**One to one relation in Dapper**

**Getting data with inner join query**

var sql = @"SELECT Capitals.CapitalId,Capitals.Name,Capitals.CountryId,

Countries.CountryId, Countries.Name

FROM Capitals

INNER JOIN Countries

ON Capitals.CountryId = Countries.CountryId"

var capitals = connection.Query<Capital, Country, Capital>(sql,

(capital, country) =>

{

capital.Country = country;

return capital;

}, splitOn: "CountryId").ToList();

**One to Many relation in Dapper**

**Getting data with inner join query**

1)

//var sql = @"SELECT S.StudentId, S.Firstname, S.Age,

// G.GroupId,G.Title

// FROM Students AS S

// INNER JOIN Groups AS G

// ON S.GroupId=G.GroupId";

//var students = connection.Query<Student, Group, Student>(sql,

// (student, group) =>

// {

// student.Group = group;

// student.GroupId = group.GroupId;

// return student;

// }, splitOn: "GroupId").ToList();

//MyDataGrid.ItemsSource = students;

2)

var sql = @"SELECT G.GroupId,G.Title,

S.StudentId, S.Firstname, S.Age

FROM Groups AS G

INNER JOIN Students AS S

ON S.GroupId=G.GroupId";

var groups = connection.Query<Group, Student, Group>(sql,

(group, student) =>

{

group.Students.Add(student);

student.GroupId = group.GroupId;

student.Group = group;

return group;

}, splitOn: "StudentId").ToList();

**Notes:**

groupun icine student lari almaq ucun group.students.Add(student); edtik

studentin de icinde groupu getirmek ucun student.GroupId = group.GroupId etdik.

splitOn: "StudentId - foreign key neye gore baglanib, neye gore elemenleri getirmeli oldugunu bildirir

connection.Query<Group, Student, Group> - burada sonuncu return tipidir. digerleri ise gelen datalarin tipleridir.

**Many To Many Relation**

**Burada ise artiq bele edirik**

var As= connection.Query<A, B, A>(sql,

(A, B) =>

{

A.Collection.Add(B);

B.Collection.Add(A);

}, splitOn: "ForeignKeyId").ToList();

Code links:

<https://github.com/Drongo-J/DapperOneToOneRelations.git>

https://github.com/Drongo-J/DapperOneToManyRelations.git