

Fremdschlüssel mit optionalen Aktionen

Das Ändern bzw. Löschen eines Primärschlüssels der übergeordneten Tabelle einer Fremdschlüsseldefinition ist standardmäßig nicht möglich, solange noch Verweise in der untergeordneten Tabelle enthalten sind.

Optional kann das Ändern bzw. Löschen von übergeordneten Primärschlüsseln durch Angabe des Ereignisses (UPDATE oder DELETE) und einer festgelegten Reaktion für untergeordnete Daten (SET NULL oder CASCADE) ermöglicht werden.

Zusatzklauseln

```
... [ON UPDATE {SET NULL|CASCADE}]
... [ON DELETE {SET NULL|CASCADE}]
```

Beispiel (Interbase) ON UPDATE SET NULL

```
CREATE TABLE TB1 (SCHLFELD VARCHAR(10) NOT NULL CHECK(SCHLFELD<>''),
                 PRIMARY KEY (SCHLFELD));
```

```
CREATE TABLE TB2 (PRIMSCHL DECIMAL(2,0) NOT NULL CHECK(PRIMSCHL BETWEEN 1 AND 99),
                 SCHLFELD VARCHAR(10) CHECK(SCHLFELD IS NULL OR SCHLFELD<>''),
                 FOREIGN KEY (SCHLFELD) REFERENCES TB1 ON UPDATE SET NULL,
                 PRIMARY KEY(PRIMSCHL));
```

Inhalt Tabelle TB1

SCHLFELD
Vorher
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
1	Vorher
2	
3	Vorher
4	Dummy

Anweisung: UPDATE TB1 SET SCHLFELD='Danach' WHERE SCHLFELD='Vorher';

Inhalt Tabelle TB1

SCHLFELD
Danach
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
1	
2	
3	
4	Dummy

Beispiel (Interbase) ON UPDATE CASCADE

```
CREATE TABLE TB1 (SCHLFELD VARCHAR(10) NOT NULL CHECK(SCHLFELD<>''),
PRIMARY KEY (SCHLFELD));
```

```
CREATE TABLE TB2 (PRIMSCHL DECIMAL(2,0) NOT NULL CHECK(PRIMSCHL BETWEEN 1 AND 99),
SCHLFELD VARCHAR(10) CHECK(SCHLFELD IS NULL OR SCHLFELD<>''),
FOREIGN KEY (SCHLFELD) REFERENCES TB1 ON UPDATE CASCADE,
PRIMARY KEY(PRIMSCHL));
```

Inhalt Tabelle TB1

SCHLFELD
Vorher
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
1	Vorher
2	
3	Vorher
4	Dummy

Anweisung: UPDATE TB1 SET SCHLFELD='Danach' WHERE SCHLFELD='Vorher';

Inhalt Tabelle TB1

SCHLFELD
Danach
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
1	Danach
2	
3	Danach
4	Dummy

Beispiel (Interbase) ON DELETE SET NULL

```
CREATE TABLE TB1 (SCHLFELD VARCHAR(10) NOT NULL CHECK(SCHLFELD<>''),  
PRIMARY KEY (SCHLFELD));
```

```
CREATE TABLE TB2 (PRIMSCHL DECIMAL(2,0) NOT NULL CHECK(PRIMSCHL BETWEEN 1 AND 99),  
SCHLFELD VARCHAR(10) CHECK(SCHLFELD IS NULL OR SCHLFELD<>''),  
FOREIGN KEY (SCHLFELD) REFERENCES TB1 ON DELETE SET NULL,  
PRIMARY KEY(PRIMSCHL));
```

Inhalt Tabelle TB1

SCHLFELD
Vorher
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
1	Vorher
2	
3	Vorher
4	Dummy

Anweisung: DELETE FROM TB1 WHERE SCHLFELD='Vorher';

Inhalt Tabelle TB1

SCHLFELD
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
1	
2	
3	
4	Dummy

Beispiel (Interbase) ON DELETE CASCADE

```
CREATE TABLE TB1 (SCHLFELD VARCHAR(10) NOT NULL CHECK(SCHLFELD<>''),
  PRIMARY KEY (SCHLFELD));
```

```
CREATE TABLE TB2 (PRIMSCHL DECIMAL(2,0) NOT NULL CHECK(PRIMSCHL BETWEEN 1 AND 99),
  SCHLFELD VARCHAR(10) CHECK(SCHLFELD IS NULL OR SCHLFELD<>''),
  FOREIGN KEY (SCHLFELD) REFERENCES TB1 ON DELETE CASCADE,
  PRIMARY KEY(PRIMSCHL));
```

