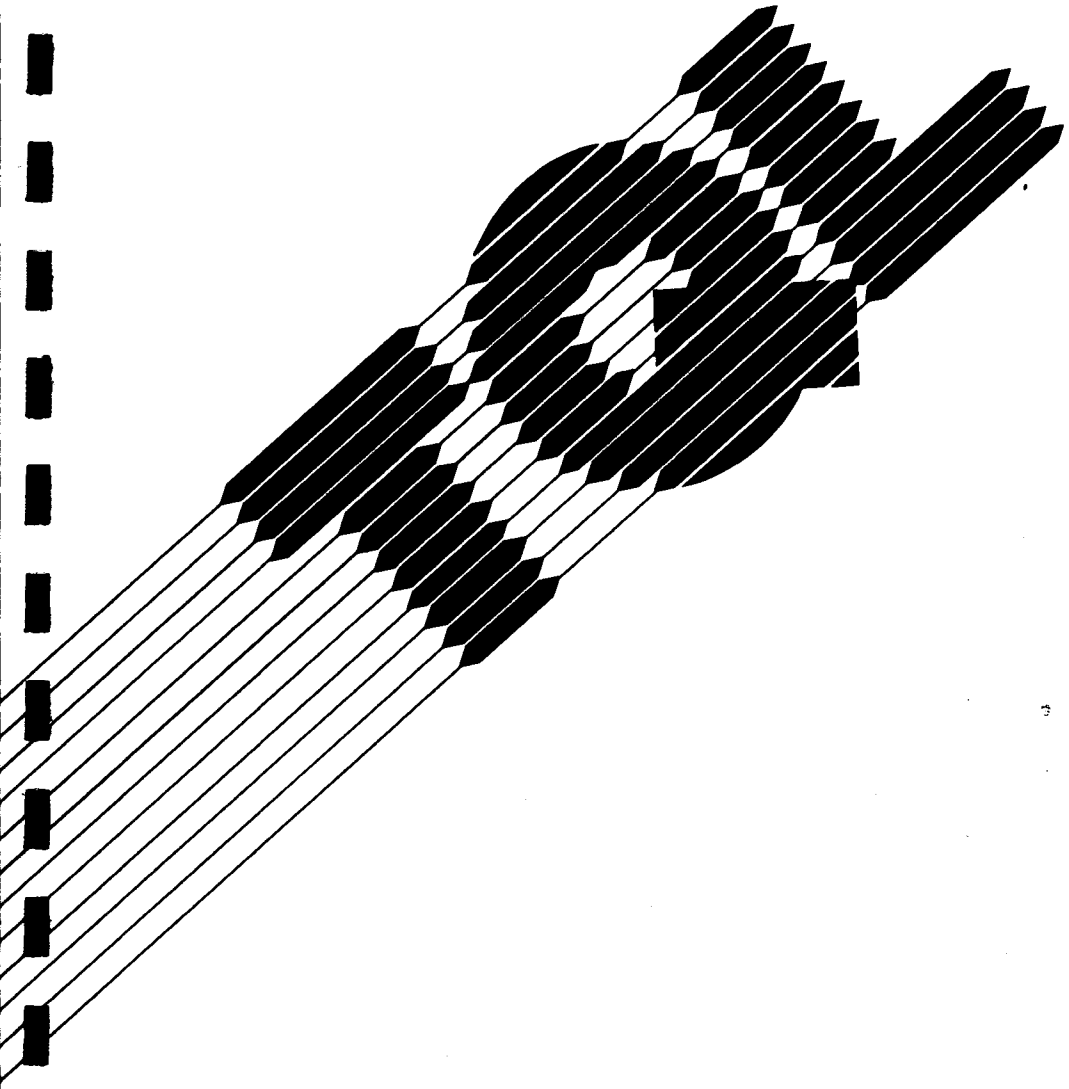
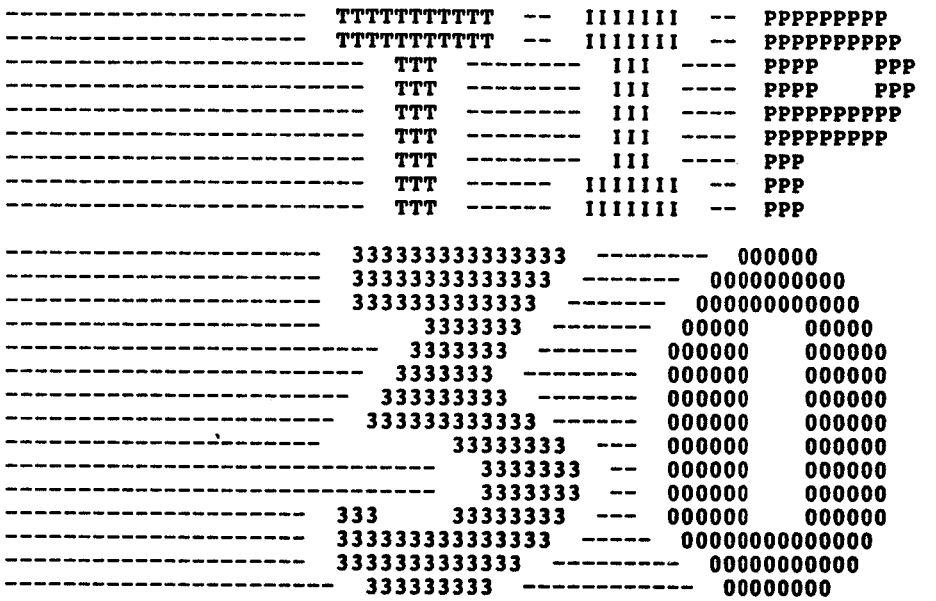


TIP/30
Query Language
User Guide



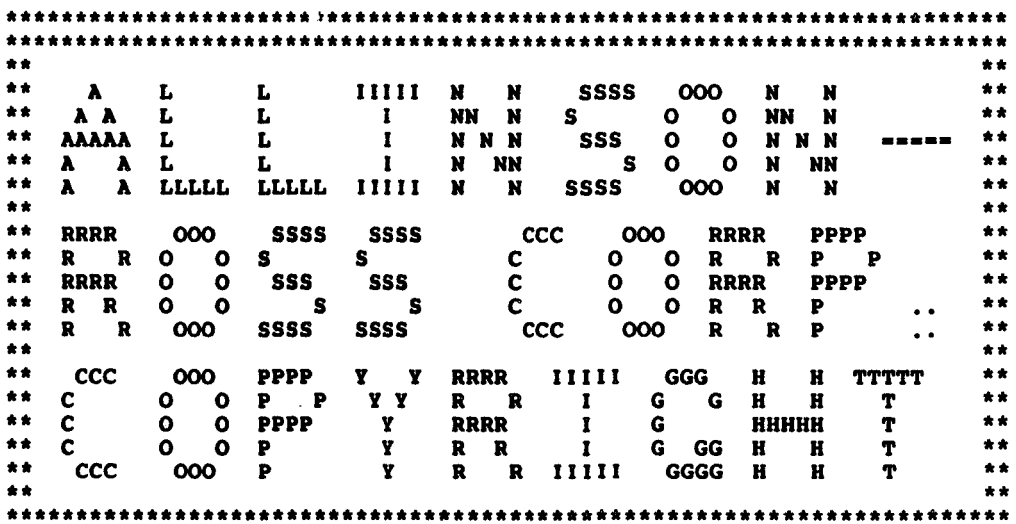
TIP/30 Query Language - User Guide

Version 3.1 (85/06/01)

ARP-603

A Product of:

Allinson-Ross Corporation
4291 Village Centre Court
Mississauga
Ontario
Canada L4Z 1S2
TEL: (416) 848-2030
TWX: (610) 492-5390



Allinson-Ross Corporation,
4291 Village Centre Court,
Mississauga, Ontario,
Canada L4Z 1S2
Tel: (416) 848-2030
Twx: (610) 492-5390

THIS DOCUMENT WAS PRODUCED USING THE
ALLINSON-ROSS CORPORATION DOCUMENT GENERATOR.

--*+*--

--*+*--

TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE
TQL USER GUIDE

1. TQL USER GUIDE

This document describes the TIP/30 Query Language (TQL) from the perspective of the terminal user. TQL is an interactive system which allows the terminal user to manipulate data that is stored in the files in the on-line system.

It is assumed that the reader has reasonable general knowledge of computers. The assumption is also made that the user is familiar with the operation of the terminal.

A more advanced user may prefer to skim through the material to bypass the tutorial information.

TQL provides the following capabilities for the terminal user:

- data may be **selected based** on the contents of one or more fields;
- data may be **sorted based** on the contents of one or more fields;
- data may be **displayed** on the terminal using pre-defined displays or a selection of fields specified at execution-time;
- data may be **added, deleted,** or **updated** at the terminal;
- data may be **printed** on the central system printer or a terminal printer using either a pre-defined report format or a selection of fields specified at execution-time;
- data may be **exported** (copied) to a file on a personal computer (PC) for manipulation by software that is running on the PC.

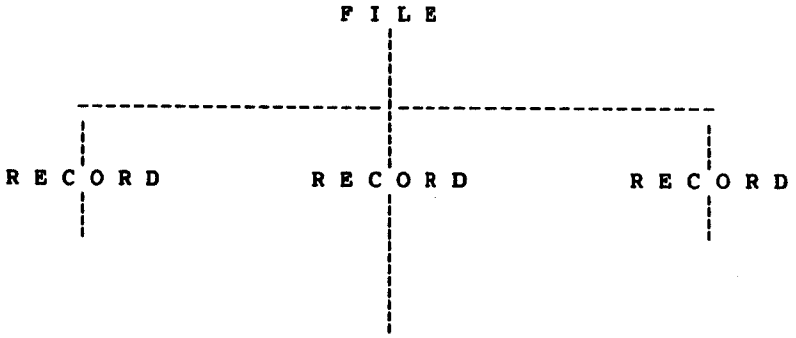
The TQL user actually executes a previously created TQL program.

All TQL programs employ the same interface with the terminal user although the capabilities of a particular TQL program may not be the same as other TQL programs.

1.1 FILES, RECORDS, AND FIELDS

DATA STRUCTURE

Information that is stored in the on-line system is typically organized in a hierarchical fashion:



A common example of such a hierarchy of information is payroll information. Payroll information would normally be stored in a file reserved for that purpose.

Within the payroll file there will be a number of payroll records. Typically there is one record for each employee.

Each payroll record is actually a collection of fields. Each field is a distinct piece of information. A field is the fundamental unit of information in the system. A field is assigned a field name (by the programmer).

The field name is a very important piece of information - the user needs to know the name of field to be able to manipulate the field (display/update/report the field).

In a payroll environment, a record in the payroll file might contain fields named as follows:

- EMP-NUMBER
- EMP-NAME
- EMP-ADDR-1
- EMP-ADDR-2
- EMP-ADDR-3
- EMP-BIRTH-DATE
- EMP-SALARY

In reality, the record would normally contain a (very) large number of fields. For our purposes, a shorter list is much more manageable.

Notice that field names may be hyphenated names. This naming convention is used to make the names of fields easier for the user to read. The computer has no difficulty distinguishing different field names - in fact, the computer is rather picky about spelling and grammar - more on that subject later.

Also notice that field names often have a common prefix ("EMP-" in the previous example). A field name prefix is a handy thing. The system may need to distinguish fields in different records and even different files. The fields named "EMPLOYEE-NAME" and "SUPPLIER-NAME" are clearly both names of some sort - but the fact that they have a different prefix just about guarantees that they are not the same thing!

Fields also have another interesting property - they have an attribute known as the field **type**. The type of a field is either **numeric** or **alphanumeric**.

The type attribute for a field is fixed by the definition of the field within a record. The user cannot alter the type of a field. The type is important to know because some TQL commands only operate on a particular type of field.

Numeric fields may only be used to represent numbers. A numeric field might be positive or negative (although a salary field is normally positive!). A numeric field may or may not be defined to include a certain number of decimal places (the salary field might be dollars and cents for example).

Alphanumeric fields may only be used to represent characters. The characters that are stored in an alphanumeric field are just characters - like a name or an address. Even if the field happens to contain characters that are all digits the field is **NOT** a numeric field as far as TQL (and the system) is concerned.

Thus by definition, fields are always a specific and unvarying type. Fields also have a predefined length. Numeric fields are always defined as a fixed number of digits before and after the decimal place. Alphanumeric fields are always defined to be a fixed number of characters long (a name field may be restricted for example to a maximum of 30 characters).

The TQL system prefers to deal with fields that are a fixed size and have a fixed type (numeric or alphanumeric).

There are occasions when a field may be defined as a multiple occurrence. For example, it may be convenient to define an address as three occurrences of a 30 character alphanumeric field. Each part of the address field can be referenced by specifying the field name followed by a number in parentheses:

```
EMP-ADDR (1)
EMP-ADDR (2)
EMP-ADDR (3)
```

The number in parentheses is often called a "subscript" or an "index". It is nothing more than a designation of the particular occurrence that is intended.

If the number in parentheses is not specified (when it should have been) the TQL system will assume that the user intended a value of (1) - the first occurrence.

If the number in parentheses is specified (when one shouldn't have been) the TQL system will normally complain that an error has been made in the name of the field.

An alternative way to define these three fields would be simply to define three distinct fields (as in our original example):

```
EMP-ADDR-1
EMP-ADDR-2
EMP-ADDR-3
```

The choice between the two methods is made by the programmer at the time the TQL program was created. Using subscripts is often preferred because this reduces the number of unique field names.

In this document, we will assume that the "EMP-ADDR" field is defined with subscripting.

1.2 KEY FIELDS

RECORD KEYS

Since there are normally a number of records stored within a single file, there needs to be some means for the system (and the user) to uniquely identify a particular record.

Usually one field (or several adjacent fields taken as an group) is designated as the unique identifier. This field (or group of fields) is called a **KEY (field)**.

A record can be organized so that there is a primary key and up to four secondary keys. A record must always have a unique primary key. Secondary keys may or may not be defined and may or may not allow duplicate values.

