

ЕРВАНСКИЙ ФИЗИЧЕСКИЙ ИНСТИТУТ

НА ПРАВАХ РУКОПИСИ

АРУТЮНЯН СУСАННА ХРИСТОФОРОВНА

АВТОМАТИЗАЦИЯ СБОРА И ОБРАБОТКИ ДАННЫХ С МНОГОКАНАЛЬНЫХ
ЭКСПЕРИМЕНТАЛЬНЫХ УСТАНОВОК ДЛЯ ИССЛЕДОВАНИЙ В ОБЛАСТИ
ФИЗИКИ КОСМИЧЕСКИХ ЛУЧЕЙ .

ОБЪЕКТ ПРИМЕНЕНИЯ ЭНЦИКЛОПЕДИИ ТЕХНИКИ .

МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ И МАТЕМАТИЧЕСКИХ
МЕТОДОВ В НАУЧНЫХ ИССЛЕДОВАНИЯХ

АВТОРЕФЕРАТ

ДИССЕРТАЦИИ НА СОИСКАНИЕ УЧЕНОЙ СТЕПЕНИ КАНДИДАТА
ФИЗИКО-МАТЕМАТИЧЕСКИХ НАУК

ЕРЕВАН-1991г.

кратия данных . При передаче больших объемов информации в МД в оперативную память, существенно увеличивается время обмена , передачи данных (сравнительно медленной по сравнению с оперативной памятью) .

Индексированная упаковка , сжатие данных перед обменом позволяет увеличить объемы передаваемой числовой информации и получить экономию во времени . Для улучшения скорости поиска , также можно выделить и обработать ту часть информации , которая необходима . Как правило , при запросе извлекается только необходимая часть хранимой информации . Таким образом , скорость поиска зависит от количества извлекаемой информации , от времени доступа и метода доступа к внешней памяти .

Но вопросы , связанные с ограничением числа параметров при поиске информации , а также отсутствием произвольного поиска по сложным запросам поставили вопрос о создании , разработке более совершенной системы , реализующей эти требования .

Автором была разработана и практически апробирована новая система реляционных баз данных AVIS , которая в целом решает эти проблемы .

Цель работы . Целью работы была разработка и применение "методов поиска" и системы , с помощью которых , при наличии больших потоков данных , поступивших с экспериментальных установок , можно было осуществить эффективный поиск по любому сложному запросу .

использования в качестве баз данных .

В результате проведенных исследований было установлено , что для хранения информации о результатах экспериментов на базе БДМД с оперативной памятью нецелесообразно .

Для обеспечения быстрого доступа к базам данных , с помощью которой осуществлялось хранение экспериментальных результатов , была разработана система хранения информации .

Начальная база данных AVIS

Автором разработана реляционная база данных AVIS , которая отличается от всех существующих реляционных баз данных тем , что в качестве основной рассматривается не двумерная плоская таблица с N столбцами , а трехмерная (с N столбцами)

Вся последующая разработка системы , создание базового языка ANI и словаря данных , построено именно на таких отношениях . Данный подход был продиктован спецификой задачи , а именно : если смоделировать экспериментальную установку в виде некоторой плоской матрицы (или несколько матриц) с N столбцами , то в момент t_1 срабатывания установки матрица (или матрицы) примет конкретное значение . Значения таблицы в момент времени t_1 будут первым рядом строящейся трехмерной таблицы . В момент времени t_2 - следующее значение матрицы , что соответствует второму ряду трехмерной таблицы . т. д. . В итоге , полностью результаты эксперимента будут расположены в трехмерной таблице (или таблиц) , с M - рядами , где M будет означать число срабатываний установки во время эксперимента .

Практическая ценность работы .

На основании разработанного автором "методов доступа" на ЭВМ БЭСМ-6 удалось экономно распределить и организовать быстрый поиск данных с установки ПИОН физики космических лучей. На основе разработанной автором системы реляционных баз данных AVIS удалось организовать банк на смоделированных данных (на ЭВМ ЕС-1046), которая базируется на трехмерных таблицах и с которой работа осуществляется с помощью базового языка реляционного исчисления ANI, работающего именно с такими таблицами.

Апюбация работы.

Результаты работы, положенные в основу диссертации, докладывались на семинарах по космическим лучам, по автоматизации научных исследований и вычислительной техники Ереванского физического института.

Публикации.

По материалам диссертации опубликовано 5 работ I-51.

Объем работы.

