iOS Development

Core Data

What is a Database?

A relational database allows related objects to be represented in a way that is fast to recall.

Student objects can be related to a number of courses, namely those which they are taking. A courses can be related to students, namely the ones who are taking that course.

Database terminology

One-to-one - One object relates to exactly one other object. Ex: A person has one spouse, and vice versa.

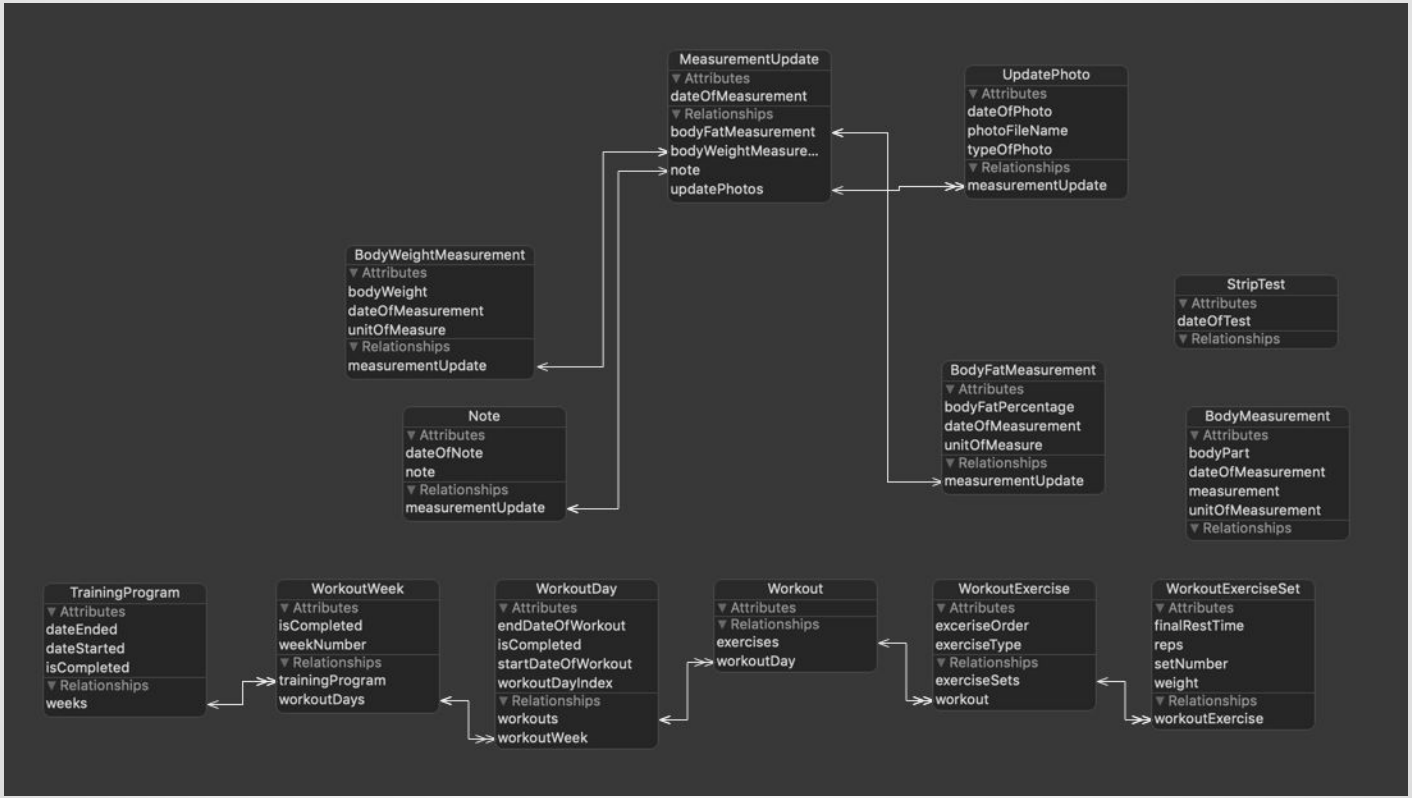
One-to-many or **many-to-one** - One object relates to many others or the other way around. Ex: A person has many possessions.

Database Terminology

Tables or Entities - Is an object that has several fields/attributes/columns. Ex: A student entity has a first name, last name and list of courses he/she is taking.

Column or attribute - An object’s properties or fields. Ex: First name, last name, age.

Example Database Schema



What is Core Data?

Several technologies that enable the storage of large amounts of data, efficiently, as well as the efficient and filtered recall of that data.

