



ClickHouse для инженеров и архитекторов БД



Проверить, идет ли запись

**Меня хорошо видно
&& слышно?**



Тема вебинара

Функции для работы с типами данных, агрегатные функции и UDF



Александра Гроховская

Senior Data Analyst / Team Lead
Ph.D.

Преподаватель курса **ClickHouse для инженеров и архитекторов БД** в OTUS

[LinkedIn](#)

Правила вебинара



Активно
участвуем



Задаем вопрос
в чат или голосом



Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения



Индивидуально



Время, необходимое
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или
задайте вопрос

Маршрут вебинара

Regular functions

Практика

Aggregate functions

Практика

User Defined Functions

Практика

Цели вебинара

1. Изучить функции для работы с типами данных и агрегатные функции в ClickHouse
2. Познакомиться с пользовательскими функциями в ClickHouse

Введение

Существует как минимум два типа функций - обычные функции (**регулярные** функции, функции) и **агрегатные** функции.

- **Регулярные функции** работают так, как будто они применяются к каждой строке отдельно (для каждой строки результат функции не зависит от других строк).
- **Агрегатные функции** накапливают набор значений из разных строк (то есть они зависят от всего набора строк).

Regular functions

Особенности

- **Сильная типизация**

Нет неявных преобразований между типами. Каждая функция работает с определенным набором типов.

- **Устранение общих подвыражений**

Все выражения в запросе, которые имеют одинаковый AST (одинаковая запись или одинаковый результат синтаксического разбора), считаются имеющими одинаковые значения. Такие выражения конкатенируются и выполняются один раз.

- **Типы результатов**

Все функции возвращают в качестве результата одно значение (не несколько значений и не нулевые значения). Тип результата обычно определяется только типами аргументов, а не их значениями.

Особенности

- **Константы**

Для простоты некоторые функции могут работать только с константами для некоторых аргументов. При этом постоянство важно только в пределах одного запроса.

- **Обработка NULL**

По умолчанию если хотя бы один из аргументов функции является NULL, результат функции также является NULL. Особенности поведения можем указывать самостоятельно.

- **Постоянство**

Функции не могут изменять значения своих аргументов. Порядок вычисления функций в запросе не имеет значения.

Особенности

- **Функции высшего порядка, оператор `->` и функция `lambda(params, expr)`**

Функции высшего порядка (те, которые принимают другие функции в качестве аргументов или возвращают функции как результат) могут принимать в качестве функционального аргумента только лямбда-функции. Чтобы передать лямбда-функцию в функцию высшего порядка, используйте оператор `->`.

Регулярные функции

- По сути, помогают решать ежедневные задачи, раскрывая мощный SQL диалект ClickHouse
- Есть как базовые, так и более продвинутые и сложные функции.
- Частота использования зависит от знаний и навыков, а также решаемой задачи

Какие бывают?

Arithmetic

Arrays

arrayJoin

UDF

Bit

Bitmap

Comparison

Conditional

Dates and Times

Dictionaries

Distance

Embedded Dictionaries

Geo

>

Encoding

Encryption

Files

Hash

IN Operator

IP Addresses

Introspection

JSON

Logical

Machine Learning

Maps

Mathematical

NLP (experimental)

Nullable

Other

Random Numbers

Replacing in Strings

Rounding

Searching in Strings

Splitting Strings

Strings

Time Series

Time Window

Tuples

Type Conversion

ULID

URLs

UUIDs

uniqTheta

Какие бывают?

