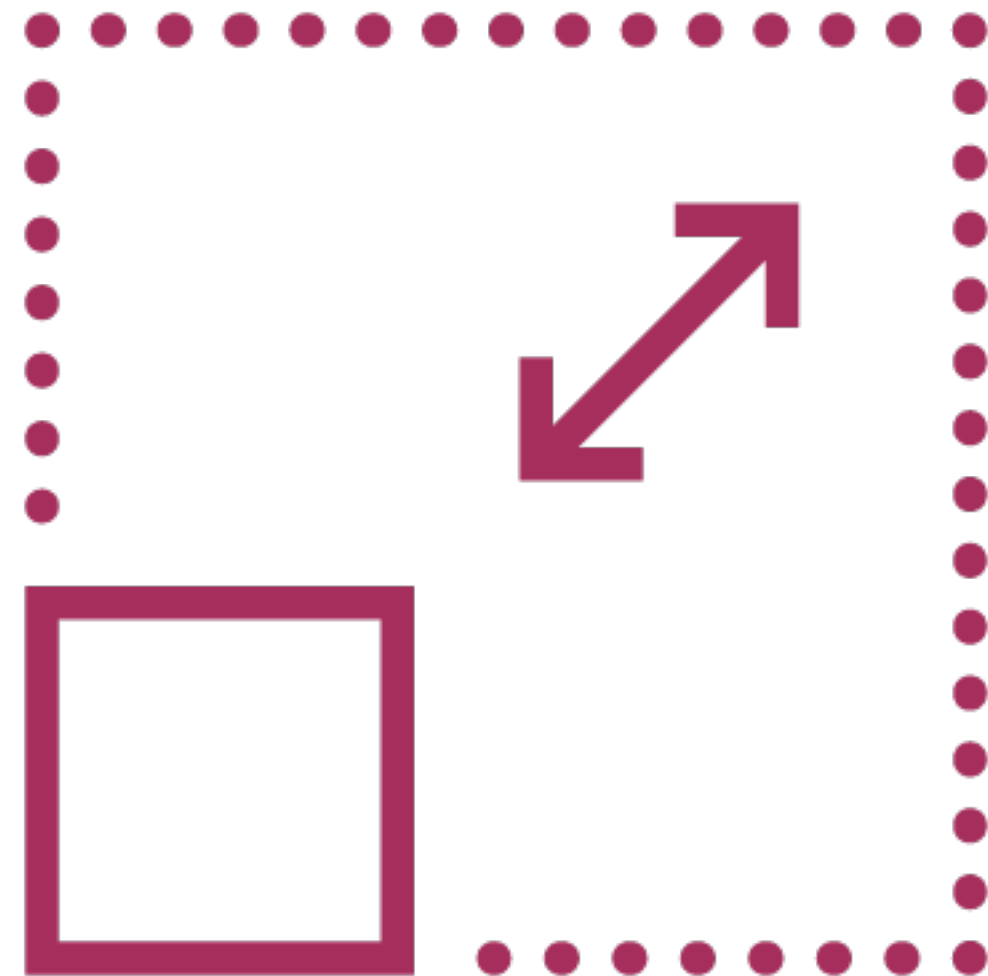
Boolean

Transformable

Date



NSCoding



Value Transformer

ShoutOut Attributes

ShoutOut Entity

message

