

53



ИНСТИТУТ ЯДЕРНОЙ ФИЗИКИ СО АН СССР

И.Е. Сапрыкин, В.И. Терехов

**ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС
АВТОМАТИЗАЦИИ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ:
ЗАПРОСНАЯ ПОДСИСТЕМА**

ПРЕПРИНТ 89-83



НОВОСИБИРСК

Инструментальный комплекс автоматизации
информационных технологий:
запросная подсистема

И.Е. Сапрыкин, В.И. Терехов

Институт ядерной физики
630090, Новосибирск 90, СССР

АННОТАЦИЯ

В препринте излагается запросная подсистема инструментального комплекса автоматизации информационных технологий. Наличие языка запросов придает определенную замкнутость, законченность набору языков, представляющих необходимый инструментарий для построения систем автоматизации ряда информационных технологий. С его помощью можно получать определенные варианты прикладных систем без программирования, т. е. самими пользователями. Предложенная реализация отличается послойной архитектурой и комбинированной стратегией поиска, совмещающей использование индексов и последовательного просмотра образованных на базе индексов подмножеств кортежей.

СО-98 11014013471

© Институт ядерной физики СО АН СССР

ВВЕДЕНИЕ

Основное назначение автоматизированных информационных систем (АИС) — обеспечение информацией пользователей путем реализации вопросно-ответного отношения между языком запросов и языком ответов [1]. Решение этой задачи определяется выбранной при построении АИС формой представления информации. В [2] предложена методология проектирования АИС, согласно которой объектами АИС являются тексты — предложения естественного языка (ЕЯ). Для построения модели базы данных (БД) сформулирована модель данных (МД), предлагающая изобразительные средства для моделирования составляющих — словосочетаний, являющихся элементами синтаксической структуры исходного предложения, и их взаимосвязей древесной совокупностью типов отношений. Эти лингвистические объекты (целиком или их осмысленные фрагменты) выступают в качестве цели запроса, адресованного к информационной системе, и предопределяют форму ответа. Таким образом, задачей запросной подсистемы является выбор объектов-предложений, элементы которых удовлетворяют определенным условиям, образующим предикат запроса.

Два характерных момента отличают предлагаемое построение запросной подсистемы: подход к решению запросной задачи и архитектура подсистемы.

1. Можно выделить два полярных подхода к выполнению работ по "вычислению" кортежей, удовлетворяющих предикату запроса:

А. Последовательная выборка кортежей БД и их проверка на удовлетворение условиям, налагаемым соответствующим предикатом запроса на атрибуты этих кортежей.

Это — лобовое, прямолинейное решение обладает очевидным

недостатком для корневого типа отношения — крайней неэффективностью вследствие полного перебора БД. В то же время для подчиненных типов, количество экземпляров которых невелико, этот алгоритм достаточно удовлетворителен.

Б. Ведение в процессе взаимодействия с БД индексов по всем или некоторым атрибутам кортежей. Наличие для специфицируемых в запросе атрибутов индексов позволяет определить список удовлетворяющих предикату запроса кортежей в результате операций пересечения и объединения над соответствующими индексами — цепочками пар: значение атрибута — список идентификаторов соответствующих кортежей.

Этот подход, вследствие непредсказуемости возможных запросов, требует индексирования по всем атрибутам, что предъявляет повышенные требования к аппаратным ресурсам — объему внешней памяти и производительности процессора и может существенно сказаться на реакции системы.

Поэтому наиболее жизнеспособным является комбинированный вариант — определение по заданным из определенных соображений индексам цепочки объектов БД, содержащей идентификаторы кортежей, удовлетворяющих подмножеству атрибутов запроса и последующего тестирования отобранных кортежей на оставшиеся условия, т. е. последовательный просмотр уже только на подмножестве, как правило, существенно меньшей мощности.

2. Одной из фундаментальных концепций построения программных систем является концепция послойной архитектуры [3, 4]. Применительно к запросной подсистеме эта концепция нашла воплощение в наличии ряда языковых уровней-интерфейсов, определяемых их ориентацией. Оказались целесообразными следующие уровни:

1. Уровень пользователя-1.
2. Уровень пользователя-2: уровень схемы-типов отношений.
3. Уровень системы виртуальных машин.
4. Уровень базовой машины.