It is always very important for the user to understand what field or fields make up the primary key of a particular record. Often it is something rather obvious (EMP-NUMBER would be a likely guess in our previous example).

The EMP-NUMBER is presumably unique and can therefore be used to guarantee that there is one (and only one) payroll record associated with an employee.

Using this example of a record (the PAYROLL record), it should be fairly obvious that the EMP-NAME could be set up as a secondary key.

TQL allows information to be retrieved by the primary key (this is the default situation) or by some secondary key.

The choice of which fields make up the key or keys of a record is normally made by the Data Processing Department. This choice is **NOT** normally changed very easily because a great deal of processing with the file (not just on-line access via TQL) depends on the definition of the record structure and key organization.

1.3 GETTING STARTED RUNNING TQL

There are a number of ways to interact with the TIP/30 Query Language (TQL). We will first discuss the direct approach.

The user normally has a choice of a number of TQL programs to run. TQL programs are created by the Data Processing staff to support one or more users of the on-line system. Each program has a unique name. This name is normally supplied to those users which need to execute the program.

In order to start the interaction with the TQL system and to "execute" the appropriate TQL program, the user must first establish the name of the TQL program to be run.

Assume that we are aware of a TQL program called "TQLDEMO". This TQL program is advertised to allow manipulation of the data in a payroll file (the payroll record and the fields of the payroll record).

To run this program, simply enter the following at the terminal:

```
OPEN TQLDEMO
```

At some sites, the word "OPEN" in this context may be redefined. The user should consult with the Data Processing Staff to find out whether "OPEN" is to be used or some other keyword.

The name of the TQL program to be executed is given after the word "OPEN" (we will assume that OPEN is in fact being used). The program name must be separated from the word "OPEN" by a space.

One possibility that might occur after entering this command at the terminal is that the system may respond with the error message:

```
Not able to locate program: TQLDEMO
```

This might occur for example, if the name of the TQL program was not spelled correctly or the user is not allowed to run that particular TQL program.

If the user receives this error message, the program name should be verified. The user may clear the terminal screen and re-enter the OPEN command to attempt the command again.

Another possibility that might occur is that the system may respond with the error message:

```
Security violation
```

This would occur if the terminal user has not been granted access to the TQL facilities of the system.

If the user receives this error message, the user should contact the Data Processing department to discuss using TQL.

Another possibility (although a relatively remote one) that might occur is that the system may respond with the error message:

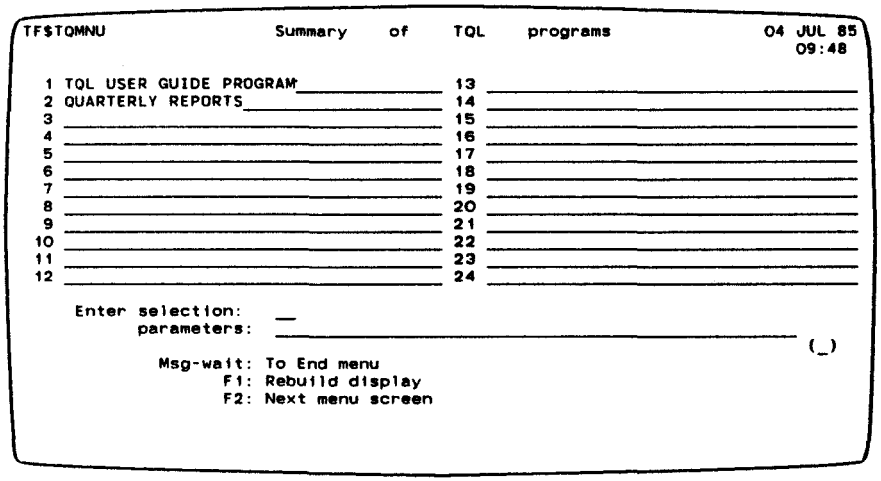
```
Invalid transaction code: xxxxxxxx
```

(The xxxxxxxx represents the word that the system objects to).

This error would occur if the user misspelled the word "OPEN" (or whatever). This error may also occur if the system is currently unable to run TQL. If this error occurs and it is NOT a spelling error, the user should contact the Data Processing department for help.

If the user enters the transaction "OPEN" (or whatever) without specifying a TQL program name the TQL system will respond with a menu of available TQL programs. This allows the terminal user to select the appropriate TQL program and eliminates the need to remember a number of TQL program names.

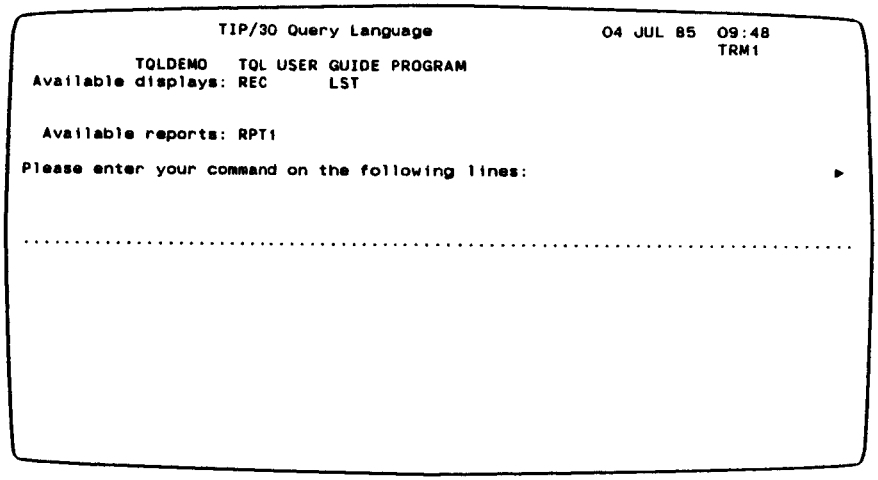
The TQL menu looks like this (your choices would probably be quite different):



The terminal user can simply enter the desired selection number and press XMIT or press MSG-WAIT to exit from the menu.

The parameter field may be used to enter a single TQL command. If this is done, TQL will OPEN the selected program and execute the specified command (only) and then immediately terminate the TQL program. This essentially allows the terminal user to execute a single TQL command as if it was a stand-alone function.

If no errors have occurred during TQL initialization the user would next be shown a display similar to the following example:



This screen display is the standard TQL prompt screen.

Almost all of the TQL commands that the user enters will be entered using this screen format.

The first line of the screen gives the current time and date. The second line will display the name of the terminal that the user is using ("TRM1" in the example above).

Starting on line 4, the screen indicates the names of the displays and reports that are available within the current TQL program (in this case the TQLDEMO program).

A display is a pre-defined format that has been set up by the programmer for use by the TQL user. A display will contain a number of fields from the record(s) in the file. The display may be used to simply view information or (in some cases) update, add or delete information.

The choice of which fields to display and the format or presentation of the fields is entirely at the discretion of the person who has defined the display (usually the programmer who wrote the TQL program).

There may be a number of available displays in a particular TQL program. For example, a PAYROLL record usually contains far more fields than would comfortably fit on one screen. The programmer normally would set up several displays. Each display would focus on some set of related fields (one display for general information, one for deductions, one for benefits etc).

A report is a pre-defined layout that defines which fields are to be printed and the order and presentation of the fields which make up the report. A pre-defined report may be printed on the main printer of the computer or may be directed to a printer that is attached to your terminal (if your terminal is equipped with an auxiliary printer).

In the example screen shown there are two displays. They are named "REC" and "LST". There is one report defined. It has the name "RPT1".

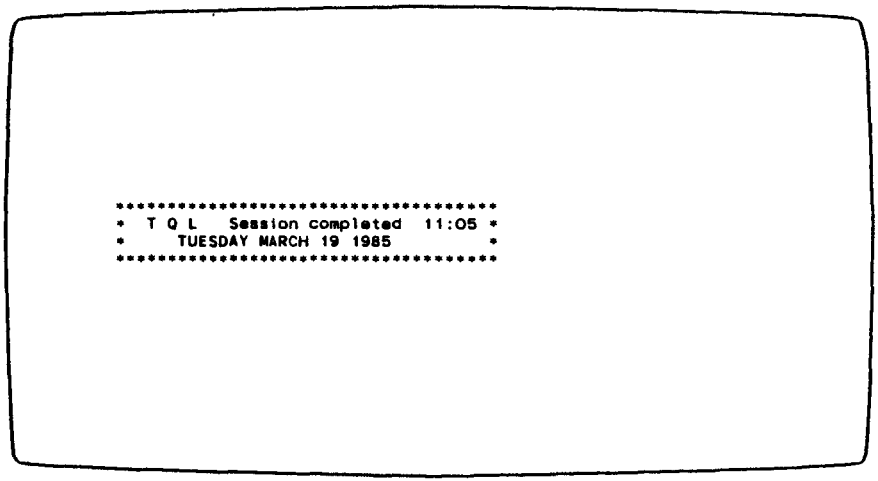
Notice that there is a command area (the three empty lines which are immediately underneath line 9 ("Please enter your command on the following lines:"). This command area is the only area in this screen where the user is allowed to enter TQL commands (more on that in a moment).

The area below the row of periods on line 10 is reserved for use (by TQL) as an informational display area.

1.4 ENDING TQL SESSION END

The command to exit from a TQL program is "END" (an alternative spelling is "CLOSE"). Either command will cause TQL to terminate the execution of the current TQL program.

If the TQL program was entered via the TQL menu, the user will simply return to the TQL menu (see previous section "GETTING STARTED") otherwise the following display will appear when the TQL program terminates:



Of course, the time and the date will always be the time and date that the TQL session was terminated.

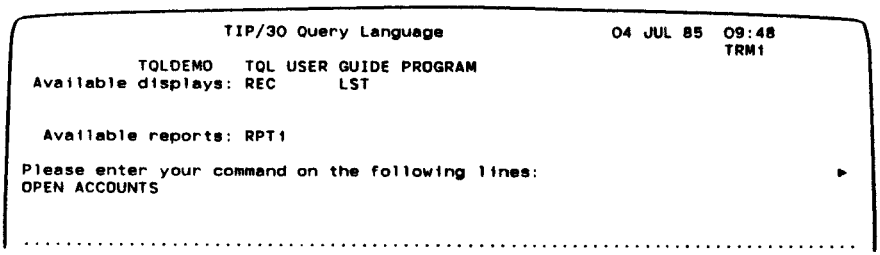
1.5 SWITCHING TQL PROGRAMS OPEN

If the terminal user is currently running a particular TQL program and wishes to both terminate the current program and immediately run a different TQL program, the "OPEN" command may be used.

The OPEN command requires the word "OPEN" followed by the name of the next TQL program to run.

The OPEN command simply performs an "END" command and immediately performs an OPEN of the specified TQL program (effectively switching from the current TQL program to the next).

For example:



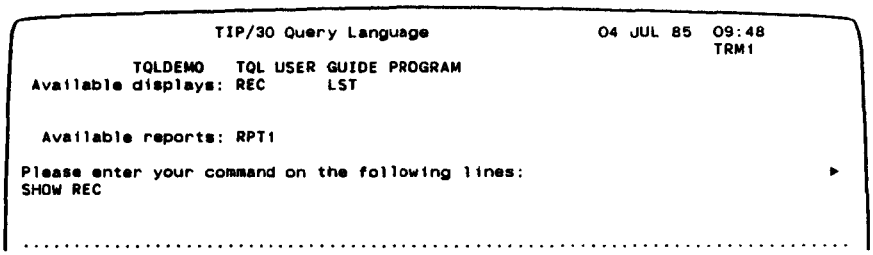
would close the program and attempt to enter a different TQL program ("ACCOUNTS").

1.6 USING THE <SHOW> COMMAND SHOW

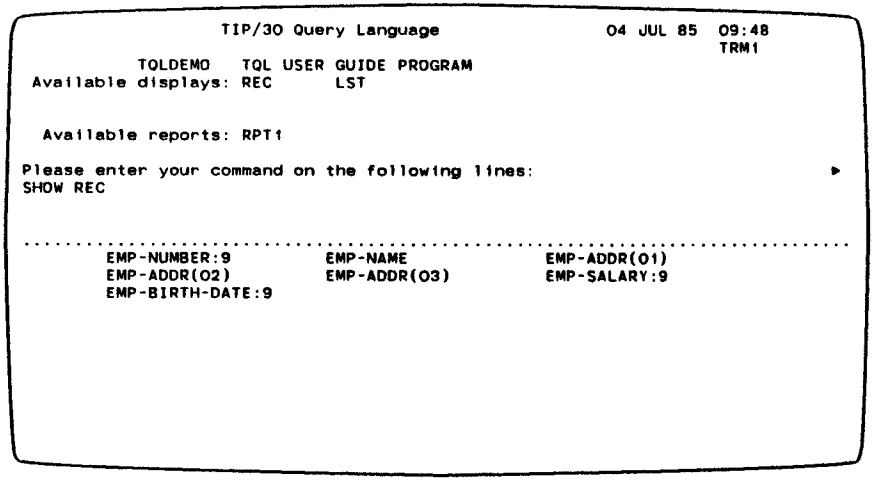
A TQL program may offer a choice of a number of pre-defined displays and/or reports. In this case the user may wish to know exactly which fields are referenced when a particular display or report is selected. A later section of this guide describes how to request a particular display or report.

The TQL **SHOW** command can be used to request that TQL display the names of all the fields that will be displayed (or reported) when a specified display or report is asked for.

To use the **SHOW** command, the user enters the word "SHOW" in the beginning of the command area on the prompt screen. Following the word "SHOW" the name of the desired display or report is entered and **XMIT** is pressed:



TQL will respond by "showing" all of the fields that are involved in the pre-defined display named "REC". The information is shown in the informational area that occupies the bottom 8 lines of the screen:



Note that this example has a rather small number of fields. If the display (or report) had a large number of fields, a message on the screen would invite the user to press function key F2 to view the next (and subsequent) batch of field names.

The field names are displayed (from left to right and top down) in the order that they appear in the display named "REC".

