

OS/2 Query Manager overview and prompted interface

by S. L. Watson

Operating System/2™ (OS/2™) Query Manager provides a user interface for both novice and sophisticated database users of the OS/2 Database Services. It offers defaults and standard options for the novice user. Prompting provides access to the database without requiring extensive knowledge of Structured Query Language (SQL), yet it also allows the advanced user to completely customize screens and reports. Direct keying of SQL statements is allowed as a fast path for the knowledgeable SQL user. Functions of OS/2 Query Manager are described, including details of the user interface.

Operating System/2™ (OS/2™) Query Manager is part of OS/2 Extended Edition. It is a component of the OS/2 Database Manager¹ and provides a user interface to the OS/2 Database Services. Developed by IBM, it consists of a set of general-purpose tools designed so that a user can easily interact with the database.

With OS/2 Query Manager, tables can be created in the database. Data can be added, changed, or deleted in these tables. *Ad hoc* queries of the database can be made, and data can be retrieved on the basis of user-specified search conditions. Data can be formatted into reports for viewing or printing, and the output can be tailored to meet specific needs. Panels (display screens) can be customized for data entry and viewing, procedures can be defined for repeated operations, and a menu can be created to tie it all together. Tools can be invoked to move data in and out of the database. All of these functions are performed in OS/2 Query Manager on a display screen by navigating through windows and action bars with consistent selection and field entry techniques.

This paper describes the functions of OS/2 Query Manager and provides details on the user interface. It is assumed that the reader has general knowledge of relational database concepts. Since another paper in this issue highlights the functions of OS/2 Database Services and describes IBM relational database technology,¹ discussion of those items is not repeated here.

Functional overview

OS/2 Query Manager provides a variety of end-user tools for interacting with the OS/2 Database Services. At the most basic level, there is a command interface which provides a full-screen window for entering Structured Query Language (SQL)² statements; this function is called the SQL window. A query can be keyed in and executed from this window. The user must know SQL syntax and be able to key in table names and column names without prompting. A query can also be saved in the database as a query object.

Prompting support is provided for defining tables, views, and indexes on the database. The user is guided through a series of pop-up windows that are displayed on the screen to create SQL data definition language statements and execute them on the database. Knowledge of SQL syntax is not required to use

© Copyright 1988 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

this function. Once tables are defined, os/2 Query Manager provides a full-screen data edit capability for entering and updating data in the tables. Data rows are entered, one row at a time, into a screen formatted according to the table definition. Data rows can also be retrieved on the basis of a search condition and then updated or deleted selectively.

Prompting support is also given for the creation of SQL SELECT statements. This facility, called *prompted query*, has built-in list capabilities for finding and selecting table names and column names. The resulting query can be executed and/or saved in the database as a query object.

When prompted queries are executed, data are retrieved from the database and displayed to the user through the report feature. The form controls the format of the report. os/2 Query Manager creates a default form for each query when the user has not specified a form. The user can change the appearance of the report by modifying the form. The form can be saved in the database as a form object and used later to print or display reports.

os/2 Query Manager offers the advanced user a set of customizing tools. Screens for data entry and viewing can be created using the panels feature. Multiple tables can be used on a single panel. Computations can be included on the panel to enhance the output of the database fields. A special actions window can be defined for a panel with actions tailored to the user's terminology. Panels can also be designed for special purposes, such as an invoice or an insurance claim, and can be set up for printing in these formats.

A procedure capability is included in os/2 Query Manager which allows os/2 Query Manager commands to be stacked for execution as a single operation. It also has basic commands for controlling logic flow through the procedure execution.

A menu feature that provides further customizing permits the user to design a full-screen display for selecting functions to execute. The functions include os/2 Query Manager commands to execute queries, print reports, and execute panels, procedures, or other menus. Panels, procedures, and menus are all os/2 Query Manager objects which can be saved in the database and executed later.

os/2 Query Manager has a tools function which gives the user an interface for invoking os/2 Database

Services utilities such as database backup and table import and export. Other tool options are available for configuring os/2 Database Services and setting system defaults for os/2 Query Manager.

Systems Application Architecture compatibility

os/2 Query Manager is based on the Query Interface elements of IBM's Systems Application Architecture. os/2 Query Manager offers a set of database tools that is consistent with the Query Management Facility (QMF) product³ on VM- and MVS-based systems.

The user interface of OS/2 Query Manager is object-action-oriented.

The QMF user should feel familiar with os/2 Query Manager on the IBM Personal Computer (IBM PC) and IBM Personal System/2[®] (PS/2[®]). The effort has been made to have a rich subset of QMF functions in os/2 Query Manager. Functions that are new in os/2 Query Manager will be considered for addition to QMF. Over time, there will be a closer affinity between os/2 Query Manager and QMF—for example, the exchange of objects and a callable interface.

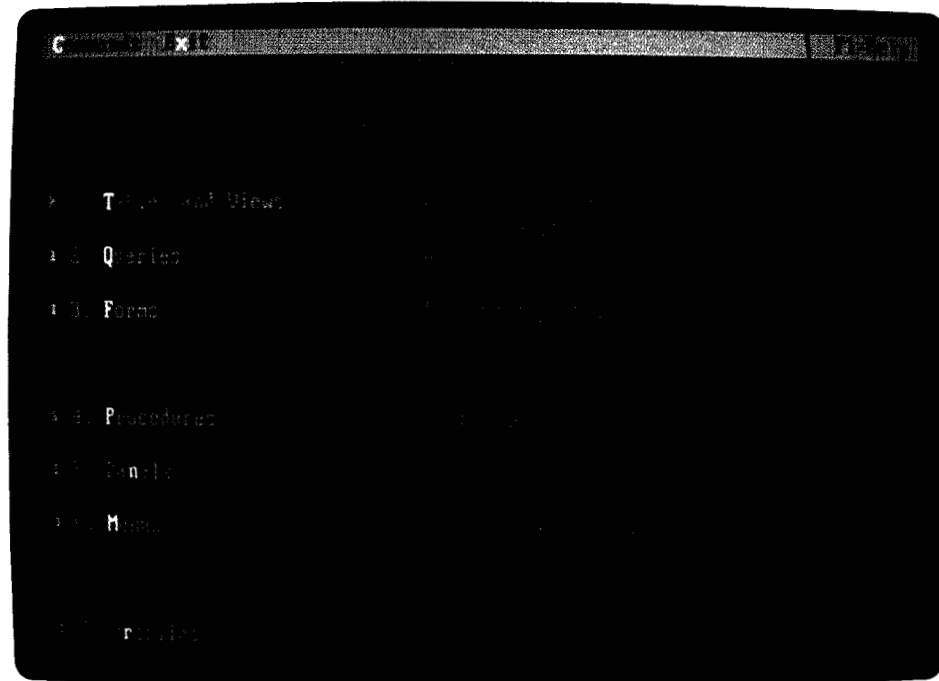
The user interface for os/2 Query Manager follows the Common User Access element of IBM's Systems Application Architecture. This interface is described in more detail in the next section.

