

Содержание

- Jinja-шаблоны 2
- Синтаксис** 2
- Jinja-макросы 2
- Логические операторы 2
- Использование Jinja-шаблонов для фильтрации в дашборде** 2

Jinja-шаблоны

Jinja Templating — это движок веб-шаблонов для Python. Он использует текстовый язык шаблонов, который можно использовать для генерации как разметки, так и исходного кода. Включение Jinja повышает гибкость функций и обеспечивает множество вариантов использования, таких как:

- Реализуйте контроль доступа на основе данных пользователя, вошедшего в систему в данный момент (`user_id` и/или `username`);
- Применяйте фильтры информационной панели непосредственно к внутреннему запросу набора данных;
- Используйте компонент фильтра для фильтрации запроса, когда имя столбца фильтра не соответствует имени в текущем запросе;
- Примените ограничения фильтра к панелям мониторинга через URL-адрес;
- Динамически изменяйте определенные элементы вашего SQL-запроса (например, вычисления, агрегации и т. д.) на основе условий фильтра панели мониторинга;
- Персонализированные информационные панели.

В Superset вы можете использовать Jinja в Лаборатории SQL, виртуальных датасетах и Фiltro на уровне строк (Row Level Security):

- Добавьте в запрос предварительно определенные макросы, чтобы возвращать динамические данные;
- Выполнять логические операторы (такие как `if`, `for` и т. д.).

Синтаксис

Jinja-макросы

Логические операторы

Использование Jinja-шаблонов для фильтрации в дашборде

Этот процесс состоит из 4 шагов:

1. Создайте виртуальный набор данных.
2. Измените запрос виртуального набора данных, включив в него структуру шаблонов Jinja.
3. Создайте диаграмму из виртуального набора данных и добавьте ее на панель мониторинга.
4. Настройте фильтры информационной панели.

Шаг 1. Создайте виртуальный набор данных

```
WITH calculation as (  
    SELECT count(*),  
    window
```

Рассмотрим следующий запрос: Если вы создадите виртуальный набор данных с помощью этого запроса, он будет иметь только два столбца: *count* и *window*. Как следствие, вы не сможете создать фильтр информационной панели для *type_call_name*. Вот почему мы будем использовать Jinja.

```
FROM
abituser.ld$telephony
WHERE
type_call_name in
('входящий', 'исходящий')
GROUP BY window
)
SELECT * FROM
calculation
```

Однако если вы выполните запрос, включающий шаблон Jinja, непосредственно в лаборатории SQL, он не вернет никаких результатов, поскольку у нас нет фильтров для передачи значений, а это означает, что выполненный запрос в конечном итоге будет иметь вид:

```
WITH calculation as (
SELECT count(*),
window
FROM
abituser.ld$telephony
WHERE
type_call_name in ('')
GROUP BY window
)
SELECT * FROM
calculation
```

Для начала создадим виртуальный набор данных без Jinja через Редактор SQL:

Сохраним запрос и используем его, как датасет для графика.

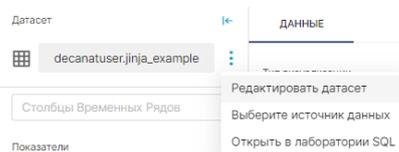
```
WITH calculation as (
SELECT count(*),
window
FROM
abituser.ld$telephony
WHERE
type_call_name in
('входящий')
GROUP BY window
)
SELECT * FROM
calculation
```

Шаг 2. Измените виртуальный набор данных, включив в него Jinja.

В вашем браузере откроется новая вкладка с вашим новым виртуальным набором данных.

В разделе **Датасет** рядом с именем вашего набора данных, выберите **Редактировать датасет**.

В разделе **Источник**, нажмите на значок замка, который предоставит вам доступ для редактирования запроса. На панели ввода **SQL**-запроса



```
замените type_call_name in
('входящий') на type_call_name
in ({{ "" + "",
"".join(filter_values('type_call_na
me')) + "" }})
```

и нажмите **Сохранить**.

О структуре шаблонов Jinja:



- Начальная (({{ "" + "", "".join() и конечная () + "" }}) части отвечают за объединение – эта структура позволяет фильтровать несколько значений.
- `filter_values('Column-Name')` – функция, которая будет учитывать значение фильтра `Column-Name`, в данном случае это `type_call_name`.

Шаг 3. Добавьте диаграмму на панель дашборда.

Теперь, когда мы правильно изменили виртуальный набор данных, мы можем запустить нашу диаграмму, выбрав **Update chart**. Диаграмма не вернет никаких результатов, поскольку она еще не настроена. Выберите **Сохранить** график и перейти к дашборду.

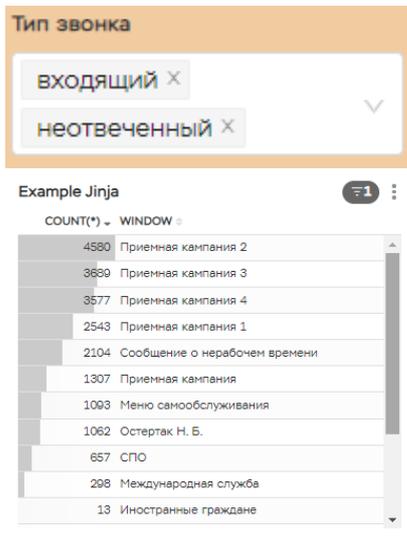
Шаг 4. Настройте фильтр на панели дашборда.

Панель управления откроется в вашем браузере. После его появления приступим к заключительному этапу настройки фильтров панели мониторинга. Начните с выбора стрелки вправо, чтобы развернуть область **Фильтры**, а затем выберите **+Add/Edit filters**.

Появится окно Добавление и редактирование фильтров.

1. В поле **Значение фильтра**

- введите имя вашего фильтра.
2. В поле **Датасет** укажите датасет, который используется в самом запросе, а не тот, который создали с помощью шаблонов. В нашем примере это *abituser.Id\$telephony*.
 3. В поле **Column** мы выберем *type_call_name*, поскольку именно оно определяет имя нашего столбца в структуре Jinja.
 4. Затем выберите вкладку **Scoping**, чтобы убедиться, что фильтр сопоставлен с вашим графиком Фильтр.
 5. По завершении выберите **Сохранить**.



Теперь, слева появится созданный фильтр, при выборе значений к области диаграммы применятся выбранные позиции.

From:

<https://kb.nstu.ru/> - База знаний НГТУ НЭТИ

Permanent link:

https://kb.nstu.ru/superset:work_in_superset:jinja_templates

Last update: 2023/09/22 12:34

