

```

import UIKit
import ArcGIS

class RelatedFeaturesListViewController: UIViewController,
UITableViewDataSource, UITableViewDelegate {

    @IBOutlet weak var tableView: UITableView!

    var censusArray = [[String: String]]()

    // results required for display
    var results = [AGSRelatedFeatureQueryResult]() {
        didSet {
            if let result = results.first {
                title = result.feature?.attributes["DIST_NO"] as? String ??
"Origin Feature"
            }
        }
    }

    override func viewDidLoad() {
        super.viewDidLoad()

        Create_CensusTable ()

    }

    func Create_CensusTable () {

        if let result = results.first {

            let feature = result.featureEnumerator().allObjects

            let dist = result.feature?.attributes["DIST_NO"] as! Int
            let male = result.feature?.attributes.value(forKey: "MALE") as? Int
            let female = result.feature?.attributes.value(forKey: "FEMALE") as?
Int

            print (dist, male, female)

        }

    }

    // MARK: - UITableViewDataSource

    func numberOfSections(in tableView: UITableView) -> Int {
        return results.count
    }

    func tableView(_ tableView: UITableView, titleForHeaderInSection section:
Int) -> String? {
        let result = results[section]
        if let tableName = result.relatedTable?.tableName {
            return "\(tableName)"
        } else {
            return "Related table \(section + 1)"
        }
    }
}

```

```

    func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int)
-> Int {
    let result = results[section]
    return result.featureEnumerator().allObjects.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath)
-> UITableViewCell {
    let result = self.results[indexPath.section]
    let feature = result.featureEnumerator().allObjects[indexPath.row]

    let displayFieldName = result.relatedTable!.layerInfo!.displayFieldName
    let displayFieldValue = feature.attributes[displayFieldName!]

    let cell = tableView.dequeueReusableCell(withIdentifier:
"RelatedFeatureCell", for: indexPath)

    cell.textLabel?.text = "\(displayFieldName)"
    cell.detailTextLabel?.text = "\(displayFieldValue)"

    return cell
    }
}

```