

ИНДЕКС 3649

Препринт ЕФИ-1218(4)-90

ԵՐԵՎԱՆԻ ՖԻԶԻԿԱՅԻ ԻՆՍՏԻՏՈՒՏ
ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ
YEREVAN PHYSICS INSTITUTE

С.Х. АРУТЮНЯН

ЯЗЫК РЕЛЯЦИОННОГО ИСЧИСЛЕНИЯ «ANI»



ЕРЕВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ

ЦНИИатоминформ
ЕРЕВАН-1990

Ա.Ք.ՀԱՐՈՒԹՅԱՆՆՅԱՆ

«ԱՆԻ» ՌԵԼՅԱՑԻՈՆ ՀԱՇՎԱՐԿԻ ԼԵՁՈՒՆ

Նկարագրված է «ԱՆԻ» ռելյացիոն հաշվարկի հրամանները «ԱՆԻ» համարակարգի համար: Լեզուն նախատեսված է ֆիզիկական փորձից ստացված տղվայունների հավաքման, պահպանման, փնտրման և ընտրության համար, ինչպես նաև կարող է կիրառվել գիտության այլ բնագավառներում: Այլ նման լեզուներից հիմնական տարբերությունն այն է, որ այստեղ հարաբերության տակ ռասկայվում է, ընդհանուր դեպքում, եռաչափ աղյուսակ (N-դոմեններով), իրի համար կիրառվում են հաշվարկների գործողությունները:

Երևանի ֆիզիկայի ինստիտուտ
Երևան 1990

Введение

Концепция баз данных уже давно стала определяющим фактором при создании эффективных систем автоматизированной обработки информации.

Существует множество банков данных, построенных на той или иной модели и созданных для реализации конкретных задач.

Особенной простотой, практической полезностью от всех других моделей данных выгодно отличается реляционная модель, которая была разработана Коддом Э.Ф. в 1970 г. и с тех пор стала бурно развиваться.

Известно, что для реализации той или иной модели используются языки описания данных (ЯОД) и языки манипулирования данными (ЯМД).

ЯОД дает возможность описать логические структуры данных в БД, отношения между элементами данных. ЯОД может использоваться и для описания внешней модели данных.

ЯМД дает возможность осуществлять доступ к данным, выполнять над ними определенные операции.

Все ЯМД в реляционных базах данных можно разделить на два главных класса: языки реляционной алгебры, языки реляционного исчисления.

В качестве примера языка реляционного исчисления можно привести язык "ALPHA" , основанный Коддом Э.Ф., другие, как например, "QUEL" , "SQL" , "PIQUE" и т.д.

Основной идеей создания языка "ANI" явилось то, что он может быть использован для сбора и обработки информации в физических экспериментах. Вместе с тем, его можно эффективно использовать в библиографии, экономике и других сферах деятельности человека.

В "ANI" , как и в вышеуказанных языках, используются фундаментальные понятия теории реляционных баз данных: отношения, атрибуты, домены, кортежи.

Создание еще одного языка реляционного исчисления "ANI" было обусловлено следующими причинами.

При наличии множества отношений, имеющих одну и ту же структуру (домены имеют одинаковые имена, одинаковые типы, ширины строк и т.д.), можно было создать либо одну очень большую двухмерную плоскую таблицу, либо каждое из этих отношений в отдельности описать и использовать как разные отношения.

Для первого случая возникают сложности с представлением такой таблицы в памяти, выборкой, сравнением этой таблицы с другими. Во втором случае число описаний отношений резко возрастает.

Рассмотрение множества подобных отношений в виде трехмерных таблиц снимает следующие проблемы:

1. Вышеописанные множества двухмерных таблиц представляются как одно отношение в виде трехмерной таблицы, и описывается такое отношение один раз;

2. С вводом понятия ряда (3-й параметр таблицы) появляется возможность сравнения разных рядов одного и того же отношения;

3. Во время эксперимента однотипные таблицы записываются одной командой, используя команду цикла, которая изменяет лишь номер ряда;

4. Есть возможность изменения во время поиска (команда SEARCH) номеров рядов для каждого домена самостоятельно (в режиме цикла);

5. Есть возможность сравнения или поиска нескольких таких отношений (их рядов) друг с другом;

6. Если имеется несколько баз данных, то поиск, выборка и т.д. между ними организована также в вышеуказанном режиме;

7. Число строк каждого ряда отношения является числом произвольным;

8. Запросы можно давать на русском, армянском, английском языках (важно, чтобы в системе, в рамках которой работает язык, и на вводных и выводных устройствах были введены буквы и их коды соответствующих естественных языков).

