Tutorial 5

Datenbankzugriff mit CICS (ASSEMBLER)

Copyright © Institut für Informatik, Universität Leipzig ph v/2010/03

Ziel dieses Tutorials ist ein Zugriff auf die im Tutorial 4 angelegte DB2-Datenbank mittels CICS. Die Business-Logik wird in dieser Version des Tutorials 5 in Assembler erstellt. Somit erfordert die Bearbeitung dieses Tutorials eine vorausgegangene zeitnahe Bearbeitung des Tutorials 4. In diesem muß die DB2-Tabelle angelegt werden, auf die hier zugegriffen werden soll.

<u>Aufgabe:</u> Bearbeiten Sie VOLLSTÄNDIG und EXAKT alle Aufgaben, die so wie diese hier in kursiver Schrift formuliert sowie eingerahmt sind.

Weiterführende Erläuterungen finden Sie im Tutorial 5, C-Version. Dieses ist auf der gleichen Web-Site herunterzuladen wie dieses Tutorial. Der Hauptunterschied und der fast einzigste Unterschied zwischen dieser Version und der C-Version des Tutorials 5 besteht in der Sprache der Business-Logik. In der C-Version ist diese C, in dieser Version hier ist diese Assembler. Informationen über CICS finden Sie auch im Buch "Einführung in z/OS und OS/390" (P. Herrmann et. al.), das im Oldenbourg-Verlag erschienen ist. Insbesondere enthält dieses Buch ein Kapitel 7: "Customer Information Control System, CICS".

<u>Aufgabe:</u> Legen Sie den Dataset "PRAKT20.CICSDB2.ASSEM" und - wenn noch nicht vorhanden - die Datasets "PRAKT20.LIB" sowie "PRAKT20.DBRMLIB.DATA" an. Verwenden Sie dazu folgende Parameter:

Space units KILOBYTE	Record format FB
Primary quantity 16	Record length 80
Secondary quantity 1	<i>Block size</i> 320
Directory blocks 2	Data set name type : PDS

Falls einer der beiden Datasets "PRAKT20.LIB" oder "PRAKT20.DBRMLIB.DATA" auf ihrem Account schon vorhanden war, empfiehlt sich das Löschen aller Members sowie ein anschließender Compress dieser Datasets.

1. Mapset und Map

Die Abbildung 1 zeigt ein BMS-Programm. Dieses definiert einen Mapset "A020SET", der nur eine einzige Map "A020MAP" enthält.

File Edit Confirm Menu Utilities Compilers Test Help EDIT PRAKT20.CICSDB2.ASSEM(ASSMAP5) - 01.02 Columns 00001 00072 * * * * * * ==MSG> -Warning- The UNDO command is not available until you change ==MSG> your edit profile using the command RECOVERY ON. 000100 //PRAKT20B JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID, 000200 // REGION=4M 000300 //ASSEM EXEC DFHMAPS, MAPNAME='A020SET', RMODE=24 000400 //COPY.SYSUT1 DD 000500 A020SET DFHMSD TYPE=MAP, MODE=INOUT, LANG=ASM, STORAGE=AUTO, 000600 TIOAPFX=YES 000700 * MENU MAP 000800 A020MAP DFHMDI SIZE=(24,80), CTRL=(PRINT, FREEKB) DFHMDF POS=(9,13),ATTRB=(ASKIP,NORM),LENGTH=20, 000900 INITIAL='VORNAME 001000 001100 DFHMDF POS=(9,34),ATTRB=(ASKIP,NORM),LENGTH=20, 001200 INITIAL='NACHNAME 001300VNAM1DFHMDFPOS=(11,13), ATTRB=(ASKIP, NORM), LENGTH=20001400NNAM1DFHMDFPOS=(11,34), ATTRB=(ASKIP, NORM), LENGTH=20001500VNAM2DFHMDFPOS=(12,13), ATTRB=(ASKIP, NORM), LENGTH=20001700VNAM3DFHMDFPOS=(13,13), ATTRB=(ASKIP, NORM), LENGTH=20 001600 NNAM2 DFHMDF POS=(12,13),ATTRB=(ASKIP,NORM),LENGTH=20 001800 NNAM3 DFHMDF POS=(13,34),ATTRB=(ASKIP,NORM),LENGTH=20 001900VNAM4DFHMDFPOS=(14,13),ATTRB=(ASKIP,NORM),LENGTH=20002000NNAM4DFHMDFPOS=(14,34),ATTRB=(ASKIP,NORM),LENGTH=20002100DFHMSDTYPE=FINAL 002200 END 002300 /* 002400 // Command ===> SUB Scroll ===> PAGE F1=Help F3=Exit F5=Rfind F6=Rchange F12=Cancel

Abbildung 1: BMS-Programm

Wir schreiben das in Abbildung 1 gezeigte JCL-Script. Anschließend geben wir "SUB" auf der Kommandozeile ein (s. Abbildung 1) und warten, bis der Job abgearbeitet wurde. Eine fehlerfreie Abarbeitung wird mit "MAXCC=0" bestätigt.

