

Developer & Reporting > Choosing the Right Widget Display in Deskpro Dashboards < قاعدة المعلومات

Choosing the Right Widget Display in Deskpro Dashboards

Kim - 2025-10-15 - رويات - Kim - 2025-10-15

When building dashboards in Deskpro, one of the most important decisions is how to display your data. The type of widget you choose determines how information is interpreted and how quickly users can draw .insights

This guide explains when to use each widget type, how it visualizes data, and includes sample DPQL queries .with real examples

Simple Stat - Display a Single Key Metric

A **Simple Stat** widget displays one key value, such as a total, count, or percentage.

.Use this type when you want to emphasize a single KPI

```
SELECT DPQL_COUNT() AS 'stat_value',
'Tickets waiting' AS 'stat_description'
FROM tickets
('WHERE tickets.status IN ('awaiting_agent', 'awaiting_user
:You can customize the display further with
```

- .(% unit_left or unit_right add a currency or symbol (e.g. £ or $\, \bullet \,$
 - .default_value set a fallback value if the query returns none •

:Example

:Use this widget when

- Showing total open tickets •
- Displaying average response or resolution times $\, ullet \,$
 - Tracking SLA performance percentages •

Backlog

665

tickets waiting

Bar and Line Charts - Compare or Track Data Over Time

Bar charts and **line charts** are the versatile display types. They are ideal for comparing categories or showing trends across time periods

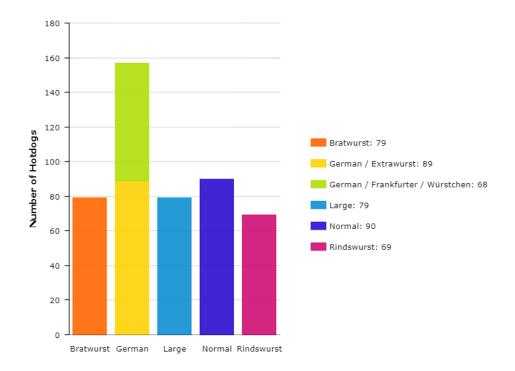
'GROUP BY DPQL_HIERARCHY(tickets.custom_data[24], 1, 3) AS 'Type

 ${\tt SELECT~DPQL_COUNT()~AS~'Number~of~Hotdogs'} \\ FROM~tickets\\ {\tt WHERE~tickets.organization} <> {\tt NULL~AND~tickets.custom_data[24]} <> {\tt NULL~tickets.custom_data[24]} <> {\tt NULL~tickets.custom_data[24]} <> {\tt NULL~tickets.custom_data[24]} <> {\tt NULL~tickets.custom_data[24]} <> {\tt$

This query groups ticket data using the $DPQL_HIERARCHY()$ function. "It aggregates nested custom field options, for example, grouping all sub-types under "German"

:Use this widget when

- Comparing ticket volume by department or agent
 - Tracking performance over time •
 - Displaying SLA compliance trends •



Layered Charts - Compare Multiple Datasets in One View

 $\label{layer} \mbox{Layered charts allow you to combine multiple result sets in a single visualization using the \mbox{LAYER WITH} \\ \mbox{keyword}.$

.This is useful when comparing related metrics, such as open and closed tickets per department

```
SELECT

DPQL_COUNT() AS 'Open Tickets',
tickets.organization.name AS 'Department',
'Tickets' AS 'value_axis_title'
FROM tickets
WHERE
tickets.organization.name <> NULL
AND tickets.status IN ('awaiting_user', 'awaiting_agent')
GROUP BY tickets.organization.name AS 'Department'
```

LAYER WITH

```
SELECT

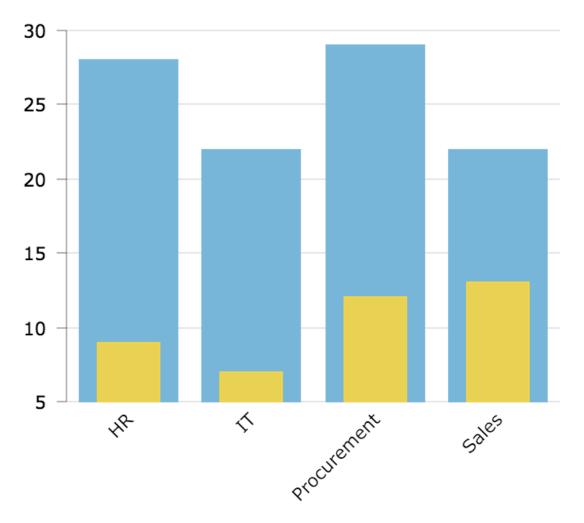
DPQL_COUNT() AS 'Closed Tickets',
tickets.organization.name AS 'Department'
FROM tickets
WHERE
tickets.organization.name <> NULL
AND tickets.status = 'resolved'
'GROUP BY tickets.organization.name AS 'Department
```

