English

FUJITSU Software BS2000

SESAM/SQL-Server V9.0

Utility Monitor

User Guide

Edition October 2016

Comments… Suggestions… Corrections…

The User Documentation Department would like to know your opinion on this manual. Your feedback helps us to optimize our documentation to suit your individual needs.

Feel free to send us your comments by e-mail to:
manuals@ts.fujitsu.com

Certified documentation according to DIN EN ISO 9001:2008

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2008.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Content

1 Preface

The functions and architectural features of the SESAM/SQL-Server database system meet all the demands placed on a powerful database server in today’s world. These characteristics are reflected in its name: SESAM/SQL-Server.

SESAM/SQL-Server is available in a standard edition for single-task operation and in an enterprise edition for multitask operation.

For the sake of simplicity, we shall use the name SESAM/SQL throughout this manual to refer to SESAM/SQL-Server.

The following introductory descriptions are located centrally in the “Core manual”:

- Brief product description
- Structure of the SESAM/SQL server documentation
- Demonstration database
- Readme file
- Changes since the last editions of the manuals

1.1 Objectives and target groups of this manual

This manual is intended for database administrators and system administrators, who manage and back up the databases and also administer the DBH.

The database administrator and the system administrator respectively should know all steps of the database design. The user should have a broad knowledge of the BS2000 operating system and SQL. The user should also be familiar with the transaction concept of SESAM/SQL and have a basic knowledge of the files and spaces of a SESAM/SQL database and the utility functions of SESAM/SQL.

1.2 Summary of contents

A brief description of the utility monitor is provided at the beginning of chapter “Working with the utility monitor” on page 71.

Chapter 2 contains examples that let you get to know the utility monitor and show you how to carry out typical administration tasks with it.

Chapters 3 and 4 contain basic information on the structure and handling of the utility monitor.

To obtain information quickly on specific functions, forms, continuation forms, sequences of forms, or tasks, you can refer to chapter 5. This chapter also contains tables that give you an overview of its contents and help you to find what you are looking for more easily.

1.3 Notational conventions

Because of the frequency with which the server names are used, the following abbreviations are employed to make things simpler and more straightforward:

- **BS2000 servers** for the servers with /390 architecture and the servers with x86 architecture. These servers are operated with the corresponding BS2000 operating system.
- Servers with /390 architecture (**/390 servers** for short) for the Server Unit /390 of the FUJITSU Servers of the BS2000 SE Series and the Business Servers of the S Series
- Servers with x86 architecture (**x86 servers** for short) for the Server Unit x86 of the FUJITSU Servers of the BS2000 SE Series and the Business Server of the SQ Series (x86-64 architecture)

The strings `<date>`, `<time>` and `<ver>`, e.g. in examples; indicate the current displays for date, time and version when the examples are otherwise independent of date, time and version.

The following notational conventions are used in this manual:

UPPERCASE	SQL keywords and names, e.g. of tables
bold	Used for emphasis in running text
italics	Variables in syntax definitions and running text
`Fixed-space font`	Program text in syntax definitions and examples
“`Quotation marks`”	Input text for runtime examples described in the running text
[*spec*]	May be omitted. The brackets are metacharacters and must not be entered in the statement.
{*spec1*/*spec2*}	Alternative specifications (on a single line): The braces and forward slash are metacharacters and must not be entered in a statement.
$\left\{\begin{matrix} \mathit{angabe1} \\ \mathit{angabe2} \\ \mathit{angabe3} \end{matrix}\right\}$	Alternative specifications (over several lines): Each line contains an alternative. The braces are metacharacters and must not be entered in an SQL statement.
ℹ	Indicates notes that are of particular importance.
⚠	Indicates warnings.

2 A quick introduction to the utility monitor

The examples in this chapter give you an overview of how to work with the utility monitor and carry out typical database administration tasks with it.

In the first example, you learn how to start and terminate the utility monitor (see page 13), and how to obtain help information on the input fields and input options available to you.

The subsequent examples show you how to use the utility monitor to carry out the following tasks:

- “Creating a database” on page 19
- “Loading user data” on page 44
- “Creating and restoring a tape backup” on page 55
- “Obtaining information from the information schemata” on page 64

The examples build successively on each other. They involve the sample database ORDERCUST (see the “Core manual”).

You send the forms off by pressing the DUE key.

You send input to the utility monitor by entering it in the command area or by pressing the relevant key, see section “The command area” on page 143.
When, for example, “enter F13” is specified in this chapter, either the entry of “F13” in the command area or pressing key $\boxed{\text{F13}}$, which has the same effect, is meant.

The following environment is used in the examples:

Computer	HOST1
Identifier	ID1
Magnetic tape cartridge	MTC001
ARCHIVE directory file	$ID1.ARCHIVE.DIR
HSMS archive	$ID1.HSMSARCH
DBH name	X
Configuration name	Z

The configuration file	SESCONF.SESUTI.ZX
	The configuration file contains the following entries::
	SEE-ADMIN=XXX
	SEE-AUTHID=UTIADM
	SEE-CATALOG=ORDERCUST
	SEE-SCHEMA=ORDERPROC
	NAM=X
	CNF=Z

Universal user	UTIUNIV receives all rights to the ORDERCUST database
Database administrator	UTIADM receives all rights to the ORDERPROC schema and can pass on privileges to other users
SQL user 1	UTIUSR1 receives all the table privileges for all the tables, but cannot pass them on
SQL user 2	UTIUSR2 receives the SELECT table privilege for all tables
Database	ORDERCUST
Schema	ORDERPROC ADDONS
Tables	CUSTOMERS CONTACTS ORDERS SERVICE ORDER_STAT DESCRIPTIONS

Table 1: Environment used in the examples

2.1 Starting the utility monitor

The utility monitor can be started with /START-SESAM-UTILITY-MONITOR command (see section “The sequence of commands for starting the utility monitor” on page 76, if necessary).

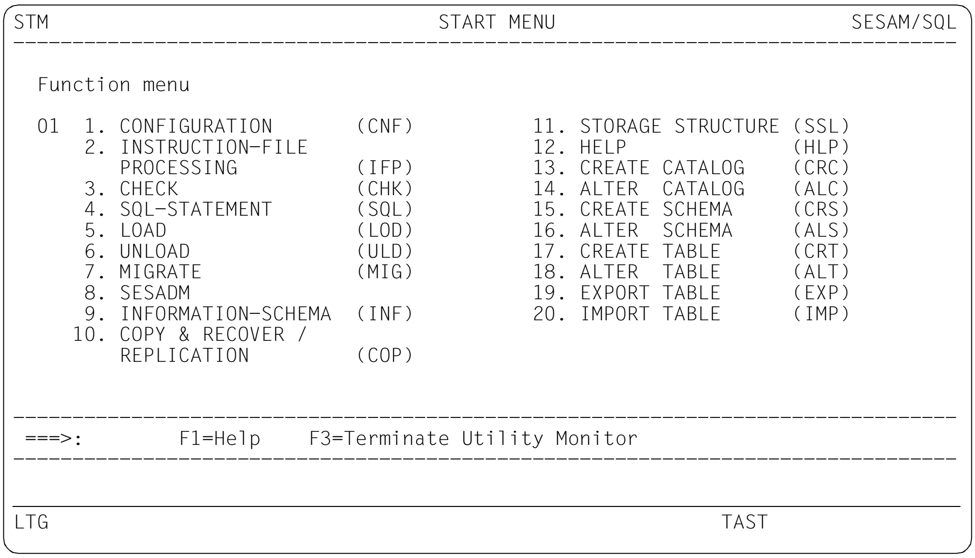
```
/ADD-FILE-LINK LINK-NAME=SESCONF,FILE-NAME=SESCONF.SESUTI.ZX, -
/              ACCESS-METHOD=SAM
/START-SESAM-UTILITY-MONITOR
```

Color settings

The input fields in the utility monitor forms are displayed in a different color from their surroundings. To optimize the way in which masks are presented you may want to adapt the colors used in your BS2000 user interface.

Start form STM - START MENU

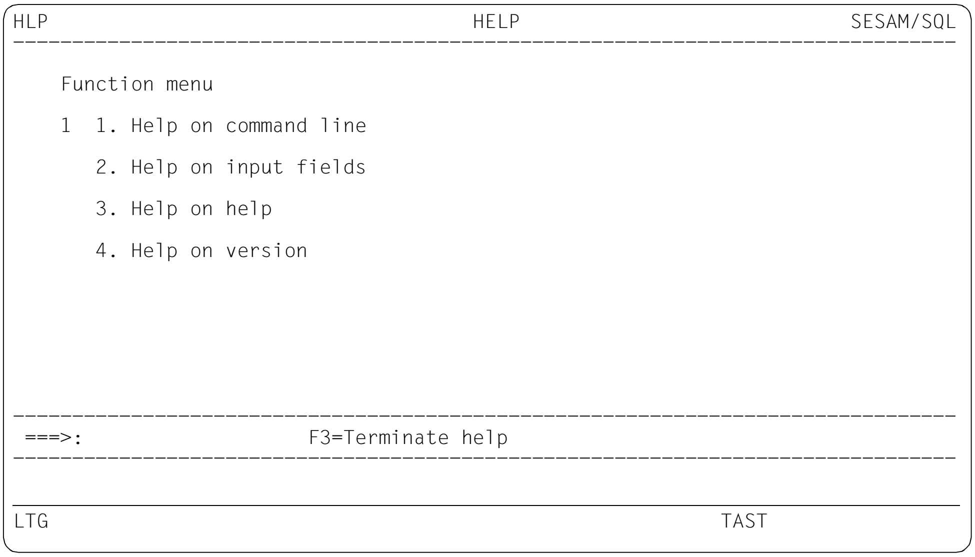
When you have started the utility monitor, the STM start form appears since the required specifications are present in the configuration file SESCONF.SESUTI.ZX.



```
STM                                 START MENU                         SESAM/SQL
--------------------------------------------------------------------------------

  Function menu

  01  1. CONFIGURATION       (CNF)          11. STORAGE STRUCTURE (SSL)
      2. INSTRUCTION-FILE                   12. HELP              (HLP)
         PROCESSING          (IFP)          13. CREATE CATALOG    (CRC)
      3. CHECK               (CHK)          14. ALTER  CATALOG    (ALC)
      4. SQL-STATEMENT       (SQL)          15. CREATE SCHEMA     (CRS)
      5. LOAD                (LOD)          16. ALTER  SCHEMA     (ALS)
      6. UNLOAD              (ULD)          17. CREATE TABLE      (CRT)
      7. MIGRATE             (MIG)          18. ALTER  TABLE      (ALT)
      8. SESADM                             19. EXPORT TABLE      (EXP)
      9. INFORMATION-SCHEMA  (INF)          20. IMPORT TABLE      (IMP)
     10. COPY & RECOVER /
         REPLICATION         (COP)


--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate Utility Monitor
--------------------------------------------------------------------------------


LTG                                                          TAST
```

2.2 Using the help system

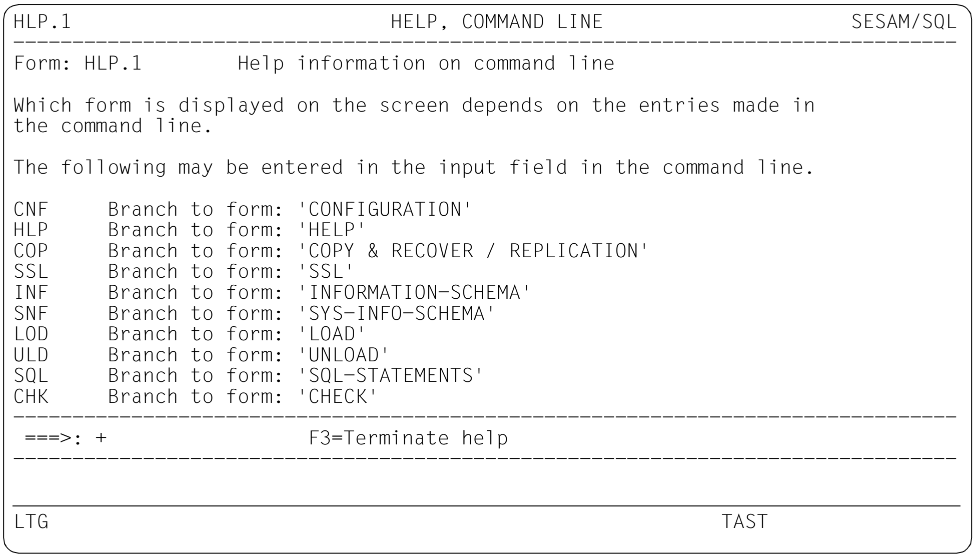
Enter the function menu “12” in the STM start form. You then branch to the HLP form. There you select the default function “1”. You then branch to the continuation form HLP.1.



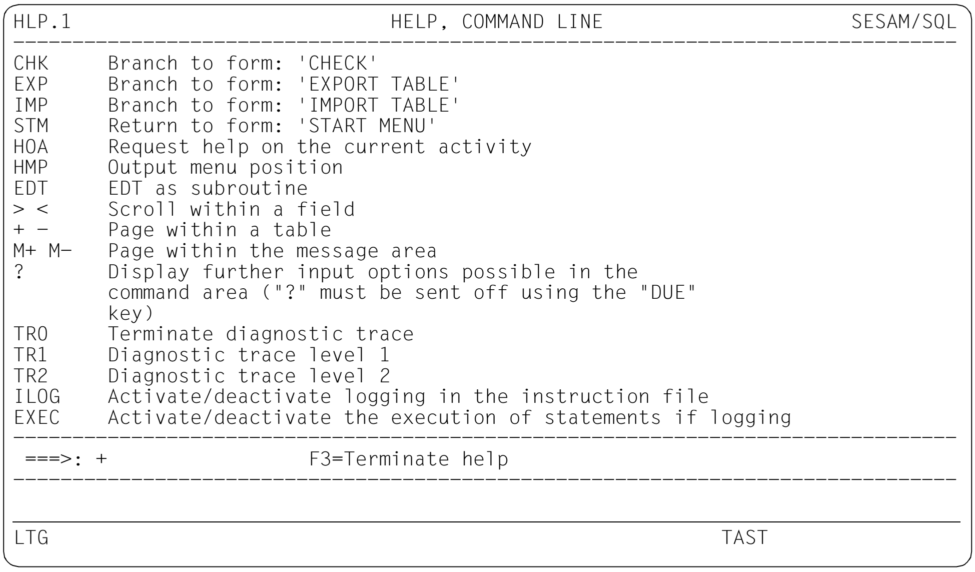
```
HLP                                     HELP                                SESAM/SQL
--------------------------------------------------------------------------------

    Function menu

    1  1. Help on command line

       2. Help on input fields

       3. Help on help

       4. Help on version




--------------------------------------------------------------------------------
 ===>:                  F3=Terminate help
--------------------------------------------------------------------------------

LTG                                                         TAST
```

Displaying a help text on the command line

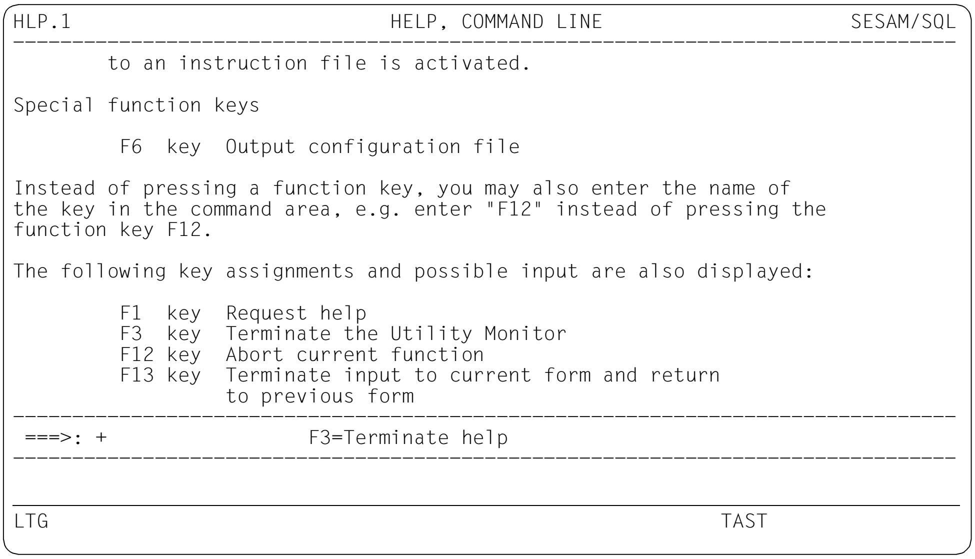
The HLP.1 continuation form contains a help text on the command line.
You enter + in the command area to display additional pages.



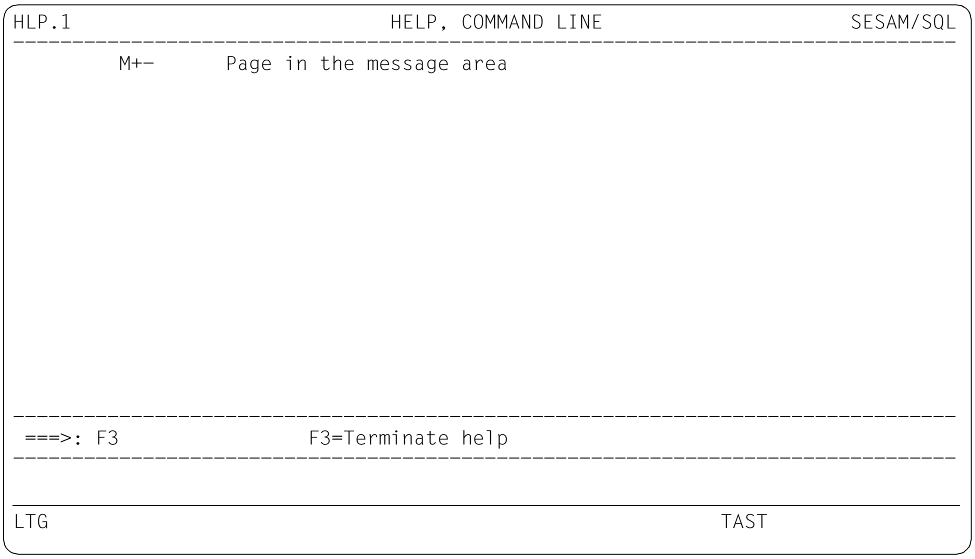
```
HLP.1                          HELP, COMMAND LINE                       SESAM/SQL
--------------------------------------------------------------------------------
Form: HLP.1        Help information on command line

Which form is displayed on the screen depends on the entries made in
the command line.

The following may be entered in the input field in the command line.

CNF     Branch to form: 'CONFIGURATION'
HLP     Branch to form: 'HELP'
COP     Branch to form: 'COPY & RECOVER / REPLICATION'
SSL     Branch to form: 'SSL'
INF     Branch to form: 'INFORMATION-SCHEMA'
SNF     Branch to form: 'SYS-INFO-SCHEMA'
LOD     Branch to form: 'LOAD'
ULD     Branch to form: 'UNLOAD'
SQL     Branch to form: 'SQL-STATEMENTS'
CHK     Branch to form: 'CHECK'
--------------------------------------------------------------------------------
 ===>: +                F3=Terminate help
--------------------------------------------------------------------------------

________________________________________________________________________________
LTG                                                         TAST
```



```
HLP.1                          HELP, COMMAND LINE                       SESAM/SQL
--------------------------------------------------------------------------------
CHK     Branch to form: 'CHECK'
EXP     Branch to form: 'EXPORT TABLE'
IMP     Branch to form: 'IMPORT TABLE'
STM     Return to form: 'START MENU'
HOA     Request help on the current activity
HMP     Output menu position
EDT     EDT as subroutine
> <     Scroll within a field
+ -     Page within a table
M+ M-   Page within the message area
?       Display further input options possible in the
        command area ("?" must be sent off using the "DUE"
        key)
TR0     Terminate diagnostic trace
TR1     Diagnostic trace level 1
TR2     Diagnostic trace level 2
ILOG    Activate/deactivate logging in the instruction file
EXEC    Activate/deactivate the execution of statements if logging
--------------------------------------------------------------------------------
 ===>: +                F3=Terminate help
--------------------------------------------------------------------------------

________________________________________________________________________________
LTG                                                         TAST
```



```
HLP.1                           HELP, COMMAND LINE                         SESAM/SQL
------------------------------------------------------------------------------------
        to an instruction file is activated.

Special function keys

         F6  key  Output configuration file

Instead of pressing a function key, you may also enter the name of
the key in the command area, e.g. enter "F12" instead of pressing the
function key F12.

The following key assignments and possible input are also displayed:

         F1  key  Request help
         F3  key  Terminate the Utility Monitor
         F12 key  Abort current function
         F13 key  Terminate input to current form and return
                  to previous form
------------------------------------------------------------------------------------
 ===>: +                 F3=Terminate help
------------------------------------------------------------------------------------

LTG                                                          TAST
```



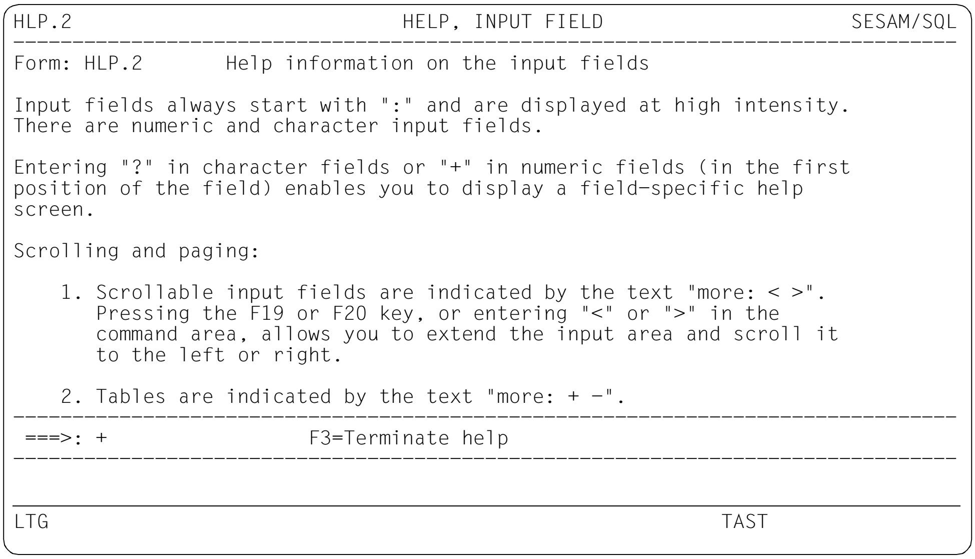
```
HLP.1                           HELP, COMMAND LINE                         SESAM/SQL
------------------------------------------------------------------------------------
         M+-      Page in the message area















------------------------------------------------------------------------------------
 ===>: F3                F3=Terminate help
------------------------------------------------------------------------------------

LTG                                                          TAST
```

You enter F3 to terminate the help function for the command line and return to the STM start form (see page 13).

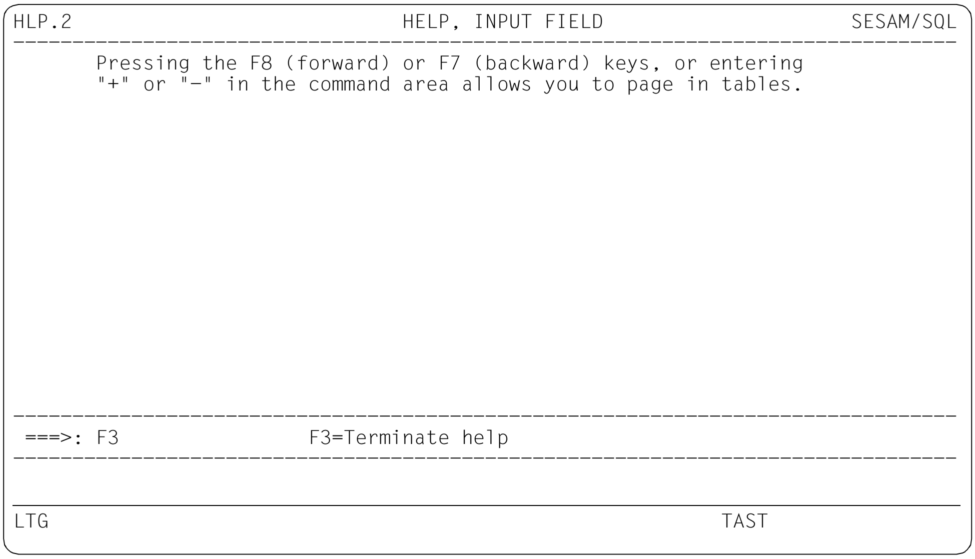
In the STM form you can select function “12” again.
You then branch to the HLP form (see page 14), in which you then select function “2”. You then branch to the continuation form HLP.2.

Displaying a help text on the input fields

The HLP.2 form appears and contains a help text on the input fields.
You enter + in the command area to display additional pages.



```
HLP.2                           HELP, INPUT FIELD                     SESAM/SQL
-------------------------------------------------------------------------------
Form: HLP.2       Help information on the input fields

Input fields always start with ":" and are displayed at high intensity.
There are numeric and character input fields.

Entering "?" in character fields or "+" in numeric fields (in the first
position of the field) enables you to display a field-specific help
screen.

Scrolling and paging:

    1. Scrollable input fields are indicated by the text "more: < >".
       Pressing the F19 or F20 key, or entering "<" or ">" in the
       command area, allows you to extend the input area and scroll it
       to the left or right.

    2. Tables are indicated by the text "more: + -".
-------------------------------------------------------------------------------
 ===>: +                F3=Terminate help
-------------------------------------------------------------------------------

LTG                                                         TAST
```



```
HLP.2                           HELP, INPUT FIELD                     SESAM/SQL
-------------------------------------------------------------------------------
       Pressing the F8 (forward) or F7 (backward) keys, or entering
       "+" or "-" in the command area allows you to page in tables.

-------------------------------------------------------------------------------
 ===>: F3               F3=Terminate help
-------------------------------------------------------------------------------

LTG                                                         TAST
```

You enter F3 to terminate the help function for the input fields and return to the STM start form.

2.3 Terminating the utility monitor

Enter F3 to exit the utility monitor. The following message is issued:

```
SEE1123 EXIT UTILITY MONITOR? IF SO, REPEAT INPUT.
```

You confirm your specification be entering F3 again. The utility monitor is not terminated if you do not enter anything or specify something other than F3.

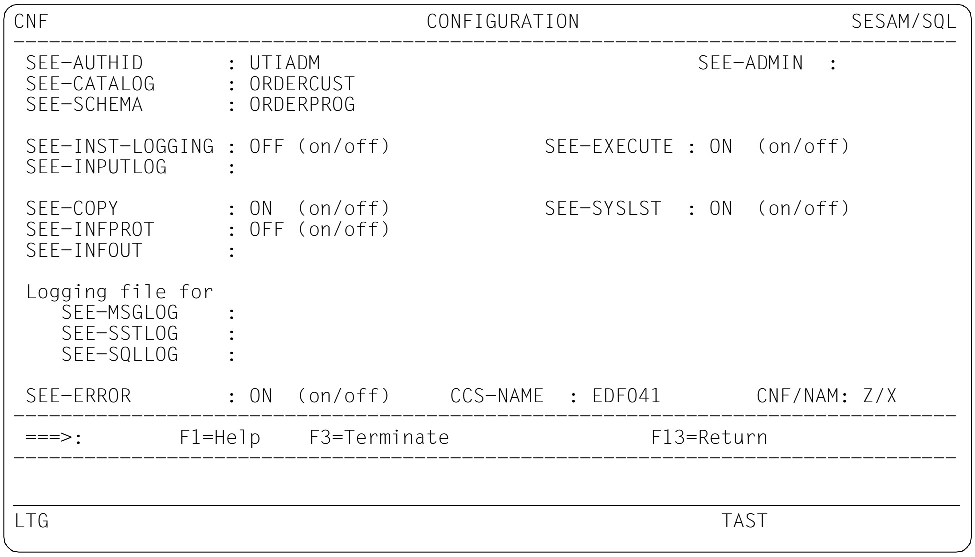
```
%  SEZ4601 PROGRAM 'SESUTI/SQL' TERMINATED NORMALLY
```

2.4 Creating a database

In this example, the ORDERCUST demonstration database is created.

The configuration file SESCONF.SESUTI.ZX is used for the demonstration database. When you start the utility monitor, the STM start form appears, see page 13.

You send off the start form with the preset value “01”. You then branch to the CNF form.



```
CNF                                  CONFIGURATION                                SESAM/SQL
-------------------------------------------------------------------------------------------
 SEE-AUTHID        : UTIADM                                        SEE-ADMIN  :
 SEE-CATALOG       : ORDERCUST
 SEE-SCHEMA        : ORDERPROG

 SEE-INST-LOGGING  : OFF (on/off)             SEE-EXECUTE : ON  (on/off)
 SEE-INPUTLOG      :

 SEE-COPY          : ON  (on/off)             SEE-SYSLST  : ON  (on/off)
 SEE-INFPROT       : OFF (on/off)
 SEE-INFOUT        :

 Logging file for
    SEE-MSGLOG     :
    SEE-SSTLOG     :
    SEE-SQLLOG     :

 SEE-ERROR         : ON  (on/off)     CCS-NAME  : EDF041       CNF/NAM: Z/X
-------------------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------------------

LTG                                                         TAST
```

Changing the authorization key and creating an instruction file

You want to use the authorization key UTIUNIV to create the database.
For the current session, you therefore have to change the authorization key UTIADM defined in the configuration file. You do this in the CNF form (SEE-AUTHID parameter). You also want the statements to be logged in the SESUTI.USRDAT.CREATE instruction file (SEE-INST-LOGGING and SEE-INPUTLOG parameters).

In the CNF form, you overwrite the default values for SEE-AUTHID and SEE-INST-LOGGING and enter the name of the instruction file in the SEE-INPUTLOG field.

Automatic backup

The SEE-COPY parameter determines whether or not SESAM/SQL automatically asks you whether to backup database objects whenever you open and exit the forms ALC - ALTER CATALOG, ALS - ALTER SCHEMA, CRS - CREATE SCHEMA, IMP - IMPORT TABLE and LOD - LOAD as well as when you exit the form CRC - CREATE CATALOG. For further information, refer to section “Selecting configuration parameters” on page 79.

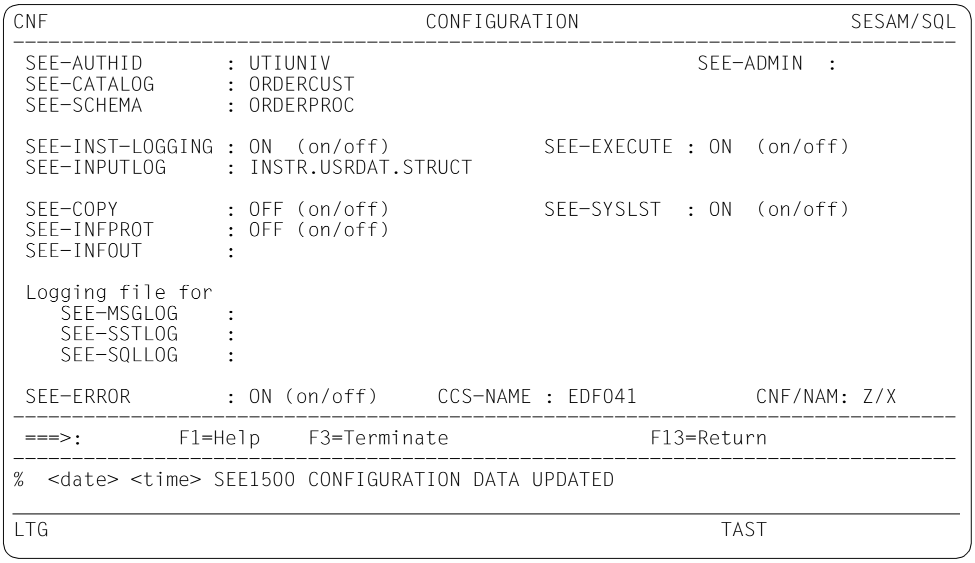
The SESAM/SQL default value for SEE-COPY is ON. You should overwrite the default value with OFF if you want to determine the time at which your database objects are backed up yourself.

When you open or exit any of the above-mentioned forms, the form COP - COPY & RECOVER / REPLICATION is opened with the title “AUTOMATIC BACKUP, COPY”.

Press the DUE key to send off the form. This displays the continuation form COP.1. In the COP.1 form you can select the required backup method (see section “Creating backup copies and carrying out recovery (COP - COPY & RECOVER / REPLICATION)” on page 208).

Enter F13 to return to the form COP - AUTOMATIC BACKUP, COPY. In this form, you again enter F13 to return to the form you originally selected.

The following examples in the manual assume SEE-COPY=OFF.

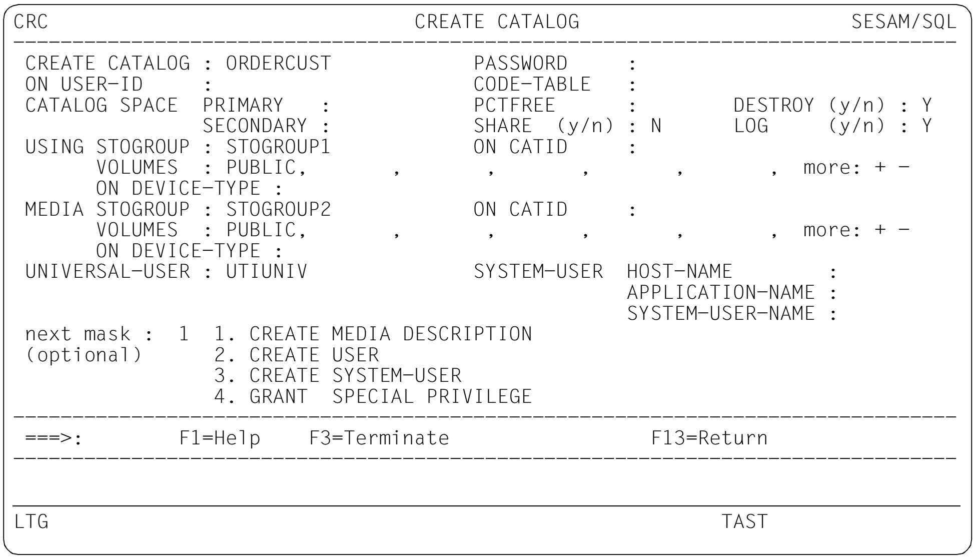

```
CNF                                CONFIGURATION                             SESAM/SQL
--------------------------------------------------------------------------------------
 SEE-AUTHID       : UTIUNIV                                       SEE-ADMIN  :
 SEE-CATALOG      : ORDERCUST
 SEE-SCHEMA       : ORDERPROC

 SEE-INST-LOGGING : ON  (on/off)          SEE-EXECUTE : ON  (on/off)
 SEE-INPUTLOG     : INSTR.USRDAT.STRUCT

 SEE-COPY         : OFF (on/off)          SEE-SYSLST  : ON  (on/off)
 SEE-INFPROT      : OFF (on/off)
 SEE-INFOUT       :

 Logging file for
    SEE-MSGLOG    :
    SEE-SSTLOG    :
    SEE-SQLLOG    :

 SEE-ERROR        : ON (on/off)     CCS-NAME : EDF041          CNF/NAM: Z/X
--------------------------------------------------------------------------------------
 ===>:         F1=Help     F3=Terminate                F13=Return
--------------------------------------------------------------------------------------
%  <date> <time> SEE1500 CONFIGURATION DATA UPDATED

LTG                                                            TAST
```

You can enter F13 to display the STM form.

Creating the catalog space and entering a universal user

Now enter 13 in the function menu in the STM form (see page 13). You now branch to the CRC form.
In the CRC form, enter the name of the database to be created and the universal user, in the CREATE CATALOG and UNIVERSAL-USER fields respectively.

The catalog space should be created in the storage group STOGROUP1 while the CAT-REC file and the first CAT-LOG file should be created in STOGROUP2. The ensures that a media recovery can be performed even if a storage group is lost.



```
CRC                                CREATE CATALOG                                SESAM/SQL
-----------------------------------------------------------------------------------------
 CREATE CATALOG : ORDERCUST               PASSWORD     :
 ON USER-ID     :                         CODE-TABLE   :
 CATALOG SPACE  PRIMARY   :               PCTFREE      :      DESTROY (y/n) : Y
                SECONDARY :               SHARE  (y/n) : N    LOG     (y/n) : Y
 USING STOGROUP : STOGROUP1               ON CATID     :
       VOLUMES  : PUBLIC,       ,       ,       ,       ,       ,  more: + -
       ON DEVICE-TYPE :
 MEDIA STOGROUP : STOGROUP2               ON CATID     :
       VOLUMES  : PUBLIC,       ,       ,       ,       ,       ,  more: + -
       ON DEVICE-TYPE :
 UNIVERSAL-USER : UTIUNIV                 SYSTEM-USER  HOST-NAME        :
                                                       APPLICATION-NAME :
                                                       SYSTEM-USER-NAME :
 next mask :  1  1. CREATE MEDIA DESCRIPTION
 (optional)      2. CREATE USER
                 3. CREATE SYSTEM-USER
                 4. GRANT  SPECIAL PRIVILEGE
-----------------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
-----------------------------------------------------------------------------------------

_________________________________________________________________________________________
LTG                                                          TAST
```

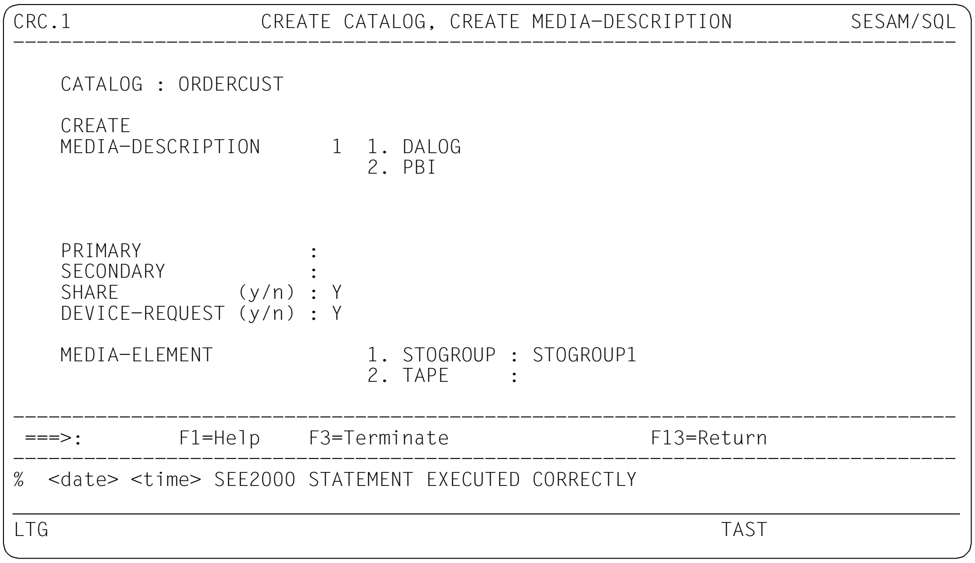
You can now call the continuation forms CRC.1 through CRC.4.

- With CREATE MEDIA DESCRIPTION you define the file attributes of the DA-LOG file and the PBI file if you do not want to take over the default configuration (see “SQL Reference Manual Part 2: Utilities”).
- With the continuation forms CRC.2 - CRC.4 you design access control mechanisms for the database that you are going to create (see the “Core manual”). To do this, you must perform the following steps in sequence:
  - Create the authorization keys, form CRC.2.
  - Generate system entries, form CRC.3.
  - Grant special privileges, form CRC.4.

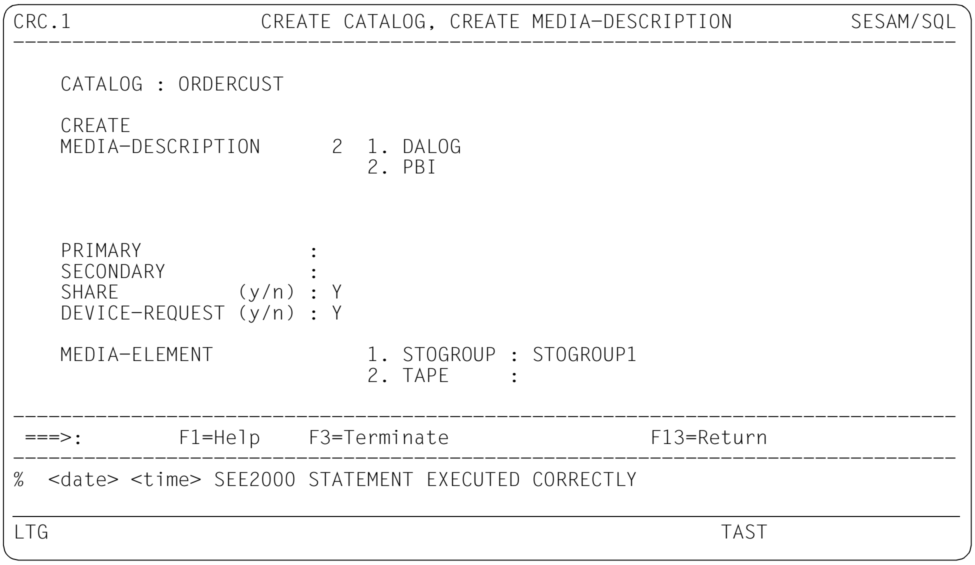
Choose 1 or 2 in the function menu to define the file attributes of the DA-LOG and PBI files or to generate the authorization keys.
If you want to skip these steps, press the DUE key to send the CRC form. The database ORDERCUST is created.

Creating database-specific files

In the form CRC.1 you now create the file attributes of the database-specific DA-LOG and PBI files by selecting function 1 and then function 2. The utility monitor confirms that your entries have been accepted.



```
CRC.1                CREATE CATALOG, CREATE MEDIA-DESCRIPTION         SESAM/SQL
-------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    CREATE
    MEDIA-DESCRIPTION      1  1. DALOG
                              2. PBI



    PRIMARY              :
    SECONDARY            :
    SHARE          (y/n) : Y
    DEVICE-REQUEST (y/n) : Y

    MEDIA-ELEMENT             1. STOGROUP : STOGROUP1
                              2. TAPE     :

-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                         TAST
```



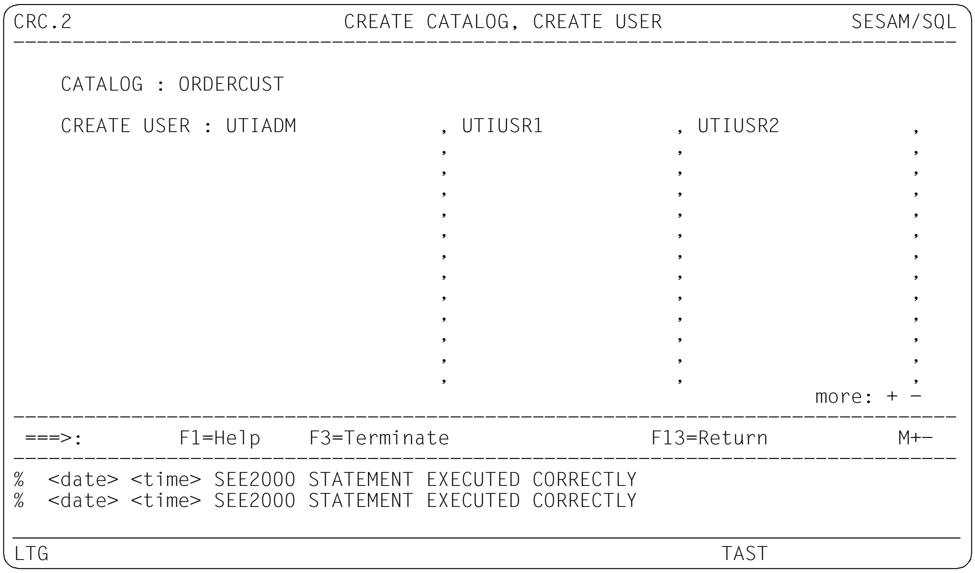
```
CRC.1                CREATE CATALOG, CREATE MEDIA-DESCRIPTION         SESAM/SQL
-------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    CREATE
    MEDIA-DESCRIPTION      2  1. DALOG
                              2. PBI



    PRIMARY              :
    SECONDARY            :
    SHARE          (y/n) : Y
    DEVICE-REQUEST (y/n) : Y

    MEDIA-ELEMENT             1. STOGROUP : STOGROUP1
                              2. TAPE     :

-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                         TAST
```

Enter F13 to return to the CRC form.

Creating an authorization key

Enter 2 in the next mask selection of the CRC form (see page 21).
This opens the continuation form CRC.2.

In the CRC.2 continuation form you create the authorization keys UTIADM, UTIUSR1 and UTISUSR2. The utility monitor confirms that your entries have been accepted.



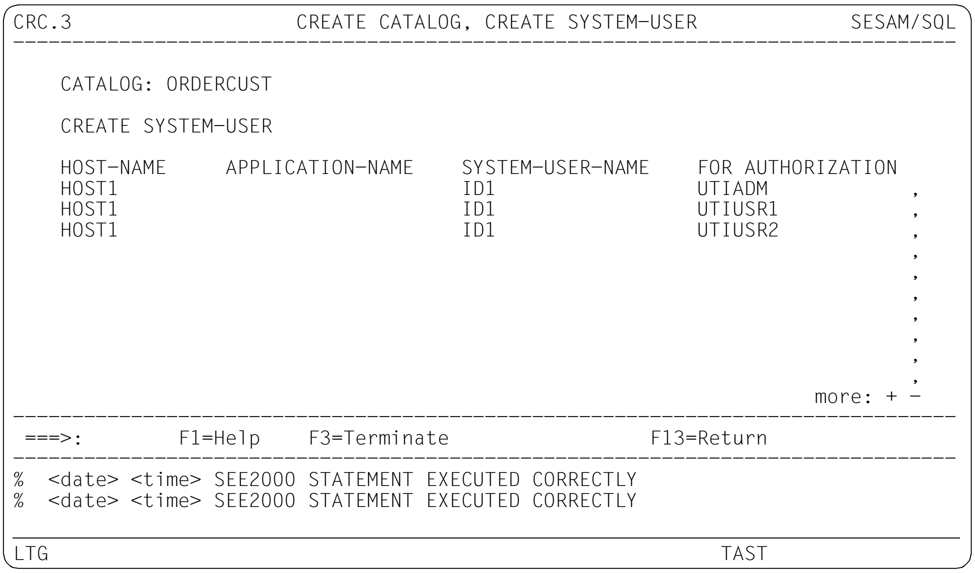
```
CRC.2                        CREATE CATALOG, CREATE USER                SESAM/SQL
-------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    CREATE USER : UTIADM            , UTIUSR1           , UTIUSR2          ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                    ,                   ,                  ,
                                                                  more: + -
-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return           M+-
-------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                          TAST
```

Enter F13 to return to the CRC form.

Creating system entries

Enter 3 in the next mask selection of the CRC form (see page 21).
You now branch to the CRC.3 continuation form.

In the CRC.3 continuation form you create the system entries. The utility monitor confirms that your entries have been accepted.



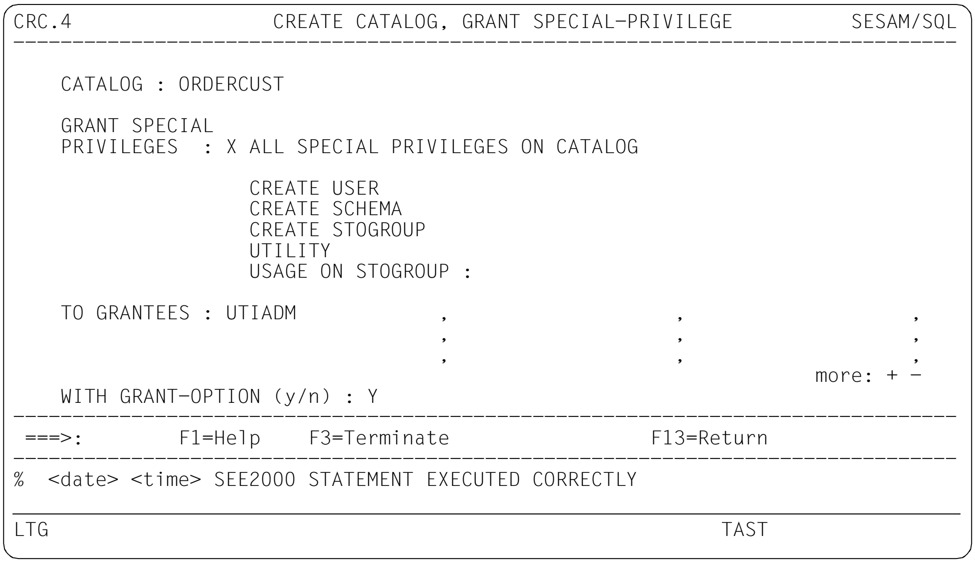
```
CRC.3                  CREATE CATALOG, CREATE SYSTEM-USER            SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG: ORDERCUST

    CREATE SYSTEM-USER

    HOST-NAME     APPLICATION-NAME    SYSTEM-USER-NAME    FOR AUTHORIZATION
    HOST1                             ID1                 UTIADM               ,
    HOST1                             ID1                 UTIUSR1              ,
    HOST1                             ID1                 UTIUSR2              ,
                                                                               ,
                                                                               ,
                                                                               ,
                                                                               ,
                                                                               ,
                                                                               ,
                                                                               ,
                                                                        more: + -
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                  F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                         TAST
```

Enter F13 to return to the CRC form.

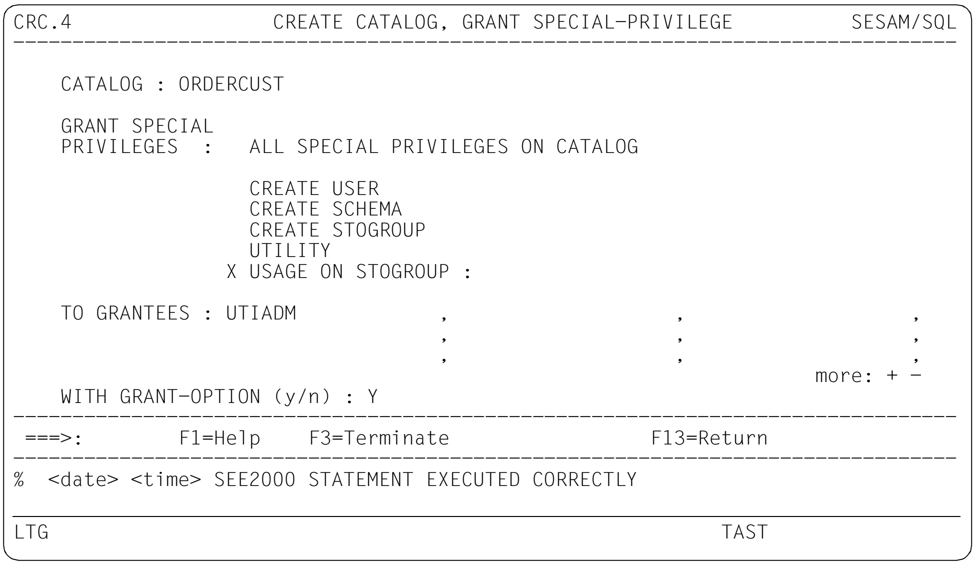
Granting special privileges

Enter 4 in the next mask selection of the CRC form (see page 21).
You now branch to the CRC.4 continuation form .

In the CRC.4 continuation form, you grant the special privileges `ALL SPECIAL PRIVILEGES ON CATALOG` and `USAGE ON STOGROUP : STOGROUP1` in two steps by marking them with an x. The utility monitor confirms that your entries have been accepted.



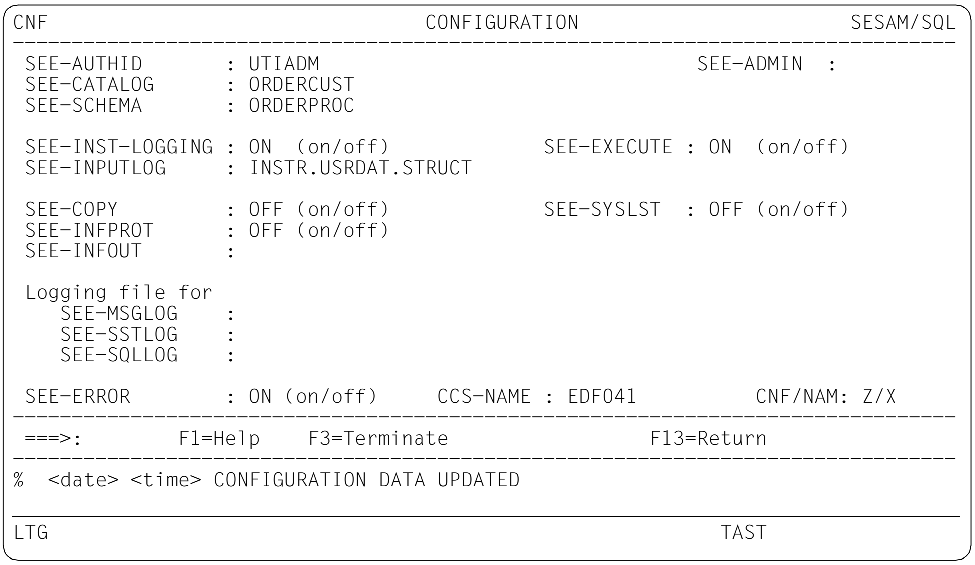
```
CRC.4                  CREATE CATALOG, GRANT SPECIAL-PRIVILEGE          SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    GRANT SPECIAL
    PRIVILEGES  : X ALL SPECIAL PRIVILEGES ON CATALOG

                  CREATE USER
                  CREATE SCHEMA
                  CREATE STOGROUP
                  UTILITY
                  USAGE ON STOGROUP :

    TO GRANTEES : UTIADM             ,                   ,                   ,
                                     ,                   ,                   ,
                                     ,                   ,                   ,
                                                                    more: + -
    WITH GRANT-OPTION (y/n) : Y
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                         TAST
```



```
CRC.4                  CREATE CATALOG, GRANT SPECIAL-PRIVILEGE          SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    GRANT SPECIAL
    PRIVILEGES  :   ALL SPECIAL PRIVILEGES ON CATALOG

                  CREATE USER
                  CREATE SCHEMA
                  CREATE STOGROUP
                  UTILITY
                X USAGE ON STOGROUP :

    TO GRANTEES : UTIADM             ,                   ,                   ,
                                     ,                   ,                   ,
                                     ,                   ,                   ,
                                                                    more: + -
    WITH GRANT-OPTION (y/n) : Y
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                         TAST
```

Changing the authorization key

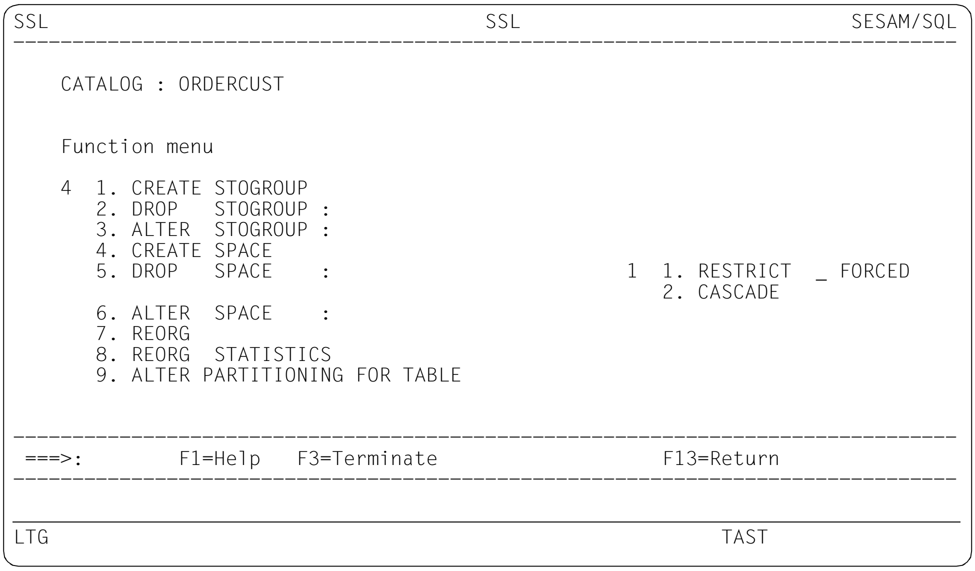
You enter cnf in the command area to display the CNF form, in which you change the authorization key again from UTIUNIV to UTIADM (see page 19). The utility monitor confirms that your entries have been accepted.



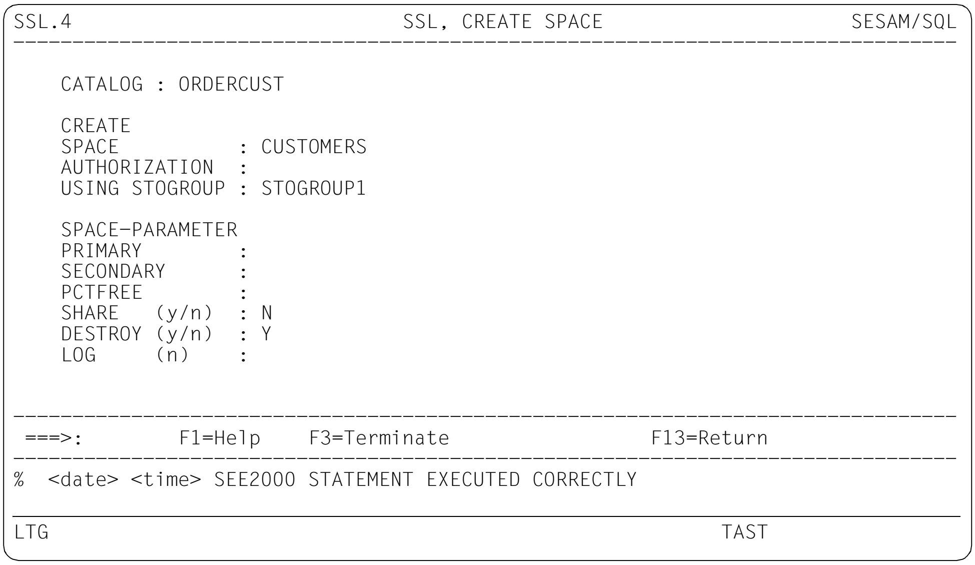
```
CNF                                  CONFIGURATION                               SESAM/SQL
-------------------------------------------------------------------------------------------
 SEE-AUTHID       : UTIADM                                       SEE-ADMIN  :
 SEE-CATALOG      : ORDERCUST
 SEE-SCHEMA       : ORDERPROC

 SEE-INST-LOGGING : ON  (on/off)            SEE-EXECUTE : ON  (on/off)
 SEE-INPUTLOG     : INSTR.USRDAT.STRUCT

 SEE-COPY         : OFF (on/off)            SEE-SYSLST  : OFF (on/off)
 SEE-INFPROT      : OFF (on/off)
 SEE-INFOUT       :

 Logging file for
    SEE-MSGLOG    :
    SEE-SSTLOG    :
    SEE-SQLLOG    :

 SEE-ERROR        : ON (on/off)     CCS-NAME : EDF041          CNF/NAM: Z/X
-------------------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
-------------------------------------------------------------------------------------------
%  <date> <time> CONFIGURATION DATA UPDATED

LTG                                                          TAST
```

Creating user spaces

You enter ssl in the command area to display the SSL form, in which you create the user spaces Select function 4 in the function menu.
This displays the SSL.4 continuation form. There you create the user space CUSTOMERS.
The utility monitor confirms that your entries have been accepted.



```
SSL                                     SSL                               SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST


    Function menu

    4  1. CREATE STOGROUP
       2. DROP   STOGROUP :
       3. ALTER  STOGROUP :
       4. CREATE SPACE
       5. DROP   SPACE    :                        1  1. RESTRICT  _ FORCED
                                                      2. CASCADE
       6. ALTER  SPACE    :
       7. REORG
       8. REORG  STATISTICS
       9. ALTER PARTITIONING FOR TABLE


--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                      F13=Return
--------------------------------------------------------------------------------

LTG                                                            TAST
```



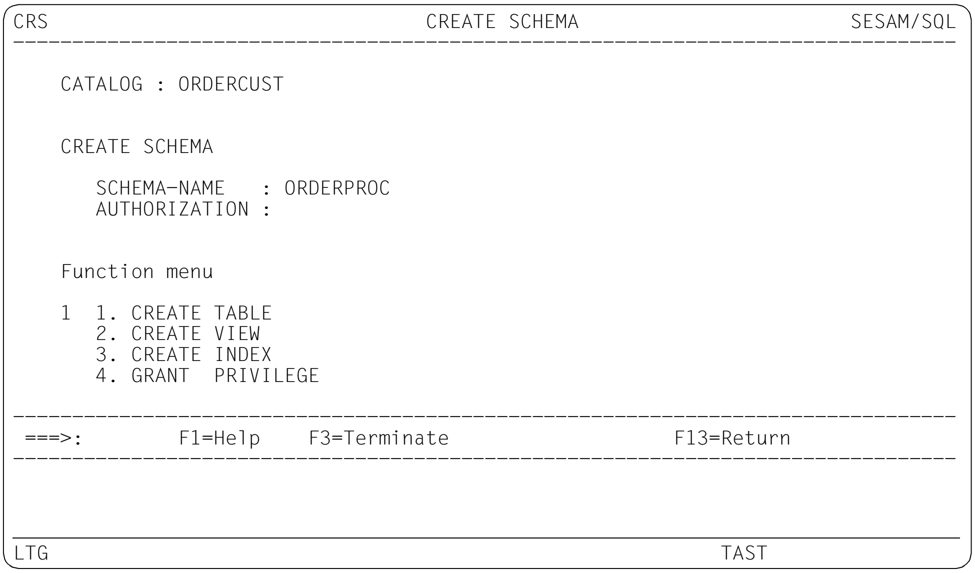
```
SSL.4                            SSL, CREATE SPACE                        SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    CREATE
    SPACE          : CUSTOMERS
    AUTHORIZATION  :
    USING STOGROUP : STOGROUP1

    SPACE-PARAMETER
    PRIMARY        :
    SECONDARY      :
    PCTFREE        :
    SHARE   (y/n)  : N
    DESTROY (y/n)  : Y
    LOG     (n)    :


--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                    F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                            TAST
```

You create the user spaces CONTACTS, ORDERS, SERVICE and ORDER_STAT in the same way.

Creating a schema

Once you have created all the user spaces, you enter stm in the command area. This displays the STM form (see page 13), in which you select function 15 (CREATE SCHEMA) from the function menu.

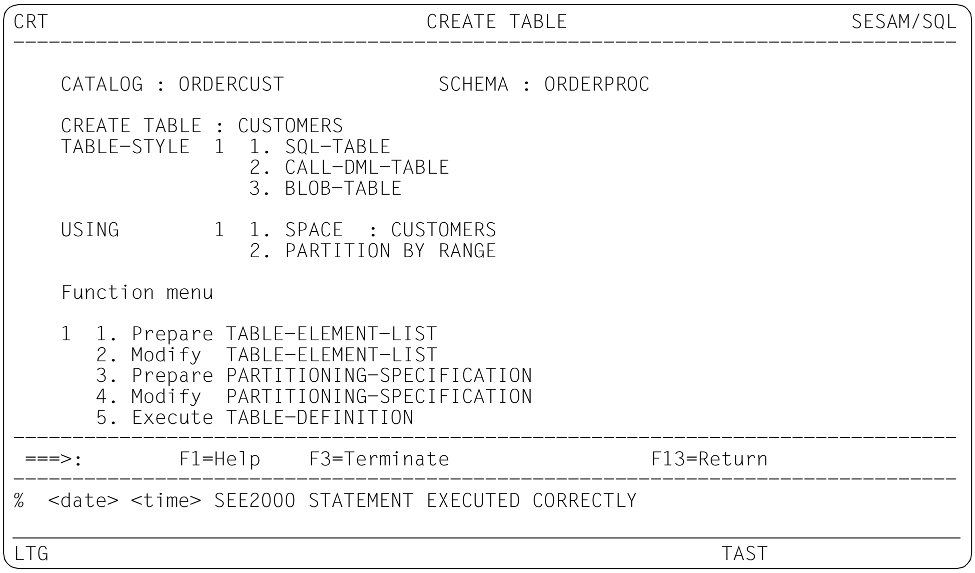
In the CRS form you enter the schema name Select function 1, the default, from the function menu. You send the mask by pressing the DUE key. The ORDERPROC schema is created and the utility monitor confirms that your entries have been accepted.
This operation simultaneously opens the CRT form.



```
CRS                                   CREATE SCHEMA                           SESAM/SQL
-------------------------------------------------------------------------------

    CATALOG : ORDERCUST


    CREATE SCHEMA

       SCHEMA-NAME    : ORDERPROC
       AUTHORIZATION :


    Function menu

    1  1. CREATE TABLE
       2. CREATE VIEW
       3. CREATE INDEX
       4. GRANT  PRIVILEGE

-------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                  F13=Return
-------------------------------------------------------------------------------



LTG                                                        TAST
```

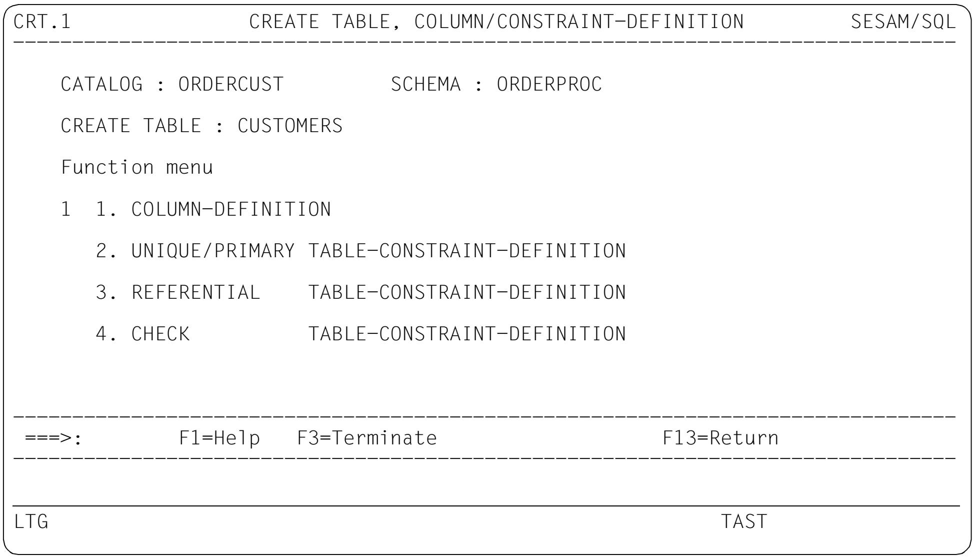
Defining a base table

In the CRT form you want to define the CUSTOMERS base table (in the CREATE TABLE field). The table is to be created in the user space CUSTOMERS (USING SPACE field) and is to be an SQL table (TABLE-STYLE field, default value 1 for SQL-TABLE). In the function menu, you select function 1, Prepare TABLE-ELEMENT-LIST“, which is the default.
This displays the CRT.1 continuation form.


```
CRT                                CREATE TABLE                            SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST             SCHEMA : ORDERPROC

    CREATE TABLE : CUSTOMERS
    TABLE-STYLE  1  1. SQL-TABLE
                    2. CALL-DML-TABLE
                    3. BLOB-TABLE

    USING        1  1. SPACE  : CUSTOMERS
                    2. PARTITION BY RANGE

    Function menu

    1  1. Prepare TABLE-ELEMENT-LIST
       2. Modify  TABLE-ELEMENT-LIST
       3. Prepare PARTITIONING-SPECIFICATION
       4. Modify  PARTITIONING-SPECIFICATION
       5. Execute TABLE-DEFINITION
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                         TAST
```

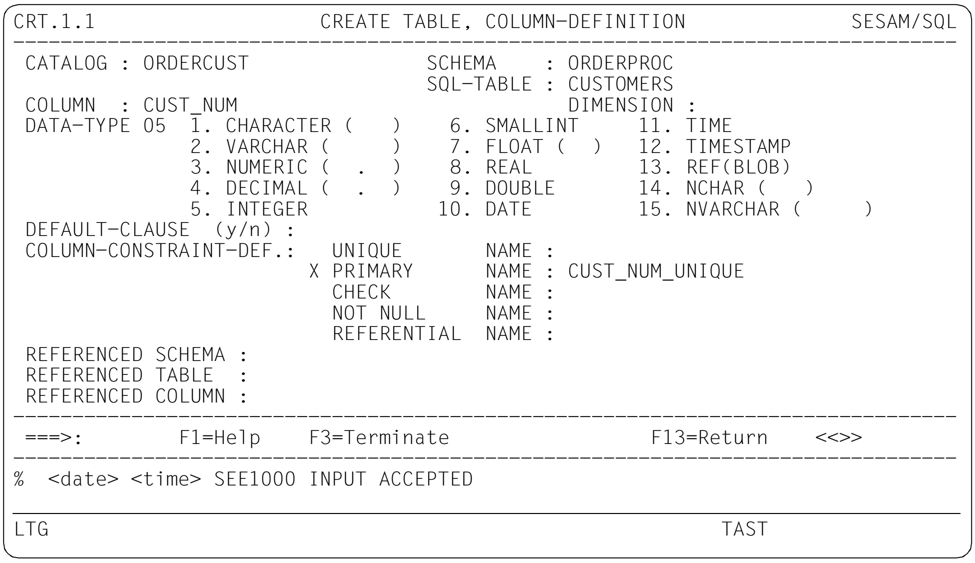
Defining columns

In the CRT.1 form you select function 1, COLUMN-DEFINITION.
This displays the CRT.1.1 continuation form.

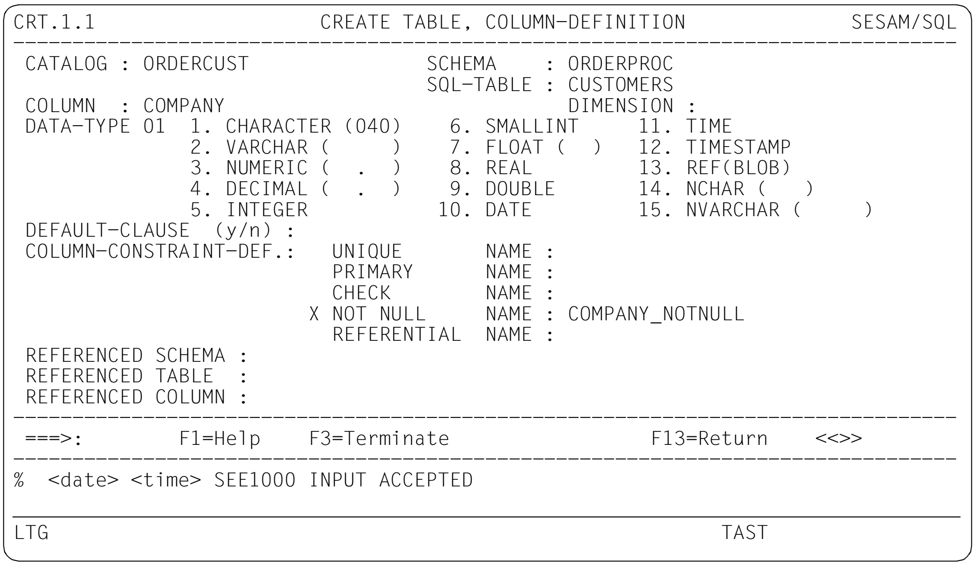


```
CRT.1               CREATE TABLE, COLUMN/CONSTRAINT-DEFINITION         SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST         SCHEMA : ORDERPROC

    CREATE TABLE : CUSTOMERS

    Function menu

    1  1. COLUMN-DEFINITION

       2. UNIQUE/PRIMARY TABLE-CONSTRAINT-DEFINITION

       3. REFERENTIAL    TABLE-CONSTRAINT-DEFINITION

       4. CHECK          TABLE-CONSTRAINT-DEFINITION


--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------

LTG                                                        TAST
```

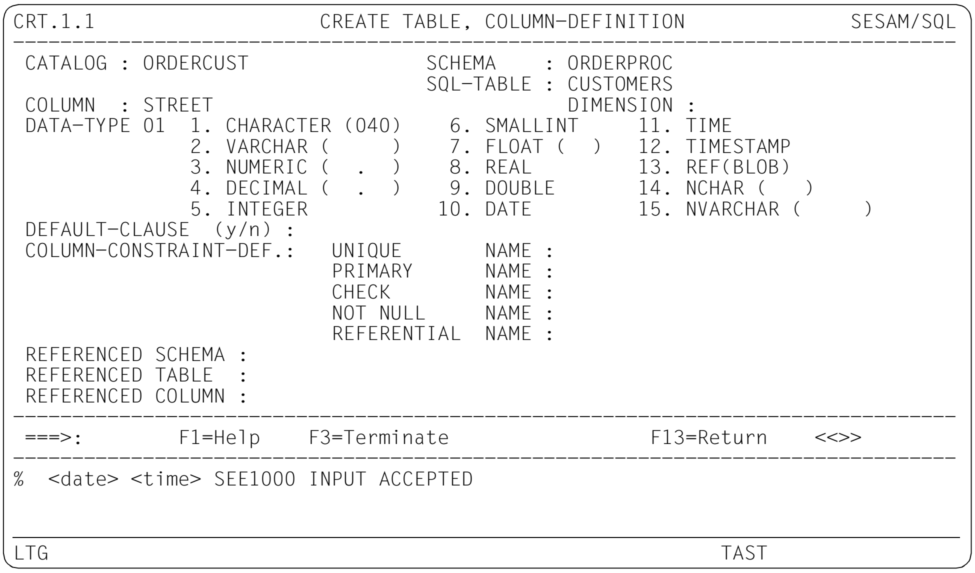
In the CRT.1.1 form you define the CUST_NUM, COMPANY, STREET, ZIP, CITY, COUNTRY, CUST_TEL and CUST_INFO columns, one after the other. You send off each column by pressing the DUE key. The utility monitor confirms that your entries have been accepted. By entering << and >> in the command area you can page through entries you have already sent off.



