

Поддержка строк переменной длины в генераторе баз данных Mill-DB

Орлов Владислав

Руководитель: Коновалов А. В.

Цель

Целью данной курсовой работы является расширение возможностей MillDB, с помощью введение нового типа данных **Text**, в котором возможно было бы хранить любое произвольное количество символов.

Этапы:

- 1) Изучение архитектуры и исходного кода генератора баз данных.
- 2) Расширение синтаксиса генератора: лексический анализ, синтаксический анализ, семантический анализ.
- 3) Изменение формата файла базы данных для поддержки строк переменной длины.
- 4) Тестирование

Описание Mill-DB

- На вход - спецификация на собственном диалекте SQL, описывающая набор таблиц и процедуры чтения и записи
- программа генерирует исходный код .c и заголовочный файл .h
- в сгенерированных файлах - методы и структуры для записи, чтения и хранения данных
- поддерживает только чтение и запись, но не модификацию или удаление

Пример конфигурации SQL

```
CREATE TABLE Example ( id int pk, description text));
```

```
CREATE PROCEDURE add_example (@id int in, @desc text in)
```

```
BEGIN
```

```
    INSERT TABLE Example VALUES (@id, @desc);
```

```
END;
```

```
CREATE PROCEDURE get_example(@id int in, @desc text out)
```

```
BEGIN
```

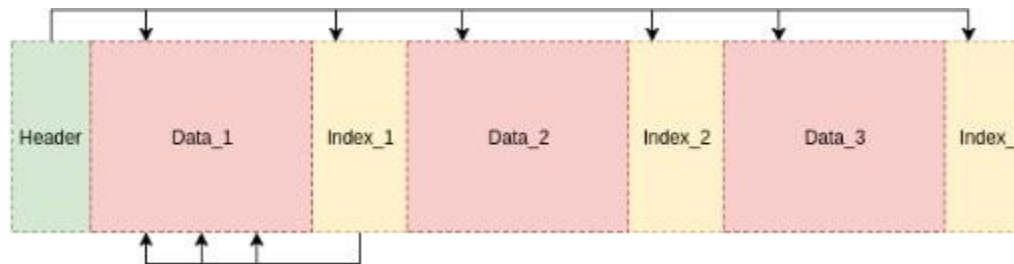
```
    SELECT description SET @desc FROM Example WHERE id=@id;
```

```
END;
```

Пример использования

```
int main() {  
    demo_open_write("FILE");  
    add_example(1, 'This is example');  
    demo_close_write();  
    struct demo_handle* handle = demo_open_read("FILE");  
    struct get_example_out iter;  
    get_example_init(&iter, handle, 1);  
    while (get_person_next(&iter))  
        printf("%s\n", iter.data.desc);  
    demo_close_read();  
}
```

Хранение данных



Header: количество записей и сдвиги данных и индексов

Data: хранимые данные для определенной таблицы

Index: узлы индексного дерева, узел включает информацию о первичном ключе и смещении в разделе данных

Изменения в проекте

- Изменение лексера и парсера, добавление TEXT_KEYWORD
- Исправление бага множественной выборки
- Изменение чтения данных: от постраничного к поточному
- Если таблица содержит колонку с типом 'text', нестандартная запись и чтение:

```
CREATE TABLE example (  
    id int pk,  
    description text,  
    value int  
);
```

```
write ID type INT  
write DESCRIPTION_LEN type UINT16  
write VALUE type INT  
write DESCRIPTION type CHAR[]
```

Тестирование: SQL

```
CREATE SEQUENCE Pet_sequence;
CREATE TABLE owner (
oid int pk,
oname text,
pet_id int
);
CREATE TABLE pet (
pid int pk,
pname text
);
CREATE PROCEDURE add_owner_pet(@oid int in, @name text in, @pname text in)
BEGIN
INSERT TABLE owner VALUES (@oid, @name, NEXTVAL(Pet_sequence));
INSERT TABLE pet VALUES (CURRVAL(Pet_sequence),@pname);
END;
CREATE PROCEDURE get_pet_by_pid(@id int in, @out text out)
BEGIN
SELECT oname SET @out FROM owner WHERE oid=@id;
SELECT pname SET @out FROM pet WHERE pid=@id;
END;
```


Тестирование: программа

```
#include <stdio.h>
#include <time.h>
#include "pet_by_owner.h"
int main() {
    int owners[6] = {5, 2, 1, 4, 3, 6};
    pet_by_owner_open_write("FILE2");
    add_owner_pet(1, "Vova", "Ni");
    add_owner_pet(2, "Petya", "Chi");
    add_owner_pet(3, "Maria", "Shuia");
    add_owner_pet(4, "Sasha", "Kawai");
    add_owner_pet(5, "Natal", "Kazem");
    add_owner_pet(6, "Tanya", "Charl");
    pet_by_owner_close_write();
    struct pet_by_owner_handle *handle1 = pet_by_owner_open_read("FILE2");
    struct get_pet_by_pid_out iter1;
    int i;
```

```
    for (i = 0; i < 6; i++) {
        printf("%d\n", owners[i]);
        get_pet_by_pid_init(&iter1, handle1, owners[i]);
        int flag = 1;
        while (get_pet_by_pid_next(&iter1)) {
            printf("%s\n", iter1.data.out);
            flag = 0;
        }
        if (flag) {
            printf("\n");
        }
    }
    pet_by_owner_close_read(handle1);
    return 0;
}
```

Тестирование: вывод

5

Natal

Kazem

2

Petya

Chi

1

Vova

Ni

4

Sasha

Kawai

3

Maria

Shuia

6

Tanya

Char

Заключение

- Исследованы технологии flex+bison, хранение информации в СУБД, индексы и их хранение в виде В-дерева
- Расширена грамматика
- Изменен способ считывания данных
- Добавлен новый тип 'text', хранящий строки переменного размера

Благодарю за внимание!