

---

*TD 05 : Core Data*

---

Ce TD portera sur l'utilisation de Core Data et la persistance de données

## 1. Ma liste d'étudiants

Dans cette application, grand classique des bases de données, vous gérez une table unique d'étudiants, avec l'ajout, la suppression, la recherche d'un étudiant.

Pour réaliser cette application, à l'aide d'un storyboard, vous utiliserez :

- un view controller
- une tableview
- des boutons pour commander les ajouts, effacements et reset de toutes les données
- des UITextField pour permettre à l'utilisateur d'entrer les données de ses propres étudiants

Vous utiliserez aussi Core Data pour déclarer un xcdatamodeld avec:

- une entité (entity)
- des attributs de type nom (string), prenom (string) note (float) numero(string)
- le NSManagedObject (classe) et son header associé

En Objective C vous utiliserez :

- NSRequest comme container d'exécution de requête
- NSEntityDescription pour déclarer l'entity à utiliser
- Eventuellement un NSSortDescriptor
- NSPredicate pour générer une requête
- NSManagedObjectContext pour récupérer le contexte

Pensez à créer un projet sous XCode en cochant la case coredata dans un single view.

Dans un premier temps, créez le modèle (xcdatamodel)

Dans un deuxième temps, dans la classe AppDelegate.m, testez votre modèle en ajoutant des données et en travaillant dans la méthode :

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
```

Exemple de code qui affiche dans la console le contenu d'une table à partir de son entity.

```
NSManagedObjectContext *context = [self managedObjectContext];
NomPrenomMatiere * etudiant1 =
[NSEntityDescription insertNewObjectForEntityForName:@"NomPrenomMatiere"
inManagedObjectContext:context];

etudiant1.nom = @"vernes";
etudiant1.prenom = @"jules";
etudiant1.numero = @"987654";
etudiant1.note = [NSNumber numberWithInt:8.0f];
[context insertObject:etudiant1];

NSError *error;
if (![context save:&error]) {
    NSLog(@"Whoops, couldn't save: %@", [error localizedDescription]);
}

NSFetchRequest * fetchRequest = [[NSFetchRequest alloc] init];
NSEntityDescription * entity = [NSEntityDescription
entityForName:@"NomPrenomMatiere" inManagedObjectContext:context];
[fetchRequest setEntity:entity];

