

Mobile Applications Development 2 (MAD2)

Diploma in IT

Teaching Team:
Mr Ng Poh Oon
Mr Charles Keck

MAD2 Oct 2019

Chapter 5

Core Data

Objectives

Core Data

Summary

Discussion 1

Objectives

To be able to understand:

- Core Data Framework
 - Persist or cache data and support undo on a single device

<https://developer.apple.com/documentation/coredata>



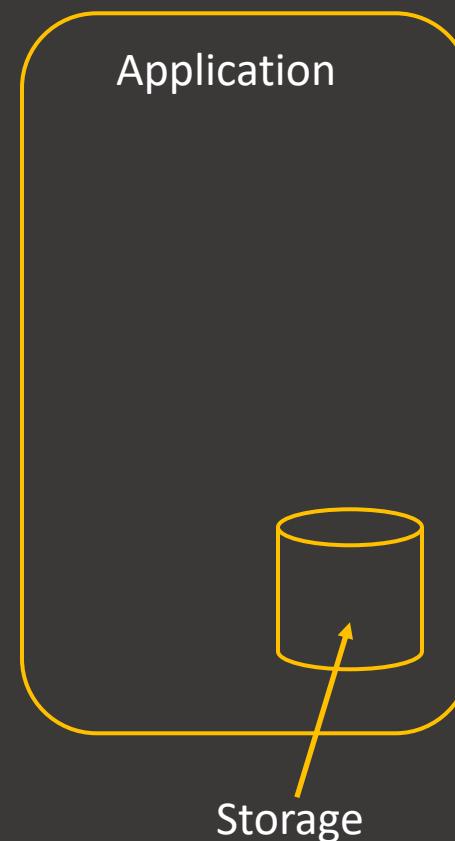
Core Data Framework

What is Core Data?

- Core Data is a **framework** that you use to manage the model layer objects in your application.
- It provides generalized and automated solutions to common tasks associated with object life cycle and object graph management, including **persistence**.

Persistent Storage

- Persistent storage is any data storage device that retains data after power to that device is shut off.
- It is also sometimes referred to as non-volatile storage.



Entity Class

Person
firstname
lastname
:
:

```
import CoreData
```

```
class Person: NSManagedObject {  
    @NSManaged var firstname : NSString  
    @NSManaged var lastname : NSString  
}
```

