

Оптимизация типов данных и структуры БД в ClickHouse

Ivan Leveev
Lead JavaCard
Developer





ВЕЩАЕТ

Иван Левеев

Занимаюсь разработкой JavaCard аплетов уже лет пять. Последний год занимаюсь всякими дата-инженерными штуками. Успел поработать с ClickHouse – понравилось 🎉

СВЯЗАТЬСЯ СО МНОЙ



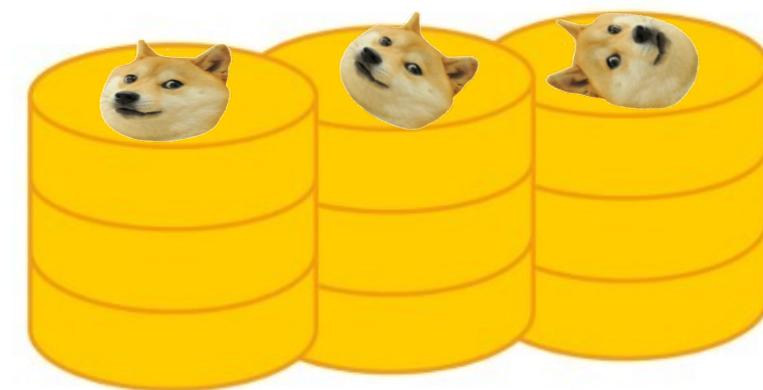
@i.leveev



@elise_azure



Click here to increase efficiency



Основные типы данных

1. Numeric (Int/UInt/Float/Decimal)
2. String/FixedString
3. Date/DateTime
4. UUID
5. Enum
6. Array(T)
7. LowCardinality(T)

Оптимизация строк

1. FixedString(N)
2. Enum8 (16)
3. LowCardinality(String)



Enum

- + Значения хранятся в схеме таблицы.
- + Оптимизация GROUP BY, ORDER BY, IN, DISTINCT.
- + Оптимизация некоторых случаев.
- Нужен ALTER для добавления значений.

ENUM ALL THE STRINGS



LowCardinality(String)

- + Динамическое расширение.
- + Оптимизация GROUP BY и некоторых функций.
- + Оптимизации из-за работы со словарем.
- + Кеширование вычислений.
- + Не нужен ALTER.

- Иногда работает медленнее Enum и даже обычных строк.

OR NOT



Немного чиселок (размеры колонок)

1. Uniq values: 6.

type	marks	rows	compressed_size	uncompressed_size	marks_size
enum	3373	27618755	336.37 KiB	26.34 MiB	79.05 KiB
dict	3373	27618755	343.79 KiB	26.39 MiB	158.11 KiB

2. Uniq values: 130, avg value length: 4.

type	marks	rows	compressed_size	uncompressed_size	marks_size
dict	3373	27618755	17.61 MiB	26.39 MiB	158.11 KiB
string	3373	27618755	35.98 MiB	128.29 MiB	79.05 KiB

Немного чиселок (размеры колонок)

3. Uniq values: 20508 , avg value length: 7.

type	marks	rows	compressed_size	uncompressed_size	marks_size
dict	3373	27618755	25.30 MiB	53.60 MiB	158.11 KiB
string	3373	27618755	44.39 MiB	224.64 MiB	79.05 KiB

4. Uniq values: 26 , avg value length: 40.

type	marks	rows	compressed_size	uncompressed_size	marks_size
dict	3373	27618755	16.57 MiB	26.39 MiB	158.11 KiB
string	3373	27618755	64.58 MiB	1.03 GiB	79.05 KiB

Обычные выборки

```
SELECT  
    field  
FROM  
    table  
PREWHERE  
    received_at >= '2020-05-01 00:00:00'  
    AND received_at < '2020-05-01 01:00:00'
```

Result: **389296** rows

Обычные выборки

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.11 s	27618755	380 KB
dict	0.12 s	27618755	435 KB

