

Praktikum ‚Datenbanken‘

Aufgabenblatt 3

Aufgabe 1:

In den Aufgaben 1 und 2 des letzten Aufgabenblatts haben Sie eine Tabelle für Städte und eine für Attraktionen angelegt. Vermutlich ist Ihnen der Zusammenhang zwischen den Tabellen aufgefallen. Tatsächlich sind sie ein gutes Beispiel für referenzielle Integrität.

a. Löschen Sie die Tabellen und legen Sie sie mit künstlichen, vom DBMS verwalteten Primärschlüsseln und referenzieller Integrität erneut an.

b. Fügen Sie Daten in die Tabellen ein. Welche Fehlermeldung erhalten Sie, wenn Sie versuchen gegen die referenzielle Integrität zu verstoßen?

Aufgabe 2:

a. Wenn Sie eine Tabelle wie folgt definieren:

```
create table t(  
    c1 int  
);
```

können Sie mit

```
alter table t add column c2 int
```

eine Spalte hinzufügen. Fügen Sie auf diese Weise die Spalten `laengengrad` und `breitengrad` vom Datentyp `decimal` zu Ihrer Städtetabelle aus Aufgabe 1 hinzu.

b. In Vorlesung und Praktikum haben Sie die SQL-Anweisung `update` kennengelernt. Nutzen Sie die Anweisung, um die beiden Spalten `laengengrad` und `breitengrad` sinnvoll mit Daten zu füllen.

c. Wenn Sie eine Tabelle wie folgt anlegen:

```
create table t(  
    c1 int,  
    c2 int,  
);
```

können Sie mit

```
alter table t drop column c2
```

die Spalte `c2` löschen. Löschen Sie auf diese Weise die Spalte mit der Beschreibung der Attraktionen.

d. Wenn Sie zwei Tabellen wie folgt anlegen:

```
create table t1(  
    c int primary key  
);  
create table t2(  
    c1 int primary key,
```

```
        c2 int  
    );
```

Dann können Sie nachträglich referenzielle Integrität hinzufügen:

```
alter table t2  
add constraint ref_tabelle_t1 foreign key(c2) references t1(c);
```

die referenzielle Integrität auch nachträglich definieren. Probieren Sie das am Beispiel Ihrer Tabellen aus Aufgabe 1 aus: Sie löschen die Tabellen und legen sie zunächst *ohne* referenzielle Integrität erneut an. Die referenzielle Integrität definieren Sie dann.

Aufgabe 3:

Die folgenden Aufgabenteile lösen Sie mit Hilfe von SQL-Anweisungen, die Sie mit H2 testen. Fügen Sie nach jeder Teilaufgabe auch Datensätze in die Tabelle(n) ein.

a. Das Design der folgenden Tabelle enthält einen gravierenden Fehler. Identifizieren und beheben Sie den Fehler.

```
create table reise(  
    id int,  
    ziel varchar(20),  
    preis int,  
    dauer int,  
    verkehrsmittel varchar(20),  
    anfangsdatum varchar(20)  
)
```

b. Ergänzen Sie die create-table-Anweisung um mindestens zwei sinnvolle statische Integritätsregeln.

c. Die Spalte `dauer` enthält die Dauer der Reise in Tagen. Ergänzen Sie die create-table-Anweisung um eine statische Integritätsregel, die sicherstellt, dass der Preis pro Tag mindestens 50 beträgt.

d. Die Spalten `ziel` und `verkehrsmittel` kann man mit Hilfe von referenzieller Integrität und künstlichen Schlüsseln besser gestalten: Löschen Sie die Tabelle `reisen` definieren Sie drei Tabellen, die mit Hilfe von referenzieller Integrität verbunden sind. Eine der drei Tabellen enthält die Ziele, eine die Verkehrsmittel und die dritte die Reise. Fügen Sie auch in jede der drei Tabellen mindestens drei Datensätze ein.

Aufgabe 4:

Im Rahmen des letzten Aufgabenblatts haben Sie die Tabelle `titanic` definiert und mit Daten befüllt.

a. Ergänzen Sie die Tabelle um einen künstlichen, vom DBMS verwalteten Primärschlüssel.

b. Es sei eine Tabelle wie folgt definiert:

```
create table t(  
    c1 int primary key,  
    c2 int  
);
```

Die Spalte `c2` kann Dubletten enthalten. Eine Liste mit allen dublettenfreien Werten aus der Spalte `c2`, erhält man mit

```
select distinct c2  
from t
```

Ermitteln Sie mit dieser Anweisung alle verschiedenen Werte in den Spalten `embarked`, `sibsp` und `parch`.

c. Ergänzen Sie die Tabelle um *mindestens* neun sinnvolle Integritätsregeln.

d. Die Spalte `embarked` enthält das Kürzel des Ortes, an dem der Passagier zugestiegen ist. Dieses Merkmal soll zusammen mit einem künstlichen Primärschlüssel in einer eigenen Tabelle abgelegt werden. Formulieren Sie eine passende `create-table`-Anweisung. Fügen Sie noch keine Daten in die Tabelle ein.

e. Wenn zwei Tabellen wie folgt definiert sind:

```
create table t1(  
    c1 int  
);  
create table t2(  
    c2 int  
);
```

Dann können die Daten aus `t2` wie folgt in die Tabelle `t1` kopiert werden:

```
insert into t1(  
    select *  
    from t2)
```

Kopieren Sie nach diesem Muster und mit Hilfe von Aufgabenteil b. alle verschiedenen Werte der Spalte `embarked` in die Tabelle, die Sie in Aufgabenteil d. definiert haben.

f. Ergänzen Sie die Tabelle aus Aufgabenteil e. um eine Spalte, mit dem vollständigen Ortsnamen. Ergänzen Sie auch sinnvolle Integritätsregeln.

g. Die Anweisung

```
select * from embarked;
```

sollte Ihnen jetzt die drei Kürzel `,C'`, `,Q'` und `,S'` anzeigen. Nutzen Sie die `update`-Anweisung um auch die zugehörigen Städtenamen `Cherbourg`, `Queenstown` und `Southampton` zu ergänzen.