The notation ":9" as the suffix of a field name is used by TQL to indicate that the field is a numeric field. The ":9" is **NOT** actually part of the field name.

Note that fields which are subscripted (indexed) are shown with a numeric occurrence number in parentheses after the name.

Field names are usually chosen by the programmer to be meaningful. If the user does not find the names meaningful, the user should consult whatever documentation was provided (by the Data Processing department) for the particular TQL program.

In a similar fashion, let us examine the field names that are involved in the other display ("LST"):

```
                TIP/30 Query Language                  04 JUL 85  09:48
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                      ►
SHOW LST

.....................................................................................
       EMP-NUMBER:9          EMP-NAME              EMP-SALARY:9
       EMP-BIRTH-DATE:9
```

... and the pre-defined report ("RPT1"):

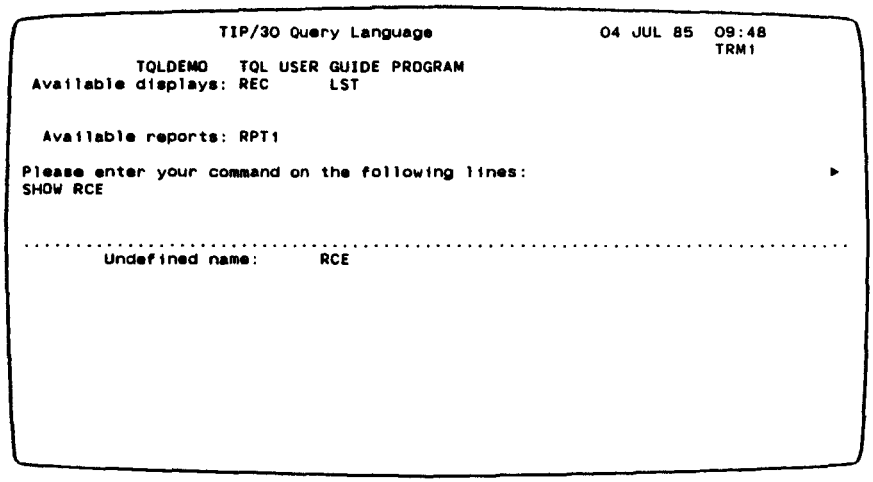
```
                TIP/30 Query Language                  04 JUL 85  09:48
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                      ▶
SHOW RPT1


.....................................................................................
        PAGE$:9                 EMP-NAME              EMP-NUMBER:9
        EMP-SALARY:9
```

The field "PAGE$" is a special field name that is used internally by TQL and is not discussed in this user guide.

If a display or report name is not supplied after the "SHOW" command, a flashing error message will appear at the top of the screen:

SUPPLY DISPLAY OR REPORT NAME

If the name following the "SHOW" command is not a valid display name or report name, the following display will appear to indicate that the "SHOW" command detected an error:



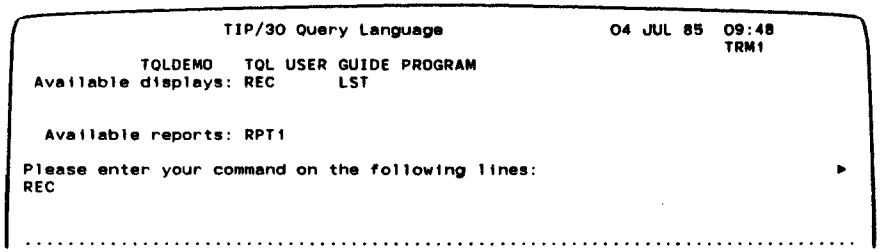
And a flashing error message will appear at the top of the screen:

Errors encountered!

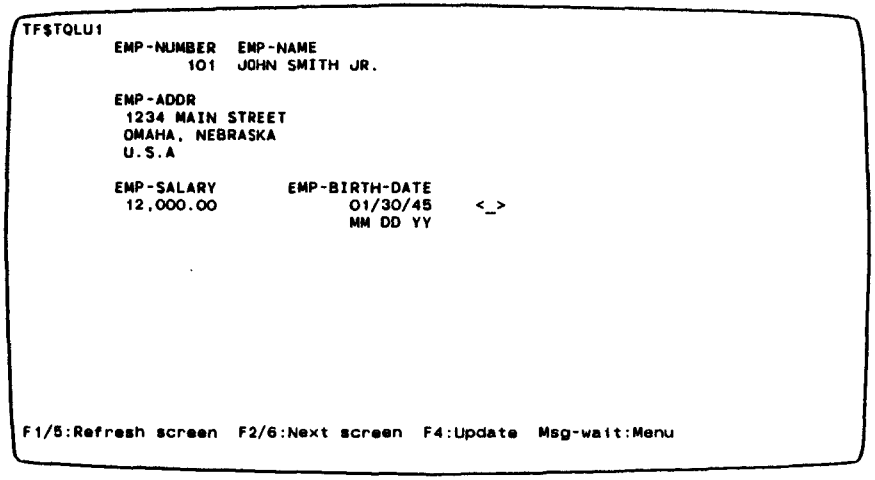
1.7 USING A PREDEFINED DISPLAY DISPLAYS

A TQL program may offer a choice of a number of pre-defined displays. These displays have been specifically defined by the programmer. Usually, the displays will show frequently used combinations of fields.

A particular display can be selected by simply using the display name as a command:



Since no information was provided with the "REC command" (more on that point in a moment) TQL will react by using the first record of the payroll file to fill in the fields as defined by the REC display:



This information is the result of the display which was defined by the programmer who created the TQL program. The order of fields, the headings for the fields, and all other heading information is at the discretion of the programmer.

Note that each of the fields that was displayed by the "SHOW REC" command are displayed. The address field (which is in fact three fields) happens to be displayed as three fields with one simplified heading.

A subtle but important point is that the headings are just decorations - they are not necessarily the actual field names.

USING A PREDEFINED DISPLAY

At this point the "REC" display (as it was defined) has been used to display the first record from the payroll file. There are several standard function keys that are available to the terminal user at this point:

MSG-WAIT	return to the TQL command screen
F1 (or F5)	refresh the current screen
F2 (or F6)	display the "next" record from the file
F4 (or F8)	update the fields that are shown in the display

Option F4 (to update) may not be available in all TQL programs. Remember that some TQL programs deliberately impose restrictions on the terminal user's capability to update, add or delete information.

F9	view additional "child" records

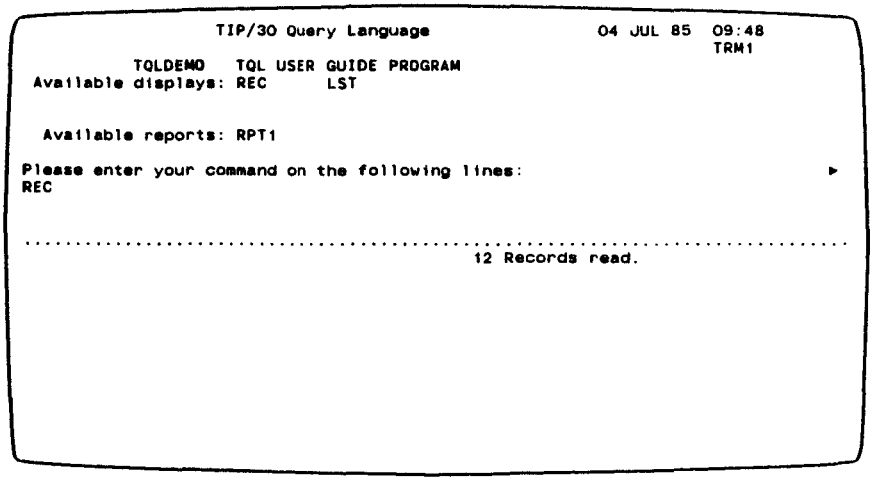
Function F9 may or may not be available in a particular TQL program. It is sometimes the case that information displayed by a TQL program consists of data from a "parent" file along with supplementary information from one or more "child" records (usually from a different file). Function key 9 is intended to be a request to view "MORE" child records (while leaving the parent information intact).

F9 differs from a NEXT request (F2/F6) because F9 deals with child records whereas F2/F6 deals with parent records.

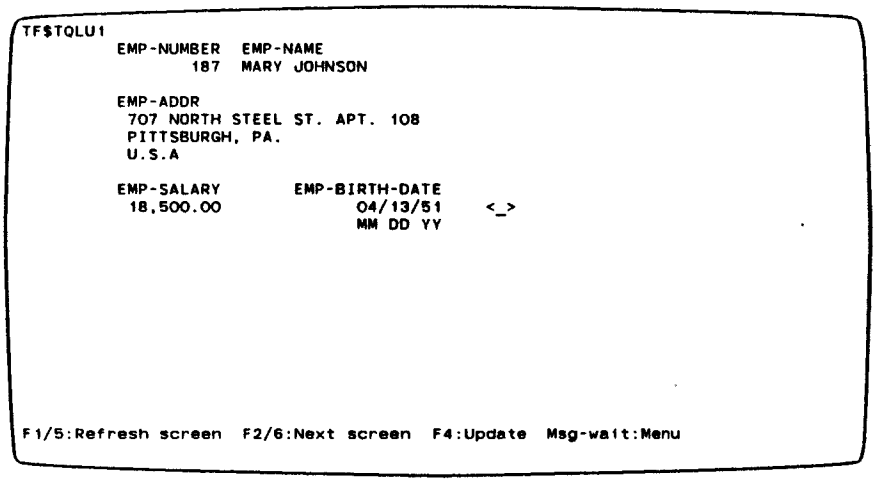
This parent-child relationship is not discussed further in this document - the Data Processing personnel will advise you if the TQL program that you are using makes use of a parent-child relationship.

Pressing MSG-WAIT causes TQL to return to the TQL prompt screen. When this prompt screen is displayed, TQL will automatically leave the text of the last command you entered in the command area (this allows you to make modifications to the command and submit it again).

If the last command processed (displayed) more than one record this fact will be noted by TQL by an informational message in the lower area of the prompt screen:



Pressing F2 (or F6) will cause TQL to advance the "REC" display to the next record in the file:



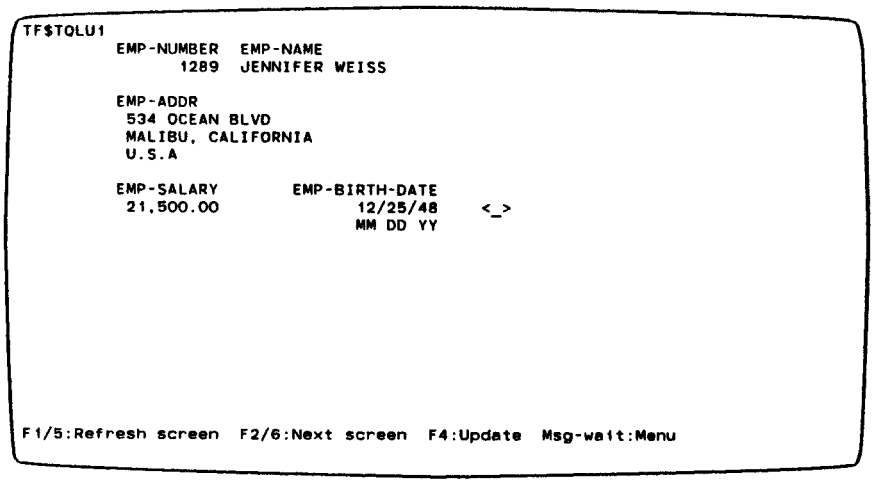
Notice that after pressing F2, the information displayed is the next record from the payroll file (employee number 187 apparently follows 101 in our example).

Every time F2 is pressed, the display will advance through the file. This technique would be fine if the record that you want to display happens to be near the beginning of the file.

The display can be invoked in a slightly different manner to view a specific employee record. Assume that we wish to see employee number 1289. (Remember that the employee number is the primary key of our payroll file).

```
                TIP/30 Query Language                  04 JUL 85  09:48
                                                                  TRM1
          TOLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                      ►
REC 1289


.....................................................................................
```

In this case, the REC display would presumably find record number 1289 and display that information:



If the value specified (after the command) is not found as a valid key of the file, the informational message:

End of selection

will be displayed to indicate that the specified key value was not found.

In this example TQL program there is another pre-defined display ("LST"). As the name of this display implies, selected information from several records will be displayed when LST is requested:

```
                    TIP/30 Query Language                  04 JUL 85  09:48
                                                                      TRM1
            TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                          ▸
LST


 ..........................................................................
```

would produce the following display:

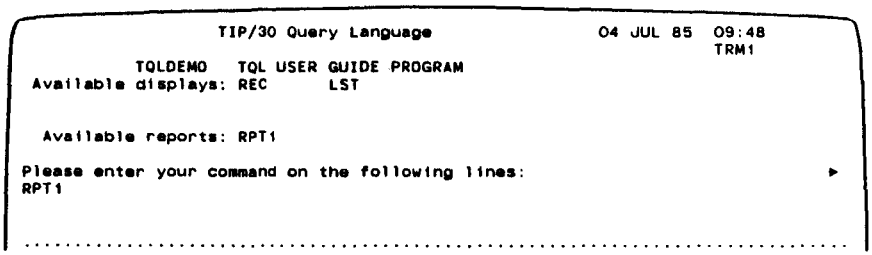
```
TF$TQLU2
 EMP-NUMBER  EMP-NAME                    EMP-SALARY  EMP-BIRTH-DATE
        101  JOHN SMITH JR.               12,000.00        01/30/45
        187  MARY JOHNSON                 18,500.00        04/13/51
        807  DAVE HARRISON                18,750.00        03/19/46
       1024  WILLIAM MARTIN               20,500.00        02/29/44
       1289  JENNIFER WEISS               21,500.00        12/25/48
       3356  MICHAEL HARRIS               34,500.00        01/01/48
       3376  RONALD DAWSON                14,800.00        09/11/37
       5645  DONALD TRACEY                34,000.00        05/11/44




F1/5:Refresh screen  F2/6:Next screen  F4:Update  Msg-wait:Menu _
```

1.8 USING A PREDEFINED REPORT REPORTS

A TQL program may offer a choice of a number of pre-defined reports. These reports have been specifically defined by the programmer.

