

Praktikum iOS-Entwicklung

Wintersemester 2017 / 2018

Prof. Dr. Linnhoff-Popien

Kyrill Schmid, Markus Friedrich

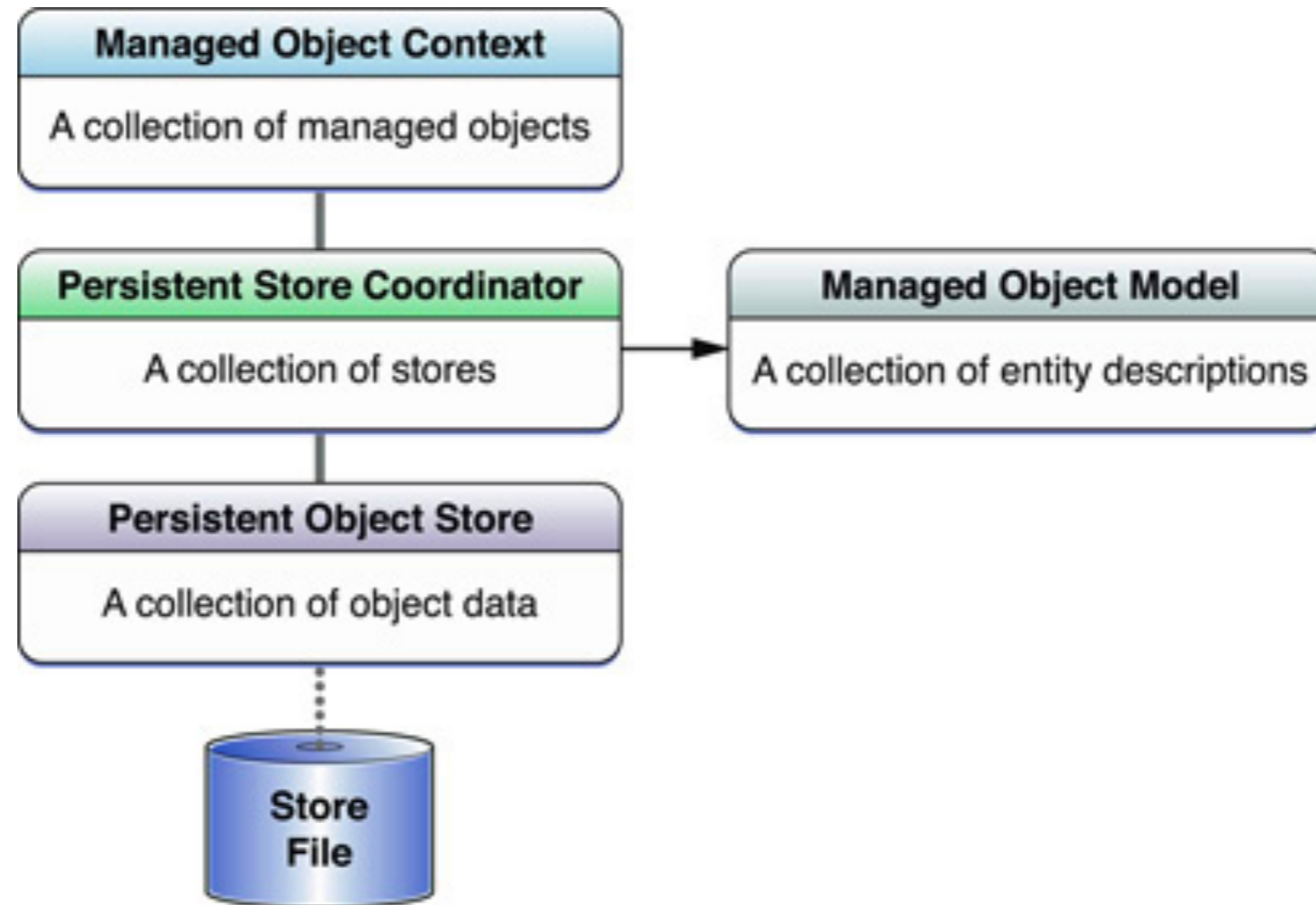


Core Data

Framework zum verwalten von *Model-Layer* Objekten:

- Change Tracking
- Change Propagation
- Konsistenz von Objekt-Beziehungen
- Integration in App-Controller
- Gruppieren und Filtern
- ...

