Описание задания

Реализовать однонаправленный список структур следующего вида:

```
class record
{
  char * name;
  int  phone;
  int  group;
public:
  // ...
};
```

Длина строки в поле name неизвестна, и память под нее выделяется динамически.

Программа должна понимать следующие управляющие последовательности.

1. quit; – завершить работу
2. insert (<name>, <phone>, <group>); – добавить структуру с указанными полями в список;
3. select <аргументы> – вывести элементы списка, удовлетворяющие указанным в команде условиям, и в указанном в команде виде. Аргументы могут быть:
   (a) задающие набор выводимых столбцов (обязательный аргумент):

   <список полей> – выводить указанные столбцы в указанном порядке, например, group, name – выводить только третье и первое поля (в этом порядке); в качестве списка может быть * – выводить все столбцы, эквивалентен name, phone, group

   (b) задающие критерий, которому должны удовлетворять выводимые строки (необязательный аргумент):

   where <условия> – выводить строки, удовлетворяющие указанным условиям. Условия могут быть:

   - <поле> <оператор> <выражение>, где
     - <поле> – имя поля таблицы (name, phone, value)
     - <оператор> – логический оператор отношения: = – равно, <> – не равно, <, >, <=, >= – соответствуют языку C
     - <выражение> – константное выражение соответствующего типа

     Пример: where name = Ivanov
   - <поле> like <образец>, где
     - <поле> – имя поля таблицы символьного типа (name)
     - <образец> – образец поиска. Может включать в себя специальные символы:

* `%` – соответствует 0 или более любым символам
* `_` – соответствует 1 любому символу

Пример: `where name like Iv%`

Из двух таких условий можно строить более сложные:

- <условие1> and <условие2>
- <условие1> or <условие2>

На условия здесь накладывается ограничение: <условие1> и <условие2> задают условия на разные поля записи.

Пример команд:

```
select group, name where phone = 1234567 and name = Student;
select * where phone >= 1234567 and name like St%;
select * where group = 208 and phone <> 1234567;
select * where name = Student or phone = 1234567;
```

4. delete <аргументы> – удалить строки таблицы, удовлетворяющие указанным в команде условиям. Аргументы могут быть:
   (a) нет аргументов – удалить все строки таблицы
   (b) where <условия> – удалить строки, удовлетворяющие указанным условиям (см. описание условий в команде select).

Разделителем команд является “;”, разделителями аргументов команды являются пробел, символ табуляции и символ новой строки.

Программа после запуска загружает структуру из файла с фиксированным именем (`a.txt`), где находятся записи вида

<name> <phone> <group>

(по одной записи на строку), выполняя для каждой такой записи команду

insert (<name>, <phone>, <group>);

Затем программа начинает интерпретировать команды со своего стандартного ввода и выдавать результаты на стандартный вывод. Программа завершает работу, встретив команду quit;

Дополнительные параметры запроса 1

В запросе select можно задавать дополнительные аргументы, управляющие выводом:

select [<дополнительные аргументы>] <аргументы>;

Список дополнительных аргументов:

1. avg(<поле>)

   Выводит среднее значение указанного поля для всех найденных записей числового типа. Например, select avg (phone) where name = Student; выводит среднее значение поля phone для всех найденных записей.

2. sum(<поле>)

   Выводит сумму значений указанного поля для всех найденных записей числового типа. Например, select sum (phone) where name = Student; выводит сумму значений поля phone для всех найденных записей.

3. count(<поле>)

   Выводит количество найденных полей с заданным (ненулевым) значением указанного поля. Например, select count (name) where group = 208; выводит количество найденных записей с ненулевым полем name.

4. min(<поле>)

   Выводит минимальное значение указанного поля для всех найденных записей. Например, select min (name) where group = 208; выводит минимальное значение поля name для всех найденных записей.

5. max(<поле>)

   Выводит максимальное значение указанного поля для всех найденных записей. Например, select max (name) where group = 208; выводит максимальное значение поля name для всех найденных записей.

6. top N

   Выводит первые N найденных записей. Например, select top 3 where phone = 1234567; выводит первые 3 найденные записи.

7. top N percent

   Выводит первые N% найденных записей. Например, select top 10 percent where phone = 1234567; выводит первые 10% найденных записей.

8. distinct <поле>

   Выводит только найденные записи с различным значением указанного поля. Например, select distinct name where phone = 1234567; выводит найденные записи с различным значением поля name.

Дополнительные параметры запроса 2

В запросе select можно задавать дополнительные аргументы, управляющие выводом:

select <аргументы> [<дополнительные аргументы>];

Список дополнительных аргументов:

9. limit N Выводит первые N найденных записей. Например, select * where phone = 1234567 limit 3; выводит первые 3 найденные записи.

10. order by <поле>

    Выводит найденные записи, отсортированные по значению указанного поля. Например, select * where phone = 1234567 order by name; выводит найденные записи отсортированными по значению поля name.

Комбинации параметров

В запросе select можно задавать дополнительные аргументы, управляющие выводом:

select [<доп. аргументы 1>] <аргументы> [<доп. аргументы 2>];

Список дополнительных аргументов:

11. top N + order by <поле>

    Выводит первые N найденных записей, отсортированные по значению указанного поля. Например, select top 3 where phone = 1234567 order by name; выводит 3 первые найденные записи отсортированными по значению поля name.

12. top N percent + order by <поле>

    Выводит первые N% найденных записей, отсортированные по значению указанного поля. Например, select top 10 percent where phone = 1234567 order by name; выводит 10% первых найденных записей отсортированными по значению поля name.