Research Product 85-27

Soldier's Guidebook to INGRES for Advanced Technology Unit Training and Management System (ATUTMS)

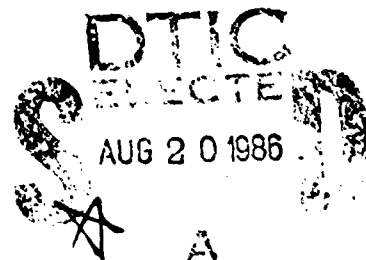ARI Field Unit at Presidio of Monterey, California
Training Research Laboratory

July 1985

DTIC
ELECTE
AUG 2 0 1986

ari

86   8   20   035

U. S. Army Research Institute for the Behavioral and Social Sciences

U. S. ARMY RESEARCH INSTITUTE
FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the
Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

WM. DARRYL HENDERSON
COL, IN
Commanding

Technical review by

Mr. Thomas Antczak
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

LTC N. Harris
1/11 Field Artillery Battalion
Fort Lewis, Washington 98433

NOTICES

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARI Research Product 85-27	**2. GOVT ACCESSION NO.** AD.A170 928	**3. RECIPIENT'S CATALOG NUMBER**
4. TITLE *(and Subtitle)* SOLDIER'S GUIDEBOOK TO INGRES FOR ADVANCED TECHNOLOGY UNIT TRAINING AND MANAGEMENT SYSTEM (ATUTMS)		**5. TYPE OF REPORT & PERIOD COVERED** Final Report July-September 1984
		6. PERFORMING ORG. REPORT NUMBER --
7. AUTHOR(s) Hamza, A. N.		**8. CONTRACT OR GRANT NUMBER(s)** --
9. PERFORMING ORGANIZATION NAME AND ADDRESS U.S. Army Research Institute Field Unit P.O. Box 5787 Presidio of Monterey, CA 93944-5011		**10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS** 2Q263743A794 Task 6610 4413 100
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Research Institute for the Behavioral and Social Sciences 5001 Eisenhower Avenue, Alexandria, VA 22333-5600		**12. REPORT DATE** July 1985
		13. NUMBER OF PAGES 38
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)* --		**15. SECURITY CLASS. *(of this report)*** Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE --

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

--

18. SUPPLEMENTARY NOTES

--

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*
Data Base Management System (DBMS)
INGRES
Advanced Technology Unit Training and Management System (ATUTMS)

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*
This guidebook on INGRES, a relational database management system (DBMS), provides instruction on the main functions of the system, which involve creating a database and tables, designing forms for those tables, appending, updating, and retrieving data, and outputting data in the form of a report. The report guides users through the procedures required to create and manage an INGRES database. Guidebook is specifically tailored for low- to medium-level instruction for soldiers on the use of the Advanced Unit Training and Management System (ATUTMS).

DD FORM 1473 JAN 73   EDITION OF 1 NOV 65 IS OBSOLETE

Soldier's Guidebook to INGRES for Advanced Technology Unit Training and Management System (ATUTMS)

Ann N. Hamza

Jack Hiller, Chief
ARI Field Unit at Presidio of Monterey, California

**Training Research Laboratory
Harold F. O'Neil, Jr., Director**

Education and Training

Research and development on combat unit training management by the Presidio of Monterey Field Unit developed a prototype Battalion Training Management System, referred to as the Advanced Technology Unit Training and Management System (ATUTMS). ATUTMS was developed and installed on site at the 1/11 Field Artillery Battalion, Fort Lewis, Washington.

A relational Data Base Management System (DBMS) was used to develop the database which, by design, integrated data on personnel, training, and logistics. A working knowledge of the relational DBMS is required to operate the system (i.e., data entry, updates, queries, and requests). Documentation is available to describe and explain in detail how to make full use of the DBMS. For most uses, however, that level of technical information is not needed; and this Guidebook is designed to present, in convenient form, sufficient information for more routine use of the database.

EDGAR M. JOHNSON
Technical Director

SOLDIER'S GUIDEBOOK TO INGRES FOR ADVANCED TECHNOLOGY UNIT TRAINING
AND MANAGEMENT SYSTEM (ATUTMS)

CONTENTS
_____

CONTENTS (Continued)

A LITTLE ABOUT DATABASES

o   A DATABASE is a computer-based, record-keeping system.
- o   Databases are used to record, maintain data.
- o   Databases are used to speed and simplify the job of getting information.

o   Data may be stored in <u>one</u> <u>or</u> <u>more</u> databases.

Organization of Data Bases

A database can be thought of like a closet. We use closets to store a bunch of "stuff". It's usually convenient for us to have more than one closet, where each closet contains "stuff" that's somewhat related. For example, we have linen closets to store towels, sheets, table cloths, rags, etc. We also have bedroom closets to store shirts, pants, sweaters, jackets, shoes, ties, etc. We also tend to organize the closet so that shirts are placed in one area of the closet, pants next to them, sweaters above the shelf, and shoes below. We divide the closet into organized and related compartments.