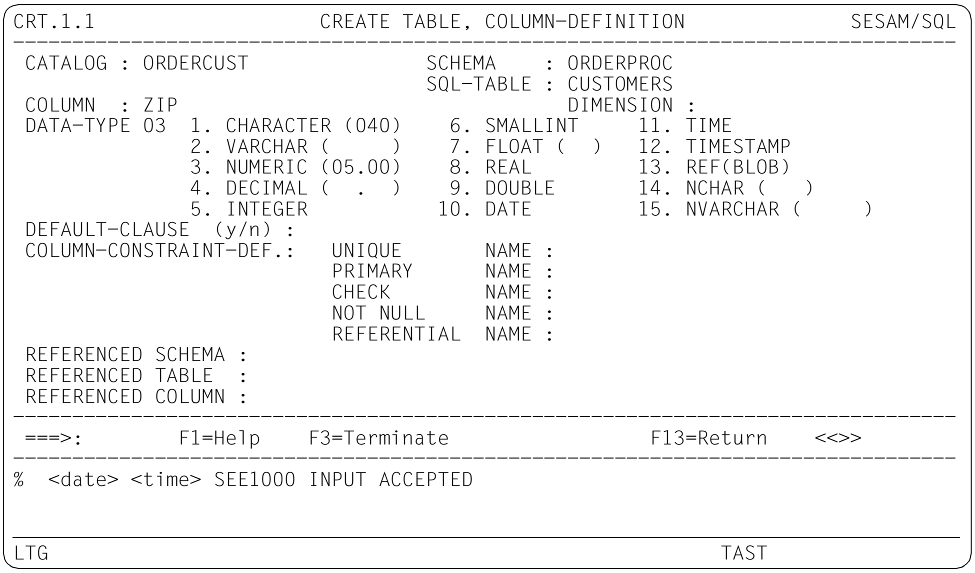
```
CRT.1.1                   CREATE TABLE, COLUMN-DEFINITION              SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                  SCHEMA    : ORDERPROC
                                      SQL-TABLE : CUSTOMERS
 COLUMN  : CUST_NUM                               DIMENSION :
 DATA-TYPE 05  1. CHARACTER (   )    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (  .  )    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                         X PRIMARY      NAME : CUST_NUM_UNIQUE
                           CHECK        NAME :
                           NOT NULL     NAME :
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED

LTG                                                          TAST
```



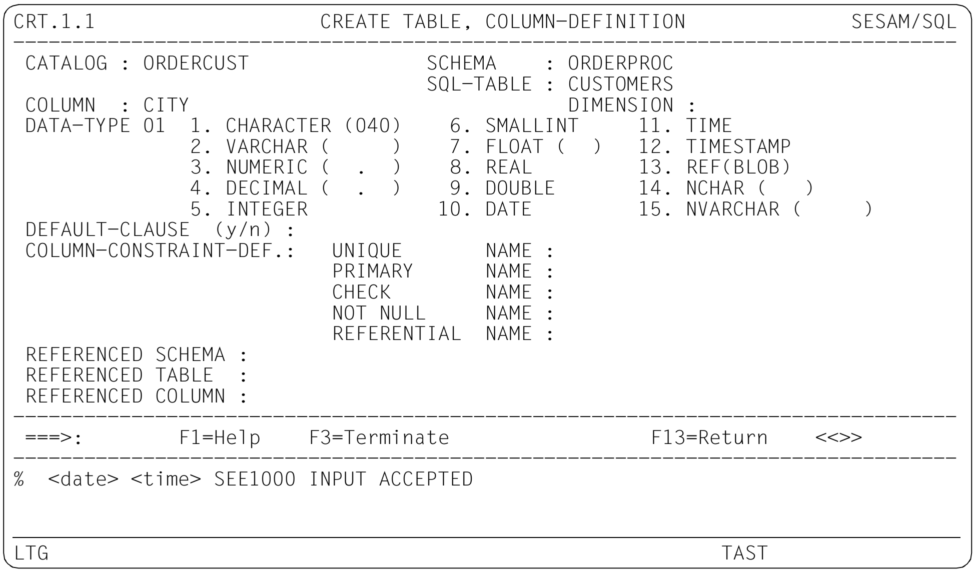
```
CRT.1.1                   CREATE TABLE, COLUMN-DEFINITION              SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                  SCHEMA    : ORDERPROC
                                      SQL-TABLE : CUSTOMERS
 COLUMN  : COMPANY                                DIMENSION :
 DATA-TYPE 01  1. CHARACTER (040)    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (  .  )    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                           PRIMARY      NAME :
                           CHECK        NAME :
                         X NOT NULL     NAME : COMPANY_NOTNULL
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED

LTG                                                          TAST
```



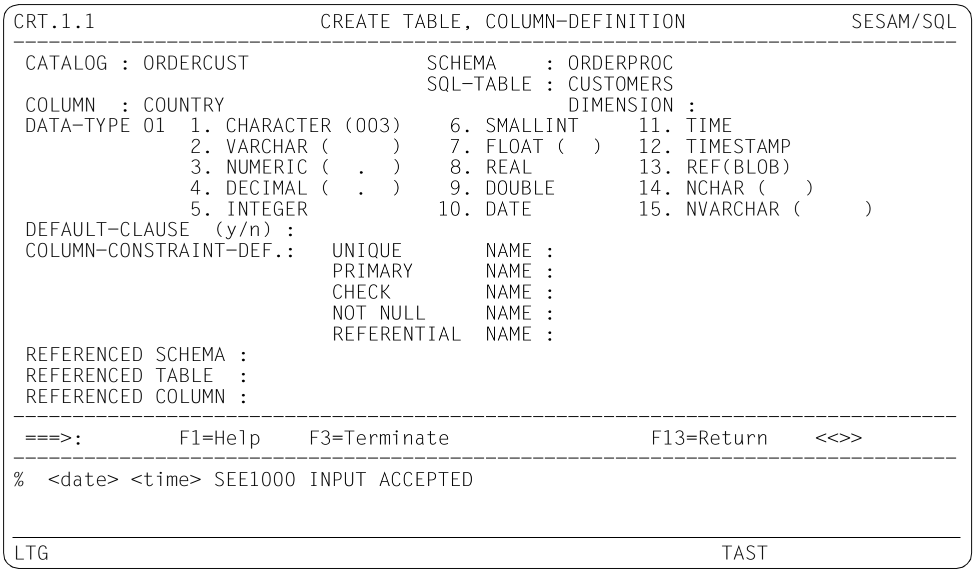
```
CRT.1.1                    CREATE TABLE, COLUMN-DEFINITION              SESAM/SQL
---------------------------------------------------------------------------------
 CATALOG : ORDERCUST                 SCHEMA    : ORDERPROC
                                     SQL-TABLE : CUSTOMERS
 COLUMN  : STREET                                DIMENSION :
 DATA-TYPE 01  1. CHARACTER (040)    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (  .  )    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                           PRIMARY      NAME :
                           CHECK        NAME :
                           NOT NULL     NAME :
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
---------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
---------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                          TAST
```



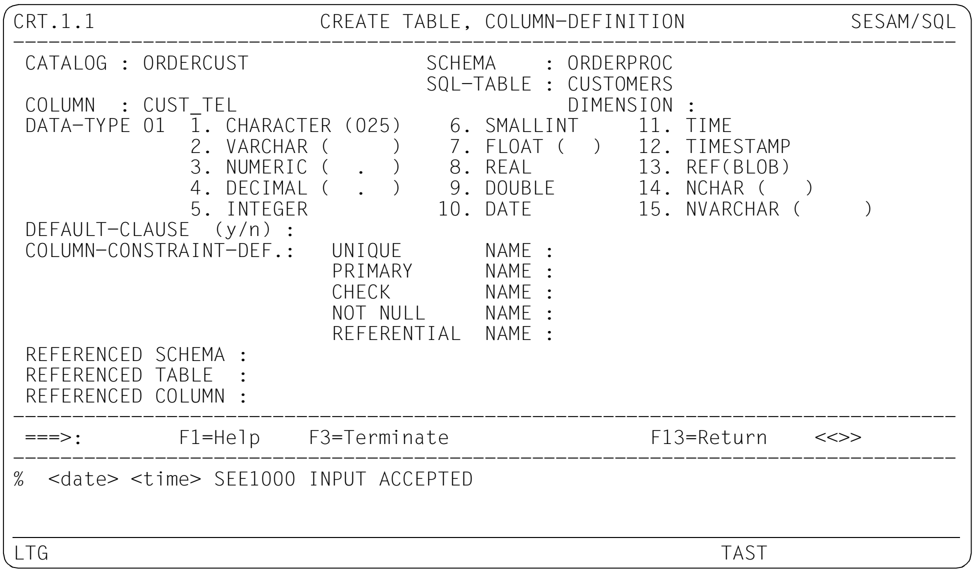
```
CRT.1.1                    CREATE TABLE, COLUMN-DEFINITION              SESAM/SQL
---------------------------------------------------------------------------------
 CATALOG : ORDERCUST                 SCHEMA    : ORDERPROC
                                     SQL-TABLE : CUSTOMERS
 COLUMN  : ZIP                                   DIMENSION :
 DATA-TYPE 03  1. CHARACTER (040)    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (05.00)    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                           PRIMARY      NAME :
                           CHECK        NAME :
                           NOT NULL     NAME :
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
---------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
---------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                          TAST
```



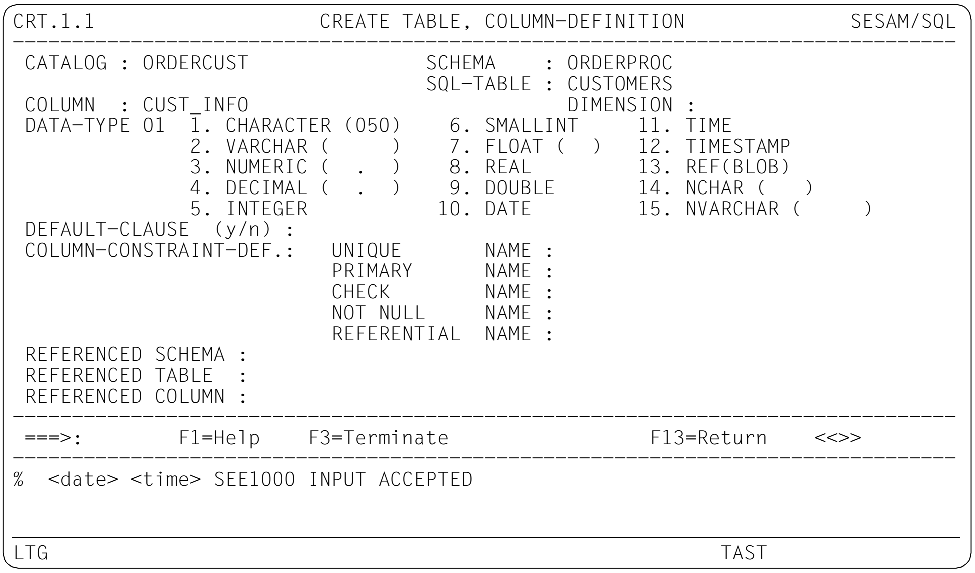
```
CRT.1.1                   CREATE TABLE, COLUMN-DEFINITION             SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                  SCHEMA    : ORDERPROC
                                      SQL-TABLE : CUSTOMERS
 COLUMN   : CITY                                  DIMENSION :
 DATA-TYPE 01  1. CHARACTER (040)    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (  .  )    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                           PRIMARY      NAME :
                           CHECK        NAME :
                           NOT NULL     NAME :
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                         TAST
```



```
CRT.1.1                   CREATE TABLE, COLUMN-DEFINITION             SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                  SCHEMA    : ORDERPROC
                                      SQL-TABLE : CUSTOMERS
 COLUMN   : COUNTRY                               DIMENSION :
 DATA-TYPE 01  1. CHARACTER (003)    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (  .  )    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                           PRIMARY      NAME :
                           CHECK        NAME :
                           NOT NULL     NAME :
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                         TAST
```



```
CRT.1.1                   CREATE TABLE, COLUMN-DEFINITION              SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                 SCHEMA    : ORDERPROC
                                     SQL-TABLE : CUSTOMERS
 COLUMN  : CUST_TEL                              DIMENSION :
 DATA-TYPE 01  1. CHARACTER (025)    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (  .  )    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                           PRIMARY      NAME :
                           CHECK        NAME :
                           NOT NULL     NAME :
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                         TAST
```

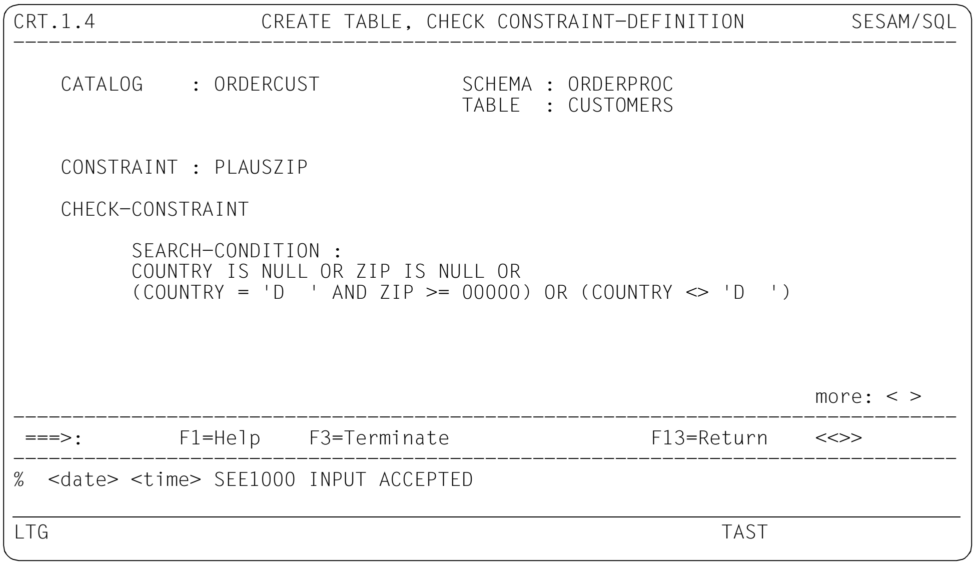


```
CRT.1.1                   CREATE TABLE, COLUMN-DEFINITION              SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                 SCHEMA    : ORDERPROC
                                     SQL-TABLE : CUSTOMERS
 COLUMN  : CUST_INFO                             DIMENSION :
 DATA-TYPE 01  1. CHARACTER (050)    6. SMALLINT     11. TIME
               2. VARCHAR (     )    7. FLOAT (  )   12. TIMESTAMP
               3. NUMERIC (  .  )    8. REAL         13. REF(BLOB)
               4. DECIMAL (  .  )    9. DOUBLE       14. NCHAR (   )
               5. INTEGER           10. DATE         15. NVARCHAR (     )
 DEFAULT-CLAUSE  (y/n) :
 COLUMN-CONSTRAINT-DEF.:   UNIQUE       NAME :
                           PRIMARY      NAME :
                           CHECK        NAME :
                           NOT NULL     NAME :
                           REFERENTIAL  NAME :
 REFERENCED SCHEMA :
 REFERENCED TABLE  :
 REFERENCED COLUMN :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                         TAST
```

Defining a check constraint

Enter F13 to return to the CRT.1 form (see page 30). Select function 4, CHECK TABLE-CONSTRAINT-DEFINITION, from the function menu.

This displays the CRT.1.4 continuation form. There you define a check constraint for the table. The utility monitor confirms that your entries have been accepted.

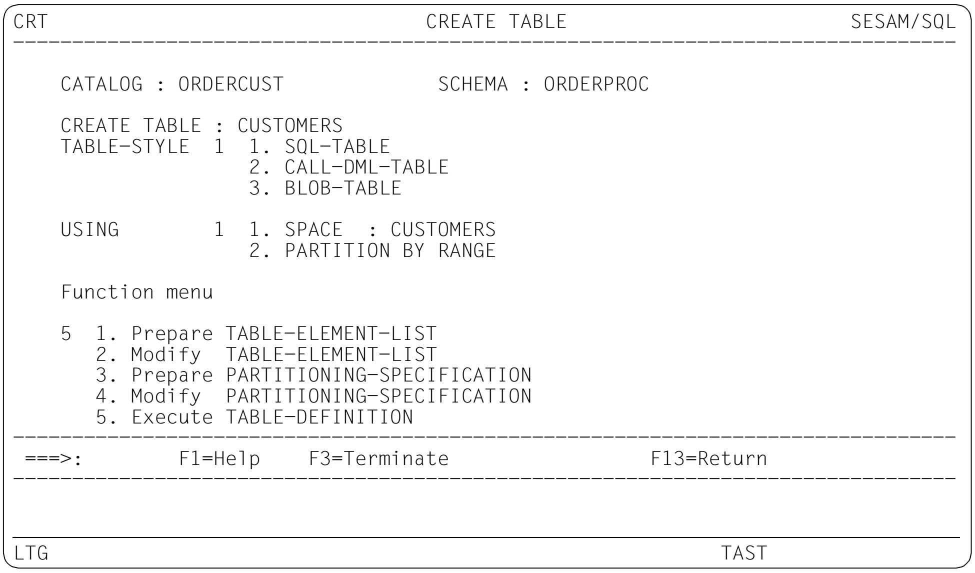


```
CRT.1.4              CREATE TABLE, CHECK CONSTRAINT-DEFINITION        SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG    : ORDERCUST            SCHEMA : ORDERPROC
                                      TABLE  : CUSTOMERS


    CONSTRAINT : PLAUSZIP

    CHECK-CONSTRAINT

          SEARCH-CONDITION :
          COUNTRY IS NULL OR ZIP IS NULL OR
          (COUNTRY = 'D  ' AND ZIP >= 00000) OR (COUNTRY <> 'D  ')




                                                                    more: < >
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED

LTG                                                          TAST
```

Creating base table

Enter F13 to return to the CRT.1 form. In this form, you again enter F13 to return to the CRT form.

You must now select the default function 5, Execute TABLE-DEFINITION, to create the defined table (if necessary, you can use function 2, Modify TABLE-ELEMENT-LIST, to edit the table once it has been created).

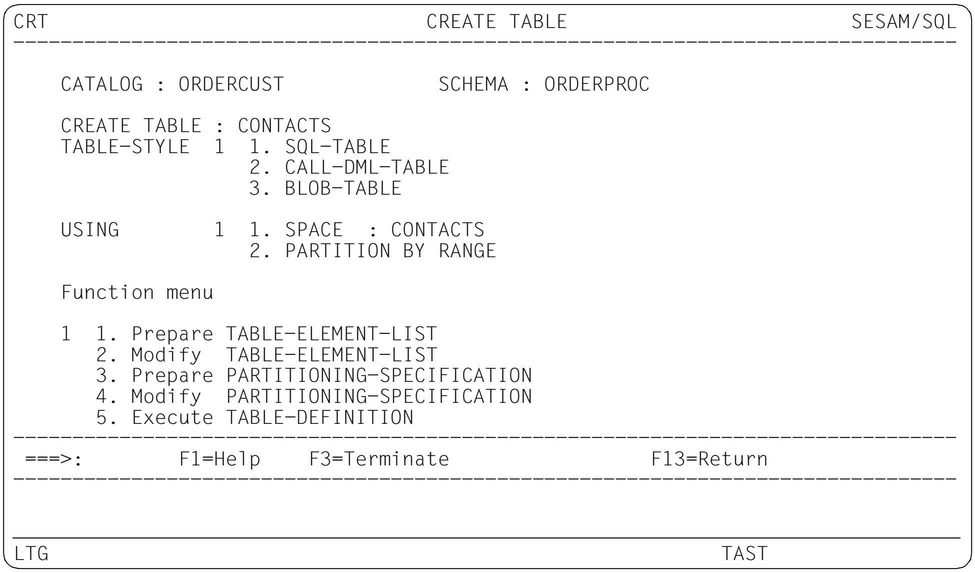


```
CRT                                 CREATE TABLE                             SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST             SCHEMA : ORDERPROC

    CREATE TABLE : CUSTOMERS
    TABLE-STYLE  1  1. SQL-TABLE
                    2. CALL-DML-TABLE
                    3. BLOB-TABLE

    USING        1  1. SPACE  : CUSTOMERS
                    2. PARTITION BY RANGE

    Function menu

    5  1. Prepare TABLE-ELEMENT-LIST
       2. Modify  TABLE-ELEMENT-LIST
       3. Prepare PARTITIONING-SPECIFICATION
       4. Modify  PARTITIONING-SPECIFICATION
       5. Execute TABLE-DEFINITION
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
--------------------------------------------------------------------------------


LTG                                                         TAST
```

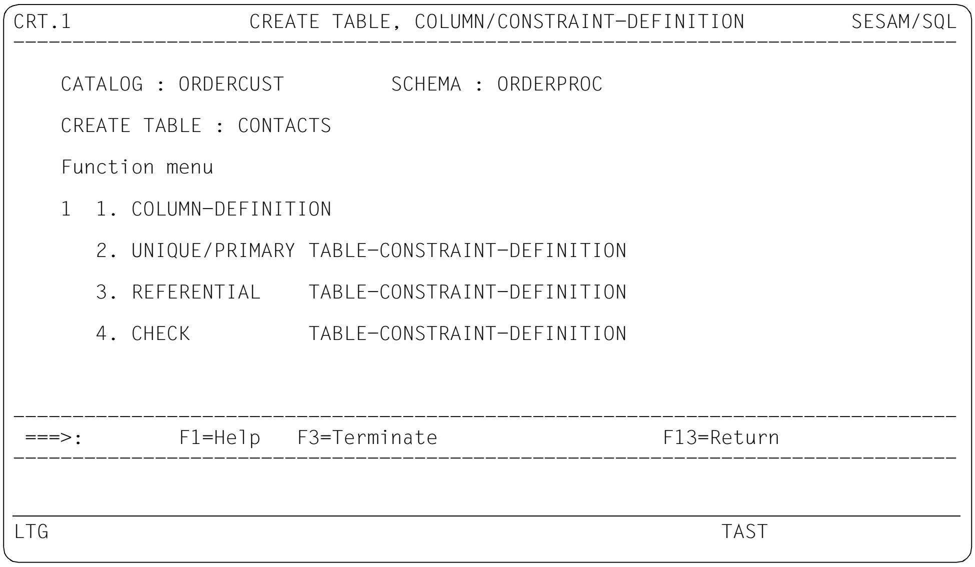
The utility monitor confirms that your entries have been accepted. Function 1 is then the default selection from the function menu again.

Defining and creating further tables

Now you want to define the CONTACTS base table.



```
CRT                                 CREATE TABLE                            SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST             SCHEMA : ORDERPROC

    CREATE TABLE : CONTACTS
    TABLE-STYLE  1  1. SQL-TABLE
                    2. CALL-DML-TABLE
                    3. BLOB-TABLE

    USING        1  1. SPACE  : CONTACTS
                    2. PARTITION BY RANGE

    Function menu

    1  1. Prepare TABLE-ELEMENT-LIST
       2. Modify  TABLE-ELEMENT-LIST
       3. Prepare PARTITIONING-SPECIFICATION
       4. Modify  PARTITIONING-SPECIFICATION
       5. Execute TABLE-DEFINITION
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
--------------------------------------------------------------------------------


LTG                                                         TAST
```



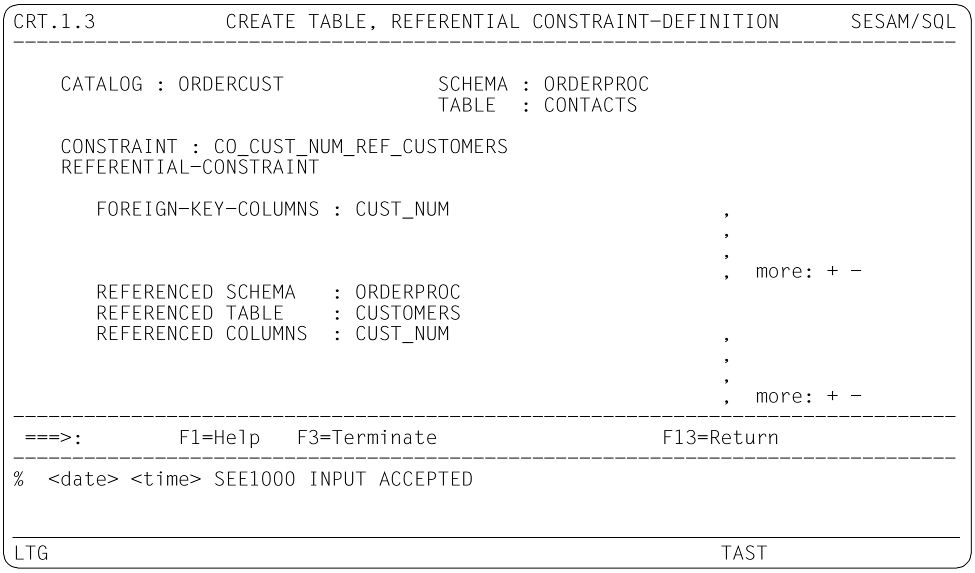
```
CRT.1                CREATE TABLE, COLUMN/CONSTRAINT-DEFINITION         SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST         SCHEMA : ORDERPROC

    CREATE TABLE : CONTACTS

    Function menu

    1  1. COLUMN-DEFINITION

       2. UNIQUE/PRIMARY TABLE-CONSTRAINT-DEFINITION

       3. REFERENTIAL    TABLE-CONSTRAINT-DEFINITION

       4. CHECK          TABLE-CONSTRAINT-DEFINITION


--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                   F13=Return
--------------------------------------------------------------------------------


LTG                                                         TAST
```

You define the CONTACT_NUM, CUST_NUM, FNAME, LNAME, TITLE, CONTACT_TEL, POSITION, DEPARTMENT and CONTACT_INFO columns, one after the other, in the same way as described above for the CUSTOMERS table.

Defining a referential constraint

Once you have defined all the columns, you enter F13 and return to the CRT.1 form (see page 30).
Now select 3 from the function menu. This displays the CRT.1.3 continuation form.

In continuation form CRT.1.3 you define a referential constraint for the CONTACTS table. The utility monitor confirms that your entries have been accepted.


```
CRT.1.3          CREATE TABLE, REFERENTIAL CONSTRAINT-DEFINITION      SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST                 SCHEMA : ORDERPROC
                                        TABLE  : CONTACTS

    CONSTRAINT : CO_CUST_NUM_REF_CUSTOMERS
    REFERENTIAL-CONSTRAINT

       FOREIGN-KEY-COLUMNS : CUST_NUM                          ,
                                                               ,
                                                               ,
                                                               ,  more: + -
       REFERENCED SCHEMA    : ORDERPROC
       REFERENCED TABLE     : CUSTOMERS
       REFERENCED COLUMNS   : CUST_NUM                         ,
                                                               ,
                                                               ,
                                                               ,  more: + -
--------------------------------------------------------------------------------
 ===>:         F1=Help   F3=Terminate                   F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                         TAST
```

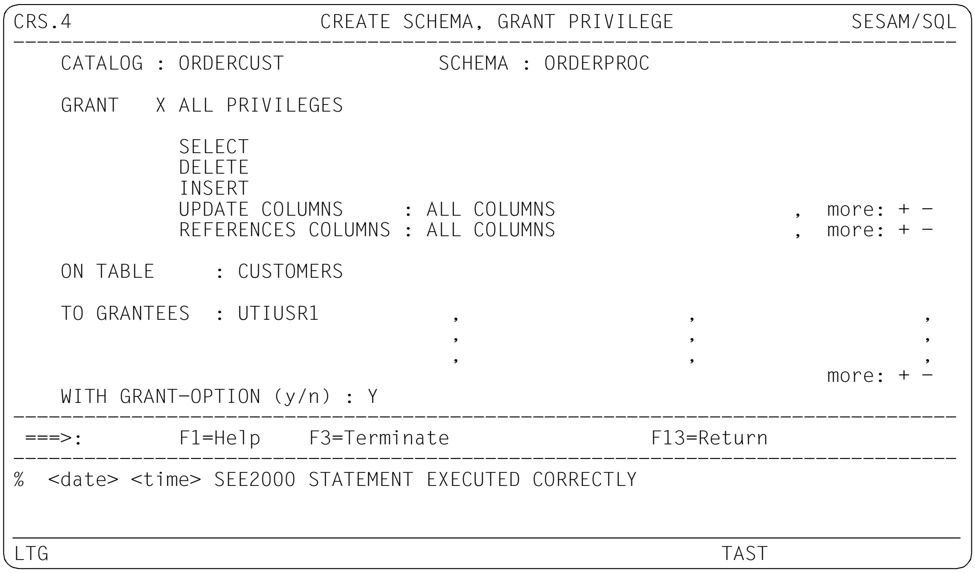
Enter F13 to return to the CRT.1 form. In this form, you again enter F13 to return to the CRT form. In the CRT form, you must select the default function 5,
Execute TABLE-DEFINITION, to create the defined CONTACTS table. The utility monitor confirms that your entries have been accepted.

You define and create the ORDERS, SERVICE and ORDER_STAT tables in the same way (see page 29).

Granting table privileges

Once you have created all the tables, you enter F13.
Return to the CRS form (see page 28), in which you select function 4 from the function menu. This displays the CRS.4 continuation form.

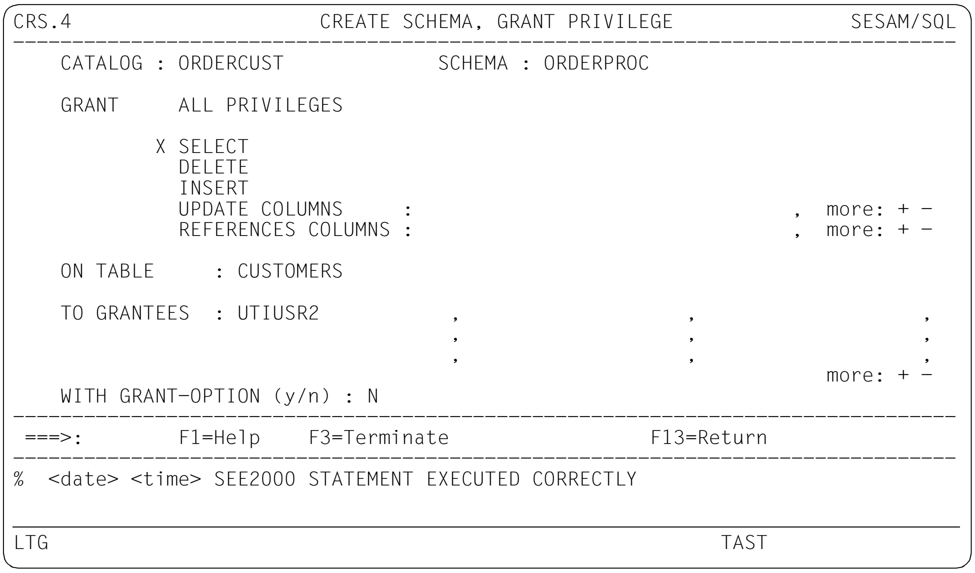
In the CRS.4 continuation form you grant table privileges. You grant the user UTIUSR1 all the table privileges for the CUSTOMERS table. The utility monitor confirms that your entries have been accepted.



```
CRS.4                      CREATE SCHEMA, GRANT PRIVILEGE                  SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST             SCHEMA : ORDERPROC

    GRANT   X ALL PRIVILEGES

              SELECT
              DELETE
              INSERT
              UPDATE COLUMNS     : ALL COLUMNS                         ,  more: + -
              REFERENCES COLUMNS : ALL COLUMNS                         ,  more: + -

    ON TABLE     : CUSTOMERS

    TO GRANTEES  : UTIUSR1          ,                   ,                   ,
                                    ,                   ,                   ,
                                    ,                   ,                   ,
                                                                      more: + -
    WITH GRANT-OPTION (y/n) : Y
--------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY


LTG                                                         TAST
```

You grant the privileges for the CONTACTS, ORDERS, SERVICE and ORDER_STAT tables in the same way.

You grant the user UTIUSR2 the SELECT table privilege for the CUSTOMERS table. The utility monitor confirms that your entries have been accepted.

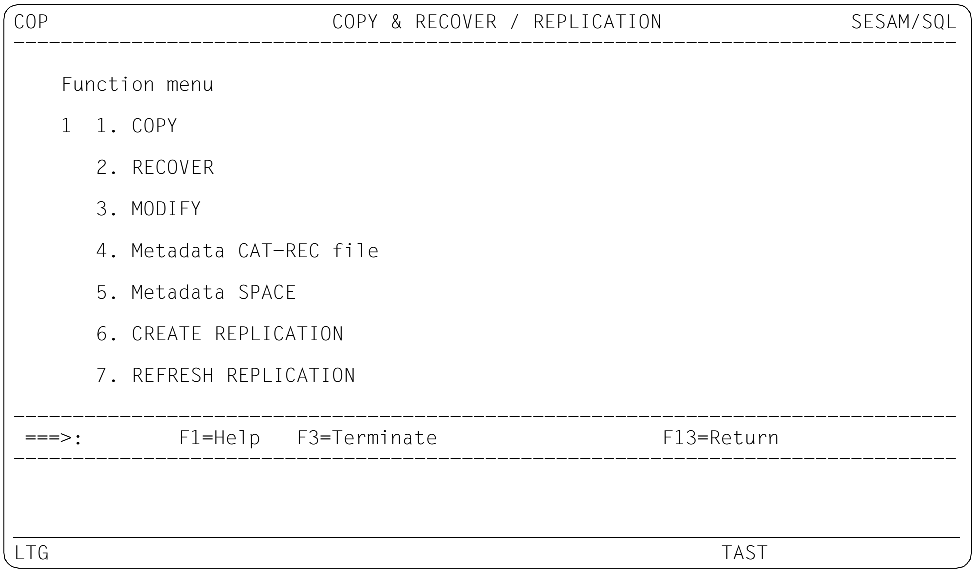


```
CRS.4                      CREATE SCHEMA, GRANT PRIVILEGE              SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST             SCHEMA : ORDERPROC

    GRANT     ALL PRIVILEGES

            X SELECT
              DELETE
              INSERT
              UPDATE COLUMNS     :                              ,  more: + -
              REFERENCES COLUMNS :                              ,  more: + -

    ON TABLE     : CUSTOMERS

    TO GRANTEES  : UTIUSR2          ,                   ,                   ,
                                    ,                   ,                   ,
                                    ,                   ,                   ,
                                                                   more: + -
    WITH GRANT-OPTION (y/n) : N
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

--------------------------------------------------------------------------------
LTG                                                         TAST
```

You grant the table privileges for the CONTACTS, ORDERS, SERVICE and ORDER_STAT tables in the same way.

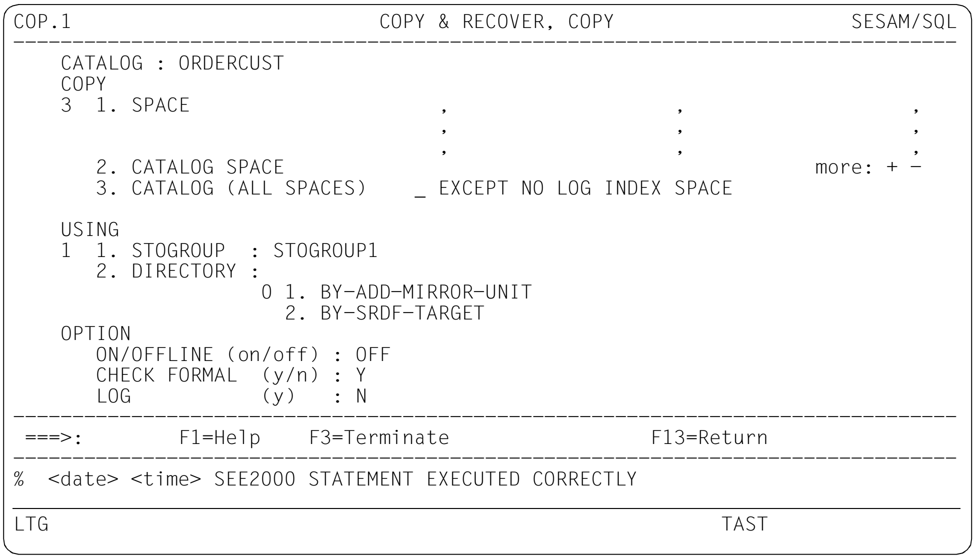
Performing backups

Enter “cop” in the command area to display the COP form.
By default, COPY is set to be function 1 in the function menu. You send the mask by pressing the DUE key. This displays the continuation form COP.1.



```
COP                          COPY & RECOVER / REPLICATION                 SESAM/SQL
-------------------------------------------------------------------------------

    Function menu

    1  1. COPY

       2. RECOVER

       3. MODIFY

       4. Metadata CAT-REC file

       5. Metadata SPACE

       6. CREATE REPLICATION

       7. REFRESH REPLICATION

-------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                      F13=Return
-------------------------------------------------------------------------------



LTG                                                         TAST
```

In the COP.1 form, choose function 3 from the function menu below COPY in order to back up the entire database. The utility monitor confirms that your entries have been accepted.



```
COP.1                            COPY & RECOVER, COPY                     SESAM/SQL
-------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    COPY
    3  1. SPACE                        ,                   ,                   ,
                                       ,                   ,                   ,
                                       ,                   ,                   ,
       2. CATALOG SPACE                                              more: + -
       3. CATALOG (ALL SPACES)     _ EXCEPT NO LOG INDEX SPACE

    USING
    1  1. STOGROUP  : STOGROUP1
       2. DIRECTORY :
                   0 1. BY-ADD-MIRROR-UNIT
                     2. BY-SRDF-TARGET
    OPTION
       ON/OFFLINE (on/off) : OFF
       CHECK FORMAL  (y/n) : Y
       LOG           (y)   : N
-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                    F13=Return
-------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                         TAST
```

Terminating database creation

Exit the utility monitor as described on page 18.

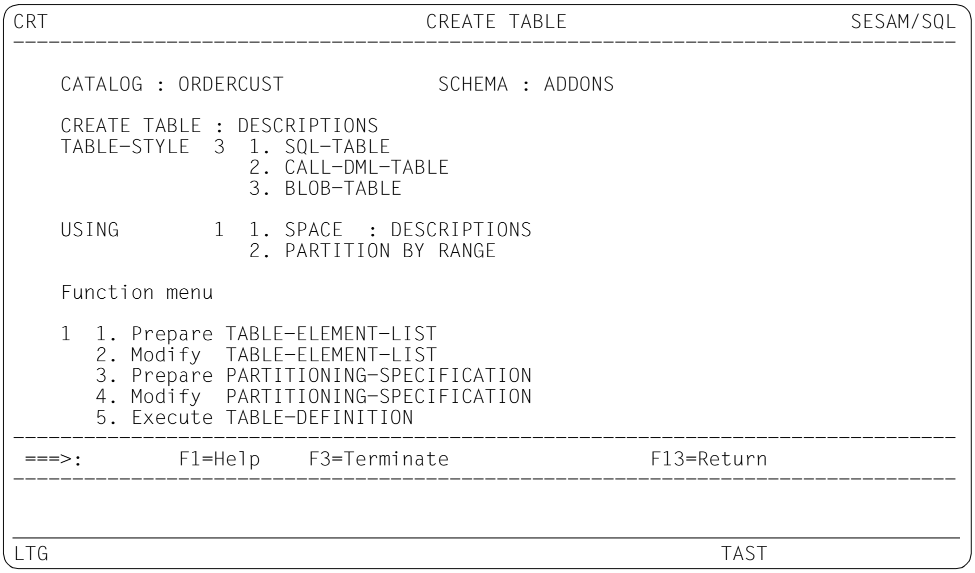
You have now finished creating the ORDERCUST database.

Defining a BLOB table

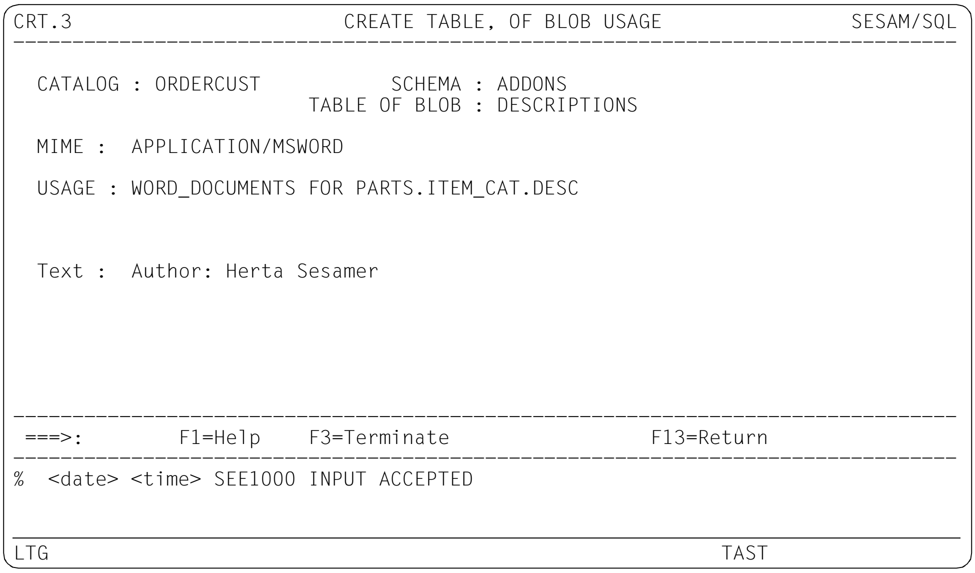
To follow the example application relating to BLOB tables, open the SSL form and create the user space DESCRIPTIONS as described in section “Creating a database” on page 19.

You next create the schema ADDONS in the CRS form as described in section “Creating a schema” on page 28.

You now open the CRT form in which you define the DESCRIPTIONS table (CREATE TABLE field). The table is to be created in the user space DESCRIPTIONS (USING SPACE field)
and is to be a BLOB table
(TABLE-STYLE field, value 3 BLOB-TABLE). In the function menu, select the default function „1, Prepare TABLE-ELEMENT-LIST.
This displays the CRT.3 continuation form.


```
CRT                                 CREATE TABLE                                SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST                 SCHEMA : ADDONS

    CREATE TABLE : DESCRIPTIONS
    TABLE-STYLE  3  1. SQL-TABLE
                    2. CALL-DML-TABLE
                    3. BLOB-TABLE

    USING        1  1. SPACE   : DESCRIPTIONS
                    2. PARTITION BY RANGE

    Function menu

    1  1. Prepare TABLE-ELEMENT-LIST
       2. Modify  TABLE-ELEMENT-LIST
       3. Prepare PARTITIONING-SPECIFICATION
       4. Modify  PARTITIONING-SPECIFICATION
       5. Execute TABLE-DEFINITION
--------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                 F13=Return
--------------------------------------------------------------------------------


LTG                                                            TAST
```

In form CRT.3 you can specify the MIME type and USAGE of the BLOB table and can enter a user-defined text. There are no restrictions on the format of this user-defined text. The DESCRIPTIONS table is intended to store BLOB objects of Word document class that can be edited using Microsoft Word.

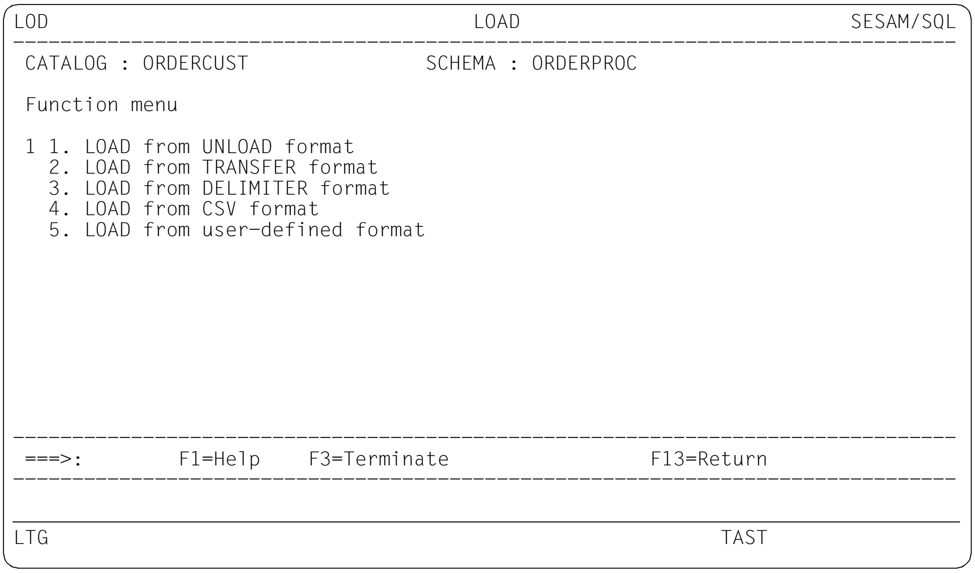


```
CRT.3                          CREATE TABLE, OF BLOB USAGE                   SESAM/SQL
--------------------------------------------------------------------------------

  CATALOG : ORDERCUST             SCHEMA : ADDONS
                           TABLE OF BLOB : DESCRIPTIONS

  MIME :  APPLICATION/MSWORD

  USAGE : WORD_DOCUMENTS FOR PARTS.ITEM_CAT.DESC


  Text :  Author: Herta Sesamer




--------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED

LTG                                                         TAST
```

Enter F13 to return to the CRT.1 form.
In this form, you again enter F13 to return to the CRT form. You now create the table as described in section “Creating a database” on page 36.

2.5 Loading user data

In the example below, data is to be loaded from an input file into the CUSTOMERS base table. Enter the function menu “05” in the STM start form (see page 13). This displays the LOD form.



```
LOD                                       LOAD                              SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                SCHEMA : ORDERPROC

 Function menu

 1 1. LOAD from UNLOAD format
   2. LOAD from TRANSFER format
   3. LOAD from DELIMITER format
   4. LOAD from CSV format
   5. LOAD from user-defined format










--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------

LTG                                                        TAST
```

The functions available in the LOD form depend on the data format of the input file. The procedure for input files in delimiter format and user-defined format is described below.

2.5.1 Loading user data in delimiter format

The data is to be loaded in the OFFLINE mode from the input file SES.CUSTOMER.DEL into the CUSTOMERS base table.

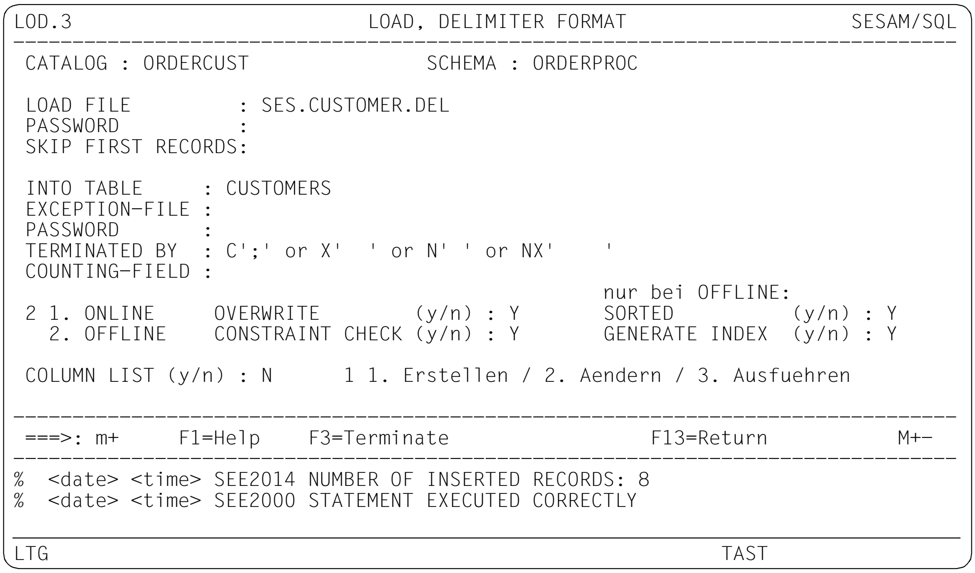
The input file SES.CUSTOMER.DEL contains the following entries:

```
100;Siemens AG;Otto-Hahn-Ring 6;81739;Munich;D;089/636-8;Electrical
101;Login GmbH;Rosenheimer Str.34;81667;Munich;D;089/4488870;PC Networks
102;JIKO GmbH;Posener Str. 12;30659;Hanover;D;0551/123874;Import/Export
103;Plenzer Trading;Paul-Heyse-Str. 12;80336;Munich;D;089/923764;Fruit market
____Freddy's Fishery                  Hirschgartenstr. 12
12;12587;Berlin;D;016/5739921;Unit retail;
105;The Poodle Parlor;Am Muehlentor 26;41179;Moenchengladbach;D;040/873562;Service
106;Foreign Ltd.;26 West York St.;;New York, NY;USA;001703/2386532;Commercial agency
107;Externa & Co KG;Berner Weg 78;03000;Bern 33;CH;;Lawyer
```

Defining the format

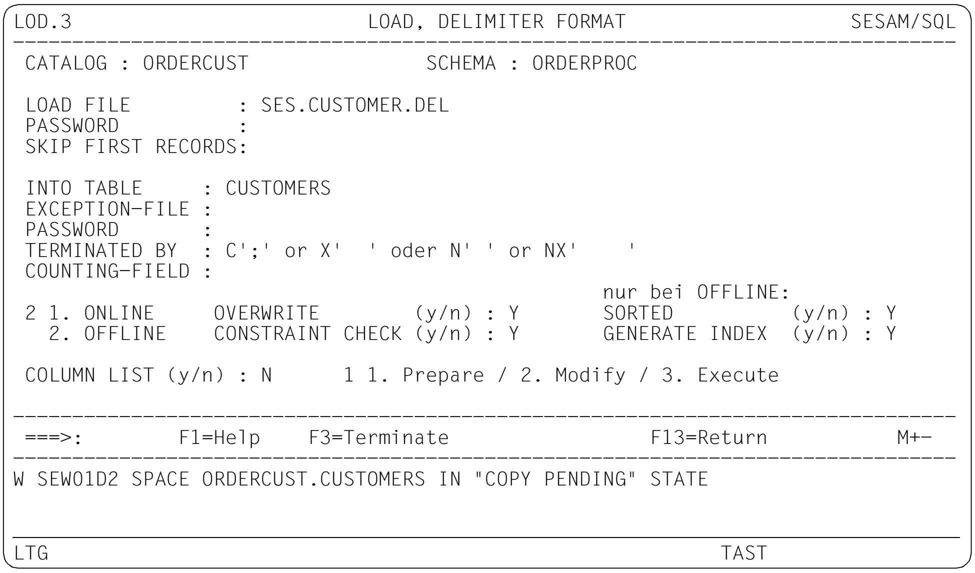
Select “3” (LOAD from DELIMITER Format) in the function menu of the LOD form (see page 44). You then branch to the continuation form LOD.3.

There you enter the name of the input file (LOAD FILE), if required the number of header lines to be skipped in the file, and the name of the base table (INTO TABLE).
Specify that the values are separated by the DELIMITER character “;” (TERMINATED BY).
Enter “2” (OFFLINE) for the operating mode. For the parameters OVERWRITE, SORTED, CHECK CONSTRAINT and GENERATE INDEX you enter “Y”. Press the DUE key to send off the form.



```
LOD.3                           LOAD, DELIMITER FORMAT                          SESAM/SQL
-------------------------------------------------------------------------------
 CATALOG : ORDERCUST                SCHEMA : ORDERPROC

 LOAD FILE         : SES.CUSTOMER.DEL
 PASSWORD          :
 SKIP FIRST RECORDS:

 INTO TABLE     : CUSTOMERS
 EXCEPTION-FILE :
 PASSWORD       :
 TERMINATED BY  : C';' or X'  ' or N' ' or NX'    '
 COUNTING-FIELD :
                                                   nur bei OFFLINE:
 2 1. ONLINE     OVERWRITE        (y/n) : Y        SORTED         (y/n) : Y
   2. OFFLINE    CONSTRAINT CHECK (y/n) : Y        GENERATE INDEX  (y/n) : Y

 COLUMN LIST (y/n) : N      1 1. Erstellen / 2. Aendern / 3. Ausfuehren

-------------------------------------------------------------------------------
 ===>: m+      F1=Help    F3=Terminate                F13=Return          M+-
-------------------------------------------------------------------------------
%  <date> <time> SEE2014 NUMBER OF INSERTED RECORDS: 8
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                          TAST
```

The number of records that are loaded into the table is displayed in the message area. The utility monitor confirms that the LOAD statement has been performed successfully. The display “M+-” in the command area indicates that further utility monitor or SQLSTATE messages are present. Enter m+ or m- in the command area to scroll through the message area.

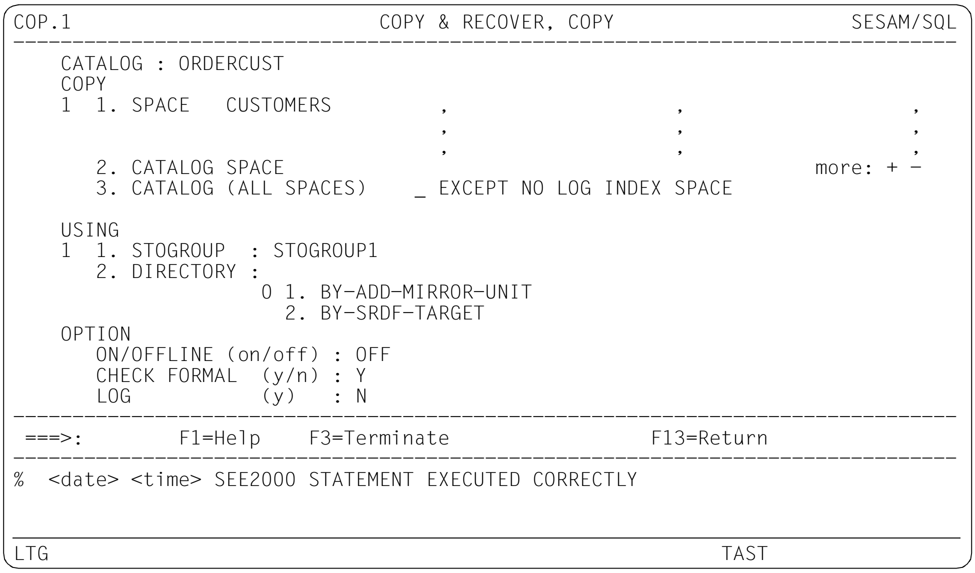


```
LOD.3                         LOAD, DELIMITER FORMAT                    SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST               SCHEMA : ORDERPROC

 LOAD FILE         : SES.CUSTOMER.DEL
 PASSWORD          :
 SKIP FIRST RECORDS:

 INTO TABLE     : CUSTOMERS
 EXCEPTION-FILE :
 PASSWORD       :
 TERMINATED BY  : C';' or X'  ' oder N' ' or NX'    '
 COUNTING-FIELD :
                                                  nur bei OFFLINE:
 2 1. ONLINE     OVERWRITE        (y/n) : Y       SORTED          (y/n) : Y
   2. OFFLINE    CONSTRAINT CHECK (y/n) : Y       GENERATE INDEX  (y/n) : Y

 COLUMN LIST (y/n) : N      1 1. Prepare / 2. Modify / 3. Execute

--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return          M+-
--------------------------------------------------------------------------------
W SEW01D2 SPACE ORDERCUST.CUSTOMERS IN "COPY PENDING" STATE


LTG                                                            TAST
```

The user space ORDERCUST.CUSTOMERS has now the state “copy pending”.

Backing up a user space

Enter cop in the command area to display the COP form (see page 41), which you can send by pressing the DUE key. This opens the COP.1 form.


```
COP.1                          COPY & RECOVER, COPY                    SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    COPY
    1  1. SPACE   CUSTOMERS          ,                   ,                   ,
                                     ,                   ,                   ,
                                     ,                   ,                   ,
       2. CATALOG SPACE                                               more: + -
       3. CATALOG (ALL SPACES)     _ EXCEPT NO LOG INDEX SPACE

    USING
    1  1. STOGROUP  : STOGROUP1
       2. DIRECTORY :
                     0 1. BY-ADD-MIRROR-UNIT
                       2. BY-SRDF-TARGET
    OPTION
       ON/OFFLINE (on/off) : OFF
       CHECK FORMAL  (y/n) : Y
       LOG           (y)   : N
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY


LTG                                                        TAST
```

In the COP.1 form, choose function 1, SPACE, and enter the name of the user space in the state “copy pending”. Send the COP.1 form by pressing the DUE key. The utility monitor confirms that your entries have been executed correctly.

Keep entering F13 until you get back to the previous form.

2.5.2 Loading user data in user-defined format

You want to load the data in the OFFLINE mode from the input file LOAD.U.CUSTOMERS into the CUSTOMERS base table.

The input file LOAD.U.CUSTOMERS contains the following entries:

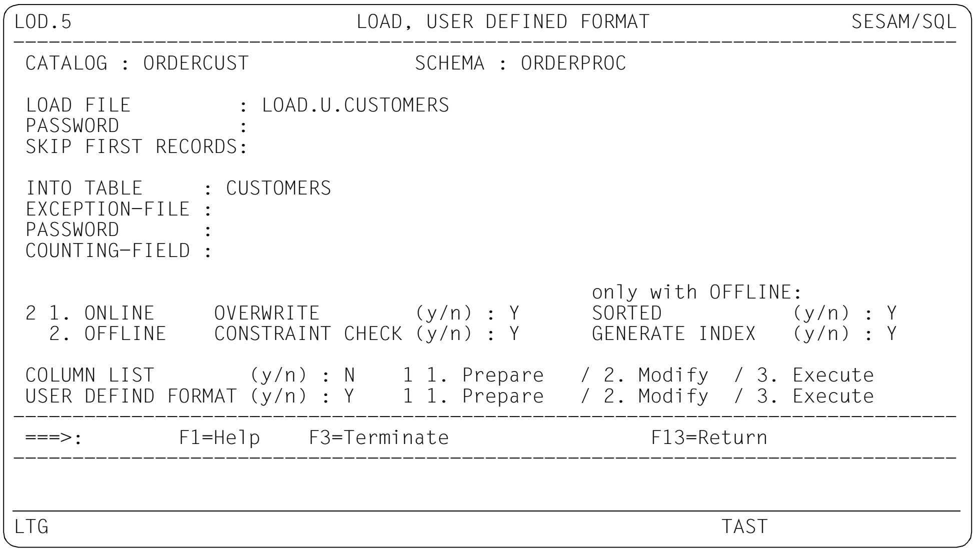
```
____Siemens AG                     Otto-Hahn-Ring 6
81739Munich               D  089/636-8              Electrical
____Login GmbH                     Rosenheimer Str.34
81667Munich               D  089/4488870             PC Networks
____JIKO GmbH                      Posener Str. 12
30659Hanover               D  0551/123874             Import-Export
____Plenzer Trading                Paul-Heyse-Str. 12
80336Munich               D  089/923764              Fruit market
____Freddy's Fishery               Hirschgartenstr. 12 12
12587Berlin                 D  016/5739921             Unit retail
____The Poodle Parlor                 Am Muehlentor 26
41179Moenchengladbach       D  040/873562              Service
____Foreign Ltd.                   26 West York  St.
00000New York, NY           USA001703/2386532          Commercial agency
____Externa & Co KG                Berner Weg 78
03000Bern 33                CH                         Lawyer
```

The primary keys are of the type INTEGER and must be entered in the input file in binary form. In the above example, they are represented by the string “____”.

Defining the format

Select “5” (LOAD from user-defined format) in the function menu of the LOD form (see page 44). You then branch to the continuation form LOD.5.

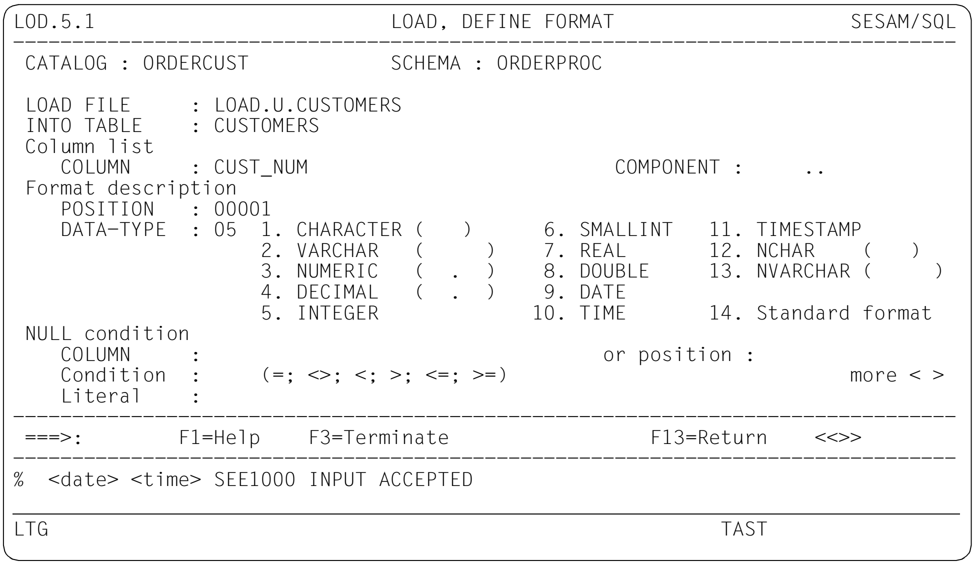
There you enter the name of the input file and the base table.
Enter “2” (OFFLINE) for the operating mode. For the parameters OVERWRITE, SORTED, CHECK CONSTRAINT and GENERATE INDEX you enter “Y”.
Enter “Y” and “1” (Prepare) in the USER DEFINED FORMAT input field.
Press the DUE key to send off the form.



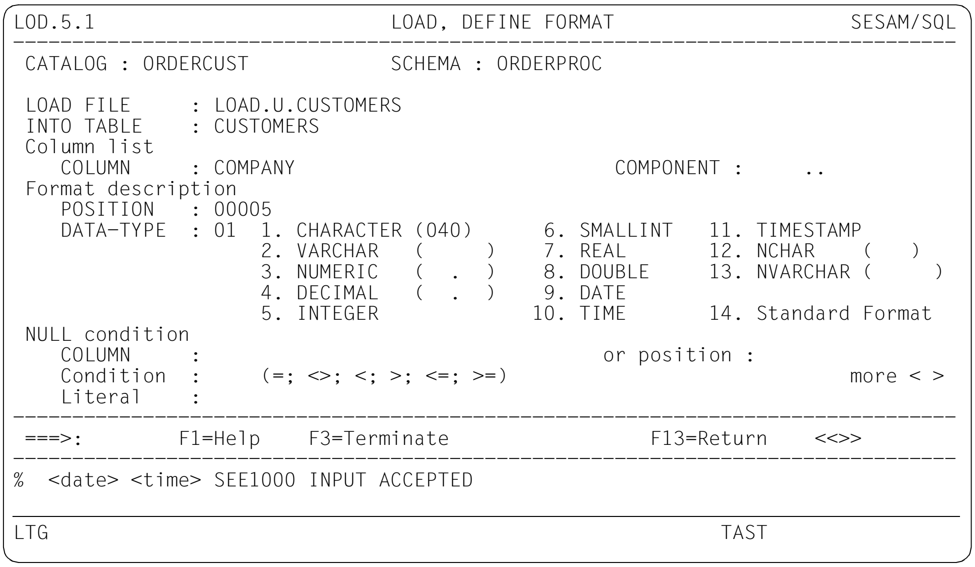
```
LOD.5                         LOAD, USER DEFINED FORMAT                 SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST              SCHEMA : ORDERPROC

 LOAD FILE          : LOAD.U.CUSTOMERS
 PASSWORD           :
 SKIP FIRST RECORDS:

 INTO TABLE      : CUSTOMERS
 EXCEPTION-FILE :
 PASSWORD        :
 COUNTING-FIELD :

                                                 only with OFFLINE:
 2 1. ONLINE     OVERWRITE        (y/n) : Y      SORTED           (y/n) : Y
   2. OFFLINE    CONSTRAINT CHECK (y/n) : Y      GENERATE INDEX   (y/n) : Y

 COLUMN LIST        (y/n) : N    1 1. Prepare   / 2. Modify   / 3. Execute
 USER DEFIND FORMAT (y/n) : Y    1 1. Prepare   / 2. Modify   / 3. Execute
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
--------------------------------------------------------------------------------


LTG                                                         TAST
```

You branch to the continuation form LOD.5.1.

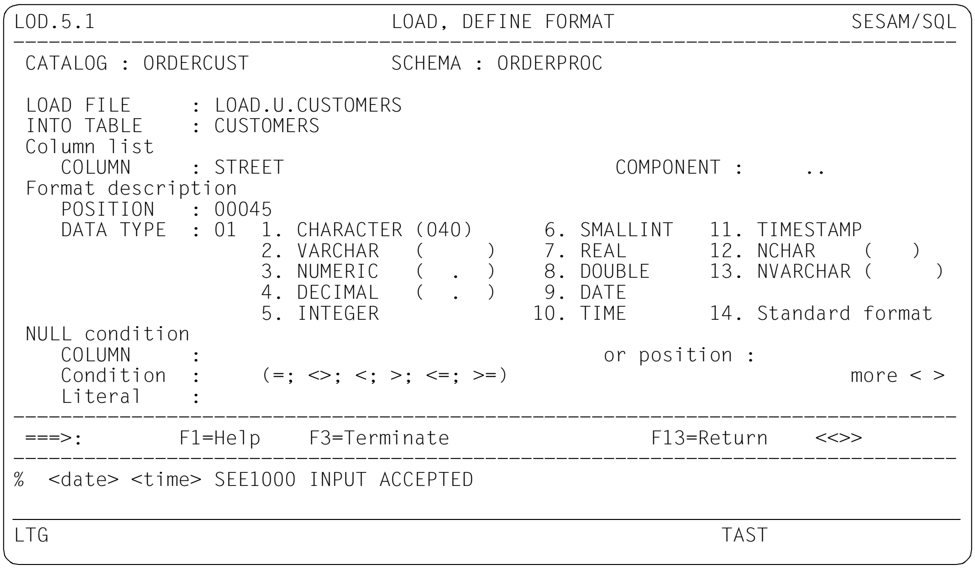
In the LOD.5.1 form, you specify the location of the data in the associated input file for each individual column in the table. The definitions declared when the database was set up apply to the columns.
You send the forms by pressing the DUE key. The utility monitor confirms that your entries have been accepted. By entering << and >> in the command area you can page through entries you have already sent off.



```
LOD.5.1                         LOAD, DEFINE FORMAT                        SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : CUST_NUM                         COMPONENT :     ..
 Format description
    POSITION   : 00001
    DATA-TYPE  : 05  1. CHARACTER (    )      6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (      )    7. REAL       12. NCHAR    (    )
                     3. NUMERIC   (  .   )    8. DOUBLE     13. NVARCHAR (      )
                     4. DECIMAL   (  .   )    9. DATE
                     5. INTEGER              10. TIME       14. Standard format
 NULL condition
    COLUMN     :                             or position :
    Condition  :     (=; <>; <; >; <=; >=)                           more < >
    Literal    :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED

LTG                                                         TAST
```



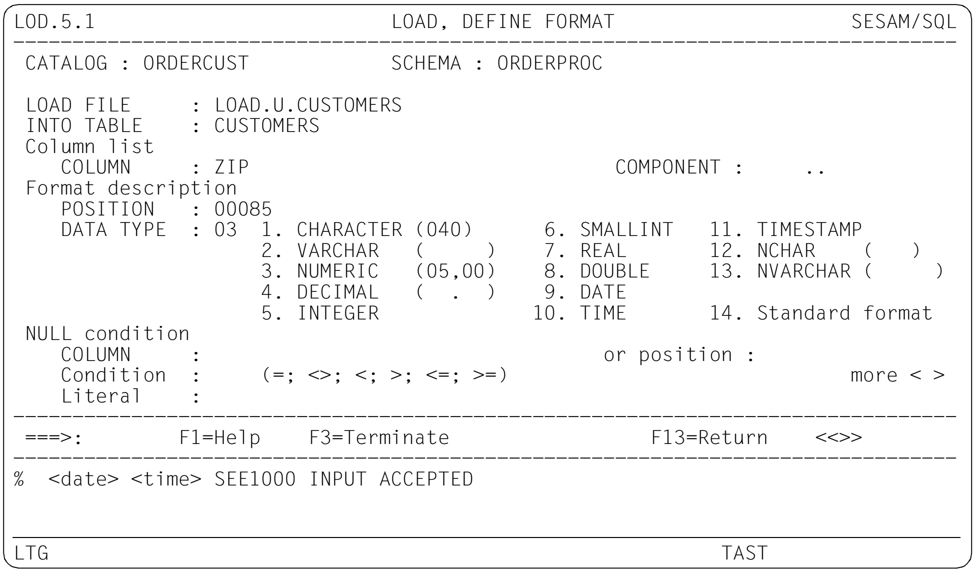
```
LOD.5.1                         LOAD, DEFINE FORMAT                        SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : COMPANY                          COMPONENT :     ..
 Format description
    POSITION   : 00005
    DATA-TYPE  : 01  1. CHARACTER (040)       6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (      )    7. REAL       12. NCHAR    (    )
                     3. NUMERIC   (  .   )    8. DOUBLE     13. NVARCHAR (      )
                     4. DECIMAL   (  .   )    9. DATE
                     5. INTEGER              10. TIME       14. Standard Format
 NULL condition
    COLUMN     :                             or position :
    Condition  :     (=; <>; <; >; <=; >=)                           more < >
    Literal    :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED

LTG                                                         TAST
```



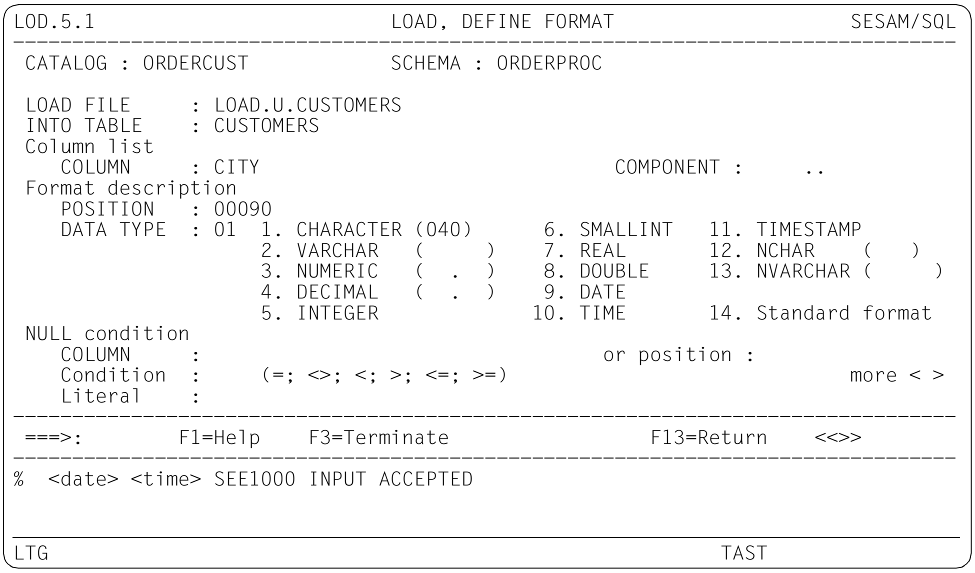
```
LOD.5.1                       LOAD, DEFINE FORMAT                       SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : STREET                           COMPONENT :     ..
 Format description
    POSITION   : 00045
    DATA TYPE  : 01  1. CHARACTER (040)      6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (     )    7. REAL       12. NCHAR    (   )
                     3. NUMERIC   (  .  )    8. DOUBLE     13. NVARCHAR (     )
                     4. DECIMAL   (  .  )    9. DATE
                     5. INTEGER             10. TIME       14. Standard format
 NULL condition
    COLUMN     :                                 or position :
    Condition  :     (=; <>; <; >; <=; >=)                            more < >
    Literal    :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED

LTG                                                          TAST
```

Defining a NULL constraint

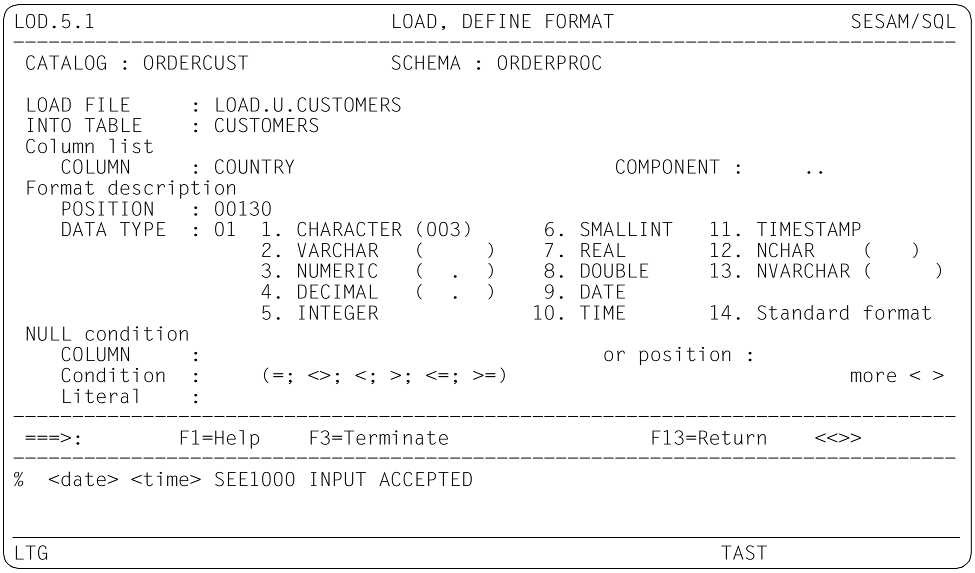
You define a NULL constraint for the ZIP column.



```
LOD.5.1                       LOAD, DEFINE FORMAT                       SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : ZIP                              COMPONENT :     ..
 Format description
    POSITION   : 00085
    DATA TYPE  : 03  1. CHARACTER (040)      6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (     )    7. REAL       12. NCHAR    (   )
                     3. NUMERIC   (05,00)    8. DOUBLE     13. NVARCHAR (     )
                     4. DECIMAL   (  .  )    9. DATE
                     5. INTEGER             10. TIME       14. Standard format
 NULL condition
    COLUMN     :                                 or position :
    Condition  :     (=; <>; <; >; <=; >=)                            more < >
    Literal    :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                          TAST
```

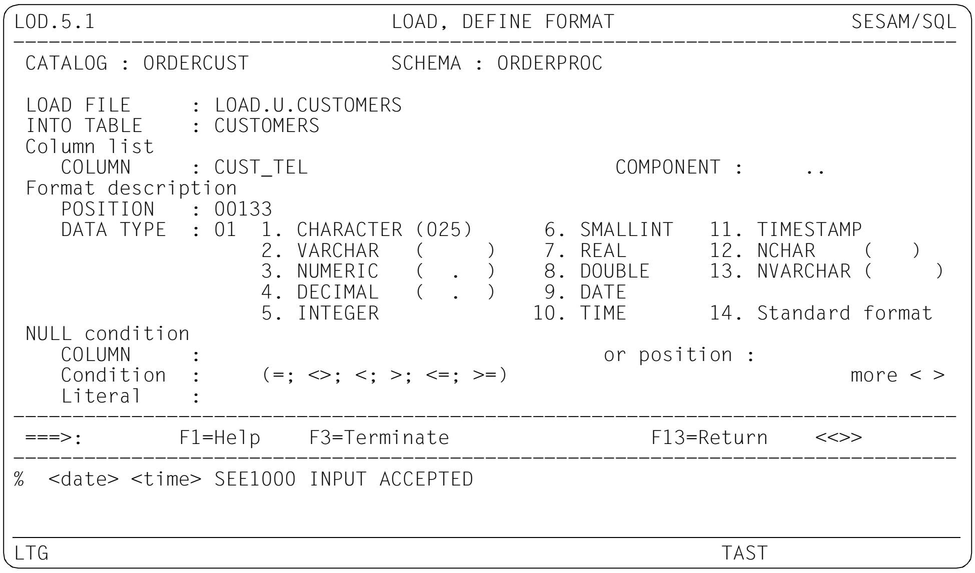


```
LOD.5.1                         LOAD, DEFINE FORMAT                      SESAM/SQL
-----------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : CITY                                 COMPONENT :     ..
 Format description
    POSITION   : 00090
    DATA TYPE  : 01  1. CHARACTER (040)      6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (     )    7. REAL       12. NCHAR    (   )
                     3. NUMERIC   (  .  )    8. DOUBLE     13. NVARCHAR (     )
                     4. DECIMAL   (  .  )    9. DATE
                     5. INTEGER             10. TIME       14. Standard format
 NULL condition
    COLUMN     :                                     or position :
    Condition  :     (=; <>; <; >; <=; >=)                                more < >
    Literal    :
-----------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
-----------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                          TAST
```

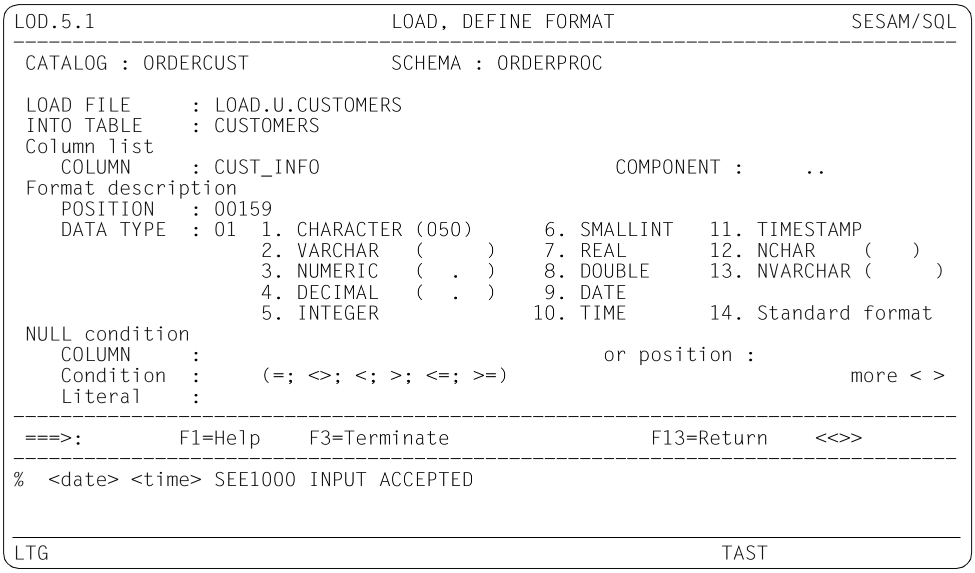


```
LOD.5.1                         LOAD, DEFINE FORMAT                      SESAM/SQL
-----------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : COUNTRY                              COMPONENT :     ..
 Format description
    POSITION   : 00130
    DATA TYPE  : 01  1. CHARACTER (003)      6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (     )    7. REAL       12. NCHAR    (   )
                     3. NUMERIC   (  .  )    8. DOUBLE     13. NVARCHAR (     )
                     4. DECIMAL   (  .  )    9. DATE
                     5. INTEGER             10. TIME       14. Standard format
 NULL condition
    COLUMN     :                                     or position :
    Condition  :     (=; <>; <; >; <=; >=)                                more < >
    Literal    :
-----------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return    <<>>
-----------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                          TAST
```

You then define a NULL constraint for the CUST_TEL column.



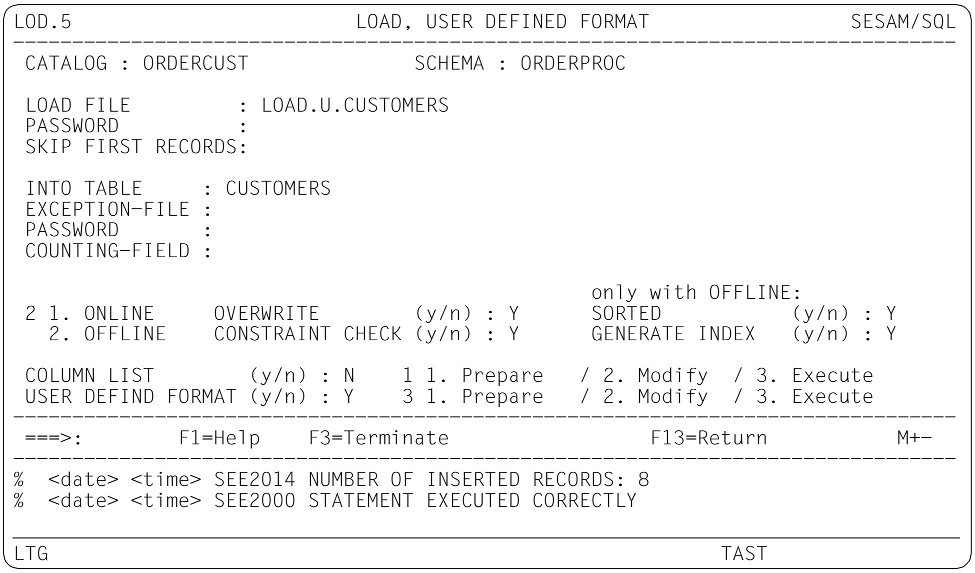
```
LOD.5.1                        LOAD, DEFINE FORMAT                      SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : CUST_TEL                            COMPONENT :     ..
 Format description
    POSITION   : 00133
    DATA TYPE  : 01  1. CHARACTER (025)      6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (     )    7. REAL       12. NCHAR    (    )
                     3. NUMERIC   (  .  )    8. DOUBLE     13. NVARCHAR (      )
                     4. DECIMAL   (  .  )    9. DATE
                     5. INTEGER             10. TIME       14. Standard format
 NULL condition
    COLUMN     :                                    or position :
    Condition  :     (=; <>; <; >; <=; >=)                                more < >
    Literal    :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                            TAST
```



```
LOD.5.1                        LOAD, DEFINE FORMAT                      SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST            SCHEMA : ORDERPROC

 LOAD FILE     : LOAD.U.CUSTOMERS
 INTO TABLE    : CUSTOMERS
 Column list
    COLUMN     : CUST_INFO                           COMPONENT :     ..
 Format description
    POSITION   : 00159
    DATA TYPE  : 01  1. CHARACTER (050)      6. SMALLINT   11. TIMESTAMP
                     2. VARCHAR   (     )    7. REAL       12. NCHAR    (    )
                     3. NUMERIC   (  .  )    8. DOUBLE     13. NVARCHAR (      )
                     4. DECIMAL   (  .  )    9. DATE
                     5. INTEGER             10. TIME       14. Standard format
 NULL condition
    COLUMN     :                                    or position :
    Condition  :     (=; <>; <; >; <=; >=)                                more < >
    Literal    :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return    <<>>
--------------------------------------------------------------------------------
%  <date> <time> SEE1000 INPUT ACCEPTED


LTG                                                            TAST
```

Loading records into the base table

You enter F13 to return to the LOD.5 form.
“3” (Execute) is now set in the USER DEFINED FORMAT input field, to load the records into the base table in accordance with the defined format. If you enter “2”, you can change the defined columns again.



```
LOD.5                         LOAD, USER DEFINED FORMAT                 SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST              SCHEMA : ORDERPROC

 LOAD FILE         : LOAD.U.CUSTOMERS
 PASSWORD          :
 SKIP FIRST RECORDS:

 INTO TABLE     : CUSTOMERS
 EXCEPTION-FILE :
 PASSWORD       :
 COUNTING-FIELD :

                                                only with OFFLINE:
 2 1. ONLINE     OVERWRITE        (y/n) : Y      SORTED            (y/n) : Y
   2. OFFLINE    CONSTRAINT CHECK (y/n) : Y      GENERATE INDEX    (y/n) : Y

 COLUMN LIST         (y/n) : N    1 1. Prepare   / 2. Modify  / 3. Execute
 USER DEFIND FORMAT (y/n) : Y    3 1. Prepare   / 2. Modify  / 3. Execute
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return          M+-
--------------------------------------------------------------------------------
%  <date> <time> SEE2014 NUMBER OF INSERTED RECORDS: 8
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                        TAST
```

The utility monitor confirms that the LOAD statement has been executed successfully but indicates that the ORDERCUST.CUSTOMERS user space has the “copy pending” state.

