Allgemeines

In einer relationalen Datenbank werden tabellenübergreifende Beziehungen bekanntlich durch Fremdschlüssel definiert. Das Umsetzen einer Aufgabenstellung in ein relationales Gebilde aus Tabellen, Primärschlüsseln und Fremdschlüsseln erfordert ein hohes Abstraktionsvermögen und sollte deshalb in mehreren Phasen erfolgen.

Beispiel

Für eine Pension soll eine Datenbank zur Zimmerbelegung entworfen werden.

Sammeln möglicher Schlüsselattribute

z.B.: Zimmer, Betten, Gäste, Mitreisende, Belegungseinheiten (Tage, Wochen etc.)

Konzentrieren Sie sich auf die Schlüsselattribute!

Relevante Fragestellungen:
- Werden Zimmer oder Betten vermietet?
- Wie werden Reisegruppen, Mitreisende, Familien etc. eingestuft?
- In welchen Zeitintervallen (Stunden, Tage, Wochen etc.) werden die Objekte vermietet?

z.Zt. nicht relevante Fragestellungen:
- Sollen Vorname und Familienname eines Gastes separat gespeichert werden?
- Soll eine Telefon-, Fax- oder Handynummer des Gastes gespeichert werden?
- Welche Eigenschaften definieren ein Zimmer (Balkon, TV, Telefon, Minibar etc.)?

Definieren der übergeordneten Tabellen

Ein Organisationsgespräch mit der Pensionswirtin ergibt folgenden Sachverhalt:
- Vermietet werden ausschließlich Zimmer (keine Betten).
- Jedes Zimmer wird tageweise vermietet.
- Bei Belegung eines Zimmers mit mehreren Personen wird nur eine Person als Kunde (Gast) betrachtet.

Daraus ergeben sich die folgenden übergeordneten Tabellen:

Zimmer

Gäste

Festlegen der Primärschlüssel für die übergeordneten Tabellen

Laut Pensionswirtin wird das bisherige Zimmernummernsystem seit Jahren unverändert angewandt und hat sich bewährt. Die Zimmernummer ist generell dreistellig. Die erste Stelle kennzeichnet die Etage (1..3). Die beiden letzten Stellen kennzeichnen das Zimmer innerhalb der Etage (01..25).

Für die Pensionsgäste gibt es bisher kein eindeutiges Unterscheidungsmerkmal. Mittels Karteikarten werden Name, Anschrift und Telefonnummer schriftlich festgehalten. Dieses System sei fehleranfällig, da z.B. aufgrund von Namensänderungen (Heirat, Scheidung etc.) oder Umzügen mehrere Karten für den selben Gast vorhanden seien. Die Anzahl unterschiedlicher Gäste wird seitens der Pensionswirtin auf ca. 1000 pro Kalenderjahr geschätzt. Im gegenseitigen Einvernehmen wird eine sechsstellige fortlaufende Nummer als Identifikationskriterium für Gäste festgelegt.

Tabelle	Primärschlüssel
Zimmer	Zimmernummer (3 Stellen numerisch)
Gäste	Gastnummer (6 Stellen numerisch)

Definieren der untergeordneten Tabellen

Zur Speicherung der Belegung ist offensichtlich eine Tabelle mit den Spalten
Zimmernummer, Gastnummer und Datum erforderlich.

Erstellen Sie eine Hilfstabelle mit allen möglichen Schlüsselkombinationen:

Kombination	Spalte	Spalte	Spalte
1	Zimmernummer	Datum	Gastnummer
2	Zimmernummer	Datum	
3	Zimmernummer	Gastnummer	
4	Zimmernummer		
5	Datum	Gastnummer	
6	Datum		
7	Gastnummer		

Prüfen Sie die Eindeutigkeit jeder einzelnen Kombination anhand von Beispieldaten (hier Kombination 1):

Zimmernummer	**Datum**	**Gastnummer**
101	15.11.2006	1
101	15.11.2006	2

Kombination 1 ist hier offensichtlich wenig hilfreich, da ein Zimmer am selben Tag mit mehreren Gästen belegt sein kann. Die Gäste werden sich bedanken.