Databases are like closets or big storage areas set up in some organized way to put data in (storage) and take it out (retrieval). There's no need to waste time looking for a shoe where we only store sweaters. What happens when the components within the "closet" are not organized? Well, what happens to us when our closets are messy? Not only do we become frustrated, because we cannot find things, but it's also time consuming and inefficient. The same is true with databases. Their whole purpose is to group data which may be placed in one or more tables in a convenient, organized way. Tables are groups of closely related items of data. Tables which share common elements of data may be included in the same database. Tables lacking any relationships to each other would not be included in the same database. The database program we are using is called INGRES. This guide shows you how INGRES lets you store and retrieve data.

Tables

Data in a database is stored in tables. You can think of a table as a square or rectangle which contains data organized in rows and columns. The rows correspond to individual records and the columns correspond to the kinds of data. The organization of a Table in rows and columns (also called fields) is shown below.

```
C           C           C           C
o           o           o           o
l           l           l           l
u           u           u           u
m           m           m           m
n           n           n           n
```

EXAMPLE TABLE

	NAME	SSN	RANK	MOS
Row	Barnes	586XXX474	E5	11B20
Row	Miller	498XXX110	E2	11B10
Row	Johnson	234XXX341	E7	74F40
Row	Terris	987XXX891	E1	76Y10

As we see in the table, a row contains data for an individual entry (Barnes, 586XXX474, E5, 11B20) and the columns contain specific kinds of data (name is one field, ssn is another field, etc.).

Creating/Destroying Databases

Databases are destroyed/created in VMS ("$").

Create a database to practice with while using this guidebook. Use your last name for the name of the database. For example if your last name was Smith, you would type: $ createdb SMITH

$ createdb databasename <return>

    For example: $ createdb SMITH
                      Making new database

$ destroydb databasename <ret>

    For example: $ destroydb SMITH
                      Database SMITH destroyed

The destroy command completely eliminates the entire database—tables and all!

Note: For clarity in specific examples, any information that you will need to fill in is underlined and a darker printed message indicates the computer's response after a command has been entered. Also, bold print within text indicates keyboard entries.

2

Overview of INGRES Functions

This manual describes the five main functions in INGRES. These five areas are (1) query language, (2) visual forms editors, (3) query by form, (4) report by form and (5) report.

QUEL - Query Language

QUEL is the INGRES Query Language. A query language lets you ask the database to display the information you want to see. The main use of QUEL is to retrieve tables. Also it is used to create tables, add rows to a table (append), change information in a table (update), and delete information from a table. QUEL also provides a print command to print a table.

VIFRED - Visual Forms Editors

The VIFRED function creates a form from a table. A form is displayed on the screen with labels and empty spaces for entering data just like a paper form. VIFRED lets you set up a form for entering data on the screen. This form can then be used by QBF for retrieving, appending and updating information.

QBF - Query By Forms

QBF is mainly used to append, retrieve (query), update, and delete information from a table. This is all done using a form set up in VIFRED. QBF is quicker than QUEL for appending individual records (rows) to a table. It lets you see tables one row at a time.

RBF - Report By Form

RBF is used to set up a report of a table. A report is a listing of all rows of a table. You can also tell RBF how to sort the rows.

Report

The report command actually runs and prints out the report designed in RBF, to a file and/or to the terminal screen.

Summary Chart

	QUEL	QBF	RBF	REPORT	VIFRED
CREATE	TABLE • specify table name, columns, data types		• generates a default report • modifies default report		FORM • generates default form • modifies default form
APPEND		• Adds one or more rows to a table			
RETRIEVE	• Pulls (a little) information from many rows. • Searches all rows and pulls out specific information, organizes it, performs calculations, makes comparisons, and produces summary statistics.	• Queries for information from one row. • Displays data one row at a time. May display all columns or selected columns according to form. • Used for "browsing" individual rows.			
UPDATE	• Replace Changes or adds (the same) data for the same column(s) in different rows. • Delete Deletes many rows at one time. This can be dangerous; consider using QBF.	• Update Insert: Change or add to data in one or more fields, one row at a time.			
PRINT	TABLE • Prints the responses to the queries in the form of a table.			• Runs RBF and prints out the report.	

4

QUEL (QUEry Language) is the INGRES query language, which enables you to
retrieve, manage, and maintain data in an INGRES database. A query language
lets you ask the database to display the information you want to see.

There are five main QUEL statements, each beginning with one of the following
commands:

    RETRIEVE (displays the data you ask for)

    APPEND   (adds new rows to a table)

    DELETE   (removes rows from a table)

    REPLACE   (updates data in a row with new values)

    CREATE   (makes new tables)

QUEL also has the following statements:

    PRINT   (prints all the rows of a table)

    HELP   (shows what tables are in the database)

    COPY   (copies data to or from a file)

Since QUEL uses a tedious method of appending to tables, more emphasis will be
placed on appending to tables from QBF.