2. Uniq values: 91, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.10 s	27618755	1 MB
string	0.12 s	27618755	5 MB

3. Uniq values: 2986, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.13 s	27618755	87 MB
string	0.15 s	27618755	6 MB

4. Uniq values: 3, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.17 s	27618755	891 KB
string	0.60 s	27618755	15 MB

Запросы с группировкой

много данных

```
SELECT
    field,
    count()
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-02 00:00:00'
GROUP BY
    field
```

Запросы с группировкой

МНОГО данных

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.09 s	27618755	26 MB
dict	0.10 s	27618755	26 MB

2. Uniq values: 130, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.09 s	27618755	27 MB
string	0.17 s	27618755	339 MB

3. Uniq values: 20508, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.14 s	27618755	109 MB
string	0.20 s	27618755	435 MB

4. Uniq values: 26, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.10 s	27618755	27 MB
string	0.20 s	27618755	1 GB

Запросы с группировкой

мало данных

```
SELECT
    field,
    count()
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-01 00:00:10'
GROUP BY
    field
```

Запросы с группировкой

мало данных

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.07 s	27618755	21 MB
dict	0.07 s	27618755	21 MB

3. Uniq values: 220, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.09 s	27618755	46 MB
string	0.07 s	27618755	21 MB

2. Uniq values: 52, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.07 s	27618755	21 MB
string	0.07 s	27618755	21 MB

4. Uniq values: 2, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.07 s	27618755	21 MB
string	0.08 s	27618755	63 MB

Запросы с сортировкой

```
SELECT
    field
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-01 01:00:00'
ORDER BY
    field
```

Result: **389296** rows

Запросы с сортировкой

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.23 s	27618755	380 KB
dict	0.40 s	27618755	435 KB

3. Uniq values: 2986, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.60 s	27618755	87 MB
string	0.35 s	27618755	6 MB

2. Uniq values: 91, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.45 s	27618755	1 MB
string	0.35 s	27618755	5 MB

4. Uniq values: 3, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.45 s	27618755	891 KB
string	0.70 s	27618755	15 MB

Запросы с DISTINCT

МНОГО данных

```
SELECT
    distinct(field)
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-02 00:00:00'
```

Запросы с DISTINCT

МНОГО данных

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.07 s	27618755	26 MB
dict	0.30 s	27618755	26 MB