1. УРОВЕНЬ ПОЛЬЗОВАТЕЛЯ-1

Спецификация запроса выполняется на интенционале соответствующего предложения. Для обеспечения пользователя требуемой информацией используется графическое изображение на экране типов отношений, представляющих фрагменты соответствующего документа (как и при операции ввода данных). Только вводи-

мыми данными являются знаки арифметических операторов сравнения (“<”, “=”, поиск в диапазоне — “> <” и т. д.) для специфицируемых атрибутов. Атрибуты одного типа отношения образуют конъюнкт, а наличие нескольких экземпляров одного типа определяет дизъюнкцию конъюнктов, т. е. формула запроса представлена в дизъюнктивной нормальной форме. Кванторы EX, ALL, относящиеся к какому-либо типу, указываются явно. Причем, на этом уровне специфицируется не конкретный запрос, а только его тип, т. е. указываются атрибуты поиска и операции сравнения. Например, на рис. 1.1 изображен запрос (описание соответствующей базы данных — схемы АРХ — приведено в [5]): «Выдать документы «ПРИХОД» за xxxx месяц поставщика xxxxxx, содержащие изделия аааааааааа».

АРХИВ ДОКУМЕНТОВ

ДАТА:=

АРХИВ ПРИХОДОВ

Номер документа	Отправитель	Поставщик	Номер посылки	Сумма				
=								
ГР.	П/ГР	Номер	Количество	Сумма	Тип	Кеик	Цена	Кеиц

> <

Составной атрибут: НМП
Квантор: E

Рис. 1.1.

2. УРОВЕНЬ ПОЛЬЗОВАТЕЛЯ-2: УРОВЕНЬ ТИПОВ ОТНОШЕНИЙ

Уровень пользователя-2 как и Уровень-1 доступен пользователю, но является более формализованным и наряду с функцией определенного слоя в программном комплексе предназначен для контроля и уточнения первоначального запроса.

Предложение языка запросов на этом уровне — уровне типов отношений, входящих в схему — представляет собой перечень пар:

(имя отношения — формула реляционного исчисления с переменными-кортежами [6]). (1)

Формулы построены из имен атрибутов соответствующих отношений и представлены в дизъюнктивной нормальной форме. Запрос на языке пользователя транслируется и на уровне типов отношений представлен в виде (1) с окнами-шаблонами для задания соответствующих значений (рис. 2.2). Кроме того, в этой форме индексируемые атрибуты помечены звездочками и пользователь имеет право удалить некоторые из них, перенося соответствующие проверки на блоки тестирования и, таким образом, задавая алгоритм решения поисковой задачи (алгоритмы изложены в разделе 5).

Предложения запроса на этом уровне выглядят следующим образом (для вышеприведенного примера):

Имя запроса	Имя схемы		Имя файла	Комментарий
	исходной	результата		
А	Арх	Док	При	Выборка из архива
Дат	KEY = 9999			
При	*Поставщ = xxxxxx			
Нмп	&EX Нмп: (KEY > 9999999999 & KEY < 9999999999)			
	KEY > 9999999999 & KEY < 9999999999			

Рис. 2.2.

Подготовленные в таком виде запросы хранятся в библиотеке Q(QUERY) (рис. 2.3). Запуск конкретного запроса на выполнение осуществляется командой ВПП. При этом потребуются в вышеприведенной форме указать конкретные значения атрибутов.

Уровень поддерживается с помощью интерактивной модели [7] следующей схемой (рис. 2.3):

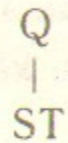


Рис. 2.3.

Здесь,

- Q — идентификация запроса (рис. 2.4);
- ST — предложение запроса (рис. 2.5).

Имя запроса	Имя схемы		Имя файла	Комментарий
	исходной	результата		
xxxxxx	xxxxxx	xxxxxx	xxxxxx	xxxxxxxxxxxxxxxxxxxxxxxx

Рис. 2.4. Отношение Q.

Имя отношения	Условие
xxxxxxx	xx

Рис. 2.5. Отношение ST.

3. УРОВЕНЬ СИСТЕМЫ ВИРТУАЛЬНЫХ МАШИН

3.1. Структура и генерация виртуальных машин

Уровень виртуальных машин (ВМ) реализует так называемый “машинный подход” [8] к программированию, когда выполнение программы рассматривается как результат функционирования и взаимодействия виртуальных машин (ВМ): машины баз данных, машины вычисления предикатов, дисплей-машины и т. д. Эта идея аналогична идее семейства языков, когда вместо монолитного единого и большого языка предлагается набор сравнительно малых языков с разными средствами и функциями [9]. “Машинный подход” способствует технологичности программного изделия благодаря систематичности дисциплины разработки и структурированности программного изделия.