Inhalt Tabelle TB1

SCHLFELD
Vorher
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
1	Vorher
2	
3	Vorher
4	Dummy

Anweisung: DELETE FROM TB1 WHERE SCHLFELD='Vorher';

Inhalt Tabelle TB1

SCHLFELD
Dummy

Inhalt Tabelle TB2

PRIMSCHL	SCHLFELD
2	
4	Dummy

Anmerkungen

Die Klausel **ON UPDATE {SET NULL|CASCADE}** ist in Oracle nicht implementiert.

In Interbase können beide Klauseln beliebig miteinander kombiniert werden.

Beispiel

```
... FOREIGN KEY (SCHLFELD) REFERENCES TB1 ON UPDATE SET NULL ON DELETE CASCADE
```

Die Angabe **SET NULL** ist natürlich nur zulässig, wenn die Spalten des Fremdschlüssels **ohne** den Zusatz **NOT NULL** deklariert sind.

Übungsaufgabe

Löschen Sie die Tabelle ART und ergänzen Sie den auf Tabelle LIE bezogenen Fremdschlüssel mit dem Zusatz ON DELETE SET NULL.

Erzeugen Sie anschließend die Tabelle ART mit der neuen Fremdschlüsseldefinition und fügen Sie die bereits bekannten Beispielsätze in die Tabelle erneut ein. Experimentieren Sie mit Löschanweisungen für Lieferanten, um die Wirkungsweise der neuen Fremdschlüsseldefinition zu überprüfen.

Lösung (Oracle)

DROP TABLE ART;

```
CREATE TABLE ART (ARTIKEL VARCHAR2(10) NOT NULL CHECK(ARTIKEL=UPPER(ARTIKEL)),
    BEZEICHN VARCHAR2(40) NOT NULL,
    HPTLIE NUMBER(6,0),
    MEH VARCHAR2(3) NOT NULL,
    PREIS1 NUMBER(8,2) NOT NULL CHECK(PREIS1>=0),
    PRS2AB DATE CHECK(PRS2AB=TRUNC(PRS2AB) OR PRS2AB IS NULL),
    PREIS2 NUMBER(8,2) NOT NULL CHECK(PREIS2>=0),
    FOREIGN KEY (HPTLIE) REFERENCES LIE ON DELETE SET NULL,
    FOREIGN KEY (MEH) REFERENCES MEH,
    PRIMARY KEY (ARTIKEL));
```

```
INSERT INTO ART VALUES('10.1001','Gartenpumpe',1,'Stk',39.99,TO_DATE('01.06.2006','DD.MM.YYYY'),49.99);
INSERT INTO ART VALUES('10.1016','Hochdruckreiniger',1,'Stk',99.99,NULL,0);
INSERT INTO ART VALUES('25.3282','Holzlasur',2,'Lit',4.98,TO_DATE('01.04.2006','DD.MM.YYYY'),3.98);
INSERT INTO ART VALUES('56.7954','Fliesenkleber',3,'kg',0.4,NULL,0);
INSERT INTO ART VALUES('80.0001','Montageleistungen',NULL,'Std',49.95,NULL,0);
```

Lösung (Interbase)