User interaction

The user interface of os/2 Query Manager is object-action-oriented. The user selects an object and then performs an action on that object. The Main Selection Window, which lists all of the os/2 Query Manager object types, is the pivotal primary window for os/2 Query Manager. Through the Main Selection Window, shown in Figure 1, a user interacts with all of the functions of the product.

The screen is designed with an application action bar across the top, on which action categories are

Figure 1 Main Selection Window



listed. Each category has a pull-down window that displays all of the actions possible for this category.

The body of the screen is called a primary window; when the user is in a primary window, the action bar is active. When the user takes an action from the action bar, it causes the primary window to change, or initiates a dialogue. The dialogue is displayed to the user in a series of overlapping pop-up windows. When the user has responded to the prompting in the pop-up windows, the dialogue completes, and the user is returned to a primary window. Examples of these windows are given in this paper.

Keys are used in OS/2 Query Manager consistent with the Common User Access. On-line help, which is available on primary and pop-up windows, gives the user descriptive information about the use of the function in the window. Additionally, field-level help is provided for each action in the action bar pull-down windows and in each entry field in the pop-up windows. The user moves the cursor to the action or the field and presses the Help key (F1) to get the field-level help. Error messages also offer help by providing the user information to correct the error.

Prompting versus SQL

OS/2 Query Manager offers the user two ways to enter SQL statements: prompting and the SQL window. With prompting, the user is guided through a sequence of pop-up windows to fill in the information needed for the SQL statement. Then OS/2 Query Manager builds the SQL statement and executes it in the database. The prompting gives the user a way to access the database without knowing the specific SQL syntax. In the SQL window, the user types in the SQL statement directly.

Prompting is available for these SQL statements:

- CREATE/ALTER/DROP TABLE
- CREATE/DROP VIEW
- CREATE/DROP INDEX
- SELECT
- INSERT
- UPDATE
- DELETE

Some complex options such as subqueries in the SELECT statement are not supported in the prompting

interface (refer to the OS/2 Extended Edition User's Guide⁴ for a discussion of these limitations). In addition, prompting is provided for use of the following OS/2 Database Services utility commands:

- CREATE/ALTER/DROP DATABASE
- BACKUP/RESTORE DATABASE
- IMPORT/EXPORT TABLE
- REORGANIZE TABLE
- RUN STATISTICS
- CATALOG/UNCATALOG DATABASE

The remainder of this paper focuses on the user interface of the OS/2 Query Manager and its use of prompting to guide the user through database actions. Prompting for a table definition and a query are described in detail, and formatting a report is illustrated. Data entry, customizing, and utilities are not described further. More information on these tasks can be found in the referenced User's Guide.⁴

Prompted table definition

Definition of table structures in the database is required before data can be entered. First, each column in a table is given a data type. The user is prompted for a column name and a column length if the data type has a variable length. Column attributes are specified to identify whether the column is required or contains text data. As each column is defined, it is echoed to the user in the underlying window. The following sequence of windows illustrates the prompting for creating a table to be named STAFF.

The user chooses Tables and Views from the Main Selection Window. Each item selected from the Main Selection Window displays a primary window with names of the objects of the selected type listed. The names of the tables and views defined in the database are retrieved and displayed in alphabetical order in a scrollable list in a primary window. Beside each name is an optional comment that was specified by the user when the object was saved in the database. The comment may be up to 55 characters in length and may be used to give a brief description of the object.

To select an object (table or view), the user places the cursor on the name of the desired object and presses the Enter key. To create a new object, the user selects the name ---NEW---, which always appears as the first name in the list. Then the user selects Actions from the action bar at the top of the screen. The Actions pull-down window is displayed, listing the actions that can be taken by the user.

In the example, the user selects ---NEW--- in the Table and View Names primary window and takes the action to Open the table definition. The Table primary window is displayed showing a blank table definition with no columns specified.

Now the user selects the Actions pull-down window from the action bar. The Actions window is shown pulled down over the underlying primary window in Figure 2.

Actions that are valid for the object selected (---NEW--- table) are indicated by a square symbol (■) to the left of the action. Since the table is new and has no columns, Change a column, Delete a column, Convert to SQL, and Show SQL are not valid actions and do not have a square indicating that they are selectable actions. The first valid action (Add a column) is selected as a default. When an action is selected, the square is replaced with a right-pointing triangular (►) character.

The user may select an action in one of five ways:

1. Move the cursor over the action in the pull-down window and press the Enter key.
2. Use the mnemonic, which is shown highlighted on monochrome displays and as a different color on color displays.
3. Use the function key, if one is assigned. The user may press the function key without moving to the action bar and entering the Actions pull-down window.
4. Use the number of the action in the Actions pull-down window.
5. Move a mouse cursor to the action in the Actions pull-down window and click the mouse.

The user selects the action to add a column. The three dots after the action text indicate to the user that additional prompting is required to complete the action. The Actions pull-down window disappears and a pop-up window is displayed prompting the user to select a data type for the column. Figure 3 shows the Column Data Type pop-up window with Character selected.

The data types correspond to SQL data types. In the prompted interface, some of the data types are more descriptive than their SQL counterparts to guide the user in making the correct choice. For example, Character (fixed length) is used instead of CHAR, and Small integer is used instead of SMALLINT. The user moves the cursor up and down over the list of items.