To get into one of the INGRES Functions:

	Example
INGRES dbname	INGRES SMITH
VIFRED dbname	VIFRED SMITH
QBF dbname	QBF SMITH
RBF dbname	RBF SMITH

NOTE: Remember to use your last name. We will use SMITH as an example.

Accessing INGRES Function

```
$ INGRES dbname
continue
*
```

For example:
```
$ INGRES SMITH <ret>
continue
*
```

5

Creating Tables

To create a table you must specify the table name and its column names. In addition, you must define what type of data will go in each column. There are five main data types:

- o   character string
- o   text
- o   date
- o   integer (no decimal point)
- o   floating point (decimal point)

* create <u>tablename</u> (<u>fieldname1=datatype,fieldname2=datatype,...</u>)
* \g
* continue

The next example creates a table containing data about soldiers allowing 24 characters for the name, 4 characters for grade, 3 characters for mpc (military position code: O=Officer, W=Warrant, E=Enlisted), one integer for gt_score (must be an integer greater than -128 and less than +127), 4 characters for civ_educ, 7 characters for mos, 5 characters for upc (unit processing code: DGLAO=A BTRY, DGLBO=B BTRY, DGLCO=C BTRY, DGLSO=SVC BTRY, DGLTO=HQ BTRY), and one integer for age.

For example:

* create soldier (name=text(24),grade=c4,mpc=c3,
* gt_score=i1,civ_educ=c4,mos=c7,upc=c5,age=i1) \g
* continue

<u>Note</u>: The fields in this example are: name, grade, mpc, gt score, civ_educ, mos, upc, and age. The data types are: text(24), c4, c3, i1, c4, c7, c5, and i1.

Appending to Tables

To put data into this table use the append. If you are going to add a lot of data, consider using QBF (page 20).

* append to <u>tablename</u> (<u>Fieldname1="value",Fieldname2="value",...</u>)
* \g
Executing. . . .
(1 row)
continue
*

For example:

* append to soldier (name="KNIGHT DAVID G",grade="E5",mpc="E",
* gt_score=105,civ_educ="12",mos="13B20",upc="DGLCO",age=27) \g

<u>Note</u>: All the data must be enclosed in double quotes except for numbers that are integers or floating points (e.g., 105 or 10.2).

Printing Tables

* print _tablename_ \g
    This command will print out the table:

* print soldier \g
Executing. . . .

soldier table

name	grade	mpc	gt_sco	civ_ed	mos	upc	age
KNIGHT DAVID G	E5	E	105	12	13B20	DGLCO	27

continue
*

Help

Help may be used to print information about the contents of the database and specific tables within the database.

* help \g
    Lists all tables that exist in the current database.

* help tablename \g
    Gives information about the specified tables.

Queries

Retrieve Statement in QUEL

Retrieve (soldier.all)

So that you do not have to keep typing the full table name, you can abbreviate it with a range statement:

* range of s is soldier \g
* ret (s.all) \g

Notes: "all" retrieves all fields within the table.

Capital or lower case letters may be used for the INGRES commands. The data should always be in capitals.

The following are examples of retrieve statements:

```
* ret (s.name,s.grade,s.civ_educ) where s.grade>"E5" and s.civ_educ="*COLL*"
* \g
```

> The above statement retrieves all soldiers who have a grade greater than E5 and who have some college education.

```
* ret (s.name,s.grade) where s.grade="01" or s.grade="02" \g
```

> The OR statement requires that one of the conditions be met, either s.grade="01" or s.grade="02". This query will retreive the name and grade of those officers in the battalion who are 2LT or 1LT (grade="01" or grade="02").

```
* ret (s.name,s.gt_score) where s.gt_score>100 and s.gt_score<120 \g
```

> The AND statement requires that both the conditions be met. This query will retrieve the name and gt scores that are greater than 100 and less than 120.

```
* ret (s.name,s.gt_score) where s.civ_educ="*COLL*" and s.grade>="E4" and
* s.grade<="E7" \g
```

> This statements asks for the names and gt scores of those whose civilian education includes some college and whose rank is between E4 and E7.

To retrieve from two or more tables, there must a common field among all tables.

```
* range of s is soldier
* range of d is drivers
* ret (s.name,s.grade,s.mos,d.bumper) where d.name=s.name \g
```

> This retrieve statement asks for the names, grades, mos's and vehicle bumper numbers of those soldiers who are drivers.

Replace Statement in QUEL

```
* replace s (grade="E7") where s.name="Brown*" \g
Executing......
(1 row)
continue
*
```

This replace statement will change the value for the column called grade to E7 for the soldier whose last name is Brown.

> Note: The asterisk matches any values beginning with the name Brown.