Danach finden wir in dem Dataset PRAKT20.LIB den Member A020SET. Dieser enthält die Mapset-Definition der Abbildung 1 in Form von Assembler-Code. Ausserdem wird ein Member A020SET in die Bibliothek CICSTS13.CICS.PRAKLOAD gestellt. Hier wird CICS später auf das Member und somit auf den Mapset zugreifen.

2. DB2-Precompile des Assembler-Programms

Wir geben das Assemblerprogramm (Abbildung 3 und Abbildung 4) und anschließend das in Abbildung 2 dargestellte JCL-Script ein.

```
File Edit Confirm Menu Utilities Compilers Test Help
EDIT
      PRAKT20.CICSDB2.ASSEM(ASSDB205) - 01.02
                                     Columns 00001 00072
==MSG> -Warning- The UNDO command is not available until you change
==MSG> your edit profile using the command RECOVERY ON.
000100 //PRAKT20B JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID,
000200 //
               REGION=4M
000300 //PCOMP EXEC DB2ASS
000400 //DBRMLIB DD DSN=&SYSUID..DBRMLIB.DATA(AP020),DISP=SHR
000500 //SYSCIN DD DSN=&SYSUID..CICSDB2.ASSEM(AP0200UT),DISP=SHR
000600 //SYSLIB DD DSN=&SYSUID..CICSDB2.ASSEM,DISP=SHR
000700 //SYSIN DD DSN=&SYSUID..CICSDB2.ASSEM(AP020),DISP=SHR
Command ===> SUB
                                            Scroll ===> PAGE
F1=Help
         F3=Exit
                  F5=Rfind F6=Rchange F12=Cancel
```

Abbildung 2: Job für DB2-Precompile

Die Jobverarbeitung wird durch Eingabe von "SUB" gestartet (Abbildung 2). Nach einigen Sekunden erscheint "MAXCC=0". Nun ist die Abarbeitung des Jobs beendet.

File Edit Confirm Menu Utilities Compilers Test Help EDIT PRAKT20.CICSDB2.ASSEM(AP020) - 01.13 Columns 00001 00072 ==MSG> -Warning- The UNDO command is not available until you change ==MSG> your edit profile using the command RECOVERY ON. 000100 AP020 CSECT 000200 AP020 AMODE 31 000300 AP020 RMODE ANY 000400 SQDWSREG EQU 7 SQDWSREG, SQDWSTOR PICK UP ADRESSE VON SQLDSECT 000500 LA SQDWSREG, SQDWSTOK FICK OF ADALGOL USING SQLDSECT, SQDWSREG ADRESSIERE SQLDSECT MIT REG 7 LA 000600 000700 * SQLDSECT WIRD DURCH DB2-PRECOMPILER AM ENDE HINZUGEFUEGT 000800 EXEC SQL * 000900 DECLARE C1 CURSOR FOR 001000 SELECT VNAME, NNAME FROM PRAKT20.TAB020 001100 001200 EXEC SQL OPEN C1 001300 EXEC SQL FETCH C1 INTO :VORNAME, :NACHNAME 001400 VNAM1I, VORNAME 001500 MVC MVC 001600 NNAM1I, NACHNAME 001700 EXEC SQL FETCH C1 INTO :VORNAME, :NACHNAME MVC VNAM2I,VORNAME 001800 001900 002000 MVC NNAM2I, NACHNAME 002100 EXEC SQL FETCH C1 INTO :VORNAME, :NACHNAME 002200 MVC VNAM3I, VORNAME 002300 002400 MVC NNAM3I, NACHNAME Command ===> Scroll ===> PAGE F5=Rfind F6=Rchange F12=Cancel F1=Help F3=Exit

Abbildung 3: Assembler-Programm (Panel 1/2)

