

Grouping (GROUP)

The *group* operator creates a [property](#) that divides all object collections in the system into *groups*, and calculates an [aggregating function](#) for each group following specified *order*. Accordingly, the set for which this aggregating function is calculated is determined as all the object collections belonging to this group.

Groups are defined for this operator as a set of properties (*groups*), and the order is defined as a list of properties and an increasing or decreasing marker. If the aggregation function is not [commutative](#), then the order must be uniquely determined.

A uniquely determined order can be guaranteed if for example, the IDs of all objects for which grouping is performed are specified when the order is defined

In addition to the standard types of aggregate functions for grouping, there are three additional types: **EQUAL**, **AGGR** and **NAGGR**.

- **EQUAL** is a special case of the aggregation function **MAX** (or **MIN**), with the additional [constraint](#) that the value of the operand of the aggregating function within each group must be the same.
- **AGGR** and **NAGGR** are a special case of **EQUAL**, but with an even more strict constraint: for each group there is no more than one object collection, the operand of the aggregating function is one of the objects, and the groups include all other objects. Aggregate function **NAGGR** only differs from **AGGR** in the fact that if it is used, no constraint is created (it is assumed that the constraint follows from the semantics of the properties of the operands and / or groupings themselves).

Language

To declare a property that implements grouping, use the [GROUP operator](#).

Examples

```
1 CLASS Game;
2 CLASS Team;
3 hostGoals = DATA INTEGER (Game);
4 hostTeam = DATA Team (Game);
5 hostGoalsScored(team) = GROUP SUM hostGoals
6 (Game game) BY hostTeam(game);
7
8 name = DATA STRING[100] (Country);
9 countryName = GROUP AGGR Country country WHERE
10 country IS Country BY name(country); //
11 property (STRING[100]) -> Country is obtained
12
13 CLASS Book;
14 CLASS Tag;
15 name = DATA STRING[100] (Tag);
16 in = DATA BOOLEAN (Book, Tag);
17 tags(Book b) = GROUP CONCAT name(Tag t) IF in
18 (b, t), ', ' ORDER name(t), t;
```