Yandex

Yandex

EXPLAIN query

Nikolai Kochetov, ClickHouse developer

Server log (before 20.3, or with `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
```

EXPLAIN types

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

AST - abstract syntax tree
SYNTAX - query text after AST-level optimizations
PLAN - query execution plan
PIPELINE - query execution pipeline

Query AST

```
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
```

Query syntax

```
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
```

Query execution plan

```
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)
```

- Step (and query) cost estimation is not supported
- TODO: EXPLAIN ANALYZE

Query execution plan

```
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
```

Query execution plan

```
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)
```

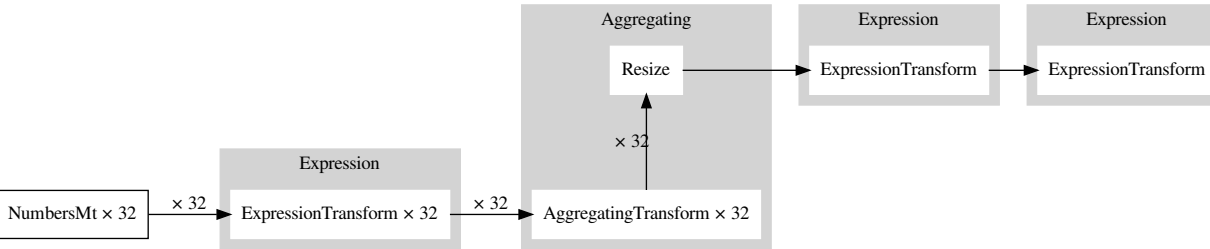
Query execution pipeline

```
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
```

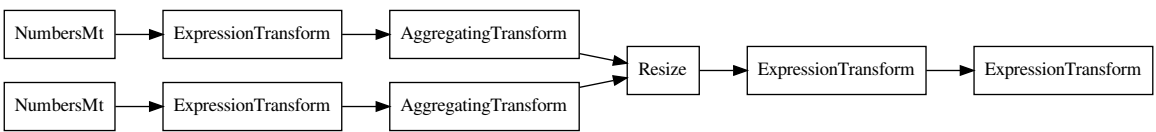
Query execution pipeline

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

digraph
{
...
```



Query execution pipeline

```
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
{
...
```



Query analysis example

```
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);
```

Last 10 events

```
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
```

Query analysis example

```
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)
```

Query analysis example

```
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)
```

Query plan optimization

```
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
```

Query plan optimization

```
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)
```

Query plan optimization

LIMIT push down

- Support for subqueries - from 20.7
- Support for VIEW - pr, review

Other optimizations

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

Summary

EXPLAIN

- Available since 20.6
- In the development stage
- Helpful for query analysis

TODO

- ANALYZE and query profiling #15261
- Support for Distributed queries
- Query plan optimizations