Figure 2 Actions pull-down window for the Table primary window

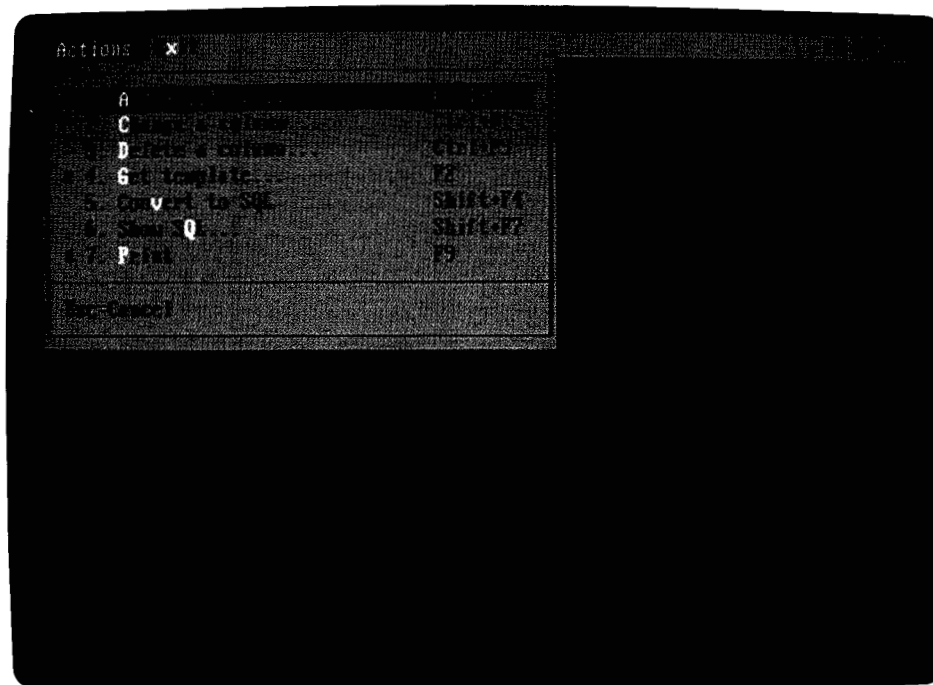
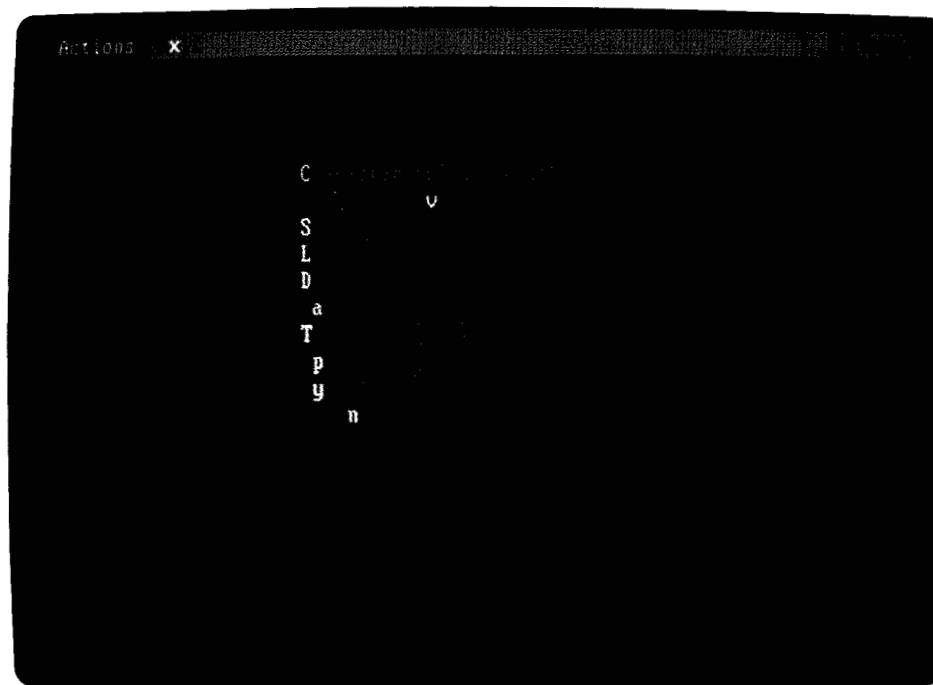


Figure 3 Table primary window with Column Data Type pop-up window



When the cursor is on the data type desired for the column, the user presses the Enter key. In the example table, the first column is a character column.

When the data type is selected, a pop-up window that prompts the user to complete the column definition is overlaid on the screen. The column name, which can be up to 18 characters in length, is keyed into the name field. Since the Character data type

Prompted query is a function that greatly enhances the usability of a relational database.

requires that a length be specified, a desired length is keyed into the length field. The user also specifies whether data are required, indicating that this column must contain data for each row in the table. This corresponds to the SQL NOT NULL attribute. Character data types have one final field to indicate whether the data are text. The text attribute says that the data in this column consist of valid ASCII characters. The user is not prompted for the text attribute for numeric data types.

After the fields that define a column are filled in, the user presses the Enter key. The data from the pop-up window are displayed in the underlying window so that the user can see the result of the Add column action. Figure 4 shows the STAFF table definition after all of the columns have been added.

In the Table primary window, the user can view the table definition after each column is specified. OS/2 Query Manager also provides a way for the user to see the SQL statement that is being built as a result. By selecting the Show SQL action from the Actions pull-down window, the user may interrupt the prompting sequence and see the current SQL statement. This is shown in a pop-up window across the bottom of the screen. The SQL statement is displayed in an easy-to-read format. Scrolling is available if it is necessary to see the entire SQL statement. Figure 5 shows the first six lines of the SQL statement for the

example table definition. Since no name has yet been given to the table, the table name is shown as an underscore line. This feature is designed for the user who wants to learn SQL but who also wants to be productive while learning.

Once all of the columns are defined, the table definition is ready to be created in the database. OS/2 Query Manager prompts the user for a table name (in this case STAFF) and a comment. The table is defined in the database by executing the SQL CREATE TABLE statement. More columns may later be added to the table, or the table definition may be recalled and used as a template to create a similar table. When the user exits from the Table primary window, the primary window labeled Table and View Names is redisplayed, with the newly created table name appearing in the names list.

Prompted query

Prompted query is a function in OS/2 Query Manager that greatly enhances the usability of a relational database for first-time users and for novice SQL users. The power of a relational database lies in the ease with which data may be retrieved from tables in the database under various conditions. Thus, the select query becomes the most important SQL statement.

To illustrate the power and use of the prompted query function, an example problem is given. From the STAFF table created earlier, the user needs a list of all of the employees whose yearly earnings (salary plus commission) are greater than \$10,000. The list should be broken out by department, with the employees ordered by earnings and the highest earnings listed first.

As the first step, the user selects the table for the query. This is done by entering the name of the table (STAFF) in the Tables pop-up window, as shown in Figure 6.

A List feature (F4=List) is available throughout the prompting windows of the product. It enables the user to select from a list of valid choices, and also helps the user to remember the name of a table, column, or other object. Since only valid choices can be selected, the List option also reduces the chance of error.

The user's Authorization identification (ID), which is set when the user starts the OS/2 Query Manager program, is appended in front of the table name. All

