NSQ - Database Management

This section covers the following topics:

- SYSSQL Utility

- Natural System Commands for SQL/DS

SYSSQL Utility

The Natural interactive catalog utility SYSSQL allows you to do SQL/DS database management without leaving your development environment.

With SYSSQL you can maintain SQL/DS tables and other SQL/DS objects.

The SYSSQL utility incorporates an SQL generator that automatically generates from your input the SQL code required to maintain the desired SQL/DS object. You can display, modify, save and retrieve the generated SQL code.

The DDL/DCL definitions are stored in the library SYSSQL on the FDIC system file.

The SYSSQL utility offers two modes of operation: Fixed Mode and Free Mode. To switch between the two modes, you press PF4.

- Fixed Mode

- Free Mode

Fixed Mode

In fixed mode, input screens with syntax graphs help you to specify correct SQL code. You simply enter the required data on input screens, and the data are automatically checked to ensure that they comply with the SQL syntax of SQL/DS. Then, SQL members are generated from the entered data. The members can be executed directly by pressing PF5. But you can also switch to free mode, where the generated SQL code can be modified.

For each field where a window can be invoked, you can specify an "S". When you press ENTER, the window appears and you can select or enter the necessary information. If such a selection is required, an "S" is already preset when the corresponding screen is invoked.

When you press ENTER again, the window closes and if data have been entered, the field is marked with "X" instead of "S". If not, the field is left blank or marked with "S" again.

This continues each time you press ENTER until no "S" remains. To redisplay a window where data have been entered, you change its "X" mark back to "S".

If another letter or character is used, an appropriate error message appears on the screen. The wrong character is automatically replaced by an "S" and if you press ENTER again, the corresponding window appears.

In fields where keywords are to be entered, you have to enter one of the keywords displayed beneath the field. Default keywords are highlighted.

Creating an SQL/DS Table

The following example illustrates how to use the SYSSQL utility to create an SQL/DS table in fixed mode.

When you log on to library SYSSQL and issue the command MENU, the SYSSQL Main Menu appears:

```
  14:41:38                    **** SYSSQL Utility ****              2006-05-25
                                  - Main Menu -


      +---------- Maintenance ---------+   +--------- Authorizations -------+
      !   x CREATE                     !   !   _ GRANT                      !
      !   _ ACQUIRE DBSPACE            !   !   _ REVOKE                     !
      !   _ ALTER                      !   !   _ LOCK TABLE                 !
      !   _ DROP                       !   !   _ CONNECT                    !
      !   _ UPDATE STATISTICS          !   !                               !
      +-------------------------------+    +-------------------------------+
                    +-------- Descriptions ----------+
                    !   _ EXPLAIN                     !
                    !   _ COMMENT ON                  !
                    +-------------------------------+


      +-------------------------- Comments ----------------------------+
      !  Enter ? for HELP or press PF1                                 !
      !  Enter . to QUIT or press PF12                                 !
      !  Press PF4 to enter Free-Mode                                  !
      +---------------------------------------------------------------+

   Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help              Free                                              Exit
   SYSSQL4776 Please mark your choice.
```

When you select the CREATE function, a window is invoked which shows you a list of all available objects, and you are prompted for the type of object to be created, in this case a table:

```
 14:41:39                    **** SYSSQL Utility ****              2006-05-25
                                - Main Menu -


       +--------- M +------------------+   +--------- Authorizations -------+
       !  x CREATE  !   _ INDEX       !   !  _ GRANT                       !
       !  _ ACQUIRE !   _ SYNONYM     !   !  _ REVOKE                      !
       !  _ ALTER   !   x TABLE       !   !  _ LOCK TABLE                  !
       !  _ DROP    !   _ VIEW        !   !  _ CONNECT                     !
       !  _ UPDATE  !                 !   !                               !
       +----------- +------------------+   +-------------------------------+
                    +-------- Descriptions ----------+
                    !  _ EXPLAIN                 !
                    !  _ COMMENT ON              !
                    +------------------------------+

       +------------------------- Comments -----------------------------+
       !  Enter ? for HELP or press PF1                                 !
       !  Enter . to QUIT or press PF12                                 !
       !  Press PF4 to enter Free-Mode                                  !
       +----------------------------------------------------------------+

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help              Free                                         Exit
 SYSSQL4776 Please mark your choice.
```

