

# CICS Cobol Tutorial 2

## Anlegen einer DB2 Datenbank

© Abteilung Technische Informatik, Institut für Informatik, Universität Leipzig  
© Abteilung Technische Informatik, Wilhelm Schickard Institut für Informatik, Universität Tübingen

### Übersicht

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.

Die Aufgabe besteht aus den folgenden Schritten

1. DB2
2. Anlegen benötigter Datasets
3. Einloggen ins z/OS DB2
4. Einstellen des Subsystem identifiers (SSIDs)
5. Starten von SPUFI
6. Überblick über die vier vorzunehmenden Definitionen
  - 6.1 DB2
  - 6.2 DB2 Storage Management
  - 6.3 Definitionen
  - 6.4 Tabellenform
7. Definition des Speicherplatzes für Datenbanken
8. Anlegen einer Datenbank
9. Definition von Tablespace für DB2-Tabellen
10. Erstellen der Tabelle
11. Datensätze in die Tabelle einfügen
12. Ansehen sämtlicher Datensätze der Tabelle

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

# 1. DB2

Das Datenbankmanagementsystem DB2 existiert in zwei unterschiedlichen Implementierungen, die jedoch weitestgehend kompatibel miteinander sind:

- DB2 for z/OS, (frühere Bezeichnungen: DB2 UDB for z/OS)
- DB2 UDB for Linux, UNIX and Windows

DB2 Express für Windows und Linux ist eine kostenlose (nicht open Source) Version, welche den gleichen Code wie DB2 UDB benutzt.

DB2 Everyplace ist eine slimmed-down, embedded Version von DB2 UDB für Cellphones, PDAs und ähnliche Devices.

DB2 for z/OS verfügt über zahlreiche Funktionen, die unter DB2 UDB nicht verfügbar sind. Unterstützung für Sysplex, Coupling Facility, Workload Manager sind nur einige Beispiele.

DB2 verwaltet Daten in Tables und speichert sie in Tablespaces. DB2 unterstützt neben den Standard-SQL-Datentypen auch binäre Datentypen (Text, Töne, Bilder, Videos, XML-Daten). Zusammen mit Oracle Database und MS SQL-Server gehört DB2 zu den Datenbanksystemen mit den größten Marktanteilen.

Der Datenzugriff der Anwendungsschicht erfolgt mit SQL, das weitgehend dem ANSI-SQL Standard entspricht. Zugriffe auf gespeicherte Daten können daher aus vielen Programmiersprachen heraus mit eingebettetem SQL erfolgen. Darüber hinaus hat DB2 APIs für REXX, PL/I, COBOL, RPG, FORTRAN, C/C++, Delphi, .NET CLI, Java, andere.

Um bei der Ausführung von DB-Zugriffen optimale Performance zu erzielen, wird ein sog. Optimizer eingesetzt, ein regelbasiertes Expertensystem, welches bei der Programmvorbereitung den Zugriff auf die betreffenden Tabellen festlegt. Dies beruht unter anderem auf den sogenannten Tabellenstatistiken, die mittels des Tools „RUNSTATS“ periodisch aktualisiert werden können, berücksichtigt aber auch andere Kennwerte wie z. B. die Anzahl der CPUs, den Systemzustand, die verfügbare Speichermenge und die physische Verteilung der Daten.

## 2. Anlegen benötigter Datasets

Wir loggen uns als TSO-Benutzer ein und wählen aus dem "ISPF Primary Option Menu" die Option "3" (Utilities) aus. Anschließend gehen wir mittels "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 . . . PRAK085
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 "PRAK085" drei neue Partitioned DataSets (PDS) an:

PRAK085.CICSDDB2.TEST01 (s. Abbildung 1)  
PRAK085.SPUFI.IN  
PRAK085.DBRMLIB.DATA

Wir verwenden dazu die in der Abbildung 2 angegebenen Parameter.