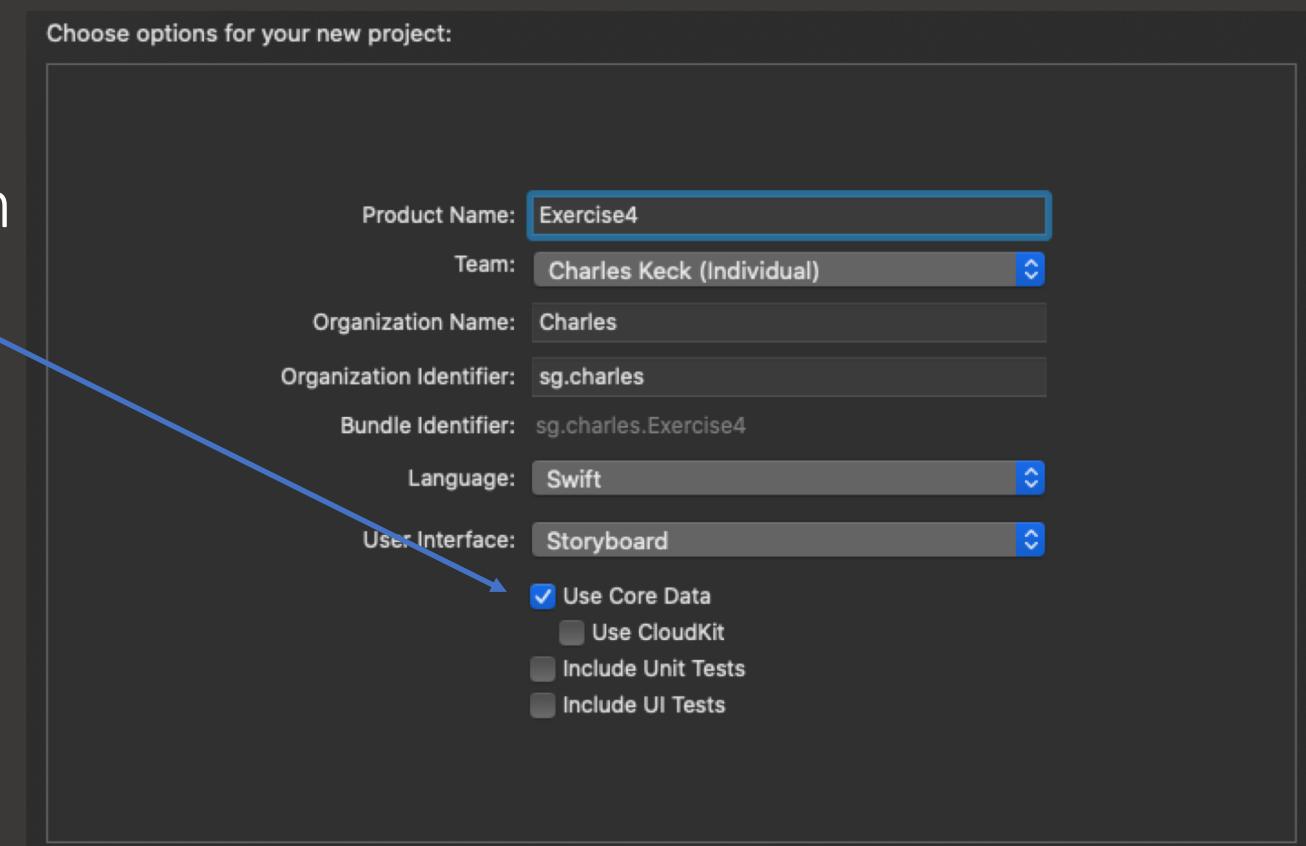
Properties/Attributes?

The screenshot shows the Xcode Data Model Editor. On the left, there is a list of entities: Person, Address, and Note. The Person entity is selected and highlighted with a blue border. Inside the Person entity, there is a section titled "Attributes" which contains three entries: "firstname", "lastname", and "mobileno". Below the attributes is a section titled "Relationships". To the right of the entity list, there is a table titled "Attributes" with three rows. Each row corresponds to one of the attributes in the Person entity: "firstname" (String type), "lastname" (String type), and "mobileno" (String type). There are also "+" and "-" buttons at the bottom of the attributes table.

Attribute	Type
firstname	String
lastname	String
mobileno	String

Xcode – Use Core Data

- Checking the "Use Core Data" box will cause Xcode to generate boilerplate code for what's known as a **Core Data stack** in `AppDelegate.swift`.



Core Data BoilerPlate

- AppDelegate.swift

The screenshot shows the Xcode interface with the file `AppDelegate.swift` open. Several annotations are present:

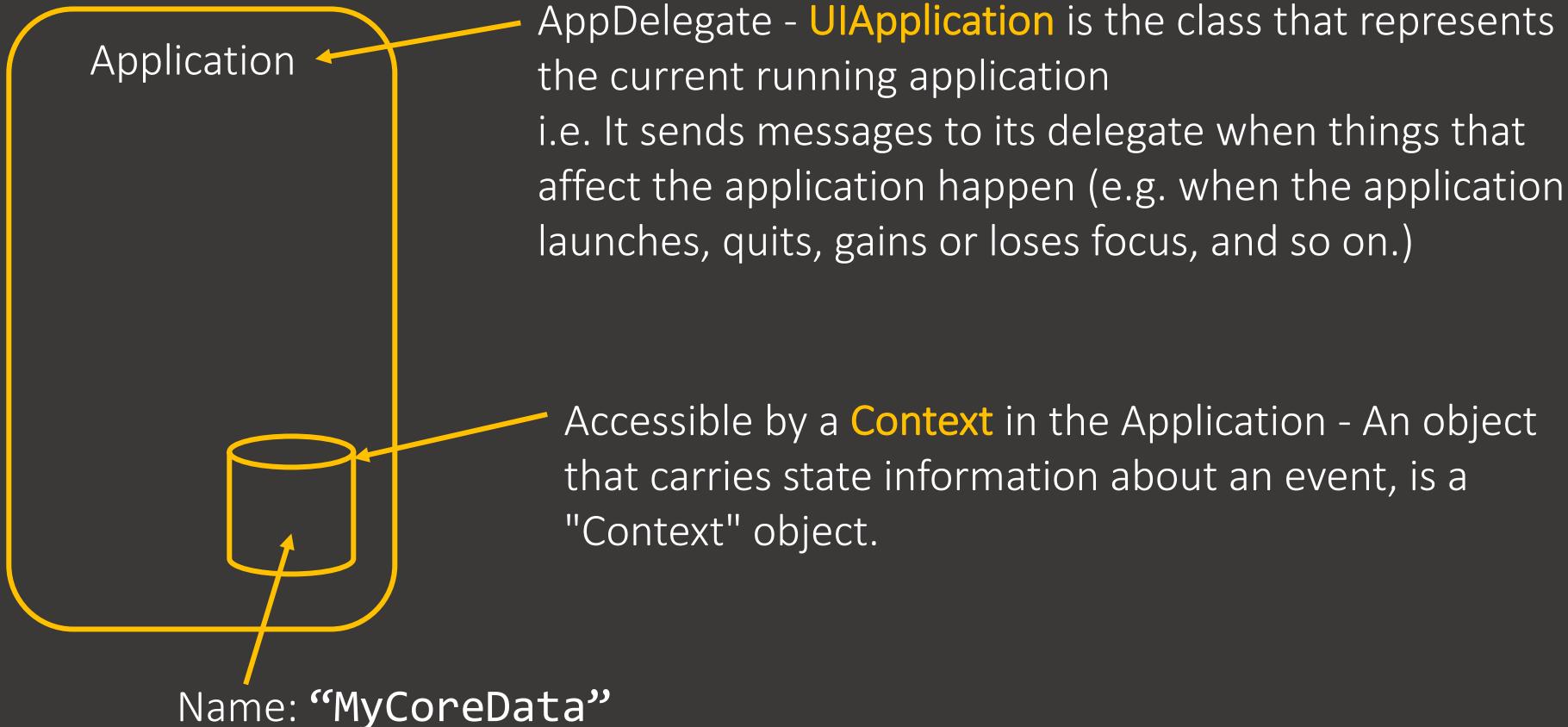
- A yellow arrow points from the left margin to the line `// MARK: - Core Data stack`.
- A red box highlights the line `let container = NSPersistentContainer(name: "MyCoreData")`, which is annotated with a yellow arrow pointing to the right.
- A yellow arrow points from the left margin to the line `// MARK: - Core Data Saving support`.
- A yellow arrow points from the left margin to the line `func saveContext () {`.

```
36 // MARK: - Core Data stack
37
38 lazy var persistentContainer: NSPersistentContainer = {
39
40     let container = NSPersistentContainer(name: "MyCoreData")
41
42     container.loadPersistentStores(completionHandler: { (storeDescription,
43         error) in
44         if let error = error as NSError? {
45             fatalError("Unresolved error \(error), \(error.userInfo)")
46         }
47     }
48     return container
49 }()
50 // MARK: - Core Data Saving support
51
52 func saveContext () {
53     let context = persistentContainer.viewContext
54     if context.hasChanges {
55         do {
56             try context.save()
57         } catch {
58             let nserror = error as NSError
59             fatalError("Unresolved error \(nserror), \(nserror.userInfo)")
60         }
61     }
62 }
63
64 }
```