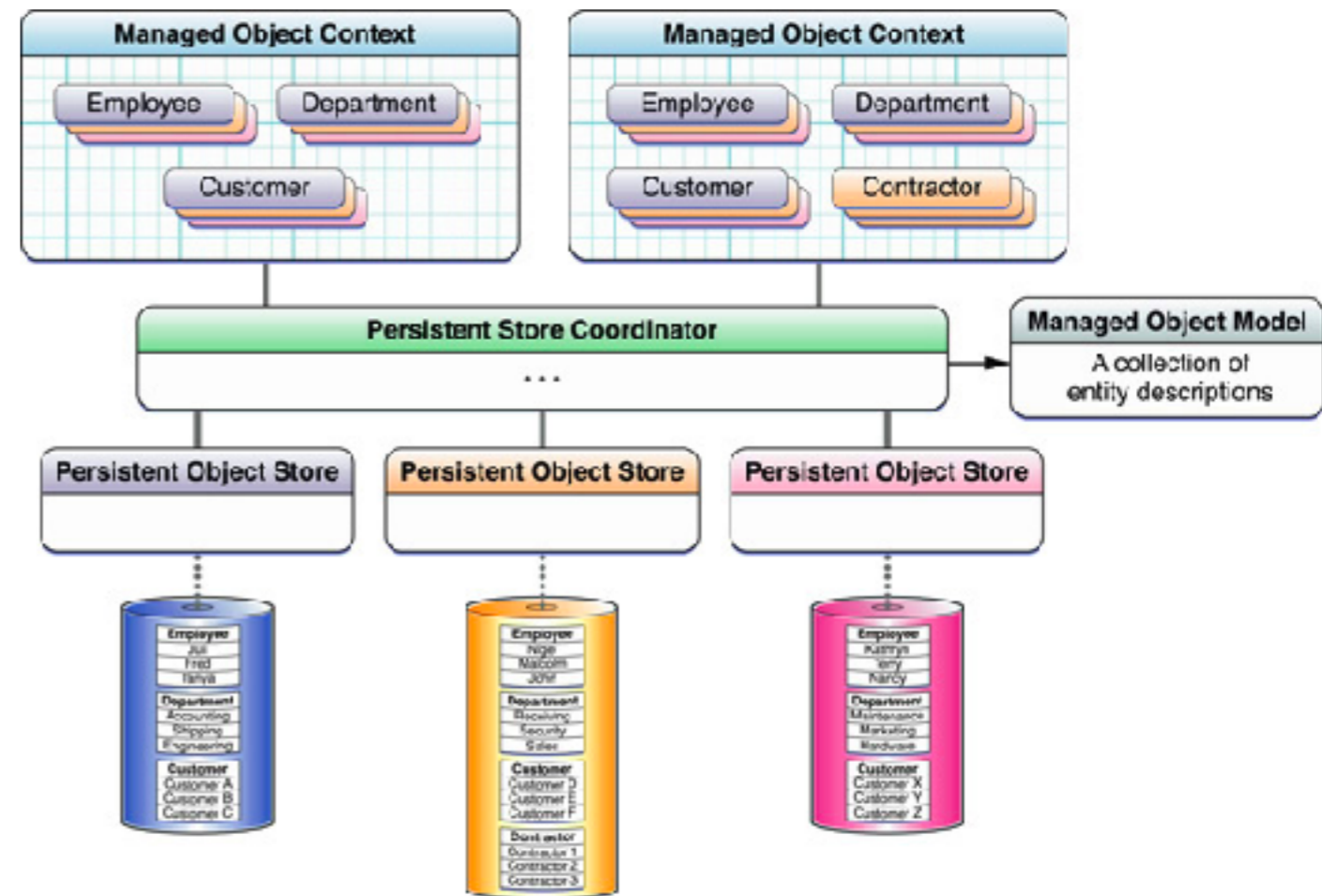
Core Data Stack



<https://developer.apple.com>

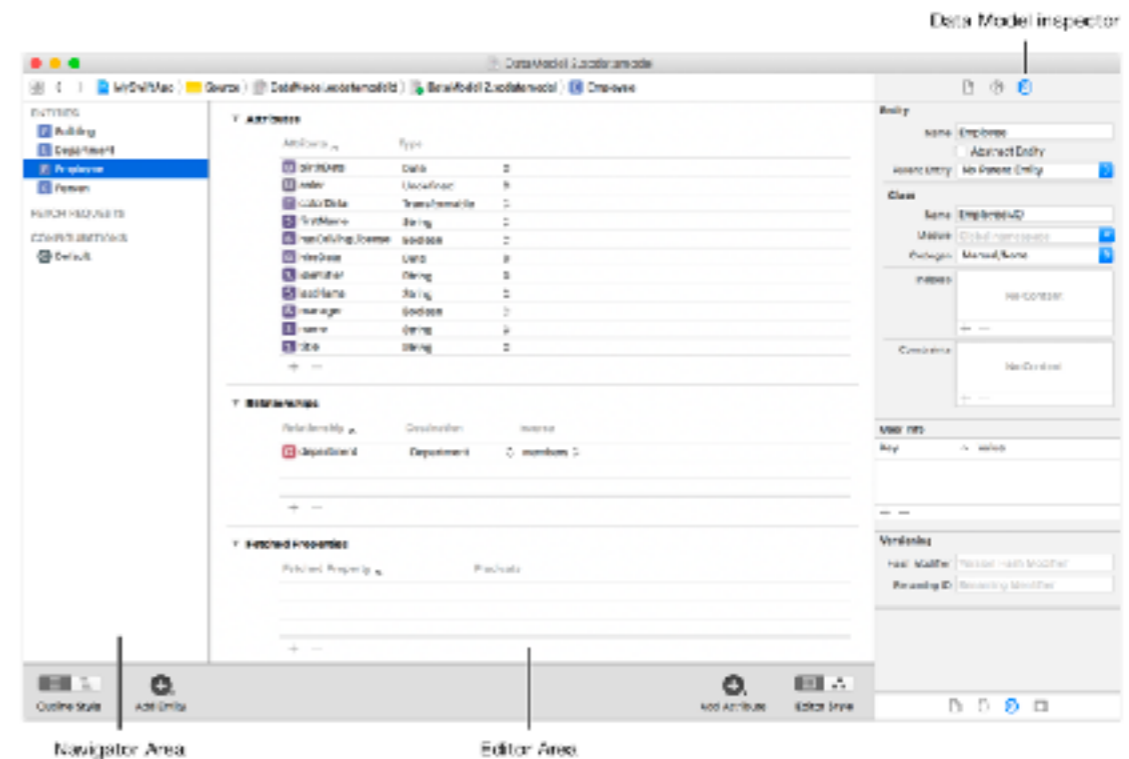
Persistent Store Coordinator

- Stack hat genau einen Persistent Store Coordinator
- Fassade für MOC: Gruppe von Stores erscheint als aggregierter Store