Die Member von "PRAK085.CICSDDB2.TEST01" nehmen das von uns zu erstellende C-Programm (Tutorial 5), das dazugehörige BMS-Programm und das JCL-Script auf. Der Dataset wird im später zu absolvierenden Tutorial 5 (C-Version) benötigt.

```

Menu  RefList  Utilities  Help
-----
                          Allocate New Data Set

Data Set Name . . . : PRAK085.CICSDB2.TEST01

Management class . . . (Blank for default management class)
Storage class . . . . (Blank for default storage class)
Volume serial . . . . (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 (M, K, or U)
Primary quantity . . 18 (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, LARGE, BASIC, *
                       EXTREQ, EXTPREF or blank)

Expiration date . . . (YY/MM/DD, YYYY/MM/DD
Enter "/" to select option YY.DDD, YYYY.DDD in Julian form
Allocate Multiple Volumes DDDD for retention period in days
                               or blank)

( * Specifying LIBRARY may override zero directory block)

( ** Only one of these fields may be specified)

Command ==>
F1=Help      F2=Split      F3=Exit      F7=Backward  F8=Forward  F9=Swap
F10=Actions  F12=Cancel
4B                                                  SC0TCP11          021/028

```

Abbildung 2: Die Parameter

Die Member von "PRAK085.SPUFI.IN" nehmen DB2-Kommandos auf, deren Ausführung unsere Datenbank irgendwie modifizieren wird. Wir werden DB2-Kommandos zum Anlegen von Datenbanken, Tabellen, Einfügen von Daten etc. kennen lernen.

In dem Partioned Dataset "PRAK085.DBRMLIB.DATA" werden Zwischenergebnisse abgespeichert, welche in Precompile-, Compile- sowie Link-Prozessen anfallen werden.

Für die spätere Bearbeitung des Tutorial 5 benötigen wir außerdem noch einen Partitioned Dataset "PRAK085.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 . . . PRAK085.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 "PRAK085.LIB"

Der Dataset "PRAK085.LIB" soll wie der Dataset "PRAK085.DBRMLIB.DATA" 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 zweimal die F3-Taste und kehren so in das "ISPF Primary Option Menu"-Panel zurück.

### 3. Einloggen ins z/OS DB2

Mit der Eingabe "M" für die Option "More" und danach "11" für die Option "DB2 V8" rufen wir das z/OS-DB2-Subsystem auf, das uns das Anlegen einer neuen DB2-Datenbank sowie deren Modifikation ermöglicht.

```
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)
 Q QMF          (Query Management Facility)
 X EXIT        (Leave DB2I)

PRESS:                END to exit        HELP for more information

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE
4B                                     SC0TCP26      002/015
```

Abbildung 4: das "DB2I PRIMARY OPTION MENU"

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

## 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 das Einstellen der SSID nicht mehr erforderlich.

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

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

```

                                DB2I DEFAULTS PANEL 1
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, 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)

PRESS:  ENTER to process      END to cancel      HELP for more information

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP     F10=LEFT      F11=RIGHT     F12=RETRIEVE
4B          SC0TCP26      002/015
```

Abbildung 5: Datenbankbezeichnung

Wir geben die beiden gekennzeichneten Werte ein, wenn sie nicht schon da stehen, lassen den Rest unverändert und bestätigen mit der Eingabetaste (s. Abbildung 5).

```
DB2I DEFAULTS PANEL 2
COMMAND ===>

Change defaults as desired:

1 DB2I JOB STATEMENT: (Optional if your site has a SUBMIT exit)
  ===> //PRAK085A JOB (ACCOUNT),'NAME'
  ===> /**
  ===> /**
  ===> /**

COBOL DEFAULTS: (For COBOL, COB2, or IBMCOB)
2 COBOL STRING DELIMITER ===> DEFAULT (DEFAULT, ' or ")
3 DBCS SYMBOL FOR DCLGEN ===> G (G/N - Character in PIC clause)

PRESS: ENTER to process      END to cancel      HELP for more information

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: DB2I DEFAULTS PANEL 2

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



```
DB2I PRIMARY OPTION MENU          SSID: D931
COMMAND ==>

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).

