

Datenbanksysteme, 1. Praktikumsabschnitt

Informationssystem zur Verwaltung und Buchung von Ferienwohnungen über Internet (IVBF)

Jan Tammen <foobar@fh-konstanz.de>

17. Mai 2005

Inhaltsverzeichnis

0	Einleitung	3
1	Konzeptioneller Entwurf	3
1.1	Systemanforderungen	3
1.1.1	Anwendungsfälle	4
1.2	Layout einiger Internetseiten	4
1.2.1	Suche nach einer Ferienwohnung	5
1.2.2	Anzeige der Suchergebnisse	6
1.2.3	Detailansicht einer Ferienwohnung	6
1.2.4	Buchung einer Ferienwohnung	7
1.3	Datenschutzkonzept	7
1.3.1	Zugriffsberechtigte	7
1.3.2	Zugriffsrechte	8
1.4	Erweitertes ER-Modell	9
2	Implementierung	9
2.1	Relationen	9
2.2	Datenbank	10
2.2.1	Anlegen von Tabellen, CREATE	10
2.2.2	Zugriffsrechte, GRANT	12
2.2.3	Datenbefüllung, INSERT	13
2.2.4	Abfragen, SELECT	14

0 Einleitung

Es soll ein Internet-System entwickelt werden, welches Interessenten das Abrufen von Informationen über Ferienwohnungen ermöglicht sowie Buchungen von Wohnungen durch registrierte Kunden aufnimmt und verwaltet.

1 Konzeptioneller Entwurf

1.1 Systemanforderungen

Die an das System gestellten Anforderungen enthalten die folgenden Abläufe und Daten:

- Speicherung von Ferienwohnungen mit Angaben: Zuordnung zu einem Land, Anzahl Zimmer, Größe, Preis/Tag, hat die W. eine Sauna?, hat die W. ein Schwimmbad?, bis zu vier Bilder
- Speicherung von Touristenattraktionen mit Angaben: Zuordnung zu einem Land, Typ (z.B. „Freizeitpark“) und Name
- Speicherung der Entfernung zwischen einzelnen Ferienwohnung und Attraktionen
- Speicherung von Fluglinien mit Name sowie eine Zuordnung zu angeflogenen Ländern
- Abrufen von Informationen über eine Wohnung durch einen Interessenten, Anzeige aller Informationen, suchen nach speziellen Kriterien
- Registrierung, dadurch Erreichen des „Kunden“-Status
- Speicherung von Kundendaten: Name, Adresse, Zuordnung zu Land, Bankverbindung
- Buchung einer Ferienwohnung für einen speziellen Zeitraum durch registrierte Kunden
- Nach erfolgreicher Buchung: Erstellung einer Rechnung mit eindeutiger Buchungsnummer. Nach Zahlungseingang: Speicherung des Datums des Zahlungseinganges.

Aus diesen Anforderungen lässt sich das im folgenden Abschnitt aufgeführte UML Use-Case-Diagramm erstellen.

