

Chapter 12

iOS Core Data

iOS Core Data

- ◉ Core Data เป็น ORM framework บน iOS และ OS X
- ◉ ช่วยให้นักพัฒนา focus ที่ Object-Oriented และ functionality โดยไม่ต้องกังวลกับการทำ mapping ระหว่าง Object และ Relational Database
- ◉ บน iOS จะเป็น local database (ไม่ใช่ client/server) ใช้ SQLite เป็น database และเป็น In-Process Database ไม่สามารถ share ข้อมูลกับ App อื่นได้
- ◉ เราสามารถเขียน code เพื่อจัดการกับ SQLite ได้โดยตรง แต่การใช้ Core Data จะช่วยลด code ใน การจัดการ database ลง 50% - 70% โดยเฉพาะการทำ Lazy Loading และ Data Migration

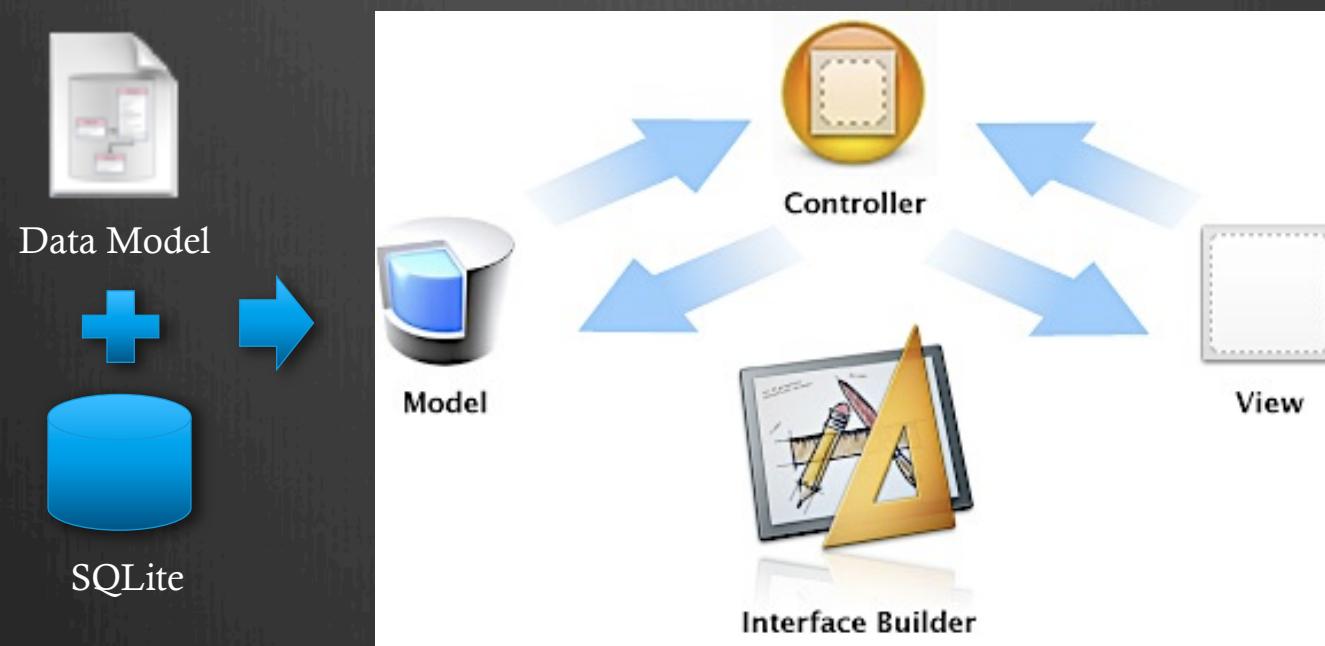
ดูข้อมูลเพิ่มเติมได้ที่ <http://goo.gl/SnRS80>

Explorer SQLite File

- ⦿ เมื่อใช้ Core Data ตัว framework จะสร้างไฟล์ database และ schema ให้อัตโนมัติ และเก็บไว้ที่โฟลเดอร์ Document ภายใต้ Application บน iOS Device
- ⦿ เราสามารถเปิดดูข้อมูลในไฟล์ที่ถูกสร้างขึ้นมาได้ด้วยโปรแกรมที่อ่าน SQLite ได้ เช่น Plug-ins ของ Firefox ชื่อ SQLite Manager
- ⦿ ในการณีที่เราพัฒนาโดย run บน iOS emulator ไฟล์จะถูกเก็บอยู่ที่
`/Users/<user>/Library/Application Support/iPhone Simulator/<iOS_version#>/<random_guid>/<app_name>/Documents/<app_name>.sqlite`

Note 1 : โดยปกติ Folder ~ / Library / จะถูกกำหนดเป็น hidden folder เราสามารถเข้าถึงได้จาก Finder แล้วเลือกเมนู Go > Go to Folder... หรือกด Command + Shift + G

iOS Core Data



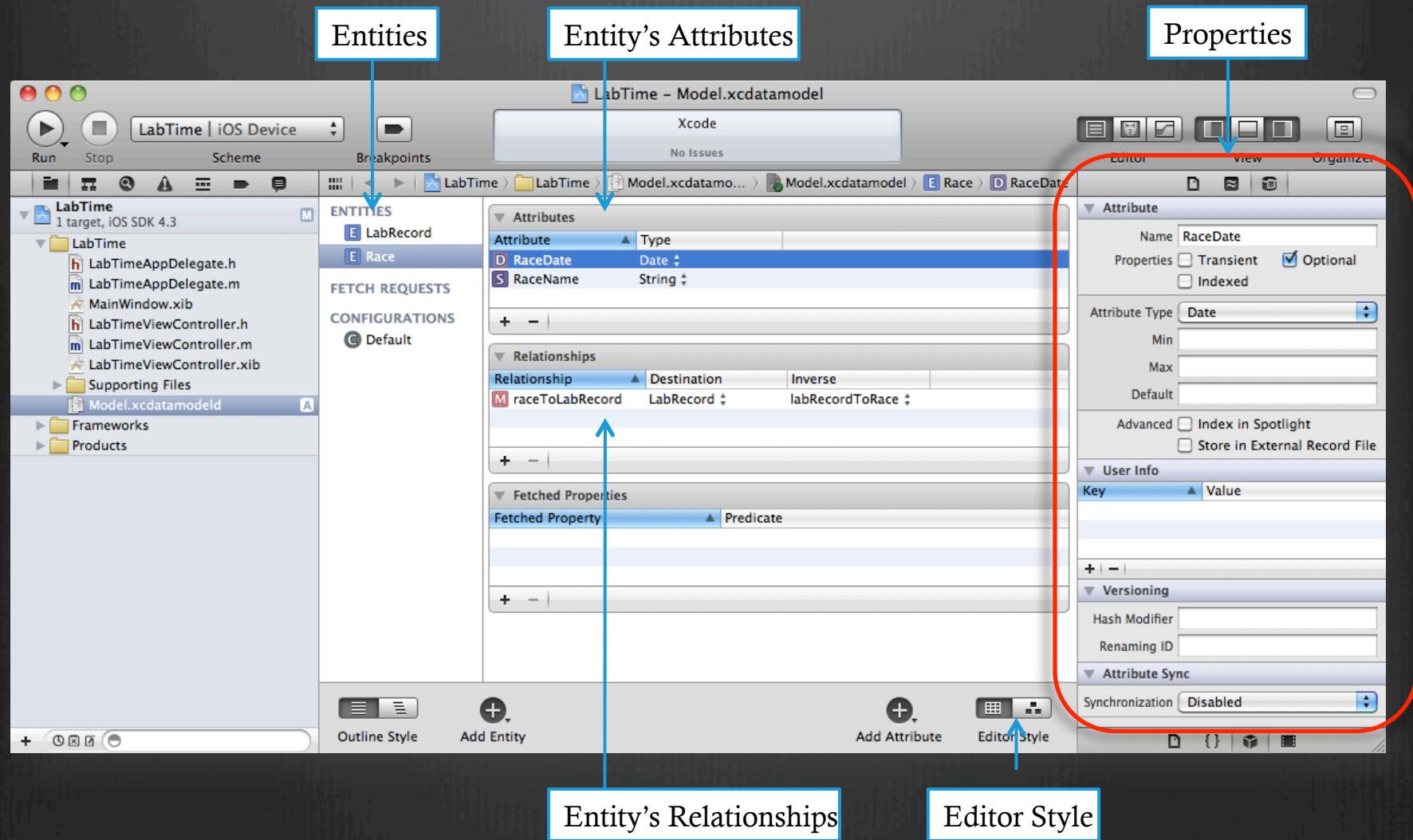
Core Data Features

- 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...