Диссертация состоит из введения, трех глав заключения, списка литературы, содержит 99 печатных страниц машинописного текста, включая 49 рисунков, 1 таблицу.

Содержание работы.

Введение: Диссертационная работа посвящена созданию специализированных баз данных в области физики космических лучей. Огромный объем информации, поступающий с экспериментальных установок (количество $\approx 10^6$ сообщений в год), требует оптимизации его хранения, систематизации и

дальнейшей его обработки.

Эти задачи в определенной степени удалось решить с помощью выполненных работ, описанных в диссертации.

В первой главе дается обзор по работам в области баз данных физики высоких энергий и космических лучей. В частности, в работе [7] коротко рассматриваются возможности некоторых систем, такие как SQL/DS, KARASK, ZEBRA-RZ, ORACLE, которые применялись в работах по организации баз данных коллаборацией L3 в области физики высоких энергий. В физике космических лучей были созданы базы данных, которые, в частности, содержали экспериментальные данные с установки ШАЛ на Тянь-Шане.

В первой главе также дается введение в теорию реляционных баз данных.

Общая структура СУБД (система управления базами данных) состоит из следующих уровней: внешнего, логического, физического, и каждый уровень имеет свою модель.

В диссертационной работе рассматривается задача разработки реляционных баз данных для применения в экспериментальной физике. В первой главе приводятся основные операции реляционной алгебры, а именно: проекция, выборка, соединение, деление, декартово произведение, частное, разность, пересечение, объединение. Дается определение понятий отношение, домен, кортеж, нормализованное отношение, определение 1НФ, 2НФ, 3НФ, 4НФ (НФ - нормальная форма).

Во второй главе диссертации описывается проблемно -

ориентированный банк данных с установки ПИОН на горе Арагац .

В результате заполнения банка данных с помощью разработанной системы программ удалось осуществить запись ≈ 200 МЛ (магнитных лент) на МД в 29 МГбайт . На каждой МЛ было записано 400 зон по 422 36-и разрядных слов , в которых находятся в упакованном виде четыре 9-и битовых целых числа . В каждой зоне размещена информация трех срабатываний установки (3-и кадра) , количество всех кадров равно $\approx 10^6$. Передача всех МЛ с ЭВМ НАИРИ-2 на ЭВМ БЭСМ-6 осуществлялось как программно , так и аппаратно .

Набор программ , разработанный автором , обеспечивающих выполнение задачи по созданию банка данных , скорее всего был "методом доступа " или простой информационно - поисковой системой [1].

Перед разработчиком стояли следующие требования по организации банка :

1. Быстрота выборки .
2. Экономное распределение памяти
3. Защищенность системы от несанкционированного доступа
4. Компактность для совместного использования с программами пользователей .

Для получения быстрого ответа на запросы применялись известные в литературе [6] методы и средства . Перед заполнением области базы данных производилась упаковка и сжатие данных . В числовых массивах также было достаточно много случайно расположенных нулей . Их сжатие также позволяло экономно

использовать внешнюю память , отведенную под банк . И несмотря на пу сжатия , упаковки , распаковки и организации части справочника - каталога , которые ухудшали временные характеристики , быстрота выборки или время на запрос составили доли секунды ($\approx 0.3с.$) .

В третьей главе диссертации дается общее описание системы реляционных баз данных AVIS (от имени AVETIS) , которая состоит из следующих основных частей .

1. Управляющие программы
2. Собственный язык реляционного исчисления ANI [3,4] .
3. Язык описания схем , команды которых входят в язык реляционного исчисления ANI [3,4]
4. Словарь данных (со своими командами)

Для работы в системе AVIS необходима предварительная организация трех областей на внешнем носителе (МД) .

- а) область , где сосредоточены все управляющие таблицы базы данных и сами данные .
- б) область , которая используется для рабочих таблиц языка реляционного исчисления ANI , а также для команд словаря данных
- в) рабочая область , в которую будут извлечены из базы данные и в соответствии с запросом отсортированы . Эта область организована как SCRATCH и информация в них сохраняется до следующего запроса .

Управляющие программы системы AVIS выполняют следующие функции (перечислим основные) :

1. Первоначальная подготовка области а) перед первым вводом

данных .

2. Обнуление областей б) и в) .
3. Запуск системы (при обращении к ней) .
4. Организация интерфейса между программами пользователя и системой (ввод , вывод) .
5. Обращение к языку реляционного исчисления ANSI при поступлении запроса со стороны пользователя и к командам словаря данных .
6. Физическая организация базы данных
7. Сообщение диагностик в процессе работы системы .
8. Организация средств защиты .
9. Обеспечение ограничения целостности данных .