<u>Type</u>	<u>Optional</u>	<u>Indexed</u>
string	yes	no

from

<u>Type</u>	<u>Optional</u>	<u>Indexed</u>
string	yes	no

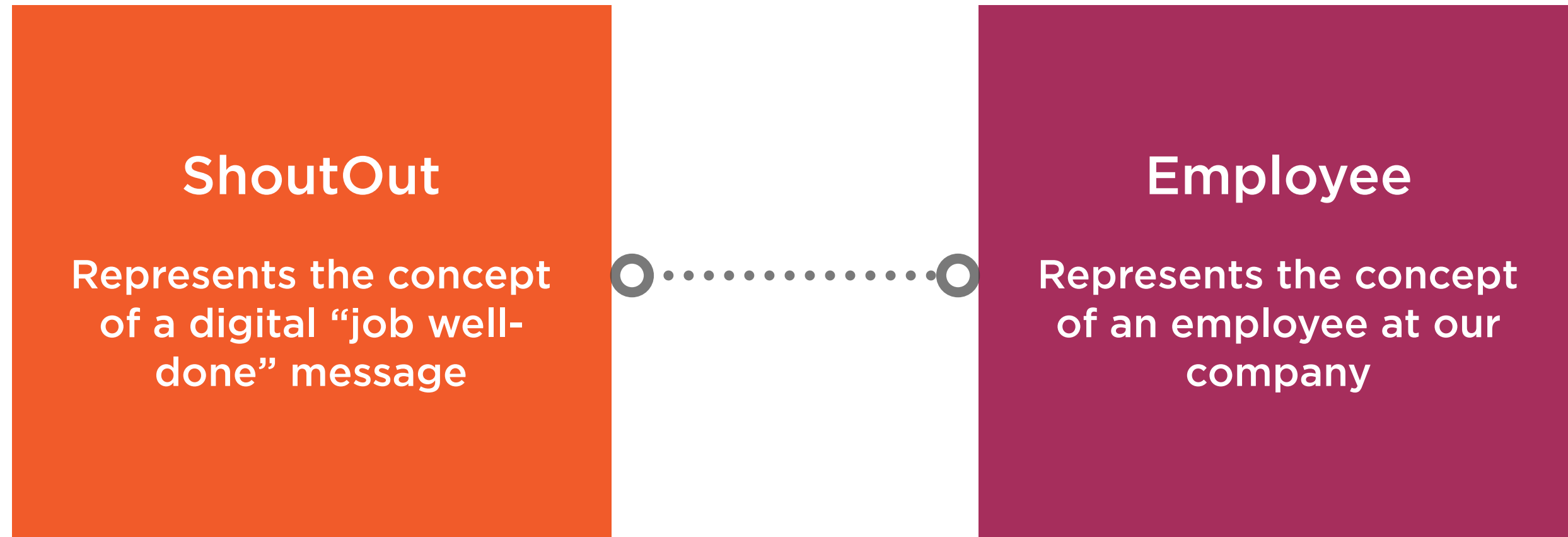