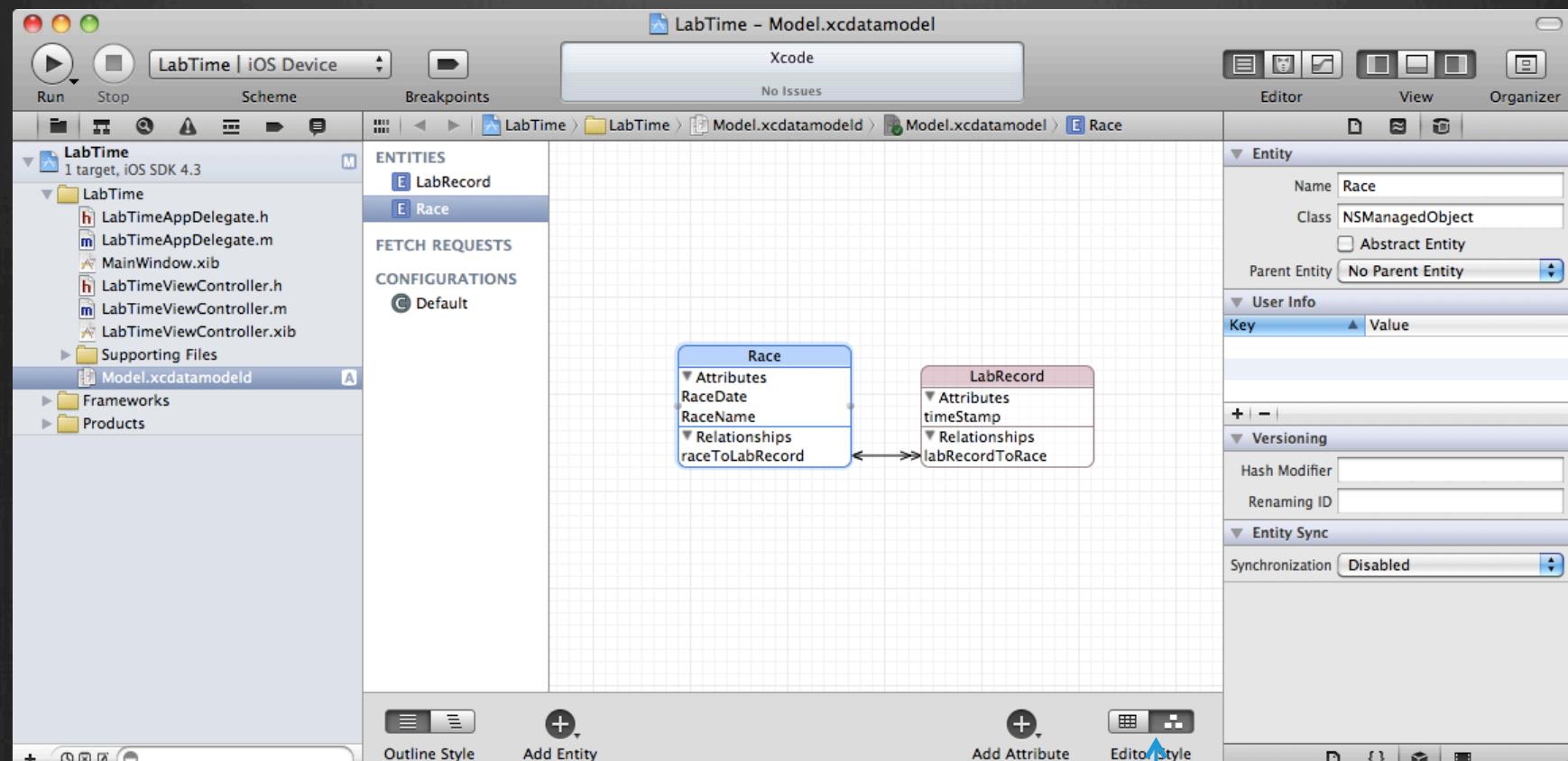
Core Data Features (cont.)

- Full, automatic, support for key-value coding and key-value observing.
- Grouping, filtering, and organizing data in memory and in the user interface.
- Automatic support for storing objects in external data repositories.
- Sophisticated query compilation.
- Merge policies.

Core Data Editor – Table Style



Core Data Editor – Graph Style



Change Editor Style to Graph

Lab 1-2 : LapTime (1/13)

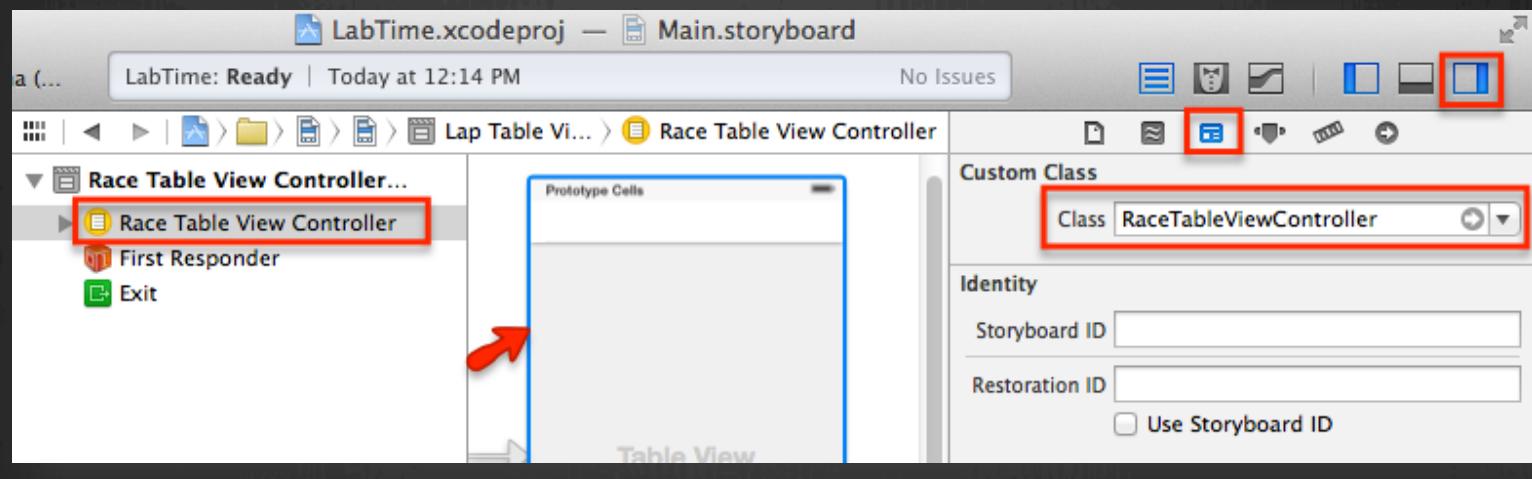
- ◎ วัตถุประสงค์
 - ◎ เพื่อให้สามารถพัฒนา iOS App เพื่อติดต่อกับ database โดยใช้ Core Data ได้
 - ◎ เข้าใจการนำข้อมูลใน SQLite มาแสดงผลบน Table View ด้วย Core Data
- ◎ ขั้นตอน
 - ◎ สร้าง project ใช้ Table View และ View ปกติ
 - ◎ สร้าง Data Model
 - ◎ เพิ่ม Entity และ Attributes ใน Entity
 - ◎ เขียน code เพื่อ access data ผ่าน Core Data
 - ◎ Read/Create/Delete data ผ่าน Core Data

Task : Create Project (2/13)

1. จาก Xcode สร้าง project ใหม่โดยเลือก iOS > Application > Single View Application
2. ตั้งชื่อ project ว่า “LapTime” และเลือก Devices เป็น iPhone
3. Click “Next” เลือก folder ที่จะ save project แล้ว click ปุ่ม “Create”
4. เปิด Main.storyboard แล้วลบ View Controller ออกจาก storyboard
5. ลบ ViewController.h และ ViewController.m ออกจาก project โดย click ขวา เลือก Delete และเลือก Move To Trash