The first Create Table syntax input screen is displayed. You can enter the creator and table names on this screen, as well as the individual column names, formats and lengths, as shown below:

```
 14:44:52                   **** SYSSQL/DS Utility ****             2006-05-25
                                - Create Table -                   Page: 01

 >>---- CREATE TABLE ----- SAG_____ . PERSONNEL_____ --------------------->
                            <creator.>table-name

 >- PERS-NO_____   DECIMAL_____ ( 8_____ ) NN -- _ -- _ -- _ -( S_ - A +
 +- NAME_____   CHAR_____ ( 25___ ) NN -- _ -- _ -- _ -- __ - _ +
 +- FIRST-NAME_____   CHAR_____ ( 25___ ) NN -- _ -- _ -- _ -- __ - _ +
 +- AGE_____   DECIMAL_____ ( 2_____ ) NN -- _ -- _ -- _ -- __ - _ +
 +- SALARY_____   DECIMAL_____ ( 5,2__ ) __ -- _ -- _ -- _ -- __ - _ +
 +- FUNCTION_____   INTEGER_____ ( _____ ) __ -- _ -- _ -- _ -- __ - _ +
 +- EMPL_SINCE_____   DATE_____ ( _____ ) NN -- _ -- _ -- _ -- __ - _ +
 +- _____   _____ ( _____ ) __ -- _ -- _ -- _ -- __ - _ )
        column-name          format        length  NN    S   field CCS  PRIMARY !
                                                    NU    M   proc  ID   KEY A/D !
                                                    NP    B      +---------------+
                                                            +- PCTFREE=  __ ->
                                                                          0-99




 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next        Free  Exec Top  Bwd  Fwd  Bot       Error Menu
```

Note:
Since the specification of any special characters as part of a Natural field or DDM name does not comply with Natural naming conventions, any special characters allowed within SQL/DS should be avoided. The same applies to SQL/DS delimited identifiers, which are not supported by Natural.

In addition, various attributes can be specified for each column.

- In the NN/NU/NP field you can specify:

    ○ NN (NOT NULL) if the column may not contain null values,

    ○ NP (NOT NULL PRIMARY KEY) if the column is the primary key

    ○ NU (NOT NULL UNIQUE) if the column is a unique key

- In the S/M/B field you can specify the following for character columns:

    ○ S (FOR SBCS DATA)

    ○ B (FOR BIT DATA)

    ○ M (FOR MIXED DATA)

- You can mark the field "fieldproc" to display a window where you can specify a field procedure which has to be executed for that column.

- For character and graphic columns you can mark the CCSID to display a window where you can specify a CCSID to be used for that column.

You can also specify which columns are to be part of a primary key if the primary key is comprised of multiple columns. To do so enter an "S" or the positional number in the first column of the field PRIMARY KEY.

A primary key is a set of column values that enforce referential integrity. Only one primary key definition is allowed per table. Primary key values must be unique and must be defined as NOT NULL.

If a column is to be part of a primary key, you also have to specify whether the values from this column are to be arranged in ascending ("A") or descending order ("D"), where "A" (Asc) is the default value. In addition, you can specify the percentage of space within each index page for later insertions and updates of the primary key (the default value is 10%).

If a letter or character other than those mentioned above is used, an appropriate error message appears on the screen and the wrong character is automatically replaced by the appropriate one.

