

# Tutorial 17

## Erstellen und Benutzen von VSAM-Datasets

Copyright © Abt. Computersysteme, Institut für Informatik, Universität Leipzig

Dies ist ein einführendes Tutorial zur Benutzung von VSAM-Datasets.  
Dieses Tutorial behandelt „Key Sequential Datasets“ (KSDS).

**Das Tutorial findet komplett unter TSO statt.  
Es werden Erfahrungen im Umgang mit ISPF (Umgang mit Datasets), dem ISPF-Editor  
und SDSF vorausgesetzt.**

Als Beispiel dient folgendes:

Es sollen für die Universität Matrikelnummern, studserv-Kürzel sowie Vor- und Nachname gespeichert werden. Dabei soll in einem Programm einmal über die Matrikel-Nummer und einmal über das studserv-Kürzel Zugriff auf die restlichen Daten erfolgen.

Für diesen Zweck soll zuerst ein VSAM-Dataset angelegt werden, der die Daten beinhaltet und einen Index auf die Matrikelnummer anlegt. Dieser wird danach mit den Daten der Studenten gefüllt werden.

Um auf die Daten zuzugreifen, ist ein COBOL-Programm zu schreiben. Das Programm soll über einen JCL-Skript gestartet werden und dann als Batch-Job laufen.

Anschließend wird der VSAM-Dataset um einen alternativen Index – die Studserv-Kennung – erweitert und ein weiteres COBOL-Programm geschrieben (bzw. das vorhandene angepasst), das über diesen Index die Daten sucht.

### Erstellen des VSAM-Datasets

Für das Erstellen eines VSAM-Datasets gibt es mehrere Möglichkeiten. Man kann sie zum Beispiel über Dialoge innerhalb von ISPF erstellen (Option 3.2 – v). In diesem Tutorial wird der Dataset mit einem JCL-Skript erstellt.

Das Programm für die VSAM-Erstellung ist IDCAMS. Genauere Details der Befehle von IDCAMS kann man in „DFSMS Access Method Services for Catalogs“ (SC26-7394-03) nachlesen.

Bevor der VSAM-Cluster angelegt wird, hier erst einmal ein paar Überlegungen für die Datenhaltung.

Ein Record soll beinhalten:

- Matrikelnummer – 7 Zeichen
- Studserv-Kennung – 8 Zeichen
- Nachname – 20 Zeichen
- Vorname – 20 Zeichen

Damit hat ein Record 55 Zeichen. Die Daten werden in der oben angegebenen Reihenfolge im Record gespeichert.

Zum Bearbeiten der nachfolgenden Aufgaben können Sie die Datei "vsam.zip" nutzen. Diese kann unter "Datasets zum Tutorial" von der gleichen Web-Site heruntergeladen werden wie dieses Tutorial.

Bei der Anlage aller Datasets beachten Sie bitte, dass Sie für jeden Dataset nur so viel Plattenspeicher reservieren wie Sie auch benötigen. Ein oder zwei Tracks reichen in der Regel aus.

**Aufgabe:** Legen Sie einen Partitioned Dataset <Ihre User-ID>.VSAM.CNTL (im FB 80-Format) an, der zukünftig alle JCL-Scripte aufnehmen soll. Beachten Sie, dass dieser eine Größe von 2 Tracks nicht überschreitet.

**Aufgabe:** Erstellen Sie im Dataset <Ihre User-ID>.VSAM.CNTL einen Member DEFCLUST (Abbildung 1), mit dessen Hilfe der VSAM-Cluster erstellt werden kann.

**Aufgabe:** Führen Sie das JCL-Script DEFCLUST aus und prüfen Sie, ob der VSAM-Cluster tatsächlich angelegt wurde.

```
//PRAKT20D JOB ( ),NOTIFY=&SYSUID
//*
//* DEFINE VSAM CLUSTER
//*
//DEFCLS EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD SYSOUT=A
//SYSIN DD *

DELETE PRAKT20.VSAM.STUDENT

DEFINE CLUSTER ( -
    NAME(PRAKT20.VSAM.STUDENT) -
    VOL(SMS009) -
    RECORDSIZE(55 55) -
    RECORDS(10 10) -
    KEYS(7 0) -
    INDEXED ) -
DATA ( -
    NAME(PRAKT20.VSAM.STUDENT.DATA) ) -
INDEX ( -
    NAME(PRAKT20.VSAM.STUDENT.INDEX) )
/*
```

