```swift
//
//  ViewController.swift
//  cd_demo1
//
//  Created by MAC on 2/7/23.
//

import UIKit
import CoreData

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }

    @IBAction func btn_add(_ sender: Any) {
        let appD = UIApplication.shared.delegate as! AppDelegate
        let context = appD.persistentContainer.viewContext
        let newitem = NSEntityDescription.insertNewObject(forEntityName: "Stud", into: context)
        newitem.setValue(2, forKey: "sid")
        newitem.setValue("Manish", forKey: "sname")
        newitem.setValue(28, forKey: "sage")
        do{

            try context.save()
            print("Record inserted")

        }catch
        {
            print("Error in inerting record")
        }


    }

    @IBAction func btn_display(_ sender: Any) {
        let appD2 = UIApplication.shared.delegate as! AppDelegate
        let context2 = appD2.persistentContainer.viewContext
```

```swift
    let fr = NSFetchRequest<NSFetchRequestResult>(entityName: "Stud")
    do{
        let result = try context2.fetch(fr)
        for data in result as! [NSManagedObject]
        {
            print(data.value(forKey: "sid")!)
            print(data.value(forKey: "sname")!)
            print(data.value(forKey: "sage")!)
        }
    }
    catch{
        print("Error in fetching")
    }
}


@IBAction func btn_update(_ sender: Any) {
    let appD3 = UIApplication.shared.delegate as! AppDelegate
    let context = appD3.persistentContainer.viewContext
    let fr = NSFetchRequest<NSFetchRequestResult>(entityName: "Stud")
    fr.predicate = NSPredicate(format: "sname=%@", "Manish")
    do{
        // update code
        /*
        let test = try context.fetch(fr)
        let obj = test[0] as! NSManagedObject
        obj.setValue("Aashish", forKey: "sname")
        */

        //Delete code
        let test = try context.fetch(fr)
        let obj = test[0] as! NSManagedObject
        context.delete(obj)

        do{
            try context.save()
        }catch
        {

        }

    }catch{
```

```
        }

    }

}
```