The name must match the model's filename

MAD2 Oct 2019

AppDelegate and Context

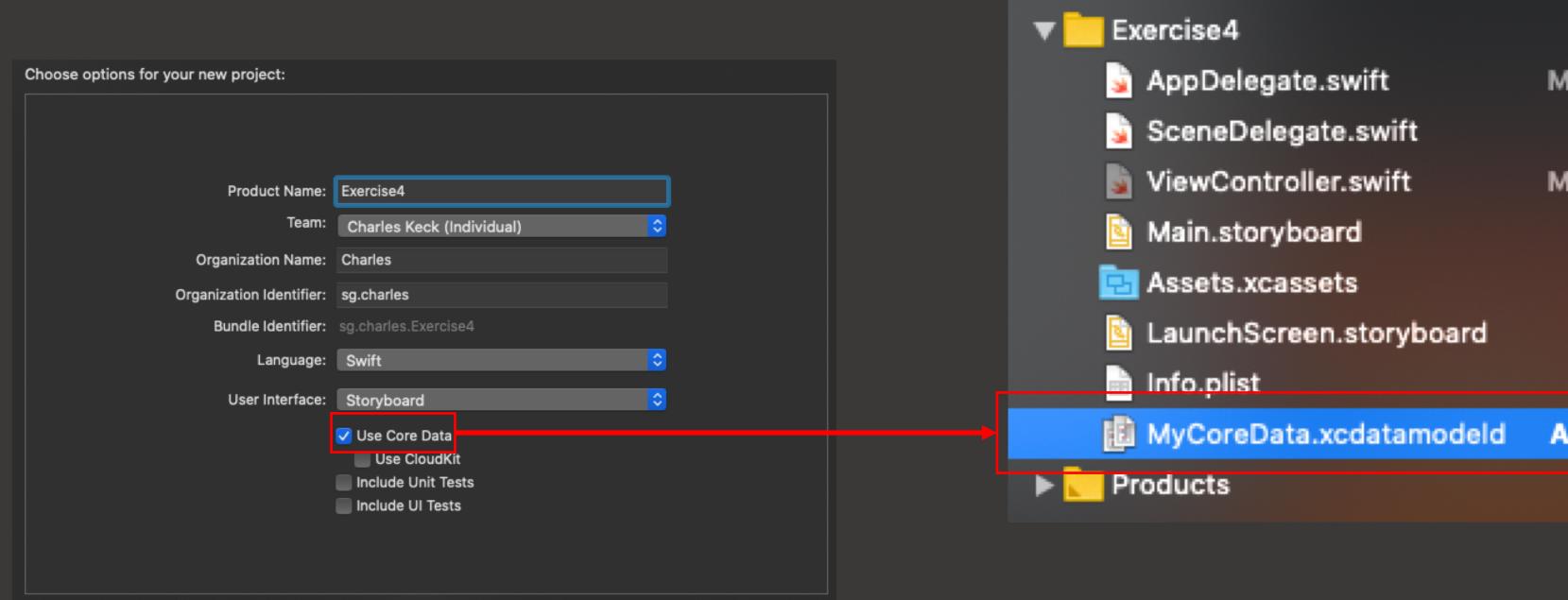


Swift Code (View Controller)