Prüfen Sie die nächste Kombination (hier Kombination 2):

Zimmernummer	**Datum**	Gastnummer
101	15.11.2006	1
101	16.11.2006	1

Die Kombination aus Zimmernummer und Datum ist als möglicher Primärschlüssel anscheinend geeignet. Ein Zimmer kann an mehreren unterschiedlichen Tagen jeweils an genau einen Gast vermietet werden.

Prüfen Sie die verbleibenden Möglichkeiten (Kombination 3 bis 7):

Zimmernummer	Datum	**Gastnummer**
101	15.11.2006	1
101	15.11.2006	2

Ein bestimmtes Zimmer kann an denselben Gast nur für genau einen Tag vermietet werden. Ein täglicher Umzug in ein anderes Zimmer ist unausweichlich. Die Gäste werden dies sicher nicht als besonderen Service betrachten.

Zimmernummer	Datum	Gastnummer
101	15.11.2006	1
102	15.11.2006	1

Ein Zimmer kann genau für einen Tag belegt werden. Davor und danach steht es leer.
Ein solcher Primärschlüssel würde die Pensionswirtin vermutlich in den Wahnsinn treiben.

Zimmernummer	**Datum**	**Gastnummer**
101	15.11.2006	1
101	16.11.2006	1
101	15.11.2006	2
101	16.11.2006	2

Leider ist es bei dieser Konstellation möglich, ein bestimmtes Zimmer am selben Tag an mehrere Gäste zu vergeben. Diese Kombination ist daher als Primärschlüssel ebenfalls nicht geeignet.

Zimmernummer	**Datum**	Gastnummer
101	15.11.2006	1
102	16.11.2006	2

Pro Tag kann leider immer nur ein Zimmer vermietet werden. Die restlichen x Zimmer bleiben zwangsläufig leer. Die Pensionswirtin ist schwer gezeichnet und wird entgültig depressiv. Folglich verlieren wir unseren Auftraggeber.

Zimmernummer	Datum	**Gastnummer**
101	15.11.2006	1
102	15.11.2006	2

Gäste dürfen nur einen Tag bleiben und niemals wiederkommen. Stammgäste sind nicht erwünscht. Und einen mehrtägigen Urlaub gönnen sich ja so oder so nur die Superreichen. Eine solche Lösung ist ein herausragendes Ereignis in Ihrer Karriere als Softwareentwickler(in), denn es beendet diese abrupt.

Fazit: Als Primärschlüssel ist nur die Kombination 2 (Zimmernummer und Datum) sinnvoll.

Reagieren auf neue Erkenntnisse

Ein Telefongespräch mit der Pensionswirtin führt zu neuen Erkenntnissen:

„Einen Punkt habe ich leider bei unserem letzten Gespräch übersehen. Es muss möglich sein, die Belegung eines Zimmers, z.B. für Renovierungsarbeiten, zu sperren. Das ist doch sicher kein Problem für Sie, oder etwa doch?“.

Solche Situationen sind auch bei gründlicher Analyse der Aufgabenstellung eher die Regel als die Ausnahme. Das bisherige Konzept muss hinsichtlich der neuen Anforderungen überprüft werden. Schlimmstenfalls muss das bisherige Konzept gänzlich verworfen und ein neuer konzeptioneller Ansatz gesucht werden.

Dies ist hier aber augenscheinlich nicht der Fall. Wir können die Spalte Gastnummer in der Belegungstabelle so definieren, dass diese auch leer (NULL) sein darf. Ein Eintrag ohne Gastnummer wird somit als Reservierung für interne Zwecke interpretiert.

Beispiel:

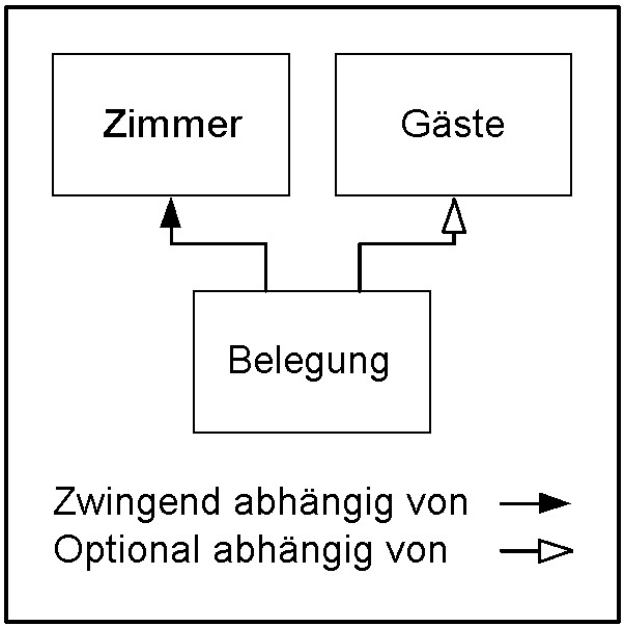
Zimmernummer	**Datum**	Gastnummer
101	15.11.2006	1
101	16.11.2006	1
101	17.11.2006	

Am 15. und 16. November 2006 ist das Zimmer 101 durch den Gast mit der Nummer 1 belegt. Am 17.11.2006 ist das Zimmer 101 für interne Zwecke gesperrt.

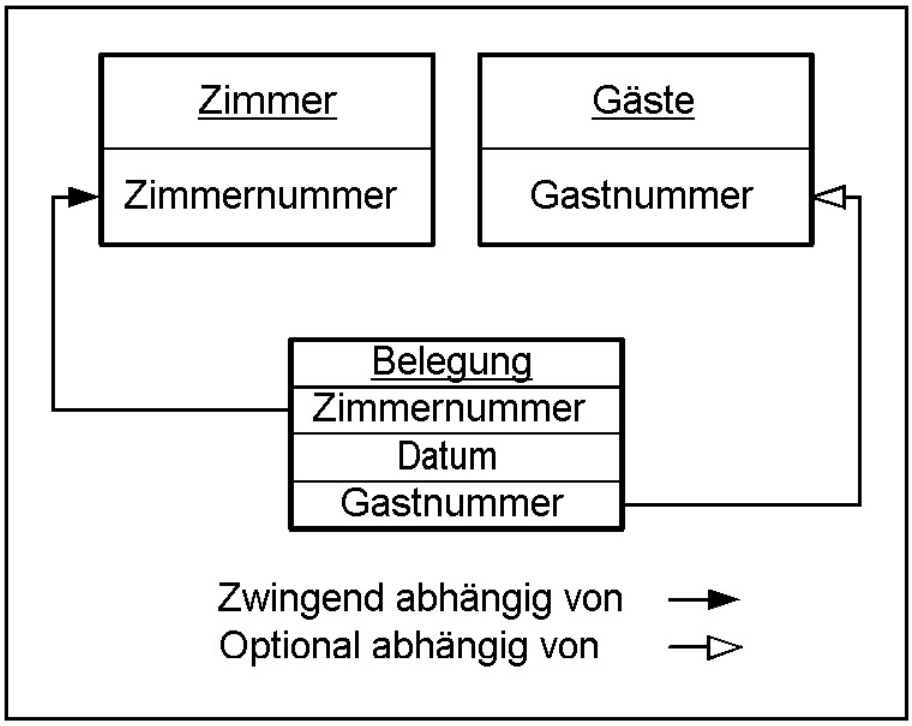
Die Relationen zwischen den Spalten der Belegungstabelle und den Primärschlüsseln der übergeordneten Tabellen Zimmer und Gäste werden als Fremdschlüssel definiert.

Die daraus abzuleitenden Beziehungen sind nachfolgend grafisch dargestellt.