Der Job hat ein DB2-Precompile des Assembler-Programms durchgeführt. Er hat in den Dataset PRAKT20.DBRMLIB.DATA den Member AP020 angelegt.

```
File Edit Confirm Menu Utilities Compilers Test Help
PRAKT20.CICSDB2.ASSEM(AP020) - 01.13
                                             Columns 00001 00072
EDIT
002500
002600
            EXEC SQL FETCH C1 INTO :VORNAME, :NACHNAME
002700
            MVC
                 VNAM4I, VORNAME
            MVC
                 NNAM4I, NACHNAME
002800
002900
003000
           EXEC SQL CLOSE C1
003100
003200
            EXEC CICS SEND MAP('A020MAP')
003300
                 MAPSET('A020SET')
003400
                 ERASE
003500
            В
                ENDROUT
            EXEC SQL INCLUDE SQLCA
003600
003700
            DS
               0F
003800 SQDWSTOR DS
                 (SQLDLEN)C RESERVIERE STORAGE FUER SQLDSECT
003900 VORNAME DS CL20
004000 NACHNAME DS CL20
004100
            COPY A020SET
004200
            DS
                 OН
004300 ENDROUT EQU
004400 * HIER WIRD SQLDSECT VOM DB2-PRECOMPILER ZUGEFUEGT
004500 *
         AN ERSTER STELLE STEHT ALS VOLLWORT DIE LAENGE VON SQLDSECT,
004600 *
        DANACH DIE DSECT.
004700 *
        DESWEGEN UEBERSPRINGEN WIR 8 BYTES: DEN BRANCH UND DAS WORT
        В
004800
                 *+8
004900
            END
                 AP020
Command ===>
                                                 Scroll ===> PAGE
F1=Help F3=Exit
                    F5=Rfind F6=Rchange F12=Cancel
```

Abbildung 4: Assembler-Programm (Panel 2/2)

3. CICS-Precompile, Assembler-Compile, LINKEDIT, BIND, GRANT

Wir benötigen noch ein JCL-Script, das einen CICS-Precompile, Assembler-Compile sowie LINKEDIT, BILD und GRANT durchführt. Dies leistet das in Abbildung 5 und in Abbildung 6 dargestellte Script.

File Edit Confirm Menu Utilities Compilers Test Help EDIT PRAKT20.CICSDB2.ASSEM(ASSSTA05) - 01.05 Columns 00001 00072 ==MSG> -Warning- The UNDO command is not available until you change your edit profile using the command RECOVERY ON. ==MSG> 000100 //PRAKT20S JOB (),CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),NOTIFY=&SYSUID, 000200 // REGION=4M 000400 //STEP1 EXEC ASSCICS 000500 //TRN.SYSIN DD DISP=SHR, DSN=&SYSUID..CICSDB2.ASSEM(AP0200UT) 000600 //ASM.SYSLIB DD 000601 // DD 000602 // DD 000603 // DD 000610 // DD DSN=&SYSUID..LIB,DISP=SHR 000700 //LKED.SYSIN DD * 000701 INCLUDE SYSLIB(DSNCLI) 000710 ENTRY AP020 000800 NAME AP020(R) 000900 /* 002200 //********************** 002300 //* BIND 002400 //******************** 002500 //BIND EXEC PGM=IKJEFT01 002600 //STEPLIB DD DSN=SYS1.DSN.V910.SDSNEXIT,DISP=SHR 002700 // DD DSN=SYS1.DSN.V910.SDSNLOAD,DISP=SHR 002800 //DBRMLIB DD DISP=SHR,DSN=&SYSUID..DBRMLIB.DATA 002900 //SYSPRINT DD SYSOUT=* 003000 //SYSTSPRT DD SYSOUT=* Command ===> SUB Scroll ===> PAGE F3=Exit F5=Rfind F1=Help F6=Rchange F12=Cancel

Abbildung 5: CICS-Precompile, Assembler-Compile, LINKEDIT, BIND, GRANT (Panel 1/2)

Nach der Eingabe dieses Scriptes geben wir wie gewohnt "SUB" ein (s. Abbildung 5) und gedulden uns eine Weile, bis der Job beendet wurde. Eine korrekte Abarbeitung des Jobs signalisiert "MAXCC=4".

File Edit Confirm Menu Utilities Compilers Test Help EDIT PRAKT20.CICSDB2.ASSEM(ASSSTA05) - 01.05 Columns 00001 00072 003100 //SYSUDUMP DD SYSOUT=* 003200 //SYSTSIN DD * 003300 DSN S(D931) BIND PLAN (A020ZGR) MEMBER (AP020) ACTION (REP) RETAIN ISOLATION (CS) 003400 003500 END 003600 /* 003700 //********************** 003800 //* GRANT 003900 //********************* 004000 //GRANT EXEC PGM=IKJEFT01 004100 //STEPLIB DD DSN=SYS1.DSN.V910.SDSNLOAD, DISP=SHR 004200 //SYSPRINT DD SYSOUT=* 004300 //SYSTSPRT DD SYSOUT=* 004400 //SYSUDUMP DD SYSOUT=* 004500 //SYSTSIN DD 004600 DSN S(D931) 004700 RUN PROGRAM(DSNTIAD) PLAN(DSNTIA91) -004800 LIBRARY('SYS1.DSN.V910.RUNLIB.LOAD') 004900 END 005000 /* 005100 //SYSIN * תם 005200 GRANT EXECUTE ON PLAN A020ZGR TO PUBLIC 005300 /* Command ===> Scroll ===> PAGE F3=Exit F5=Rfind F6=Rchange F12=Cancel F1=Help

Abbildung 6: CICS-Precompile, Assembler-Compile, LINKEDIT, BIND, GRANT (Panel 2/2)

Dieser Job legt den Member AP020OUT im Dataset PRAKT20.CICSDB2.ASSEM an. Der Member AP020OUT enthält ein Assembler-Programm, welches das Ergebnis der bisher gelaufenen Precompile-Jobs ist.