## 5. Starten von SPUFI

Es existieren zahlreiche Möglichkeiten, eine DB2 Datenbank anzulegen. Wir verwenden hier das DB2-Subsystem "SPUFI".

Unter z/OS ist SPUFI (SQL Processing Using File Input) ein Werkzeug, das es ermöglicht, interaktiv Datenbankstrukturen zu ändern, SQL-Statements zu editieren, auszuführen und die Ergebnisse aufbereitet anzuzeigen. Ansonsten bezeichnet SPUFI ein SQL-Statement, welches nicht in ein Programm eingebettet, sondern über eine Datei oder eine manuelle Eingabe abgesetzt wird.

Wir geben eine "1" ein (COMMAND ==> 1), betätigen die Eingabetaste und erhalten den folgenden Screen.

```
====>                                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

4B                                                    005/29
```

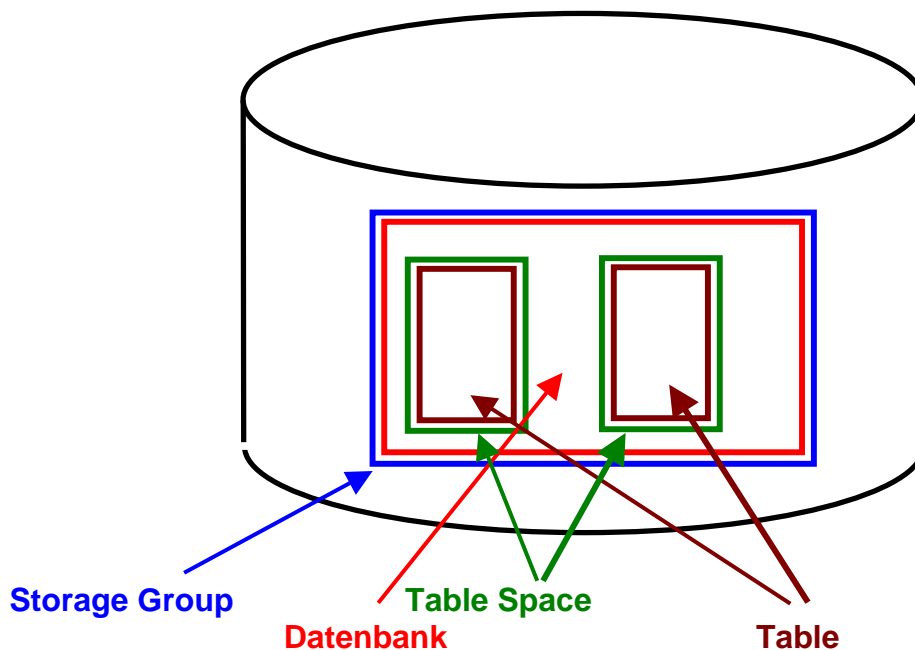
SPUFI-Panel

Abbildung 8:

## 6. Überblick über die vier vorzunehmenden Definitionen

### 6.1 DB2 Spaces

Für die Anlage einer Datenbank brauchen wir Definitionen, die eine Aussage über das "was", "wie" und "wo" machen. Aus der Sicht des Benutzers besteht eine DB2 Datenbank aus mindestens einer, meistens aber mehreren Tabellen.



Auf einem Plattenspeicher wird für die Datenbank Platz in der Form einer Storage Group definiert und angelegt. Diese nimmt eine DB2 Datenbank auf.

Innerhalb der Storage Group werden eine oder mehrere Table Spaces angelegt. Jeder Table Space nimmt eine DB2-Tabelle auf.

Als Beispiel sei eine DB2 Großrechnerinstallation erwähnt, die mehr als 50 000 Tabellen oder Indices für ERP/CRM Anwendungen verwendet.

