

Выпускная квалификационная работа

Улучшение аналитической СУБД ClickHouse

Третьяков Александр Александрович, гр. 155

Руководитель: Миловидов Алексей Николаевич

Цель дипломной работы

Добавление новой функциональности в аналитической СУБД ClickHouse для использования prepared statements

Prepared statements – подготовленные запросы с параметрами, используются для многократного выполнения похожих запросов



Актуальность задачи

- Востребованная пользователями возможность, предоставляющая удобную работу с prepared statements, ранее нереализованная в ClickHouse
- Поиск нужной информации через обращения к серверу, используя заранее заданный запрос с параметрами



Задачи дипломной работы

Для поддержки prepared statements:

1. Изучить официальную документацию ClickHouse
2. Разработать концепцию взаимодействия с сервером
3. Реализовать методы парсинга для обработки параметризованных запросов
4. Написать тесты для системы с внедренным решением

Метод решения

Выбор синтаксиса параметризованных запросов

Сложности:

- обычный синтаксис, где подстановки обозначаются с помощью знака '?'
- в ClickHouse строгая типизация, поэтому нужно знать тип подставляемых данных
- знак вопроса уже используется для тернарного оператора '?'

Решение:

- указывать тип и имя подстановки в фигурных скобках

Поддержка новых токенов

Решение:

- использование фигурных скобок «{», «}»
- использование двоеточия для разделения имени подстановки от типа «:»

Пример:

- `WHERE id = {Name:Type}`

Взаимодействие с сервером

Сложности:

- передать значения подстановок серверу

Решение:

- использовать аргументы командной строки
- использовать параметры URL

Разбор нового аргумента запроса

Сложности:

- совместить в аргументе как название подстановки, так и его значение

Решение:

- использовать префикс «param_»
- разделить нижним подчеркиванием аргумент от названия подстановки

Пример:

- --param_Name=Value

Примеры запросов

Использование через аргументы командной строки:

```
clickhouse-client --param_user_id=123 --param_phrase='Hello, world'  
--query="SELECT * FROM table WHERE id = {user_id:UInt64} and  
s = {phrase:String}"
```

Использование через параметры URL:

```
curl http://localhost:8123/?param_ps_ids=[123,456,789]  
-d 'SELECT * FROM table WHERE ids = {ps_ids:Array(UInt64)}'
```

Программная реализация

Реализация prepared statements

Для поддержки prepared statements необходимо:

1. Принимать и разбирать аргументы с подстановками
2. Добавить новые токены для нахождения параметров
3. Реализовать разбор нового выражения
4. Написать метод десериализации для разных типов данных
5. Обработать синтаксическое дерево разбора для подстановки значения

Обработка подстановок

- считывание и парсинг аргументов
- сохранение в словарь, где ключ — название подстановки
- создание нового типа токенов
- обработка символов
- разбор выражения с параметром в запросе:
 - считывание имени и типа, удовлетворяющих выражению {Name:Type}

Обработка запроса

- создание нового класса элемента синтаксического дерева
- во время разбора выражения сохранение внутри класса данных подстановки
- обход в глубину построенного дерева по запросу для поиска этого элемента дерева

Обработка запроса

- нахождение элемента в дереве с нужным типом
- получение из словаря строкового значения подстановки
- создание класса по типу подстановки для дальнейшей десериализации строкового значения
- замена элемента дерева на новый с полем, заполненным данными подстановки со значением переданного типа

Десериализация данных

- добавление нового метода чтения строк без экранирования
- проверка наличия мусорных символов
- реализация для разных типов данных

Тестирование

Функциональные тесты

- проверка корректности обработки prepared statements

Поддерживаемые типы данных:

- строки
- числа
- даты, даты с временем
- Nullable(T) (содержит значения типа или NULL)
- кортежи
- массивы

Демонстрация примеров из теста

```
$CLICKHOUSE_CLIENT --param aui="[1, 2]" \  
-q "SELECT t FROM ps WHERE a = {aui:Array(UInt16)}";  
  
$CLICKHOUSE_CLIENT --param d a="[[1, 1], [2, 2]]" \  
-q "SELECT dt FROM ps WHERE da = {d_a:Array(Array(UInt8))}";  
  
$CLICKHOUSE_CLIENT --param tisd="(10, 'Test')" \  
-q "SELECT a FROM ps WHERE t = {tisd:Tuple(Int16, String)}";  
  
$CLICKHOUSE_CLIENT --param d t="(10, ('dt', 10))" \  
-q "SELECT da FROM ps WHERE dt = {d_t:Tuple(UInt8, Tuple(String, UInt8))}";  
  
$CLICKHOUSE_CLIENT --param nd="2015-02-15" \  
-q "SELECT * FROM ps WHERE n = {nd:Nullable(Date)}";
```

Результаты работы

- Проведено изучение архитектуры системы
- Разработана концепция передачи параметров
- Реализованы методы для обработки prepared statements
- Написаны тесты для проверки работоспособности

Спасибо за внимание!