When a report has been defined by the programmer a default print destination has been set up. The next section of this guide describes the various valid print destinations. For simplification of this discussion, we will assume that there are only two possible destinations: PRNTR (the central computer printer) and AUX1 (an auxiliary printer that is attached to the user's terminal).

A particular report can be generated by simply using the report name as a command:



TQL will respond with a message "**NOW PRINTING**" while the report is being generated (the message will be erased when the report has been completed):

```
                    TIP/30 Query Language                  04 JUL 85  09:48
                                                                      TRM1
            TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                          ▶
RPT1 << NOW PRINTING >>


.....................................................................................
```

Since no constraints were provided with the "RPT1 command" (more on that point in a moment) TQL will generate the report (as defined by RPT1) using all records in the PAYROLL file.

In this example, the default destination is PRNTR. If the user preferred, the report could have been directed to an auxiliary printer that is attached to the terminal by using the <ON> clause:

```
                TIP/30 Query Language                  04 JUL 85  09:48
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                      ▸
RPT1 ON AUX1

.....................................................................
```

The only difference is the final destination of the report. (Auxiliary printers are typically poor quality and slow printers - they are generally handy for LOW volume printing).

(See following section describing the print destinations supported by TQL).

USING A PREDEFINED REPORT

A sample of the information generated by the RPT1 pre-defined report follows:

```
_____top of page_____

        FOR  username       850201  13:33       TERMINAL:TRM1
RPT1


_____top of page_____

<--------- EMPLOYEE  NAME/NUMBER --------->       SALARY   PAGE:      1


JOHN SMITH JR.                          101      12000.00
MARY JOHNSON                            187      18500.00
DAVE HARRISON                           807      18750.00
WILLIAM MARTIN                         1024      20500.00
JENNIFER WEISS                         1289      21500.00
MICHAEL HARRIS                         3356      34500.00
RONALD DAWSON                          3376      14800.00
DONALD TRACEY                          5645      34000.00



_____top of page_____

                                                 PAGE:      2

TOTAL EMP-SALARY:       174550.00
```

This information is the result of the report which was defined by the programmer who created the TQL program. The order of fields, the headings for the fields, and all other heading information is at the discretion of the programmer.

Notice that the report includes an initial header page that is automatically generated by TQL. This header page serves to identify the originating user and terminal and also documents the actual TQL command which was used to generate the report (this makes getting a similar report 3 months later much less difficult).

This report was programmed to include a trailer page that includes some additional information (namely: TOTAL EMP-SALARY). The programmer had to include specific coding in the TQL program to generate this total. Obviously, more complicated totals or computations could have been pre-programmed.

1.9 TQL PRINT DESTINATIONS ON

Various TQL commands (and pre-defined reports) allow the terminal user to specify a "print destination" via the optional **ON** clause. This section describes the various choices that may be available to the terminal user. Keep in mind, however, that each system can customize these printer destinations. The user is advised to double check print destinations with the system administrator for their site.

The ON clause may be specified in one of three basic forms:

- ON printer-name
- ON AUXn
- ON ?:??????

printer-name is the name of a main site printer. Usually, there will always be a printer named "PRNTR" at every site. This printer represents standard forms on the central computer high-speed printer. Some sites may define other printer-names for specific types of forms (eg: PRNTR2 PRNTR3 etc).

AUXn is a special printer name that represents an auxiliary printer that is **attached to the terminal.** The "n" represents an auxiliary number and usually is specified as a "1". AUX1 is the standard name for the (first) auxiliary printer attached to the terminal.

?:?????? represents a file name on a Personal Computer. This name consists of a disk drive identifier (the single alphabetic character preceding the colon) and a file name (the 1 to 6 characters that follow the colon). TQL will create a print file (on the personal computer) on the specified disk drive (A: B: C: etc) with the specified name and the file name extension of "PRN".

This latter facility is available **only if** you are executing the TQL program on a Personal Computer that is equipped with the Sperry Terminal Emulator Package (STEP) or the Computer Logics Personal Emulator Package (PEP). Your data processing department can advise you whether or not this combination of hardware and software is available to you.

1.10 <FROM> CLAUSE FROM

There may be instances when a predefined display (or report) is desired and a specific key value is not known. For example, we may wish to view the payroll information of employees with employee numbers greater than 1287. All that is known is a specific starting point.

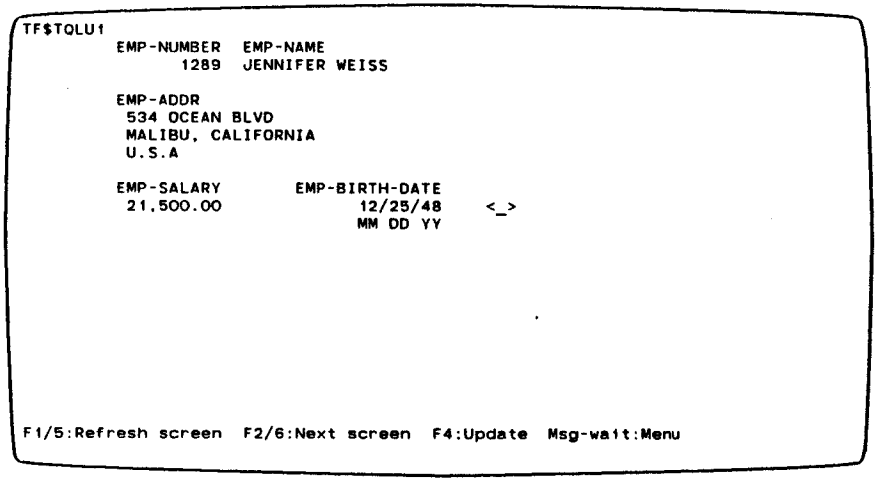
The TQL **FROM** clause may be used in conjunction with the predefined display or report to indicate that the display or report is to begin at a specific point in the file.

The FROM clause provides a way to specify a key (or a partial key) which will define the lower limit of the view of the data in the file. TQL will make available the first record in the file that has a key greater than or equal to the key specified.

```
                TIP/30 Query Language              04 JUL 85  09:48
                                                              TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▶
REC FROM 1287
```

In this case, since there are no records on file with employee number 1287 or 1288, TQL will find record number 1289:

<FROM> CLAUSE



The FROM clause requires the word "FROM" to be followed by the key of the record that is to be the starting point of the display. The value specified for the key need not be a complete key. The value does **not** need to be placed in quotes if the key field is a numeric field (remember EMP-NUMBER:9). If the key field is an alphanumeric field, the value should be placed in quotes.

The display will start with the first record in the file that has a key value **equal to or greater than** the specified value.

For example, we might have wanted to begin the display with the first employee with an employee number of the form 12xx.

The following command would accomplish this:

```
                TIP/30 Query Language                  04 JUL 85  09:48
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                      ▸
REC FROM 1200
```

If a FROM value is specified that repesents a key that is higher than any key in the file, the informational message:

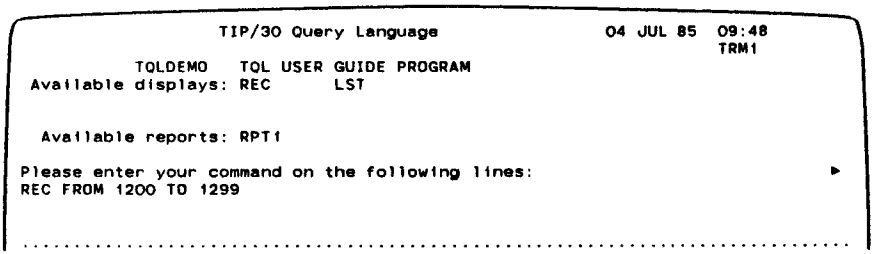
End of selection

will be displayed to indicate that there are no (more) records available for further display.

Incidently, this message would also appear if the user uses the F2 key to advance past the last record in the file.

1.11 <TO> CLAUSE TO

A companion to the FROM clause is the TO clause. The TO clause can be used to put an upper limit on the range of the information that may be accessed:



This command would invoke the "REC" display starting with the first employee number greater than or equal to 1200 and would allow the user to press F2 (to see the next employee record information) until the employee number exceeds 1299. This effectively restricts the display to employee numbers 1200-1299 inclusive.

The TO clause requires the word "TO" to be followed by the key of the record that is to be the upper limit of the records available. The value specified for the key need not be a complete key. The value does not need to be placed in quotes if the key field is a numeric field (remember EMP-NUMBER:9).

1.12 <BY> CLAUSE BY

Up to this point we have assumed that we wish to view information in sequence by the primary key of the file. If the file has one or more secondary keys defined (remember that files may have from one to five keys), TQL can be instructed to use a key other than the primary key.

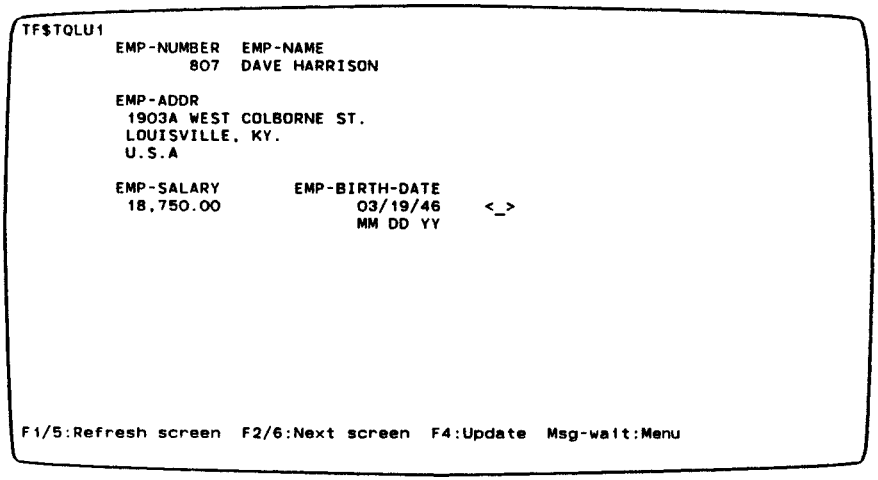
The BY clause may be used to specify the name of the field which is to be used as the controlling key. The field name specified must be a valid field name and must have been properly defined as a secondary key of the file.

If the BY clause is used, it must be specified before other clauses (the FROM and TO clauses are obviously sensitive to which key of the file is being used).

```
                TIP/30 Query Language              04 JUL 85  09:48
                                                              TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▶
REC BY EMP-NAME

.....
```

This command would invoke the "REC" display starting with the first employee in sequence by the field EMP-NAME. The field EMP-NAME is specified as a secondary key of the file.

We now see (as the first record): "DAVE HARRISON" (EMPLOYEE #807) because that name appears first in the file when the file is accessed in sequence by **EMP-NAME:**



An important point to realise here is that the EMP-NAME field (in this example file) is basically a free-format field. It contains the employee's name in a "natural" format: first name followed by last name. A very important consequence is that the computer will sort these names in order according to the characters in the field (from left to right).

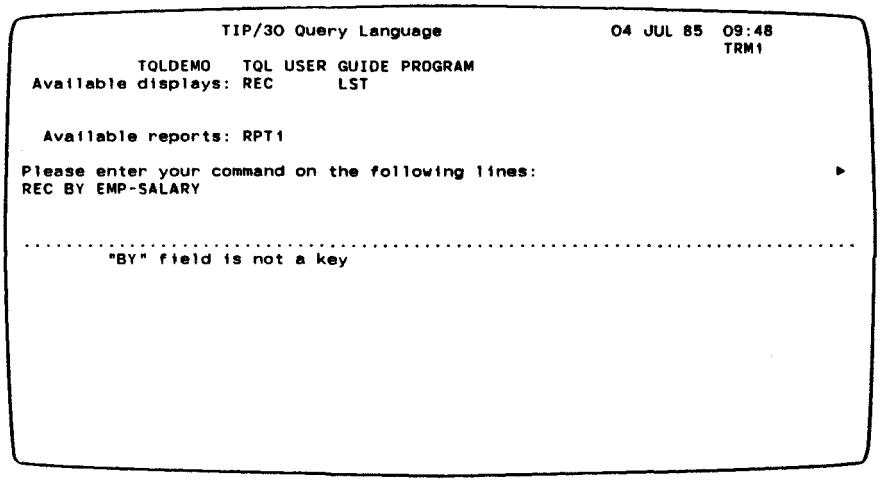
This explains why "Dave Harrison" appears before "Ronald Dawson" ("D" is before "R"). Although most humans would have unconciously used the last name to sort the names, the field EMP-NAME was NOT defined in such a way to make that sort possible.

Pressing F2 (to view the next record) would then display the following record:

```
TF$TQLU1
          EMP-NUMBER  EMP-NAME
                5645  DONALD TRACEY

          EMP-ADDR
           8911 EAST 53 STREET
           NEW YORK, N.Y.
           U.S.A

          EMP-SALARY        EMP-BIRTH-DATE
           34,000.00              05/11/44     <_>
                                  MM DD YY




F1/5:Refresh screen  F2/6:Next screen  F4:Update  Msg-wait:Menu
```

and so on ...

If the field name specified in the BY clause is not a properly defined key for the file, the following message will appear:



If this occurs, the terminal operator should consult the documentation provided with the TQL program to determine which field or fields represent the keys for the file.

1.13 <SUM> CLAUSE SUM

TQL provides the capability for the user to total (or sum) numeric fields. The summing process can be requested as a side-effect of some other TQL command (eg: in conjunction with a pre-defined display or report) or may be invoked directly as a command.

Up to **seven** field names may be specified after the reserved word SUM. Each field specified will be summed separately.

It is important to realize that ONLY the records that are actually processed are included in the final total.

The field names that are specified must be numeric fields.

When the user returns to the standard TQL prompt screen (after viewing the data or whatever), TQL will display a summary of the the field names and the totals.