. You can also use LAYER WITH $\,$ LINE to combine chart types, for example, bars for totals and a line for targets

$: Use\ this\ widget\ when$

- Comparing two or more datasets with the same categories •
- Showing changes in open vs. closed or inbound vs. resolved counts •

 $. \textbf{Avoid using when:} \ \text{only a single metric is being displayed, or when datasets differ significantly in scale} \\$



Department

Pie Charts - Show Proportions of a Whole

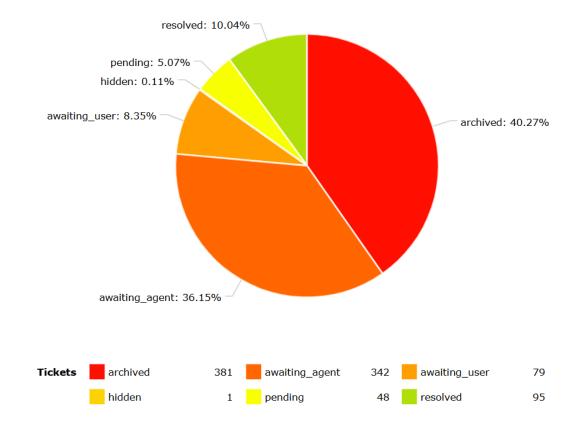
A **Pie** chart displays how each part contributes to a total. .It's most effective when you have a limited number of categories to compare

SELECT DPQL_COUNT() AS 'Tickets', tickets.status AS 'Status'
FROM tickets
'GROUP BY tickets.status AS 'Status

:Use this widget when

- Showing ticket distribution by status or channel •
- Comparing the share of resolved vs. pending tickets •

.Avoid using when: you have more than 5-6 segments or when categories have similar values



Gauge Widgets - Measure Performance Against a Target

A **Gauge** widget is useful when you need to visualize progress toward a specific target, such as SLA compliance .or total revenue

```
SELECT SUM(ticket_charges.amount) AS 'stat_value',

1500000 AS 'stat_total',

DPQL_CONCAT('f', DPQL_FORMAT(SUM(ticket_charges.amount), 'number', 0),

' of f1.5M budget') AS 'tooltip_text'

FROM ticket_charges

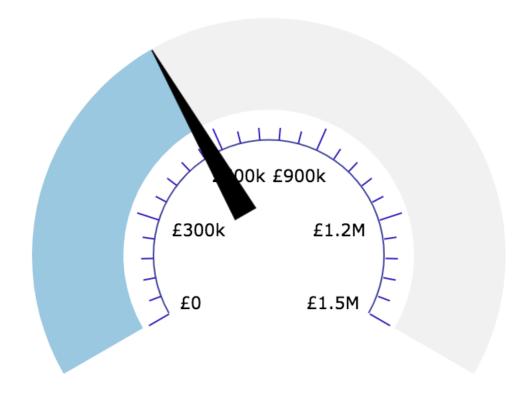
%WHERE ticket_charges.ticket.date_created = %THIS_YEAR
```

- .stat_value determines the current position of the needle or fill $\, ullet \,$
 - .stat_total defines the maximum range $\ \bullet$
 - .tooltip_text adds context for the displayed value •

:Use this widget when

- Comparing progress against a set goal or budget •
- Tracking utilization rates or performance thresholds •

