

**Y**andex



# Запрос EXPLAIN

Николай Кочетов, разработчик ClickHouse

## Лог сервера (до 20.3, или с `experimental_use_processors = 0`)

```
<Debug> executeQuery: SELECT sum(number) FROM numbers(10) GROUP BY number % 2
```

```
<Debug> executeQuery: Query pipeline:
```

```
Expression
```

```
  Expression
```

```
    Aggregating
```

```
      Concat
```

```
        Expression
```

```
          Numbers
```

## EXPLAIN (20.6+)

```
EXPLAIN description = 0
```

```
SELECT sum(number) FROM numbers(10) GROUP BY number % 2
```

```
Expression
```

```
  Expression
```

```
    Aggregating
```

```
      Expression
```

```
        ReadFromStorage
```

# Варианты EXPLAIN

```
EXPLAIN AST
  SYNTAX
  PLAN header = 0,
        description = 1,
        actions = 0,
        optimize = 1
  PIPELINE header = 0,
           graph = 0,
           compact = 1
```

SELECT ...

AST - абстрактное синтаксическое дерево

SYNTAX - текст запроса после оптимизаций AST

PLAN - план выполнения запроса

PIPELINE - конвейер выполнения запроса

# AST запроса

```
EXPLAIN AST  
SELECT 1, 2 + 3
```

```
SelectWithUnionQuery (children 1)  
  ExpressionList (children 1)  
    SelectQuery (children 1)  
      ExpressionList (children 2)  
        Literal UInt64_1  
        Function plus (children 1)  
          ExpressionList (children 2)  
            Literal UInt64_2  
            Literal UInt64_3
```

# Синтаксис запроса

```
EXPLAIN SYNTAX
```

```
SELECT * FROM system.numbers AS a, system.numbers AS b, system.numbers AS c
```

```
SELECT
```

```
  `--a.number` AS `a.number`,  
  `--b.number` AS `b.number`,  
  number AS `c.number`
```

```
FROM
```

```
(
```

```
  SELECT
```

```
    number AS `--a.number`,  
    b.number AS `--b.number`
```

```
  FROM system.numbers AS a
```

```
  CROSS JOIN system.numbers AS b
```

```
) AS `--s`
```

```
CROSS JOIN system.numbers AS c
```

# План выполнения запроса

```
EXPLAIN PLAN
SELECT sum(number) + 1 AS x
FROM numbers(10)
GROUP BY number % 2
```

Expression (Projection)

Expression (Before ORDER BY and SELECT)

Aggregating

Expression (Before GROUP BY)

ReadFromStorage (Read from SystemNumbers)

- › Нет оценки стоимости выполнения
- › TODO: EXPLAIN ANALYZE

# План выполнения запроса

```
EXPLAIN header = 1  
SELECT sum(number) + 1 AS x  
FROM numbers(10)  
GROUP BY number % 2
```

Expression (Projection)

Header: x UInt64

Expression (Before ORDER BY and SELECT)

Header: modulo(number, 2) UInt8

plus(sum(number), 1) UInt64

Aggregating

Header: modulo(number, 2) UInt8

sum(number) UInt64

Expression (Before GROUP BY)

Header: number UInt64

modulo(number, 2) UInt8

ReadFromStorage (Read from SystemNumbers)

Header: number UInt64



# План выполнения запроса

```
EXPLAIN actions = 1  
SELECT sum(number) + 1 AS x FROM numbers(10) GROUP BY number % 2
```

Expression (Projection)

Actions: PROJECT plus(sum(number), 1) AS x

Expression (Before ORDER BY and SELECT)

Actions: ADD 1 UInt8 Const(UInt8)

FUNCTION plus(sum(number), 1) UInt64 = plus(sum(number), 1)

REMOVE sum(number)

REMOVE 1

Aggregating

Keys: modulo(number, 2)

Aggregates:

sum(number)

Function: sum(UInt64) → UInt64

Arguments: number

Argument positions: 0

Expression (Before GROUP BY)

Actions: ADD 2 UInt8 Const(UInt8)

FUNCTION modulo(number, 2) UInt8 = modulo(number, 2)

REMOVE 2

ReadFromStorage (Read from SystemNumbers)