Perform a backup of the user space as described on page 47.

2.6 Creating and restoring a tape backup

In section “Creating a tape backup of a database”, the ORDERCUST database is to be backed up on the magnetic tape cartridge MTC001. The BS2000 software product HSMS is used for backup and restore operations. For information on setting up an HSMS archive and HSMS parameter file, refer to “SQL Reference Manual Part 2: Utilities”.

In section “Restoring tape backups of a database or a space” on page 58 you will find a description of how to restore the created tape backup.

2.6.1 Creating a tape backup of a database

Before you back up the database on magnetic tape cartridge, you have to:

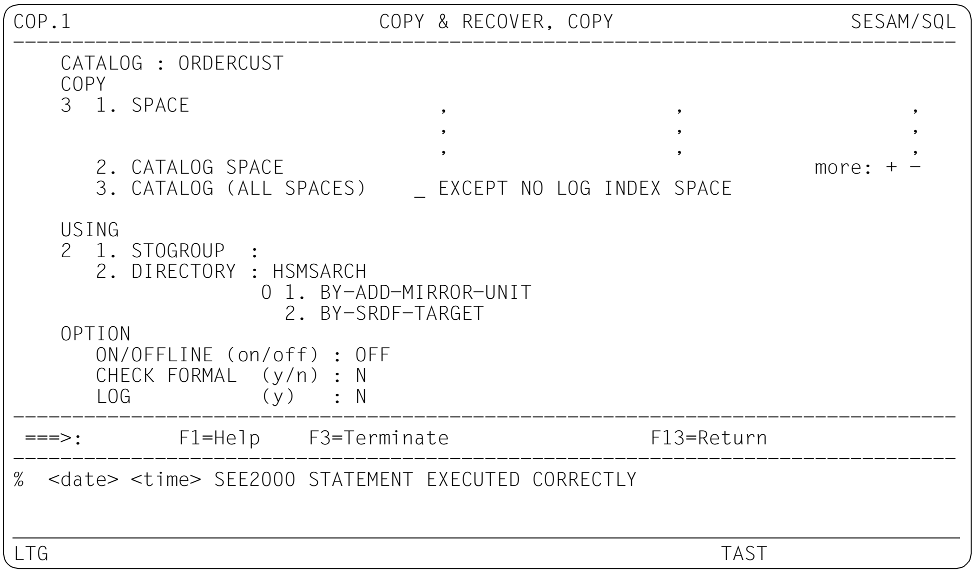
- create an HSMS archive under the DBH user ID
- create an HSMS parameter file under the DBH user ID

For details on using HSMS for tape backups, refer to “SQL Reference Manual Part 2: Utilities” and the “HSMS (BS2000)” manual.

Backing up the database on a magnetic tape cartridge

When you start the utility monitor, the STM start form appears (see page 13). Select function 10 in the function menu.
This opens the COP form (see page 41). By default, COPY is set to be function 1 in the function menu. You send the mask by pressing the DUE key. This displays the continuation form COP.1.

In the COP.1 form, enter 3 in the function menu under COPY. In the function menu under USING, you enter 2 and the name of the existing HSMS archive.
The utility monitor confirms that the backup has been executed successfully.



```
COP.1                          COPY & RECOVER, COPY                    SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    COPY
    3  1. SPACE                     ,                   ,                   ,
                                    ,                   ,                   ,
                                    ,                   ,                   ,
       2. CATALOG SPACE                                           more: + -
       3. CATALOG (ALL SPACES)     _ EXCEPT NO LOG INDEX SPACE

    USING
    2  1. STOGROUP   :
       2. DIRECTORY : HSMSARCH
                    0 1. BY-ADD-MIRROR-UNIT
                      2. BY-SRDF-TARGET
    OPTION
       ON/OFFLINE (on/off) : OFF
       CHECK FORMAL  (y/n) : N
       LOG           (y)   : N
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY


LTG                                                           TAST
```

Exit the utility monitor as described on page 18.

Querying the HSMS archive

You can now call HSMS and use the HSMS statement SHOW-ARCHIVE to list the files backed up in the HSMS archive.

```
/START-HSMS
%  HSMLOAD Program 'HSMS', version '<version>' of '<date>' LOADED from file
':2OSG:$TSOS.SYSLNK.HSMS.<version>'
%//SHOW-ARCHIVE ARCHIVE-NAME=HSMSARCH, SELECT=*FILES

















________________________________________________________________________________
LTG                                                         TAST
```

```
SHOW-ARCHIVE (FILES)                      SHOW-FILE-VERSIONS = DIFFERENT
ENVIRONMENT     = SM(2OSG)                ARCHIVE-NAME = *$ID1.HSMSARCH
CATALOG-ID      = 2OSG                    SV-NAME  = ANY
USER-ID         = ID1                     SV-DATE  = INTERVAL  EARLIEST LATEST
FILE-SAVE-STATE = ANY                     EXP-DATE = ANY
INFORMATION     = SUMMARY
-------------------------------------------------------------------------------
M FILE-NAME                                 VERS SAV-DATE SAV-TIME EXP-DATE TYPE
  ORDERCUST.CATALOG                         1     <date>    <time>  <date>  FULL
  ORDERCUST.CONTACTS                        1     <date>    <time>  <date>  FULL
  ORDERCUST.CUSTOMERS                       1     <date>    <time>  <date>  FULL
  ORDERCUST.ORDERS                          1     <date>    <time>  <date>  FULL
  ORDERCUST.ORDSTAT                         1     <date>    <time>  <date>  FULL
  ORDERCUST.SERVICE                         1     <date>    <time>  <date>  FULL







-------------------------------------------------------------------------------
%  HSM0012 END OF OUTPUT LIST REACHED


________________________________________________________________________________
LTG                                                         TAST
```

After you have performed the backup, further changes can be made to the ORDERCUST database and utility statements can be entered.

2.6.2 Restoring tape backups of a database or a space

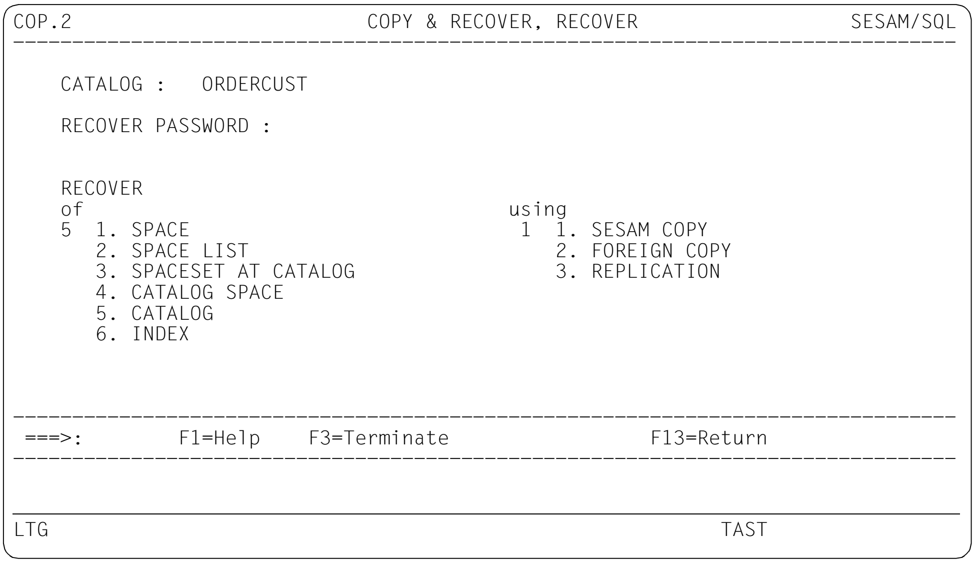
Assume that the system crashes as a result of a power failure. Usually a database is recovered on the restart that follows the failure.

Below you will find a description of how to recover a database or space using RECOVER if the restart fails or a disk is defective. To do this, you use the backup on the magnetic tape cartridge MTC001 and then apply the modifications logged in the log files.

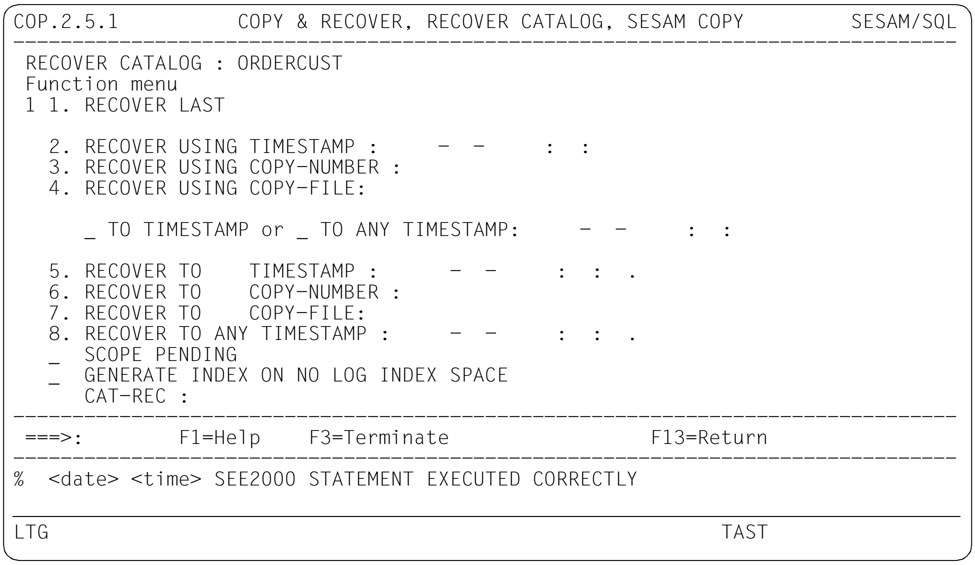
Restoring a tape backup of a database

When you start the utility monitor, the STM start form appears (see page 13). The database is to be recovered using COPY & RECOVER.
Enter “cop” in the command area to display the COP form (see page 41).

In the COP form, choose 2 for RECOVER in the function menu. This displays the COP.2 form.
In the COP.2 form, choose 5 under “RECOVER of” and choose 1 under “using” in order to recover the entire database from a SESAM backup. This displays the COP.2.5.1 form.


```
COP.2                          COPY & RECOVER, RECOVER                    SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG :   ORDERCUST

    RECOVER PASSWORD :


    RECOVER
    of                                   using
    5  1. SPACE                           1  1. SESAM COPY
       2. SPACE LIST                         2. FOREIGN COPY
       3. SPACESET AT CATALOG                3. REPLICATION
       4. CATALOG SPACE
       5. CATALOG
       6. INDEX



--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------


LTG                                                         TAST
```

In the COP.2.5.1 choose 1 from the function menu to use the most recently created backup. The utility monitor confirms that the recovery has been executed successfully.



```
COP.2.5.1          COPY & RECOVER, RECOVER CATALOG, SESAM COPY        SESAM/SQL
-------------------------------------------------------------------------------
 RECOVER CATALOG : ORDERCUST
 Function menu
 1 1. RECOVER LAST

   2. RECOVER USING TIMESTAMP :     -  -     :  :
   3. RECOVER USING COPY-NUMBER :
   4. RECOVER USING COPY-FILE:

      _ TO TIMESTAMP or _ TO ANY TIMESTAMP:     -  -     :  :

   5. RECOVER TO    TIMESTAMP :      -  -     :  :  .
   6. RECOVER TO    COPY-NUMBER :
   7. RECOVER TO    COPY-FILE:
   8. RECOVER TO ANY TIMESTAMP :     -  -     :  :  .
   _  SCOPE PENDING
   _  GENERATE INDEX ON NO LOG INDEX SPACE
      CAT-REC :
-------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY

LTG                                                          TAST
```

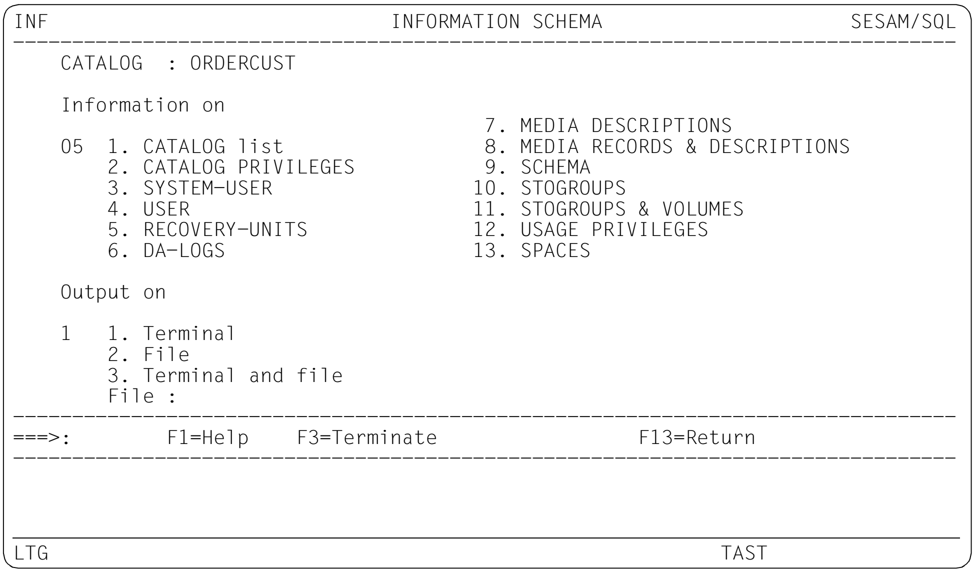
Exit the utility monitor as described on page 18.

Restoring a tape backup of a space

When you start the utility monitor, the STM start form appears (see page 13).

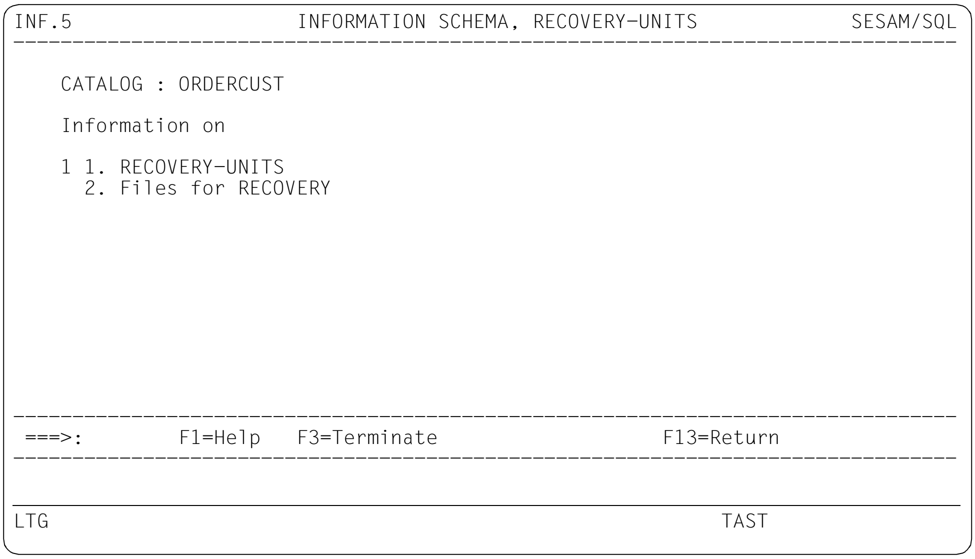
If you want to use a specific backup to perform recovery and the catalog space is intact, you can use the information schema INFORMATION_SCHEMA to provide you with information on the backups of the ORDERCUST database.
In the function menu of the STM start form, you enter 09
This displays the INF form.

In the INF form, you enter 05 under “Information on” to obtain information on RECOVERY_UNITS. The INF.5 form is now displayed.



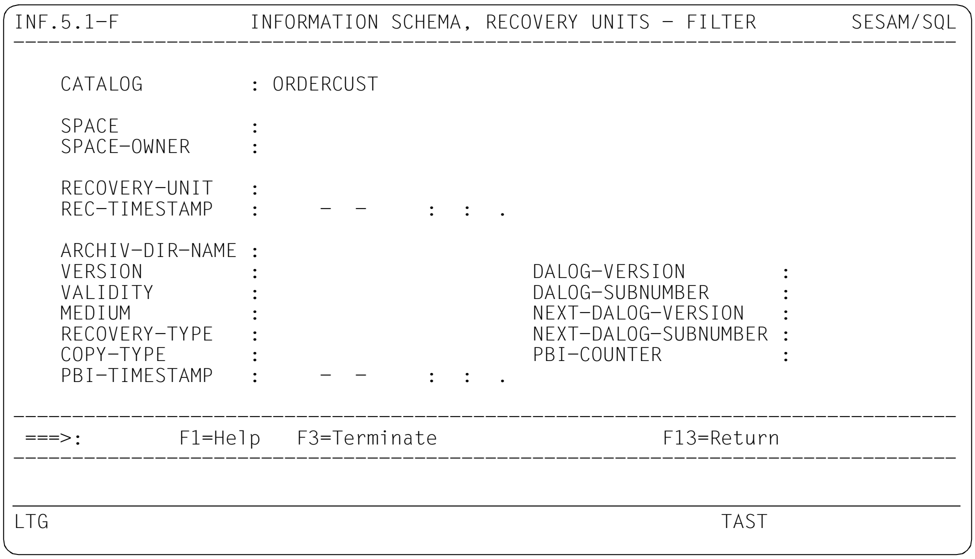
```
INF                                INFORMATION SCHEMA                          SESAM/SQL
-----------------------------------------------------------------------------------------
    CATALOG  : ORDERCUST

    Information on
                                        7. MEDIA DESCRIPTIONS
    05  1. CATALOG list                 8. MEDIA RECORDS & DESCRIPTIONS
        2. CATALOG PRIVILEGES           9. SCHEMA
        3. SYSTEM-USER                 10. STOGROUPS
        4. USER                        11. STOGROUPS & VOLUMES
        5. RECOVERY-UNITS              12. USAGE PRIVILEGES
        6. DA-LOGS                     13. SPACES

    Output on

    1   1. Terminal
        2. File
        3. Terminal and file
        File :
-----------------------------------------------------------------------------------------
===>:        F1=Help    F3=Terminate                     F13=Return
-----------------------------------------------------------------------------------------



LTG                                                                   TAST
```

In the INF.5 form, you select function 1. In the continuation form INF 5.1 (RECOVERY-UNITS) you can branch to the selection form INF.5.1-F.



```
INF.5                   INFORMATION SCHEMA, RECOVERY-UNITS               SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    Information on

    1 1. RECOVERY-UNITS
      2. Files for RECOVERY









--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------


LTG                                                         TAST
```

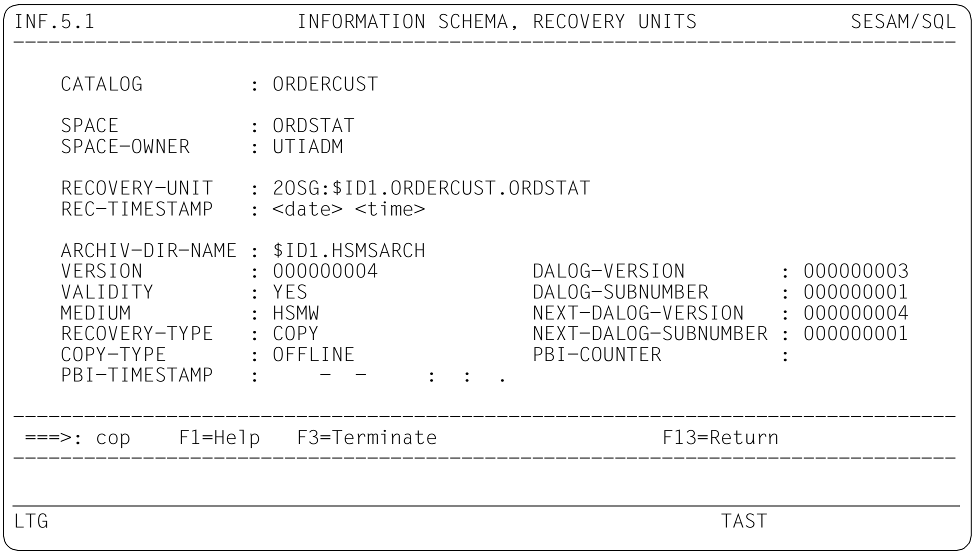
You send off the INF.5.1-F selection form by pressing the DUE key (see also section "Specifying the extent of the output – selection forms" on page 248).



```
INF.5.1-F           INFORMATION SCHEMA, RECOVERY UNITS - FILTER          SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG          : ORDERCUST

    SPACE            :
    SPACE-OWNER      :

    RECOVERY-UNIT    :
    REC-TIMESTAMP    :     -  -     :  :  .

    ARCHIV-DIR-NAME  :
    VERSION          :                      DALOG-VERSION        :
    VALIDITY         :                      DALOG-SUBNUMBER      :
    MEDIUM           :                      NEXT-DALOG-VERSION   :
    RECOVERY-TYPE    :                      NEXT-DALOG-SUBNUMBER :
    COPY-TYPE        :                      PBI-COUNTER          :
    PBI-TIMESTAMP    :     -  -     :  :  .

--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------


LTG                                                         TAST
```

The recovery units of the user spaces are output in alphabetical order. The recovery units of the user space ORDER_STAT are output first.

If further backups are available, the input field in the command area automatically contains a “+”. Press the DUE key to display further backups of the ORDER_STAT user space.



```
INF.5.1                  INFORMATION SCHEMA, RECOVERY UNITS              SESAM/SQL
-------------------------------------------------------------------------------

    CATALOG          : ORDERCUST

    SPACE            : ORDSTAT
    SPACE-OWNER      : UTIADM

    RECOVERY-UNIT    : 2OSG:$ID1.ORDERCUST.ORDSTAT
    REC-TIMESTAMP    : <date> <time>

    ARCHIV-DIR-NAME  : $ID1.HSMSARCH
    VERSION          : 000000004             DALOG-VERSION        : 000000003
    VALIDITY         : YES                   DALOG-SUBNUMBER      : 000000001
    MEDIUM           : HSMW                  NEXT-DALOG-VERSION   : 000000004
    RECOVERY-TYPE    : COPY                  NEXT-DALOG-SUBNUMBER : 000000001
    COPY-TYPE        : OFFLINE               PBI-COUNTER          :
    PBI-TIMESTAMP    :     -  -     :  :  .

-------------------------------------------------------------------------------
 ===>: cop    F1=Help   F3=Terminate                       F13=Return
-------------------------------------------------------------------------------


LTG                                                             TAST
```

The depicted form displays the backup of the ORDSTAT user space that you are looking for. If you want to recover an individual space, memorize the version number VERSION of the RECOVERY-UNIT.

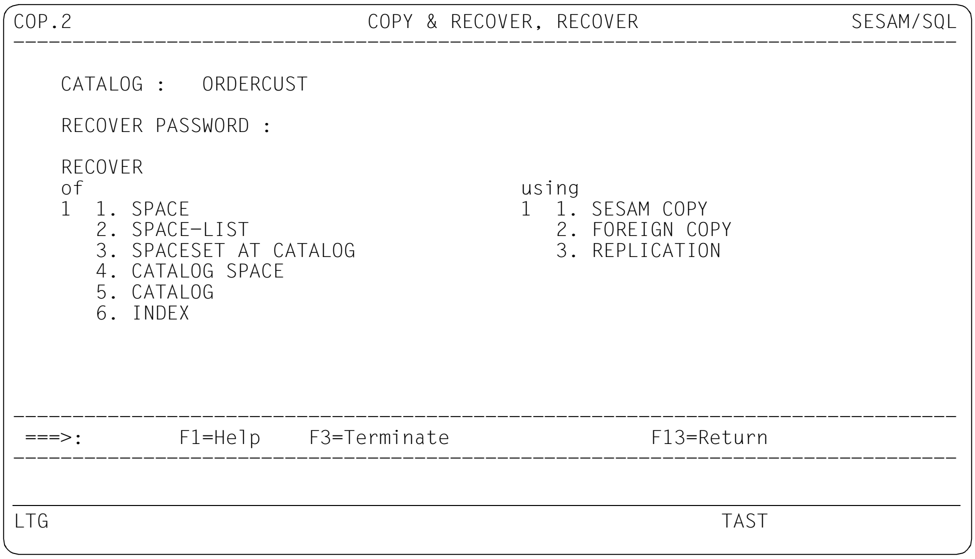
By entering “+” in the command area, you can display the backups of the other user spaces of the ORDERCUST database.

The backup of the entire database was performed in section “Creating a tape backup of a database” on page 55. Consequently, all the spaces (catalog space and user spaces) have the same time stamp. However, the version number of the SESAM backup copy for the individual spaces may be different. The version number depends on the number of copies of the space that have already been created.

Once you have identified the version number or the time stamp you require, you recover the database using COPY & RECOVER. Enter “cop” in the command area to display the COP form (see page 41).

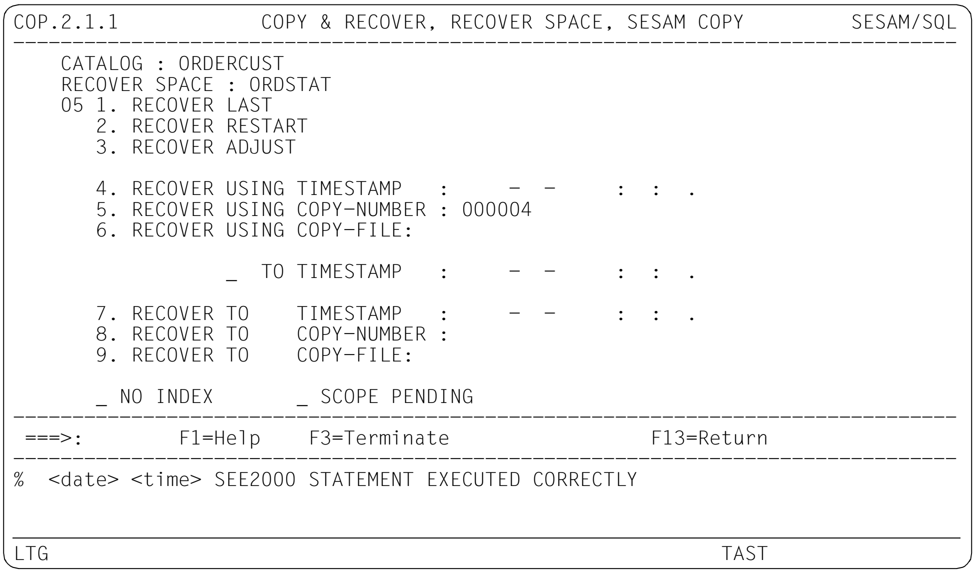
In the COP form, choose 2 for RECOVER in the function menu.
This displays the COP.2 form.

In the COP.2 form, choose 1 under “RECOVER of” and choose 1 under “using” in order to recover an individual user space from a SESAM backup. This displays the COP.2.1.1 form.



```
COP.2                          COPY & RECOVER, RECOVER                  SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG :   ORDERCUST

    RECOVER PASSWORD :

    RECOVER
    of                                      using
    1  1. SPACE                             1  1. SESAM COPY
       2. SPACE-LIST                           2. FOREIGN COPY
       3. SPACESET AT CATALOG                  3. REPLICATION
       4. CATALOG SPACE
       5. CATALOG
       6. INDEX




--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------


LTG                                                           TAST
```

In the form COP.2.1.1, choose function 5 in the function menu and enter the version number (COPY-NUMBER) that you have ascertained from the information schema INFORMATION_SCHEMA.
The utility monitor confirms that the recovery has been executed successfully.



```
COP.2.1.1            COPY & RECOVER, RECOVER SPACE, SESAM COPY        SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    RECOVER SPACE : ORDSTAT
    05 1. RECOVER LAST
       2. RECOVER RESTART
       3. RECOVER ADJUST

       4. RECOVER USING TIMESTAMP   :     -  -     :  :  .
       5. RECOVER USING COPY-NUMBER : 000004
       6. RECOVER USING COPY-FILE:

                   _  TO TIMESTAMP   :     -  -     :  :  .

       7. RECOVER TO    TIMESTAMP   :     -  -     :  :  .
       8. RECOVER TO    COPY-NUMBER :
       9. RECOVER TO    COPY-FILE:

       _ NO INDEX       _ SCOPE PENDING
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY


LTG                                                           TAST
```

2.7 Obtaining information from the information schemata

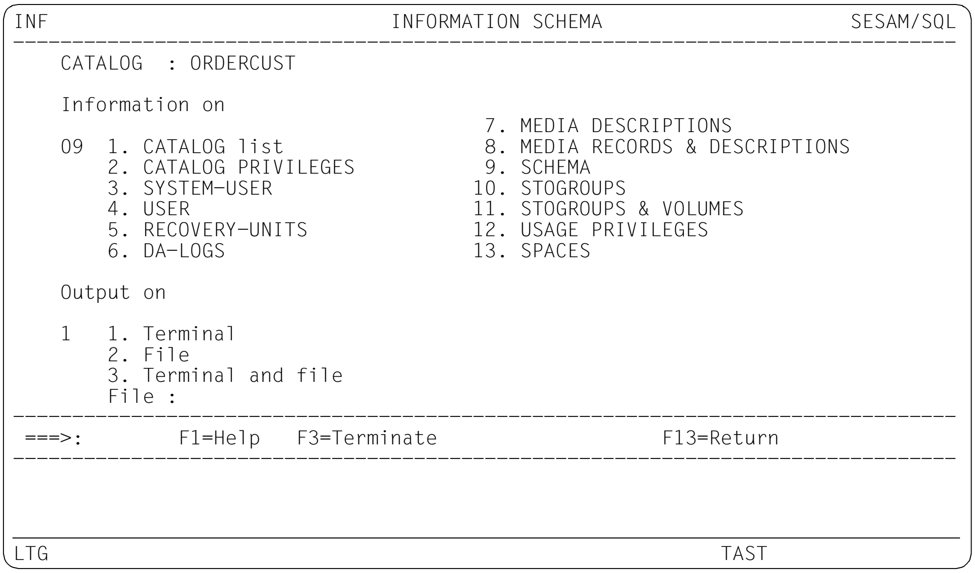
In this example, information is obtained from the information schemata INFORMATION_SCHEMA and SYS_INFO_SCHEMA.

Calling INFORMATION_SCHEMA

In the STM start form (see page 13), you enter 09. This displays the INF form.

Obtaining information on a schema

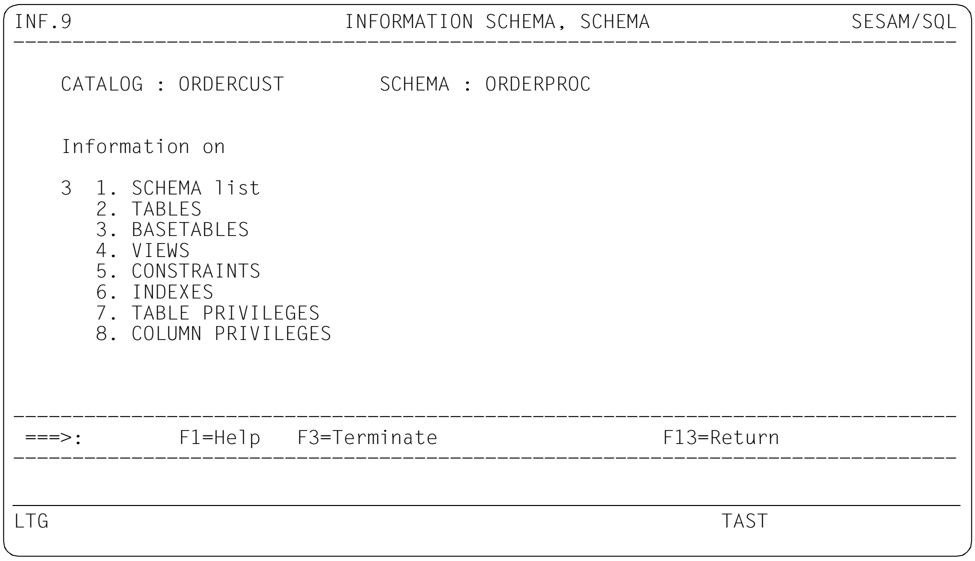
In the INF form you enter the name of the database and select function 9 “SCHEMA”. This displays the INF.9 continuation form.



```
INF                                INFORMATION SCHEMA                            SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG   : ORDERCUST

    Information on
                                         7. MEDIA DESCRIPTIONS
    09  1. CATALOG list                  8. MEDIA RECORDS & DESCRIPTIONS
        2. CATALOG PRIVILEGES            9. SCHEMA
        3. SYSTEM-USER                  10. STOGROUPS
        4. USER                         11. STOGROUPS & VOLUMES
        5. RECOVERY-UNITS               12. USAGE PRIVILEGES
        6. DA-LOGS                      13. SPACES

    Output on

    1   1. Terminal
        2. File
        3. Terminal and file
        File :
--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                   F13=Return
--------------------------------------------------------------------------------



LTG                                                          TAST
```

Displaying information on tables

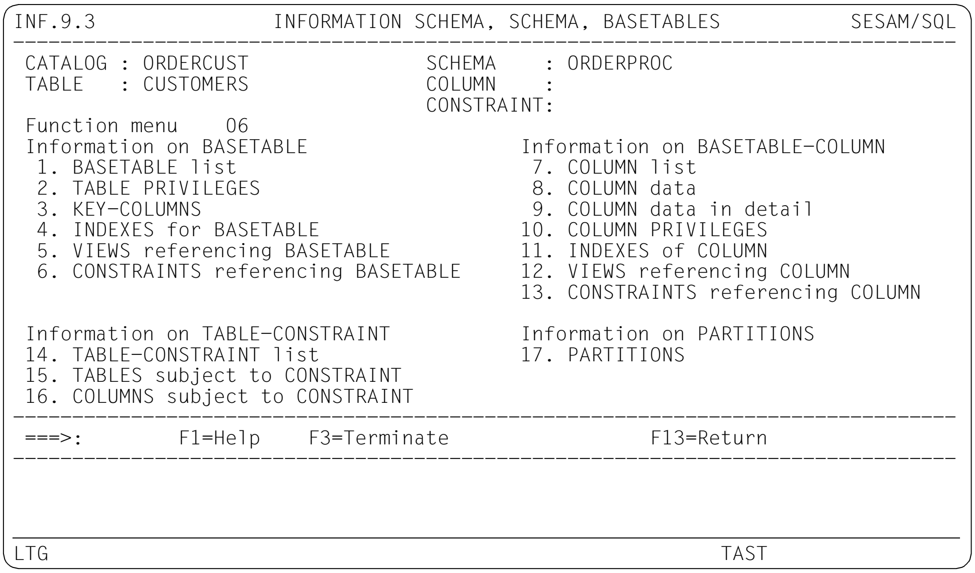
In the INF.9 continuation form you select function 3 “BASETABLES” to display further information on tables. This opens continuation form INF.9.3.



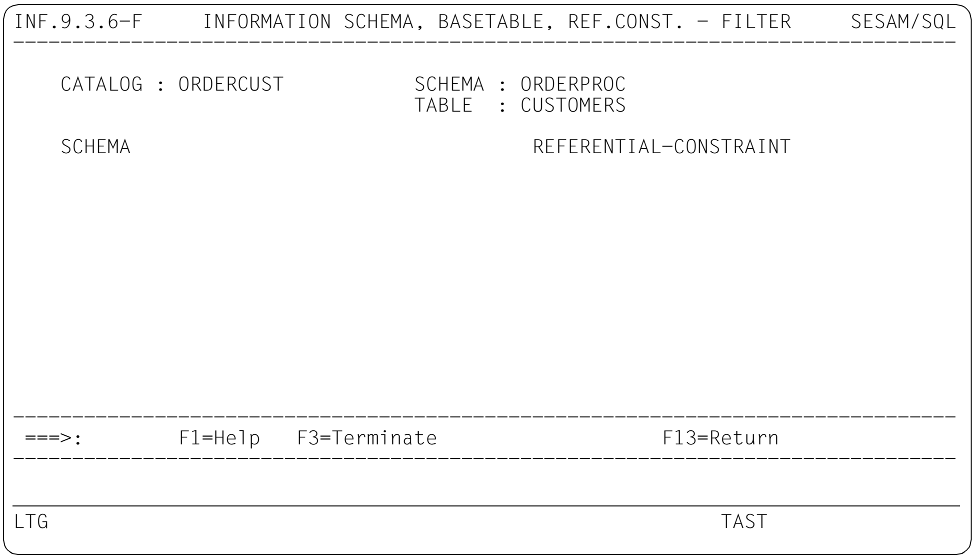
```
INF.9                        INFORMATION SCHEMA, SCHEMA                 SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST        SCHEMA : ORDERPROC


    Information on

    3  1. SCHEMA list
       2. TABLES
       3. BASETABLES
       4. VIEWS
       5. CONSTRAINTS
       6. INDEXES
       7. TABLE PRIVILEGES
       8. COLUMN PRIVILEGES



--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------

LTG                                                          TAST
```

Displaying information on integrity constraints

In the INF.9.3 continuation form you specify the CUSTOMERS base table and select function 6 to display information on integrity constraints that reference the CUSTOMERS table. You branch to the selection form INF.9.3.6-F.



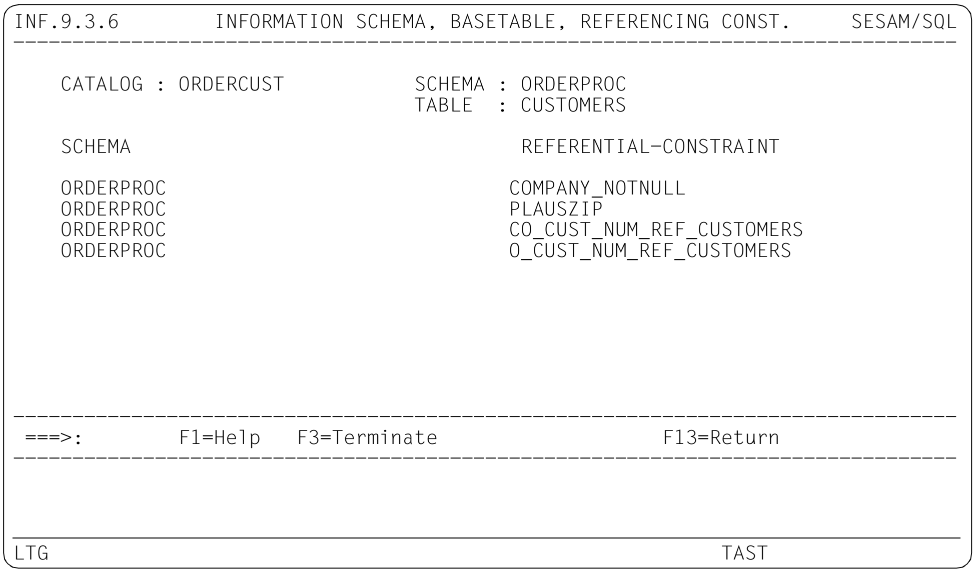
```
INF.9.3              INFORMATION SCHEMA, SCHEMA, BASETABLES          SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST                  SCHEMA     : ORDERPROC
 TABLE   : CUSTOMERS                  COLUMN     :
                                      CONSTRAINT:
 Function menu    06
 Information on BASETABLE                     Information on BASETABLE-COLUMN
  1. BASETABLE list                            7. COLUMN list
  2. TABLE PRIVILEGES                          8. COLUMN data
  3. KEY-COLUMNS                               9. COLUMN data in detail
  4. INDEXES for BASETABLE                    10. COLUMN PRIVILEGES
  5. VIEWS referencing BASETABLE              11. INDEXES of COLUMN
  6. CONSTRAINTS referencing BASETABLE        12. VIEWS referencing COLUMN
                                              13. CONSTRAINTS referencing COLUMN

 Information on TABLE-CONSTRAINT              Information on PARTITIONS
 14. TABLE-CONSTRAINT list                    17. PARTITIONS
 15. TABLES subject to CONSTRAINT
 16. COLUMNS subject to CONSTRAINT
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------



LTG                                                         TAST
```



```
INF.9.3.6-F     INFORMATION SCHEMA, BASETABLE, REF.CONST. - FILTER     SESAM/SQL
--------------------------------------------------------------------------------

   CATALOG : ORDERCUST           SCHEMA : ORDERPROC
                                 TABLE  : CUSTOMERS

   SCHEMA                                  REFERENTIAL-CONSTRAINT










--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------

LTG                                                         TAST
```

You send the INF.9.3.6-F selection form by pressing the DUE key (see also section "Specifying the extent of the output – selection forms" on page 248).

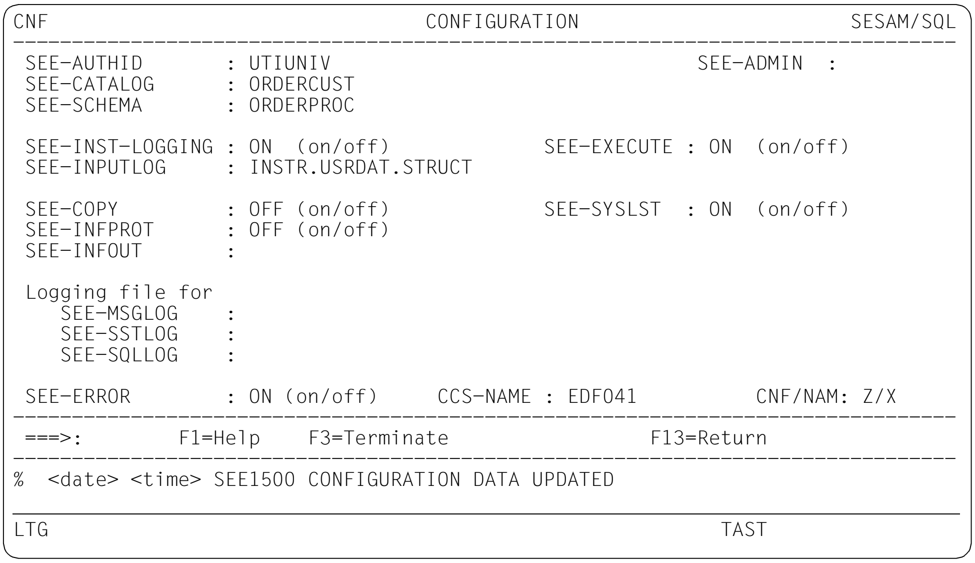
The INF.9.3.6 continuation form displays all the integrity constraints that reference the CUSTOMERS table.



```
INF.9.3.6        INFORMATION SCHEMA, BASETABLE, REFERENCING CONST.     SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST               SCHEMA : ORDERPROC
                                      TABLE  : CUSTOMERS

    SCHEMA                                     REFERENTIAL-CONSTRAINT

    ORDERPROC                                 COMPANY_NOTNULL
    ORDERPROC                                 PLAUSZIP
    ORDERPROC                                 CO_CUST_NUM_REF_CUSTOMERS
    ORDERPROC                                 O_CUST_NUM_REF_CUSTOMERS








--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                   F13=Return
--------------------------------------------------------------------------------



LTG                                                           TAST
```

Changing the authorization key

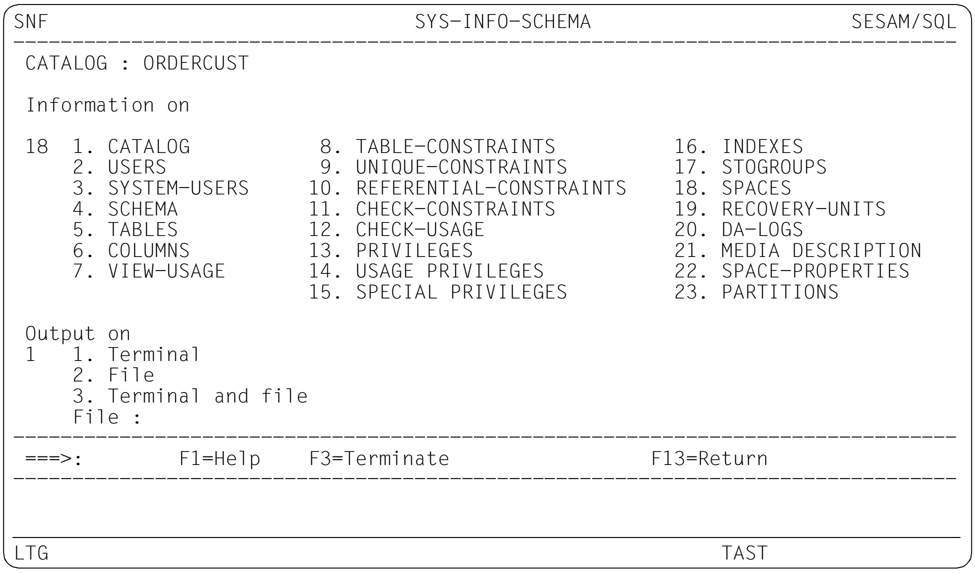
In order to display information from SYS_INFO_SCHEMA, you have to change the authorization key. To do this, you enter cnf in the command area to display the CNF form, in which you then enter the universal user UTIUNIV as the new authorization key. The utility monitor confirms that your entries have been accepted.



```
CNF                                 CONFIGURATION                        SESAM/SQL
-------------------------------------------------------------------------------
 SEE-AUTHID       : UTIUNIV                                  SEE-ADMIN  :
 SEE-CATALOG      : ORDERCUST
 SEE-SCHEMA       : ORDERPROC

 SEE-INST-LOGGING : ON  (on/off)           SEE-EXECUTE : ON  (on/off)
 SEE-INPUTLOG     : INSTR.USRDAT.STRUCT

 SEE-COPY         : OFF (on/off)           SEE-SYSLST  : ON  (on/off)
 SEE-INFPROT      : OFF (on/off)
 SEE-INFOUT       :

 Logging file for
    SEE-MSGLOG    :
    SEE-SSTLOG    :
    SEE-SQLLOG    :

 SEE-ERROR        : ON (on/off)     CCS-NAME : EDF041         CNF/NAM: Z/X
-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------
%  <date> <time> SEE1500 CONFIGURATION DATA UPDATED

LTG                                                         TAST
```

Calling SYS_INFO_SCHEMA

In the CNF form, you enter snf in the command area to display the SNF form. Select function 18 in the function menu. This opens the selection form SNF.18-F.

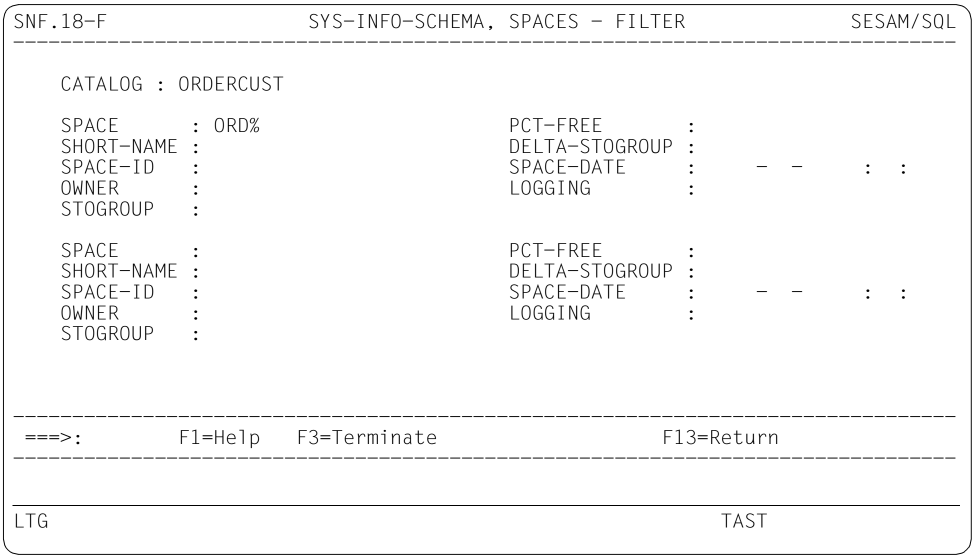


```
SNF                                SYS-INFO-SCHEMA                        SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST

 Information on

 18  1. CATALOG           8. TABLE-CONSTRAINTS         16. INDEXES
     2. USERS             9. UNIQUE-CONSTRAINTS        17. STOGROUPS
     3. SYSTEM-USERS     10. REFERENTIAL-CONSTRAINTS   18. SPACES
     4. SCHEMA           11. CHECK-CONSTRAINTS         19. RECOVERY-UNITS
     5. TABLES           12. CHECK-USAGE               20. DA-LOGS
     6. COLUMNS          13. PRIVILEGES                21. MEDIA DESCRIPTION
     7. VIEW-USAGE       14. USAGE PRIVILEGES          22. SPACE-PROPERTIES
                         15. SPECIAL PRIVILEGES        23. PARTITIONS

 Output on
 1   1. Terminal
     2. File
     3. Terminal and file
     File :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
--------------------------------------------------------------------------------


LTG                                                          TAST
```

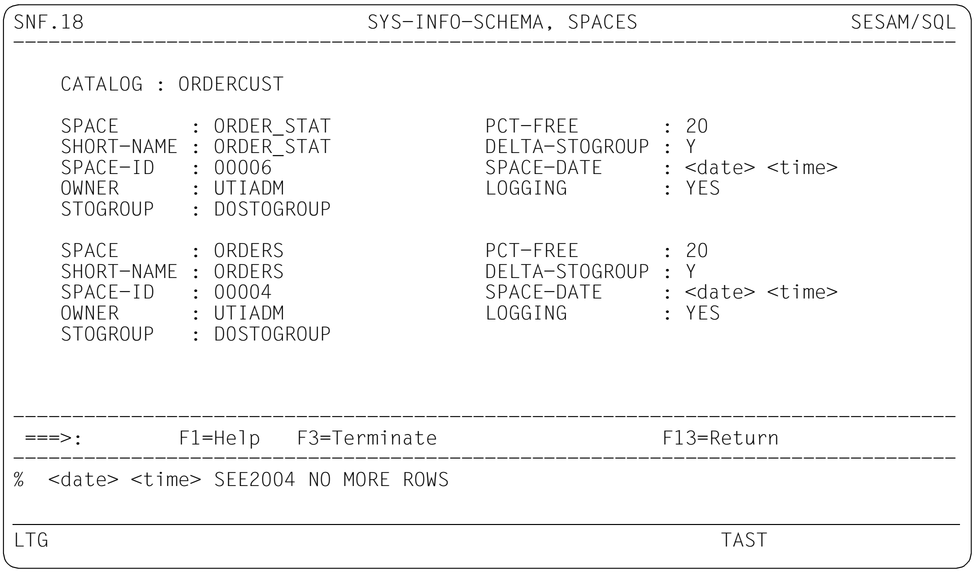
Displaying information on spaces

In the SNF.18-F selection form, you enter the partially qualified value ORD in the SPACE input field (see also section "Specifying the extent of the output – selection forms" on page 248).

In the SNF.18 continuation form information is displayed on the ORDER_STAT and ORDERS user spaces.



```
SNF.18-F                  SYS-INFO-SCHEMA, SPACES - FILTER              SESAM/SQL
-------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    SPACE      : ORD%                          PCT-FREE       :
    SHORT-NAME :                               DELTA-STOGROUP :
    SPACE-ID   :                               SPACE-DATE     :      -  -      :  :
    OWNER      :                               LOGGING        :
    STOGROUP   :

    SPACE      :                               PCT-FREE       :
    SHORT-NAME :                               DELTA-STOGROUP :
    SPACE-ID   :                               SPACE-DATE     :      -  -      :  :
    OWNER      :                               LOGGING        :
    STOGROUP   :



-------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                        F13=Return
-------------------------------------------------------------------------------


LTG                                                                 TAST
```



```
SNF.18                         SYS-INFO-SCHEMA, SPACES                  SESAM/SQL
-------------------------------------------------------------------------------

    CATALOG : ORDERCUST

    SPACE      : ORDER_STAT                  PCT-FREE       : 20
    SHORT-NAME : ORDER_STAT                  DELTA-STOGROUP : Y
    SPACE-ID   : 00006                       SPACE-DATE     : <date> <time>
    OWNER      : UTIADM                      LOGGING        : YES
    STOGROUP   : DOSTOGROUP

    SPACE      : ORDERS                      PCT-FREE       : 20
    SHORT-NAME : ORDERS                      DELTA-STOGROUP : Y
    SPACE-ID   : 00004                       SPACE-DATE     : <date> <time>
    OWNER      : UTIADM                      LOGGING        : YES
    STOGROUP   : DOSTOGROUP



-------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                        F13=Return
-------------------------------------------------------------------------------
%  <date> <time> SEE2004 NO MORE ROWS


LTG                                                                 TAST
```

The utility monitor informs you that there is no more information to be displayed. Exit the utility monitor as described on page 18.

3 Working with the utility monitor

The utility monitor is a SESAM/SQL tool with which the database administrator or system administrator can carry out administration tasks. It supports the user with predefined activities and automatic backup functions. You can use it to:

- create, back up and recover a database
- reorganize and carry out checks on spaces
- load and unload user data
- import and export tables
- back up database objects automatically before and after a change
- evaluate a database's information schemata INFORMATION_SCHEMA and SYS_INFO_SCHEMA
- read and delete metadata from the spaces and the catalog recovery file (CAT-REC file)
- manage backup copies
- administer the DBH using the SESADM administration program
- Issues dynamically compilable SQL statements

These tasks can be carried out either in a form-driven dialog or by processing an instruction file in interactive or batch mode.

The utility monitor works with both the independent DBH and with the linked-in DBH (/390 servers only).

The following diagram shows the two input options for starting the utility monitor.

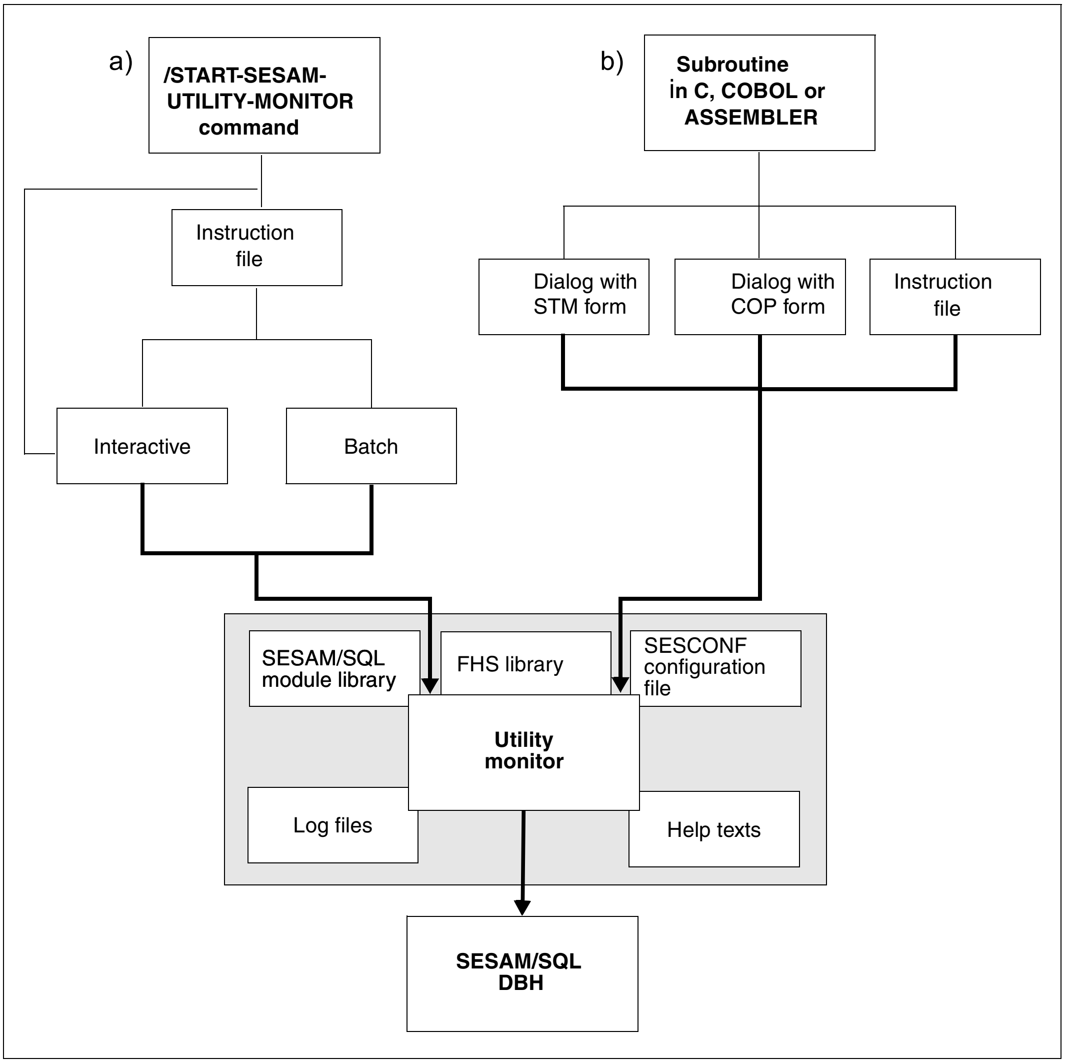


Figure 1: Input options for starting the utility monitor (variant for the independent DBH)

Explanations

a) You can start the utility monitor with the /START-SESAM-UTILITY-MONITOR command. The name of the corresponding command for the utility monitor with linked-in DBH is /START-SESLK-UTILITY-MONITOR.
Both commands can also be embedded in a BS2000 procedure.

In interactive mode, the utility monitor receives input via FHS forms. SQL statements are formed based on this input and sent to the DBH. The utility monitor then evaluates the replies and results of the DBH and displays them to the user.

The utility monitor can also read input from an instruction file. It can be created by the user with an editor or by the utility monitor as a dialog protocol. The instruction file can be processed in interactive or batch mode.

b) The utility monitor can be called as a subroutine from a C, COBOL or Assembler program. It is started either via the STM - START MENU or COP - COPY & RECOVER / REPLICATION form, or by processing an instruction file.

The modules required to start the utility monitor are loaded dynamically from the SESAM/SQL module library. The name of the hardware-dependent module library is SYSLNK.SESAM-SQL.<ver> for /390 servers and SKULNK.SESAM-SQL.<ver> for x86 servers.

The utility monitor logs the SQL statements sent to the DBH, the SQLSTATEs and its own messages in log files.

The user enters the configuration data in a configuration file with the link name SESCONF or in the global configuration file.

The FHS library and the file containing the help texts are available in both English and German. When the utility monitor is started the files are assigned according to the language setting for the message output. This can be set with the /MODIFY-MSG-ATTRIBUTES command (see manual “Commands”).

Access from the World Wide Web

You can also access the administration program SESADM, the performance monitor SESMON and the utility monitor SESUTI all from a unified access on the World Wide Web (WWW or Web for short) with the aid of the software product WebTransactions (WebTA).

To access the SESAM programs via the Web, you only need a standard browser in addition to the software product WebTransactions.

Web access is described in the document “WebTA access for SESAM/SQL” shipped together with SESAM/SQL-Server. This document is also available from our manual server under the software product SESAM/SQL.

The program run of the utility monitor

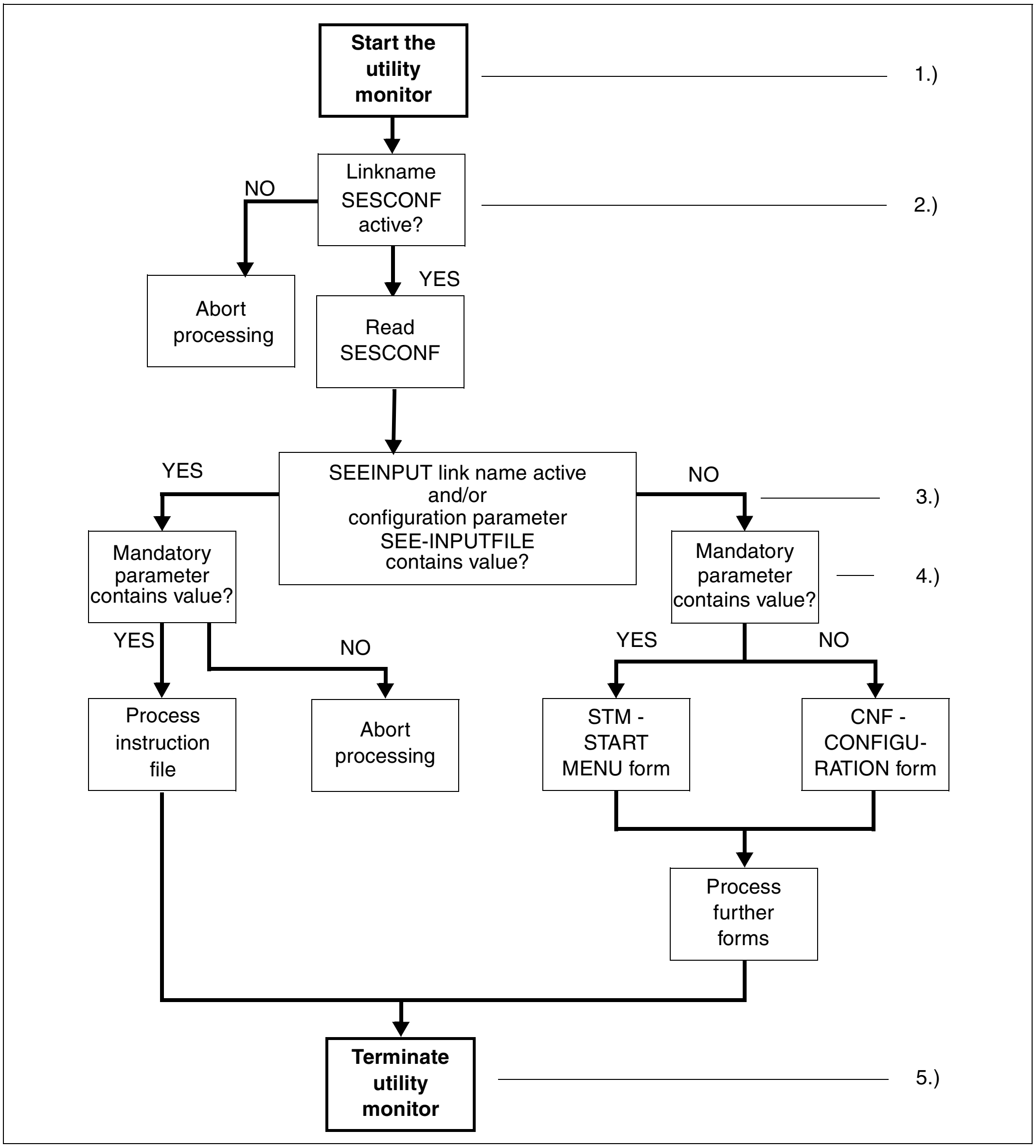

Start the utility monitor
1.)
Linkname SESCONF active?
NO
YES
2.)
Abort processing
Read SESCONF
SEEINPUT link name active and/or configuration parameter SEE-INPUTFILE contains value?
YES
NO
3.)
Mandatory parameter contains value?
Mandatory parameter contains value?
4.)
YES
NO
YES
NO
Process instruction file
Abort processing
STM - START MENU form
CNF - CONFIGU-RATION form
Process further forms
Terminate utility monitor
5.)

Figure 2: The program run of the utility monitor

Explanations

1. The utility monitor is started by means of the START-SESAM-UTILITY-MONITOR command (see also section “The sequence of commands for starting the utility monitor” on page 76).
2. If the link name SESCONF has been assigned, the SESCONF configuration file is opened and the configuration parameters are read. The configuration file is then closed again. If SESCONF has not been assigned, the program is aborted.
3. If the link name SEEINPUT has been assigned or the configuration parameter SEE-INPUTFILE has been assigned a value and the mandatory parameters have been supplied, the instruction file thus assigned is opened and processed in batch mode. If both SEEINPUT and SEE-INPUTFILE are specified, only SEEINPUT is evaluated. If the mandatory parameters have not been assigned, processing of the instruction file is aborted.

   If there is no instruction file, the input is processed in interactive mode. You can process instruction files in interactive mode by specifying them in the IFP - INSTRUCTION FILE PROCESSING form.
4. If the SESCONF configuration file contains all the required information, the STM - START MENU form appears.

   If no configuration file has been assigned or the mandatory parameters have not been supplied, the CNF - CONFIGURATION form appears, in which you are requested to make an entry for the SEE-AUTHID parameter.
5. In interactive mode, you exit the utility monitor by pressing the $\boxed{\text{F3}}$ key or entering F3 in the command area. To confirm your choice, press the $\boxed{\text{F3}}$ key again or re-enter F3 in the command area.

   When processing an instruction file in batch mode, the utility monitor terminates when it finds the END statement. If there is no END statement, it terminates at the end of the file and issues a warning.

3.1 The sequence of commands for starting the utility monitor

```
[/ADD-FILE-LINK,LINK-NAME=SESCONF,FILE-NAME=configuration_file
     ,ACCESS-METHOD=SAM]
or
[/CONNECT-SESAM-CONFIGURATION TO-FILE=configuration_file.global
     ,CONFIGURATION-LINK=linkname]------------------------------------------- (1)
[/ADD-FILE-LINK,LINK-NAME=SEETRACE,FILE-NAME=trace_file
     ,ACCESS-METHOD=SAM] ---------------------------------------------------- (2)
[/ADD-FILE-LINK,LINK-NAME=SEEINPUT,FILE-NAME=instruction_file
     ,ACCESS-METHOD=SAM] ---------------------------------------------------- (3)
[/ADD-FILE-LINK,LINK-NAME=SESAMCID,FILE-NAME=catid_list_file
     ,ACCESS-METHOD=SAM] ---------------------------------------------------- (4)
/START-SESAM-UTILITY-MONITOR
or
/START-SESLK-UTILITY-MONITOR ------------------------------------------------ (5)
```

(1) You can assign a configuration file with the link name SESCONF.
The configuration file must be a SAM file (see the “Core manual”, configuration file). It is also possible to group together local configuration files to form a global configuration file (see the section entitled “The global configuration file” in the “Core manual”). In this case you assign the configuration file using the CONNECT-SESAM-CONFIGURATION command.

configuration_file
User-defined name of the configuration file

configuration_file.global
User-defined name of the global configuration file

linkname
Link name under which the DBH options are defined in the global configuration file.

See also section “Entering configuration data” on page 85.

(2) You can assign an output file with the link name SEETRACE for the diagnostic trace. The utility monitor generates the file implicitly as a SAM file. If the file already exists, the utility monitor extends it.

trace_file
User-defined name of the diagnostic trace file Default name:
SESUTI.INPUTLOG.*tsn.yyyymmddhhmmss*

tsn Task sequence number

yyyymmddhhmmss
Current date and time

(3) If you want to start the utility monitor in the batch mode, then you need to create an instruction file. The instruction file must be a SAM file. If the instruction file is stored as a BS2000 file, then you can assign it using the line name SEEINPUT or via the configuration parameter SEE-INPUTFILE. If the instruction file is stored as a member of an LMS library, you must assign it via the configuration parameter SEE-INPUTFILE.

instruction_file
User-defined name of the instruction file Default name: SESUTI.INPUTLOG.*tsn.yyyymmddhhmmss*

tsn Task sequence number

yyyymmddhhmmss
Current date and time

See also section “Creating and processing an instruction file” on page 89.

(4) You can restrict the search for files in SESAM-DBH to predefined catalog IDs. To do this, you enter a list of the required IDs in a file. To ensure that the DBH and the utility monitor access the same IDs, you assign this CATID list to the utility monitor via the link name SESAMCID. The CATID list is evaluated the first time a file is accessed. Changes in the file do not take effect until the utility monitor is rebooted (see the “Core manual”).

catid_list_file
User-defined name of the CATID list file.

(5) Starts SESUTI or SESUTIL with the start command (see the “Database Operation” manual). The following variants are permitted:

for SESUTI utility monitor (independent)	for SESUTIL utility monitor (linked-in)
START-SESAM-UTILITY-MONITOR SESAM-UTILITY-MONITOR START-SESUTI SESUTI	START-SESLK-UTILITY-MONITOR SESLK-UTILITY-MONITOR START-SESUTIL SESUTIL

You start the utility monitor with one of these commands.

If you have assigned an instruction file with the link name SEEINPUT or with the configuration parameter SEE-INPUTFILE, the corresponding instruction file is processed. If not, the STM - START MENU start form appears. If there is no configuration file or mandatory parameters have not been supplied, the CNF - CONFIGURATION form appears first, in which you can then make the missing entries.

See also section “Entering configuration data” on page 85.

Starting the utility monitor from a BS2000 procedure

The utility monitor can be started from a BS2000 procedure in either interactive or batch mode.

The examples below show you how to include the utility monitor call in a BS2000 procedure.

Example 1: Starting the utility monitor in interactive mode

You create a procedure file called UTIMON.START.

```
/BEGIN-PROCEDURE LOGGING=ALL,PARAMETERS=NO
/ADD-FILE-LINK LINK-NAME=SESCONF,FILE-NAME=SESCONF.SESUTI.ZX        -
/                              ,ACCESS-METHOD=SAM
/START-SESAM-UTILITY-MONITOR
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/END-PROCEDURE
```

The procedure is started with the following command:

```
/CALL-PROCEDURE FROM-FILE=UTIMON.START
```

Example 2: Starting the utility monitor in batch mode

You create a procedure file called UTIMON.START.ENTER. The instruction file SESUTI.USRDAT is stored as a BS2000 file.

```
/LOGON
/ASSIGN-SYSOUT TO-FILE=ENTER.OUT
/ADD-FILE-LINK LINK-NAME=SESCONF,FILE-NAME=SESCONF.SESUTI.ZX         -
/                              ,ACCESS-METHOD=SAM
/ADD-FILE-LINK LINK-NAME=SEEINPUT,FILE-NAME=SESUTI.ANWDAT            -
/                               ,ACCESS-METHOD=SAM
/START-SESAM-UTILITY-MONITOR
/ASSIGN-SYSDTA TO-FILE=*PRIMARY
/LOGOFF
```

i If the instruction file is stored as a member of an LMS library, you must assign it using the configuration parameter SEE-INPUTFILE in the configuration file. In this event, the link assignment SEEINPUT is omitted.

The procedure is started with the following command:

```
/ENTER-JOB FROM-FILE=UTIMON.START.ENTER,JOB-CLASS=JCB32000
```

See BEGIN-PROCEDURE, CALL-PROCEDURE and ENTER-JOB (manual “Commands”).

3.2 Define configuration data

The configuration data you enter controls how the utility monitor runs. Once you have specified the configuration parameters you require, you can enter them in a configuration file, the CNF - CONFIGURATION form or an instruction file.

3.2.1 Selecting configuration parameters

You define the configuration by making entries for the various configuration parameters. The table below contains all the configuration parameters available for controlling the utility monitor. Those configuration parameters that are not fully described in the table are explained in more detail below it.

Parameter	**Default value/ default file name**	**Meaning**
SEE-ADMIN=*password*		Specifies a three-character password for DBH administration. In administration via the CALL DML interface, this is a **mandatory parameter**. See also page 129 and the DBH option ADMINISTRATOR in the “Database Operation” manual.
SEE-AUTHID= *authorization_key*		Specifies an authorization key. **This parameter is mandatory.**
SEE-CATALOG= *catalog_name*	D0CATALOG	Sets a default logical database name so that partially qualified object names can be specified.
SEE-COPY={**ON**/OFF}	ON	Controls automatic backup; ON: activates automatic backup. OFF: deactivates automatic backup.
SEE-ERROR={**ON**/OFF}	ON	Controls the response to DBH error messages during processing of an instruction file; ON: processing is aborted when error messages occur. OFF: processing is continued if error messages occur. If there are syntax errors in the instruction file, processing of the file is **always** aborted.

Table 2: Configuration parameter (part 1 of 4)

Parameter	**Default value/ default file name**	**Meaning**
SEE-EXECUTE={**ON**/OFF}	ON	Controls execution of the statements to be logged; ON: the statements are logged in the instruction file and executed. OFF: the statements are logged in the instruction file but not executed. This parameter is only evaluated if you specify SEE-INST-LOGGING=ON.
SEE-INFOUT= *file_name/libmem*	SESUTI.INFOUT. *tsn.yyyymmddhhmmss*	Name of the output file in which the information (requested by CMD INF statements from a configuration file or instruction file) is written from the information schemata INFORMATION_SCHEMA and SYS_INFO_SCHEMA; *file_name*: any name complying with BS2000 conventions *libmem*: any LMS member name complying with LMS conventions
SEE-INFPROT={ON/**OFF**}	OFF	Controls the logging of inquiries to the information schemata INFORMATION_SCHEMA and SYS_INFO_SCHEMA. ON: The inquiries are logged in the instruction file. OFF: The inquiries are not logged in the instruction file.
SEE-INPUTFILE= *file_name/libmem*		Specifies the name of the instruction file which is to run in batch mode. *file_name*: any file name complying with BS2000 conventions *libmem*: any LMS member name complying with LMS conventions
SEE-INPUTLOG= *file_name/libmem*	SESUTI.INPUTLOG. *tsn.yyyymmddhhmmss*	Specifies the name of the instruction file in which the logged information is to be written. *file_name*: any file name complying with BS2000 conventions *libmem*: any LMS member name complying with LMS conventions

Table 2: Configuration parameter (part 2 of 4)

Parameter	**Default value/ default file name**	**Meaning**
SEE-INST-LOGGING= {ON/**OFF**}	OFF	Controls the creation of an instruction file. ON: activates logging OFF: deactivates logging
Log files: – SEE-MSGLOG= *file_name*/*libmem* – SEE-SQLLOG= *file_name*/*libmem* – SEE-SSTLOG= *file_name*/*libmem*	SESUTI.STDLOG. *tsn.yyyymmddhhmmss*	Assigns a log file for messages. Assigns a log file for SQL statements. Assigns a log file for SQLSTATEs. *file_name*: any file name complying with BS2000 conventions *libmem*: any LMS member name complying with LMS conventions
SEE-SCHEMA= *schema_name*	D0SCHEMA	Sets a default schema name so that partially qualified object names can be specified.
SEE-STOGROUP=*stogroup*	D0STOGROUP	Specifies the storage group to be filled. You cannot change the parameter in the CNF - CONFIGURATION form; it applies throughout the session.
SEE-SYSLST={ON/**OFF**}	OFF	Controls logging to SYSLST; ON: data is also to be logged to SYSLST OFF: no data is to be logged to SYSLST
SEE-TRACE={**0**/1/2}	0 SESUTI.TRACE. *tsn.yyyymmddhhmmss*	Specifies the diagnostic trace level 1. 0: no diagnostic trace 1: diagnostic trace for main functions and their parameters 2: diagnostic trace for all functions; special diagnostic information is also output.

Table 2: Configuration parameter (part 3 of 4)

Parameter	**Default value/ default file name**	**Meaning**
Parameters for creating user spaces:		Sets defaults for user space parameters (see the SQL statement CREATE SPACE in the “SQL Reference Manual Part 1: SQL Statements”)
– SEE-DESTROY={**Y**/N}	Y	When the space is deleted, the storage space is to be: Y: overwritten with binary zero N: simply released
– SEE-LOG=N		Specifies that there is to be no logical data backup.
– SEE-PCTFREE=*integer*	20	Specifies the percentage of space that is free. *integer*: integer from 0 to 70
– SEE-PRIMARY=*integer*	24	Specifies the primary assignment of the space in kilobytes. *integer*: integer from 1 to 2 147 483 640
– SEE-SECONDARY= *integer*	24	Specifies the secondary assignment of the space in kilobytes. *integer*: integer from 1 to 32767
– SEE-SHARE={Y/**N**}	N	Controls the shareability of a space: Y: the space is shareable. N: the space is not shareable. You cannot change the parameters for user spaces in the CNF - CONFIGURATION form; they apply throughout the session.

Table 2: Configuration parameter (part 4 of 4)

SEE-AUTHID=*authorization_key*

You use this parameter to specify for the current SQL session the authorization key set by means of the SQL statement SET SESSION AUTHORIZATION and under which the utility monitor is to work. See the SQL statement SET SESSION AUTHORIZATION in the “SQL Reference Manual Part 1: SQL Statements”.

The SEE-AUTHID parameter is mandatory. You can change it during the session.

