НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Факультет Компьютерных Наук
БПМИ 175 Ильговский Роман Максимович

Генерация искусственных данных для тестирования заданных запросов. Обфускация запросов для тестирования ClickHouse

Курсовая работа, программный проект.

Москва, 2020

ClickHouse

ClickHouse - это колоночная аналитическая СУБД с открытым кодом, позволяющая выполнять аналитические запросы в режиме реального времени на структурированных больших данных, разрабатываемая компанией Яндекс

Зачем нужна генерация базы данных

https://github.com/ClickHouse/ClickHouse/issues/11215

WHERE 1 or WHERE 1=1 causes bugs when using GLOBAL JOIN #11215

New issue

Open **toannhu96** opened this issue 8 days ago · 3 comments

toannhu96 commented 8 days ago • edited

Describe the bug
I use **GLOBAL ANY LEFT JOIN** to join metadata table with fact table.
When I use **WHERE 1=1** to add additional conditions with **AND...** then it return nothings when left join with metadata table

How to reproduce

- Which ClickHouse server version to use: **19.17**
- Which interface to use, if matters: **DataGrip**
- Queries to run that lead to unexpected result`

Assignees
4ertus2

Labels
bug
comp-joins
obsolete-version

Projects
None yet

```
SELECT *
FROM (
      SELECT *
      FROM cdp.recommendation_metadata_prod
      GLOBAL ANY
      LEFT JOIN
           (SELECT recommend_id       AS uuid,
                   sum(sends)         AS sends,
                   sum(opens)         AS opens,
                   sum(clicks)        AS clicks,
                   sum(unique_clicks) AS unique_clicks,
                   sum(unique_opens)  AS unique_opens,
                   sum(orders)        AS orders,
                   sum(revenue)       AS revenue,
                   sum(activations)   AS activations,
                   sum(revenue_gmv)   AS revenue_gmv,
                   sum(revenue_cmv)   AS revenue_cmv,
                   sum(revenue_nmv)   AS revenue_nmv,
                   sum(orders_nmv)    AS orders_nmv,
                   sum(orders_cmv)    AS orders_cmv
            FROM (SELECT date_key,
                         recommend_id,
                         sends,
                         opens,
                         clicks,
                         unique_clicks,
                         unique_opens,
                         orders,
                         transaction_revenue AS revenue,
                         activations,
                         transaction_gmv     AS revenue_gmv,
                         cmv                 AS revenue_cmv,
                         nmv                 AS revenue_nmv,
                         net_orders          AS orders_nmv,
                         confirmed_orders    AS orders_cmv
                  FROM cdp.view_campaign_performance
                  WHERE date_key BETWEEN '2020-05-21' AND '2020-05-27'
                        AND isNotNull(recommend_id)
                        AND notEmpty(recommend_id)

                  UNION ALL

                  SELECT date_key,
                         recommend_id,
                         sends,
                         opens,
                         clicks,
                         unique_clicks,
                         unique_opens,
                         total_orders  AS orders,
                         total_revenue AS revenue,
                         activations,
                         0.0           AS revenue_gmv,
                         0.0           AS revenue_cmv,
                         0.0           AS revenue_nmv,
                         0             AS orders_nmv,
                         0             AS orders_cmv
                  FROM cdp.realtime_campaign_performance
                  WHERE date_key BETWEEN '2020-05-21' AND '2020-05-27'
                        AND isNotNull(recommend_id)
                        AND notEmpty(recommend_id)
                )
           GROUP BY uuid)
      USING (uuid)
      WHERE 1=1
      ORDER BY event_time DESC
      LIMIT 1 BY uuid)
 ORDER BY created_at DESC
LIMIT 10 OFFSET 0;
```