Einfache Darstellung der Beziehungen



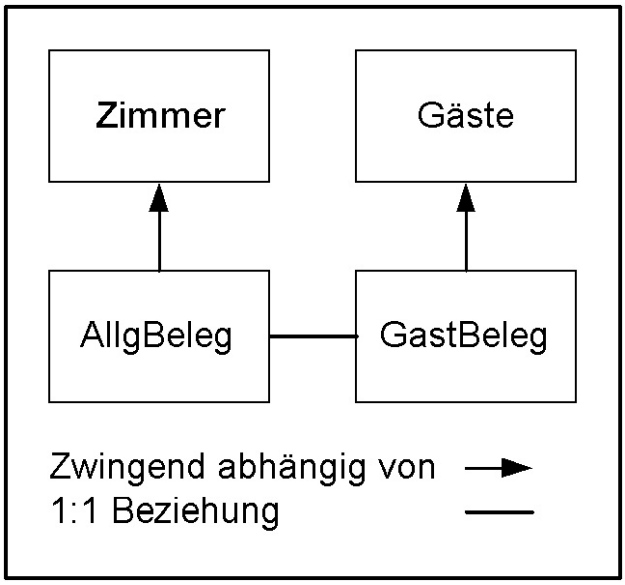
Erweiterte Darstellung der Beziehungen



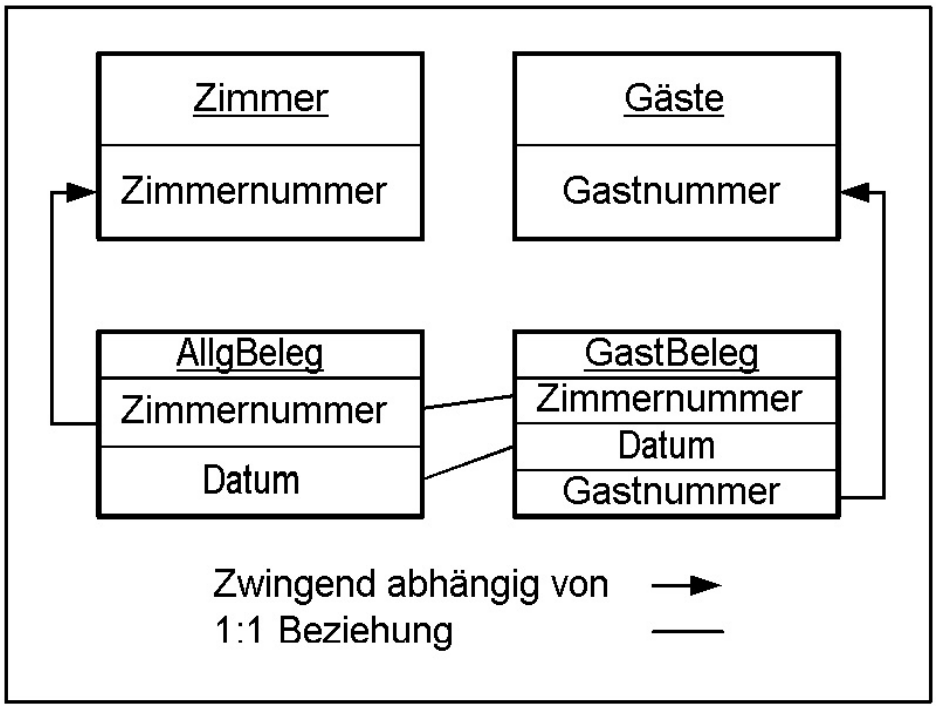
Alternativ könnte man die Belegung auch in zwei unterschiedlichen Tabellen speichern.

Die erste Tabelle enthält nur die Information, welches Zimmer an welchem Tag belegt ist (AllgBeleg).
Die zweite Tabelle enthält zusätzlich Einträge, wenn ein Zimmer durch einen Gast belegt ist (GastBeleg).

Einfache Darstellung der Beziehungen



Erweiterte Darstellung der Beziehungen



Ein Eintrag in der Tabelle Gastbeleg ist zwingend abhängig von einem Eintrag in der Tabelle Gäste und zusätzlich zwingend abhängig von der Existenz eines entsprechenden Satzes in der Tabelle AllgBeleg.

Die Spalte Gastnummer in der Tabelle GastBeleg darf bei dieser Tabellenstruktur nicht leer sein.

Die Beziehung zwischen den Tabellen AllgBeleg und GastBeleg ist dadurch gekennzeichnet, dass in beiden Tabellen nur jeweils ein korrespondierender Eintrag bezüglich Zimmernummer und Datum vorhanden sein kann. Eine solche Relation wird als 1:1 Beziehung bezeichnet.