1.1.1 Anwendungsfälle

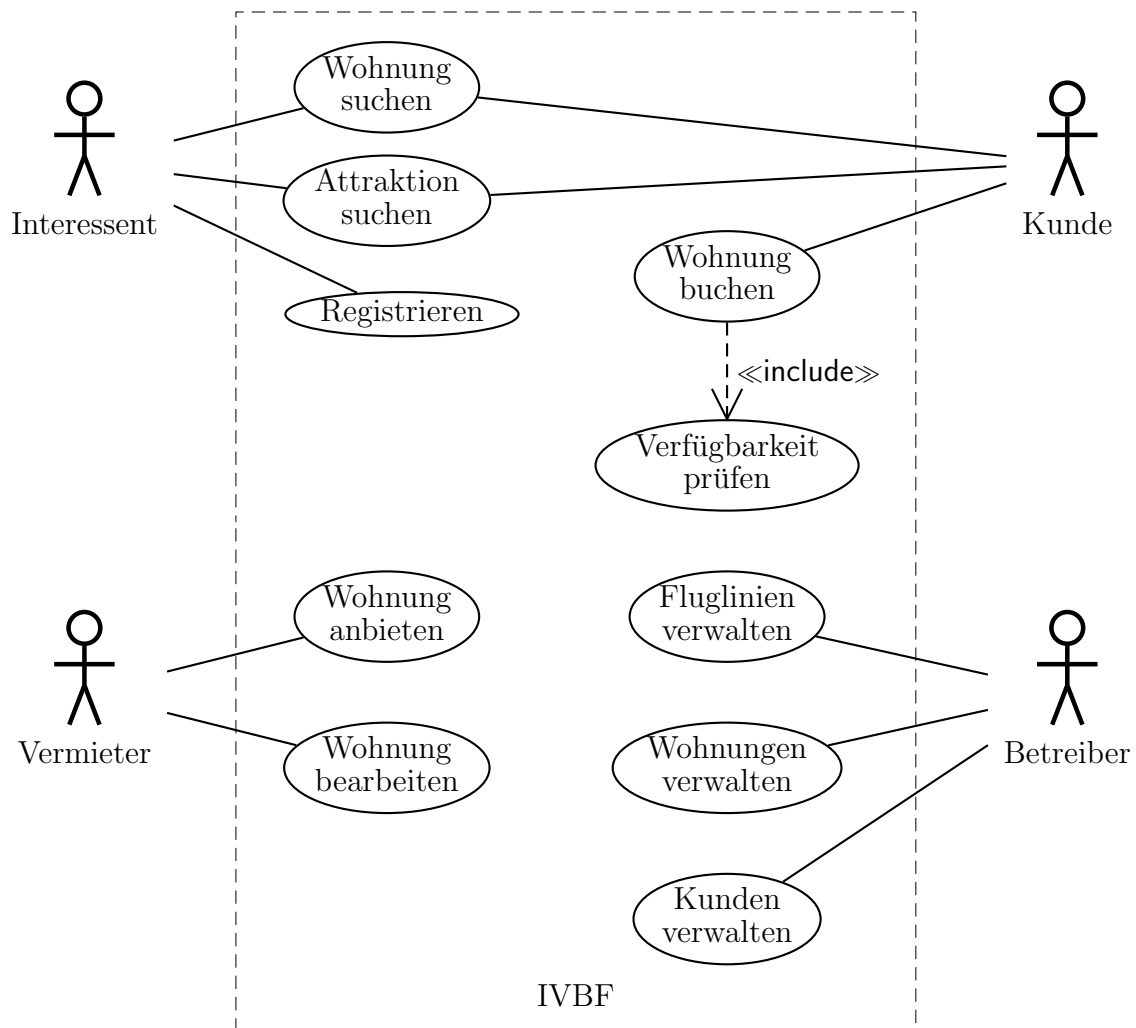


Abbildung 1: Use-Case-Diagramm

1.2 Layout einiger Internetseiten

Die Interaktion mit dem Systembenutzer erfolgt über Eingabemasken, zu deren Illustration hier einige schematische Beispiele aufgeführt sind.

1.2.1 Suche nach einer Ferienwohnung

Ferienwohnung suchen

Allgemeines

Land

Anzahl Zimmer

Größe in qm

Preis / Tag

nur Ergebnisse mit Bild? ☒

Zusatzausstattung

Sauna? ☒

Schwimmbad? ☐

Attraktionen

Die Wohnung soll max. km von folgenden Attraktionen entfernt gelegen sein:

Disneyland Paris (Freizeitpark) ☒

Connyland (Freizeitpark) ☐

Acquapark (Freizeitpark) ☐

[...]