If you do not specify SEE-AUTHID, the utility monitor displays the CNF - CONFIGURATION form at the beginning of interactive processing and requests you to enter this parameter (see also section “Entering configuration data in the CNF form” on page 86 and section section “Entering configuration data (CNF - CONFIGURATION)” on page 206).
In batch mode, processing is aborted if SEE-AUTHID is not specified before the first SQL statement in either the configuration file or the instruction file.
See also section “Specifying access authorization” on page 129.

SEE-COPY={ON/OFF}

You can use this parameter to specify that database objects be backed up automatically. A backup may include the catalog space containing the database's metadata, and all spaces containing tables and indexes belonging to the database.

You are offered an automatic backup in the following situations:

- after creating a catalog space (CREATE CATALOG)
- before and after changing a database's metadata (ALTER CATALOG)
- before and after creating a schema (CREATE SCHEMA)
- before and after changing a schema (ALTER SCHEMA)
- before and after executing the IMPORT TABLE utility statement
- before and after executing the LOAD utility statement

The COP - COPY & RECOVER / REPLICATION form appears with default settings and the heading AUTOMATIC BACKUP, COPY when you call or exit the ALC - ALTER CATALOG, ALS - ALTER SCHEMA, CRS - CREATE SCHEMA and LOD - LOAD forms and when you exit the CRC - CREATE CATALOG form.

SEE-MSGLOG=*file_name/libmem*
SEE-SQLLOG=*file_name/libmem*
SEE-SSTLOG=*file_name/libmem*

You can merge the SEE-MSGLOG, SEE-SQLLOG and SEE-SSTLOG log files in a single file by assigning the same file to each of the three parameters. The log files can be stored as BS2000 files or as members of an LMS library.

If you do not assign any log files, the messages, SQLSTATEs and SQL statements are written to a file with the default name SESUTI.STDLOG.*tsn.yyyymmddhhmmss*. If you assign less than three log files, everything else is written to the default log file.

tsn Task sequence number

yyyymmddhhmmss
Current date and time

SEE-TRACE={0/1/2}

The diagnostic trace is needed only for error diagnosis and should only be activated in concrete cases of error, since it significantly impairs performance. The diagnostic trace contains the following:
– the function call and name
– the function's parameters
– specific data areas within the function
– entries in the internal log
– the end of the function
– internal statements issued by the utility monitor
– SELECT statements issued by the utility monitor

The following are not logged:
– irrelevant help functions
– the results of SELECT statements

Instead of using the configuration file, you can set the diagnostic trace in the command area (tr0/tr1/tr2).

You can also control the diagnostic trace with the following BS2000 command:
```
/INFORM-PROGRAM MSG=C'SEE,TRACE={0/1/2}'
```

Diagnostic trace level 0 is the default: no diagnostic trace is performed.

Level 1 means a diagnostic trace for the main functions and their parameters. With level 2, a diagnostic trace is performed for all functions; in addition, special diagnostic information is output.

You can assign an output file with the link name SEETRACE for the diagnostic trace (see also section “The sequence of commands for starting the utility monitor” on page 76). If you do not assign an output file, the diagnostic trace is written to a file with the default name SESUTI.TRACE.*tsn.yyyymmddhhmmss*.

tsn Task sequence number

yyyymmddhhmmss
Current date and time

3.2.2 Entering configuration data

You can enter configuration data in the following ways:

- in a configuration file
- in the CNF - CONFIGURATION form
- in an instruction file

Entering configuration data in a configuration file

In the configuration file, you can enter **all** the configuration data. You create the file with an editor. Please note:

- The configuration file must be a SAM file.
- You can enter only one parameter per line.
- The parameters must begin in column 1 and must not contain any blanks.
- You can enter the parameters in any order you like.
- If you enter a parameter more than once, the last value entered applies.
- Start comment lines with “//REMARK”.

The utility monitor opens the configuration file, reads it and then closes it again. In the event of a syntax error, a message appears with the number of the relevant line in the configuration file, and the utility monitor aborts.

In addition to the configuration data for the utility monitor, the configuration file can also contain configuration data for other components, such as DBCON. You should enter the DBH name, the configuration name (connection module parameters NAM and CNF) and the CCS name (connection module parameter CCSN) in the configuration file, since you cannot make these entries in the CNF - CONFIGURATION form even though they can be displayed there.

If you do not enter the DBH name, the configuration name and the CCS name (connection module parameter CCSN) in the configuration file, the utility monitor is started with the defaults for the DBH name and configuration name (see the “Database Operation” manual). If the CCS name is not specified in the configuration file, the default value *NONE is used.

The utility monitor only reads the DBH name, the configuration name and the CCS name from the configuration file together with the data intended for it. This data is marked with the escape symbol SEE. It ignores all other data.

Before starting the utility monitor, assign the configuration file with the link name SESCONF. See section “The sequence of commands for starting the utility monitor” on page 76.

It is also possible to group together local configuration files to form a global configuration file (see the section entitled “The global configuration file” in the “Core manual”). In this case you assign the configuration file using the CONNECT-SESAM-CONFIGURATION command.

You can display the utility monitor configuration file with the F6 function, see page 88.

Entering configuration data in the CNF form

In interactive mode, you can enter or change the dynamically modifiable configuration data in the CNF - CONFIGURATION form. The following configuration data is dynamically modifiable:

SEE-ADMIN	SEE-ERROR	SEE-INPUTLOG	SEE-SQLLOG
SEE-AUTHID	SEE-EXECUTE	SEE-INST-LOGGING	SEE-SSTLOG
SEE-CATALOG	SEE-INFOUT	SEE-MSGLOG	SEE-SYSLST
SEE-COPY	SEE-INFPROT	SEE-SCHEMA	

If you have not assigned a configuration file or not supplied mandatory parameters, the utility monitor displays the CNF - CONFIGURATION form and requests you to enter the missing information. See also section “Specifying access authorization” on page 129.

During the runtime of the utility monitor you can change the configuration data in the CNF - CONFIGURATION form. The changes apply only to the current session and are not written to the configuration file.

The current DBH name, the configuration name (connection module parameters NAM and CNF) and the CCS name (connection module parameter CCSN) are displayed in the CNF - CONFIGURATION form. You can only edit these parameters in the configuration file.

See also section “Entering configuration data (CNF - CONFIGURATION)” on page 206.

Entering configuration data in an instruction file

You can also define some of the dynamically modifiable configuration data in an instruction file:

SEE-AUTHID
SEE-ERROR
SEE-INFOUT
SEE-MSGLOG
SEE-SQLLOG
SEE-SSTLOG
SEE-TRACE

This configuration data applies until it is changed or until processing of the specified instruction file is completed. If an instruction file containing configuration parameters that are changed is processed during the dialog, once processing of the file is completed, then the values set before the file was processed apply again.

The instruction file can be processed in interactive or batch mode.

In interactive mode, if a configuration file has not been assigned or mandatory parameters have not been supplied, the utility monitor displays the CNF - CONFIGURATION form.

In batch mode, processing of the instruction file is aborted if mandatory parameters have not been supplied.

3.2.3 Outputting configuration file

You can output the configuration file assigned to the utility monitor by pressing the F6 key or entering F6 in the command field.

The configuration file can be output in each work step of the utility monitor. The setting of the F6 key is not output in the command field of the forms. The functionality of the F6 key is given in form HLP.1 (command line help).

The configuration file is displayed for reading via an implicit `SHOW-FILE` command. To return to the current form, terminate the `SHOW-FILE` command with “END”.

The utility monitor displays the configuration file which was assigned when starting with the link-name “SESCONF” or within a global configuration file with the corresponding configuration link. If this file is no longer available, (e.g. because it has been deleted or renamed), an error is reported by the `SHOW-FILE` command.

Changes in the configuration data in the current session (using the form CNF - CONFIGURATION) are **not** entered in the configuration file. They cannot therefore be displayed with the F6 function.
The current DBH name and the configuration name (connection module parameter NAM or CNF) of the displayed configuration file are always the current ones because they cannot be changed via the form CNF - CONFIGURATION.

The **current** configuration parameters can be displayed and changed as previously via the form CNF - CONFIGURATION, see page 86.

Comments in the configuration file are also displayed with the F6 function.

3.3 Creating and processing an instruction file

An instruction file contains instructions for the utility monitor that are processed one after the other. It can contain:

– Configuration data
– SQL statements
– CMD statements
– Comments
– END statement

You can create the instruction file using an editor (see page 104), or you can create it as an interactive log using the utility monitor (see page 105).

Entries in the instruction file

Parameter/statement	Default value/ default file name	Meaning
Configuration data:		
SEE-AUTHID= *authorization_key*		Specifies an authorization key. This is a **mandatory parameter** if it is not specified in the configuration file. It must be specified before the first SQL statement.
SEE-ERROR={**ON**/OFF}	ON	Controls the response to error messages: ON: processing is aborted if error messages occur. OFF: processing is continued if error messages occur.
SEE-INFOUT= *file_name/libmem*	SESUTI.INFOUT. *tsn.yyyymmddhhmmss*	Outputs the CMD INF instructions.
SEE-MSGLOG= *file_name/libmem*	SESUTI.STDLOG *tsn.tsn.yyyymmddhhm mss*	Assigns a log file for messages.
SEE-SQLLOG= *file_name/libmem*	SESUTI.STDLOG. *tsn.yyyymmddhhmmss*	Assigns a log file for SQL statements.
SEE-SSTLOG= *file_name/libmem*	SESUTI.STDLOG. *tsn.yyyymmddhhmmss*	Assigns a log file for SQLSTATEs.
SEE-TRACE={**0**/1/2}	0 SESUTI.TRACE. *tsn.yyyymmddhhmmss*	Specifies the diagnostic trace level 1. 0: no diagnostic trace 1: diagnostic trace for main functions and their parameters 2: diagnostic trace for all functions

Table 3: Entries in the instruction file (part 1 of 2)

Parameter/statement	Default value/ default file name	Meaning
SQL statements:		
SQL *sql_statement*		Issues any dynamically compilable SQL statement (except for SELECT). See the “SQL Reference Manual Part 1: SQL Statements”.
CMD statements:		
CMD INF *form-short-name* [*object-list*]		Requests information from the information schemata INFORMATION_SCHEMA and SYS_INFO_SCHEMA.
CMD CATREC DEL_LAST_RU *file_name*		Deletes CAT-LOG records after the last recovery unit record.
CMD COPJV *object*		Determines the version number of the SESAM backup copy.
CMD CHECK FORMAL		Checks the formal correctness of a database.
Comments:		
**text*		Issues a user-defined text as a comment.
End of statement:		
END		Terminates processing of the instruction file.

Table 3: Entries in the instruction file (part 2 of 2)

Configuration data

The configuration data applies as of the point at which it is read and until the processing of the instruction file is completed. The values are not written to the configuration file with the link name SESCONF.
If the same item of configuration data is read twice, the new value replaces the old one.
If the instruction file does not contain any configuration data, the utility monitor uses the configuration data from the configuration file. In batch mode, if mandatory parameters are missing when the instruction file is processed, the utility monitor terminates and displays an error message.
In interactive mode, a warning is issued.

See also section “Selecting configuration parameters” on page 79.

SQL statements

The instruction file can contain any dynamically compilable SQL statements, with the exception of SELECT statements. While processing the instruction file, the utility monitor does not add any statements that control transactions. It does not commit or roll back any transactions automatically.
To ensure that the utility monitor does not terminate while a transaction is open, statements that control transactions (COMMIT WORK, ROLLBACK WORK) must be written explicitly in the instruction file.

See the SQL statements COMMIT WORK and ROLLBACK WORK in the “SQL Reference Manual Part 1: SQL Statements”.

CMD statements

The user can use CMD statements in the instruction file to output information from the information schemata, delete the CAT-LOG records after the last recovery unit record, and determine the version number of the SESAM backup copy after a COPY and store it in a job variable or check the formal correctness of a database.
For more information on job variables, see section “Output in job variables” on page 117 and the “Database Operation” manual.

- CMD INF *form-short-name* [*object-list*]

  Statements for outputting information from the information schemata are introduced with CMD INF.

  To select the desired information, the form short name must be specified (see table 4 on page 92).

  The selection is limited to particular records by specifying one or more selection criteria in the form of an object list (see table 5 on page 99).
  The selection criteria correspond to the input fields in the respective dialog forms or to the preset output fields in the activities INF - INFORMATION_SCHEMA and SNF - SYS_INFO_SCHEMA (INF or SNF form and their continuation forms).
  If there are various different selection criteria of the same type then they are terminated with a number in order to make it possible to distinguish between them. You should not specify the number itself.

  Example

  “SCHEMA” keyword in the INF.9.3.13 form, table 4 on page 92:

  CATALOG,SCHEMA (of the table),TABLE,COLUMN,schema2 (of the view),view

  Here an additional schema is specified as an optional selection criterion with “schema2”. The information in parentheses is explanatory.

Short form name	**Information** [1]	**Selection criteria (lowercase letters = optional)**
INF.2	CATALOG_PRIVILEGES	CATALOG,privilege,grantor, grantee,grantable
INF.3	SYSTEM_ENTRIES	CATALOG,user,sysuser,host,application
INF.4.1	USERS	CATALOG,USER
INF.4.2	TABLE_PRIVILEGES	CATALOG,USER,privilege,schema, table,grantor,grantee,grantable
INF.4.3	COLUMN_PRIVILEGES	CATALOG,USER,privilege,schema, table,column,grantor,grantee,grantable
INF.5.1	RECOVERY_UNITS	CATALOG,space,owner,recunit,archive
INF.5.2	Files for RECOVERY	Syntax see page 97
INF.6	DA_LOGS	CATALOG,version,subnumber
INF.7	MEDIA_DESCRIPTIONS	CATALOG
INF.8	MEDIA_RECORDS	CATALOG,file-type
INF.9.1	SCHEMATA	CATALOG,schema,owner
INF.9.2	TABLES	CATALOG,SCHEMA,table
INF.9.3.1	BASE_TABLES	CATALOG,SCHEMA,table,space, table-style
INF.9.3.2	TABLE_PRIVILEGES	CATALOG,SCHEMA,TABLE, privilege,grantor,grantee,grantable
INF.9.3.3	KEY_COLUMN_USAGE	CATALOG,SCHEMA,TABLE, column,constraint
INF.9.3.4	INDEXES	CATALOG,SCHEMA,TABLE, index,space,constraint
INF.9.3.5	VIEW_TABLE_USAGE	CATALOG,SCHEMA (of the table), TABLE, schema2 (of the view),view
INF.9.3.6	CONSTRAINT_TABLE_USAGE	CATALOG,SCHEMA (of the table), TABLE,schema2 (of the integrity constraint),constraint
INF.9.3.7	ROUTINE_TABLE_USAGE	CATALOG,SCHEMA (of the table), TABLE,schema2 (of the routine),routine [2]
INF.9.3.8	BASE_TABLE_COLUMNS (list)	CATALOG,SCHEMA,TABLE,column
INF.9.3.9	BASE_TABLE_COLUMNS (description)	CATALOG,SCHEMA,TABLE,COLUMN

Table 4: Short form names (part 1 of 6)

Short form name	**Information** [1]	**Selection criteria (lowercase letters = optional)**
INF.9.3.10	BASE_TABLE_COLUMNS (detailed data)	CATALOG,SCHEMA,TABLE,COLUMN
INF.9.3.11	COLUMN_PRIVILEGES	CATALOG,SCHEMA,TABLE, COLUMN,privilege,grantor,grantee, grantable
INF.9.3.12	INDEX_COLUMN_USAGE	CATALOG,SCHEMA,TABLE, COLUMN,index
INF.9.3.13	VIEW_COLUMN_USAGE	CATALOG,SCHEMA (of the table), TABLE,COLUMN, schema2 (of the view),view
INF.9.3.14	CONSTRAINT_COLUMN_USAGE	CATALOG,SCHEMA (of the table), TABLE,COLUMN,schema2 (of the integrity constraint),constraint
INF.9.3.15	ROUTINE_COLUMN_USAGE	CATALOG,SCHEMA (of the table), TABLE,COLUMN, schema2 (of the routine),routine
INF.9.3.16	TABLE_CONSTRAINTS	CATALOG,SCHEMA,TABLE, constraint,constraint-type
INF.9.3.17	CONSTRAINT_TABLE_USAGE	CATALOG,SCHEMA (of the integrity constraint),CONSTRAINT, schema2 (of the table),table
INF.9.3.18	CONSTRAINT_COLUMN_USAGE	CATALOG,SCHEMA (of the integrity constraint),CONSTRAINT, schema2 (of the table),table,column
INF.9.3.19	PARTITIONS	CATALOG,SCHEMA,TABLE, serial-number
INF.9.4.1	VIEWS (list)	CATALOG,SCHEMA,view
INF.9.4.2	VIEWS (description)	CATALOG,SCHEMA,VIEW
INF.9.4.3	VIEW_TABLE_USAGE	CATALOG,SCHEMA (of the view), VIEW,schema2 (of the table),table
INF.9.4.4	VIEW_COLUMN_USAGE	CATALOG,SCHEMA (of the view), VIEW,schema2 (of the table), table,column
INF.9.4.5	TABLE_PRIVILEGES	CATALOG,SCHEMA,VIEW, privilege,grantor,grantee,grantable
INF.9.4.6.1	COLUMNS (list)	CATALOG,SCHEMA,VIEW,column
INF.9.4.6.2	COLUMNS (description)	CATALOG,SCHEMA,VIEW,COLUMN

Table 4: Short form names (part 2 of 6)

Short form name	**Information** [1]	**Selection criteria (lowercase letters = optional)**
INF.9.4.7	VIEW_ROUTINE_USAGE	CATALOG,SCHEMA (of the view), VIEW,schema2 (of the routine),routine
INF.9.5.1	REFERENTIAL_CONSTRAINTS	CATALOG, SCHEMA (of the integrity constraint), constraint (of the integrity constraint), schema2 (of the integrity constraint), constraint2 (of the integrity constraint)
INF.9.5.2	CHECK_CONSTRAINTS (list)	CATALOG,SCHEMA,constraint
INF.9.5.3	CHECK_CONSTRAINTS (description)	CATALOG,SCHEMA,CONSTRAINT
INF.9.5.4	CONSTRAINT_TABLE_USAGE	CATALOG,SCHEMA (of the check or referential constraint),CONSTRAINT, schema2 (of the table),table
INF.9.5.5	CONSTRAINT_COLUMN_USAGE	CATALOG,SCHEMA (of the integrity constraint),CONSTRAINT, schema2 (of the table),table,column
INF.9.6.1	INDEXES (list)	CATALOG,SCHEMA,index,table,space
INF.9.6.2	INDEXES (description)	CATALOG,SCHEMA,INDEX
INF.9.6.3	INDEX_COLUMN_USAGE	CATALOG,SCHEMA,INDEX,column
INF.9.7	TABLE_PRIVILEGES	CATALOG,SCHEMA,privilege, table,grantor,grantee,grantable
INF.9.8	COLUMN_PRIVILEGES	CATALOG,SCHEMA,privilege,table, column,grantor,grantee,grantable
INF.9.9.1	ROUTINES (list)	CATALOG,SCHEMA, routine,routine-type
INF.9.9.2	ROUTINES (description)	CATALOG,SCHEMA,ROUTINE
INF.9.9.3	ROUTINE_PRIVILEGES	CATALOG,SCHEMA,ROUTINE, grantor,grantee,grantable
INF.9.9.4	PARAMETER (list)	CATALOG,SCHEMA,ROUTINE, parameter
INF.9.9.5	PARAMETER (description)	CATALOG,SCHEMA,ROUTINE, PARAMETER
INF.9.9.6	ROUTINE_TABLE_USAGE	CATALOG,SCHEMA (of the routine), ROUTINE,schema2 (of the table),table

Table 4: Short form names (part 3 of 6)

Short form name	**Information** [1]	**Selection criteria (lowercase letters = optional)**
INF.9.9.7	ROUTINE_COLUMN_USAGE	CATALOG,SCHEMA (of the routine), ROUTINE,schema2 (of the table), table,column
INF.9.9.8	ROUTINE_ROUTINE_USAGE (called routines of a routine, „called routines“)	CATALOG (for calling and called routines),SCHEMA,ROUTINE (in each case of the calling routine), schema2,routine2,routine-type (in each case of the called routine) [2]
INF.9.9.9	ROUTINE_ROUTINE_USAGE (calling routines of a routine, „calling routines“)	CATALOG (for calling and called routines),SCHEMA,ROUTINE (in each case of the called routine), schema2,routine2,routine-type (in each case of the calling routine)
INF.9.9.10	VIEW_ROUTINE_USAGE	CATALOG,SCHEMA (of the routine), ROUTINE,schema2 (of the view),view
INF.10	STOGROUPS	CATALOG,stogroup,owner
INF.11	STOGROUP_VOLUME_USAGE	CATALOG,stogroup,owner,catid, volume,dev-type
INF.12	USAGE_PRIVILEGES	CATALOG,grantor,grantee, grantable,schema,object-name, object-type
INF.13.1	SPACES	CATALOG,space,owner,stogroup, logging
INF.13.2	INDEXES	CATALOG,SPACE,index,table
INF.13.3	BASE_TABLES	CATALOG,SPACE,table,table-style
INF.13.4	RECOVERY_UNITS	CATALOG,SPACE,recunit
SNF.1	SYS_CATALOGS	CATALOG
SNF.2	SYS_USERS	CATALOG,user,usershort
SNF.3	SYS_SYSTEM_ENTRIES	CATALOG,user,sysuser,host,application
SNF.4	SYS_SCHEMATA	CATALOG,schema,owner
SNF.5	SYS_TABLES	CATALOG,schema,table, table-type,space
SNF.6	SYS_COLUMNS	CATALOG,schema,table,column, data-type

Table 4: Short form names (part 4 of 6)

Short form name	Information [1]	Selection criteria (lowercase letters = optional)
SNF.7	SYS_VIEW_USAGE	CATALOG,view,schema (of the view), table,schema2 (of the table), column (of the table), object-indicator,column2 (of the view)
SNF.8	SYS_TABLE_CONSTRAINTS	CATALOG,schema,table, constraint,constraint-type
SNF.9	SYS_UNIQUE_CONSTRAINTS	CATALOG,schema,table, constraint-type,column,constraint
SNF.10	SYS_REFERENTIAL_CONSTRAINTS	CATALOG,schema,schema2, constraint, constraint2,table,table2, column,column2 without suffix “2”: of the referential constraint with suffix “2”“: of the table
SNF.11	SYS_CHECK_CONSTRAINTS	CATALOG,constraint,schema
SNF.12	SYS_CHECK_USAGE	CATALOG,schema (of the check constraint),constraint, schema2 (of the table),table,column, object-indicator,not-null-column
SNF.13	SYS_PRIVILEGES	CATALOG,grantee,object-indicator, schema,privilege,table,grantor, column,grantable
SNF.14	SYS_USAGE_PRIVILEGES	CATALOG,grantee,grantor, grantable,schema,object-name, object-type
SNF.15	SYS_SPECIAL_PRIVILEGES	CATALOG,grantee,privilege, grantor,grantable
SNF.16	SYS_INDEXES	CATALOG,index,schema,table,column
SNF.17	SYS_STOGROUPS	CATALOG,stogroup,owner,catid, volume,dev-type
SNF.18	SYS_SPACES	CATALOG,space,owner,logging, stogroup
SNF.19	SYS_RECOVERY_UNITS	CATALOG,space,recunit,archive, medium
SNF.20	SYS_DA_LOGS	CATALOG,dalogversion,dalogsub
SNF.21	SYS_MEDIA_DESCRIPTIONS	CATALOG,file-type,dev-type, medium,share

Table 4: Short form names (part 5 of 6)

Short form name	**Information** [1]	**Selection criteria (lowercase letters = optional)**
SNF.22	SYS_SPACE_PROPERTIES	CATALOG,space,lock-check-pend, lock-copy-pend,lock-recover-pend, lock-load-run,lock-is-copy,lock-is-repl, flag-opened,flag-modified,flag-defect
SNF.23	SYS_PARTITIONS	CATALOG,schema,table,serial-number
SNF.24	SYS_ROUTINES	CATALOG,schema,routine,routine-type
SNF.25	SYS_PARAMETERS	CATALOG,schema,routine,parameter
SNF.26	SYS_ROUTINE_PRIVILEGES	CATALOG,schema,routine, grantor,grantee,grantable
SNF.27	SYS_ROUTINE_USAGE	CATALOG,schema (of the routine), routine,schema2 (of the table),table, column,object-indicator
SNF.28	SYS_ROUTINE_ROUTINE_USAGE (called (sub)routines of a routine)	CATALOG,schema,routine (in each case of the calling routine), schema2,routine2 (in each case of the called routine)
SNF.29	SYS_VIEW_ROUTINE_USAGE	CATALOG,schema (of the view),view, schema2 (of the routine),routine

Table 4: Short form names (part 6 of 6)

[1] see also page 253

[2] see examples below

Syntax description for output from form INF.5.2 “Files for RECOVERY” (see short form name INF.5.2, table 4 on page 92):

```
CMD INF INF.5.2 CATALOG=<catalogname>,
{SPACE=<spacename>|CATALOG-SPACE=<file>|SPACESET|ALL SPACES}
[, [UNIT ]{LAST|FILE=<file>|NUMBER=<number>|TIMESTAMP=<timestamp>} ]
[, TYPE={TO|USING} ]
```

Examples

```
CMD INF INF.9.3.7 CATALOG='ORDERCUST',-
SCHEMA='PARTS', TABLE='ITEMS', -
SCHEMA='ORDERPROC', ROUTINE='TAKE_ORDER'
```

Output for the ITEMS table from the PARTS schema and the ORDERCUST catalog and for the referencing routine TAKE_ORDER from the ORDERPROC schema.
PARTS is the first schema here.
ORDERPROC is the second (optional) schema here.

```
CMD INF INF.9.9.8 CATALOG='ORDERCUST',-
SCHEMA='ORDERPROC', ROUTINE='WHICH_DAY',-
SCHEMA='ORDERPROC', ROUTINE='DAYSTRING’
```

Output for the called routine DAYSTRING and for the calling routine WHICH_DAY from the ORDERPROC schema and the ORDERCUST catalog. ORDERPROC is the first and the second (optional) schema here. WHICH_DAY is the first routine here; DAYSTRING is the second (optional) routine here.

The following objects can be specified as selection criteria in the form of an object list in order to restrict the output of information from the information schemata to certainrecords.
The selection criteria correspond to the input fields in the respective dialog forms or to the preset output fields in the activities INF - INFORMATION_SCHEMA and SNF - SYS_INFO_SCHEMA (INF or SNF form and their continuation forms).

Object	**Permitted values**
APPL[ICATION]	<utm applicationname>:
ARCH[IV]	<archive directory file>/<hsms archive>
CAT[ALOG]	<catalog>
CATID	<catid>
COL[UMN]	<column>
CONST[RAINT]	<simple.condition_name>
CONST[RAINT]-TYPE	PRIMARY KEY/FOREIGN KEY/UNIQUE/CHECK
DALOGS[UB]	<number>
DALOGV[ERSION]	<number>
DATA[-TYPE]	INTEGER/SMALLINT/NUMERIC/DECIMAL/FLOAT/REAL/ DOUBLE PRECISION/CHARACTER/NATIONAL CHAR/ CHARACTER VARYING/NATIONAL CHAR VARYING/ DATE/ TIME/TIMESTAMP
DEV[-TYPE]	<device_type>
FILE[-TYPE]	CATLOG/CATREC/DALOG/PBI
FLAG-DEFECT	YES/NO
FLAG-OPENED	YES/NO
FLAG-MODIFIED	YES/NO
GRANTA[BLE]	YES/NO
GRANTE[E]	<authorization_key>/PUBLIC
GRANTO[R]	<authorization_key>
HOST	<hostname>
IND[EX]	<simple indexname>
LOCK-CHECK-PEND	YES/NO
LOCK-COPY-PEND	YES/NO
LOCK-IS-COPY	YES/NO
LOCK-IS-REPL	YES/NO
LOCK-LOAD-RUN	YES/NO

Table 5: Selection criteria (part 1 of 2)

Object	Permitted values
LOCK-RECOVER-PEND	YES/NO
LOCK-REORG-PEND	YES/NO
LOG[GING]	YES/NO
MEDIUM	DISC/TAPE/HSMS/HSMW/HSMB/SRDF
NOT-NULL-COL[UMN]	Y/N
OBJECT-IND[ICATOR]	T/C
OBJECT-NAME	<simple name>
OBJECT-TYPE	STOGROUP/COLLATION/CHARACTER SET
OWN[ER]	<authorization_key>
PAR[AMETER]	<routine_parameter>
PRIV[ILEGE]	SELECT/INSERT/UPDATE/DELETE/REFERENCES/ UTILITY/USAGE/CREATE USER/CREATE SCHEMA/ CREATE STOGROUP
REC[UNIT]	<file>
ROUTINE	<simple routinename>
ROUTINE-T[YPE]	PROCEDURE/FUNCTION
SCH[EMA]	<simple schemaname>
SERIAL-NUMBER	<number>
SHARE	YES/NO
SPA[CE]	<simple spacename>
STO[GROUP]	<simple stogroupname>
SUB[NUMBER]	<number>
SYSU[SER]	<utm user_id> <bs2000 user_id>
TAB[LE]	<unqual.basetablename>/<unqual. viewname>
TABLE-S[TYLE]	NEW/OLD/OLDEST
TABLE-T[YPE]	BASE TABLE/VIEW
USER	<authorization_key>
USERS[HORT]	<authorization_key (max. 10 characters)>
VERS[ION]	<number>
VIEW	<simple viewname>
VOL[UME]	<volumename>

Table 5: Selection criteria (part 2 of 2)

Specified objects must be defined with a valid value in the form *object*='*value*'. Each of the objects specified in an object list must be separated by a comma.

If objects that are not optional according to table 4 on page 92 are not defined with a valid value, the statement is rejected with an error message.
Specification of partially qualified selection criteria is only permitted for the corresponding input fields in the forms. The placeholders “_” and “%” may not be used in the <number> specification.

i If the form contains two selection criteria of the same data type, the specified criteria are assigned in accordance with the sequence in the form.
For example, the form INF.9.4.4 could contain the object schema name (SCHEMA) both as a preset output field and as an input field.

The records are output to a default output file or to the file specified with the configuration parameter SEE-INFOUT in the configuration file SESCONF or to the file specified in the instruction file. The output file corresponds to the file created in the activity INF - INFORMATION_ SCHEMA or SNF - SYS_INFO_SCHEMA.

Example

All RECOVERY_UNITS for the user space “ordercust.customer” should be determined:

```
CMD INF INF.5.1 CAT='ORDERCUST',SPACE='CUSTOMER'
```

For the output of the CMD INF INF.5.1 statement, the utility monitor creates temporary job variables which name the files required for a RECOVER or REFRESH REPLICATION of the database. The job variables also contain the indicator “I”. This makes it possible to recover a destroyed database in batch mode using these job variables.

Example

The RECOVERY FILES (i.e. the files which are required for RECOVER) are determined for all user spaces of the catalog “ordercust”:

```
CMD INF INF.5.2 CATALOG='ORDERCUST',ALL SPACES,LAST
```

- CMD CATREC DEL_LAST_RU *catrec-file*

  This statement deletes the CAT-LOG records after the last recovery unit record. A temporary job variable with the name “#SESAM.RU. CATALOG” is thereby created, in which the COPY-NUMBER of the last recovery unit record is stored. The job variable also contains the indicator “D”.

  Processing the CMD CATREC DEL_LAST_RU *catrec-file* statement produces the same result as calling the COP.4 form, function 5.

  Example

  ```
  CMD CATREC DEL_LAST_RU ordercust.cat-rec.copy
  ```

- CMD COPJV *object*

  With the CMD COPJV *object* statement you can output the version number of the last SQL statement COPY for the corresponding spaces. The output is made to job variables. The following two *objects* can be specified:

  – CATALOG=*catalog* [,USER-SPACE=$\left\{\begin{array}{l}\textit{space}\\ \textit{(space,space}[,\textit{space},...])\end{array}\right\}$

    The version numbers of user spaces of the specified database are stored in job variables with the name “#SESAM.RU.*space*”. If only CATALOG *catalog* is specified, the version numbers of all user spaces are output.

  – CATALOG-SPACE *catrec-file*

    The version number of the catalog space is stored in the job variable with the name “#SESAM.RU.CATALOG”.

  The job variables also contain the indicator “C”.

  Processing the SQL COPY ... and CMD COPJV *object* statements produces the same result as interactively calling the COP.1 form, whereby the job variables of the user space are defined by an implicit SELECT statement.

  Example

  ```
  CMD COPJV CATALOG='ORDERCUST',
            USER-SPACE=('CUSTOMER','CONTACT','ORDSTAT')

  CMD COPJV CATALOG-SPACE ordercust.cat-rec
  ```

- CMD CHECK FORMAL CATALOG *catalog* [NO ACTION]

  You use this statement to check the formal correctness of a database or a replication. When you do this, all the spaces belonging to the database or the replication are checked. In the case of a partial replication only the user spaces contained in the partial replication can be checked by the DBH.

  If an error is detected when CHECK FORMAL is performed for a space, the SQLSTATE is entered in the log file for SQLSTATEs which is assigned with SEE-SSTLOG. The check of the other spaces is continued provided the SEE-ERROR=OFF parameter ("continue processing") is set. The NO ACTION parameter in the statement specifies that a space should not be set to defective if an error is detected.

  Executing the CMD CHECK FORMAL statement has the same effect as choosing function 5 in the CHK form.

  Example

  ```
  CMD CHECK FORMAL CATALOG ordercust
  ```

Comments

Comment lines begin with an asterisk (*) and can contain any alphanumeric string. The end of the comment is the end of the line. Comment lines must not come between continuation lines, otherwise they will be interpreted as part of the statement.

End of the statement

In batch mode, the utility monitor terminates when it finds the END statement.
In interactive mode, a message appears stating that processing of the instruction file has been completed.
If there is no END statement in the batch mode, the utility monitor terminates with a warning when it detects the end of the file. In interactive mode, a warning is issued.

3.3.1 Creating an instruction file using an editor

When creating an instruction file, please note the following points:

- The instruction file must be a SAM file.
- The statements begin in column 1. They have no length restriction. SQL comments and pragmas (beginning with “--”) also have no length restriction.
- The parameters of the configuration data contain no blanks. There can be only one parameter per line.
- A distinction is drawn between uppercase and lowercase. SQL statements are formulated in accordance with SQL conventions. See also the section entitled “SQL keywords” in the “SQL Reference Manual Part 1: SQL Statements”.
- SQL objects must be fully qualified if the SEE-CATALOG and SEE-SCHEMA configuration parameters are not preset with defaults.
- The statements are processed one after the other. Lines connected with the continuation sign (-) form a single record.

  If an SQL expression contains a minus sign at the end of the line, it is only interpreted as a minus sign if the next character is also a minus sign (≙ continuation sign).

  In other words, “-” at the end of a line is always interpreted as a continuation sign, and “--” at the end of the line is interpreted as a minus sign and a continuation sign.

- In the following cases, the continuation sign “-” is interpreted as a newline character:
  - at the end of a line in an SQL comment or pragma (beginning with “--”)
  - at the end of a line ahead of an SQL comment or pragma (beginning with “--”) provided the SQL comment or the pragma is located at the (next) start of line
- For information on comment lines, see section “Comments” on page 103.
- The “--” string should be avoided in literals and delimiters. This string is interpreted as the start of an SQL comment.

3.3.2 Creating an instruction file as a dialog log

You can also create the instruction file as the log of a dialog with the utility monitor. To do this, you must specify SEE-INST-LOGGING=ON in the configuration file or the CNF - CONFIGURATION form. The log information is written to the file specified for SEE-INPUTLOG (BS2000 file or LMS library member) or the default file SESUTI.INPUTLOG.*tsn. yyyymmddhhmmss*.

If you have specified SEE-EXECUTE=ON, the statements generated are both logged to the instruction file and executed.
If you have specified SEE-EXECUTE=OFF, the statements generated are merely logged to the instruction file.

By specifying the configuration parameter SEE-INFPROT=ON, you define that all interactive queries from the INF and SNF activities (but not interactive queries (SELECTs) from the SQL form) are logged in the instruction file.
SEE-INFPROT=OFF switches off this logging.

All SQL statements are logged, together with transaction commits set by the utility monitor, which are sent to the DBH as a result of the forms being processed.

The following information is not logged:

- SELECT statements resulting from the processing of forms
- incorrect statements
  (statements with an SQLSTATE≠00000 and statements whose SQLSTATE does not begin with “01” (i.e. `WARNING`)

You can make changes to the instruction file retroactively. In particular, you must check and, if necessary, make changes to transaction control (see also section “Creating and processing an instruction file” on page 89).

i
– The line length of the statement file is not limited.
– Literals, SQL comments, and pragmas (beginning with “--”) are not wrapped.

3.3.3 Example of information output in batch and interactive mode

In the following example, records are to be output from the INFORMATION_SCHEMA. From the ORDERCUST database, information is requested on indexes of the CUSTOMERS table located in the user space CUSTOMERS.

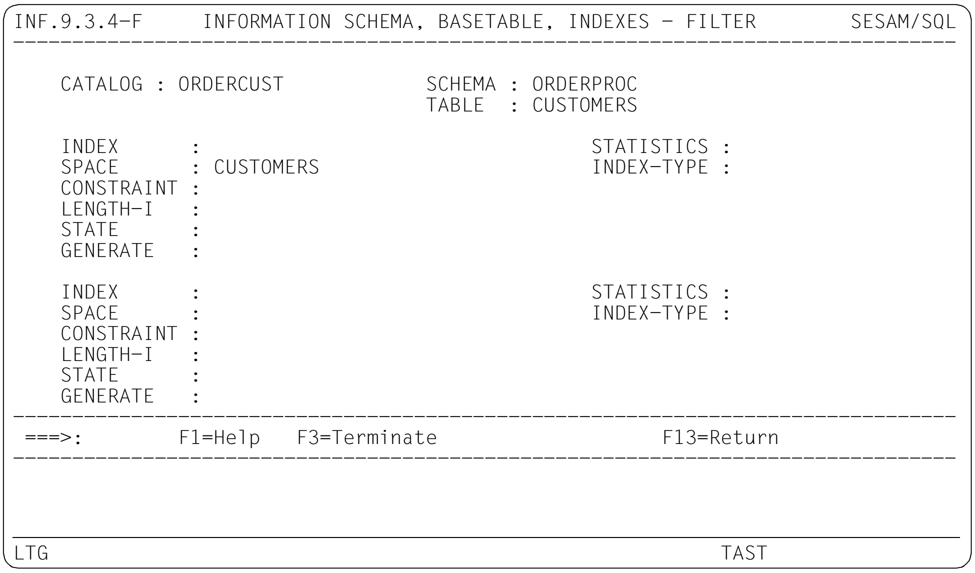
Batch mode

Excerpt from an instruction file:

```
:

* the default catalog is called ORDERCUST
SQL SET CATALOG 'ORDERCUST'
SQL COMMIT WORK

* the information output is routed to the file LIST.INF.CUSTOMERS
SEE-INFOUT=LIST.INF.CUSTOMERS

* information is to be output on indexes of the CUSTOMERS table
* which are located in the CUSTOMERS space
CMD INF INF.9.3.4
CAT='ORDERCUST',SCH='ORDERPROC',TAB='CUSTOMERS',SPACE='CUSTOMERS'
```

Interactive mode

In interactive mode, the above excerpt from an instruction file corresponds to the following entries in the INF.9.3.4-F form:



```
INF.9.3.4-F     INFORMATION SCHEMA, BASETABLE, INDEXES - FILTER       SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST            SCHEMA : ORDERPROC
                                   TABLE  : CUSTOMERS

    INDEX      :                                STATISTICS :
    SPACE      : CUSTOMERS                      INDEX-TYPE :
    CONSTRAINT :
    LENGTH-I   :
    STATE      :
    GENERATE   :

    INDEX      :                                STATISTICS :
    SPACE      :                                INDEX-TYPE :
    CONSTRAINT :
    LENGTH-I   :
    STATE      :
    GENERATE   :
--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------



LTG                                                        TAST
```

3.3.4 Example of an instruction file

This example contains sections of the INSTR.USRDAT.CREATE instruction file created in the example of database creation described in section “Creating a database” on page 19.

The authorization key is taken from the configuration file:

```
SQL SET SESSION AUTHORIZATION 'UTIADM'
SQL COMMIT WORK
```

The authorization key is changed:

```
SQL SET SESSION AUTHORIZATION 'UTIUNIV'
SQL COMMIT WORK
```

The catalog space is created and the universal user specified:

```
SQL CREATE CATALOG ORDERCUST CATALOG_SPACE SHARE DESTROY STOGROUP -
STOGROUP1 PUBLIC MEDIA STOGROUP STOGROUP2 PUBLIC USER UTIUNIV
SQL COMMIT WORK
```

Database-specific files are created:

```
SQL CREATE MEDIA DESCRIPTION FOR DALOG AT CATALOG ORDERCUST SHARE -
DEVICE REQUEST
SQL COMMIT WORK
SQL CREATE MEDIA DESCRIPTION FOR PBI AT CATALOG ORDERCUST SHARE -
DEVICE REQUEST
SQL COMMIT WORK
```

Authorization keys are created:

```
SQL CREATE USER UTIADM AT CATALOG ORDERCUST
SQL COMMIT WORK
SQL CREATE USER UTIUSR1 AT CATALOG ORDERCUST
SQL COMMIT WORK
SQL CREATE USER UTIUSR2 AT CATALOG ORDERCUST
SQL COMMIT WORK
```

System entries are created:

```
SQL CREATE SYSTEM_USER ('HOST1', , 'ID1') FOR UTIADM AT CATALOG ORDERCUST
SQL COMMIT WORK
SQL CREATE SYSTEM_USER ('HOST1', , 'ID1') FOR UTIUSR1 AT CATALOG ORDERCUST
SQL COMMIT WORK
SQL CREATE SYSTEM_USER ('HOST1', , 'ID1') FOR UTIUSR2 AT CATALOG ORDERCUST
SQL COMMIT WORK
```

Special privileges are assigned:

```
SQL GRANT ALL SPECIAL PRIVILEGES ON CATALOG ORDERCUST TO UTIADM -
WITH GRANT OPTION
SQL COMMIT WORK
```

The authorization key is changed:

```
SQL SET SESSION AUTHORIZATION 'UTIADM'
SQL COMMIT WORK
```

User spaces are created:

```
SQL CREATE SPACE CUSTOMERS NO SHARE DESTROY
SQL COMMIT WORK

SQL CREATE SPACE ORDER NO SHARE DESTROY
SQL COMMIT WORK

:  (continue with spaces CONTACT, ORDSTAT and SERVICE)
```

A schema is created:

```
SQL CREATE SCHEMA ORDERPROC
SQL COMMIT WORK
```

Tables are created:

```
SQL CREATE TABLE CUSTOMERS (-
CUST_NUM INTEGER CONSTRAINT CUST_NUM_PRIMARY PRIMARY KEY,-
COMPANY CHARACTER(040) CONSTRAINT COMPANY_NOTNULL NOT NULL,-
STREET CHARACTER(040),-
ZIP NUMERIC(05,00),-
CITY CHARACTER(040),-
COUNTRY CHARACTER(003),-
CUST_TEL CHARACTER(025),-
CUST_INFO CHARACTER(050),-
CONSTRAINT PLAUSPLZ CHECK (-
country IS NULL -
OR zip IS NULL -
OR (country = 'D  ' AND zip >= 00000) -
OR (country <> 'D  '))) -
USING SPACE CUSTOMERS
SQL COMMIT WORK
```

```
: (continue with spaces ORDERS, CONTACT, ORDSTAT and SERVICE)
```

Table privileges are assigned:

```
SQL GRANT ALL PRIVILEGES ON CUSTOMERS TO UTIUSR1 WITH GRANT OPTION
SQL COMMIT WORK

SQL GRANT ALL PRIVILEGES ON ORDERS TO UTIUSR1 WITH GRANT OPTION
SQL COMMIT WORK

: (continue with tables for CONTACT, ORDSTAT and SERVICE)

SQL GRANT SELECT ON CUSTOMERS TO UTIUSR2
SQL COMMIT WORK

SQL GRANT SELECT ON ORDERS TO UTIUSR2
SQL COMMIT WORK

: (continue with tables for CONTACT, ORDSTAT and SERVICE)
```

A backup is performed:

```
SQL COPY CATALOG ORDERCUST OFFLINE CHECK FORMAL
SQL COMMIT WORK
```

End of the statement

```
END
```

3.3.5 Processing an instruction file

The instruction file can be processed in batch or interactive mode.

Batch mode

In batch mode, you must assign the instruction file before the utility monitor is started.
If the instruction file is stored as a BS2000 file, you can assign it either with the link name SEEINPUT or via the configuration parameter SEE-INPUTFILE. If the instruction file is stored as a member of an LMS library, you must assign it via the configuration parameter SEE-INPUTFILE.
If both SEEINPUT and SEE-INPUTFILE are assigned, only SEEINPUT is evaluated. See also section “The sequence of commands for starting the utility monitor” on page 76.
If the assigned file does not exist, the utility monitor terminates with an error message.

Interactive mode

In interactive mode, you can process an instruction file by branching to the IFP - INSTRUCTION FILE PROCESSING form and entering the file name or library member name of the instruction file. See also section “Specifying an instruction file (IFP - INSTRUCTION FILE PROCESSING)” on page 243.
If you have not assigned a configuration file, or if mandatory parameters are missing, the utility monitor displays the CNF - CONFIGURATION form and requests you to enter the missing information. Only then can you make an entry in the IFP - INSTRUCTION FILE PROCESSING form.

Status display during processing

When processing of an instruction file starts, the message `SEE1124 REQUEST IN PROCESS` (abbreviated form) is issued. This message is also logged.

The message `SEE1620 CURRENT STATEMENT: <statement>` is issued when processing of a statement begins.
These messages inform you on SYSOUT about the current processing status and are not logged.

As most statements would normally only be visible for a very short time, only the following statements <*statement*> are displayed:

- <*sql_statement*>
- CMD INF ...
- CMD CATREC DEL_LAST_RU ...
- CMD COPJV ...
- CMD CHECK FORMAL CATALOG ...

Status display and interrupting processing

The BS2000 command INFORM-PROGRAM MSG=C'SEE,INFO' interrupts processing of the instruction file after the current SQL statement. The next statement which has not yet been started is displayed. Processing is continued when you press the DUE key.

If an instruction file is not being processed, a message to this effect appears.

Aborting processing

The BS2000 command INFORM-PROGRAM MSG=C'SEE,BREAK' cancels processing of the instruction file and causes an appropriate message to be displayed. However, the utility monitor waits until the current SQL statement is completed. You can also use this command to cancel file outputs (e.g. when a table derived from a SELECT statement is output to a file).

3.4 Structure of the log files

During a session, all utility monitor messages, SQLSTATEs and SQL statements are written to a SAM file with the default name SESUTI.STDLOG.*tsn.yyyymmddhhmmss*. If you want to log the different message types separately, you can explicitly assign a different file for each type by means of the SEE-MSGLOG, SEE-SQLLOG and SEE-SSTLOG configuration parameters in the configuration file, the CNF - CONFIGURATION form or the instruction file. The log files can be stored either as a BS2000 file or as an LMS library member. See also section "Define configuration data" on page 79.

By entering the abbreviation "edt" in the command area, you call the file editor EDT as a subroutine, with which you can read the log files during the dialog, see page 130. You can switch between log files during the session. Newer entries in the log files do not overwrite older ones in this case; they are added to them.

Example

```
SEE-MSGLOG=PROTMON1
:
SEE-MSGLOG=PROTMON2
:
SEE-MSGLOG=PROTMON1
```

In the above example, the messages of the utility monitor are logged first to the PROTMON1 file, then to PROTMON2, and finally to PROTMON1 again. The second set of messages logged to the PROTMON1 file are added to the first set; they do not overwrite them.

Before each entry, a time stamp and a consecutive number is set. The consecutive number always applies to a statement. In other words, the entries for SQLSTATEs, SQL statements and the corresponding messages receive the same number.
When different log files are used, entries in the different files that belong together can be recognized by the fact that they have the same consecutive number.

DBH input is represented by the character string "<<", while DBH output is represented by ">>".

For a better overview of the session, the log file specifies the length of time between the issuing of the statement and the confirmation message from the DBH. In the SQLSTATE line - in addition to the time stamp of the DBH acknowledgment - the length of time is thus specified in hours, minutes and seconds.

Each entry corresponds to a record in the log file.

In the event of syntax errors, the location of the error (line/column) is output in addition to the text of the error message for the SQLSTATEs. The line and column refer to the lines in the logged statement.

By specifying the configuration parameter SEE-SYSLST=ON you can define that data is also logged to SYSLST. In this case, the log is retained for the duration of the entire BS2000 session. It contains all the messages which are also written to the log files, i.e. the SQL statements, the SQLSTATEs, and the messages from the utility monitor.

Entries in the log files

The records in the log file are structured as follows:

$$\textit{time}\text{␣}\textit{mess\_num}\text{␣}\textit{statement\_num}\text{␣}\left\{\begin{array}{l}\texttt{<<}\\ \texttt{>>}\end{array}\right\}\text{␣}\left\{\begin{array}{l}\textit{sql-statement}\\ \textit{timespan}\text{␣}/\text{␣}\textit{sql\_state}[\text{␣}\textit{state\_text}]\end{array}\right\}$$

or

time␣*mess_num*␣*mess_text*

Entry	**Meaning**
time	Date and time in the format: *yyyymmddhhmmss*
message_number	Message number of the utility monitor
statement_number	Consecutive statement number
<<	DBH input
>>	DBH output
sql_statement	SQL statement sent to the DBH
timespan	Output of the time required by the DBH in the form *hhmmss*
sql_state	SQLSTATE returned by the DBH
state_text	SQLSTATE error message text
message_text	Message text of the utility monitor

Table 6: Entries in the log files

Example of a log file

This example contains sections of the log file created automatically when the ORDERCUST database is created using the SESUTI.USRDAT.CREATE instruction file.

The catalog space is created and the universal user specified:

```
<date> <time> SEE2100 0013 >> CREATE CATALOG AUFTRAGKUNDEN CATALOG_SPACE
SHARE DESTROY STOGROUP STOGROUP1 PUBLIC MEDIA STOGROUP STOGROUP2 PUBLIC USER
UTIUNIV
<date> <time> SEE2200 0013 << 00:01:53 / 00000
<date> <time> SEE2100 0014 >> COMMIT WORK
<date> <time> SEE2200 0014 << 00:00:00 / 00000
:
```

Authorization keys are created:

```
<date> <time> SEE2100 0019 >> CREATE USER UTIADM AT CATALOG ORDERCUST
<date> <time> SEE2200 0019 << 00:00:01 / 0000
<date> <time> SEE2100 0020 >> COMMIT WORK
<date> <time> SEE2200 0020 << 00:00:00 / 00000
<date> <time> SEE2100 0021 >> CREATE USER UTIUSR1 AT CATALOG ORDERCUST
<date> <time> SEE2200 0021 << 00:00:00 / 00000
<date> <time> SEE2100 0022 >> COMMIT WORK
<date> <time> SEE2200 0022 << 00:00:01 / 00000
<date> <time> SEE2100 0023 >> CREATE USER UTIUSR2 AT CATALOG ORDERCUST
<date> <time> SEE2200 0023 << 00:00:01 / 00000
<date> <time> SEE2100 0024 >> COMMIT WORK
<date> <time> SEE2200 0024 << 00:00:00 / 00000
```

System entries are created:

```
<date> <time> SEE2100 0025 >> CREATE SYSTEM_USER ('HOST1', ,      'ID1') FOR
UTIADM AT CATALOG ORDERCUST
<date> <time> SEE2200 0025 << 00:00:00 / 00000
<date> <time> SEE2100 0026 >> COMMIT WORK
<date> <time> SEE2200 0026 << 00:00:01 / 00000 <date> <time> SEE2100 0027 >>
CREATE SYSTEM_USER ('HOST1', ,      'ID1') FOR UTIUSR1 AT CATALOG ORDERCUST
<date> <time> SEE2200 0027 << 00:00:00 / 00000
<date> <time> SEE2100 0028 >> COMMIT WORK
<date> <time> SEE2200 0028 << 00:00:00 / 00000
<date> <time> SEE2100 0029 >> CREATE SYSTEM_USER ('HOST1', ,      'ID1') FOR
UTIUSR2 AT CATALOG ORDERCUST
<date> <time> SEE2200 0029 << 00:00:00 / 00000
<date> <time> SEE2100 0030 >> COMMIT WORK
<date> <time> SEE2200 0030 << 00:00:00 / 00000
:
```

Processing of the instruction file is terminated:

```
<date> <time> SEE1600 EXECUTION OF INSTRUCTION FILE COMPLETED WITHOUT ERROR
<date> <time> SEE2100 0082 >> COMMIT WORK
<date> <time> SEE2200 0082 << 00:00:01 / 00000
```

3.5 Specifying output files

You can have the following information written to output files:

- output from INFORMATION_SCHEMA (INF - INFORMATION-SCHEMA form) and SYS_INFO_SCHEMA (SNF - SYS-INFO-SCHEMA form)
- results of SELECT statements (SQL - SQL-STATEMENTS form)
- metadata from the CAT-REC file (COP.4 form - COPY & RECOVER, CAT-REC METADATA form) or the spaces (COP.5 form - COPY & RECOVER, SPACE METADATA form)

You assign these files in the appropriate form by specifying the file name or library member name in the function menu for information output. See also section “Entering a library member name (LIB - LIBRARY ELEMENT)” on page 265.

You can also specify the following output files:

- log files for messages, SQLSTATEs and SQL statements, via the SEE-MSGLOG, SEE-SSTLOG and SEE-SQLLOG configuration parameters
- a diagnostic trace file, by assigning the link name SEETRACE before starting the utility monitor
- an instruction file, by means of the SEE-INPUT-LOG configuration parameter
- information output file for CMD INF statements by the SEE-INFOUT configuration parameter

The output files are SAM files; new information written to them is added to the existing contents. They can be stored as BS2000 files or as members of an LMS library.

To edit the output files during the dialog, enter edt in the command area to call the file editor EDT as a subroutine, see page 130.

3.6 Output in job variables

The utility monitor creates temporary job variables. A temporary job variable enables the output of particular information about the recovery units (RU) of a database in the event of a RECOVER or REFRESH REPLICATION.

The utility monitor stores information in the following job variables:

SESAM.SESUTI.JV	State of the utility monitor (see page 138)
#SESAM.SESUTI.JV	State of a utility monitor. The temporary job variable enables the parallel monitoring of several utility monitors which run on the same system and user ID in different tasks. (Value sets as SESAM.SESUTI.JV, see page 138)
#SESAM.RU.CATALOG [1]	catalog space backup unit
#SESAM.RU.*space* [1]	user spaces backup unit
#SESAM.RU.CAT-LOG [1]	log file of catalog space backup unit
#SESAM.RU.DA-LOG [1]	log file of user spaces backup unit

[1] The layout of these job variables has changed incompatibly with SESAM/SQL V3.2, see the next page

All job variables of the utility monitor are defined in the module SEZTXT in the SESAM/SQL-module library. The associated source SEZTXT.ASS is supplied as a component of the library SIPANY.SESAM-SQL.<ver>.SPEZ. The user can therefore change the standard names of the job variables, see “Core manual”, section “Job variables”.

Recovery units for individual spaces

The job variables for the recovery units of the individual spaces are defined using a fixed structure. The job variables contain the names of the files which are required to perform a RECOVER or REFRESH REPLICATION of the user spaces or catalog spaces of a database.

The data consists of the database name, the space name (in the case of user spaces) or the character string CATALOG (in the case of catalog spaces), a version number, the time stamp, and an indicator I, C or D.

$$\mathit{catalog}.\left\{\begin{matrix}\mathit{space}\\ \text{CATALOG}\end{matrix}\right\}\text{␣}\mathit{version}\text{␣}\mathit{timestamp}\text{␣}\left\{\begin{matrix}\text{I}\\ \text{C}\\ \text{D}\end{matrix}\right\}$$

version specifies the version number of the SESAM backup copy or the COPY-NUMBER of the last recovery unit record.

The indicator is located at the 70th position in the data. It has the following meaning:

I INFORMATION: The job variable was defined after calling the INF.5.2 form or after executing the CMD INF INF.5.2... statement.

C COPY: The job variable was defined after calling the COP.1 form or after executing the CMD COPJV ... statement.

D DELETE: The job variable was defined after deleting the CAT-LOG records in the CAT-REC file after the last recovery unit record (calling COP.4 form, function 5, or executing the CMD CATREC DEL_LAST_RU ... statement).

For more information on the CMD INF, CMD COPJV and CMD CATREC DEL_LAST_RU statements, see page 91ff.

Log files of the recovery units

The data is specified in the job variables #SESAM.RU.CAT-LOG and #SESAM.RU.DA-LOG in accordance with a syntax which can be used immediately by ARCHIVE for a reconstruction. The file names comprise the database name, the version number of the recovery unit, the letter “C” for CAT-LOG or “D” for DA-LOG, and the subnumber of the log files. The job variable #SESAM.RU.DA-LOG is passed the physical database name by means of an internal administration command. This is only possible if the administration password has been entered for the SEE-ADMIN configuration parameter in the CNF configuration form. If no value has been specified for SEE-ADMIN then the logical database name is used for the job variable #SESAM.RU.DA-LOG. In this case, automatic recovery is only successful if the physical and logical names are the same.

$$\mathit{catalog}.\mathit{version}.\left\{\begin{matrix}\mathrm{C}\\\mathrm{D}\end{matrix}\right\}.\mathit{subnumber}\,,\mathrm{THRU}{=}\mathit{catalog}.\mathit{version}.\left\{\begin{matrix}\mathrm{C}\\\mathrm{D}\end{matrix}\right\}.\mathit{subnumber}$$

Examples of creating job variables

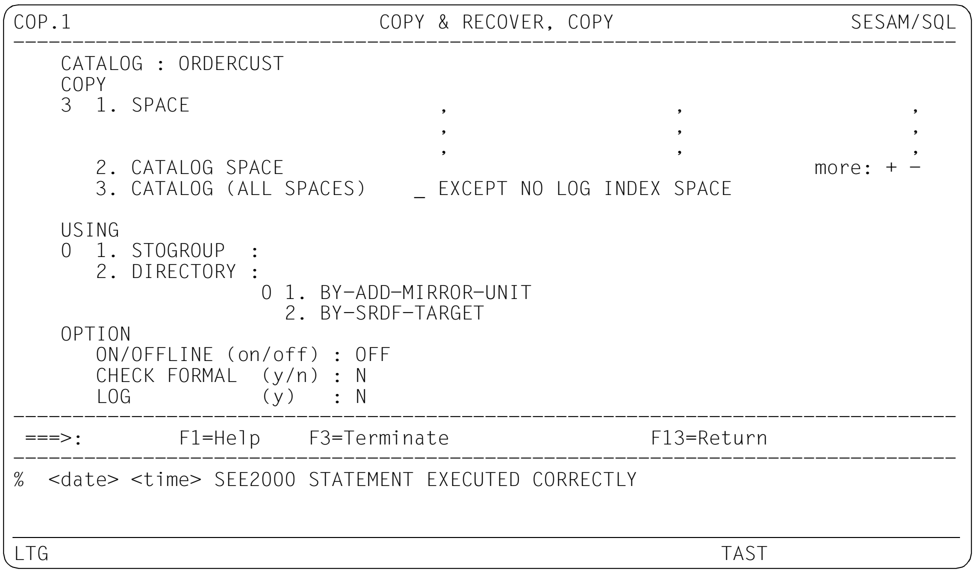
The following examples illustrate the creation of temporary job variables by the utility monitor. They do not show the continuous process nor the further processing of the job variables received, because job variables are only significant in batch mode (see the example on page 126).

Job variables are created or updated by the processing of CMD statements (see page 91ff) or by calling their corresponding forms.

Example 1

Job variables are created with each COPY (COP.1 form).

Enter “COPY CATALOG (ALL SPACES)” (function 3). The utility monitor confirms that the backup has been performed successfully.



```
COP.1                          COPY & RECOVER, COPY                   SESAM/SQL
-------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    COPY
    3  1. SPACE                        ,                   ,                   ,
                                       ,                   ,                   ,
                                       ,                   ,                   ,
       2. CATALOG SPACE                                             more: + -
       3. CATALOG (ALL SPACES)    _ EXCEPT NO LOG INDEX SPACE

    USING
    0  1. STOGROUP  :
       2. DIRECTORY :
                    0 1. BY-ADD-MIRROR-UNIT
                      2. BY-SRDF-TARGET
    OPTION
       ON/OFFLINE (on/off) : OFF
       CHECK FORMAL  (y/n) : N
       LOG           (y)   : N
-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------
%  <date> <time> SEE2000 STATEMENT EXECUTED CORRECTLY


LTG                                                         TAST
```

You can now press the K2 key to switch to command mode in order to enter BS2000 commands.

```
/
%  CMD0170 ARE COMMANDS TO BE INSERTED? ANSWER (Y=YES; N=NO)?y
%  SSM2014 TASK IN ESCAPE MODE ON LEVEL NUMBER '1'

/show-jv-attributes #
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.ORDSTAT
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.ORDERS
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.CONTACTS
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.CUSTOMERS
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.SERVICE
%SUM    00005 JV'S; JV-VALUE = 00000350 BYTES
/show-jv #sesam.ru.ordstat
%ORDERCUST.ORDSTAT                      000006 2010-07-11 09:31:57.584 C
/show-jv #sesam.ru.orders
%ORDERCUST.ORDERS                       000006 2010-07-11 09:31:57.584 C
/show-jv #sesam.ru.contacts
%ORDERCUST.CONTACTS                     000006 2010-07-11 09:31:57.584 C
/show-jv #sesam.ru.customers
%ORDERCUST.CUSTOMERS                    000006 2010-07-11 09:31:57.584 C
/show-jv #sesam.ru.service
%ORDERCUST.SERVICE                      000006 2010-07-11 09:31:57.584 C
/resume-program



LTG                                                          TAST
```

You use the SHOW-JV-ATTRIBUTES command to display a list of all the temporary job variables.

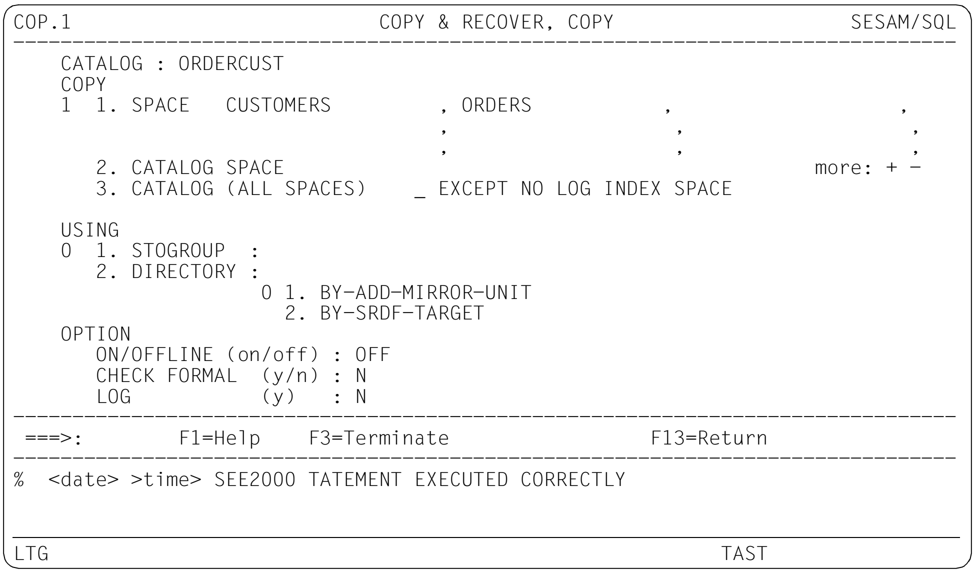
A job variable has been created for each user space:

- All spaces have been backed up six times up to this point; all have the version number “000006”.
- All spaces backed up at a particular time have the same time stamp).
- The “C” indicates that they were defined after a COPY.

Although all spaces of the catalog were backed up with function 3, i.e. including the catalog space itself, no job variable has been created for it. This is connected with the fact that the associated CAT-REC file must be evaluated for the contents of the #SESAM.RU.CATALOG job variable.

After the database has been processed (e.g. by modifying the user spaces CUSTOMER and ORDER), a SESAM backup copy should again be created of the modified spaces.

Choose function 1 for “COPY SPACE” and specify the CUSTOMER and ORDER spaces. The utility monitor confirms that the backup has been performed successfully.



```
COP.1                            COPY & RECOVER, COPY                     SESAM/SQL
-------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    COPY
    1  1. SPACE   CUSTOMERS         , ORDERS           ,                   ,
                                    ,                   ,                   ,
                                    ,                   ,                   ,
       2. CATALOG SPACE                                               more: + -
       3. CATALOG (ALL SPACES)    _ EXCEPT NO LOG INDEX SPACE

    USING
    0  1. STOGROUP  :
       2. DIRECTORY :
                    0 1. BY-ADD-MIRROR-UNIT
                      2. BY-SRDF-TARGET
    OPTION
       ON/OFFLINE (on/off) : OFF
       CHECK FORMAL  (y/n) : N
       LOG           (y)   : N
-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
-------------------------------------------------------------------------------
%  <date> >time> SEE2000 TATEMENT EXECUTED CORRECTLY


_______________________________________________________________________________
LTG                                                        TAST
```

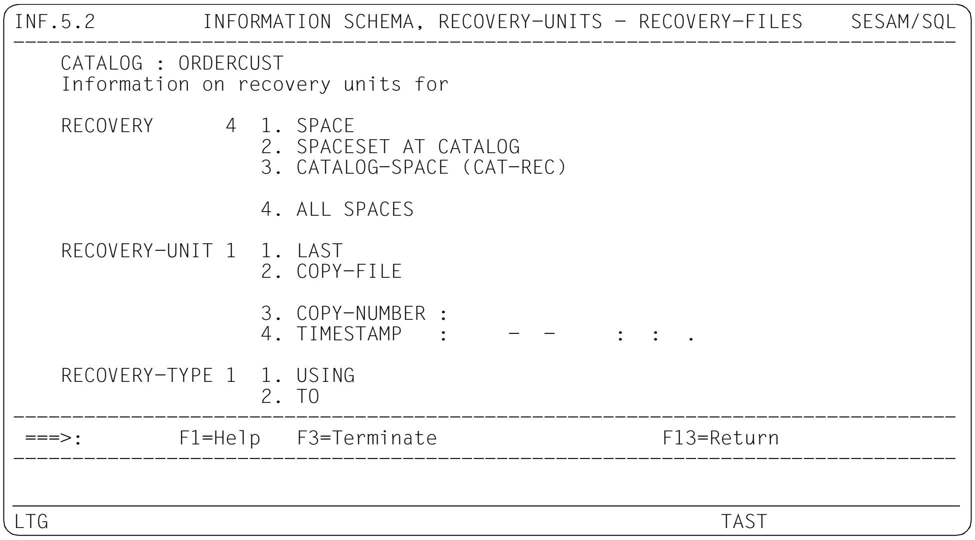
You can press the K2 key to switch to command mode.

```
/show-jv-attributes #
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.ORDSTAT
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.ORDERS
%0000060 :2OSG:$ID1.S.152.8EC7.SESAM.RU.DA-LOG
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.CONTACTS
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.CUSTOMERS
%0000068 :2OSG:$ID1.S.152.8EC7.SESAM.RU.SERVICE
%SUM    00006 JV'S; JV-VALUE = 00000422 BYTES
/show-jv #sesam.ru.da-log
%ORDERCUST.000008.D.0001       ,THRU=ORDERCUST.000008.D.0001
/show-jv #sesam.ru.customers
%ORDERCUST.CUSTOMERS                   000007 2010-07-11 09:36:52.641 I
/show-jv #sesam.ru.orders
%ORDERCUST.ORDERS                      000007 2010-07-11 09:36:52.641 I
/show-jv #sesam.ru.service
%ORDERCUST.SERVICE                     000006 2010-07-11 09:31:57.584 I
/show-jv #sesam.ru.contacts
%ORDERCUST.CONTACTS                    000006 2010-07-11 09:31:57.584 I
/show-jv #sesam.ru.ordstat
%ORDERCUST.ORDSTAT                     000006 2010-07-11 09:31:57.584 I
/delete-jv #
%  JVS0465 ALL JV'S ':2OSG:$ID1.S.152.8EC7.' DELETE USER-ID?
REPLY (Y=YES; N=NO; T=END; ,CHECK=NEW MODE)?y
/resume-program

LTG                                                         TAST
```

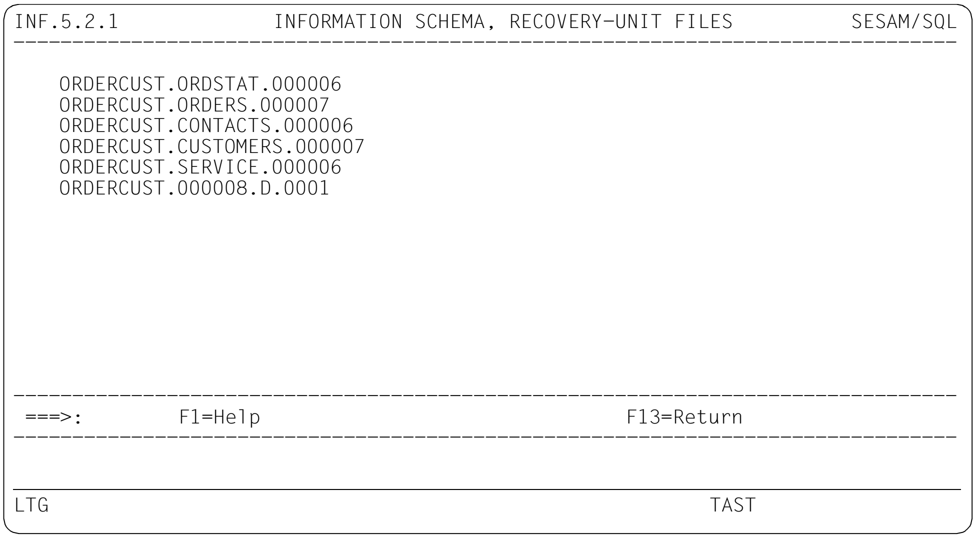
You can use the SHOW-JV-ATTRIBUTES command again to obtain a new list of all temporary job variables. The job variables of the recovery units for the user spaces CUSTOMER and ORDER have been updated. The version number has been incremented to “000007” and the time stamp adapted accordingly.

Example 2

Job variables are also defined when the INF.5.2 form is called. In this example, the files required for a RECOVER/REFRESH REPLICATION are to be output, i.e. the last ones backed up (“LAST”). The names of the files required for a RECOVER or REFRESH REPLICATION of all spaces of the database are queried: “ALL SPACES” (function 4). The INF.5.2.1 continuation form is displayed.



```
INF.5.2         INFORMATION SCHEMA, RECOVERY-UNITS - RECOVERY-FILES    SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    Information on recovery units for

    RECOVERY      4  1. SPACE
                     2. SPACESET AT CATALOG
                     3. CATALOG-SPACE (CAT-REC)

                     4. ALL SPACES

    RECOVERY-UNIT 1  1. LAST
                     2. COPY-FILE

                     3. COPY-NUMBER :
                     4. TIMESTAMP   :     - -     : :  .

    RECOVERY-TYPE 1  1. USING
                     2. TO
--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------


LTG                                                           TAST
```



```
INF.5.2.1             INFORMATION SCHEMA, RECOVERY-UNIT FILES         SESAM/SQL
--------------------------------------------------------------------------------

    ORDERCUST.ORDSTAT.000006
    ORDERCUST.ORDERS.000007
    ORDERCUST.CONTACTS.000006
    ORDERCUST.CUSTOMERS.000007
    ORDERCUST.SERVICE.000006
    ORDERCUST.000008.D.0001









--------------------------------------------------------------------------------
 ===>:        F1=Help                              F13=Return
--------------------------------------------------------------------------------


LTG                                                          TAST
```

You press the K2 key to switch to command mode.

```
/show-jv-attributes #
%0000068 :20SG:$ID1.S.152.8EC7.SESAM.RU.ORDSTAT
%0000068 :20SG:$ID1.S.152.8EC7.SESAM.RU.ORDERS
%0000060 :20SG:$ID1.S.152.8EC7.SESAM.RU.DA-LOG
%0000068 :20SG:$ID1.S.152.8EC7.SESAM.RU.CONTACTS
%0000068 :20SG:$ID1.S.152.8EC7.SESAM.RU.CUSTOMERS
%0000068 :20SG:$ID1.S.152.8EC7.SESAM.RU.SERVICE
%SUM    00006 JV'S; JV-VALUE = 00000422 BYTES
/show-jv #sesam.ru.da-log
%ORDERCUST.000008.D.0001       ,THRU=ORDERCUST.000008.D.0001
/show-jv #sesam.ru.customers
%ORDERCUST.CUSTOMERS                   000007 2010-07-11 09:36:52.641 I
/show-jv #sesam.ru.orders
%ORDERCUST.ORDERS                      000007 2010-07-11 09:36:52.641 I
/show-jv #sesam.ru.service
%ORDERCUST.SERVICE                     000006 2010-07-11 09:31:57.584 I
/show-jv #sesam.ru.contacts
%ORDERCUST.CONTACTS                    000006 2010-07-11 09:31:57.584 I
/show-jv #sesam.ru.ordstat
%ORDERCUST.ORDSTAT                     000006 2010-07-11 09:31:57.584 I
/delete-jv #
%  JVS0465 ALL JV'S ':20SG:$ID1.S.152.8EC7.' DELETE USER-ID?
REPLY (Y=YES; N=NO; T=END; ,CHECK=NEW MODE)?y
/resume-program

-------------------------------------------------------------------------------
LTG                                                          TAST
```

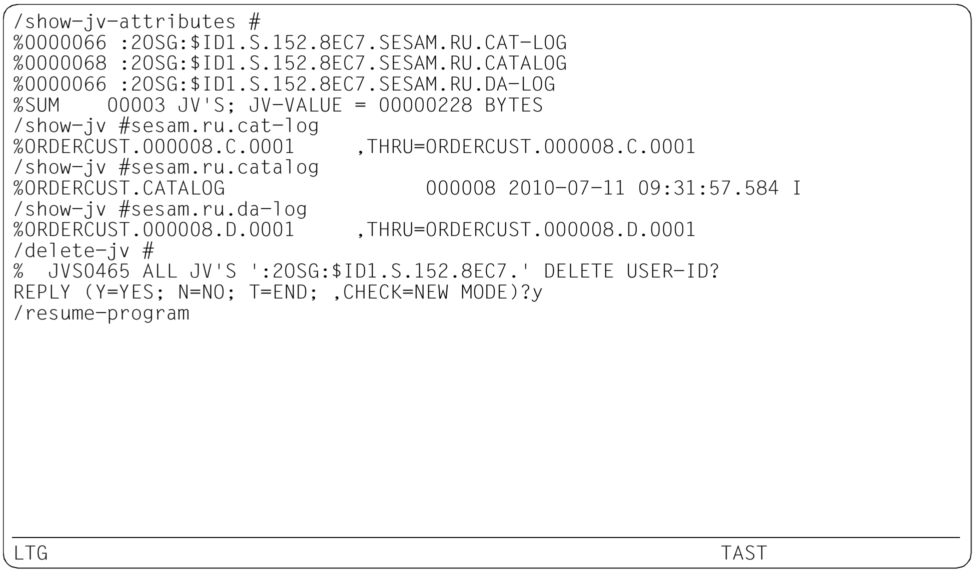
You use the SHOW-JV-ATTRIBUTES command to display a list of all the temporary job variables.

The list shows a job variable for each user space and the job variable #SESAM.RU.DA-LOG:

- The last backup of the user spaces CUSTOMER and ORDER was performed at the specified time; they have the version number “000007”.
- All other user spaces have the version number “000006” with the specified time stamp.
- The “I” indicates that they were defined by output from the information schema.
- The names of the log files required for a RECOVER/REFRESH REPLICATION of the user spaces can be taken from the #SESAM.RU.DA-LOG job variable.

Although information on all spaces of the database was queried with function 4, there is no output for the catalog space. This is again connected with the fact that a CAT-REC file must be evaluated for the contents of the #SESAM.RU.CATALOG job variable, which is not known for the function selection 1, 2 or 4 in the INF.5.2 form.

The next information output should now relate to the catalog space. Select function 3 “CATALOG-SPACE (CAT-REC)” and specify the CAT-REC file ORDERCUST.CAT-REC to send off the INF.5.2 form. The INF.5.2.1 continuation form is displayed for the output.
You can press the K2 key to switch to command mode.



```
/show-jv-attributes #
%0000066 :20SG:$ID1.S.152.8EC7.SESAM.RU.CAT-LOG
%0000068 :20SG:$ID1.S.152.8EC7.SESAM.RU.CATALOG
%0000066 :20SG:$ID1.S.152.8EC7.SESAM.RU.DA-LOG
%SUM    00003 JV'S; JV-VALUE = 00000228 BYTES
/show-jv #sesam.ru.cat-log
%ORDERCUST.000008.C.0001     ,THRU=ORDERCUST.000008.C.0001
/show-jv #sesam.ru.catalog
%ORDERCUST.CATALOG                 000008 2010-07-11 09:31:57.584 I
/show-jv #sesam.ru.da-log
%ORDERCUST.000008.D.0001     ,THRU=ORDERCUST.000008.D.0001
/delete-jv #
%  JVS0465 ALL JV'S ':20SG:$ID1.S.152.8EC7.' DELETE USER-ID?
REPLY (Y=YES; N=NO; T=END; ,CHECK=NEW MODE)?y
/resume-program









LTG                                                         TAST
```

You can again use the SHOW-JV-ATTRIBUTES command to display a new list of the temporary job variables. The job variable for the recovery unit of the catalog space and the job variable #SESAM.RU.CAT-LOG were created:

- The last backup of the catalog space was performed at the specified time; it has the version number “000008” (see page 120).
- The “I” indicates that this is output from the information schema.
- The names of the log files required for a RECOVER of the catalog space can be taken from the #SESAM.RU.CAT-LOG job variable.

