

УДК 681. 142.2

Э. М. БУЗНИЦКАЯ, канд. техн. наук, *В. А. КУДРЯВЦЕВ*

**ОБ ОДНОЙ РЕАЛИЗАЦИИ ЯЗЫКА ЗАПРОСОВ В СИСТЕМЕ УПРАВЛЕНИЯ
БАЗОЙ ДАННЫХ**

Современная система управления базами данных (СУБД) должна быть способна ответить на любой неожиданный запрос, не требуя перестройки структуры данных и не влияя на предыдущие запросы. Она дает возможность пользователям-

непрограммистам работать непосредственно с базой данных, т. е. удовлетворять большой процент их потребностей без консультаций с программистом. Отсюда вытекает необходимость развития непроцедурного языка высокого уровня, который позволит неподготовленному пользователю непосредственно и эффективно работать с базой данных. Чтобы удовлетворить свои информационные потребности, пользователь с помощью этого языка формирует СУБД свой запрос. Непроцедурный язык высокого уровня будем называть языком запросов (ЯЗ).

Разработано большое количество языков запросов. Реализация их в конкретной СУБД в значительной степени зависит от используемой в системе модели данных. Существуют три подхода к построению модели [1]: иерархический, сетевой, реляционный. Рассматриваемая СУБД автоматизированной информационно-поисковой системы кадры — табельный учет с использованием дисплеев (АИСКТД) использует иерархическую модель данных. При проектировании и реализации ЯЗ разработчики руководствовались принципом предоставления максимума удобств пользователям системы. Анализ опыта функционирования СУБД различного назначения позволил конкретизировать группы работающих с системой пользователей.

Наиболее приемлемой принято считать систему с четырьмя уровнями пользователей: администратор системы; пользователь — административный работник; программист-прикладник; пользователь-параметрист (например, оператор ЭВМ).

Функции этих типов пользователей хорошо известны в существующей практике. Однако перечень выполняемых ими работ в значительной мере зависит от конкретной СУБД и реализованного в ней языка запросов. В некоторых системах используются несколько языков запросов, каждый из которых ориентирован на определенный уровень пользователей, что ведет к значительному удорожанию стоимости и увеличению сроков ее реализации. Кроме того, пользователям любого уровня свойственно быстрое освоение заложенных в ЯЗ возможностей (для данного уровня). Следствием этого процесса является повышение потребностей пользователей. Неизбежно наступает такой момент, когда для их удовлетворения необходимо выполнить расширение языковых возможностей системы. Большинство наиболее известных СУБД содержат в составе средства расширения возможностей. Однако, как правило, они выполняются на каком-нибудь одном уровне пользователя. Обычно это уровень программиста-прикладника.

Предпринята попытка разработать и реализовать язык запросов, способный удовлетворить запросы различных уровней пользователей и обладающего средствами координированного расширения своих возможностей. Общий вид ЯЗ представляется четырьмя уровнями, каждый из которых соответствует своему уровню пользователей.

Первый уровень (самый низший) состоит из набора элементарных функций, представляющих собой операторы — расширения базового языка программирования *PL/1*, который был использован при реализации АИСКТД. Эти функции используются любой прикладной программой и позволяют программистам модифицировать базу данных и (или) выдавать интересующую его информацию. Возможности имеющихся функций таковы: инициализация файлов базы данных; замена, добавление и удаление элементов данных, а также групп элементов; выдача информации о хранимом определении данных; выдача значений элементов базы данных. При использовании функций низшего уровня должны выдерживаться все спецификации и ограничения, накладываемые конкретной реализацией АИСКТД.

Каждый последующий уровень ЯЗ представляет собой надстройку над возможностями низшего уровня. Поскольку расширение возможной системы выполняется всегда сначала на низшем уровне ЯЗ (путем программирования новых функций), а потом включается в набор операторов высшего уровня, такой подход к реализации ЯЗ оказывается более предпочтительным. При реализации многоуровневого ЯЗ нужно следовать основному требованию: в каждом высшем уровне ЯЗ должны быть предусмотрены средства определения возможностей низшего уровня, причем они должны быть обязательно доступны администратору системы, под руководством которого выполняется расширение возможностей системы.