**Abbildung 1: JCL-Script zum Erstellen des VSAM-Clusters**

Die beiden Parameter bei RECORDSIZE geben die mittlere und die maximale Größe eines Records an, die in diesem Fall gleich groß sind. Wir wählen genau 55, weil ja ein Datensatz - bestehend aus Matrikel-Nr, Studserv-Kennung, Vor- und Zuname - genau 55 Zeichen lang ist.

INDEXED legt fest, dass es sich um einen KSDS handelt.

KEYS gibt an, dass der Schlüssel für den Zugriff 7 Zeichen lang ist und bei Offset 0 beginnt.

Als nächstes soll der Cluster mit Studenten-Daten gefüllt werden. Diese später in den VSAM-Cluster zu kopierenden Daten erstellen wir zunächst in einem sequentiellen Dataset.

Aufgabe: Legen Sie diesen sequentiellen Dataset an. Er sollte 55 Bytes/Record haben (Record-Format FB) und den Namen PRAKT20.VSAM.SEQDATA erhalten. Da dieser nur sehr wenige Beispiel-Daten aufnehmen soll, reicht eine Dataset-Größe von 1 Track.

Aufgabe: Füllen Sie unter Nutzung des ISPF-Editors den gerade angelegten Dataset mit einigen Beispiel-Datensätzen (siehe Abbildung 2). Sie können einige Fantasie-Datensätze verwenden, doch ein Datensatz muß wahrheitsgemäß Ihre persönlichen Daten enthalten: Ihr eigener Datensatz. Arbeiten mehrere Personen zusammen, dann muss für jede Person ein entsprechender wahrheitsgemäßer Datensatz erstellt werden.

```

File Edit Confirm Menu Utilities Compilers Test Help
-----
EDIT          PRAKT20.VSAM.SEQDATA                               Data set saved
*****      ***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>          your edit profile using the command RECOVERY ON.
000001 1187579mai91jhzMichaelson           Nils
000002 1207747mai02wdrMueller              Georg
000003 1207864mai03werFriedrich            Boris
000004 1213435mai03fgtHeinrich             Albrecht
000005 1309745mai03hgeFischer              Katrin
*****      ***** Bottom of Data *****

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

```

**Abbildung 2: PRAKT20.VSAM.SEQDATA mit 4 Datensätzen füllen**

Aufgabe: Legen Sie im Partitioned Dataset „PRAKT20.VSAM.CNTL“ den Member REPRO an, ein JCL-Script zum Kopieren der Daten aus dem sequentiellen Dataset in den VSAM-Cluster (Abbildung 3).

Aufgabe: Kopieren Sie die Studenten-Daten hinüber, indem Sie mittels SUB das JCL-Script ausführen.

```

//PRAKT20R JOB ( ),NOTIFY=&SYSUID
//*
//* COPY SEQUENTIAL DATASET INTO VSAM CLUSTER
//*
//DEFCLS EXEC PGM=IDCAMS,REGION=4096K
//SEQDD DD DSN=PRAKT20.VSAM.SEQDATA,DISP=SHR
//VSAMDD DD DSN=PRAKT20.VSAM.STUDENT,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
REPRO INFILE(SEQDD) -
      OUTFILE(VSAMDD)
/*

```

**Abbildung 3: Inhalt des Members REPRO**

## Schreiben des COBOL-Programms

Um auf den VSAM-Dataset zuzugreifen, erstellen wir ein COBOL-Programm.

Aufgabe: Erstellen Sie einen weiteren Partitioned Dataset <Ihre User-ID>.VSAM.COBOLE, der zukünftig die beiden Cobol-Programm-Codes aufnehmen soll (Record length = 80).

Aufgabe: Legen Sie in diesem einen Member "STUD" mit dem im folgenden angegebenen Programm-Code an.