Применение языка "ANI" даст возможность в режиме "on line" организовать ввод поступающих с эксперимента данных, сформировать определенным образом их выборку, поиск, обработку.

Опишем команды языка "ANI" .

2. Описание команд языка "ANI"

Часть команд языка "ANI" предназначена для описания отношений, их атрибутов, типов и т.д., а другая часть - для поиска выборки, соединения, исключения и т.д. При описании каждой команды будут даны ссылки на примеры. Все команды завершаются символом %.

1. ATRIBU (NAM, 0 : A1 : A2 ... AN) - описывает новое отношение с именем NAM, именами атрибутов A1, A2, ... AN (пример на рис.8).

ATRIBU <идентификатор отношения> : <идентификатор атрибута>
{<идентификатор атрибута>}

<идентификатор отношения> ::= <имя отношения>, <номер ряда отношения>

<имя отношения> ::= <идентификатор>

<номер ряда отношения> :- <число> - номер ряда трехмерной таблицы отношения NAM

<идентификатор> ::= {буква} | <идентификатор> <цифра> |
<идентификатор> {<цифра>}

Число символов в идентификаторе не более шести.

<буква> ::= . A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
X | Y | Z .

А также все буквы русского и армянского алфавитов.

<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<число> ::= <последовательность цифр>

<последовательность цифр> ::= {цифра}

<идентификатор атрибута> ::= <идентификатор> .

2. LENGTH (NAM, 0 : I : 2 : I : ... : 7 : 3) - описывает ширину строк всех доменов отношения NAM. Под шириной подразумевается число слов, образующих данную строку и имеющих атомарное значение (пример на рис.8).

LENGTH (<идентификатор отношения> : <ширина> { : <ширина> })
<ширина> ::= =число слов в строке

3. TIP (NAM, 0 : R : I : ... : I : T) - описывает типы всех атрибутов отношения NAM (рис.8).

TIP (<идентификатор отношения> : <тип> { : <тип> })

<TIP> ::= I | R | T | D

I - целая константа

R - вещественная - - - -

T - текстовая

D - вещественная с двойной точностью

4. SS (NAM, 0 : BB < 20.4 & NAM, 0 : BB ≠ 100.)

- описывает условие ограничения целостности, которое накладывается на атрибут BB отношения NAM. SS (логическое выражение)

<логическое выражение> ::= <безусловно булево выражение> <⊗>

<безусловно булево выражение> ::= { ⊗ <безусловно булево выражение> } | ⊗ <безусловно булево выражение>

<безусловно булево выражение> ::= <безусловно арифметическое выражение> <сравнение> <безусловно арифметическое выражение>

<безусловно арифметическое выражение> ::= формула, построенная из чисел, идентификатора отношения и его идентификаторов атрибутов, арифметических операций + | - | * | / | %, круглых скобок и элементарных функций SUMM | MAXC | MINI

<сравнение> ::= < | < | > | > | = | ≠

<⊗> ::= ⊗ | & | ∨

5. CIPHER (ISK) - описывает шифр ISK пользователя, его пароль (рис.8).

CIPHER (шифр)

< шифр > :: = идентификатор.

6. STEP A(I;2,...4;I) - описывает шаги и их ограничение сверху для каждого отношения, домена, участвующих в запросе (рис.8).
STEPS (< шаг I > : < ограничение I > { , < шаг I > : < ограничение I > })

< шаг I > :: = целая константа

< ограничение I > :: = целая константа

7. STEPB (2:100) - описывает шаг цикла и его ограничитель, равный двум и 100, который применяется для всех отношений (доменов), участвующих в запросе.

