App.MapGet("/PlantsInfo", ([FromServices] ModelContext db) =>

{

    return db.plants.ToList();

}

App.MapGet("/BooksInfo", ([FromServices] ModelContext db) =>

{

    var books= db.Books.ToList();

    var authors= db.Authors.ToList();

var merge =

}

persons.Select(p => new Student

{

id = p.id,

name = p.name,

isStudying = students.Any(s => s.id == p.id

&& s.isStudying)

});

select Nombre, Titulo from Books, Persona p

where author = p.id

var query = from A in Books

join B in Persona

on A.author equals B.id

select new

{

B.Nombre,

A.Titulo

};

return query;

var results = workOrders.Join(plans,

wo => wo.WorkOrderNumber,

p => p.WorkOrderNumber,

(order,plan) => new {order.WorkOrderNumber, order.WorkDescription, plan.ScheduledDate}

);

var query = from A in workOrders

join B in plans

on A.WorkOrderNumber equals B.WorkOrderNumber

select new

{

A.WorkOrderNumber,

A.Description,

B.ScheduledDate

};

var students = new List<Student> { new Student { id = 2, name = "Ace", isStudying = true },

new Student { id = 3, name = "Mike", isStudying = true },

new Student { id = 5, name = "Ken", isStudying = true } };

var allPersons = (from P in persons

join S in students on new { first = p.id } equals new { first = s.id } into sjoin

from slj in sjoin.DefaultIfEmpty()

select new

{

id = p.id,

name = p.name,

isStudying = (slj != null ? (slj.isStudying ? "TRUE" : "FALSE") : string.Empty)

}).ToList();

<https://unaura.com/join-keyword-with-linq-c/>

var cdbips = from cdb in CDBs

join ip in IPs

on cdb.Cdbno equals ip.Cdbno

orderby cdb.Cdbno descending

select new classIP(cdb.Cdbno, ip.Ipno);

var cdbips = from cdb in CDBs

join ip in IPs

on cdb.Cdbno equals ip.Cdbno

orderby cdb.Cdbno descending

select new classIP(cdb.Cdbno, ip.Ipno);

foreach (var cdbip in cdbips)

{

CDBIPs.Add(cdbip);

}

var cdbips = from cdb in CDBs

join ip in IPs

on cdb.Cdbno equals ip.Cdbno

orderby cdb.Cdbno descending

select new classIP(cdb.Cdbno, ip.Ipno);