Совокупность команд, реализующих определенный тип специализированной обработки, образует виртуальную машину (ВМ). Каждая ВМ имеет имя, по которому она может быть вызвана, и таблицу команд. Команда вызова ВМ имеет один параметр — имя файла, содержащего последовательность команд, подлежащих выполнению этой машиной.

Команда — основная синтаксическая конструкция языка виртуальной машины. Она представляет собой последовательность лексем трактуемых следующим образом (в порядке следования):

- метка;
- имя команды — именем команды может служить идентификатор, знак операции, числовой код и т. д.

- метка перехода, если значение, выработанное командой, в качестве условия завершения равно 1. Это значение хранится в переменной STATUS;
- метка перехода, если это значение равно 0;
- параметры — команды имеют разное число параметров.

Выбранный синтаксис придает компактность и ясность программному коду, способствует синтезу программы (разрешению ссылок) и является следствием специфики наборов команд VM, заключающейся в обязательной явной проверке условия выполнения команд для определения дальнейшего пути выполнения.

Введение языкового слоя в виде виртуальных машин основывается на возможности соответствия каждому типу отношения, с целью доступа к его кортежам при решении запросной задачи, программного блока на языке виртуальных машин DB (описание приведено ниже, в разделе 4):

Метка	Команда	След. команда		Комментарий
		OK=1	OK=0	
гп.D	DW	гп.G	гп.U	установка текущего типа отношения; выборка кортежа; проверка предиката — запуск TST-программы; пересылка кортежа в БД-результат; восстановление текущего типа отношения.
гп.G	GN	гп.T	гп.U	
гп.T	TST	гп.O	гп.G	
гп.O	OUTD		1	
гп.U	UP		2	

Для создания и поддержания VM создана структура данных следующей схемы CVM:

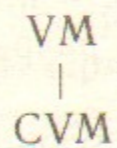


Рис. 3.1.

Здесь:

VM — идентификация виртуальной машины (рис. 3.2);
CVM — описание команд (рис. 3.3).

Имя машины	Комментарий
xxxxx	xx

Рис. 3.2. Отношение VM.

Команда	Ссылка	Комментарий
xxxxxx	xxxxxxxxxxx	xx

Рис. 3.3. Отношение CVM

Определение виртуальной машины начинается с создания экземпляра отношения VM с именем новой машины в атрибуте МАШИНА. В подчиненном отношении CVM определяются команды виртуальной машины. Значение атрибута КОМАНДА — мнемоника команды. Значение атрибута ССЫЛКА — ссылка на программу, реализующую выполнение команды в формате:

метка ↑ имя программы.

На рис. 3.4 приведены две VM — DB и TST (набор команд виртуальных машин приведен в Приложении 1)

Машина		Комментарий:
DB		Машина без данных

Команда	Ссылка	Комментарий
RDY	RDY ↑VMCDB	«открытие» схемы ВХОД (актуализация) в отношение ВЫХОД из отношения (к родителю) чтение следующего кортежа чтение кортежа по ключу чтение по ключу в диапазоне чтение по индексу в диапазоне вывод кортежа в базу данных
DW	DW ↑VMCDB	
UP	UP ↑NDB1	
GN	GN ↑NDB2	
GK	GK ↑NDB2	
GNK	GNK ↑NDB2	
GNI	GNI ↑NDB2	
OUTD	OUTD ↑NDB2	

TST ТЕСТ-машина

Команда	Ссылка	Комментарий
≠	NE ↑VMCTST	
=	EQ ↑VMCTST	
<	LT ↑VMCTST	
≤	LE ↑VMCTST	
>	GT ↑VMCTST	
≥	GE ↑VMCTST	

Рис. 3.4.

текста. TST-модули реализуют вычисление предикатов, соответствующих типам отношений, а DB-модули — вызовы в случае наличия в формуле кванторов EX и ALL, содержащих условия на подчиненные типы.

4.1. Трансляция запроса

- Процесс трансляции включает в себя четыре основных этапа:
- построение схемы, описывающей структуру БД-результата;
 - построение внутреннего представления запроса;
 - генерация комплекса DB- и TST-модулей;
 - разрешение логических ссылок.