STEPS (< шаг I > : < огр I >).

8. EQU (NAM 1 ; NAM 2) - объявляет отношение с именем NAM 2 копией отношения NAM 1. Но новое отношение NAM 2 записывается в рабочую область СУБД, в базу данных не записывается.

EQU (< идентификатор отношения > ; < идентификатор отношения >)

9. SEARCH (A, 1 : BV ; C, 1 : DD) WHERE A, I : BV = "вероятность" & C, 1 : DD < I7,5 %.

Осуществляет поиск и выборку кортежей отношений или строк доменов отношений при выполнении некоторого условия. В нашем примере осуществляется выборка строк доменов A, 1 : BV и C, 1 : DD (ряд I-й отношения A и ряд I-й отношения C), которые удовлетворяют условиям, изображенным на рис.1.

A, I : BV = "вероятность" & C, I : DD < I7,5

Рис.1

SEARCH (< список >) WHERE < условие >
< список > :: = { < идентификатор отношения > : < ALL > ; < идентификатор отношения > : < идентификатор атрибута > ; | < переменная > = < безусловно арифметическое выражение > | < переменная > : = < имя функций > (< безусловно арифметическое выражение >) } ; < список > ; < список > }
< имя функций > :: = SUMM | MAXC | MINI - вычисляют сумму всех элементов множества, максимальный элемент множества, минимальный элемент множества.

< ALL :: > = все атрибуты данного отношения

< переменная > :: = идентификатор

< условие > :: = < логическое выражение >

10. SEARCH (< список >) WHERE < условие > *)

Осуществляет выборку строк доменов (отношений) из списка, если условие выполняется (TRUE,).

11. SEARCH (< список >) WHERE < условие > - осуществляет выборку всех строк доменов отношений из списка, если условие TRUE.

12. DELETE (A) - исключить отношение A (всю трехмерную таблицу)

DELETE (< имя отношения >).

*) Условие может быть опущено при желании, т.е. SEARCH (< список >). То же для команды SEARCH.

13. DELETE (A,2:ALL) -исключить отношение A(второй ряд трехмерной таблицы).

DELETE (< идентификатор отношения > : < ALL >)

14. RENAME (A; C)-переименовать отношение A на C.

RENAME (< имя отношения > ; < имя отношения >)

15. RENAM1 (A,0:AI:A2)-переименовать имя атрибута AI отношения A, на имя A2.

RENAM1 (< идентификатор отношения > : < идентификатор атрибута > : < идентификатор атрибута >)

16. UNITED (A,1:ALL;B,1:ALL;C,1:ALL) WHERE A,1=BB-E,1=BB' -осуществляет соединение отношений A и B в новое отношение C при выполнении условия, а именно: наличие общих атрибутов в отношении A и B.

UNITED (< идентификатор отношения > : ALL ; < идентификатор отношения > : < ALL > ; < идентификатор отношения > : < ALL >)
WHERE < условие соединения >

< условие соединения > ::= < идентификатор отношения > : < идентификатор атрибута > < θ > < идентификатор отношения > : < идентификатор атрибута >

< θ > ::= < | < | > | > | = | ≠

17. WRITE (A,I: ALL)-осуществляет запись данных отношения A,I, передаваемых пользователем.

WRITE (< идентификатор отношения > : < ALL >) .

Приведем пример работы на языке "ANI" . Допустим, что реляционная СУБД содержит базу данных о некоторой организации, которая разрабатывает программы для других организаций [1] .

В этой базе существуют два отношения:

ОТДЕЛ (номер отдела, название отдела, число программистов, число терминалов).

ПРОГРАММА (номер отдела, код программы, идентификатор, размер, язык, операционная система).

То, что находится в скобках, - это названия атрибутов.

Вид запроса: получить название всех отделов, которые разрабатывают программы на языке "ПАСКАЛЬ" для операционной системы ОС ЕС.