Windows like the one below may help you in making a valid selection. They are invoked by entering the help character "?" in the appropriate field on the screen:

```
 14:50:09                    **** SYSSQL Utility ****              2006-05-25
                               - Create Table -                     Page: 01

  >>--- CREATE TABLE ----- SAG_____ . PERSONNEL_____ ------------------->
                            <creator.>table-na +----------------------------+
                                               ! Please mark your choice:   !
 >-( PERS-NO_____ - DECIMAL_____ ( 8_ !  _ INTEGER               !
 >-- NAME_____ - CHAR_____ ( 25 !  _ SMALLINT             !
 >-- FIRST-NAME_____ - CHAR_____ ( 25 !  _ FLOAT(integer,integer)   !
 >-- AGE_____ - DECIMAL_____ ( 25 !  _ DECIMAL(integer,integer) !
 >-- SALARY_____ - DECIMAL_____ ( 2_ !  _ CHAR(integer)         !
 >-- FUNCTION_____ - INTEGER_____ ( 5, !  _ VARCHAR(integer)      !
 >-- EMPL-SINCE_____ - DATE_____ ( __ !  _ LONG VARCHAR          !
 >-- _____ - ?_____ ( __ !  _ GRAPHIC(integer)      !
      column-name              format      ( __ !  _ VARGRAPHIC(integer)   !
                                           ( __ !  _ LONG VARGRAPHIC       !
                                                !  _ DATE                  !
                                                !  _ TIME                  !
                                                !  _ TIMESTAMP             !
                                                ! Valid abbreviations:     !
                                                ! I,S,F,DE,C,VARC,L VARC,G, !
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7 ! VARG,L VARG,DA,TIME,TIMES !
      Help  Next         Free  Exec  Top  Bwd ! !
                                                +----------------------------+
```

Press ENTER to close the window again.

As you can see on the above screen, the beginning of the syntax specification for an SQL statement is always indicated by ">>".

In the case of complex SQL statements, more than one input screen may be required. If so, you can switch to the following screen by pressing PF2 (Next).

If you press PF2 (Next), the next Create Table input screen screen is displayed, where you can specify up to 16 foreign keys for the current table together with their corresponding parent table and up to 16 unique keys.

```
 14:52:52                     **** SYSSQL/DS Utility ****              2006-05-25
                                 - Create Table -                      Page: 01

 >-+-+-------------------------------------------------------------------+-+-)->
    ! +- , - FOREIGN KEY --- AUTO-NAME_____ --- ( --- X --- ) ---->    ! !
    !                        <constraint-name>      column-names        ! !
    !                                                                    ! !
    !  >---- REFERENCES ---->                                           ! !
    !  >--- SAG_____ . AUTOMOBILES_____ - ON DELETE -+- _ - RESTRICT -+-+ !
    !      <creator>     table-name               +- _ - CASCADE --+   !
    !                                             +- S - SET NULL -+   !
    +--------------------------------<--------------------------------+

 >-+-+-------------------------------------------------------------------+-+-)->
    ! !                                                                   !
    ! +- , - UNIQUE -------- _____ --- ( --- _ --- ) ---->    !
    !                        <constraint-name>      column-names        !
    !  >---- PCTFREE= ------ __                                         !
    !                        0-99                                       !
    +--------------------------------<--------------------------------+
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Prev  Free  Exec  Top   Bwd   Fwd   Bot         Error Menu
```

On this screen, you can specify a referential constraint to another table. To do so, enter an "S" in the first "column names" field. A list of all columns available in the current table (dependent table) is displayed, where you can select the column(s) to comprise the foreign key related to another table (parent table). You can also specify a name for the constraint. If not, the constraint name is derived from the first column of the foreign key.

A foreign key consists of one or more columns in a dependent table that together must take on a value that exists in the primary key of the related parent table.

In the REFERENCES part, you must specify the table name (with an optional creator name) of the parent table which is to be affected by the specified constraint. In addition, you must specify the action to be taken when a row in the referenced parent table is deleted. You have three options available:

- RESTRICT prevents the deletion of the parent row until all dependent rows are deleted (this is the default value).

- CASCADE deletes all dependent rows, too.

- SET NULL sets to null all columns of the foreign key in each dependent row that can contain null values.

You can also specify a unique key for that table. To do so, enter an "S" in the second Column Names field. A list of all columns available in the current table is displayed, where you can select the column(s) to comprise the key. All selected columns must have been defined with the NOT NULL attribute. If this is not the case, a window is displayed where you can set NOT NULL for this column. You can also specify a name for the constraint. If you do not, the constraint name is derived from the first column of the unique key.