Система реляционных баз реализована на ЭВМ ЕС 1046 на языке ФОРТРАН -77 .

Система avis рассчитана на работу с отношениями , представленными в виде трехмерных таблиц (нормализованных) с n атрибутами , в отличие от всех существующих систем реляционного типа , разработанных для двумерных таблиц (с n - столбцами) . Это является отличительной чертой этой системы от всех существующих . Такое построение таблиц было обусловлено требованиями к данным в экспериментальной физике .

Пусть имеется установка , схему которой можно представить в виде некоторой таблицы (смоделировать) и каждое срабатывание установки представляет собой уникальным образом заполненную таблицу . (рис 1)

Если , в процессе эксперимента происходит несколько

срабатываний установки , то естественно , что можно данные , поступившие с эксперимента , представить в виде , изображенном на рис . 1 .

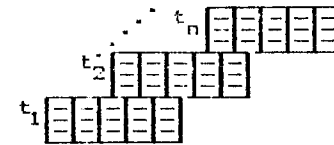


рис . 1

где t_1, t_2, \dots, t_n - времена срабатываний установки . В эксперименте может участвовать не одна , а несколько взаимосвязанных установок и тогда данную систему можно представить в виде совокупности таблиц с различным числом строк и столбцов , как показано на рис . 2 .

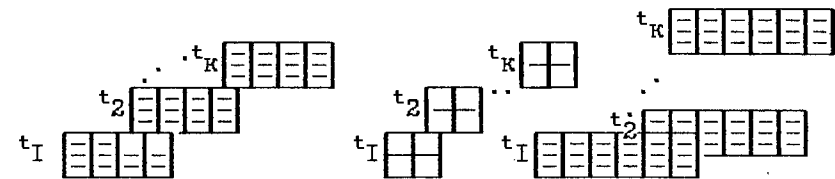


рис 2.

Именно рассмотрение таких структур легло в основу разработанной системы avis , к которой были применены

требования построения реляционных баз данных .

Это и стимулировало создание языка реляционного исчисления AMI , которая и базируется на работе с такими таблицами .

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

- 1) Вышеописанные множества двумерных таблиц представляются как одно отношение в виде трехмерной таблицы и описывается такое отношение один раз
- 2) С вводом понятия ряда (3-ий параметр таблицы) появляется возможность сравнения разных рядов одного и того же отношения
- 3) Однотипные таблицы , полученные из эксперимента , записываются одной командой , используя команду цикла , которая изменяет лишь номер ряда
- 4) Есть возможность изменения во время поиска (команда SEARCH) номеров рядов для каждого атрибута 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 в вышеуказанном режиме
- 7) Число строк каждого ряда отношения является числом произвольным .
- 8) Запросы реализованы на английском языке , но можно запросы организовать и на русском , армянском (важно , чтобы в операционной системе , в рамках которого работает язык , и на

вводных и выводных устройствах были введены буквы и их коды соответствующих языков).

Опишем язык реляционного исчисления AMI

ЯЗЫК РЕЛЯЦИОННОГО ИСЧИСЛЕНИЯ AMI

Часть запросов языка AMI предназначена для описания схем отношений , их атрибутов , типов и т. д. , а другая часть для манипулирования данными , для поиска , выборки , соединения , исключения и т. д. .

При описании каждого запроса будут даны ссылки на примеры. Все запросы (команды) завершаются символом % .

I). АТКВНУ(НАМ,О:А1:А2:....:АН) — описывает новое отношение (схему) с именем НАМ и именами атрибутов А1,А2,...АН

АТКВНУ(< идентификатор отношения > < идентификатор атрибута > { : < идентификатор атрибута > }) .

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

< имя отношения > ::= < идентификатор >

< номер ряда отношения > ::= < число > ,

где число — это номер ряда трехмерной таблицы отношения НАМ

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

<идентификатор> {<цифра>}

Число символов в идентификаторе не более четырех (в работе [3] говорится о числе символов не более шести для ЭВМ БЭСМ-6)

<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 .

< число > ::= < последовательность цифр >

< последовательность цифр > ::= { < цифра > }

цифра ::= 0|1|2|3|4|5|6|7|8|9

< идентификатор атрибута > ::= < идентификатор >

2) LENGTH(NAM, 0:1:2:.....:7:3) - описывает ширины строк всех атрибутов отношения NAM. Под шириной подразумевается число слов, образующих данную строку и имеющих атомарное значение

LENGTH(< идентификатор отношения > : < ширина > { : < ширина > }

< ширина > ::= число слов в строке

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

TYP(< идентификатор отношения > : < тип > { : < тип > }

< тип > ::= I|R|T|

I - целая константа

R - вещественная константа...