На языке "ANI" запрос выглядит в следующем виде (рис.2).

```
SEARCH (отдел,I:название отдела) WHERE отдел,I: номер от-
дела=программа, I:номер отдела & программа, I:язык="пас-
каль" & программа, I:операционная система="ос ес"%
```

Рис.2

Индекс I означает, что рассматривается первый ряд трехмерной таблицы отношений ОТДЕЛ и ПРОГРАММА.

Если трехмерная таблица имеет всего один ряд, то она рассматривается как двухмерная таблица отношения.

Пусть в результате некоторого эксперимента через определенные промежутки времени получаются таблицы (отношения) ALPHA и BETA с атрибутами A1, A2, A3, A4, A5 и B1, B2, B3, B4 (рис.3)

Мы имеем две трехмерные таблицы отношений ALPHA и BETA .

Для того, чтобы записать запросы для таких таблиц на языках "ALPHA" , "RQUEL" и т.д. , необходимо прежде всего как-то назвать, описать все ряды от I-ого до 10⁶-ого своими именами, так как каждая двухмерная таблица -это самостоятельное отношение.

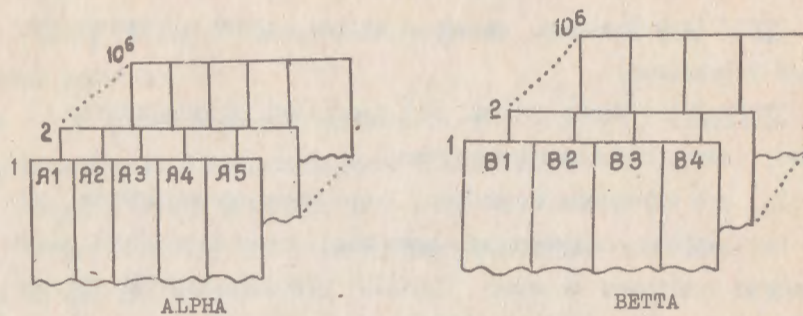


Рис.3 Все таблицы ALPHA и BETA, полученные во время эксперимента

Для некоторых языков описание всех имен отношений, участвующих в запросе, или описание всех смысловых множеств очень трудно. Это очень усложняет вид запроса, а для некоторых языков делает его практически невозможным.

Так как возможность рассмотрения всех двумерных таблиц в этих отношениях (ALPHA и BETA) велика, а именно $2 \cdot 10^6$, то естественно возникает вопрос об организации цикла над этими запросами.

На языке "ANI" это можно организовать следующим образом.

Допустим, нас интересует ряд 2-ой отношения ALPHA (рис.3) и 1000-ый ряд отношения BETA.

Запрос: Найти все строки доменов A2, A3 отношения ALPHA, если в домене A5 отношения ALPHA встречается часть "электрон", а в таблице BETA значение домена B4 должно быть меньше 4.5.

На языке "ANI" запрос выглядит следующим образом (рис.4)

```
SEARCH (ALPHA, 2: A2; ALPHA, 2: A3; BETA, 1000: B4)
WHERE ALPHA, 2: A5=" электрон" & BETA, 1000: B4 < 4.5%
```

Рис.4

Пусть этот запрос необходимо выполнить в цикле, начиная с ряда 2-ого таблицы ALPHA с шагом 2 и ряда 1000 таблицы BETA с шагом 1 (без верхнего ограничителя ряда).

Тогда запрос на языке "ANI" выглядит следующим образом:

```
STEPS(2:0; 2:0; 1:0; 2:0; 1:0)%
SEARCH (ALPHA, 2: A2; ALPHA, 2: A3; BETA, 1000: B4) WHERE ALPHA,
2: A5 = "электрон" & BETA, 1000: B4 < 4.5%
```

Рис.5

В результате будут рассмотрены следующие ряды отношений ALPHA и BETA (рис.6):

ALPHA	BETA	
2	1000	} указатели номеров рядов
4	1001	
.	.	
.	.	
10^6	10^3	

Рис.6

Как только во время работы цикла номер рассматриваемого ряда выйдет за количество существующих рядов или указателя ограничителя рядов хотя бы для одного отношения ALPHA или BETA, этот процесс остановится.