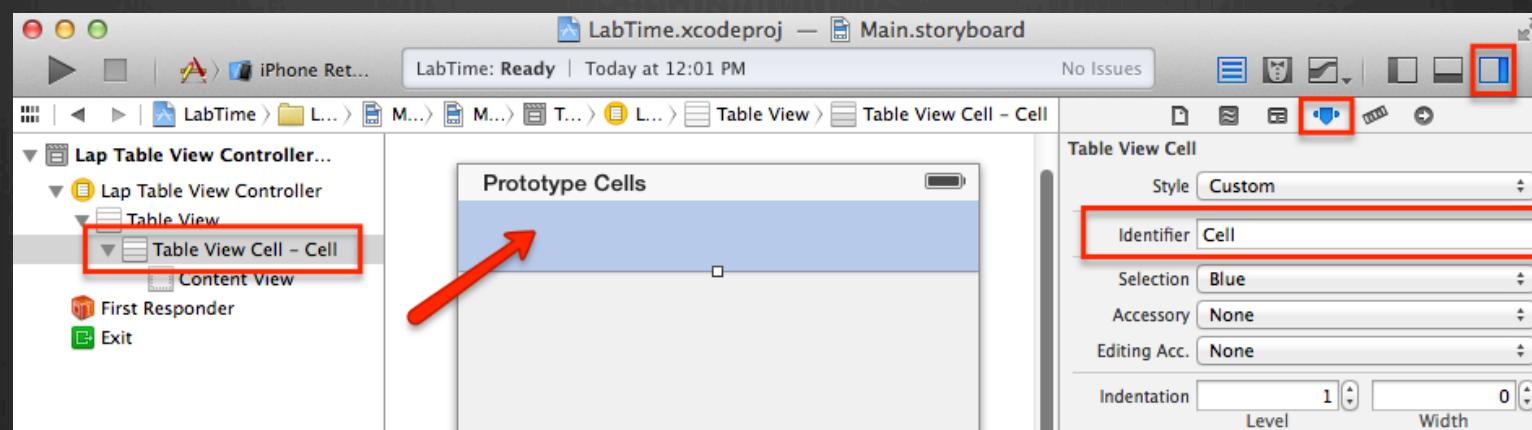
Task : Add Table View (3/13)

6. สร้าง class ใหม่ โดย click ขวาที่ project ใน Navigation Pane เลือก New File... > iOS > Cocoa Touch > Objective-C class และ click ปุ่ม “Next”
7. ตั้งชื่อ class ว่า “RaceTableViewController” และเลือก Subclass of เป็น “UITableViewController” (ไม่ต้องเลือก check box ทั้ง 2 ตัว) จากนั้น click ปุ่ม Next และ Create ตามลำดับ
8. เปิดไฟล์ Main.storyboard และลาก Table View Controller จาก Library Pane มาวางลงบน storyboard
9. Click ที่ table view และเปิด Identity Inspector บน Inspector Pane และเปลี่ยน class เป็น “RaceTableViewController”



Task : Define Cell Identifier (4/13)

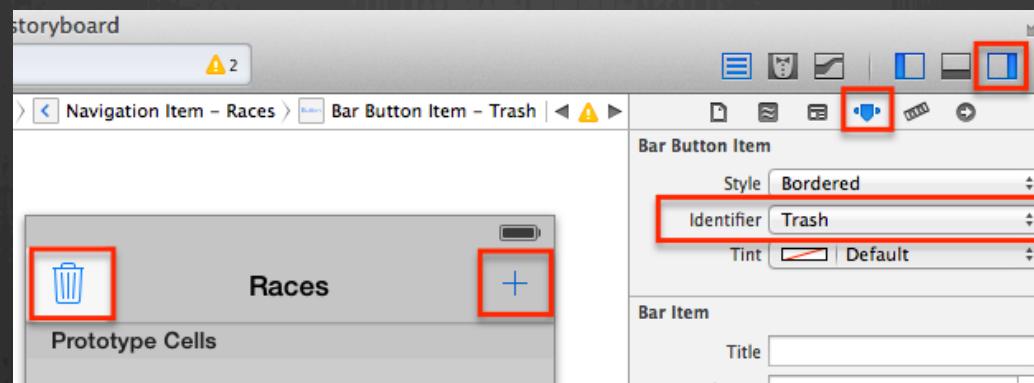
10. Click ที่ node “Table View Cell” ใน Document Outline และเปิด Attributes inspector ใน Inspector Pane และกรอก property “Identifier” ว่า “RaceCell”



11. เพิ่ม Navigation Controller เข้าไปใน Table View โดย click เลือก Table View ใน Storyboard (หรือ Document Outline) และ click ที่เมนู Editor > Embed In > Navigation Controller

Task : Embed in Navigation (5/13)

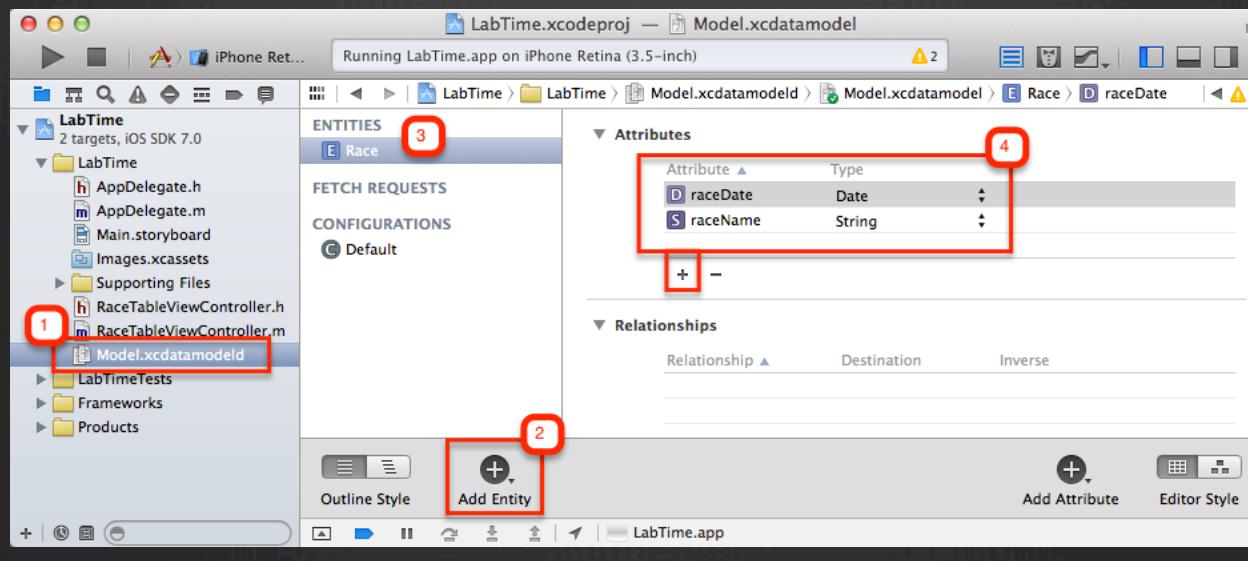
12. Double click บน Navigation Bar ของ Race Table View และตั้งชื่อ title ว่า “Races”
13. เพิ่ม “Bar Button Item” ที่มุมขวาบนของ Navigation Bar เปลี่ยน property “Identifier” ใน Attributes inspector เป็น “Add”
14. เพิ่ม “Bar Button Item” ที่มุมซ้ายบนของ Navigation Bar เปลี่ยน property “Identifier” ใน Attributes inspector เป็น “Trash”



15. เปลี่ยน editor mode เป็น Assistant editor และผูก IBAction ของปุ่ม Add กับ RaceTableViewController.h ตั้งชื่อว่า “btnAddTapped” และผูก IBAction ของปุ่ม Trash และตั้งชื่อว่า “btnDeleteTapped”

