

## Übung Nr. 4

```

TCPIP MSG10 ==> SOURCE DATA SET = SYS1.LOCAL.VTAMLST(USSTCPIP)

02/04/01                W E L C O M E   T O                16:54:39

          SSSSSS  //   3333333  9999999  0000000
        SS      //   33  33  99  99  00  00
       SS      //           33  99  99  00  00
      SSSS    //   33333  9999999  00  00
     SS     //           33           99  00  00
    SS //   33  33  99  99  00  00
   SSSSSS //   3333333  9999999  0000000

YOUR TERMINAL NAME IS :                YOUR IP ADDRESS IS : 217.002.090.033

          APPLICATION DEVELOPMENT SYSTEM
          OS/390 RELEASE 2.7.0

===> ENTER "L " FOLLOWED BY THE APPLID YOU WISH TO LOGON TO.  EXAMPLE "L TSO"
      FOR TSO/E OR "L C001" FOR THE CICSC001 CICS APPLICATION.

l tso

```

Wir wollen für den Benutzer Spruth eine Transaktion erzeugen, welche auf eine OS/390 DB2 relationale Datenbank zugreift. In dem hier beschriebenen Schritt wird von uns die Datenbank mit einer einzigen einfachen Tabelle angelegt und anschließend mit einigen wenigen Daten gefüllt.

Wir loggen uns als TSO Benutzer ein. Enter

```

----- TSO/E LOGON -----

Enter LOGON parameters below:                RACF LOGON parameters:

Userid    ==> SPRUTH
Password  ==>
Procedure ==> IKJACCNT                       Group Ident  ==>

Acct Nmbr ==> ACCT#
Size      ==>
Perform   ==>
Command   ==>

Enter an 'S' before each option desired below:
      -Nomail      -Nonotice      -Reconnect      -OIDcard

PF1/PF13 ==> Help    PF3/PF15 ==> Logoff    PA1 ==> Attention    PA2 ==> Reshow
You may request specific help information by entering a '?' in any entry field

```

Enter, warten bis neuer Screen erscheint, wieder enter, enter

CUSTOMPAC MASTER APPLICATION MENU

OPTION ==>

SCROLL ==> PAGE

IS ISMF - Interactive Storage Management Facility  
P PDF - ISPF/Program Development Facility  
ATC ATC - Application Testing Collection  
ART ARTT - Automated Regression Testing Tool  
DB2 DB2 - Perform DATABASE 2 interactive functions  
QMF QMF - QMF Query Management Facility  
C CPSM - CICSplex/SM  
M MQ - MQSeries  
IP IPCS - Interactive Problem Control Facility  
OS SUPPORT - OS/390 ISPF System Support Options  
OU USER - OS/390 ISPF User Options  
SM SMP/E - SMP/E Dialogs  
SD SDSF - System Display and Search Facility  
R RACF - Resource Access Control Facility  
DI DITTO - Data Interfile Transfer, Testing and Operations  
HC HCD - Hardware Configuration Definition  
S SORT - DF/SORT Dialogs  
BMR BMR READ - BookManager Read (Read Online Documentation)

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE  
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

Mit P, 3, 4, enter arbeiten wir uns zum Data Set Utility Panel durch

Menu RefList Utilities Help

-----  
Data Set Utility

A Allocate new data set C Catalog data set  
R Rename entire data set U Uncatalog data set  
D Delete entire data set S Data set information (short)  
blank Data set information M Allocate new data set  
V VSAM Utilities

ISPF Library:

Project . . SPRUTH  
Group . . . CICSDB2  
Type . . . . TEST01

Other Partitioned, Sequential or VSAM Data Set:

Data Set Name . . .  
Volume Serial . . . (If not cataloged, required for option "C")

Data Set Password . . (If password protected)

Option ==> A

F1=Help F3=Exit F10=Actions F12=Cancel

Wir legen für den Benutzer SPRUTH vier neue Partitioned Data Sets an, mit den Namen  
SPRUTH.CICSDB2.TEST01  
SPRUTH.SPUFI.IN  
SPRUTH.DBRMLIB.DATA

Die Members von SPRUTH.CICSDB2.TEST01 nehmen die von uns erstellten Programme und JCL Scripts auf.