DROP TABLE ART;

```
CREATE TABLE ART (ARTIKEL VARCHAR(10) NOT NULL CHECK(ARTIKEL<>' ' AND ARTIKEL=UPPER(ARTIKEL)),
    BEZEICHN VARCHAR(40) NOT NULL CHECK(BEZEICHN<>' '),
    HPTLIE DECIMAL(6,0),
    MEH VARCHAR(3) NOT NULL,
    PREIS1 DECIMAL(8,2) NOT NULL CHECK(PREIS1 BETWEEN 0 AND 999999.99),
    PRS2AB TIMESTAMP CHECK(PRS2AB=CAST(PRS2AB AS DATE) OR PRS2AB IS NULL),
    PREIS2 DECIMAL(8,2) NOT NULL CHECK(PREIS2 BETWEEN 0 AND 999999.99),
    FOREIGN KEY (HPTLIE) REFERENCES LIE ON DELETE SET NULL,
    FOREIGN KEY (MEH) REFERENCES MEH,
    PRIMARY KEY (ARTIKEL));
```

```
INSERT INTO ART VALUES('10.1001','Gartenpumpe',1,'stk',39.99,'01-JUN-2006',49.99);
INSERT INTO ART VALUES('10.1016','Hochdruckreiniger',1,'Stk',99.99,NULL,0);
INSERT INTO ART VALUES('25.3282','Holzlasur',2,'Lit',4.98,'01-APR-2006',3.98);
INSERT INTO ART VALUES('56.7954','Fliesenkleber',3,'kg',0.4,NULL,0);
INSERT INTO ART VALUES('80.0001','Montageleistungen',NULL,'Std',49.95,NULL,0);
```

Übungsaufgaben

- Einige Lieferanten sind mit mehreren Niederlassungen bzw. Werken an evtl. unterschiedlichen Orten ansässig. Entwerfen Sie eine Tabelle mit dem Namen LAA, in der solche Anschriften gespeichert werden können.

Die Tabelle soll die folgenden Information beinhalten:

Name	Datentyp	Länge	Kommentar
ADRNR	Numerisch	3	<u>Eindeutige</u> fortlaufende Adressnummer (1..999, Primärindex)
ADRSUCH	Beliebige Zeichen	20	<u>Eindeutiger</u> Suchbegriff bezüglich der Adressen eines Lieferanten (Null nicht zulässig, Grossbuchstaben)
NAME1	Beliebige Zeichen	40	Firmenname-1 (Null nicht zulässig)
NAME2	Beliebige Zeichen	40	Firmenname-2 (Null zulässig)
STRASSE	Beliebige Zeichen	40	Straße und Hausnummer (Null nicht zulässig)
PLZORT	Beliebige Zeichen	40	Postleitzahl und Ort (Null nicht zulässig)

Beim Löschen eines Lieferanten sollen alle zugeordneten Anschriften der Niederlassungen bzw. Werke automatisch entfernt werden. Fügen Sie mindestens zwei Beispielsätze in die neue Tabelle ein.

- Erzeugen Sie eine Sql-Abfrage zur Selektion aller Niederlassungsadressen mit der Lieferantenummer und dem Kurznamen des Lieferanten sowie Nummer, Postleitzahl, Ort und Straße der Niederlassungsadresse. Sortieren Sie das Ergebnis aufsteigend nach Lieferantenummer und Nummer der Niederlassungsadresse.

Ausgabebeispiel:

LIEFNR	KURZNAME	ADRNR	PLZORT	STRASSE
1	WEBER, PADERBORN	1	33604 Bielefeld	Brunnenweg 7
1	WEBER, PADERBORN	2	32051 Herford	Mindener Str. 5

Lösung Aufgabe 1 (Oracle)

```

CREATE TABLE LAA (LIEFNR NUMBER(6,0) NOT NULL,
                  ADRNR  NUMBER(3,0) NOT NULL CHECK(ADRNR>0),
                  ADRSUCH VARCHAR2(20) NOT NULL CHECK(ADRSUCH=UPPER(ADRSUCH)),
                  NAME1  VARCHAR2(40) NOT NULL,
                  NAME2  VARCHAR2(40),
                  STRASSE VARCHAR2(40) NOT NULL,
                  PLZORT  VARCHAR2(40) NOT NULL,
                  FOREIGN KEY (LIEFNR) REFERENCES LIE ON DELETE CASCADE,
                  PRIMARY KEY (LIEFNR,ADRNR));

CREATE UNIQUE INDEX LAA_UNIQUE_1 ON LAA(LIEFNR,ADRSUCH);

INSERT INTO LAA
VALUES(1,1,'BIELEFELD','Weber Maschinenbau GmbH',NULL,'Brunnenweg 7','33604 Bielefeld');
INSERT INTO LAA
VALUES(1,2,'HERFORD','Weber Maschinenbau GmbH','Zentrallager','Mindener Str. 5','32051 Herford');