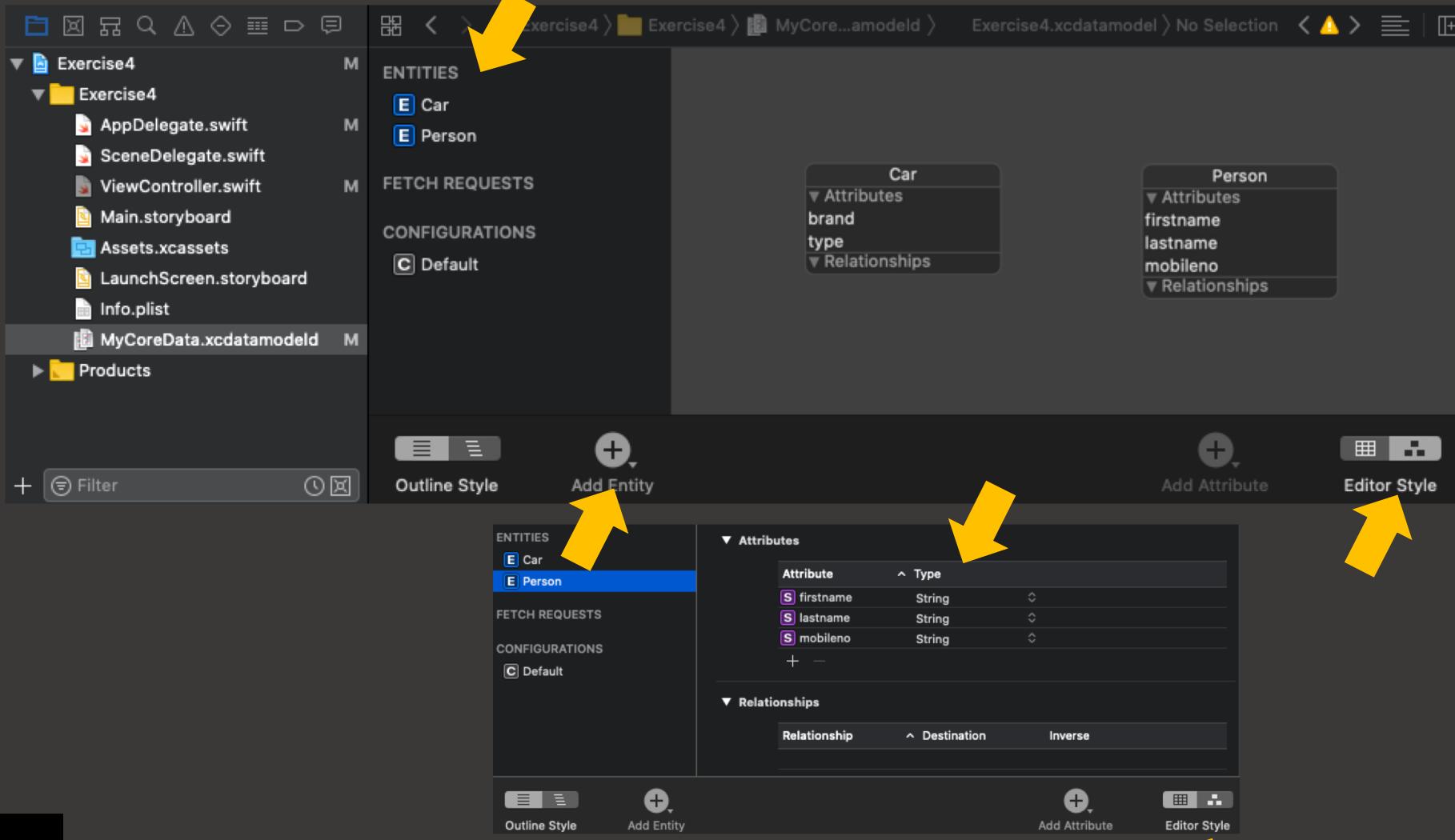
```
let appDelegate = (UIApplication.shared.delegate) as! AppDelegate  
  
let context = appDelegate.persistentContainer.viewContext  
  
//or write a getContext method in the appDelegate  
let context = appDelegate.getContext()
```