Wir benötigen außerdem einen Partitioned Data Set SPRUTH.LIB. Hier ist angenehm, daß er in der letzten Sitzung angelegt wurde, und deshalb bereits existiert.

```

Menu  RefList  Utilities  Help
-----
                          Allocate New Data Set
                                          More:  +

Data Set Name . . . : SPRUTH.CICSDB2.TEST01

Management class . . . DEFAULT          (Blank for default management class)
Storage class . . . . PRIM90           (Blank for default storage class)
  Volume serial . . . . SMS001         (Blank for system default volume) **
  Device type . . . . .                (Generic unit or device address) **
Data class . . . . .                  (Blank for default data class)
  Space units . . . . . MEGABYTE       (BLKS, TRKS, CYLS, KB, MB, BYTES
                                         or RECORDS)
Average record unit                    (M, K, or U)
Primary quantity . . 2                 (In above units)
Secondary quantity . . 1               (In above units)
Directory blocks . . 5                 (Zero for sequential data set) *
Record format . . . . FB
Record length . . . . 80
Block size . . . . . 11440
Data set name type : PDS                (LIBRARY, HFS, PDS, or blank) *
                                         (YY/MM/DD, YYYY/MM/DD)

Command ==>
  F1=Help      F3=Exit      F10=Actions  F12=Cancel

```

Wir verwenden die gleichen Partitoned Data Set Definitionen wie bisher. Enter.

```

Menu  RefList  Utilities  Help
-----
                          Data Set Utility

  A Allocate new data set                C Catalog data set
  R Rename entire data set              U Uncatalog data set
  D Delete entire data set              S Data set information (short)
blank Data set information              M Allocate new data set
                                         V VSAM Utilities

ISPF Library:
  Project . . SPRUTH
  Group . . . SPUFI
  Type . . . . IN

Other Partitioned, Sequential or VSAM Data Set:
  Data Set Name . . .
  Volume Serial . . . (If not cataloged, required for option "C")

Data Set Password . . (If password protected)

Option ==>  A
  F1=Help    F3=Exit    F10=Actions  F12=Cancel

```

Die Members von SPRUTH.SPUFI.IN nehmen unsere Datenbankdefinitionen auf.

Enter

```

Menu  RefList  Utilities  Help
-----
                                Data Set Utility                                Data set allocated

    A Allocate new data set                                C Catalog data set
    R Rename entire data set                               U Uncatalog data set
    D Delete entire data set                               S Data set information (short)
blank Data set information                                 M Allocate new data set
                                                            V VSAM Utilities

ISPF Library:
Project . . . SPRUTH
Group . . . . DBRMLIB
Type . . . . . DATA

Other Partitioned, Sequential or VSAM Data Set:
Data Set Name . . .
Volume Serial . . . (If not cataloged, required for option "C")

Data Set Password . . . (If password protected)

Option ==> A
F1=Help      F3=Exit      F10=Actions  F12=Cancel

```

In dem Partioned Data Set SPRUTH.DBRMLIB.DATA werden Zwischenergebnisse abgespeichert.

Enter

```

Menu  RefList  Utilities  Help
-----
                                Data Set Utility

    A Allocate new data set                                C Catalog data set
    R Rename entire data set                               U Uncatalog data set
    D Delete entire data set                               S Data set information (short)
blank Data set information                                 M Allocate new data set
                                                            V VSAM Utilities

ISPF Library:
Project . . . SPRUTH
Group . . . .
Type . . . . .

Other Partitioned, Sequential or VSAM Data Set:
Data Set Name . . . 'SPRUTH.LIB'
Volume Serial . . . (If not cataloged, required for option "C")

Data Set Password . . . (If password protected)

Option ==>
F1=Help      F3=Exit      F10=Actions  F12=Cancel

```

Wie bisher auch, legen wir einen Partitioned Data Set SPRUTH.LIB an, in dem während eines Übersetzungsschrittes Templates für den nächstn Schritt abgelegt werden. Wurde der Partitioned Data Set user.LIB schon vorher angelegt, ist dieser Schritt nicht mehr notwendig.