T - текстовая константа....

4) SS(NAM, 0: BV < 20.4 & NAM, 0: BV # 10.0) - описывает ограничение целостности, которое накладывается на атрибут BV отношения NAM.

SS (< логическое выражение >)

< логическое выражение > ::= < безусловно булевское выражение

> < > < безусловно булевское выражение > | (< логическое выражение

> < > < безусловно булевское выражение >)

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

< безусловно арифметическое выражение > ::= формула ,

построенная из чисел, идентификаторов отношений и идентификаторов атрибутов, арифметических операций +|-|*|:|**|, круглых скобок и элементарных функций SUMM|MAX|MIN|

< > ::= | , & , \

< сравнение > ::= < > | < = | > = | = | * |

5) CIPHER (ISK) - описывает шифр пользователя, его пароль CIPHER (< шифр >)

< шифр > ::= < идентификатор >

6) STER (1:2.....:4:8) - описывает шаги и их ограничения сверху для каждого отношения, атрибута, участвующего в запросе

STER (< шаг I > : < ограничение I > { , < шаг n > : < ограничение n > }

< шаг I > ::= целая константа

< ограничение I > ::= целая константа

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

STEP (< шаг I > : < ограничение I >)

8) EQU (NAM1; NAM2) - объявляет отношение с именем NAM2 копией отношения NAM1. Но новое отношение NAM2 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, 1: BV = " ВЕРОЯТНОСТЬ " & C, 1: DD <= 17.5%

Осуществляет поиск и выборку кортежей отношений или строк доменов отношений при выполнении некоторого условия. В нашем

примере осуществляется выборка строк атрибутов A,1:BB и C,1: DD (ряд первый отношения A и ряд первый отношения C), которые удовлетворяют условиям, изображенным на рис 3.

A,1:BB= "ВЕРОЯТНОСТЬ" & C,1:DD<= 17.5

рис . 3

SEARCH (< СПИСОК >) WHERE < УСЛОВИЕ >

< СПИСОК > ::= < идентификатор отношения > : < ALL > | < идентификатор отношения > : < идентификатор атрибута > | < переменная > = < безусловно арифметическое выражение > | < переменная > = < имя функции > < безусловно арифметическое выражение > | { < список > ; < список > }

< ИМЯ ФУНКЦИИ > ::= SUM | MAX | MIN - вычисляет сумму всех элементов множества, максимальный элемент множества, минимальный элемент множества.

< ALL > ::= все атрибуты данного отношения

< переменная > ::= идентификатор

< условие > ::= < логическое выражение >

10. SEARCH (< СПИСОК >) - осуществляет выборку из списка без какого-либо условия.

11. SEARCH (< СПИСОК >) WHERE < УСЛОВИЕ >

Осуществляет выборку всех строк атрибутов отношений из списка, если условие TRUE.

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

DELETE (< ИМЯ ОТНОШЕНИЯ >)

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. RENAME (A, 0: A1: A2) - переименовать имя атрибута A1 отношения A на имя отношения A2

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

16. UNITED (A, 1: ALL; B, 1: ALL; C, 1: ALL) WHERE A, 1: BB = B, 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 > < @ > < идентификатор отношения > : < идентификатор атрибута >

< @ > ::= < | < = | > | > = | = | * |

17. WRITE (A, 1: ALL) - осуществляет запись данных отношения A, 1, передаваемых пользователем списка, если условие TRUE.

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

Для команд словаря данных следующие имена подпрограмм: MENU, MENUA, MENU4, MENU3, MENU5, MENU1. В работе приводится краткий лексический, синтаксический, семантический анализ языка реляционного исчисления AMI, дается более подробное описание некоторых команд, их возможностей.

Критерии , определяющие выбор физической организации , отличаются от тех , которые определяют выбор логической организации данных. При выборе физической организации решающими факторами являются эксплуатационная эффективность , минимальные затраты , обеспечение малого времени ответа .

При разработке физической организации системы AVIS основными факторами явились :

1. Экономная организация памяти ;
2. Произвольная обработка файла;
3. Обработка одиночных транзакций ;
4. Добавление групп записей;
5. Поиск по нескольким ключам ;
6. Требования к поиску в файле;
7. Независимость данных;
8. Время ответа.