Es folgt der Quelltext des Cobol-Programms:

```

000100 IDENTIFICATION DIVISION.
000200*
000300* Programm zum Zugriff auf VSAM-Dataset
000400*
000500 PROGRAM-ID.  STUD.
000600/
000700 ENVIRONMENT DIVISION.
000800*-----
000900 CONFIGURATION SECTION.
001000 SPECIAL-NAMES.
001100 INPUT-OUTPUT SECTION.
001200 FILE-CONTROL.
001300     SELECT STUD-FILE
001400     ASSIGN TO STUDDS
001500     ORGANIZATION IS INDEXED
001600     ACCESS IS DYNAMIC
001700     RECORD KEY IS MATNR
001800     FILE STATUS IS FSTAT-CODE VSAM-CODE.
001810     SELECT MATNR-IN
001820     ASSIGN TO SYSIN
001830     ORGANIZATION IS SEQUENTIAL
001840     FILE STATUS IS MATNR-IN-CODE.
001900 DATA DIVISION.
002000*-----
002100 FILE SECTION.
002200 FD STUD-FILE
002300     RECORD CONTAINS 55 CHARACTERS.
002400     01 STUDENT-RECORD.
002500         05 MATNR          PIC X(7).
002600         05 STUDSERV       PIC X(8).
002700         05 NNAME          PIC X(20).
002800         05 VNAME          PIC X(20).
002810 FD MATNR-IN
002820     RECORDING MODE F
002830     BLOCK 0 RECORDS
002840     RECORD 80 CHARACTERS
002850     LABEL RECORD STANDARD.
002860     01 MATNR-RECORD    PIC X(80).
002900/
003000 WORKING-STORAGE SECTION.
003100 01 STATUS-AREA.
003200     05 FSTAT-CODE PIC X(2).
003300     88 I-O-OKAY  VALUE ZEROES.
003310     05 MATNR-IN-CODE PIC X(2).
003400     05 VSAM-CODE.
003500     10 VSAM-R15-RETURN-CODE    PIC 9(2) COMP.
003600     10 VSAM-FUNCTION-CODE      PIC 9(1) COMP.
003700     10 VSAM-FEEDBACK-CODE      PIC 9(3) COMP.
003800 01 WS-STUDENT.

```

```

003900    05 WS-MATNR      PIC X(7) .
004000    05 WS-STUDSERV  PIC X(8) .
004100    05 WS-NNAME     PIC X(20) .
004200    05 WS-VNAME     PIC X(20) .
004210    01 WS-MATNR-IN-RECORD .
004220    05 WS-MATNR-IN  PIC X(7) .
004230    05 FILLER      PIC X(73) .
004300 /
004400    PROCEDURE DIVISION .
004500        OPEN INPUT MATNR-IN .
004600        READ MATNR-IN INTO WS-MATNR-IN-RECORD .
004700        DISPLAY "SUCHE " WS-MATNR-IN .
004800        OPEN INPUT STUD-FILE .
004900        IF FSTAT-CODE NOT = "00"
005000            DISPLAY "OPEN INPUT VSAM FILE FS-CODE: " FSTAT-CODE
005100            PERFORM VSAM-CODE-DISPLAY
005200            STOP RUN
005300        END-IF .
005310        MOVE WS-MATNR-IN TO MATNR .
005400        READ STUD-FILE RECORD KEY IS MATNR .
005500        IF FSTAT-CODE NOT = "00" AND FSTAT-CODE NOT = "02"
005600            DISPLAY "READ STUD-FILE FS-CODE: " FSTAT-CODE
005700            PERFORM VSAM-CODE-DISPLAY
005800            STOP RUN
005900        END-IF .
006000        MOVE MATNR TO WS-MATNR .
006100        MOVE STUDSERV TO WS-STUDSERV .
006200        MOVE NNAME TO WS-NNAME .
006300        MOVE VNAME TO WS-VNAME .
006310        CLOSE STUD-FILE .
006320        CLOSE MATNR-IN .
006400        DISPLAY "MATNR:      " WS-MATNR .
006500        DISPLAY "STUDSERV:   " WS-STUDSERV .
006600        DISPLAY "NACHNAME:    " WS-NNAME .
006700        DISPLAY "VORNAME:     " WS-VNAME .
006800        STOP RUN .
006810
006900    VSAM-CODE-DISPLAY .
007000        DISPLAY "VSAM CODE -->"
007100            " RETURN: " VSAM-R15-RETURN-CODE ,
007200            " COMPONENT: " VSAM-FUNCTION-CODE ,
007300            " REASON: " VSAM-FEEDBACK-CODE .