Delete Statement in QUEL

```
* delete s where s.name="Franklin Thomas Paul" \g
Executing
(1 row)
continue
*
```

   The DELETE command will delete all the fields in the row where
   name="Franklin Thomas Paul" and not just the name itself.


```
* delete s where soldier name="Edwin Michael Jones" \g
```

   Beware! The statement above will delete all the rows in a table
   because both s and soldier are used for the table name within the
   statement. Within a statement you must be consistent in either
   spelling out the full table name each time it is used or abbreviating
   the table name each time it is used.

Copying Tables from/to Files

To move tables from one database to another, use the COPY command. First, you must copy the table from the original database to a file, then you copy into the second database from the file.

For example:
```
* copy bnpersonn ( ) into "DUAØ:[MYDIR]BNPERSONN.DAT" \g
```

This copies the personnel roster into a file in [MYDIR] directory called BNPERSONN.DAT. The empty parentheses means copy everything. The "DUAØ:" is the name of the disk where the file is located.

Now, in the second database, create the table and copy in the data from the file.

```
* copy bnpersonn ( ) from "DUAØ:[MYDIR]BNPERSONN.DAT" \g
```

Editor

o   INGRES lets you type in as many commands as you need before you enter the
    \g to execute them. While you are typing, the commands are saved in a
    workspace. If you make a mistake or want to add something new, INGRES
    lets you edit the workspace. This means you can make changes without
    retyping the whole command.

o   The computer creates a workspace for you when you begin to type text on a
    line. This workspace, or temporary storage area, is called a buffer.

        The editor works with the text that is stored in the buffer – that
        means the most current text. To make changes or add to the text,
        from INGRES, use the editor.

o   For example, you are typing a retrieve statement in INGRES and toward the
    end of the statement you realize you misspelled retrieve and forgot the
    "e" on name, as shown below:

        * ritrieve (s.nam,s.grade,s.gt_score) where <ret>
        * s.gt_score>=100 and s.gt_score<110

o   You cannot back up with the arrow keys, or use the delete to return to the
    first line; you can use the editor to correct the text. Say you just
    completed typing 110 and now realize your errors. Type \e (backslash e)
    and this will appear:

        * \e <ret>
        >>editor
        1 ritrieve (s.nam,s.grade,s.gt_score) where
        *

o   At the INGRES sign *, type \e and press return to complete your command.

o   The editor brings up one line at a time (line mode).
    Each time you press return it brings up the next line of the text until it
    reaches the end of the buffer [EOB].

        * \e <ret>
        >>editor
        1 ritrieve (s.nam,s.grade,s.gt_score) where <ret>
        * <ret>
        2 s.gt_score>=100 and s.gt_score<110 <ret>
        * <ret>
        [EOB]
        * (the asterisk is called the line editing prompt which lets you
        know you started the editor)

o   The line mode is for browse only. Therefore, to modify your text, type
    the CHANGE command (c) to change to full-screen editing.

        * c <ret>

o   The screen clears. The cursor and the [EOB] symbol appear at the upper
    left of the screen. Whatever you type at this point is displayed on the
    screen before the cursor. Whatever you typed in (or what ever was stored
    in the buffer) is displayed on the screen. The cursor will rest on the
    first letter. The [EOB] symbol always appears after the last character in
    the text buffer. It stands for End of Buffer and indicates it is at the
    bottom of the file.

Terminal Screen

**ritrieve (s.nam,s.grade,s.gt_score) where s.gt_score>=100 and
s.gt_score<100
[EOB]**

*c

o   Text changes can easily be made within the editor by using arrow keys to
    move around the text. The keypad is the small group of keys to the right
    of the larger keyboard on your terminal. Each key performs at least one
    editing function; most of the keys perform two.

o   For information about keypad editing, press **HELP** (PF2, below). The screen
    goes blank for a few seconds after you press the *HELP* key, and then the
    diagram (below) of keypad functions appears. (Requesting help does not
    affect your editing session; pressing the space bar returns you to where
    you left off.)

↑ UP	↓ DOWN	← LEFT	→ RIGHT

PF1 GOLD	PF2 HELP	PF3 FNDNXT FIND	PF4 DEL L UND L
7 PAGE COMMAND	8 SECT FILL	9 APPEND REPLACE	— DEL W UND W
4 ADVANCE BOTTOM	5 BACKUP TOP	6 CUT PASTE	DEL C UND C
1 WORD CHNGCASE	2 EOL DEL EOL	3 CHAR SPECINS	*ENTER* ENTER
0 LINE OPEN LINE		. SELECT RESET	SUBS

Keypad Editing Keys
VT100-Type Terminals

o   When you are done, type a **CTRL-z** (press down the control key and at the same time hit the z key).

o   There are two options to end an editing session. You can either <u>save</u> or <u>delete</u> the text in the buffer. If you save the buffer, you can execute the statement immediately once you get out of the editor and back into INGRES.

o   After doing the CTRL-z, you will see another asterisk (*) at the bottom of the screen. Remember, CTRL-z only gets you out of the change mode, and not completely out of the editor.