Инвертированные файлы представляют собой особый случай выборки по нескольким ключам , допускающей обработку запросов :

а) Простой запрос, когда определенному атрибуту задается конкретное значение , например (рис. 4):

[СПЕЦИАЛЬНОСТЬ=МАТЕМАТИКА]

рис. 4

б) Запрос по области значений , когда для определенного атрибута задается конкретная область значений , например (рис. 5) :

[21<=ВОЗРАСТ <=23]

рис. 5

в) Булев запрос состоит из запросов двух первых типов ,

соединенных операциями И , ИЛИ , НЕ (рис 6):

[КУРС=ВТОРОКУРСНИК) И (МЕСТОЖИТЕЛЬСТВО=ЕРЕВАН)]

рис. 6

Первый важный класс методов выборки по вторичным ключам основан на идее "инвертированного файла".

В системе AVIS был реализован принцип инвертированных списков . Сами данные в базе записаны в виде нормализованных таблиц . При поступлении запроса , в котором требуется произвести выборку по вторичным ключам , из базы извлекаются множества тех атрибутов , которые соответствуют ключам . После этого переэnumerуются все значения этих множеств в отдельности (устанавливаются порядковые номера) . Для каждого атрибута создается список , в котором индекс содержит : а) в первом столбце - значения самого атрибута , б) во втором столбце - значения порядковых номеров . На основе запроса (операции отношения или логические операции) , производится выборка из этих списков и отдельно формируются инвертированные списки , содержащие значения , соответствующие значениям вторичных ключей из запроса . Одновременно начала адресов этих инвертированных списков записываются в специальные таблицы .

Система AVIS была протестирована на отношениях смешанного типа . Были смоделированы данные , диапазон которых находился в пределах значений экспериментальных данных . Данные были типа INTEGER , REAL, TEXT.

Далее была проверена возможность дозаписи к уже

существующим отношениям новых кортежей для каждого конкретного ряда таблицы в отдельности. После ввода этих отношений в базу были протестированы все основные команды языка ANI. Результаты работы каждой команды или запроса получались на экране дисплея и хранились в выводном файле операционной системы. Особое внимание было уделено команде SEARCH, как основной, и были проверены различные ее виды применения над вышеуказанными отношениями. Были выявлены ошибки, которые были устранены.

Проведена сравнительная характеристика системы AVIS с другими системами, как это сделано в работе [73].

Воспользуемся 5-ти бальной системой для указания того, насколько эффективна система AVIS в каждой из категорий, которые считаются важными для специализированных прикладных систем, в частности, в физике высоких энергий и космических лучей, а именно:

1. Полный набор признаков
2. Эффективность
3. Фортранный доступ
4. Терминальный доступ
5. Конкурентные записи
6. Мобильность фортран программ
7. Мобильность данных
8. Живучесть
9. Защита
10. Дешевизна

Рассмотрим таблицу, которая приводится в работе [73] и

введем в нее систему AVIS в качестве 6-той системы (как на таб. I)

Некоторые пункты требуют дополнительного пояснения.

1. Сравнение по категории "полный набор признаков" показывает, какой из этих системных продуктов является коммерческой базой данных, и какой является пакетом, разработанным для удовлетворения более ограниченных требований.

2. Эффективность. Здесь подразумевается эффективный поиск больших объемов информации со скоростями, близкими к аппаратным, достаточно эффективный ввод данных.

5. Конкурентные записи.

Должна быть возможность блокировки (в идеале автоматическая) достаточной малой части базы данных.

Мы постарались по возможности объективно оценить и обосновать возможности системы AVIS (в диссертации подробный разбор проведен) и в таблице I она имеет следующие баллы по всем категориям: 3,3,5,3,4,2,4,3,3,5. Можно сказать, что система AVIS по своим характеристикам близка к коммерческим системам, но как специализированная, в ней особое место было уделено вопросам организации удобного фортранного доступа со стороны пользователя, максимальное отображение рассматриваемых в эксперименте схем, таблиц при построении базы данных и с этим связанные средства доступа к данным. В предлагаемом варианте системы реализованы не все ее возможности, она способна к расширению и дальнейшее ее развитие в том или другом направлении во многом будет зависеть от областей ее

ПРИМЕНЕНИЯ.