3 mal F3 Taste kehren wir in das CUSTOMPAC MASTER APPLICATION MENU Panel zurück

```

CUSTOMPAC MASTER APPLICATION MENU
OPTION ====> DB2                                SCROLL ====> PAGE

IS ISMF - Interactive Storage Management Facility
P PDF - ISPF/Program Development Facility
ATC ATC - Application Testing Collection
ART ARTT - Automated Regression Testing Tool
DB2 DB2 - Perform DATABASE 2 interactive functions
QMF QMF - QMF Query Management Facility
C CPSM - CICSplex/SM
M MQ - MQSeries
IP IPCS - Interactive Problem Control Facility
OS SUPPORT - OS/390 ISPF System Support Options
OU USER - OS/390 ISPF User Options
SM SMP/E - SMP/E Dialogs
SD SDSF - System Display and Search Facility
R RACF - Resource Access Control Facility
DI DITTO - Data Interfile Transfer, Testing and Operations
HC HCD - Hardware Configuration Definition
S SORT - DF/SORT Dialogs
BMR BMR READ - BookManager Read (Read Online Documentation)

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Mit der Eingabe "DB2" rufen wir ein ISPF Subsystem auf, welches uns das Anlegen einer neuen DB2 Datenbank ermöglicht.

Enter

```

DB2I PRIMARY OPTION MENU                        SSID:
COMMAND ====> d

Select one of the following DB2 functions and press ENTER.

1 SPUFI (Process SQL statements)
2 DCLGEN (Generate SQL and source language declarations)
3 PROGRAM PREPARATION (Prepare a DB2 application program to run)
4 PRECOMPILE (Invoke DB2 precompiler)
5 BIND/REBIND/FREE (BIND, REBIND, or FREE plans or packages)
6 RUN (RUN an SQL program)
7 DB2 COMMANDS (Issue DB2 commands)
8 UTILITIES (Invoke DB2 utilities)
D DB2I DEFAULTS (Set global parameters)

P DB2 PM (Performance Monitor)
C DC Admin (Data Collector Admin)

X EXIT (Leave DB2I)

F1=HELP F2=SPLIT F3=END F4=RETURN F5=RFIND F6=RCHANGE
F7=UP F8=DOWN F9=SWAP F10=LEFT F11=RIGHT F12=RETRIEVE

```

Es erscheint das DB2I Primary Option Panel. Ehe wir mit dem Anlegen der Datenbank beginnen müssen wir Default Werte setzen. Wir geben " d " ein.

Enter

DB2I DEFAULTS

COMMAND ===>

Change defaults as desired:

```
1 DB2 NAME ..... ===> DBA1      (Subsystem identifier)
2 DB2 CONNECTION RETRIES ===> 0    (How many retries for DB2 connection)
3 APPLICATION LANGUAGE ===> IBMCOB (ASM, C, CPP, COBOL, COB2, IBMCOB,
  FORTRAN, PLI)
4 LINES/PAGE OF LISTING ===> 60    (A number from 5 to 999)
5 MESSAGE LEVEL ..... ===> I      (Information, Warning, Error, Severe)
6 SQL STRING DELIMITER ===> DEFAULT (DEFAULT, ' or ")
7 DECIMAL POINT ..... ===> .      (. or ,)
8 STOP IF RETURN CODE >= ===> 8    (Lowest terminating return code)
9 NUMBER OF ROWS ..... ===> 20     (For ISPF Tables)
10 CHANGE HELP BOOK NAMES? ===> NO  (YES to change HELP data set names)
11 DB2I JOB STATEMENT: (Optional if your site has a SUBMIT exit)
    ===> //SPRUTHA JOB (ACCOUNT), 'NAME'
    ===> //*
    ===> //*
    ===> /*
```

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE  
F7=UP      F8=DOWN      F9=SWAP      F10=LEFT      F11=RIGHT      F12=RETRIEVE

Eine SSID ist eine Datenbankbezeichnung die systemintern benutzt wird. Auf unserer Maschine *jedi.informatik.uni-leipzig.de* ist dies die Bezeichnung "DBA1". Auf einem anderen OS/390 Rechner kann die Bezeichnung anders sein. Bei Bedarf beim Systemadministrator erfragen.

Wir geben die beiden gekennzeichneten Werte ein, und lassen den Rest unverändert.

Enter

COBOL DEFAULTS

COMMAND ===>

Change defaults as desired:

```
1 COBOL STRING DELIMITER ===> DEFAULT (DEFAULT, ' or ")
2 DBCS SYMBOL FOR DCLGEN ===> G      (G/N - Character in PIC clause)
```

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE  
F7=UP      F8=DOWN      F9=SWAP      F10=LEFT      F11=RIGHT      F12=RETRIEVE

Dieses Panel unverändert übernehmen, keine Änderungen, enter

```

DB2I PRIMARY OPTION MENU                SSID: DBA1
COMMAND ==> 1

Select one of the following DB2 functions and press ENTER.