2. Uniq values: 130, avg value length: 4.

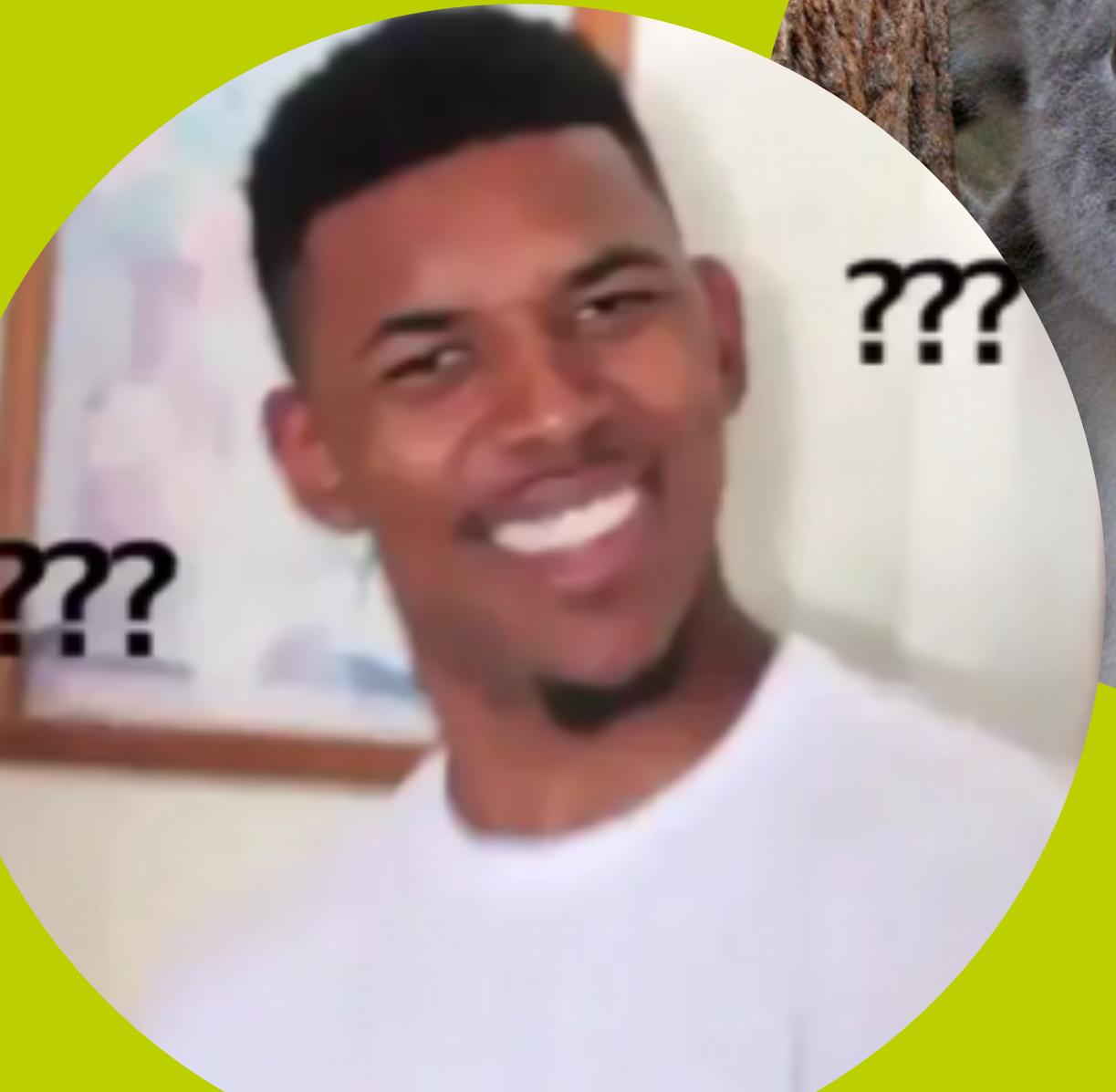
type	elapsed	rows_read	bytes_read
dict	0.30 s	27618755	27 MB
string	0.17 s	27618755	339 MB

3. Uniq values: 20508, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.33 s	27618755	109 MB
string	0.20 s	27618755	435 MB

4. Uniq values: 26, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.37 s	27618755	27 MB
string	0.17 s	27618755	1 GB



Запросы с DISTINCT

мало данных

```
SELECT
    distinct(field)
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-01 01:00:00'
```

Запросы с DISTINCT

мало данных

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.07 s	27618755	380 KB
dict	0.07 s	27618755	435 KB

2. Uniq values: 91, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.08 s	27618755	1 MB
string	0.12 s	27618755	5 MB

3. Uniq values: 2986, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.09 s	27618755	87 MB
string	0.14 s	27618755	6 MB

4. Uniq values: 3, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.08 s	27618755	891 KB
string	0.12 s	27618755	15 MB

Запросы с фильтрами

по одному значению

```
SELECT
    field
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-01 01:00:00'
WHERE
    field = 'value'
```

Запросы с фильтрами

по одному значению

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.07 s	27618755	380 KB
dict	0.07 s	27618755	435 KB

2. Uniq values: 91, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.07 s	27618755	1 MB
string	0.12 s	27618755	5 MB

3. Uniq values: 2986, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.09 s	27618755	87 MB
string	0.14 s	27618755	6 MB

