

Tutorial 4

DB2

Copyright © Institut für Informatik, Universität Leipzig

ph v/2010/03

Ziel dieses Tutorials ist es, eine z/OS - DB2 relationale Datenbank zu erzeugen und auszulesen. In dem hier beschriebenen Schritt wird von uns die Datenbank mit einer einzigen einfachen Tabelle angelegt und anschließend mit einigen wenigen Daten gefüllt.

Aufgabe: Beschäftigen Sie sich mit diesem Tutorial und lösen Sie exakt und gewissenhaft die kursiv geschriebenen und umrahmten Aufgaben.

Anlegen benötigter Datasets

Wir loggen uns als TSO-Benutzer ein und wählen aus dem "CUSTOMPAC MASTER APPLICATION MENU" die Option "P" aus. Anschließend gehen wir mittels "3.2" zum "Data Set Utility" (s. Abbildung 1) .

```

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 . . . PRAKT20
Group . . . CICSDDB2
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

```

Abbildung 1: "Data Set Utility"

Wir legen für den Benutzer "PRAKT20" drei neue **Partitioned DataSets** (PDS) an:

PRAKT20.CICSDDB2.TEST01 (s. Abbildung 1)
PRAKT20.SPUFI.IN
PRAKT20.DBRMLIB.DATA

Wir verwenden dazu die in der Abbildung 2 angegebenen Parameter.

Die Members von "PRAKT20.CICSDB2.TEST01" nehmen das von uns zu erstellende C-Programm, den das dazugehörige BMS-Programm und das JCL-Scripte auf. Der Dataset wird im später zu absolvierenden Tutorial 5 (C-Version) benötigt.

```

Menu  RefList  Utilities  Help
-----
                          Allocate New Data Set                      More:      +
Data Set Name      . . . : PRAKT20.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       . . . KILOBYTE     (BLKS, TRKS, CYLS, KB, MB, BYTES
or RECORDS)
Average record unit
Primary quantity  . . 16             (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        . . . 320
Data set name type : PDS             (LIBRARY, HFS, PDS, or blank) *
(Y Y/MM/DD, Y YYY/MM/DD)
Command ==>
F1=Help      F3=Exit      F10=Actios  F12=Cancel

```

Abbildung 2: Die Parameter

Die Members von "PRAKT20.SPUFLIN" nehmen DB2-Kommandos auf, deren Ausführung unsere Datenbank irgendwie modifizieren wird. Wir werden DB2-Kommandos zum Anlegen von Datenbanken, Tabellen, Einfügen von Daten etc. kennenlernen.

In dem Partitioned Dataset "PRAKT20.DBRMLIB.DATA" werden Zwischenergebnisse abgespeichert, welche in Precompile-, Compile- sowie Link-Prozessen anfallen werden.

Für die spätere Bearbeitung der Tutorien 5 benötigen wir außerdem noch einen Partitioned Dataset "PRAKT20.LIB". Hier wird angenommen, dass er in der letzten Sitzung angelegt wurde und deshalb bereits existiert. Sollte dies nicht der Fall sein, legen wir diesen unter Nutzung der Parameter laut Abbildung 2 an (s. auch Abbildung 3).

```

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 . . .
Group . . .
Type . . .

Other Partitioned, Sequential or VSAM Data Set:
Data Set Name . . . 'PRAKT20.LIB'
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

```

Abbildung 3: Anlegen des Datasets "PRAKT20.LIB"

Der Dataset "PRAKT20.LIB" nimmt soll wie der Dataset "PRAKT20.DBRMLIB.DATA" in den Tutorien 5 Zwischenergebnisse von Precompile-, Compile- und Link-Schritten aufnehmen.

Aufgabe: Erstellen Sie die benötigten Datasets. Verwenden Sie dazu die in Abbildung 2 angegebenen Parameter.

Wir betätigen dreimal die F3-Taste und kehren so in das "CUSTOMPAC MASTER APPLICATION MENU"-Panel zurück.