      * Exit
         Exit saves the text (only until the next execution).
      * Quit
         Quit deletes the text; nothing is saved.

o   For example if you start an editing session and make mistakes or are dissatisfied with your edits, you may find it easier to use the QUIT command and then start a new editing session.

o   After typing **Exit**, you will be back in INGRES. You can then type **\g** to execute the commands you have edited.

Canned Queries

Queries that you execute many times can be saved so they do not have to be retyped each time. These queries can be written out with the WRITE command (backslash w [\w]). Then each time you want to rerun the query, use the INCLUDE command (backslash i [\i]) to bring it back into INGRES.

For Example:

* retrieve (s.name,s.grade,s.upc) where s.mos="s" sort by upc,name
* \w rations.qry

This retrieves all soldiers who are on separate rations and sorts them by unit and name. The query is saved to be used again in the file named rations.qry.

To reuse the retrieve command:

* \i rations.qry
* \g

The \i command reads the file and places it in INGRES. You still need to use the \g to execute.

VIFRED: VISUAL FORMS EDITOR is used to define, create, edit and store forms in an INGRES database. INGRES generates a default form from a table. VIFRED is then used to improve upon the default form. QBF uses the form developed in VIFRED for updating, appending and retrieving information. VIFRED is concerned with forms manipulation, and QBF is concerned with data manipulation.

The VIFRED Command

    $ VIFRED dbname
    For example: $ VIFRED SMITH

To Create a Form in VIFRED

o   After invoking VIFRED, the screen will clear and display a menu on the bottom of the screen:

Help   Blankform   Catalogs   Form   Table   Exit

o   Select the option **table** to start a default form and give a table name when it promts for **table:**.

o   It will generate a default form like the one shown below. On the bottom of the screen will be a list of items to choose from (menu). Each menu item will be discussed below.

```
┌─────────────────────────────────────────────────────────────────────┐
│  Table Is Soldier                                                     │
│                                                                       │
│  name: c_____        ssn: c_____        mpc: c__           │
│                                                                       │
│  grade: c__       rank: c___       dor: c_____       race: c__        │
│                                                                       │
│  mos: c____     asi: c___     lang: c___     scty_status: c____       │
│                                                                       │
│  scty-clnc: c_____       pulhes: c_____       phys_categ: c__     │
│                                                                       │
│ ─────────────────────────End_of_form────────────────────────────────│
│                                                                       │
│                                                                       │
│                                                                       │
│  Help   Delete   Edit   Field   Insert   Move   Tblfld   Trim   Undo   End  │
└─────────────────────────────────────────────────────────────────────┘
```

To Change a Form

To Move a Field

o   Place cursor on field to be moved (move around form with tab key)
o   Depress PF1; select **move** (PF1 will be used interchangeably with the word menu key)
o   Move the cursor in desired position to be placed before depressing menu key (PF1)
o   Select **place**, or
o   **Right** or **Left** options to move the entire field all the way to the right or left margins.

To Change a Data Type

To change from "c9" to "cl2," for example, name: c_____

o   place cursor on the data field c
o   depress menu key
o   select **edit**
o   select **data**
     The menu disappears and the message below appears on the bottom of the screen:

     Edit Data Window (<MENU KEY>) to terminate   input [INSERT]

o   now begin to change the data
o   type cl2 over the old data c9
o   It will bump the entire line down if additional space is needed.
o   depress menu key to terminate
o   depress menu key again and select **end**

To Change the Title of a Field

o   used for simple editing; basically used as an overstrike facility
o   place cursor on the field to be changed
o   depress menu key
o   select **edit**, depress <ret>. The menu will appear:

     Help Attributes Data Title undo:

o   select **title**
o   the cursor will return to the selected field and the following message will appear on the bottom of the screen:

     Edit Title (<MENU KEY>) to terminate   input [INSERT]

o   This message indicates the insert mode. Begin typing over the old title.

16

o   depress menu key when done

      o   if new title is longer than old title keep typing even though it
              types over the data; it will bump the text over to the right.

o   select **end**

Example 1

         name: c_____   **Statul c**_____

      o   To correct the mispelled word "Statul" depress menu key and
            select **insert**.
      o   The screen will indicate it is in the insert mode:

                Insert (menu key to terminate)

      o   place cursor over the "1" and type an **s**
      o   depress menu key to end the insert mode

To Add Trim

o   Trim is considered as anything other than a field or data. Trim will
      appear on the form yet it has no relation to the table and the data
      which it was created from. Usually trim is used to include
      instruction in a form, title a table, or add borders.

Example

Table is Soldier

     Name: c_____   **ssn: c**____ ___ ____

    — — — — — — — — — — — — **End-of-Form** — — — — — — — — — —

To add trim to this form simply:

o   place cursor at position where trim will be inserted
o   depress menu key then select **trim**
o   as indicated on the bottom of the screen, enter trim:

         Enter Trim (type <MENU KEY> to exit)

Delete

o   to delete a field, position cursor on field to be deleted
o   depress menu key and select **delete**
o   delete a blank line by positioning cursor at any point on the line to
      be deleted

Insert

o   The INSERT command inserts a blank line above the line that the
     cursor is on.

