

DB2 for z/OS  
Musterlösungen zu den Übungen

4. Januar 2013

Eine Ausarbeitung von:

**cps4it**

Ralf Seidler • Stromberger Straße 36A • 55411 Bingen  
Fon: +49-6721-992611 • Fax: +49-6721-992613 • Mail: [ralf.seidler@cps4it.de](mailto:ralf.seidler@cps4it.de)  
Internet : <http://www.cps4it.de>  
Steuernummer: 08/220/2497/3, Finanzamt Bingen, Ust-ID : DE214792185

# Inhaltsverzeichnis

<b>1</b>	<b>VORBEREITUNGEN.....</b>	<b>4</b>
	<i>Zum Schmunzeln.....</i>	<i>4</i>
1.1	ANMELDEN UND TEST DER USER-IDENTEN .....	4
	<i>Vorgehensweise.....</i>	<i>4</i>
1.2	BIBLIOTHEK FÜR ÜBUNGSAUFGABEN ERSTELLEN .....	5
	<i>Vorgehensweise.....</i>	<i>5</i>
1.3	ÜBUNGSDATENBANKEN .....	6
1.3.1	<i>Datenbank Lieferant.....</i>	<i>6</i>
1.3.2	<i>Datenbank Teil.....</i>	<i>6</i>
1.3.3	<i>Datenbank Projekt .....</i>	<i>7</i>
1.3.4	<i>Datenbank Auftrag.....</i>	<i>7</i>
<b>2</b>	<b>DATENBANKEN ERSTELLEN .....</b>	<b>8</b>
2.1	TABELLE ERSTELLEN .....	8
	<i>Vorgehensweise.....</i>	<i>8</i>
2.2	INDEX ERSTELLEN .....	9
	<i>Vorgehensweise.....</i>	<i>9</i>
2.3	SYNONYM.....	9
	<i>Vorgehensweise.....</i>	<i>9</i>
<b>3</b>	<b>DATEN IN TABELLEN FINDEN.....</b>	<b>10</b>
3.1	PROJEKTDATEN ALLER PROJEKTE.....	10
	<i>Vorgehensweise.....</i>	<i>10</i>
3.2	PROJEKTDATEN EINER LOKATION.....	10
	<i>Vorgehensweise.....</i>	<i>10</i>
3.3	SORTIEREN .....	11
	<i>Vorgehensweise.....</i>	<i>11</i>
3.4	AUSWAHL.....	11
	<i>Vorgehensweise.....</i>	<i>11</i>
<b>4</b>	<b>JOIN .....</b>	<b>12</b>
4.1	AUFTRÄGE MIT NAMEN DER LIEFERANTEN .....	12
	<i>Vorgehensweise.....</i>	<i>12</i>
4.2	TEILE VON BESTIMMTEN LIEFERANTEN .....	12
	<i>Vorgehensweise.....</i>	<i>12</i>
4.3	ZUSÄTZLICHE ZEILEN .....	13
	<i>Vorgehensweise.....</i>	<i>13</i>
<b>5</b>	<b>FUNKTIONEN.....</b>	<b>14</b>
5.1	STATUS VON LIEFERANTEN .....	14
	<i>Vorgehensweise.....</i>	<i>14</i>
5.2	ANZAHL VON LIEFERANTEN FÜR PROJEKTE .....	14
	<i>Vorgehensweise.....</i>	<i>14</i>
5.3	GRUPPIERUNG VON TEILEN .....	15
	<i>Vorgehensweise.....</i>	<i>15</i>
5.4	ORTE SUCHE.....	15
	<i>Vorgehensweise.....</i>	<i>15</i>
<b>6</b>	<b>ÄNDERN VON TABELLEN (1).....</b>	<b>16</b>
6.1	EIGENE TABELLENINHALTE ERZEUGEN .....	16
	<i>Vorgehensweise.....</i>	<i>16</i>
6.2	ÄNDERN VON ZEILEN .....	16
	<i>Vorgehensweise.....</i>	<i>16</i>
6.3	LÖSCHEN VON ZEILEN.....	16
	<i>Vorgehensweise.....</i>	<i>16</i>