shoutCategory

<u>Type</u>	<u>Optional</u>	<u>Indexed</u>
string	no	yes

sentDate

<u>Type</u>	<u>Optional</u>	<u>Indexed</u>
date	yes	no

Relationships Between Entities



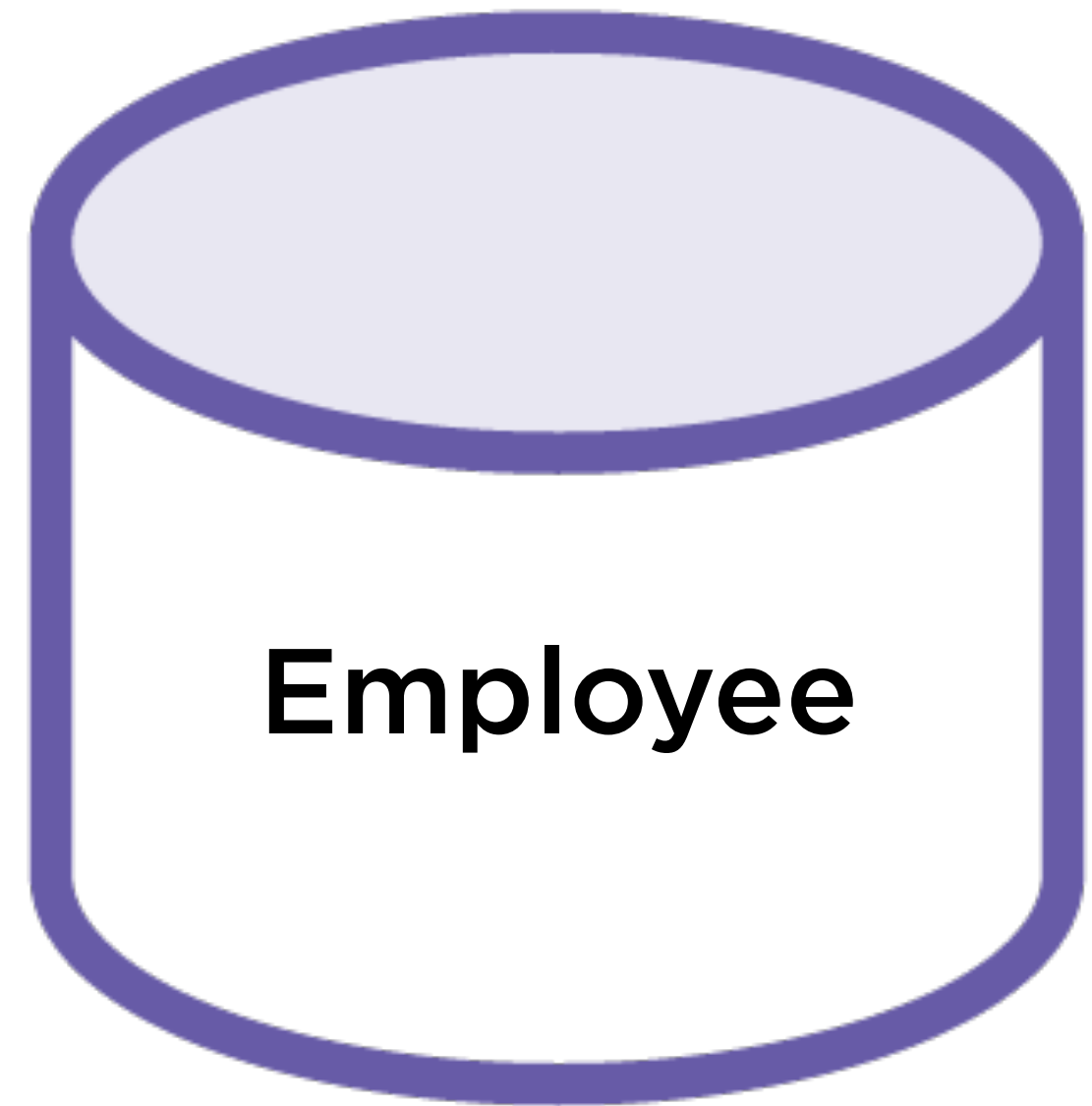
Demo

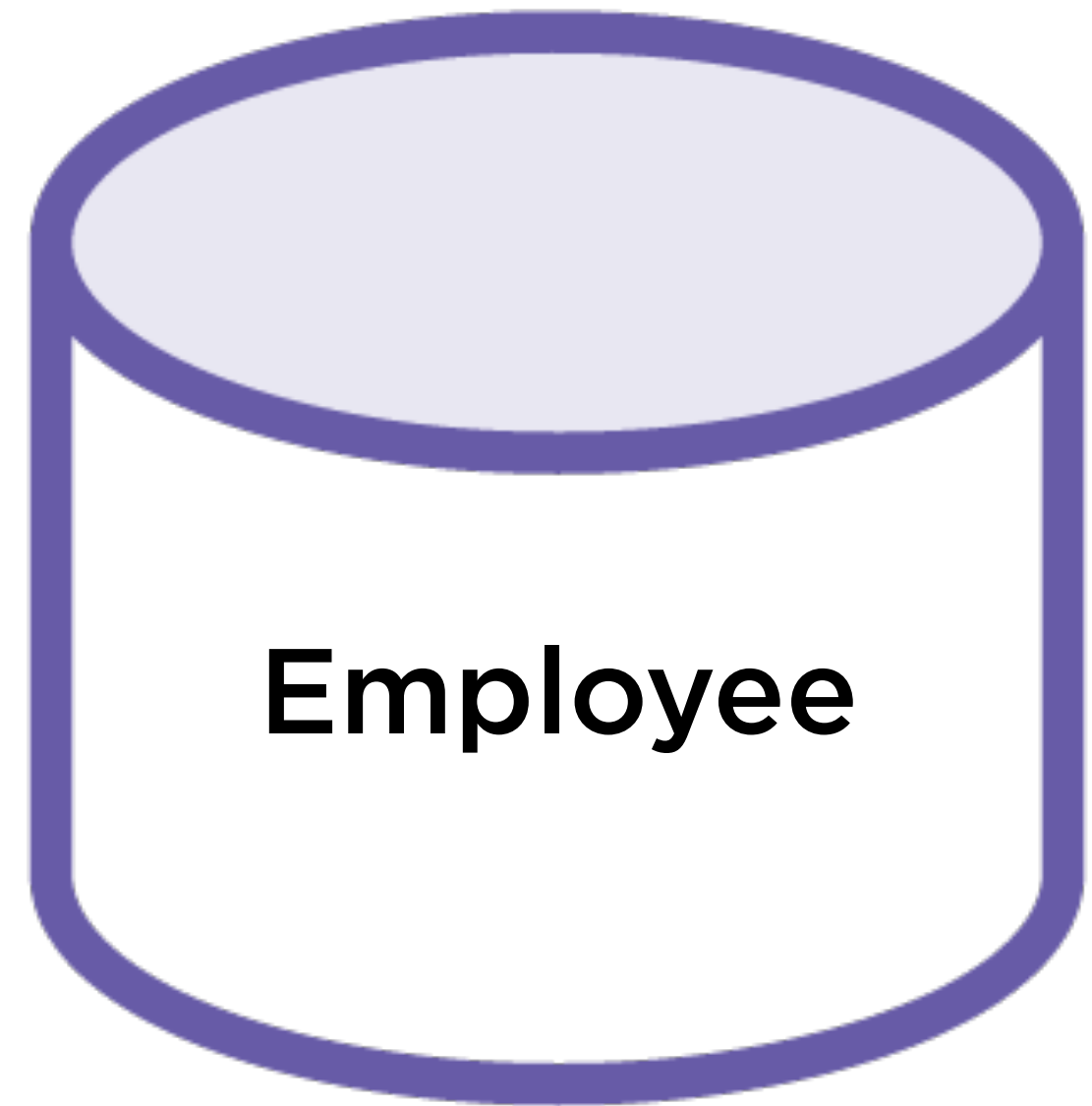
Create ShoutOut Entity

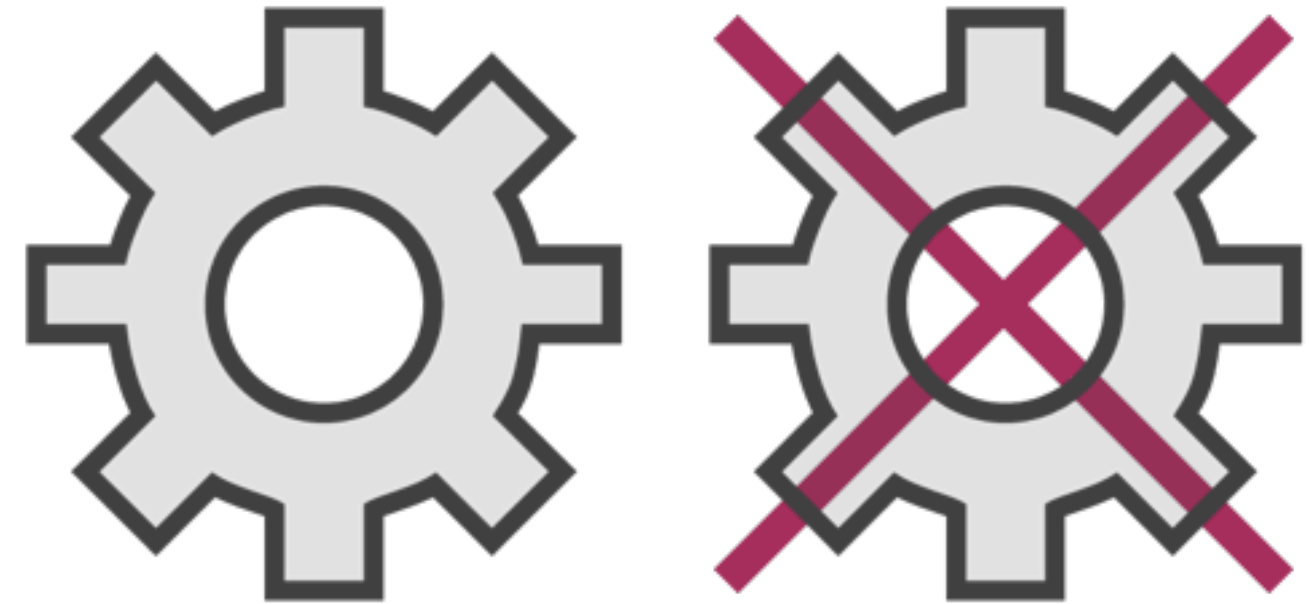