Преобразование запроса во внутреннее представление позволяет эффективно решить задачу генерации комплекса DB- и TST-модулей, составляющих программу выполнения конкретного запроса. Внутреннее представление запроса имеет древовидную структуру (рис. 4.1):

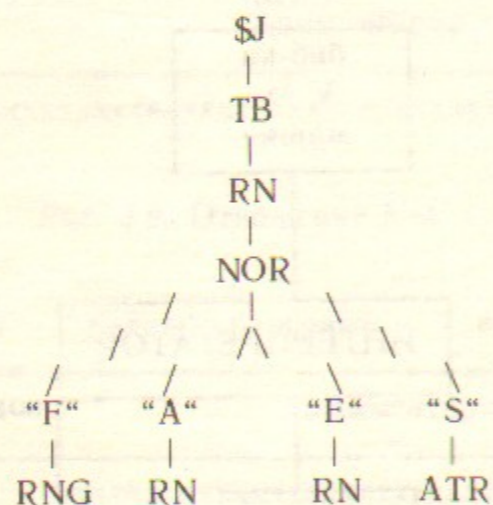


Рис. 4.1.

Здесь

Имя запроса	Имя схемы		Имя файла
	исходной	результата	

Рис. 4.2. Отношение \$J

TB — тип блока — принимает значения “M” (MAIN), “A” (ALL) или “E” (EX).

Имя сына	Имя брата	Флаг TST	Команда выборки	Парам 1	Парам 2	Парам 3
----------	-----------	----------	-----------------	---------	---------	---------

Рис. 4.3. Отношение RN.

NOR — номер дизъюнкта.

“F” — список условий на поисковые атрибуты.

“S” — список условий на простые атрибуты.

“A” — список составных атрибутов, для которых в формуле отбора указан квантор всеобщности “All”.

“E” — список составных атрибутов, для которых в формуле отбора указан квантор существования “EX”.

Ссылки на значения границ диапазона	
на нижнюю	на верхнюю

Рис. 4.4. Отношение RNG.

Номер атрибута	Арифметический оператор сравнения	Ссылка на значение
----------------	-----------------------------------	--------------------

Рис. 4.5. Отношение ATR.

Сгенерированные модули записываются в библиотеку PVM и, как уже было сказано выше, используются интерпретатором VM при выполнении программы запроса. Имена программам присваиваются по следующим правилам:

⟨имя головной DB-программы⟩ :: = ⟨имя запроса⟩

⟨имя DB-подпрограммы⟩ :: = ⟨имя запроса⟩ . ⟨имя отношения⟩

⟨имя TST-программы⟩ :: = ⟨имя запроса⟩ . ⟨имя отношения⟩

4.2. Структура модулей обращения к БД

Как уже говорилось введение языкового слоя в виде виртуальных машин основывается на возможности соответствия каждому типу отношения определенного программного блока на языке виртуальных машин DB и TST:

Метка	Команда	След. команда		Комментарий
		OK=1	OK=0	
гп.D	DW	гп.G	гп.U	установка текущего типа отношения;
гп.G	GN	гп.T	гп.U	выборка кортежа;
гп.T	TST	гп.O	гп.G	проверка предиката — запуск TST-программы;
гп.O	OUTD		1)	пересылка кортежа в БД-результат;
гп.U	UP		2)	восстановление текущего типа отношения.

Здесь гп — переменная, содержащая имя отношения. Т. е. имя метки имеет переменную часть, префикс — имя отношения, и постоянную (после точки), ассоциируемую с командой. Таким образом, задача разрешения ссылок имеет статическое решение.

Указанная структура программного блока имеет принципиальный характер. В зависимости от положения типа отношения в дереве, вида предиката запроса, выбранного алгоритма поиска блок видоизменяется. Возможные вариации структуры программных блоков реализуются метками переходов G0 и G1, принимаемые значения которых в зависимости от тех или иных факторов приведены в описании команд ВМ (приложение 1).

Таким образом, программа взаимодействия с БД состоит из совокупности описанных блоков обращения к БД, составленных из команд виртуальной машины БД и соответствующих именам отношений. Существование логической связи между командами блока (и блоков между собой) позволяет введением логических меток решить задачу синтеза соответствующей запросу программы на языке ВМ.

4.3. Структура программ вычисления предикатов

Каждой формуле соответствует TST-модуль, составленный с помощью команд TST-машины (прил. 1).