For example:

     age: 1____
     name: c_____

o   To add a line between age and name, place cursor anywhere on the next
     line below where the new line will be inserted—in this case on the
     "name" line.

Undo

o   The UNDO command "undoes" the <u>last</u> executed command only! UNDO
     reverses the previously executed command, and returns the screen back
     to where it was before the last execution. For instance, if a field
     was accidently deleted, select **undo** from the menu and it will undo
     the DELETE command and return the field to where it was before the
     delete. If you delete a field and execute another command and later
     decide you did not want that field deleted, undo will not work. The
     field must be added again.

To Add a Field

o   place cursor in position where field is to be added
o   depress menu key and select **field**, then
o   depress menu key and select **title**
o   enter title, as the screen indicates
o   always add a colon (:) at the end of the field title
o   depress menu key to terminate

To Add Data

Also see page 16, "To Change a Data Type"
o   position cursor where data is to be placed
o   depress menu key and select **data** <ret>·
o   enter new data type
o   depress menu key to end
o   depress menu key again and select **end**

<u>Note</u>: Always remember to **WRITE** any changes in order to save them!

QBFname

o   A "QBF name" must be assigned in order to link the VIFRED Form to QBF.

o   select **write**

      o   The QBF name and the form name can be the same name as long as only one form is created. If you create more than one form using the same table, each form's QBF name must have a different name.

o   select **qbfnames**

      This will provide you a catalog of QBF names for the form.

o   select **addnames**

      It will prompt with this message: **name to use with QBF:**

o   give it a name (same as table name)

o   When it prompts for a table, give it the table name it will be used with.

o   select **end** and **WRITE** it into the database

o   <u>Note</u>: To access a modified form, follow the same steps for creating a form in VIFRED (page 15), and select **form** rather than **table**. It will prompt for the form name. The form name will be the same as the table name.

Query by Forms (QBF) allows users to access tables in INGRES databases to perform routine data retrieval and update tasks, so that users need not learn the basic data manipulation commands of QUEL. QBF provides forms on the terminal screen as tools for accessing and manipulating information from a specific table.

QBF performs any of the three basic functions: APPEND, UPDATE, and RETRIEVE. The APPEND function corresponds to the INGRES "append" command, the UPDATE function to the INGRES "replace" and "delete" commands and the RETRIEVE function to the INGRES "retrieve" command.

The QBF Command

> $ QBF dbname

> For example: $ QBF SMITH

This command will prompt for a table and will provide a default form. This is used if you do not want to use a VIFRED form specifically set up the way you want, otherwise,

o   To invoke "VIFRED forms," you must specify the "-f" flag on the command line:

> $ QBF dbname -f qbfname

> For example: $ QBF SMITH -f qbfsoldier

> Note: This command searches the system tables in the database for the specified form instead of creating a default form.

The screen will clear and a menu will appear on the bottom of the screen:

HELP   UPDATE   RETRIEVE   APPEND   TABLE   EXIT <command>:

After selecting any of the three QBF functions (update, retrieve or append) the screen will clear and a form containing the fields from the table will appear.

Basic Functions of QBF

Retrieve

The RETRIEVE command gets information from the table that you ask for (query). After calling the retrieve function, QBF puts you in a query state. Enter query and press menu key to run.

For example, to find all the officers (O) under the field MPC:
o   position cursor on the field MPC
o   type O
o   depress menu key, the screen clears and the menu appears:

HELP QUERY GO END <Command>:

o   select **GO** to run your query
o   now you will view the rows you have just specified
o   as the instruction on the bottom of the screen indicates, press
     CTRL-f to go forward to the next row

BROWSE DATA (<MENU KEY> to return, CTRL-f for next row)

<u>Note</u>: You cannot make changes in RETRIEVE. This is a browse mode only.

Append

The APPEND command allows you to add rows to the table via the QBF form.

Append #1 (control-Z to add, <MENU KEY> to return)

This message will appear on the bottom of the screen. This indicates that
the first entry is being appended to the table. Enter the data in the
appropriate corresponding fields.

o   **TAB** moves from field to field
o   **CTRL-p** takes the cursor to the previous field, sequentially
     (i.e., the cursor moves backward through the form, one field at
     a time)

Update

The UPDATE command is ordinarily used to make changes to data that already
exists in a table, on a row by row basis. You can query for rows in a
simple manner using forms, and you can edit those rows using forms.

Two States of Update: Query and Go

o   To get data that you want, fill out the form with the values you are
     searching for.
o   After filling out this form, QBF enters the go state, retrieves the
     rows you asked for and allows you to change or delete that row.
o   After changing all the rows you wish to change, save these changes by
     selecting **WRITE**, which writes out the changes to the database.