Figure 4 Table primary window with columns for STAFF table defined

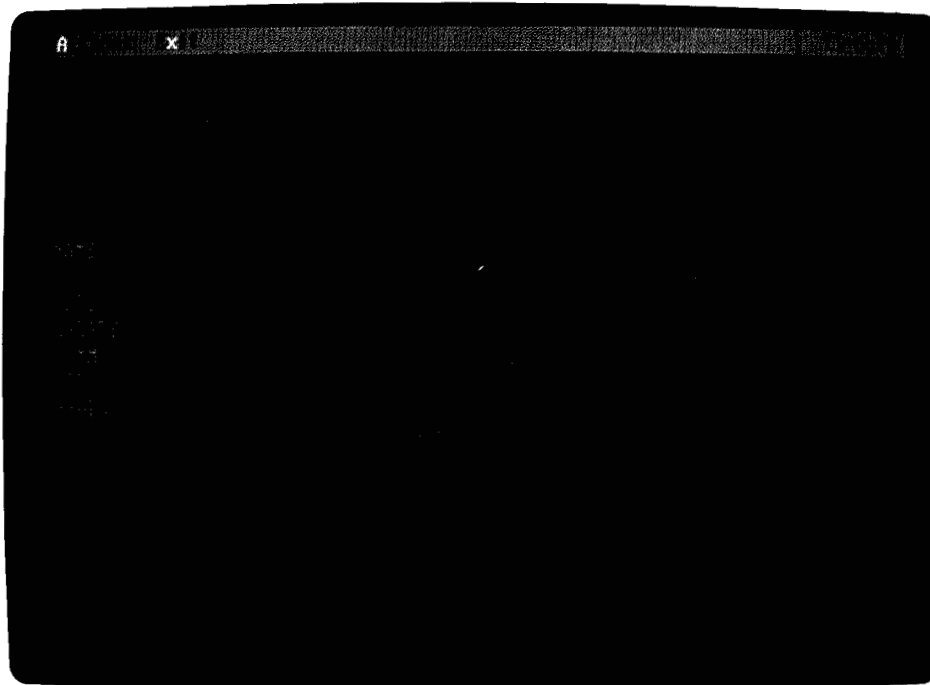


Figure 5 Table primary window with Show SQL pop-up window

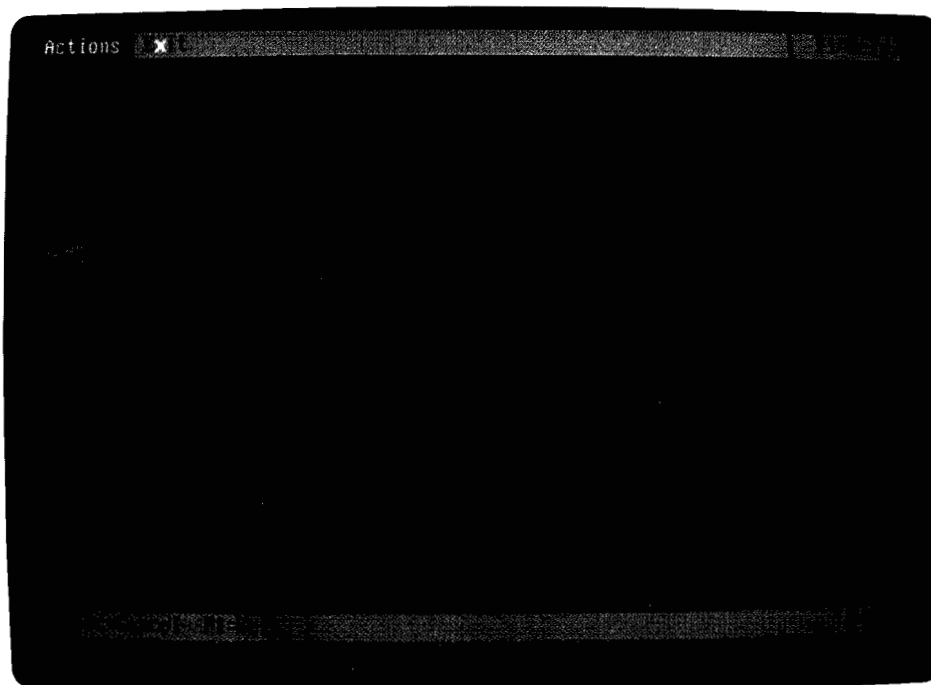
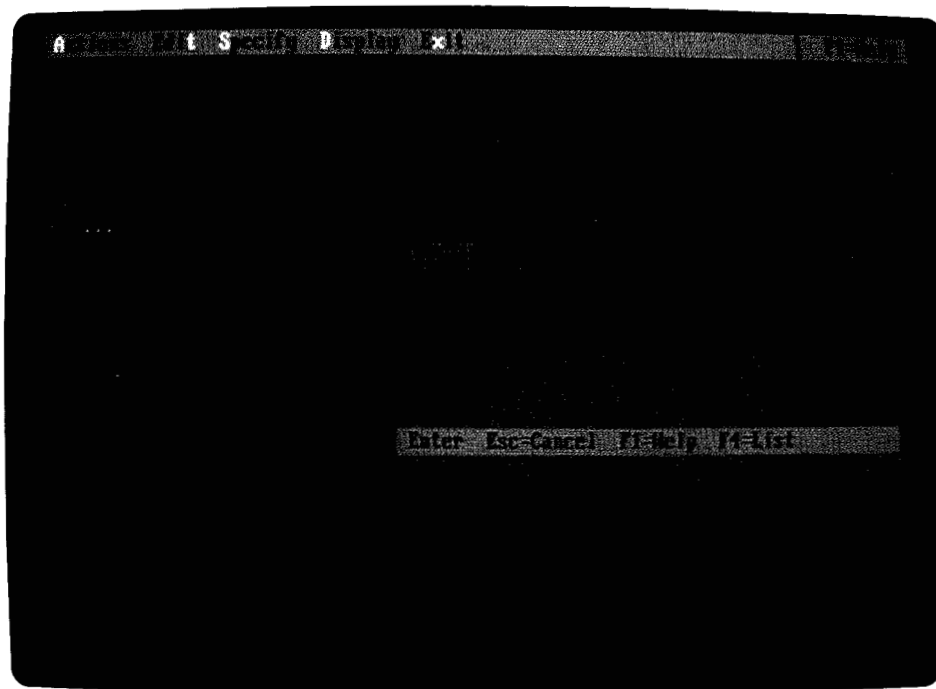


Figure 6 Prompted Query primary window with Tables pop-up window



tables and objects created in the database by the user are uniquely identified by the user's Authorization ID. PUBLIC is the Authorization ID used in the example.

The Tables step and the other steps for prompting the user for the query information are given below. As each step is completed, the results are echoed in the underlying primary window. The terms for each step are designed to be more descriptive and easier for the user to follow than the corresponding SQL terms.