Если шаги циклов для каждого отношения, участвующего в запросе, равны, то используется команда STEPB

Пусть с эксперимента поступают разные таблицы: ALPHA, BETA, GAMMA, NNR. Допустим, что ALPHA и BETA образуют одну базу данных, а GAMMA и NNR — другую (рис.7)

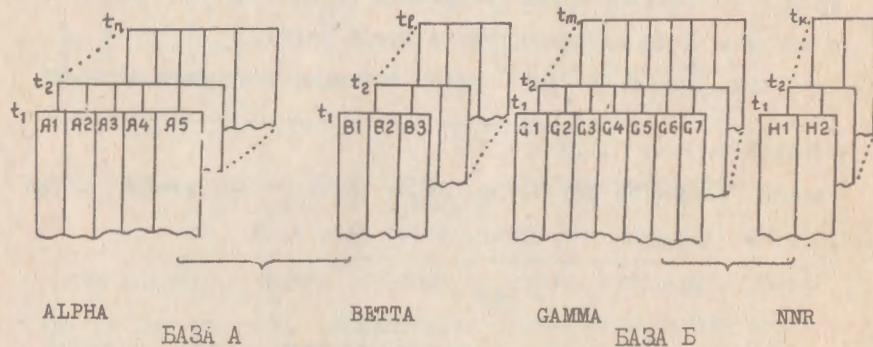


Рис.7 Пример организации базы данных А и В

```

CIPHER (ISS)%
ATRIBU (ALPHA, 0: A1:A2:A3:A4:A5)%
ATRIBU (BETA, 0: B1:B2:B3)%
LENGTH (ALPHA, 0: 1:4:1:1:1)%
LENGTH (BETA, 0: 1:5:2)%
TIP (BETA, 0: I: I:T: T: T)%
TIP (ALPHA, 0: I: I: T)%
STEPB(1:1000000)%
WRITE(ALPHA;1:ALL)%
STEPB(1:20000)%
WRITE (BETA. 1: ALL)%
    
```

Рис.8

В результате действия этих команд все таблицы типа ALPHA и BETA будут записаны в базу данных. То же можно написать для GAMMA и NNR.

Пусть теперь необходимо организовать поиск данных, используя данные двух баз данных (число отношений из разных баз данных, участвующих в запросе, не ограничено).

```

CIPHER (ISS)%
STEPB (1:2000)%
SEARCH(ALPHA, 1:A3;S=MAXC (BETA, 1:B3 x5))WHERE
GAMMA, 1:G7="электрон" & NNR, 1:N2 <= 20V (ALPHA,1:A1
= 4.5)%
    
```

Рис.9

В результате работы этих команд в режиме цикла будут просмотрены ряды с I-ого по 2000-ый отношений ALPHA, BETA, GAMMA, NNR, и для каждого ряда будет проверено условие, а именно: если столбец с именем G7 отношения GAMMA содержит название "электрон" и столбец с именем N2 отношения NNR меньше 20, то необходимо передать пользователю содержимое домена A3 отношения ALPHA, а также переменную с именем S, в которой находится максимальное значение всех элементов домена B3 отношения BETA, умноженного на 5. Или извлечь ту же информацию при выполнении условия, что домен A1 отношения ALPHA равно 4,5 (но это может быть другая выборка строк).

СПИСОК ЛИТЕРАТУРЫ

1. Костин А.Е., Шаньгин В.Ф. Организация и обработка структур данных в вычислительных системах. М.: Высшая школа, 1967.
2. Мейер Д. Теория реляционных баз данных. М.: Мир, 1987 .

Рукопись поступила 9 декабря 1969 г.

С.Х. АРУТКНЯН

ЯЗЫК РЕЛЯЦИОННОГО ИСЧИСЛЕНИЯ " АЛГІ "

Редактор Л.П. Мукаян

Технический редактор А.С. Абрамян

Подписано в печать 16/IV-90г. ВФ-01337 Формат 60x64/16

Офсетная печать. Уч.изд.л.1,0 Тираж 299 экз. Ц.15 к.

Зак.тип.№ 177

Индекс 3649

Отпечатано в Ереванском физическом институте

Ереван 36, ул.Братьев Алиханян 2