Arithmetic

Arrays ♥

arrayJoin ♥

UDF

Bit

Bitmap

Comparison ♥

Conditional ♥

Dates and Times ♥

Dictionaries ♥

Distance

Embedded Dictionaries

Geo

>

Encoding

Encryption ♥

Files

Hash

IN Operator ♥

IP Addresses

Introspection

JSON ♥

Logical

Machine Learning

Maps ♥

Mathematical

NLP (experimental)

Nullable ♥

Other ♥

Random Numbers

Replacing in Strings

Rounding ♥

Searching in Strings ♥

Splitting Strings ♥

Strings ♥

Time Series

Time Window

Tuples

Type Conversion ♥

ULID

URLs

UUIDs ♥

uniqTheta

Пришлите в чат ответ, который будет получен при исполнении запроса

```
select sum(score) as ans
from table1
where date = toDate('2020-02-01') - interval 1 month
      and name ilike('%a%')
```

№	id	name	date	score
1	0	Max	2020-01-01	50
2	1	Dan	2020-01-01	12
3	2	Alex	2020-01-01	95
4	3	Alex	2020-01-02	72
5	4	Max	2020-01-02	66
6	5	Dan	2020-01-02	69
7	6	Dan	2020-01-03	18
8	7	Alex	2020-01-03	60
9	8	Max	2020-01-03	86

Вопросы?



Ставим “+”,
если вопросы есть

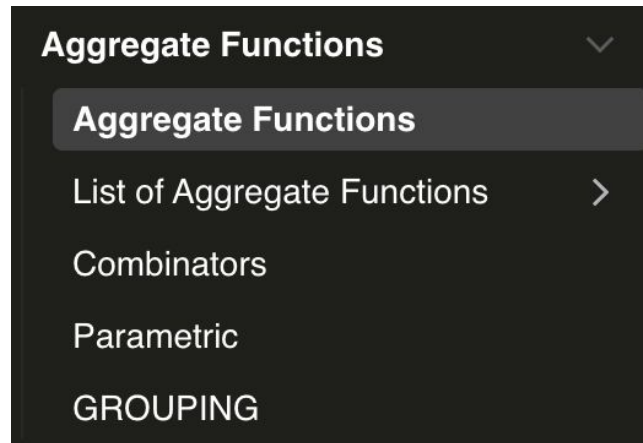


Ставим “-”,
если вопросов нет

Aggregate functions

Агрегатные функции

Функции, которые получают результирующее значение путем вычисления на множестве значений, называются агрегатными функциями. В ClickHouse агрегатные функции работают обычным образом, как и ожидает специалист по базам данных.



Какие бывают?

Standard aggregate functions:	ClickHouse-specific aggregate functions:			
count	anyHeavy	groupBitAnd	kurtSamp	quantileTiming
min	anyLast	groupBitOr	kurtPop	quantileTimingWeighted
max	argMin	groupBitXor	uniq	quantileDeterministic
sum	argMax	groupBitmap	uniqExact	quantileTDigest
avg	avgWeighted	groupBitmapAnd	uniqCombined	quantileTDigestWeighted
any	topK	groupBitmapOr	uniqCombined64	quantileBFloat16
stddevPop	topKWeighted	groupBitmapXor	uniqHLL12	quantileBFloat16Weighted
stddevSamp	groupArray	sumWithOverflow	quantile	simpleLinearRegression
varPop	groupUniqArray	sumMap	quantiles	stochasticLinearRegression
varSamp	groupArrayInsertAt	minMap	quantileExact	stochasticLogisticRegression
covarPop	groupArrayMovingAvg	maxMap	quantileExactLow	categoricalInformationValue
covarSamp	groupArrayMovingSum	skewSamp	quantileExactHigh	
		skewPop	quantileExactWeighted	

Параметрические агрегатные функции

- Некоторые агрегатные функции могут принимать не только столбцы аргументов (используются для сжатия), но и набор параметров - констант для инициализации.
- Синтаксис - две пары скобок вместо одной. Первая - для параметров, вторая - для аргументов.

histogram

```
sequenceMatch(pattern)(timestamp, cond1,  
cond2, ...)
```

```
sequenceCount(pattern)(time, cond1, cond2, ...)
```

```
windowFunnel
```

```
retention
```

```
uniqUpTo(N)(x)
```

```
sumMapFiltered
```

```
sumMapFilteredWithOverflow
```

```
sequenceNextNode
```

Комбинаторы агрегатных функций

- К имени агрегатной функции может быть добавлен суффикс. Это изменяет принцип работы агрегатной функции.