Abbildung 2: Layoutbeispiel: Suche nach einer Ferienwohnung

1.2.2 Anzeige der Suchergebnisse

Suchergebnisse						
Treffer für Ihre Suchanfrage: X						
Land	Zimmer	Größe in qm	Preis / Tag	Sauna?	Schwimmbad?	Details
F	3	40,5	80,50	nein	ja	Details
F	2	72,5	127,50	ja	ja	Details
[...]						

Abbildung 3: Layoutbeispiel: Anzeige der Suchergebnisse




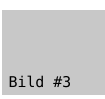
1.2.3 Detailansicht einer Ferienwohnung

Detailansicht Ferienwohnung

Allgemeines

Land	Frankreich
Anzahl Zimmer	3
Größe in qm	40,5
Preis / Tag	80,50 €

Bilder

Zusatzausstattung

Sauna?	<input checked="" type="checkbox"/>
Schwimmbad?	<input type="checkbox"/>

[Ich möchte diese Wohnung buchen!](#)

Attraktionen

Es werden Attraktionen im Umkreis von km angezeigt:

Name	Entfernung in km
Disneyland Paris (Freizeitpark)	60,40
[...]	

Abbildung 4: Layoutbeispiel: Detailansicht einer Ferienwohnung

1.2.4 Buchung einer Ferienwohnung

Buchung Ferienwohnung

Sie haben sich für eine Wohnung entschieden. Bitte geben Sie in den folgenden Eingabefeldern den gewünschten Zeitraum Ihres Aufenthaltes ein.

Das System wird dann prüfen, ob die Wohnung in dieser Zeit frei ist.

Gewählte Wohnung: **Frankreich, 3 Zimmer, 80,50 € / Tag**

Gewünschter Zeitraum:

vom bis zum

Sollen wir Ihnen im Falle einer Belegung einen alternativen Terminvorschlag machen?

ja ☐

Abbildung 5: Layoutbeispiel: Buchung einer Ferienwohnung

1.3 Datenschutzkonzept

1.3.1 Zugriffsberechtigte

Aus dem Use-Case-Diagramm (1) lassen sich folgende Benutzer- bzw. Benutzergruppen ableiten:

- Kunde (entspricht Datenbankuser dbsys01)
- Interessent (entspricht Datenbankuser dbsys33)
- Vermieter (entspricht Datenbankuser dbsys02)
- Betreiber (entspricht Datenbankuser dbsys34)

1.3.2 Zugriffsrechte

Ebenfalls dem Use-Case-Diagramm (1) entnommen werden können die benötigten Zugriffsrechte, welche in der folgenden Matrix aufgeführt sind. (Zugriffsrechte = {**r**ead, **w**rite, **d**elete})

Relation/Objekt	Subjekt			
	Interessent	Kunde	Vermieter	Betreiber
Kunde		r,w	r	r,w,d
Buchung		r,w	r	r,w,d
Rechnung		r	r	r,w,d
Ferienwohnung	r	r	r,w,d	r,w,d
Land	r	r	r	r,w,d
Fluglinie	r	r	r,w,d	r,w,d
Entfernung	r	r	r,w,d	r,w,d
Attraktion	r	r	r,w,d	r,w,d

Tabelle 1: Zugriffsrechte

Es wurden lediglich Zugriffsrechte für *komplette Tabellen* verwendet, nicht für einzelne Spalten. Daraus folgt, dass z.B. für die Tabelle „Kunde“ keine differenzierten Rechte existieren, die den Zugriff auf Kundendaten nur dem dazugehörigen Kunden erlauben.

Dazu müsste zu jedem Kunden ein Datenbankuser angelegt werden, dem dann per speziellem VIEW Zugriff auf „seine“ Daten gewährt werden könnte.

1.4 Erweitertes ER-Modell

Aus den Systemanforderungen wird das konzeptionelle Datenbank-Design in Form folgenden ER-Modelles entwickelt.