Core Data Features

- Change tracking and undo support.
- Relationship maintenance.
- Futures (faulting).
- Automatic validation of property values.
- Schema migration.
- Optional integration with the application’s controller layer to support user interface synchronization.
- Full, automatic, support for key-value coding and key-value observing.
- Grouping, filtering, and organizing data in memory and in the user interface.
- Sophisticated query compilation.(NSPredicate)
- Merge policies.

Why Use Core Data

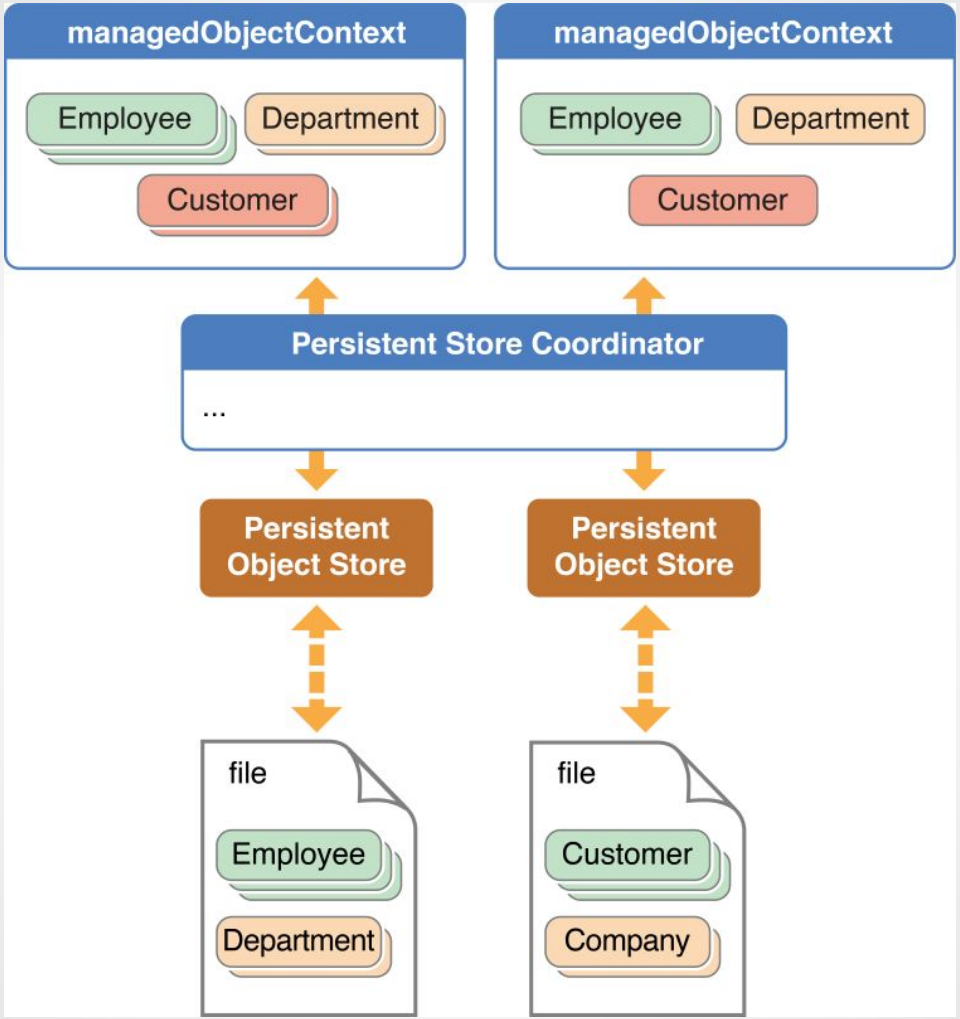
- Code is 50-70% smaller
- Built in, optimized and tested code
- Mature (announced at WWDC 2005)