If the SUM clause is specified in conjunction with a report an additional page will be printed (at the end of the pages generated by the report). This additional page will display the field names and the computed sums.

In the following example, SUM is used directly as a command. Since no range of operation is specified (a FROM or TO clause is not present) TQL will default to scanning the entire file:

```
                TIP/30 Query Language              04 JUL 85  09:49
                                                          TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▶
SUM EMP-SALARY

.....................................................................
                8 RECORDS
        DATA FIELD                    TOTAL        AVERAGE  #COUNTED
        EMP-SALARY                174,550.00      21,818.75         8
```

Note that one field is summarized on each line and that the total, the average and the number of records involved is shown.

1.14 <IF> CLAUSE IF

One of the most powerful TQL facilities is the ability to specify conditions that must be met before a record is processed by the actual TQL command (to control record selection for a display, report, update etc).

The TQL <IF> clause allows the user to "filter" records for the command by defining the acceptable contents of one or more fields.

In the IF clause, an expression (or several expressions) may be given. Each expression is a **qualification condition** that a record must meet before being considered for processing by the associated TQL command (a pre-defined display, report or other command).

An expression normally involves one or more fields from the record and an **operator**. The next paragraphs describe the various operators that are available - a simple example of an IF clause is:

```
IF EMP-SALARY > 13000
```

All of the standard arithmetic and relational operators are available along with a number of operators that are unique to TQL.

T Q L O P E R A T O R S

Symbol	Description	Alternative Symbol	Type
+	Addition		N
-	Subtraction		N
*	Multiplication		N
/	Division		N
%	Remainder		N
=	Equality	EQ	N,C
<>	Inequality	NE	N,C
>	Greater than	GT	N,C
<	Less than	LT	N,C
>=	Greater than or equal	GE	N,C
<=	Less than or equal	LE	N,C
=*	Begins with	BEGINS WITH	C
=!	Does not begin with	DOES NOT BEGIN WITH	C
=:	Contains	CONTAINS	C
	Does not contain	DOES NOT CONTAIN	C

The TYPE notation above indicates on which type of arguments the operator may operate. "N" implies that the operator's arguments must be numeric fields or values. "C" implies that the arguments must be character fields or values.

In many situations, a field is compared to a specifc value. A specific value is called a literal. A numeric literal may be entered as a number - without any comma separators but with a decimal place if appropriate.

Valid numeric literals: 12.85
10000
0.50

Invalid numeric literals: .85
10,000
.50

Character literals must be entered as a string of characters enclosed in single quotes.

Example character literals:
```
'JONES'
'123 MAIN STREET'
'18.50'
```

Character literals are internally considered to be padded with spaces (on the right) during comparison operations.

<IF> CLAUSE

The IF clause also can make use of **connectors**. Connectors are used to link several expressions together. For example, the connector AND is used in this simple example to specify that both the left and right expressions must be satisfied:

```
IF (EMP-SALARY > 15000) AND (EMP-NAME CONTAINS 'JR')
```

T Q L C O N N E C T O R S

Symbol	Description	Alternative Symbol	Type
&	Logical AND	AND	N,C
¦	Logical OR	OR	N,C
	Logical negation	NOT	N,C

Connectors and parentheses may be used to force a desired order of evaluation of a complex expression.

EXAMPLES OF <IF> CLAUSES

[1] IF EMP-SALARY GT 13000.75

[2] IF (EMP-SALARY * 1.10) < 25000

Check whether a 10% increase in salary will (still) be less than $25,000

[3] IF (EMP-NAME CONTAINS ' JR') AND (EMP-SALARY > 30000)

All favourite sons with salary over 30,000

[4] IF EMP-NAME BEGINS WITH 'ALLINSON'

[5] IF NOT (EMP-SALARY > 15500)

A (somewhat bizarre) way of saying: salary <= 15,500

<IF> CLAUSE

IF clauses must be specified as a sort of additional provision in conjunction with a TQL command (an IF clause cannot be used as a free standing TQL command).

```
                    TIP/30 Query Language                    04 JUL 85  09:49
                                                                        TRM1
            TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                          ▸
REC IF (EMP-SALARY * 1.10) < 25000
```

Would find the first record in the file (starting at the beginning of the file since a FROM clause was NOT given) that satisfies the constraint that the (salary+10%) is less than 25000.

```
                    TIP/30 Query Language                    04 JUL 85  09:49
                                                                        TRM1
            TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                          ▸
RPT1 IF EMP-SALARY > 25000
```

Would produce the pre-defined report RPT1 including only records for employees who have a salary greater than $25,000.

1.15 ADDING RECORDS ADD

A TQL program may allow the terminal user to add "new" information to the file that is being processed. To be able to accomplish this the program must have been pre-conditioned to allow the use of the "ADD" command. This section describes the use of the ADD command, but keep in mind that ADDing records may or may not be a feature allowed by all TQL programs.

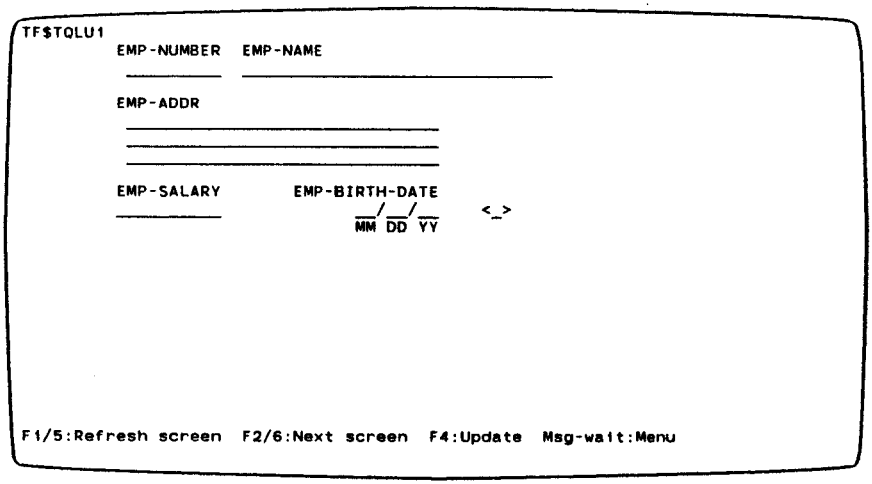
To be able to add information to our PAYROLL file (in the form of a new record) there **must** exist a pre-defined display that displays all of the fields of the record. (In fact, only the crucial fields need be present on the screen - in normal practice ALL fields would be present and could be entered by the terminal operator).

Obviously, if the record has more fields than would fit on the screen at one time, an ADD operation would have to be performed first to "create" the record (and some subset of the fields) and then a subsequent UPDATE could be performed (using another screen format with the balance of the fields). This grotesque possibility is not discussed here.

The syntax of the ADD command requires the reserved word "ADD" followed by the name of the pre-defined display that is to be used to determine the screen format to be used for the ADD operation:

```
                TIP/30 Query Language              04 JUL 85  09:49
                                                              TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▶
ADD REC
```

This command would cause TQL to display the "REC" pre-defined screen format in **update mode** (more about this mode in a moment):



Notice that all of the familiar fields from the "REC" display are on the screen, but each field has been filled with underscores (_____). This is what is meant by "update mode". The underscores are placed in the fields to aid the terminal user when data is being entered.

The terminal user can now enter the appropriate information in each field.

The **TAB BACK** and/or **TAB FWD** keys on the keyboard may be used to step backward or forward from one field to another. There is no need to remove any of the underscore characters. They are in the fields for visual reference ONLY and will be removed automatically by TQL when the data is put in the file.

When all relevant data has been entered, the user should place the cursor in the resting place provided and press XMIT to cause TQL to attempt to ADD the record.

Obviously, the EMP-NUMBER field is crucial because it is (in this case) the primary key for the PAYROLL file. TQL will attempt to create (ADD) a new record to the PAYROLL file with a primary key equal to the value that is entered in this field.

TQL will not allow a record to be added that has a primary key which already is on file. If an attempt is made to add a record with a key that already exists, the flashing error message:

Duplicate key

will appear at the top of the screen format after XMIT is pressed. The terminal user can then correct the key value and attempt to add the data again.

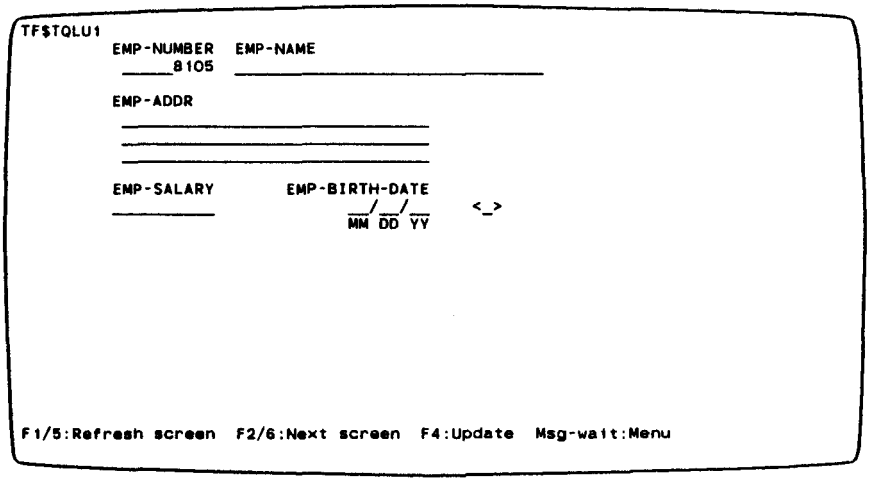
If the record is successfully added, TQL will return to the TQL prompt screen and display the flashing message:

Record added

If the terminal operator knows (in advance) the key of the record that is to be added, the ADD command may include the key value:

```
                    TIP/30 Query Language              04 JUL 85  09:49
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC       LST


  Available reports: RPT1

Please enter your command on the following lines:                    ▶
ADD REC 8105
```

This will cause TQL to copy the key information from the command into the screen format selected ("REC" in this case):



TQL will respond with the message:

Record already exists

if a record with the specified key already exists instead of blindly providing the screen format in update mode.

This method provides early warning that a record already exists and may be preferred because the terminal operator can avoid wasting time keying erroneous data.

A TQL program may have built-in restrictions on the values for certain fields. For example, in our TQL program, the programmer has insisted that ALL fields (on the REC display) must be entered (this means that the terminal operator MUST supply data for each field on the screen).

If a field is designated as a mandatory field the terminal operator must supply a value for that field. For character fields, this means that the terminal operator cannot leave the field empty; for numeric fields, the field must contain a non-zero value.

If a particular field must be present (or must have data within a specific range of values) and does not meet these constraints when XMIT is pressed, TQL will automatically cause the field to blink (flash) and will also display a flashing error message at the top of the screen.

For example, if the user does not enter data for the EMP-NAME field, that field (the underscores) will blink and the flashing error message:

EMP-NAME?

will appear at the top of the screen format.

The terminal operator should correct the field identified and press XMIT again. Before pressing XMIT, the terminal operator should check other fields to see if there are other opportunities for a correction.

Some fields may simply be a mandatory field (EMP-NAME); other fields may be restricted to a specific range (eg: EMP-SALARY must be greater than zero). Such field constraints are normally documented for the user of the TQL program by the Data Processing staff at your site.

If the terminal user decides that the ADD command is to be aborted (cancelled), then the terminal user should simply press MSG-WAIT.

TQL will return to the TQL prompt screen and display the flashing message:

Record NOT added

This message will confirm that the previous ADD command was not performed.

1.16 DATA ENTRY MODE ENTER

If a number of records have to be added to the file at the same time, the terminal operator could use the ADD command over and over to ADD each record.

TQL provides an alternative method to approach this situation - the ENTER command. The ENTER command is syntactically similar to the ADD command (see previous section) but it acts as a continuous repetition of an ADD command.

When an ADD command completes (or is aborted by the terminal operator) TQL will return to the standard TQL prompt screen. The ENTER command, on the other hand, will immediately assume that another record is to be added and will (in effect) automatically prepare for another ADD operation.

When the last record has been added via the ENTER command (and a fresh screen format is displayed), the terminal user must press MSG-WAIT to terminate the ENTER processing.

Note that this final pressing of MSG-WAIT will return the user to the standard TQL prompt screen with the flashing message:

Record NOT added

thus indicating that the ENTER operation has been terminated by the user.

Also note that the ENTER command ignores any key value that may be supplied along with the ENTER command (whereas the ADD command would copy such information into the first screen format).

1.17 DELETING RECORDS DELETE

A TQL program may allow the terminal user to delete information from the file that is being processed. To be able to accomplish this the program must have been pre-conditioned to allow the use of the "DELETE" command. This section describes the use of this command, but keep in mind that the ability to DELETE records may not be allowed by some TQL programs.

The DELETE command requires:

- the reserved word "DELETE" followed by
- optional name of a pre-defined display followed by
- the key of the record to be deleted

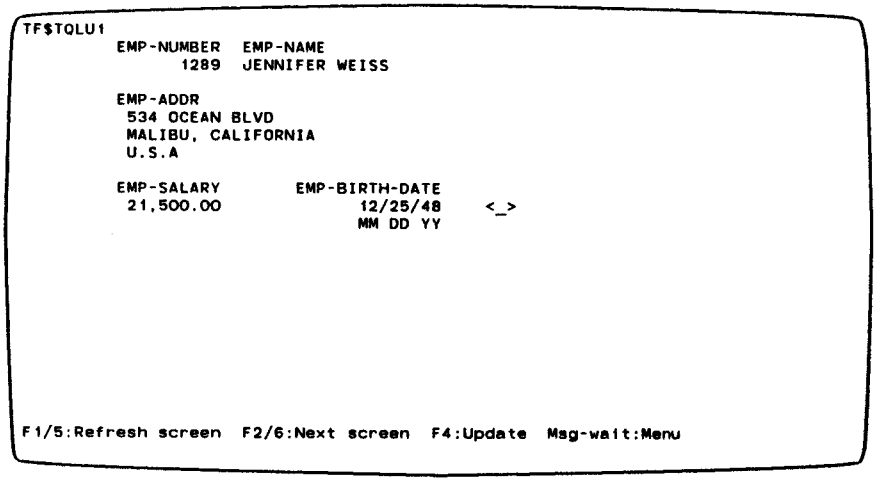
The pre-defined display format is specified because TQL will always display the record to be deleted and request confirmation from the terminal operator that this is indeed the correct record.