```

Lösung Aufgabe 1 (Interbase)

```

CREATE TABLE LAA (LIEFNR  DECIMAL(6,0) NOT NULL,
                  ADRNR    DECIMAL(3,0) NOT NULL CHECK(ADRNR BETWEEN 1 AND 999),
                  ADRSUCH  VARCHAR(20) NOT NULL CHECK(ADRSUCH<>' ' AND ADRSUCH=UPPER(ADRSUCH)),
                  NAME1    VARCHAR(40) NOT NULL CHECK(NAME1<>' '),
                  NAME2    VARCHAR(40) CHECK(NAME2 IS NULL OR NAME2<>' '),
                  STRASSE  VARCHAR(40) NOT NULL CHECK(STRASSE<>' '),
                  PLZORT   VARCHAR(40) NOT NULL CHECK(PLZORT<>' '),
                  FOREIGN KEY (LIEFNR) REFERENCES LIE ON DELETE CASCADE,
                  PRIMARY KEY (LIEFNR,ADRNR));

CREATE UNIQUE INDEX LAA_UNIQUE_1 ON LAA(LIEFNR,ADRSUCH);

INSERT INTO LAA
VALUES(1,1,'BIELEFELD','Weber Maschinenbau GmbH',NULL,'Brunnenweg 7','33604 Bielefeld');
INSERT INTO LAA
VALUES(1,2,'HERFORD','Weber Maschinenbau GmbH','Zentrallager','Mindener Str. 5','32051 Herford');

```

Lösung Aufgabe 2

```

SELECT LIE.LIEFNR,LIE.KURZNAME,LAA.ADRNR,LAA.PLZORT,LAA.STRASSE FROM LAA,LIE
WHERE LIE.LIEFNR=LAA.LIEFNR ORDER BY LIE.LIEFNR,LAA.ADRNR;

```

Übungsaufgaben

- Zu den Anschriften der Niederlassungen bzw. der Werke eines Lieferanten sollen bis zu 99 unterschiedliche Ansprechpartner mit Name und Telefonnummer hinterlegt werden können. Die Tabelle soll unter dem Namen LAP angelegt werden und soll die folgenden Information beinhalten:

Name	Datentyp	Länge	Kommentar
PARTNR	Numerisch	2	Eindeutige fortlaufende Partnernummer (1..99, Primärindex)
PARTSUCH	Beliebige Zeichen	20	Eindeutiger Suchbegriff je Lieferant und Niederlassung (Null nicht zulässig, Grossbuchstaben)
VORNAME	Beliebige Zeichen	40	Vorname (Null nicht zulässig)
FAMNAME	Beliebige Zeichen	40	Familienname (Null nicht zulässig)
TELEFON	Beliebige Zeichen	20	Telefonnummer (Null nicht zulässig)

Beim Löschen einer Lieferantenniederlassung (LAA) sollen alle zugeordneten Partner der Niederlassung bzw. des Werkes automatisch entfernt werden. Fügen Sie mindestens vier Beispielsätze in die neue Tabelle ein.

- Erzeugen Sie eine Sql-Abfrage zur Selektion aller Partner sämtlicher Niederlassungen aller Lieferanten. Die Ausgabe soll die Lieferantennummer, den Kurznamen des Lieferanten, Postleitzahl und Ort der Niederlassung, den Familien- und Vornamen sowie die Telefonnummer des Ansprechpartners enthalten. Die Ausgabe soll nach Lieferantennummer, Postleitzahl und Ort der Niederlassung sowie nach Familiennamen der Ansprechpartner sortiert sein.

Ausgabebeispiel:

LIEFNR	KURZNAME	PLZORT	FAMNAME	VORNAME	TELEFON
1	WEBER, PADERBORN	32051 Herford	Kleine	Werner	05221/34187-56
1	WEBER, PADERBORN	32051 Herford	Lorenz	Konrad	05221/34187-54
1	WEBER, PADERBORN	33604 Bielefeld	Kleine	Heinz	0521/3029-12
1	WEBER, PADERBORN	33604 Bielefeld	Krause	Margit	0521/3029-26

