Command use in a relational database system

by JOHN D. JOYCE and DAVID R. WARN
General Motors Research Laboratories
Warren, Michigan

ABSTRACT

A study of commands in a relational database system was undertaken to provide a basis for improving future implementations of relational techniques. The use of Regis (Relational General Information System), an interactive relational database system developed at General Motors Research Laboratories, was monitored to accumulate a large amount of data about distribution of command uses across a variety of applications developed by users with a diverse set of capabilities. Of the basic relational commands, it is important that searching, PROJECTION, and JOIN operations be efficient. However, optimizing INTERSECTION and “exclusive or” may not warrant the time and effort it would require. Although proportions of use varied, the rankings of use of relational operations were reasonably independent of applications and the programming expertise level of the users.
INTRODUCTION

An interactive relational system has been measured to determine relative use of different kinds of commands by industrial users. This information was gathered to gain some insight into where emphasis should be placed for efficient implementation of future relational systems, either in hardware or software. Data have been acquired over a 6-month period by monitoring all completed sessions on one computer system.

Descriptions of the salient characteristics of the system, the command language, the monitoring facilities, and the major applications are presented in following sections. The characteristics of command use will be presented by command categories and by application area.

REGIS DATABASE SYSTEM

Regis (Relational General Information System) is an interactive relational database system developed at the General Motors Research Laboratories. Since 1975 it has been in production use by many GM divisions and staffs. The system is designed to provide convenient, powerful, and flexible information manipulation facilities for information storage, retrieval, and analysis. All data are represented in a simple tabular form to provide an easily understandable and easily manipulated view of data. Graphical display facilities to plot the data and statistical functions to analyze the data are also incorporated in the package.

Regis is designed for the interactive user who asks questions, receives answers, and then asks further questions based on the answers. It is particularly well suited to applications in which the queries cannot be defined in advance. Regis, however, also provides the ability to package commands in parameterized command files with sufficient logic control commands to permit the development of sophisticated applications.

HIGHLIGHTS OF THE COMMAND LANGUAGE

Regis uses an interpreted algebraic command language. The command name appears as the first word of each command except when the results of the command are stored in a table different from the source table. In that case the command name is preceded by “table_name = ”.

There are in total 66 commands. These have been grouped into six major categories for analysis in this paper.

Basic Relational Commands

Eight basic relational commands were selected as the focus for analysis in this paper because they correspond most closely to the relational operations described in the literature. This group consists of SUBSET, PROJECTION, JOIN, JOIN-ALL, UNION, INTERSECTION, DIFFERENCE, and XOR. An explanation of how these commands relate to, or differ from, relational terminology will be given.

The SUBSET command selects rows from a table using a boolean expression of column relationships to be satisfied by the selected rows. An extensive set of pattern-matching functions is provided for text columns in addition to the typical functions for numeric relationships. The selected rows are placed in a result table, which has the same form (columns) as the source table. The SUBSET command is sometimes referred to in the literature as “selection” or “restriction.” SUBSET is a “search” command.

The PROJECTION command selects specified columns from a table. There are options for reordering of columns, sorting of rows and eliminating duplicate rows.

JOIN and JOINALL combine columns from two source tables into a new target table where values match in specified columns. The resulting table will contain all columns from both source tables, except that the columns being matched occur only once. There are a number of variations in terminology in the literature describing various facets of the join capability. The Regis implementation of JOIN is usually referred to as a natural join. A minor extension is that the named columns (domains or attributes) that are being matched to perform the join operation in the two source tables need not have the same names, although the data types must match. The theta-join operations (where theta can be any of the comparison operators =, >, <, ...) are not supported, except for =. M-way joins where M > 2 are also not supported directly. Users can, of course, repeatedly apply a join operator to the results of a previous “join” to simulate an M-way join. Neither the users nor the implementors during seven years of use have perceived much need for these kinds of additional primitive operations. M-way joins would be a convenience at times, however.

A different extension of join, however, was often requested by users. The JOINALL command is an implementation of the generalized symmetric natural join as defined by Lacroix and Pirotte. It differs from this natural join in cases where a match does not exist on the columns being joined; the target row (tuple) will be augmented by null values for the columns of the appropriate source table.

DIFFERENCE, INTERSECTION, UNION, and XOR (exclusive or, i.e., symmetric difference) perform the normal set operations on two tables. The Regis implementation permits the comparisons to be made on the entire row (all columns) or on selected columns whose values are to be matched. UNION and XOR require both tables to have the same form for all columns. DIFFERENCE and INTER-
SECTION always take rows from the first source table only so that the second source table has only to have the same data types for the columns whose values are to be matched. There can be any number or type of additional columns in the second source table.

Logic Control Commands

The logic control commands provide the ability to package Regis commands in a command file with optional parameters. The command procedures can be used to implement new functions or to provide a complete application system. The commands include conditional execution, repeated execution of a group of commands, parameter definition, and local variables. An extensive set of built-in functions complements these commands and provides string manipulation, access to table, column, and row information, arithmetic operations, and so on.