Второй уровень ЯЗ можно назвать уровнем неподготовленного пользователя. К этому уровню относятся административные работники предприятия — самый главный тип пользователей, для которых и разрабатываются все возможности СУБД. Основное назначение ЯЗ данного уровня — оперативное обеспечение достоверной информации в требуемом пользователями виде. Средства указанного уровня включают операторы поиска и редактирования данных. В некоторых случаях пользователям могут представляться средства манипулирования данными, хотя функционирование подобных систем в нашей стране показывает, что упомянутые выше функции должны быть в ведении администратора системы (или находиться под его наблюдением).

Третий уровень используемого ЯЗ в системе АИСКТД ориентирован на пользователя-параметриста, к которому можно отнести и оператора ЭВМ, работающего с системой в пакетном режиме. Взаимодействие пользователей данного уровня с базой данных состоит в вызове предварительно определенных команд, а также в присваивании значений некоторым их параметрам.

Четвертый уровень ЯЗ ориентирован на администратора системы, несущего ответственность за различные стороны использования базы данных. В первую очередь это относится к ее первоначальному созданию и выполнению в ней любых структур-

ных изменений, в которых может возникнуть необходимость в ходе ее эксплуатации. Средства ЯЗ этого уровня включают набор всех возможностей системы. Сюда включено такое мощное средство, как ограничение языковых возможностей отдельных уровней, что существенно для ограничения несанкционированного доступа к некоторым функциям или данным в разрезе этих функций для некоторых пользователей уровня административных работников, а также перераспределение возможностей системы между уровнями пользователей по усмотрению администратора системы. Работа с указанными средствами осуществляется как на стадии предкомпиляции компонентов системы АИСКТД, так и на стадии выполнения системой запрошенных пользователем функций. Подготовка информации и выполнение процесса предкомпиляции достаточно изучены в литературе и зависят от реализации конкретной версии СУБД.

Поэтому в данной работе такие вопросы не рассматриваются.

Грамматика ЯЗ описана с помощью бэкусовской нормальной формы и представлена в таблице. Структура таблицы определена логической последовательностью алгоритма компиляции.

С семантической точки зрения ЯЗ соответствует подмножеству языка исчисления предикатов, что позволяет строить поисковые предписания любой логической сложности [2].

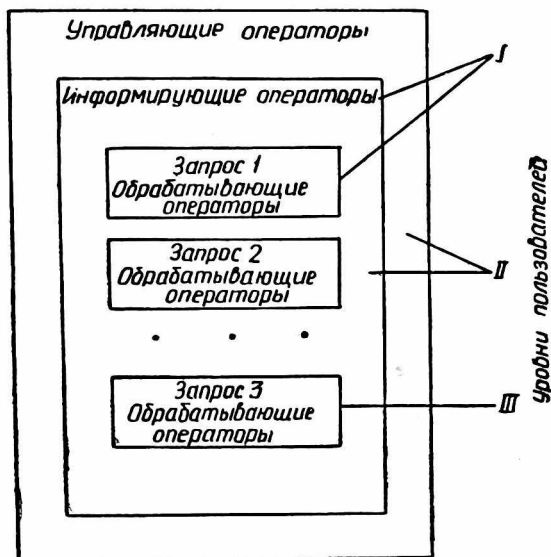
Операторы ЯЗ разбиты на три типа: управляющие, информирующие, обрабатывающие.

С помощью управляющих операторов системы сообщается вся необходимая информация для настройки системы на выполнение той или иной функции. Операторы этого типа используются администратором системы в диалоговом режиме для составления и отладки запросов по плановой статистической отчетности. После отладки эти запросы помещаются в специальную библиотеку запросов. При пакетной обработке группа операторов может использоваться пользователем — оператором ЭВМ с привлечением средств языка управления заданиями.