Recovery in batch mode

The ability of the utility monitor to store information from the INFORMATION_SCHEMA information schema in temporary job variables makes it easier to recover a database in batch mode.

The following tasks must be performed in an appropriate BS2000 procedure:

1. All temporary job variables with the name #SESAM.RU.* are deleted before each call of the utility monitor.
2. The utility monitor is started and the first instruction file (USRDAT.1, see below) is processed: the names of the backup files for a RECOVER CATALOG-SPACE are stored in the job variables. The utility monitor is terminated.
3. If the backup files and log files no longer exist in the file catalog, they must be made available with ARCHIVE by accessing the job variables. To do this you can, for example, use the procedure PRC.RESTORE.
4. The utility monitor is restarted and the second instruction file (USRDAT.2, see below) is processed: it contains a RECOVER CATALOG-SPACE and a specification of the backup files for a RECOVER of all user spaces, which are again stored in job variables. The utility monitor is terminated.
5. If necessary, the relevant files are made available by ARCHIVE (see point 3).
6. The utility monitor is restarted and the third instruction file (USRDAT.3, see page 127) is processed: a RECOVER CATALOG is performed. The database is thus recovered in full. It contains all the modifications made up to and including the last backed up log file.

Contents of the instruction files

USRDAT.1

```
* Names of the backup files for RECOVER CATALOG-SPACE
CMD INF INF.5.2 CATALOG='ORDERCUST',-
CATALOG-SPACE='ORDERCUST.CAT-REC',-
LAST,TYPE=USING
END
```

USRDAT.2

```
* RECOVER CATALOG-SPACE
SQL RECOVER CATALOG_SPACE ORDERCUST
SQL COMMIT WORK
* Names of the backup files for RECOVER of user spaces
CMD INF INF.5.2 CATALOG='ORDERCUST',ALL SPACES,-
LAST,TYPE=USING
END
```

USRDAT.3

```
   * RECOVER CATALOG
   SQL RECOVER CATALOG ORDERCUST
   SQL COMMIT WORK
   END
```

Example procedure PRC.RESTORE

Makes available backups and log files under the user ID.

```
/BEGIN-PROCEDURE LOGGING=*ALL,PAR=YES(PROC-PAR=(-
   /   &SPACE,-
   /   &VERSION,-
   /   &LOGS),ESC-CHAR=C'&')
   /ASSIGN-SYSDTA TO-FILE=*SYSCMD
   /START-ARCHIVE
   FILES NAME = (&SPACE..&VERSION)
   FILES NAME = (&LOGS)
   RESTORE DIR=ARCHIVE.DIR,DEVICE=TAPE-C6
   END
   /END-PROCEDURE
```

Example

The starting point for this example is a disk backup of the database which was then subsequently backed up to magnetic tape cartridge along with the associated CAT-LOG and DA-LOG files using ARCHIVE.

In the BS2000 procedure listed below, the utility monitor stores the names of the save files in temporary job variables. ARCHIVE uses these job variables to make the files available under the user ID. In the final step, the database is recovered.

```
/BEGIN-PROCEDURE LOGGING=*ALL
/ASSIGN-SYSDTA TO-FILE=*SYSCMD
/ADD-FILE-LINK LINK-NAME=SEEINPUT,FILE-NAME=ANWDAT.1
/ADD-FILE-LINK LINK-NAME = SESCONF,FILE-NAME=SESCONF.SESUTI.ZX
/DELETE-JV #SESAM.RU*
/SET-JOB-STEP
/" READOUT OF JOB VARIABLES "
/START-SESAM-UTILITY-MONITOR
/DELETE-JV #CATALOG.VERS
/SET-JOB-STEP
/CREATE-JV #CATALOG.VERS
/MODIFY-JV JV=#CATALOG.VERS,-
/  SET-VALUE = *SUBSTRING (JV-NAME=#SESAM.RU.CATALOG,-
/  POSITION=39,LENGTH=4)
/SET-JOB-STEP
/" READ IN THE CATALOG AND THE CAT-LOGS "
/CALL-PROCEDURE FROM-FILE=PRC.RESTORE,-
/   PROCEDURE-PARAMETERS = (ORDERCUST.CATALOG,-
/   '&(#CATALOG.VERS)',-
/   '&(#SESAM.RU.CAT-LOG)')
/" RECOVER CATALOG_SPACE "
/" NAMES OF THE SAVE FILES FOR USER SPACES "
/ADD-FILE-LINK LINK-NAME=SEEINPUT,FILE-NAME=USRDAT.2
/START-SESAM-UTILITY-MONITOR
/DELETE-JV #CUSTOMERS.VERS
/SET-JOB-STEP
/CREATE-JV #CUSTOMERS.VERS
/MODIFY-JV JV=#CUSTOMERS.VERS,-
/  SET-VALUE = *SUBSTRING (JV-NAME=#SESAM.RU.CUSTOMERS,-
/  POSITION=39,LENGTH=4)
/SET-JOB-STEP
/" READ IN THE USER SPACES CUSTOMERS AND DALOGS "
/CALL-PROCEDURE FROM-FILE=PRC.RESTORE,-
/   PROCEDURE-PARAMETERS = (ORDERCUST.CUSTOMERS,-
/   '&(#CUSTOMERS.VERS)',-
/   '&(#SESAM.RU.DA-LOG)')
/" THE OTHER USER SOURCES ARE READ IN THE SAME WAY"
/" ... "
/" THE DALOGS ONLY HAVE TO BE READ IN ONCE"
/" RECOVER CATALOG "
/ADD-FILE-LINK LINK-NAME=SEEINPUT,FILE-NAME=USRDAT.3
/START-SESAM-UTILITY-MONITOR
/END-PROCEDURE
```

3.7 Specifying access authorization

In order to work with the utility monitor, you must specify your access authorization to the DBH by entering an authorization key for the SEE-AUTHID configuration parameter. There are three ways to do this:

– in the SESCONF configuration file (see page 85)
– in an instruction file (see page 87)
– in the CNF - CONFIGURATION form (see page 86)

The utility monitor always works under the authorization key you specify and does not question your right to use it.
However, depending on how access protection is organized, you may need several authorization keys in a single session in order to carry out different tasks with the utility monitor.
However, if you are not authorized to work with the specified authorization key, an error message is output when you issue the first statement under this authorization key.
You can also change the authorization key by issuing the SQL statement SET SESSION AUTHORIZATION in the SQL form. The modified value is only valid with respect to the DBH and only in this function; the preset configuration data does not change.
In interactive mode, you change to a different authorization key by specifying it in the CNF - CONFIGURATION form.
When processing an instruction file, you can change authorization keys as often as you like by changing the value specified for the SEE-AUTHID parameter in the file. However, it is only possible to do this when the utility monitor does not have any transactions open.
You must know the administration password in order to output certain information from the INFORMATION_SCHEMA information schema. In addition, the physical database name is passed to the job variable #SESAM.RU.DA-LOG by means of an internal administration command. The physical database name is required for the purposes of automatic recovery.
You can specify the administration password in the configuration file or in the CNF - CONFIGURATION form by entering a value for the SEE-ADMIN configuration parameter.

Administration via CALL DML

If you want to carry out administration work via the CALL DML interface, you can call the SESADM administration program from the STM - START MENU form. You then have to enter an administration password in SESADM. See the DBH option ADMINISTRATOR in the “Database Operation” manual.
You can change the SEE-ADMIN configuration parameter during the dialog.

3.8 Call file editor EDT as a subroutine

Using EDT as a subroutine enables you, for example, to read log files (see page 112) or output files (see page 116) of the utility monitor during a dialog.

i At present EDT V17.0 (with Unicode) is called.
The line length is not limited.

Calling EDT in the utility monitor

The file editor EDT can be called as a subroutine from any form of the utility monitor by specifying the abbreviation “edt” in the command area.

Before EDT is called, all the files opened by the utility monitor are closed. They can then be opened in EDT for editing.

EDT is called by the utility monitor in interactive mode.

After EDT has been terminated (@HALT), the utility monitor files closed beforehand are opened again.

Coded character sets in EDT

In SESAM/SQL, EDT is called in a mode that is compatible with EDT V16.6.

The following EDT statements are important in conjunction with coded character sets:

– @STATUS=CCS outputs the coded character set (CCSN) currently selected.

– @SHOW CCS outputs a list of the coded character sets which are possible in the system. In addition to the CCS name, details of whether the coded character set can be displayed on the terminal are also output.

– @CODENAME selects the required coded character set in EDT.

A detailed description of the EDIT statements is provided in the manual “EDT (BS2000)”.

3.9 Starting the utility monitor as a subroutine

The utility monitor can be called as a subroutine of Assembler, COBOL or C application programs by means of the SEEUPA function. Processing can be carried out in interactive or batch mode from the following entry points:

- the STM - START MENU form (in interactive mode)
- the COP - COPY & RECOVER / REPLICATION form (in interactive mode)
- an instruction file (in batch mode)

It is not possible to call the utility monitor as a subroutine of UTM program units or from DRIVE.

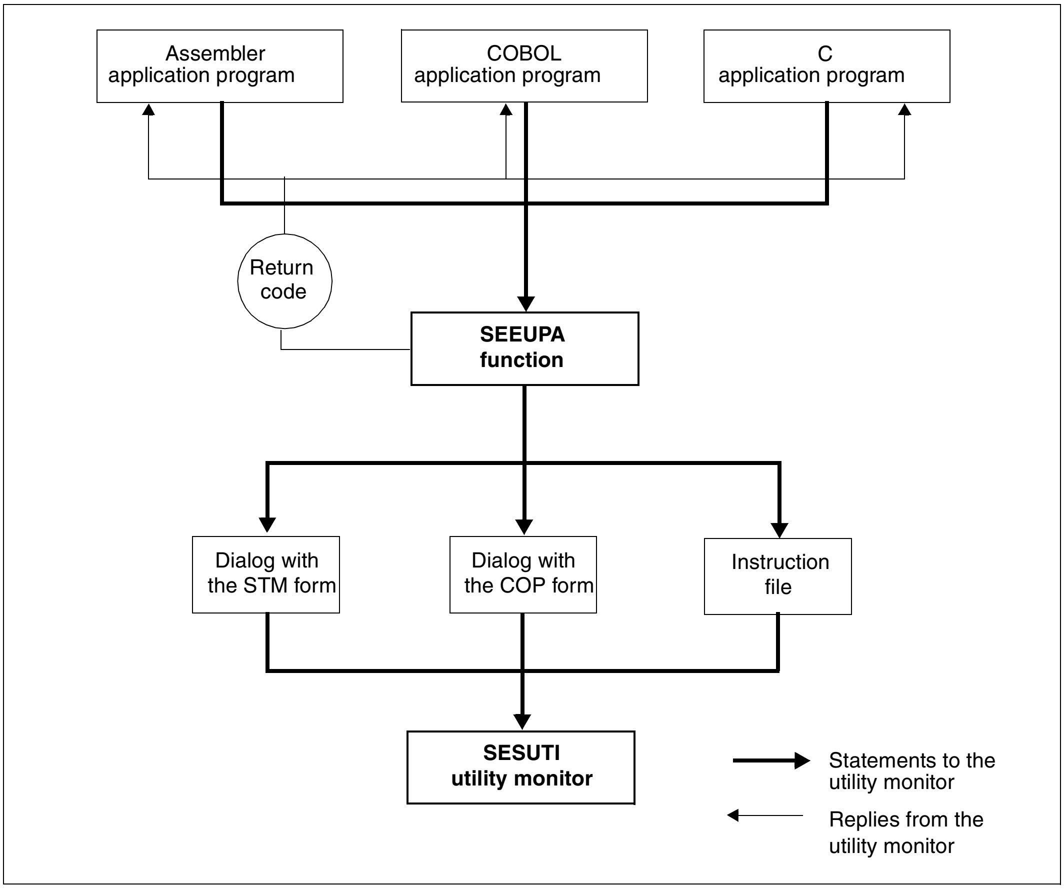


Figure 3: Starting the utility monitor as a subroutine

The application program calls the SEEUPA function, which in turn calls the utility monitor. The utility monitor dynamically loads the required modules, establishes the link to the DBH and processes the desired function. SEEUPA then sends a return code to the application program. The utility monitor communicates with the calling application program exclusively by means of return codes. It can issue the following return codes to the calling application program:

␣	The request has been carried out successfully.
W	The request has been carried out and a warning issued.
E	The request has not been carried out, or not fully, as a result of an error.

To start the utility monitor in interactive mode, you must specify the start form in the application program.
The values STM and COP are permitted, for the STM - START MENU and COP - COPY & RECOVER / REPLICATION forms respectively. If you enter a different value, processing is aborted with an error message.

To start the utility monitor in batch mode, you must specify an instruction file in the application program. The instruction file can be stored as a BS2000 file or as an LMS library member.

The following parameters must be reserved with the specified lengths in the application program:

Parameter	**String length**	**Meaning**
retcode	1	1 character must be reserved for the return code.
usrdat	195	195 characters must be reserved for the instruction file.
formnum	3	3 characters must be reserved for the start form.

Table 7: Reserving parameters in the application program

The following table shows the different ways of combining the parameters in the application program and the effects of the different combinations.

usrdat	**formnum**	**Effect**
instruction_file	-	The instruction file *instruction_file* is processed.
Not specified	STM	The utility monitor is started in interactive mode with the STM - START MENU form as an entry point.
Not specified	COP	The utility monitor is started in interactive mode with the COP - COPY & RECOVER / REPLICATION form as an entry point.

Table 8: Possible parameter combinations (part 1 of 2)

usrdat	**formnum**	**Effect**
Not specified	not STM or COP	Processing is aborted with an error message.
instruction_file	STM or COP	The instruction file *instruction_file* is processed; masknr is ignored

Table 8: Possible parameter combinations (part 2 of 2)

The three examples that follow illustrate how the utility monitor is called in application programs as a subroutine.

Example 1: Calling the utility monitor in an Assembler application program

The utility monitor is called with ILCS-capable macros of ASSEMBH. The parameter area is called by means of the @PAR macro. SEEUPA is called by means of the @PASS macro.

```
:
OK       EQU   CL1' '
WARNING  EQU   CL1'W'
ERROR    EQU   CL1'E'

* Set parameters
         MVC   RC,=OK
         MVC   DATA,=CL195' '
         MVC   FORM,=C'STM'

* Call "SEEUPA"
         @PASS EXTNAME=SEEUPA,PAR=PARLST

PARLST   @PAR  PLIST=((3)),VLIST=(RC,DATA,FORM)

* Return code
RC       DS    CL1

* Instruction file name
DATA     DS    CL195
* Start mask name (STM or COP)
FORM     DS    CL3
:
```

Example 2: Calling the utility monitor in a COBOL application program

```
 01 retcode         pic x.
* Return code
 01 data            pic x(195).
* Instruction file name
 01 form            pic x(3).
* Start mask name (STM or COP)
:
 CALL "SEEUPA" USING retcode, data, form.
* Call "SEEUPA"
:
```

Example 3: Calling the utility monitor in a C application program

```
char *retcode; /* Return code */
char *data;  /* Instruction file name */
char *form;  /* Start mask name (STM or COP) */

/* Call SEEUPA */
void SEEUPA (retcode, data, form);
```

Linking and starting the application program

When you link the application program, you do not have to include any SESAM/SQL or utility monitor components. In the following example, a C application program is linked:

Example

```
/START-BINDER
//START-LLM-CREATION INTERNAL-NAME=application_program
//INCLUDE-MODULES LIBRARY=application_lib,ELEMENT=c_module
//SAVE-LLM LIB=application_lib
//END
```

application_program
Name of the application program

application_lib
Name of the application library

c_module
User-specific C modules

Before you start the application program, you must assign the following files to the appropriate link names:

```
[/ADD-FILE-LINK,LINK-NAME=SESCONF,FILE-NAME=configuration_file
     ,ACCESS-METHOD=SAM]
```
or
```
[/CONNECT-SESAM-CONFIGURATION TO-FILE=global_configuration_file
     ,CONFIGURATION-LINK=linkname] ————————————————————————————————————   (1)
/ADD-FILE-LINK LINK-NAME=MAPLIB,FILE-NAME=fhs_lib ————————————————————   (2)
/ADD-FILE-LINK LINK-NAME=SESAMOML,FILE-NAME=sesam_modlib——————————————   (3)
/ADD-FILE-LINK LINK-NAME=SEEHELP,FILE-NAME=help_text_file ————————————   (4)
/ADD-FILE-LINK LINK-NAME=BLSLIB01,FILE-NAME=crte_lib —————————————————   (5)
/ADD-FILE-LINK LINK-NAME=BLSLIB02,FILE-NAME=sesam_modlib——————————————   (6)
```

(1) You can assign a configuration file with the link name SESCONF. The configuration file must be a SAM file (see the “Core manual”, configuration file). It is also possible to combine local configuration files in a global configuration file (see the “Core manual”, global configuration file). In this case you assign the configuration file using the CONNECT-SESAM-CONFIGURATION command.

configuration_file
User-defined name of the configuration file

configuration_file.global
User-defined name of the global configuration file

linkname
Link name under which the DBH options are defined in the global configuration file.

See also section “Entering configuration data” on page 85.

(2) You must assign the FHS library for the FHS forms module in the German or English language with the link name MAPLIB.

fhs_lib
User-defined name of the FHS library
Default name: SYSFHS.SESAM-SQL.<ver>.UTI.{D/E}
Suffix D: The forms will be output in the German language.
Suffix E: The forms will be output in the English language.

(3) You must assign the SESAM/SQL module library with the link name SESAMOML.

sesam_modlib
SESAM/SQL module library.
(SYSLNK.SESAM-SQL.<ver> for /390 servers, SKULNK.SESAM-SQL.<ver> for x86 servers).

(4) You must assign the help text file to the forms in the German or English language using the link name SEEHELP. The help text files are ISAM files.

help_text_file
User-defined name of the help text file
Default name: SYSMAN.SESAM-SQL.<ver>.UTI.{D/E}
Suffix D: The help text files will be output in the German language.
Suffix E: The help text files will be output in the English language.

See also section “Requesting help information on forms” on page 154.

(5) You must assign the CRTE library with the link name BLSLIB01.

crte_lib
Name of the CRTE runtime library from which the runtime modules of the compiler are prepared.
Default names:
$.SYSLNK.CRTE for /390 servers or $.SKULNK.CRTE for x86 servers

(6) The SESAM/SQL module library must be assigned with the link name BLSLIB02.

sesam_modlib
SESAM/SQL module library.
(SYSLNK.SESAM-SQL.<ver> for /390 servers or SKULNK.SESAM-SQL.<ver> for x86 servers).

The application program is then started with one of the following statements:

usually:

```
/START-EXECUTABLE-PROGRAM FROM-FILE=
   *LIBRARY-ELEMENT(LIBRARY=application_lib,ELEMENT-OR-SYMBOL=application_program) -
   ,DBL-PARAMETERS=(RESOLUTION=(ALTERNATE-LIBRARIES=*BLSLIB##) -
                    ,ERROR-PROCESSING=(UNRESOLVED-EXTRNS=*DELAY) -
                    ,LOADING=(LOAD-INFORMATION=*REFERENCES))
```

for compatibility reasons:

```
/START-PROGRAM FROM-FILE=
   *MODULE(LIBRARY=application_lib,ELEMENT=application_program) -
   ,PROGRAM-MODE=ANY,RUN-MODE=ADVANCED(ALTERNATE-LIBRARIES=YES -
   ,UNRESOLVED-EXTRNS=DELAY,LOAD-INF=REF))
```

3.10 Terminating the utility monitor

In interactive mode, you terminate the utility monitor by pressing the F3 key or entering F3 in the command area. You must then confirm your input by pressing the F3 key again or by re-entering F3 in the command area.

You can only terminate processing from forms that have F3=Terminate in the command area. During an activity (see page 139), you can terminate the utility monitor in the following ways:

- You can enter “stm” in the command area. This terminates all the forms, and the utility monitor returns to the STM - START MENU start form.
- You can terminate the current processing step by pressing the F13 key or entering F13 in the command area and going back through the forms until you reach one in which it is possible to enter F3.
- You can press the F12 key or enter F12 in the command area. This cancels the current form and the entries made in it. You can then press the F3 key or enter F3 in the command area to terminate the utility monitor.

When processing an instruction file in batch mode, the utility monitor terminates when it finds the END statement. If there is no END statement, a warning is issued and the utility monitor terminates at the end of the file.

If no error has occurred, the utility monitor reports error-free termination.

Behavior in the event of abnormal program termination or an error

If an error has occurred, the utility monitor terminates with an error message.

In batch mode process switches 11 and 12 are set at the end of the program under the following circumstances.

Switch 11:	The utility monitor has received an error message from the DBH of the form SEWxxxx [1].
Switch 12:	The utility monitor is terminated prematurely. This could have been caused by one of the following, amongst other causes: – the configuration data is incorrect – a statement/instruction in the instruction file is incorrect – the DBH sends an error message of the form SEWxxxx [1] and the configuration switch SEE-ERROR (see page 79) is set to ON.

The SESAM.SESUTI.JV and #SESAM.SESUTI.JV job variables are supplied with one of the following values:

```
0:timestamp:UTILITY-MONITOR END WITHOUT ERROR

1:timestamp:UTILITY-MONITOR END WITH ERROR

2:timestamp:UTILITY-MONITOR IN DIALOG MODE

3:timestamp:UTILITY-MONITOR IN INSTRUCTION-FILE-PROCESSING MODE
```

After being terminated involuntarily (e.g. after a system crash or power failure), the database may be in an inconsistent state from the user's viewpoint. In the log file with the default name SESUTI.STDLOG*.yyyymmddhhmmss*, you can see which processing steps have been carried out and which have not. You can then decide whether you want to repeat only the statements that have not been executed or whether you want to restart from an earlier database status.

For the purpose of troubleshooting you can use the diagnostic trace, which is logged in the default file SESUTI.TRACE.*yyyymmddhhmmss* (see the description of the SEE-TRACE configuration parameter on page 84).

[1] xxxx is not null and does not begin with 01 when this is the case. This means that the message is not a positive acknowledge or a warning.

4 Form layout and handling

The functions of the utility monitor are offered to you at the user interface in **forms**.

Self-contained functions which may involve the processing of several related forms are called **main functions**. You can call these from the STM - START MENU start form.

Further functions can be executed in the forms of the main functions. Either these functions are executed immediately, or the utility monitor branches to one or more **continuation forms** for further processing.

Functions in which a related set of database objects (database metadata, schema, table) are created or modified are known as **activities**.

The forms of the utility monitor can be called in a predefined hierarchy or, in many cases, by addressing them directly.

The table below lists the main functions of the utility monitor and the forms via which they can be executed. The forms and continuation forms involved are described in detail in section "Main functions and their continuation forms" on page 187.

Form	Function	Meaning
ADT	ALTER DATA	Shuffle column values, anonymize user data
ALC	ALTER CATALOG	Modifies the database's metadata.
ALS	ALTER SCHEMA	Alter a schema
ALT	ALTER TABLE	Modify a base table
CHK	CHECK	Carries out checks.
CNF	CONFIGURATION	Entering configuration data
COP	COPY & RECOVER / REPLICATION	Backs up and recovers spaces; queries and deletes metadata on backup resources; modifies backup tables; obtains information on backup copies; creates and updates replications

Table 9: Main functions of the utility monito (part 1 of 2)

Form	Function	Meaning
CRC	CREATE CATALOG	Create catalog space
CRS	CREATE SCHEMA	Create a schema
CRT	CREATE TABLE	Create a base table
EXP	EXPORT TABLE	Exports a base table.
HLP	HELP	Call the help function
IDE	DELIMITER IDENTIFIER	Specifies a delimiter identifier; can only be called by entering an exclamation mark (!) in any input field and then pressing F2 or entering F2 in the command area
IFP	INSTRUCTION FILE PROCESSING	Specifies an instruction file to be processed in interactive mode.
IMP	IMPORT TABLE	Imports a base table.
INF	INFORMATION-SCHEMA	Queries metadata from INFORMATION_SCHEMA.
LIB	LIBRARY ELEMENT	Specifies a library member name; can only be called by entering an exclamation mark (!) in an input field for a file name and then pressing F2 or entering F2 in the command area.
LOD	LOAD	Loads a base table with data from a file.
MIG	MIGRATE	Converts a SESAM/SQL V1 database to a SESAM/SQL table or table type of the current version
	SESADM	Calls the SESADM administration program; can only be called from the function menu in the STM start form
SNF	SYS-INFO-SCHEMA	Queries metadata from SYS_INFO_SCHEMA; can only be called by entering snf in the command area
SQL	SQL-STATEMENTS	Issues dynamically compilable SQL statements
SSL	SSL	Controls the management of storage space.
STM	START MENU	This is the start form (call main functions and SESADM).
ULD	UNLOAD	Unloads data from a base table or a view into a file.

Table 9: Main functions of the utility monito (part 2 of 2)

The following sections provide an overview of the structure of the forms and describe the various ways of calling them.

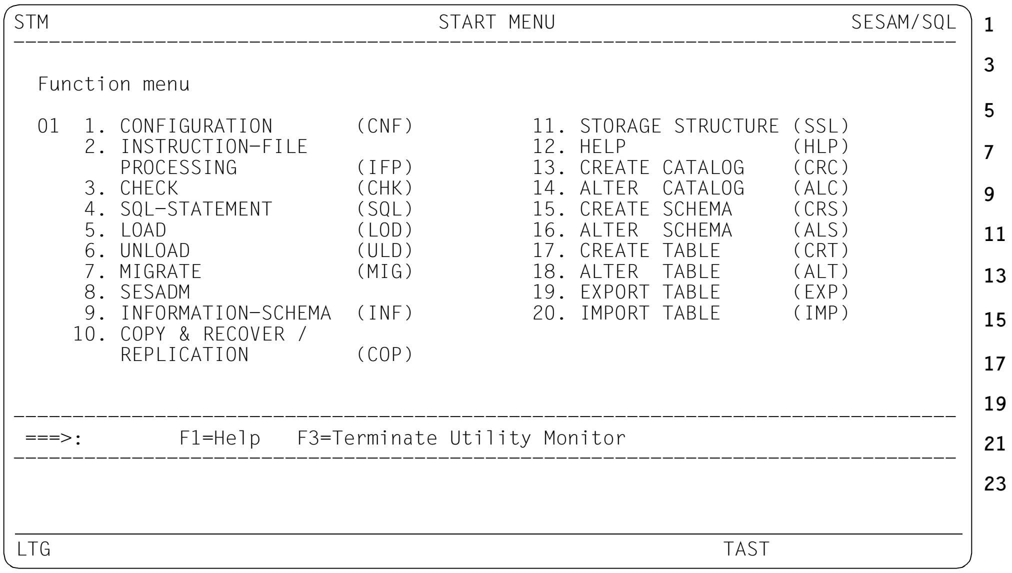
4.1 Form layout

The forms of the utility monitor consist of 24 lines x 80 columns and are divided from top to bottom into the following areas:

- the status area (line 1)
- the work area (lines 3 - 19)
- the command area (line 21)
- the message area (line 23)

The different areas are separated from each other by continuous lines.

Example



```
STM                                 START MENU                           SESAM/SQL  1
-------------------------------------------------------------------------------
                                                                                    3
  Function menu
                                                                                    5
  01  1. CONFIGURATION       (CNF)          11. STORAGE STRUCTURE (SSL)
      2. INSTRUCTION-FILE                   12. HELP              (HLP)             7
         PROCESSING          (IFP)          13. CREATE CATALOG    (CRC)
      3. CHECK               (CHK)          14. ALTER  CATALOG    (ALC)             9
      4. SQL-STATEMENT       (SQL)          15. CREATE SCHEMA     (CRS)
      5. LOAD                (LOD)          16. ALTER  SCHEMA     (ALS)             11
      6. UNLOAD              (ULD)          17. CREATE TABLE      (CRT)
      7. MIGRATE             (MIG)          18. ALTER  TABLE      (ALT)             13
      8. SESADM                             19. EXPORT TABLE      (EXP)
      9. INFORMATION-SCHEMA  (INF)          20. IMPORT TABLE      (IMP)             15
     10. COPY & RECOVER /
         REPLICATION         (COP)                                                  17

-------------------------------------------------------------------------------     19
 ===>:        F1=Help   F3=Terminate Utility Monitor                                21
-------------------------------------------------------------------------------
                                                                                    23

_______________________________________________________________________________
LTG                                                           TAST
```

4.1.1 The status area

The status area (line 1) contains the form's short name (e.g. ALT), its full name (e.g. ALTER TABLE) and the product name. No entries can be made here.

The **short name** consists of three characters that indicate the function to which the form belongs. Each form has one, e.g. ALT for ALTER TABLE.

The short names of continuation forms also include an indexed number indicating the position of the form in the call hierarchy of the function to which the form belongs. The number generally also corresponds to the number in the function menu of the preceding form.

Example

You can call the CRT - CREATE TABLE form from the STM - START MENU start form. When you select function 1 from the CRT form, the CRT.1 continuation form appears. When you select function 1 from the CRT.1 continuation form, the CRT.1.1 continuation form appears, and so on.

The form's short name also defines the entry point when the utility monitor is called as a subroutine (see page 131) and allows errors to be pinpointed exactly.

Each form or continuation form is assigned a **form name** that indicates its function.

Example

CRC form	CREATE CATALOG
CRC.1 continuation form	CREATE CATALOG, CREATE MEDIA-DESCRIPTION

The form name can also include the name of the previous form in the call hierarchy. In the case of an activity, the names of the previous form and the current form are separated by an arrow.

Example

ALTER SCHEMA −−> CREATE TABLE

The **product name** is always SESAM/SQL.

4.1.2 The work area

The work area (lines 3 - 19) contains the functions and their parameters. The database, schema and table names may already be entered when the form appears.

In the work area, you can select functions and make entries for them.

See also section “Input and output” on page 146.

4.1.3 The command area

The command area (line 21) allows you to control the processing of the forms.

The utility monitor displays the key assignments and available input options. If your terminal does not have the relevant function keys, you can simulate these by entering the name of the key in the command area instead.

Key assignments

F1	Call the help function
F2	Calls the IDE form (after you have entered an exclamation mark (!) in any input field) or the LIB form (after you have entered an exclamation mark (!) in any input field for a file name).
F3	Terminates the utility monitor or the help function.
F6	Output configuration file
F7	Pages backward in a table.
F8	Pages forward in a table.
F12	Cancels the current function.
F13	Terminates the current form and displays the previous form.
F19	Scrolls to the left in a field.
F20	Scrolls to the right in a field.

The F1, F3, F12 and F13 keys are displayed.

Only those keys permitted in the current form are displayed. It is not possible to terminate the utility monitor from every form, for example.

Entries in the command area

You can make the following entries in the command area, just to the right of the arrow (===>:):

Entry	**Meaning**
adt	Calls the ADT - ALTER DATA form
chk	Calls the CHK - CHECK form.
cnf	Calls the CNF - CONFIGURATION form.
cop	Calls the COP - COPY & RECOVER / REPLICATION form.
edt	Call file editor EDT as a subroutine
exec	Activate or deactivate logging of statements in the instruction file.
exp	Calls the EXP - EXPORT TABLE form.
F1	Calls the help function; ≙ pressing the F1 key
F2	1. Calls the IDE - DELIMITER IDENTIFIER form; has the same effect as pressing the F2 key Before you do this, you must enter “!” in any field 2. Calls the LIB - LIBRARY ELEMENT form; has the same effect as pressing the F2 key Before you do this, you must enter “!” in an input field for a file name
F3	Terminates the utility monitor or the help function; has the same effect as pressing the F3 key
F6	Outputs the configuration file; has the same effect as pressing the F6 key
F7	Pages backward in a table; has the same effect as pressing the F7 key or entering “-”
F8	Pages forward in a table; has the same effect as pressing the F8 key or entering “+”
F12	Cancels the current function; has the same effect as pressing the F12 key
F13	Terminates the current form and displays the previous form; has the same effect as pressing the F13 key
F19	Scrolls to the left in a field; has the same effect as pressing the F19 key or entering “<”
F20	Scrolls to the right in a field; has the same effect as pressing the F20 key or entering “>”
hoa	Calls help on an activity.
hlp	Calls the HLP - HELP form.

Table 10: Entries in the command area (part 1 of 2)

Entry	**Meaning**
hmp	Displays the current form position.
ilog	Switches logging on or off in the instruction file.
imp	Calls the IMP - IMPORT TABLE form.
inf	Calls the INF - INFORMATION-SCHEMA form.
lod	Calls the LOD - LOAD form.
snf	Calls the SNF - SYS-INFO-SCHEMA form.
sql	Calls the SQL - SQL-STATEMENTS form.
ssl	Calls the SSL - SSL form.
stm	Calls the STM - START MENU start form.
tr0 [1]	Switches off the diagnostic trace.
tr1 [1]	Switches on diagnostic trace level 1.
tr2 [1]	Switches on diagnostic trace level 2.
uld	Calls the ULD - UNLOAD form.
>	Scrolls to the right in a field; has the same effect as pressing the $\boxed{\text{F20}}$ key or entering F20
>>	Displays the next form belonging to the same level (e.g. the LOD.1 form)
<	Scrolls to the left in a field; has the same effect as pressing the $\boxed{\text{F19}}$ key or entering F19
<<	Displays the previous form belonging to the same level (e.g. the LOD.1 form)
+	Pages forward in a table; has the same effect as pressing the $\boxed{\text{F8}}$ key or entering F8
-	Pages backward in a table; has the same effect as pressing the $\boxed{\text{F7}}$ key or entering F7
m+	Pages forward in the message area.
m-	Pages backward in the message area.
? [2]	Displays help forms that list all the possible entries (the first part is in the HLP.CMD form, the second part in the HLP.CMD.1 continuation form).

Table 10: Entries in the command area (part 2 of 2)

[1] See the description of the diagnostic trace on page 84

[2] ? must be sent off with the DUE key

4.1.4 The message area

The message area (lines 23 - 24) displays messages of the utility monitor and SQLSTATEs.

If M+- is displayed in the command area, you can page through the message area by entering m+ and m-. This is possible when long message texts or several messages are issued.

You cannot make any entries in the message area.

4.1.5 Input and output

The input fields are either numeric or alphanumeric, begin after a colon (:) and are displayed at high intensity. In addition, some input fields are single-choice and some are multiple-choice (see section “Selecting fields” on page 150).

The maximum length of an input field is defined by a certain number of characters (blanks for alphanumeric characters and null characters for numeric characters). The input fields may be prefilled by the utility monitor (see below). You can accept these default entries or overwrite them. Lowercase letters are only taken into account inside double quotes, all text outside double quotes is assumed to be uppercase. See also the section entitled “SQL keywords” in the “SQL Reference Manual Part 1: SQL Statements”.
You press the DUE key to send off the entries. The output form is refreshed on the screen, or a continuation form appears.

Output fields and text fields are displayed at lower intensity. The utility monitor outputs default values and tables. The default values are in input fields so they are displayed at high intensity. Fields can be prefilled with default names for the database, schema, storage group (see also section “Define configuration data” on page 79) and table, as well as parameters of SQL statements that are executed in the relevant form. They can also be prefilled with values from previously processed forms.

4.2 Form handling

The following describes how to call, use, control and request help information on the forms of the utility monitor.

4.2.1 Calling forms

If you have assigned a configuration file, and no mandatory parameters are missing, the STM - START MENU start form appears when you start the utility monitor. From the start form, you can call the main functions by either selecting one of those offered or entering the short name of a form in the command area. By specifying the name of a form, you can break out of a predefined call hierarchy (see page 144).

The following diagram shows how the forms of the utility monitor's main functions can be called.

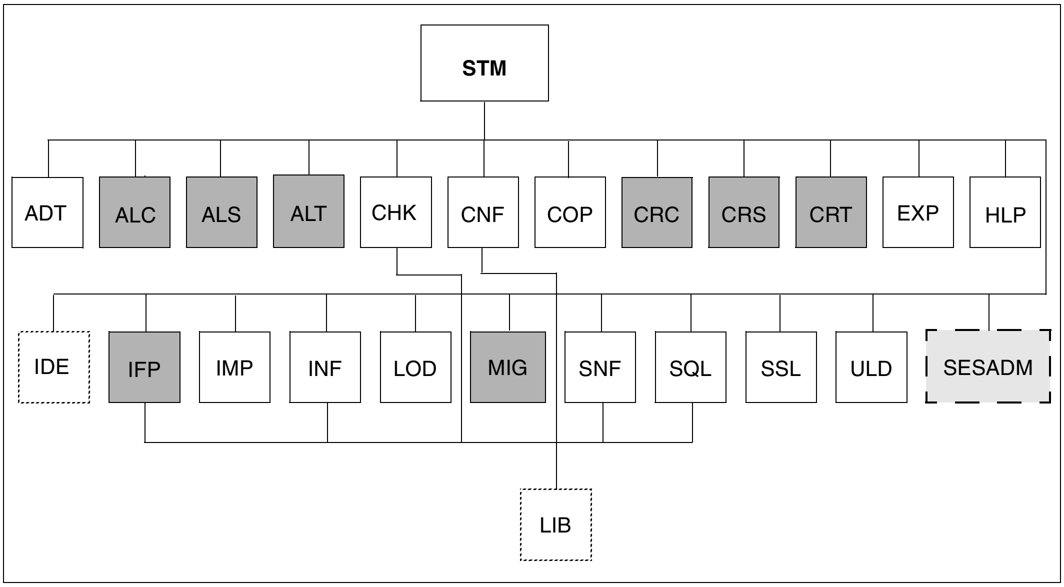


Figure 4: Forms of the utility monitor's main functions

Key to figure 4

☐ You can call one of these forms from any other form by entering its short name in the command area (see also section “The command area” on page 143). With the exception of the ADT - ALTER DATA and SNF - SYS_INFO_SCHEMA forms, you can also call these forms from the function menu in the STM - START MENU start form.

■ These forms can only be called from the function menu of the STM - START MENU start form, with the following exceptions:

– You can call the ALT - ALTER TABLE and CRT - CREATE TABLE forms from the ALS - ALTER SCHEMA form.

– You can call the CRT - CREATE TABLE form from the CRS - CREATE SCHEMA form.

⬚ You can call the IDE form from any other form by entering an exclamation mark (!) in any input field and pressing the F2 key or entering F2 in the command area to send it off. See also section “Entering a delimiter identifier (IDE - DELIMITER IDENTIFIER)” on page 241.

You can call the LIB form from the CNF, COP.4, COP.5, IFP, INF, SNF and SQL forms by entering an exclamation point (!) in any input field and pressing the F2 key or entering F2 in the command area to send it off. See also section “Entering a library member name (LIB - LIBRARY ELEMENT)” on page 265.

▢ The SESADM administration program can be called only from the function menu in the STM - START MENU start form.

If the database, schema and table are displayed in the called form, the values set for these in the calling form apply.

If, from within the INFORMATION_SCHEMA or SYS_INFO_SCHEMA output forms, you enter the short name of a form to branch to a form function, the contents of the output forms are lost and the previous form reappears. The data is not stored because it may no longer correspond to the current status.

4.2.2 Branching and returning to forms

You can branch to a number of other forms from any particular form, and then return to the original form.
You branch to another form by making an entry in the function menu or by marking a function (entering a character next to it).
You send the form off by pressing the DUE key.

You terminate the current form and return to the form from which you called it by pressing the F13 key or entering F13 in the command area. In some cases, you may have to press the F13 key repeatedly to return to the original form (in the case of activities, for example). The previous forms to which you return successively by pressing the F13 key indicate your current position within the form. See also "Help information on the current form position" on page 155.

If you enter stm in the command area, all forms are terminated and you return to the STM - START MENU start form.

In section "Main functions and their continuation forms" on page 187 you will find a detailed description of all the options available for branching to other forms.

4.2.3 Interrupting forms

The K2 key interrupts the utility monitor and takes you to the operating system. You may have to press the K2 key more than once to achieve this.

The BS2000 command RESUME-PROGRAM returns you to the utility monitor.

4.2.4 Scrolling and paging forms

When fields are scrollable, "more: < >" appears next to them.

By pressing the F19 or F20 key or entering < or > in the command area you can scroll the field to the left or right respectively.

When tables are scrollable, "more: + -" appears next to them.

By pressing the F8 or F7 key or entering a plus (+) or minus sign (-) in the command area you can page forward or backward through the table, respectively.

4.2.5 Entering library member names and delimiter identifiers

If you want to enter a library member name or a special name in an input field which is not long enough to accommodate the required name, enter the character “!” in the corresponding input field. Send the form by pressing the F2 key or by entering F2 in the command area

- In the case of an input name for file names in the CNF, COP.4, COP.5, IFP, INF, SNF or SQL forms, the utility monitor branches to the LIB form.
  Here, you can enter a library member name in its full length in accordance with LMS conventions (see section “Entering a library member name (LIB - LIBRARY ELEMENT)” on page 265).
- In the case of input fields for which a special name is permitted by SESAM/SQL, the utility monitor branches to the IDE form.
  Here, you can enter a special name in its full length (see section “Entering a delimiter identifier (IDE - DELIMITER IDENTIFIER)” on page 241). You can call the IDE form from any other form.

After you have sent the form with DUE, you return to the original form. As many characters of the delimiter identifier appear in it as the length of the input field allows.

4.2.6 Selecting fields

Some of the input fields in the forms are single-choice or multiple-choice fields.

Single-choice fields

In single-choice fields you select a single function or object.
There are two ways of doing this:

- by specifying its number
- by marking it (entering a character other than a period next to it)

The utility monitor indicates whether you are to specify a number or mark it.

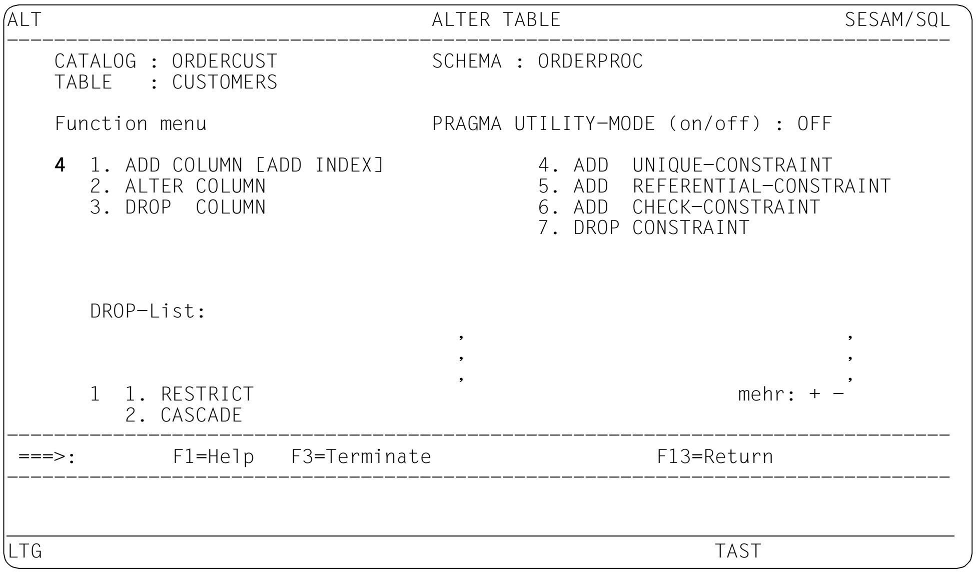
- Selecting an item by specifying a number

  In a list of numbered items, you can select one of them by entering its number.

  Example

  The ALT - ALTER TABLE form offers seven functions for selection.

  You enter 4 to select the ADD UNIQUE-CONSTRAINT function for the CUSTOMERS table.



```
ALT                                ALTER TABLE                      SESAM/SQL
-----------------------------------------------------------------------------
    CATALOG : ORDERCUST            SCHEMA : ORDERPROC
    TABLE   : CUSTOMERS

    Function menu                  PRAGMA UTILITY-MODE (on/off) : OFF

    4  1. ADD COLUMN [ADD INDEX]            4. ADD  UNIQUE-CONSTRAINT
       2. ALTER COLUMN                      5. ADD  REFERENTIAL-CONSTRAINT
       3. DROP  COLUMN                      6. ADD  CHECK-CONSTRAINT
                                            7. DROP CONSTRAINT



       DROP-List:
                                     ,                               ,
                                     ,                               ,
                                     ,                               ,
       1  1. RESTRICT                                      mehr: + -
          2. CASCADE
-----------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                 F13=Return
-----------------------------------------------------------------------------


LTG                                                         TAST
```

- by marking it (entering a character other than a period next to it)

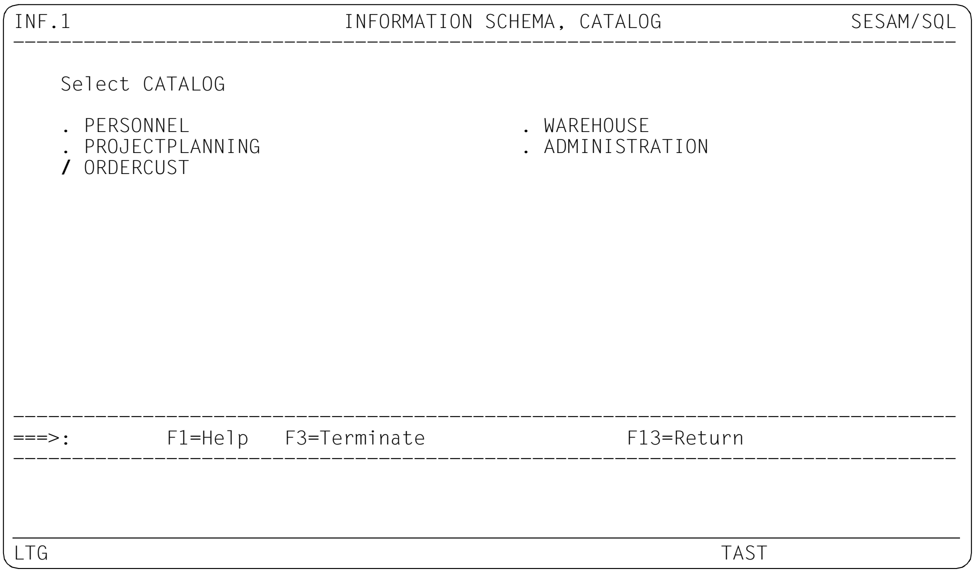
  You use this method to select an item from a list or pageable table. The individual items are preceded by a left-justified input field for marking the selection. Each of these input fields is prefilled with a period (.).

  You select an item by entering a character other than "." (period) in its input field, and you then exit the form by pressing the F13 key or entering F13 in the command area. To deselect an item you have selected, you enter a period in the input field again.

  Example

  The INF.1 - INFORMATION-SCHEMA, CATALOG continuation form offers five databases for selection. You can select one of them.

  You select the ORDERCUST database by entering a slash (/) in its input field.



```
INF.1                        INFORMATION SCHEMA, CATALOG                SESAM/SQL
--------------------------------------------------------------------------------

    Select CATALOG

    . PERSONNEL                             . WAREHOUSE
    . PROJECTPLANNING                       . ADMINISTRATION
    / ORDERCUST










--------------------------------------------------------------------------------
===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------



LTG                                                           TAST
```

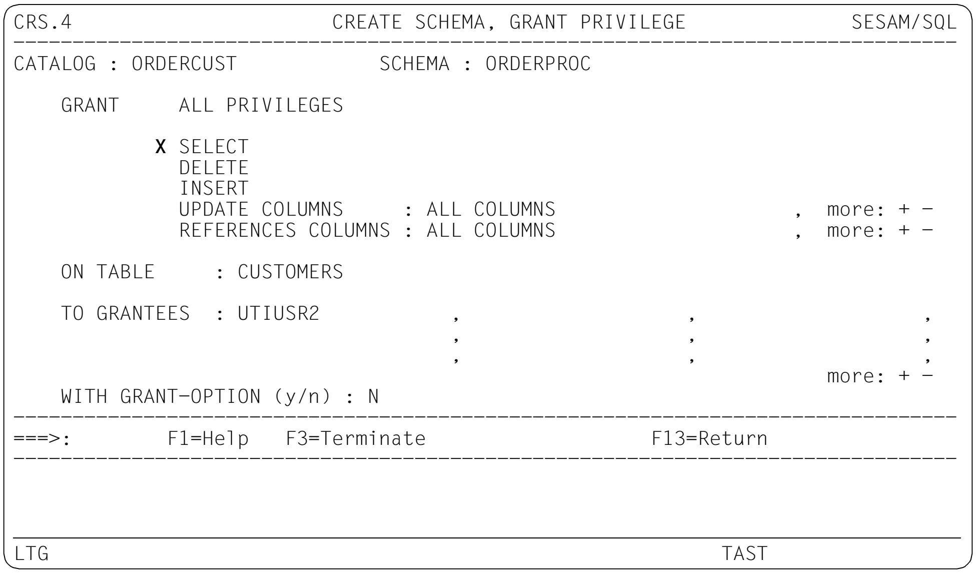
Multiple-choice fields

In multiple-choice fields, you select one or more items, or alternatively no items, from a list. The individual items are preceded by a left-justified input field for marking the selection. You select the required item(s) by marking the corresponding input field(s) with an arbitrary character.

Example

In the CRS.4 - CREATE SCHEMA, GRANT-PRIVILEGE continuation form, you can grant access rights for a schema.

You grant the SELECT table privilege for the CUSTOMERS table to the authorization key UTIUSR2 by entering X in the corresponding input field.



```
CRS.4                          CREATE SCHEMA, GRANT PRIVILEGE              SESAM/SQL
--------------------------------------------------------------------------------
CATALOG : ORDERCUST            SCHEMA : ORDERPROC

    GRANT     ALL PRIVILEGES

            X SELECT
              DELETE
              INSERT
              UPDATE COLUMNS     : ALL COLUMNS                   ,  more: + -
              REFERENCES COLUMNS : ALL COLUMNS                   ,  more: + -

    ON TABLE     : CUSTOMERS

    TO GRANTEES  : UTIUSR2          ,                   ,                   ,
                                    ,                   ,                   ,
                                    ,                   ,                   ,
                                                                    more: + -
    WITH GRANT-OPTION (y/n) : N
--------------------------------------------------------------------------------
===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------



LTG                                                         TAST
```

4.3 Requesting help information on forms

You can request context-sensitive help information on the form in which you are currently working.

You control whether the help texts are displayed in English or German by assigning the appropriate help text file with the link name SEEHELP before starting the utility monitor. See also section “The sequence of commands for starting the utility monitor” on page 76.

Help information on the current form

If you press the F1 key or enter F1 in the command area, a help text on the current form appears. Depending on what type of form it is, the text explains:

- the function of the statement processed in the current form and the purpose of its parameters
- in what way the user interface (function menu) or utility monitor (configuration, instruction file) can be controlled in the current form

The help text on the form consists of a general description of the function and handling of the form and descriptions of the various input fields.

Synonyms are given for some of the terms in the help texts.

Example

```
Constraint (-> Integrity constraint)
```

The term in brackets preceded by an arrow is explained in the glossary.

The help texts are stored in the ISAM files SYSMAN.SESAM-SQL.<ver>.UTI.D (German) and SYSMAN.SESAM-SQL.<ver>.UTI.E (English). You can print these files by means of the BS2000 command /PRINT-DOCUMENT.

Help information on the current input field

If you enter a question mark (?) in the first position of an input field and then press the F1 key or enter F1 in the command area, a help text on this input field appears.

The help text on the field is that part of the help text on the form that describes the field. In other words, the help text on the form is displayed as of the point at which the field is described.

Help information on the current activity

If you enter hoa (help on activity) in the command area, a list appears of the SQL statements issued and the error messages logged during the activity.

When activities are nested, this applies only to the current activity. For example, in the case of CREATE SCHEMA **-->** CREATE TABLE, only the statements and error messages of the CREATE TABLE activity are displayed.

Help information on the current form position

If you enter hmp (help function menu position) in the command area, the current form position is displayed.

Help information on the entries possible in the command area

If you enter a question mark (?) in the command area and then send it off with the DUE key, the HLP.CMD help form appears.

This form contains the first half of the list of possible entries in the command area. The second half of the list is displayed in a continuation form. The output fields are preceded by an input field in which you can select an entry.

Example

Global help information

If you enter hlp in the command area or select the HELP function in the
STM - START MENU form, the
HLP - HELP form appears, from which you can request global help, e.g. on working with the utility monitor.

Help information on syntax

The syntax elements that appear in a form are the syntax elements of the SQL statement or utility statement issued in the form.

The SQL statements are described in the “SQL Reference Manual Part 1: SQL Statements”.

They are described in the “SQL Reference Manual Part 2: Utilities”.

In section “Main functions and their continuation forms” on page 187, you will find a reference to the relevant manual and statement for each function in the forms.

5 Functions of the utility monitor

You can use this chapter to obtain information quickly on specific functions, forms, continuation forms, sequences of forms, or tasks.

The section “Task-oriented overview” on page 158 provides an overview in the form of a table, indicating which forms or sequences of forms you have to call to carry out specific tasks.

The section “Overview of the short names of the forms” on page 172 contains a table of the short names of all the forms in alphabetical order, with a brief description for each form of that what you can use it for.

In section “Overview of the information schemata” on page 182 you will find object-oriented overviews of the information schemata INFORMATION_SCHEMA and SYS_INFO_SCHEMA.

The section “Main functions and their continuation forms” on page 187 describes all the main functions and the continuation forms in alphabetical order.

5.1 Task-oriented overview

The table in this section provides an overview of how to use the utility monitor to carry out specific functions. The different columns in the table are explained below.

Search criterion

This column contains search criteria that help you find the task you are looking for more quickly.

Task

This column contains all the different tasks.

SQL statement/utility statement

This column contains the statements executed for the various tasks. It also indicates whether the statement is an SQL statement or a utility statement and the manual in which you will find a complete syntax description of the statement. It means:

(sql)	SQL statement See the “SQL Reference Manual Part 1: SQL Statements”.
(uti)	Utility statement See the “SQL Reference Manual Part 2: Utilities”.

Form/sequence of forms

This column contains the forms and sequences of forms you have to call to carry out each task using the utility monitor.
Square brackets around forms or sequences of forms indicate that you do not necessarily have to call them. For example, to create a catalog space, it is enough to make the appropriate entries in the CRC form. Only if you want to define further properties of the catalog space do you have to branch to the continuation forms.

Entry via

This column indicates the form from which, or the entry in the command area with which, you can call the relevant form or sequence of forms.

The names of the forms are presented in uppercase, and the short names in the command area are in lowercase.

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
Space	Catalog space			
	Creating	CREATE CATALOG (uti)	CRC [CRC.1 - CRC.4]	STM
	Modifying (metadata)	CREATE/DROP/ALTER MEDIA DESCRIPTION FOR ... (uti)	ALC [ALC.1 - ALC.10]	STM
		CREATE/DROP USER (sql)		
		CREATE/DROP SYSTEM_USER (sql)		
		GRANT/REVOKE SPECIAL_PRIVILEGE (sql)		
		ALTER CODE-TABLE (uti)		
	Reorganizing	REORG (uti)	SSL SSL.7	STM or ssl
	User space			
	Creating	CREATE SPACE (sql)	SSL SSL.4	STM or ssl
	Modifying (parameters)	ALTER SPACE (sql)	SSL SSL.6	STM or ssl
	Deleting	DROP SPACE (sql)	SSL	STM or ssl
	Checking the format of	CHECK FORMAL (uti)	CHK	STM or chk
	Reorganizing	REORG (uti)	SSL SSL.7	STM or ssl
Storage group	Storage group			
	Creating	CREATE STOGROUP (sql)	SSL SSL.1	STM or ssl
	Modifying	ALTER STOGROUP (sql)	SSL SSL.3	STM or ssl
	Deleting	DROP STOGROUP (sql)	SSL	STM or ssl

Table 11: Task-oriented overview (part 1 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
database-specific files and their properties	Description of the storage media and properties of the database-specific DA-LOG, PBI, CAT-LOG and CAT-REC files:			
	Creating – i.e. the first media record for DA-LOG and PBI	CREATE MEDIA DESCRIPTION FOR ... (uti)	CRC CRC.1	STM
	– for all database-specific file types	CREATE MEDIA DESCRIPTION FOR ...(uti)	ALC ALC.1	STM
	Modifying (modifying, adding or deleting media records for all database-specific file types)	ALTER MEDIA DESCRIPTION FOR ... (uti)	ALC ALC.3	STM
	Deleting (all entries for the specified file type)	DROP MEDIA DESCRIPTION FOR ... (uti)	ALC ALC.2	STM
	Changes the coded character set of the database	ALTER CODE-TABLE	ALC ALC.10	STM
Universal user	Specifying the universal user	CREATE CATALOG ... USER (uti)	CRC	STM

Table 11: Task-oriented overview (part 2 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
User privileges	Authorization keys for SQL users			
	Creating	CREATE USER (sql)	CRC CRC.2	STM
			ALC ALC.4	STM
	Deleting	DROP USER (sql)	ALC ALC.5	STM
	System entries for SQL users			
	Creating	CREATE SYSTEM_USER (sql)	CRC CRC.3	STM
			ALC ALC.6	STM
	Deleting	DROP SYSTEM_USER (sql)	ALC ALC.7	STM
	Special privileges			
	Granting	GRANT SPECIAL_PRIVILEGE (sql)	CRC CRC.4 or	STM
			ALC ALC.8	STM
	Revoking	REVOKE SPECIAL_PRIVILEGE (sql)	ALC ALC.9	STM
	Privileges			
	Granting	GRANT (sql)	CRS CRS.4	STM
			ALS ALS.9	STM
	Revoking	REVOKE (sql)	ALS ALS.10	STM

Table 11: Task-oriented overview (part 3 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
Schema Base Table Table View	Schema			
	Creating	CREATE SCHEMA (sql)	CRS [CRT] [CRS.2 - CRS.4]	STM

	Modifying	CREATE/DROP/
		ALTER TABLE (sql)
		CREATE/DROP VIEW (sql)
		CREATE/DROP INDEX (sql)
		GRANT/REVOKE (sql)
			ALS
			[CRT]
			[ALT]
			[ALS.5 - ALS.10]
				STM

	Deleting	DROP SCHEMA (sql)	ALS	STM
	SQL table			
	Creating	CREATE TABLE (sql)	CRT CRT.1 - CRT.1.4 [CRT.4]	STM, ALT or CRS
	Modifying	ALTER TABLE (sql)	ALT [ALT.1 - ALT.7]	STM or ALS
	Modify partitioning	ALTER PARTITIONING (uti)	SSL SSL.9	STM or ssl
	Shuffle row values	ALTER DATA (uti)	ADT	adt
	Deleting	DROP TABLE (sql)	ALS	STM
	Checking the format of	CHECK FORMAL (uti)	CHK	STM or chk
	CALL DML table			
	Creating	CREATE TABLE (sql)	CRT CRT.2 [CRT.4]	STM, ALT or CRS
	Modifying	ALTER TABLE (sql)	ALT ALT.1.2 [ALT 2.2]	STM or ALS
	Modify partitioning	ALTER PARTITIONING (uti)	SSL SSL.9	STM or ssl
	Shuffle row values	ALTER DATA (uti)	ADT	adt
	Deleting	DROP TABLE (sql)	ALS	STM
	Checking the format of	CHECK FORMAL (uti)	CHK	STM or chk

Table 11: Task-oriented overview (part 4 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
Schema Base Table Table View (cont.)	BLOB table			
	Creating	CREATE TABLE (sql)	CRT CRT.3 [CRT.4]	STM, ALT or CRS
	Modifying	ALTER TABLE (sql)	ALT [ALT.1-ALT.3]	STM or ALS
	Deleting	DROP TABLE (sql)	ALS	STM
	Checking the format of	CHECK FORMAL (uti)	CHK	STM or chk
	View			
	Creating	CREATE VIEW (sql)	CRS CRS.2	STM
			ALS ALS.5	STM
	Deleting	DROP VIEW (sql)	ALS	STM

Table 11: Task-oriented overview (part 5 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
Index Column Integrity constraints	Index			
	Creating	CREATE INDEX (sql)	CRS CRS.3 - CRS.3.1	STM
			ALS ALS.7 - ALS.7.1	STM
	Deleting	DROP INDEX (sql)	ALS	STM
	Checking the format of	CHECK FORMAL (uti)	CHK	STM or chk
	Reorganizing global statistics	REORG STATISTICS (sql)	SSL SSL.8	STM or ssl
	Column for SQL table			
	Defining	CREATE TABLE (sql)	CRT CRT.1 - CRT.1.1 [CRT.1.1.1 - CRT.1.1.2] [CRT.1.2 -CRT.1.4]	STM,ALS or CRS
	Modifying	ALTER TABLE (sql)	ALT ALT.2 ALT.2.1	STM,ALS or CRS
	Deleting	ALTER TABLE (sql)	ALT ALT.3	STM
	Column for CALL DML tab.			
	Defining	CREATE TABLE (sql)	CRT CRT.2	STM,ALS or CRS
	Modifying	ALTER TABLE (sql)	ALT ALT.2 ALT.2.2	STM,ALS or CRS
	Deleting	ALTER TABLE (sql)	ALT ALT.3	STM

Table 11: Task-oriented overview (part 6 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
Index Column Integrity constraints (cont.)	Integrity constraint			
	Defining	CREATE TABLE (sql)	CRT CRT.1 CRT.1.2 - CRT.1.4	STM,ALS or CRS
	Modifying	ALTER TABLE (sql)	ALT ALT.4 - ALT.7	STM or ALS
	Deleting	ALTER/DROP TABLE (sql)	ALT ALS	STM or ALS STM
	Checking	CHECK CONSTRAINTS (uti)	CHK CHK.4 - CHK.4.1.1	STM or chk
SQL statements	Issues dynamically compilable SQL statements	(sql)	SQL [SQL.1]	STM or sql
Loading and unloading	Loading user data from an input file into a base table	LOAD (uti)	LOD LOD.1 - LOD.5 [LOD.1.1] LOD.5.1	STM or lod
	Unloading user data from a base table into an output file	UNLOAD (uti)	ULD ULD.1 - ULD.5 [ULD.1.1 - ULD.1.3] ULD.5.1	STM or uld
Exporting and importing	Exporting a base table from a database to an export file	EXPORT TABLE (uti)	EXP	STM or exp
	Importing a base table from an export file into a database	IMPORT TABLE (uti)	IMP [IMP.1]	STM or imp

Table 11: Task-oriented overview (part 7 of 13)

Search criterion	Task	SQL statement/ utility statement	Form/ sequence of forms	Entry via
Backup and recovery	Backing up a database (Catalog space and user spaces)	COPY (uti)	COP COP.1	STM or cop
	Recovering a database (repairing the catalog space and user spaces, resetting to a previous backup, rebuilding the indexes)	RECOVER (uti)	COP COP.2 COP.2.1.1 - COP.2.6	STM or cop
	Deletes records from the RECOVERY_UNITS and DA_LOGS tables or deletes records from the CAT-REC file (online update).	MODIFY RECOVERY (uti)	COP COP.3	STM or cop
	Editing the metadata of the CAT-REC file (only possible with independent DBH)			
	Querying metadata		COP COP.4 COP.4.1 - COP.4.3	STM or cop
	Deleting the metadata (offline update)		COP COP.4 COP.4.4	STM or cop
	Reading replication info block		COP COP.4 COP.4.6	STM or cop
	Querying the metadata of a space (only possible with independent DBH)		COP COP.5 COP.5.1 - COP.5.2	STM or cop
	Create a replication	CREATE REPLICATION (uti)	COP COP.6	STM or cop
	Update a replication	REFRESH REPLICATION (uti) REFRESH SPACE (uti)	COP COP.7	STM or cop

Table 11: Task-oriented overview (part 8 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
INFOR MATION_ SCHEMA	Querying metadata from INFORMATION_ SCHEMA on:	(sql)		STM or inf
	– databases		INF INF.1	
	– Privileges		INF INF.2	
	– System entries		INF INF.3	
	– Authorization identifier		INF INF.4 INF.4.1 - INF.4.3	
	– RECOVERY_UNITS		INF INF.5 - INF.5.2	
	– DA_LOGS (backup copies for the user spaces)		INF INF.6	
	– database-specific files and their properties		INF INF.7	
	– database-specific files and their media		INF INF.8	
	– Schemas		INF INF.9 INF.9.1 - INF.9.2	
	– base tables		INF INF.9 INF.9.3 INF.9.3.1 - INF.9.3.19	
	– views		INF INF.9 INF.9.4 INF.9.4.1 - INF.9.4.7	
	– Integrity constraints		INF INF.9 INF.9.5 INF.9.5.1 - INF.9.5.5	

Table 11: Task-oriented overview (part 9 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
INFOR MATION_ SCHEMA (continued)	– indexes		INF INF.9 INF.9.6 INF.9.6.1 - INF.9.6.3	
	– table privileges, not specifying individual columns		INF INF.9 INF.9.7	
	– UPDATE and REFERENCES table privileges for individual columns		INF INF.9 INF.9.8	
	– Routines		INF INF.9 INF.9.9 INF.9.9.1 - INF 9.9.10	
	– Storage groups		INF INF.10	
	– storage groups and the associated volumes		INF INF.11	
	– granted USAGE special privileges		INF INF.12	
	– User spaces		INF INF.13 INF.13.1 - INF.13.4	

Table 11: Task-oriented overview (part 10 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
SYS_ INFO_ SCHEMA	Querying metadata from SYS_INFO-SCHEMA on:	(sql)		snf
	– universal users		SNF	
	– default value of the LOG parameter		SNF.1	
	– coded character set		SNF.1	
	– Authorization identifier		SNF SNF.2	
	– System entries		SNF SNF.3	
	– Schemas		SNF SNF.4	
	– Tables		SNF SNF.5	
	– columns		SNF SNF.6	
	– tables and columns referenced by a view		SNF SNF.7	
	– table constraints		SNF SNF.8	
	– UNIQUE and primary key constraints		SNF SNF.9	
	– referential constraints		SNF SNF.10	
	– Check constraints		SNF SNF.11	
	– tables and columns to which a check constraint refers		SNF SNF.12	
	– granted privileges		SNF SNF.13	
	– granted USAGE special privileges		SNF SNF.14	

Table 11: Task-oriented overview (part 11 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
SYS_ INFO_ SCHEMA (continued)	– granted special privileges except USAGE		SNF SNF.15	
	– indexes		SNF SNF.16	
	– Storage groups		SNF SNF.17	
	– User spaces		SNF SNF.18	
	– RECOVERY_UNITS		SNF SNF.19	
	– DA_LOGS		SNF SNF.20	
	– Media table		SNF SNF.21	
	– Space properties		SNF SNF.22	
	– Partitions		SNF SNF.23	
	– Routines		SNF SNF.24	
	– Routine parameters		SNF SNF.25	
	– Routine privileges		SNF SNF.26	
	– Routine: tables and columns used		SNF SNF.27	
	– Routine: called routines		SNF SNF.28	
	– View: routines used		SNF SNF.29	

Table 11: Task-oriented overview (part 12 of 13)

Search criterion	**Task**	**SQL statement/ utility statement**	**Form/ sequence of forms**	**Entry via**
Other	Converting a SESAM/SQL V1 database to a SESAM/SQL table of the current version	MIGRATE (uti)	MIG	STM
	Migrate CALL DML/SQL table to SQL table	MIGRATE (uti)	MIG	STM
	Migrate CALL DML only table to CALL DML/SQL table.	MIGRATE (uti)	MIG	STM
	Define configuration data		CNF	STM or cnf
	Entering a delimiter identifier		IDE	see [1]
	Processing an instruction file		IFP	STM
	Entering a library member name		LIB	see [2]
	Administering a database		STM SESADM function	STM
	Call file editor EDT as a subroutine			edt

Table 11: Task-oriented overview (part 13 of 13)

[1] Entry via all forms by entering “!” in the current input field and “F2” in the command area, or by pressing the F2 key.

[2] Entry via the forms CNF, COP.4, COP.5, IFP, INF, SNF and SQL by entering “!” in the input field for file names and “F2” in the command area, or by pressing the F2 key.

5.2 Overview of the short names of the forms

The table below lists in alphabetical order the short names of all the forms together with their purpose. You will find a detailed description of the main functions and continuation forms in section "Main functions and their continuation forms" on page 187.

Short name of form	Meaning
ADT (ALTER DATA)	**Shuffle column values, anonymize user data**
ALC (ALTER CATALOG)	**Modifies a database's metadata, including all its properties.**
ALC.1	Define the properties of the database- or space-specific DA-LOG, PBI, CAT-LOG, CAT-REC and DDL-TA-LOG files, and the media on which these files are to be stored
ALC.2	Delete the entries of the DA-LOG, PBI, CAT-LOG, CAT-REC and DDL-TA-LOG files
ALC.3	Modify the properties of the DA-LOG, PBI, CAT-LOG, CAT-REC and DDL-TA-LOG files
ALC.4	Create an authorization identifier
ALC.5	Delete an authorization identifier
ALC.6	Create a system entry
ALC.7	Delete a system entry
ALC.8	Grant special privileges
ALC.9	Revoke special privileges
ALC.10	Changes the coded character set of the database
ALS (ALTER SCHEMA)	**Alter a schema**
ALS.5	Create a view
ALS.7	Create an index
ALS.7.1	Defines an index.
ALS.9	Grant privileges
ALS.10	Revoke privileges
ALT (ALTER TABLE)	**Modify a base table**
ALT.1	Inserts a column and possibly an index.
ALT.1.1	Inserts a column in an SQL table.
ALT.1.1.1	Sets a default value.
ALT.1.1.2	Defines a search condition for a check constraint.
ALT.1.2	Inserts a column in a CALL DML table.
ALT.1.3	Defines an index.

Table 12: Short names of utility monitor forms (part 1 of 10)

Short name of form	Meaning
ALT.2	Creates or modifies the data type of a column; displays error file.
ALT.2.1	Modifies the data type of a column in an SQL table; sets or deletes the default value.
ALT.2.1.2	Sets a default value.
ALT.2.2	Modifies the data type of a column in a CALL DML table.
ALT.2.3	Displays the error file.
ALT.4	Adds a UNIQUE constraint.
ALT.5	Adds a referential constraint.
ALT.6	Adds a check constraint.
CHK (**CH**EC**K**)	**Carries out checks.**
CHK.4	Checks spaces, indexes and tables for formal accuracy.
CHK.4.1	Check integrity constraints
CNF (**C**O**NF**IGURATION)	**Checks and modifies configuration data.**
COP (**COP**Y & RECOVER/ REPLICATION)	**Backs up and recovers spaces, queries and deletes metadata on backup copies, modifies backup tables, obtains information on backup copies, creates and refreshes replications.**
COP.1	Creates backup copies of spaces.
COP.2	Repairs spaces, resets to recovery units and rebuilds indexes.
COP.2.1.1 COP.2.1.2 COP.2.1.3	Repairs a specified space: from a backup from a foreign copy from a replication
COP.2.1.1.1 COP.2.1.2.1 COP.2.1.3.1	Outputs the time stamps of the recovery unit records.
COP.2.1.1.2 COP.2.1.2.2 COP.2.1.3.2	Outputs a warning that all references to DA-LOG information are lost (logically canceled) following a RECOVER TO.
COP.2.2.1 COP.2.2.2 COP.2.2.3	Repairs a space list: from a backup from a foreign copy from a replication
COP.2.2.1.1	Outputs the time stamps of the recovery unit records.
COP.2.2.1.2 COP.2.2.2.2 COP.2.2.3.2	Outputs a warning that all references to DA-LOG information are lost (logically canceled) following a RECOVER TO.

Table 12: Short names of utility monitor forms (part 2 of 10)

Short name of form	Meaning
COP.2.3	Recovers a number of spaces as space set
COP.2.3.1	Outputs the time stamps of the recovery unit records.
COP.2.3.2	Outputs a warning that all references to DA-LOG information are lost (logically canceled) following a RECOVER TO.
COP.2.4.1 COP.2.4.2 COP.2.4.3	Repairs catalog space: from a backup from a foreign copy from a replication
COP.2.4.1.1	Outputs the time stamps of the recovery unit records.
COP.2.4.1.2 COP.2.4.2.2 COP.2.4.3.2	Outputs a warning that all references to DA-LOG information are lost (logically canceled) following a RECOVER TO.
COP.2.5.1 COP.2.5.2 COP.2.5.3	Repairs entire database: from a backup from a foreign copy from a replication
COP.2.5.1.1	Outputs the time stamps of the recovery unit records.
COP.2.5.1.2 COP.2.5.2.2 COP.2.5.3.2	Outputs a warning that all references to DA-LOG information are lost (logically canceled) following a RECOVER TO.
COP.2.6	Rebuild indexes
COP.3	Deletes records from the RECOVERY_UNITS and DA_LOGS tables or deletes records from the CAT-REC file (online update).
COP.4	Reads the metadata of the CAT-REC file and deletes recovery unit records from the CAT-REC file.
COP.4.1	Outputs the identification record and CREATE-CATALOG record of the CAT-REC file.
COP.4.2	Outputs the CAT-REC file.
COP.4.3	Outputs the recovery unit records of the CAT-REC file.
COP.4.4	Deletes recovery unit records from the CAT-REC file (offline update).
COP.4.4.1	Outputs the time stamps of the recovery unit records.
COP.4.6	Reads the replication info block
COP.5	Outputs the metadata of a space.
COP.5.1	Outputs the metadata of the specified space.
COP.5.2	Outputs the metadata of the tables in the specified space.
COP.5.3	Outputs the metadata of the indexes of tables in the specified space.

Table 12: Short names of utility monitor forms (part 3 of 10)

Short name of form	Meaning
COP.5.4	Outputs the metadata of the columns of tables in the specified space.
COP.6	Creates replications.
COP.7	Refreshes replications.
COP.7.1	Outputs the time stamps of the recovery unit records.
CRC (**CR**EATE **C**ATALOG)	**Creates a catalog space with properties.**
CRC.1	Defines the properties of the database-specific DA-LOG and PBI files and specifies the media on which these files are to be stored.
CRC.2	Create an authorization identifier
CRC.3	Create a system entry
CRC.4	Granting special privileges
CRS (**CR**EATE **S**CHEMA)	**Create a schema**
CRS.2	Create a view
CRS.3	Create an index
CRS.3.1	Defines an index.
CRS.4	Grant privileges
CRT (**CR**EATE **T**ABLE)	**Create a base table**
CRT.1	Defines a column or table constraint for an SQL table.
CRT.1.1	Defines a column.
CRT.1.1.1	Sets a default value.
CRT.1.1.2	Defines a search condition for a check constraint.
CRT.1.2	Defines a UNIQUE constraint for a table constraint.
CRT.1.3	Defines a referential constraint for a table constraint.
CRT.1.4	Defines a search condition for a table constraint.
CRT.2	Defines a column in a CALL DML table.
CRT.3	Defines a BLOB table.
CRT.4	Defines partitions of the table
EXP (**EXP**ORT TABLE)	**Exports a base table.**
HLP (**H**E**LP**)	**Call the help function**
HLP.1	Displays the entries possible in the command area.
HLP.2	Displays a help text on working with the input fields.
HLP.3	Displays a help text on how to use the help functions.

Table 12: Short names of utility monitor forms (part 4 of 10)

Short name of form	Meaning
HLP.4	Displays help information on the version.
IDE (DELIMITER **IDE**NTIFIER)	**Enters a delimiter identifier.**
IFP (**I**NSTRUCTION **F**ILE **P**ROCESSING)	**Specifies an instruction file to be processed in interactive mode.**
IMP (**IMP**ORT TABLE)	**Imports a base table.**
IMP.1	Defines partitions of the table
INF (**INF**ORMATION-SCHEMA)	**Queries metadata from INFORMATION_SCHEMA.**
INF.1	Outputs all the databases known to the DBH.
INF.2	Outputs the privileges for the selected database.
INF.3	Outputs the system entries of the selected database.
INF.4	Selects an authorization key by means of which further information is to be output.
INF.4.1	Outputs all the authorization keys of the selected database.
INF.4.2	Outputs the table privileges for which the selected authorization key is the GRANTOR or GRANTEE.
INF.4.3	Outputs, for individual columns, the table privileges for which the selected authorization key is the GRANTOR or GRANTEE.
INF.5	Outputs information on the recovery unit of a selected database or on which files are required for a RECOVER or REFRESH REPLICATION.
INF.5.1	Outputs the recovery unit records of the user spaces of the selected database.
INF.5.2	Outputs information on which files are required for a RECOVER or REFRESH REPLICATION.
INF.5.2.1	Outputs the data records found with INF.5.2 if the information is (also) to be output to the screen.
INF.6	Outputs the DA-LOG files of the selected database.
INF.7	Outputs the database-specific files and their properties.
INF.8	Outputs the database-specific files and their media.
INF.9	Selects a schema on which further information is to be output.
INF.9.1	Outputs all the schemata of the selected database.
INF.9.2	Outputs the tables of the selected schema.