Analysis and Modification Commands

A variety of commands are provided to analyze and modify tables. Table modification functions include sorting a table on one or more columns, performing arithmetic on columns, replacing particular values in a column, appending one table to another, and transposing a table to interchange rows and columns. Analysis functions are available to generate statistics such as minimum, maximum, average, count, sum, and standard deviation for numerical columns of a table. These statistics can be obtained for all rows of a table or for groups of rows within a table where each group has equal values in one or more selected columns. Curves for trend prediction can be fitted to data using polynomial or least-squares fit. Other commands are available to perform multivariate regression analysis and to generate bivariate multiple correlation coefficients.

Table and File-Handling Commands

The table-handling commands provide the ability to create, delete, or rename tables or columns and to add or delete rows or columns. File functions include reading, writing, releasing and backing up files.

Printed and Graphical-Output Commands

A variety of output commands are provided for both printed and graphical formats. Printed output ranges from a simple listing of a table at the terminal to a completely formatted report. Simple bar charts and plots may also be printed. Graphical output is produced through a tightly coupled interface to SIMON (a GMR-developed interactive graph-plotting package) and a command interface with data transfer by a disk file to a commercial interactive package. These packages can be used to produce high-quality graphs from data contained in Regis tables. Other output commands display messages to the terminal, clear the terminal display and control pagination of printed output.

Miscellaneous Commands

Other commands are provided to display debugging information about tables and files, set Regis default values, invoke DO-IT Menu System, establish command synonyms, and terminate a Regis session. A HELP command will display the syntax for any command, and a NEWS command will display announcements for the users. An OBEY command can be used to issue operating system commands without leaving Regis.

Instrumentation Capabilities

Monitoring capabilities had previously been incorporated in Regis to provide information on every Regis session. A 1-row table is created to describe a session: the user ID, total commands, total central processing unit (cpu) time, total elapsed time, and so on. For every session that terminates normally, this table is written to a file that is periodically analyzed to provide a global summary of Regis use.

For purposes of the current study, the command interpreter was modified to maintain an array containing the number of times each command is executed during a session. At the end of each session the array of command counts was put into a table and written out along with the session data. These data were collected for all Regis sessions on one computer over a 6-month period.

Characteristics of Interactive Usage

As background for the study on relational command use, some characteristics of the interactive sessions will be outlined. The distribution of the durations of the on-line interactive sessions is shown in Figure 1. There is a high proportion
of sessions in which the total duration is much shorter than expected. We theorize that either there are numerous short sessions for the purposes of updating databases with small amounts of data, or there are numerous requests for small amounts of information.

The number of commands used per session varies greatly because commands are used in two basic modes. One mode is to key in each command at an interactive terminal for immediate execution. In this mode a relatively small number of commands are issued in an interactive session. In the second mode commands are executed from command files. Some user applications rely heavily on prepackaged command files. Between the two extremes, there is a fairly smooth distribution. (See Figure 2.) To some extent this gives an indication of the degree to which ad hoc and preplanned procedures are used.

USE OF BASIC RELATIONAL COMMANDS

Use Across All Applications

Regis applications come from a variety of areas, including workload scheduling, engineering tests, quality control, machine failure analysis, and analysis of business operations. For most long-term applications, users have developed extensive collections of command files to handle routine database functions and produce standard reports. In addition, two major functions of Regis have been implemented in command files. A composite analysis of all command executions will be presented first.

The data in Figure 3 include all the sessions that were monitored beginning December 1980 through May 1981. The TOTALS value for the number of sessions column in Figure 3 is the total number of sessions that were monitored; it is not the sum of the number-of-sessions entries tabulated for each command. Clearly, searching of tables (SUBSET) is the most heavily used relational function. PROJECTION and JOIN are also very important in the total of uses. UNION, DIFFERENCE, and JOINALL are used considerably less often, but are still important capabilities for a number of applications. Both the XOR command and the INTERSECTION command could reasonably be implemented as command files of UNION or DIFFERENCE commands.

Relational Command Usage in Two Regis Extensions

The next analysis focuses on relational commands used in providing two major extensions to Regis, the Regis Report Generator and the DO-IT Menu System. Both are implemented with Regis command files. For that reason, the use of either facility could potentially distort the interactive characteristics and distribution of command uses for a session, because many commands are executed for each command that is issued by a user.

The Regis Report Generator provides controls to format reports from Regis tables for printing or display at a terminal. This facility has proven to be very popular with users for preparing reports suitable for presentation or publication. The DO-IT Menu System provides a menu interface to almost all Regis commands. Instead of having to learn or remember the syntax of commands, users make selections from a menu. The capabilities of the system are evident in the choices displayed and there is far less typing involved. A custom-menu facility provides the ability to create, modify, and execute specialized menus adapted to the needs and terminology of an application.

