

Tutorial 4

DB2

Ziel dieses Tutorials ist es, eine OS/390 - 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: Arbeiten Sie nachfolgendes Tutorial durch.

Wir loggen uns als TSO-Benutzer ein und wählen aus dem "CUSTOMPAC MASTER APPLICATION MENU" die Option "P" aus. Anschließend arbeiten wir uns zum "Data Set Utility" durch.

```

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

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

ISPF Library:
Project . . . PRAKT20
Group . . . CICSDB2
Type . . . . TEST01

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

Data Set Password . . . (If password protected)

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

```

Abbildung 1: "Data Set Utility"

Wir legen für den Benutzer "PRAKT20" drei neue Partitioned Datasets an, mit den Namen:

PRAKT20.CICSDB2.TEST01 (Abbildung 1)
PRAKT20.SPUFI.IN
PRAKT20.DBRMLIB.DATA

Die Members von "PRAKT20.CICSDB2.TEST01" nehmen die von uns erstellten Programme und JCL-Scripts auf.

```

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       . . . . MEGABYTE     (BLKS, TRKS, CYLS, KB, MB, BYTES
or RECORDS)
Average record unit
Primary quantity  . . 2                (In above units)
Secondary quantity . . 1                (In above units)
Directory blocks  . . 5                (Zero for sequential data set) *
Record format    . . . . FB
Record length    . . . . 80
Block size       . . . . 11440
Data set name type : PDS                (LIBRARY, HFS, PDS, or blank) *
(YEAR/MM/DD, YYYY/MM/DD)

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

```

Abbildung 2: Die Parameter

Wir verwenden für alle Datasets die gleichen "Partitioned Data Set"-Definitionen wie bisher (s. Abbildung 2).

Die Members von "PRAKT20.SPUFI.IN" nehmen unsere Datenbankdefinitionen auf. In dem Partioned Dataset "PRAKT20.DBRMLIB.DATA" werden Zwischenergebnisse abgespeichert.

Wir benötigen außerdem einen Partitioned Dataset "PRAKT20.LIB". Hier wird angenommen, dass er in der letzten Sitzung angelegt wurde und deshalb bereits existiert.

```

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"

Auch für diesen Dataset benutzen wir die gleichen Parameter.

In dem Partitioned Dataset "PRAKT20.LIB" werden während eines Übersetzungsschrittes Templates für den nächsten Schritt abgelegt. Wurde der Partitioned Dataset "PRKAT20.LIB" schon vorher angelegt, ist dieser Schritt nicht mehr notwendig.

Aufgabe: Erstellen Sie die benötigten Datasets.

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

Mit der Eingabe "DB2" rufen wir ein ISPF-Subsystem auf, das uns das Anlegen einer neuen DB2-Datenbank ermöglicht. Anschließend drücken wir die Eingabetaste.

```

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

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). Ehe wir mit dem Anlegen der Datenbank beginnen, müssen wir Defaultwerte setzen. Wir geben "d" ein und bestätigen anschließend mit der Eingabetaste.

```

                                DB2I DEFAULTS
COMMAND ==>

Change defaults as desired:

 1 DB2 NAME ..... ==> DBA1      (Subsystem identifier)
 2 DB2 CONNECTION RETRIES ==> 0    (How many retries for DB2 connection)
 3 APPLICATION LANGUAGE ==> IBMCOB (ASM, C, CPP, COBOL, COB2, IBMCOB,
  FORTRAN, PLI)
 4 LINES/PAGE OF LISTING ==> 60    (A number from 5 to 999)
 5 MESSAGE LEVEL ..... ==> I      (Information, Warning, Error, Severe)
 6 SQL STRING DELIMITER ==> DEFAULT (DEFAULT, ' or ")
 7 DECIMAL POINT ..... ==> .      (. or ,)
 8 STOP IF RETURN CODE >= ==> 8    (Lowest terminating return code)
 9 NUMBER OF ROWS ..... ==> 20    (For ISPF Tables)
10 CHANGE HELP BOOK NAMES?==> NO   (YES to change HELP data set names)
11 DB2I JOB STATEMENT: (Optional if your site has a SUBMIT exit)
    ==> //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 SSID ist eine Datenbankbezeichnung, die systemintern benutzt wird. Auf unserer Maschine "jedi.informatik.uni-leipzig.de" ist dies die Bezeichnung "DBA1". Auf einem anderen OS/390-Rechner kann die Bezeichnung anders sein. Ist das der Fall, so kann die Bezeichnung beim Systemadministrator erfragt werden.

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: DBA1
COMMAND ==> 1
Select one of the following DB2 functions and press ENTER.

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

P  DB2 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 "DBA1" eingetragen (s. Abbildung 7).

Das zum Anlegen einer Datenbank verwendete ISPF-Subsystem hört auf den Namen "SPUFI". Wir geben eine "1" ein und betätigen die Eingabetaste.

```

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

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

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

For remote SQL processing:
10 CONNECT LOCATION ==>

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

```

Abbildung 8: Der SPUFI Panel

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

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 Partitioned 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 (STOGR1 in dem vorliegenden Beispiel).

Für die Bezeichnung (symbolischer Name) der Datenbank wählen wir den Namen "DBPRAK1".

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).

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 SPAPRAK1). In unserem einfachen Beispiel begnügen wir uns mit einer einzigen Tabelle, benötigen also auch nur einen Tablespace.

Schließlich muß die Tabelle selbst bezüglich ihres Namens (TABPRAK1), ihrer Struktur und der Bezeichnung ihrer Felder (Spalten) definiert werden. Unsere Tabelle soll über 2 Spalten verfügen und das folgende Format haben:

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

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

Member-Name	Aufgabe	Bezeichnung
STOPRAK	Speicherplatz für unsere DB2 Datenbank	STOGR1
DBPRAK1	Die Datenbank selbst aufnehmen	DBPRAK1
SPAPRAK1	Speicherplatz für eine Tabelle	SPAPRAK1
TABPRAK1	Die Tabelle selbst aufnehmen	TABPRAK1

Wir unterscheiden zwischen:

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

Die Bezeichnungen können, müssen aber nicht identisch sein.

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