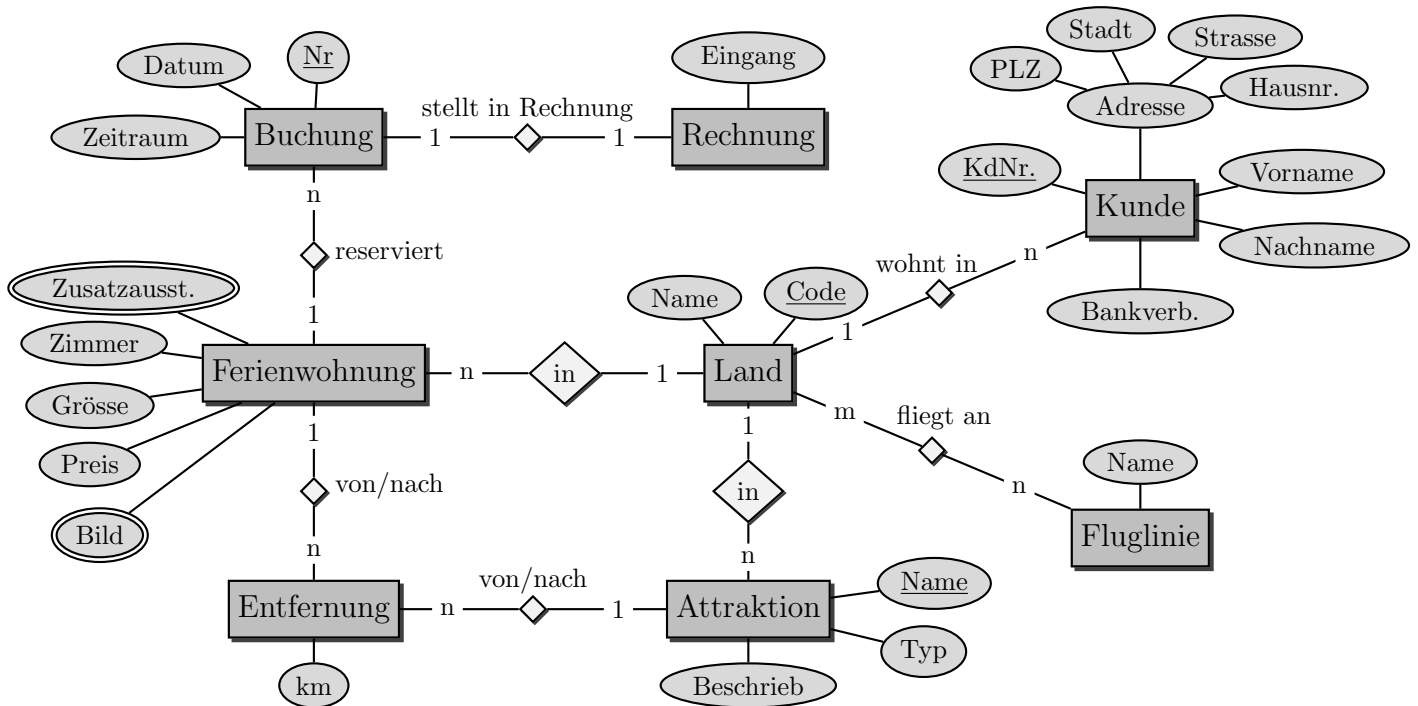


Abbildung 6: EERM

2 Implementierung

2.1 Relationen

Durch Transformation des ER-Modells erhält man die nachstehenden flachen Relationen.