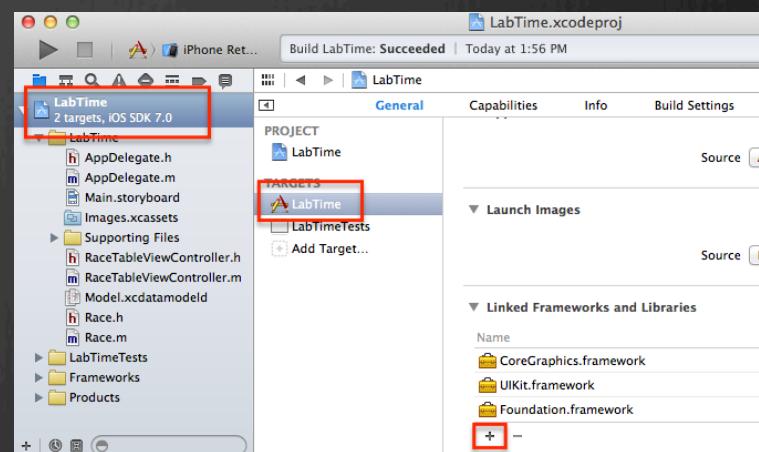
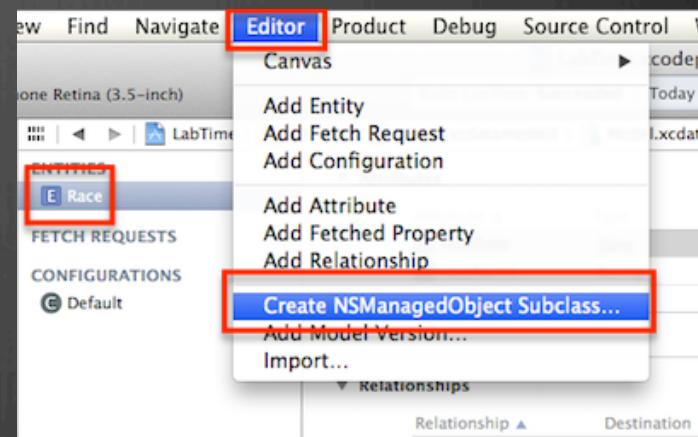
Task : Add Core Data (6/13)

16. เพิ่ม Core Data Model โดย click ขวาที่ project ใน Navigation Pane เลือก New File... > iOS > Core Data > Data Model แล้ว click ปุ่ม “Next” เลือก folder ที่จะ save และ click “Create” (ไม่ต้องเปลี่ยนชื่อไฟล์ - default ชื่อ Model.xcdatamodeld)
17. เปิดไฟล์ Model.xcdatamodeld และ click ปุ่ม “Add Entity” จากนั้น double click ที่ Entity เปลี่ยนชื่อเป็น “Race”
18. เพิ่ม attributes ของ Race 2 attributes คือ raceName : String และ raceDate : Date
Note : ชื่อ Attribute จะต้องขึ้นต้นด้วยตัวเล็กเท่านั้น



Task : Create Entity Class (7/13)

19. สร้าง class จาก Entity โดย click ที่ entity “Race” จากนั้น click ที่เมนู Editor > Create NSManagedObject Subclass... แล้ว click Next จากนั้นเลือก Race และ click “Create” จะได้ไฟล์ Race.h และ Race.m
20. เพิ่ม Library CoreData.framework เข้ามาใน project โดย click เลือก project ใน Navigation Pane และ click ที่ TARGETS > LapTime ให้ scroll ลงมาด้านล่างจะเห็น “Linked Frameworks and Libraries” และ click ที่ปุ่ม +
21. พิมพ์คำว่า CoreData ในช่อง search เลือก “CoreData.framework” และ click Add



Task : Add Helper Context (8/13)

22. เพิ่ม class ContextHelper เข้ามาใน project โดย click ขวาที่ folder “LapTime” แล้วเลือกเมนู Add Files to “LapTime...” จากนั้น browse ไปที่ “Resources 7/Day 3 – Lab11” แล้วเลือกไฟล์ ContextHelper.h และ ContextHelper.m และ click ปุ่ม “Add”

Note : ContextHelper เป็นตัวอย่าง wrapper class เพื่อให้เขียน code เรียก Managed Object Context (ORM) ง่ายขึ้น ส่วนการเขียน Core Data จริงๆ ให้ดูใน detail ของ method ต่างๆ

ใน class จะมี static method ต่างๆ ดังนี้

```
+ (NSManagedObjectContext *)getManagedObjectContext;
+ (id)insertNewObjectForEntityForName:(NSString *)className;
+ (void)saveContext;
+ (void)deleteObjectFromContext:(id)object;
+ (NSMutableArray *)getAllObjectsFromEntity:(NSString *)entityName
    initWithKey:(NSString *)keyName
    ascending:(bool)ascending;
```

Task : Add Core Data (9/13)

23. เปิดไฟล์ RaceTableViewController.h เพิ่ม property “races” เป็น type “NSMutableArray” เพื่อใช้เก็บ object Race

```
#import <UIKit/UIKit.h>

@interface RaceTableViewController : UITableViewController

@property(strong, nonatomic)NSMutableArray * races;
- (IBAction)btnAddTapped:(id)sender;
- (IBAction)btnDeleteTapped:(id)sender;

@end
```

24. เปิดไฟล์ RaceTableViewController.m เพิ่ม code การ import class “ContextHelper” และ “Race”

```
#import "RaceTableViewController.h"
#import "ContextHelper.h"
#import "Race.h"
```

Task : Read Objects from Core Data (10/13)

25. เพิ่ม code ใน method “viewDidLoad:” เพื่ออ่าน object “Race” ขึ้นมาจาก database

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.races = [ContextHelper getAllObjectsFromEntity:@"Race"
                                                initWithKey:@"raceName"
                                               ascending:YES];
}
```

26. แก้ Data Source method “numberOfSectionsInTableView:” และ “tableView:numberOfRowsInSection:” เพื่อบอก Table View ว่ามีจำนวน object Race กี่ object (ลบ #warning ออกรายการ)

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView
 numberOfRowsInSection:(NSInteger)section
{
    return [self.races count];
}
```

Task : Display in Table View (11/13)

27. แก้ code ใน method “tableView:cellForRowAtIndexPath:” เพื่อสร้าง cell ให้กับ Table View สำหรับแสดงข้อมูล Race

```
- (UITableViewCell *)tableView:(UITableView *)tableView
    cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"RaceCell";
    UITableViewCell *cell =
        [tableView dequeueReusableCellWithIdentifier:CellIdentifier
                                forIndexPath:indexPath];

    NSDateFormatter * dateFormatter = [[NSDateFormatter alloc] init];
    [dateFormatter setDateFormat:@"dd/mm/yyyy HH.mm.ss"];

    Race * race = [self.races objectAtIndex:indexPath.row];
    cell.textLabel.text = [NSString stringWithFormat:@"%@ [%@]",
                           race.raceName,
                           [dateFormatter stringFromDate:race.raceDate]];

    return cell;
}
```

28. Run โปรแกรม เพื่อดูผลลัพธ์

สังเกตว่าใน folder “Document” ภายใน Application จะมีไฟล์ “LapTime.sqlite” เราสามารถ copy file ทั้งหมดออกมารอแล้วเปิดดูข้อมูลได้ด้วย SQLite Viewer (*ดู path ของ file LapTime.sqlite ใน slide หน้า 12-03*)

Task : Add Object to DB (12/13)

29. เพิ่ม code ใน method “btnAddTapped:” เพื่อสร้าง object ใหม่และ save ข้อมูลลง Core Data รวมทั้ง reload ข้อมูลใน Table View ใหม่

```
- (IBAction)btnAddTapped:(id)sender
{
    Race *race = [ContextHelper insertNewObjectForEntityForName:@"Race"];
    race.raceName = [NSString stringWithFormat:@"Race - %lu",
                     (unsigned long)[self.races count]];
    race.raceDate = [NSDate date];

    [ContextHelper saveContext];

    self.races = [ContextHelper getAllObjectsFromEntity:@"Race"
                                                initWithKey:@"raceDate"
                                                ascending:NO];
    [self.tableView reloadData];
}
```

Task : Remove Objects (13/13)