Информирующие операторы служат для связывания пользователей с конкретными файлами базы данных (через хранимые в системе описания данных), а также описывают информацию уточняющую внешний вид и особенности печати отчетов, запрашиваемых пользователями. С помощью операторов данной группы пользователь может задавать формат печатаемого отчета, размер листа, управлять печатью верхней шапки, подписями, начинать печатать с любого листа, задавать количество печатаемых листов, задавать условия печати информации в верхней шапке отчета и т. п.

Основной массе пользователей-непрограммистов иногда необходимо знать и использовать возможности указанных группы операторов. Система имеет средства задания этих операторов на более высоком уровне пользователей. В основ

средств положен принцип умолчания. Администратор системы может сгенерировать любую версию системы с заданием стандартного значения конкретному информирующему оператору. Если в процессе диалога пользователя с системой в обрабатываемом запросе будет отсутствовать один из требуемых по контексту информирующих операторов, то его значение будет принято по умолчанию (оно будет равно значению, установленному при генерации). Если же информирующий



оператор не задан ни при генерации, ни в самом запросе, то система выдаст пользователю требование на его задание.

Кроме перечисленных выше возможностей информирующие операторы могут задаваться на уровне оператора ЭВМ. Тогда они устанавливают значения этим операторам только на время данного вызова системы (переопределению могут подлежать и значения операторов, установленные при генерации системы).

На рисунке показана область действия значений информирующих операторов и указаны уровни пользователей, на которых они устанавливаются: 1 — уровень администратора системы (область действия — до следующей регенерации); 2 — уровень оператора ЭВМ (область действия — время выполнения шага задания, в котором вызвана система АИСКТД); 3 — уровень пользователя-административного работника (область действия — один запрос).

Обрабатывающие операторы задают условия отбора информации при выполнении указанной пользователем функции. Они

описывают также форму и основные характеристики выходных данных при построении отчетов. Эта группа операторов — основное средство общения с системой пользователей — административных работников подразделений предприятия. Структура и формат этих операторов подбирались с особой тщательностью, что позволяет пользователям данного уровня легко и быстро составлять запросы на получение достаточно сложных отчетов.

При таком подходе к распределению языковых возможностей системы АИСКТД основной контингент пользователей-непрограммистов при составлении запроса акцентирует свое внимание на содержательной стороне описания требуемого отчета. Задание оформительской части описания отчета перекладывается либо на систему (по принципу умолчания), либо на высшие уровни пользователей.

Использованный в системе подход к реализации ЯЗ позволяет в минимально возможный срок пользователям основного уровня освоить возможности системы АИСКТД.

В ходе накопления пользователями опыта работы с системой можно выполнять перераспределение средств использования возможностей ЯЗ между уровнями пользователей.

В разработанной системе АИСКТД применяются специальные средства ограничения доступа как к самим данным БД, так и к языковым возможностям системы. Поэтому администратором системы ведется специальный файл обеспечения доступа, где для каждого конкретного пользователя указываются доступные для него режимы функционирования (что соответствует заданию тех или иных операторов ЯЗ), файлы и данные. Поскольку связь с системой устанавливается посредством задания пользователем своего индивидуального кода, система автоматически способна на основании информации указанного выше файла сформулировать управляющие и информирующие операторы.

В процессе функционирования системы в файл обеспечения доступа можно добавлять новых пользователей или модифицировать информацию по имеющимся в нем пользователям, сужая или расширяя область их доступа.

В целях более удобного взаимодействия пользователей с системой для каждого уровня пользователей используется своя форма записи запроса.

Язык уровня пользователя-оператора ЭВМ представляется в форме ключевых слов или в форме разделителей.