Regis commands were used to implement these extensions because it took much less effort than it would to use a conventional programming language such as PL/I. Sessions that used either of these extensions were separated for analysis to determine whether their characteristics or the programming experience of the implementors would make any significant differences in the uses distribution of relational commands. User application commands may also be present in these sessions, but it is assumed that the two major Regis extensions dominate these data.

The distribution of uses of basic relational commands for these two extensions is quite similar to the distribution for all

<table>
<thead>
<tr>
<th>COMMAND NAME</th>
<th>NUMBER OF SESSIONS</th>
<th>NUMBER OF COMMANDS</th>
<th>PERCENT OF RELATIONAL COMMANDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSET</td>
<td>1 901</td>
<td>182 213</td>
<td>72.4</td>
</tr>
<tr>
<td>PROJECTION</td>
<td>1 623</td>
<td>38 990</td>
<td>15.5</td>
</tr>
<tr>
<td>JOIN</td>
<td>836</td>
<td>18 601</td>
<td>7.4</td>
</tr>
<tr>
<td>UNION</td>
<td>568</td>
<td>5 460</td>
<td>2.2</td>
</tr>
<tr>
<td>DIFFERENCE</td>
<td>477</td>
<td>3 498</td>
<td>1.4</td>
</tr>
<tr>
<td>JOINALL</td>
<td>190</td>
<td>2 410</td>
<td>1.0</td>
</tr>
<tr>
<td>INTERSECTION</td>
<td>27</td>
<td>275</td>
<td>0.1</td>
</tr>
<tr>
<td>XOR</td>
<td>1</td>
<td>1</td>
<td>0.0</td>
</tr>
</tbody>
</table>

TOTAL NUMBER OF ALL COMMANDS FOR THESE SESSIONS = 14 094 675

Figure 3—Basic relational commands sorted by usage
The order of frequency of use of relational commands is identical. See Figure 4.

**Analysis of User Applications**

The next analysis includes only the sessions that did not have any DO-IT menu or Regis report generator commands. These results are shown in Figure 5. These data would seem to indicate that users in normal user applications do more searching of tables than is done in the Regis extensions and that fewer PROJECTION operations are done. However, subsequent examination of some of the major applications shows that there is sufficient variability in the use of relational commands that this conclusion would not be valid. One can also observe that both the JOIN and JOINALL operations are used much more than any of the INTERSECTION, UNION, or XOR operations. For these user applications, the latter three commands are little used. Thus, nonoptimized implementations of these three commands would be tolerable.

Four of the largest user applications have also been summarized. For each application, there is a group of anywhere from 5 to 15 users so that no application represents the characteristics of any individual. Application A is an application run by manufacturing people who are tracking quality control for the manufacture of small parts. Application B is the monitoring of trends of failures in major automotive assemblies in another manufacturing division. Application C is the analysis of sales forecasts to make plans periodically for future capital, tooling, space, and other plant requirements for component manufacturing. Application D consists of monitoring project schedules, computing effects on workloads when plans for new project target dates are shifted, and minimizing overtime of a variety of skilled people working on various facets of these projects.

The variations in use of relational commands are portrayed in Figure 6. Note that the applications A, B, C, and D include some of the menu and report generator commands summarized in the second column. The four applications listed separately account for 41.6% of all the monitored commands. In examining individual applications, the trend of high use of SUBSET (searching) continues across all applications. In most cases, PROJECTION is the next most used operation. The proportion of SUBSET to PROJECTION operations varies considerably from one application to another. The JOIN operation is generally the third most popular relational command. In the six summaries shown in Figure 6, the first three operations account for a low of 88.8% to a high of 99.9% of the basic relational commands in use. Generally, the next three operations, UNION, DIFFERENCE, and JOINALL are used rather sparingly, though enough to be an important part of many applications.

**USE OF ALL COMMANDS**

Command use by categories has been summarized in Figure 7. Use of prepackaged command files built either by users for their own applications or general functions provided by the implementors dominates the distribution of all commands executed. Seventy-nine percent of the commands executed are logic control commands, which can only be executed from command files. The single most used command is an IF command to do conditional testing, which constitutes over 34% of command frequency.

In prior studies of computer execution time resource use,
we observed that the logic control commands make up only a small percentage of the total for an application. The basic relational commands category and the analysis and modification categories constitute the heaviest resource use. Implementors need to pay close attention to the efficiency of the commands in these two categories. Even though their frequency is low, they place a heavy load on the system.

Although some of the database input/output (I/O) operations are represented by the commands in the table and file handling category, most of the actual I/O is carried out directly and implicitly by the operating system through virtual-memory paging operations. Data are referenced directly in virtual memory. This has proven to be a fast and efficient mechanism for accessing data, since it uses an optimized path through the operating system and avoids continually having to manage record buffers and copy data to and from buffer areas. This indicates why, in the case of Regis, the frequency of I/O commands is relatively small. One normally expects a database system to have vast numbers of conventional I/O operations, although they still might be buried inside other functions with explicit subroutine calls and not appear as explicit I/O commands.

REFERENCES