NSArray * fetchedObjects = [context executeFetchRequest:fetchRequest error:&error];
for (NSManagedObject *info in fetchedObjects) {
    NSLog(@"Name: %@", [info valueForKey:@"nom"]);
}
```

Dans un troisième temps, ajoutez la tableview et les interactions avec l'utilisateur.

Enfin, associez le tout à des actions de chargement, de remove etc... avec Core Data.

Au final :

- l'utilisateur doit pouvoir renseigner les données d'un nouvel étudiant pour l'ajouter
- l'utilisateur doit pouvoir effacer une entrée dans la tableview et la table et l'affichage doivent être mis à jour automatiquement
- l'utilisateur doit pouvoir effacer toute la table avec une seule action
- l'utilisateur doit pouvoir afficher la liste des étudiants ayant des notes inférieures ou supérieures à 10 dans la table view
- vous afficherez la moyenne sur l'ensemble des étudiants existants dans un champ à part

Pour aller plus loin :

- utilisez un deuxième VC pour gérer les interactions avec l'utilisateur (ajout, effacement etc.)
- un swipe sur une cellule de tableview permet d'afficher un bouton delete ... essayez d'implémenter cette feature
- utilisez des opérations dans les prédicats (sum par exemple)

Conseils :

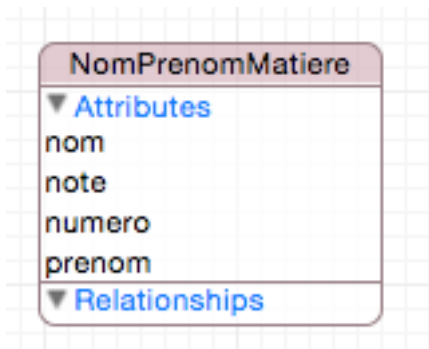
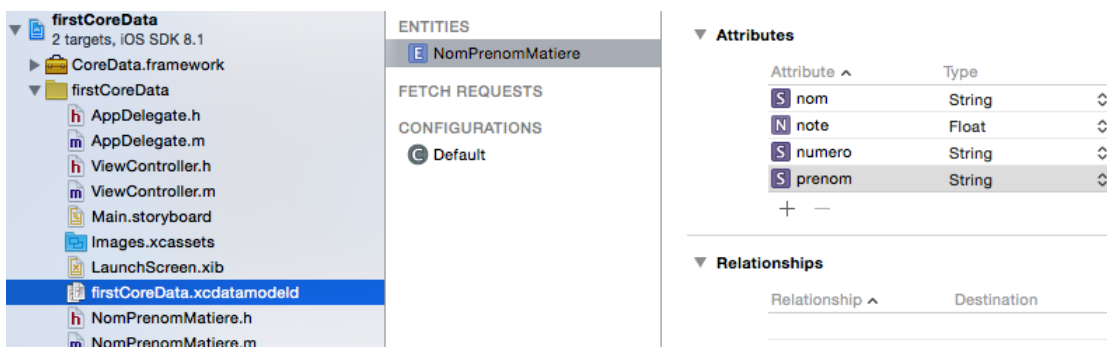
Pensez à ajouter vos entrées dans la table

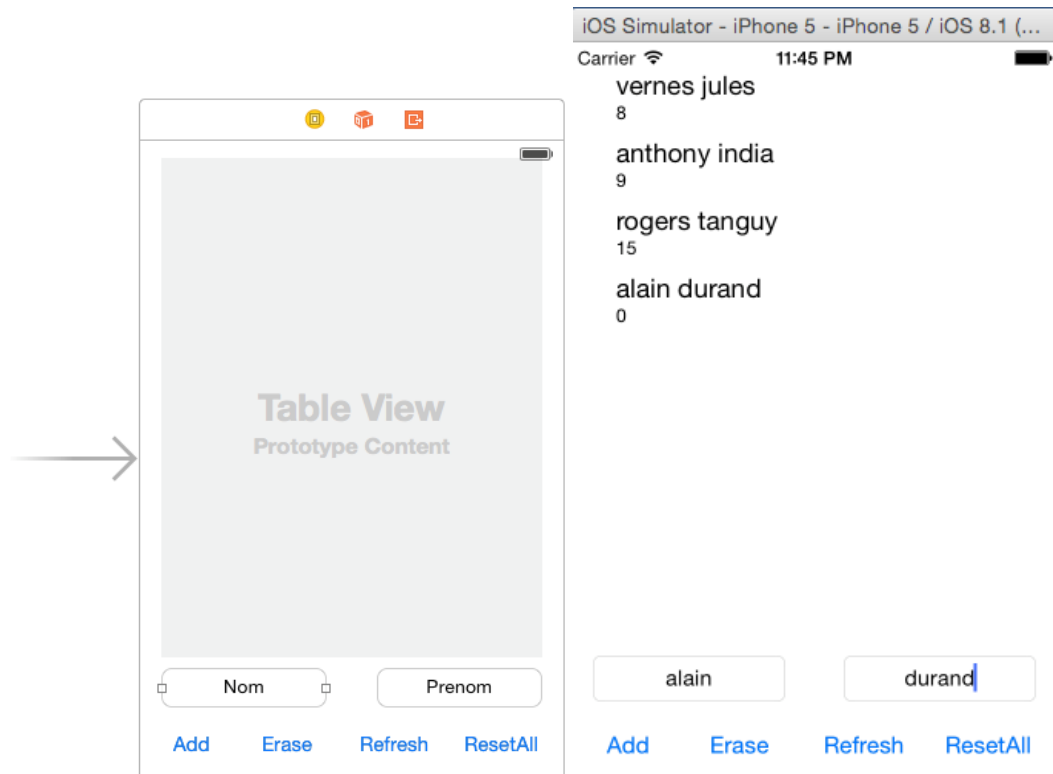
Pensez à sauvegarder (commiter) vos entrées au moins une fois avant de sortir de l'application

Si vous changez de modèle de table, alors que vous avez déjà lancé l'application au moins une fois, pensez à effacer l'application du simulateur et à la lancer à nouveau. Cela a pour effet de recréer la sauvegarde des données dans le nouveau format.

Pensez à fixer le format de vos entrées dans la table ... un nom est composé de 5 à n lettres etc...

Résultat attendu sous storyboard, sous éditeur et sous émulateur:





Ces 2 captures ne sont pas représentatives de l'application finale, elles ne sont qu'un guide.