4. Uniq values: 3, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.08 s	27618755	891 KB
string	0.13 s	27618755	15 MB

Запросы с фильтрами

по нескольким значениям

```
SELECT
    field
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
        AND received_at < '2020-05-01 01:00:00'
WHERE
    field IN ('value1', 'value2')
```

Запросы с фильтрами

по нескольким значениям

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.07 s	27618755	380 KB
dict	0.07 s	27618755	435 KB

2. Uniq values: 91, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.07 s	27618755	1 MB
string	0.13 s	27618755	5 MB

3. Uniq values: 2986, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.10 s	27618755	87 MB
string	0.14 s	27618755	6 MB

4. Uniq values: 3, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.12 s	27618755	891 KB
string	0.13 s	27618755	15 MB

Запросы с агрегацией

uniq

```
SELECT
    uniq(field)
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-02 00:00:00'
```

Запросы с агрегацией

uniq

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.11 s	27618755	26 MB
dict	0.17 s	27618755	26 MB

2. Uniq values: 130, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.17 s	27618755	27 MB
string	0.20 s	27618755	339 MB

3. Uniq values: 20508, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.19 s	27618755	109 MB
string	0.21 s	27618755	435 MB

4. Uniq values: 26, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.20 s	27618755	27 MB
string	0.20 s	27618755	1 GB

Запросы с агрегацией

topK

```
SELECT
    topK(3)(field)
FROM
    table
PREWHERE
    received_at >= '2020-05-01 00:00:00'
    AND received_at < '2020-05-02 00:00:00'
```

Запросы с агрегацией

topK

1. Uniq values: 6.

type	elapsed	rows_read	bytes_read
enum	0.13 s	27618755	26 MB
dict	0.19 s	27618755	26 MB

2. Uniq values: 130, avg value length: 4.

type	elapsed	rows_read	bytes_read
dict	0.19 s	27618755	27 MB
string	0.21 s	27618755	339 MB

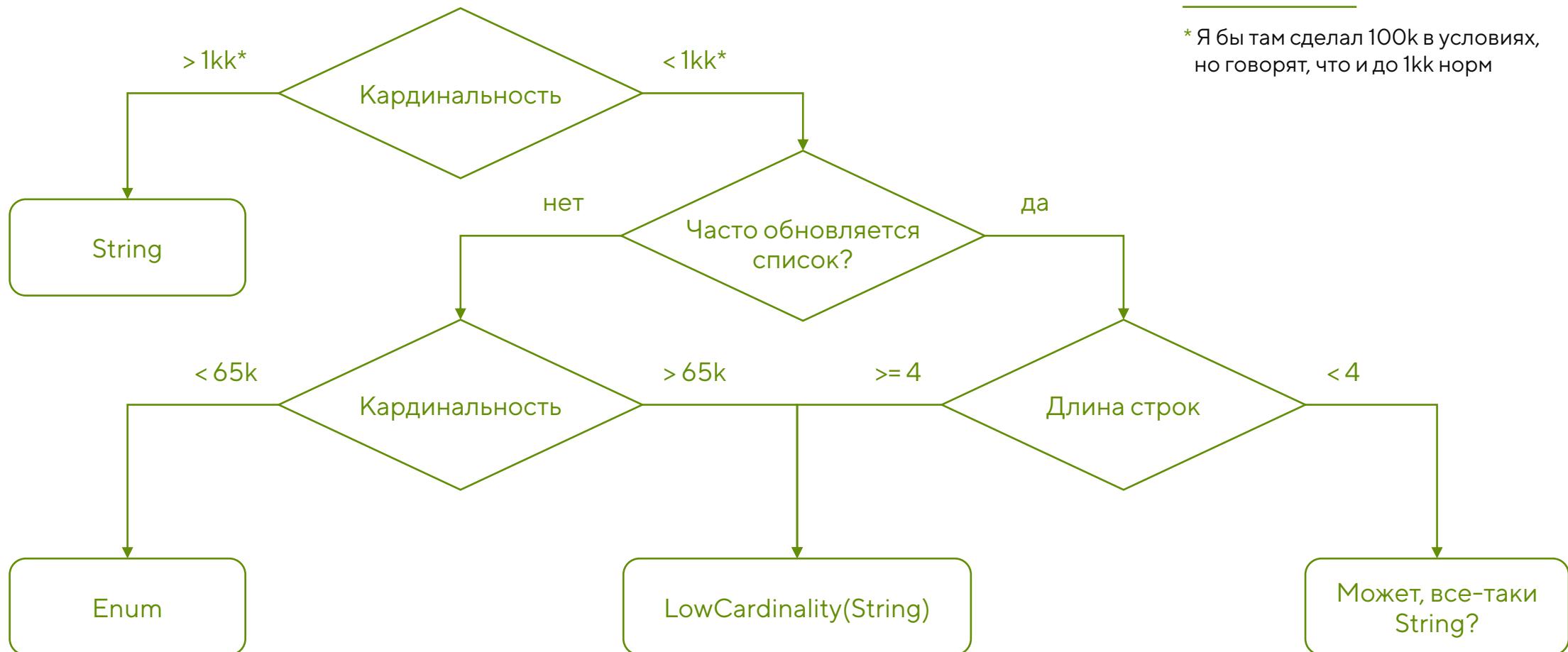
3. Uniq values: 20508, avg value length: 7.