### 6.2 DB2 Storage Management

Storage Management System (SMS) ist die z/OS Komponente, welche für die Plattenspeicherplatz-Verwaltung zuständig ist. Data Sets werden von SMS generell in Data Classes, Storage Classes und Management Classes zusammengefasst. Data Sets mit identischen Data Classes, Storage Classes und Management Class Eigenschaften werden von SMS zu einer Storage Group zusammen gefasst. Eine Storage Group ist eine Gruppe von Plattenspeicher Volumes, die SMS-verwaltete Data Sets enthält.

Eine DB2 Storage Group ist ebenfalls eine derartige SMS Storage Group und besteht aus einzelnen Data Sets.

DB2 speichert seine Daten in VSAM Linear Data Sets (LDS). Jeder DB2 Table Space oder Index Space erfordert wenigstens einen (möglicherweise mehrere) VSAM Data Sets. DB2 benutzt den VSAM Media Manager für seine I/O Operationen. Für jede I/O Anforderung erstellt der VSAM Media Manager ein Kanalprogramm und sendet eine Request an den z/OS I/O Supervisor.

Solutions Journal: DB2 and Storage Management

<http://www.craigsmullins.com/DB2%20and%20Storage%20Management.pdf>

## 6.3 Definitionen

Im Einzelnen werden nun vier Definitionen erstellt, 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 ("SYSDEFLT" in dem vorliegenden Beispiel), siehe Abbildung 12.
2. Für den symbolischen Namen der Datenbank wird der Name "PRAK085" gewählt.

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.

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 "TABSP085"). In unserem einfachen Beispiel legen wir nur eine einzige Tabelle an, benötigen also auch nur einen Tablespace (siehe Abbildung 20).
4. Schließlich muss die Tabelle selbst bezüglich ihres Namens ("TAB085"), ihrer Struktur und der Bezeichnung ihrer Felder (Spalten) definiert werden (siehe Abbildung 23).

Für die vier Definitionen erstellen wir je einen Member. Die folgende Tabelle enthält eine Übersicht über die Membernamen, die Aufgaben und die symbolischen Namen. Hierbei unterscheiden wir zwischen

- den symbolischen Namen der Datenbank, der Tabelle und des jeweiligen zugeordneten Speicherplatzes (rechte Seite obiger Tabelle),
- den Namen der Member von "PRAK085.SPUFI.IN", die diese Definitionen aufnehmen (linke Seite obiger Tabelle).

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

SPUFI interner Member Name	Aufgabe	symbolischer Name
STOGR1	Speicherplatz für unsere DB2-Datenbank anlegen	SYSDEFLT
DB1	Die Datenbank selbst anlegen	PRAK085
TABSP1	Speicherplatz für eine Tabelle anlegen	TABSP085
TAB1	Die Tabelle selbst anlegen	TAB085

Das Anlegen dieser Definitionen ist in den Abschnitten 6 bis 9 beschrieben.

**Problem:**

Diese Definitionen können teilweise nur mit bestimmten Privilegien erzeugt werden, über welche Ihre Accounts nicht verfügen. Im Falle des Praktikumaccounts sind Storage Group und Database vorgegeben und die Privilegien nur auf den Tablespace und den (die) Table begrenzt. Als Storage Group verwenden Sie die Default-Storage Group "SYSDEFLT".

In den Abschnitten 6, 7 und 8 wird beschrieben, wie ein User mit den entsprechenden Privilegien eine Storage Group und eine Database anlegt. Als Praktikumsteilnehmer können Sie dies nicht tun. Ihr Betreuer hat deshalb diese Schritte für Sie bereits vorgenommen; Sie finden deshalb Storage Group und Datenbank bereits vor. Schauen Sie sich die Vorgehensweise in den Abschnitten 6, 7 und 8 dennoch an.

Sie selbst setzen Ihre Arbeit mit Abschnitt 9 fort.