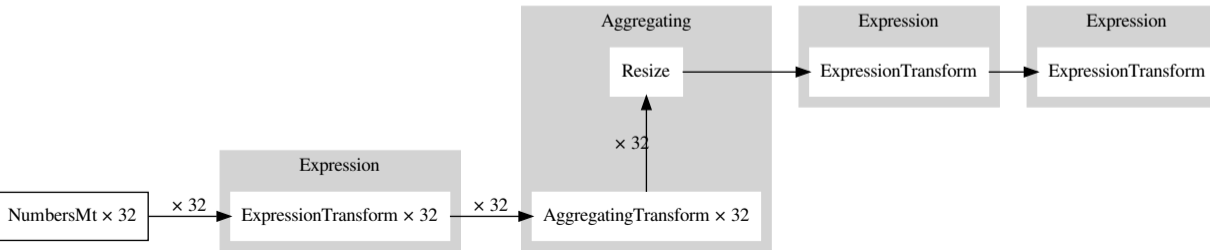
# Конвейер выполнения запроса

```
EXPLAIN PIPELINE
SELECT sum(x + 1) as y FROM (SELECT number + 2 AS x FROM system.numbers_mt LIMIT 100000)
(Expression)
ExpressionTransform
(Expression)
ExpressionTransform
(Aggregating)
Resize 32 → 1
AggregatingTransform × 32
StrictResize 32 → 32
(Expression)
ExpressionTransform × 32
(Expression)
ExpressionTransform × 32
(Expression)
ExpressionTransform × 32
(Limit)
Limit 32 → 32
(ReadFromStorage)
NumbersMt × 32
```

# Конвейер выполнения запроса

```
EXPLAIN PIPELINE graph = 1
SELECT sum(number) AS x
FROM numbers_mt(1000000)
GROUP BY number % 2
FORMAT TSV
```

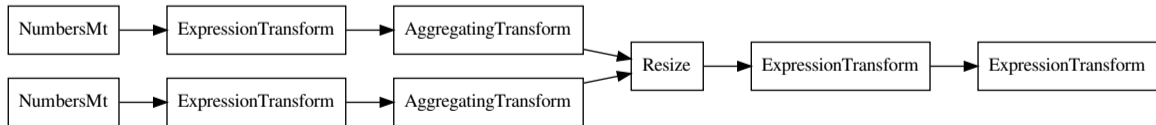
```
digraph
{
  ...
```



# Конвейер выполнения запроса

```
EXPLAIN PIPELINE graph = 1, compact = 0
SELECT sum(number) AS x
FROM numbers_mt(1000000)
GROUP BY number % 2
FORMAT TSV
SETTINGS max_threads = 2
```

```
digraph
{
  ...
}
```



# Пример анализа запроса

```
CREATE TABLE events (t DateTime, value UInt64) ENGINE = MergeTree ORDER BY t;  
  
INSERT INTO events SELECT  
    toDate('2000-01-01') + toIntervalSecond(number * 30), number  
FROM numbers(100000000);
```

## Последние 10 событий

```
SELECT toStartOfMinute(t) AS m, value FROM events ORDER BY m DESC LIMIT 10
```

```
2095-01-24 05:19:00      99999999  
...
```

```
10 rows in set. Elapsed: 0.015 sec. Processed 901.38 thousand rows, 10.82 MB
```

```
SELECT toStartOfMinute(t, 'UTC') AS m, value FROM events ORDER BY m DESC LIMIT 10
```

```
2095-01-24 02:19:00      99999999  
...
```

```
10 rows in set. Elapsed: 0.188 sec. Processed 100.00 million rows, 1.20 GB
```

# Пример анализа запроса

EXPLAIN

```
SELECT toStartOfMinute(t) AS m, value FROM events ORDER BY m DESC LIMIT 10
```

Expression (Projection)

Limit (preliminary LIMIT)

FinishSorting

Expression (Before ORDER BY and SELECT)

ReadFromStorage (Read from MergeTree)

EXPLAIN

```
SELECT toStartOfMinute(t, 'UTC') AS m, value FROM events ORDER BY m DESC LIMIT 10
```

Expression (Projection)

Limit (preliminary LIMIT)

MergingSorted (Merge sorted streams for ORDER BY)