```

                                SPUFI                                SSID: DBA1
===>

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

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

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

For remote SQL processing:
10 CONNECT LOCATION ==>

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

```

Abbildung 9: Definition für "STOPRAK"

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

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

Nachdem diese Änderungen (s. Abbildung 9) vorgenommen wurden, drücken wir die Eingabetaste.

SPUFI teilt uns mit, dass damit die Edit Session – Erstellen des Members SPUFI.IN(STOPRAK) - beendet wurde. Unsere Eingabe wird übersetzt, das Ergebnis in SPUFI.OUT gestellt (s. Abbildung 13).

Wir drücken nun die Eingabetaste.

```

Menu  Utilities  Compilers  Help
-----
  BROWSE      PRAKT20.SPUFI.OUT                               Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+
CREATE STOGROUP STOGR1                                       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

Das Ergebnis der Übersetzung wird mitgeteilt (s. Abbildung 14). Offensichtlich ist alles gut gelaufen.

Wir verlassen diesen Screen mit der F3-Taste.

Aufgabe: Definieren Sie den Speicherplatz für die Datenbank.

```

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

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

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

For remote SQL processing:
10 CONNECT LOCATION ==>

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

```

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 für den bereitgestellten Speicherplatz "STOPRAK" eine Datenbank definiert.

```

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

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

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

For remote SQL processing:
10 CONNECT LOCATION ==>

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

```

Abbildung 16: Definition für DBPRAK

Wir speichern die Definition für die Datenbank in einem Member mit den Namen "PRAKT20.SPUFI.IN(DBPRAK)" ab (s. Abbildung 16) und drücken die Eingabetaste.

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

Aufgabe: Definieren Sie Ihre Datenbank.

```

                                SPUFI                                SSID: DBA1
====>
DSNE361I SPUFI PROCESSING COMPLETE
Enter the input data set name:      (Can be sequential or partitioned)
 1  DATA SET NAME ... ==> SPUFI.IN(SPAPRAK1)
 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: Definition für SPAPRAK1

In unserem Beispiel soll die DB2-Datenbank aus einer einzigen Tabelle bestehen. Wir erstellen eine Definition des Speicherplatzes (Table Space) für unsere Tabelle in dem Member "SPAPRAK1" (s. Abbildung 19).

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

Die Einträge in der Tabelle "TABPRAK1" (s. Abbildung 23) haben zwei Spalten: "VNAME" (zur Aufnahme des Vornamens einer Person) und "NNAME" (zur Aufnahme des Nachnamens einer Person). Die Felder haben eine Länge von je 20 Zeichen. Die Tabelle ist Teil der Datenbank "DBPRAK1" und verwendet den Tablespace "SPAPRAK1".

Wir verlassen den Screen mit F3 und bestätigen anschließend mit der Eingabetaste.

```

Menu  Utilities  Compilers  Help
-----
  BROWSE      PRAKT20.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+
CREATE TABLE TABPRAK1                                00010000
(                                                       00020000
  VNAME CHAR(20) NOT NULL,                             00030000
  NNAME CHAR(20) NOT NULL                             00040000
)                                                       00050000
  IN DBPRAK1.SPAPRAK1;                                00060000
-----+-----+-----+-----+-----+-----+-----+
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 ==>                                         Scroll ==> PAGE
  F1=Help      F3=Exit      F5=Rfind  F12=Cancel

```

Abbildung 24: Erfolgreiche Definition

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

Aufgabe: Definieren Sie Ihre Tabelle.


```

Menu Utilities Compilers Help
-----
BROWSE      PRAKT20.SPUFI.OUT                      Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO TABPRAK1                                00010001
      VALUES ('HANS', 'BAUER');                      00020002
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO TABPRAK1                                00030001
      VALUES ('FRED', 'MAYER');                      00040002
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
DSNE615I NUMBER OF ROWS AFFECTED IS 1
DSNE616I STATEMENT EXECUTION WAS SUCCESSFUL, SQLCODE IS 0
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
INSERT INTO TABPRAK1                                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 Aufgabe an.

```

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: der restliche Screen

Wir haben unsere Datenbank mit Daten "gefüttert".
Mit der F3-Taste rufen wir SPUFI erneut auf.

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.SPUFI.OUT                               Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM PRAKT20.TABPRAK1;
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 Datenbank wieder. Die mit dem "INSERT"-Statement eingegebenen drei Vornamen und Nachnamen sind in der Datenbank gespeichert.

Wir können jetzt die Lösung 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, "INSERT" eingeben, zweimal die Eingabetaste drücken, um zum "Edit Screen" zu gelangen, neue Daten eingeben, mit F3 und anschließend mit der Eingabetaste wieder zum "Browse Screen".

```

Menu  Utilities  Compilers  Help
-----
BROWSE      PRAKT20.SPUFI.OUT                               Line 00000000 Col 001 080
***** Top of Data *****
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
SELECT * FROM PRAKT20.TABPRAK1;
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

Das Ergebnis könnte 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 immer, dass wir hier Prinzipien erklären und in der Praxis mit anderen und weitgehend automatisierten Prozessen gearbeitet wird.

Organisatorisches

- Bearbeiten Sie die Aufgaben bis zum 25.1.2002.
- Löschen Sie nichts von Ihrer Lösung.
- Bei Bedarf sind Konsultationen nach Vereinbarung möglich.