If apps requires data with complex relationships or serializing is too expensive

What Core Data is not

Not a relational database

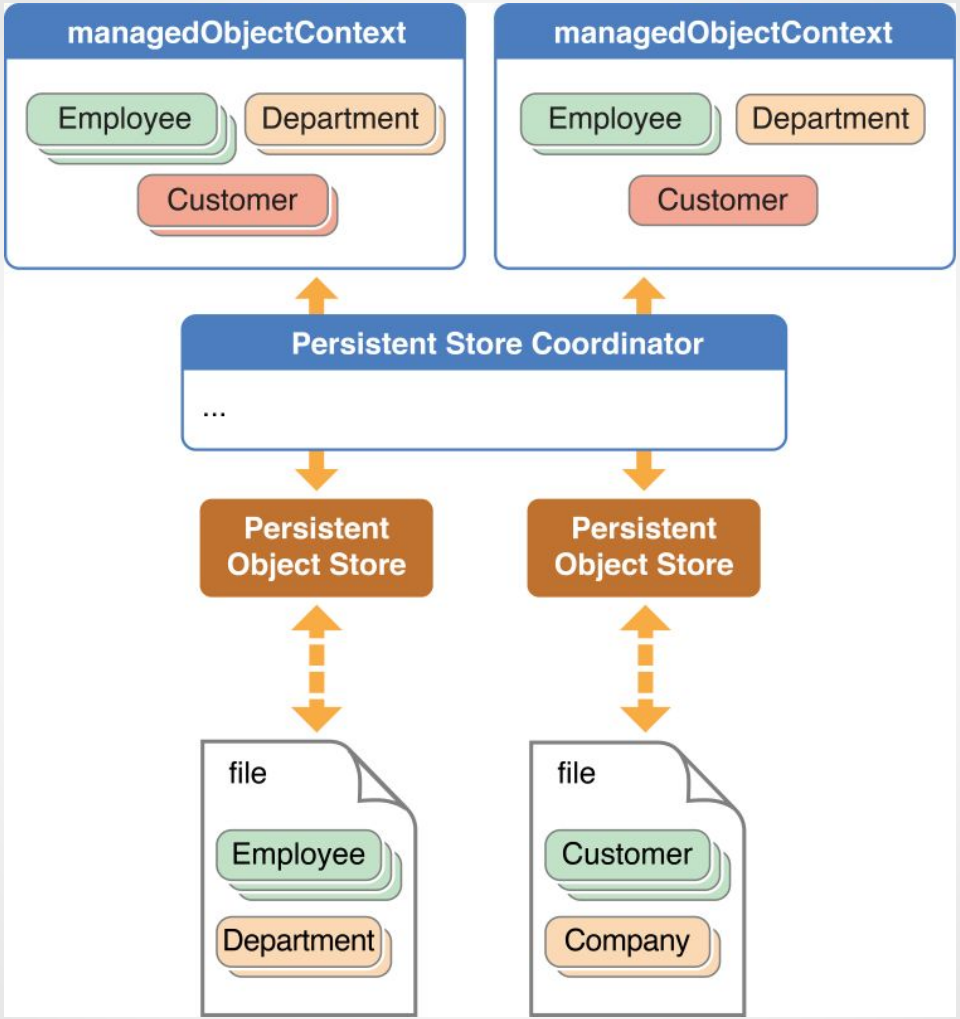
Not a end-all, be-all (but close)


managedObjectContext
Employee
Department
Customer
managedObjectContext
Employee
Department
Customer
Persistent Store Coordinator
...
Persistent Object Store
Persistent Object Store
file
Employee
Department
file
Customer
Company

Core Data Terminology

Persistent Store - the data storage object (SQLite, binary, XML)

Persistent Store Coordinator - Handles loading and control of the persistent store


managedObjectContext
Employee
Department
Customer
managedObjectContext
Employee
Department
Customer
Persistent Store Coordinator
...
Persistent Object Store
Persistent Object Store
file
Employee
Department
file
Customer
Company