You can specify up to 16 constraint blocks. In each block you can define a foreign key and a unique key. In the top right-hand corner of the screen, the index of the currently displayed referential constraint block (1) is displayed. You can page forward and backward through the contraint blocks by pressing PF7 and PF8.

When you have entered all information, you can press either PF3 (Prev) to return to the previous screen, or PF2 (Next) to go to the last screen as shown below:

```
  15:05:38                    **** SYSSQL/DS Utility ****               2006-05-25
                                 - Create Table -                        Page: 01




    >-----------+-------------------------------------------------+-------->< 
               !                                                 !
               +-------- IN -- SAG_____ . DEMO_____ ---------+
                                <owner.>dbspace-name







  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
        Help         Prev  Free  Exec                              Error Menu
```

On this screen, you can specify the dbspace where the table is to be created.

As you can see on the above screen, the end of the syntax specification for an SQL statement is always indicated by "><".

If you press PF2 (Prev) on this screen, you return to the previous screen.

When all information has been entered, you can either switch to free mode (PF4) or submit the created member directly to SQL/DS for execution (PF5). If execution is successful, you receive the message:

```
Statement(s) successful, SQLCODE = 0
```

If not, an error code is returned.

Once a table has been created, the data type of its columns cannot be changed and columns cannot be deleted. However, new columns can be added using the ALTER TABLE function as described in the following section.

Altering an SQL/DS Table

With the ALTER TABLE function you can add single columns to an existing table. You can also add, drop, activate or deactivate primary and foreign keys.

The following example illustrates how to use the SYSSQL utility to alter an SQL/DS table in fixed mode.

When you mark the ALTER function in the SYSSQL Main Menu and press ENTER, a window prompts you for the type of object to be altered - in this case a TABLE:

```
 15:07:33                    **** SYSSQL Utility ****                2006-05-25
                              - Main Menu -

      +---------- Maintenance ---------+   +--------- Authorizations -------+
      ! _ CREATE                       !   ! _ GRANT                        !
      ! _ ACQUIRE +------------------+  !   ! _ REVOKE                       !
      ! x ALTER   ! _ DBSPACE    !  !   ! _ LOCK TABLE                   !
      ! _ DROP    ! x TABLE      !  !   ! _ CONNECT                      !
      ! _ UPDATE  !              !   ! !                                !
      +----------- +------------------++   +--------------------------------+
                   +-------- Descriptions ----------+
                   ! _ EXPLAIN                 !
                   ! _ COMMENT ON              !
                   +------------------------------+

      +------------------------- Comments ----------------------------+
      ! Enter ? for HELP or press PF1                                 !
      ! Enter . to QUIT or press PF12                                 !
      ! Press PF4 to enter Free-Mode                                  !
      +---------------------------------------------------------------+

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help              Free                                        Exit
 SYSSQL4776 Please mark your choice.
```

When you press ENTER again, the first Alter Table input screen is displayed:

```
 15:07:04                    **** SYSSQL/DS Utility *               2006-05-25
                              - Alter Table -

   >>--- ALTER TABLE ---------- _____ . _____ ---------------->
                                <creator.>table-name

   >-+-- ADD -- _____ _____ ( _____ ) -- _ -- _ -- _-+->
     !           column-name         format      length    S  field CCS !
     !                                                      M  proc  ID  !
     !                                                      B             !
     !                                                                    !
     +--+-------+-- PRIMARY KEY --- ( --- _ --- ) ---- PCTFREE= -- __   -----+
     ! +- ADD -+               column-names                0-99        !
     !                                                                    !
     +-- DROP --+-- PRIMARY KEY --- _ -------------------------------------+
               !                                                          !
               +-- FOREIGN KEY --- _____ ---------------------+
               !                      constraint-name                     !
               +-- UNIQUE KEY  --- _____ ---------------------+
                                      constraint-name

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next        Free  Exec                            Error Menu
```

You can enter the creator and table names on this screen, as well as the name, format and length of an additional column.

In addition, you can define a primary key as described in the section Creating an SQL/DS Table. You can also drop an already existing primary key, thereby removing all referential constraints in which the current table is a parent table.