Пользователи уровней администратора системы и административных работников используют форму фиксированных позиций. При составлении запросов по такой форме каждый элемент языка появляется в фиксированной позиции входного носителя запроса. Поскольку основной тип пользователей системы — административные работники предприятия, которые привыкли в повседневной работе заполнять и обрабатывать печатные фор-

- 1 символ : : = буквенно—цифровой—символ | пробел | ' | = | + | — | * |
| / | (|) | ; | . | : | | < | > | ! | % | & | @ | # | } |
- 2 буквенно—цифровой—символ : : = буква | десятичная—цифра
- 3 буква : : = A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | R | Q |
| S | T | U | V | W | X | Y | Z | Б | Г | Д | Ж | З | И | Й | Л | П | У | Ф | Ц | Ч |
| Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я
- 4 пробел : : = _
- 5 разделитель : : = , | . | ; | : | = | знак—операции | заполнитель }
| ограничитель
- 6 знак—операции : : = знак—арифметической—операции | знак—опе-
рации—сравнения | знак—логической—операции
- 7 знак—арифметической—операции : : = * | / | + | —
- 8 знак—операции—сравнения : : = > | > = | = | < = | < | > | не > |
| > = | > < | не <
- 9 знак—логической—операции : : = И | ИЛИ
- 10 заполнитель : : = пробел ...
- 11 ограничитель : : = ' | скобка начало—сложного—условия | конец—
сложного—условия
- 12 скобка : : = (|)
- 13 начало—сложного—условия : : = десятичная цифра ...
- 14 конец—сложного—условия : : = десятичная цифра ...
- 15 десятичная—цифра : : = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
- 16 идентификатор : : = { [& | @ | # | } | } | буквенно—цифровой—символ.
... } | { буквенно—цифровой—символ... } :
- 17 константа : : = арифметическая—константа | строковая—константа
- 18 арифметическая—константа : : = константа—с—фиксированной—точкой
- 19 константа—с—фиксированной—точкой : : = десятичное—число
- 20 десятичное—число : : = целое—десятичное—число [.] | [целое—де-
сятичное—число] . целое—десятичное—число
- 21 целое—десятичное—число : : = десятичная—цифра ...
- 22 строковая—константа : : = строка—символов
- 23 строка—символов : : = символ ...
- 24 выражение : : = [выражение ИЛИ] выражение—5
- 25 выражение—5 : : = [выражение—5] И выражение—4
- 26 выражение—4 : : = [выражение—4 знак—операции—сравнения] выра-
жение—3
- 27 выражение—3 : : = [выражение—3 { + | — }] выражение—2
- 28 выражение—2 : : = [выражение—2 { * | / }] выражение—1
- 29 выражение—1 : : = простое—выражение
- 30 простое—выражение : : = (выражение) | начало—сложного—условия
выражение конец—сложного—условия | имя | константа | вызов—под-
программы—пользователя | указатель—функции
- 31 имя : : = имя—элемента—данного [(целое—десятичное—число [, це-
лое—десятичное число ...] ...)]
- 32 имя—элемента—данного : : = [& | @ | ≠ | } | } | буквенно—цифровой—
символ ...
- 33 вызов—подпрограммы—пользователя : : = ВЫХОД номер—подпро-
граммы—пользователя
- 34 номер—подпрограммы—пользователя : : = 1 | 2 | 3 | 4 | 5
- 35 указатель—функции : : = имя—функции [(список—аргументов)]
- 36 имя—функции : : = } буквенно—цифровой—символ ...
- 37 список аргументов : : = аргумент [, аргумент] ...
- 38 аргумент : : = константа | имя
- 39 оператор : : = управляющий—оператор | информирующий—оператор |
обрабатывающий—оператор