Table 12: Short names of utility monitor forms (part 5 of 10)

Short name of form	**Meaning**
INF.9.3	Selects a base table on which further information is to be output.
INF.9.3.1	Outputs all the base tables of the selected schema.
INF.9.3.2	Outputs the privileges of the selected base table.
INF.9.3.3	Outputs the UNIQUE and referential constraints of the selected base table.
INF.9.3.4	Outputs the indexes of the selected base table.
INF.9.3.5	Outputs the views that reference the selected base table.
INF.9.3.6	Outputs the integrity constraints that reference the selected base table.
INF.9.3.7	Outputs routines that reference the selected base table.
INF.9.3.8	Outputs all the columns of the selected base table.
INF.9.3.9	Outputs the data of the selected column.
INF.9.3.10	Outputs detailed data on the selected column.
INF.9.3.11	Outputs the privileges of the selected column.
INF.9.3.12	Outputs the indexes of the selected column.
INF.9.3.13	Outputs views that reference the selected column.
INF.9.3.14	Outputs the integrity constraints that reference the selected column.
INF.9.3.15	Outputs the routines that reference the selected column.
INF.9.3.16	Outputs the table constraints of the selected base table.
INF.9.3.17	Outputs the tables dependent on the selected table constraint.
INF.9.3.18	Outputs the columns dependent on the selected integrity constraint.
INF.9.3.19	Outputs properties of a table’s partitions

Table 12: Short names of utility monitor forms (part 6 of 10)

Short name of form	Meaning
INF.9.4	Selects a view on which further information is to be output.
INF.9.4.1	Outputs all the views of the selected schema.
INF.9.4.2	Outputs the data of the selected view.
INF.9.4.3	Outputs the referenced tables of the selected view.
INF.9.4.4	Outputs the referenced columns of the selected view.
INF.9.4.5	Outputs the privileges of the selected view.
INF.9.4.6	Selects the column of the selected view on which more information is to be output.
INF.9.4.6.1	Outputs the columns of the selected view.
INF.9.4.6.2	Outputs the data of the selected column.
INF.9.4.7	Referenced routines of the view.
INF.9.5	Selects an integrity constraint on which further information is to be output.
INF.9.5.1	Outputs all the referential constraints of the selected schema; the integrity-constraint names assigned by the system are output in the CONSTRAINT-NAME field and can be checked.
INF.9.5.2	Outputs all the check constraints of the selected schema.
INF.9.5.3	Outputs the data of the selected check constraint.
INF.9.5.4	Outputs the tables dependent on the selected integrity constraint.
INF.9.5.5	Outputs the columns dependent on the selected integrity constraint.
INF.9.6	Selects an index on which further information is to be output.
INF.9.6.1	Outputs all the indexes of the selected schema.
INF.9.6.2	Outputs the data of the selected index.
INF.9.6.3	Outputs all the indexed columns of the selected schema.
INF.9.7	Outputs all the table privileges of the selected schema, without specifying individual columns.
INF.9.8	Outputs all the UPDATE and REFERENCES table privileges for individual columns of the selected schema.

Table 12: Short names of utility monitor forms (part 7 of 10)

Short name of form	Meaning
INF.9.9	Selects a routine and possibly a parameter on which further information is to be output.
INF.9.9.1	Routines of the schema.
INF.9.9.2	Data of the selected routine.
INF.9.9.3	Privileges of the routine.
INF.9.9.4	Parameters of the routine.
INF.9.9.5	Data of the selected parameter.
INF.9.9.6	Tables which the routine uses.
INF.9.9.7	Columns which the routine uses.
INF.9.9.8	Called routines of the selected routine.
INF.9.9.9	Calling routines of the selected routine.
INF.9.9.10	Views which use the routine.
INF.10	Outputs the storage groups of the selected database.
INF.11	Outputs the storage groups and the associated volumes of the selected database.
INF.12	Outputs the USAGE special privileges granted for the selected database.
INF.13	Selects a space on which more information is to be output.
INF.13.1	Outputs all the spaces of the selected database.
INF.13.2	Outputs the indexes stored in the selected space.
INF.13.3	Outputs the tables stored in the selected space.
INF.13.4	Outputs the recovery unit records of the selected space.
LIB (**LIB**RARY ELEMENT)	**Entering a library member name**
LOD (**LO**A**D**)	**Loads a base table with data from a file.**
LOD.1	Input file in UNLOAD format
LOD.1.1	Specifies the LOAD-COLUMN list.
LOD.2	Input file in TRANSFER format
LOD.3	Input file in DELIMITER format
LOD.4	Input file in CSV format
LOD.5	Input file in user-defined format
LOD.5.1	User definition of the format of a column and specification of the column in the table in which loading is to take place

Table 12: Short names of utility monitor forms (part 8 of 10)

Short name of form	Meaning
MIG (**MIG**RATE)	**Converts a SESAM/SQL V1 database to a SESAM/SQL table of the current version or a CALL DML/SQL table to an SQL table or a CALL DML-only table to a CALL DML/SQL table**
SNF (**S**YS-I**NF**O-SCHEMA)	**Queries metadata from SYS_INFO_SCHEMA.**
SNF.1	Outputs the universal user, default value for the LOG parameter and coded character set of the specified database.
SNF.2	Outputs all the authorization keys of the specified database together with their short names.
SNF.3	Outputs all the system entries of the specified database.
SNF.4	Outputs all the schemata of the specified database together with their owners.
SNF.5	Outputs all the base tables and views of the specified database.
SNF.6	Outputs all the columns of the specified database.
SNF.7	Outputs all the tables and columns of the specified database that are referenced by a view.
SNF.8	Outputs all the table constraints of the specified database.
SNF.9	Outputs all the UNIQUE and primary key constraints of the specified database.
SNF.10	Outputs all the referential constraints of the specified database.
SNF.11	Outputs all the check constraints of the specified database.
SNF.12	Outputs all the tables and columns of the specified database to which a check constraint refers; the integrity-constraint names assigned by the system are output in the CONSTRAINT-NAME field and can be checked.
SNF.13	Outputs all the privileges of the specified database.
SNF.14	Outputs all the USAGE privileges of the specified database.
SNF.15	Outputs all the special privileges of the specified database.
SNF.16	Outputs all the indexed columns of the specified database.
SNF.17	Outputs all the storage groups of the specified database.
SNF.18	Outputs all the spaces of the specified database.
SNF.19	Outputs the recovery unit records of the specified database.
SNF.20	Outputs the DA-LOG files of the specified database.
SNF.21	Outputs the records of the media table of the specified database.
SNF.22	Outputs the space properties.

Table 12: Short names of utility monitor forms (part 9 of 10)

Short name of form	Meaning
SNF.23	Outputs properties of a table’s partitions.
SNF.24	Outputs the properties of routines.
SNF.25	Outputs the parameters of routines.
SNF.26	Outputs the privileges of routines.
SNF.27	Outputs tables and columns used by routines.
SNF.28	Outputs routines called by routines.
SNF.29	Outputs routines used by views.
SQL (**SQL**-STATEMENTS)	**Issues dynamically compilable SQL statements.**
SQL.1	Outputs the records found for the SELECT statement issued in the SQL form.
SSL (**SSL**)	**Controls the management of storage space.**
SSL.1	Creates a storage group.
SSL.3	Modifies the description of a storage group.
SSL.4	Create user space
SSL.6	Modifies the properties of a user space.
SSL.7	Reorganizes spaces
SSL.8	Reorganizes global statistics for an index.
SSL.9	Changing the partitioning of a base table
STM (**ST**ART **M**ENU)	**Start form** Allows you to branch to any main function; calls the SESADM administration program
ULD (**U**N**L**OA**D**)	**Unloads data from a base table to a file.**
ULD.1	Output file in LOAD format
ULD.1.1	Specifies the column list.
ULD.1.2	Defines the WHERE clause.
ULD.1.3	Defines the ORDER BY clause.
ULD.2	Output file in TRANSFER format
ULD.3	Output file in DELIMITER format
ULD.4	Output file in CSV format
ULD.5	Output file in user-defined format
ULD.5.1	Defines the format for the column to be unloaded to the output file and specifies which column is to be unloaded.

Table 12: Short names of utility monitor forms (part 10 of 10)

5.3 Overview of the information schemata

The tables below indicate which INFORMATION_SCHEMA or SYS_INFO_SCHEMA views contain information on which database objects and in which utility monitor forms this information is displayed. The views of the information schemata are described in the “SQL Reference Manual Part 1: SQL Statements”.

Views of the INFORMATION_SCHEMA

Object	View name	Information on	Form
Schema	SCHEMATA	Schemas in the database	INF.9 - INF.9.1
Table	TABLES	Tables in the database	INF.9.2
	BASE_TABLES	Base tables in the database	INF.9.2 INF.9.3 - INF.9.3.1 INF.9.3.18 INF.13.2
	PARTITIONS	Partitions of a table	INF.9.3.19
	VIEWS	Views of the database	INF.9.4 - INF.9.4.2
	VIEW_TABLE_USAGE	Tables on which the views are based	INF.9.3.5 INF.9.4.3
	CONSTRAINT_TABLE_USAGE	Tables on which integrity constraints are based	INF.9.3.6 INF.9.3.17 INF.9.5.4
	ROUTINE_TABLE_USAGE	Routines which reference a table	INF.9.3.7
View	VIEWS	Views of the database	INF.9.4 - INF.9.4.2
	VIEW_ROUTINE_USAGE	Referenced routines of the view.	INF.9.4.7

Table 13: Views of the INFORMATION_SCHEMA (part 1 of 3)

Object	**View name**	**Information on**	**Form**
Column	COLUMNS	Columns in the database	INF.9.4.6 - INF.9.4.6.2
	BASE_TABLE_COLUMNS	Columns in the base tables	INF.9.3.8 - INF.9.3.10
	VIEW_COLUMN_USAGE	Columns on which views are based	INF.9.3.13 INF.9.4.4
	CONSTRAINT_COLUMN_USAGE	Columns on which integrity constraints are based	INF.9.3.14 INF.9.3.18 INF.9.5.5
	INDEX_COLUMN_USAGE	Columns on which indexes are based	INF.9.3.12 INF.9.6.3
	KEY_COLUMN_USAGE	Columns for which a primary key or UNIQUE constraint is defined	INF.9.3.3
	ROUTINE_COLUMN_USAGE	Routines which reference a column	INF.9.3.15
Privilege	TABLE_PRIVILEGES	Table privileges	INF.4.2 INF.9.3.2 INF.9.4.5 INF.9.7
	COLUMN_PRIVILEGES	Column privileges	INF.4.3 INF.9.3.11 INF.9.8
	CATALOG_PRIVILEGES	Special privileges	INF.2
	USAGE_PRIVILEGES	USAGE privileges	INF.12
Index	INDEXES	Indexes in the database	INF.9.3.4 INF.9.6 - INF.9.6.2 INF.13.2
Integrity constraint	TABLE_CONSTRAINTS	Integrity constraints	INF.9.3.16
	REFERENTIAL_CONSTRAINTS	Reference constraints	INF.9.5.1
	CHECK_CONSTRAINTS	Check constraints	INF.9.5.2 - INF.9.5.3

Table 13: Views of the INFORMATION_SCHEMA (part 2 of 3)

Object	View name	Information on	Form
Routine	ROUTINES	Routines of the database	INF.9.9 - INF.9.9.2
	ROUTINE_PRIVILEGES	Privileges of the routine.	INF.9.9.3
	PARAMETERS	Parameters of the routine.	INF.9.9.4 - INF.9.9.5
	ROUTINE_TABLE_USAGE	Tables which the routine uses.	INF.9.9.6
	ROUTINE_COLUMN_USAGE	Columns which the routine uses.	INF.9.9.7
	ROUTINE_ROUTINE_USAGE	Called and calling routines	INF.9.9.8 - INF.9.9.9
	VIEW_ROUTINE_USAGE	Views which use the routine.	INF.9.9.10
Storage group	STOGROUPS	Storage groups in the database	INF.10 INF.11
Volume	STOGROUP_VOLUME_USAGE	Volumes used for storage groups	INF.11
Space	SPACES	Spaces	INF.13 - INF.13.1
User	USERS	Authorization identifier	INF.4 - INF.4.1
	SYSTEM_ENTRIES	System entries	INF.3
DA-LOG file	DA_LOGS	DA-LOG files	INF.6
Media table	MEDIA_DESCRIPTIONS MEDIA_RECORDS	Media records of the database-specific files	INF.7 - INF.8
Recovery unit	RECOVERY_UNITS	Recovery units for spaces	INF.5 - INF.5.2.1 INF.13.4
Character set	CHARACTER_SETS	Character set	---
Sort sequence	COLLATIONS	Sort sequence	---
Features and confor-mance	SQL_FEATURES SQL_IMPL_INFO SQL_LANGUAGES_S SQL_SIZING	Features, subfeatures, implementations, implemented host languages, embedments and implementation-specific maximum values	---

Table 13: Views of the INFORMATION_SCHEMA (part 3 of 3)

Views of the SYS_INFO_SCHEMA

Object	View name	Information on	Form
Database	SYS_CATALOGS	Database	SNF.1
Schema	SYS_SCHEMATA	Schemas in the database	SNF.4
Table	SYS_TABLES	Tables in the database	SNF.5
	SYS_PARTITIONS	Partitions of the base tables	SNF.23
	SYS_VIEW_USAGE	Tables on which the views are based	SNF.7
	SYS_CHECK_USAGE	Tables of a check constraint	SNF.12
Column	SYS_COLUMNS	Columns in the database	SNF.6
	SYS_VIEW_USAGE	Columns on which views are based	SNF.7
	SYS_CHECK_USAGE	Columns of a check constraint	SNF.12
Privilege	SYS_PRIVILEGES	Table privileges	SNF.13
	SYS_SPECIAL_PRIVILEGES	Special privileges	SNF.15
	SYS_USAGE_PRIVILEGES	USAGE privileges	SNF.14
Index	SYS_INDEXES	Indexes in the database	SNF.16
Integrity constraint	SYS_TABLE_CONSTRAINTS	Integrity constraints	SNF.8
	SYS_REFERENTIAL_ CONSTRAINTS	Reference constraints	SNF.10
	SYS_CHECK_CONSTRAINTS	Check constraints	SNF.11
	SYS_UNIQUE_CONSTRAINTS	UNIQUE constraints	SNF.9
Routine	SYS_ROUTINES	Routines of the database	SNF.24
	SYS_PARAMETERS	Routine parameters	SNF.25
	SYS_ROUTINE_PRIVILEGES	Routine privileges	SNF.26
	SYS_ROUTINE_USAGE	Tables and columns which the routine uses.	SNF.27
	SYS_ROUTINE_ROUTINE_ USAGE	Routines called by routines	SNF.28
	SYS_VIEW_ROUTINE_USAGE	Views which use the routine.	SNF.29
Storage group	SYS_STOGROUPS	Storage groups in the database	SNF.17
Space	SYS_SPACES	Spaces	SNF.18
Space properties	SYS_SPACE_PROPERTIES	Space properties	SNF.22

Table 14: Views of the SYS_INFO_SCHEMA (part 1 of 2)

Object	View name	Information on	Form
User	SYS_USERS	Authorization identifier	SNF.2
	SYS_SYSTEM_ENTRIES	System entries	SNF.3
DA-LOG file	SYS_DA_LOGS	DA-LOG files	SNF.20
Media table	SYS_MEDIA_DESCRIPTIONS	Media records of the database-specific files	SNF.21
Recovery unit	SYS_RECOVERY_UNITS	Recovery units for spaces	SNF.19

Table 14: Views of the SYS_INFO_SCHEMA (part 2 of 2)

5.4 Main functions and their continuation forms

The main functions of the utility monitor and the options available for branching to continuation forms are described below.

All the main function forms are depicted in the manual as they appear on screen. The options available for branching to continuation forms are illustrated in diagrams.

All the functions in the function menus of the main function forms are explained. The individual fields are not described here. Of the functions in the continuation forms, only those that call further continuation forms are explained.

You will find more information on the forms, continuation forms, functions and fields in the help texts for the forms (see also section “Requesting help information on forms” on page 154).

Shuffling column values, anonymizing data (ADT - ALTER DATA)

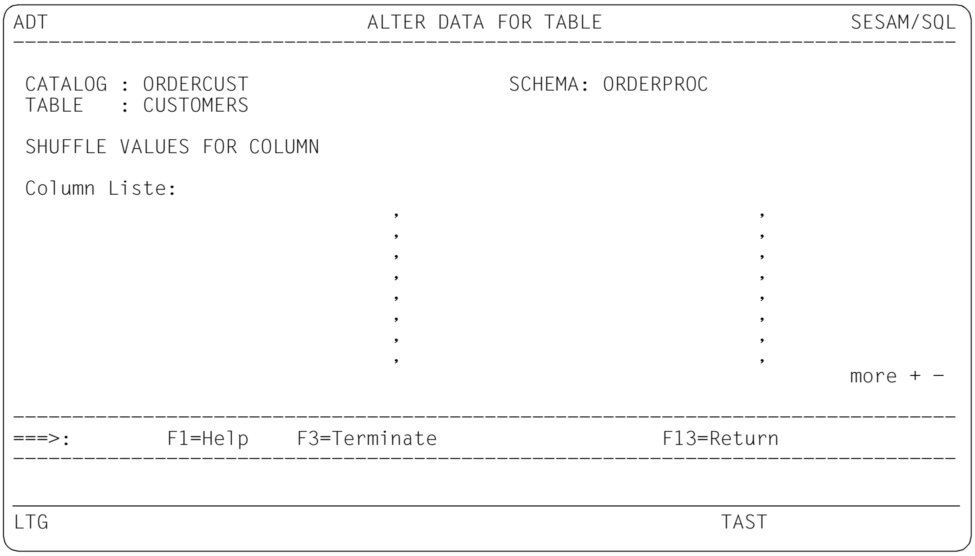
You call the ADT form just by entering the abbreviation “adt” in the command area.

In the ADT form you can shuffle the column values of a base table in such a manner that no conclusions can be drawn about the original content. The original values of a column and its frequency distribution are retained. The shuffling differs from column to column and from case to case.

Important (e.g. person-related) data is thus anonymized.

The ADT form has no continuation forms.

The ADT form



```
ADT                            ALTER DATA FOR TABLE                          SESAM/SQL
--------------------------------------------------------------------------------------

 CATALOG : ORDERCUST                        SCHEMA: ORDERPROC
 TABLE   : CUSTOMERS

 SHUFFLE VALUES FOR COLUMN

 Column Liste:
                                ,                              ,
                                ,                              ,
                                ,                              ,
                                ,                              ,
                                ,                              ,
                                ,                              ,
                                ,                              ,
                                ,                              ,
                                                                          more + -
--------------------------------------------------------------------------------------
===>:        F1=Help     F3=Terminate                   F13=Return
--------------------------------------------------------------------------------------

LTG                                                         TAST
```

In this form columns can be specified whose values are shuffled in order to anonymize a data set. You can specify the column names.

Columns whose names are specified in parentheses are regarded as a unit and are shuffled contiguously. The logical relationship between these columns is retained. The parentheses are set in front of the first and behind the last column name affected. Multiple column names in parentheses can be specified.

Further columns can be specified by entering “+”.

See the “SQL Reference Manual Part 2: Utilities”, utility statement ALTER DATA FOR TABLE.

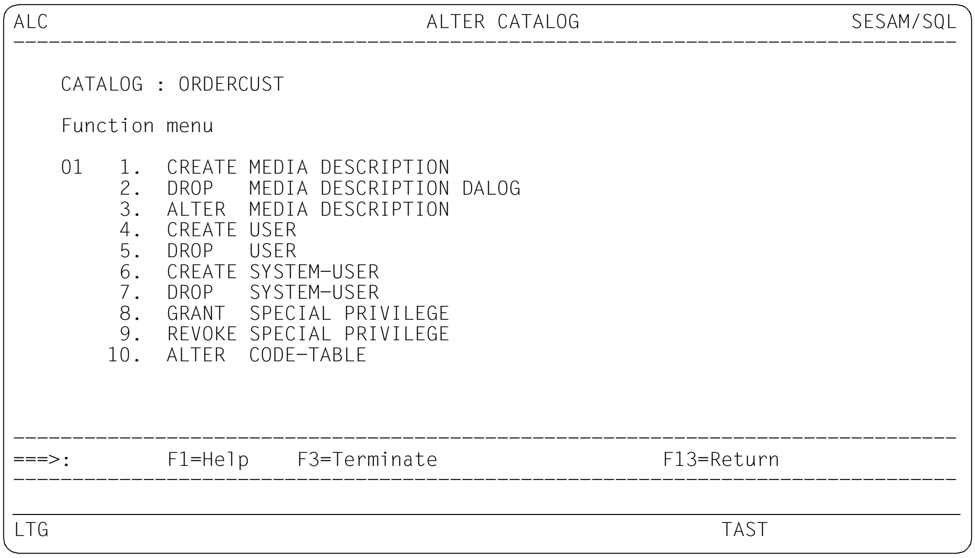
Modifying the metadata of the database (ALC - ALTER CATALOG)

You call the ALC form by selecting function 14, ALTER CATALOG, from the STM - START MENU start form.

In the ALC form and its continuation forms, you can modify the metadata of a database and all its properties.

You are offered an automatic backup before and after the activity provided that you have specified the configuration parameter SEE-COPY = ON (see also the SEE-COPY configuration parameter on page 83).

The ALC form



```
ALC                                  ALTER CATALOG                          SESAM/SQL
-------------------------------------------------------------------------------------


    CATALOG : ORDERCUST

    Function menu

    01   1.  CREATE MEDIA DESCRIPTION
         2.  DROP   MEDIA DESCRIPTION DALOG
         3.  ALTER  MEDIA DESCRIPTION
         4.  CREATE USER
         5.  DROP   USER
         6.  CREATE SYSTEM-USER
         7.  DROP   SYSTEM-USER
         8.  GRANT  SPECIAL PRIVILEGE
         9.  REVOKE SPECIAL PRIVILEGE
        10.  ALTER  CODE-TABLE



-------------------------------------------------------------------------------------
===>:         F1=Help    F3=Terminate                       F13=Return
-------------------------------------------------------------------------------------

LTG                                                               TAST
```

When you select functions 1 - 10, you branch to the corresponding continuation forms.

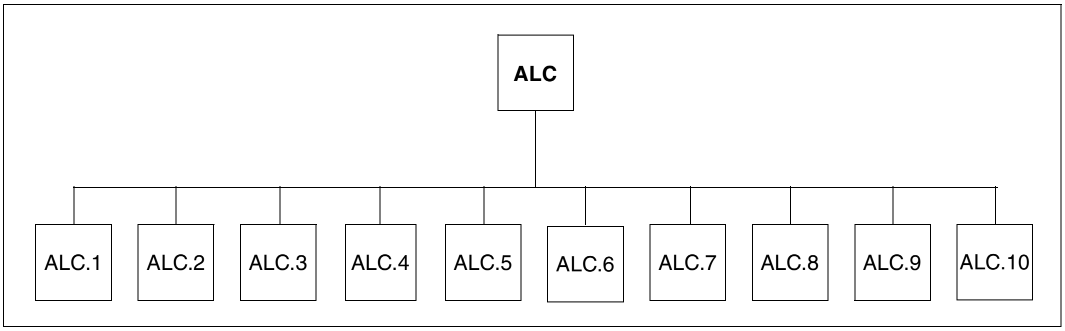


Figure 5: The ALC form and its continuation forms

Explanation of the functions

1. CREATE MEDIA DESCRIPTION

   When you select this function, you branch to the ALC.1 continuation form, in which you can:

   - specify the properties of the database-specific or space-specific DA-LOG, PBI, CAT-LOG, CAT-REC and DDL-TA-LOG files;
   - specify the media on which these files are to be stored

   See the utility statement CREATE MEDIA DESCRIPTION FOR ... in the “SQL Reference Manual Part 2: Utilities”.

2. DROP MEDIA DESCRIPTION

   When you select this function, you branch to the ALC.2 continuation form, in which you can delete the DA-LOG, PBI, CAT-LOG, CAT-REC or DDL-TA-LOG files of the media table.

   See the utility statement DROP MEDIA DESCRIPTION FOR ... in the “SQL Reference Manual Part 2: Utilities”.

3. ALTER MEDIA DESCRIPTION

   When you select this function, you branch to the ALC.3 continuation form, in which you can:

   - change the properties of the DA-LOG, PBI, CAT-LOG, CAT-REC and DDL-TA-LOG files;
   - add media records
   - delete media records

   See the utility statement ALTER MEDIA DESCRIPTION FOR ... in the “SQL Reference Manual Part 2: Utilities”.

4. CREATE USER

   When you select this function, you branch to the ALC.4 continuation form, in which you specify one or more authorization keys. See the SQL statement CREATE USER in the “SQL Reference Manual Part 1: SQL Statements”.

5. DROP USER

   When you select this function, you branch to the ALC.5 continuation form, in which you delete one or more authorization keys and the associated system entries. See the SQL statement DROP USER in the “SQL Reference Manual Part 1: SQL Statements”.

6. CREATE SYSTEM-USER

   When you select this function, you branch to the ALC.6 continuation form, in which you create one or more system entries. See the SQL statement CREATE SYSTEM_USER in the “SQL Reference Manual Part 1: SQL Statements”.

7. DROP SYSTEM-USER

   When you select this function, you branch to the ALC.7 continuation form, in which you delete one or more system entries. See the SQL statement DROP SYSTEM_USER in the “SQL Reference Manual Part 1: SQL Statements”.

8. GRANT SPECIAL PRIVILEGE

   When you select this function, you branch to the ALC.8 continuation form, in which you grant special privileges to one ore more authorization keys. See the SQL statement GRANT in the “SQL Reference Manual Part 1: SQL Statements”.

9. REVOKE SPECIAL PRIVILEGE

   When you select this function, you branch to the ALC.9 continuation form, in which you revoke special privileges from one or more authorization keys. You can specify whether a REVOKE RESTRICT or a REVOKE CASCADE operation is to be performed. REVOKE RESTRICT is the default value. See the SQL statement REVOKE in the “SQL Reference Manual Part 1: SQL Statements”.

10. ALTER CODE-TABLE

    When you select this function, you branch to the ALC.10 continuation form, in which you can enter a different coded character set (synonym: code table) for the database. The character set must be defined in BS2000.
    You can also specify that no coded character set is to be used. See the utility statement ALTER CATALOG in the “SQL Reference Manual Part 2: Utilities”.

    i When a coded character set is specified (`CODE-TABLE` not equal to `*NONE`), only users who have specified the same coded character set in the user configuration file (connection module parameter CCSN) can access the database.
    The SNF, INF and ALC functions can also be executed when different character sets are used.

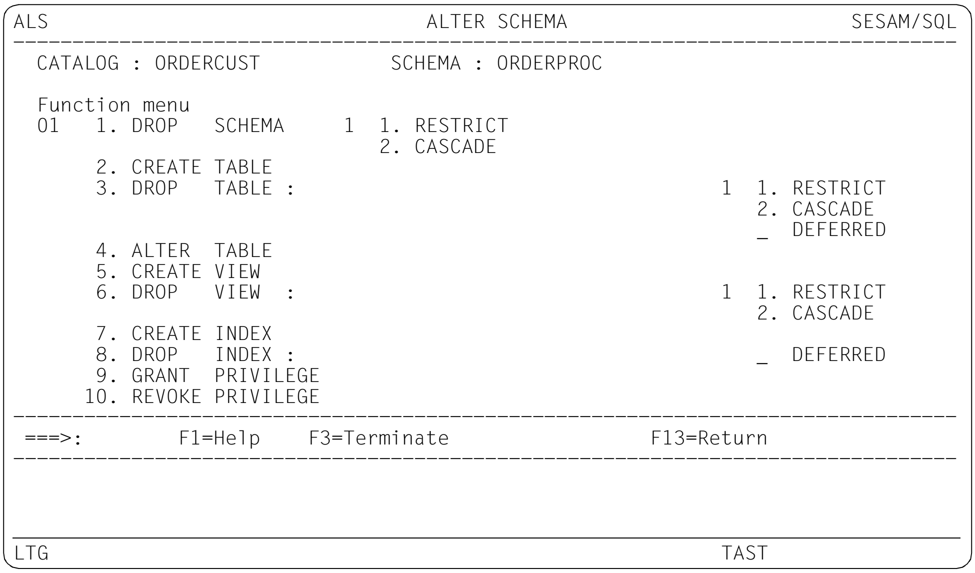
Modifying a schema (ALS - ALTER SCHEMA)

You call the ALS form by selecting 16, ALTER SCHEMA, from the STM - START MENU start form.

In the ALS form and its continuation forms, you can modify a schema within a database. You can delete the schema; create, modify and delete tables; create and delete views and indexes; and grant and revoke privileges.

You are offered an automatic backup before and after each activity provided that you have specified the SEE-COPY = ON configuration parameter (see also the SEE-COPY configuration parameter on page 83).

The ALS form



```
ALS                                  ALTER SCHEMA                          SESAM/SQL
--------------------------------------------------------------------------------
  CATALOG : ORDERCUST              SCHEMA : ORDERPROC

  Function menu
  01    1. DROP   SCHEMA     1  1. RESTRICT
                                2. CASCADE
        2. CREATE TABLE
        3. DROP   TABLE :                                 1  1. RESTRICT
                                                             2. CASCADE
                                                             _  DEFERRED

        4. ALTER  TABLE
        5. CREATE VIEW
        6. DROP   VIEW  :                                 1  1. RESTRICT
                                                             2. CASCADE

        7. CREATE INDEX
        8. DROP   INDEX :                                    _  DEFERRED
        9. GRANT  PRIVILEGE
       10. REVOKE PRIVILEGE
--------------------------------------------------------------------------------
 ===>:        F1=Help     F3=Terminate                 F13=Return
--------------------------------------------------------------------------------



LTG                                                         TAST
```

When you select functions 1, 3, 6 and 8 (DROP ...), the statements are executed immediately when you specify the name of the database object.

When you select function 2, you branch to the CRT form, see also section “Creating a base table (CRT - CREATE TABLE)” on page 233.

When you select function 4, you branch to the ALT form, see also section “Modifying a base table (ALT - ALTER TABLE)” on page 196.

When you select functions 5, 7, 9 and 10, you branch to continuation forms.

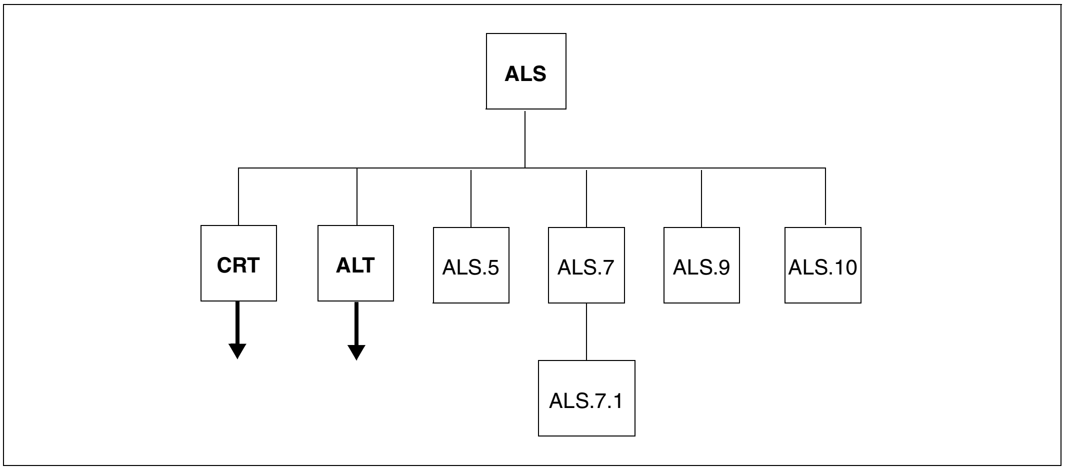


Figure 6: The ALS form and its continuation forms

Explanation of the functions

1. DROP SCHEMA

   When you select this function, the specified schema is deleted. You can specify whether a DROP RESTRICT or a DROP CASCADE operation is to be performed. DROP RESTRICT is the default value.
   See the SQL statement DROP SCHEMA in the “SQL Reference Manual Part 1: SQL Statements”.

2. CREATE TABLE

   When you select this function, you branch to the CRT - CREATE TABLE form (see page 233).

3. DROP TABLE

   When you select this function, the specified table is deleted. You can specify whether a DROP RESTRICT or a DROP CASCADE operation is to be performed. DROP RESTRICT is the default value. You activate the DEFERRED parameter by marking it.
   See the SQL statement DROP TABLE in the “SQL Reference Manual Part 1: SQL Statements”.

4. ALTER TABLE

   When you select this function, you branch to the ALT - ALTER TABLE form (see page 196).
   See the SQL statement ALTER TABLE in the “SQL Reference Manual Part 1: SQL Statements”.

5. CREATE VIEW

   When you select this function, you branch to the ALS.5 continuation form, in which you create a view. See the SQL statement CREATE VIEW in the “SQL Reference Manual Part 1: SQL Statements”.

6. DROP VIEW

   When you select this function, the specified view is deleted. You can specify whether a DROP RESTRICT or a DROP CASCADE operation is to be performed. DROP RESTRICT is the default value. See the SQL statement DROP VIEW in the “SQL Reference Manual Part 1: SQL Statements”.

7. CREATE INDEX

   When you select this function, you branch to the ALS.7 continuation form, in which you create one or more indexes. See the SQL statement CREATE INDEX in the “SQL Reference Manual Part 1: SQL Statements”.

   ALS.7 continuation form, function 1 “Create INDEX DEFINITION”

   When you select this function you branch to the ALS.7.1 continuation form in which you can define an index. The index name and the name of at least one column (COLUMN) must be specified. You are only permitted to specify the length if the associated COLUMN is of data type (N)CHAR, (N)VARCHAR or is a data type of SESAM V12 or older.
   When you press the DUE key, the input is accepted and the ALS.7.1 form is displayed again to allow you to define the next index.
   The defined indexes are not created until you return to the ALS.7 form and select function 3, “Execute INDEX DEFINITION”.

8. DROP INDEX

   When you select this function, the specified index is deleted. You activate the DEFERRED parameter by marking it. See the SQL statement DROP INDEX in the “SQL Reference Manual Part 1: SQL Statements”.

9. GRANT PRIVILEGE

   When you select this function, you branch to the ALS.9 continuation form, in which you grant privileges to one or more authorization keys. See the SQL statement GRANT in the “SQL Reference Manual Part 1: SQL Statements”.

10. REVOKE PRIVILEGE

    When you select this function, you branch to the ALS.10 continuation form, in which you revoke privileges from one or more authorization keys. You can specify whether a REVOKE RESTRICT or a REVOKE CASCADE operation is to be performed. REVOKE RESTRICT is the default value. See the SQL statement REVOKE in the “SQL Reference Manual Part 1: SQL Statements”.

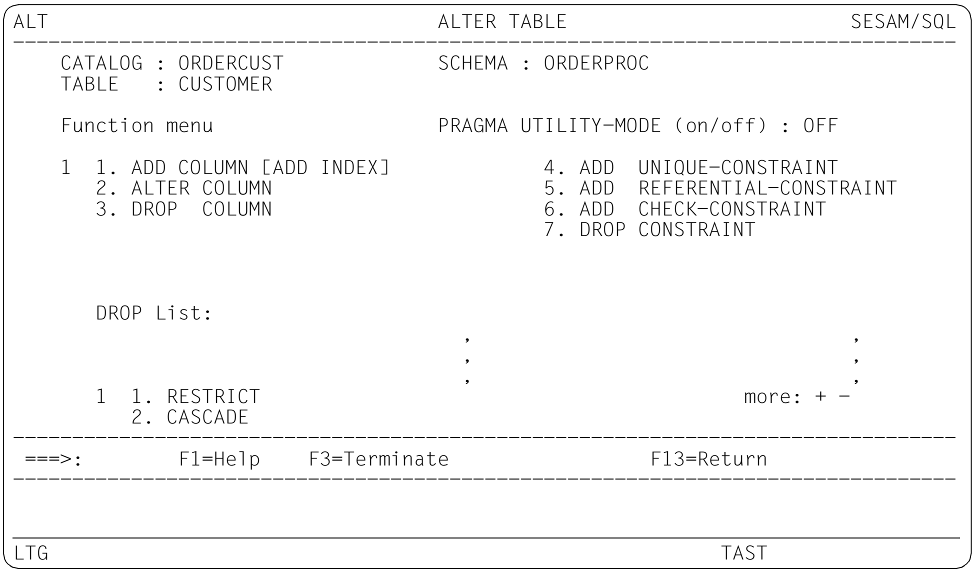
Modifying a base table (ALT - ALTER TABLE)

You call the ALT form by selecting function 18, ALTER TABLE, from the STM - START MENU start form.

In the ALT form and its continuation forms, you can modify an SQL table, a CALL DML table or a BLOB table and its objects. You can add, modify or delete columns or table constraints and indexes.

BLOB tables are treated in the same way as SQL tables. You can define new columns or constraints. Whenever you modify or delete columns, you must take care not to change the characteristic underlying structure of the BLOB table. Otherwise it will no longer be possible to access this table using BLOB operations.
See the SQL statement ALTER TABLE in the “SQL Reference Manual Part 1: SQL Statements”.

The ALT form



```
ALT                                   ALTER TABLE                         SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST               SCHEMA : ORDERPROC
    TABLE   : CUSTOMER

    Function menu                     PRAGMA UTILITY-MODE (on/off) : OFF

    1  1. ADD COLUMN [ADD INDEX]               4. ADD  UNIQUE-CONSTRAINT
       2. ALTER COLUMN                         5. ADD  REFERENTIAL-CONSTRAINT
       3. DROP  COLUMN                         6. ADD  CHECK-CONSTRAINT
                                               7. DROP CONSTRAINT



       DROP List:
                                        ,                                 ,
                                        ,                                 ,
                                        ,                                 ,
       1  1. RESTRICT                                            more: + -
          2. CASCADE
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
--------------------------------------------------------------------------------


LTG                                                       TAST
```

The PRAGMA UTILITY MODE input field (only relevant when you select function 1, 2 or 3) is used to select whether the backup mechanism – which enables you to reset the data to a consistent state in the event of an error – is to be activated (ON) or deactivated (OFF). The input field is preset with OFF.

If “ON” is entered, the ALTER TABLE statement is executed with the UTILITY MODE ON pragma, i.e. resetting is not possible. This improves performance considerably. However, to ensure that a full data backup is available again, a backup must then be created.

If “OFF” is specified, the statement is issued without the pragma, and resetting is possible in the event of an error.

When you select functions 1, 2 and 4 - 6, you branch to continuation forms.

When you select functions 3 and 7, the statement is executed immediately.

Only functions 1 - 3 are permitted for a CALL DML table.

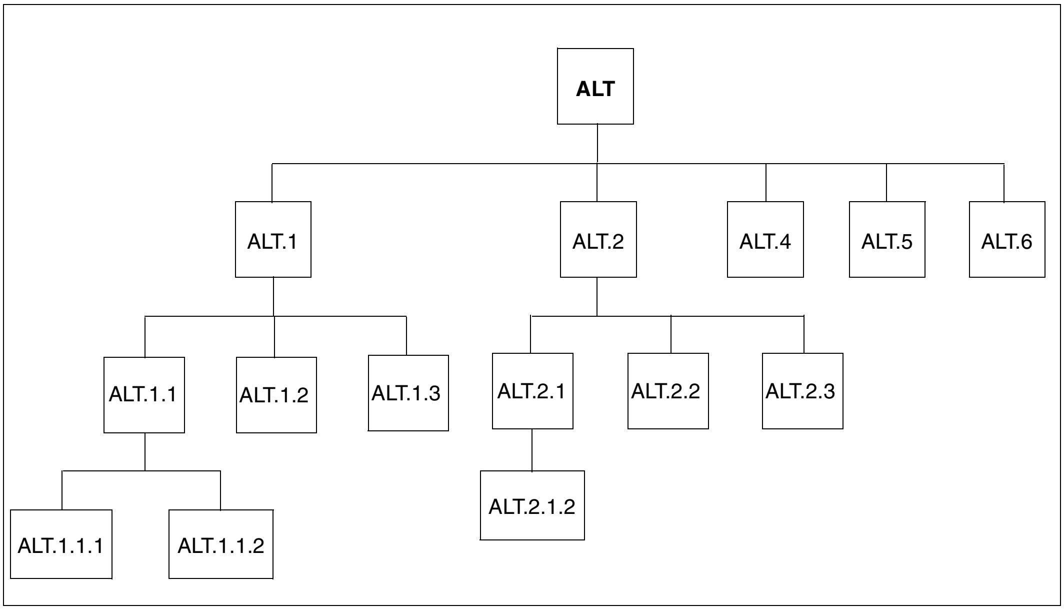


Figure 7: The ALT form and its continuation forms

Explanation of the functions

1. ADD-COLUMN [ADD INDEX]

   When you select this function, you branch to the ALT.1 continuation form. In this continuation form, you can:

   - define and modify a column (functions 1 and 2)
   - define and modify indexes for the defined columns (functions 3 and 4)
   - add the newly created columns and any newly created indexes to a table (function 5)

   The PRAGMA UTILITY MODE field (see page 196) is preset with OFF. A value already entered in the ALT form is accepted.

   1. Create ADD COLUMN DEFINITION

      When you select this function, you branch to the ALT.1.1 continuation form for SQL tables and the ALT.1.2 continuation form for CALL DML tables.
      In these continuation forms you add a column to a table.

      ALT.1.1 continuation form, DEFAULT-CLAUSE function
      If you enter y for this function, you branch to the ALT.1.1.1 continuation form, in which you can set a default value.
      Not until you return to the ALT.1 form and select function 5 is the statement executed.

      ALT.1.1 continuation form, CHECK function
      When you select this function, you branch to the ALT.1.1.2 continuation form, in which you define a search condition for a check constraint.
      Not until you return to the ALT.1 form and select function 5 is the statement executed.

   2. Modify ADD COLUMN DEFINITION

      When you select this function, you branch to the ALT.1.1 or ALT.1.2 continuation form, in which you modify a defined column format. You proceed as described for function 1, Prepare ADD COLUMN DEFINITION.

      You can call function 2 either after function 1, Prepare ADD COLUMN DEFINITION, to modify a defined column, or after function 5, Execute ADD COLUMN [ADD INDEX] DEFINITION, to correct a defined column after the ALTER-TABLE statement has been rejected with an error message.

3. Create ADD INDEX DEFINITION

   Select this function after you have defined the columns in the table. When you select this function, you branch to the ALT.1.3 continuation form. In this continuation form you add another index to a table.

   For each index you can specify the associated columns, and with data types (N)CHAR and (N)VARCHAR, the associated lengths.

   Pressing the DUE key defines the index. The ALT.1.3 form is then displayed again to define the next index. All index definitions must be created in the same space.

   i When you have defined an index, only the value OFF can be specified in the PRAGMA UTILITY MODE field (see page 196).

4. Modify ADD INDEX DEFINITION

   When you select this function, you branch to the ALT.1.3 continuation form, in which you can modify a defined index. You proceed as described for function 3, Prepare ADD INDEX DEFINITION.

   You can call function 4 either after function 3, Prepare ADD INDEX DEFINITION, to modify a defined index, or after function 5, Execute ADD COLUMN [ADD INDEX] DEFINITION, to correct a defined index after the ALTER-TABLE statement has been rejected with an error message.

5. Execute ADD COLUMN [ADD INDEX] DEFINITION

   Only when you select this function are the defined columns and any defined indexes created.

2. ALTER COLUMN

   When you select this function, you branch to the ALT.2 continuation form, in which you can create or change the data type of a column, as well as display a specified error file.

   In the function menu, a distinction is made between preparing (function 1), modifying (function 2) and executing (function 3) a column definition. If you select function 1, you branch to the ALT.2.1 continuation form for SQL tables or the ALT 2.2 form for CALL DML tables.

   The created statement is not issued until you return to the ALT.2 form and select function 3. If you select function 2, you can modify the definition before issuing the statement, e.g. to correct it following erroneous execution.

   The PRAGMA UTILITY-MODE field (see page 196) is preset with OFF. A value already entered in the ALT form is accepted.

Non-convertable or truncated values can be logged in an error file. The error file is only created if such values occur and a file name has been specified in the EXCEPTION-FILE field.
If this file is created by the DBH, the utility monitor branches to the ALT.2.3 continuation form.

ALT.2 continuation form, function 1 COLUMN-DEFINITION for SQL tables

If you select this function, you branch to the ALT.2.1 continuation form in which you can change the data type of a column (function 1) or define that a default value is to be added (function 2) or deleted (function 3). If function 2 is selected, you branch to the 2.1.2 continuation form.

ALT.2.1 continuation form, function 2: DEFAULT-CLAUSE

When you select this function, you branch to the ALT.2.1.2 continuation form, in which you can set a default value. You cannot specify CALL DML default values for SQL tables.
Not until you return to the ALT.2.1 form and press the DUE key is the statement executed.

ALT.2 continuation form, function 1 COLUMN-DEFINITION for CALL-DML tables

If you select this function, you branch to the ALT.2.2 continuation form in which you can change the data type of a column in a CALL DML table. You can only select one of the data types permitted for CALL-DML tables:

CHARACTER, NUMERIC, DECIMAL, INTEGER and SMALLINT.

Columns for CALL DML-compatible tables must not contain any default definitions unless they are of type OLDEST-STYLE or OLD-STYLE.

ALT.2 continuation form, function 2 EXCEPTION-FILE

If an error file was written by the DBH, the utility monitor branches to the ALT.2.3 continuation form. Here you can have the created file displayed.

Depending on the specifications in the preceding forms, not all functions offered here may be available (functions that cannot be selected are locked). If you select function 1, the created error file is displayed, whereby the utility monitor implicitly issues a `SHOW-FILE` command.

If the UTILITY-MODE=OFF pragma was specified in the ALT.2. form, you can now decide whether a COMMIT WORK (function 2) or a ROLLBACK WORK (function 3) is to be executed. Following the execution of functions 2 and 3, you return immediately to the preceding form ALT.2.

If UTILITY-MODE=ON, you can only view the error file. The transaction cannot be rolled back with ROLLBACK WORK.

3. DROP-COLUMN

   If you select this function, you delete the columns specified in the input table provided for this purpose (DROP list).
   You can specify whether a DROP RESTRICT or a DROP CASCADE operation is to be performed.
   If you specify “1” (RESTRICT), the columns are only deleted if no other objects are dependent on them. If you specify “2” (CASCADE), the columns and the objects dependent on them are deleted.
   DROP RESTRICT is the default value.

4. ADD UNIQUE-CONSTRAINT

   When you select this function, you branch to the ALT.4 continuation form, in which you can add a UNIQUE constraint to a table.
   The UNIQUE-SPECIFICATION selection field is preset with a value of 1, which cannot be changed.

5. ADD REFERENTIAL-CONSTRAINT

   When you select this function, you branch to the ALT.5 continuation form, in which you add a referential constraint to a table.

6. ADD CHECK-CONSTRAINT

   When you select this function, you branch to the ALT.6 continuation form, in which you add a check constraint to a table.

7. DROP CONSTRAINT

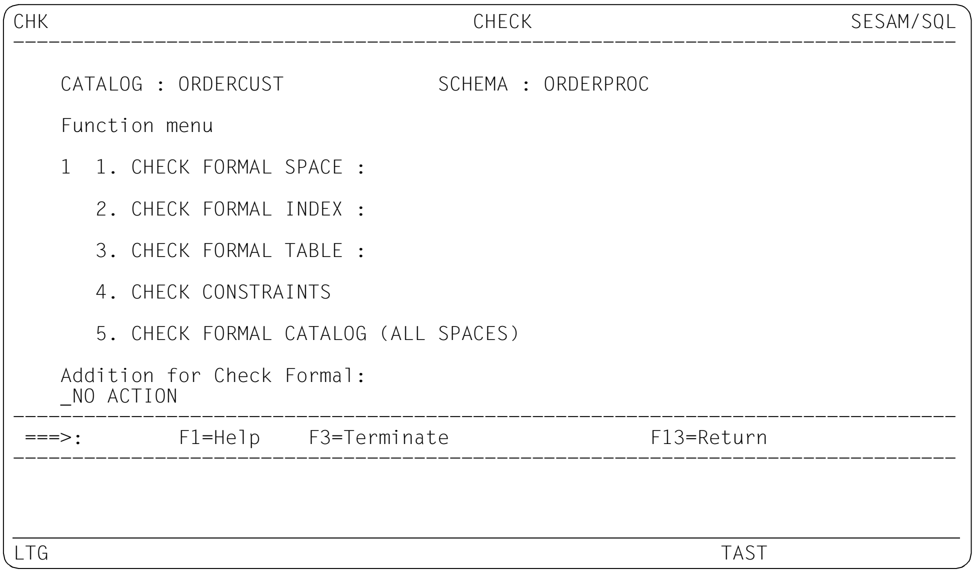
   When you select this function, you delete the table constraint specified in the input table provided for this purpose (DROP list).
   You can specify whether a DROP RESTRICT or a DROP CASCADE operation is to be performed.
   If you specify “1” (RESTRICT), the table constraints are only deleted if no other objects are dependent on them. If you specify “2” (CASCADE), the table constraints and the objects dependent on them are deleted.
   DROP RESTRICT is the default value.

Carrying out checks (CHK - CHECK)

You call the CHK form either by selecting function 3, CHECK, from the STM - START MENU start form or by entering chk in the command area.

In the CHK form and its continuation forms, you can check the format of spaces, indexes and tables. You can also check whether integrity constraints are violated.

The CHK form



```
CHK                                     CHECK                                   SESAM/SQL
-------------------------------------------------------------------------------------------

    CATALOG : ORDERCUST                 SCHEMA : ORDERPROC

    Function menu

    1  1. CHECK FORMAL SPACE :

       2. CHECK FORMAL INDEX :

       3. CHECK FORMAL TABLE :

       4. CHECK CONSTRAINTS

       5. CHECK FORMAL CATALOG (ALL SPACES)

    Addition for Check Formal:
    _NO ACTION
-------------------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------------------



LTG                                                               TAST
```

When you select functions 1 - 3 and 5, the statements are executed immediately. When you select function 4, you branch to a continuation form.

You can set the NO ACTION parameter for CHECK-FORMAL statements. It means that a space is not set to defective as soon as an error is detected.

You can also use CHECK-FORMAL statements for replications.

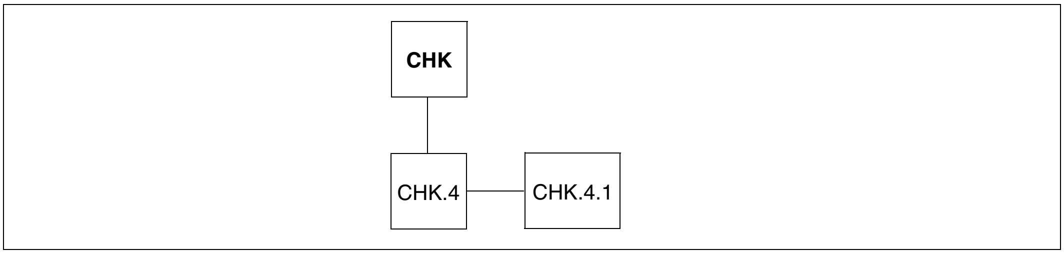


Figure 8: The CHK form and its continuation form

Explanation of the functions

1. CHECK FORMAL SPACE

   When you select this function, you specify the name of the space whose format is to be checked. The format of all tables and indexes in the specified space is checked, as described for functions 2 and 3. The space name CATALOG must be specified as the delimiter identifier for the catalog space: “CATALOG”.
   See the utility statement CHECK FORMAL in the “SQL Reference Manual Part 2: Utilities”.

2. CHECK FORMAL INDEX

   When you select this function, you specify the name of the index whose format is to be checked. The format of the index data of the specified index is checked.
   See the utility statement CHECK FORMAL in the “SQL Reference Manual Part 2: Utilities”.

3. CHECK FORMAL TABLE

   When you select this function, you specify the name of the base table whose format is to be checked. The format of the primary data for the specified table and the internal SESAM/SQL blocks and tables is checked.
   See the utility statement CHECK FORMAL in the “SQL Reference Manual Part 2: Utilities”.

4. CHECK CONSTRAINTS

   When you select this function, you branch to the CHK.4 continuation form, in which you can check the following integrity constraints:
   - one or more specified integrity constraints
   - the integrity constraints of one or more tables
   - the integrity constraints of all the tables in one or more spaces

   You can also specify whether all tables are to be checked or only those with the “check pending” state. You should not check all the tables because, when an integrity constraint is violated, all tables may have the “pending” state.
   The PENDING function is only effective if you have selected CHECK CONSTRAINTS ON TABLE list or CHECK CONSTRAINTS ON SPACE list.
   See the utility statement CHECK CONSTRAINTS in the “SQL Reference Manual Part 2: Utilities” and the section entitled “Processing input/output and error files” in the chapter “Basic information on working with utility statements” of that manual.

   If you enter “y” in the “Output of constraint violations” field and if violations of integrity constraints are identified in the subsequent CHECK CONSTRAINT, the CHK.4.1 continuation form is displayed. The field is preset with “y”.

   CHK.4 continuation form, function 1 “CONSTRAINTS VIOLATION”
   It is possible to identify violations of the following integrity constraints:
   - REFERENTIAL CONSTRAINT
   - CHECK CONSTRAINT
   - UNIQUE CONSTRAINT
   - NOT NULL CONSTRAINT

   In addition to the database name and schema name, the name and type of the integrity constraint and the name of the corresponding table are also output.
   For the output of table records, you can choose the function SELECT to determine whether all columns (function 1) or only certain columns (function 2) are to be output. (In the case of REFERENTIAL CONSTRAINTS, the records of the referencing table are determined.)
   The output location is determined by another SELECT specification: You can choose to output the hits to the screen (function 1), to a file (function 2) or to both media (function 3). With functions 2 and 3, you must specify a file name. The output is processed in accordance with the SQL.1 output form for SELECT output of the SQL form.
   The SELECT statements executed by the utility monitor to determine the corresponding records are likewise recorded in the log file.

   i To execute these functions you require SELECT privilege to the respective tables. If you do not have this privilege (since this privilege is generally only granted to the owner of a table), the statement will be rejected by the utility monitor with an error message.

5. CHECK FORMAL CATALOG (ALL SPACES)

   When you select this function, the utility monitor determines the names of all user spaces in the database using appropriate SELECT statements on the information schema.
   The CHECK FORMAL *space* utility statement is then issued for each space.

   If the database specified is a partial replication, only the user spaces contained in the partial replication can be checked by the DBH. Checking the spaces contained in the INFORMATION SCHEMA but not in the partial replication results in SQLSTATEs which are entered in the log file assigned with SEE-SSTLOG for SQLSTATEs or in the standard log file. The result of each utility statement CHECK FORMAL *space* is output in the message area of the form. All the other spaces continue to be checked.

Entering configuration data (CNF - CONFIGURATION)

You call the CNF form either by selecting function 1, CONFIGURATION, from the STM - START MENU start form or by entering cnf in the command area. In addition, the CNF form appears automatically when the utility monitor starts in interactive mode and a configuration file has not been assigned or mandatory parameters have not been supplied.

In the CNF form you can check the current configuration data and change it for the duration of the current SQL session.

If you have not changed the configuration data, the values from the configuration file (i.e. the default values) are displayed. To change the configuration data, you simply overwrite the old value with a new value.

In the CNF form you can specify or change the following:

- the authorization key under which the utility monitor is working (by means of the SEE-AUTHID parameter)
- the password for administration via the CALL DML interface (by means of the SEE-ADMIN parameter)
- the default database with which you want to work (by means of the SEE-CATALOG parameter); you can then specify partially qualified object names for this database.
- the default schema with which you want to work (by means of the SEE-SCHEMA parameter); you can then specify partially qualified object names for this schema.
- whether or not the statements issued are to be logged to an instruction file (by means of the SEE-INST-LOGGING parameter)
- the name of the instruction file (by means of the SEE-INPUTLOG parameter). The instruction file can be stored either as a BS2000 file or as an LMS library member. If the input field is too short for the library member name, you can branch to the LIB - LIBRARY ELEMENT form to enter the complete library member name (see page 265).
- whether or not, in the case of logging to an instruction file, the statements created are to be executed (by means of the SEE-EXECUTE parameter)
- that automatic backup be activated or deactivated (by means of the SEE-COPY parameter)
- whether or not the output in the log file is also to be output to SYSLST (by means of the SEE-SYSLST parameter)
- whether or not the information queries from the INF and SNF activities are to be logged in an instruction file which is to be created (by means of the SEE-INFPROT parameter)
- name of the output file to which the results of the information queries are to be written from an instruction file (by means of the SEE-INFOUT parameter)

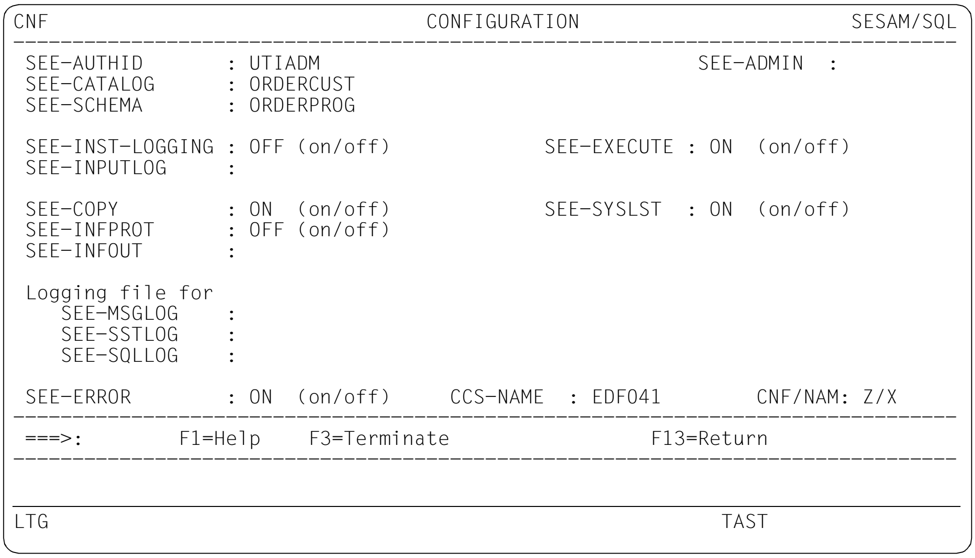
- log files for messages, SQL statements and SQLSTATEs (by means of the SEE-MSGLOG, SEE-SSTLOG and SEE-SQLLOG parameters). The log files can be stored either as a BS2000 file or as an LMS library member. If the input field is too short for the library member name, you can branch to the LIB - LIBRARY ELEMENT form to enter the complete library member name (see page 265).
- the response to error messages of the DBH during processing of the instruction file (by means of the SEE-ERROR parameter)

See also section “Define configuration data” on page 79.

The current DBH name, the configuration name (connection module parameter NAM or CNF) and the CCS name (connection module parameter CCSN) are displayed in the CNF form. You can only edit these parameters in the configuration file.

The CNF form has no continuation forms.

The CNF form



```
CNF                                 CONFIGURATION                          SESAM/SQL
--------------------------------------------------------------------------------------
 SEE-AUTHID        : UTIADM                                    SEE-ADMIN  :
 SEE-CATALOG       : ORDERCUST
 SEE-SCHEMA        : ORDERPROG

 SEE-INST-LOGGING  : OFF (on/off)             SEE-EXECUTE : ON  (on/off)
 SEE-INPUTLOG      :

 SEE-COPY          : ON  (on/off)             SEE-SYSLST  : ON  (on/off)
 SEE-INFPROT       : OFF (on/off)
 SEE-INFOUT        :

 Logging file for
    SEE-MSGLOG     :
    SEE-SSTLOG     :
    SEE-SQLLOG     :

 SEE-ERROR         : ON  (on/off)     CCS-NAME  : EDF041       CNF/NAM: Z/X
--------------------------------------------------------------------------------------
 ===>:         F1=Help     F3=Terminate                 F13=Return
--------------------------------------------------------------------------------------


LTG                                                            TAST
```

The SEE-TRACE parameter is no longer offered in this form in SESAM/SQL V7.0 and higher. It can only be specified in the configuration file.

Creating backup copies and carrying out recovery (COP - COPY & RECOVER / REPLICATION)

You call the COP form either by selecting function 10, “COPY & RECOVER / REPLICATION”, from the STM - START MENU start form or by entering cop in the command area.

In the COP form and its continuation forms, you can:

- back up and recover user spaces and the catalog space
- delete metadata on the backup copies of spaces in the catalog space
- output and delete metadata on database backup copies from the CAT-REC file
- read metadata from a space or a space backup
- create and refresh replications

You are offered automatic backup when you call or exit the ALC, ALS, CRS, IMP and LOD forms and when you exit the CRC form provided that you have set the configuration parameter SEE-COPY=ON (see also the SEE-COPY configuration parameter on page 83).
The COP - COPY & RECOVER / REPLICATION form appears with the default settings and the heading AUTOMATIC BACKUP, COPY.
Press the DUE key to send off the form. This displays the continuation form COP.1. In the COP.1 form you can select the required backup method.
If you press the F13 key, the backup is rejected and you return from the COP form to the original form.
If there are several spaces involved, you are offered another backup when you press the F13 key.

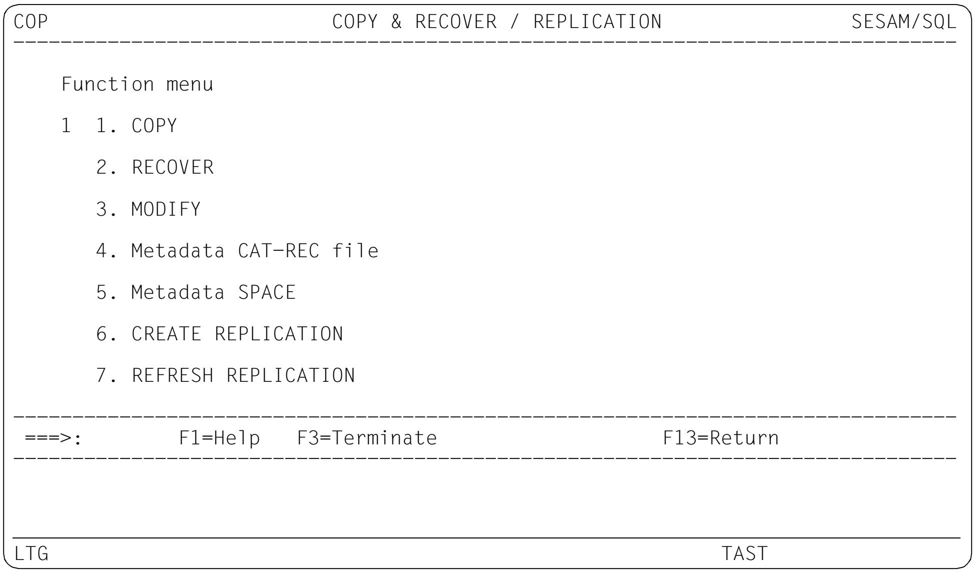
Following a COPY, the utility monitor issues an inquiry to the information schema to determine the version number of the SESAM backup copy[1] and stores this number in a temporary job variable. A job variable is thus created for each user space.

The names of the job variables are assigned as follows:

#SESAM.RU.*space* for each user space.

[1] The SELECT statement is issued implicitly within the dialog to determine the backup copy number. For batch mode, the CMD COPJV *objekt* statement must be issued within the instruction file after each SQL COPY statement. See also page 102 and page 117.

The COP form



```
COP                          COPY & RECOVER / REPLICATION                SESAM/SQL
--------------------------------------------------------------------------------

    Function menu

    1  1. COPY

       2. RECOVER

       3. MODIFY

       4. Metadata CAT-REC file

       5. Metadata SPACE

       6. CREATE REPLICATION

       7. REFRESH REPLICATION

--------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                  F13=Return
--------------------------------------------------------------------------------



LTG                                                         TAST
```

When you select functions 1 - 7, you branch to the corresponding continuation forms.

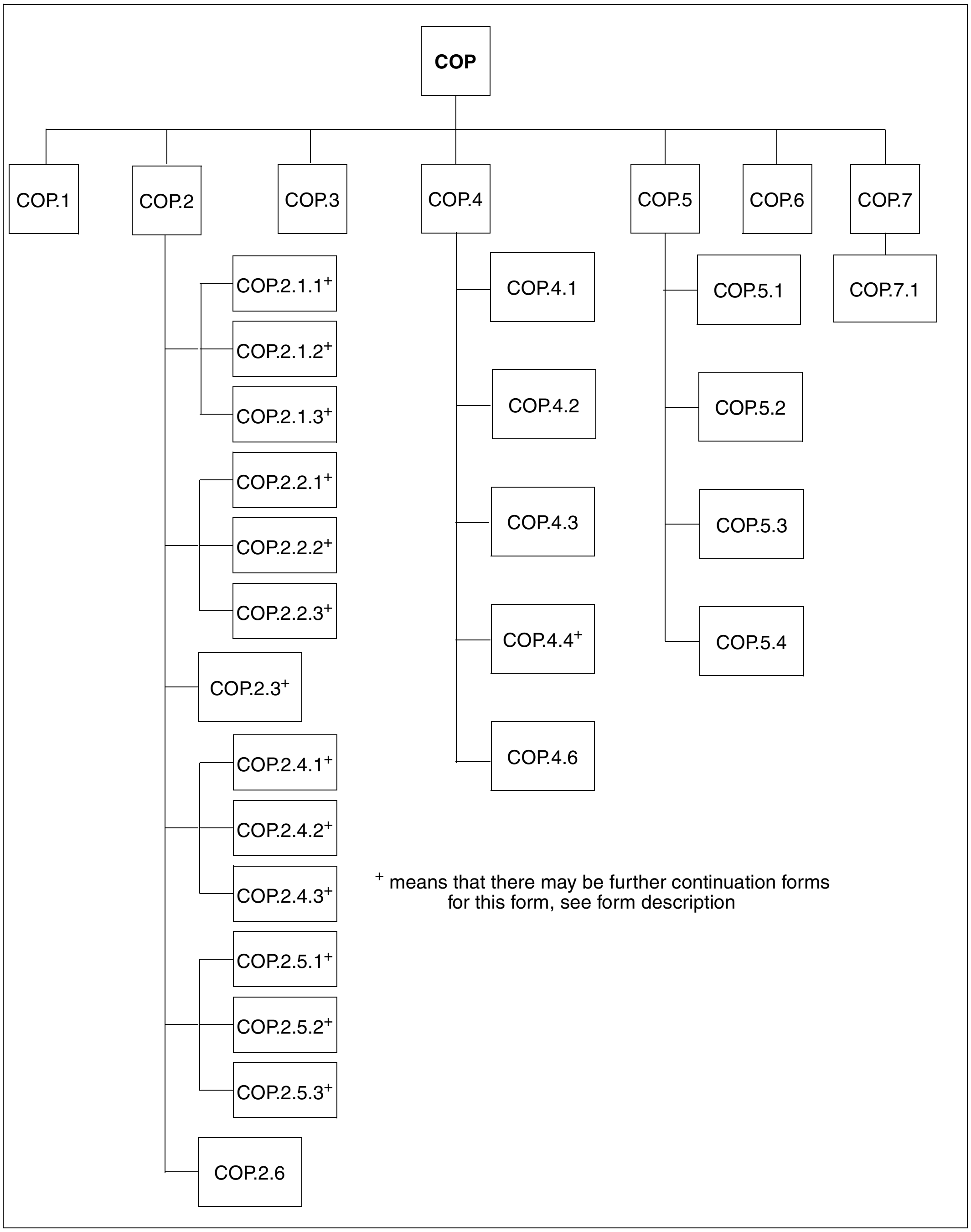
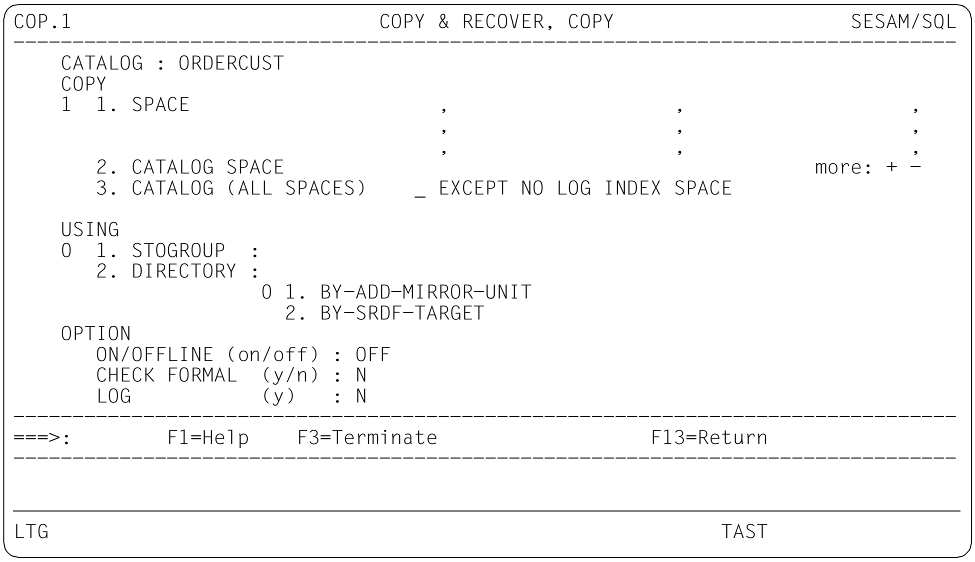


Figure 9: The COP form and its continuation forms

Explanation of the functions

1. COPY

   When you select this function, you branch to the COP.1 continuation form. This allows you to create a backup copy for the catalog space, for all spaces of the database or for specific spaces.



```
COP.1                            COPY & RECOVER, COPY                           SESAM/SQL
-----------------------------------------------------------------------------------------
    CATALOG : ORDERCUST
    COPY
    1  1. SPACE                     ,                   ,                   ,
                                    ,                   ,                   ,
                                    ,                   ,                   ,
       2. CATALOG SPACE                                                 more: + -
       3. CATALOG (ALL SPACES)     _ EXCEPT NO LOG INDEX SPACE

    USING
    0  1. STOGROUP  :
       2. DIRECTORY :
                     0 1. BY-ADD-MIRROR-UNIT
                       2. BY-SRDF-TARGET
    OPTION
       ON/OFFLINE (on/off) : OFF
       CHECK FORMAL  (y/n) : N
       LOG           (y)   : N
-----------------------------------------------------------------------------------------
===>:        F1=Help    F3=Terminate                   F13=Return
-----------------------------------------------------------------------------------------


LTG                                                          TAST
```

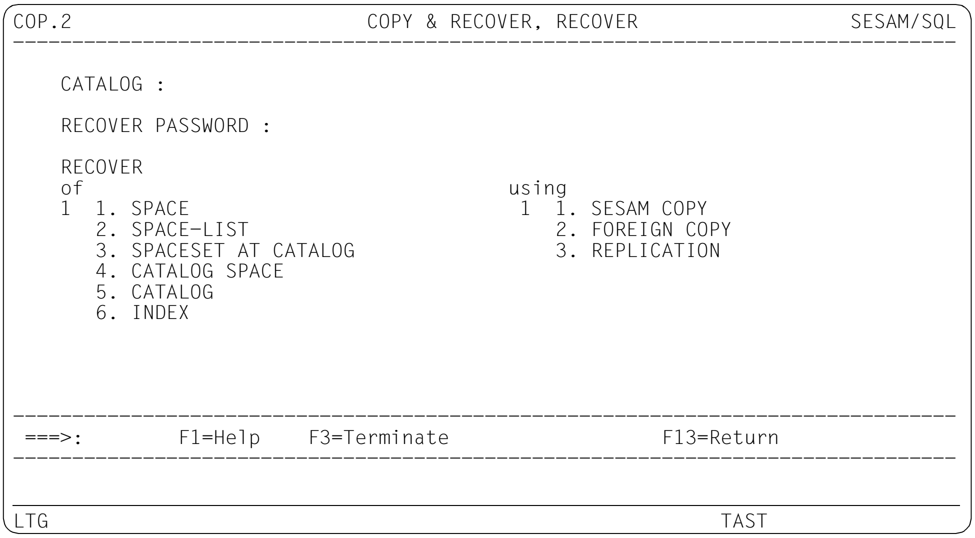
You can use the EXCEPT NO LOG INDEX SPACE parameter to back up the database without taking account of spaces which contain only indexes and are not included in logging.

Database files which are mirrored onto an additional mirror unit can also be backed up in an HSMS archive.
The BY-ADD-MIRROR-UNIT parameter is used to back up database files of an additional mirror unit of the local symmetrix system. The BY-SRDF-TARGET parameter is used to back up database files of the additional mirror unit of the remote symmetrix system. There is no need to concern yourself at this point with splitting up or subsequently synchronizing the mirror units.
You may only specify BY-ADD-MIRROR-UNIT or BY-SRDF-TARGET if the backup is carried out with HSMS and the database files are located in the same mirror unit.
See the utility statement COPY in the “SQL Reference Manual Part 2: Utilities”.

You may only specify CHECK FORMAL=Y together with ONLINE (ON/OFFLINE=ON) and DIRECTORY if only one of the parameters BY-ADD-MIRROR-UNIT or BY-SRDF-TARGET has been selected.

2. RECOVER

When you select this function, you branch to the COP.2 continuation form, which you can use to process user spaces, the catalog space or the entire database and recreate indexes.



```
COP.2                       COPY & RECOVER, RECOVER                      SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG :

    RECOVER PASSWORD :

    RECOVER
    of                                    using
    1  1. SPACE                            1  1. SESAM COPY
       2. SPACE-LIST                            2. FOREIGN COPY
       3. SPACESET AT CATALOG                   3. REPLICATION
       4. CATALOG SPACE
       5. CATALOG
       6. INDEX




--------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                     F13=Return
--------------------------------------------------------------------------------

LTG                                                         TAST
```

See the utility statement RECOVER in the “SQL Reference Manual Part 2: Utilities”. By selecting the object and backup unit you branch to continuation forms for processing the selected objects:

RECOVER of	**where**	**continuation form**
1. SPACE	1. SESAM COPY 2. FOREIGN COPY 3. REPLICATION	COP.2.1.1 COP.2.1.2 COP.2.1.3
2. SPACE-LISTE	1. SESAM COPY 2. FOREIGN COPY 3. REPLICATION	COP.2.2.1 COP.2.2.2 COP.2.2.3
3. SPACESET AT CATALOG		COP.2.3
4. CATALOG SPACE	1. SESAM COPY 2. FOREIGN COPY 3. REPLICATION	COP.2.4.1 COP.2.4.2 COP.2.4.3
5. CATALOG	1. SESAM COPY 2. FOREIGN COPY 3. REPLICATION	COP.2.5.1 COP.2.5.2 COP.2.5.3
6. INDEX		COP.2.6

Table 15: Selection in the form COP2 and continuation forms

Continuation forms COP.2.1.1: RECOVER of SPACE using SESAM COPY
COP.2.1.2: RECOVER of SPACE using FOREIGN COPY
COP.2.1.3: RECOVER of SPACE using REPLICATION

In these forms you can repair a space or reset to a specified backup. A backup which has been generated with the utility statement COPY, a foreign copy or a replication must be present for the space.

You may only select the functions RECOVER RESTART and RECOVER ADJUST in the form COP.2.1.1.

If you insert a cross next to the parameter TO TIMESTAMP, you specify the time stamp of a backup or a mark in the case of RECOVER USING.

If you insert a cross next to the parameter NO INDEX, indexes that have become invalid are not created again but are identified as invalid in the case of RECOVER USING...TO..., RECOVER TO, and RECOVER ADJUST (COP.2.1.1).

If you insert a cross next to the parameter SCOPE PENDING (COP.2.1.1 and 3), a space is only repaired if it is defective.

In the form COP.2.1.3 you specify whether the replication space is to be copied (and therefore continues to exist for the replication) or is to be renamed by selecting function 1: COPY or function 2: RENAME. Function 1 is the default setting.
In the case of RENAME the replication space is renamed as a space in the catalog. The replication space is therefore no longer present and the replication becomes a partial replication. The replication space can be added again with REFRESH SPACE (see page 226) on the basis of a backup. The space removed from the replication with RENAME is marked as deleted in the replication info block of the replication-CAT-REC file. The replication info block can be displayed with the form COP.4, function 6: Read replication info block (see page 224).

COP.2.1.1.1, COP.2.1.2.1, COP.2.1.3.1 continuation forms; select time stamp
If you do not specify a time stamp or if you specify an incomplete time stamp when selecting a function with TIMESTAMP, one of these continuation forms appears, which displays the time stamps of all the recovery unit records from the RECOVERY_UNITS view. You can impose limits on the time stamps to be displayed by specifying a year, a year and month, or a year, month and day in the predecessor form.
You select a time stamp by inserting a cross next to it. It is then transferred to the predecessor form.

COP.2.1.1.2, COP.2.1.2.2, COP.2.1.3.2 continuation forms; warning

CAUTION!
Before the statement RECOVER TO or RECOVER USING...TO... is executed, you branch to the relevant continuation form. This form indicates that all subsequent DA-LOG information will be lost (i.e. it will be logically canceled) after you reset to a recovery unit. It is thus no longer possible to change to the current status, even if the DA-LOG files still exist.
If the question regarding whether you really want to execute the function is answered explicitly with “y” and DUE, the RECOVER TO statement is executed and you return to the predecessor form.
Otherwise, the form is exited without executing the statement. The message `SEE2013   "RECOVER TO” STATEMENT NOT EXECUTED` is then output in the predecessor form.

Continuation forms COP.2.2.1: RECOVER of SPACE-LIST using SESAM COPY
COP.2.2.2: RECOVER of SPACE-LIST using FOREIGN COPY
COP.2.2.3: RECOVER of SPACE-LIST using REPLICATION

In these forms you can repair a selection of spaces with the same time stamp or reset to a specified backup. A backup which has been generated with the utility statement COPY, a foreign copy or a replication must be present for the spaces.

If you insert a cross next to the parameter NO INDEX, indexes that have become invalid are not created again but are identified as invalid in the case of RECOVER TO.

If you insert a cross next to the parameter SCOPE PENDING (COP.2.2.1 and 3), spaces are only repaired if they are defective. You may only insert a cross next to SCOPE PENDING together with RECOVER USING.

In the form COP.2.2.3 you specify, by selecting function 1: COPY or function 2: RENAME, whether the replication spaces are to be copied (and therefore continue to be present for the replication) or renamed. Function 1 is the default setting.
In the case of RENAME the replication spaces are renamed as spaces in the catalog. The replication space is therefore no longer present and the replication becomes a partial replication. The replication spaces can be added again to the replication with REFRESH SPACE (see page 226) on the basis of a backup.
The spaces removed with RENAME from the replication are marked as deleted in the replication info block of the replication CAT-REC file. The replication info block can be displayed with the form COP.4, function 6: Read replication info block (see page 224).

COP.2.2.1.1 continuation form; select time stamp
If you do not specify a time stamp or if you specify an incomplete time stamp when selecting a function with TIMESTAMP, this continuation form appears, which displays the time stamps of all the recovery unit records from the RECOVERY_UNITS view. You can impose limits on the time stamps to be displayed by specifying a year, a year and month, or a year, month and day in the predecessor form.
You select a time stamp by inserting a cross next to it. It is then transferred to the predecessor form.