If the DELETE command does not specify a pre-defined display, TQL will (by default) use the first pre-defined display defined in the TQL program.

For example, to delete PAYROLL record number 1289, the terminal user would key in:

```
                TIP/30 Query Language              04 JUL 85  09:49
                                                              TRM1
        TQLDEMO   TQL USER GUIDE PROGRAM
Available displays: REC      LST

 Available reports: RPT1

Please enter your command on the following lines:                    ▶
DELETE REC 1289
```

TQL would react by using the pre-defined display "REC" to display the existing information for record number 1289 (presuming of course that record 1289 did exist - otherwise an error message would appear instead):



The flashing informational message:

Press F2 to delete record

will appear at the top of this screen.

The terminal user may press function key 2 (F2) to cause TQL to delete the record OR may press any other function key to cancel the delete command.

This verification procedure is designed to minimize the chances of deleting the wrong record.

1.18 UPDATING INFORMATION UPDATE

A TQL program may allow the terminal operator to update information that is contained in a pre-defined display. Remember that some TQL programs specifically do not allow records to be altered - some TQL programs may allow only certain fields to be changed.

The **UPDATE** (or the alternative spelling "**CHANGE**") command requires:

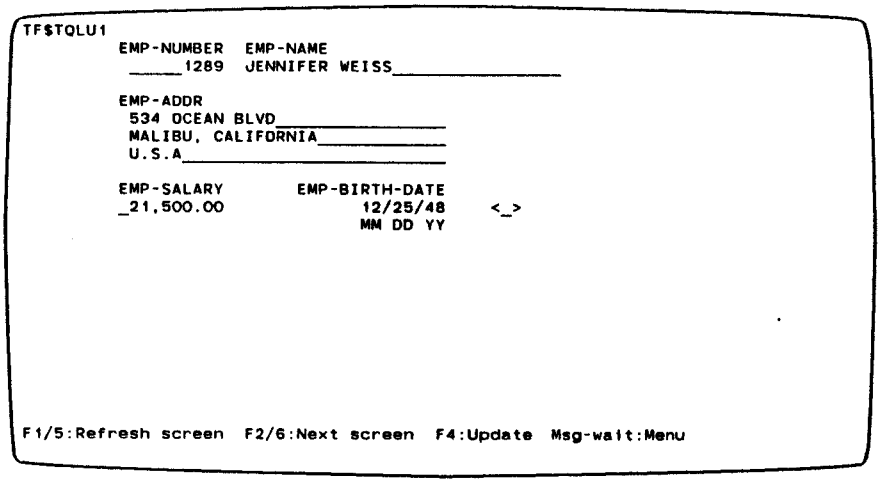
- the word "UPDATE" (or "CHANGE") followed by
- the name of a pre-defined display that is to be used to control the fields that are displayed in update mode
- the key of the record that is to be updated (OR some other clause that implies record selection eg: BY, FROM, IF, TO)

[Note that the pre-defined display that is used to update a record should be a display that involves a single record (in our example the best choice is REC; the LST display involves multiple records and is not suitable for ADD/ENTER/UPDATE/CHANGE commands).]

For example:

```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▶
UPDATE REC 1289
```

Would cause TQL to find record 1289 (presuming that such a record exists) and using the pre-defined display "REC", display the current information in "update mode":



Note that the fields are displayed with underscore characters to indicate the size of each field. The terminal user may now alter any of the information (except the primary key information) and press XMIT to cause the record to be updated.

The TQL program may be programmed to prevent the alteration of certain fields. Although the terminal user may be able to alter information on the screen for such a field, TQL will not change such fields in the file.

[The Data Processing department would normally provide information for the terminal user to identify which fields may or may not be changed (or, for example, define the screen format to specify that such fields are "protected" and physically not changeable on the screen).]

If the TQL program has placed restrictions on the value of fields or the range of allowed values those restrictions will be enforced by the UPDATE/CHANGE command (see discussion of field restrictions in the description of the ADD command for further details).

If the terminal operator decides to abort the UPDATE or CHANGE, pressing MSG-WAIT will cause TQL to return to the standard TQL prompt screen with the message:

Record NOT updated

to indicate that the update/change operation was not carried out.

Note that TQL will automatically offer for update all records that match any selection criteria specified in the original command.

That is, a command such as:

```
                    TIP/30 Query Language                  04 JUL 85  09:49
                                                                      TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC       LST


  Available reports: RPT1

Please enter your command on the following lines:                          ▶
UPDATE REC IF EMP-SALARY > 25000


.....................................................................................
```

will cause TQL to present (in update mode) all records with a salary field greater than $25,000.

The user may choose to update each record in turn (by making any desired changes and pressing XMIT) and move on to the next record OR may terminate the UPDATE/CHANGE command at any point by pressing MSG-WAIT.

When control returns to the standard TQL prompt screen, TQL will display a message indicating the number of records that were updated (if any records were updated).

NOTE: It is possible that information (from the file) that is displayed on the terminal was perfectly valid at the point in the past when that record was entered. This information may NOW be unacceptable to the TQL program and may be flagged as incorrect during an UPDATE/CHANGE operation. The user should be aware that TQL may flag fields that were not entered or altered at the terminal.

1.19 FREE-FORMAT LIST

LIST

If the TQL program does not provide a pre-defined display that contains the correct combination fields for the user's purposes, the terminal user can use the free-format "LIST" command to dynamically generate a multiple record display.

The LIST command requires the reserved word "LIST" to be followed by the names of one or more fields to be listed. The field names must be enclosed in parentheses.

For example, to generate a list display in the TQLDEMO program which includes (only) the EMP-NAME and the EMP-SALARY fields, the following command should be entered:

```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                      ►
LIST (EMP-NAME EMP-SALARY)

.....................................................................
```

This command would (according to the information in our test file) display the following screen of information:

FREE-FORMAT LIST

```
EMP-NAME                        EMP-SALARY      PAGE   1

JOHN SMITH JR.                    12000.00
MARY JOHNSON                      18500.00
DAVE HARRISON                     18750.00
WILLIAM MARTIN                    20500.00

JENNIFER WEISS                    21500.00
MICHAEL HARRIS                    34500.00
RONALD DAWSON                     14800.00
DONALD TRACEY                     34000.00

▸NEXT     ▸CLOSE          [Press MSG WAIT to return to menu]
```

Notice that TQL has automatically supplied headings for the fields and has listed several records (all records in our example). (A deliberate blank line appears after every four lines to improve readability of the display).

At the bottom of the screen there are two options that may be selected:

- NEXT: display the next screen full of information
- CLOSE: terminate the TQL program

The user may use the TAB BACK/FWD key to position the cursor after the desired option and press XMIT.

Another option (as described in the screen format) is to press MSG WAIT. This will return the user to the standard TQL prompt screen (effectively terminating the LIST command).

FREE-FORMAT LIST

The LIST command allows the user to make a personalized selection of field names to view. However, TQL will truncate the display at 80 columns (if necessary). This means that the resulting display may not be wide enough to accommodate all the fields requested.

NOTE: the LIST command can also include other TQL clauses that are available with other commands (BY, FROM, TO, IF, SUM etc) to further control the selection of information that is listed:

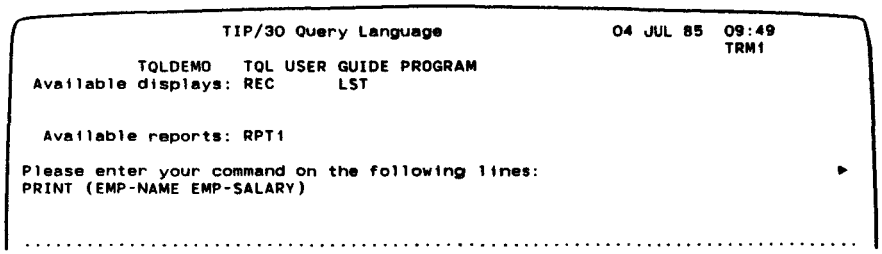
```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                      ▸
LIST (EMP-NAME EMP-SALARY) IF EMP-SALARY > 16000
```

1.20 FREE-FORMAT PRINT PRINT

If the TQL program does not provide a pre-defined report that contains the correct combination fields for the user's purposes, the terminal user can use the free-format "PRINT" command to dynamically generate a free-format report on the main site printer, a terminal's auxiliary printer or to an MS-DOS file on a personal computer.

The PRINT command requires the reserved word "PRINT" to be followed by the names of one or more fields to be printed. The field names must be enclosed in parentheses.

For example, to generate a report in the TQLDEMO program which includes (only) the EMP-NAME and the EMP-SALARY fields, the following command should be entered:



This command would (according to the information in our test file) create a report like the following. The report would be generated on the main-site printer (the default destination).

```
_____top of page_____

        FOR  username       850320  11:31       TERMINAL:TRM1
PRINT (EMP-NAME EMP-SALARY)

_____top of page_____

EMP-NAME                          EMP-SALARY     PAGE   1

JOHN SMITH JR.                       12000.00
MARY JOHNSON                         18500.00
DAVE HARRISON                        18750.00
WILLIAM MARTIN                       20500.00

JENNIFER WEISS                       21500.00
MICHAEL HARRIS                       34500.00
RONALD DAWSON                        14800.00
DONALD TRACEY                        34000.00
```

FREE-FORMAT PRINT

Notice that TQL has automatically supplied headings for the fields and has printed several records (all records in our example). (A deliberate blank line appears after every four lines to improve readability of the printout).

The PRINT command allows the user to make a personalized selection of field names to print. However, TQL will truncate each print line at 132 columns (if necessary). This means that the resulting report may not be wide enough to accommodate all the fields requested.

NOTE: the PRINT command can also include other TQL clauses that are available with other commands (BY, FROM, TO, IF, SUM etc) to further control the selection of information that is printed.

For additional information refer to the preceding sections "USING A PRE-DEFINED REPORT" and "TQL PRINT DESTINATIONS".

The PRINT command is effectively an automatic report generator with user-selected fields.

1.21 TRANSFER DATA TO PC EXPORT

TQL provides the capability to transfer data from the main computer to a Personal Computer. This capability can only be utilized if the user is actually running the TQL program on a Personal Computer that is equipped with the Sperry Terminal Emulator Package (STEP) or the Computer Logics Personal Emulator Package (PEP).

These packages are a combination of hardware and software for the personal computer.

The data processing personnel at your site can advise you whether or not this hardware and software is installed in your personal computer.

This section discusses the use of the EXPORT command assuming that all of the requisite hardware and software is in place and functioning correctly. The assumption is also made that the reader is familiar with the operation of the Personal Computer and the features of the Personal Computer operating software (MS-DOS).

The EXPORT command employs a syntax which is deliberately similar to the syntax of the free-format "PRINT" command (see also previous section).

The EXPORT command requires:

- the reserved word "EXPORT" followed by
- a list of field names to be exported (names enclosed in parentheses)
- an ON clause specifying the name of the file to receive the data (review earlier section on PRINT DESTINATION NAMES).

For example, we could export several fields from the PAYROLL file as follows:

```
                    TIP/30 Query Language                  04 JUL 85  09:49
                                                                      TRM1
            TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC       LST


  Available reports: RPT1

Please enter your command on the following lines:                          ▶
EXPORT (EMP-NUMBER EMP-NAME EMP-SALARY) ON A:TEST


.....................................................................................
```

This command would (according to the information in our test file) create a file on the "A" disk drive of the personal computer.

The file name created would have the name **A:TEST.PRN** (the .PRN file extension is automatically provided by TQL).

The EXPORT command operates by writing the data to the screen of the Personal Computer and then causing the data on the screen to be copied to the Personal Computer's disk drive. This process will be repeated until all of the data is transferred to the disk drive.

The EXPORT command (like most other TQL commands) may also include other standard TQL clauses to qualify the records that are to be processed (BY, FROM, TO, IF etc).

The data in the MSDOS file is limited (by TQL) to a maximum width of 160 characters (and will be truncated if necessary). This means that there is a limit to the number of fields that may be specified with one EXPORT command.

The MSDOS file that we have created with the previous example TQL command would contain the following data:

```
  101  "JOHN SMITH JR.              "      12000.00
  187  "MARY JOHNSON                "      18500.00
  807  "DAVE HARRISON               "      18750.00
 1024  "WILLIAM MARTIN              "      20500.00
 1289  "JENNIFER WEISS              "      21500.00
 3356  "MICHAEL HARRIS              "      34500.00
 3376  "RONALD DAWSON               "      14800.00
 5645  "DONALD TRACEY               "      34000.00
```

Notice that numeric fields (EMP-NUMBER, EMP-SALARY) are output as simple numeric values while alphanumeric fields (EMP-NAME) are output as character strings enclosed in double quote marks.

This format is accepted by some PC-based software (LOTUS 1-2-3 and SYMPHONY for example).

1.22 SELECT RECORDS SELECT

There may be occasions when the terminal user wishes to perform several operations on a well-defined subset of a large file. An example of a typical subset might be "all employees who have a salary greater than $20,000".

A brute force approach would be to perform each of the intended operations by always including a TQL <IF> clause:

```
... ... ... IF EMP-SALARY > 20000
```

This approach is (usually) very poor because TQL must read the entire file every time the IF clause is specified.