Data Model

- Extension : .xcdatamodeld

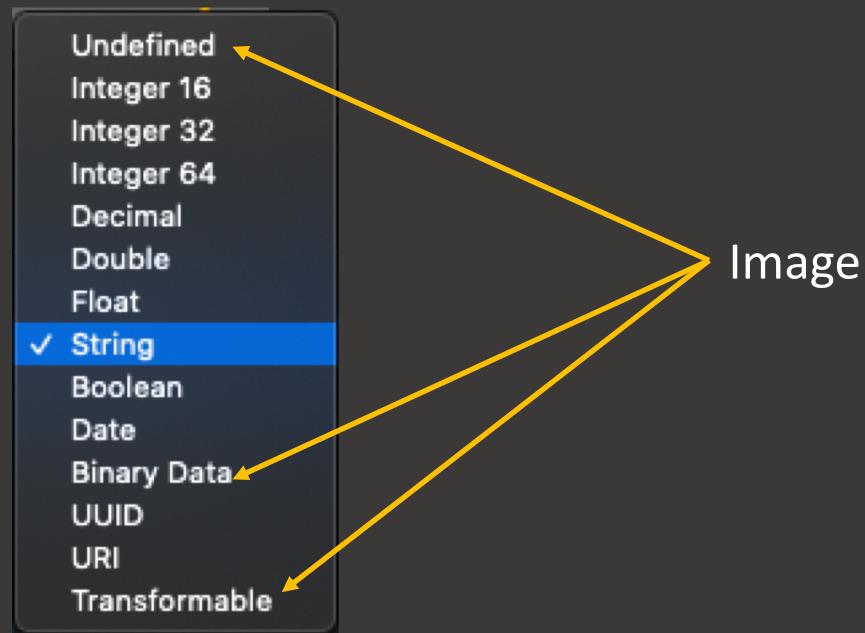


Data Model View



MAD2 Oct 2019

Data Type



Image

Add Data to Core Data

```
func AddData(){
    let appDelegate = (UIApplication.shared.delegate) as! AppDelegate
    let context = appDelegate.persistentContainer.viewContext

    let entity = NSEntityDescription.entity(forEntityName: "Person", in: context)!

    let person = NSManagedObject(entity: entity, insertInto: context)
    person.setValue("Melfred", forKeyPath: "firstname")
    person.setValue("Sawyer", forKeyPath: "lastname")
    person.setValue("91111222", forKeyPath: "mobilenumber")
        ^ Data          ^ Key
    do {
        try context.save()

    } catch let error as NSError {
        print("Could not save. \(error), \(error.userInfo)")
    }

}
```

Fetching Data from Core Data

- Fetch Request and Exception Handling

```
func FetchData(){
    var people:[NSManagedObject] = [] Empty List

    let appDelegate = (UIApplication.shared.delegate) as! AppDelegate
    let context = appDelegate.persistentContainer.viewContext

    let fetchRequest = NSFetchedResultsController<NSManagedObject>(entityName: "Person") (entityName)
    do {
        people = try context.fetch(fetchRequest)

        for p in people {
            let firstname = p.value(forKeyPath: "firstname") as? String
            let lastname = p.value(forKeyPath: "lastname") as? String
            let mobileno = p.value(forKeyPath: "mobileno") as? String
            print("\(firstname!) \(lastname!), \(mobileno!)")
        }
    } catch let error as NSError {
        print("Could not fetch. \(error), \(error.userInfo)")
    }
}
```

```
Melfred Sawyer, 91111222
Simon Wood, 92222111
```