Lösung Aufgabe 1 (Oracle)

```

CREATE TABLE LAP (LIEFNR NUMBER(6,0) NOT NULL,
                  ADRNR NUMBER(3,0) NOT NULL,
                  PARTNR NUMBER(2,0) NOT NULL CHECK(PARTNR>0),
                  PARTSUCH VARCHAR2(20) NOT NULL CHECK(PARTSUCH=UPPER(PARTSUCH)),
                  VORNAME VARCHAR2(40) NOT NULL,
                  FAMNAME VARCHAR2(40) NOT NULL,
                  TELEFON VARCHAR2(20) NOT NULL,
                  FOREIGN KEY (LIEFNR,ADRNR) REFERENCES LAA ON DELETE CASCADE,
                  PRIMARY KEY (LIEFNR,ADRNR,PARTNR));

CREATE UNIQUE INDEX LAP_UNIQUE_1 ON LAP(LIEFNR,ADRNR,PARTSUCH);

INSERT INTO LAP VALUES(1,1,1,'KLEINE','Heinz','Kleine','0521/3029-12');
INSERT INTO LAP VALUES(1,1,2,'KRAUSE','Margit','Krause','0521/3029-26');
INSERT INTO LAP VALUES(1,2,1,'KLEINE','werner','Kleine','05221/34187-56');
INSERT INTO LAP VALUES(1,2,2,'LORENZ','Konrad','Lorenz','05221/34187-54');

```

Lösung Aufgabe 1 (Interbase)

```

CREATE TABLE LAP (LIEFNR DECIMAL(6,0) NOT NULL,
                  ADRNR DECIMAL(3,0) NOT NULL,
                  PARTNR DECIMAL(2,0) NOT NULL CHECK(PARTNR BETWEEN 1 AND 99),
                  PARTSUCH VARCHAR(20) NOT NULL CHECK(PARTSUCH<>' ' AND PARTSUCH=UPPER(PARTSUCH)),
                  VORNAME VARCHAR(40) NOT NULL CHECK(VORNAME<>' '),
                  FAMNAME VARCHAR(40) NOT NULL CHECK(FAMNAME<>' '),
                  TELEFON VARCHAR(20) NOT NULL CHECK(TELEFON<>' '),
                  FOREIGN KEY (LIEFNR,ADRNR) REFERENCES LAA ON DELETE CASCADE,
                  PRIMARY KEY (LIEFNR,ADRNR,PARTNR));

CREATE UNIQUE INDEX LAP_UNIQUE_1 ON LAP(LIEFNR,ADRNR,PARTSUCH);

INSERT INTO LAP VALUES(1,1,1,'KLEINE','Heinz','Kleine','0521/3029-12');
INSERT INTO LAP VALUES(1,1,2,'KRAUSE','Margit','Krause','0521/3029-26');
INSERT INTO LAP VALUES(1,2,1,'KLEINE','werner','Kleine','05221/34187-56');
INSERT INTO LAP VALUES(1,2,2,'LORENZ','Konrad','Lorenz','05221/34187-54');

```

Lösung Aufgabe 2

```

SELECT LIE.LIEFNR,LIE.KURZNAME,LAA.PLZORT,LAP.FAMNAME,LAP.VORNAME,LAP.TELEFON FROM LAP,LAA,LIE
WHERE LAA.LIEFNR=LAP.LIEFNR AND LAA.ADRNR=LAP.ADRNR AND LIE.LIEFNR=LAP.LIEFNR
ORDER BY LIE.LIEFNR,LAA.PLZORT,LAP.FAMNAME;

```

Übungsaufgaben

1. Viele Artikel können von mehreren Lieferanten bezogen werden. Die Einkaufspreise je Artikel sind häufig von der Bestellmenge abhängig. Erzeugen Sie eine Tabelle (Name LSP) zur Speicherung lieferantenspezifischer, mengenabhängiger Preise. Preisänderungen zu einem bestimmten zukünftigen Stichtag sollen bereits im Vorfeld hinterlegt werden können (siehe Verkaufspreisinformationen PREIS1,PRS2AB,PREIS2 Tabelle ART).

Beim Löschen eines Lieferanten bzw. eines Artikels sollen die entsprechenden Preisdaten automatisch gelöscht werden. Fügen Sie mindestens 4 Beispielsätze in die neue Tabelle ein.

2. Erzeugen Sie eine Sql-Abfrage zur Selektion sämtlicher Einkaufspreisinformationen mit Artikelnummer und Artikelbezeichnung, Lieferantenummer und Kurznamen des Lieferanten. Sortieren Sie das Ergebnis nach Artikelnummer, Lieferantenummer und Mindestbestellmenge.