The following is a step by step procedure on "update."

o   <u>Main Menu</u>

     HELP UPDATE RETRIEVE APPEND TABLE EXIT <Command>:

o   From the main menu select **update** <ret>
o   As the message on the bottom of the screen indicates

     ENTER QUERY (<MENU KEY> to return or to run)

o   Fill in the form with values you want <ret>
o   Depress menu key; select GO
     The rows you have asked for will be brought up.

21

o   Depress **CTRL-f**, pressing both keys at once, until the row you want appears. Be careful because you cannot go backwards.
Values will continue to appear until the message appears:

NO MORE ROWS LEFT IN QUERY

o   Now edit your data or type in new data for each row.

o   If no rows were found for the value that you queried, the message below will appear on your screen:

NO ROWS FOUND

o   QBF then returns you to the QUERY state to enter a new query.

o   Changes are saved in the database with the INGRES WRITE command. Nothing is made permanent until you use the WRITE command.

o   The DELETE command allows you to delete the current row displayed on the screen. If you come across a row you wish to delete, display the menu and type **DELETE**. It will not be actually deleted until you write all the changes, with the WRITE command.

Report-By-Form (RBF) lets you design, on the screen, a report for a table within an INGRES database. RBF basically sets up its own report, requiring minimum specifications on the report design. The main functions of RBF are options, layout and structure. These areas are used for underlining, breaking at end of page, sorting priority, and for purposes of presenting a neat and organized report.

Style

- o   Three styles for reports
  - o   column
  - o   block
  - o   wrap
- o   —m[style] specifies the particular style of report

The RBF Command

- o   First define your terminal type to INGRES in VMS:
  $ define term_ingres "vt100w"

  previous logical name assignment replaced

  - o   This sets the terminal at 132 columns across.

- o   now begin the RBF command:
  $ rbf dbname   tablename   —mstyle

  For example: $ rbf smith soldier —mcolumn

- o   The main RBF menu will appear on the bottom of the screen.

  Help   Catalogs   Report   Table   Exit:

- o   select **Table** and give it a table name when it prompts for table
- o   The screen will clear; these messages will appear one at a time:

  Setting up default report. . .

  Initializing forms for report. . .

- o   At the bottom of the screen is the main command menu. Each command invokes submenus, requiring further choices.

  Help Layout Options Restart Structure Write Exit:

23

Layout

o   The default report works very much like VIFRED's default form.
     Fields can be moved and arranged to design a report to meet the
     specific needs. A typical report has the following elements.

     o   a title
     o   column headings
     o   lines of data

o   Unlike VIFRED, data lines do not automatically get moved along with
     its corresponding column, unless the command "column" is selected.
     Otherwise, move, trim, insert, delete, and edit work the same as in
     VIFRED and will NOT be repeated again here. The main menu for layout
     editing in RBF will appear:

              Help Delete Heading Insert Move Trim Undo End

o   There are three editing options within Layout:

              Edit   Move   Adding Components

o   To execute any of these commands the basic steps are:
     o   move the cursor to the component you want to edit
     o   depress **ESC** (escape) key
     o   select one of the commands listed in the menu at the bottom of
          the screen <ret>

o   Moving Components
     o   position cursor over column heading
     o   depress the **ESC** key; select **move** <ret>
     o $ $ will surround the field to be moved
     o   depress the **ESC** key; select **column** <ret>
     o   move cursor to the desired position
     o   depress the **ESC** key; select **place**
     o   otherwise do not move the cursor and select one of the
          following:

              Center   Left   Right   Shift

o   The column heading and its data line will be positioned at one
     of these positions.

o   To move column headings independent of data lines, use the move
     procedure in VIFRED.

Options

o The OPTIONS command provides the Report Options Form which allows you
   to specify page length and underline text.
   o   select **options** <ret>
   o   the report options form appears, with the cursor on the first
       line

       (report page length (lines):)

o use **TAB** to move to the next field; **CTRL-p** takes you back to the
   previous field
   o   report page length (lines) (defines the size of a page)
   o   specify the underlining (underlining codes defined on screen)
o depress **ESC** to return to menu

Structure

o The STRUCTURE command provides the report structure form which lists
   the column names and provides sorting sequence, sorting order, and
   break column options.
   o   use **TAB** to move around the form
   o   **ESC** takes you back to the menu. For each column name, specify
       the following:
       o   prioritize the sorting sequence that you would like the
           report set up beginning with 1, 2, 3, ... etc.
       o   sorting order
               a= ascending order (e.g., 1, 2, 3, 4)
               d= descending order (e.g., 4, 3, 2, 1)
       o   Break Column
           o   breaks are changes in data values that affect a
               formating action in a report.
           o   helps to present a clearer, more organized report

Consider the following tables

Table 1

vehicle type	bumper #
jeep	1238
jeep	1239
jeep	1849
truck	6711
truck	6712
howitzer	3842
howitzer	3843

Table 2

vehicle type	bumper #
jeep	1238
	1239
	1849
truck	6711
	6712
howitzer	3842
	3843

o No break columns were specified in Table 1. Table 2 suppresses the
   same data values, and only prints when the value changes.
o In the report Structure Form, depress **ESC** to select column options
   o   Column Options
       The column options command, within Report Structure Form
       provides you with a Column Options Form. You may change options
       for specific columns in a report, such as, selection by value at
       run time, and calculations of sums and counts, by editing the
       Column Options Form.