type	elapsed	rows_read	bytes_read
dict	0.21 s	27618755	109 MB
string	0.24 s	27618755	435 MB

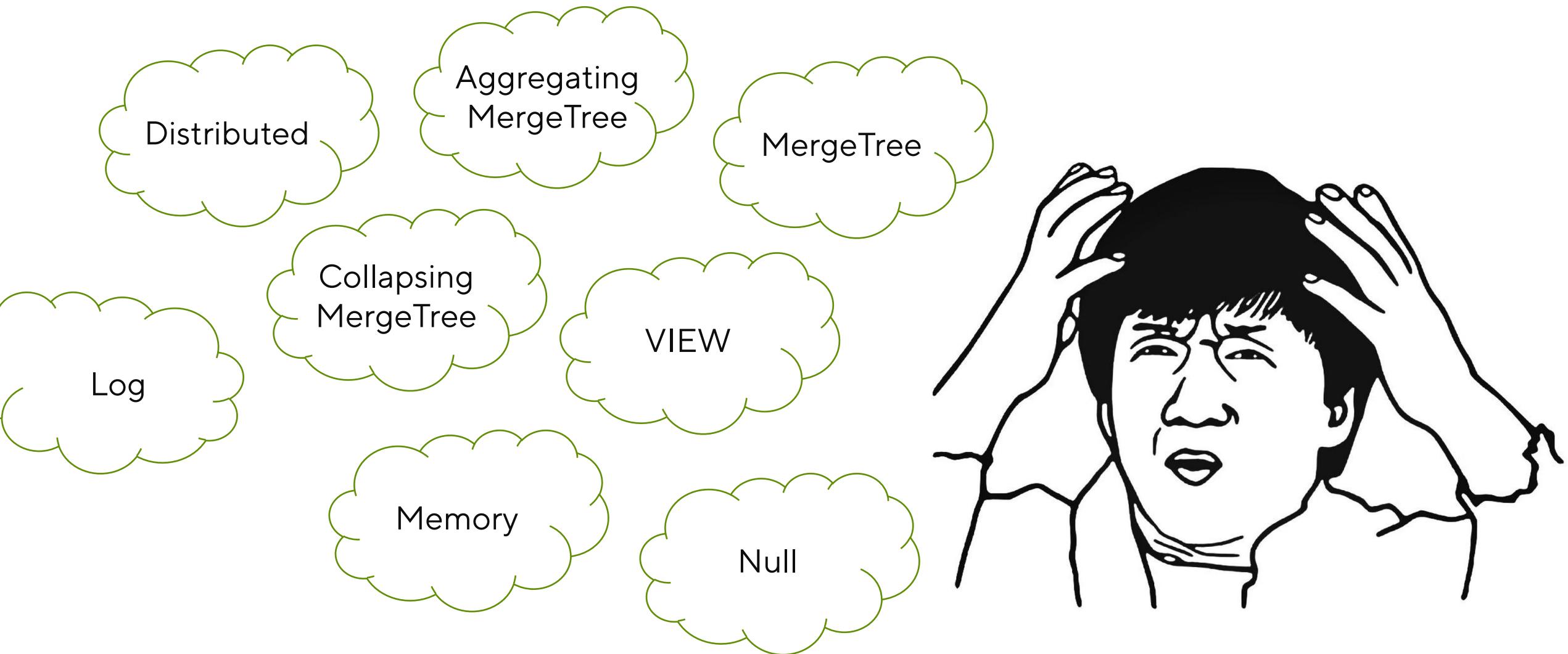
4. Uniq values: 26, avg value length: 40.