- Attraktion = ({ AttraktionNr, Name, Typ, Beschrieb, Landcode })
- Bild = ({ BildNr, Pfad, FerienwohnungNr })
- Buchung = ({ BuchungNr, Rechnungsdatum, von, bis, KundenNr, FerienwohnungNr })
- Entfernung = ({ km, FerienwohnungNr, AttraktionNr })
- Ferienwohnung = ({ FerienwohnungNr, AnzahlZimmer, Groesse, Preis, Landcode, hatSauna, hatSchwimmbad })

- Fluglinie = ({ FluglinieNr, Name })
- Fluglinie_Land = ({ FluglinieNr, Landcode })
- Kunde = ({ KundeNr, Vorname, Nachname, PLZ, Stadt, Strasse, HausNr, BLZ, KontoNr, Landcode })
- Land = ({ Landcode, Name })

Hinweis: Hier sind bereits die nachträglich eingefügten Surrogatschlüssel (bspw. „FerienwohnungNr“ in der Relation Ferienwohnung) aufgeführt, welche im ER-Modell noch nicht erscheinen.

2.2 Datenbank

2.2.1 Anlegen von Tabellen, CREATE

```

1 CREATE TABLE land (
2     landcode    VARCHAR2(4),
3     fullname    VARCHAR2(100) NOT NULL,
4
5     CONSTRAINT pk_land    PRIMARY KEY (landcode) );
6
7 CREATE TABLE ferienwohnung (
8     fwnr        NUMBER(10),
9     anz_zimmer  NUMBER(1) NOT NULL,
10    groesse     NUMBER(5,2) NOT NULL,
11    preis       NUMBER(5,2) NOT NULL,
12    landcode    VARCHAR2(4) NOT NULL,
13    hat_sauna   NUMBER(1),
14    hat_schwab  NUMBER(1),
15
16    CONSTRAINT pk_ferienwohnung PRIMARY KEY (fwnr),
17    CONSTRAINT fk_ferienwohnung FOREIGN KEY (landcode)
18        REFERENCES land (landcode)
19        ON DELETE SET NULL,
20    CONSTRAINT anz_zimmer_ok    CHECK (anz_zimmer > 0),
21    CONSTRAINT groesse_ok       CHECK (groesse > 0),
22    CONSTRAINT preis_ok         CHECK (preis > 0),
23    CONSTRAINT hat_sauna_ok     CHECK (hat_sauna IN (0, 1)),
24    CONSTRAINT hat_schwab_ok    CHECK (hat_schwab IN (0, 1)) );
25
26 CREATE TABLE bild (
27     binr        NUMBER(10),
28     path        VARCHAR2(100) NOT NULL,
29     fwnr        NUMBER(10),
30
31    CONSTRAINT pk_bild    PRIMARY KEY (binr),

```

```

32     CONSTRAINT fk_bild    FOREIGN KEY (fwnr)
33                           REFERENCES ferienwohnung (fwnr)
34                           ON DELETE CASCADE );
35
36 CREATE TABLE kunde (
37     kdnr        NUMBER(10),
38     vorname     VARCHAR2(50) NOT NULL,
39     nachname    VARCHAR2(50) NOT NULL,
40     zip         NUMBER(5)    NOT NULL,
41     stadt       VARCHAR2(50) NOT NULL,
42     strasse     VARCHAR2(50) NOT NULL,
43     hsnr       VARCHAR2(4)   NOT NULL,
44     blz        NUMBER(15),
45     kontonr     NUMBER(15),
46     landcode    VARCHAR2(4),
47
48     CONSTRAINT pk_kunde    PRIMARY KEY (kdnr),
49     CONSTRAINT fk_kunde    FOREIGN KEY (landcode)
50                           REFERENCES land (landcode)
51                           ON DELETE SET NULL );
52
53 CREATE TABLE buchung (
54     bunr        NUMBER(10),
55     rdatum      DATE          DEFAULT SYSDATE,
56     von         DATE,
57     bis         DATE,
58     kdnr        NUMBER(10),
59     fwnr        NUMBER(10),
60
61     CONSTRAINT pk_buchung  PRIMARY KEY (bunr),
62     CONSTRAINT fk1_buchung FOREIGN KEY (kdnr)
63                           REFERENCES kunde (kdnr)
64                           ON DELETE SET NULL,
65     CONSTRAINT fk2_buchung FOREIGN KEY (fwnr)
66                           REFERENCES ferienwohnung (fwnr)
67                           ON DELETE SET NULL,
68     CONSTRAINT von_bis_ok  CHECK  (bis > von) );
69
70 CREATE TABLE attraktion (
71     atnr        NUMBER(10),
72     typ         VARCHAR2(100),
73     fullname    VARCHAR2(100) NOT NULL,
74     beschrieb   VARCHAR2(255),
75     landcode    VARCHAR2(4),
76
77     CONSTRAINT pk_attraktion PRIMARY KEY (atnr),
78     CONSTRAINT fk_attraktion FOREIGN KEY (landcode)
79                           REFERENCES land (landcode)
80                           ON DELETE CASCADE );
81