Ausgabebeispiel:

ARTIKEL	BEZEICHN	LIEFNR	KURZNAME	MENGEV	PREIS1	PRS2AB	PREIS2
10.1001	Gartenpumpe	1	WEBER, PADERBORN	1	27,95	01.06.2006	32,50
10.1001	Gartenpumpe	1	WEBER, PADERBORN	10	26,50	01.06.2006	31,00
10.1001	Gartenpumpe	1	WEBER, PADERBORN	100	24,50	01.06.2006	29,00
10.1001	Gartenpumpe	4	FOSTERS, BIRMINGHAM	1	42,50	01.07.2006	42,00
10.1016	Hochdruckreiniger	1	WEBER, PADERBORN	1	78,50		
10.1016	Hochdruckreiniger	1	WEBER, PADERBORN	10	75,00		
10.1016	Hochdruckreiniger	4	FOSTERS, BIRMINGHAM	1	79,00	01.07.2006	80,00
10.1016	Hochdruckreiniger	4	FOSTERS, BIRMINGHAM	10	75,00	01.07.2006	76,00
10.1016	Hochdruckreiniger	4	FOSTERS, BIRMINGHAM	100	69,00	01.07.2006	69,90

3. Experimentieren Sie mit Löschanweisungen, um die Wirkungsweise von Fremdschlüsseldefinitionen mit dem Zusatz CASCADE zu überprüfen.

Lösung Aufgabe 1 (Oracle)

```
CREATE TABLE LSP (ARTIKEL VARCHAR2(10) NOT NULL,
LIEFNR NUMBER(6,0) NOT NULL,
MENGEV NUMBER(8,2) NOT NULL CHECK(MENGEV>=0),
PREIS1 NUMBER(8,2) NOT NULL CHECK(PREIS1>=0),
PRS2AB DATE CHECK(PRS2AB=TRUNC(PRS2AB) OR PRS2AB IS NULL),
PREIS2 NUMBER(8,2) NOT NULL CHECK(PREIS2>=0),
FOREIGN KEY (ARTIKEL) REFERENCES ART ON DELETE CASCADE,
FOREIGN KEY (LIEFNR) REFERENCES LIE ON DELETE CASCADE,
PRIMARY KEY (ARTIKEL, LIEFNR, MENGEV));

INSERT INTO LSP VALUES('10.1001',1,1,27.95,TO_DATE('01.06.2006','DD.MM.YYYY'),32.50);
INSERT INTO LSP VALUES('10.1001',1,10,26.5,TO_DATE('01.06.2006','DD.MM.YYYY'),31);
INSERT INTO LSP VALUES('10.1001',1,100,24.5,TO_DATE('01.06.2006','DD.MM.YYYY'),29);
INSERT INTO LSP VALUES('10.1001',4,1,42.5,TO_DATE('01.07.2006','DD.MM.YYYY'),42);
```

Lösung Aufgabe 1 (Interbase)

```
CREATE TABLE LSP (ARTIKEL VARCHAR(10) NOT NULL,
LIEFNR DECIMAL(6,0) NOT NULL,
MENGEV DECIMAL(8,2) NOT NULL CHECK(MENGEV BETWEEN 0 AND 999999.99),
PREIS1 DECIMAL(8,2) NOT NULL CHECK(PREIS1 BETWEEN 0 AND 999999.99),
PRS2AB TIMESTAMP CHECK(PRS2AB=CAST(PRS2AB AS DATE) OR PRS2AB IS NULL),
PREIS2 DECIMAL(8,2) NOT NULL CHECK(PREIS2 BETWEEN 0 AND 999999.99),
FOREIGN KEY (ARTIKEL) REFERENCES ART ON DELETE CASCADE,
FOREIGN KEY (LIEFNR) REFERENCES LIE ON DELETE CASCADE,
PRIMARY KEY (ARTIKEL, LIEFNR, MENGEV));

INSERT INTO LSP VALUES('10.1001',1,1,27.95,'01-JUN-2006',32.50);
INSERT INTO LSP VALUES('10.1001',1,10,26.5,'01-JUN-2006',31);
INSERT INTO LSP VALUES('10.1001',1,100,24.5,'01-JUN-2006',29);
INSERT INTO LSP VALUES('10.1001',4,1,42.5,'01-JUL-2006',42);
```