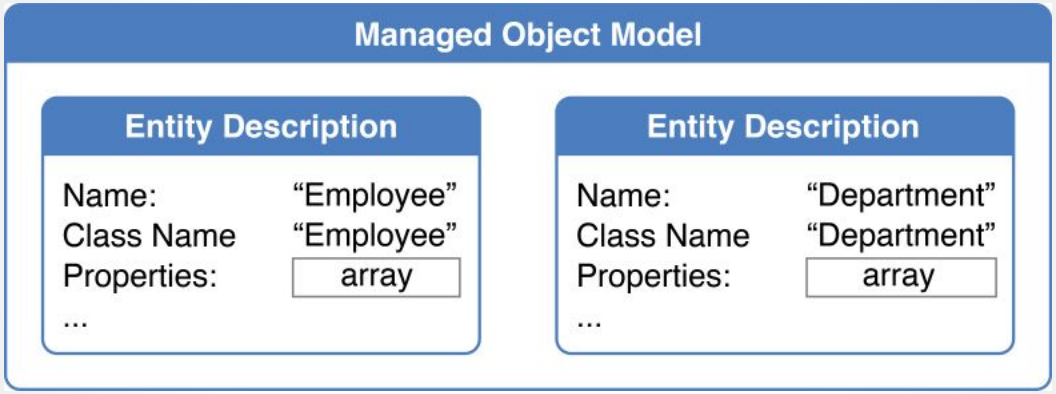
Core Data Terminology

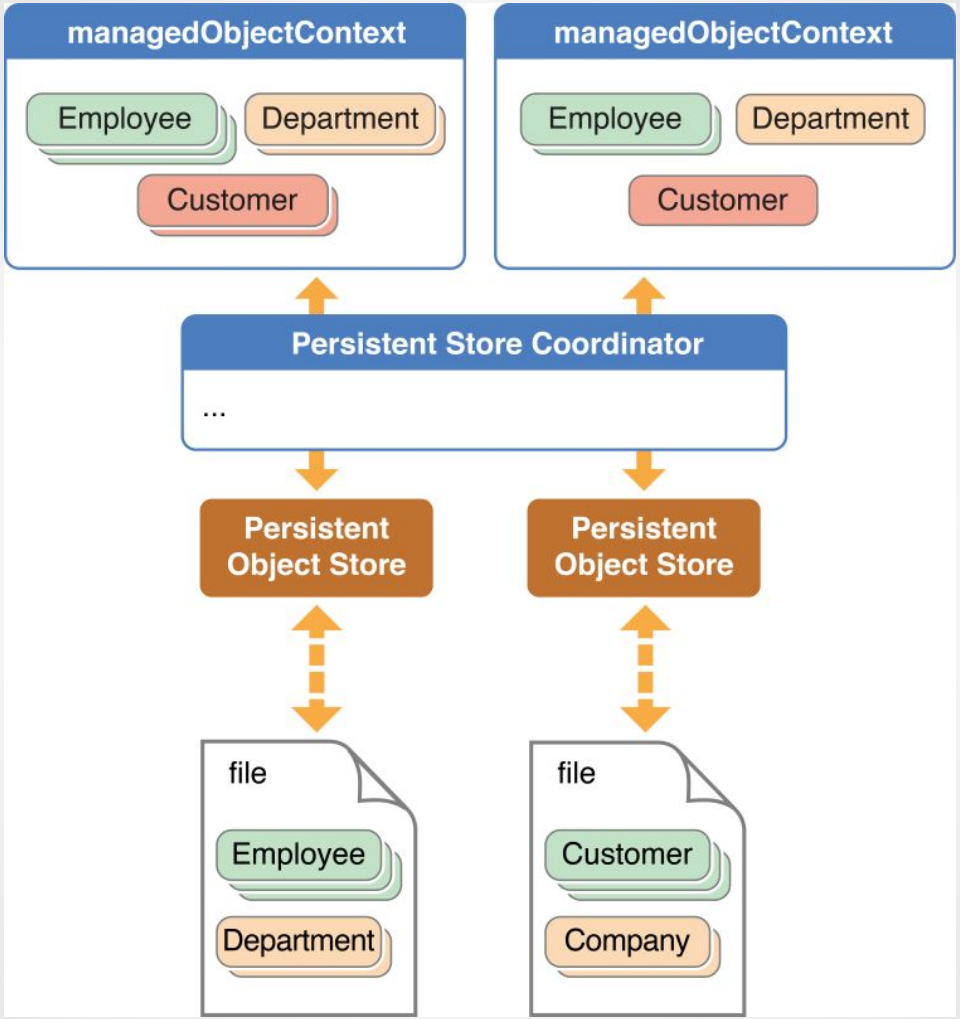
Managed Object Model - Definition of our models