Bewertung

In der relationalen Datenbanktheorie werden mehrere Beziehungstypen unterschieden:

1:1

In der Vergangenheit wurden Feldinhalte immer fix in der maximal möglichen Länge gespeichert. Ein Datenfeld mit der Deklaration VARCHAR(40) beanspruchte daher in jedem Satz, unabhängig vom tatsächlichen Inhalt, 40 Byte vom ohnehin knappen Speicherplatz auf der Festplatte. Daher wurden optionale Spalteninhalte häufig in eine zusätzliche Tabelle ausgelagert. Die Beziehung zwischen der Hauptdatentabelle mit den Pflichtfeldern und der Tabelle mit den optionalen Feldern ergab sich aus identischen Primärschlüsseln (1:1).

Moderne Datenbanken speichern Felder vom Typ VARCHAR nur in der tatsächlich benötigten Länge.
Der Datensatz enthält dann nur einen Zeiger (4 Byte) auf die Speicheradresse des Feldinhalts. Bei einer angenommenen Deklaration VARCHAR(40) und einem Feldinhalt 'Meier' werden z.B. 10 Byte benötigt.

Zeiger + 'Meier' + Endekennzeichen
4 Byte + 5 Byte + 1 Byte

Da Festplattenspeicher heute in Gigabyte bemessen wird und professionelle Datenbanksysteme Feldinhalte in variabler Länge speichern können, sind 1:1 Beziehungen nur noch als spezieller Ausnahmefall, bezogen auf Entlade- und Ladevorgänge von bestimmten Tabellenkonstellationen sinnvoll und erforderlich.

n:m

Dieser Beziehungstyp (z.B.: Zimmer <-> Gäste) lässt sich in keinem der marktüblichen Datenbanksysteme abbilden und muss in einzelne 1:n Beziehungen transformiert werden.

1:n bzw. n:1

Bleibt also nur ein relevanter Beziehungstyp übrig. Nennen wir ihn doch einfach **Beziehung**.
Beziehungen werden in Datenbanken als **Fremdschlüssel** dargestellt. **So einfach ist das!**

Leere Felder (NULL)

In der relationalen Theorie ist für alle definierten Tabellenspalten ein Feldinhalt erforderlich. Die Deklarieren einer Spalte ohne den Zusatz NOT NULL wird allerdings nur noch von wenigen echten Puristen als Fehler betrachtet. Diese Forderung resultiert ebenfalls aus der Vergangenheit und ist ähnlich wie die 1:1 Beziehung zu bewerten.

Für unser Beispiel bedeutet dies, dass eine neue Tabelle, damit verbunden ein Primärschlüssel, bestehend aus zwei Spalten, und zwei zusätzliche Fremdschlüssel notwendig sind, um dieser Forderung zu genügen.

Die Lösung mit drei Tabellen (Gastnummer darf NULL sein) ist der Lösung mit vier Tabellen in jedem Fall (Speicherplatz, Auswertegeschwindigkeit etc.) weit überlegen.

Übungsaufgabe

Erzeugen Sie Sql-Scriptdateien für die Datenbanksysteme Oracle und Interbase zum Anlegen der Datenstrukturen bezüglich der beiden skizzierten Lösungsansätze.

Mögliche Lösung mit einer Belegungstabelle

Oracle

```
CREATE TABLE ZIM (ZIMMER   NUMBER(3,0) NOT NULL CHECK(ZIMMER>0),
                  PRIMARY KEY (ZIMMER));

CREATE TABLE GST (GASTNR   NUMBER(6,0) NOT NULL CHECK(GASTNR>0),
                  PRIMARY KEY (GASTNR));

CREATE TABLE BEL (ZIMMER   NUMBER(3,0) NOT NULL,
                  DATUM    DATE NOT NULL CHECK(DATUM=TRUNC(DATUM)),
                  GASTNR   NUMBER(6,0),
                  FOREIGN KEY (ZIMMER) REFERENCES ZIM,
                  FOREIGN KEY (GASTNR) REFERENCES GST,
                  PRIMARY KEY (ZIMMER,DATUM));
```