All Output

Filter



More Information

- https://youtu.be/PMze4tT2w_U

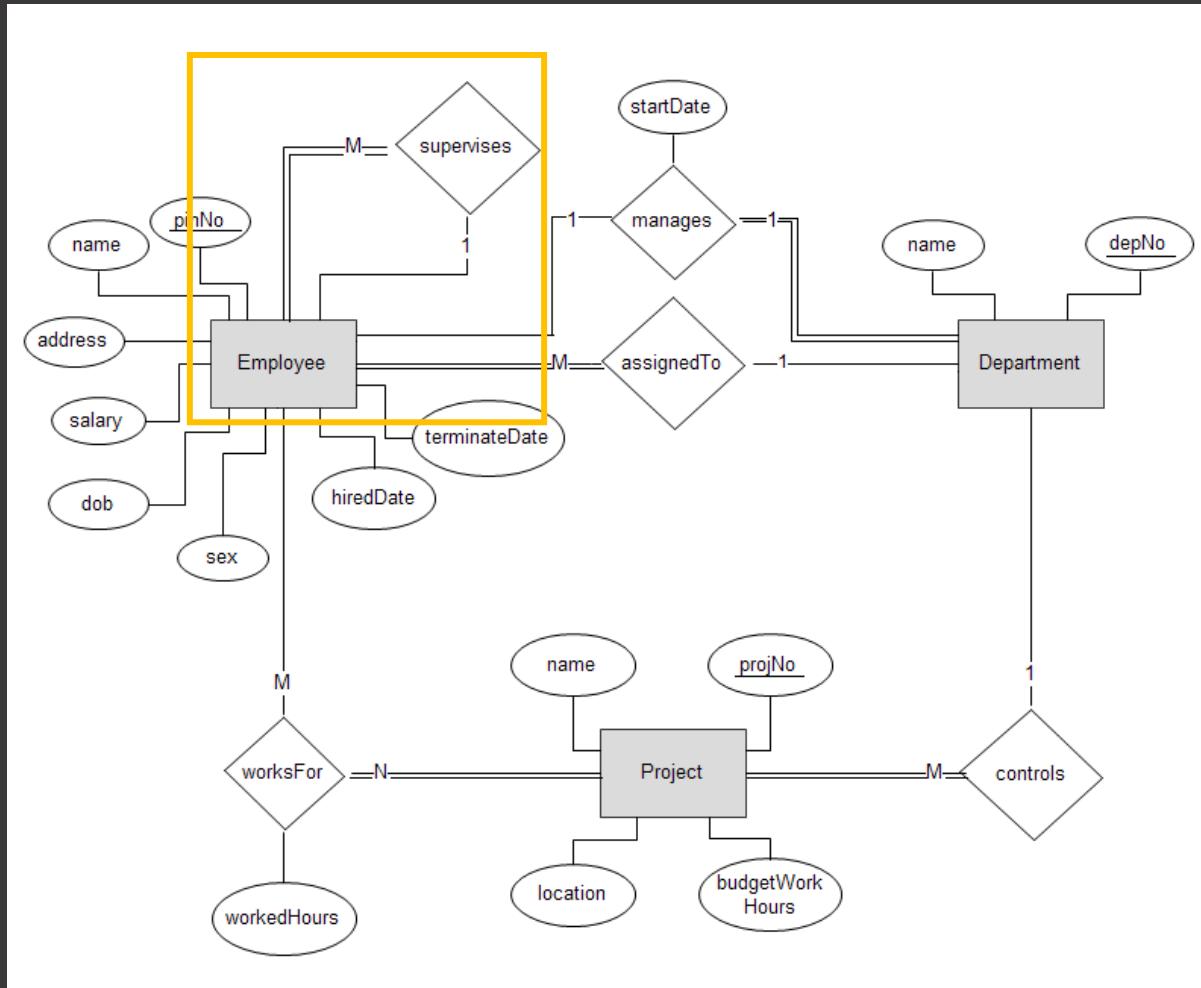


<https://www.raywenderlich.com/7569-getting-started-with-core-data-tutorial#toc-anchor-003>
MAD2 Oct 2019

Relationships

- A relationship is a link between multiple entities. In Core Data, relationships between two entities are called **to-one relationships**, while those between one and many entities are called **to-many relationships**.
- For example, a Manager can have a to-many relationship with a set of employees, whereas an individual Employee will have a to-one relationship with his manager.

