# Text query

## Documentation

## Summary

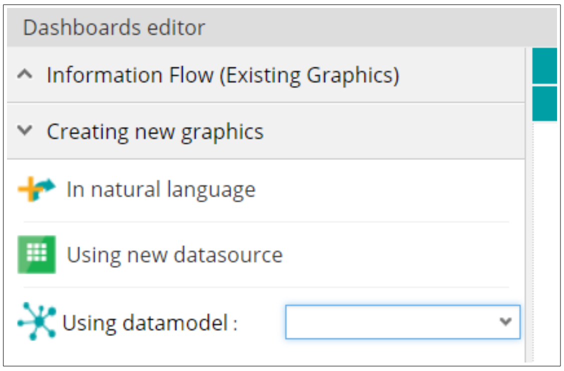
# I. PREREQUISITES

- Cubes in your wallets recently synchronized

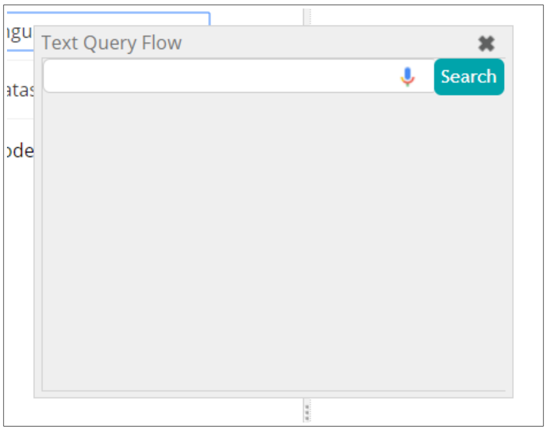
# II. CREATING FLOWS WITH TEXT QUERY

Digdash offers the possibility to create flows in some clicks with natural language.

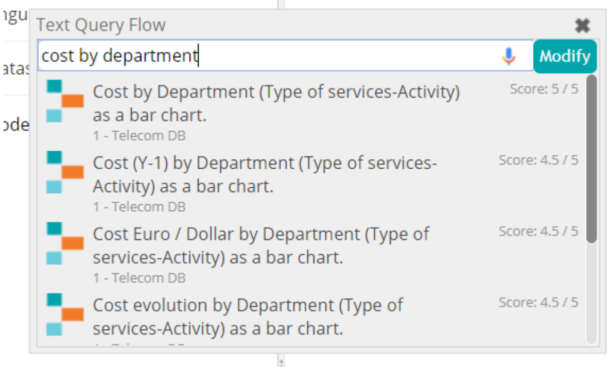
- Via the dashboards editor



**Dashboards editor** > **Creating new graphics** > **In natural language**



In the showing dialog, you can enter you query in the search bar.



cost by department

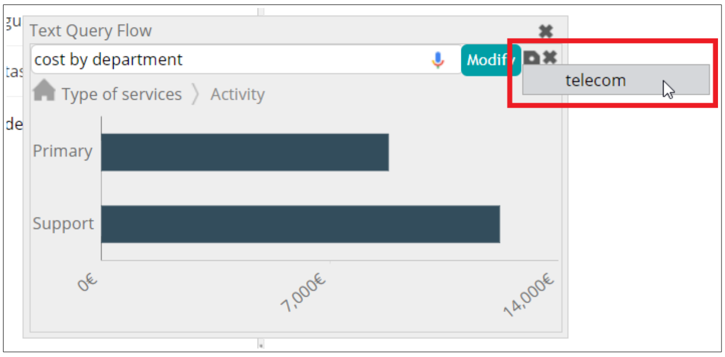
Cost by Department (Type of services-Activity) as a bar chart.
1 - Telecom DB
Score: 5 / 5

Cost (Y-1) by Department (Type of services-Activity) as a bar chart.
1 - Telecom DB
Score: 4.5 / 5