You can also drop any already existing foreign key or unique key by specifying its constraint name. If a foreign key is dropped the corresponding referential constraint is removed.

Once you have entered all necessary information, press PF2 (Next) to display the next Alter Table input screen, where you can add or drop foreign keys and unique keys.

```
 15:09:56                    **** SYSSQL/DS Utility *            2006-05-25
                              - Alter Table -



  +>>----+-------+- FOREIGN KEY --- _____ --- ( --- _ --- ) ---+->
        +- ADD -+                   constraint-name        column-names

        >---- REFERENCES ---------- _____ . _____ ----------->
                                     <creator.> table-name
        >---- ON DELETE -+- S - RESTRICT -+-+-------------------------------><
                         +- _ - CASCADE --+
                         +- _ - SET NULL -+



  +>>----+-------+- UNIQUE KEY ---- _____ --- ( --- _ --- ) ----->
        +- ADD -+                   constraint-name        column-names

        >---------- PCTFREE= ------ __ -----------------------------------------><
                            0-99

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help  Next  Prev  Free  Exec                             Error Menu
```

A foreign key or unique key is added as described in the section Creating an SQL/DS Table.

When you have entered all information you can press either PF3 (Prev) to return to the previous screen, or PF2 (Next) to go to the last screen as shown below:

```
 15:12:40                    **** SYSSQL/DS Utility ****          2006-05-25
                              - Alter Table -

  >--- ACTIVATE ---+---- _ --- ALL -----------------------------------------+-><
                   !                                                        !
                   +---- _ --- PRIMARY KEY --------------------------------+
                   !                                                        !
                   +--------- FOREIGN KEY -- _____ ----------+
                   !                              constraint-name          !
                   +--------- UNIQUE kEY --- _____ ----------+
                                                constraint-name

  >--- DEACTIVATE -+---- _ --- ALL -----------------------------------------+-><
                   !                                                        !
                   +---- _ --- PRIMARY KEY--------------------------------+
                   !                                                        !
                   +--------- FOREIGN KEY -- _____ -----------+
                   !                              constraint-name          !
                   +--------- UNIQUE KEY --- _____ -----------+
                                                constraint-name

 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help       Prev  Free  Exec                              Error Menu
```

In the ACTIVATE part you have three options available. You can activate:

- ALL, which automatically enforces all the referential constraints defined for a primary key.

- PRIMARY KEY, which automatically enforces the primary key.

- FOREIGN KEY *constraint-name*, which automatically enforces the specified referential constraint.

In the DEACTIVATE part you have three options available. You can deactivate:

- ALL, which deactivates the primary key and all active foreign keys in the table.

- PRIMARY KEY, which drops the primary key index from the table and implicitly deactivates all active dependent foreign keys.

- FOREIGN KEY *constraint-name*, which deactivates the specified referential constraint.

By specifying any of these options, the restrictions imposed by the referential constraints are suspended and the parent and dependent tables involved in a referential constraint are made unavailable to users other than the DBA and the owner of the table.

Press PF2 (Prev) to return to the previous screen.

Free Mode

When free mode is invoked from fixed mode, the data that were entered in fixed mode are shown as generated SQL code, which can be saved for later use or modification. The editor provided is an adapted version of the Natural program editor.

If you modify an SQL member in free mode, this has no effect on the fixed-mode version of the member. You can save your modified code in free mode, but when you return to fixed mode, the original data appear again. Thus, both original and modified data are available.

In free mode you can execute the member currently in the source area by pressing PF5 (as in fixed mode).