```

**Aufgabe:** Erstellen Sie einen weiteren Partitioned Dataset <Ihre User-ID>.VSAM.LOAD, der zukünftig alle ausführbaren Programme aufnehmen soll (Record format "U").

**Aufgabe:** Um das Cobol-Programm zu compilieren, existiert ein Member COMPILE (Abbildung 4). Legen Sie auch diesen in <Ihre User-ID>.VSAM.CNTL an, modifizieren Sie ihn und wenden Sie anschließend SUB auf ihn an.

```
//PRAKT20C JOB ( ),NOTIFY=&SYSUID
//*
/* COMPILIEREN DES COBOL-PROGRAMMS ZUM VSAM-ZUGRIFF
/*
//COBCOMP EXEC IGYWCL
//COBOL.SYSIN DD DSN=PRAKT20.VSAM.COBOL(STUD),DISP=SHR
//LKED.SYSLMOD DD DSN=PRAKT20.VSAM.LOAD,DISP=SHR
//LKED.SYSIN DD *
NAME STUD(R)
/*
```

Abbildung 4: JCL-Script zum Compilieren des Cobol-Programms

Um das gerade compilierte Cobol-Programm ausführen zu können, ist ein weiterer Member RUN erforderlich (Abbildung 5).

Der „STUDDS“-DD-Entry stellt die Verbindung für das COBOL-Programm dar, wo die Daten abgelegt sind. Über SYSIN übergeben wir die Matrikelnummer des Datasets, der gesucht werden soll.

Aufgabe: Erstellen Sie im Dataset <Ihre User-ID>.VSAM.CNTL den Member RUN. Passen Sie ihn wieder an Ihre Bedürfnisse an.

Aufgabe: Schreiben Sie in den Member RUN die Matrikel-Nummer hinein, nach der im VSAM-Dataset gesucht werden soll (Abbildung 5, hellgrün umrahmte Zahl). Starten Sie die Programmausführung mittels SUB.

```
//PRAKT20R JOB ( ),NOTIFY=&SYSUID
//*
/* AUSFUEHREN DES COBOL-PROGRAMMS ZUM VSAM-ZUGRIFF
/*
//RUN EXEC PGM=STUD
//STEPLIB DD DSN=PRAKT20.VSAM.LOAD,DISP=SHR
//STUDDS DD DSN=PRAKT20.VSAM.STUDENT,DISP=SHR
//SYSIN DD *
1207864
/*
```

Abbildung 5: Member RUN

Nach dem submit steht nun im Job Log der Eintrag des Studenten mit allen zu ihm vorhandenen Daten (Abbildung 6).

```
SUCHE 1207864
MATNR: 1207864
STUDSERV: mai03wer
NACHNAME: Friedrich
VORNAME: Boris
```

Abbildung 6

Wie können wir uns nun die in der Abbildung 6 gezeigte Ausgabe unseres Cobol-Programms anschauen? Dafür starten wir SDSF und rufen den Job Status auf (ST).

Nachdem der Job gefunden wurde, schauen wir uns entweder das Job Log durch das Kommando "S" direkt an oder wir geben statt "S" ein "?" ein.

Nun werden alle Ausgaben nach DD Cards gruppiert angezeigt. Wir interessieren uns nur für SYSOUT im Step RUN und schreiben vor diese Zeile ein „S“, um den Inhalt anzuzeigen.

Weitere Details dazu können Sie unserem Tutorial "Fehlersuche in OS/390 und OS/390-Anwendungen" entnehmen.

Aufgabe: Sehen Sie sich das Suchergebnis wie oben beschrieben an.

Aufgabe: Suchen Sie nach Ihrer eigenen Matrikel-Nummer. Sehen Sie sich auch dieses Suchergebnis an. Fertigen Sie von diesem einen Screenshot entsprechend der Abbildung 7 an; dieser muß mindestens die rot umrandeten Zeilen enthalten. Der Screenshot soll im JPG-Format erstellt werden und darf 90 KByte nicht überschreiten.