## 6.4 Tabellenform

Wir legen für unseren Partitioned Dataset "PRAK085.SPUFI.IN" nur zwei Members an:

Member Name	Aufgabe	symbolischer Name
<del>STOCR1</del>	<del>Speicherplatz für unsere DB2 Datenbank anlegen</del>	<del>CYSDEFLT</del>
<del>DB1</del>	<del>Die Datenbank selbst anlegen</del>	<del>PRAK085</del>
TABSP1	Speicherplatz für eine Tabelle anlegen	TABSP085
TAB1	Die Tabelle selbst anlegen	TAB085

Die von Ihnen anzulegende Tabelle soll über 2 Spalten verfügen und das folgende Format haben:

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

## 7. Definition des Speicherplatzes für Datenbanken

Wir beginnen mit der Definition des Speicherplatzes (Storage Group) für unsere Datenbank. Hierfür wird, wie bereits beschrieben, das entsprechende Privileg benötigt. Praktikumssteilnehmer besitzen dies nicht.

```
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:
CONNECT LOCATION  ====>

PRESS: ENTER to process          END to exit          HELP for more information

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 "PRAK085.SPUFI.IN(STOGR1)" gespeichert werden. Diese Eingabe wird von SPUFI übersetzt.

Für die Ausgabe der Übersetzung wird ein neuer Dataset benötigt. Wir nennen ihn PRAK085.SPUFI.OUT. Er wird automatisch angelegt.

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

Was jetzt geschieht ist die Eingabe und Übersetzung eines SQL Statements. Normalerweise sind SQL Statements Bestandteil eines Anwendungsprogramms und werden als Teil des Anwendungsprogramms übersetzt und ausgeführt. SPUFI gibt uns eine Möglichkeit, ein SQL Statement auch außerhalb eines Anwendungsprogramms zu übersetzen und auszuführen.

```

CURRENT SPUFI DEFAULTS                SSID: D931
====>

Enter the following to control your SPUFI session:
 1 SQL TERMINATOR .. ==> ;           (SQL Statement Terminator)
 2 ISOLATION LEVEL  ==> RR           (RR=Repeatable Read, CS=Cursor Stability)
 3 MAX SELECT LINES ==> 250         (Maximum number of lines to be
                                     returned from a SELECT)
 4 ALLOW SQL WARNINGS==> NO         (Continue fetching after sqlwarning)
 5 CHANGE PLAN NAMES ==> NO         (Change the plan names used by SPUFI)

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

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

PRESS:  ENTER to process      END to exit                HELP for more information

F1=HELP      F2=SPLIT      F3=END        F4=RETURN     F5=RFIND     F6=RCHANGE
F7=UP        F8=DOWN       F9=SWAP      F10=LEFT     F11=RIGHT    F12=RETRIEVE
4B                                     SC0TCP06     002/007

```

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.







```
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   ===> *        (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 ===>

PRESS: ENTER to process   END to exit   HELP for more information

DSNE808A EDIT SESSION HAS COMPLETED. PRESS ENTER TO CONTINUE
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 13: Bestätigung der beendeten Edit-Session

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.

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

```
Menu Utilities Compilers Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+
CREATE STOGROUP SYSDEFLT                                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
```

Abbildung 14: Erfolgreiche Übersetzung

Wird an dieser Stelle eine Fehlermeldung ausgegeben, dann ist der zu erstellende Speicherplatz für unsere Datenbank (Stogroup) möglicherweise bereits vorhanden. In diesem Fall gehen sie wie im Anhang beschrieben vor.

## 8. 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.

Als nächstes wird im Speicherplatz "SYSDEFLT" 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

Wir geben den in Abbildung 16 gezeigten Dataset- und Member-Namen ein und drücken anschließend die Eingabetaste.

