

order by

Last Modified on 25/05/2020 1:45 pm AEST

The **order by** keyword sorts a list of records by one or more fields, or calculations.

Keyword

```
list order by value
```

```
list order by value desc
```

```
list order by value1 asc, value2 desc
```

Arguments

Argument	Data Type	Remarks
list	record list	
value	Any of single-value: <ul style="list-style-type: none">• string• number/decimal/currency• date/time/date-time• choice-field	Typically simply a field name.
result	record list	

Comments

A list calculation (such as requesting all records of a particular type, or following a to-many relationship) is provided to the left of the **order by** keywords. A sorting calculation, typically a field name, is then provided to the right. The sorting calculation must be a single-value calculation, not a list calculation.

For example, you can sort a list of Person records by their names, but not by their related tasks, because a person may have multiple tasks, so that would be considered as a list calculation.

Direction of sort

The **asc** or **desc** keywords may be added directly after the sorting calculation to indicate that the sorting should occur in ascending or descending order. The default is ascending, so the asc keyword is redundant, but can aid readability.

A choice field may also be provided as a sort calculation

Sorting on multiple fields

More than one sorting calculation may be provided, separated by commas. If two records have the same value for

the first sorting calculation, then sorting defers to the second calculations, and so on. For example, this can be used to sort Person records by Last Name and then First Name. If two or more people have the same Last name, then they will be then sorted by first name.

Using 'order by' within a function

If you need to sort a some list of records by multiple fields, and then provide the sorted list as an input to some other function, then it is necessary to enclose the whole list, including the ordering, within a separate brackets. This is necessary to prevent the commas of the sorting from becoming mixed up with the commas of the function.

For example, the following calculation will load Person records, sort them by last name, falling back to first name, and then return the first ten records. Note the second and second-last bracket that is required to avoid ambiguity caused by the comma.

```
first(10, (all(Person) order by [Last Name], [First Name]) )
```

Context

The right-hand sorting calculations are evaluated with respect to the records returned by the left-hand side. For example, if you refer to the [Name] field on the right-hand side, then it refers to the name field of the record list being sorted, not the name field of some other parent record that may be applicable to the calculation as a whole.

Availability

The **order by** keyword is not available in reports. Calculated fields that make use of **order by** are similarly not available in reports.

Examples

```
[Incidents] order by [Reported Date]
```

```
all([Person]) order by [Last Name], [First Name]
```

```
all([Person]) order by len([Email address]) desc
```