Add Attributes

Tour Xcode's data model designer

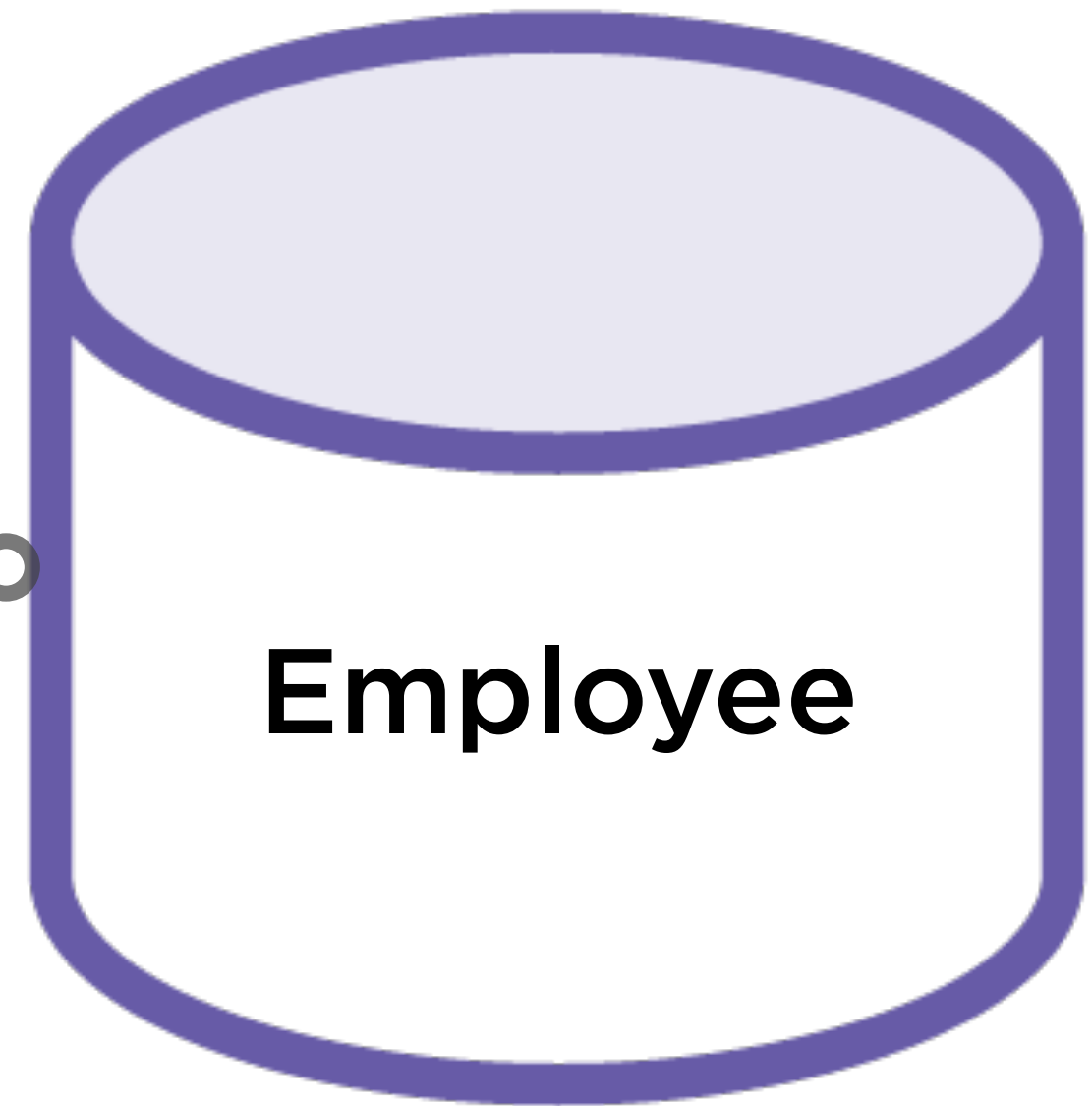

```
let fetchRequest = NSFetchRequest<ShoutOut>(. . .)
```