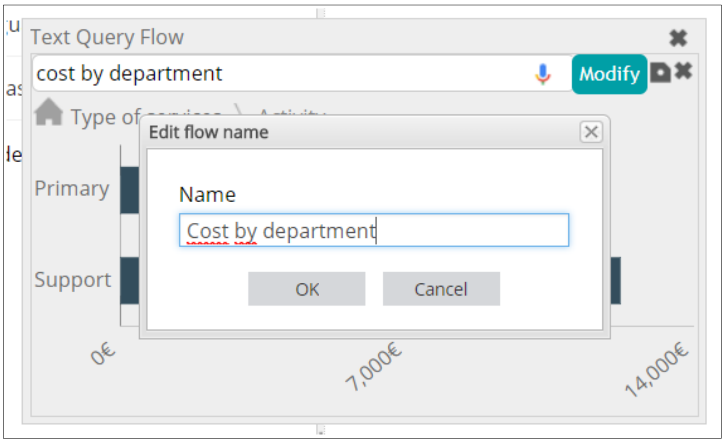
Cost Euro / Dollar by Department (Type of services-Activity) as a bar chart.
1 - Telecom DB
Score: 4.5 / 5

Cost evolution by Department (Type of services-Activity) as a bar chart.
Score: 4.5 / 5

Choose a flow in the results list (we are going to pick the first one)

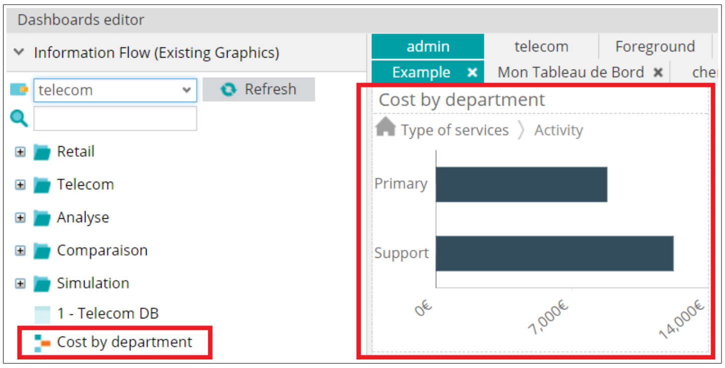


And save the flow in the current wallet.



Name: Cost by department

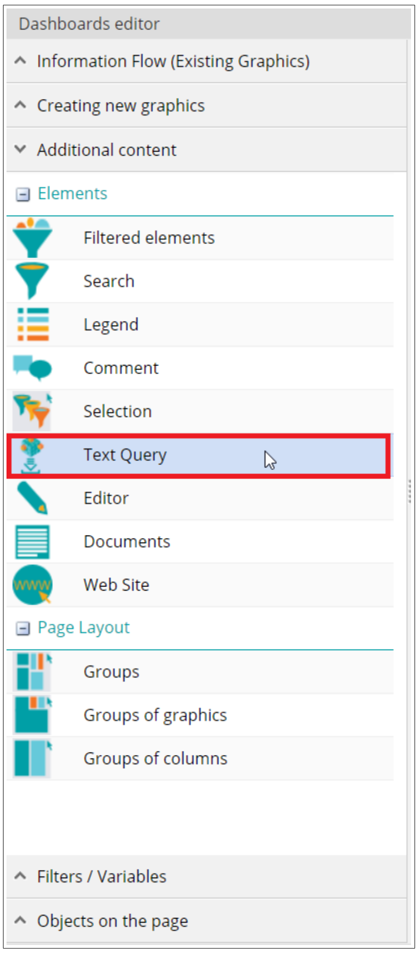
Give a name to your flow.



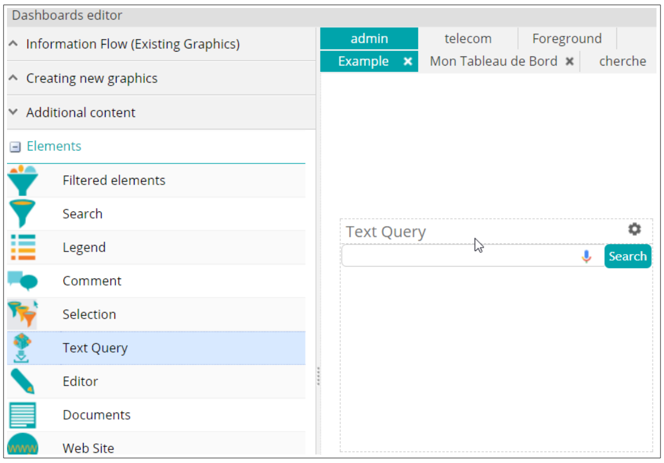
The new flow will be added to your dashboard editor and you will find it in the existing graphics list.