---

6.4	LÖSCHEN VON TABELLENINHALTEN.....	16
	<i>Vorgehensweise</i> .....	16
<b>7</b>	<b>ÄNDERN VON TABELLEN (2).....</b>	<b>17</b>
7.1	EIGENE TABELLENINHALTE ERZEUGEN .....	17
	<i>Vorgehensweise</i> .....	17
7.2	ÄNDERN VON SPALTEN .....	17
	<i>Vorgehensweise</i> .....	17
7.3	LÖSCHEN VON ZEILEN.....	17
	<i>Vorgehensweise</i> .....	17
7.4	EINFÜGEN VON ZEILEN.....	18
	<i>Vorgehensweise</i> .....	18
7.5	ERSTELLEN EINER VIEW .....	18
	<i>Vorgehensweise</i> .....	18
<b>8</b>	<b>SYSTEMTABELLEN.....</b>	<b>19</b>
8.1	SYSTEMTABELLEN AUSLESEN.....	19
	<i>Vorgehensweise</i> .....	19
<b>9</b>	<b>DB2 UTILITIES .....</b>	<b>20</b>
9.1	TABELLE VOLLSTÄNDIG ENTLADEN.....	20
	<i>Vorgehensweise</i> .....	20
9.2	TABELLE TEILWEISE ENTLADEN .....	20
	<i>Vorgehensweise</i> .....	20
9.3	RUNSTATS ERZEUGEN.....	20
	<i>Vorgehensweise</i> .....	20
9.4	DB2-TOOL AUFRUFEN.....	20
	<i>Vorgehensweise</i> .....	20
<b>10</b>	<b>SQL IM ANWENDUNGSPROGRAMM.....</b>	<b>21</b>
10.1	PROGRAMM MIT DB2-BEFEHLEN SCHREIBEN .....	21
	<i>Vorgehensweise</i> .....	21
10.2	TABELLE LIEFERANT LESEN.....	21
	<i>Vorgehensweise</i> .....	21
<b>11</b>	<b>CURSOR-VERARBEITUNG .....</b>	<b>21</b>
11.1	TABELLE LIEFERANT LESEN.....	21
	<i>Vorgehensweise</i> .....	21
11.2	TABELLE LIEFERANT LESEN UND 1 ZEILE VERÄNDERN.....	21
	<i>Vorgehensweise</i> .....	21

# 1 Vorbereitungen

---

## Zum Schmunzeln

"Glück ist: zu begreifen, wie alles zusammenhängt."

Sten Nadolny (\*1942), dt. Schriftsteller

## 1.1 Anmelden und Test der User-Iden

---

Melden Sie sich nach Vorgabe im TSO an.  
Melden Sie sich wieder ab und erneut wieder an.  
Konfigurieren Sie Ihre TSO-Session, wie Sie am besten arbeiten können.  
Testen Sie, ob Sie auf die Dateien des Referenten lesend zugreifen können.

### Vorgehensweise

wie erklärt / bekannt

## 1.2 Bibliothek für Übungsaufgaben erstellen

Erstellen Sie eine PO-Datei, um die Übungsaufgaben dieses Seminars aufnehmen zu können.

Name der Bibliothek	userid.KURS.SPUFIIN
Satzlänge	80 Byte
Satzformat	fest geblockt
Dateigröße	5 Spuren primär, 2 Spuren sekundär
Directory	Platz für 30 Member

Kontrollieren Sie das Ergebnis.

### Vorgehensweise

Definition über ISPF Menü 3.2.A (Allocate new Dataset)

```

Menu  RefList  Utilities  Help
-----
--
                                Allocate New Data Set
Command ==>
+
Data Set Name . . . : RZSR.KURS.SPUFIIN
More:
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 . . . . . TRACK (BLKS, TRKS, CYLS, KB, MB, BYTES
or RECORDS)
Average record unit (M, K, or U)
Primary quantity . . 5 (In above units)
Secondary quantity . 2 (In above units)
Directory blocks . . 8 (Zero for sequential data set) *
Record format . . . . FB
Record length . . . . 80
Block size . . . . .
Data set name type : (LIBRARY, HFS, PDS, or blank) *
(Y Y/MM/DD, YYYY/MM/DD
Expiration date . . . YY.DDD, YYYY.DDD in Julian form