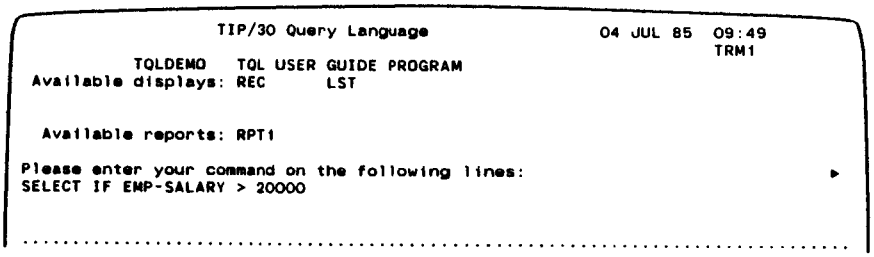
Our example file has only 8 records - brute force would not be particularily disadvantageous in this case. If the PAYROLL file had several thousand records however, the time required to (repeatedly) read the entire file would be prohibitively high.

A better approach is to use the TQL SELECT command.

The TQL SELECT command allows the terminal user to specify (once) which records are to be taken as the desired subset of the main file. After this "selection" (or "extraction") is performed, all subsequent TQL commands will apply to this subset of the main file.

TQL accomplishes this by creating an index (to the records of the file that meet the specification) that is used for all TQL commands that are then issued.

For example:



would read the entire file (since a <FROM> or <TO> clause was not provided) and create an internal index containing pointers to the records that had an EMP-SALARY field with a value exceeding $20,000.

TQL would respond to this command (after creating the index) by displaying the following information in the standard prompt screen:

```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                      ▸
SELECT IF EMP-SALARY > 20000


.....................................................................
           4 records
```

To indicate the number of records that were "selected" and are now in the internal index to the file.

A flashing informational message would also appear at the top of the display:

*** Selection active!**

This reminds the user that TQL is now working with a subset of the file (rather than the entire file).

The terminal user may now issue any of the normal TQL commands as already described in this guide (pre-defined displays/reports, LIST, PRINT, SUM, COUNT etc) **with an important difference:** now that a SELECTion is active, the TQL commands operate **only** on the subset of records that were selected.

Another possibility that the terminal user may employ is to issue another SELECT command. This would have the effect of pruning the previous selection according to an additional condition.

WARNING: the SELECT command always creates a new internal index which points to the records that match the stated <IF> clause. IF you specify a condition that results in no "winners", you will reduce the selected index to zero records!

This means that the terminal user must be extremely careful when issuing multiple SELECT commands. For example, 1000 records (say) might be selected by the first SELECT command and the second SELECT command may find no records which satisfy it - the result would be 0 records in the selection index (thereby wasting all the machine time invested in performing the first SELECT).

A better approach would be to combine multiple constraints into a single (possibly complex) <IF> clause on the initial SELECT command.

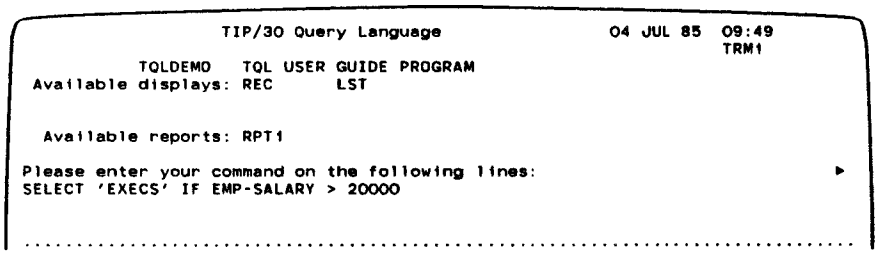
Under normal circumstances, a SELECTion remains active until:

- the user exits from the TQL program OR
- the user accidently SELECTs zero records OR
- the user issues a DROP command (the topic of the next section of this guide)

There may be situations where it is quite meaningful to retain a SELECTion for future use (ie: tomorrow, next week etc). (If a selection is used later and it happens to point to a record that is no longer on file, TQL will simply ignore that pointer).

To be able to retain a SELECTion, the terminal user must (at the time the SELECT is performed) provide a **NAME** for this particular SELECTion.

This name is restricted to a maximum of 8 characters (the first of which must be alphabetic). The naming process is accomplished as shown in this example:



The selection name must be placed in single quotes and must follow the word "SELECT". (It should be clear that a SELECTion that is performed without a name is a purely temporary selection and will not be retained by TQL if the user exits from the TQL program).

The selection illustrated above can be recalled at some point in the future by simply selecting just the name:

```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                      ▶
SELECT 'EXECS'
```

This will cause TQL to search for an existing SELECTion by the name 'EXECS'.

A so-called "named" selection can only be used by the user that created it and in the same TQL program that was used to create it.

BEWARE: the system administrator of your site may delete named SELECTions from time to time if they appear to be "forgotten" or have not been used for some time.

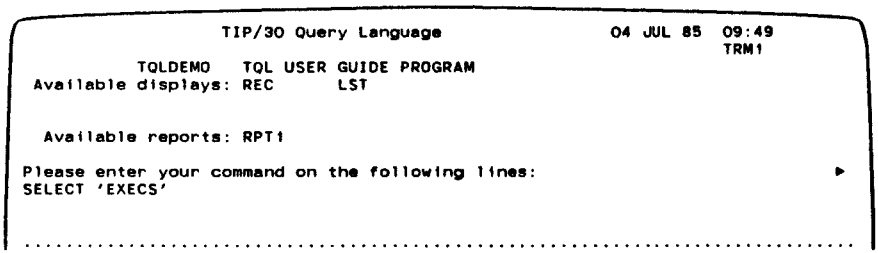
NOTE: there is no mechanism provided in this version of TQL to help a user discover what selections have been "named" in this manner. The user is advised to keep track of these selection names or to discuss naming conventions with the system administrator.

1.23 DELETE EXISTING SELECTION DROP

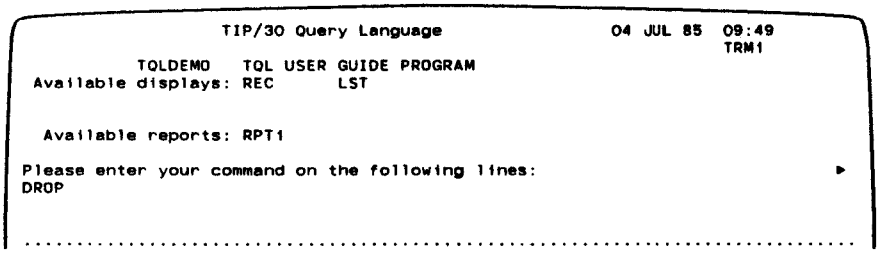
The TQL command "DROP" may be used to discard a named SELECTion that is no longer needed (also see section describing the SELECT command).

Only the user that created the SELECTion can issue a DROP command to discard the selection.

The following example illustrates how a terminal user could discard a SELECTion that was named 'EXECS':



This would make the named selection "EXECS" the active selection.



will drop the currently-active selection (namely: EXECS).

1.24 SORT A SELECTION SORT

The data that is contained in a file can normally be retrieved in sequence by an existing key of that file. The keys of a file are determined in advance by the Data Processing Department.

There are times that the terminal user wishes to display data or generate a list or report and have the data appear in an unusual sequence (that is, any sequence that is not directly provided in advance by the normal key or keys for the file).

For example, the PAYROLL file that we have been using to illustrate this user guide has (for simplicity) been defined with a single key: EMP-NUMBER. This primary key is probably sensible for most of the possible uses of the data in this file.

Imagine that management wanted a report of employees in increasing order by SALARY. With the key structure of the PAYROLL file this would normally be impossible because the SALARY field is neither a primary nor secondary key for the file.

The TQL SORT command may be used to SORT the file according to one or more selected field names!

The SORT command requires the following:

- A SELECT must be currently active for the TQL sesssion;
- The reserved words "SORT BY" followed by
- one (or more than one) field name to define the sort sequence (do not enclose the names in parentheses)

NOTE:

- The SORT command is currently restricted to ascending sequence only.
- A prior selection MUST be made (even if it is an "unconstrained" selection)
- The fields specified as the SORT fields must be specified in order from most significant to least significant (major to minor order).

SORT A SELECTION

The SORT command does not sort the actual file; it SORTS the internal index that was created by the previous SELECT command (you may wish to review the section of this guide titled "SELECT RECORDS").

The SORT command will advise the terminal user (via a flashing informational message) about the progress of the SORT process. The number of records sorted (of the total to be sorted) will appear as the SORT progresses.

A subtle point to consider is that a SELECTION must be made. If the user wishes to SORT the entire file (usually this is rather prohibitive) an otherwise empty SELECT must be made.

We will illustrate this here since our PAYROLL file contains only 8 records and we wish to select all records in this case:

```
                    TIP/30 Query Language              04 JUL 85  09:49
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                      ▶
SELECT
```

This would result in the following response by TQL (in this case):

```
                TIP/30 Query Language                  04 JUL 85  09:49
                *       8 selected                                 TRM1
           TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▶


.....
       Selection active!
```

Note that this "unconstrained" SELECT has, in fact, selected all 8 records from our file.

We may now specify the intended SORT command:

```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                           TRM1
           TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▶
SORT BY EMP-SALARY
```

When the SORT has completed (very quickly in the case of 8 records) the following display will appear - note the flashing message in line 2 of the screen.

```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                                  TRM1
            SORT complete
         TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                    ▸
```

At this point, the SELECTion is still active BUT the internal index has been SORTED according to the EMP-SALARY field.

ALL commands from this point will display or report information in EMP-SALARY sequence:

```
                TIP/30 Query Language                  04 JUL 85  09:49
                                                                  TRM1
         TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                    ▸
LST
```

would produce the following display:

```
TF$TQLU2
 EMP-NUMBER  EMP-NAME                    EMP-SALARY  EMP-BIRTH-DATE
        101  JOHN SMITH JR.               12,000.00        01/30/45
       3376  RONALD DAWSON                14,800.00        09/11/37
        187  MARY JOHNSON                 18,500.00        04/13/51
        807  DAVE HARRISON                18,750.00        03/19/46
       1024  WILLIAM MARTIN               20,500.00        02/29/44
       1289  JENNIFER WEISS               21,500.00        12/25/48
       5645  DONALD TRACEY                34,000.00        05/11/44
       3356  MICHAEL HARRIS               34,500.00        01/01/48




F1/5:Refresh screen  F2/6:Next screen  F4:Update  Msg-wait:Menu _
```

The terminal user may specify an entirely different SORT command now to reorder the SELECTion in some other fashion before issuing other TQL display or report commands...

WARNING: Sorting by its very nature consumes computer resources at an ever-increasing rate. This rate is not linear! This means that the time and resources required increases dramatically as the number of records increases.

A reasonable rule of thumb is to limit sorting to less than 500 records (although this rule is utimately dependent on the type of computer being used, the current load on the system and many other factors).

1.25 SAVING TQL COMMANDS SAVE

TQL may be configured to allow the terminal user to SAVE a TQL command for later use.

The TQL SAVE command (and the companion RECALL command) may or may not be available at your site - the terminal user should consult the system administrator or the Data Processing staff.

The terminal user may wish to save a complex command that is used on a regular basis rather than typing the entire command in every time it is needed. (Obviously saving a command is sensible only when the command is particularily lengthy or the field names are tricky to remember or type).

The TQL command "SAVE" is the mechanism whereby a TQL command may be saved for later use.

The SAVE command requires the following:

- the reserved word SAVE optionally followed by
- the command text to save.

For example, let us assume that a PAYROLL report is to be generated on a regular basis. The terminal user does not wish to remember the details of this report and wishes to create a saved command to have "on file":

```
                TIP/30 Query Language                  04 JUL 85  09:50
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC       LST


  Available reports: RPT1

Please enter your command on the following lines:                    ▶
RPT1 IF EMP-SALARY > 16500


.....................................................................
```

This command (admittedly NOT that tedious to remember or type) could be SAVED by issuing the following TQL command:

```
                TIP/30 Query Language                  04 JUL 85  09:50
                                                                  TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST


  Available reports: RPT1

Please enter your command on the following lines:                      ►
SAVE RPT1 IF EMP-SALARY > 16500


.....
```

Note that we have illustrated the inclusion of the actual command text after the reserved word "SAVE". (If you do not do this, you will have the opportunity to enter this command text on the screen format which TQL will subsequently display).

TQL will respond by displaying the following screen format:

```
                TIP/30 Query Language                          TF$TQL3

----The command you wish to save is as follows:--------------------------------
RPT1 IF EMP-SALARY > 16500


--------------------------------------------------------------------------------

 This command is used with the program:  TQLDEMO
       Enter the name of this command:   _______

 Leave cursor here (_) and press XMIT
```

Notice that TQL has copied the command text that was provided with the SAVE command ("RPT1 IF EMP-SALARY > 16500") to this screen format and has automatically provided the name of the TQL program that is currently being executed.

SAVING TQL COMMANDS

The terminal user must provide a "name" for this saved command (the name specified is used in conjunction with the procedure used to RECALL a saved command - described in the next section of this guide).

For example, assume that we wish to call this saved command "Q1". Simply provide the name ("Q1") in the appropriate field:

```
                    TIP/30 Query Language                          TF$TQL3

----The command you wish to save is as follows:--------------------------------
RPT1 IF EMP-SALARY > 16500


--------------------------------------------------------------------------------

 This command is used with the program:  TQLDEMO
       Enter the name of this command:  Q1_____

 Leave cursor here (_) and press XMIT
```

and press XMIT to cause TQL to save this command (under the name "Q1").

TQL will save this command in a special internal system file. The command "Q1" is now available to be recalled by any user that is able to run the TQL program TQLDEMO. This command will remain SAVED until it is explicitly deleted.

Your system administrator can assist you if you wish to delete a saved command that is no longer needed.

The next section of this guide illustrates the procedure for using a command that has been saved in this manner.

1.26 RECALLING A SAVED COMMAND RECALL

TQL may be configured to allow the terminal user to RECALL a command that was saved using the TQL "SAVE" command.

The TQL RECALL command (and the companion SAVE command) may or may not be available at your site - the terminal user should consult the system administrator or the Data Processing staff.

The RECALL command does nothing more than retrieve the text of the command that was originally saved. This retrieved command text can be altered (if desired) before the XMIT key is pressed to enter the command.