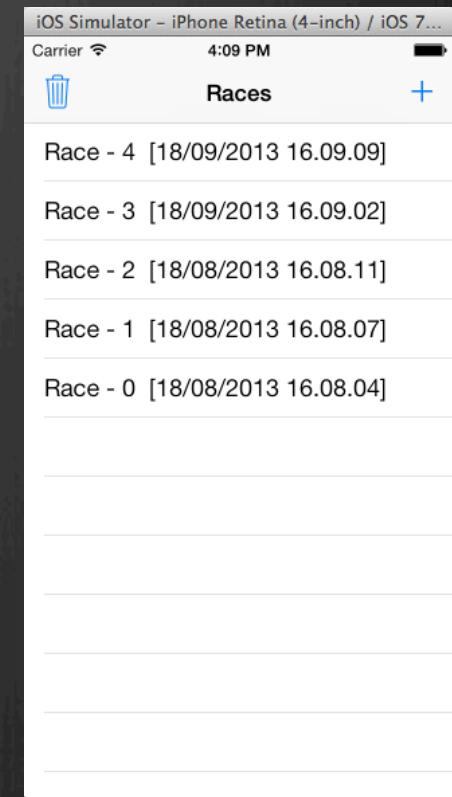
30. เพิ่ม code ใน method “btnDeleteTapped:” เพื่อลบ object ออกทั้งหมดและ save ข้อมูล (ที่ถูกลบ) ลง Core Data รวมทั้ง reload ข้อมูลใน Table View ใหม่

```
- (IBAction)btnDeleteTapped:(id)sender
{
    for (Race *r in self.races) {
        [ContextHelper deleteObjectFromContext:r];
    }
    [ContextHelper saveContext];

    [self.races removeAllObjects];

    self.races = [ContextHelper
                  getAllObjectsFromEntity:@"Race"
                  initWithKey:@"raceDate"
                  ascending:NO];
    [self.tableView reloadData];
}
```

31. Run โปรแกรมเพื่อดูผลลัพธ์



Lab 2-2 : Master/Detail (1/16)

◎ วัตถุประสงค์

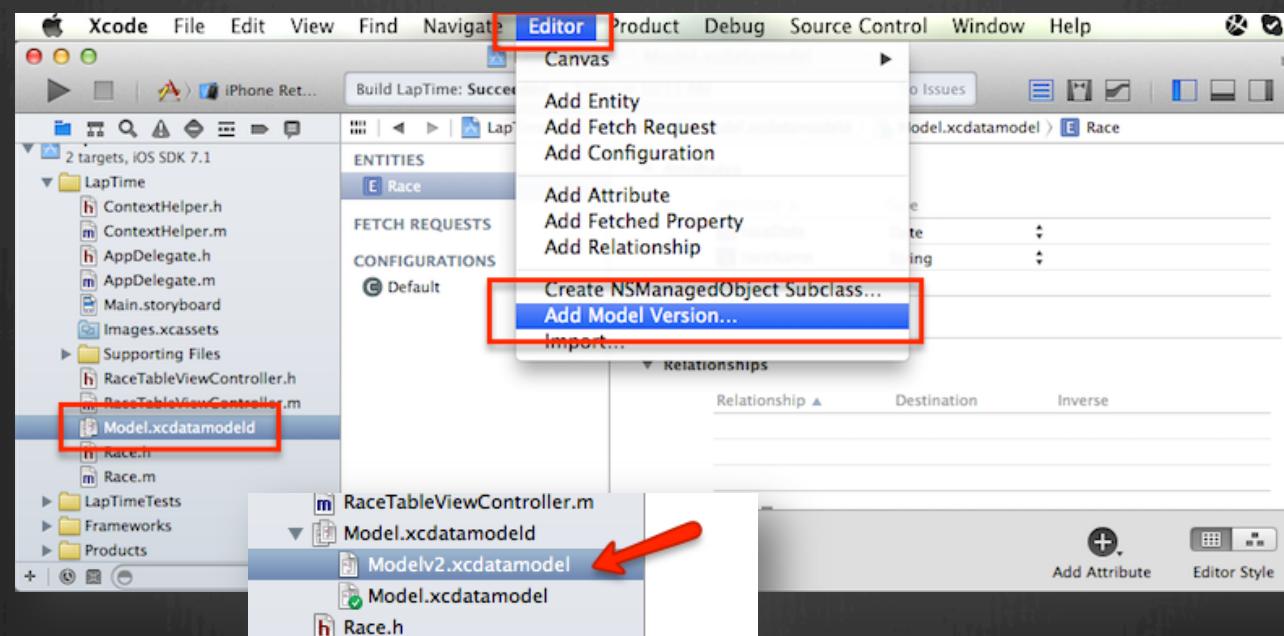
- ◎ เพื่อให้เข้าใจการใช้งาน Relationship ระหว่าง class ใน Core Data
- ◎ เข้าใจการทำ versioning บน CoreData ของ iOS
- ◎ เข้าใจการใช้ Date Formatter และการ compare time

◎ ขั้นตอน

- ◎ สร้าง Entity เพิ่ม และผูก relationship ระหว่าง Entity แบบ one-to-many หรือ master/detail
- ◎ สร้าง Table view มาแสดงผล Entity ที่เป็น detail
- ◎ เขียน code เพื่อจัดการกับ Entity detail
- ◎ เขียน code เพื่อเรียกอีก table view ใน Storyboard โดยใช้ Segue

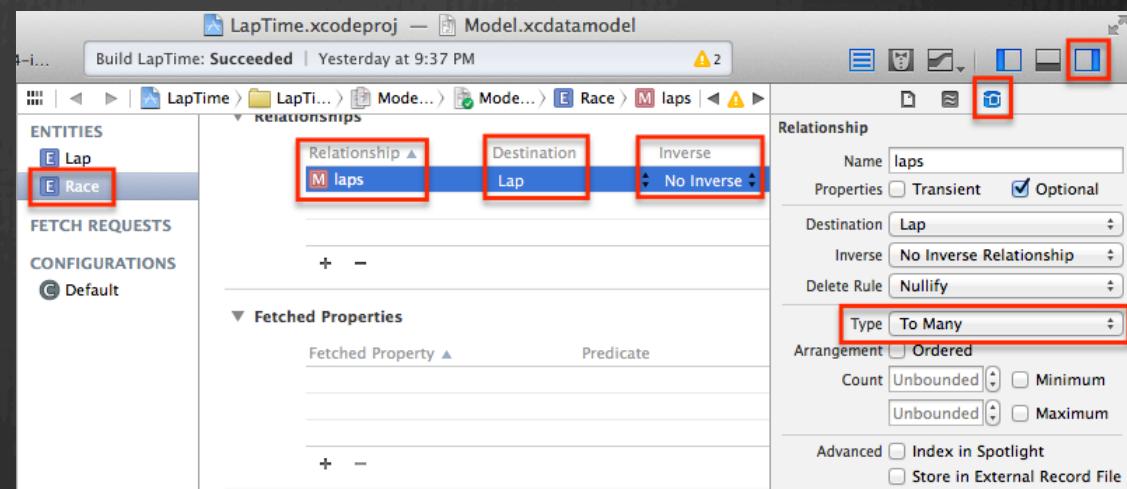
Task : Add Model Version (2/16)

- Click ที่ไฟล์ Model.xcdatamodeld จากนั้น click ที่เมนู Editor > Add Model Version... ตั้งชื่อ Model ว่า "Modelv2" สังเกตุว่าจะได้ไฟล์ Modelv2.xcdatamodel อยู่ใต้ group ของ fileเดิม