 1 SPUFI                (Process SQL statements)
 2 DCLGEN               (Generate SQL and source language declarations)
 3 PROGRAM PREPARATION  (Prepare a DB2 application program to run)
 4 PRECOMPILE           (Invoke DB2 precompiler)
 5 BIND/REBIND/FREE     (BIND, REBIND, or FREE plans or packages)
 6 RUN                  (RUN an SQL program)
 7 DB2 COMMANDS         (Issue DB2 commands)
 8 UTILITIES            (Invoke DB2 utilities)
 D DB2I DEFAULTS        (Set global parameters)

 P DB2 PM               (Performance Monitor)
 C DC Admin             (Data Collector Admin)

 X EXIT                 (Leave DB2I)

F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP  F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Das DB2I PRIMARY OPTION MENU Panel erscheint wieder. Als SSID ist jetzt DBA1 eingetragen.

Das zum Anlegen einer Datenbank verwendete ISPF Subsystem hört auf den erotischen Namen SPUFI. Wir geben eine "1" ein.

Enter

```

SPUFI                SSID: DBA1
==>

Enter the input data set name:          (Can be sequential or partitioned)
 1 DATA SET NAME ... ==>
 2 VOLUME SERIAL ... ==>              (Enter if not cataloged)
 3 DATA SET PASSWORD ==>             (Enter if password protected)

Enter the output data set name:         (Must be a sequential data set)
 4 DATA SET NAME ... ==>

Specify processing options:
 5 CHANGE DEFAULTS ==> YES             (Y/N - Display SPUFI defaults panel?)
 6 EDIT INPUT ..... ==> YES           (Y/N - Enter SQL statements?)
 7 EXECUTE ..... ==> YES              (Y/N - Execute SQL statements?)
 8 AUTOCOMMIT ..... ==> YES           (Y/N - Commit after successful run?)
 9 BROWSE OUTPUT ... ==> YES          (Y/N - Browse output data set?)

For remote SQL processing:
10 CONNECT LOCATION ==>

F1=HELP   F2=SPLIT   F3=END   F4=RETURN   F5=RFIND   F6=RCHANGE
F7=UP     F8=DOWN    F9=SWAP  F10=LEFT   F11=RIGHT  F12=RETRIEVE

```

Dasr SPUFI Panel erscheint.

Für die Anlage einer Datenbank brauchen wir Definitionen, die eine Aussage über das wie, wo und was machen. Diese Definitionen müssen wir erzeugen. Sie werden in Members des Partitioned Data Sets SPRUTH.SPUFI.IN abgespeichert.

Im Einzelnen müssen wir vier Definitionen erstellen, die in vier Members abgespeichert werden. Diese Definitionen enthalten:

1. Art, Ort (Bereich auf einem von mehreren Plattenspeichern), Größe und Eigenschaften des Speicherplatzes, der unsere Datenbank aufnehmen soll. Dieser Speicherplatz wird als Storage Group (STOGROUP) bezeichnet und erhält einen symbolischen Namen (STOSP1 in dem vorliegenden Beispiel).

2. Bezeichnung (symbolischer Name) der Datenbank selbst. Wir wählen den Namen DBSP1.

Eine Datenbank zwischenspeichert (cached) normalerweise einen Teil der aktiven Daten innerhalb des Hauptspeichers. Dieser Cache wird allgemein als "Bufferpool" bezeichnet und erhält ebenfalls einen symbolischen Namen (BP0 in unserem Beispiel).

3. Eine relationale Datenbank besteht aus mindestens einer, meistens aber aus mehreren Tabellen (Relationen). Für jede Tabelle muß Speicherplatz (als Tablespace bezeichnet) reserviert werden. Dieser erhält ebenfalls einen symbolischen Namen (hier SPASP1). In unserem einfachen Beispiel begnügen wir uns mit einer einzigen Tabelle, benötigen also auch nur einen Tablespace.
4. Schließlich muß die Tabelle selbst bezüglich ihres Namens (TABSP1), ihrer Struktur und der Bezeichnung ihrer Felder (Spalten) definiert werden. Unsere Tabelle soll über 2 Spalten verfügen und das folgende Format haben:

VNAME	NNAME
.....	.....
.....	.....
.....	.....
.....	.....