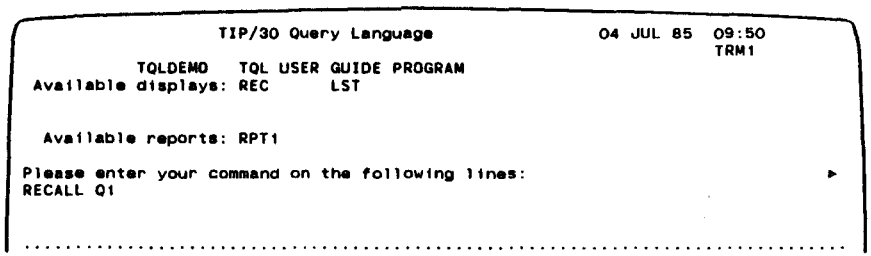
The RECALL command may be entered in one of two ways:

- RECALL xxxxxx
- RECALL

The first format (with the name of the saved command) can be used if the terminal user knows the name of the saved command that he wishes to recall. This format will either retrieve the specified command or complain that it could not be found (spelling error!?).

The second format (just the reserved word "RECALL") will display a "menu" of the available saved commands.

For example, assume that the terminal user is aware that a saved command named "Q1" exists. This command can be recalled (retrieved) by entering the following TQL command:



RECALLING A SAVED COMMAND

If there is indeed a saved command named "Q1", TQL will redisplay the command screen with the retrieved text of the command that was saved with the name "Q1":

```
                    TIP/30 Query Language                  04 JUL 85  09:50
                                                                      TRM1
            TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC       LST


  Available reports: RPT1

Please enter your command on the following lines:                          ▶
RPT1 IF EMP-SALARY > 16500


.....................................................................................
```

The terminal user can now make any desired modifications to this command and press XMIT to have TQL process the command. Of course, the terminal user can also change his mind and enter some other command.

The SAVE/RECALL facility simply allows the terminal user to quickly and easily retrieve commands that are used frequently or are so complex that they warrant saving.

If the terminal user cannot remember the name of a saved command or just wishes to "browse" through the saved commands for something to do, the second format of the RECALL command may be used:

```
                TIP/30 Query Language              04 JUL 85  09:50
                                                              TRM1
          TQLDEMO   TQL USER GUIDE PROGRAM
 Available displays: REC      LST

  Available reports: RPT1

Please enter your command on the following lines:                    ▸
RECALL

.....
```

When the RECALL command is issued without a following command name, TQL will react either by stating that there are:

NO SAVED COMMANDS AVAILABLE

or by displaying a "menu" of saved commands:

RECALLING A SAVED COMMAND

```
TIP/30 Query Language  --- Summary of saved commands for TQLDEMO        TF$TQL6

 Enter selection:
       parameters: _____________________________________________________ [_]
       Msg-wait: no selection          F2: next page of commands

Selection: 1      Saved Name: Q1
RPT1 IF EMP-SALARY > 16500


Selection: 2      Saved Name: Q2
LST IF EMP-SALARY < 20000 AND EMP-NAME BEGINS WITH 'D'


Selection:        Saved Name:
```

The terminal user can now:

- Press MSG-WAIT to cancel the RECALL command
- Press F2 to view the next screen of saved commands (if there are more available)
- Enter the number of the command desired (in line 3 of the format) and press XMIT to have that particular command recalled.

The "parameters" field is an area where optional command parameters may be entered. That feature is not discussed in this guide - the terminal user can discuss this more advanced feature with the system administrator.

1.27 TQL COMMAND REFERENCE Commands

These notes apply to the description of the syntax of TQL run-time commands which follows this section:

- items that are presented in a **bold** typeface must be entered exactly as shown.
- items that are in square brackets [] are considered to be optional entries.
- a number appearing in braces {} is a reference to a numbered note that appears after the presentation of the syntax.
- the use of the word "quote" or "quotes" always means the single quote character (eg: ').

<display> the name of a pre-defined TQL display

<report> the name of a pre-defined TQL report

<field> the name of a field in a record or the WORKING-STORAGE area of the TQL program

The field name may be followed by a subscript (that is, a number in parentheses) if a subscript is appropriate for that particular field

<key> a key value (enclosed in quotes if the value is not strictly numeric).

If a key field is defined as multiple fields, more than one value must be supplied; each value separated from the preceding value by a space or a comma.

<pkey> a (possibly partial) key value used generally to indicate the relevant portion of the key that is desired.

If the key field is a numeric field, trailing zeroes are required when specifying a partial key value.

EG: 'SM' or 25000

<keyfield>	The name of a field which is a key of a file that is being used. A keyfield is normally a secondary key field.
<expr>	A relationship between a field and a literal value or between two fields. EG: EMP-SALARY <> 25000 An expression may be combined with other expressions by using a TQL connector or parentheses: EG: (EMP-SALARY <> 25000) AND (EMP-NAME CONTAINS 'JR')
...	Indicates that the preceding item may be repeated a number of additional times.
...n...	Indicates that the preceding item may be repeated up to n additional times (n is a number).
<printer>	The name of a printer in the system. EG: PRNTR,AUX1,B:TEST,AUX1*BYP etc
'name'	An arbitrary name (maximum of 8 characters, first of which must be alphabetic, enclosed in quotes).
<TQL command>	The text of an entire TQL command.
<saved name>	The name by which a TQL command was saved.
<value>	A literal value. Enclosed in quotes unless the value is strictly numeric. Numeric values must not contain comma separators and must begin with a digit (0 thru 9). If appropriate, a numeric value may contain a decimal place and trailing decimal digits.
<program>	The name of a TQL program.

Syntax:

```
<display>  [ <key>          ] {1}
           [ BY <keyfield>  ] {2}
           [ FROM <pkey>    ]
           [ TO <pkey>      ]
           [ IF <expr>      ]
           [ SUM <field> [ ...6...] ]
```

{1} If a <key> value is supplied, all other clauses are superfluous since the presence of an explicit <key> value implies a request for a particular record.

{2} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

Syntax:

```
<report>   [ BY <keyfield> ] {1}
           [ FROM <pkey>   ]
           [ TO <pkey>     ]
           [ IF <expr>     ]
           [ SUM <field> [ ...6...] ]
           [ ON <printer>  ] {2}
```

{1} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

{2} If the ON clause is not specified, the report will be directed to the default printer as specified in the TQL program.

Syntax:

```
ADD         [ <display>     ] {1}
            [ <key>         ] {2}
```

{1} If the <display> is omitted, the first available display in the TQL program is assumed.

The <display> controls the screen format that is to be used to collect the input data.

{2} If a <key> value is provided, the value will be carried forward into the (primary) key field(s) in the <display>.

TQL will also verify (in advance) that a record with the specified <key> does not already exist.

Syntax:

CLOSE {1}

{1} The current TQL program is terminated.

Syntax:

```
COUNT [ <display> ] {1}
                    [ FROM <pkey>    ]
                    [ TO <pkey>      ]
                    [ IF <expr>      ]
                    [ SUM <field> [ ...6...] ]
```

{1} If the <display> is omitted, the first available display in the TQL program is assumed.

The <display> determines which record (type) is to be counted (if there happen to be multiple record types defined in the TQL program).

Syntax:

DELETE [<display>] {1} **<key>**

{1} If the <display> is omitted, the first available display in the TQL program is assumed.

The <display> controls the screen format that is to be used to display the record for verification of the DELETE.

Syntax:

DROP ['name'] {1}

{1} If the name is omitted, the current selection is discarded and TQL continues without a selection in effect.

If the name is provided and is the retained name of the currently active selection, the retained selection will be discarded.

Syntax:

END {1}

{1} The current TQL program is terminated.

Syntax:

ENTER [<display>] {1}

{1} If the <display> is omitted, the first available display in the TQL program is assumed.

The <display> controls the screen format that is to be used to collect the input data.

Syntax:

```
EXPORT ( <field> ... )
                    [ BY <keyfield> ] {1}
                    [ FROM <pkey>   ]
                    [ TO <pkey>     ]
                    [ IF <expr>     ]
                    [ SUM <field> [ ...6...] ]
                    [ ON <printer>  ] {2}
```

{1} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

{2} If the ON clause is not specified, the information will be exported to MSDOS file A:TQLDTA.PRN

Syntax:

```
EXPORT <report>
                    [ BY <keyfield> ] {1}
                    [ FROM <pkey>   ]
                    [ TO <pkey>     ]
                    [ IF <expr>     ]
                    [ SUM <field> [ ...6...] ]
                    [ ON <printer>  ] {2}
```

{1} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

{2} If the ON clause is not specified, the information will be exported to MSDOS file A:TQLDTA.PRN

Syntax:

```
LIST  ( <field> ... )
                    [ BY <keyfield> ] {1}
                    [ FROM <pkey>   ]
                    [ TO <pkey>     ]
                    [ IF <expr>     ]
                    [ SUM <field> [ ...6...] ]
```

{1} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

Syntax:

```
MORE  {1}
```

{1} More "child" record information (if available) is displayed.

An equivalent to this command is function key F9.

Syntax:

```
NEXT  {1}
```

{1} The next screen full of data is displayed (if available).

An equivalent to this command is function key F2.

Syntax:

```
OPEN  <program>  {1}
```

{1} The current TQL program is terminated and TQL attempts to OPEN the specified <program>.

Syntax:

```
PRINT  ( <field> ... )
                    [ BY <keyfield> ] {1}
                    [ FROM <pkey>   ]
                    [ TO <pkey>     ]
                    [ IF <expr>     ]
                    [ SUM <field> [ ...6...] ]
                    [ ON <printer>  ] {2}
```

{1} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

{2} If the ON clause is not specified, the information will be directed to the system printer (PRNTR).

Syntax:

```
RECALL  [ <saved-named> {1} [ <value> ... ] ] {2}
```

{1} If the <saved-name> is provided, TQL will attempt to RECALL a command which was previously saved by that name.

Otherwise, a menu of commands that may be recalled will be presented.

{2} These (optional) values may be used to supply values for parameters that are anticipated by the command that is to be recalled.

Syntax:

```
SAVE  [ <TQL-command> ] {1}
```

{1} If the <TQL command> is provided, the text of the command will be carried forward to the subsequent screen format.

Syntax:

```
SELECT [ 'name' ] {1}
                  [ BY <keyfield> ] {2} {3}
                  [ FROM <pkey>   ]     {3}
                  [ TO <pkey>     ]     {3}
                  [ IF <expr>     ]
```

{1} If a 'name' is specified an existing selection (by that name) will be made active.

If an existing selection (by that name) cannot be found this selection will be retained by the name specified.

{2} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

{3} This clause may only be used on an initial SELECT (ie: if no SELECTION is in effect).

Syntax:

```
SHOW <display>
SHOW <report>
```

Syntax:

```
SORT BY {1}  <field> ... {2}
```

{1} The SORT BY command cannot be used unless it has been preceded by a SELECT command.

The SORT is restricted to ascending order only.

{2} At least one field must be specified.

If more than one field is specified, the first field is considered the major sort field and the subsequent fields are minor sort keys.

Syntax:

```
SUM  <field> [...6...]   {1}
                  [ BY <keyfield> ] {2}
                  [ FROM <pkey>   ]
                  [ TO <pkey>     ]
                  [ IF <expr>     ]
```

{1} At least one field may be specified; up to seven fields may be specified to SUM.

{2} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

Syntax:

```
UPDATE  {1} [ <display> ] {2}  [ <key> ] {3}
                          [ BY <keyfield> ] {4}
                          [ FROM <pkey>   ]
                          [ TO <pkey>     ]
                          [ IF <expr>     ]
                          [ MOVE <expr> TO <field> ... ] {5}
```

{1} A valid synonym for **UPDATE** is **CHANGE**.

{2} If <display> is omitted, the first defined display in the TQL program is assumed.

The <display> controls the format of the data that may be altered.

{3} If the <key> value is specified, all other clauses are superfluous since the implication is that a single record is to be updated.

If the <key> value is omitted, all records which match the specified BY/FROM/TO/IF clauses will be presented (in turn) for possible alteration.

{4} If the BY clause is used, it must precede any other clause which refers to a <key> or <pkey> value.

{5} The MOVE statement is executed immediately before the record is displayed in update mode.

1.28 TQL FUNCTION KEYS FKEYS

Function Key Summary

MSG-WAIT Synonym for **END** command if detected at the standard TQL prompt screen.

May also be used to abort a command and return to the standard TQL prompt screen.

F1 / F5 If pressed once, will redisplay the last output screen.

If pressed again, will cause TQL to display (to a maximum of 10) previous output screens.

F2 / F6 When a pre-defined display or the **LIST** command is being used, F2/F6 is equivalent to the **NEXT** command and will advance the display to the next screen.

If this function key is pressed when a record is currently being displayed in "update mode", TQL will NOT update that record and will proceed to the next record (if any).

F4 / F8 When a pre-defined display is being used, F4/F8 will cause TQL to go into UPDATE mode for the record that is currently displayed on the terminal. The screen will be re-displayed in UPDATE mode.

F9 When a pre-defined display or the **LIST** command is being used, F9 is equivalent to the **MORE** command and will advance the display to the next group of "child" records (if such records are supported by the executing TQL program).

1.29 TQL EXPRESSIONS <expr>

T Q L O P E R A T O R S

Symbol	Description	Alternate
+	Addition	
-	Subtraction	
*	Multiplication	
/	Division	
%	Remainder	
=	Equality	EQ
<>	Inequality	NE
>	Greater than	GT
<	Less than	LT
>=	Greater than or equal	GE
<=	Less than or equal	LE
=*	Begins with	BEGINS WITH
=!	Does not begin with	DOES NOT BEGIN WITH
=:	Contains	CONTAINS
	Does not contain	DOES NOT CONTAIN

T Q L C O N N E C T O R S

Symbol	Description	Alternate
&	Logical AND	AND
¦	Logical OR	OR
	Logical negation	NOT

--*+*--

KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX

--*+*--

KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX
KWIC INDEX

2. KWIC INDEX

KWIC INDEX

- E n d o f D o c u m e n t -

A•R

ALLINSON—ROSS CORPORATION