COP.2.2.1.2, COP.2.2.2.2, COP.2.2.3.2 continuation forms; warning

CAUTION!
Before the statement RECOVER TO is executed, you branch to the relevant continuation form. This form indicates that all subsequent DA-LOG information will be lost (i.e. it will be logically canceled) after you reset to a recovery unit. It is thus no longer possible to change to the current status, even if the DA-LOG files still exist.
If the question regarding whether you really want to execute the function is answered explicitly with “y” and DUE, the RECOVER TO statement is executed and you return to the predecessor form.
Otherwise, the form is exited without executing the statement. The message `SEE2013    "RECOVER TO” STATEMENT NOT EXECUTED` is then output in the predecessor form.

Continuation form COP.2.3: RECOVER of SPACESET AT CATALOG

In this form you can repair a space set or reset it to a specified backup. A unit of several user spaces with a common time stamp can be designated as a space set. Spaces with the same time stamp are generated when several spaces are backed up with a common COPY statement.

If you insert a cross next to the parameter NO INDEX, indexes that have become invalid are not created again but are identified as invalid in the case of RECOVER TO.

If you insert a cross next to the parameter SCOPE PENDING, spaces are only repaired if they are defective. You may only insert a cross next to SCOPE PENDING together with RECOVER USING.

COP.2.3.1 continuation form; select time stamp

If you do not specify a time stamp or if you specify an incomplete time stamp when selecting a function with TIMESTAMP, this continuation form appears, which displays the time stamps of all the recovery unit records from the RECOVERY_UNITS view. You can impose limits on the time stamps to be displayed by specifying a year, a year and month, or a year, month and day in the predecessor form.
You select a time stamp by inserting a cross next to it. It is then transferred to the predecessor form.

COP.2.3.2 continuation form; warning

CAUTION!
Before the statement RECOVER TO is executed, you branch to the continuation form COP.2.3.2. This form indicates that all subsequent DA-LOG information will be lost (i.e. it will be logically canceled) after you reset to a recovery unit. It is thus no longer possible to change to the current status, even if the DA-LOG files still exist.
If the question regarding whether you really want to execute the function is answered explicitly with “y” and DUE, the RECOVER TO statement is executed and you return to the predecessor form.
Otherwise, the form is exited without executing the statement. The message `SEE2013   "RECOVER TO” STATEMENT NOT EXECUTED` is then output in the predecessor form.

Continuation forms COP.2.4.1: RECOVER of CATALOG SPACE using SESAM COPY
COP.2.4.2: RECOVER of CATALOG SPACE using FOREIGN COPY
COP.2.4.3: RECOVER of CATALOG SPACE using REPLICATION

In these forms you can repair the catalog space or reset to a specified backup. For the catalog space, a backup which has been generated with the utility statement COPY, a foreign copy or a replication must be present for the space.

Form COP.2.4.3, function 1: RECOVER USING REPLICATION WITH CAT-REC
When this function is selected, the catalog space is repaired by means of the replication specified in the REPLICATION field. The modifications resulting from the log files identified by CAT-REC are applied accordingly.
The specification of a CAT-REC file is mandatory.
By selecting function 1: COPY or function 2: RENAME you determine whether the catalog space of the replication is to be copied (and therefore continues to exist for the replication) or renamed. Function 1 is the default setting.

Form COP.2.4.3, function 2: RECOVER TO REPLICATION
When you select this function, the catalog space is reset to the state of the replication specified in the REPLICATION field.
By selecting function 1: COPY or function 2: RENAME you determine whether the catalog space of the replication is to be copied (and therefore continues to exist for the replication) or renamed. Function 1 is the default setting.

Note on “RENAME” in the form COP.2.4.3, functions 1 and 2:
The replication catalog space is renamed as the catalog space of the catalog. The replication catalog space therefore no longer exists and the replication is defective. It must be created again with CREATE REPLICATION.

COP.2.4.1.1 continuation form; select time stamp
If you do not specify a time stamp or if you specify an incomplete time stamp when selecting a function with TIMESTAMP, the form COP.2.4.1 reappears, supplemented by the input field “CAT-REC” and a message requesting you to specify the current CAT-REC file. After you have entered the name of the CAT-REC file and sent off the form COP.2.4.1, this continuation form appears, which displays the time stamps of all the recovery unit records from the CAT-REC file. You can impose limits on the time stamps to be displayed by specifying a year, a year and month, or a year, month and day in the predecessor form.
You select a time stamp by inserting a cross next to it. It is then transferred to the predecessor form.

COP.2.4.1.2, COP.2.4.2.2, COP.2.4.3.2 continuation forms; warning

CAUTION!
Before the statement RECOVER TO is executed, you branch to the relevant continuation form. This form indicates that all subsequent DA-LOG information will be lost (i.e. it will be logically canceled) after you reset to a recovery unit. It is thus no longer possible to change to the current status, even if the DA-LOG files still exist.
If the question regarding whether you really want to execute the function is answered explicitly with “y” and DUE, the RECOVER TO statement is executed and you return to the predecessor form.
Otherwise, the form is exited without executing the statement. The message `SEE2013   "RECOVER TO” STATEMENT NOT EXECUTED` is then output in the predecessor form.

Continuation forms COP.2.5.1: RECOVER of CATALOG using SESAM COPY
COP.2.5.2: RECOVER of CATALOG using FOREIGN COPY
COP.2.5.3: RECOVER of CATALOG using REPLICATION

In these forms you can repair the entire database or reset to a specified backup or to a freely selectable time. When you specify a freely selectable time, the last backup before this time is read in and the changes are then applied to the database to bring it up to the status at the time specified. For the database, a backup which has been generated with the utility statement COPY, a foreign copy or a replication must be present for the space.

If you insert a cross next to the parameter SCOPE PENDING (COP.2.5.1), the catalog space is always repaired. The user spaces are only repaired if they are defective.

If you insert a cross next to the parameter GENERATE INDEX ON NO LOG INDEX SPACE, no backups are read in of spaces that contain only indexes and are not in the logic data backup. These spaces must not form part of the replication. The index spaces are reset and the indexes are created again.

Form COP.2.5.3, function 1: RECOVER USING REPLICATION WITH CAT-REC
When this function is selected the database is repaired by means of the replication specified in the REPLICATION field. The modifications resulting from the log files identified by CAT-REC are applied accordingly.
The specification of a CAT-REC file is mandatory.
By selecting function 1: COPY or function 2: RENAME, you specify whether the replication is to be copied (and therefore continues to exist) or renamed.
Function 1 is the default setting.

Form COP.2.5.3, function 2: RECOVER TO REPLICATION
When this function is selected the database is reset to the state of the replication specified in the REPLICATION field.
By selecting function 1: COPY or function 2: RENAME, you specify whether the replication is to be copied (and therefore continues to exist) or renamed.
Function 1 is the default setting.

COP.2.5.1.1 continuation form; select time stamp
If you do not specify a time stamp or if you specify an incomplete time stamp when selecting a function with TIMESTAMP, form COP.2.5.1 reappears, supplemented by the input field “CAT-REC:” and a message which requests you to specify the current CAT-REC file. After you have entered the name of the CAT-REC file and sent off the form COP.2.5.1, this continuation form appears, which displays the time stamps of all the recovery unit records from the CAT-REC file. You can impose limits on the time stamps to be displayed by specifying a year, a year and month, or a year, month and day in the predecessor form.

You select a time stamp by inserting a cross next to it. It is then transferred to the predecessor form.
When ANY TIMESTAMP is specified, only the values for seconds and milliseconds may be omitted. These are padded internally with zeros.

COP.2.5.1.2, COP.2.5.2.2, COP.2.5.3.2 continuation forms; warning

CAUTION!
Before the statement RECOVER TO is executed, you branch to the relevant continuation form. This form indicates that all subsequent DA-LOG information will be lost (i.e. it will be logically canceled) after you reset to a recovery unit. It is thus no longer possible to change to the current status, even if the DA-LOG files still exist.
If the question regarding whether you really want to execute the function is answered explicitly with “y” and DUE, the RECOVER TO statement is executed and you return to the predecessor form.
Otherwise, the form is exited without executing the statement. The message `SEE2013   "RECOVER TO” STATEMENT NOT EXECUTED` is then output in the predecessor form.

Continuation form COP.2.6: INDEX

In this form you can recreate a specified index, all the indexes in a certain table or all the indexes in tables which are located in a particular space.

3. MODIFY

   When you select this function, you branch to the COP.3 continuation form. This allows you to delete the following in the metadata of SESAM backup copies:

   - Delete records from the RECOVERY_UNITS and DA_LOGS catalog tables for all or specific user spaces
     (selection “1” (ALL SPACES) or selection “3” (SPACE ...)).
   - Delete records from the CAT-REC file
     (selection “2” (CATALOG-SPACE)
     In this case the catalog must be entered in a database catalog when the function is performed (online update of the CAT-REC file). The catalog may already be open.

   You can also specify whether all records of the specified space are to be deleted or only records that are older than a specified number of days or a specified date. The UNRESTRICTED parameter simplifies your work with foreign copies and should only be specified when you want to use a foreign copy.

   CAUTION!
   The UNRESTRICTED option may only be used for selection “1” (ALL SPACES) in conjunction with selection “2” (DELETE COPY-AGE) or “3” (DELETE COPY-DATE).
   If you specify UNRESTRICTED, all the entries of the specified age are deleted from RECOVERY_UNITS and DA_LOGS (with the exception of the last entry).
   A RECOVER is no longer possible if the required entries have been deleted from the RECOVERY_UNITS and DA_LOGS catalog tables.

   See the utility statement MODIFY RECOVERY in the “SQL Reference Manual Part 2: Utilities”.

4. Metadata CAT-REC file

   When you select this function, you branch to the COP.4 continuation form. This allows you to read the metadata of the CAT-REC file, to delete recovery unit records from the CAT-REC file, to delete CAT-LOG records after the last recovery unit record or to read the replication info block.

   i You can only edit the metadata in the CAT-REC file if the utility monitor was started with the independent DBH or without a DBH.

   You can output the information to the screen, a file (BS2000 file or a member of an LMS library) or both by making the appropriate entry in the “Output on” field. See also the LIB - LIBRARY ELEMENT form on page 265.
   When you select functions 1 - 4 and 6 you branch to continuation forms to process the selected objects. Function 5 is executed immediately.

   COP.4 continuation form, function 1: Read identification and CREATE-CATALOG record
   When you select this function, you branch to the COP.4.1 continuation form, in which the identification record and CREATE-CATALOG record of the CAT-REC file are output.

   COP.4 continuation form, function 2: Read CATALOG-LOGGING records
   When you select this function, you branch to the COP.4.2 continuation form, in which the CAT-REC file is output.

   COP.4 continuation form, function 3: Read RECOVERY UNIT records
   When you select this function, you branch to the COP.4.3 continuation form, in which the recovery unit records of the CAT-REC file are output.
   You can restrict the records to be output by specifying the year, year and month, or year month and day in the input field (DATE). If you do not specify anything here, all recovery unit records are output in the COP.4.3 continuation form.

   COP.4 continuation form, function 4: Delete RECOVERY UNIT records
   When you select this function, you branch to the COP.4.4 continuation form. This allows you to delete recovery unit records from the CAT-REC file. The associated CAT-LOG records are also deleted. You can delete records before or after a specified time.
   When you delete recovery unit records from the CAT-REC file, it must not be opened by a DBH.

   i When you delete recovery unit records from the CAT-REC file, the file may not be opened by a DBH (offline update of the CAT-REC file). An online update of the CAT-REC file can be performed with the MODIFY function, see page 222.

COP.4.4 continuation form, function 1: Records before TIMESTAMP and function 2: Records after TIMESTAMP

If you do not specify a time stamp or if you specify an incomplete time stamp when selecting this function, the COP.4.4.1 continuation form appears, which displays the time stamps of all the recovery unit records from the CAT-REC file (You can impose limits on the time stamps to be displayed by specifying a year, a year and month, or a year, month and day in the COP.4.4 form).
You select a time stamp by inserting a cross next to it. It is then transferred to the predecessor form.

COP.4 continuation form, function 5: Delete CAT-LOG records after the last RECOVERY-UNIT record

When you select this function, the CAT-LOG records are deleted.
A temporary job variable with the name “#SESAM.RU.CATALOG” is created in which the version number (COPY-NUMBER) of the last recovery unit record is stored (with backup unit, version, time stamp, and the indicator “D”).
For more information on job variables, see page 117 and the “Database Operation” manual.
Function 5 can also be called in batch mode from an instruction file using CMD CATREC DEL_LAST_RU *file_name* (see page 102).

Continuation form COP.4, function 6 Read replication info block

When you select this function you branch to continuation form COP.4.6., in which the spaces currently or previously belonging to the replication are output. These are read from the replication info block of the CAT-REC file.

5. Metadata SPACE

   When you select this function, you branch to the COP.5 continuation form. This allows you to output the metadata of a space.
   You can output the information to the screen, a file (BS2000 file or a member of an LMS library) or both by making the appropriate entry in the “Output on” field. See also the LIB - LIBRARY ELEMENT form on page 265.
   When you select a function, you branch to the corresponding continuation form, in which you process the selected objects:

   COP.5 continuation form, function 1: Read SPACE metadata
   When you select this function, you branch to the COP.5.1 continuation form, in which the metadata of the specified space is output.

   COP.5 continuation form, function 2: Read TABLE metadata
   When you select this function, you branch to the COP.5.2 continuation form, in which the metadata of all tables in the specified space is output.

   COP.5 continuation form, function 3: Read INDEX metadata
   When you select this function, you branch to the COP.5.3 continuation form, in which the metadata of all the indexes of the tables in the specified space is output.

   COP.5 continuation form, function 4: Read COLUMN metadata
   When you select this function, you branch to the COP.5.4 continuation form, in which the metadata of all the columns of the tables in the specified space is output.

6. CREATE REPLICATION

   When you select this function you branch to the COP.6 continuation form, in which you create a replication from a SESAM backup copy (COPY CATALOG). The name of the database and the name of a CAT-REC copy must be specified.
   If you want to create a partial replication, the appropriate user spaces must be entered in the “FOR SPACES” input table.
   If FOREIGN is specified, the replication is created from a foreign copy: The files of the foreign copy used are derived from the name of the CAT-REC file. RENAME causes the replication to be created by renaming the foreign copy.

   See the utility statement CREATE REPLICATION in the “SQL Reference Manual Part 2: Utilities”.

7. REFRESH REPLICATION

   When you select this function you branch to the COP.7 continuation form, in which an existing replication is refreshed using log files.

   By selecting the function you may specify whether

   1. the entire replication is refreshed (REFRESH REPLICATION).
      The name of the replication catalog and the name of a CAT-REC copy must be specified. The modifications resulting from the log files identified by CAT-REC are applied accordingly.
   2. a selection of spaces in the replication is refreshed
      (REFRESH REPLICATION FOR SPACE).
      The name of the replication catalog, the name of a CAT-REC copy and the spaces must be specified. The modifications resulting from the log files identified by CAT-REC are applied accordingly.
   3. spaces are added to a replication (REFRESH SPACE).
      The space names and data on the backup or foreign copy must be specified.

   COP.7.1 continuation form; select time stamp
      If you do not specify a time stamp or if you specify an incomplete time stamp when selecting a backup with TIMESTAMP, this continuation form appears, which displays the time stamps of all the recovery unit records from the RECOVERY_UNITS view. You can impose limits on the time stamps to be displayed by specifying a year, a year and month, or a year, month and day in the predecessor form.
      You select a time stamp by inserting a cross next to it. It is then transferred to the predecessor form.

   See the utility statements REFRESH REPLICATION and REFRESH SPACE in the "SQL Reference Manual Part 2: Utilities".

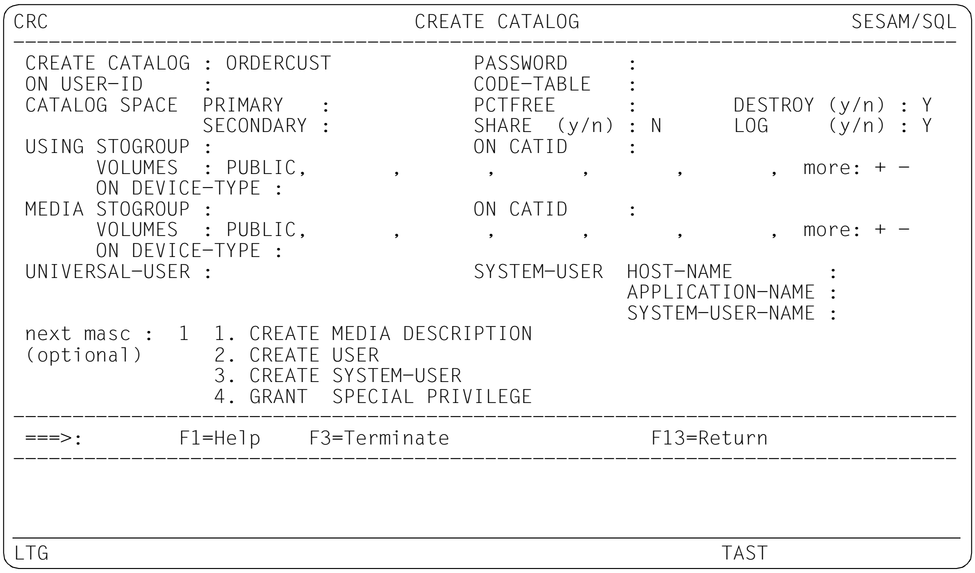
Creating a catalog space (CRC - CREATE CATALOG)

You call the CRC form by selecting function 13, CREATE CATALOG, from the STM - START MENU start form.

In the CRC form and its continuation forms, you can create a catalog space with all its properties.

You are offered an automatic backup after the activity provided that you have specified the configuration parameter SEE-COPY = ON (see also the SEE-COPY configuration parameter on page 83).
See the utility statement CREATE CATALOG in the “SQL Reference Manual Part 2: Utilities”.

The CRC form


```
CRC                                CREATE CATALOG                                SESAM/SQL
-------------------------------------------------------------------------------
 CREATE CATALOG : ORDERCUST           PASSWORD     :
 ON USER-ID     :                     CODE-TABLE   :
 CATALOG SPACE  PRIMARY   :           PCTFREE      :       DESTROY (y/n) : Y
                SECONDARY :           SHARE  (y/n) : N     LOG     (y/n) : Y
 USING STOGROUP :                     ON CATID     :
       VOLUMES  : PUBLIC,       ,       ,       ,       ,       ,  more: + -
       ON DEVICE-TYPE :
 MEDIA STOGROUP :                     ON CATID     :
       VOLUMES  : PUBLIC,       ,       ,       ,       ,       ,  more: + -
       ON DEVICE-TYPE :
 UNIVERSAL-USER :                     SYSTEM-USER  HOST-NAME        :
                                                   APPLICATION-NAME :
                                                   SYSTEM-USER-NAME :
 next masc :  1  1. CREATE MEDIA DESCRIPTION
 (optional)      2. CREATE USER
                 3. CREATE SYSTEM-USER
                 4. GRANT  SPECIAL PRIVILEGE
-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------



LTG                                                         TAST
```

To create a catalog space without specifying it more closely, the entries in the CRC form suffice. You do not need to select a function from the function menu to branch to a continuation form. If you specify LOG=Y, the CAT-REC file and the CAT-LOG file are created automatically on the storage group defined with “MEDIA STOGROUP”. If you do not define a storage group with “MEDIA STOGROUP”, the CAT-REC file and the CAT-LOG file are created on the standard storage group D0STOGROUP.

In SESAM/SQL V5.0 and higher the default value for `CODE-TABLE` has changed, see the "SQL Reference Manual Part 2: Utilities".

When a coded character set is specified (`CODE-TABLE` not equal to `*NONE`), only users who have specified the same coded character set in the user configuration file (connection module parameter CCSN) can access the database.
The SNF, INF and ALC functions can also be executed when different character sets are used.

When you select functions 1 - 4, you branch to the corresponding continuation forms.

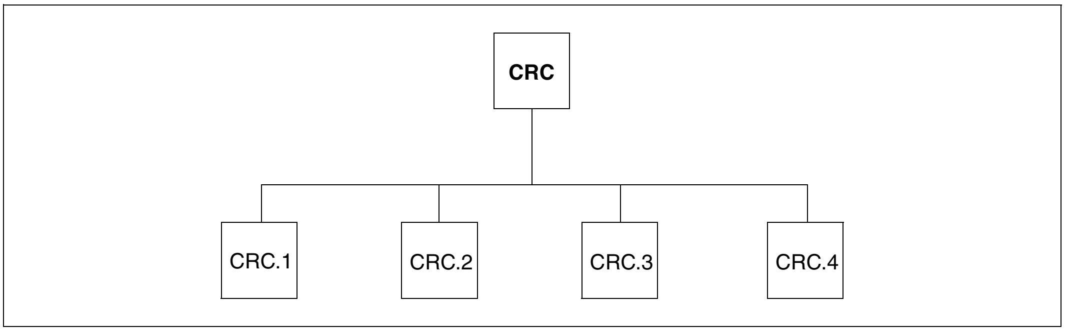


Figure 10: The CRC form and its continuation forms

Explanation of the functions

1. CREATE MEDIA DESCRIPTION

   When you select this function, you branch to the CRC.1 continuation form. This allows you to:

   - specify the properties of the database-specific DA-LOG or PBI files
   - specify the media on which these files are to be stored

   See the utility statement CREATE MEDIA DESCRIPTION FOR ... in the "SQL Reference Manual Part 2: Utilities".

2. CREATE USER

   When you select this function, you branch to the CRC.2 continuation form. This allows you to create one or more authorization keys.
   See the SQL statement CREATE USER in the "SQL Reference Manual Part 1: SQL Statements".

3. CREATE SYSTEM-USER

   When you select this function, you branch to the CRC.3 continuation form. This allows you to create one or more system entries.

   See the SQL statement CREATE SYSTEM_USER in the “SQL Reference Manual Part 1: SQL Statements”.

4. GRANT SPECIAL PRIVILEGE

   When you select this function, you branch to the CRC.4 continuation form. This allows you to grant special privileges to one or more authorization keys.

   See the SQL statement GRANT in the “SQL Reference Manual Part 1: SQL Statements”.

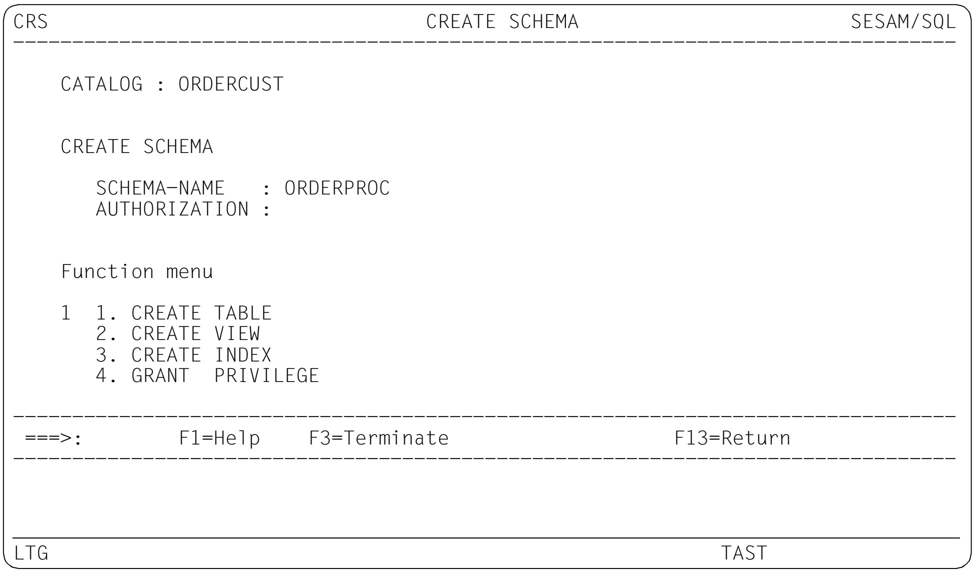
Create a schema (CRS - CREATE SCHEMA)

You call the CRS form by selecting function 15, CREATE SCHEMA, from the STM - START MENU start form.

In the CRS form and its continuation forms, you create a schema in a database. You can define tables, views and indexes for the schema, and grant privileges.

You are offered an automatic backup before and after the activity provided that you have specified the configuration parameter SEE-COPY = ON (see also the SEE-COPY configuration parameter on page 83).
See the SQL statement CREATE SCHEMA in the “SQL Reference Manual Part 1: SQL Statements”.

The CRS form


```
CRS                                  CREATE SCHEMA                          SESAM/SQL
--------------------------------------------------------------------------------

    CATALOG : ORDERCUST


    CREATE SCHEMA

       SCHEMA-NAME   : ORDERPROC
       AUTHORIZATION :


    Function menu

    1  1. CREATE TABLE
       2. CREATE VIEW
       3. CREATE INDEX
       4. GRANT  PRIVILEGE

--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                   F13=Return
--------------------------------------------------------------------------------



LTG                                                         TAST
```

When you select function 1, you branch to the CRT form (see also page 233).

When you select functions 2 - 4, you branch to continuation forms.

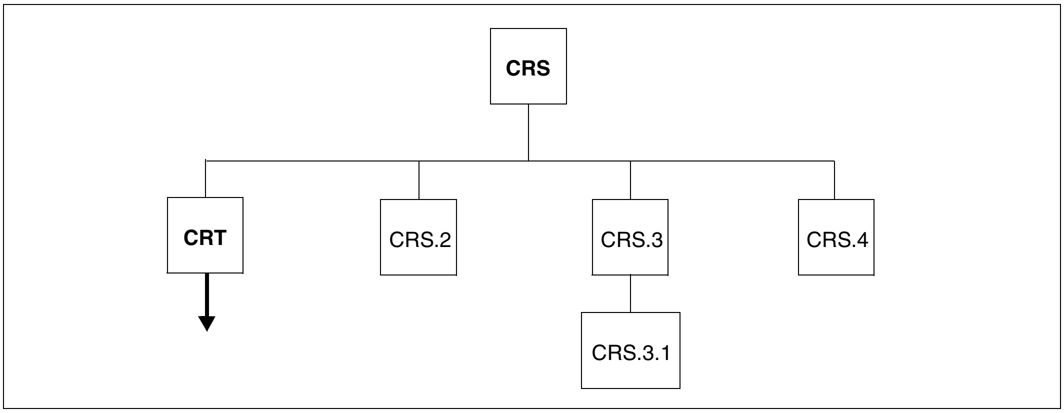


Figure 11: The CRS form and its continuation forms

Explanation of the functions

1. CREATE TABLE

   When you select this function, you branch to the CRT form (see also page 233).

2. CREATE VIEW

   When you select this function, you branch to the CRS.2 continuation form. This allows you to create a view.

   See the SQL statement CREATE VIEW in the “SQL Reference Manual Part 1: SQL Statements”.

3. CREATE INDEX

   When you select this function, you branch to the CRS.3 continuation form. This allows you to create one or more indexes.

   See the SQL statement CREATE INDEX in the “SQL Reference Manual Part 1: SQL Statements”.

   CRS.3 continuation form, function 1: Create INDEX DEFINITION

   When you select this function you branch to the CRS.3.1 continuation form, in which you can define an index. The index name and the name of at least one column (COLUMN) must be specified. You are only permitted to specify the length if the associated COLUMN is of data type (N)CHAR, (N)VARCHAR or is a data type of SESAM V12 or older.
   When you press the DUE key the entries are accepted and the CRS.3.1 form is displayed again, thus allowing you to define the next index.
   The defined indexes are not created until you return to the CRS.3 form and select function 3 “Execute INDEX DEFINITION”.

4. GRANT PRIVILEGE

   When you select this function, you branch to the CRS.4 continuation form. This allows you to grant privileges to other authorization keys.

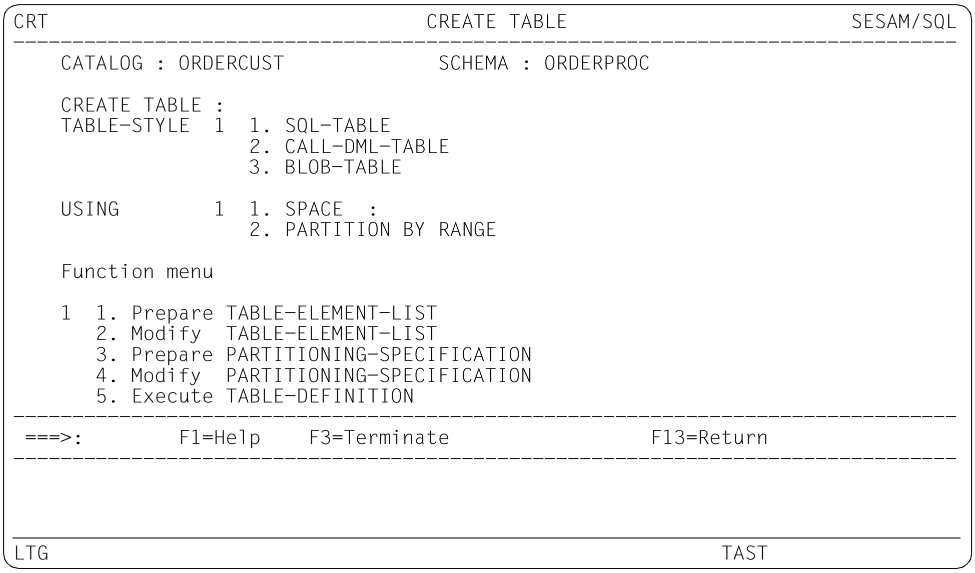
   See the SQL statement GRANT in the “SQL Reference Manual Part 1: SQL Statements”.

Creating a base table (CRT - CREATE TABLE)

You call the CRT form by selecting function 17, CREATE TABLE, from the STM - START MENU start form.

In the CRT form and its continuation forms, you can create an SQL table, a CALL DML table or a BLOB table with all the associated objects. The table can also be partitioned.
See the SQL statement CREATE TABLE in the “SQL Reference Manual Part 1: SQL Statements”.

The CRT form



```
CRT                                CREATE TABLE                              SESAM/SQL
--------------------------------------------------------------------------------
    CATALOG : ORDERCUST              SCHEMA : ORDERPROC

    CREATE TABLE :
    TABLE-STYLE  1  1. SQL-TABLE
                    2. CALL-DML-TABLE
                    3. BLOB-TABLE

    USING        1  1. SPACE   :
                    2. PARTITION BY RANGE

    Function menu

    1  1. Prepare TABLE-ELEMENT-LIST
       2. Modify  TABLE-ELEMENT-LIST
       3. Prepare PARTITIONING-SPECIFICATION
       4. Modify  PARTITIONING-SPECIFICATION
       5. Execute TABLE-DEFINITION
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------



LTG                                                           TAST
```

The procedure differs for non-partitioned and partitioned tables:

- If you wish to create a non-partitioned table, select USING SPACE (“1”, default). To create the new table you must then execute both function 1 and subsequently function 5. If you wish to modify a non-partitioned table, you must execute both function 2 and subsequently function 5.

- If you wish to create a partitioned table, select USING PARTITION BY RANGE (“2”). You then execute function 1 for the initial creation, you define the partitions with function 3, and you create the table with function 5.
  If you wish to modify a partitioned table, you must execute functions 2, 3 and 5. If you wish to modify individual partitions, you must execute functions 4 and 5.

When you select functions 1, 2, 3 and 4, you branch to continuation forms. The specification for USING is evaluated only after function 1 has been executed.

When you select function 5, the statement is executed immediately, provided function 1 or function 2 (non-partitioned table) or function 3 or 4 (partitioned table) has been executed. See also section “Creating a database” on page 19ff.

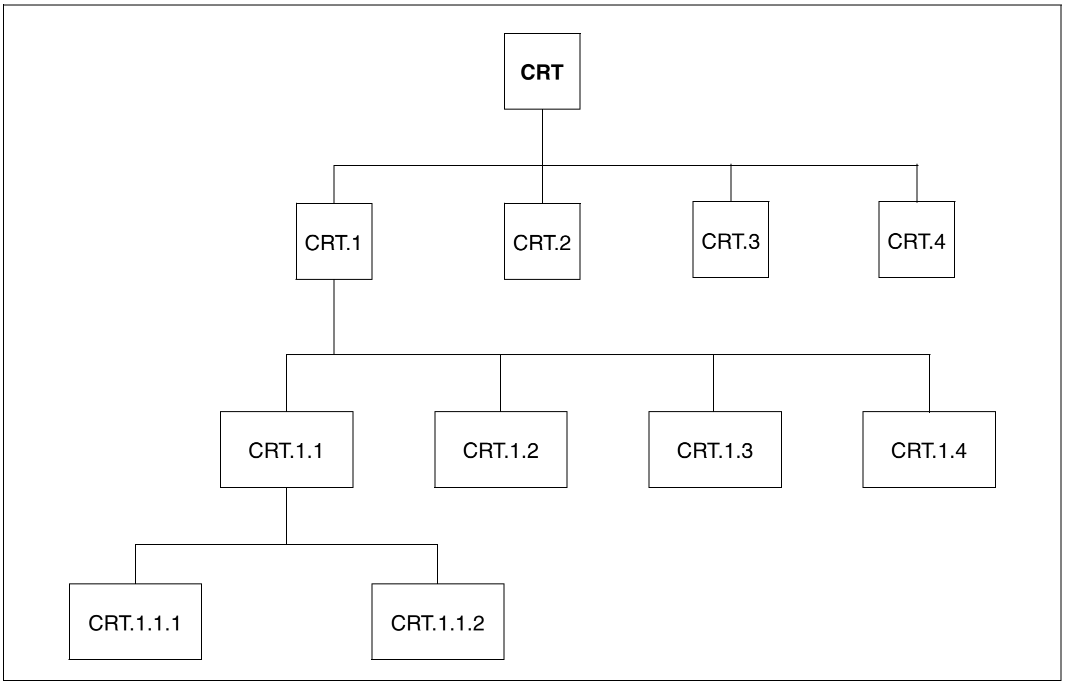


Figure 12: The CRT form and its continuation forms

Explanation of the functions

1. Prepare TABLE-ELEMENT-LIST

- When you select table style 1, SQL-TABLE, from the CRT form, you branch to the CRT.1 continuation form. This allows you to define a column or table constraint for an SQL table.

  CRT.1 continuation form, function 1: COLUMN-DEFINITION
  When you select this function, you branch to the CRT.1.1 continuation form. This allows you to define a column.

  CRT.1.1 continuation form, DEFAULT-CLAUSE function
  When you enter y for this function, you branch to the CRT.1.1.1 continuation form. This allows you to set a default value.
  Not until you return to the CRT.1.1 form is the definition displayed. If the set default value is a literal, a maximum of 42 characters are displayed.

  CRT.1.1 continuation form, CHECK function
  When you select this function, you branch to the CRT.1.1.2 continuation form. This allows you to define a search condition for a check constraint.

  CRT.1 continuation form,
  function 2: UNIQUE/PRIMARY TABLE-CONSTRAINT definition
  When you select this function, you branch to the CRT.1.2 continuation form. This allows you to define a UNIQUE constraint for a table constraint.

  CRT.1 continuation form,
  function 3: REFERENTIAL TABLE-CONSTRAINT definition
  When you select this function, you branch to the CRT.1.3 continuation form. This allows you to define a referential constraint for a table constraint.

  CRT.1 continuation form,
  function 4: CHECK TABLE-CONSTRAINT definition
  When you select this function, you branch to the CRT.1.4 continuation form. This allows you to define a search condition for a table constraint.

- When you select table style 2, CALL-DML-TABLE, from the CRT form, you branch to the CRT.2 continuation form. This allows you to define a column for a CALL DML table.

- When you select table style 3, BLOB-TABLE, from the CRT form, you branch to the CRT.3 continuation form. This allows you to define a BLOB table. You can also specify a MIME type and a USAGE and enter a user-defined text. There are no restrictions on the format of this user-defined text.

2. Modify TABLE-ELEMENT-LIST

   When you select this function, you branch to the CRT.1 or CRT.2 continuation form. This allows you to modify the defined table format of a table. You proceed as described for function 1, Prepare TABLE-ELEMENT-LIST.

   You can call function 2 either after function 1, Prepare TABLE-ELEMENT-LIST, to modify the defined table format of a table, or after function 5, Execute TABLE-DEFINITION, to correct the defined table format of a table after the CREATE-TABLE statement has been rejected with an error message.

3. Prepare PARTITIONING-SPECIFICATION

   When you select this function, you branch to the CRT.4 continuation form. This allows you to define the properties of a partition for partitioned tables.

   The entries correspond to the specifications in the SQL statement CREATE TABLE, see the "SQL Reference Manual Part 1: SQL Statements":

   - Serial number of the partition (1 through 16)
   - Comparison operator < or <= and the upper limit of the primary key interval (without parentheses). If the primary key consists of multiple columns, the individual values must be separated by commas.
   - Space name of the partition

   After you transfer the form with the DUE key you are shown the CRT.4 form again in order to define the next partition's properties.

   For the last partition you must check mark LAST PARTITION; no upper limit for the primary key interval may be specified here.
   After the last partition has been defined you return to the CRT form by pressing the F13 key or entering F13 in the command area.

4. Modify PARTITIONING-SPECIFICATION

   When you select this function, you branch to the CRT.4 continuation form. This allows you to modify the properties of a partition for partitioned tables. The procedure is the same as that described for function 3 “Prepare PARTITIONING-SPECIFICATION”.

   You can call function 4 either after function 3 “Prepare PARTITIONING-SPECIFICATION” in order to modify a partition’s properties, or after function 5 “Execute TABLE-DEFINITION” in order to correct a defined partition after the CREATE-TABLE statement has been rejected with an error message.

5. Execute TABLE-DEFINITION

   This creates the base table. The parameters are used that you have fully defined in the continuation forms of function 1 to 4.

Exporting a base table (EXP - EXPORT TABLE)

You can call the EXP form either by selecting function 19, EXPORT TABLE, in the start form STM - START MENU, or by typing “exp” in the command area.

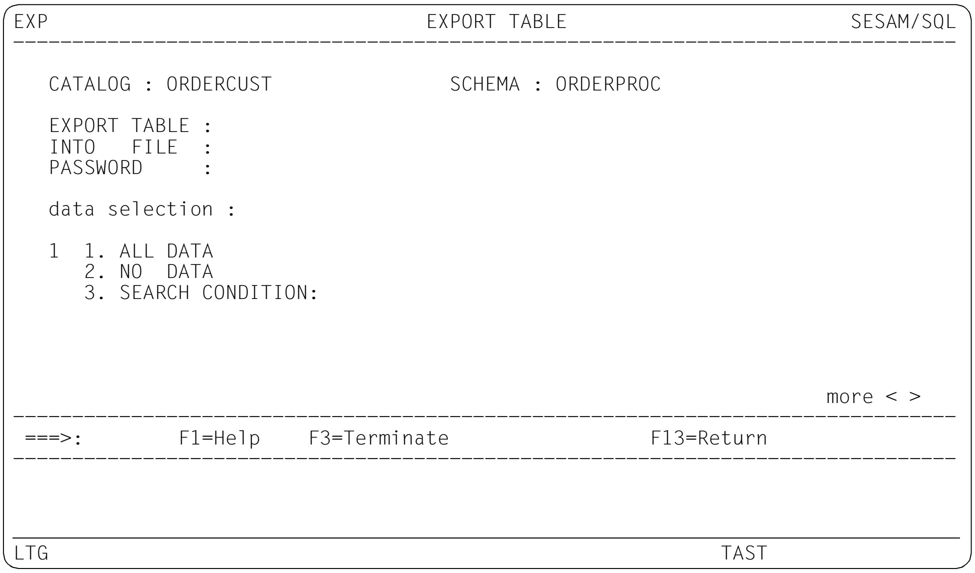
You use the EXP form to export a base table to a BS2000 file. This file is known as the export file. The base table can also be partitioned. You can choose whether to store all the user data, selected data or only metadata in this file. You can then use this export file to import a base table with this structure into any SESAM/SQL database.
See the EXPORT TABLE utility statement in the “SQL Reference Manual Part 2: Utilities”.

If you choose function 3 then you must specify a full search condition. It may not contain any user variables or question marks as placeholders

The number of records that are loaded from the base table into the export file is displayed in the message area. If other utility monitor messages or SQLSTATEs are present, this is indicated by a “M+-” in the command area. In this case, you can use “m+” and “m-” in the command area to scroll through the messages.

The EXP form has no continuation screens.

The EXP form



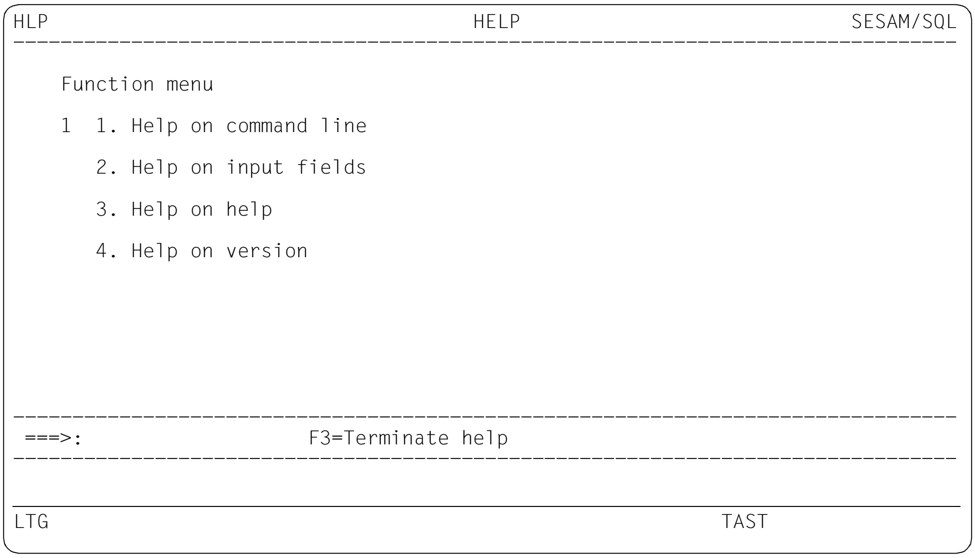
```
EXP                                  EXPORT TABLE                          SESAM/SQL
--------------------------------------------------------------------------------

   CATALOG : ORDERCUST                SCHEMA : ORDERPROC

   EXPORT TABLE :
   INTO   FILE  :
   PASSWORD     :

   data selection :

   1  1. ALL DATA
      2. NO  DATA
      3. SEARCH CONDITION:



                                                                       more < >
--------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                F13=Return
--------------------------------------------------------------------------------



LTG                                                         TAST
```

Requesting help (HLP - HELP)

You call the HLP form either by selecting function 12, HELP, from the STM - START MENU form or by entering hlp in the command area.

In the HLP form and its continuation forms, you can request global help information on working with the utility monitor and the forms.
See also section “Requesting help information on forms” on page 154.

The HLP form



```
HLP                                   HELP                               SESAM/SQL
-----------------------------------------------------------------------------------

    Function menu

    1  1. Help on command line

       2. Help on input fields

       3. Help on help

       4. Help on version







-----------------------------------------------------------------------------------
 ===>:                   F3=Terminate help
-----------------------------------------------------------------------------------

LTG                                                          TAST
```

When you select functions 1 - 4, you branch to the corresponding continuation forms.

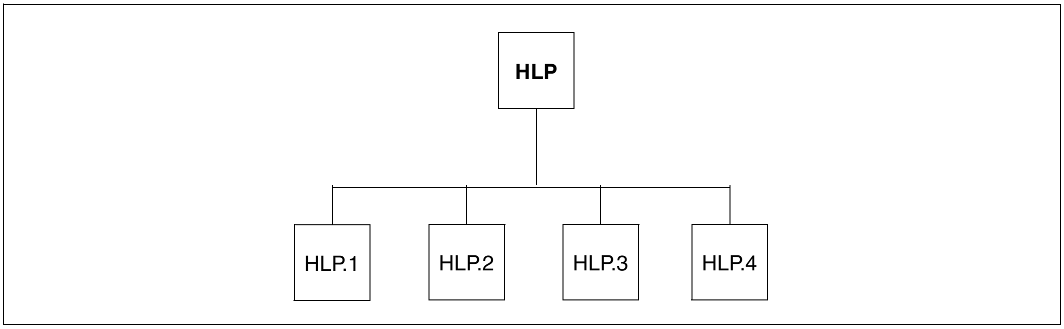


Figure 13: The HLP form and its continuation forms

Explanation of the functions

1. Help on command line

   When you select this function, you branch to the HLP.1 continuation form. This displays the entries that can be made in the command area. The output corresponds to that in the HLP.CMD and HLP.CMD.1 help forms, except that, in the HLP.1 continuation form, the output fields are not preceded by input fields.

2. Help on input fields

   When you select this function, you branch to the HLP.2 continuation form. This displays a help text on how to work with the input fields.

3. Help on help

   When you select this function, you branch to the HLP.3 continuation form. This explains how to work with the help functions.

4. Help on version

   When you select this function, you branch to the HLP.4 continuation form. The version of SESAM/SQL is displayed there.

Entering a delimiter identifier (IDE - DELIMITER IDENTIFIER)

You can call the IDE form from any other form.

If you want to enter a delimiter identifier in an input field in accordance with SESAM/SQL conventions, but the input field is not long enough, then enter an exclamation mark (!) in the input field and press the F2 key to send the form off.

The utility monitor then displays the IDE form, in which you can enter the delimiter identifier in full. When doing this, note the following:

- The delimiter identifier is enclosed in quotes and, including the quotes, can have a maximum length of 64 characters.
- Blanks count as characters.
- You cannot enter the delimiter identifier in hexadecimal notation.
- When the delimiter identifier is logged, quotes within the delimiter identifier appear twice.

See the section entitled “Basic SQL language constructs” in the “Core manual”.

Example

Entry in the IDE form:

```
''delimiter_identifier ''ORDERCUST_sample_database''''
```

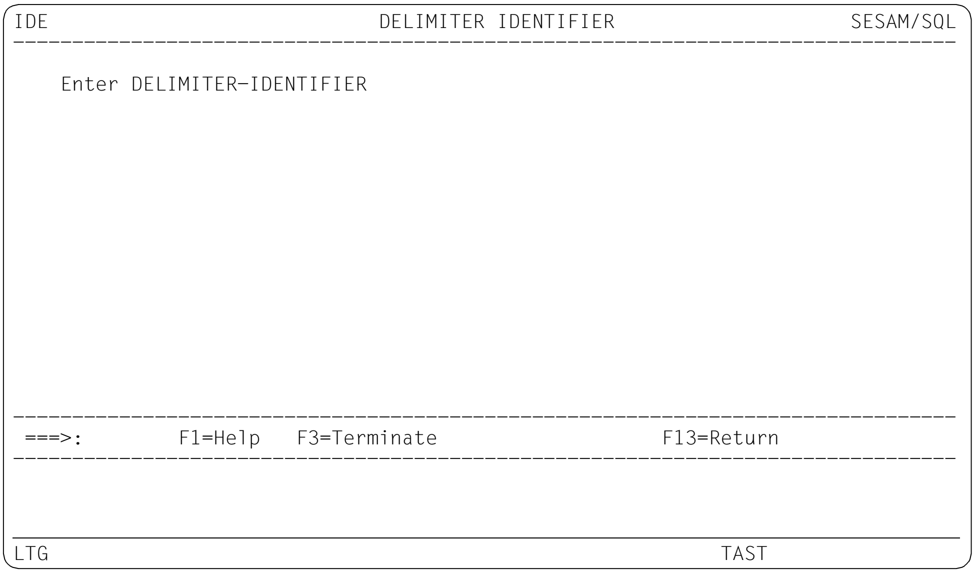
The delimiter identifier is logged as follows:

```
''delimiter_identifier ''''ORDERCUST_sample_database''''''
```

When you press the DUE key to send off the IDE form, the original form appears again. As many characters of the delimiter identifier appear in it as the length of the input field allows.

The IDE form has no continuation forms.

The IDE form



```
IDE                             DELIMITER IDENTIFIER                    SESAM/SQL
---------------------------------------------------------------------------------

    Enter DELIMITER-IDENTIFIER












---------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate                   F13=Return
---------------------------------------------------------------------------------



LTG                                                          TAST
```

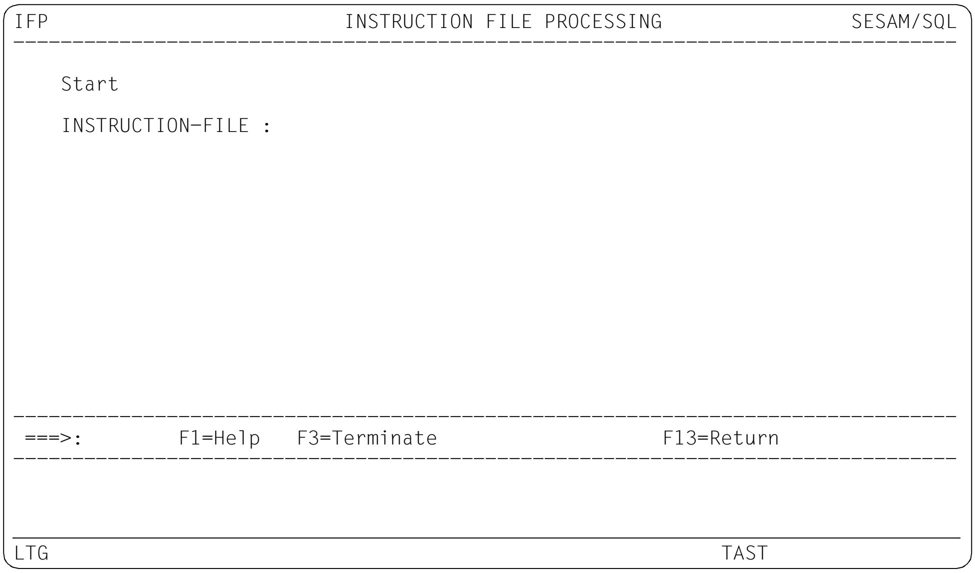
Specifying an instruction file (IFP - INSTRUCTION FILE PROCESSING)

You call the IFP form by selecting function 2, INSTRUCTION FILE PROCESSING, from the STM - START MENU start form.

In the IFP form, you specify an instruction file to be processed in interactive mode. The instruction file can be stored either as a BS2000 file or as an LMS library member. If the input field is too short for the library member name, you can branch to the LIB - LIBRARY ELEMENT form to enter the complete library member name (see page 265).

The IFP form has no continuation forms.

The IFP form



```
IFP                           INSTRUCTION FILE PROCESSING                SESAM/SQL
--------------------------------------------------------------------------------

    Start

    INSTRUCTION-FILE :













--------------------------------------------------------------------------------
 ===>:         F1=Help   F3=Terminate                      F13=Return
--------------------------------------------------------------------------------


________________________________________________________________________________
LTG                                                          TAST
```

While the instruction file is being processed, the statement currently being processed is displayed in the form's message area, see section "Processing an instruction file" on page 110.

Importing a base table (IMP - IMPORT TABLE)

You call the IMP form either via function 20, IMPORT TABLE, in the start form STM - START MENU, or by typing “imp” in the command area.
You can use the IMP form to import a table from an export file into any SESAM/SQL database where it acts as a base table. This table has the same structure as the base table that was used to create the export file with EXPORT-TABLE. The column definition and primary key (if present) are identical. You can decide whether or not to transfer user data and which table properties are to be taken over.
See the IMPORT TABLE utility statement in the “SQL Reference Manual Part 2: Utilities”.

You are offered an automatic backup before and after the activity provided that you have specified the configuration parameter SEE-COPY = ON (see also the SEE-COPY configuration parameter on page 83).

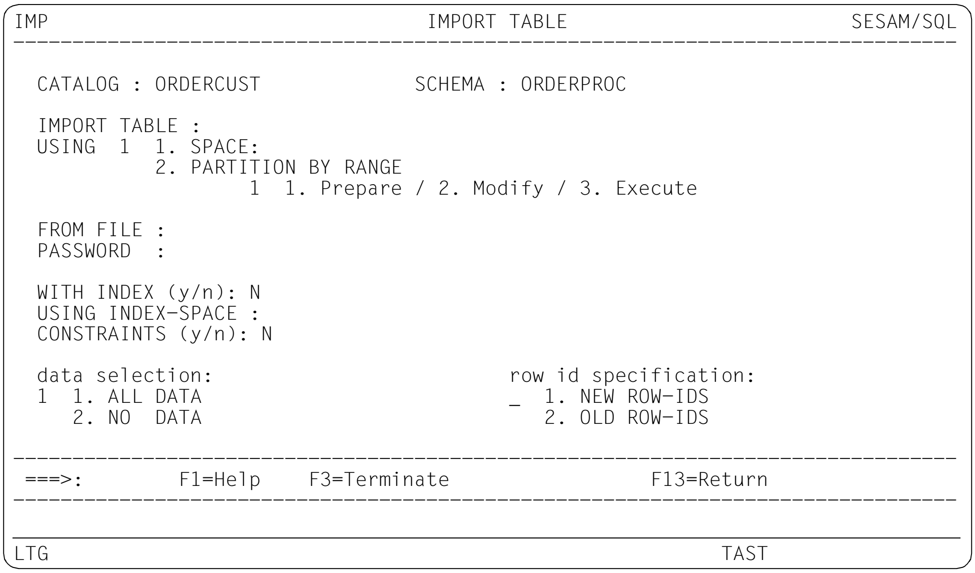
When importing, you can also use the relevant setting in USING to control whether a partitioned or a non-partitioned table is created.

You can transfer UNIQUE constraints and check constraints as well as indexes. If you decide to import indexes, you can use the USING INDEX SPACE field to specify a storage location.

When you import a table, you specify whether all or no user data is to be transferred. In the case of tables with primary keys, you can assign new record numbers for the transferred data.

The number of records that are loaded from the export file into the table is displayed in the message area. If other utility monitor messages or SQLSTATEs are present, this is indicated by a “M+-” in the command area. In this case, you can use “m+” and “m-” in the command area to scroll through the messages.

The IMP form



```
IMP                                  IMPORT TABLE                              SESAM/SQL
-------------------------------------------------------------------------------------

  CATALOG : ORDERCUST                SCHEMA : ORDERPROC

  IMPORT TABLE :
  USING  1  1. SPACE:
            2. PARTITION BY RANGE
                    1  1. Prepare / 2. Modify / 3. Execute

  FROM FILE :
  PASSWORD  :

  WITH INDEX (y/n): N
  USING INDEX-SPACE :
  CONSTRAINTS (y/n): N

  data selection:                            row id specification:
  1  1. ALL DATA                             _  1. NEW ROW-IDS
     2. NO  DATA                                2. OLD ROW-IDS

-------------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------------

LTG                                                          TAST
```

The procedure differs for non-partitioned and partitioned tables:

- If you wish to create a non-partitioned table, select USING SPACE (“1”, default) and, as required, enter the space name under SPACE. To create the table and execute the import, transfer the form with the DUE key.
- If you wish to create a partitioned table, select USING PARTITION BY RANGE (“2”). You then use the functions under PARTITION BY RANGE to control the further procedure, see the next page. In order to create the partitioned table and execute the import, enter function 3 “Execute” under PARTITION BY RANGE and transfer the form with DUE.

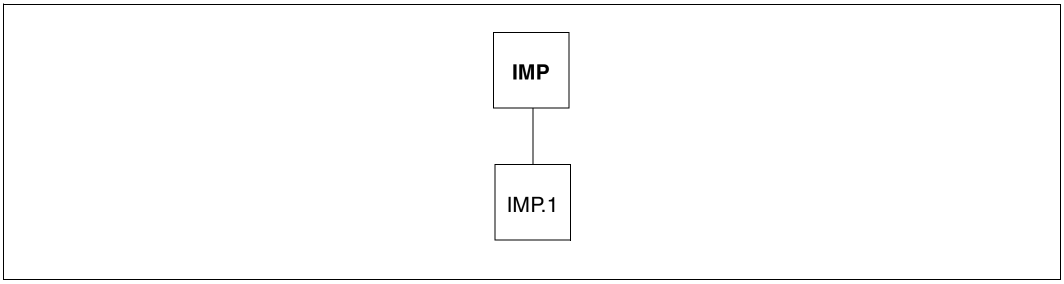


Figure 14: IMP form with continuation form

Explanation of the functions under PARTITION BY RANGE

1. Prepare

   With function 1 “Prepare” you branch to the IMP.1 continuation form. There you define the properties of the partitions.

   The entries correspond to the specifications in the SQL statement CREATE TABLE, see the “SQL Reference Manual Part 1: SQL Statements”:

   - Serial number of the partition (1 through 16)
   - Comparison operator < or <= and the upper limit of the primary key interval (without parentheses). If the primary key consists of multiple columns, the individual values must be separated by commas.
   - Space name of the partition

   After you transfer the form with the DUE key you are shown the IMP.1 form again in order to define the next partition’s properties.

   For the last partition you must check mark LAST PARTITION; no upper limit for the primary key interval may be specified here.
   After the last partition has been defined you return to the IMP form by pressing the F13 key or entering F13 in the command area.

2. Modify

   Function 2 “Modify” enables you, after you have returned to the IMP form, to branch to the IMP.1 form again in order to modify the properties of the partitions.

3. Execute

   The partitioned table is created and the import executed using function 3 “Execute” (default after branch back from IMP.1), specifying the other options, and transferring the form with the DUE key.

   i For partitioned tables “NEW ROW-IDS” must be selected under “row id specification” in the IMP form.

Querying metadata from INFORMATION_SCHEMA (INF - INFORMATION-SCHEMA)

You call the INF form either by selecting function 9, INFORMATION-SCHEMA, from the STM - START MENU start form or by entering inf in the command area.

The INF form and its continuation forms allow you to obtain information from the INFORMATION_SCHEMA on the database objects for which you have access authorization. See section “Overview of the information schemata” on page 182 and of the “SQL Reference Manual Part 1: SQL Statements”.

You can output the information to the screen, a file (BS2000 file or a member of an LMS library) or both.

The INF - INFORMATION-SCHEMA form has a hierarchical structure: you select a database object, such as a schema, about which you can then request more information. Information on privileges for the associated database object is also issued. The selected database objects appear by default in the continuation forms.

When you select database objects, you move around the following INFORMATION_SCHEMA hierarchy:

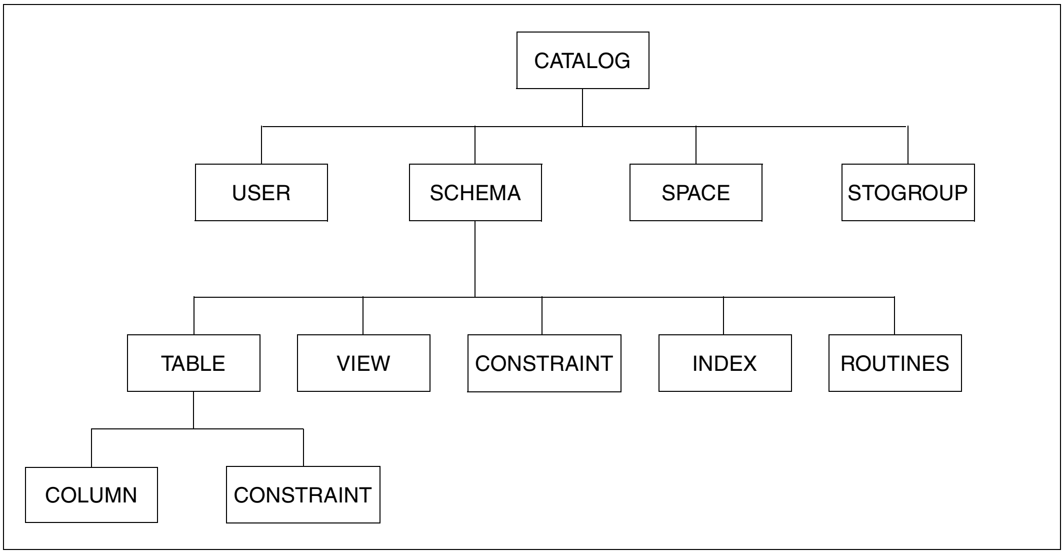


Figure 15: The INFORMATION_SCHEMA hierarchy

Specifying the extent of the output – selection forms

Before a table is output in a continuation form, a selection form appears that differs from the continuation form for table output only in the following ways:

- The selection form contains input fields instead of output values.
- The suffix -F is appended to the form's short name.
- The suffix -FILTER is appended to the form name.

The selection form allows you to limit the extent of the output records of an INFORMATION_SCHEMA view by entering values in the input fields for specific columns.

You can specify partially qualified alphanumeric values by means of the placeholders “%” and “_”:

- The “%” character stands for any n characters where n ≥ 0.
- The “_” character stands for any character.

If the input value in command mode contains “_”, more values than expected may be output. (Reason: When the characters “%”, “_”, and/or “\” are included in the input value, a “%” character is appended to the value by SESAM and compared with LIKE.)

If the placeholders “_” or “%” are to be interpreted as normal characters, they must be preceded by a backslash “\”. The value specification must confirm to the conventions for delimiter identifiers (see section “Entering a delimiter identifier (IDE - DELIMITER IDENTIFIER)” on page 241).

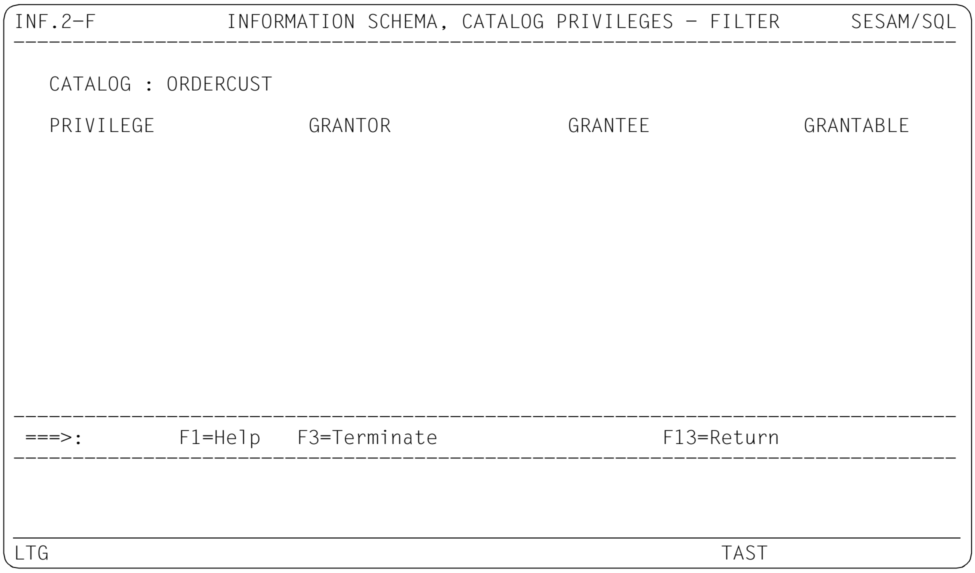
After you have pressed the DUE key to send the entry off, only records containing all the specified values are output.

The input fields now contain values, and no further entries can be made in them.

Example

INF.2-F and INF.2 continuation forms

If you select function 2, Information on CATALOG PRIVILEGES, in the INF form (see page 251), you branch to the INF.2-F continuation form.


```
INF.2-F          INFORMATION SCHEMA, CATALOG PRIVILEGES - FILTER      SESAM/SQL
--------------------------------------------------------------------------------

   CATALOG : ORDERCUST

   PRIVILEGE              GRANTOR                GRANTEE             GRANTABLE











--------------------------------------------------------------------------------
 ===>:         F1=Help   F3=Terminate                    F13=Return
--------------------------------------------------------------------------------



LTG                                                         TAST
```

An input field appears under each of the column headings PRIVILEGE, GRANTOR, GRANTEE and GRANTABLE. In these input fields, you can preselect the records to be output. The selection constraints are linked by AND. After you have specified a preselection, you press the DUE key to send the form off.
The INF.2 continuation form is output with the selected records.

If no corresponding record is found, the INF.2-F continuation form appears with an appropriate message.

Here are a number of preselection examples:

Example 1

You make no entries in the input fields, or you enter a percent sign (%) in one or more input fields. You enter nothing in the other input fields.

Output:
All records are output.

Example 2

You enter the name USERADM in the GRANTOR input field. You enter nothing in the other input fields.

Output:
All records with the value USERADM in the GRANTOR column are output.

Example 3

You enter the name USERADM% in the GRANTOR input field, the value YES in the GRANTABLE input field. You enter nothing in the other input fields.

Output:
All records that have a value beginning with USERADM in the GRANTOR column (e.g. USERADM, USERADM-A or USERADM-B) and that also have the value YES in the GRANTABLE column are output.

Example 4

You enter the name USERADM_ in the GRANTOR input field. You enter nothing in the other input fields.

Output:
All records containing a value in the GRANTOR column that begins with USERADM and is followed by another character (e.g. USERADM1 or USERADM2) are output.

Viewing and terminating output

In the continuation forms for table output, only a specific number of records are output, depending on the size of the table.

Default values that are longer than 256 characters in printable format cannot be output. If this happens, the text TRUNCATED appears.

If you want to output more records or page back, proceed as described on page 149.

In the output file, all records that correspond to the search statement are output.

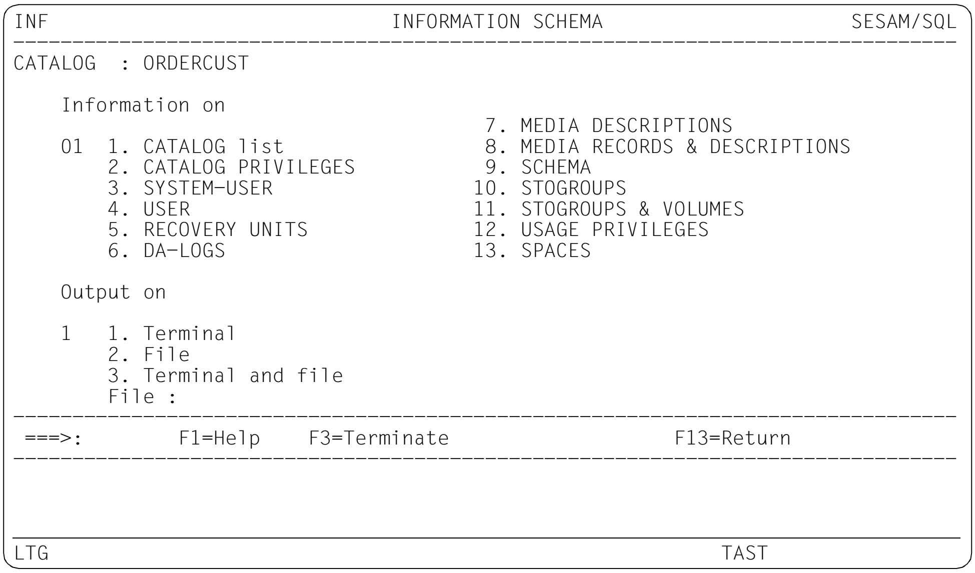
You can cancel the file output by pressing the K2 key and entering the BS2000 command INFORM-PROGRAM MSG=C'SEE,BREAK' . After the command is issued, the utility monitor returns to the form displayed last and issues an appropriate message.

You terminate file output and output to the screen by pressing the F13 key or entering F13 in the command area.

If you branch from the output forms to a form function by entering the form short name, the contents of the output forms are lost. The data is not stored because it may no longer correspond to the current status. When you return from the called form function, the predecessor form of the output form is displayed.

In batch mode, the CMD INF *short_form_name* statement can be used to route the information output to the file specified with the SEE-INFOUT configuration parameter. See also page 80 and page 91.

The INF form



```
INF                             INFORMATION SCHEMA                       SESAM/SQL
--------------------------------------------------------------------------------
CATALOG  : ORDERCUST

    Information on
                                       7. MEDIA DESCRIPTIONS
    01  1. CATALOG list                8. MEDIA RECORDS & DESCRIPTIONS
        2. CATALOG PRIVILEGES          9. SCHEMA
        3. SYSTEM-USER                10. STOGROUPS
        4. USER                       11. STOGROUPS & VOLUMES
        5. RECOVERY UNITS             12. USAGE PRIVILEGES
        6. DA-LOGS                    13. SPACES

    Output on

    1   1. Terminal
        2. File
        3. Terminal and file
        File :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                   F13=Return
--------------------------------------------------------------------------------



LTG                                                             TAST
```

When you select functions 1 - 13, you branch to the corresponding continuation forms.

Under “Output on”, you specify whether the information is to be output to the screen, a file (BS2000 file or member of an LMS library) or both. If you select function 2 or 3, you must specify a file name or library member name in the input field provided. If the input field is too short for the library member name, you can branch to the LIB - LIBRARY ELEMENT form to enter the complete library member name (see page 265).

Function 1 is the default.

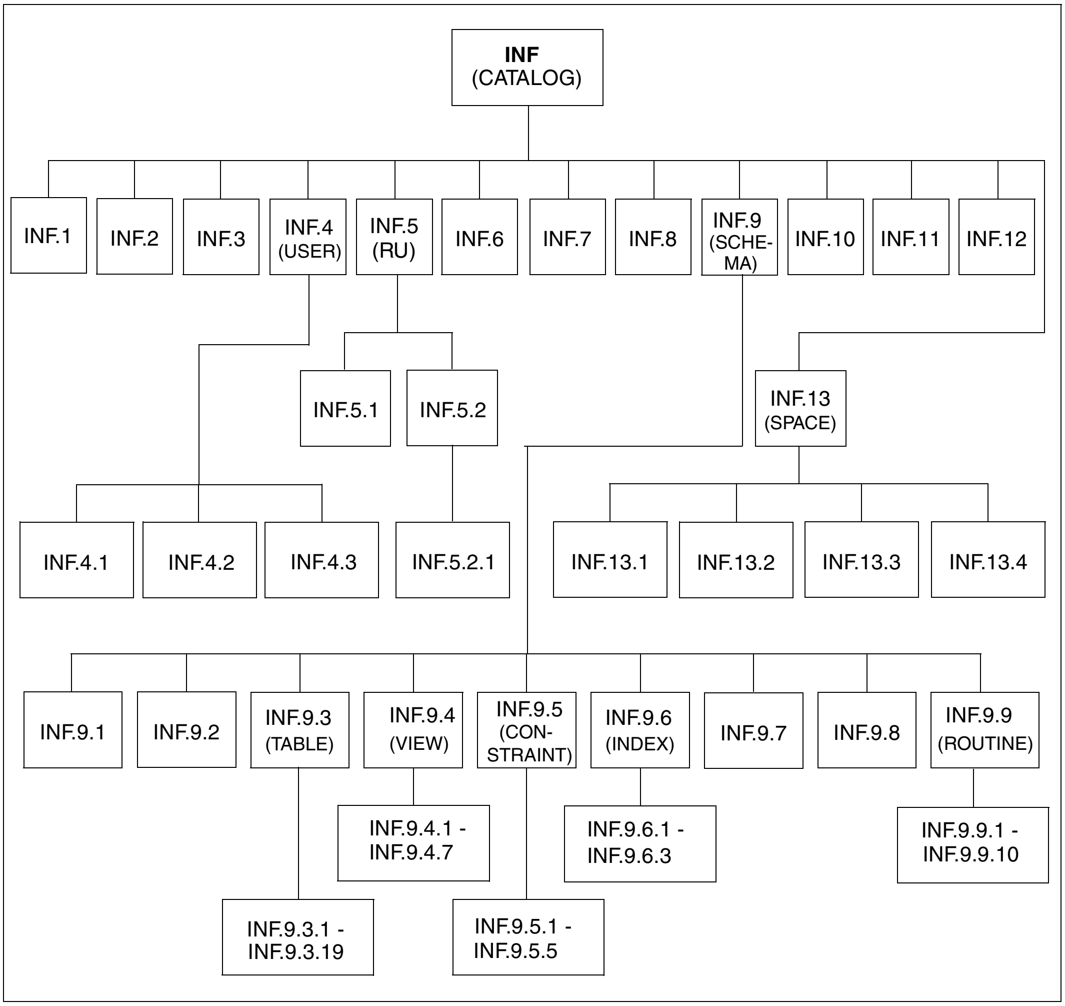


Figure 16: The INF form and its continuation forms

Explanation of the functions

1. CATALOG list

   When you select this function, you branch to the INF.1 continuation form, in which all the databases connected to the DBH are output.
   To select a database for further processing, you mark it with an x or another character.
   The DBH must be loaded with the ADM option, and the administration password must be specified in the configuration data (by means of the SEE-ADMIN configuration parameter).

2. CATALOG-PRIVILEGES

   When you select this function, you branch to the INF.2 continuation form, in which the privileges granted for the selected database are output. The CATALOG_PRIVILEGES view is accessed.

3. SYSTEM-USER

   When you select this function, you branch to the INF.3 continuation form, in which the system entries of the selected database are output. The SYSTEM_ENTRIES view is accessed.

4. USER

   When you select this function, you branch to the INF.4 continuation form, in which you specify the desired authorization key in the USER field. When you select a function, you branch to the corresponding continuation form:

   INF.4 continuation form, function 1: USER list
   When you select this function, you branch to the INF.4.1 continuation form, in which the authorization keys of the selected database are output.
   The USERS view is accessed.
   You select an authorization key for further processing by marking it with an x or another character.

   INF.4 continuation form, function 2: TABLE-PRIVILEGES
   When you select this function, you branch to the INF.4.2 continuation form, in which all the table privileges are output for which the selected authorization key is the GRANTOR or GRANTEE.
   The TABLE_PRIVILEGES view is accessed.

   INF.4 continuation form, function 3: COLUMN-PRIVILEGES
   When you select this function, you branch to the INF.4.3 continuation form, in which, for individual columns, all the table privileges are output for which the selected authorization key is the GRANTOR or GRANTEE.
   The COLUMN_PRIVILEGES view is accessed.
   If the GRANTOR and GRANTEE fields have default values, the USER field is not evaluated.

5. RECOVERY-UNITS

   When you select this function you branch to the INF.5 continuation form, in which you can choose whether you want information on the recovery unit of a selected database or on which files are required for a RECOVER. When you select a function, you branch to the corresponding continuation form:

   INF.5 continuation form, function 1 “RECOVERY-UNITS”
   When you select this function you branch to the INF.5.1 continuation form, in which the recovery units for the selected database are output. The RECOVERY_UNITS view is accessed.

   INF.5 continuation form, function 2 “Files for RECOVERY”
   When you select this function you branch to the INF.5.2 continuation form, which displays an overview of which files are required for a RECOVER or REFRESH REPLICATION.
   Depending on the function selected for RECOVERY, temporary job variables are created which contain the names of the files required for a RECOVER or REFRESH REPLICATION of the user spaces or catalog space of a database.
   If you select functions 1, 2 and 4, a job variable with the name #SESAM.RU.DA-LOG is created in addition to one or more job variables for user spaces (#SESAM.RU.*space*). This job variable contains the name of the log files of the user spaces.
   If you select function 3, the job variables #SESAM.RU.CATALOG for the catalog space and #SESAM.RU.CAT-LOG for the log files of the catalog space are created.
   For more information on job variables, see page 117 and the “Database Operation” manual.
   If data records are found and if not only “File” was selected in the INF main form in the information output field, you switch to the INF.5.2.1 continuation form to output these data records.

   i Information cannot be output for RECOVER CATALOG. However, it is possible to determine the files for a RECOVER CATALOG by carrying out the following steps:

   1. Information query for RECOVER CATALOG-SPACE
   2. RECOVER CATALOG-SPACE
   3. Information query for RECOVER ALL SPACES
   4. RECOVER CATALOG

   If the time stamp of all backups of the spaces is identical, i.e. these backups originate from one COPY CATALOG, a RECOVER SPACESET can be issued as step 4.
   See also the section “Recovery in batch mode” on page 126.

6. DA-LOGS

   When you select this function, you branch to the INF.6 continuation form, in which the names of the DA-LOG files of the selected database are output.
   The DA_LOGS view is accessed.

7. MEDIA DESCRIPTIONS

   When you select this function, you branch to the INF.7 continuation form, in which the names of the database-specific files and their properties are output from the media tables of the selected database.
   The MEDIA_DESCRIPTIONS view is accessed.

8. MEDIA RECORDS & DESCRIPTIONS

   When you select this function, you branch to the INF.8 continuation form, in which the names of the database-specific files and their media are output from the media table of the selected database.
   The MEDIA_RECORDS view is accessed.

9. SCHEMA

   When you select this function, you branch to the INF.9 continuation form, in which you specify the desired schema in the SCHEMA field. When you select a function, you branch to the corresponding continuation form:

   INF.9 continuation form, function 1: SCHEMA list
   When you select this function, you branch to the INF.9.1 continuation form, in which the schemata of the selected database are output.
   The SCHEMATA view is accessed.
   You select a schema for further processing by marking it with an x or another character.

   INF.9 continuation form, function 2: TABLES
   When you select this function, you branch to the INF.9.2 continuation form, in which the tables of the selected schema are output.
   The TABLES and BASE_TABLES views are accessed.

   INF.9 continuation form, function 3: BASETABLES
   When you select this function, you branch to the INF.9.3 continuation form, in which you can specify a base table for further processing in the TABLE input field. You can then specify a column for it in the COLUMN input field and a table constraint for it in the CONSTRAINT input field.
   If you subsequently select a different base table, the preselected values for the columns and table constraints of the previously specified base table disappear automatically.
   When you select a function, you branch to the corresponding continuation form:

INF.9.3 continuation form, function 1: BASETABLE list

When you select this function from under the Information on BASETABLE heading, you branch to the INF.9.3.1 continuation form, in which the base tables of the selected schema are output. The "_PARTITIONS_" value is output in the SPACE field in the case of partitioned tables.
The BASE_TABLES view is accessed.
You select a base table for further processing by marking it with an x or another character.

INF.9.3 continuation form, function 2: TABLE-PRIVILEGES

When you select this function from under the Information on BASETABLE heading, you branch to the INF.9.3.2 continuation form, in which the privileges for the selected base table are output.
The TABLE_PRIVILEGES view is accessed.

INF.9.3 continuation form, function 3: KEY-COLUMNS

When you select this function from under the Information on BASETABLE heading, you branch to the INF.9.3.3 continuation form, in which the columns of the selected base table are output for which a UNIQUE or referential constraint has been defined.
The KEY_COLUMN_USAGE view is accessed.