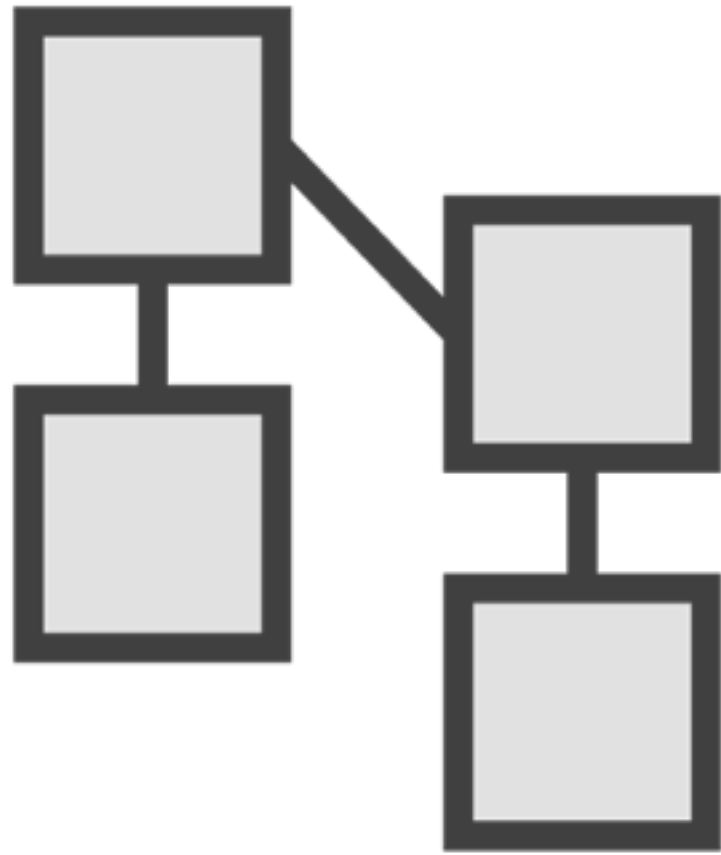




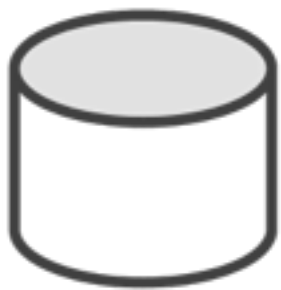


**Multiple
Configurations**

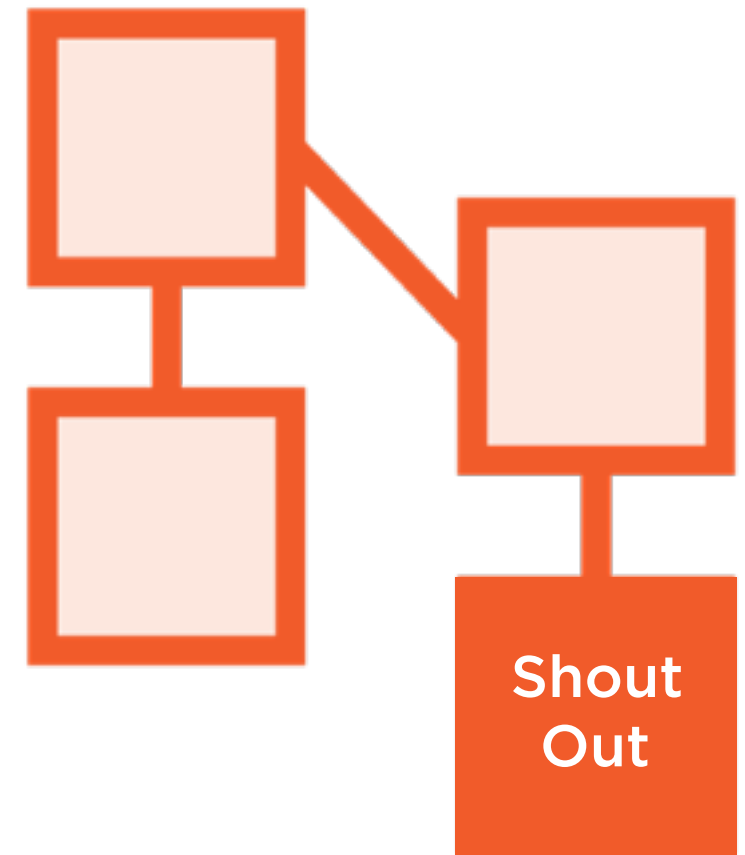
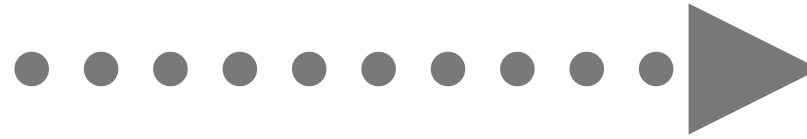