Wir legen für unseren Partitioned Data Set SPUFI.IN vier Members an. Sie sollen die folgenden Aufgaben übernehmen:

Bezeichnung, symbolischer Name	Aufgabe	definiert in SPUFI.IN Member
STOSP1	Speicherplatz für unsere DB2 Datenbank	STOPRAK
DBSP1	Die Datenbank selbst aufnehmen	DBPRAK
SPASP1	Speicherplatz für eine Tabelle	SPAPRAK
TABSP1	Die Tabelle selbst aufnehmen	TABPRAK

Wir unterscheiden zwischen:

- der Bezeichnung der Datenbank, der Tabelle, und des zugeordneten Speicherplatzes (linke Spalte),
- dem Namen der Member von SPRUTH.SPUFI.IN, die diese Definitionen aufnehmen. (rechte Spalte).

Die Bezeichnungen in der rechten und in der linken Spalte können, müssen aber nicht identisch sein.

```

SPUFI                                SSID: DBA1
===>

Enter the input data set name:        (Can be sequential or partitioned)
 1 DATA SET NAME ... ===> SPUFI.IN(STOPRAK)
 2 VOLUME SERIAL ... ===>           (Enter if not cataloged)
 3 DATA SET PASSWORD ===>          (Enter if password protected)

Enter the output data set name:       (Must be a sequential data set)
 4 DATA SET NAME ... ===> SPUFI.OUT

Specify processing options:
 5 CHANGE DEFAULTS ... ===> YES     (Y/N - Display SPUFI defaults panel?)
 6 EDIT INPUT ..... ===> YES       (Y/N - Enter SQL statements?)
 7 EXECUTE ..... ===> YES           (Y/N - Execute SQL statements?)
 8 AUTOCOMMIT ..... ===> YES       (Y/N - Commit after successful run?)
 9 BROWSE OUTPUT ... ===> YES       (Y/N - Browse output data set?)

For remote SQL processing:
10 CONNECT LOCATION ... ===>

F1=HELP      F2=SPLIT    F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE

```

Wir beginnen mit der Definition des Speicherplatzes (Storage Group) für unsere Datenbank. Diese unsere Eingabe soll in dem Member SPRUTH.SPUFI.IN(STOPRAK) gespeichert werden. Die Eingabe wird von SPUFI übersetzt.

Für die Ausgabe der Übersetzung wird ein neue Data Set benötigt. Wir nennen ihn SPUFI.OUT. Er wird automatisch angelegt.

Enter

```

CURRENT SPUFI DEFAULTS              SSID: DBA1
===>

Enter the following to control your SPUFI session:
 1 ISOLATION LEVEL ... ===> RR      (RR=Repeatable Read, CS=Cursor Stability)
 2 MAX SELECT LINES ... ===> 250    (Maximum number of lines to be
                                     returned from a SELECT)

Output data set characteristics:
 3 RECORD LENGTH ... ===> 4092     (LRECL=Logical record length)
 4 BLOCK SIZE ..... ===> 4096     (Size of one block)
 5 RECORD FORMAT ... ===> VB       (RECFM=F, FB, FBA, V, VB, or VBA)
 6 DEVICE TYPE ..... ===> SYSDA    (Must be DASD unit name)

Output format characteristics:
 7 MAX NUMERIC FIELD ... ===> 33    (Maximum width for numeric fields)
 8 MAX CHAR FIELD .. ===> 80       (Maximum width for character fields)
 9 COLUMN HEADING .. ===> NAMES    (NAMES, LABELS, ANY or BOTH)

F1=HELP      F2=SPLIT    F3=END      F4=RETURN    F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP    F10=LEFT    F11=RIGHT   F12=RETRIEVE

```

In diesem Screen werden Data Set Parameter angezeigt. Wir übernehmen hier (und auch in Zukunft für alle weiteren SPUFI Definitionen) alle Default Werte ohne Änderung.