Write

The WRITE option saves all the modifications and specifications made to the default report in RBF.
When you are ready to save the report
  o   select **WRITE** <ret>
The SAVE REPORT FORM will appear

Report name:
Data table name:

The report name, by default, will be the table name. As in assigning a form name, it is not necessary to change the report name unless you create more than one report using the same table.

  o   Depress **ESC** (escape) and select **SAVE** in order to save the current report specifications in the database for later use by the report command.

  o   **END** resumes without saving the report specifications and returns back to the menu. Any changes will <u>not</u> be saved.

REPORT

The REPORT command actually runs and prints out a computer generated report designed in RBF, to a file and/or to the terminal screen.

To output a RBF report to a file:

$ Report -f filename -r dbname reportname

For example:
$ Report -fsoldier.lis -r smith soldier

This command reads the report specifications, retrieves and formats data, but no report appears on the terminal screen.

To send the RBF report from the file to a printer for a hard copy:

$ print12 filename (use print17 if the report is too wide)

To move a RBF report from the file to the terminal screen

$ type filename

For example: $ type smith.fil

To send a RBF report to the terminal:

$ report dbname tablename

For example: $ report smith soldier

Note: The RBF will automatically prompt you for a missing database name and a report/table name. If no -m[style] is specified, by default, the style of the report will be determined by the width of the columns in the tables being reported.

A report, that is output to a terminal is displayed one page at a time with the following information appearing at the bottom of each page:

ENTER c,s,Help or <return>:

To get to the next page, type in c (contine).
To stop reading the page, type in s (stop).

GLOSSARY

<u>Terms</u>:

APPEND: A command used to add information to a database.

BROWSE: A command used to view or look at information in a database.
            Information in the database cannot be changed in any way when using
            the browse command.

BUFFER: A temporary storage area or work space created in the computer
            until the user "saves" the information in the database. The work
            space is called a buffer.

COLUMN: Refers to the type of data which belongs to a table; also referred
            to as a field (of information), this data is connected to a type of
            vertical, or "top-to-bottom," format.

CURSOR: A blinking square or line on the terminal screen (VT100) which
            indicates where the next character will be placed.

DATA: Any item of information such as a number, letter, symbol, fact, or
            statement which can be processed by a computer.

DATABASE: A collection of information organized in a computer such that it
            can be added to, stored and retrieved in a fast and efficient way.

DEFAULT: When the computer receives a command, it must respond to the
            command. If no specific instruction is given by the user, it will
            provide a preprogrammed value. Default is the preprogrammed action
            that is taken.

DELETE: A command used to remove a value or a row of information in a
            database.

DISK: A flat rotating ciruclar magnetic material used for storing data in
            the computer.

EDIT: A command which allows the user to make changes to data fields.
            The edit command can be used in time mode for changing data one
            line at a time or for full screen editing.

FIELD: Refers to the type of data which belongs to a table; also referred
            to as a column (of information).

FILE: A collection of information stored with an identifiable name on a
            disk or tape.

INGRES: A relational database management system used to create a database
            and to add, update, and retrieve data.

INPUT/
OUTPUT   A process used to put data into a computer or take out data from a
(I/O):   computer

29

INTEGER: A whole number, either positive, or negative (-3, -2, -1, 0, 1, 2,
            3, etc.)

MENU: A list of choices, or items, that when made, perform a certain
            function. Menu choices are displayed on the bottom of the screen.

PROMPT: A message given by the computer to the user asking for information
            or giving instuctions, e.g., showing an error message.

QUERY: A request for information from the database by the user.

RETRIEVE: A command used to pull out information from a database.

ROW: A horizontal series of field(s) running across the screen. Also
            called record.

STORE: A function which saves data in the computer memory.

TERMINAL: A computer and a CRT (Cathode Ray Tube) screen used both for data
            input and output.

Symbols:

<ret>   Press the return key on the terminal.

\g   Press the backslash key, then the g key.

\i   Press the backslash key, then the i key.

\w   Press the backslash key, then the w key.

CTRL-f   Press the CTRL key and the f key at the same time.

CTRL-p   Press the CTRL key and the p key at the same time.

CTRL-z   Press the CTRL key and the z key at the same time.

$ VMS, the operating system of the computer, is ready to accept a
            command or to enter INGRES or the editor (ed/edt).

*   The INGRES Terminal Monitor used to formulate, review, or execute a
            query to INGRES.

[EOB]   Indicates the End of Buffer.

30