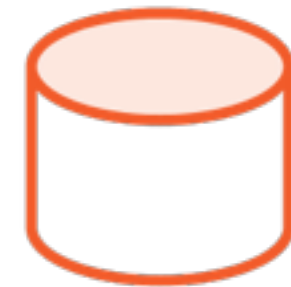
Empty Data Model



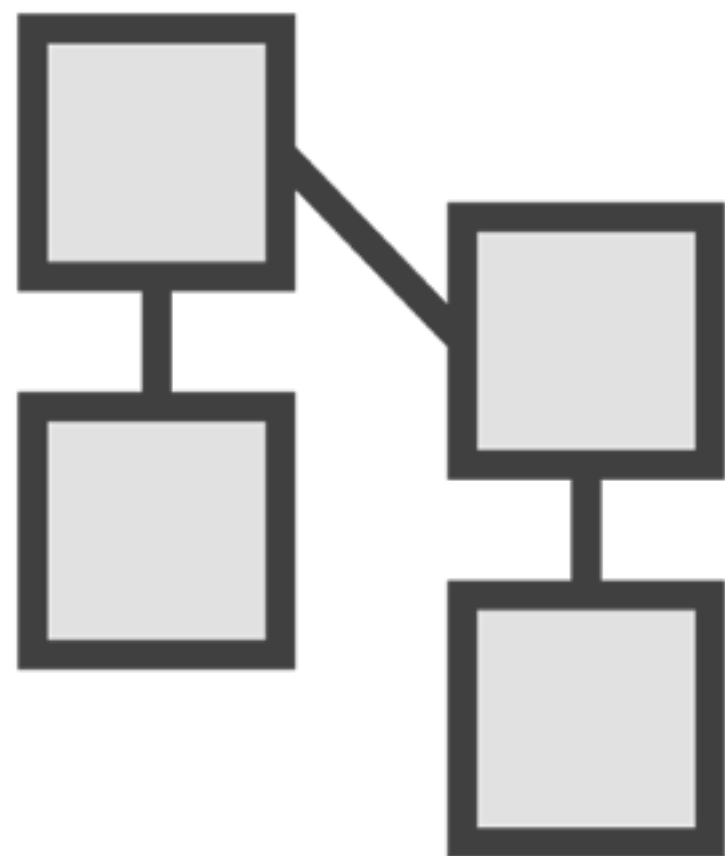
(Version 1)



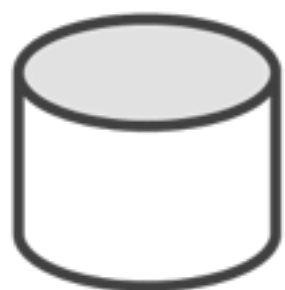
Data Model w/ ShoutOut



(Version 2)