```

---

## 1.3 Übungsdatenbanken

---

### 1.3.1 Datenbank Lieferant

LNR	LNAME	LSTATUS	ORT
L1	NEUMANN	30	BERLIN
L2	SCHMIDT	20	HAMBURG
L3	KRAUSE	30	HAMBURG
L4	MEIER	10	BERLIN
L5	SCHULZ	20	FRANKFURT

LNR ist Primär-Schlüssel

### 1.3.2 Datenbank Teil

TNR	TNAME	FARBE	GEWICHT	ORT
T1	C	BLAU	19	BERLIN
T2	D	GELB	12	HAMBURG
T3	S	ROT	14	STUTTGART
T4	S	BLAU	17	BERLIN
T5	B	ROT	17	HAMBURG
T6	N	BLAU	12	BERLIN

TNR ist Primär-Schlüssel

### 1.3.3 Datenbank Projekt

PNR	PNAME	ORT
P1	S4711	HAMBURG
P2	P0815	STUTTGART
P3	R111	FRANKFURT
P4	C007	FRANKFURT
P5	C1234	BERLIN
P6	TZ1	MAINZ
P7	TX25	BERLIN
P8	P888	MUENCHEN

PNR ist Primär-Schlüssel

### 1.3.4 Datenbank Auftrag

LNR	TNR	PNR	MENGE
L1	T1	P1	300
L1	T1	P4	800
L2	T3	P1	500
L2	T3	P2	300
L2	T3	P3	300
L2	T3	P4	600
L2	T3	P5	700
L2	T3	P6	500
L2	T3	P7	900
L2	T5	P2	200
L3	T3	P1	300
L3	T4	P2	600
L4	T6	P3	400
L4	T6	P7	400
L5	T2	P2	300
L5	T2	P4	100
L5	T5	P5	500
L5	T5	P7	100
L5	T6	P2	200
L5	T1	P4	100
L5	T3	P4	200
L5	T4	P4	800
L5	T5	P4	400
L5	T6	P4	500

Die Werte LNR, TNR und PNR bilden zusammen die eindeutige Identifikation eines Auftrags. Sie stellen die Auftragsnummer dar. Das heißt, dass die Spalten LNR, TNR und PNR zusammen den Primär-Schlüssel bilden.

---

## 2 Datenbanken erstellen

---

### 2.1 Tabelle erstellen

---

Member: SPIN201

Schreiben Sie den SQL, der die Tabelle PROJEKT definiert. Die Projekte werden durch die Projektnummer eindeutig identifiziert. Das zugehörige Feld heißt PNR und ist 5 Stellen alphanumerisch. Von einer anderen Tabelle aus soll auf Ihre Tabelle verwiesen werden. Die von der Projektnummer abhängigen Attribute sind der Name des Projektes mit dem Feldnamen PNAME sowie der Ort mit dem Feldnamen ORT. PNAME ist 8 Stellen lang und alphanumerisch, Ort ist 10 Stellen lang und ebenfalls alphanumerisch.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN201) mit dem Inhalt

```
CREATE TABLE PROJEKT
( PNR          CHAR(05) NOT NULL,
  PNAME        CHAR(08) ,
  ORT          CHAR(10) ,
  PRIMARY KEY  (PNR)
)
IN db-name.ts-name
;
```



## 2.2 Index erstellen

---

Member: SPIN202

Die Tabelle Projekt soll nur eindeutig zu identifizierende Zeilen enthalten. Definieren Sie den notwendigen Index, um die Eindeutigkeit bei Hinzufügen und Ändern zu erzwingen. Der verkürzte Namen des Index soll PNRX sein.

### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN202) mit dem Inhalt

```
CREATE UNIQUE INDEX PNRX
      ON      PROJEKT (PNR)
;
```

## 2.3 Synonym

---

Member: SPIN203

Bilden Sie für die Tabellen LIEFERANT, TEIL, PROJEKT und AUFTRAG des Referenten Synonyme mit den Namen L, T, P, LTP.

### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN203) mit dem Inhalt

```
CREATE SYNONYM L      FOR uid-ref.LIEFERANT
;
CREATE SYNONYM T      FOR uid-ref.TEIL
;
CREATE SYNONYM P      FOR uid-ref.PROJEKT
;
CREATE SYNONYM LTP    FOR uid-ref.AUFTRAG
;
```

---

## 3 Daten in Tabellen finden

---

### 3.1 Projektdaten aller Projekte

---

Member: SPIN301

Erstellen Sie eine Projektliste mit allen Daten der Tabelle PROJEKT. Lösen Sie die Aufgabe auf 2 unterschiedlichen Wegen. Welcher Weg ist für einen Test sinnvoller? Welcher Weg ist für die Aufnahme in ein Programm sinnvoller?

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN301) mit dem Inhalt

```
SELECT PNR, PNAME, ORT
  FROM P
;
oder
SELECT *
  FROM P
;
```

---

### 3.2 Projektdaten einer Lokation

---

Member: SPIN302

Listen Sie alle Daten der Projekte in Berlin auf. Lösen Sie die Aufgabe auf 2 unterschiedlichen Wegen. Welcher Weg ist für einen Test sinnvoller? Welcher Weg ist für die Aufnahme in ein Programm sinnvoller?

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN302) mit dem Inhalt

```
SELECT PNR, PNAME, ORT
  FROM P
 WHERE ORT = 'BERLIN'
;
oder
SELECT *
  FROM P
 WHERE ORT = 'BERLIN'
;
```

---

### 3.3 Sortieren

---

Member: SPIN303

Suchen Sie die Lieferantennummern aller Lieferanten in sortierter Reihenfolge, für die die Projekte P1, P4 oder P6 Teile liefern. Lösen Sie diese Aufgabe auf 2 unterschiedlichen Wegen. Welche ist vorzuziehen?

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN303) mit dem Inhalt

```
SELECT DISTINCT LNR
  FROM LTP
 WHERE PNR IN ('P1', 'P4', 'P6')
 ORDER BY LNR
;
oder
SELECT DISTINCT LNR
  FROM LTP
 WHERE PNR = 'P1'
        OR PNR = 'P4'
        OR PNR = 'P6'
 ORDER BY LNR
;
```

---

### 3.4 Auswahl

---

Member: SPIN304

Geben Sie die Aufträge aus, deren gelieferte Mengen zwischen 400 und 800 liegen. Die Grenzen sind jeweils eingeschlossen.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN304) mit dem Inhalt

```
SELECT LNR, TNR, PNR, MENGE
  FROM LTP
 WHERE MENGE >= 400
        AND MENGE <= 800
;
oder
SELECT LNR, TNR, PNR, MENGE
  FROM LTP
 WHERE MENGE BETWEEN 400 AND 800
;
```

---

## 4 JOIN

---

### 4.1 Aufträge mit Namen der Lieferanten

---

Member: SPIN401

Erstellen Sie eine Liste aller Aufträge mit den Namen der Lieferanten. Lösen Sie die Aufgabe auf 2 unterschiedlichen Wegen.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN401) mit dem Inhalt

```
SELECT L.LNR, LNAME, TNR, PNR, MENGE
      FROM LTP, L
     WHERE LTP.LNR = L.LNR
;
oder:
SELECT L.LNR, LNAME, TNR, PNR, MENGE
      FROM LTP INNER JOIN L
     ON LTP.LNR = L.LNR
;
```

### 4.2 Teile von bestimmten Lieferanten

---

Member: SPIN402

Listen Sie alle Teilenummern der Teile, die von Lieferanten aus Berlin geliefert werden. Lösen Sie die Aufgabe auf 2 unterschiedlichen Wegen.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN402) mit dem Inhalt

```
SELECT DISTINCT TNR
      FROM LTP, L
     WHERE LTP.LNR = L.LNR
           AND      ORT = 'BERLIN'
;
oder:
SELECT DISTINCT TNR
      FROM LTP INNER JOIN L
     ON LTP.LNR = L.LNR
     WHERE      ORT = 'BERLIN'
;
```

### 4.3 zusätzliche Zeilen

---

Member: SPIN403

Listen Sie alle Aufträge mit ihren Projektdaten auf. Außerdem sollen in der Liste die Projektdaten angezeigt werden, für die es keine Aufträge gibt.

