

## Referenzielle Integrität

### Ausgangsbasis (Oracle)

```
CREATE TABLE KND (KUNDNR    NUMBER(6,0) NOT NULL CHECK(KUNDNR>=1),
                  KURZNAME  VARCHAR2(20) NOT NULL CHECK(KURZNAME=UPPER(KURZNAME)),
                  PKTOFIBU  NUMBER(5,0) NOT NULL CHECK(PKTOFIBU BETWEEN 10000 AND 69999),
                  LAND      VARCHAR2(3) CHECK(LAND=UPPER(LAND) OR LAND IS NULL),
                  KRELIMIT  NUMBER(10,2) NOT NULL CHECK(KRELIMIT>=0),
                  PRIVAT    VARCHAR2(1) NOT NULL CHECK(PRIVAT IN ('J','N')),
                  ANLDAT    DATE NOT NULL CHECK(ANLDAT=TRUNC(ANLDAT)),
                  PRIMARY KEY (KUNDNR));

CREATE INDEX KND_SEARCH_1 ON KND(KURZNAME);
CREATE UNIQUE INDEX KND_UNIQUE_1 ON KND(PKTOFIBU);
```

### Problematik

Der Inhalt der Spalte Land kann leer sein (NULL) oder aus maximal drei beliebigen Zeichen in Großschreibung bestehen.

Folgende Inhalte werden z.B. von der Datenbank problemlos akzeptiert:

```
XXX
ABC
D
012
!!
[$
```

Wir können natürlich darauf hoffen, dass die Benutzer dieser Tabelle diszipliniert arbeiten und nicht solche schwachsinnigen Länderkennzeichen eingeben. Besser wäre es, die vorhandene CHECK-Bedingung so zu verändern, dass nur bestimmte ausgewählte Länderkennzeichen zulässig sind.

#### Beispiel:

```
CHECK(LAND IN ('GB','F','I','USA','DK') OR LAND IS NULL)
```

Und dann haben wir einen Kunden in den Niederlanden (NL). Hoffentlich wird dann nicht gerade krank oder in Urlaub. Denn wir als Datenbankadministratoren müssen jetzt die CHECK-Bedingung verändern, was Übrigens nicht gerade einfach ist, wenn man nicht vorher die gesamte Kundentabelle löscht und anschließend neu aufbaut.

```
CHECK(LAND IN ('GB','F','I','USA','DK','NL') OR LAND IS NULL)
```

Einige Tage später noch ein neuer Kunde, diesmal aus Spanien (E).

```
CHECK(LAND IN ('GB','F','I','USA','DK','NL','E') OR LAND IS NULL)
```

So jetzt reicht's aber. Kunden aus weiteren Ländern sollen gefälligst woanders kaufen! Besser fürs Geschäft wäre allerdings der Einsatz einer Schlüsseltable, in der wir die bei uns gebräuchlichen Länderkennzeichen hinterlegen.