Einloggen ins z/OS DB2

Mit der Eingabe "DB2" rufen wir das z/OS-DB2-Subsystem auf, das uns das Anlegen einer neuen DB2-Datenbank sowie deren Modifikation ermöglicht. Anschließend drücken wir die Eingabetaste.

```

                                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 OM               (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

```

Abbildung 4: das "DB2I PRIMARY OPTION MENU"

Es erscheint das "DB2I PRIMARY OPTION MENU"-Panel (s. Abbildung 4).

Einstellen des SubSystem IDentifiers (SSIDs)

Ehe wir mit dem Anlegen der Datenbank beginnen, müssen wir die "SubSystem IDentifier" (SSID) setzen. Dies ist nur ein einziges Mal erforderlich, und zwar beim erstmaligen Gebrauch des DB2-Subsystems. Loggt man sich wiederholt in das DB2-Subsystem ein, ist dieses Einstellen der "SubSystem IDentifier" nicht mehr erforderlich.

Wir geben "d" ein und bestätigen anschließend mit der Eingabetaste.

```

                                DB2I DEFAULTS
COMMAND ==>>

Change defaults as desired:

 1  DB2 NAME ..... ==>> D931      (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)
    ==>> //PRAKT20 JOB (ACCOUNT),'NAME'lector Admin)
    ==>> /**
    ==>> /**
    ==>> /**

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

```

Abbildung 5: Datenbankbezeichnung

Eine "SubSystem IDentifier" ist eine Datenbankbezeichnung, die systemintern benutzt wird. Auf unserer Maschine "binks.informatik.uni-leipzig.de" ist dies die Bezeichnung "D931". Auf einem anderen z/OS-Rechner kann die Bezeichnung anders sein.

Wir geben die beiden gekennzeichneten Werte ein, lassen den Rest unverändert und bestätigen mit der Eingabetaste (s. Abbildung 5).

```

                                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

```

Abbildung 6: COBOL DEFAULTS

Das Panel in der Abbildung 6 bedarf keiner Änderungen, wir drücken lediglich die Eingabetaste.

```

                                DB2I PRIMARY OPTION MENU                                SSID: D931
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 OM                                (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

```

Abbildung 7: "DB2I PRIMARY OPTION MENU"

Das "DB2I PRIMARY OPTION MENU"-Panel erscheint wieder. Als SSID ist jetzt "D931" eingetragen (s. Abbildung 7).

Starten von SPUFI

Man verwendet zum Anlegen einer Datenbank das DB2-Subsystem "SPUFI". Wir geben eine "1" ein und betätigen die Eingabetaste.

```

                                     SPUFI                               SSID: D931
====>

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

```

Abbildung 8: Der SPUFI-Panel

Das SPUFI-Panel erscheint (s. Abbildung 8).

Überblick über die vier vorzunehmenden Definitionen

Für die Anlage einer Datenbank brauchen wir Definitionen, die eine Aussage über das "was", "wie" und "wo" machen. Diese Definitionen müssen wir erzeugen. Sie werden in Members des Partioned Datasets "PRAKT20.SPUFI.IN" abgespeichert.

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

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 ("STOGR020" in dem vorliegenden Beispiel), siehe Abbildung 12.

Für den symbolischen Namen der Datenbank wählen wir den Namen "DB020". Eine Datenbank speichert (cached) normalerweise einen Teil der aktiven Daten innerhalb des Hauptspeichers temporär ab. Dieser Cache wird allgemein als "Bufferpool" bezeichnet und erhält ebenfalls einen symbolischen Namen ("BP0" in unserem Beispiel), siehe Abbildung 17.

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 "TABSP020"). In

unserem einfachen Beispiel legen wir nur eine einzige Tabelle an, benötigen also auch nur einen Tablespace (siehe Abbildung 20).