- Via the dashboard

It is also possible to add a « Text query » element in your dashboard in order to create flows on the fly.



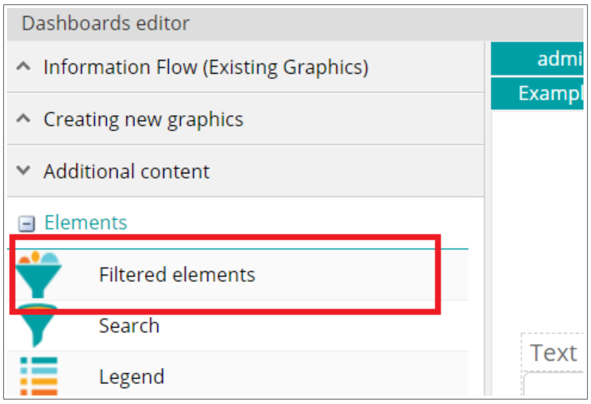
**Dashboards editor** > **Additional content** > **Elements** > **Text query**



Add the element in your dashboard editor.

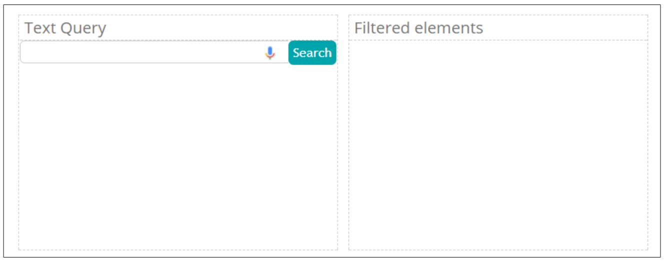
According to your query, you may filter on some measures/dimensions.

You would better have a « Filtered elements » element in your dashboard to remove these filters for the potential



next queries.

**Dashboard editors** > **Additional content** > **Elements** > **Filtered elements**



You will find those elements dashboard side.

# III. COMPOSITION OF A TEXT QUERY

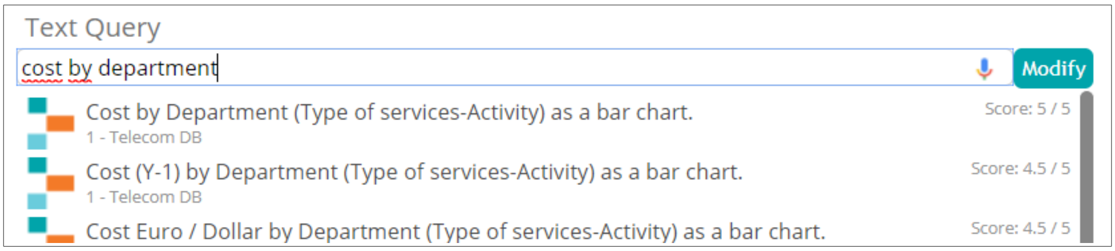
Digdash allows you to create flows from data models in natural language.

Digdash mainly rests on keywords to suggest you the most relevant flows and the names of your data models' columns.

So a query is basically composed of measures names from your data model and/or dimensions names, followed (or not) by flows names and/or sorting operations.

We shall later explain how important query terms are important.

# IV. RESULTS FOR A TEXT QUERY



cost by department

Cost by Department (Type of services-Activity) as a bar chart.
1 - Telecom DB
Score: 5 / 5

Cost (Y-1) by Department (Type of services-Activity) as a bar chart.
1 - Telecom DB
Score: 4.5 / 5

Cost Euro / Dollar by Department (Type of services-Activity) as a bar chart.
Score: 4.5 / 5

*Screenshot: Results for a text query*

Results for a text query are presented in list, with its associated flow, sorted by its relevance and a score. The more the score is high, the more the suggested flow is considered as relevant. Also, the cube name is mentioned so is a description of the flow.

We shall later see in this document which are the criteria that could have an influence on the score of a result.