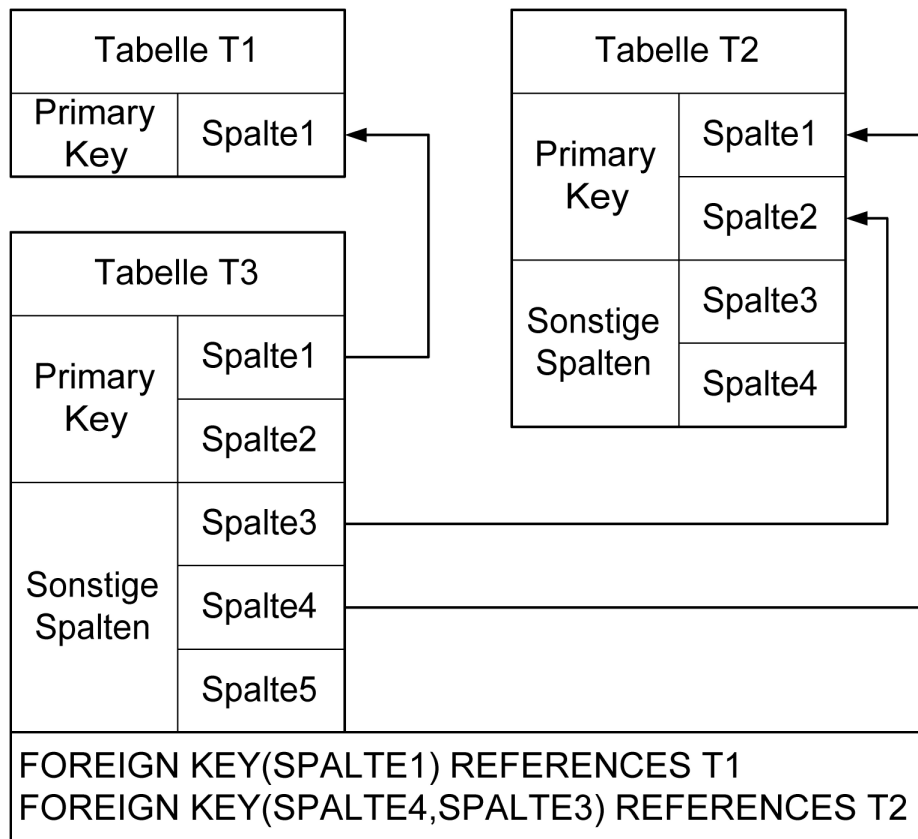
Es müsste dann allerdings sichergestellt sein, dass nur die Länderkennzeichen aus der Schlüsseltable in der Kundentabelle verwendet werden. Außerdem darf natürlich kein Länderkennzeichen aus der Schlüsseltable entfernt oder verändert werden, solange noch diesbezügliche Einträge in der Kundentabelle vorhanden sind.

Dieses Verhalten erreichen wir durch die Definition von Fremdschlüsseln oder Bezugsschlüsseln (FOREIGN KEY).

#### Erweitern wir also die Liste der unterschiedlichen Schlüsselarten:

- PRIMARY KEY (Primärschlüssel bzw. Primärindex)
- INDEX (Suchschlüssel bzw. Suchindex)
- UNIQUE INDEX (Eindeutiger Schlüssel bzw. Index)
- FOREIGN KEY (Fremdschlüssel bzw. Bezugsschlüssel)

## Fremdschlüssel



### Syntax

```
FOREIGN KEY(colname[,colname,...]) REFERENCES tblname;
```

### Besonderheiten

1. Die Anzahl der angegebenen Spalten muss mit der Anzahl der Spalten des Primärschlüssels der referenzierten Tabelle identisch sein.
2. Der Datentyp, die maximale Gesamtlänge und die Anzahl der maximal möglichen Nachkommastellen der angegebenen Spalten muss exakt mit der entsprechenden Spaltendefinition des Primärschlüssels der referenzierten Tabelle übereinstimmen.
3. Ein Bezug wird in Oracle und Interbase nicht hergestellt, falls eine der angegebenen Spalten leer (NULL) ist. Einige Datenbanksysteme stellen nur dann keinen Bezug her, wenn alle angegebenen Spalten leer sind.

### Begriffe

Die referenzierte Tabelle wird auch als übergeordnete Tabelle bezeichnet. Die Tabelle, in der ein Fremdschlüssel definiert ist, bildet dann die untergeordnete Tabelle.

Diese Betrachtung bezieht sich immer auf genau zwei Tabellen. Die untergeordnete Tabelle beinhaltet dann die zu betrachtende Fremdschlüsseldefinition. Die übergeordnete Tabelle ist die in der Klausel REFERENCES benannte Tabelle.