MergeSorting (Merge sorted blocks for ORDER BY)

PartialSorting (Sort each block for ORDER BY)

Expression (Before ORDER BY and SELECT)

ReadFromStorage (Read from MergeTree)

# Пример анализа запроса

```
SELECT toStartOfMinute(t, 'UTC') AS m, value FROM events ORDER BY t DESC LIMIT 10
2095-01-24 02:19:00      99999999
...
10 rows in set. Elapsed: 0.013 sec. Processed 901.38 thousand rows, 10.82 MB
```

```
EXPLAIN
SELECT toStartOfMinute(t, 'UTC') AS m, value FROM events ORDER BY t DESC LIMIT 10

Expression (Projection)
  Limit (preliminary LIMIT)
    FinishSorting
      Expression (Before ORDER BY and SELECT)
        ReadFromStorage (Read from MergeTree)
```

# Оптимизации плана запроса

```
SELECT number + 1 FROM numbers(100000000) ORDER BY number DESC LIMIT 10
```

10 rows in set. Elapsed: 1.171 sec. Processed 100.03 million rows, 800.21 MB

```
SELECT number + 1 FROM (  
    SELECT number FROM numbers(100000000) ORDER BY number DESC  
) LIMIT 10
```

10 rows in set. Elapsed: 1.132 sec. Processed 100.03 million rows, 800.21 MB

```
CREATE VIEW numbers_100m_desc AS  
SELECT number FROM numbers(100000000) ORDER BY number DESC
```

```
SELECT number + 1 FROM numbers_100m_desc LIMIT 10
```

10 rows in set. Elapsed: 5.961 sec. Processed 100.03 million rows, 800.21 MB



# Оптимизации плана запроса

```
SELECT number + 1 FROM (  
    SELECT number FROM numbers(100000000) ORDER BY number DESC  
) LIMIT 10
```

```
EXPLAIN actions = 1, optimize = 0
```

```
Expression (Projection)  
Actions: PROJECT plus(number, 1)  
    Limit (preliminary LIMIT)  
    Limit 10  
    Offset 0  
    Expression (Before ORDER BY and SELECT)  
    Actions: ADD 1 UInt8 Const(UInt8)  
            FUNCTION plus(number, 1)  
            REMOVE number  
            REMOVE 1  
    Expression (Projection)  
    Actions: PROJECT number  
    MergingSorted (Merge sorted streams for ORDER BY)  
    Sort description: number DESC  
  
    MergeSorting (Merge sorted blocks for ORDER BY)  
    Sort description: number DESC  
  
    PartialSorting (Sort each block for ORDER BY)  
    Sort description: number DESC  
  
    Expression (Before ORDER BY and SELECT)  
    ReadFromStorage (Read from SystemNumbers)
```

```
EXPLAIN actions = 1, optimize = 1
```

```
Expression (Projection)  
Actions: PROJECT plus(number, 1)  
    Expression (Before ORDER BY and SELECT)  
    Actions: ADD 1 UInt8 Const(UInt8)  
            FUNCTION plus(number, 1) UInt64  
            REMOVE number  
            REMOVE 1  
    Expression (Projection)  
    Actions: PROJECT number  
    Limit (preliminary LIMIT)  
    Limit 10  
    Offset 0  
    MergingSorted (Merge sorted streams for ORDER BY)  
    Sort description: number DESC  
    Limit 10  
    MergeSorting (Merge sorted blocks for ORDER BY)  
    Sort description: number DESC  
    Limit 10  
    PartialSorting (Sort each block for ORDER BY)  
    Sort description: number DESC  
    Limit 10  
    Expression (Before ORDER BY and SELECT)  
    ReadFromStorage (Read from SystemNumbers)
```

# Оптимизации плана запроса

## LIMIT push down

- › Для подзапросов - с версии 20.7
- › Для VIEW - на стадии ревью

## В очереди

- › Predicate push down
- › GROUP BY push down
- › ORDER BY lift up

# Summary

## EXPLAIN

- › Доступна с версии 20.6
- › В процессе разработки
- › Полезна для анализа запроса

## TODO

- › ANALYZE и профилировка запроса [#15261](#)
- › Поддержка Distributed запросов
- › Оптимизации плана запроса