Lösung Aufgabe 2

```
SELECT ART.ARTIKEL,ART.BEZEICHN,LIE.LIEFNR,LIE.KURZNAME,LSP.MENGEV,LSP.PREIS1,LSP.PRS2AB,LSP.PREIS2
FROM LSP,ART,LIE
WHERE ART.ARTIKEL=LSP.ARTIKEL AND LIE.LIEFNR=LSP.LIEFNR
ORDER BY ART.ARTIKEL,LIE.LIEFNR,LSP.MENGEV;
```

Übungsaufgaben

- Lieferanten erwarten bei Bestellvorgängen häufig die Angabe ihrer eigenen Artikelnummer. Zu diesem Zweck sollen Sie eine Tabelle (Name LAN) entwerfen, in der lieferantenspezifische Artikelnummern gespeichert werden können. Beim Löschen eines Lieferanten oder Artikels sollen evtl. vorhandene lieferantenspezifische Artikelnummern automatisch entfernt werden. Fügen Sie mindestens 4 Beispielsätze in die neue Tabelle ein.
- Erzeugen Sie eine Sql-Abfrage zur Selektion aller lieferantenspezifischen Artikelnummern. Die Ausgabe soll die eigene Artikelnummer und Bezeichnung sowie die Lieferantenummer und dessen Kurznamen beinhalten. Das Ergebnis soll aufsteigend nach eigener Artikelnummer und Lieferantenummer sortiert sein.

Ausgabebeispiel:

ARTIKEL	BEZEICHN	LIEFNR	KURZNAME	LIEARTNR
10.1001	Gartenpumpe	1	WEBER, PADERBORN	A-482.191
10.1001	Gartenpumpe	4	FOSTERS, BIRMINGHAM	128300
10.1016	Hochdruckreiniger	1	WEBER, PADERBORN	B-452.957
10.1016	Hochdruckreiniger	4	FOSTERS, BIRMINGHAM	123853

Lösung Aufgabe 1 (Oracle)

```
CREATE TABLE LAN (ARTIKEL VARCHAR2(10) NOT NULL,  
LIEFNR NUMBER(6,0) NOT NULL,  
LIEARTNR VARCHAR2(20) NOT NULL,  
FOREIGN KEY (ARTIKEL) REFERENCES ART ON DELETE CASCADE,  
FOREIGN KEY (LIEFNR) REFERENCES LIE ON DELETE CASCADE,  
PRIMARY KEY (ARTIKEL, LIEFNR));  
  
INSERT INTO LAN VALUES('10.1001', 1, 'A-482.191');  
INSERT INTO LAN VALUES('10.1001', 4, '128300');  
INSERT INTO LAN VALUES('10.1016', 1, 'B-452.957');  
INSERT INTO LAN VALUES('10.1016', 4, '123853');
```

Lösung Aufgabe 1 (Interbase)

```
CREATE TABLE LAN (ARTIKEL VARCHAR(10) NOT NULL,  
LIEFNR DECIMAL(6,0) NOT NULL,  
LIEARTNR VARCHAR(20) NOT NULL CHECK (LIEARTNR<>''),  
FOREIGN KEY (ARTIKEL) REFERENCES ART ON DELETE CASCADE,  
FOREIGN KEY (LIEFNR) REFERENCES LIE ON DELETE CASCADE,  
PRIMARY KEY (ARTIKEL, LIEFNR));  
  
INSERT INTO LAN VALUES('10.1001', 1, 'A-482.191');  
INSERT INTO LAN VALUES('10.1001', 4, '128300');  
INSERT INTO LAN VALUES('10.1016', 1, 'B-452.957');  
INSERT INTO LAN VALUES('10.1016', 4, '123853');
```

Lösung Aufgabe 2

```
SELECT ART.ARTIKEL, ART.BEZEICHN, LIE.LIEFNR, LIE.KURZNAME, LAN.LIEARTNR  
FROM LAN, ART, LIE  
WHERE ART.ARTIKEL=LAN.ARTIKEL AND LIE.LIEFNR=LAN.LIEFNR  
ORDER BY ART.ARTIKEL, LIE.LIEFNR;
```