## Schlüsseltabelle für Länderkennzeichen (Oracle)

Zusätzlich zum Länderkennzeichen nehmen wir eine Spalte für die Landbezeichnung auf, die allerdings nur im Bedarfsfall, d.h. bei weniger bekannten Länderschlüsseln, ausgefüllt werden soll.

```
CREATE TABLE LND (LANDSCHL VARCHAR2(3) NOT NULL CHECK(LANDSCHL=UPPER(LANDSCHL)),
                  BEZEICHN VARCHAR2(40),
                  PRIMARY KEY (LANDSCHL));
```

## Änderungen Kundentabelle (Oracle)

```
CREATE TABLE KND (KUNDNR NUMBER(6,0) NOT NULL CHECK(KUNDNR>=1),
  KURZNAME VARCHAR2(20) NOT NULL CHECK(KURZNAME=UPPER(KURZNAME)),
  PKTOFIBU NUMBER(5,0) NOT NULL CHECK(PKTOFIBU BETWEEN 10000 AND 69999),
  LAND VARCHAR2(3),
  KRELIMIT NUMBER(10,2) NOT NULL CHECK(KRELIMIT>=0),
  PRIVAT VARCHAR2(1) NOT NULL CHECK(PRIVAT IN ('J','N')),
  ANLDAT DATE NOT NULL CHECK(ANLDAT=TRUNC(ANLDAT)),
  FOREIGN KEY (LAND) REFERENCES LND,
  PRIMARY KEY (KUNDNR));
```

```
CREATE INDEX KND_SEARCH_1 ON KND(KURZNAME);
CREATE UNIQUE INDEX KND_UNIQUE_1 ON KND(PKTOFIBU);
```

### Hinweise

Die Definition der Spalte LAND kann ohne CHECK-Bedingung erfolgen, da ihr Inhalt entweder nicht angegeben (NULL) ist, oder dem Primärschlüssel eines Satzes der Tabelle LND entsprechen muss.

Wir gehen davon aus, dass Kunden, die in Deutschland ansässig sind, ohne Länderkennzeichen geführt werden.

### Beispielhafter Inhalt der Tabelle LND

LANDSCHL	BEZEICHN
BM	Bermudas
DK	
E	
F	
GB	
GCA	Guatemala
I	
NL	
PNG	Papua Neuguinea
USA	

### Beispielhafter Inhalt der Tabelle KND

KUNDNR	KURZNAME	PKTOFIBU	LAND	KRELIMIT	PRIVAT	ANLDAT
1	ABZOCKER, BERLIN	10001		1000000,00	N	15.04.06
2	MADGIRLS, LONDON	10002	GB		N	15.04.06
3	STUPIDO, ROMA	10003	I	2,50	N	15.04.06
4	FRAUDEUR, PARIS	10004	F	3000,00	N	15.04.06
5	BRITNEY SPEARS, LA	10005	USA	1,00	J	15.04.06
6	BRAD PITT, NY	10006	USA	1,50	J	15.04.06

Hinweis: Die Spalte ANLDAT wurde mit dem Systemdatum des Datenbankservers belegt.

SYSDATE ist eine Systemvariable des Datenbanksystems ORACLE, die das aktuelle Datum und die aktuelle Uhrzeit des Datenbankservers enthält. Durch Einsatz der Funktion TRUNC(SYSDATE) wird die Uhrzeit abgeschnitten.

In Interbase wird statt SYSDATE die Systemvariable 'NOW' benutzt, um Datum und Uhrzeit des Datenbankservers zu erhalten. Der Uhrzeitanteil kann mittels der Funktion CAST('NOW' AS DATE) abgeschnitten werden.

## Übungsaufgabe

Ändern Sie Ihre Scriptdateien UEB2OR.SQL (Oracle) und UEB2IB.SQL (Interbase)

- Fügen Sie die neue Tabellendefinition LND zu Beginn in Ihre Scriptdatei ein.
- Löschen Sie die Tabelle KND und ändern Sie im Anschluss die Definition der Tabelle KND.
- Nehmen Sie, den o.g. Daten entsprechend, INSERT-Anweisungen in Ihre Scriptdatei auf.  
Belegen Sie dabei das Feld ANLDAT der Tabelle KND mit dem aktuellen Datum des Datenbankservers.

Führen Sie die Scriptdateien im jeweiligen Datenbanksystem aus.  
Lassen Sie sich die Inhalte der Tabellen LND und KND anzeigen.