.Avoid using when: you need to compare multiple categories or historical data



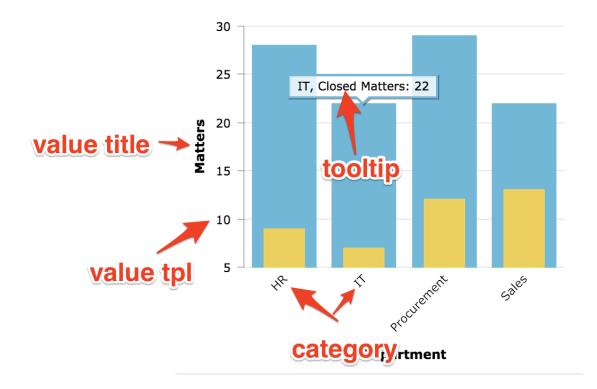
Tooltips and Templates - Enhance Readability

 $. Tooltips \ and \ templates \ provide \ a \ way \ to \ customize \ axis \ labels, \ tooltips, \ and \ values \ for \ clearer \ visualization$

- .formatCurrency displays numeric values as currency
 - .formatNumber adds digit grouping and decimals •
 - .formatPercent converts fractions to percentages •
 - .math performs simple operations within the template $\, \bullet \,$

:When layering datasets

- {{First layer uses {{value}}}, {{category •
- $\{\{Second\ uses\ \{\{0_value\}\},\ \{\{0_category\ \bullet$
 - .Third uses {{1_value}}, etc ●



Advanced Example - Four-Layer Template Query

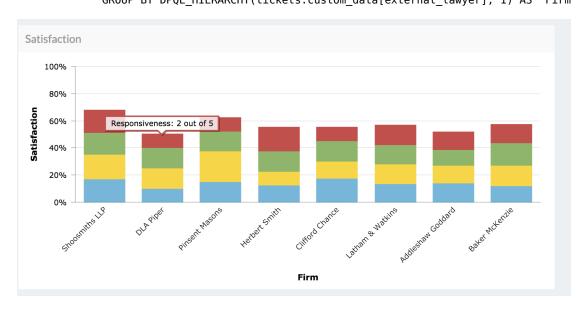
The following query shows how to create a chart that layers four metrics, each with its own tooltip template.

Each layer calculates the average rating for a different metric, such as responsiveness or friendliness

```
AVG(tickets.custom data[rate responsiveness])*5 AS 'Responsiveness',
'Responsiveness: {{formatNumber (math value "/" 5) minimumFractionDigits=0
            maximumFractionDigits=1}} out of 5' as 'tooltip_text_template'
                                                               FROM tickets
                                                                      WHERE
                    tickets.custom_data[rate_responsiveness].value <> NULL
                          AND tickets.custom data[external lawyer] <> NULL
GROUP BY DPQL_HIERARCHY(tickets.custom_data[external_lawyer], 1) AS 'Firm'
                                                                 LAYER WITH
                                                                     SELECT
        AVG(tickets.custom data[rate commerciality])*5 AS 'Commerciality',
'Commerciality: {{formatNumber (math 0_value "/" 5) minimumFractionDigits=0
            maximumFractionDigits=1}} out of 5' as 'tooltip_text_template'
                                                               FROM tickets
                                                                      WHERE
                      tickets.custom data[rate commerciality].value <> NULL
                          AND tickets.custom_data[external_lawyer] <> NULL
GROUP BY DPQL_HIERARCHY(tickets.custom_data[external_lawyer], 1) AS 'Firm'
                                                                 LAYER WITH
                                                                     SELECT
                        AVG(tickets.custom_data[rate_value])*5 AS 'Value',
       'Value: {{formatNumber (math 1_value "/" 5) minimumFractionDigits=0
            maximumFractionDigits=1}} out of 5' as 'tooltip text template'
                                                               FROM tickets
                             tickets.custom_data[rate_value].value <> NULL
                          AND tickets.custom_data[external_lawyer] <> NULL
GROUP BY DPQL HIERARCHY(tickets.custom data[external lawyer], 1) AS 'Firm'
```

LAYER WITH

SELECT



Summary

:Each widget type in Deskpro serves a distinct purpose

Key DPQL Function	Best Widget	Goal
<pre>DPQL_COUNT() AS 'stat_value' ''Tickets this month' AS 'stat_description</pre>	' Simple Stat	Display a single KPI
GROUP BY, LAYER WITH	Bar / Line	Compare categories or time periods
GROUP BY	Pie	Show proportions
stat_total,DPQL_FORMAT	Gauge	Track progress against targets
GROUP BY with multiple fields	Table	Display detailed data