Enter



```

SPUFI                      SSID: DBA1
===>
DSNE808A EDIT SESSION HAS COMPLETED. PRESS ENTER TO CONTINUE
Enter the input data set name:      (Can be sequential or partitioned)
 1 DATA SET NAME ... ===> SPUFI.IN(STOPRAK)
 2 VOLUME SERIAL ... ===>          (Enter if not cataloged)
 3 DATA SET PASSWORD ===>        (Enter if password protected)

Enter the output data set name:     (Must be a sequential data set)
 4 DATA SET NAME ... ===> SPUFI.OUT

Specify processing options:
 5 CHANGE DEFAULTS  ===> *        (Y/N - Display SPUFI defaults panel?)
 6 EDIT INPUT ..... ===> *        (Y/N - Enter SQL statements?)
 7 EXECUTE .....   ===> YES       (Y/N - Execute SQL statements?)
 8 AUTOCOMMIT ..... ===> YES      (Y/N - Commit after successful run?)
 9 BROWSE OUTPUT ... ===> YES     (Y/N - Browse output data set?)

For remote SQL processing:
10 CONNECT LOCATION ===>

F1=HELP      F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP        F8=DOWN     F9=SWAP    F10=LEFT   F11=RIGHT   F12=RETRIEVE

```

SPUFI teilt uns mit, daß damit die Edit Session - Erstellung des Members SPUFI.IN(STOPRAK) - beendet wurde. Unsere Eingabe wird übersetzt, das Ergebnis in SPUFI.OUT gestellt, und -

Enter

```

Menu Utilities Compilers Help
-----
BROWSE      SPRUTH.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
CREATE STOGROUP STOSP1                                00010000
  VOLUMES (SCPMV5)                                   00020000
  VCAT DSN510;                                       00030000
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE617I COMMIT PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS 1
DSNE621I NUMBER OF INPUT RECORDS READ IS 3
DSNE622I NUMBER OF OUTPUT RECORDS WRITTEN IS 15
***** Bottom of Data *****

Command ===>
F1=Help      F3=Exit      F5=Rfind    F12=Cancel

                                           Scroll ===> PAGE

```

- das Ergebnis der Übersetzung mitgeteilt. Offensichtlich ist alles gut gelaufen. Soweit ok.

F3 Taste

```

SPUFI                                SSID: DBA1
===>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:      (Can be sequential or partitioned)
 1 DATA SET NAME ... ===> SPUFI.IN(STOPRAK)
 2 VOLUME SERIAL ... ===>          (Enter if not cataloged)
 3 DATA SET PASSWORD ===>         (Enter if password protected)

Enter the output data set name:     (Must be a sequential data set)
 4 DATA SET NAME ... ===> SPUFI.OUT

Specify processing options:
 5 CHANGE DEFAULTS  ===> YES      (Y/N - Display SPUFI defaults panel?)
 6 EDIT INPUT ..... ===> YES      (Y/N - Enter SQL statements?)
 7 EXECUTE .....   ===> YES      (Y/N - Execute SQL statements?)
 8 AUTOCOMMIT ..... ===> YES      (Y/N - Commit after successful run?)
 9 BROWSE OUTPUT ... ===> YES      (Y/N - Browse output data set?)

For remote SQL processing:
10 CONNECT LOCATION ===>

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Wir kehren zum SPUFI Panel zurück. Die von uns bisher vorgenommene Reservierung von Speicherplatz für die Datenbank wird normalerweise vom Systemadministrator vorgenommen; dem einzelnen Benutzer fehlen hierfür in der Regel die Zugriffsrechte. Die weiteren Schritte kann der Benutzer aber selbst vornehmen. Als nächstes wird für den bereitgestellten Speicherplatz STOPRAK eine Datenbank definiert.

```

SPUFI                                SSID: DBA1
===>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:      (Can be sequential or partitioned)
 1 DATA SET NAME ... ===> SPUFI.IN(DBPRAK)
 2 VOLUME SERIAL ... ===>          (Enter if not cataloged)
 3 DATA SET PASSWORD ===>         (Enter if password protected)

Enter the output data set name:     (Must be a sequential data set)
 4 DATA SET NAME ... ===> SPUFI.OUT

Specify processing options:
 5 CHANGE DEFAULTS  ===> YES      (Y/N - Display SPUFI defaults panel?)
 6 EDIT INPUT ..... ===> YES      (Y/N - Enter SQL statements?)
 7 EXECUTE .....   ===> YES      (Y/N - Execute SQL statements?)
 8 AUTOCOMMIT ..... ===> YES      (Y/N - Commit after successful run?)
 9 BROWSE OUTPUT ... ===> YES      (Y/N - Browse output data set?)

For remote SQL processing:
10 CONNECT LOCATION ===>

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Wir speichern die Definition für die Datenbank in einem Member mit den Namen SPRUTH.SPUFI.IN(DBPRAK) ab.