## Musterlösung

### Datei UEB2OR.SQL

```

CREATE TABLE LND (LANDSCHL VARCHAR2(3) NOT NULL CHECK(LANDSCHL=UPPER(LANDSCHL)),
  BEZEICHN VARCHAR2(40),
  PRIMARY KEY (LANDSCHL));

DROP TABLE KND;
CREATE TABLE KND (KUNDNR NUMBER(6,0) NOT NULL CHECK(KUNDNR>=1),
  KURZNAME VARCHAR2(20) NOT NULL CHECK(KURZNAME=UPPER(KURZNAME)),
  PKTOFIBU NUMBER(5,0) NOT NULL CHECK(PKTOFIBU BETWEEN 10000 AND 69999),
  LAND VARCHAR2(3),
  KRELIMIT NUMBER(10,2) NOT NULL CHECK(KRELIMIT>=0),
  PRIVAT VARCHAR2(1) NOT NULL CHECK(PRIVAT IN ('J','N')),
  ANLDAT DATE NOT NULL CHECK(ANLDAT=TRUNC(ANLDAT)),
  FOREIGN KEY (LAND) REFERENCES LND,
  PRIMARY KEY (KUNDNR));

CREATE INDEX KND_SEARCH_1 ON KND(KURZNAME);
CREATE UNIQUE INDEX KND_UNIQUE_1 ON KND(PKTOFIBU);

INSERT INTO LND VALUES('BM','Bermudas');
INSERT INTO LND VALUES('DK',NULL);
INSERT INTO LND VALUES('E',NULL);
INSERT INTO LND VALUES('F',NULL);
INSERT INTO LND VALUES('GB',NULL);
INSERT INTO LND VALUES('GCA','Guatemala');
INSERT INTO LND VALUES('I',NULL);
INSERT INTO LND VALUES('NL',NULL);
INSERT INTO LND VALUES('PNG','Papua Neuguinea');
INSERT INTO LND VALUES('USA',NULL);

INSERT INTO KND VALUES(1,'ABZOCKER, BERLIN',10001,NULL,1000000,'N',TRUNC(SYSDATE));
INSERT INTO KND VALUES(2,'MADGIRLS, LONDON',10002,'GB',0,'N',TRUNC(SYSDATE));
INSERT INTO KND VALUES(3,'STUPIDO, ROMA',10003,'I',2.5,'N',TRUNC(SYSDATE));
INSERT INTO KND VALUES(4,'FRAUDEUR, PARIS',10004,'F',3000,'N',TRUNC(SYSDATE));
INSERT INTO KND VALUES(5,'BRITNEY SPEARS, LA',10005,'USA',1,'J',TRUNC(SYSDATE));
INSERT INTO KND VALUES(6,'BRAD PITT, NY',10006,'USA',1.5,'J',TRUNC(SYSDATE));

COMMIT;