Wir übernehmen im nächsten Screen wieder alle Werte unverändert und bestätigen mit der Eingabetaste.



```

Menu Utilities Compilers Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+
CREATE DATABASE PRAK085                               00010000
STOGROUP SYSDEFLT                                   00020000
BUFFERPOOL BP0;                                     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 18: Erfolgsmeldung der Anlage der Datenbank

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

Die von uns bisher vorgenommene Reservierung von Speicherplatz für die Datenbank und das Erstellen der Datenbank wird normalerweise vom Systemadministrator vorgenommen, dem einzelnen Praktikumsbenutzer fehlen hierfür in der Regel die Zugriffsrechte. Die weiteren Schritte kann der Benutzer aber selbst vornehmen.



## 9. Definition von Tablespace für DB2-Tabellen

**Aufgabe:** Erstellen Sie den Tablespace wie im folgenden Beispiel beschriebene. Dieser soll den Namen "TABSPxxx" erhalten, wobei "xxx" für Ihre dreistellige Nummer Ihres PRAK-Accounts steht.

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).

```
          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

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



```

Menu Utilities Compilers Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+
CREATE TABLESPACE TABSP085                        00010000
  IN PRAK085                                        00020000
  USING STOGROUP SYSDEFLT                          00030000
  PRIQTY 60                                         00040000
  SECQTY 60                                         00050000
  ERASE NO                                          00085000
  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 Tablespace "TABSP085" 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.

## 10. Erstellen der Tabelle

**Aufgabe:** Definieren Sie Ihre Tabelle, wie im Folgenden beschrieben. Diese soll den Namen "TABxxx" erhalten, wobei "xxx" für Ihre dreistellige Nummer Ihres PRAK-Accounts steht.

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

```
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

Um weiter zu kommen, drücken wir zweimal die Eingabetaste.



```

Menu Utilities Compilers Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+
CREATE TABLE TAB085                                00010000
(                                                    00020000
  VNAME CHAR(20) NOT NULL,                          00030000
  NNAME CHAR(20) NOT NULL                          00040000
)                                                    00050000
  IN PRAK085.TABSP085;                             00085000
-----+-----+-----+-----+-----+-----+-----+
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 6
DSNE622I NUMBER OF OUTPUT RECORDS WRITTEN IS 18
Command ==>
F1=Help    F3=Exit    F5=Rfind   F12=Cancel
                                           Scroll ==> PAGE

```

Abbildung 24: Erfolgreiche Definition

Da die Definiton erfolgreich war (s. Abbildung 24), verlassen wir den Screen mit F3.

## 11. Datensätze in die Tabelle einfügen

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

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

Hierzu legen wir (zusätzlich zu den bisherigen 4 Members) einen weiteren SPUFI-Member "INSERT" an (s. Abbildung 25)...

```
SPUFI                                SSID: D931
====>

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

... und drücken zweimal die Eingabetaste.





```

Menu Utilities Compilers Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO TAB085                                00010001
  VALUES ('HANS', 'BAUER');                       00020002
-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+
INSERT INTO TAB085                                00030001
  VALUES ('FRED', 'MAYER');                       00040002
-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+
INSERT INTO TAB085                                00050001
  VALUES ('JORG', 'WAGNER');                     00060002
-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
Command ==>
F1=Help    F3=Exit    F5=Rfind   F12=Cancel
                                         Scroll ==> PAGE

```

Abbildung 27: Erfolgreiche Definition

Die Eingabe war erfolgreich (s. Abbildung 27). Mit der F8-Taste sehen wir uns den Rest der Ausgabe an.

```
Menu Utilities Compilers Help
-----
BROWSE      PRAK085.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 nun unsere Datenbank mit Daten beschrieben.  
Mit der F3-Taste rufen wir den SPUFI-Panel erneut auf.

## 12, Ansehen sämtlicher Datensätze der Tabelle

Frage: Befinden sich nun die korrekten Namen in der Datenbank? Mit SPUFI können wir uns auch den Datenbank-Inhalt ansehen.