Постановка задачи

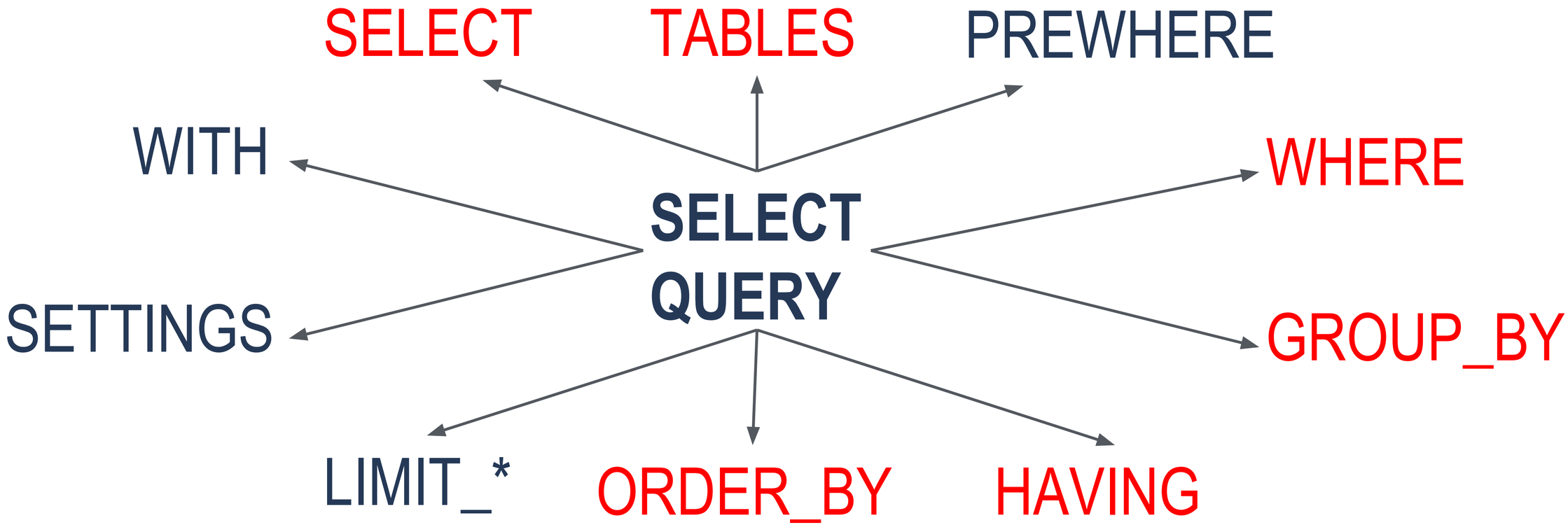
Задача:

- Дана строка запроса. Необходимо сгенерировать базу данных, на которой запрос выполнится без ошибок.

Подзадачи:

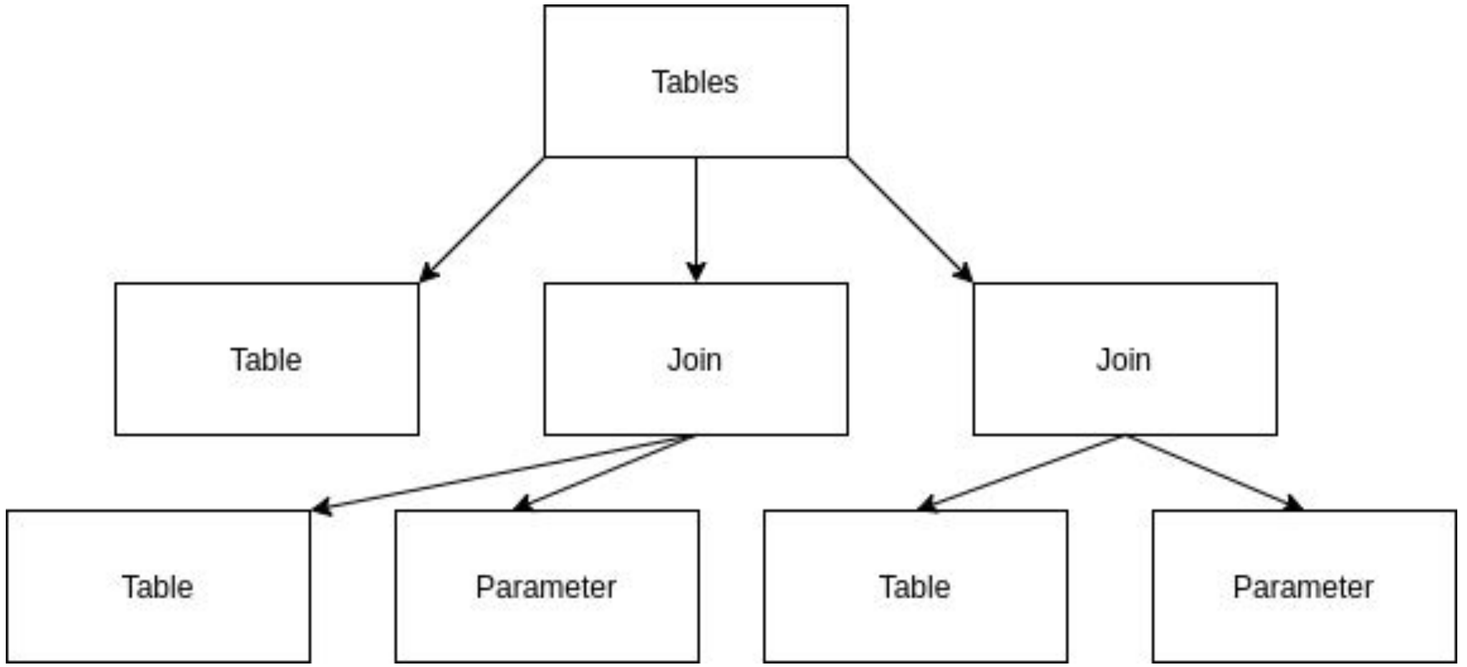
- Определить структуру базы данных по запросу
- Заполнить базу данных значениями