```

### Datei UEB2IB.SQL

```

CREATE TABLE LND (LANDSCHL VARCHAR(3) NOT NULL CHECK(LANDSCHL<>' ' AND LANDSCHL=UPPER(LANDSCHL)),
  BEZEICHN VARCHAR(40) CHECK(BEZEICHN<>' ' OR BEZEICHN IS NULL),
  PRIMARY KEY (LANDSCHL));

DROP TABLE KND;
CREATE TABLE KND (KUNDNR DECIMAL(6,0) NOT NULL CHECK(KUNDNR BETWEEN 1 AND 999999),
  KURZNAME VARCHAR(20) NOT NULL CHECK(KURZNAME<>' ' AND KURZNAME=UPPER(KURZNAME)),
  PKTOFIBU DECIMAL(5,0) NOT NULL CHECK(PKTOFIBU BETWEEN 10000 AND 69999),
  LAND VARCHAR(3),
  KRELIMIT DECIMAL(10,2) NOT NULL CHECK(KRELIMIT BETWEEN 0 AND 99999999.99),
  PRIVAT VARCHAR(1) NOT NULL CHECK(PRIVAT IN ('J','N')),
  ANLDAT TIMESTAMP NOT NULL CHECK(ANLDAT=CAST(ANLDAT AS DATE)),
  FOREIGN KEY (LAND) REFERENCES LND,
  PRIMARY KEY (KUNDNR));

CREATE INDEX KND_SEARCH_1 ON KND(KURZNAME);
CREATE UNIQUE INDEX KND_UNIQUE_1 ON KND(PKTOFIBU);

INSERT INTO LND VALUES('BM','Bermudas');
INSERT INTO LND VALUES('DK',NULL);
INSERT INTO LND VALUES('E',NULL);
INSERT INTO LND VALUES('F',NULL);
INSERT INTO LND VALUES('GB',NULL);
INSERT INTO LND VALUES('GCA','Guatemala');
INSERT INTO LND VALUES('I',NULL);
INSERT INTO LND VALUES('NL',NULL);
INSERT INTO LND VALUES('PNG','Papua Neuguinea');
INSERT INTO LND VALUES('USA',NULL);

INSERT INTO KND VALUES(1,'ABZOCKER, BERLIN',10001,NULL,1000000,'N',CAST('NOW' AS DATE));
INSERT INTO KND VALUES(2,'MADGIRLS, LONDON',10002,'GB',0,'N',CAST('NOW' AS DATE));
INSERT INTO KND VALUES(3,'STUPIDO, ROMA',10003,'I',2.5,'N',CAST('NOW' AS DATE));
INSERT INTO KND VALUES(4,'FRAUDEUR, PARIS',10004,'F',3000,'N',CAST('NOW' AS DATE));
INSERT INTO KND VALUES(5,'BRITNEY SPEARS, LA',10005,'USA',1,'J',CAST('NOW' AS DATE));
INSERT INTO KND VALUES(6,'BRAD PITT, NY',10006,'USA',1.5,'J',CAST('NOW' AS DATE));

COMMIT;

```

## Übungsaufgaben

### Aufgabe-1

Die Tabellen LND und KND enthalten die Daten gemäß den Anweisungen aus vorgenannten Scriptdateien. Warum werden die nachfolgenden Anweisungen vom DBMS als fehlerhaft abgewiesen? Begründung!

**Achtung:** In einer Anweisung können mehrere Fehler enthalten sein!

1	INSERT INTO KND VALUES(7,'Julia Roberts, Atlanta',10006,'USA',200000,'J',TRUNC(SYSDATE));
2	INSERT INTO KND VALUES(8,'ROBBIE WILLIAMS',10008,'UK',400000,'X',SYSDATE);
3	INSERT INTO KND VALUES(9,'RICKIE MARTIN, ',70008,'Bm',10,'J',TRUNC(SYSDATE));
4	INSERT INTO KND VALUES(10.5,'KARL MOIK, INGOLST.',10010,'D',NULL,'j',TRUNC(SYSDATE));
5	UPDATE LND SET LANDSCHL='FR' WHERE LANDSCHL='F';
6	DELETE FROM LND WHERE LAND='GB';
7	UPDATE KND SET LANDSCHL='E ' WHERE LANDSCHL='I';
8	DROP TABLE LND;

### Aufgabe-2

Erzeugen Sie eine Tabelle (LIE) für Lieferanten. Die Tabelle soll weitgehend mit der Kundentabelle identisch sein.

Abweichungen:

- Spaltenname LIEFNR statt KUNDNR
- Wertebereich Personenkonto Finanzbuchhaltung (70000..99999)
- Die Spalte PRIVAT entfällt

LIEFNR	KURZNAME	PKTOFIBU	LAND	KRELIMIT	ANL DAT
1	WEBER, PADERBORN	70001		10000,00	15.04.06
2	MILLER, HAMILTON	70002	BM		15.04.06
3	LAHO, BAGNEUX	70012	F	5000,00	15.04.06
4	FOSTERS, BIRMINGHAM	70004	GB	7500,00	15.04.06

Erzeugen Sie Scriptdateien (UEB3OR.SQL und UEB3IB.SQL) für die Datenbanksysteme Oracle und Interbase. Nehmen Sie die Lieferanten gemäß Beispieldaten in die neue Tabelle auf.