If you switch to free mode after you have created an SQL/DS table in fixed mode as described in the section Creating an SQL/DS Table, the free-mode editor displays the generated SQL code as in the following sample screen:

```
 15:15:39                     **** SYSSQL Utility ****                    2006-05-25
                               - Free Mode -              Member:

     Command:
     +-------------------------------------------------------------------------+
     ! CREATE TABLE SAG.PERSONNEL                                              !
     !  (PERS-NO              DECIMAL(8)             NOT NULL,                 !
     !   NAME                 CHAR(25)               NOT NULL,                 !
     !   FIRST-NAME           CHAR(25)               NOT NULL,                 !
     !   AGE                  DECIMAL(2)             NOT NULL,                 !
     !   SALARY               DECIMAL(5,2),                                    !
     !   FUNCTION             INTEGER,                                         !
     !   EMPL-SINCE           DATE                   NOT NULL,                 !
     !   PRIMARY KEY (PERS-NO),                                               !
     !   FOREIGN KEY  AUTO-NAME (NAME)                                         !
     !     REFERENCES SAG.AUTOMOBILES                                          !
     !      ON DELETE SET NULL                                                 !
     !   )                                                                     !
     !   IN SAG.DEMO                                                           !
     +-------------------------------------------------------------------------+


 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help                Fix   Exec  Top   Bwd   Fwd   Bot         Error Menu
```

Free-Mode Editor

The free-mode editor available is almost identical to the Natural program editor and allows you to edit the generated SQL code. All program editor line commands and the following editor commands are available:

Command	Function
ADD d*n*f	Adds *n* empty lines.
CHANGE	Scans for the value entered as *scandata* and replaces each such value found with the value entered as *replacedata*. The syntax for this command is: CHANGE '*scandata*' '*replacedata*'
CLEAR	Clears the editor source area (including the line markers "X" and "Y").
DX, DY, DX-Y	Deletes the X-marked line or the Y-marked line or the block of lines delimited by "X" and "Y".
EX, EY, EX-Y	Deletes source lines from the top of the source area to - but not including - the X-marked line, or from the source line following the Y-marked line to the bottom of the source area, or all source lines in the source area excluding the block of lines delimited by "X" and "Y".
LET	Undoes all modifications made to the current screen since the last time ENTER was pressed, including all line commands already entered but not yet executed.
POINT	Positions the line in which the line command ".N" was entered to the top of the current screen.
RESET	Deletes the current X and/or Y line markers and any marker previously set with the line command ".N".
SCAN ['*scan-value*']	Scans for the string *scan-value* in the source area.
SCAN = [+|-]	Scans forwards (+) or backwards (-) for the next occurrence of the scan value.
SHIFT [-|+ *nn*]	Shifts the block of source lines delimited by the X and Y markers to the left (-) or right (+). "*nn*" represents the number of characters the source line is to be shifted.

For further details, refer to Program Editor in the Natural Editors documentation.

In addition, the following SQL code maintenance commands are available:

Command	Function
INSERT *member-name*	Saves the code in the source area as a member. If you press PF5, the code in the source area can also be executed as in fixed mode.
SELECT *member-name*	Reads the specified member into the source area.
DELETE *member-name*	Deletes the specified member.
LIST QUERY *member-name*	Displays a list of members on the screen using asterisk notation (*). For example, "L Q A*" would display a list of all SQL code members beginning with "A".

Member names must correspond to the naming conventions for Natural objects, which means they can be up to eight characters long and must start with a letter.

You can also always refer to the SYSSQL help system, which is invoked via PF1.

Natural System Commands for SQL/DS

This section describes special Natural system commands for the use with SQL/DS. There are three Natural system commands which perform SQL/DS-specific functions:

-

  LISTSQL Command
  Lists Natural DML statements and their corresponding SQL statements.

-

  SQLERR Command
  Provides diagnostic information about an SQL/DS error

-

  LISTDBRM Command
  Displays either a list of packages for a particular Natural program or a list of Natural programs that reference a particular package.

Note:
LISTDBRM has to be issued from library SYSSQL, which means you have to LOGON to SYSSQL first and then enter the command LISTDBRM.

LISTSQL Command

```
LISTSQL [ object-name ]
```

The LISTSQL command lists the Natural statements in the source code of a programming object that are associated with a database access, and the corresponding SQL statements into which they have been translated. LISTSQL is issued from the Natural NEXT prompt.

Thus, before executing a Natural program which accesses an SQL/DS table, you can view the generated SQL code by using the command LISTSQL.

If a valid object name is specified, the object to be displayed must be stored in the library to which you are currently logged on.

