For what it is worth, I tried to map the ODATA requests in 7.2 of in [odata-data-aggregation](http://docs.oasis-open.org/odata/odata-data-aggregation-ext/v4.0/cs02/odata-data-aggregation-ext-v4.0-cs02.html#_Toc435016610) to LINQ.

[LINQ](https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/%5D) is the technology in .NET that allows to express queries – integrated in the programming language - that allows to be translated to various query languages (including but not limited to SQL) .

I create the following dataset (all in memory)

 Id| Amount| Currency/Code| Product/Name

 1001| 6| USD| Coffee

 1002| 6| USD| Coffee

 1003| 3| EUR| Paper

 1004| 5| USD| Paper

 1005| 2| EUR| Sugar

 1006| 2| USD| Sugar

## Example 53:

GET ~/Products?$apply=groupby((Name),

aggregate(Sales/Amount with sum as Total))

can be translated to

 from product in data.Products

 group product by product.Name into grp

 select new

 {

 Name = grp.Key,

 Sales = new

 {

 Total = (

 from p in grp

 from s in p.Sales

 select s.Amount

 ).Sum()

 }

 }

Please Note that there are two steps. First group by name. Next, for each create one record with the sum of all sales amounts. The sum must be built over a collection of collections (each element of a group (a product) has many sales records hence the two inner selects).

Which leads to the result:

[

 {

 "Name": "Coffee",

 "Sales": {

 "Total": 12

 }

 },

 {

 "Name": "Paper",

 "Sales": {

 "Total": 8

 }

 },

 {

 "Name": "Pencil",

 "Sales": {

 "Total": 0

 }

 },

 {

 "Name": "Sugar",

 "Sales": {

 "Total": 4

 }

 }

]

Please note that in the specification, the Sales properties value is a collection (of an object with a Total property) whereas here it is a single object.

## Example 54

GET ~/Products?$apply=groupby((Name,Sales/Currency/Code),

 aggregate(Sales/Amount with sum as Total))

Could be translated to the following, essentially just changing the previous query in the way the key is formed: it was just the name and now it is the record consisting of Name and currency/code.

    from p in data.Products

    group p by new {

        Name = p.Name,

        Code = from s in p.Sales select s.Currency.Code

    } into grp

    select new

    {

        Name = grp.Key.Name,

        Code = grp.Key.Code,

        Sales = new

        {

            Total = (

                from p in grp

                from s in p.Sales

                select s.Amount

            ).Sum()

        }

    }

This leads to the wrong result

[

 ...,

 {

 "Name": "Paper",

 "Code": [

 "EUR",

 "USD"

 ],

 "Sales": {

 "Total": 8

 }

 },

 ...

Something more must be done to ensure that there is one unique key per name and currency-code combination, a sort of preprocessing step that in some way splits a single product entity in one per currency with only the sales records of the given currency.

    from split in (

        from p in data.Products

        from c in (

                from s in p.Sales

                select new { Code = s.Currency.Code }

            ).Distinct()

        select new

        {

            // p.Id,

            p.Name,

            c.Code,

            Sales = from s in p.Sales where s.Currency.Code == c.Code select s

        }

    )

    group split by new { split.Name, split.Code } into grp

    select new

    {

        Name = grp.Key.Name,

        Code = grp.Key.Code,

        Sales = new

        {

            Total = (

                from p in grp

                from s in p.Sales

                select s.Amount)

            .Sum()

        }

    }

Which leads to the following result. Please note that the two grouping keys name and currency code appear on the top level, different from what the specification shows.

 {

 "Name": "Coffee",

 "Currency": {

 "Code": "USD"

 },

 "Sales": {

 "Total": 12

 }

 },

 {

 "Name": "Paper",

 "Currency": {

 "Code": "EUR"

 },

 "Sales": {

 "Total": 3

 }

 },

 {

 "Name": "Paper",

 "Currency": {

 "Code": "USD"

 },

 "Sales": {

 "Total": 5

 }

 },

 {

 "Name": "Sugar",

 "Currency": {

 "Code": "EUR"

 },

 "Sales": {

 "Total": 2

 }

 },

 {

 "Name": "Sugar",

 "Currency": {

 "Code": "USD"

 },

 "Sales": {

 "Total": 2

 }

 }

]