#### **Vorgehensweise**

Editieren Member userid.KURS.SPUFIIN(SPIN403) mit dem Inhalt

```
SELECT P.*, LNR, TNR, MENGE
FROM P FULL OUTER JOIN LTP
ON LTP.PNR = P.PNR
;
```

---

## 5 Funktionen

---

### 5.1 Status von Lieferanten

---

Member: SPIN501

Erstellen Sie eine Liste der Lieferantennummern von denjenigen Lieferanten, die einen niedrigeren Status als der Lieferant L1 haben.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN501) mit dem Inhalt

```
SELECT LNR
  FROM L
 WHERE LSTATUS <
        ( SELECT LSTATUS
          FROM L
          WHERE LNR = 'L1'
        )
;
```

---

### 5.2 Anzahl von Lieferanten für Projekte

---

Member: SPIN502

Wie viele Projekte werden von dem Lieferanten L1 beliefert?

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN502) mit dem Inhalt

```
SELECT COUNT (DISTINCT PNR)
  FROM LTP
 WHERE LNR = 'L1'
;
```

### 5.3 Gruppierung von Teilen

---

Member: SPIN503

Erstellen Sie eine Liste mit den Teilen, die für mindestens 1 Projekt geliefert werden. Zu jedem Teil geben Sie die Teilenummer, die Projektnummer und die jeweilige Gesamtmenge an.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN503) mit dem Inhalt

```
SELECT TNR, PNR, SUM(MENGE)
  FROM LTP
 GROUP BY TNR, PNR
;
```

### 5.4 Orte suchen

---

Member: SPIN504

Listen Sie die Namen der Orte in sortierter Reihenfolge auf, in denen wenigstens ein Lieferant, ein Teil oder ein Projekt seinen Ort hat.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN504) mit dem Inhalt

```
SELECT ORT FROM L
 UNION
SELECT ORT FROM T
 UNION
SELECT ORT FROM P
ORDER BY 1
;
```

## 6 Ändern von Tabellen (1)

---

### 6.1 Eigene Tabelleninhalte erzeugen

---

Member: SPIN601

Erstellen Sie in den vorliegenden Tabellen L, T, LT, P jeweils eigene Zeilen. Wählen Sie dazu Schlüssel, die nur Ihnen zugeordnet worden sind.

#### **Vorgehensweise**

Keine Musterlösung

### 6.2 Ändern von Zeilen

---

Member: SPIN602

Spielen Sie mit Ihren Tabelleninhalten, indem Sie die Inhalte der Spalten verändern. Prüfen Sie Ihre Änderungen, ob sie Ihren Erwartungen entsprechen.

#### **Vorgehensweise**

Keine Musterlösung

### 6.3 Löschen von Zeilen

---

Member: SPIN603

Spielen Sie mit Ihren Tabelleninhalten, indem Sie einzelne Tabellenzeilen löschen. Prüfen Sie Ihre Änderungen, ob sie Ihren Erwartungen entsprechen.

#### **Vorgehensweise**

Keine Musterlösung

### 6.4 Löschen von Tabelleninhalten

---

Member: SPIN604

Löschen Sie die von Ihnen erzeugten Tabelleninhalte wieder.

#### **Vorgehensweise**

Keine Musterlösung



---

## 7 Ändern von Tabellen (2)

---

### 7.1 Eigene Tabelleninhalte erzeugen

---

Member: SPIN701

Erstellen Sie in den vorliegenden Tabellen L, T, LT, P jeweils eigene Zeilen. Wählen Sie dazu Schlüssel, die nur Ihnen zugeordnet worden sind. Die Zeilen sollen bis auf die Schlüssel echte Duplikate der ursprünglich vorhandenen sein.

#### Vorgehensweise

Keine Musterlösung

### 7.2 Ändern von Spalten

---

Member: SPIN702

Setzen Sie den Status aller Lieferanten in Berlin auf 40. Prüfen Sie das Ergebnis.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN702) mit dem Inhalt

```
UPDATE LIEFERANT
      SET LSTATUS = 40
      WHERE ORT      = 'BERLIN'
;

```

### 7.3 Löschen von Zeilen

---

Member: SPIN703

Löschen Sie alle Lieferanten in Hamburg, deren Status kleiner als 25 ist. Prüfen Sie das Ergebnis.

#### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN703) mit dem Inhalt

```
DELETE FROM LIEFERANT
      WHERE ORT      = 'HAMBURG'
      AND LSTATUS < 25
;

```

---

## 7.4 Einfügen von Zeilen

---

Member: SPIN704

Der neue Lieferant Schulze in Münster ist mit der Lieferantenummer L10 in die Lieferantentabelle einzutragen. Sein Lieferantenstatus ist nicht bekannt. Lösen Sie die Aufgabe auf 2 unterschiedlichen Wegen.

### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN704) mit dem Inhalt

```
INSERT
  INTO LIEFERANT (LNR, LNAME, ORT)
VALUES
      ('L10', 'SCHULZE', 'MUENSTER'
;
oder:
INSERT
  INTO LIEFERANT (LNR, LNAME, LSTATUS, ORT)
VALUES
      ('L10', 'SCHULZE', NULL, 'MUENSTER'
;
```

---

## 7.5 Erstellen einer View

---

Member: SPIN705

Erstellen Sie die View VAUFTRAG mit den folgenden Spalten aus den Tabellen LIEFERANT, TEIL und AUFTRAG:

LNR, LNAME, TNR, TNAME, MENGE

Die Spaltennamen in der View sollen für LNAME LIEF\_NAME und für TNAME TEIL\_NAME heißen.

### Vorgehensweise

Editieren Member userid.KURS.SPUFIIN(SPIN705) mit dem Inhalt

```
CREATE VIEW VAUFTRAG
      ( LNR, LIEF_NAME, TNR, TEIL_NAME, MENGE)
AS
  SELECT LIEFERANT.LNR, LNAME, TEIL.TNR,
         TNAME, MENGE
     FROM LIEFERANT, TEIL, AUFTRAG
    WHERE LIEFERANT.LNR = AUFTRAG.LNR
      AND AUFTRAG.TNR   = TEIL.TNR
;
SELECT *
  FROM VAUFTRAG
;
```

## 8 Systemtabellen

---

### 8.1 Systemtabellen auslesen

---

Member: SPIN801

Lesen Sie verschiedene Informationen zu den Beispieltabellen aus den Systemtabellen aus. Prüfen Sie, ob die Ergebnisse Ihren Erwartungen entsprechen.

#### **Vorgehensweise**

Keine Musterlösung

## 9 DB2 Utilities

---

### 9.1 Tabelle vollständig entladen

---

Member: SPIN901

Entladen Sie die Tabelle LIEFERANT in eine sequentielle Datei.

#### **Vorgehensweise**

Keine Musterlösung

### 9.2 Tabelle teilweise entladen

---

Member: SPIN902

Entladen Sie Teile der Tabelle LIEFERANT in eine sequentielle Datei.

#### **Vorgehensweise**

Keine Musterlösung

### 9.3 Runstats erzeugen

---

Wie ist es möglich, die Statistik-Informationen zu der Tabelle LIEFERANT zu aktualisieren?

#### **Vorgehensweise**

Keine Musterlösung

### 9.4 DB2-Tool aufrufen

---

Rufen Sie ein DB2-Tool auf, das in Ihrer Firma implementiert ist.

#### **Vorgehensweise**

Keine Musterlösung

## 10 SQL im Anwendungsprogramm

---

### 10.1 Programm mit DB2-Befehlen schreiben

---

Member: SPINA01

Schreiben Sie ein Programm, das fehlerfrei umgewandelt wird und nur SQLCA als DB2-Source enthält.

#### **Vorgehensweise**

Keine Musterlösung

### 10.2 Tabelle Lieferant lesen

---

Member: SPINA02

Schreiben Sie ein Programm, das eine Zeile der Tabelle Lieferant ausliest und anzeigt.

#### **Vorgehensweise**

Keine Musterlösung

## 11 Cursor-Verarbeitung

---

### 11.1 Tabelle Lieferant lesen

---

Member: SPINB01

Schreiben Sie ein Programm, das alle Zeilen der Tabelle Lieferant ausliest und anzeigt.

#### **Vorgehensweise**

Keine Musterlösung

### 11.2 Tabelle Lieferant lesen und 1 Zeile verändern

---

Member: SPINB02

Schreiben Sie ein Programm, das alle Zeilen der Tabelle Lieferant ausliest und anzeigt. Mittendrin soll, wenn eine bestimmte Bedingung (frei wählbar) erfüllt ist, ein UPDATE der aktuellen Zeile erfolgen.

#### **Vorgehensweise**

Keine Musterlösung