№№ п/п	Грамматика языка запросов
40	управляющий—оператор : : = оператор—ЗАПРОС оператор—RETCN
41	оператор—ЗАПРОС : : = ЗАПРОС = имя—запроса
42	имя—запроса : : = идентификатор
43	оператор—RETCN : : = RETCN = { ФОРМ [, A] ДОЗАП [, A] ОБНОВ [, A] ВЕДОМ СПИСОК [, { S T } { A] РЕСТР КОЛИЧ СОРТ }
44	информирующий—оператор : : = оператор—TIPM оператор—LIST оператор—NLIST оператор—CONTR оператор—SORTK оператор—SHAP оператор—NSHAP оператор—KONSH оператор—KEVR оператор—NREP оператор—ARXIB оператор—MDOZAP оператор—ЗАГЛАВИЕ оператор—КОНЦОВКА оператор—KLIST
45	оператор—TIPM : : = TIRM = номер—файла [, V]
46	номер—файла : : = десятичная—цифра
47	оператор—LIST : : = LIST = размер—листа
48	размер—листа : : = целое—десятичное—число
49	оператор—NLIST : : = NLIST = номер—листа
50	номер—листа : : = целое—десятичное—число
51	оператор—CONTR : : = CONTR = имя—элемента—данного
52	оператор—SORTK : : = SORTK = ДА НЕТ
53	оператор—SHAP : : = SHAP = СТАНД НЕСТД
54	оператор—NSHAP : : = NSHAP = ПК ПМ ЛВ
55	оператор—KONSH : : = KONSH = ДА
56	оператор—KEVR : : = KEVR = ДА
57	оператор—NREP : : = NREP = НЕТ
58	оператор—ARXIB : : = ARXIB = ДА [, { S T } { A]]
59	оператор—MDOZAP : : = MDOZAP = номер—массива—дозаписи [, длина—блока—массива]
60	номер—массива—дозаписи : : = 1 2 3 4 5 6
61	длина—блока—массива : : = целое—десятичное—число
62	оператор—ЗАГЛАВИЕ : : = ЗАГЛАВИЕ = литерал—заглавия
63	оператор—КОНЦОВКА : : = КОНЦОВКА = литерал—концовки
64	оператор—KLIST : : = KLIST = целое—десятичное—число
65	обрабатывающий—оператор : : = оператор—отбора—записей оператор—построения—граф оператор—построения—строки
66	оператор—ШП : : = ШП : выражение
67	оператор—отбора—записей : : = оператор—ШП оператор—УО оператор—УИ оператор—УД оператор—УУ
68	оператор—УО : : = УО : выражение [выражение ...]
69	оператор—УИ : : = УИ : выражение ; имя—элемента—данного = константа [{ имя—элемента—данного = константа } ...]
70	оператор—УД : : = УД : выражение
71	оператор—УУ : : = УУ : выражение
72	оператор—построения—граф : : = ОГ : ТЕКСТ = целое—десятичное—число условие—формирования—графы { * КОЛ } [{ условие—формирования—графы { * КОЛ } } ...]
73	условие—формирования—графы : : = имя—элемента—данного = { X K + выражение идентификатор }
74	оператор—построения—строки : : = номер—строки—отчета : { выражение КОЛ ТЕКСТ = строковая—константа [{ ТЕКСТ = строковая—константа } ...]
75	номер—строки—отчета : : = целое—десятичное—число
76	библиотечный—запрос : : = оператор—ЗАПРОС запрос [КЗК описание—верхней—шапки КЗП]