Ebenfalls wurde von diesem Job der Member AP020 in der CICS-Library CICSTS13.CICS.PRAKLOAD angelegt. Auf diesen Member wird CICS später zugreifen.

4. Definitionen und Installation unter CICS

Bisher wurden ein Mapset (= Presentation Logic, Member A020SET) sowie ein Programm (= Business Logic, Member AP020) erstellt und unter der CICS-internen Library CICSTS13.CICS.PRAKLOAD abgelegt.

Wir möchten nun über eine vierstellige Transaktions-ID auf beide zugreifen. Wie man das tut, wird im Tutorial 5, C-Version, behandelt. Benutzen Sie den CICS-Teil dieses Dokuments, um die folgenden Aufgaben zu lösen.

Warnung:

Ihr DB2ENTRY ist nur ein einziges Mal von Ihnen installierbar. Deshalb könnte z.B. Ihre zweite Anwendung von "CEDA INSTALL GROUP ..." die Fehlermeldung "INSTALL UNSUCCESSFUL" produzieren. Dieser Fehler kann von Ihnen mangels Ihrer CICS-Zugriffsrechte nicht behoben werden. Informieren Sie deshalb umgehend Ihren Tutor.

<u>Aufgabe:</u>	Erstellen Sie eine Transaktion S <ihre prakt-id="">, die auf eine DB2-Datenbank mit 4 Vor- und Zunamen zugreift. Unter diesen Vor- und Zunamen müssen sich die Namen des Autors / der Autoren der Transaktion S<xxx> befinden. Somit müssen alle Autorennamen auf dem Screen ähnlich Abbildung 7 erscheinen.</xxx></ihre>
	Weiterhin müssen "Tutorial 5 in Assembler", das Datum der Bearbeitung sowie die Transaktions-ID, mit der Ihr Betreuer Ihre Transaktion aufrufen kann, auf diesem Screen erscheinen.
<u>Aufgabe:</u>	Ersetzen Sie in den Bezeichnern für Ihren Mapset, Ihren Map, Ihr Assembler-Programm sowie Ihres Zeigers auf die D2-Datenbank die Ziffern "020" durch die Nummer Ihres Prakt <xx> oder PRAK<xxx>-Accounts. Wenn Sie das nicht tun und mehrere Teilnehmer dieses Tutorial gleichzeitig bearbeiten, kommt es zu sehr unschönen Effekten, wie z.B. dass Sie den Screen von jemand anderem anstelle ihres eigenen als Ergebnisscreen erhalten.</xxx></xx>
<u>Aufgabe:</u>	Erzeugen Sie einen Screenshot (unter Windows durch den Shortcut ALT-Druck) Ihrer Version der Abbildung 7 und schicken Sie diesen ihrem Betreuer per Mail zu. Der Screenshot darf eine Größe von 300 KByte nicht überschreiten, benutzen Sie möglichst das JPG-Format, dass mit Dateigrößen unter 90 KByte auskommt. Löschen Sie nichts von Ihrer Lösung, so dass Ihr Betreuer Ihre Transaktion aufrufen kann.
<u>Aufgabe:</u>	Gehen Sie vom CUSTOMPAC MASTER APPLICATION MENU aus in die System Display and Search Facility. Im erscheinenden SDSF PRIMARY OPTION MENU wählen Sie die Option ST. Löschen Sie alle angezeigten Jobs, die sich in der PRINT-Queue befinden, indem Sie links neben einen jeden Jobnamen "p" (purge) eintragen und anschließend die Eingabetaste (mehrfach) drücken. Einen Job dürfen Sie natürlich nicht löschen: Den, der sich in der EXECUTION-Queue befindet. Denn das ist der Job, unter dem Sie zur Zeit eingeloggt sind.

TUTORIAL 5 IN ASSEMBLER DATUM der Bearbeitung dieses Tutorials: 02.03.2004 TRID, die diesen Screen erzeugt: S020 VORNAME NACHNAME FRANZISKA RICHTER SEBASTIAN SCHULZE KATHRIN LANDMANN FRANK SCHNEIDER

Abbildung 7: Musterscreen zu den Aufgaben