```

```

82 CREATE TABLE entfernung (
83     km          NUMBER(6,2) NOT NULL,
84     fwnr        NUMBER(10),
85     atnr        NUMBER(10),
86
87     CONSTRAINT fk_entfernung1 FOREIGN KEY (fwnr)
88                     REFERENCES ferienwohnung (fwnr)
89                     ON DELETE CASCADE,
90     CONSTRAINT fk_entfernung2 FOREIGN KEY (atnr)
91                     REFERENCES attraktion (atnr)
92                     ON DELETE CASCADE );
93
94 CREATE TABLE fluglinie (
95     flnr        NUMBER(10),
96     fullname    VARCHAR2(100) NOT NULL,
97
98     CONSTRAINT pk_fluglinie PRIMARY KEY (flnr) );
99
100 CREATE TABLE fluglinie_land (
101     flnr        NUMBER(10),
102     landcode    VARCHAR2(4),
103
104     CONSTRAINT fk_fluglinie_land1 FOREIGN KEY (flnr)
105                     REFERENCES fluglinie (flnr)
106                     ON DELETE CASCADE,
107     CONSTRAINT fk_fluglinie_land2 FOREIGN KEY (landcode)
108                     REFERENCES land (landcode)
109                     ON DELETE CASCADE );

```

2.2.2 Zugriffsrechte, GRANT

```

1 GRANT SELECT ON attraktion TO dbsys01;
2 GRANT SELECT ON bild TO dbsys01;
3 GRANT SELECT, INSERT ON buchung TO dbsys01;
4 GRANT SELECT ON entfernung TO dbsys01;
5 GRANT SELECT ON ferienwohnung TO dbsys01;
6 GRANT SELECT ON fluglinie TO dbsys01;
7 GRANT SELECT ON fluglinie_land TO dbsys01;
8 GRANT SELECT, ALTER, INSERT ON kunde TO dbsys01;
9 GRANT SELECT ON land TO dbsys01;
10
11 GRANT SELECT, ALTER, INSERT, DELETE ON attraktion TO dbsys02;
12 GRANT SELECT, ALTER, INSERT, DELETE ON bild TO dbsys02;
13 GRANT SELECT ON buchung TO dbsys02;
14 GRANT SELECT, ALTER, INSERT, DELETE ON entfernung TO dbsys02;
15 GRANT SELECT, ALTER, INSERT, DELETE ON ferienwohnung TO dbsys02;
16 GRANT SELECT ON fluglinie TO dbsys02;
17 GRANT SELECT ON fluglinie_land TO dbsys02;

```

```

18 GRANT SELECT ON kunde TO dbsys02;
19 GRANT SELECT ON land TO dbsys02;
20
21 GRANT SELECT ON ferienwohnung TO dbsys33;
22 GRANT SELECT ON land TO dbsys33;
23 GRANT SELECT ON fluglinie TO dbsys33;
24 GRANT SELECT ON entfernung TO dbsys33;
25 GRANT SELECT ON attraktion TO dbsys33;