type	elapsed	rows_read	bytes_read
dict	0.21 s	27618755	27 MB
string	0.23 s	27618755	1 GB

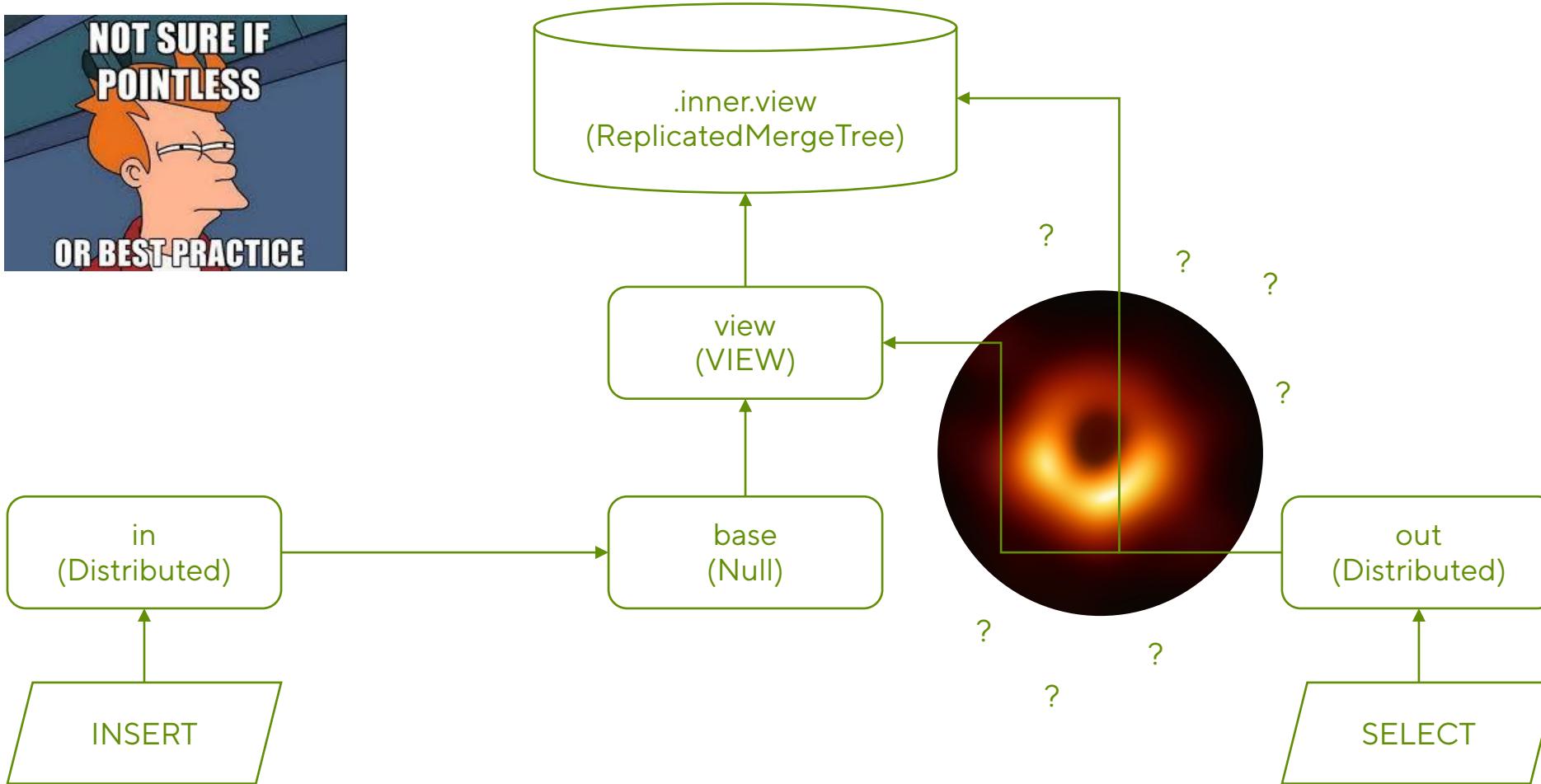
Что выбрать?