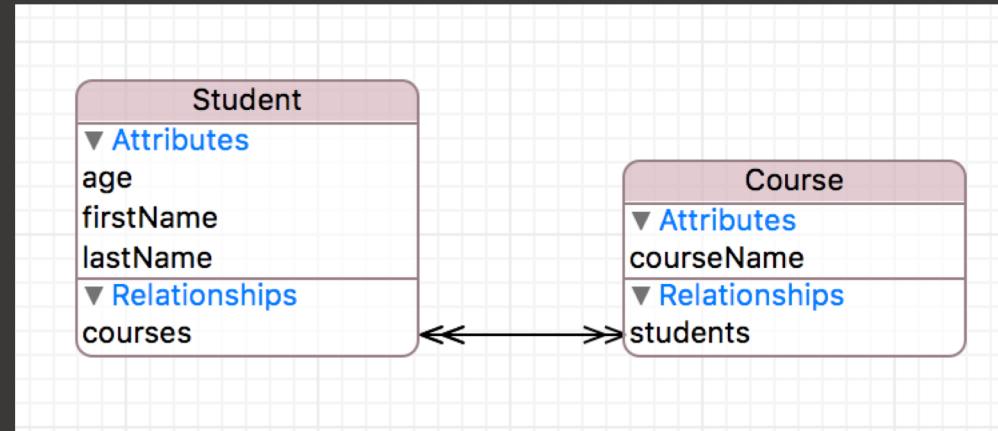
Relationships



MAD2 Oct 2019

Quick Exercise

- Without creating NSManagedObject Subclass

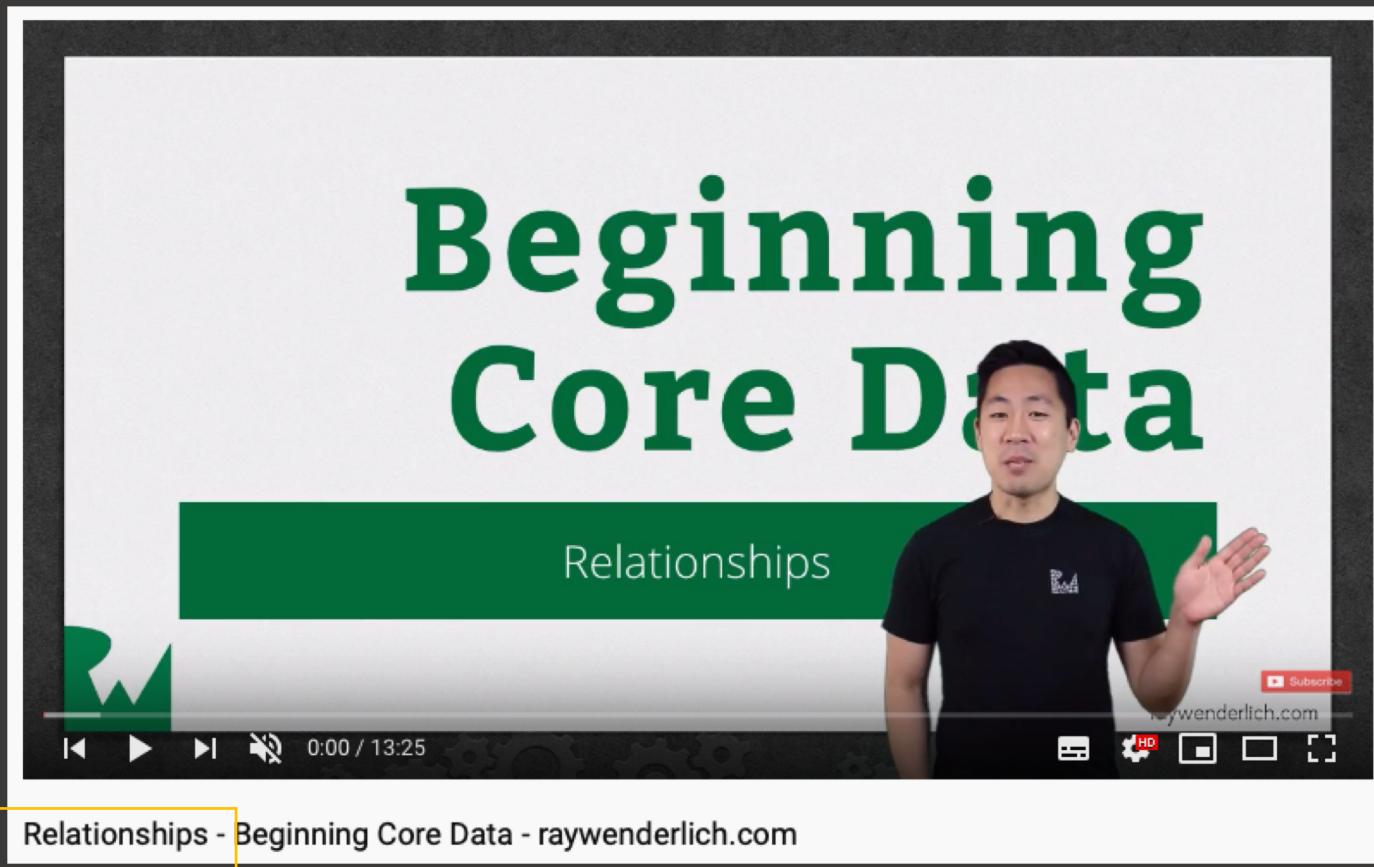


XCode created the derived classes for developer

For advanced developer, you can generate the Classes (re-declaration error may occurs)

More Information

- <https://youtu.be/uJuLk1niBYA>



Summary

Understand the Core Data framework