Schließlich muß die Tabelle selbst bezüglich ihres Namens ("TAB020"), ihrer Struktur und der Bezeichnung ihrer Felder (Spalten) definiert werden (siehe Abbildung 23). Unsere Tabelle soll über 2 Spalten verfügen und das folgende Format haben:

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

Zusammengefasst:

Wir legen für unseren Partitioned Dataset "PRAKT20.SPUFI.IN" vier Members an:

Member-Name	Aufgabe	symbolischer Name
STOGR1	Speicherplatz für unsere DB2-Datenbank anlegen	STOGR020
DB1	Die Datenbank selbst anlegen	DB020
TABSP1	Speicherplatz für eine Tabelle anlegen	TABSP020
TAB1	Die Tabelle selbst anlegen	TAB020

Wir unterscheiden zwischen

- den symbolischen Namen der Datenbank, der Tabelle und des jeweiligen zugeordneten Speicherplatzes (rechte Seite obiger Tabelle),
- den Namen der Member von "PRAKT20.SPUFI.IN", die diese Definitionen aufnehmen (linke Seite obiger Tabelle).

Symbolischer Name und Membername können, müssen aber nicht identisch sein.

Definition des Speicherplatzes für Datenbanken

Wir beginnen mit der Definition des Speicherplatzes (Storage Group) für unsere Datenbank.

```

====>                                SPUFI                                SSID: D931

Enter the input data set name:          (Can be sequential or partitioned)
1  DATA SET NAME ... ====>  SPUFI.IN(STOGR1)
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

```

Abbildung 9: Definition für die Storage Group

Unsere Eingabe soll in dem Member "PRAKT20.SPUFI.IN(STOGR1)" gespeichert werden. Diese Eingabe wird von SPUFI übersetzt.

Für die Ausgabe der Übersetzung wird ein neuer Dataset benötigt. Wir nennen ihn PRAKT20.SPUFI.OUT. Er wird automatisch angelegt.

Nachdem wir die in Abbildung 9 gekennzeichneten Änderungen vorgenommen haben, drücken wir die Eingabetaste.

```

====>                                CURRENT SPUFI DEFAULTS                    SSID: D931

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

```

Abbildung 10: Dataset-Parameter

In diesem Screen (Abbildung 10) werden Dataset-Parameter angezeigt. Wir übernehmen hier (und auch in Zukunft für alle weiteren SPUFI Definitionen) alle Default-Werte ohne Änderung, indem wir mit der Eingabetaste bestätigen.

Es erscheint ein (leerer) Edit-Screen (s. Abbildung 11):

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.SPUFI.IN(STOGR1) - 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.
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
|||||
Command ===>          Scroll ===> PAGE
F1=Help      F3=Exit      F5=Rfind    F6=Rchange  F12=Cancel

```

Abbildung 11: der Edit-Screen

SPUFI teilt uns mit, dass damit die Edit Session – Erstellen des Members SPUFI.IN(STOGR1) – beendet wurde (s. Abbildung 13). Nach dem Drücken der Eingabetaste wird unsere Definition übersetzt und das Ergebnis in SPUFI.OUT gestellt.

```

Menu  Utilities  Compilers  Help
-----
BROWSE      PRAKT20.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
CREATE STOGROUP STOGR020                                00010000
  VOLUMES ('**')                                       00020000
  VCAT PRAKT20;                                         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

```

Abbildung 14: Erfolgreiche Übersetzung

Das Ergebnis der Übersetzung wird mitgeteilt (s. Abbildung 14).

Löschen von alten Objekten, um neue anlegen zu können

Wird an dieser Stelle eine Fehlermeldung ausgegeben, dann ist der zu erstellende Speicherplatz für unsere Datenbank (Stogroup) möglicherweise bereits vorhanden (s. Abbildung 14a). Er wurde zu einem früheren Zeitpunkt von jemand anderem schon einmal angelegt. In diesem Fall ist der alte Speicherplatz erst einmal zu entfernen. Dies ist mit dem folgenden SQL-Statement möglich:

```
DROP STOGROUP STOGR020
```

Die Stogroup wird aber nur gelöscht, wenn sich in ihr keine Speicherplatzreservierung für eine Tabelle befindet (Tablespace). Befindet sich eine solche in der Stogroup, wird die in Abbildung 14b dargestellte Fehlermeldung ausgegeben sowie die Stogroup nicht gelöscht. Die Fehlermeldung gibt aber den Namen des Objektes aus, welches vor der Stogroup noch gelöscht werden muß, in diesem Beispiel der Tablespace "DB020.TABSP020". Dies geschieht durch das SQL-Statement

```
DROP TABLESPACE DB020.TABSP020
```

Beide SQL-Statements lassen sich wie folgt ausführen: Zuerst sind im SPUFI-Panel (s. Abbildung 9 die grün umrahmten Werte einzutragen; anstelle von "STOGR1" wählt man einen in dem Dataset "SPUFI.IN" noch nicht benutzten Member-Namen, z.B. "DELTABSP" (DElete TABLE Space). Dieses Member nimmt das auszuführende SQL-Statement z.B. "DROP TABLESPACE DB020.TABSP020" auf.

Aufgabe: Definieren Sie den Speicherplatz für die Datenbank. Bezeichnen Sie diesen mit dem Namen "STOGRxxx" (= Storage Group "xxx"), wobei Sie "xxx" durch Ihre dreistellige Nummer Ihres PRAKT- oder PRAK-Accounts ersetzen sollen. Erstellen Sie einen Screenshot entsprechend der Abbildung 14. Dieser muss natürlich demonstrieren, dass Sie Ihre Storage Group FEHLERFREI anlegen konnten. Dieser Screenshot und der später zu erstellende Screenshot sollten im JPEG-Format erstellt werden (< 90 Kbyte). Im Ausnahmefall ist auch das Bitmap-Format (< 250 Kbyte) erlaubt. Alle Screenshots müssen für Ihren Betreuer gut lesbar sein (entsprechend zweckmäßige Farben und Größen benutzen).

Hinweis: Falls Sie die Storage Group nicht anlegen können, weil sich schon eine entsprechende Storage Group auf dem Mainframe befindet, dürfen und müssen Sie die alte Storage Group löschen. Wie man das tun kann, ist auf Seite 12 dieses Tutorials beschrieben.

```

Menu Utilities Compilers Help
-----
BROWSE   PRAKT20.SPUFI.OUT                               Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
CREATE STOGROUP STOGR020                                00010000
  VOLUMES ('*')                                         00020000
  VCAT PRAKT20;                                         00040000
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNT408I  SQLCODE = -601, ERROR:  THE NAME OF THE OBJECT TO BE CREATED OR THE
          TARGET OF A RENAME STATEMENT IS IDENTICAL TO THE EXISTING NAME STOGR020
          OF THE OBJECT TYPE STOGROUP
DSNT418I  SQLSTATE   = 42710 SQLSTATE RETURN CODE
DSNT415I  SQLERRP    = DSNXICSG SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD    = 10 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD    = X'0000000A' X'00000000' X'00000000' X'FFFFFFFF'
          X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE618I  ROLLBACK PERFORMED, SQLCODE IS 0
DSNE616I  STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE601I  SQL STATEMENTS ASSUMED TO BE BETWEEN COLUMNS 1 AND 72
Command ==>
          F1=Help   F3=Exit   F5=Rfind  F12=Cancel          Scroll ==> PAGE

```

Abbildung 14a: Fehlermeldung, dass Objekt bereits existiert