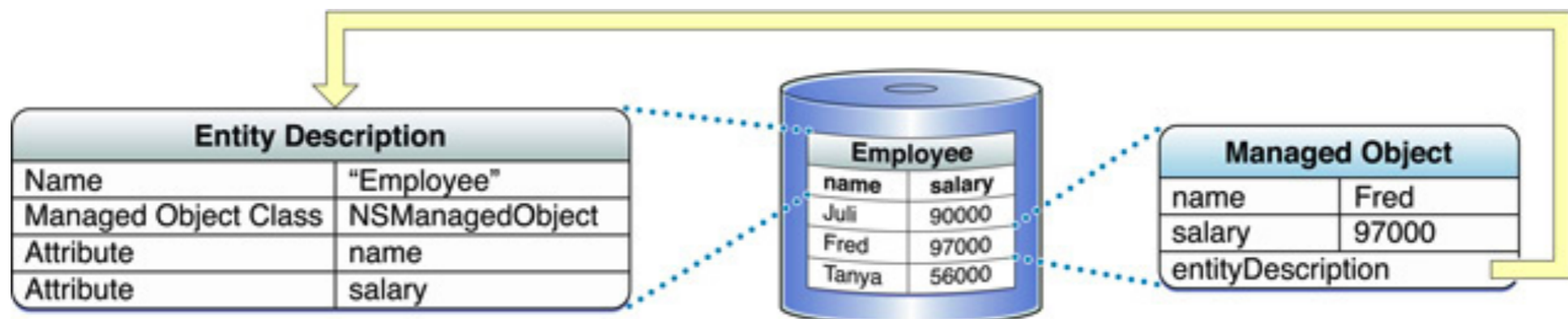
Managed Object Model

- Managed Object Model = Daten-Schema
- Managed Object Model beschreibt die Entitäten (Name, Klassenname, Properties)
- Kann in XCode mit Data Model Editor erstellt werden