```

2.2.3 Datenbefüllung, INSERT

```

1  INSERT INTO land VALUES ( 'de', 'Deutschland' );
2  INSERT INTO land VALUES ( 'fr', 'Frankreich' );
3  INSERT INTO land VALUES ( 'ch', 'Schweiz' );
4  INSERT INTO land VALUES ( 'it', 'Italien' );
5
6  INSERT INTO attraktion VALUES ( 1, 'Freizeitpark', 'Europapark', 'Beschrieb', 'de' );
7  INSERT INTO attraktion VALUES ( 2, 'Freizeitpark', 'Disneyland Paris', 'Beschrieb', 'fr' );
8  INSERT INTO attraktion VALUES ( 3, 'Freizeitpark', 'Connyland', 'Beschrieb', 'ch' );
9  INSERT INTO attraktion VALUES ( 4, 'Freizeitpark', 'Acquapark', 'Beschrieb', 'it' );
10
11 INSERT INTO ferienwohnung VALUES ( 1, 1, 10.5, 20.5, 'de', 0, 0 );
12 INSERT INTO ferienwohnung VALUES ( 2, 2, 22.5, 67.5, 'de', 1, 0 );
13 INSERT INTO ferienwohnung VALUES ( 3, 3, 40.5, 80.5, 'fr', 0, 1 );
14 INSERT INTO ferienwohnung VALUES ( 4, 2, 72.5, 127.5, 'fr', 1, 1 );
15 INSERT INTO ferienwohnung VALUES ( 5, 2, 10.5, 20.5, 'ch', 0, 0 );
16 INSERT INTO ferienwohnung VALUES ( 6, 4, 92.5, 97.5, 'ch', 1, 0 );
17 INSERT INTO ferienwohnung VALUES ( 7, 3, 50.5, 70.5, 'it', 0, 1 );
18 INSERT INTO ferienwohnung VALUES ( 8, 1, 12.5, 27.5, 'it', 0, 0 );
19
20 INSERT INTO entfernung VALUES ( 100.98, 1, 1 );
21 INSERT INTO entfernung VALUES ( 220.38, 2, 1 );
22 INSERT INTO entfernung VALUES ( 60.4, 3, 2 );
23 INSERT INTO entfernung VALUES ( 45.5, 4, 2 );
24 INSERT INTO entfernung VALUES ( 600.98, 5, 3 );
25 INSERT INTO entfernung VALUES ( 9.18, 6, 3 );
26 INSERT INTO entfernung VALUES ( 122.94, 7, 4 );
27 INSERT INTO entfernung VALUES ( 56.38, 8, 4 );
28
29 INSERT INTO fluglinie VALUES ( 1, 'Lufthansa' );
30 INSERT INTO fluglinie VALUES ( 2, 'Air France' );
31 INSERT INTO fluglinie VALUES ( 3, 'Swiss Air' );
32 INSERT INTO fluglinie VALUES ( 3, 'British Airways' );
33
34 INSERT INTO fluglinie_land VALUES ( 1, 'de' );
35 INSERT INTO fluglinie_land VALUES ( 1, 'ch' );
36 INSERT INTO fluglinie_land VALUES ( 2, 'fr' );
37 INSERT INTO fluglinie_land VALUES ( 2, 'ch' );