Анализ запроса в ClickHouse



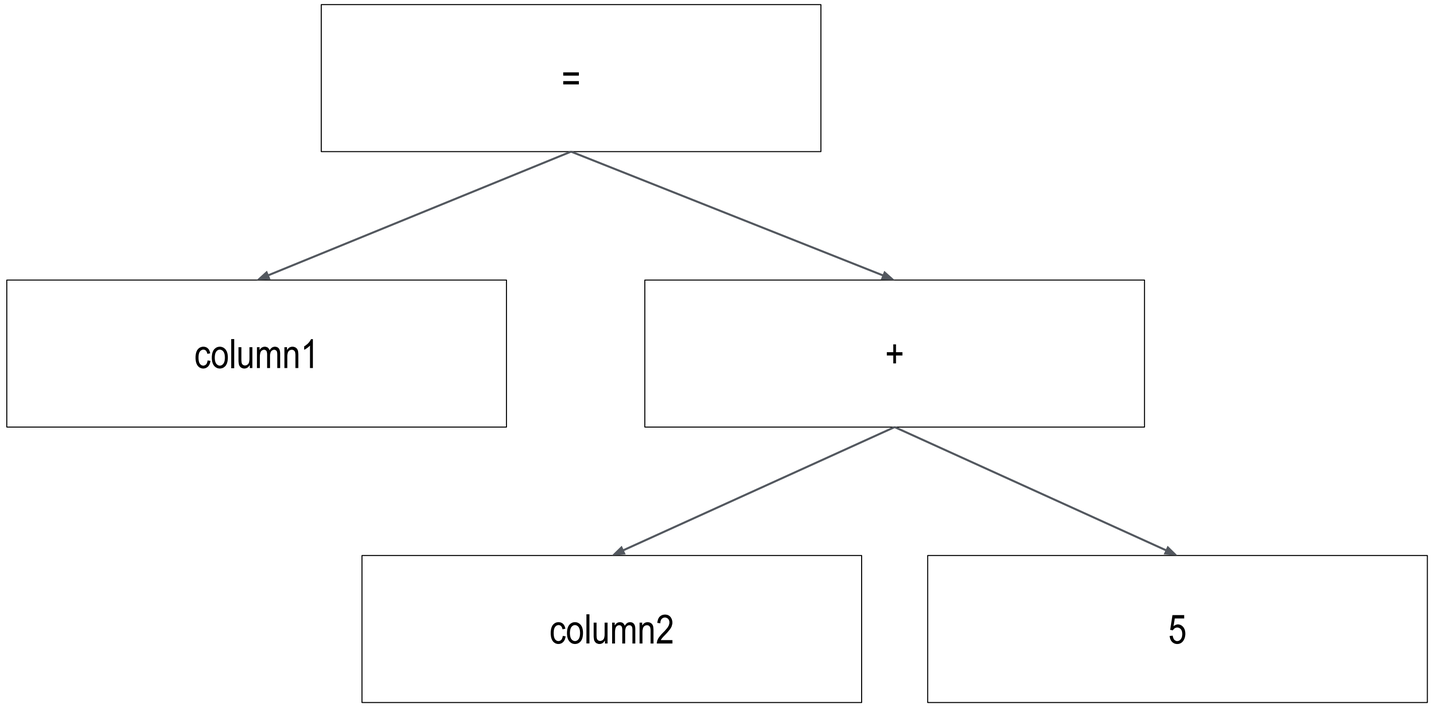
Определение таблиц

```
SELECT
*
FROM db.table1
JOIN db.table2 AS second
ON
db.table1.sim_value1 = second.sim_value2
```