-If
-Array
-Map
-SimpleState
-State
-Merge
-MergeState
-ForEach
-Distinct
-OrDefault
-OrNull
-Resample
-ArgMin
-ArgMax

Grouping

ROLLUP и CUBE являются модификаторами GROUP BY - вычисляют промежуточные итоги.

- ROLLUP берет упорядоченный список столбцов, например (день, месяц, год), и вычисляет промежуточные итоги на каждом уровне агрегирования, а затем общий итог.
- CUBE вычисляет промежуточные итоги по всем возможным комбинациям указанных столбцов.

Пришлите в чат ответ, который будет получен при исполнении запроса

```
select groupUniqArray(winner_of_day) as winners
from (
  select argMax(name, score) as winner_of_day
  from table1
  group by date) as t1
```

№	id	name	date	score
1	0	Max	2020-01-01	50
2	1	Dan	2020-01-01	12
3	2	Alex	2020-01-01	95
4	3	Alex	2020-01-02	72
5	4	Max	2020-01-02	66
6	5	Dan	2020-01-02	69
7	6	Dan	2020-01-03	18
8	7	Alex	2020-01-03	60
9	8	Max	2020-01-03	86

Вопросы?



Ставим “+”,
если вопросы есть



Ставим “-”,
если вопросов нет

UDF

Executable User Defined Functions

Исполняемые функции, определяемые пользователем

- ClickHouse может вызывать любую внешнюю исполняемую программу или скрипт для обработки данных.
- Конфигурация исполняемых функций, определяемых пользователем, может быть расположена в одном или нескольких xml-файлах. Путь к конфигурации указывается в параметре **user_defined_executable_functions_config**.

Команда должна считывать аргументы из **STDIN** и выводить результат в **STDOUT**. Команда должна обрабатывать аргументы итеративно. То есть после обработки фрагмента аргументов она должна дожидаться следующего фрагмента.

Пример

```
<functions>
  <function>
    <type>executable</type>
    <name>test_function_python</name>
    <return_type>String</return_type>
    <argument>
      <type>UInt64</type>
      <name>value</name>
    </argument>
    <format>TabSeparated</format>
    <command>test_function.py</command>
  </function>
</functions>
```

```
#!/usr/bin/python3

import sys

if __name__ == '__main__':
    for line in sys.stdin:
        print("Value " + line, end='')
        sys.stdout.flush()
```

SQL User Defined Functions

- Создает функцию, определяемую пользователем (UDF), из лямбда-выражения. Выражение должно состоять из параметров функции, констант, операторов или других вызовов функций.
- Функция может иметь произвольное количество параметров.

Существует несколько ограничений:

- Имя функции должно быть уникальным среди пользовательских и системных функций.
- Рекурсивные функции не допускаются.
- Все переменные, используемые функцией, должны быть указаны в ее списке параметров.

Пришлите в чат ответ, который будет получен при исполнении запроса

```
CREATE FUNCTION parity_str AS (n) -> if(n % 2, 'odd', 'even');  
  
SELECT 11 as number, parity_str(number);
```

Слайд с домашним заданием

1. Закрепить навыки использования регулярных, агрегирующих и пользовательских функций

2.

3.

4.



Сроки выполнения:

Рефлексия

Рефлексия



Какие отличия в работе регулярных и агрегирующих функций в ClickHouse от СУДБ из вашего текущего стека вы заметили?



Полезны ли эти отличия в вашей практике?

**Заполните, пожалуйста,
опрос о занятии
по ссылке в чате**

Спасибо за внимание!

Приходите на следующие вебинары



Александра Гроховская

Senior Data Analyst / Team Lead
Ph.D.

Преподаватель курса ***ClickHouse для инженеров и архитекторов БД*** в OTUS

[LinkedIn](#)