Немножко про схему БД



Немножко про схему БД



Бонус: PREWHERE MAGIC

```
CREATE TABLE foo(A Int64, V Int64, S String) Engine = NULL;

CREATE MATERIALIZED VIEW fooM Engine = MergeTree
ORDER BY A
AS SELECT A, V, S FROM foo;

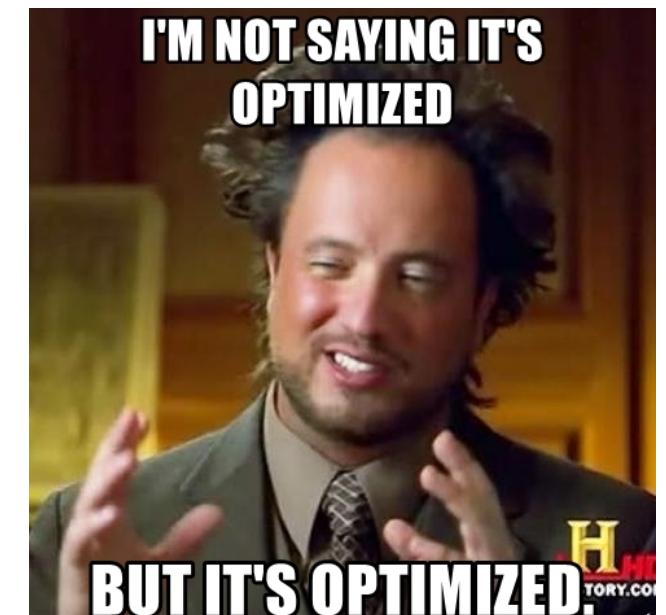
INSERT INTO foo
SELECT number, sipHash64(number), toString(arrayMap(x -> sipHash64(number*x), RANGE(10)))
FROM numbers(10000000);

SET optimize_move_to_prewhere = 1;

SELECT * FROM ".inner.fooM" WHERE V = 777;
0 rows in set. Elapsed: 0.009 sec. Processed 10.00 million rows

SELECT * FROM fooM PREWHERE V = 777;
0 rows in set. Elapsed: 0.008 sec. Processed 10.00 million rows

SELECT * FROM fooM WHERE V = 777;
0 rows in set. Elapsed: 0.485 sec. Processed 10.00 million rows
```



Вопросы?



Ivan Leveev

Lead JavaCard Developer
BigData Enthusiast

OPTIMISATION



**Спасибо
за внимание!**