Aufgabe: Schicken Sie diesen sowie einen später anzufertigenden zweiten Screenshot unverpackt per E-Mail an Ihren Tutor, und zwar genau eine Abgabe-E-Mail mit je einem Anhang pro Screenshot.

```

. . . . .
  Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY PRAKT20R JOB04593  DSID      4 LINE 9      COLUMNS 02- 81
COMMAND INPUT ===>                                SCROLL ===> 1
IEF142I PRAKT20R RUN - STEP WAS EXECUTED - COND CODE 0000
IGD104I PRAKT20.VSAM.LOAD                          RETAINED,  DDNAME=STEPLIB
IGD104I PRAKT20.VSAM.STUDENT                       RETAINED,  DDNAME=STUDDS
IEF285I  PRAKT20.PRAKT20R.JOB04593.D0000101.?      SYSIN
IEF285I  PRAKT20.PRAKT20R.JOB04593.D0000102.?      SYSIN
IEF373I STEP/RUN      /START 2006017.1928
IEF374I STEP/RUN      /STOP 2006017.1928 CPU        0MIN 00.03SEC SRB      0MIN 00.00S
IEF375I JOB/PRAKT20R/START 2006017.1928
IEF376I JOB/PRAKT20R/STOP 2006017.1928 CPU        0MIN 00.03SEC SRB      0MIN 00.00S
SUCHE 1187579
MATNR:   1187579
STUDSERV: mai91jhz
NACHNAME: Michaelsen
VORNAME: Nils
***** BOTTOM OF DATA *****

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

```

Abbildung 7: Ergebnis der Suche nach dem eigenen Namen

### Erweitern um alternativen Index

Um nun auch über das Studserv-Kürzel auf einen Record zugreifen zu können benötigt man einen alternativen Index. Der alternative Index hat seine eigene Kopie der Records und wird über einen PATH mit dem VSAM-Cluster verbunden.

Einen alternativen Index kann man wieder mit einem geeigneten JCL-Script definieren. Es folgt ein geeigneter Programmcode.

```
//PRAKT20D JOB ( ),NOTIFY=&SYSUID
//*
//* DEFINE VSAM ALTERNATE INDEX
//*
//DEFCLS EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD SYSOUT=A
//SYSIN DD *

DEFINE ALTERNATEINDEX ( -
    NAME(PRAKT20.VSAM.STUDENT.ALTINDEX) -
    RELATE(PRAKT20.VSAM.STUDENT) -
    VOL(SMS009) -
    RECORDSIZE(55 55) -
    KILOBYTES(100 100) -
    KEYS(8 7) -
    UPGRADE )

DEFINE PATH ( -
    NAME(PRAKT20.VSAM.STUDENT.PATH) -
    PATHENTRY(PRAKT20.VSAM.STUDENT.ALTINDEX) )

BLDINDEX INDATASET(PRAKT20.VSAM.STUDENT) -
    OUTDATASET(PRAKT20.VSAM.STUDENT.ALTINDEX) -
    SORTCALL
/*
```

**Aufgabe:** Legen Sie ein entsprechendes JCL-Script als Member DEFAIX im PDS „PRAKT20.VSAM.CNTL“ an.

**Aufgabe:** Führen Sie dieses JCL-Script aus und kontrollieren Sie, ob der alternative Index tatsächlich angelegt wurde. Der VSAM-Cluster müßte unter anderem um weitere Komponenten PRAK<xxx>.VSAM.STUDENT.ALTINDEX.\* erweitert worden sein.

Dieser JCL übernimmt verschiedene Aufgaben. Zuerst wird der alternative Index angelegt, der Key hat 8 Zeichen Länge und beginnt bei Offset 7.

Damit der alternative Index funktioniert, muss ein PATH angelegt werden.

Zuletzt muss der alternative Index erstmalig erstellt werden. Das Wiederholen dieses Schrittes ist durch die Angabe von UPGRADE beim Definieren des alternativen Indexes nicht nötig.

