УДК 004.658

АВТОМАТИЗАЦИЯ ПРОЦЕССА СОГЛАСОВАНИЯ ДАННЫХ В ГЕТЕРОГЕННЫХ РАСПРЕДЕЛЕННЫХ БАЗАХ ДАННЫХ

Чабанов О.В.

Науковий керівник - доц. каф. «Системне програмне забезпечення», канд. техн. наук Блажко О.А.

Существует система тиражирование операций согласования в гетерогенных распределенных базах данных, недостатком работы, которой является алгоритм согласования. Этот алгоритм предполагает согласование данных лишь один к одному, т.е. не учитывается то, что информация может быть искажена при вводе.

Одним из известных способов сравнения 2х строк являются расстояния Левенштейна и равенство отображений “SOUNDEX” [2]. Эти алгоритмы позволяют сравнивать длины и звучание строк соответсвенно,а не посимвольно и к тому же имеют свои недостатки.

Поскольку проблемой является согласование данных в ручном режиме, цель данной работы – разработка алгорима автоматизированного согласования данных.

Решение. Для решения поставленной задачи необходимо усовершенствовать имеющуюся ИС [1], дополнив ее новыми структурами данных. Представим дополнительные структуры данных в виде тройки

<SM, RS, D>

где *SM* (*Schema Maching*) – множество атрибутов таблиц, которые будут сравниваться с использованием преобразующих функций;

RS (*Replace Symbol*) – множество гласных букв русского и украинского алфавитов, которые чаще всего может спутать оператор при вводе информации, и замена на их русский либо украинский эквивалент;

D (*Domens*) – множество значения атрибутов всех таблиц, которые имеют строковый тип и состоящие из нескольких слов, разделенных знаками «пробел» или «-».

Элемент $sm \in SM$ представим в виде двойки $< a1, a2, AM >$, где $a1 \in R$ $a2 \in R$ – атрибуты БД-подписчика; *AM* – упорядоченное множество преобразований с элементами $F(pa)$, $pa \in R$ – атрибут ЛБД-издателя, *F* – функция преобразования атрибута.

Элемент $rs \in RS$ представим в виде двойки $<s\_in, s\_out>$, где $s\_in \in S$ и $s\_out \in S$ – атрибуты таблицы, значения которых соответствуют гласным буквам русского и украинского алфавитов.

Элемент $d \in D$ представим в виде тройки <*tn, an, val*>, где $tn \in TN$ – имя таблицы, $an \in AN$ – атрибут таблицы *tn, val* – значение атрибута *an.*

Прокомментируем представленный код:

Db_1, Db_2 – множество дубликатов таблиц T_1 и T_2 до выполнения замены;

```
Db1:= getDoublets(R1);
Db2:= getDoublets(R2);
for c:=1 to |F| do{
 R1new:= {0};
 R2new:= {0};
 for i:=1 to |R1| do{
        for j:=1 to ri do{
               x:= replace(ri.ai, fc);
               R1new:= R1new ∪ x;
        }
 }
 for i:=1 to |R2| do{
        for j:=1 to ri do{
               x:= replace(ri.ai, fc);
               R2new:= R2new ∪ x;
        }
 }
 Da1:= getDoublets(R1new);
 Da2:= getDoublets(R2new);
 if(|Db1|=|Da1| && |Db2|=|Da2|){
 for i:=1 to |R1new| do
 for j:=1 to |rinew| do
 for k:=1 to |R2new| do
 for l:=1 to |rknew| do
 if(rinew.ai=rknew.al)then{
        x:= rinew +rknew;
        R:= R ∪ x;
        R1:=R1 ∩ rinew;
        R2:=R2 ∩ rknew;
        Db1:= getDoublets(R1);
        Db2:= getDoublets(R2);
 }
 }
}
```

Da_1, Da_2 – множество дубликатов таблиц *T1* и *T2* после выполнения замены;

R_1new, R_2new – сформированные множества, включающие в себя измененные значения множества R_1 и R_2 соответственно;

F – множество гласных букв, которые встречаются в значениях атрибутов таблиц T_1 и T_2, $f_c \in F$ – *c*-й картеж F;

|<множество>| - размерность множества;

getDoublets(<множество>) - функция получения количества дубликатов для множества.

Эксперимент. Для проведения эксперимента были взяты таблицы из разных источников: таблица успеваемости студентов деканата, таблица списка студентов отдела кадров и таблица со списком студентов, проживающих в общежитии. Согласовывались строки, при совпадении групп и фамилий студентов. Нами предложены следующие правила замен RS={<’i’,’и’>,<’ь’,’ ’>,<’е’,’є’>,<’ы’,’и’>,<’’’,’`’>}. На основании этих правил из множества не совпавших строк получили 49% записей, которые можно согласовать. Остальные строки согласовать не удалось из-за ошибок, определяющих разный порядок следования букв в словах.

Вывод. Проведя эксперименты с алгоритмом Левенштейна, было установлено, что при увеличении значения расстояния сравниваемых строк возникает ошибочное согласование. Используя множество не совпавших данных в зависимости от значения расстояния Левенштейна, процент возникновения таких ошибок: при lv<4 – 0%, lv<5 – 1.2%, lv<6 – 2,3%, lv<7 – 7%. Ошибочное согласование связано с совпадением разных фамилий с одинаковыми длинами.

Проведенные эксперименты показывают, что дальнейшее повышение процента согласованных строк возможно лишь при создании алгоритма, совмещающего свойства нами предложенного алгоритма и алгоритма Левенштейна.

СПИСОК ЛІТЕРАТУРИ

1. *Джайяб Т.Д. Альсаффаді, О.А.Блажко* Підтримка гетерогенних розподілених баз даних з тиражуванням // The Procs. of International Conferences on Computer Science and Information Technologies. – Lviv, Ukraine, 2007. – С. 237–240.

2. Чухрай Андрій Григорович. Методи та засоби підвищення якості даних в автоматизованих системах організаційного управління: дисертація канд. техн. наук: 05.13.06 / Національний аерокосмічний ун-т ім. М.Є.Жуковського;Харківський авіаційний ін-т;. - Х., 2003. – С. 7– 9.