### Aufgabe-3

Erzeugen Sie eine Schlüsseltable für Mengeneinheiten. Eine Mengeneinheit kann maximal drei beliebige Zeichen umfassen (Stk,Kg,m,...). Die Tabelle soll unter dem Namen MEH gespeichert werden. Die Spalte für die Mengeneinheit soll mit MENGEINH benannt werden.

Erzeugen Sie eine Tabelle für Teile bzw. Dienstleistungen unter dem Namen ART.

- ARTIKEL (Max. 10 Zeichen in Großschreibung, Primärschlüssel)
- BEZEICHN (Max. 40 beliebige Zeichen, NULL nicht zulässig)
- HPTLIE (Lieferant gemäß Lieferantentabelle, NULL zulässig)
- MEH (Mengeneinheit gemäß Schlüsseltable, NULL nicht zulässig)
- PREIS1 (Numerisch mit 8 Ziffernstellen, davon 2 Nachkommastellen, negative Werte oder NULL nicht zulässig)
- PRS2AB (Datum ohne Uhrzeit, NULL zulässig)
- PREIS2 (Numerisch mit 8 Ziffernstellen, davon 2 Nachkommastellen, negative Werte oder NULL nicht zulässig)

Erläuterungen:

- Ein Hauptlieferant (HPTLIE) soll nur bei Kaufteilen, nicht bei Erzeugnissen bzw. Leistungen hinterlegt werden.
- Preis2 soll erst ab einem bestimmten Datum (PRS2AB) gültig sein, sofern überhaupt ein Datum angegeben ist.

ARTIKEL	BEZEICHN	HPTLIE	MEH	PREIS1	PRS2AB	PREIS2
10.1001	Gartenpumpe	1	Stk	39,99	01.06.06	49,99
10.1016	Hochdruckreiniger	1	Stk	99,99		
25.3282	Holzlasur	2	Lit	4,98	01.04.06	3,98
56.7954	Fliesenkleber	3	Kg	0,40		
80.0001	Montageleistungen		Std	49,95		

Erzeugen Sie Scriptdateien (UEB4OR.SQL und UEB4IB.SQL) für die Datenbanksysteme Oracle und Interbase. Nehmen Sie die oben genannten Beispieldaten in die Tabellen auf.

## Musterlösungen

### Aufgabe-1

1	KURZNAME zu lang KURZNAME enthält Kleinbuchstaben PKTOFIBU bereits vergeben
2	PRIVAT nicht 'J' oder 'N' Uhrzeit SYSDATE unzulässig Länderschlüssel 'UK' nicht in Tabelle LND gespeichert
3	PKTOFIBU nicht im zulässigen Bereich Länderschlüssel 'Bm' ist ungleich 'BM'
4	KRELIMIT darf nicht NULL sein PRIVAT nicht 'J' oder 'N' Länderschlüssel 'D' nicht in Tabelle LND gespeichert
5	Länderschlüssel 'F' wird in Kundentabelle verwendet
6	Spalte LAND nicht in Tabelle LND vorhanden Länderschlüssel 'GB' wird in Kundentabelle verwendet
7	Spalte LANDSCHL nicht in Tabelle KND vorhanden 'E' ist ungleich 'E'
8	Tabelle LND kann nicht gelöscht werden, solange Verweise in Tabelle KND vorhanden sind