Enter Taste. Der CURRENT SPUFI DEFAULTS Bildschirm erscheint wieder. Wir übernehmen alle Werte unverändert. Nochmals Enter.





```

Menu Utilities Compilers Help
-----
BROWSE      SPRUTH.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
CREATE TABLESPACE SPASP1                          00010000
  IN DBSP1                                          00020000
  USING STOGROUP STOSP1                            00030000
    PRIQTY 20                                       00040000
    SECQTY 20                                       00050000
    ERASE NO                                        00060000
  BUFFERPOOL BP0                                   00070000
  CLOSE NO;                                         00080000
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE617I COMMIT PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS 1
Command ==>                                         Scroll ==> PAGE
  F1=Help    F3=Exit    F5=Rfind    F12=Cancel

```

Ebenfalls erfolgreich (sind wir nicht gut?). Durch Drücken der F8 Taste (Scroll Forward) können wir den Rest der Nachricht ansehen. Mit der F7 Taste (Scroll Backward) geht es wieder zurück.

Mit der F3 Taste verlassen wir das Panel.

```

SPUFI                      SSID: DBA1
==>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:          (Can be sequential or partitioned)
 1 DATA SET NAME ... ==> SPUFI.IN(TABPRAK)
 2 VOLUME SERIAL ... ==>                (Enter if not cataloged)
 3 DATA SET PASSWORD ==>                (Enter if password protected)

Enter the output data set name:         (Must be a sequential data set)
 4 DATA SET NAME ... ==> SPUFI.OUT

Specify processing options:
 5 CHANGE DEFAULTS ==> YES              (Y/N - Display SPUFI defaults panel?)
 6 EDIT INPUT ..... ==> YES             (Y/N - Enter SQL statements?)
 7 EXECUTE ..... ==> YES                (Y/N - Execute SQL statements?)
 8 AUTOCOMMIT ..... ==> YES             (Y/N - Commit after successful run?)
 9 BROWSE OUTPUT ... ==> YES            (Y/N - Browse output data set?)

For remote SQL processing:
10 CONNECT LOCATION ==>

F1=HELP    F2=SPLIT    F3=END      F4=RETURN   F5=RFIND    F6=RCHANGE
F7=UP      F8=DOWN    F9=SWAP    F10=LEFT   F11=RIGHT   F12=RETRIEVE

```

Wir sind wieder im SPUFI Screen. In den Member TABPRAK speichern wir die Definition der Tabelle selbst.

Enter, und nochmals Enter



```

SPUFI                                SSID: DBA1
===>

Enter the input data set name:        (Can be sequential or partitioned)
 1 DATA SET NAME ... ===> SPUFI.IN(ININSERT)
 2 VOLUME SERIAL ... ===>          (Enter if not cataloged)
 3 DATA SET PASSWORD ===>         (Enter if password protected)

Enter the output data set name:       (Must be a sequential data set)
 4 DATA SET NAME ... ===> SPUFI.OUT

Specify processing options:
 5 CHANGE DEFAULTS  ===> YES        (Y/N - Display SPUFI defaults panel?)
 6 EDIT INPUT ..... ===> YES        (Y/N - Enter SQL statements?)
 7 EXECUTE .....   ===> YES        (Y/N - Execute SQL statements?)
 8 AUTOCOMMIT ..... ===> YES        (Y/N - Commit after successful run?)
 9 BROWSE OUTPUT ... ===> YES        (Y/N - Browse output data set?)

For remote SQL processing:
10 CONNECT LOCATION ===>

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE

```

Damit haben wir eine Datenbank angelegt. Aber sie ist zur Zeit noch leer. Es existieren zahlreiche Software Funktionen, um leere Datenbanken mit bereits existierenden Daten zu füllen. Wir machen es hier auf eine einfache Art zu Fuß.

Hierzu legen wir (zusätzlich zu den bisherigen 4 Members) einen weiteren SPUFI Member INSERT an.