**Aufgabe:** Kopieren Sie den Cobol-Quelltext in einen neuen Member STUD2.

**Aufgabe:** Modifizieren Sie den Member STUD2 entsprechend der nachfolgenden Liste.

Hier die Liste der Änderungen in STUD2 (- heißt weg, + heißt hinzu):

```

      ORGANIZATION IS INDEXED
      ACCESS IS DYNAMIC
      RECORD KEY IS MATNR
+     ALTERNATE RECORD KEY IS STUDSERV
      FILE STATUS IS FSTAT-CODE VSAM-CODE.
-     SELECT MATNR-IN
+     SELECT STUDSERV-IN
      ASSIGN TO SYSIN
      ORGANIZATION IS SEQUENTIAL
-     FILE STATUS IS MATNR-IN-CODE.
+     FILE STATUS IS STUDSERV-IN-CODE.
DATA DIVISION.
-----
```

```

FILE SECTION.
@@ -30,18 +31,18 @@
    05 STUDSERV    PIC X(8).
    05 NNAME      PIC X(20).
    05 VNAME      PIC X(20).
-   FD MATNR-IN
+   FD STUDSERV-IN
        RECORDING MODE F
        BLOCK 0 RECORDS
        RECORD 80 CHARACTERS
        LABEL RECORD STANDARD.
-   01 MATNR-RECORD PIC X(80).
+   01 STUDSERV-RECORD PIC X(80).

WORKING-STORAGE SECTION.
01 STATUS-AREA.
    05 FSTAT-CODE PIC X(2).
    88 I-O-OKAY  VALUE ZEROES.
-   05 MATNR-IN-CODE PIC X(2).
+   05 STUDSERV-IN-CODE PIC X(2).
    05 VSAM-CODE.
        10 VSAM-R15-RETURN-CODE PIC 9(2) COMP.
        10 VSAM-FUNCTION-CODE PIC 9(1) COMP.
@@ -51,22 +52,22 @@
    05 WS-STUDSERV PIC X(8).
    05 WS-NNAME    PIC X(20).
    05 WS-VNAME    PIC X(20).
-   01 WS-MATNR-IN-RECORD.
-   05 WS-MATNR-IN PIC X(7).
-   05 FILLER      PIC X(73).
+   01 WS-STUDSERV-IN-RECORD.
+   05 WS-STUDSERV-IN PIC X(8).
+   05 FILLER      PIC X(72).

PROCEDURE DIVISION.
-   OPEN INPUT MATNR-IN.
-   READ MATNR-IN INTO WS-MATNR-IN-RECORD.
-   DISPLAY "SUCHE " WS-MATNR-IN.
+   OPEN INPUT STUDSERV-IN.
+   READ STUDSERV-IN INTO WS-STUDSERV-IN-RECORD.
+   DISPLAY "SUCHE " WS-STUDSERV-IN.
    OPEN INPUT STUD-FILE.
    IF FSTAT-CODE NOT = "00"
        DISPLAY "OPEN INPUT VSAM FILE FS-CODE: " FSTAT-CODE
        PERFORM VSAM-CODE-DISPLAY
        STOP RUN
    END-IF.
-   MOVE WS-MATNR-IN TO MATNR.
-   READ STUD-FILE RECORD KEY IS MATNR.
+   MOVE WS-STUDSERV-IN TO STUDSERV.
+   READ STUD-FILE RECORD KEY IS STUDSERV.
    IF FSTAT-CODE NOT = "00" AND FSTAT-CODE NOT = "02"
        DISPLAY "READ STUD-FILE FS-CODE: " FSTAT-CODE
        PERFORM VSAM-CODE-DISPLAY
@@ -77,7 +78,7 @@
    MOVE NNAME TO WS-NNAME.
    MOVE VNAME TO WS-VNAME.
    CLOSE STUD-FILE.
-   CLOSE MATNR-IN.
+   CLOSE STUDSERV-IN.
    DISPLAY "MATNR: " WS-MATNR.
    DISPLAY "STUDSERV: " WS-STUDSERV.
    DISPLAY "NACHNAME: " WS-NNAME.

```

Die Liste der Änderungen im Cobol-Programm STUD sollte helfen, alle Änderungen einzubauen.

Es wird nun ein JCL-Script als Hilfsmittel benötigt, um das neue COBOL-Programm zu compilieren.