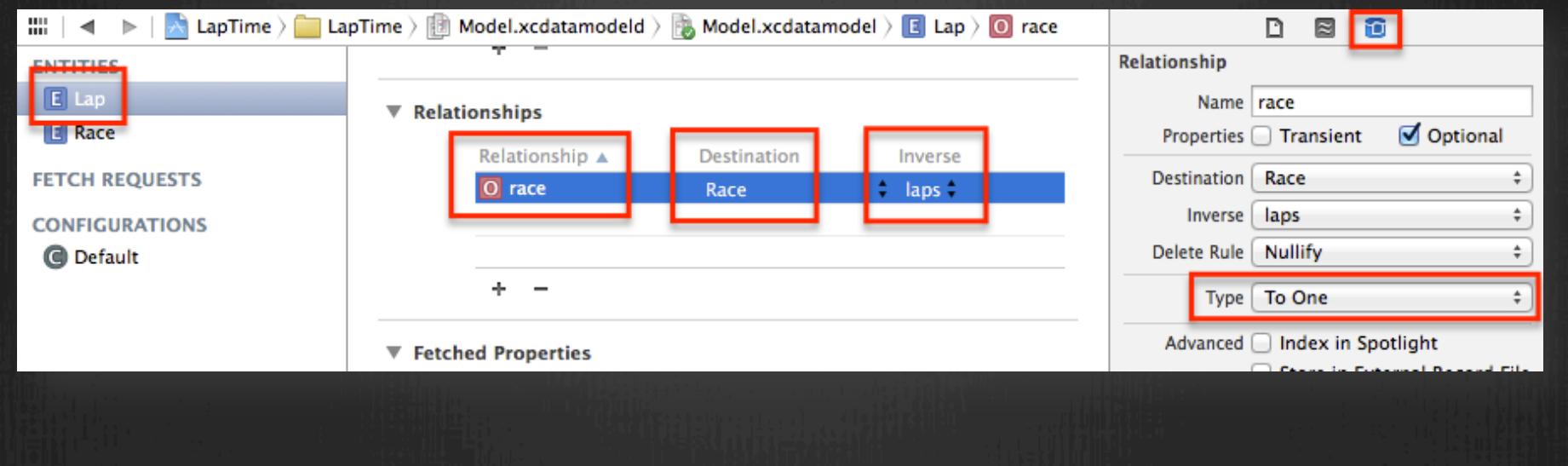
Task : Add Relationships (3/16)

2. ที่ไฟล์ Modelv2.xcdatamodeld เพิ่ม Entity ใหม่อีก 1 Entity ตั้งชื่อว่า “Lap”
3. เพิ่ม Attributes ให้กับ Entity Lap ชื่อ “**timeStamp**” ประเภท “Date”
4. เพิ่ม Relationship จาก Race ไปยัง Lap โดย
 - click ที่ Entity “Race” และเพิ่ม Relationship โดยกดปุ่ม + ใต้ block Relationship
 - ตั้งชื่อว่า Relationship ว่า “laps” (เดิมส่วนใน race 1 ครั้งมี lap หลายรอบ)
 - เปลี่ยนค่าของ Destination เป็น “Lap” (class ของปลายที่ relationship link ไป)
 - Click ที่ relationship “laps” และเปลี่ยน property “Type” ใน Inspector Pane เป็น “To Many”
 - เปลี่ยนค่า property “Delete Rule” เป็น “Cascade” เพื่อให้ Core Data ลบ Lap เมื่อเราลบข้อมูลใน Race



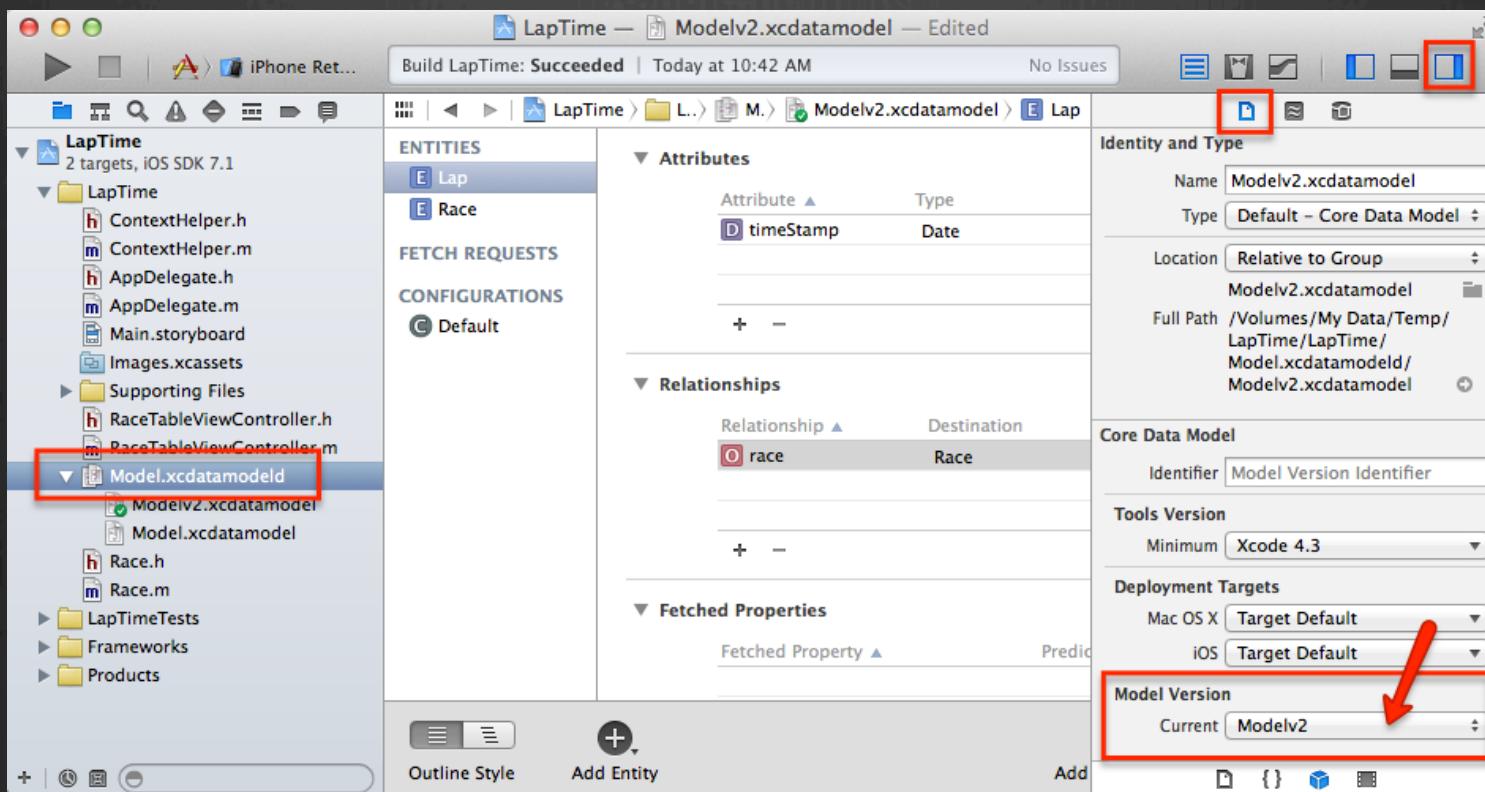
Task : Add Relationships (4/16)

5. เพิ่ม Relationship จาก Lap ไปยัง Race โดย
- click ที่ Entity “Lap” และเพิ่ม Relationship โดยกดปุ่ม + ใต้ block Relationship
 - ตั้งชื่อ Relationship ว่า “race” (เพราะ Lap แต่ละรอบอยู่ใน Race เดียว)
 - เปลี่ยนค่าของ Destination เป็น “Race”
 - เปลี่ยนค่าของ Inverse เป็น “laps”**
 - (เช็คความถูกต้อง) Click ที่ relationship “races” property “Type” ใน Inspector Pane ควรจะเป็น “To One”



Task : Add Relationships (5/16)

6. Click เลือกไฟล์ Model.xcdatamodeld และเปลี่ยน Inspector Pane เป็น “File Inspector” (icon ซ้ายสุดของ pane) จากนั้นทำการเปลี่ยนค่า Current ของ Model Version จาก Model เป็น “Modelv2”



Task : Update Entity Classes (6/16)

7. ลบไฟล์ Race.h และ Race.m ออกจาก project โดยกด Command ค้างไว้แล้ว click เลือกทั้ง 2 ไฟล์ จากนั้น click ขวา เลือก “Delete” และ click ปุ่ม “Move to Trash”
8. Click เลือก Entity Race และ Lap ใน Model.xcdatamodeld แล้วเลือกเมนู “Editor > Create NSManagedObject Subclass...” จะมี dialog ให้เลือกว่าจะ เอา Entity จาก Model version ไหน ให้เลือก Modelv2 เพียงตัวเดียว
9. เลือก Race และ Lap ใน dialog หน้าถัดไป click Next แล้ว click Create จะได้ Model class “Race” และ “Lab”