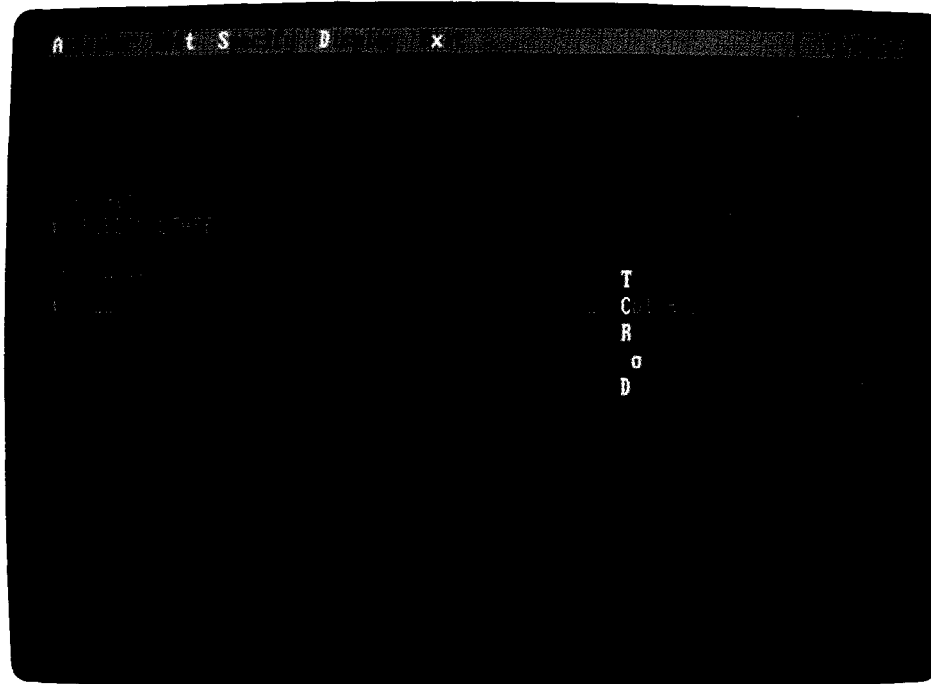
- **Tables:** Defines the table names to be used in the query. If the user specifies more than one table name, OS/2 Query Manager automatically prompts for the join condition(s). The contents of Tables are translated into an SQL FROM clause.
- **Columns:** Defines the columns to be retrieved for output. Only columns that exist in the tables selected may be used in Columns. The contents of this step are translated into an SQL SELECT clause.
- **Row Conditions:** Defines the conditions for selecting the output rows. The contents of Row Conditions are translated into an SQL WHERE clause.

- **Sort:** Defines the sort order in which the output rows are to appear. The contents of Sort are translated into an SQL ORDER BY clause.
- **Duplicate Rows:** Defines whether duplicate rows are discarded from the output rows. Single Copy of Each Row option is translated into the DISTINCT keyword in the SQL SELECT clause.
- **Join On:** Defines the table join criteria when more than one table is selected. The contents of Join On are translated into an SQL WHERE clause.

The user is provided with assisted navigation through the steps of the prompted query creation. As each step is completed, the user is prompted by the Specify pop-up window, with the next option in the query selected. All of the steps in prompted query are listed in the Specify pop-up window except the Join On step. Prompting for the join condition is done when more than one table is specified in the Tables pop-up window.

The assisted navigation is presented to the user but does not force him or her through a structured path. The user can follow a different sequence or skip steps in the sequence without penalty. Figure 7 shows the Specify pop-up window with Columns selected. To

Figure 7 Prompted Query primary window with Specify pop-up window



specify Columns (which are needed for the example query), the user simply presses the Enter key.

The columns for the query can be selected from the Columns pop-up window. The user chooses the required columns from a list of all columns in the selected table(s). In the example shown in Figure 8, there is one table with seven columns. The columns are presented in a list in a scrollable box within the Columns pop-up window. The scrolling indicators in the upper right corner of the box are present if there are more column names than appear in the box. Four column names are shown as having been selected in Figure 8.

In the example problem, an expression is also used as a column. An expression is an arithmetic calculation using the values in the columns. The user rerequests the Columns pop-up window, this time selecting the Expression option. An Expression pop-up window is displayed overlaying the bottom of the screen. The pop-up windows overlay one another through the prompting sequence; their placement is designed to enhance the user's flow through the sequence. The user keys an expression (SALARY+COMM) into the Expression pop-up window and

presses the Enter key. Once entered, the expression is checked for validity.

At this point, assisted navigation guides the user to specify the row conditions, i.e., which rows are to be selected from the table. In the example table, each row contains data on one employee. The requirement is to select only those employees whose yearly earnings (salary plus commission) are greater than \$10,000.

Each row condition is entered in three parts: left operand, operator, and right operand. The Row Conditions pop-up window looks similar to the Columns pop-up window. The column names and expressions that can be used as operands are listed in a scrollable box. For the left operand, the user makes a selection (SALARY+COMM) from the list.

The next step in creating the row condition is specifying the comparison operator, which is composed of a verb and a comparison. A pop-up window overlays the Row conditions pop-up window and displays a list of comparison operators (see Figure 9). From the list, the user selects a verb (Is or Is not) and a comparison. Since the row condition given

Figure 8 Prompted Query primary window with Columns pop-up window

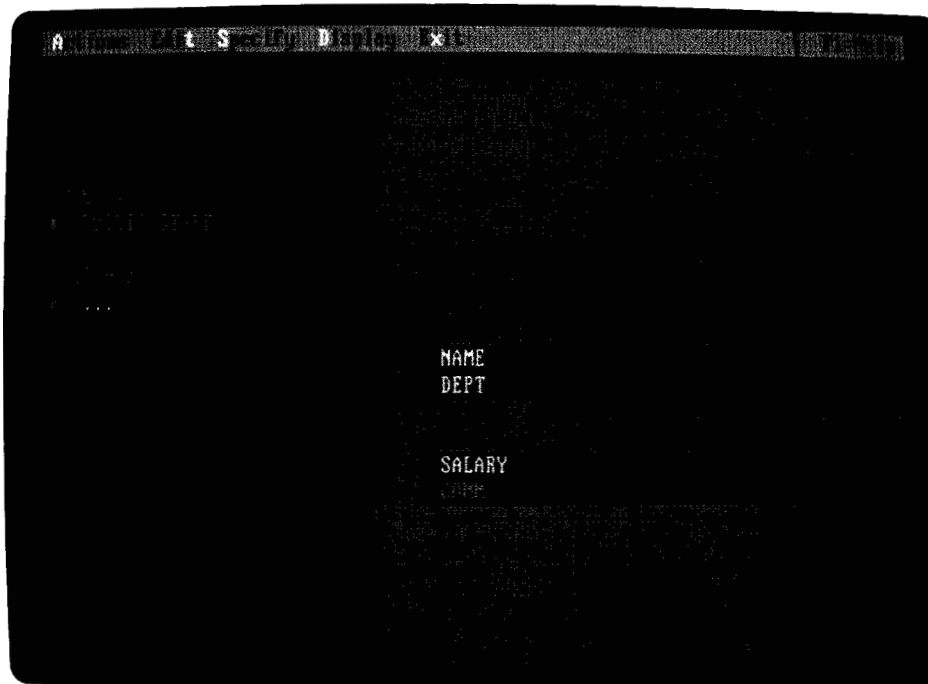
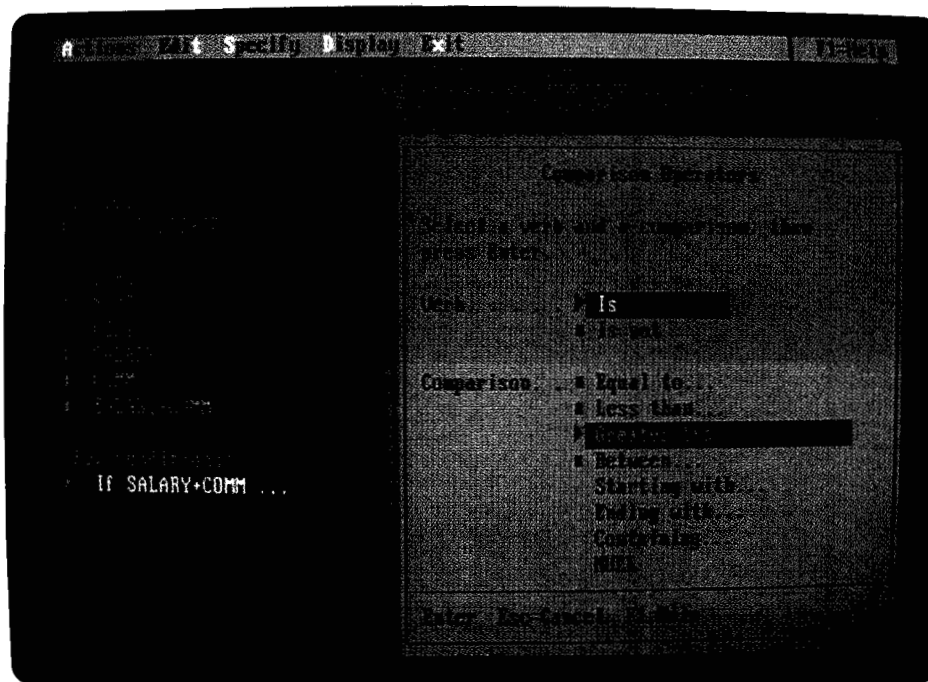