Managed Object - An object in Core Data

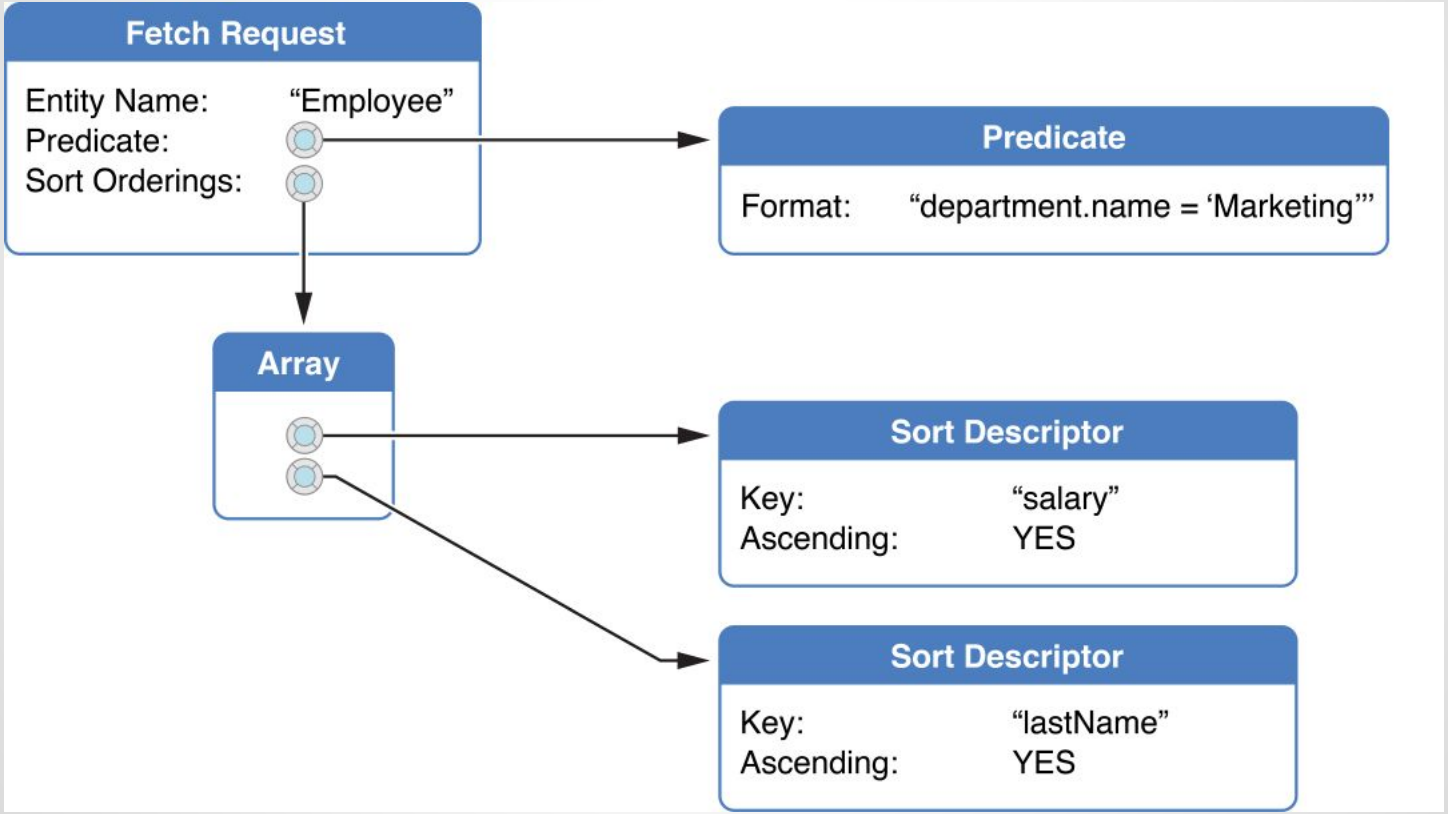
Managed Object Context - an intelligent scratch pad

Managed Object Model




managedObjectContext
Employee
Department
Customer
managedObjectContext
Employee
Department
Customer
Persistent Store Coordinator
...
Persistent Object Store
Persistent Object Store
file
Employee
Department
file
Customer
Company

Fetching Data



SQLite Browser

SQLite Browser

NSEntityDescription

Description of your entity.

Use it to insert or reference managed objects.

Core Data

Demo

NSEntity Description Usage

```
NSEntityDescription.insertNewObjectForEntityForName(entityName: String, inManagedObjectContext: NSManagedObjectContext)
// OR
let newStudent:Student = Student(context: DatabaseController.persistentContainer.viewContext)
newStudent.studentName = "Mike"
newStudent.studentMajor = "Computer Science"

let newCourse:Course = Course(context: DatabaseController.persistentContainer.viewContext)
newCourse.classTitle = "CS402"
newCourse.classTeacherName = "Mike"
newStudent.courses = [newCourse, otherNewCourse]
```

Fetching Data from the DB

```
let fetchRequest:NSFetchRequest = Student.fetchRequest()

do{
  let fetchResults = try
appDelegate.managedObjectContext.executeFetchRequest(fetchRequest)
  if fetchResults.count > 0 {
    for( var index = 0; index < fetchResults.count; index++ ){
      let currentBuilding = fetchResults[index]
      print( currentBuilding.buildingName )
    }
  }
} // catch omitted
```

Fetching Specific Data from the DB

```
let fetchRequest:NSFetchRequest = Student.fetchRequest()

let fetchPredicate:NSPredicate = NSPredicate(format: "studentName ==
%@", "Mike")

fetchRequest.predicate = fetchPredicate

// execute fetch...
```

Removing Data from the DB

```
DatabaseController.persistentContainer.viewContext.delete(object)
```