Note: สังเกตุ code ของ class Race จะมี property “laps” เป็น set (to-many) พร้อมทั้งมี add / remove ให้ด้วย ส่วน code ของ class Lap จะมี property “race” เป็น object (to-one) ตามที่เรากำหนดไว้ใน relationship

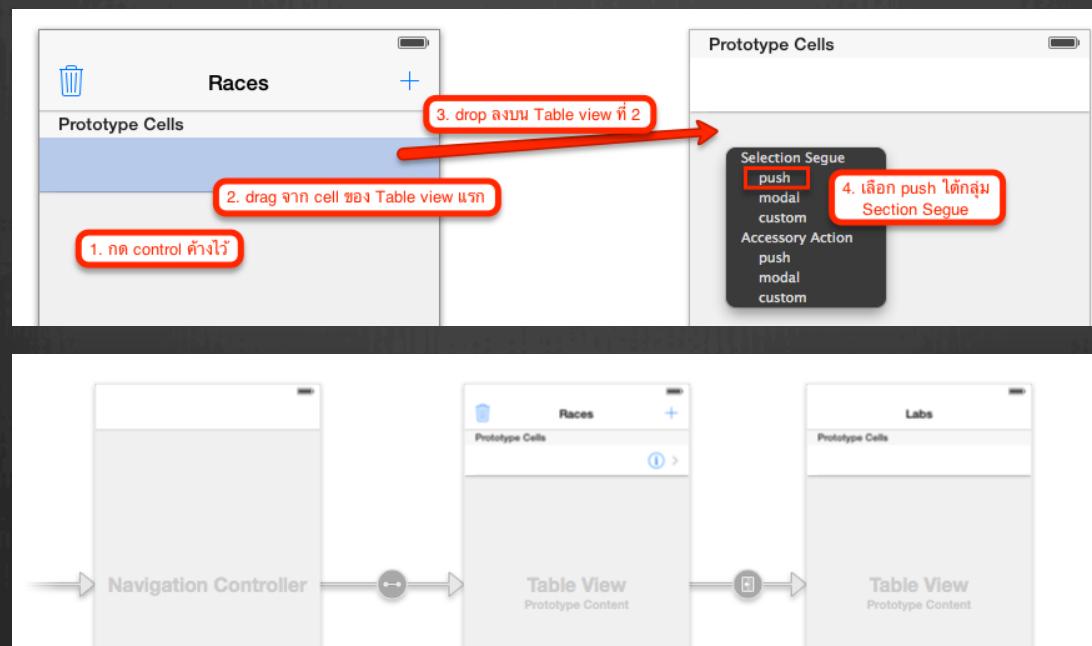
Task : Add Auto-Migration Code (7/16)

10. เปิดไฟล์ ContextHelper.m เพิ่ม code ใน method “getManagedObjectContext” เพื่อให้ app ทำการ migrate version ของ schema อัตโนมัติ ดังนี้

```
+ (NSManagedObjectContext *)getManagedObjectContext {
    .....
    managedObjectModel_ = [[NSManagedObjectModel alloc] initWithContentsOfURL:modelPath];
    NSDictionary * options = [NSDictionary dictionaryWithObjectsAndKeys:
        [NSNumber numberWithBool:YES],
        NSMigratePersistentStoresAutomaticallyOption,
        [NSNumber numberWithBool:YES],
        NSInferMappingModelAutomaticallyOption, nil];
    NSError *error = nil;
    persistentStoreCoordinator_ =
        [[NSPersistentStoreCoordinator alloc] initWithManagedObjectModel:managedObjectModel_];
    if (![persistentStoreCoordinator_ addPersistentStoreWithType:NSSQLiteStoreType
                                                 configuration:nil
                                                       URL:dbPath
                                                       options:options
                                                       error:&error])
    {
        NSLog(@"Unresolved error %@", error, [error userInfo]);
        abort();
    }
    .....
}
```

Task : Create Segue (8/16)

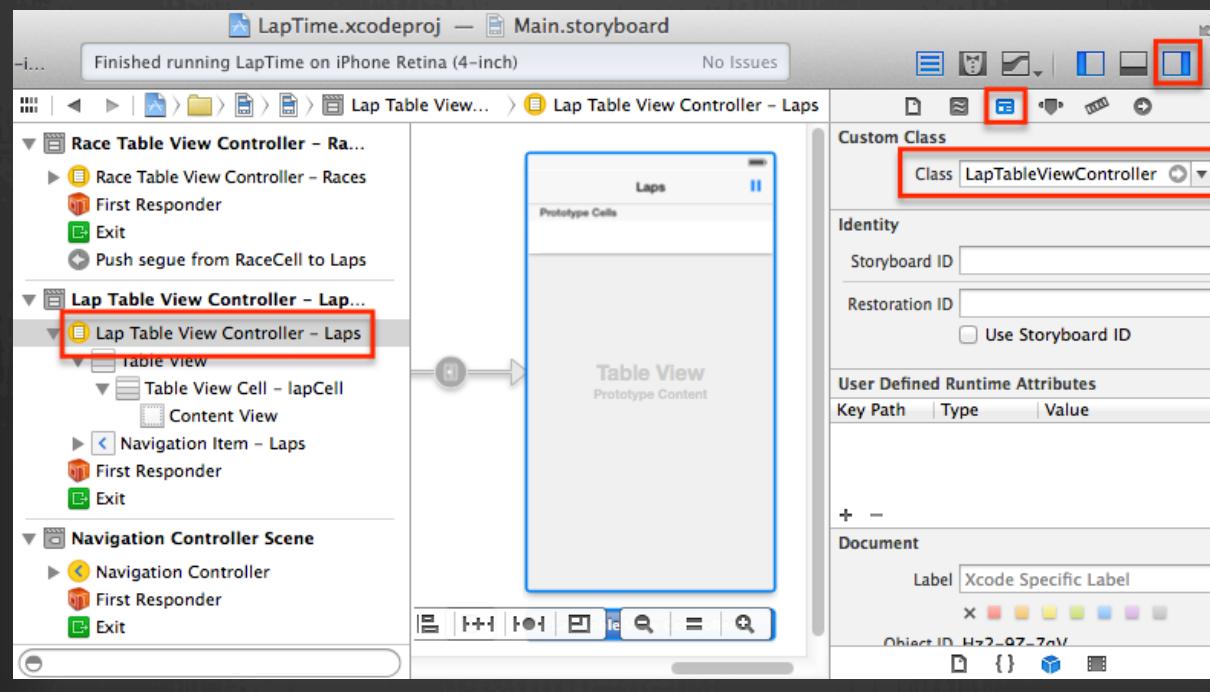
11. เปิดไฟล์ Main.storyboard เพิ่ม Table View Controller จาก Library Pane ลงบน Storyboard อีก 1 view
12. กดปุ่ม Control บน keyboard ค้างไว้แล้ว drag จาก cell ของ Table View แรกไปยัง Table view ที่ 2 จะปรากฏ popup เมนูขึ้นมา ให้เลือก push. ตัวกลุ่ม Selection Action จะได้เล่นที่เชื่อมจาก table view แรกไปยัง table view ตัวที่สอง เล่นที่ปรากฏขึ้นมาเรียกว่า “Segue”



13. Double ที่ title ของ Navigation Bar ของ Table view ที่ 2 และเปลี่ยน title เป็น “Laps”

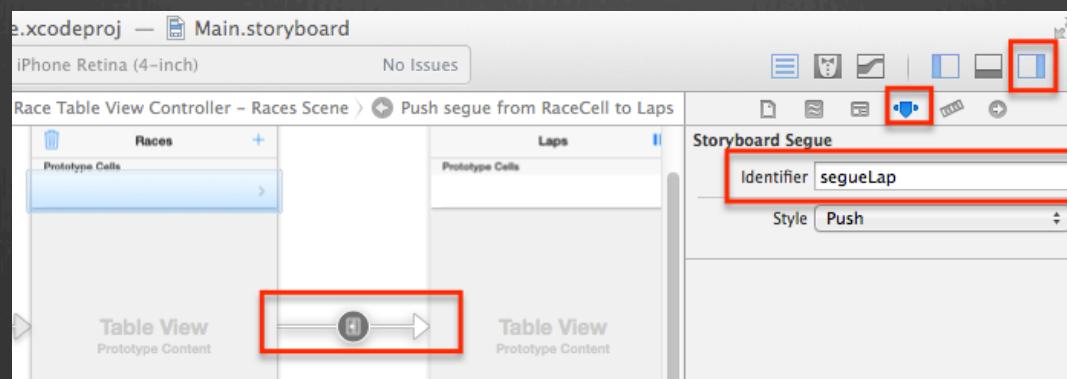
Task : Add view controller (9/16)