```

Menu Utilities Compilers Help
-----
BROWSE   PRAKT20.SPUFI.OUT                               Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DROP STOGROUP STOGR020                                00010000
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNT408I  SQLCODE = -616, ERROR:  STOGROUP STOGR020 CANNOT BE DROPPED BECAUSE IT
          IS REFERENCED BY TABLESPACE DB020.TABSP020
DSNT418I  SQLSTATE   = 42893 SQLSTATE RETURN CODE
DSNT415I  SQLERRP    = DSNXIDSG SQL PROCEDURE DETECTING ERROR
DSNT416I  SQLERRD    = 60 0 0 -1 0 0 SQL DIAGNOSTIC INFORMATION
DSNT416I  SQLERRD    = X'0000003C' X'00000000' X'00000000' X'FFFFFFFF'
          X'00000000' X'00000000' SQL DIAGNOSTIC INFORMATION
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE618I  ROLLBACK 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 1
DSNE622I  NUMBER OF OUTPUT RECORDS WRITTEN IS 18
Command ==>
          F1=Help   F3=Exit   F5=Rfind  F12=Cancel          Scroll ==> PAGE

```

Abbildung 14b: Fehlermeldung, dass Stogroup nicht gelöscht werden kann

Anlegen einer Datenbank

```

                                SPUFI                                SSID: D931
====>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:      (Can be sequential or partitioned)
 1 DATA SET NAME ... ==> SPUFI.IN(STOGR1)
 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

```

Abbildung 15: Der SPUFI-Screen

Wir kehren zum SPUFI-Bildschirm (s. Abbildung 15) 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 im Speicherplatz "STOGR020" eine Datenbank angelegt.

```

                                SPUFI                                SSID: D931
====>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:      (Can be sequential or partitioned)
 1 DATA SET NAME ... ==> SPUFI.IN(DB1)
 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

```

Abbildung 16: Definition für DB1

Auch diese Anlage der Datenbank war erfolgreich (s. Abbildung 18). Mit der F3-Taste kehren wir zum SPUFI-Screen zurück.

Aufgabe: Definieren Sie Ihre Datenbank. Diese soll den Namen "DBxxx" (=Database "xxx") erhalten, wobei "xxx" für ihre dreistellige PRAKT- oder PRAK-Account-Nummer steht.

```

                                SPUFI                                SSID: D931
====>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:      (Can be sequential or partitioned)
 1  DATA SET NAME ... ==> SPUFI.IN(TABSP1)
 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

```

Abbildung 19: Member TABSP1 zum Editieren öffnen

In unserem Beispiel soll die DB2-Datenbank aus einer einzigen Tabelle bestehen. Wir erstellen eine Definition des Speicherplatzes (Tablespace) für unsere Tabelle in dem Member "TABSP1" (s. Abbildung 19).

Wir drücken zweimal die Eingabetaste, um in den "Edit Entry Panel" zu gelangen.

Defininition von Tablespace für DB2-Tabellen

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.SPUFI.IN(TABSP1) - 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.
'''''' CREATE TABLESPACE TABSP020
''''''      IN DB020
''''''      USING STOGROUP STOGR020
''''''          PRIQTY 20
''''''          SECQTY 20
''''''          ERASE NO
''''''      BUFFERPOOL BP0
''''''      CLOSE NO;
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
Command ==>
F1=Help      F3=Exit      F5=Rfind     F6=Rchange   F12=Cancel
Scroll ==> PAGE
    
```

Abbildung 20: Speicherplatz für Tabelle anlegen

Wir erstellen einen Tablespace mit dem symbolischen Namen "TABSP020" (s. Abbildung 20). Die darin enthaltene Tabelle gehört zur Datenbank "DB020", die wiederum in der Storage Group "STOGR020" untergebracht ist. Der hierzu gehörige Buffer Pool (Hauptspeicher Cache) heißt "BP0". Wir verlassen den Bildschirm mit F3 und bestätigen mit der Eingabetaste.

```

Menu Utilities Compilers Help
-----
BROWSE        PRAKT20.SPUFI.OUT          Line 00000000 Col 001 080
*****      ***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+
CREATE TABLESPACE TABSP020                00010000
      IN DB020                              00020000
      USING STOGROUP STOGR020              00030000
          PRIQTY 20                        00040000
          SECQTY 20                        00050000
          ERASE NO                          00060000
      BUFFERPOOL BP0                       00070000
      CLOSE NO;                            00080000
-----+-----+-----+-----+-----+-----+-----+
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+
                                                                00090000
-----+-----+-----+-----+-----+-----+-----+
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
Command ==>
F1=Help      F3=Exit      F5=Rfind     F12=Cancel
Scroll ==> PAGE
    
```

Abbildung 21: Erfolgreiche Anlage des Tablespace

Die Anlage des Tablespaces "TABSP020" war ebenfalls erfolgreich. 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 den Screen.