If no object name is specified, LISTSQL refers to the object currently in the Natural source area.

The generated SQL statements contained in the specified object are listed one per page.

Sample LISTSQL Screen:

```
  15:20:18               * * * NATURAL Tools for SQL * * *               2006-05-25
   Member N2PIGDDM                     LISTSQL                    Library SYSSQL

   NATURAL statement  at line 3820                               Stmt   4 / 4

      FIND SYSTEM-SYSCOLUMNS WITH TNAME EQ TABLE-NAME AND
        CREATOR = ICREATOR SORTED BY COLNO
        IF NO RECORDS FOUND DO

   Generated SQL statement   Mode : dynamic  DBRM :               Line   1 / 5

      SELECT   COLNO, CNAME, COLTYPE, SYSLENGTH, NULLS, REMARKS, REMARKS,
               CLABEL, LENGTH
      FROM     SYSTEM.SYSCOLUMNS
      WHERE    TNAME = ? AND CREATOR = ?
      ORDER BY COLNO




   Command ===>                                    Queryno for EXPLAIN 1____
   Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
             Error Exit  Expl        Parms  -     +         Prev  Next  Canc
```

Within the listed results, you can go from one listed SQL statement to another by pressing PF10 (Prev) or PF11 (Next). If a single SQL statement does not fit on the screen, you can scroll backwards or forwards by pressing PF7 or PF8, respectively.

If a static DBRM has been generated, the name of this DBRM is displayed in the DBRM field of the LISTSQL screen; otherwise, the DBRM field remains empty.

If an error occurs, PF2 (Error), which executes the SQLERR command, can be used to provide information about SQL/DS errors.

With PF4 (Expl), a SQL/DS EXPLAIN command can be executed for the SQL statement currently listed. The query number (Queryno) for the EXPLAIN command is set to "1" by default, but you can overwrite this default.

With PF6 (Parms), a further screen is displayed which lists all parameters from the SQLDA for the currently displayed SQL statement:

```
 15:27:25              * * * NATURAL Tools for SQL * * *           2006-05-25
  Member N2PIGDDM                      LISTSQL                    Library SYSSQL

         Mode : dynamic   DBRM :         Contoken :

         static parms : (1st)
                        (2nd)
         SQLDA
                                  DBID : 250  FNR :   3  CMD : S2 3820 08
     Nr   Type    Length
      1.  SMALLINT    2          0F5C C0C2 0002 01F5 0000 0000 0901 0000
      2.  CHAR       18          0F5E 0012 0012 01C5 0000 0000 0D01 0000
      3.  CHAR        8          0F70 0008 0008 01C5 0000 0000 0901 0000
      4.  SMALLINT    2          0F78 C0C2 0002 01F5 0000 0000 0901 0000
      5.  CHAR        1          0F7A 0001 0001 01C5 0000 0000 0901 0000
      6.  VARCHAR   127          002A 00FE 007F 01C1 0000 0000 0901 0000
      7.  VARCHAR   127          0038 00FE 007F 01C1 0000 0000 0901 0000
      8.  CHAR       30          1079 001E 001E 01C5 0000 0000 0901 0000
      9.  CHAR        7          1097 0007 0007 01C5 0000 0000 0801 0000


 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                 Exit                          -     +                    Canc
```

In static mode, static information is also displayed, which includes the static DBRM name, the SQL/DS consistency token and some internal static parameters.

EXPLAIN Command

LISTSQL enables you to use the SQL/DS command EXPLAIN, which provides information on the SQL/DS optimizer's choice of strategy for executing SQL statements.

Natural executes the EXPLAIN command for the SQL statement that is displayed on the LISTSQL screen.

The information determined by the SQL/DS optimizer is written into your PLAN_TABLE. Natural then reads the table and displays the contents.

```
  15:33:42                  * * * NATURAL Tools for SQL * * *              2006-05-25
  Queryno 1                       EXPLAIN Result                       Row  1 / 1

                     Estimated cost :    16.3  timerons

              Qblockno                     Table
                   Planno Method Tabno creator      Tablename
                 ---  ---   ------  ---  --------  -----------------
                  1    1                1    SYSTEM    SYSCOLUMNS




              Access  Access
               type   creator      Accessname      sort_new sort_comp
               ----   --------   -----------------  -------- ---------
                I     SYSTEM     ICOL                   Y        N




  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
                    Exit          Del         -     +                      Canc
```

The Query Number is set to "1" by default, but you can overwrite this default.