Результат работы программы — RET 0 или RET 1, свидетельствует о том, удовлетворяет ли считанный кортеж формуле запроса. Так как формула представлена дизъюнкцией конъюнкций, т. е. в виде:

$$(A1 < C1 \& A2 = C2) \text{ OR } (A3 = C3 \& A4 = C4) \text{ OR } \dots \text{ OR } (A_n > = C_n)$$

целесообразно метки операторов TST-программы образовывать по следующему правилу: $\langle O \rangle . \langle A \rangle$, где префикс $\langle O \rangle$ — номер конъюнкта, $\langle A \rangle$ — номер условия в конъюнкте. Нижеследующий пример (п. 4.4) иллюстрирует структуру TST-программ.

4.4. Пример трансляция запроса

В качестве примера приведем текст программы на языке ВМ для вышесформулированного запроса А (рис. 1.1):

ДАТ : KEY = 9999

ПРИ : *ПОСТАВЦ = xxxxxxx

:& EX НМП: (KEY > 999999999999 & KEY < 999999999999)

НМП: KEY > 999999999999 & KEY < 999999999999

В результате трансляции запроса получим следующие модули:

Программа		Комментарий				
А		Головная DB-программа				
Метка	Мнемоника команды	Метки перехода при		Параметры		
		OK=0	OK=1	N 1	N 2	N 3
ДАТ.D	RDY	RET	ДАТ.G	АРХ		
ДАТ.G	GK	RET	ДАТ.O	%V(ПРИ,1)		
ДАТ.O	OUTD		ПРИ.D	ДОК		
ПРИ.D	DW	ПРИ.U	ПРИ.G	ПРИ		
ПРИ.G	GI	ПРИ.U	ПРИ.T	%V(ПРИ,2) 3		
ПРИ.T	TST	ПРИ.G	ПРИ.O	А.ПРИ		
ПРИ.O	OUTD		НМП.D	ДОК		
ПРИ.U	UP		RET			
НМП.D	DW	НМП.U	НМП.G	НМП		
НМП.G	GNK	НМП.U	НМП.O	%V(НМП,5) %V(НМП,6)		
НМП.O	OUTD		НМП.G	ДОК		
НМП.U	UP		ПРИ.G			
RET	RET			1		

Программа	Комментарий
А.НМП	Вызывается из TST-программы А.ПРИ

Метка	Мнемоника команды	Метки перехода при		Параметры		
		OK=0	OK=1	N 1	N 2	N 3
НМП.D	DW	U0	НМП.G	НМП		
НМП.G	GNK	U0	U1	%V(НМП,3) %V(НМП,4)		
U0	UP		R0			
U1	UP		R1			
R0	RET			0		
R1	RET			1		

Программа	Комментарий
А.ПРИ	TST-программа вычисляет выражение EX Нмп:(KEY > ... ' KEY < ...)

Метка	Мнемоника команды	Метки перехода при		Параметры		
		OK=0	OK = 1	N 1	N 2	N 3
I.1	DB	R0	R1	А.НМП		
R0	RET			0		
R1	RET			1		

5. АЛГОРИТМЫ ПОИСКА

Как уже говорилось в разделе 3, в программном блоке, соответствующем работе с определенным типом отношения, команда выборки кортежа из БД является переменной, ее точный вид определяется принятой стратегией решения задачи запроса. Ниже перечислены эти стратегии. Первые три применимы в случае, когда формула запроса представляет собой единственный конъюнкт.

СТРАТЕГИЯ 1. Последовательный перебор БД командой GN и отбор кортежей с помощью TST-программы, т. е. кортежей соответствующих истинности предиката.

СТРАТЕГИЯ 2. Основывается на существовании идентификатора записи KEY. Наличие служебного слова KEY в формуле транслируется в блок обращения к БД с командой виртуальной машины

GNK [нижняя граница значения ключа;
верхняя граница].

В TST-программе исключаются строки, связанные с проверкой условий на KEY.

СТРАТЕГИЯ 3. Основывается на существовании индексируемых атрибутов, а именно, в формуле хотя бы один атрибут индексируемый. Блок обращения к БД включает команду

GNI [имя атрибута;
нижняя граница значения атрибута;
верхняя граница],

и, таким образом, перебор в БД ограничивается подцепочкой соответствующего индекса. Если в формуле присутствуют несколько индексируемых атрибутов, то выбор индекса определяется минимальной длиной подцепочки.