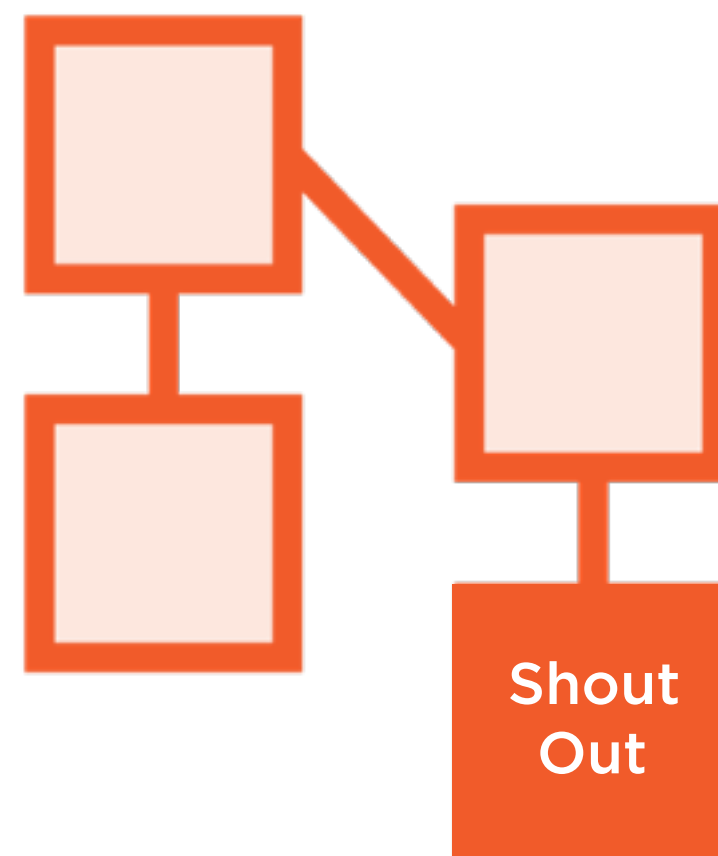
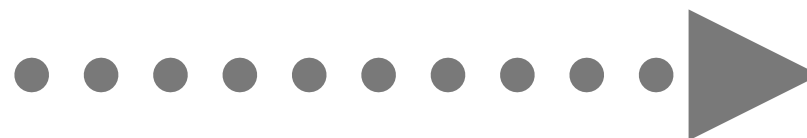


Empty Data Model

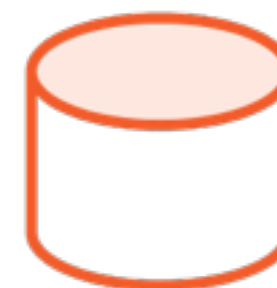


(Version 1)

Migration



Data Model w/ ShoutOut

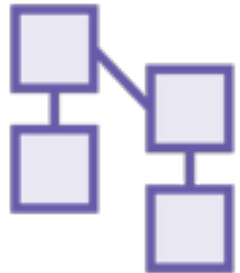


(Version 2)

Representing Entities in Code

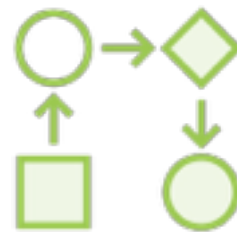
The Core Data Stack

Model



NSManagedObjectModel

Manage



NSManagedObject

NSManagedObjectContext

Persist



NSPersistentStoreCoordinator

NSPersistentStore

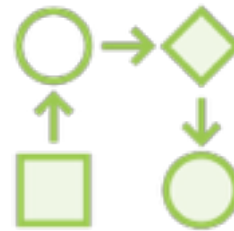
The Core Data Stack

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 NSObject Subclasses



Type safety

Convenience

Convenience of NSManagedObject Subclasses

NSManagedObject

```
shoutOut.setValue("Good job...", forKey: "message")
```



ShoutOut

- message
- from
- shoutCategory
- sentDate

```
shoutOut.message = "Good job..."
```

Advantages of NSManagedObject Subclasses



Type safety

Convenience

Custom logic

- entityName property**

Demo

Explore Xcode's NSManagedObject subclass creation methods

Create NSManagedObject subclass for ShoutOut Entity

Summary

Created a data model with Xcode

Configured an Entity called ShoutOut

**Represented a ShoutOut in code as an
NSManagedObject subclass**

Coming up: Relationships!