SQLERR Command

The SQLERR command is used to obtain diagnostic information about an SQL/DS error.

When an SQL/DS error occurs, Natural issues an appropriate error message. When you enter the SQLERR command, the following information on the most recent SQL/DS error is displayed:

- the Natural error message number;

- the corresponding reason code (if applicable);

- the variable SQLCODE returned by SQL/DS;

- the SQL/DS error message.

The SQLERR command can be issued either from the Natural NEXT prompt or from within a Natural program (by using the FETCH statement).

Sample SQLERR Diagnostic Information Screen:

```
 *** SQLERR Diagnostic Information *** ----------------------
NATURAL SQL Interface Codes ------------------------ Return Code: 3700 Reason
Code: 0 SQL code : -204 ------------------------------ SQLCA -------------------------------------
SQLERRP (Adabas SQL Subroutine where error occurred) : ARIXOCA SQLERRD (Adabas
SQL Internal State) RDS Return Code : 100 DBSS Return Code : 0 Number of Rows
Processed : 0 Estimated Cost : 1.0 Syntax error on PREPARE or EXECUTE IMMEDIATE
: 0 Buffer Manager ERROR Code : 0 SQLWARN (Warning Flags) Data truncated Null
Values ignored(AVG,SUM,MAX,MIN) : No. of columns greater than no. of host variables
: UPDATE/DELETE without WHERE clause : SQL statement causes a performance degradation
: Adjustment to DATE/TIMESTAMP Variable made : SQL/DS Error Message : SAG.SYSTABLES
not found in system catalog
```

LISTDBRM Command

The LISTDBRM command is used to display either existing packages of Natural programs or Natural programs referencing a given package.

Since LISTDBRM has to be issued from the library SYSSQL, first LOGON to SYSSQL and then enter the command LISTDBRM. The following menu is displayed:

```
  15:35:20                * * * LISDBRM Command * * *              2006-05-25




                     Code Function
                     ---- ------------------------------
                      D   Display DBRMs of Programs
                      R   List Programs Referencing DBRM
                      ?   Help
                      .   Exit
                     ---- ------------------------------
                 Code .. _   Library .. EXAMPLE_
                             Member  .. _____
                             DBRM ..... _____



  Command ===>
  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
       Help      Exit                                                      Canc
```

The following functions are available:

Code	Description
D	This function displays programs with SQL/DS access and their corresponding package (DBRM). If no package name is shown, the corresponding program uses dynamic SQL.
R	This function lists all programs that use a given package (DBRM). If no package name is specified, all programs that use dynamic SQL are listed.

The following parameters apply:

Parameter	Description
Library	Specifies the name of a Natural library. Library names beginning with "SYS" are not permitted. This parameter must be specified.
Member	Specifies the name of the Natural program (member) to be displayed. This parameter is optional and can be used to limit the output. If a value is specified followed by an asterisk (*), all members in the specified library with names beginning with this value are listed. If this field is left blank, or if an asterisk is specified only, all members in the specified library are listed.
DBRM	Specifies a valid package name. If left blank, programs that run dynamically are referenced. This parameter applies to function code "R" only.

Sample LISTDBRM Result Screen:

```
 15:42:20                    * * * LISTDBRM Command * * *                  2006-05-25


        Library  Name     Type         DBRM       User ID  Date      Time
        -------- -------- ------------ --------    -------- -------- --------
        EXAMPLE  PROG1    Program      PACK1       SAG      2006-05-25 15:10:43
        EXAMPLE  PROG2    Program      PACK1       SAG      2006-05-25 15:10:48
        EXAMPLE  PROG3    Program      PACK2       SAG      2006-05-25 15:11:04
        EXAMPLE  PROG4    Program                  SAG      2006-05-25 08:16:07
```