Figure 9 Prompted Query primary window with Comparison Operators pop-up window



depends on the left operand, only comparisons that are valid for the left side of the condition will be selectable (indicated by a square). In the example, the left operand (SALARY+COMM) is numeric; therefore, only those comparisons which operate with numeric fields are selectable. The comparison operator (Is greater than) is chosen in Figure 9.

A third pop-up window appears which prompts for the right operand, the final part of the row condition. The List key is available for the user to list columns from the selected table that would complete a valid row condition. If an expression is entered, the expression is verified to the extent that its results will be consistent with the left operand; otherwise the user could be creating a query with an invalid row condition. By being presented with only valid choices and having any expressions verified, the user is aided in creating a valid query statement. In the example, a value (10000) is entered for the right operand.

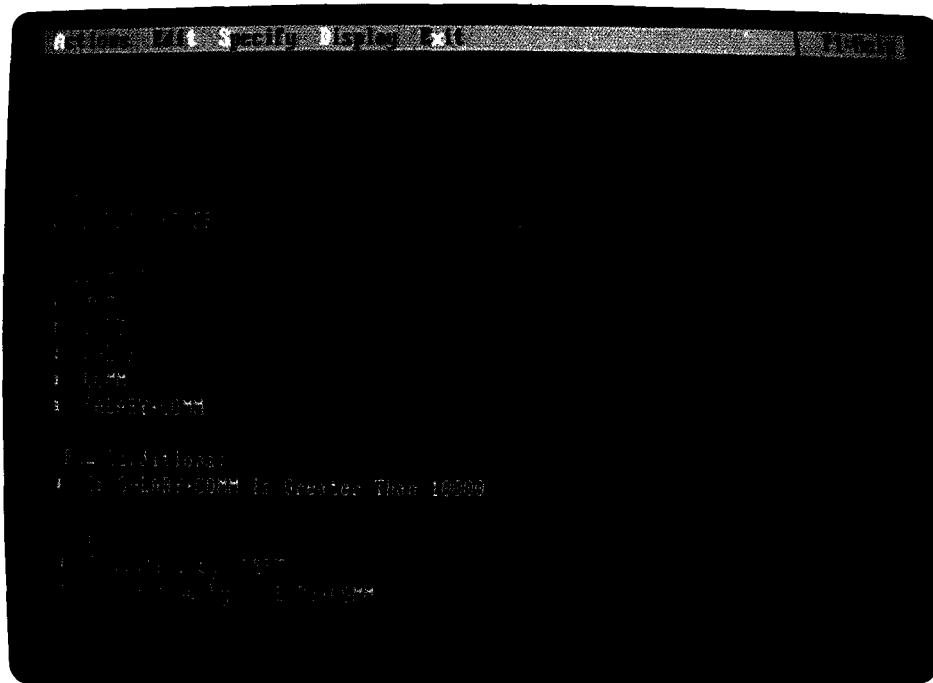
To complete the query, the user requests the sort option, which allows the user to specify the order in which the rows appear in the report. The Sort pop-up window displays the column names specified for

this query in a selection list. The user picks an order sequence (Ascending) and a column name (DEPT) from the list and then presses the Enter key. A second sort key is needed in the example. The user rerequests the sort option, this time specifying Descending by SALARY+COMM. The order of precedence for the sort keys is determined by the sequence in which the sort options are specified.

The example query is now complete. Figure 10 shows the final version of the example query. All the user's selections are displayed in the underlying window. If the user decides to modify the query, editing may be performed on the underlying window. Selectable fields are indicated with squares. The cursor is moved to a field and selected by pressing the Enter key. Editing options to insert and delete are available in the Edit pull-down window presented on the action bar.

During the prompting sequence, the user may view the query in the SQL format by using the Show SQL action in the Actions pull-down window. This action provides the same function as described earlier for the prompted table definition. A pop-up window is displayed in the bottom half of the screen and shows

Figure 10 Prompted Query primary window with completed query



the query as an SQL SELECT statement. In this way, the user can relate the prompted query actions to the resulting SQL statement. Translations and adjustments made to conform to SQL structure and syntax are apparent. The final example query is shown in a Show SQL pop-up window in Figure 11.

At any point in the prompting sequence, the user can convert the prompted query to an SQL statement and see the query displayed in the SQL window. This is a one-way transfer. The query in the SQL window can be modified, executed, and/or saved as a query object. Both SQL queries and prompted queries are recognized as query objects by os/2 Query Manager.

Report formatting

When the query is run, os/2 Query Manager creates a three-sided structure called a triangle. The query represents one side of the triangle. When the data are retrieved from the database, the column attributes are collected into an object called a form, which becomes the second side of the triangle. The data are formatted according to the form into a report, which is the third side of the triangle. The user can view any side of the triangle: the query, the form, or the report.

Figure 12 shows the first page of the report generated by running the example query. A default form created by os/2 Query Manager is used to format the report.

Once the triangle exists, the user may move freely from one side of the triangle to another side. The form may be revised to change the appearance of the data in the report. When the report is redisplayed, it uses the updated form. The data for the report remain intact and can be reformatted by changing the form. The user may view the query and modify it. When the query is modified, the data become invalid. The query may be rerun and data are again retrieved from the database. When the query is rerun, the form may no longer match the data selected in the query. For instance, the revised query may contain more columns of data than the form. When the form does not match the data in the query, the user may discard the form and replace it with another previously saved form or let the os/2 Query Manager create a default form to match the revised query.