СТРАТЕГИЯ 4. Наиболее общая стратегия для запросов либо с наличием дизъюнкции в формуле, либо при использовании для решения поисковой задачи нескольких индексов. Эта стратегия предполагает предварительную обработку с помощью операций над индексными цепочками:

— выборка подцепочки, S-операция

GList [имя атрибута;
нижняя граница значения атрибута;
верхняя граница];

— A(AND) — операция

AList [имя результирующей цепочки;
имя1 цепочки;
имя2 цепочки;
.....
имяп цепочки];

— O(OR) — операция

OList [имя результирующей цепочки;
имя1 цепочки;
имя2 цепочки;
.....
имяп цепочки].

С каждой цепочкой, образованной в результате AND-операции, связан соответствующий TST-блок, т. е. имеется пара:

[имя цепочки — имя TST-блока].

Результатом OR-операции является цепочка, в которой с каждым идентификатором кортежа, удовлетворяющим после первой стадии отбора, связан список имен TST-блоков, т. е. результирующая цепочка состоит их пар:

[идентификатор кортежа — список имен TST-блоков (TST1, TST2,...)]

Таким образом, выборка из БД ограничивается цепочкой, определенной в результате SL-, AL-, OL-операций. В блок обращения к БД включена команда GNLlist [имя цепочки].

РЕЗЮМЕ

Наличие языка запросов придает определенную замкнутость, законченность набору языков, представляющих необходимый инструментарий для построения систем автоматизации ряда информационных технологий. С его помощью можно получать определенные варианты систем без программирования, т. е. самими пользователями. Предложенная реализация отличается послойной архитектурой и комбинированной стратегией поиска, совмещающей использование индексов и последовательного просмотра образованных на базе индексов подмножеств кортежей.

ЛИТЕРАТУРА

1. Цаленко М.Ш. Моделирование семантики в базах данных.—М.: Наука. Гл. ред. физ.-мат. лит.,1989, 288 с.
2. Гейци И.И., Терехов В.И. Методологические аспекты построения систем автоматизации информационных технологий.—Новосибирск, 1988, 24с.—Препринт АН СССР. Сиб. отд-ние. ВЦ, 804.
3. Шоу А. Логическое проектирование операционных систем.—М.: Мир, 1981, 360 с.
4. Крамаренко Р.П., Голощук Н.А. Мультипроцессорная схема управления данными в СУБД с послойной архитектурой.— УСиМ, 1983, № 4, 1983, с.97—103.
5. Гейци И.И., Терехов В.И. и др. Построение оперативных систем автоматизации учреждений информационных технологий. —Новосибирск, 1989.—Препринт АН СССР. Сиб. отд-ние. ВЦ, 838.
6. Ульман Дж. Основы систем баз данных.—М.:Финансы и статистика, 1983, 334 с.
7. Гейци И.И., Канаков С.А. и др. Интерактивная подсистема технологического комплекса автоматизации информационных технологий.—Новосибирск,1988, 30с.—Препринт АН СССР. Сиб. отд-ние. ВЦ, 805.
8. Левин Д.Я. Инструментальный комплекс программирования на основе языков высокого уровня/Под ред. А.П. Ершова.—М.: Наука. Гл. ред. физ.-мат. лит., 1987. 200 с.
9. Агафонов В.Н. Спецификация программ: понятийные средства и их организация.—Новосибирск: Наука, 1987.

КРАТКОЕ ОПИСАНИЕ КОМАНД ВМ

Формат команд:

Метка	Мнемоника команды	Метки перехода при		Параметры		
		KO=0	OK = 1	N 1	N 2	N 3
%M	%CD	%G0	%G1	%P1	%P2	%P3

При описании команд используются следующие обозначения:

- gn — имя текущего типа отношения;
- ch — имя сына для текущего типа отношения;
- br — имя брата для текущего типа отношения;
- pr — имя отца для текущего типа отношения.
- tb — тип блока, в котором используется команда;
- sf — имя схемы, которая описывает исходную БД;
- st — имя схемы, которая описывает БД-результат;
- lw — нижняя граница диапазона значений;
- hi — верхняя граница диапазона значений;
- var — ссылка на параметр запроса.
- pa — номер атрибута в кортеже.