### Aufgabe-2

#### Datei UEB3OR.SQL

```
CREATE TABLE LIE (LIEFNR NUMBER(6,0) NOT NULL CHECK(LIEFNR>=1),
  KURZNAME VARCHAR2(20) NOT NULL CHECK(KURZNAME=UPPER(KURZNAME)),
  PKTOFIBU NUMBER(5,0) NOT NULL CHECK(PKTOFIBU>=70000),
  LAND VARCHAR2(3),
  KRELIMIT NUMBER(10,2) NOT NULL CHECK(KRELIMIT>=0),
  ANLDAT DATE NOT NULL CHECK(ANLDAT=TRUNC(ANLDAT)),
  FOREIGN KEY (LAND) REFERENCES LND,
  PRIMARY KEY (LIEFNR));
```

```
CREATE INDEX LIE_SEARCH_1 ON LIE(KURZNAME);
CREATE UNIQUE INDEX LIE_UNIQUE_1 ON LIE(PKTOFIBU);
```

```
INSERT INTO LIE VALUES(1, 'WEBER, PADERBORN', 70001, NULL, 10000, TRUNC(SYSDATE));
INSERT INTO LIE VALUES(2, 'MILLER, HAMILTON', 70002, 'BM', 0, TRUNC(SYSDATE));
INSERT INTO LIE VALUES(3, 'LAHO, BAGNEUX', 70012, 'F', 5000, TRUNC(SYSDATE));
INSERT INTO LIE VALUES(4, 'FOSTERS, BIRMINGHAM', 70004, 'GB', 7500, TRUNC(SYSDATE));
```

```
COMMIT;
```

#### Datei UEB3IB.SQL

```
CREATE TABLE LIE (LIEFNR DECIMAL(6,0) NOT NULL CHECK(LIEFNR BETWEEN 1 AND 999999),
  KURZNAME VARCHAR(20) NOT NULL CHECK(KURZNAME<>' ' AND KURZNAME=UPPER(KURZNAME)),
  PKTOFIBU DECIMAL(5,0) NOT NULL CHECK(PKTOFIBU BETWEEN 70000 AND 99999),
  LAND VARCHAR(3),
  KRELIMIT DECIMAL(10,2) NOT NULL CHECK(KRELIMIT BETWEEN 0 AND 99999999.99),
  ANLDAT TIMESTAMP NOT NULL CHECK(ANLDAT=CAST(ANLDAT AS DATE)),
  FOREIGN KEY (LAND) REFERENCES LND,
  PRIMARY KEY (LIEFNR));
```

```
CREATE INDEX LIE_SEARCH_1 ON LIE(KURZNAME);
CREATE UNIQUE INDEX LIE_UNIQUE_1 ON LIE(PKTOFIBU);
```

```
INSERT INTO LIE VALUES(1, 'WEBER, PADERBORN', 70001, NULL, 10000, CAST('NOW' AS DATE));
INSERT INTO LIE VALUES(2, 'MILLER, HAMILTON', 70002, 'BM', 0, CAST('NOW' AS DATE));
INSERT INTO LIE VALUES(3, 'LAHO, BAGNEUX', 70012, 'F', 5000, CAST('NOW' AS DATE));
INSERT INTO LIE VALUES(4, 'FOSTERS, BIRMINGHAM', 70004, 'GB', 7500, CAST('NOW' AS DATE));
```

```
COMMIT;
```

### Aufgabe-3

#### Datei UEB4OR.SQL

```

CREATE TABLE MEH (MENGEINH VARCHAR2(3) NOT NULL,
                  PRIMARY KEY (MENGEINH));

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,
                  FOREIGN KEY (MEH) REFERENCES MEH,
                  PRIMARY KEY (ARTIKEL));

INSERT INTO MEH VALUES('Kg');
INSERT INTO MEH VALUES('Lit');
INSERT INTO MEH VALUES('Std');
INSERT INTO MEH VALUES('Stk');

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);

COMMIT;

```

#### Datei UEB4IB.SQL

```

CREATE TABLE MEH (MENGEINH VARCHAR(3) NOT NULL CHECK(MENGEINH<>''),
                  PRIMARY KEY (MENGEINH));

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,
                  FOREIGN KEY (MEH) REFERENCES MEH,
                  PRIMARY KEY (ARTIKEL));

INSERT INTO MEH VALUES('Kg');
INSERT INTO MEH VALUES('Lit');
INSERT INTO MEH VALUES('Std');
INSERT INTO MEH VALUES('Stk');

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);

COMMIT;

```

### Übungsaufgabe

Experimentieren Sie mit allen bisher bekannten Anweisungen und versuchen Sie die Konsistenzmechanismen der Datenbanksysteme bewusst zu unterminieren.