# V. CHOICES OF GRAPHICS

For a given query, you will be suggested a list of results of multiple graphics, depending on your query members.

Nevertheless, you can require a particular graphic, as long as it remains coherent (the query « Cost in line » has no sense).

The numbers of members for a cube has an incidence. Indeed, when a query, even coherent, has too many members to show, you will be suggested a more adapted graphic.

Here are the keywords to use for the choice of graphics:

| Charts name | Keywords |
|---|---|
| Pie chart | « pie » |
| Gauge | « gauge » |
| Progress bar | « progress bar » |
| Energy bars | « energy » |
| Arrow indicator | « arrow » |
| Column chart | « column »<br>« histogram » |
| Bar chart | « bar » |
| Map chart | « map » |
| Scatter chart | « scatter » |
| Bubble chart | « bubble » |
| Line chart | « line » |
| Zone chart | « area »<br>« zone » |
| Radar chart | « radar » |
| Tab chart | « table » |
| Indicator | « indicator » |
| Cross table | « cross table » |
| OLAP table | « OLAP table »<br>« OLAP» |
| Text chart | « text » |

*Table presenting keywords for the different flows*

# VI. SORTING

It is possible for you to use sorting operations with the following keywords :

| Sorting | Keywords | Queries examples |
|---|---|---|
| Ascending | « sort » (ascending)<br>« sorted » (ascending)<br>« ascending » | « Cost by region in France in 2006 sorted by cost in table »<br>« Cost by region in France in 2006 sorted in table »<br>« Cost by region in France in 2006 ascending in table »<br>« Cost by region in France in 2006 sorted in ascending order in table » |
| Descending | « sort » (descending)<br>« sorted » (descending)<br>« descending » | « Cost by region in France in 2006 sorted by cost in table »<br>« Cost by region in France in 2006 descending in table »<br>« Cost by region in France in 2006 sorted in descending order in table » |

*Table presenting keywords for sorting operations*

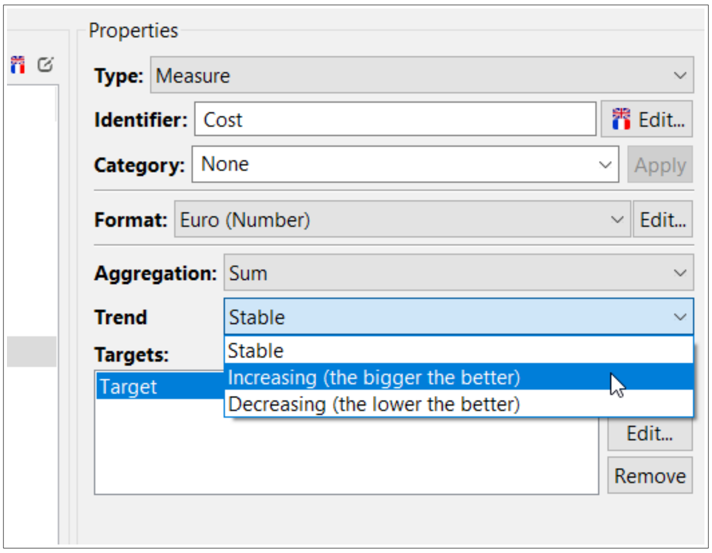
# VII. TREND OF A MEASURE

A measure has a trend. It can be stable (trend by default), increasing or decreasing:

| Tendance | Signification |
|---|---|
| STABLE | Default trend: The bigger the better |
| INCREASING | The bigger the better<br>*Example : for a quality* |
| DECREASING | The lowest the better<br>*Example : for a cost* |

*Table presenting possible trends for a measure*

You can change the trend of a measure via the Digdash Enterprise Studio, in the data source advanced configuration, in the properties part of a measure.