КОМАНДЫ ДВ-МАШИНЫ

1. Команды установки текущего типа отношения

RDY — начать работу со схемой.

gn.D	RDY	RET	rn.G	sf	[ln]
------	-----	-----	------	----	------

- gn — в данном случае имя корневого отношения;
- ln — флаг для запуска программы формирования списка; Необходим, если далее в программе используется команда выборки GNL.
- DW — войти в составной атрибут.

гп.D	DW	гп.U	гп.G	гп	[ln]
------	----	------	------	----	------

UP — выйти из составного атрибута.

гп.U	UP		xxx.x
------	----	--	-------

метка перехода %G1 может принимать следующие значения :

%G1	в случае, если ...
br.D	для данного типа отношения в запросе определен брат
pr.U	нет брата и экземпляр отца выбран командой GK
pr.G	экземпляр отца выбран командой, отличной от GK

2. Команды выборки

Общим для всех команд выборки является

- метка команды - гп.G
- метки переходов %G0 и %G1, принимающие следующие значения

%G0	в случае, если ...
RET	команда в блоке для корневого типа отношения
гп.U	в противном случае
%G1	в случае, если ...
гп.T	в блоке есть инструкция TST
гп.O	в противном случае

GN — дать следующий кортеж

гп.G	GN	%G0	%G1
------	----	-----	-----

GNL — дать следующий кортеж из списка

гп.G	GNL	%G0	%G1
------	-----	-----	-----

GNI — дать кортеж по назначению поискового атрибута в диапазоне

гп.G	GNI	%G0	%G1	[lw]	[hi]	na
------	-----	-----	-----	------	------	----

Если не задан %P1, то поиск с начала файла.

Если не задан %P2, то поиск до конца файла.

GI — дать кортеж по значению поискового атрибута

гп.G	GI	%G0	%G1	var	na
------	----	-----	-----	-----	----

var указывает на элемент массива %v. это двумерный массив, где значение первого индекса — имя отношения, значение второго индекса — номер параметра.

GNK — дать следующий по ключу в диапазоне

гп.G	GNK	%G0	%G1	[lw]	[hi]
------	-----	-----	-----	------	------

Если не задан %P1, то поиск с начала файла.

Если не задан %P2, то поиск до конца файла.

GK — дать кортеж по значению ключа

гп.G	GK	%G0	%G1	var
------	----	-----	-----	-----

var — значение ключа.

3. Команда вывода

OUTD — вывод экземпляра в базу данных

гп.O	OUTD		%G1	st
------	------	--	-----	----

%G1 принимает следующие значения

%G1	в случае, если ...
ch.D	для текущего типа отношения определен сын
gn.G	нет сына и команда выборки не GK
gn.U	в блоке команда выборки GK и нет сына
RET	то же, но для корневого отношения.

КОМАНДЫ TST-МАШИНЫ

or.and	CD	%G0	%G1	na	ref
--------	----	-----	-----	----	-----

Метка команд — or.and, где

or — номер дизъюнкта;

and — номер конъюнкта в дизъюнкте.

Метки переходов %G0 и %G1 принимают следующие значения:

%G1 = R1, если текущая команда не последняя в дизъюнкте;

= or.(and + 1) — в противном случае.

%G0 = (or + 1).and, если текущий дизъюнкт не последний;

= R0 — в противном случае.

Параметры команд:

%P1 = na — номер атрибута;

%P2 = ref — ссылка на значение атрибута.

EQ — равно.

or.and	EQ	%G0	%G1	na	ref
--------	----	-----	-----	----	-----

GT — больше чем.

or.and	GT	%G0	%G1	na	ref
--------	----	-----	-----	----	-----

LT — меньше чем.

or.and	LT	%G0	%G1	na	ref
--------	----	-----	-----	----	-----

И.Е. Сапрыкин, В.И. Терехов

**Инструментальный комплекс автоматизации
информационных технологий:
запросная подсистема**

Ответственный за выпуск С.Г. Попов

Работа поступила 2 июня 1989 г.

Подписано в печать 8.06. 1989 г. МН 10260

Формат бумаги 60×90 1/16 Объем 1,3 печ.л., 1,0 уч.-изд.л.

Тираж 200 экз. Бесплатно. Заказ № 83

*Набрано в автоматизированной системе на базе фото-
наборного автомата ФА1000 и ЭВМ «Электроника» и
отпечатано на ротапинтере Института ядерной физики
СО АН СССР,
Новосибирск, 630090, пр. академика Лаврентьева, 11.*