Определение столбцов

```
SELECT
 integer, second.float, arrayjoin(array)
FROM db.table1
JOIN db.table2 AS second
ON
db.table1.sim_value1  =  second.sim_value2
WHERE
integer > 5 AND
second.float > 100/3 AND
date = yesteray() - 5;
```



Анализ столбцов

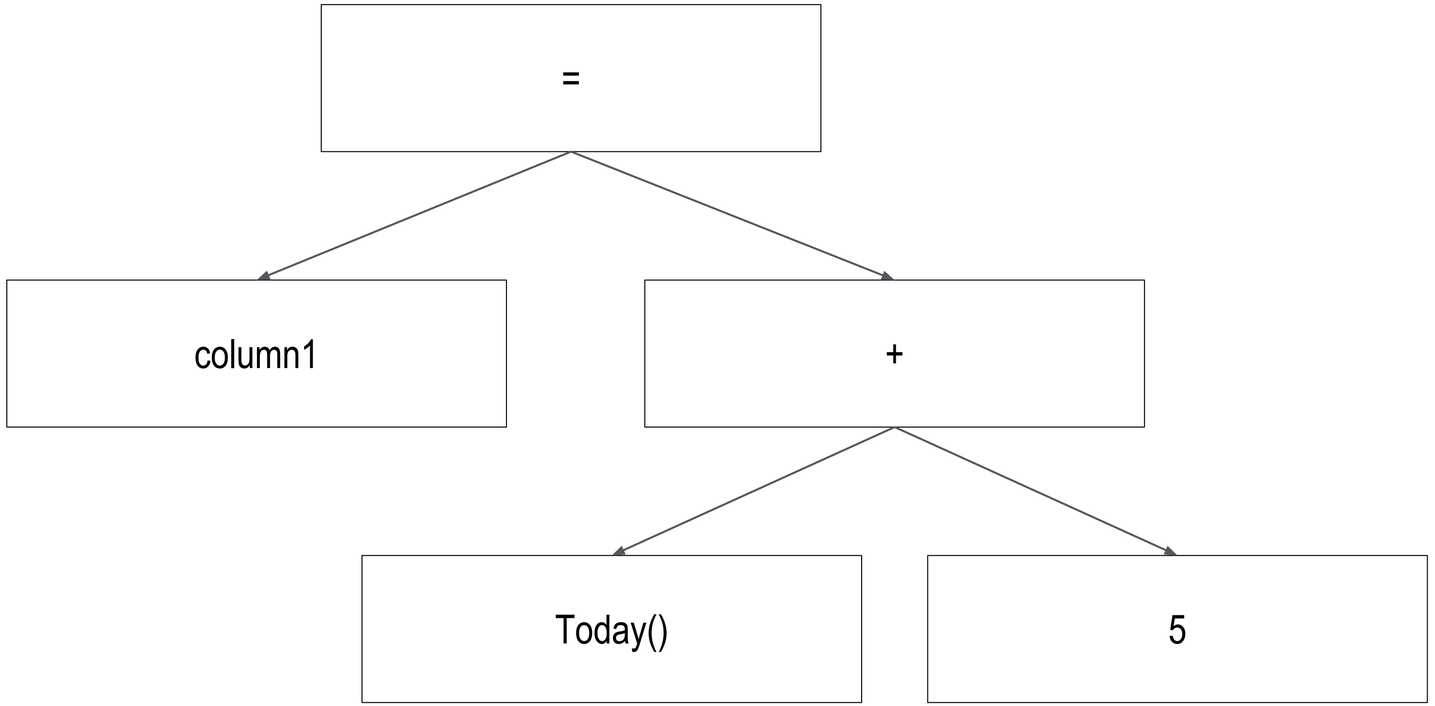
Необходимо выяснить

• Принадлежность столбца к таблице	*table1*.*column1*, *column2*, *alias*.*column3*
• Тип значений в столбце: INT, FLOAT, DATE, DATETIME, ARRAY	column1 = 5, column2 LIKE “%string%”
• Возможные значения столбца	column1 > 5 / 100, column3 = today() - 10
• Связанные столбцы	join … on table1.column1 = table2.column3

Анализ столбцов

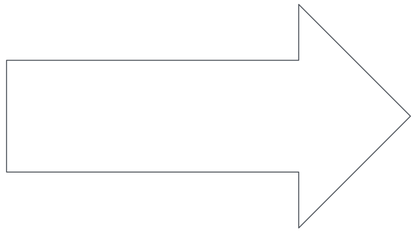
Обработчик функции

- Возвращаемый тип
- Тип параметров
- Связанность элементов
- Возвращаемое значение (если есть)



Результаты анализа

- Множество таблиц
- Для каждой таблицы множество колонок
- Для каждой колонки:
  - Название
  - Тип
  - Множество значений



Запросы для создания базы данных

Заполнение базы данных значениями из декартового произведения значений столбцов для каждой таблицы

Пример использования

```
SELECT
 integer, second.float, arrayjoin(array)
FROM db.table1
JOIN db.table2 AS second
ON
db.table1.sim_value1  =  second.sim_value2
WHERE
 integer > 5 AND
 second.float > 100/3 AND
 date = yesteray() - 5;
```

Пример использования

Table: *db.table1*
Columns:
array (*db.table1.array*)
type: I, ARR
values: [-648416411, -1030677187, 2087194344, 896488399, -1400453402, -958290909], [-850244462, 1449933719, 1492774445],
