# Output configuration

# Dashboard block



## General considerations

The output settings for data or graphs will always use the columns aliases to identify the output fields.

## Table type

Three table shape can be choosen for output data. Some table types rely on some available attributes:

**Raw flat data:** Data are presented as the query output, record by record.

**Cross table:** Data are shown within a bidimensional array. The configuration will tell:

- The output column alias that represents the horizontal dimension (one choice possible only)
- The output column aliases that provide the vertical dimensions.
- The content of the value cell.

**Tree shaped data:** If the output data provide a hierarchical organization (noticed by id,parent), it may be practical to use a tree representation of the output.

## Output columns

This setting lists in order the list of output fields that will be shown in the output. It must use the aliased names of the columns and cannot be SQL expressions. The output list uses semi-column (;) as separator.

*E.g. if the query is:*

```
SELECT
   YEAR(FROM_UNIXTIME(time)) as year,
   count(*) as access
FROM
   mdl_log
GROUP BY
year
```

*Then the output column list could be:*

```
year;access
```

### Formatting output data

This setting accepts a data formatting expression on the “sprintf” syntax basis that will post-format the output values. You should provide as may formatters than output columns. The formatters will apply in order of the output description list, and will complete with “ignore format” formatters the missing descriptors. An empty formatter stands also for “ignore format” signal.

In the preceding sample, you would use:

```
%d;%s
```

to format first columns as an integer and the second column as a string.

Second exampe:

for, say, a popularity ratio, you may want to format the output as a float value with one single digit in the decimal part. You would use the following formatter for a result giving the ratio per month:

```
month;ratio
;%.1f
```

Note here that the first column (month) has no formatter applied.

Read more about formatting syntax

### Outputing HTML glue with referenced columns

Say you may want to show course names having links to browse quickly to the corresponding course. The formatting syntax allows using static HTML glue and reference another column in the output result to produce the cell content.

E.g the following syntax will show this formatting:

```
<a href="/course/view.php?id=%{cid}">%s</a>
```

References to other columns in the output result uses the %{fieldalias} syntax and will be processed befor the “sprintf” applies.

### Output fields labels

As technical fieldnames in DB may not have full sense to the end user of the dashboard, you may here rename the field names to more comprehensible names.

Again; use semi-columns (;) to separate labels, and give labels in same order of the output field list.

Example:

```
Category;Subcategory
```

### Paging size

If given, this will add a LIMIT,OFFSET clause to the query to paginate the output. Use pagination on queries that may potentially output a lot of rows.

### Cache results

when cache results is enabled, the query outputs will be cached to save time when accessing back to the same query output. The database will not have to process again the joins and the aggregators.

The TTL (Time To Live) delay of the cache can be adjusted.

The cron task of the dashboard can be used to program caches refresh and provide some fresh results.

### Clean the table display

If this option is used, subsequent rows repeating the same value than the row above will not show the value again. This often leads to a cleaner and more readable view of the data, specially for dimensional context columns that are usually displayed first.

There is visually a trap to clean values in the last (main payload) columns, as this might be misinterpreted as a lack of data (null) rather than undestanding the data has same value as above. You can fix this asking the cleanup filter no to operate further a given column number.

### Sortable table

If your query has not a structural hardcoded SORT clause in its body, but you may want the end user sort the data as he wants, you may declare the table sortable. the dashboard will add sorting controls and sort SQL clause for you.

### Subtotal separation column

Using summators usually provide a single final summed value of some output fields. In case you need having subtotals in some value output subdomains, you may tell which output field will be used to discriminate subtotals. the dashboard will add an additional subtotal row (for declared summators) summing all rows in a subtotal column single modality.

Note that subtotals can only be calculated if the global output result is sorted on the subtotal separation column.

## Credits

- Valéry Frémaux (valery@activeprolearn.com)- Main design and development
- Florence Labord (florence@activeprolearn.com) - documentation and testing

Return to the configuration guide index - Return to the component index - Generic query catalog - Return to the plugins index - Home

From:
https://docsen.activeprolearn.com/ - **Documentation Moodle ActiveProLearn**

Permanent link:
**https://docsen.activeprolearn.com/doku.php?id=blockdashboardconfigoutputfields&rev=1503329204**

Last update: **2026/01/13 08:58**