Aufgabe: Erstellen Sie den Tablespace für Ihre Tabelle. Dieser soll den Namen "TABSPxxx" erhalten, wobei "xxx" für Ihre dreistellige Nummer Ihres PRAK- oder PRAKT-Accounts steht.

Erstellen der Tabelle

```

                                SPUFI                                SSID: D931
====>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:      (Can be sequential or partitioned)
 1  DATA SET NAME ... ==> SPUFI.IN(TAB1)
 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

```

Abbildung 22: Definition für TAB1

Wir sind wieder im SPUFI-Panel. In den Member "TAB1" speichern wir die Definition der Tabelle selbst (s. Abbildung 22).

Um weiter zu kommen, drücken wir zweimal die Eingabetaste.

Aufgabe: Definieren Sie Ihre Tabelle. Diese soll den Namen "TABxxx" erhalten, wobei "xxx" für Ihre dreistellige Nummer Ihres PRAKT- oder PRAK-Accounts steht.

Datensätze in die Tabelle einfügen

```

====>                                SPUFI                                SSID: D931

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

```

Abbildung 25: Definition für INSERT

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 (s. Abbildung 25) und drücken zweimal die Eingabetaste.


```

Menu Utilities Compilers Help
-----
BROWSE      PRAKT20.SPUFI.OUT                               Line 00000019 Col 001 080
-----+-----+-----+-----+-----+-----+-----+-----+
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 3
DSNE621I NUMBER OF INPUT RECORDS READ IS 6
DSNE622I NUMBER OF OUTPUT RECORDS WRITTEN IS 27
***** Bottom of Data *****
-----

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

Scroll ==> PAGE

```

Abbildung 28: Die restliche Ausgabe

Wir haben unsere Datenbank mit Daten beschrieben.

Mit der F3-Taste rufen wir den SPUFI-Panel erneut auf.

***Aufgabe:** Füllen Sie Ihre Tabelle mit einigen Namen einschließlich Ihres eigenen Namens. Sollten Sie die Übung zu zweit absolvieren (unter Nutzung eines gemeinsamen PRAK- oder PRAKT-Accounts), dann müssen Ihre beiden Namen in die Tabelle aufgenommen werden.*

Ansehen sämtlicher Datensätze der Tabelle

```

SPUFI                      SSID: D931
==>
DSNE800A NO DEFAULT VALUES WERE CHANGED. PRESS ENTER TO CONTINUE
Enter the input data set name: (Can be sequential or partitioned)
 1 DATA SET NAME ... ==> SPUFI.IN(SELECT)
 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 ..... ==> 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

```

Abbildung 29: Definition für SELECT

Frage: Befinden sich nun die korrekten Namen in der Datenbank? Mit SPUIFI können wir uns auch den Datenbank-Inhalt ansehen. Wir erstellen noch einen weiteren SPUIFI-Member "SELECT" (s. Abbildung 29) und drücken zweimal die Eingabetaste.

```
File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.SPUIFI.IN(SELECT) - 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.
'''''' SELECT * FROM PRAKT20.TAB020;
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
Command ==>
F1=Help      F3=Exit      F5=Rfind      F6=Rchange    F12=Cancel    Scroll ==> PAGE
```