equal:

date(*db.table1.date*)
type: DT
values: yesterday() - 5, toDateTime(yesterday() - 5) + 142, toDateTime(yesterday() - 5) - 7930
equal:

integer(db.table1.integer)
type: I
values: 5, 5 + 1, 5 - 4
equal:

sim_value1(*db.table1.sim_value1*)
type: I
values: -1454026639, 1136820438, 1832565398
equal: *db.table2.sim_value2*

Table: *db.table2*
Columns:
float (*db.table2.float*)
type: F
values: 100 / 3, 100 / 3 + 1.166667, 100 / 3 - 1.235294
equal:

sim_value2 (*db.table2.sim_value2*)
type: I
values: -1454026639, 1136820438, 1832565398
equal: *db.table1.sim_value1*,

Пример использования

```
CREATE DATABASE IF NOT EXISTS db;

CREATE TABLE IF NOT EXISTS db.table2 (
float Float64,
sim_value2 Int64
) ENGINE = Log;

INSERT INTO db.table2
(float, sim_value2) VALUES
(100 / 3, -1454026639),
...
```

```
CREATE DATABASE IF NOT EXISTS db;

CREATE TABLE IF NOT EXISTS db.table1 (
array Array(Int64),
date DateTime,
integer Int64,
sim_value1 Int64
) ENGINE = Log;

INSERT INTO db.table1
(array, date, integer, sim_value1) VALUES
([-648416411, -1030677187, 2087194344, 896488399,
-1400453402, -958290909], yesterday() - 5, 5,
-1454026639),
...
```

Заключение

Результат

Создан скрипт, который уменьшит затраты разработчиков на решение проблем пользователей.

Дальнейшая работа

- Поддержка необходимых функций
- Тестирование на запросах пользователей
- Улучшение работы с алиасами колонок

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