Enter, und dann nochmals Enter

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          SPRUTH.SPUFI.IN(ININSERT) - 01.00          Columns 00001 00072
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000100 INSERT INTO SPRUTH.TABSP1
000200     VALUES ('HEINO', 'BAUER');
000300 INSERT INTO SPRUTH.TABSP1
000400     VALUES ('BORIS', 'FAERBER');
000500 INSERT INTO SPRUTH.TABSP1
000600     VALUES ('SEBASTIAN', 'RICHTER');
000700 INSERT INTO SPRUTH.TABSP1
000800     VALUES ('FRITZ', 'SCHULTE');
***** Bottom of Data *****

Command ===>
F1=Help      F3=Exit      F5=Rfind     F6=Rchange   F12=Cancel

```

Wir geben in die SQL Tabelle TABSP1 Vornamen und Nachnamen für 4 Personen ein.

F3 Taste, und dann Enter

```

Menu Utilities Compilers Help
-----
BROWSE      SPRUTH.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO SPRUTH.TABSP1                          00010000
VALUES ('HEINO', 'BAUER');                          00020002
-----+-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO SPRUTH.TABSP1                          00030000
VALUES ('BORIS', 'FAERBER');                        00040002
-----+-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO SPRUTH.TABSP1                          00050000
VALUES ('SEBASTIAN', 'RICHTER');                    00060002
-----+-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
Command ==>                                         Scroll ==> PAGE
F1=Help      F3=Exit      F5=Rfind    F12=Cancel

```

Die Eingabe war erfolgreich. Mit der F8 Taste sehen wir uns den Rest der Ausgabe an.

```

Menu Utilities Compilers Help
-----
BROWSE      SPRUTH.SPUFI.OUT                      Line 00000019 Col 001 080
-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO SPRUTH.TABSP1                          00070000
VALUES ('FRITZ', 'SCHULTE');                        00080001
-----+-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+
DSNE617I COMMIT PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+
DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS 4
DSNE621I NUMBER OF INPUT RECORDS READ IS 8
DSNE622I NUMBER OF OUTPUT RECORDS WRITTEN IS 33
***** Bottom of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+
Command ==>                                         Scroll ==> PAGE
F1=Help      F3=Exit      F5=Rfind    F12=Cancel

```

OK, wir haben unsere Datenbank mit Daten gefüttert.

Mit der F3 Taste SPUFI erneut aufrufen.



```

Menu  Utilities  Compilers  Help
-----
BROWSE      SPRUTH.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM SPRUTH.TABSP1;                      00010000
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
VNAME              NNAME
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
HEINO              BAUER
BORIS              FAERBER
SEBASTIAN          RICHTER
FRITZ              SCHULTE
DSNE610I NUMBER OF ROWS DISPLAYED IS 4
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE617I COMMIT PERFORMED, SQLCODE IS 0
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE601I SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
DSNE620I NUMBER OF SQL STATEMENTS PROCESSED IS 1
Command ==>                                         Scroll ==> PAGE
F1=Help   F3=Exit   F5=Rfind  F12=Cancel

```

Die Abfrage war erfolgreich.

SPUFI gibt den Inhalt der Datenbank wieder; die Datenbank ist da. Die mit dem INSERT Statement eingegebenen vier Vornamen und Nachnamen sind in der Datenbank gespeichert.

Wir können jetzt das Spiel wiederholen, indem wir den SPUFI Screen mit INSERT erneut aufrufen, und weitere Namen eingeben. Sie werden an den vorhandenen Datenbestand angehängt. Machen Sie es doch mal als Exercise:

F3 um den SPUFI Screen aufzurufen, INSERT eingeben, 2 x Enter um zum Edit Screen zu gelangen, neue Daten eingeben, mit F3 und Enter wieder zum Browse Screen.

---

Das war es, Sie sind auf dem besten Weg, ein toller OS/390 Experte zu werden.

Wir haben erfolgreich eine DB2 Datenbank angelegt und mit Daten bevölkert. Bitte bedenken Sie immer, daß wir hier Prinzipien erklären, und daß man in der Praxis mit anderen und weitgehend automatisierten Prozessen arbeiten würde.

Wichtig ist auch, daß wir es hier mit einem Großrechner zu tun haben, der an die unterschiedlichsten Bedingungen angepaßt werden kann. Das Anlegen einer DB2 Datenbank z.B. unter dem Windows NT oder Linux Betriebssystem ist unter anderem schon deshalb einfacher, weil es weniger Optionen gibt.

Als nächsten Schritt werden wir mit Hilfe einer CICS Transaktion auf unsere Datenbank zugreifen.

Wir verlassen OS/390 mit F3, F3, F3, F3, 3, logoff. Ende der Session.