Interbase

```
CREATE TABLE ZIM (ZIMMER   DECIMAL(3,0) NOT NULL CHECK (ZIMMER BETWEEN 1 AND 999),
                  PRIMARY KEY (ZIMMER));

CREATE TABLE GST (GASTNR     DECIMAL(6,0) NOT NULL CHECK (GASTNR BETWEEN 1 AND
999999),
                  PRIMARY KEY (GASTNR));

CREATE TABLE BEL (ZIMMER   DECIMAL(3,0) NOT NULL,
                  DATUM    TIMESTAMP NOT NULL CHECK(DATUM=CAST(DATUM AS DATE)),
                  GASTNR   DECIMAL(6,0),
                  FOREIGN KEY (ZIMMER) REFERENCES ZIM,
                  FOREIGN KEY (GASTNR) REFERENCES GST,
                  PRIMARY KEY (ZIMMER,DATUM));
```

Mögliche Lösung mit zwei Belegungstabellen

Oracle

```
CREATE TABLE ZIM (ZIMMER   NUMBER(3,0) NOT NULL CHECK(ZIMMER>0),
                  PRIMARY KEY (ZIMMER));

CREATE TABLE GST (GASTNR   NUMBER(6,0) NOT NULL CHECK(GASTNR>0),
                  PRIMARY KEY (GASTNR));

CREATE TABLE ZBA (ZIMMER   NUMBER(3,0) NOT NULL,
                  DATUM    DATE NOT NULL CHECK(DATUM=TRUNC(DATUM)),
                  FOREIGN KEY (ZIMMER) REFERENCES ZIM,
                  PRIMARY KEY (ZIMMER,DATUM));

CREATE TABLE ZBG (ZIMMER   NUMBER(3,0) NOT NULL,
                  DATUM    DATE NOT NULL,
                  GASTNR   NUMBER(6,0) NOT NULL,
                  FOREIGN KEY (ZIMMER,DATUM) REFERENCES ZBA,
                  FOREIGN KEY (GASTNR) REFERENCES GST,
                  PRIMARY KEY (ZIMMER,DATUM));
```

Interbase

```
CREATE TABLE ZIM (ZIMMER   NUMBER(3,0) NOT NULL CHECK(ZIMMER>0),
                  PRIMARY KEY (ZIMMER));

CREATE TABLE GST (GASTNR   NUMBER(6,0) NOT NULL CHECK(GASTNR>0),
                  PRIMARY KEY (GASTNR));

CREATE TABLE ZBA (ZIMMER   DECIMAL(3,0) NOT NULL,
                  DATUM    TIMESTAMP NOT NULL CHECK(DATUM=CAST(DATUM AS DATE)),
                  FOREIGN KEY (ZIMMER) REFERENCES ZIM,
                  PRIMARY KEY (ZIMMER,DATUM));

CREATE TABLE ZBG (ZIMMER   DECIMAL(3,0) NOT NULL,
                  DATUM    TIMESTAMP NOT NULL,
                  GASTNR   DECIMAL(6,0) NOT NULL,
                  FOREIGN KEY (ZIMMER,DATUM) REFERENCES ZBA,
                  FOREIGN KEY (GASTNR) REFERENCES GST,
                  PRIMARY KEY (ZIMMER,DATUM));
```

Extrahieren der Nicht-Schlüssel-Attribute

Alle Nicht-Schlüssel-Attribute dürfen ausschließlich vom Primärindex der entsprechenden Tabelle abhängig sein. Redundanzen (identische Nicht-Schlüssel-Attribute in mehreren Tabellen) müssen unbedingt vermieden werden.

D.h.: Eine Tabelle darf, zusätzlich zu den Nicht-Schlüssel-Attributen,
nur die Spalten der Primärschlüssel anderer Tabellen beinhalten,
sofern diese zur Definition von Fremdschlüsselbeziehungen erforderlich sind.

Nicht-Schlüssel-Attribute lassen sich problemlos, auch im Nachhinein, an vorhandene Tabellen anfügen.
Die Datenbanktabellen vieler Softwareprodukte wurden z.B. erst kürzlich durch eMail-Adressen, Handynummern, Homepageangaben etc. erweitert. Im Gegensatz dazu ist eine lückenhafte Datenbankstruktur im Bezug auf Tabellen, Primärschlüssel und Fremdschlüssel nur mit erheblichem Aufwand zu korrigieren.