Managed Object

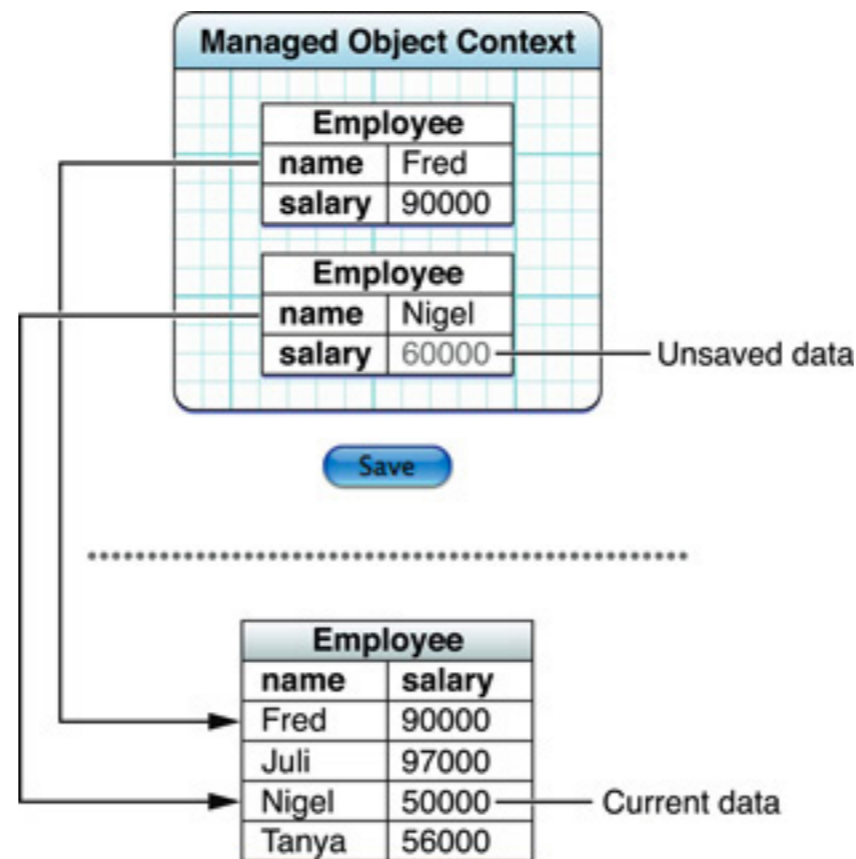
- Managed Object repräsentiert Record aus Persistent Store
- In einem Managed Context gibt es nur ein Managed Object zu einem Eintrag in einem Store



<https://developer.apple.com>

Managed Object Context

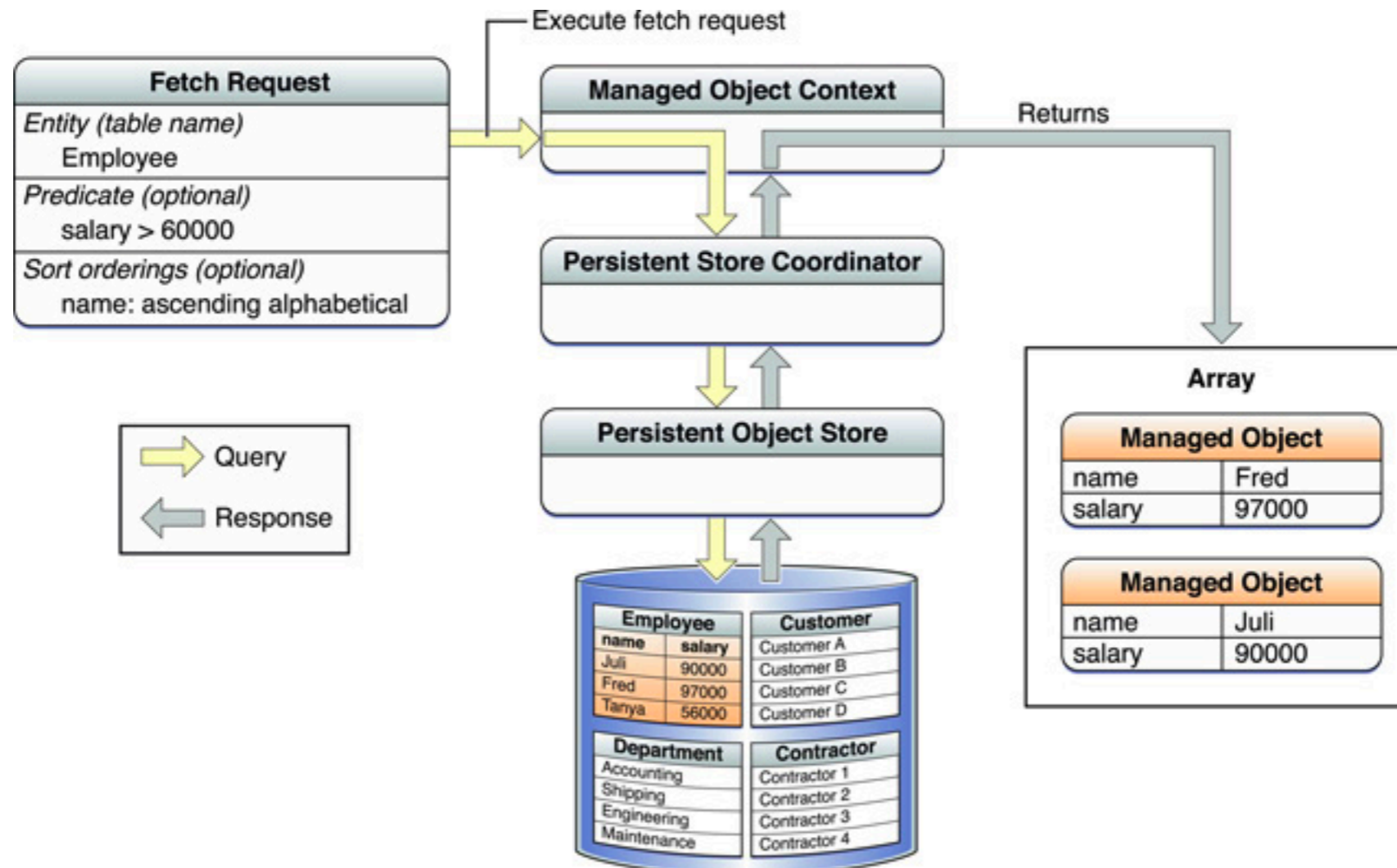
- MOC verwaltet eine Menge von Managed Objects
- MOC ermöglicht Anlegen und Fetchen von Objekten
- Undo/Redo