Properties

Type: Measure

Identifier: Cost

Category: None

Format: Euro (Number)

Aggregation: Sum

Trend: Stable

Targets: Target

Edit...
Apply
Edit...
Stable
Increasing (the bigger the better)
Decreasing (the lower the better)
Edit...
Remove

*Definition of a measure's trend in Digdash's Studio Enterprise*

- Impact of a measure’s trend

Trend has an impact on the sorting order. Indeed, if the type of sorting is not explicitly mentioned, the kind of sorting will be based on the measure’s trend. Consequently, we will get an ascending sort for a stable or increasing trend and a descending sort for a decreasing trend.

| Trend | Queries examples | Sort |
|---|---|---|
| STABLE | « Duration by state sorted »<br>Duration is a measure with a stable trend | *The sorting of measure Duration on dimension state will be descendant* |
| INCREASING | « Quality by state sorted »<br>Quality is a measure with an increasing trend | *The sorting of measure Quality on dimension state will be descendant* |
| DECREASING | « Cost by state sorted »<br>Cost is a measure with a decreasing trend | *The sorting of measure Cost on dimension state will be ascendant* |

*Table with examples presenting impact of a measure’s trend on sorting orders*

# VIII. THE BEST / THE WORST, THE TOP / BOTTOM

It is possible to get the X best/worst members of the results of your query using these keywords:

| Cases | Keywords | Example |
|---|---|---|
| The X best | « Top »<br>« biggest »<br>« best » | The best cost in France<br>The 5 best costs in 2016<br>The 2 biggest costs in Europe<br>Top 3 of costs in France in 2016 |
| The X worst | « Worst »<br>« bottom »<br>« smallest » | The worst cost in France<br>The 5 worst costs in 2016<br>The 2 smallest costs in Europe |

*Table presenting keywords for best/worst, top/bottom*

# IX. AGGREGATION

You can define an aggregation method for the measures of your query using these keywords :

| Aggregation | Keywords | Queries examples |
|---|---|---|
| Sum | « sum » | « Sum of cost » |
| Average | « average » | « Average of cost » |
| Minimum | « min » | « Min of cost  » |
| Maximum | « max » | « Max of cost  » |

*Table presenting keywords for aggregation*

# X. TARGETS

It is possible to apply targets on measures mentioning the following keyword :

| Keyword |
|---|
| « target » |

*Table presenting the keyword for targets*

You can also directly mention the name of the targets you want to apply.

| Example |
|---|
| Given a data model with the following columns |

| Dimensions | Measures |
|---|---|
| Date | Quality |
| Type of line | Cost (with target « Targ ») |

Example 1 : « Cost in gauge with target »

➔ Every measure has one target applied on it.

Example 2 : « Cost in gauge avec Obj »

➔ The target « Targ » is applied on the associated measure « Cost ».

*Table with example presenting the use of target in text query*

# XI. USE OF SYNONYMS

Text query takes in charge synonyms of your query members.

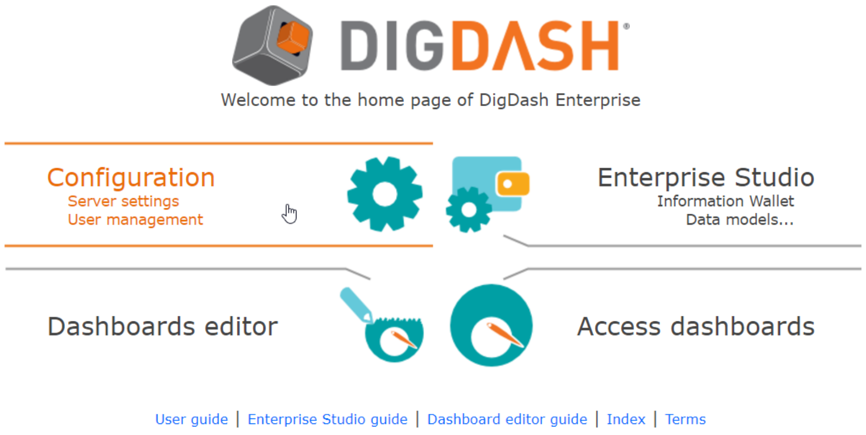
## XI.1 Creating synonyms dictionaries

To use synonyms in text query, you first need to import a synonyms dictionary in Digdash.