**Aufgabe:**            *Selektieren Sie alle Einträge in Ihrer Tabelle, wie im folgenden Beispiel beschrieben.*

Wir erstellen noch einen weiteren SPUFI-Member "SELECT" (s. Abbildung 29) ...

```
SPUFI                               SSID: DB7G
====>
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

... und drücken zweimal die Eingabetaste.



```

Menu Utilities Compilers Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM PRAK085.TAB085;                      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.

SPUFI 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 SPUFI-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 SPUFI-Screen aufzurufen, den aufzurufenden Membernamen mit "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.

Wenn wir uns nun mittels des SQL-Kommandos "SELECT \* FROM PRAK085.TAB085;" wieder den Inhalt unserer Tabelle anschauen, könnte dieser wie in Abbildung 32 dargestellt aussehen.

```

Menu Utilities Compilers Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM PRAK085.TAB085;                      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

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

```
Menu Utilities Compilers Help
-----
BROWSE      PRAK085.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

**Aufgabe:** Erstellen Sie einen Screenshot entsprechend der Abbildung 31 bzw. 32, \_\_\_\_\_ der alle Einträge Ihrer Tabelle zeigt.

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.

## Anhang

### Fehlermeldung beim Anlegen der Storage Group Löschen von alten Objekten, um Neue anlegen zu können

Dieser Anhang ist für die Praktikumssteilnehmer von theoretischem Interesse. Hier wird der Vollständigkeit halber beschrieben, wie ein privilegierter User vorgeht, wenn Fehler beim Anlegen einer Storage Group passieren. Folgende Fehlersituation: Nehmen wir an, wir sind folgendermaßen vorgegangen:

```
====>                                SPUFI                                SSID: D931

Enter the input data set name: _____ (Can be sequential or partitioned)
 1 DATA SET NAME ... ==> SPUFI.IN(STOGR129)
 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
```

Enter



**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  
F7=UP

F2=SPLIT  
F8=DOWN

F3=END  
F9=SWAP

F4=RETURN  
F10=LEFT

F5=RFIND  
F11=RIGHT

F6=RCHANGE  
F12=RETRIEVE



SPUFI

SSID: D931

====>

**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(STOGR129)
- 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

Und jetzt Enter

```

Menu Utilities Compilers Help
-----
BROWSE      PRAK129.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+
CREATE STOGROUP STOGR129                               00010000
  VOLUMES (SCPMV5)                                     00020000
  VCAT DSN510;                                         00030000
-----+-----+-----+-----+-----+-----+-----+-----+
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
          STOGR129 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

```

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 STOGR129**

Hierzu gehen Sie folgendermaßen vor:

```
====>                                SPUFI                                SSID: D931

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

Im SPUFI Panel machen Sie die obenstehende Eingabe. 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 STOGROUP STOGR129" auf.



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 unten 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 "DB129.TABSP129". Dies geschieht durch das SQL-Statement

```

Menu  Utilities  Compilers  Help
-----
BROWSE      PRAK085.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+
DROP STOGROUP STOGR129                                00010000
-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNT408I  SQLCODE = -616, ERROR:  STOGROUP STOGR129 CANNOT BE DROPPED BECAUSE IT
          IS REFERENCED BY TABLESPACE DB129.TABSP129
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

```

Die Fehlermeldung gibt aber den Namen des Objektes aus, welches vor der Stogroup noch gelöscht werden muss, in diesem Beispiel der Tablespace "DB129.TABSP129". Dies geschieht durch das SQL-Statement

**DROP TABLESPACE DB129.TABSP129**

Weiterhin ist folgende Situation möglich. Sie haben Ihre Database angelegt, hierbei aber einen Fehler gemacht. Wenn Sie das Edit Panel erneut aufrufen, den Fehler korrigieren und dann F3 und Enter machen, erscheint eine Fehlermeldung, dass das Objekt schon existiert. In diesem Fall müssen Sie das fälschlich angelegte Objekt mit

**DROP DATABASE DB129**

erst eliminieren, ehe Sie die korrekte Version erneut anlegen können.