<https://developer.apple.com>

Fetch Requests

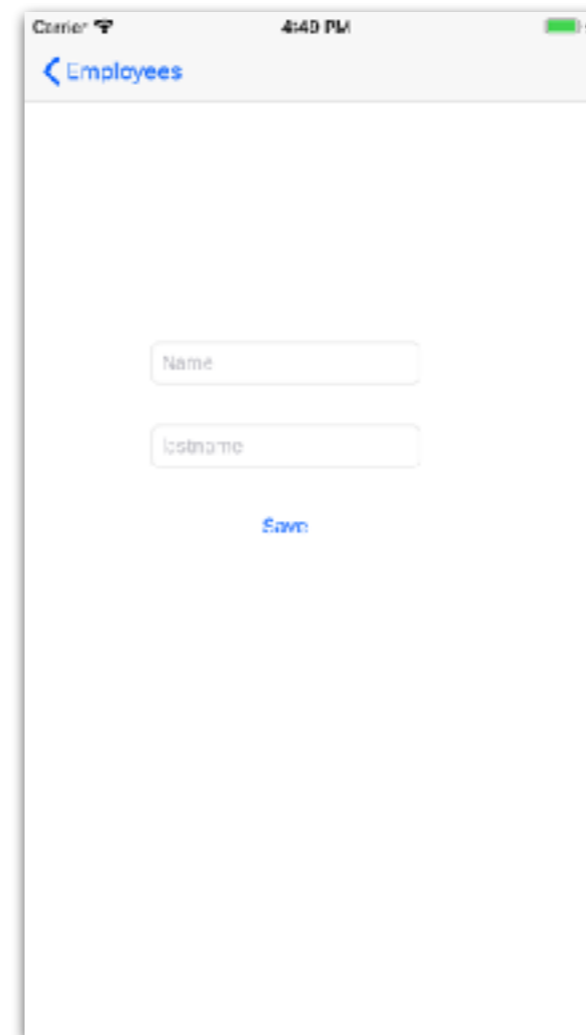
- Ermöglicht bestimmte Objekte zu fetchen und sortieren



<https://developer.apple.com>

Aufgabe

- Erstelle neues Projekt:
- Zwei Views:
 - TableView mit Einträgen Employee (Name, Geburtstag)
 - View zum Anlegen neuer Employees



Vorgehen

- Anlegen Daten Model:
 - Entität : Mitarbeiter
 - Attribute: name, nachname -> String
- Implement: `func initializeFetchedResultsController()`
- Implement: `override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)`
- Implement: `func save(_ sender: UIButton)`