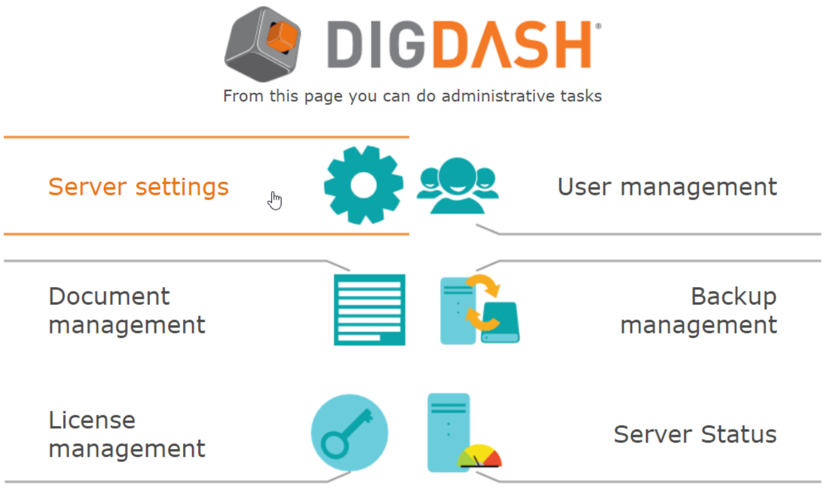
Please refer to the documentation called « synonyms_dictionary_en.pdf » to import a synonyms dictionary in Digdash.

## XI.2 Activating synonyms dictionaries

You then need to check if the use of synonyms dictionaries is activated for text query in the server configuration.

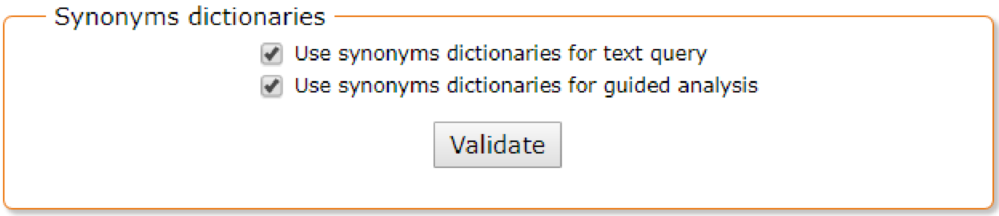


Welcome to the home page of DigDash Enterprise



From this page you can do administrative tasks

In the server configuration page, at the bottom of the page > **Advanced >>** > Category **Synonyms dictionaries** > Checkbox **Use synonyms dictionaries for text query**



Synonyms dictionaries

☑ Use synonyms dictionaries for text query

☑ Use synonyms dictionaries for guided analysis

*Screenshot: Activation of the use of synonyms dictionaries for text query*

## XI.3 Use case

| Example |
|---|
| Given a data model with the following columns |

| Dimensions | Measures |
|---|---|
| Date | Quality |
| Type of line | Cost |

In that case, the query

« Price by sort of line »

is equivalent to th query

« Cost by type of line »

(« price » is a synonym of « cost » and « sort » is a synonym of « type »).

**NB** : This is valid only if your synonyms dictionaries contains these synonyms.

*Table with example presenting a use case of text query with synonyms*

# XII. SCORES FOR A RESULT OF A TEXT QUERY

Queries results are ordered according to their relevance and the best score can reach a 5 out of 5.

The score for a result may vary for different reasons.

- Importance of the query members

Indeed, a result will be better marked if the query is composed of the exact columns' names of your data models.

A query is then considered as less good if it contains partial names of your columns, or synonyms.

**Example**

Given a data model with these columns

| Dimensions | Measures |
|---|---|
| Date | Quality |
| Type of line | Cost |

| Query 1 :<br>With the exact names | Query 2 :<br>With partial names | Query 3 :<br>With synonyms |
|---|---|---|
| « Cost by type of line » | « Cost by type » | « Price by sort of line » |
| With only exact terms, the results can be well-marked. | With partial names, the results can be less well-marked. | « price » is a synonym of « cost », « sort of line » is a synonym of « type of line ».<br>This query is different from the original query, the score will be low. |
| Score* : 5/5 | Score* : 3/5 | Score* : 2/5 |

* indicated scoes are just to illustrate our example

*Table with example presenting how important query members are important*

- Importance of the suggested type of graphics

The results list for a query suggests graphics more or less relevant for what is expected. Given the nature of the query members, some flows will be considered less relevant, hence a lower score for them.