The form may be named and saved as an object in the database. It allows the user to change column headings on the report, omit columns from the report, and indicate which columns cause breaks in

Figure 11 Prompted Query primary window with Show SQL pop-up window

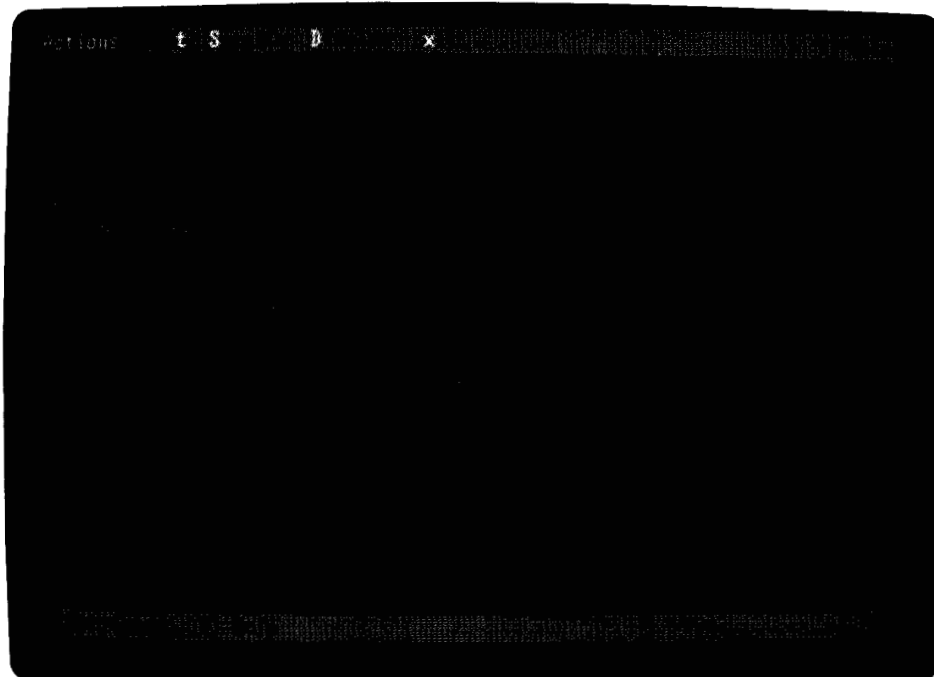
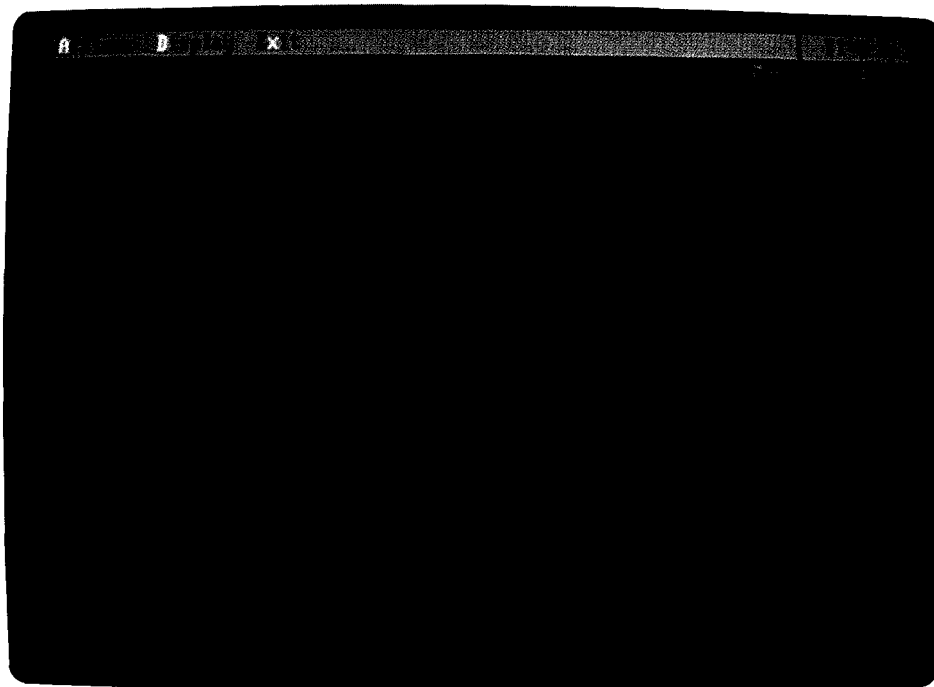


Figure 12 Default report



the report. The form primary window defines the column heading text, the width for each column, the spacing between columns, the edit format for each column, and the order position of each column across the report. Four pop-up window sequences are available to specify report headings and footings, break text, final summary text, and various options for spacing and separators.

The example report shown in Figure 12 may be modified by making changes to the form, as shown in Figure 13.

The Column heading is a scrollable field indicated by the "greater-than" symbol (>) on the right end of the field. As the text is entered, scrolling occurs to the right until the right boundary is reached. Then the > is changed to a right bracket (]) indicating the end of the field. Column heading text may be split over multiple lines by imbedding an underscore in the text for each line split.

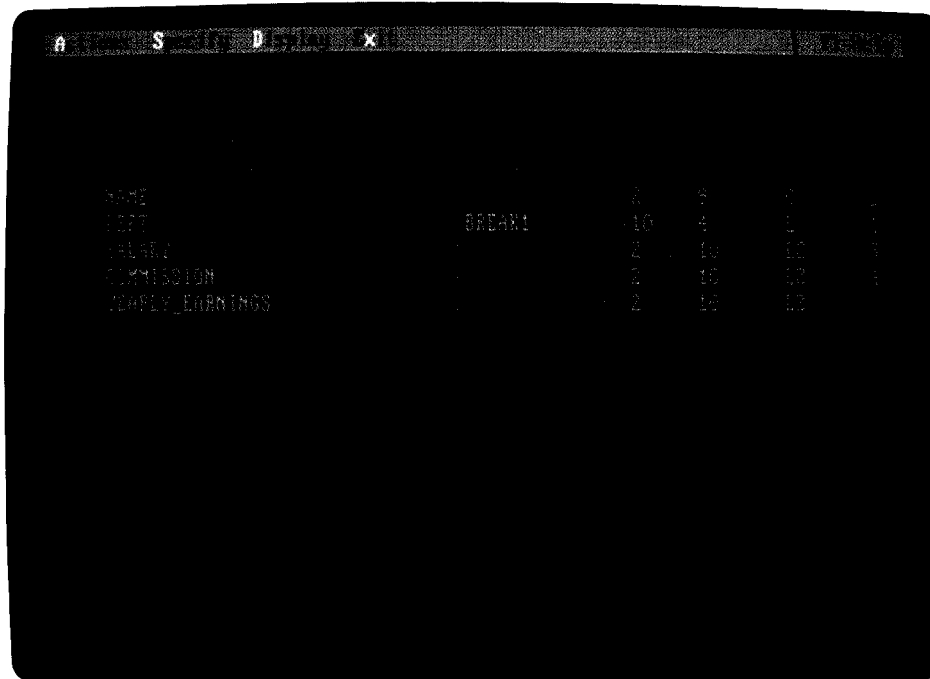
The user makes changes in the form window by keying into the appropriate fields. The Column heading for COMM is expanded to COMMISSION. The col-

umn heading for EXPRESSION 5 is revised to be YEARLY EARNINGS split across two lines. DEPT is indicated as a break column in the Usage field. The indent parameter for DEPT is increased to 10. The width of the YEARLY EARNINGS column is decreased to 10. Finally the columns are reordered to move the DEPT column to the leftmost position by changing the number sequence in the Seq field.