Aufgabe: Kopieren Sie den Member COMPILE im PDS „PRAKT20.VSAM.CNTL“ nach COMP2 und passen Sie diesen so an, dass Sie ihn zum Compilieren des neuen Cobol-Programms einsetzen können. Diese Anpassung muß bewirken, dass der alte ausführbare Member "STUD" in "PRAKT20.VSAM.LOAD" nicht überschrieben wird. Statt dessen soll hier ein zweiter Member "STUD2" angelegt werden.

Aufgabe: Compilieren Sie das neue Cobol-Programm.

Nun bleibt, das gerade compilierte neue Cobol-Programm auch auszuführen. Dazu kopieren wir den Member RUN nach RUN2 und passen ihn an. Bei einem Vergleich von RUN mit RUN2 (Abbildung 8) fällt auf: In RUN2 muss noch zusätzlich etwas eingefügt werden, um den alternativen Index bekannt zu machen (über den erstellten PATH).

Nach Ausführung findet man die Daten des Studenten wie beim vorigen Cobol-Programm im Job Log des Jobs.

Aufgabe: Erstellen Sie einen letzten neuen Member RUN2 in PRAKT20.VSAM.CNTL (siehe Abbildung 8) und führen Sie mit dessen Hilfe das zweite Cobol-Programm zu Test-Zwecken mehrfach aus.

Aufgabe: Suchen Sie wieder nach Ihrem eigenen Datensatz, diesmal über Ihr eigenes Studserv-Kürzel. Fertigen Sie nach dem Muster von Abbildung 9 einen zweiten Screenshot an. Auch dieser soll im JPG-Format erstellt werden und darf 90 KByte nicht überschreiten und muss mindestens die rot umrandeten Zeilen enthalten.

Aufgabe: Schicken Sie nun beide Screenshots unverpackt und im JPG-Format an Ihren Tutor, und zwar in genau einer E-Mail mit je einem Anhang pro Screenshot.

```
//PRAKT20R JOB ( ),NOTIFY=&SYSUID
//*
//* AUSFUEHREN DES COBOL-PROGRAMMS ZUM VSAM-ZUGRIFF
//* UEBER ALTERNATIVEN INDEX (STUDSERV-KUERZEL)
//*
//RUN          EXEC PGM=STUD2
//STEPLIB     DD DSN=PRAKT20.VSAM.LOAD,DISP=SHR
//STUDDS      DD DSN=PRAKT20.VSAM.STUDENT,DISP=SHR
//STUDDS1     DD DSN=PRAKT20.VSAM.STUDENT.PATH,DISP=SHR
//SYSIN       DD *
mai00aaa
/*
```

**Abbildung 8: JCL-Script zum Ausführen des zweiten Cobol-Programms**

```

. . . . .
Display Filter View Print Options Help
-----
SDSF OUTPUT DISPLAY PRAKT20R JOB04622 DSID      4 LINE 10      COLUMNS 02- 81
COMMAND INPUT ==>                                SCROLL ==> 1
IEF142I PRAKT20R RUN - STEP WAS EXECUTED - COND CODE 0000
IGD104I PRAKT20.VSAM.LOAD                          RETAINED, DDNAME=STEPLIB
IGD104I PRAKT20.VSAM.STUDENT                       RETAINED, DDNAME=STUDDS
IGD104I PRAKT20.VSAM.STUDENT.PATH                 RETAINED, DDNAME=STUDDS1
IEF285I PRAKT20.PRAKT20R.JOB04622.D0000101.?      SYSIN
IEF285I PRAKT20.PRAKT20R.JOB04622.D0000102.?      SYSIN
IEF373I STEP/RUN /START 2006018.1558
IEF374I STEP/RUN /STOP 2006018.1558 CPU           0MIN 00.04SEC SRB      0MIN 00.00S
IEF375I JOB/PRAKT20R/START 2006018.1558
IEF376I JOB/PRAKT20R/STOP 2006018.1558 CPU           0MIN 00.04SEC SRB      0MIN 00.00S
SUCHE mai91jhz
MATNR: 1187579
STUDSERV: mai91jhz
NACHNAME: Michaelsen
VORNAME: Nils
***** BOTTOM OF DATA *****

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

```

Abbildung 9: Ergebnis der Suche über das Studserv-Kürzel