```

```

38 INSERT INTO fluglinie_land VALUES ( 3, 'ch' );
39 INSERT INTO fluglinie_land VALUES ( 3, 'de' );
40 INSERT INTO fluglinie_land VALUES ( 4, 'fr' );
41
42 INSERT INTO kunde VALUES ( 1, 'Hans', 'Müller', 12345, 'Hamburg', 'Sonnenweg', '1a', 123456,
    7890123, 'de' );
43 INSERT INTO kunde VALUES ( 2, 'Erwin', 'Schulze', 12345, 'Berlin', 'Blumenweg', '10',
    123456, 6547645, 'de' );
44 INSERT INTO kunde VALUES ( 3, 'Francois', 'LeGrand', 54566, 'Paris', 'La Rue', '90', 476566,
    565665, 'fr' );
45 INSERT INTO kunde VALUES ( 4, 'Michele', 'LaPetite', 6774, 'Grenoble', 'La Rue', '10',
    645645, 3435685, 'fr' );
46 INSERT INTO kunde VALUES ( 5, 'Urs', 'Huber', 54356, 'Bern', 'Bergstrasse', '34', 5435435,
    77657, 'ch' );
47 INSERT INTO kunde VALUES ( 6, 'Chris', 'Gruetli', 65465, 'Zürich', 'Feldweg', '54', NULL,
    NULL, 'ch' );
48 INSERT INTO kunde VALUES ( 7, 'Michel', 'Grimaldi', 5435, 'Rom', 'Testweg', '46c', NULL,
    NULL, 'it' );
49 INSERT INTO kunde VALUES ( 8, 'Ernesto', 'Venecia', 5436, 'Rom', 'Teststrasse', '54', 67567,
    54544, 'it' );
50
51 INSERT INTO buchung VALUES ( 1, '2004-01-02', '2003-12-01', '2003-12-25', 1, 4 );
52 INSERT INTO buchung VALUES ( 2, '2005-01-02', '2004-11-02', '2004-12-01', 4, 7 );
53 INSERT INTO buchung VALUES ( 3, '2004-08-01', '2004-02-22', '2004-03-05', 8, 6 );
54 INSERT INTO buchung VALUES ( 4, '2004-07-12', '2004-06-12', '2004-06-25', 2, 1 );
55 INSERT INTO buchung VALUES ( 5, '2005-02-24', '2005-01-22', '2005-02-13', 3, 2 );

```

2.2.4 Abfragen, SELECT

Nachfolgende SELECT-Abfragen sollen die Korrektheit des Datenmodells verifizieren.

Sprachlich lässt sich das Ergebnis der ersten Abfrage wie folgt formulieren: „Welche Ferienwohnungen in der Schweiz mit Sauna sind in der Zeit vom 01.11.2004 bis 21.11.2004 noch frei?“

```

1 SELECT f.fwnr, b.von, b.bis
2 FROM ferienwohnung f
3 LEFT OUTER JOIN buchung b
4 ON ( b.fwnr = f.fwnr )
5 WHERE
6 (
7     ( b.von < '2004-11-01' AND b.bis < '2004-11-21' )
8     OR
9     ( b.von > '2004-11-01' AND b.bis > '2004-11-21' )
10    OR
11    ( b.bunr IS NULL )
12 )
13 AND f.landcode = 'ch'
14 AND f.hat_sauna = 1;

```

```

15
16 # Ergebnis:
17
18     FWNR VON      BIS
19 -----
20     6 22-FEB-04 05-MAR-04

```

Listing 1: SELECT-Befehl 1

„Welche Ferienwohnungen in Frankreich sind höchstens 100km von Disneyland entfernt?“

```

1 SELECT f.fwnr,f.land,e.km,e.atnr
2 FROM ferienwohnung f
3 JOIN entfernung e
4 ON (f.fwnr = e.fwnr)
5 WHERE e.atnr = 2
6 AND f.landcode = 'fr'
7 AND e.km <= 100;
8
9 # Ergebnis:
10     FWNR LAND      KM      ATNR
11 -----
12     3 fr      60.4      2
13     4 fr      45.5      2

```

Listing 2: SELECT-Befehl 2

„Wieviele Reservierungen gibt es für die einzelnen Länder?“

```

1 SELECT f.landcode,COUNT(*)
2 FROM ferienwohnung f
3 JOIN buchung b
4 ON (f.fwnr = b.fwnr)
5 JOIN land l
6 ON (l.landcode = f.landcode)
7 GROUP BY f.landcode;
8
9 # Ergebnis:
10 LAND  COUNT(*)
11 -----
12 ch      1
13 de      2
14 fr      1
15 it      1

```

Listing 3: SELECT-Befehl 3