14. เพิ่ม class ใหม่ จากเมนู File > New > File ... > iOS > Cocoa Touch > Objective-C class จากนั้น click “Next” เลือก Subclass of เป็น “UITableViewController” และตั้งชื่อ class เป็น “LapTableViewController”
15. เปิด Main.storyboard เลือก Table view “Laps” และเปลี่ยน property “Class” ที่ Identity inspector ใน Inspector Pane เป็น “LapTableViewController”

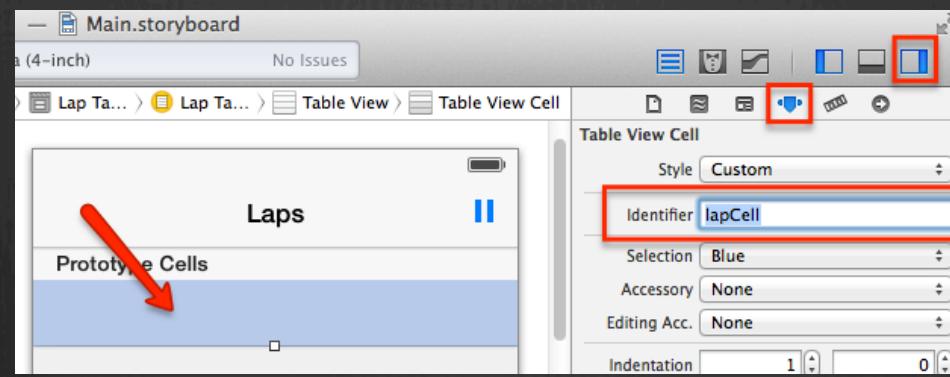


Task : Identify Segue & Cell (10/16)

16. Click ที่ segue ที่ลูกเพิ่มเข้ามา แล้วเปิด Attribute Inspector กำหนดค่า property “Identifier” เป็น “segueLap”

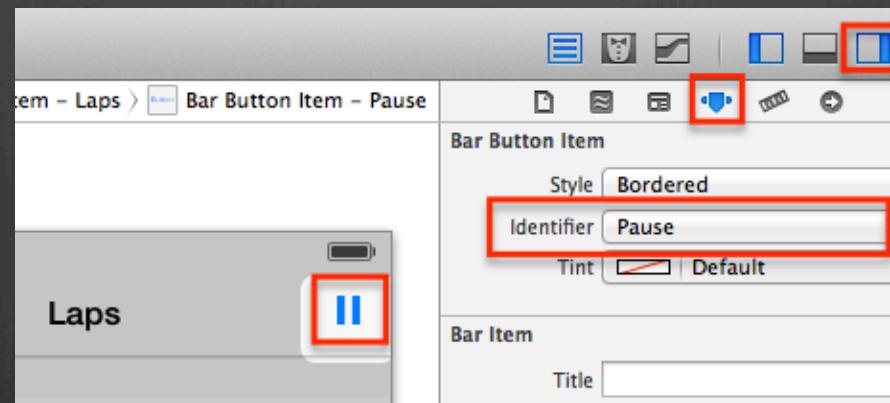


17. Click ที่ Table view cell ของ table view Lap แล้วกำหนด property “Identifier” เป็น “lapCell”



Task : Binding Button Action (11/16)

18. เพิ่ม control “Bar Button Item” ลงบน Navigation Bar ของ Table view ด้านขวาบน และเปลี่ยน property “Identifier” เป็น “Pause”



19. เปลี่ยน editor mode เป็น Assistant editor ผูก IBAction ของปุ่ม Pause ไปยังไฟล์ “LapTableViewController.h” และตั้งชื่อ action ว่า “btnPauseTapped”

Task : Coding for Detail (12/16)

20. ที่ไฟล์ LapTableViewController.h เพิ่ม #import และ property “race” ใน ดังนี้

```
#import <UIKit/UIKit.h>
#import "ContextHelper.h"
#import "Race.h"
#import "Lap.h"

@interface LapTableViewController : UITableViewController

@property (nonatomic, strong) Race * race;

- (IBAction)btnPauseTapped:(id)sender;

@end
```

21. เปิดไฟล์ “LapTableViewController.m” แก้ code ใน method “numberOfSectionsInTableView:” และ “tableView:numberOfRowsInSection:”

```
- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return [self.race.laps count];
}
```

Task : Coding for Detail (13/16)

22. แก้ code ใน method “tableView:cellForRowAtIndexPath:” ในไฟล์ “LapTableViewController.m”

```

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    static NSString *CellIdentifier = @"lapCell"; // ตามที่กำหนดไว้ในข้อ 14.
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:CellIdentifier
                                                       forIndexPath:indexPath];
    if (!cell)
        cell = [[UITableViewCell alloc] initWithStyle:UITableViewCellStyleDefault
                                      reuseIdentifier:CellIdentifier];
    // Sort Lap objects
    NSArray *laps = [self.race.laps allObjects];
    NSSortDescriptor* sortDescriptor = [NSSortDescriptor sortDescriptorWithKey:@"timeStamp"
                                                               ascending:NO];
    NSArray* sortedArray = [laps sortedArrayUsingDescriptors:[NSArray arrayWithObject:sortDescriptor]];
    Lap * lap = [sortedArray objectAtIndex:indexPath.row];

    // Time to string
    NSDateFormatter *dateFormatter = [[NSDateFormatter alloc] init];
    dateFormatter.dateFormat = @"HH:mm:ss";
    cell.textLabel.text = [dateFormatter stringFromDate:lap.timeStamp];

    // compare time
    Lap *previousLap = nil;
    if ([sortedArray count] > (indexPath.row + 1))
        previousLap = [sortedArray objectAtIndex:(indexPath.row + 1)];

    if (previousLap) {
        NSTimeInterval timeDifferent = [[lap timeStamp] timeIntervalSinceDate:[previousLap timeStamp]];
        cell.textLabel.text = [cell.textLabel.text stringByAppendingString:
                               [NSString stringWithFormat:@" +%.02f sec", timeDifferent]];
    }
    return cell;
}

```

Task : Add Detail Object (14/16)

23. เพิ่ม code ใน method “btnPauseTapped:” เพื่อเพิ่ม object ของ Lap ใหม่

```
- (IBAction)btnPauseTapped:(id)sender
{
    Lap * newLap = [ContextHelper insertNewObjectForEntityForName:@"Lap"];
    newLap.timeStamp = [NSDate date];
    newLap.race = self.race;
    [self.race addLapsObject:newLap];

    [ContextHelper saveContext];
    [self.tableView reloadData];
}
```

24. เปิดไฟล์ “RaceTableViewController.m” เพิ่ม code #import

```
#import "RaceTableViewController.h"
#import "ContextHelper.h"
#import "Race.h"
#import "LapTableViewController.h"
```

Task : Call Detail View (15/16)

25. ในไฟล์ “RaceTableViewController.m” เอา comment ของ method “prepareForSegue:sender:” ออฟ แล้วเพิ่ม code ดังนี้

```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"segueLap"]) {
        LapTableViewController * lapTableViewController =
            [segue destinationViewController];

        long row = self.tableView.indexPathForSelectedRow.row;
        lapTableViewController.race = [self.races objectAtIndex:row];

        [lapTableViewController.tableView reloadData];
    }
}
```

Task : Run & Test (16/16)

26. Run โปรแกรมเพื่อดูผลลัพธ์

- Click ปุ่ม + เพื่อสร้าง Race ใหม่
- Click ที่ cell ใน table ที่มีข้อมูล Race หน้าจอจะ push ไปยัง table ที่ 2 (ความสามารถของ Navigation Control)
- Click ปุ่ม Pause จะได้รายการ Lap แรก เมื่อ click อีกครั้งจะได้รายการ Lap ที่ 2 ขึ้นไปอยู่บนสุด และบอกเวลาที่ diff