Abbildung 30: SELECT

Wir erstellen eine Abfrage "SELECT" für unsere Datenbank (s. Abbildung 30), verlassen den Screen mit F3 und betätigen anschließend die Eingabetaste.

```
Menu Utilities Compilers Help
-----
BROWSE        PRAKT20.SPUIFI.OUT                          Line 00000000 Col 001 080
*****      ***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM PRAKT20.TAB020;                               00010000
-----+-----+-----+-----+-----+-----+-----+
VNAME          NNAME
-----+-----+-----+-----+-----+-----+-----+
HANS           BAUER
FRED           MAYER
JORG           WAGNER
DSNE610I NUMBER OF ROWS DISPLAYED IS 3
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
DSNE621I NUMBER OF INPUT RECORDS READ IS 1
Command ==>
F1=Help      F3=Exit      F5=Rfind      F12=Cancel    Scroll ==> PAGE
```

Abbildung 31: Erfolgreiche Definition

Die Abfrage war erfolgreich, wie aus Abbildung 31 hervorgeht.

SPUIFI gibt den Inhalt der Tabelle wieder. Die mit dem "INSERT"-Statement eingegebenen drei Vornamen und Nachnamen sind in der Tabelle gespeichert.

Wir können das Aufnehmen von Datensätzen wiederholen, indem wir den SPUIFI-Screen mit "INSERT" erneut aufrufen und weitere Namen eingeben. Sie werden an den vorhandenen Datenbestand angehängt. Führen Sie es als Übung durch:

F3-Taste drücken, um den SPUIFI-Screen aufzurufen, den aufzurufenden Membernamen "INSERT" eingeben, zweimal die Eingabetaste drücken, um zum "Edit Screen" zu gelangen, neue Daten eingeben, mit F3 und anschließend mit der Eingabetaste das Aufnehmen dieser Namen in die Tabelle durchführen.

```

Menu  Utilities  Compilers  Help
-----
BROWSE      PRAKT20.SPUFI.OUT                               Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM PRAKT20.TAB020;                               00010000
-----+-----+-----+-----+-----+-----+-----+
VNAME      NNAME
-----+-----+-----+-----+-----+-----+
HANS       BAUER
FRED       MAYER
JORG       WAGNER
HEINZ      BAUER
FRITZ      MAYER
RICHARD    SCHULTE
JORG       MEISTER
HANS       BERG
FRITZ      MEIER
RICHARD    SCHMITZ
MARTIN     WAGNER
DSNE610I NUMBER OF ROWS DISPLAYED IS 11
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 100
Command ==>
F1=Help    F3=Exit    F5=Rfind   F12=Cancel
Scroll ==> PAGE

```

Abbildung 32: Datenbank mit erweitertem Inhalt

Wenn wir uns nun mittels des SQL-Kommandos "SELECT * FROM PRAKT20.TAB020;" wieder den Inhalt unserer Tabelle anschauen, könnte dieser wie in Abbildung 32 dargestellt aussehen.

Nach Betätigen der F8-Taste erscheint der nächsten Screen der Ausgabe.


```

Menu  Utilities  Compilers  Help
-----
BROWSE      PRAKT20.SPUFI.OUT                      Line 00000019 Col 001 080
-----+-----+-----+-----+-----+-----+-----+-----+-----+
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 1
DSNE622I NUMBER OF OUTPUT RECORDS WRITTEN IS 27
***** Bottom of Data *****

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

Scroll ==> PAGE

```

Abbildung 33: Der letzte Teil der Ausgabe

Wir haben erfolgreich eine DB2-Datenbank angelegt und mit Daten bestückt. Bitte bedenken Sie, dass wir hier Prinzipien erklären, in der Praxis jedoch meist mit anderen und weitgehend automatisierten Prozessen gearbeitet wird.

***Aufgabe:** Erstellen Sie einen Screenshot entsprechend der Abbildung 31, der alle Einträge Ihrer Tabelle zeigt. Schicken Sie anschließend beide Screenshots (auch den, den Sie laut Aufgabe auf der Seite 13 erstellen sollen) an Ihren Betreuer.*