№№ п/п	Грамматика языка запроса
77	запрос : : = управляющий—оператор ... [информирующий—оператор ...] [обработывающий—оператор ...]
78	описание—верхней—шапки : : = [{ H строка—описания—шапки [K строка—описания—шапки } ...] H описание—разделителей—граф [K описание—разделителей—граф]
79	строка—описания—шапки : : = [символ ...] [символ ...] [' имя—элемента—данного '] [символ ...]
80	описание—разделителей—граф : : = [символ ...] разделитель—графы [{ символ ... } разделитель—графы [символ ...] } ... разделитель—графы [символ ...]
81	разделитель—графы : : = !
82	отчет : : = итоговый—блок заглавный—блок верхняя—шапка тело—отчета итоговый—блок
83	итоговый—блок : : = линия—интервала итоговая—линия литерал—концовка
84	итоговая—линия : : = ИТОГО ПО ПРЕДПРИЯТИЮ целое—десятичное—число
85	заглавный—блок : : = штриховая—линия линия—интервала линия—заглавия линия—дата—лист
86	штриховая—линия : : = - . . .
87	линия—интервала : : = [символ ...]
88	линия—заглавия : : = КОД ЗАПРОСА имя—запроса литерал—заглавия
89	литерал—заглавия : : = строковая—константа
90	литерал—концовки : : = строковая—константа
91	линия—дата—лист : : = ДАТА число—года. месяц—года. год [символ ...] ЛИСТ номер—листа [символ ...]
92	верхняя—шапка : : = штриховая—линия—шапки тело—верхней—шапки штриховая—линия—шапки
93	штриховая—линия—шапки : : = [символ ...] - . . . [символ ...]
94	тело—верхней—шапки : : = системная—шапка шапка—пользователя
95	системная—шапка : : = разделитель—графы шапка—графы [{ разделитель—графы шапка—графы } ...] разделитель—графы
96	шапка—графы : : = строка—графы—шапки—1 строка—графы—шапки—2 строки—графы—шапки—3
97	строка—графы—шапки—1 : : = [символ ...] [символ ...] КОД [символ ...] [символ ...] СУММА ПО [символ ...] [символ ...] КОЛ-ВО [символ ...]
98	строка—графы—шапки—2 : : = [символ ...] ТЕКСТ [символ ...] [символ ...] ЭЛЕМЕНТА [символ ...] [символ ...] УСЛОВИЮ [символ ...]
99	строка—графы—шапки—3 : : = [символ ...] [символ ...] имя—элемента—данного [символ ...] [символ ...] диапазон—строка—условия [символ ...]
100	диапазон—строка—условия : : = целое—десятичное—число—целое—десятичное—число
101	шапка—пользователя : : = [строка—шапки ...] строка—разделителей—граф
102	строка—шапки : : = [символ ...] [символ ...] [значение—элемента—данного] [символ ...]
103	значение—элемента—данного : : = буквенно—цифровой—символ ...
104	строка—разделителей—граф : : = [символ ...] разделитель—графы [{ символ ... } разделитель—графы [символ ...] } ... разделитель—графы [символ ...]
105	тело—отчета : : = линия—интервала ... [информационная—строка] ... [текстовая—строка] ... [итоговая—строка] ... [линия интервала] ...

№№ п/п	Грамматик языка запросов
106	информационная—строка . . = [] значение—графы
	[{ [] значение—графы } . . .] []
107	значение—графы : : = буквенно—цифровой—символ . .
108	текстовая—строка : : = [] символ . . . []
109	итоговая—строка : : = [] значение—итога [{ [] зна- чение—итога } . . .] []
110	значение—итога : : = целое—десятичное—число
111	число—года . . = целое—десятичное—число
112	месяц—года . . = целое—десятичное—число
113	год : : = целое—десятичное—число

мы документов, была выбрана форма фиксированных позиций, для которой разработан печатный бланк.

Результаты практического использования системы показывают большую эффективность языка запросов. В течение короткого времени пользователи-непрограммисты осваивают все возможности языковых средств. Использование пакетного и диалогового режима обработки запросов повышает удобство эксплуатации системы (см. таблицу).

Список литературы: 1. *Мартин Дж.* Организация баз данных в вычислительных системах. — М.: Мир, 1980. — 662 с. 2. *Котов Р. Г., Якушин Б. В.* Языки информационных систем. — М.: Наука, 1979. — 304 с. 3. *Овчаров Л. А., Селетков С. Н.* Автоматизированные банки данных. — М.: Финансы и статистика, 1982. — 262 с.

Поступила в редколлегию 15.08.83.