INF.9.3 continuation form, function 4: INDEXES for BASETABLE

When you select this function from under the Information on BASETABLE heading, you branch to the INF.9.3.4 continuation form, in which the indexes of the selected base table are output.
The INDEXES view is accessed.

INF.9.3 continuation form, function 5: VIEWS referencing BASETABLE

When you select this function from under the Information on BASETABLE heading, you branch to the INF.9.3.5 continuation form, in which the views that reference the selected base table are output.
The VIEW_TABLE_USAGE view is accessed.

INF.9.3 continuation form, function 6: CONSTRAINTS referencing BASETABLE

When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.6 continuation form, in which the integrity constraints that reference the selected base table are output.
The CONSTRAINT_TABLE_USAGE view is accessed.

INF.9.3 continuation form, function 7: ROUTINES referencing BASETABLE

When you select this function from under the Information on BASETABLE heading, you branch to the INF.9.3.7 continuation form, in which the routines that reference the selected base table are output.
The ROUTINE_TABLE_USAGE view is accessed.

INF.9.3 continuation form, function 8: COLUMN list
When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.8 continuation form, in which the columns of the selected base table are output.
The BASE_TABLE_COLUMNS view is accessed.
You select a column for further processing by marking it with an x or another character.

INF.9.3 continuation form, function 9: COLUMN data
When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.9 continuation form, in which data on the selected column is output.
The BASE_TABLE_COLUMNS view is accessed.

INF.9.3 continuation form, function 10: COLUMN data in detail
When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.10 continuation form, in which detailed data on the selected column is output.
The BASE_TABLE_COLUMNS view is accessed.

INF.9.3 continuation form, function 11: COLUMN-PRIVILEGES
When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.11 continuation form, in which the privileges for the selected column are output.
The COLUMN_PRIVILEGES view is accessed.

INF.9.3 continuation form, function 12: INDEXES of COLUMN
When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.12 continuation form, in which the indexes of the selected column are output.
The INDEX_COLUMN_USAGE view is accessed.

INF.9.3 continuation form, function 13: VIEWS referencing COLUMN
When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.13 continuation form, in which the views that reference the selected column are output.
The VIEW_COLUMN_USAGE view is accessed.

INF.9.3 continuation form, function 14: CONSTRAINTS referencing COLUMN
When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.14 continuation form, in which the integrity constraints that reference the selected column are output.
The CONSTRAINT_COLUMN_USAGE view is accessed.

INF.9.3 continuation form, function 15: ROUTINES referencing COLUMN

When you select this function from under the Information on BASETABLE-COLUMN heading, you branch to the INF.9.3.15 continuation form, in which the routines that reference the selected column are output.
The ROUTINE_COLUMN_USAGE view is accessed.

INF.9.3 continuation form, function 16: TABLE-CONSTRAINT list

When you select this function from under the Information on TABLE-CONSTRAINT heading, you branch to the INF.9.3.16 continuation form, in which the table constraints of the selected base table are output.
The TABLE_CONSTRAINTS view is accessed.
You select a table constraint for further processing by marking it with an x or another character.

INF.9.3 continuation form, function 17: TABLES subject to CONSTRAINT

When you select this function from under the Information on TABLE-CONSTRAINT heading, you branch to the INF.9.3.17 continuation form, in which the tables that are subject to the selected integrity constraint are output.
The CONSTRAINT_TABLE_USAGE view is accessed.

INF.9.3 continuation form, function 18: COLUMNS subject to CONSTRAINT

When you select this function from under the Information on TABLE-CONSTRAINT heading, you branch to the INF.9.3.18 continuation form, in which the columns that are subject to the selected integrity constraint are output.
The CONSTRAINT_COLUMN_USAGE view is accessed.

INF.9.3 continuation form, function 19 “PARTITIONS”

When you select this function from under the Information on PARTITIONS heading, you branch to the INF.9.3.19 continuation form, in which information on the partitions of a partitioned table is output.
The PARTITIONS view is accessed.

INF.9 continuation form, function 4: VIEWS

When you select this function, you branch to the INF.9.4 continuation form, in which you specify a view in the VIEW input field. When you select a function, you branch to the corresponding continuation form:

INF.9.4 continuation form, function 1: VIEW list

When you select this function, you branch to the INF.9.4.1 continuation form, in which the views of the selected schema are output.
The VIEWS view is accessed.
You select a view for further processing by marking it with an x or another character.

INF.9.4 continuation form, function 2: VIEW data
When you select this function, you branch to the INF.9.4.2 continuation form, in which the data of the selected view is output.
The VIEWS view is accessed.

INF.9.4 continuation form, function 3: TABLES referenced by VIEW
When you select this function, you branch to the INF.9.4.3 continuation form, in which the tables referenced by the selected view are output.
The VIEW_TABLE_USAGE view is accessed.

INF.9.4 continuation form, function 4: COLUMNS referenced by VIEW
When you select this function, you branch to the INF.9.4.4 continuation form, in which the columns referenced by the selected view are output.
The VIEW_COLUMN_USAGE view is accessed.

INF.9.4 continuation form, function 5: PRIVILEGES for VIEW
When you select this function, you branch to the INF.9.4.5 continuation form, in which the privileges for the selected view are output.
The TABLE_PRIVILEGES view is accessed.

INF.9.4 continuation form, function 6: COLUMNS of VIEW
When you select this function, you branch to the INF.9.4.6 continuation form, in which you specify a column in the COLUMN input field. When you select a function, you branch to the corresponding continuation form:

INF.9.4.6 continuation form, function 1: COLUMN list
When you select this function, you branch to the INF.9.4.6.1 continuation form, in which the columns of the selected view are output. You can select a column for further processing.
The COLUMNS view is accessed.

INF.9.4.6 continuation form, function 2: COLUMN data
When you select this function, you branch to the INF.9.4.6.2 continuation form, in which the data of the selected column is output.
The COLUMNS view is accessed.

INF.9.4 continuation form, function 7: ROUTINES referenced by VIEW
When you select this function, you branch to the INF.9.4.7 continuation form, in which all routines that are referenced by the selected view are output.
The VIEW_ROUTINE_USAGE view is accessed.

INF.9 continuation form, function 5: CONSTRAINTS

When you select this function, you branch to the INF.9.5 continuation form, in which you specify the desired integrity constraint in the CONSTRAINT input field. When you select a function, you branch to the corresponding continuation form:

INF.9.5 continuation form, function 1: REFERENTIAL-CONSTRAINT list

When you select this function, you branch to the INF.9.5.1 continuation form, in which the referential constraints of the selected schema are output. It also includes the integrity-constraint name assigned by the system (in the CONSTRAINT-NAME output field).
The REFERENTIAL_CONSTRAINTS view is accessed.
You select a referential constraint for further processing by marking it with an x or another character.

INF.9.5 continuation form, function 2: CHECK-CONSTRAINT list

When you select this function, you branch to the INF.9.5.2 continuation form, in which the check constraints of the selected schema are output.
The CHECK_CONSTRAINTS view is accessed.
You select a check constraint for further processing by marking it with an x or another character.

INF.9.5 continuation form, function 3: CHECK-CONSTRAINT data

When you select this function, you branch to the INF.9.5.3 continuation form, in which the data of the selected check constraint is output.
The CHECK_CONSTRAINTS view is accessed.

INF.9.5 continuation form, function 4: TABLES subject to CONSTRAINT

When you select this function, you branch to the INF.9.5.4 continuation form, in which the tables that are subject to the selected integrity constraint are output.
The CONSTRAINT_TABLE_USAGE view is accessed.

INF.9.5 continuation form, function 5: COLUMNS subject to CONSTRAINT

When you select this function, you branch to the INF.9.5.5 continuation form, in which the columns that are subject to the selected integrity constraint are output.
The CONSTRAINT_COLUMN_USAGE view is accessed.

INF.9 continuation form, function 6: INDEXES

When you select this function, you branch to the INF.9.6 continuation form, in which you specify an index in the INDEX input field. When you select a function, you branch to the corresponding continuation form:

INF.9.6 continuation form, function 1: INDEX list

When you select this function, you branch to the INF.9.6.1 continuation form, in which the indexes of the selected schema are listed. The INDEXES view is accessed.
You select an index for further processing by marking it with an x or another character.

INF.9.6 continuation form, function 2: INDEX data

When you select this function, you branch to the INF.9.6.2 continuation form, in which the data of the selected index is output.
The INDEXES view is accessed.

INF.9.6 continuation form, function 3: INDEX-COLUMNS

When you select this function, you branch to the INF.9.6.3 continuation form, in which the indexed columns of the selected schema are output.
The INDEX_COLUMN_USAGE view is accessed.

INF.9 continuation form, function 7: TABLE-PRIVILEGES

When you select this function, you branch to the INF.9.7 continuation form, in which the table privileges of the selected schema are output, without individual columns being specified.
The TABLE_PRIVILEGES view is accessed.

INF.9 continuation form, function 8: COLUMN-PRIVILEGES

When you select this function, you branch to the INF.9.8 continuation form, in which the UPDATE and REFERENCES table privileges for individual columns of the selected schema are output.
The COLUMN_PRIVILEGES view is accessed.

INF.9 continuation form, function 9: ROUTINES

When you select this function, you branch to the INF.9.9 continuation form, in which you specify the desired routine in the ROUTINE input field and possibly the desired parameter in the PARAMETER input field. When you select a function, you branch to the corresponding continuation form:

INF.9.9 continuation form, function 1: ROUTINE list

When you select this function, you branch to the INF.9.9.1 continuation form, in which the routines of the selected schema are output. The ROUTINES view is accessed.
You select a routine for further processing by marking it with an x or another character.

INF.9.9 continuation form, function 2: ROUTINE data

When you select this function, you branch to the INF.9.9.2 continuation form, in which the data of the selected routine is output.
The ROUTINES view is accessed.

INF.9.9 continuation form, function 3 PRIVILEGES

When you select this function, you branch to the INF.9.9.3 continuation form, in which the privileges of the selected routine are output.
The ROUTINE_PRIVILEGES view is accessed.

INF.9.9 continuation form, function 4: PARAMETER list

When you select this function, you branch to the INF.9.9.4 continuation form, in which the parameters of the selected routine are output. The PARAMETERS view is accessed.
You select a parameter for further processing by marking it with an x or another character.

INF.9.9 continuation form, function 5: PARAMETER data

When you select this function, you branch to the INF.9.9.5 continuation form, in which the data of the selected parameter is output.
The PARAMETERS view is accessed.

INF.9.9 continuation form, function 6: TABLES

When you select this function, you branch to the INF.9.9.6 continuation form, in which the tables used by the selected routine are output.
The ROUTINE_TABLE_USAGE view is accessed.

INF.9.9 continuation form, function 7: COLUMNS

When you select this function, you branch to the INF.9.9.7 continuation form, in which the columns used by the selected routine are output.
The ROUTINE_COLUMN_USAGE view is accessed.

INF.9.9 continuation form, function 8: CALLED-ROUTINES
When you select this function, you branch to the INF.9.9.8 continuation form, in which the routines **called** by the selected routine are output.
The ROUTINE_ROUTINE_USAGE view is accessed.

INF.9.9 continuation form, function 9: CALLING-ROUTINES
When you select this function, you branch to the INF.9.9.9 continuation form, in which the **calling** routines of the selected routine are output.
The ROUTINE_ROUTINE_USAGE view is accessed.

INF.9.9 continuation form, function 10: VIEWS
When you select this function, you branch to the INF.9.9.10 continuation form, in which the views that use the selected routine are output.
The VIEW_ROUTINE_USAGE view is accessed.

10. STOGROUPS

    When you select this function, you branch to the INF.10 continuation form, in which the storage groups of the selected database are output.
    The STOGROUPS view is accessed.

11. STOGROUPS & VOLUMES

    When you select this function, you branch to the INF.11 continuation form, in which the storage groups and the associated volumes of the selected database are output.
    The STOGROUPS and STOGROUP_VOLUME_USAGE views are accessed.

12. USAGE PRIVILEGES

    When you select this function, you branch to the INF.12 continuation form, in which the USAGE special privileges granted for the selected database are output. The USAGE special privilege permits the usage of a storage group.
    The USAGE_PRIVILEGES view is accessed.

13. SPACES

When you select this function, you branch to the INF.13 continuation form, in which you specify a space in the SPACE input field.
When you select a function, you branch to the corresponding continuation form:

INF.13 continuation form, function 1: SPACES list
When you select this function, you branch to the INF.13.1 continuation form, in which the spaces of the selected database are output.
The SPACES view is accessed.
You select a space for further processing by marking it with an x or another character.

INF.13 continuation form, function 2: INDEXES on SPACE
When you select this function, you branch to the INF.13.2 continuation form, in which the indexes stored in the selected space are output.
The INDEXES view is accessed.

INF.13 continuation form, function 3: TABLES on SPACE
When you select this function, you branch to the INF.13.3 continuation form, in which the tables stored in the selected space are output.
The BASE_TABLES view is accessed.

As SPACE is an input field for selecting output records, the "_PARTITIONS_" value for outputting partitioned tables can also be specified here. If a space name is specified for SPACE, partitioned tables are not output even if there is a partition on this space.

INF.13 continuation form, function 4: RECOVERY UNITS for SPACE
When you select this function, you branch to the INF.13.4 continuation form, in which the recovery unit records of the selected space are output.
The RECOVERY_UNITS view is accessed.

Entering a library member name (LIB - LIBRARY ELEMENT)

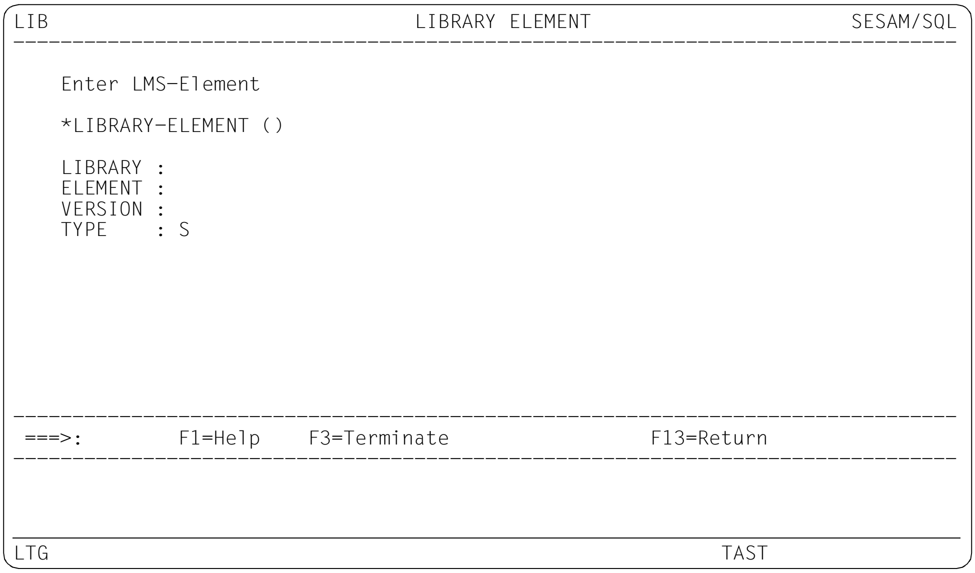
You can call the LIB form from the CNF, COP.4, COP.5, IFP, INF, SNF or SQL form.

If you want to enter a library member name compliant with LMS conventions (see the “LMS (BS2000)” manual) in the input field for file names in one of the forms listed above, and the input field is not long enough (54 characters), then enter an exclamation mark (!) in the input field and press the F2 key or enter “F2” in the command area to send the form off. The utility monitor then displays the LIB form, in which you can enter the library member name in full.

When you press the DUE key to send off the LIB form, the original form appears again. As many characters of the library member name appear in it as the length of the input field allows.

The LIB form has no continuation forms.

The LIB form



```
LIB                                 LIBRARY ELEMENT                               SESAM/SQL
-------------------------------------------------------------------------------

    Enter LMS-Element

    *LIBRARY-ELEMENT ()

    LIBRARY :
    ELEMENT :
    VERSION :
    TYPE    : S










-------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                     F13=Return
-------------------------------------------------------------------------------



LTG                                                           TAST
```

The TYPE specifications “L”, “C” and “R” are not permitted and are rejected by the utility monitor with an error message.

Loading a base table with data from a file (LOD - LOAD)

You call the LOD form either by selecting function 5, LOAD, from the STM - START MENU start form or by entering lod in the command area.

In the LOD form and its continuation forms, you load data from a file into a base table. You can load the data from predefined formats (the UNLOAD format, TRANSFER format, DELIMITER format or CSV format) or define the input format yourself.

You may select from the OFFLINE and ONLINE modes. The OFFLINE mode behaves like the LOAD statement previously used in SESAM.

The ONLINE mode offers the following advantages:

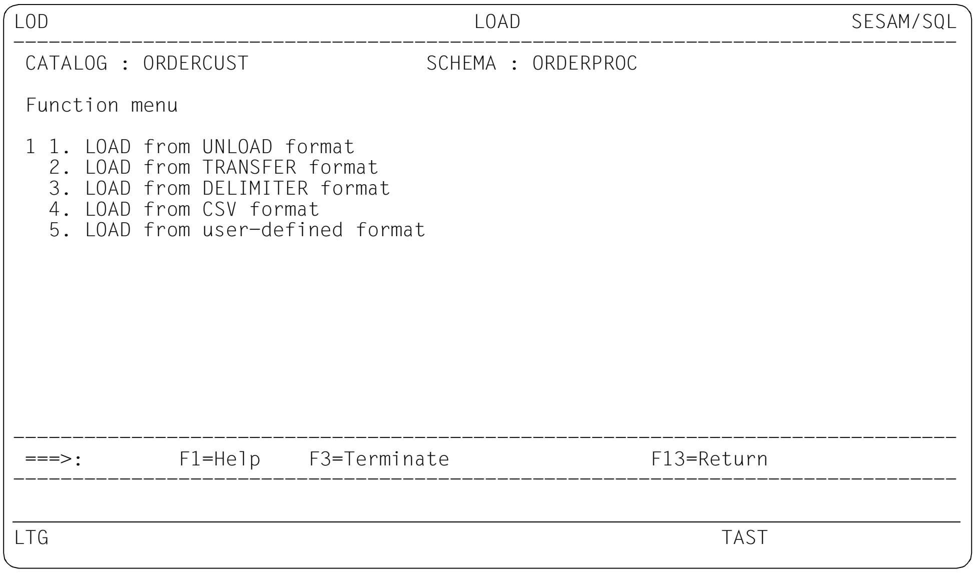
- better performance when loading small files
- logging is not interrupted during the loading process
- the user space is not exclusively locked against changing during the loading process
- the user space is not set to the “load running” state after an aborted LOAD.

You are offered an automatic backup before and after the activity provided that you have specified the configuration parameter SEE-COPY = ON (see also the SEE-COPY configuration parameter on page 83).

The number of records that are loaded into the table is displayed in the message area. If other utility monitor messages or SQLSTATEs are present, this is indicated by a “M+-” in the command area. In this case, you can use “m+” and “m-” in the command area to scroll through the messages.

See the utility statement LOAD in the “SQL Reference Manual Part 2: Utilities” and the sections entitled “Processing input/output and error files” and “Data representation in the input and output files for LOAD and UNLOAD” in the chapter “Basic information on working with utility statements” of that manual.

The LOD form



```
LOD                                     LOAD                                SESAM/SQL
-------------------------------------------------------------------------------------
 CATALOG : ORDERCUST                  SCHEMA : ORDERPROC

 Function menu

 1 1. LOAD from UNLOAD format
   2. LOAD from TRANSFER format
   3. LOAD from DELIMITER format
   4. LOAD from CSV format
   5. LOAD from user-defined format











-------------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                F13=Return
-------------------------------------------------------------------------------------

LTG                                                            TAST
```

When you select functions 1 - 5, you branch to the corresponding continuation forms.

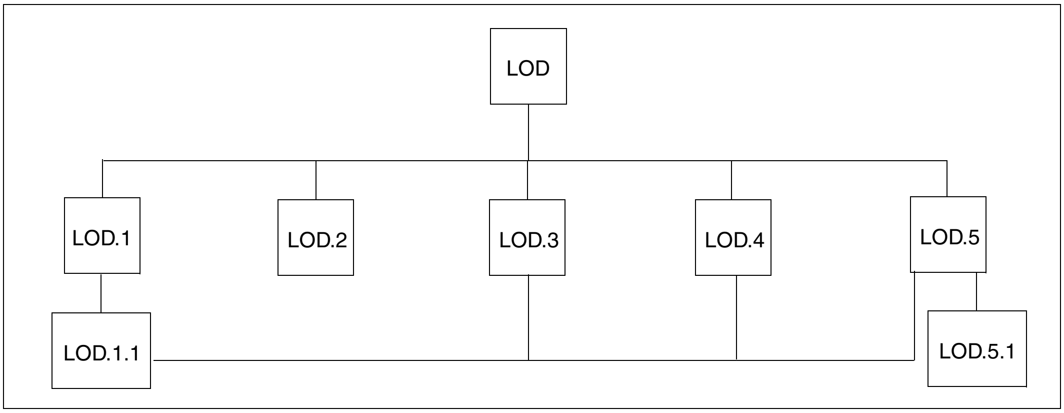


Figure 17: The LOD form with its continuation forms

Explanation of the functions

1. LOAD from UNLOAD-FORMAT

   When you select this function, you branch to the continuation form LOD.1. There you specify the input file that has the same format as an output file created by UNLOAD. See the utility statement LOAD ... UNLOAD_FORMAT in the “SQL Reference Manual Part 2: Utilities”.

   If column values are only to be loaded to specific columns in a table, you must specify COLUMN LIST=Y and function 1 (Prepare) in the LOD.1 form. You then branch to the continuation form LOD.1.1 to prepare the column list.
   After you return to the LOD.1 form, the entered list can be changed again using COLUMN LIST=Y and function 2 (Modify). The LOAD statement is executed with COLUMN LIST=Y and function 3 (Execute).

   Continuation form LOD.1.1
      In this form you specify the column of the table to which the values are to be loaded (COLUMN list input fields). If the columns are multiple columns, specify the characteristic in the “COMPONENT” field.

      After defining the first 8 columns you send off the form by pressing the DUE key. This means that when you define more than 8 columns you must send off the form LOD.1.1 several times. By entering “<<“ and ”>>” in the command area and pressing the DUE key you can page back and forward through forms in which you have already defined columns.
      After you have defined all the columns, you return to the LOD.1 form by pressing the F13 key .

2. LOAD from TRANSFER-FORMAT

   When you select this function, you branch to the continuation form LOD.2. There you specify the input file that has the same format as a transfer file created by UNLOAD. All the columns from the input file are loaded to the table. The specification of a column list is not possible.

3. LOAD from DELIMITER-FORMAT

   When you select this function, you branch to the continuation form LOD.3. There you specify the input file in delimiter format.
   You must enter the DELIMITER character in the “TERMINATED BY” field as an alphanumeric literal or as a national literal.
   See the utility statement LOAD ... UNLOAD_FORMAT in the “SQL Reference Manual Part 2: Utilities”.

   If column values are only to be loaded to specific columns in the table, you must specify COLUMN LIST=Y and function 1 (Prepare). You then branch to the continuation form LOD.1.1 to prepare the column list.
   After you return to the LOD.3 form, the entered list can be changed again by COLUMN LIST=Y and function 2 (Modify).
   The LOAD statement is executed with COLUMN LIST=Y and function 3 (Execute).

   Continuation form LOD.1.1
   See “Continuation form LOD.1.1” on page 268.

4. LOAD from CSV-FORMAT

   When you select this function, you branch to the continuation form LOD.4. There you specify the input file in CSV format. Fundamental information on the layout of CSV files is provided in the “SQL Reference Manual Part 1: SQL Statements”.

   You must enter DELIMITER characters, QUOTE characters and ESCAPE characters as alphanumeric literals or national literals in the corresponding fields.
   See the utility statement LOAD ... UNLOAD_FORMAT in the “SQL Reference Manual Part 2: Utilities”.

   If column values are only to be loaded to specific columns in the table, you must specify COLUMN LIST=Y and function 1 (Prepare). You then branch to the continuation form LOD.1.1 to prepare the column list.
   After you return to the LOD.4 form, the entered list can be changed again by COLUMN LIST=Y and function 2 (Modify).
   The LOAD statement is executed with COLUMN LIST=Y and function 3 (Execute).

   Continuation form LOD.1.1
   See “Continuation form LOD.1.1” on page 268.

5. LOAD from user-defined format

   When you select this function, you branch to the continuation form LOD.5. There you enter the specifications for the user-defined format.
   See the “SQL Reference Manual Part 2: Utilities”, utility statement LOAD, syntax elements *load_description* and *load_column*.

   To create the user-defined format, enter USER DEFINED FORMAT=Y and function 1 (Prepare). You then branch to the continuation form LOD.5.1, in which you can create or modify a user-defined format.
   After you return to the LOD.5 form, the user-defined format can be changed again in the continuation form LOD.5.1 by USER DEFINED FORMAT=Y and function 2 (Modify).
   The LOAD statement is executed with USER DEFINED FORMAT=Y and function 3 (Execute).

   Continuation form LOD.5.1
   In this form you can define or modify the format of a column in the input file (“format description” input field)

   In the format description you can specify a NULL constraint. Thus, if the corresponding column of the user-defined format in the input file has the value NULL, this column in the table is assigned the value which you specify in the “LITERAL” input field.

   You can define only one column at a time. Once you have defined the column, press the DUE key to send the form off. Thus, if you define n columns, you must send the LOD.5.1 form off n times. By entering “<<“ and ”>>” in the command area and pressing the DUE key you can page back and forward through forms in which you have already defined columns.
   Once you have defined all the columns, you return to the LOD.5 form by pressing the F13 key.

   If column values which exist in standard format are only to be loaded to specific columns in the table, you must specify COLUMN LIST=Y and function 1 (Prepare) in the LOD.5 form. You then branch to the continuation form LOD.1.1 to prepare the column list.
   After you return to the LOD.5 form, the entered list can be changed again by COLUMN LIST=Y and function 2 (Modify).
   The LOAD statement is executed with COLUMN LIST=Y and function 3 (Execute).

   Continuation form LOD.1.1
   See “Continuation form LOD.1.1” on page 268.

Converting a SESAM/SQL V1 database to a SESAM/SQL table of the current version or modifying the table type (MIG - MIGRATE)

You call the MIG form by selecting function 7, MIGRATE, from the STM - START MENU form.

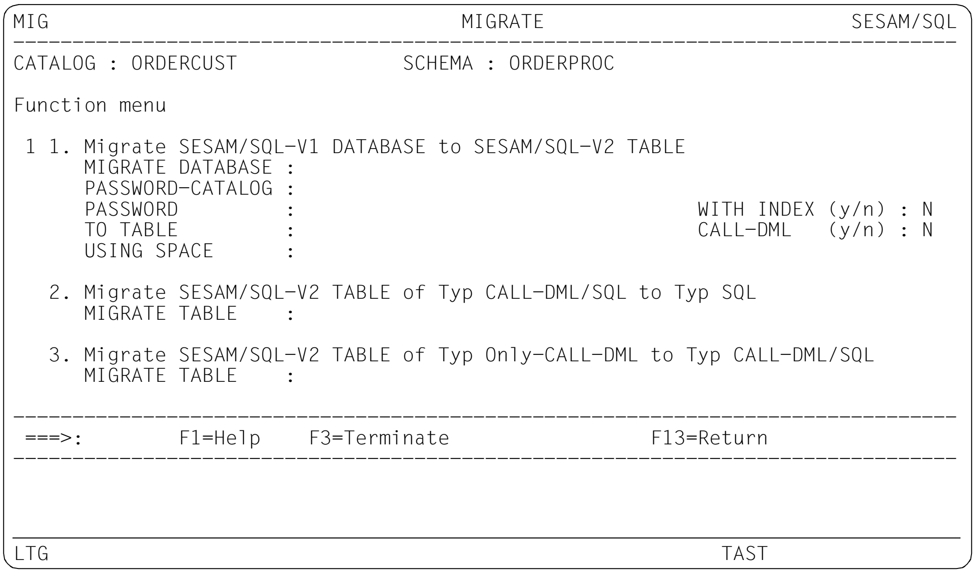
The MIG form allows you to

- convert a SESAM/SQL V1.1 or earlier database (referred to as a V.1 database below) into a base table of the current SESAM/SQL version,
- convert a CALL DML/SQL table into an SQL table, or a CALL DML-only table into a CALL DML/SQL table.

See the utility statement MIGRATE in the “SQL Reference Manual Part 2: Utilities”.

The MIG form has no continuation forms.

The MIG form


```
MIG                                      MIGRATE                                SESAM/SQL
-------------------------------------------------------------------------------------------
CATALOG : ORDERCUST                 SCHEMA : ORDERPROC

Function menu

 1 1. Migrate SESAM/SQL-V1 DATABASE to SESAM/SQL-V2 TABLE
       MIGRATE DATABASE :
       PASSWORD-CATALOG :
       PASSWORD         :                                  WITH INDEX (y/n) : N
       TO TABLE         :                                  CALL-DML   (y/n) : N
       USING SPACE      :

   2. Migrate SESAM/SQL-V2 TABLE of Typ CALL-DML/SQL to Typ SQL
       MIGRATE TABLE    :

   3. Migrate SESAM/SQL-V2 TABLE of Typ Only-CALL-DML to Typ CALL-DML/SQL
       MIGRATE TABLE    :

-------------------------------------------------------------------------------------------
 ===>:         F1=Help    F3=Terminate                 F13=Return
-------------------------------------------------------------------------------------------



LTG                                                             TAST
```

When you select functions 1 - 3, the statements are executed immediately.

The catalog space and schema in which the table is to be created must already exist and must be specified in the form (if they have not already been preset by the utility monitor).

Explanation of the functions

1. Migrate SESAM/SQL V1 DATABASE to SESAM/SQL V2 TABLE

   When you select this function, you convert a SESAM/SQL V1 database of the type CALL DML-only, CALL DML/SQL or SQL into a SESAM/SQL V2 table of the same type.

Specifications on the database to be migrated:

- The SESAM/SQL V1 database (to be specified in the MIGRATE DATABASE field) must be a backup file of type DB-SIB.
- If this database was defined with a password catalog and if the CALL DML field is defined with “y”, a backup file of this password catalog must be specified in the PASSWORD-CATALOG field. This backup file is of type PK-SIB.
- If applicable, a password for both backup files must be specified in the PASSWORD field.
- You use the WITH INDEX field to determine whether all indexes defined in the V1 database are to be transferred (“y”) or not (“n”).

Specifications on the destination database:

- The TO TABLE field must contain the name of the table to be newly created. This table must not yet exist.
- The type of the new table must be specified in the CALL DML field:
  - with “CALL DML : y”, the new table is created as a CALL DML-only table or as a CALL DML/SQL table;
  - with “CALL-DML : n”, it is created as an SQL table.
- If the USING SPACE field does not contain any space name on which the table is to be created, the table is created on the default space of the schema.

2. Migrate SESAM/SQL V2 TABLE of type CALL DML/SQL to type SQL

   When you select this function, you convert a CALL DML/SQL table specified in the MIGRATE TABLE field into an SQL table.

3. Migrate SESAM/SQL-V2 TABLE of type CALL DML-only to type CALL DML/SQL

   When you select this function, you convert a CALL DML-only table specified in the MIGRATE TABLE field into a CALL DML/SQL table.

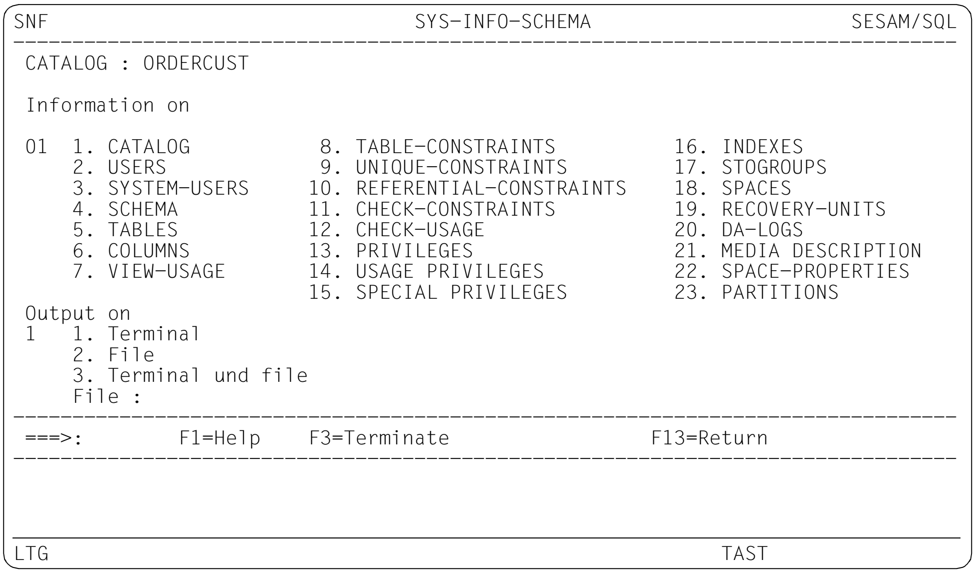
Querying metadata from SYS_INFO_SCHEMA (SNF - SYS-INFO-SCHEMA)

You can only call the SNF form by entering snf in the command area. This form is generally only available to the UNIVERSAL USER.

In the SNF form and its continuation forms you can output information from the SYS_INFO_SCHEMA information schema. The information can be output to the screen, to a file (BS2000 file or member of an LMS library), or to both.

You can handle the output as described for the INF - INFORMATION-SCHEMA form, see page 248.
See section “Overview of the information schemata” on page 182 and of the “SQL Reference Manual Part 1: SQL Statements”.

The SNF form



```
SNF                                 SYS-INFO-SCHEMA                          SESAM/SQL
--------------------------------------------------------------------------------
 CATALOG : ORDERCUST

 Information on

 01   1. CATALOG            8. TABLE-CONSTRAINTS           16. INDEXES
      2. USERS              9. UNIQUE-CONSTRAINTS          17. STOGROUPS
      3. SYSTEM-USERS      10. REFERENTIAL-CONSTRAINTS     18. SPACES
      4. SCHEMA            11. CHECK-CONSTRAINTS           19. RECOVERY-UNITS
      5. TABLES            12. CHECK-USAGE                 20. DA-LOGS
      6. COLUMNS           13. PRIVILEGES                  21. MEDIA DESCRIPTION
      7. VIEW-USAGE        14. USAGE PRIVILEGES            22. SPACE-PROPERTIES
                           15. SPECIAL PRIVILEGES          23. PARTITIONS
 Output on
 1    1. Terminal
      2. File
      3. Terminal und file
      File :
--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                   F13=Return
--------------------------------------------------------------------------------




LTG                                                           TAST
```

When you select functions 1 - 29, you branch to the corresponding continuation forms.

Under “Output on”, you specify whether the information is to be output to the screen, a file (BS2000 file or member of an LMS library) or both.

If you select function 2 or 3, you must specify a file name or library member name in the input field provided. If the input field is too short for the library member name, you can branch to the LIB - LIBRARY ELEMENT form to enter the complete library member name (see page 265).

Function 1 is the default.

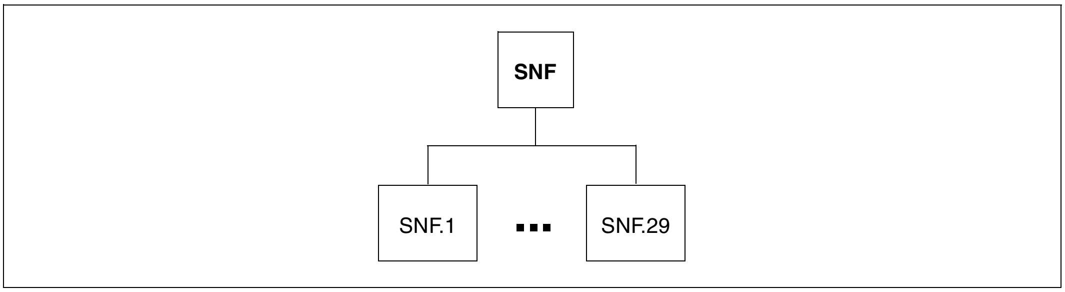


Figure 18: The SNF form and its continuation forms

Explanation of the functions

1. CATALOG

   When you select this function, you branch to the SNF.1 continuation form, in which the following information about the specified database is output:
   – the UNIVERSAL USER
   – default value for the LOG parameter (LOGGING)
   – coded character set of the database (CODE-TABLE)
   The SYS_CATALOGS view is accessed.

2. USERS

   When you select this function, you branch to the SNF.2 continuation form, in which all the authorization keys of the specified database are output together with their short names.
   The SYS_USERS view is accessed.

3. SYSTEM-USERS

   When you select this function, you branch to the SNF.3 continuation form, in which all the system entries of the specified database are output.
   The SYS_SYSTEM_ENTRIES view is accessed.

4. SCHEMA

   When you select this function, you branch to the SNF.4 continuation form, in which all the schemata of the specified database are output together with their owners.
   The SYS_SCHEMATA view is accessed.

5. TABLES

   When you select this function, you branch to the SNF.5 continuation form, in which information on all the base tables and views of the specified database is output. Each output form contains information on only one table. In the case of partitioned tables the "_PARTITIONS_" value is output in the SPACE field and the value 32767 in the SPACE-ID field.

   As SPACE is an input field for selecting output records, the "_PARTITIONS_" value for outputting partitioned tables can also be specified here. If a space name is specified for SPACE, partitioned tables are not output even if there is a partition on this space.

   The SYS_TABLES view is accessed.

6. COLUMNS

   When you select this function, you branch to the SNF.6 continuation form, in which information on all the columns of the specified database is output. Each output form contains information on only one column.
   The SYS_COLUMNS view is accessed.

7. VIEW-USAGE

   When you select this function, you branch to the SNF.7 continuation form, in which all the tables and columns of the specified database that are referenced by a view are output.
   The SYS_VIEW_USAGE view is accessed.

8. TABLE-CONSTRAINT

   When you select this function, you branch to the SNF.8 continuation form, in which all the table constraints of the specified database are output.
   The SYS_TABLE_CONSTRAINTS view is accessed.

9. UNIQUE-CONSTRAINTS

   When you select this function, you branch to the SNF.9 continuation form, in which all the UNIQUE constraints and primary key constraints of the specified database are output.
   The SYS_UNIQUE_CONSTRAINTS view is accessed.

10. REFERENTIAL-CONSTRAINTS

    When you select this function, you branch to the SNF.10 continuation form, in which all the referential constraints of the specified database are output.
    The SYS_REFERENTIAL_CONSTRAINTS view is accessed.

11. CHECK-CONSTRAINTS

    When you select this function, you branch to the SNF.11 continuation form, in which all the check constraints of the specified database are output.
    The SYS_CHECK_CONSTRAINTS view is accessed.

12. CHECK-USAGE

    When you select this function, you branch to the SNF.12 continuation form, in which all the tables and columns of the specified database that are subject to a check constraint are output. It also includes the integrity-constraint name assigned by the system (in the CONSTRAINT-NAME output field).
    The SYS_CHECK_USAGE view is accessed.

13. PRIVILEGES

    When you select this function, you branch to the SNF.13 continuation form, in which all the privileges of the specified database are output. A record is output for each privilege assigned.
    The SYS_PRIVILEGES view is accessed.

14. USAGE-PRIVILEGES

    When you select this function, you branch to the SNF.14 continuation form, in which the owners of the USAGE special privileges granted for the specified database are output. The USAGE special privilege permits the usage of a storage group. A record is output for each USAGE special privilege granted.
    The SYS_USAGE_PRIVILEGES view is accessed.

15. SPECIAL-PRIVILEGES

    When you select this function, you branch to the SNF.15 continuation form, in which all the special privileges (except USAGE) for the specified database are output. A record is output for each special privilege granted.
    The SYS_SPECIAL_PRIVILEGES view is accessed.

16. INDEXES

    When you select this function, you branch to the SNF.16 continuation form, in which all the indexed columns of the specified database are output.
    The SYS_INDEXES view is accessed.

17. STOGROUPS

    When you select this function, you branch to the SNF.17 continuation form, in which all the storage groups of the specified database are output. If a storage group is distributed across several disks you have specified, one record is output per disk.
    The SYS_STOGROUPS view is accessed.

18. SPACES

    When you select this function, you branch to the SNF.18 continuation form, in which all the spaces of the specified database are output. A record is output for each space.
    The SYS_SPACES view is accessed.

19. RECOVERY-UNITS

    When you select this function, you branch to the SNF.19 continuation form, in which the recovery unit records of the specified database are output.
    The SYS_RECOVERY_UNITS view is accessed.

20. DA-LOGS

    When you select this function, you branch to the SNF.20 continuation form, in which the DA-LOG files of the specified database are output. A record is output for each DA-LOG file.
    The SYS_DA_LOGS view is accessed.

21. MEDIA-DESCRIPTION

    When you select this function, you branch to the SNF.21 continuation form, in which all the records of the media table of the specified database are output.
    The SYS_MEDIA_DESCRIPTIONS view is accessed.

22. SPACE-PROPERTIES

    When you select this function, you branch to the SNF.22 continuation form, in which space properties are output.
    The SYS_SPACE_PROPERTIES view is accessed.

23. PARTITIONS

    When you select this function, you branch to the SNF.23 continuation form, in which the properties of partitions in a partitioned table are output.
    The SYS_PARTITIONS view is accessed.

24. ROUTINES

    When you select this function, you branch to the SNF.24 continuation form, in which the properties of routines are output.
    The SYS_ROUTINES view is accessed.

25. PARAMETERS

    When you select this function, you branch to the SNF.25 continuation form, in which the parameters of routines are output.
    The SYS_PARAMETERS view is accessed.

26. ROUTINE-PRIVILEGES

    When you select this function, you branch to the SNF.26 continuation form, in which the privileges of routines are output.
    The SYS_ROUTINE_PRIVILEGES view is accessed.

27. ROUTINE-USAGE

    When you select this function, you branch to the SNF.27 continuation form, in which the tables and columns used by routines are output.
    The SYS_ROUTINE_USAGE view is accessed.

28. ROUTINE-ROUTINE-USAGE

    When you select this function, you branch to the SNF.28 continuation form, in which the routines called by routines are output.
    The SYS_ROUTINE_ROUTINE_USAGE view is accessed.

29. VIEW-ROUTINE-USAGE

    When you select this function, you branch to the SNF.29 continuation form, in which the routines used by views are output.
    The SYS_VIEW_ROUTINE_USAGE view is accessed.

Issuing dynamically compilable SQL statements (SQL - SQL-STATEMENTS)

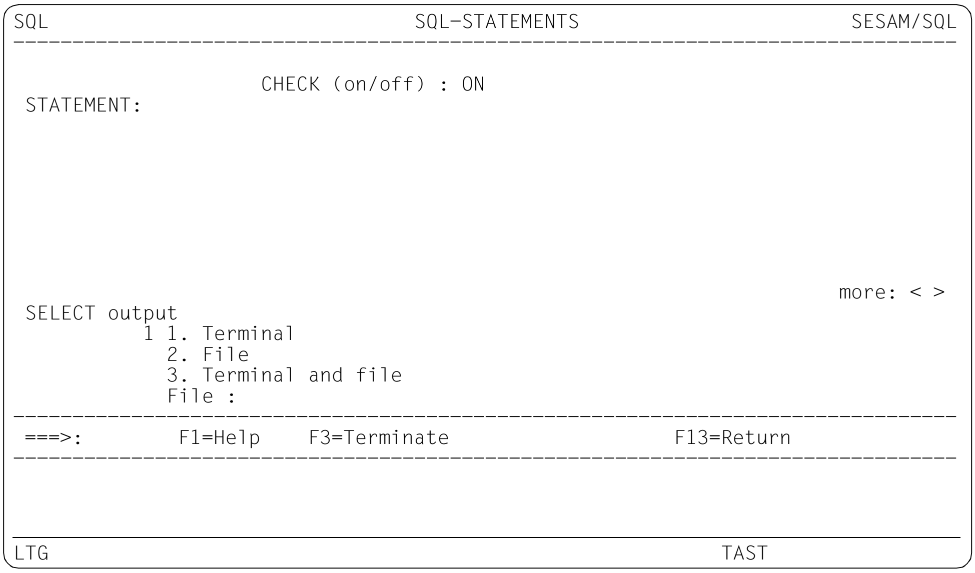
You call the SQL form either by selecting function 4, SQL-STATEMENT, from the STM - START MENU start form or by entering sql in the command area.

In the SQL form you can issue any dynamically compilable SQL statement. This is necessary, above all, in the case of corrections after a CHECK CONSTRAINTS function has been executed and it has been established that integrity constraints have been violated.

You can also change the authorization key, database and schema in this form by issuing the SQL statements SET SESSION AUTHORIZATION, SET CATALOG and SET SCHEMA. The changed values are only valid in relation to the DBH. The default configuration data does not change. In other words, as soon as you branch to another function, the default configuration data applies again.

You can output the information to the screen, a file (BS2000 file or member of an LMS library) or both.
See the section “Dynamic SQL” and the SQL statements PREPARE, EXECUTE and EXECUTE IMMEDIATELY in the “SQL Reference Manual Part 1: SQL Statements”.

The SQL form


```
SQL                                  SQL-STATEMENTS                          SESAM/SQL
---------------------------------------------------------------------------------------

                     CHECK (on/off) : ON
 STATEMENT:









                                                                            more: < >
 SELECT output
          1 1. Terminal
            2. File
            3. Terminal and file
            File :
---------------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                   F13=Return
---------------------------------------------------------------------------------------



LTG                                                          TAST
```

If you have to carry out corrections after executing the CHECK CONSTRAINTS function because integrity constraints have been violated, you must enter “off” in the CHECK input field.

If a pragma is to precede an SQL statement, the pragma must be in its own line before the SQL statement and be preceded by the string --%PRAGMA.

If a syntax error arises, a message is issued showing the location of the syntax error in terms of line and column. If the output is extensive, it is recommended that you call the program EDT (see section “Call file editor EDT as a subroutine” on page 130), open the log file and search for the relevant line in this file.

i If “off” is entered in the CHECK input field, the line number is incremented by one in the event of syntax errors, as the CHECK=OFF pragma is represented internally in a separate line.

Under “SELECT output”, you can specify whether the records found are to be output to the screen, a file (BS2000 file or member of an LMS library) or both.

If you select function 2 or 3, you must specify a file name or library member name in the input field provided. If the input field is too short for the library member name, you can branch to the LIB - LIBRARY ELEMENT form to enter the complete library member name (see page 265).
The file is closed when the activity is ended or when you branch to the EDT. You can thus display the file on the screen without terminating the utility monitor.

If you select function 1 or 3 and enter a SELECT statement that finds sets of records, you branch to the continuation form SQL.1. The SELECT statement remains in the SQL form.

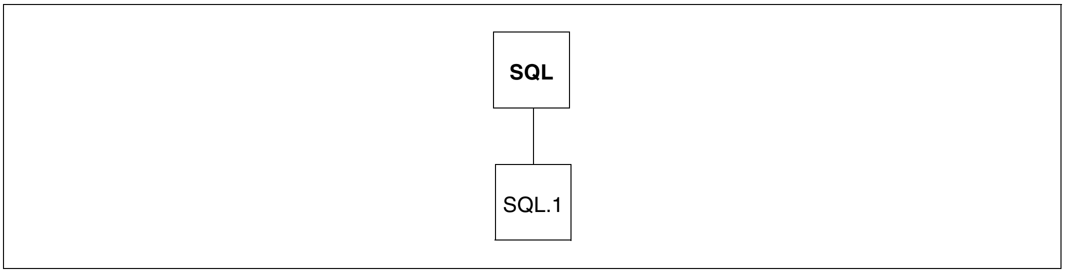


Figure 19: The SQL form and its continuation form

Explanation of the functions

1. Terminal

   When you select this function, you branch to the SQL.1 continuation form, in which the records found by the SELECT statement issued in the SQL form are output.
   The output appears line by line.
   The column name is output in the first output field (up to the colon). If the output field for the column value contains more than 45 characters, it is followed by a one-character input field and a scrollable output field for the column value. By entering < and > in the input field you can scroll the output field that follows it to the left and right.

   Up to 17 columns can be output simultaneously in a form. If a record has more than 17 columns, you can page back or forward by entering a plus (+) or minus sign (-) in the command area or by pressing the F8 or F7 key.

   Up to 100 columns can be output per record. If a record has more than 100 columns (e.g. in the case of SELECT * FROM ...), the SELECT statement is rejected with an error message.

   The records in the set of records found appear one at a time. When you press the DUE key, another record appears (assuming there is one). Only the next record in the set of those found can be output. It is not possible to skip to the end or beginning of a set of records, page back, or skip a specific number of records.

   Numerical values that are packed or stored in binary form are displayed unpacked on the screen.

   Alphanumeric values of the data type CHARACTER (VARYING) are displayed as they are contained in the database. On the screen the characters are displayed in the coded character set which is selected for the screen (see /MODIFY-TERMINAL-OPTIONS).

   To ensure correct representation of the characters, the coded character set which is selected for the screen should be the same as that used by the database. Otherwise characters may not be displayed correctly on the screen.

   National values of the data type NATIONAL CHARACTER (VARYING) are converted to the coded character set of the database for output on the screen. Characters which cannot be converted are represented by a period (.).

   When you reach the last record in a set, the utility monitor displays a message to this effect.

   By pressing the F13 key or entering F13 in the command area you terminate the output and return to the SQL form.

2. File

   When you select this function, the records found for the SELECT statement issued in the SQL form are output to the file you specify. The output file can be stored either in a BS2000 file or as a member of an LMS library.
   The output appears line by line.
   The first 31 positions are reserved for the column name. This is followed by a colon and a blank.

   The remaining 99 positions in the line are filled with the column value. If a column value has more than 99 characters, it is continued in the next line. The first 31 positions of the new line are filled with blanks.

   Alphanumeric values of the data type CHARACTER (VARYING) are written into the file as they are contained in the database. To ensure correct 8-bit representation of the characters in the file editor EDT, /MODIFY-FILE-ATTRIBUTES should be used to select the database's coded character set for the file, too.

   National values of the data type NATIONAL CHARACTER (VARYING) are converted to the database's coded character set for file output. Characters which cannot be converted are represented by a period (.).

   The different records are separated by a blank line.

   All the records found for the SELECT statement issued are output.

   You can cancel the file output by pressing the K2 key and entering the BS2000 command INFORM-PROGRAM MSG=C'SEE,BREAK' . After the command is issued, the utility monitor returns to the form displayed last and issues an appropriate message.

3. Terminal and file

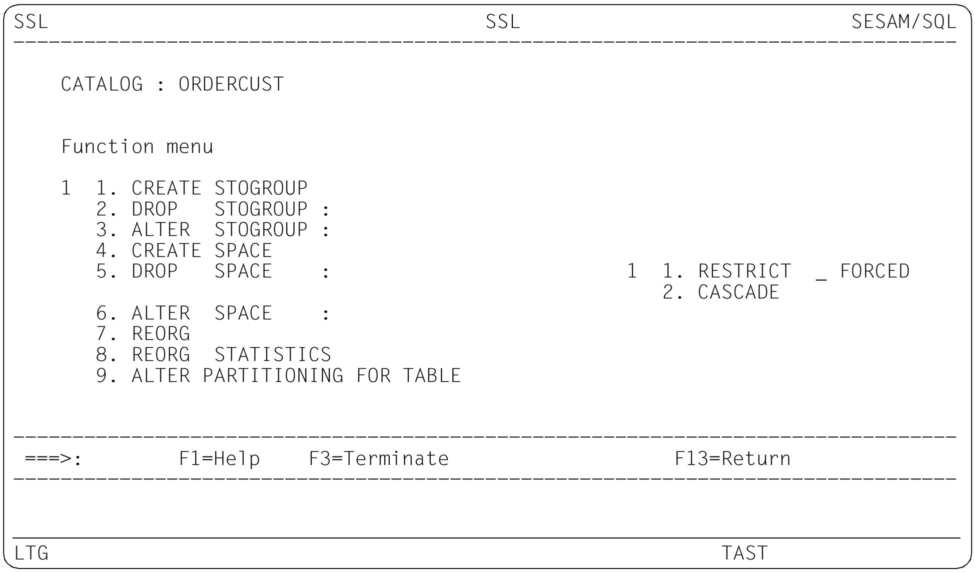
   When you select this function, the records found for the SELECT statement issued in the SQL form are output to the screen and a file (see the explanations of functions 1 and 2 above).

Controlling storage management (SSL - SSL)

You call the SSL form either by selecting function 11, SSL, from the STM - START MENU start from or by entering ssl in the command area.

In the SSL form and its continuation forms, you can create, modify and delete storage groups and spaces, reorganize spaces, reorganize global statistics for indexes and change the partitioning of a base table.

The SSL form



```
SSL                                 SSL                           SESAM/SQL
--------------------------------------------------------------------------------

   CATALOG : ORDERCUST


   Function menu

   1  1. CREATE STOGROUP
      2. DROP   STOGROUP :
      3. ALTER  STOGROUP :
      4. CREATE SPACE
      5. DROP   SPACE    :                       1  1. RESTRICT  _ FORCED
                                                    2. CASCADE
      6. ALTER  SPACE    :
      7. REORG
      8. REORG  STATISTICS
      9. ALTER PARTITIONING FOR TABLE


--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate                  F13=Return
--------------------------------------------------------------------------------



LTG                                                        TAST
```

When you select functions 2 (DROP STOGROUP) and 5 (DROP SPACE), the statements are executed immediately.

When you select functions 1, 3, 4 and 6 - 9, you branch to continuation forms.

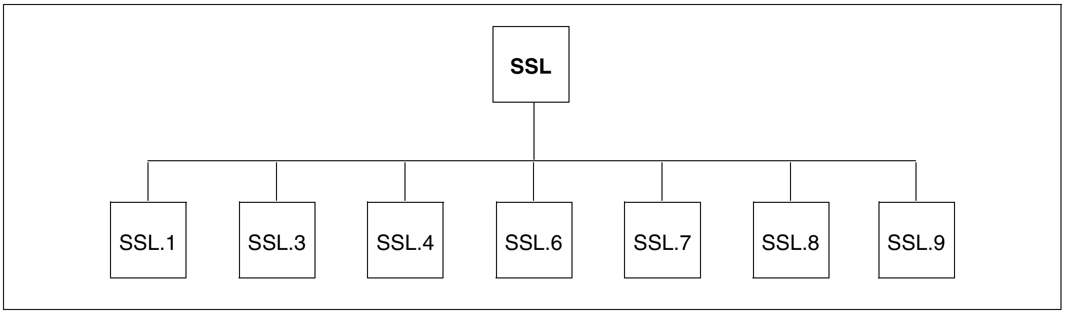


Figure 20: The SSL form and its continuation forms

Explanation of the functions

1. CREATE STOGROUP

   When you select this function, you branch to the SSL.1 continuation form. This allows you to create a storage group (see also the SEE-STOGROUP configuration parameter on page 81).
   The pageable VOLUMES input table is preset with the PUBLIC parameter. If you want to create the storage group on a private disk, you have to overwrite PUBLIC. You can specify up to 100 private disks.
   If you have specified a private disk, you also have to supply the ON DEVICE-TYPE input field with a parameter.
   See the SQL statement CREATE STOGROUP in the “SQL Reference Manual Part 1: SQL Statements”.

2. DROP STOGROUP

   When you select this function, the specified storage group is deleted.
   See the SQL statement DROP STOGROUP in the “SQL Reference Manual Part 1: SQL Statements”.

3. ALTER STOGROUP

   When you select this function, you branch to the SSL.3 continuation form. This allows you to modify the description of the specified storage group.
   See the SQL statement ALTER STOGROUP in the “SQL Reference Manual Part 1: SQL Statements”.

4. CREATE SPACE

   When you select this function, you branch to the SSL.4 continuation form. This allows you to create a user space.
   The authorization key, the storage group and the parameters for the user space are preset with defaults and can be changed (see also the table of the configuration parameters on table “Configuration parameter” on page 79).
   See the SQL statement CREATE SPACE in the “SQL Reference Manual Part 1: SQL Statements”.

5. DROP SPACE

   When you select this function, the specified user space is deleted. You can choose between the RESTRICT and CASCADE parameters and mark the optional parameter DEFERRED.
   See the SQL statement DROP SPACE in the “SQL Reference Manual Part 1: SQL Statements”.

6. ALTER SPACE

   When you select this function, you branch to the SSL.6 continuation form. This allows you to change the properties of the specified user space. You can also use the special name "CATALOG" (with double quotes) to change the properties of the catalog space. The NO LOG parameter may only not be specified for the catalog space.
   See the SQL statement ALTER SPACE in the “SQL Reference Manual Part 1: SQL Statements”.

7. REORG

   When you select this function, you branch to the SSL.7 continuation form. This allows you to reorganize a specific space, the catalog space, all spaces or a base table. If you want to reorganize all spaces, you need the appropriate authorization.

   REORG for spaces is also permitted for replications. The user spaces can also be reorganized, but the catalog space of the replication cannot.
   If the replication is a partial replication, only the user spaces contained in the partial replication can be reorganized. The reorganization of the spaces which are in the INFORMATION SCHEMA, but not in the partial replication, gives rise to SQLSTATEs which are entered in the log file assigned with SEE-SSTLOG for SQLSTATEs or in the standard log file. SQLSTATEs are also output in the message area of the form. The reorganization of all the other space is continued.

   The following parameters can be specified for spaces in the case of REORG:

   - You can use the NEW ROW_IDS parameter to reassign the table record numbers.
   - When COPY is selected, the reorganized work file is copied into the space file, i.e. the space retains its original position.

When RENAME is selected, the space file is deleted and the reorganized work file is renamed, i.e. the space takes the place of the work file after reorganization.
If the user makes no selection for COPY or RENAME, the presettings of SESAM/SQL then apply.

- If you insert a cross next to MINIMIZE, the storage space for the space file which is no longer required after reorganization is released.

The following parameters can be specified for a base table in the case of REORG:

- You can use the ON SPACE parameter to specify the name of the space on which the base table or partition resides.

See the utility statement REORG in the "SQL Reference Manual Part 2: Utilities".

8. REORG STATISTICS

   When you select this function, you branch to the SSL.8 continuation form. This allows you to reorganize the global statistics of the specified index.
   See the SQL statement REORG STATISTICS in the "SQL Reference Manual Part 1: SQL Statements".

9. ALTER PARTITIONING FOR TABLE

   When you select this function, you branch to the SSL.9 continuation form. This allows you to change the partitioning of a base table.

   The ADD PARTITION function enables you to add a new partition to a partitioned or non-partitioned base table. A non-partitioned base table is then converted into a partitioned base table. You can define the upper boundaries of the partitions affected and determine whether records are to be transferred to the new partition.

   The ALTER PARTITION function enables you to define the upper boundary of a partition and determine whether records are to be transferred.

   The DROP PARTITION function enables you to delete a partition and determine whether records are to be transferred or deleted. If the base table contains only one partition after the delete operation, it is automatically converted to a non-partitioned base table.

   The three functions mentioned above display further continuation forms. There you can specifically control the creation of indexes.

   See the utility statement ALTER PARTITIONING FOR TABLE in the "SQL Reference Manual Part 2: Utilities".

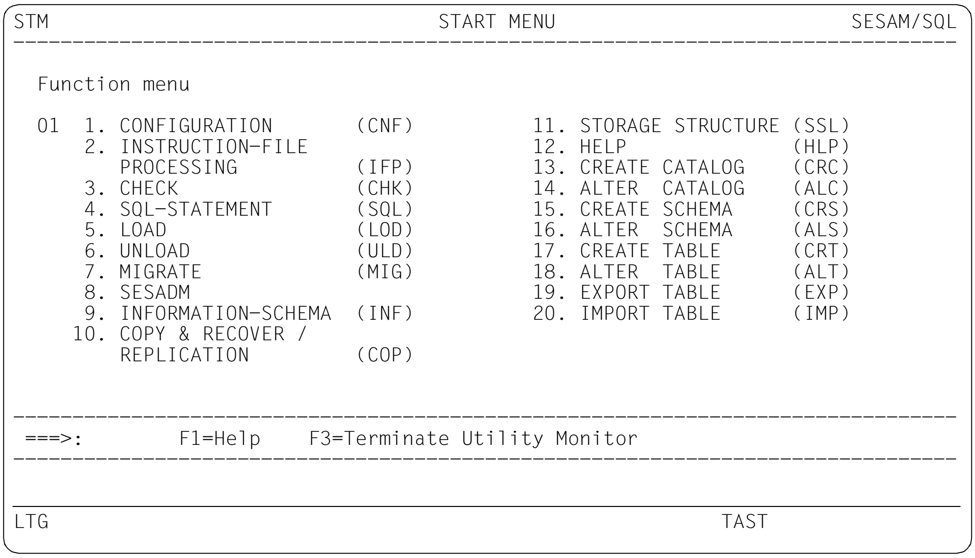
Calling main functions and SESADM (STM - START MENU)

If you have assigned all the necessary configuration data, the STM startform is the first to appear when you start the utility monitor. You can also return to this form during a session by entering stm in the command area.

Unless you terminate the utility monitor by pressing the F3 key or entering F3 in the command area, the STM form is also the last one to appear before you exit the utility monitor.

The STM form allows you to call the utility monitor's main functions and the SESADM administration program as a subroutine.

The STM form



```
STM                                START MENU                                SESAM/SQL
--------------------------------------------------------------------------------

  Function menu

  01  1. CONFIGURATION       (CNF)          11. STORAGE STRUCTURE (SSL)
      2. INSTRUCTION-FILE                   12. HELP              (HLP)
         PROCESSING          (IFP)          13. CREATE CATALOG    (CRC)
      3. CHECK               (CHK)          14. ALTER  CATALOG    (ALC)
      4. SQL-STATEMENT       (SQL)          15. CREATE SCHEMA     (CRS)
      5. LOAD                (LOD)          16. ALTER  SCHEMA     (ALS)
      6. UNLOAD              (ULD)          17. CREATE TABLE      (CRT)
      7. MIGRATE             (MIG)          18. ALTER  TABLE      (ALT)
      8. SESADM                             19. EXPORT TABLE      (EXP)
      9. INFORMATION-SCHEMA  (INF)          20. IMPORT TABLE      (IMP)
     10. COPY & RECOVER /
         REPLICATION         (COP)


--------------------------------------------------------------------------------
 ===>:        F1=Help    F3=Terminate Utility Monitor
--------------------------------------------------------------------------------

LTG                                                         TAST
```

When you select functions 1 - 7 and 9 - 20, you branch to the forms of the main functions.

When you select function 8, you call the SESADM administration program.

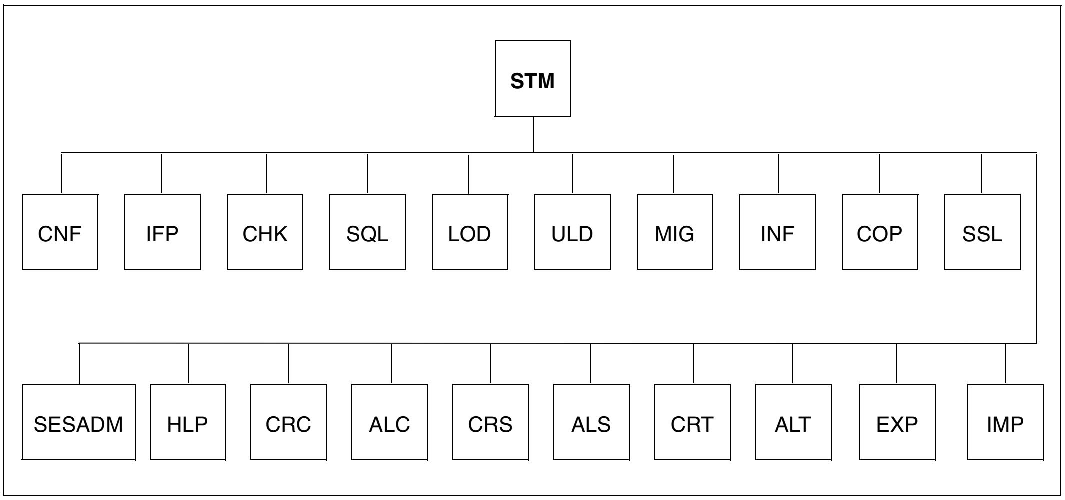


Figure 21: The STM form and its branching options

Explanation of the functions

1. CONFIGURATION (CNF)

   When you select this function, you branch to the CNF - CONFIGURATION form (see page 206).

2. INSTRUCTION-FILE PROCESSING (IFP)

   When you select this function, you branch to the IFP - INSTRUCTION FILE PROCESSING form (see page 243).

3. CHECK (CHK)

   When you select this function, you branch to the CHK - CHECK form (see page 202).

4. SQL-STATEMENT (SQL)

   When you select this function, you branch to the SQL - SQL-STATEMENTS form (see page 279).

5. LOAD (LOD)

   When you select this function, you branch to the LOD - LOAD form (see page 266).

6. UNLOAD (ULD)

   When you select this function, you branch to the ULD - UNLOAD form (see page 291).

7. MIGRATE (MIG)

   When you select this function, you branch to the MIG - MIGRATE form (see page 271).

8. SESADM

   When you select this function, the SESADM administration program is called as a subroutine. The dialog with SESADM runs via an SDF interface until SESADM is terminated.
   See also “Administration via CALL DML” on page 129 and the “Database Operation” manual.

9. INFORMATION-SCHEMA (INF)

   When you select this function, you branch to the INF - INFORMATION-SCHEMA form (see page 247).

10. COPY & RECOVER / REPLICATION (COP)

    When you select this function, you branch to the COP - COPY & RECOVER / REPLICATION form (see page 208).

11. SSL (SSL)

    When you select this function, you branch to the SSL - SSL form (see page 283).

12. HELP (HLP)

    When you select this function, you branch to the HLP - HELP form (see page 239).

13. CREATE CATALOG (CRC)

    When you select this function, you branch to the CRC - CREATE CATALOG form (see page 227).

14. ALTER CATALOG (ALC)

    When you select this function, you branch to the ALC - ALTER CATALOG form (see page 189).

15. CREATE SCHEMA (CRS)

    When you select this function, you branch to the CRS - CREATE SCHEMA form (see page 230).

16. ALTER SCHEMA (ALS)

    When you select this function, you branch to the ALS - ALTER SCHEMA form (see page 193).

17. CREATE TABLE (CRT)

    When you select this function, you branch to the CRT - CREATE TABLE form (see page 233).

18. ALTER TABLE (ALT)

    When you select this function, you branch to the ALT - ALTER TABLE form (see page 196).

19. EXPORT TABLE (EXP)

    When you select this function, you branch to the EXP- EXPORT TABLE form (see page 238).

20. IMPORT TABLE (IMP)

    When you select this function, you branch to the IMP- IMPORT TABLE form (see page 244).

Unloading data from a table into a file (ULD - UNLOAD)

You call the ULD form either by selecting function 6, UNLOAD, from the STM - START MENU form or by entering uld in the command area.

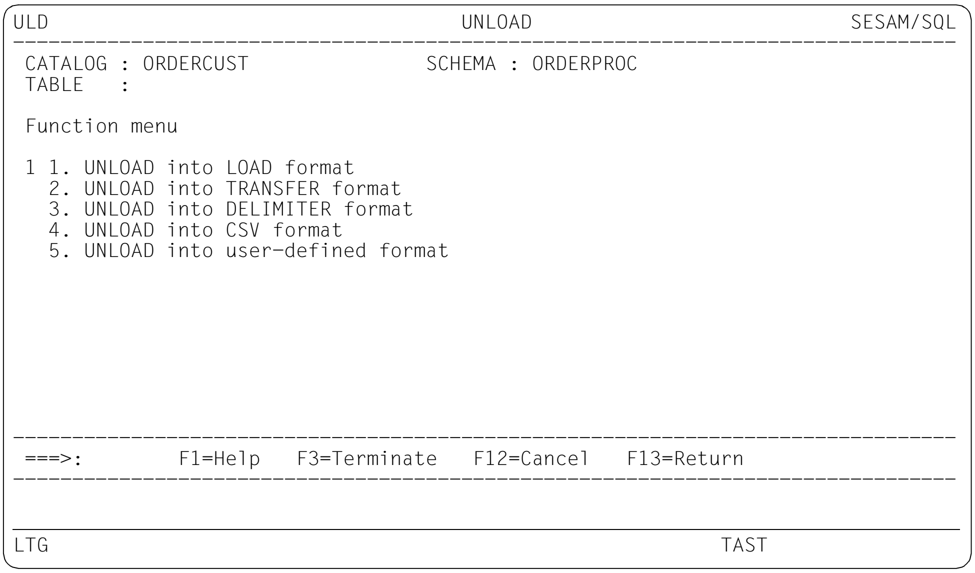
In the ULD form and its continuation forms, you unload into a file either all the data of a table or the data from specific columns of a table. You can unload data in predefined formats (the LOAD format, TRANSFER format, DELIMITER format or CSV format) or in an output format that you yourself define.

UNLOAD ONLINE enables you to unload user data from base tables or views. You can restrict the data to be unloaded by means of a search condition and define a collation sequence for the output file.
UNLOAD OFFLINE enables you to unload user data from base tables. You can also specify the FROM SPACE and FROM COPY_FILE clauses in TRANSFER format.

The number of records that are loaded from the table into a file is displayed in the message area. If other utility monitor messages or SQLSTATEs are present, this is indicated by a “M+-” in the command area. In this case, you can use “m+” and “m-” in the command area to scroll through the messages.

See the utility statement UNLOAD in the “SQL Reference Manual Part 2: Utilities”.

The ULD form



```
ULD                                        UNLOAD                                SESAM/SQL
-------------------------------------------------------------------------------------------
 CATALOG : ORDERCUST                  SCHEMA : ORDERPROC
 TABLE   :

 Function menu

 1 1. UNLOAD into LOAD format
   2. UNLOAD into TRANSFER format
   3. UNLOAD into DELIMITER format
   4. UNLOAD into CSV format
   5. UNLOAD into user-defined format







-------------------------------------------------------------------------------------------
 ===>:        F1=Help   F3=Terminate   F12=Cancel   F13=Return
-------------------------------------------------------------------------------------------


LTG                                                               TAST
```

When you select functions 1 - 5, you branch to the corresponding continuation forms.

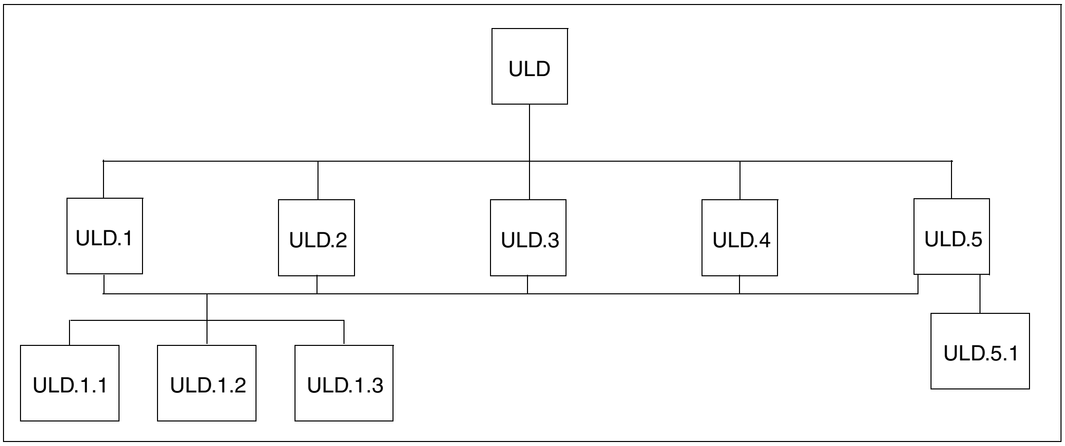


Figure 22: The ULD form and its continuation forms

Shared input fields in the forms ULD.1 through ULD.5

You use ONLINE / OFFLINE to define the UNLOAD mode.

- When UNLOAD ONLINE (entry “1”, default value) is specified, the user space is not locked exclusively. Only the same locks are requested as for a DML search statement. All the processing takes place in a DBH task. The data to be unloaded can be restricted by means of a search condition. A collation sequence can be defined for the output file. The contents of both the base tables and the views can be unloaded.
- When UNLOAD OFFLINE (entry “2”) is specified, SESAM/SQL locks the user space to prevent other users from modifying it. Processing takes place largely in a service task. The FROM SPACE and FROM COPY_FILE clauses can be specified.

You use TABLE / DATA to define whether the entire table, with all the columns, or only individual columns are to be unloaded.

- When TABLE (entry “1”, default value) is specified, the entire table, with all columns, is unloaded.
- When DATA (entry “2”) is specified, only certain columns are unloaded. You must enter COLUMN LIST=Y and select the columns, see below.

When COLUMN LIST=Y or WHERE=Y or ORDER BY=Y and function 1 (Prepare) is specified, you branch to the continuation forms ULD.1.1 and/or ULD1.2 and/or ULD1.3. You can make the entries you require there.
After you have returned to the higher-ranking form, you can change the entries again with function 2 (Modify).
The UNLOAD statement is executed with function 3 (Execute).

The continuation form ULD.1.1 (column list) is like continuation form LOD.1.1, see page 270. In the case of multiple columns, specify the occurrence in the “COMPONENT” field.

The verbal text for the clauses can be specified in the continuation forms ULD.1.2 (WHERE clause) and ULD.1.3 (ORDER BY clause). You can scroll within the field using “<“ and ”>”. Once you have made your entry you return to the higher-ranking form by pressing the F13 key.

Explanation of the functions

1. UNLOAD into LOAD-FORMAT

   When you select this function, you branch to the ULD.1 continuation form. There you specify that the output file is to have the same format as an input file required by LOAD.
   See the utility statement UNLOAD ... LOAD_FORMAT in the “SQL Reference Manual Part 2: Utilities”.

2. UNLOAD into TRANSFER-FORMAT

   When you select this function, you branch to the ULD.2 continuation form. There you specify that the output file is to have the same format as a transfer file required by LOAD.
   See the utility statement UNLOAD ... TRANSFER_FORMAT in the “SQL Reference Manual Part 2: Utilities”.

   “FROM SPACE” and “FROM COPY-FILE” are entered to support recovery and may only be specified under the following conditions:
   – An entire table is unloaded
   – OFFLINE mode is selected
   – The space specified is in the “recover pending” state (FROM COPY_FILE clause)
   – The space specified or copy specified is not marked as defective (FROM SPACE clause)

3. UNLOAD into DELIMITER-FORMAT

   When you select this function, you branch to the ULD.3 continuation form. There you specify the output file in delimiter format.
   See the utility statement UNLOAD ... DELIMITER_FORMAT in the “SQL Reference Manual Part 2: Utilities”.

   You must enter the DELIMITER character in the “TERMINATED BY” field as an alphanumeric literal or as a national literal.

4. UNLOAD in CSV-FORMAT

   When you select this function, you branch to the ULD.4 continuation form. There you specify the output file in CSV format. Fundamental information on the layout of CSV files is provided in the “SQL Reference Manual Part 1: SQL Statements”.

   You must enter DELIMITER character, QUOTE character and ESCAPE character as an alphanumeric literal or as a national literal.
   See the utility statement UNLOAD ... CSV_FORMAT in the “SQL Reference Manual Part 2: Utilities”.

5. UNLOAD into user-defined format

   When you select this function, you branch to the ULD.5 continuation form. There you specify the output file in the user-defined format.
   See the utility statement UNLOAD and the syntax elements*unload_description* and *unload_column* in the "SQL Reference Manual Part 2: Utilities".

   To create the user-defined format, enter USER DEFINED FORMAT=Y and function 1 (Prepare). You then branch to the continuation form ULD.5.1, in which you can create or modify a user-defined format.
   After you return to the ULD.5 form, the user-defined format can be changed again in the continuation form ULD.5.1 by USER DEFINED FORMAT=Y and function 2 (Modify).
   The UNLOAD statement is executed with USER DEFINED FORMAT=Y and function 3 (Execute).

   Continuation form ULD.5.1

   In this form you can define or modify the format of a column in the output file ("format description" input field)

   In the format description you can specify how a NULL value is represented in the output file. Thus, if the corresponding column of the user-defined format in the table has the value NULL, this column in the output file is assigned the value which you specify in the "LITERAL" input field.

   You can define only one column at a time. Once you have defined the column, press the DUE key to send the form off. Thus, if you define n columns, you must send the ULD.5.1 form off n times. By entering "<<" and ">>" in the command area and pressing the DUE key you can page back and forward through forms in which you have already defined columns.
   Once you have defined all the columns, you return to the LOD.5 form by pressing the F13 key.

   i If you define the columns in the format description for the user-defined format and also specify the column list (see page 293), the number of elements in the two lists must be the same.

Related publications

You will find the manuals on the internet at *http://manuals.ts.fujitsu.com*. You can order printed versions of manuals which are displayed with the order number.

SESAM/SQL-Server (BS2000)
SQL Reference Manual Part 1: SQL Statements
User Guide

SESAM/SQL-Server (BS2000)
SQL Reference Manual Part 2: Utilities
User Guide

SESAM/SQL-Server (BS2000)
CALL-DM Applications
User Guide

SESAM/SQL-Server (BS2000)
Core manual
User Guide

SESAM/SQL-Server (BS2000)
Database Operation
User Guide

SESAM/SQL-Server (BS2000)
Messages
User Guide

SESAM/SQL-Server (BS2000)
Performance
User Guide

WebTA access for SESAM/SQL
(Product document, also available on the manual server)

ESQL-COBOL (BS2000)
ESQL-COBOL for SESAM/SQL-Server
User Guide

SESAM-DBAccess
Server-Installation, Administration (available on the manual server only)

EDT (BS2000)
Statements
User Guide

LMS (BS2000)
SDF Format
User Guide

BS2000 OSD/BC
Commands
User Guide

HSMS (BS2000)
Hierarchical Storage Management System
User Guide

Index

In the index, **bold** page numbers refer to the main sources of the index entries, while *italicized* page numbers refer to examples. The collation sequence is a follows: symbols come before digits which come before letters. A punctuation mark is a symbol.