Für unsere Pension bedeutet dies:

Die bisher handschriftlich auf Karteikarten erfassten Daten eines Gastes sollen in der Datenbank gespeichert werden.

Titel, beliebige Zeichen, max. 10 Stellen
Vorname(n), beliebige Zeichen, max. 40 Stellen
Familienname, beliebige Zeichen, max. 40 Stellen
Straße und Hausnummer, beliebige Zeichen, max. 40 Stellen
Postleitzahl, numerisch, max. 5 Stellen
Ort, beliebige Zeichen, max. 40 Stellen
Telefonnummer, beliebige Zeichen, max. 20 Stellen
Telefaxnummer, beliebige Zeichen, max. 20 Stellen

Bis auf Titel, Telefon- und Telefaxnummer sind für alle Attribute Werte erforderlich.

Zu den einzelnen Zimmern müssen keine zusätzlichen Angaben gespeichert werden.

Übungsaufgaben

1. Erweitern Sie die Gasttabelle in Oracle und Interbase bezüglich der genannten Eigenschaften und fügen Sie Beispielsätze in alle vorhandenen Tabellen ein.

2. Erzeugen Sie jeweils eine Sql-Abfrage für die unterschiedlichen Lösungsansätze, aus der die aktuelle Zimmerbelegung für Gäste ersichtlich wird. Das Ergebnis soll Zimmernummer, Datum, Gastnummer, Familienname und Ort beinhalten und nach Zimmernummer und Datum sortiert sein.

   Ausgabebeispiel:

ZIMMER	DATUM	GASTNR	FAMNAME	ORT
101	10.11.2006	1	Meier	Paderborn
101	11.11.2006	1	Meier	Paderborn
101	12.11.2006	1	Meier	Paderborn
101	13.11.2006	1	Meier	Paderborn
101	14.11.2006	1	Meier	Paderborn
101	15.11.2006	1	Meier	Paderborn
102	10.11.2006	2	Krause	Bielefeld
102	11.11.2006	2	Krause	Bielefeld
102	12.11.2006	2	Krause	Bielefeld

Mögliche Lösungen

1. Oracle

```
CREATE TABLE GST (GASTNR   NUMBER(6,0) NOT NULL CHECK(GASTNR>0),
                  TITEL    VARCHAR2(10),
                  VORNAME  VARCHAR2(40) NOT NULL,
                  FAMNAME  VARCHAR2(40) NOT NULL,
                  STRASSE  VARCHAR2(40) NOT NULL,
                  PLZ      NUMBER(5,0) NOT NULL CHECK(PLZ>0),
                  ORT      VARCHAR2(40) NOT NULL,
                  TELEFON  VARCHAR2(20),
                  TELEFAX  VARCHAR2(20),
                  PRIMARY KEY (GASTNR));
```

Interbase

```
CREATE TABLE GST (GASTNR   DECIMAL(6,0) NOT NULL CHECK (GASTNR BETWEEN 1 AND 999999),
                  TITEL    VARCHAR(10),
                  VORNAME  VARCHAR(40) NOT NULL,
                  FAMNAME  VARCHAR(40) NOT NULL,
                  STRASSE  VARCHAR(40) NOT NULL,
                  PLZ      DECIMAL(5,0) NOT NULL CHECK (PLZ BETWEEN 1 AND 99999),
                  ORT      VARCHAR(40) NOT NULL,
                  TELEFON  VARCHAR(20),
                  TELEFAX  VARCHAR(20),
                  PRIMARY KEY (GASTNR));
```

2. Sql-Abfrage (BEL)

```
SELECT BEL.ZIMMER,BEL.DATUM,BEL.GASTNR,GST.FAMNAME,GST.ORT FROM BEL,GST
WHERE GST.GASTNR=BEL.GASTNR ORDER BY BEL.ZIMMER,BEL.DATUM;
```

Sql-Abfrage (ZBG)

```
SELECT ZBG.ZIMMER,ZBG.DATUM,ZBG.GASTNR,GST.FAMNAME,GST.ORT FROM ZBG,GST
WHERE GST.GASTNR=ZBG.GASTNR ORDER BY ZBG.ZIMMER,ZBG.DATUM;
```