Additional format controls are specified in the form pop-up windows by selecting one of the actions in the Specify pull-down window on the action bar. One of these actions is for specifying page headings. Up to five text lines for a heading are allowed, and each line can be up to 55 characters long. In the example, two lines of heading text are entered. The parameter "&date" is entered as the first heading line with right alignment. The &date parameter is a system variable. When the report is displayed, the system value for date is inserted in the first line of the heading. "Summary of Yearly Earnings" is the second heading line to be centered.

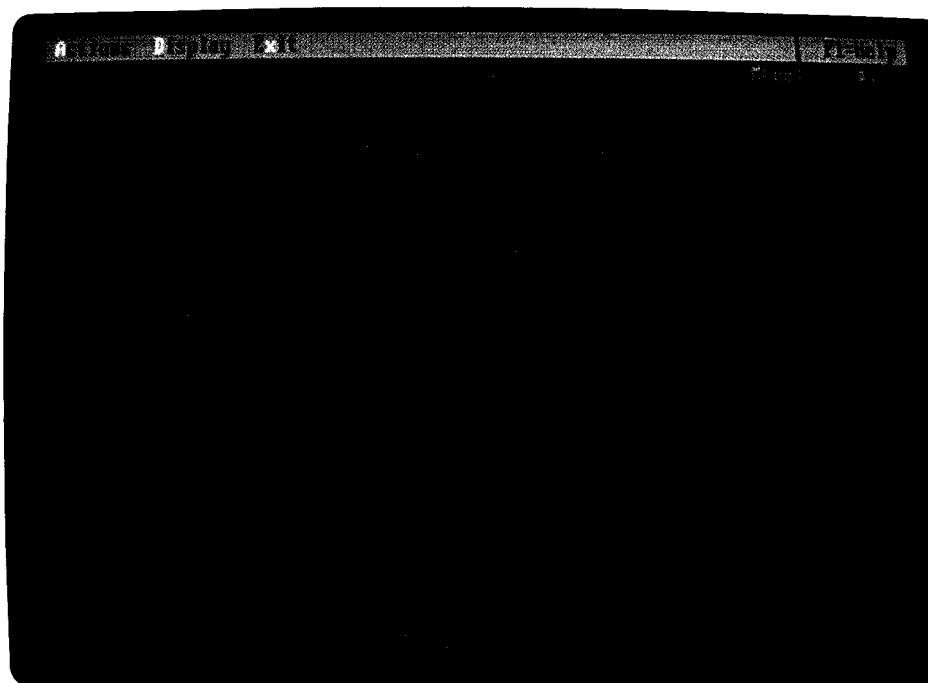
The data for the report are still available in the triangle. The user can see the reformatted report by

Figure 13 Modified Form primary window

A screenshot of a primary window titled 'ASD' with a close button 'X'. The window contains a table with 5 columns. The first column lists variables: NAME, LEFT, SALARY, COMMISSION, and YEARLY_EARNINGS. The second column lists values: 10, 2, 2, 2, and 2. The third column lists values: 10, 10, 10, 10, and 10. The fourth column lists values: 10, 10, 10, 10, and 10. The fifth column lists values: 10, 10, 10, 10, and 10. The table is as follows:

NAME				
LEFT	10	10	10	10
SALARY	2	10	10	10
COMMISSION	2	10	10	10
YEARLY_EARNINGS	2	10	10	10

Figure 14 Reformatted report

A screenshot of a reformatted report window titled 'ASD' with a close button 'X'. The window contains a table with 5 columns. The first column lists variables: NAME, LEFT, SALARY, COMMISSION, and YEARLY_EARNINGS. The second column lists values: 10, 2, 2, 2, and 2. The third column lists values: 10, 10, 10, 10, and 10. The fourth column lists values: 10, 10, 10, 10, and 10. The fifth column lists values: 10, 10, 10, 10, and 10. The table is as follows:

NAME				
LEFT	10	10	10	10
SALARY	2	10	10	10
COMMISSION	2	10	10	10
YEARLY_EARNINGS	2	10	10	10

selecting Report from the Display action pull-down window. When the report is redisplayed, it is formatted with the updated form options, as shown in Figure 14. Outlining (i.e., displaying the column values only when the value changes) is used for the break column (DEPT).

Summary

OS/2 Query Manager provides a set of end-user tools for interacting with the OS/2 Database Services. The user interface is designed to be easy to use and to meet the needs of both the beginner and the expert database user. SQL syntax and semantics are hidden from the user through the prompting interface. Use of the SQL window is also available as a fast path for the knowledgeable SQL user.

Operating System/2 and OS/2 are trademarks, and Personal System/2 and PS/2 are registered trademarks, of International Business Machines Corporation.

Cited references

1. P. Y. Chang and W. W. Myre, "OS/2 EE Database Manager overview and technical highlights," *IBM Systems Journal* 27, No. 2, 105-118 (1988, this issue).
2. *IBM Database 2, Introduction to SQL*, GC26-4082, IBM Corporation; available through IBM branch offices.
3. J. J. Sordi, "The Query Management Facility," *IBM Systems Journal* 23, No. 2, 126-150 (1984).
4. *Operating System/2 Extended Edition User's Guide*, 90X7939, IBM Corporation; available through IBM branch offices and authorized dealers.

Susan L. Watson *IBM Entry Systems Division, 11400 Burnet Road, Austin, Texas 78758.* Ms. Watson is a development programmer in the Data Systems Products Department at the Austin laboratory. She joined IBM in 1968 in the Federal Systems Division at Houston, Texas, where she worked on NASA contracts for the space program. She was involved in the development of the telemetry subsystem for the Apollo and the Skylab programs. Ms. Watson was a programming manager for development of the Ground Based Systems Telemetry and Near Realtime Telemetry subsystems in the Space Shuttle program. In 1984, she transferred to the Austin laboratory and became the development manager for the first release of OS/2 Query Manager, continuing in this position today. Ms. Watson received a B.A. in mathematics from the University of Texas in 1968.

Reprint Order No. G321-5314.