	ORACLE	SQL/ DS	ING- RES	KA- PACK	ZEBRA -RZ	AVIS
FULL FEATURES	*****	****	*****	*	**	***
EFFICIENCY	***	**	***	****	*****	***
FORTRAN ACCESS	***	**	**	*****	*****	*****
TERMINAL ACCESS	*****	****	*****			***
CONCURRENT WRITE	*****	*****	*****		***	****
PORTABILITY OF FORT	*****	***	***	*****	*****	****
PORTABILITY OF DATA	***		***	**	****	**
ROBUSTNESS	*****	*****	*****		**	***
SECURITY	****	****	****		***	***
CHEAPNESS		**		*****	*****	*****

Таблица I

В заключении сформулированы выносимые на защиту основные результаты работы:

1. Было разработано и внедрено математическое обеспечение односторонней связи ЭВМ НАИРИ-2 и ЭВМ БЭСМ-6 через канал фотосчитывающего устройства
2. Разработан и внедрен проблемно - ориентированный банк данных на ЭВМ БЭСМ-6 для информации с установки ПИОН на горе Арагац, предназначенной для изучения состава и потока вертикального космического излучения

3. Разработана и внедрена система реляционных баз данных AVIS, которая имеет следующие особенности

- а) в отличие от существующих систем реляционных баз данных под отношением здесь подразумевается трехмерная таблица с n атрибутами, частным случаем которой является двумерная плоская таблица. Такое построение отношений диктовалось спецификой представления данных, поступивших с эксперимента и существенно облегчает понимание и использование подобных структур физиками.
- б) Построение таких структур позволило однотипные отношения описывать один раз.
- в) С вводом понятия ряда (3-ий параметр таблицы), появляется возможность сравнения разных рядов одного и того же отношения.
- г) Однотипные таблицы записываются одной командой, используя команду цикла, которая изменяет лишь номер ряда
- д) Есть возможность независимого изменения друг от друга номеров рядов в режиме цикла для каждого отдельного атрибута, участвующего в запросе
- е) Есть возможность сравнения или поиска нескольких отношений (их рядов) друг с другом.
- ж) Если имеется несколько баз данных, то поиск, выборка и т. д. между ними организуется также в вышеуказанном режиме.
- з) - Число строк каждого ряда отношения является числом произвольным.
- и) В случае введения на вводных - выводных устройствах букв (и их кодов) армянского языка запросы можно организовать и на армянском языке.

ЛИТЕРАТУРА

1. В.В. АКОПОВ, В.В. АРАКЕЛ, С.Х. АСАДУБЕК. ПРОБЛЕМЫ
ОРГАНИЗАЦИИ ПРОБЛЕМНО-ОРИЕНТИРОВАННОГО ВЗАИМНОГО ОБМЕНА
СРЕДСТВАМИ ПРЯМОГО ДОСТУПА К БАЗЕ ДАННЫХ. В КНИЖКЕ
1977.

2. В.В. АКОПОВ, С.Х. АРАКЕЛ, Г.Г. БОСОВИЧ, В.В. АСАДУБЕК. В.С.
ОГАНЕЗОВА

СИСТЕМА ОДНОСТОРОННЕГО ОБМЕНА ДАННЫМИ И БАЗА ДАННЫХ
БДМ 284(15)-23, Г. ЕРЕВАН.

3. С.Х. АРАКЕЛ

БАЗА РЕЛЯЦИОННОГО МОДЕЛИРОВАНИЯ АНТ. ПРЕДУПРЕЖДЕНИЕ
Г. ЕРЕВАН.

4. С.Х. АРАКЕЛ

СИСТЕМА РЕЛЯЦИОННЫХ БАЗ ДАННЫХ AVIS. ЖУРНАЛ
"ПРОГРАММИРОВАНИЕ" (В ПЕЧАТИ).

5. ASATIANI T. L., ARUTUNIAN S. CH., GENINA L. J., MOCHAROV YU. A.

EXPERIMENTAL STUDY OF SUPERFAMILIES WITH NALO. PROCC. OF XVIII
ICDC BANGALOR 15-28 (1983)

6. ДЖ. МАРТИН

ОРГАНИЗАЦИЯ БАЗ ДАННЫХ В ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ. ИЗДАТЕЛЬСТВО
"МИР", 1979Г.

7. R.P. MOUNT

DATABASE SYSTEMS FOR NEP EXPERIMENTS

Технический Редактор А.С.Абрамян

Подписано в печать 28.12.91г.

Формат 60x84/16

Офсетная печать. Уч. изд. л. 1,0

Тираж 170 экз.

Зак. тип. № 190

Отпечатано в Ереванском физическом институте

Ереван 36, ул. Братьев Аликянян 2