Übung 2

Aufgabe 2.1

Im Zusammenhang mit dem relationalen Datenmodell haben wir die Begriffe *Superschlüssel*, *Schlüsselkandidat* und *Primärschlüssel* kennengelernt.

Erläutern Sie diese Begriffe anhand des folgenden Beispiels: Gegeben sei eine Relation

`Händler(Id, Name, Adresse, Telefonnummer, Branche, URL)`,

in der Informationen für ein Händlerverzeichnis gespeichert werden. Welche Attribute oder Attributkombinationen sind Schlüsselkandidaten? Welcher der Schlüsselkandidaten bietet sich als Primärschlüssel an? Begründen Sie Ihre Wahl und machen Sie Ihre Annahmen über die Bedeutung und möglichen Ausprägungen der Attribute explizit.

Aufgabe 2.2

Geben Sie ein SQL-Script an, das die folgenden Tabellen anlegt:

- `person(nr, name, vorname, plz, strasse)`
- `plzort(plz, ort)`

Überlegen Sie sich geeignete Datentypen und legen Sie dabei folgende Integrity-Constraints an:

- das Feld *ort* in der Tabelle *plzort* darf nicht leer sein;
- der Wert *plz* in der Tabelle *person* muss in der Tabelle *plzort* vorkommen;
- wenn in *plzort* eine *plz* geändert wird, soll diese Änderung bei allen betroffenen *person*en automatisch nachgezogen werden;
- das Löschen eines Datensatzes aus *plzort* soll unterbunden werden, wenn in diesem *ort* noch *person*en wohnen;
- die Kombination *vorname, name* soll eindeutig sein;
- die *plz* darf nur aus Ziffern bestehen (Hinweis: `SIMILAR TO`[1]);

Aufgabe 2.3

Nehmen Sie an, dass die oben definierten Tabellen die folgenden Daten enthalten:

plzort

plz	ort
47805	Krefeld

person

nr	name	vorname	plz	strasse
4711	Gans	Gustav	47805	Kölner Straße 66

Geben Sie Beispiele für *INSERT*, *UPDATE* und *DELETE*-Befehle an, die aufgrund der in Aufgabe 2.2 definierten Constraints fehlschlagen. Welche Statements schlagen warum fehl?

[1] Siehe: https://www.postgresql.org/docs/13/functions-matching.html

Übung 2

Aufgabe 2.4

Mit dem ALTER TABLE Befehl können Spalten zu einer Tabelle hinzugefügt oder entfernt werden:

- `ALTER TABLE table ADD COLUMN column type`
- `ALTER TABLE table DROP COLUMN column`
- `ALTER TABLE table RENAME COLUMN oldname TO newname`

Der Datentyp einer Spalte lässt sich in PostgreSQL wie folgt ändern:

- `ALTER TABLE table ALTER COLUMN column TYPE newtype [USING expression]`

Die optionale Klausel `USING expression` gibt dabei einen Ausdruck an, der die Werte vom alten Spaltentyp in den neuen Spaltentyp umwandelt. Wenn die USING-Klausel fehlt, versucht PostgreSQL, die alten Werte über implizites Casting in den neuen Datentyp zu wandeln.

a) Nehmen Sie an, dass Sie *person.plz* zunächst ungünstigerweise als *INTEGER* definiert hatten. Wie lautet der Befehl, um in PostgreSQL die Spalte *person.plz* von *INTEGER* in *VARCHAR(5)* zu ändern?
b) In manchen DBMS existiert kein eigener Befehl zum Ändern des Datentyps einer Spalte. Die Änderung eines Spaltentyps lässt sich dann nur über einen Umweg erreichen. Überlegen Sie, welcher Umweg das ist und geben Sie eine Folge von SQL-Statements für das obige Beispiel an.

Aufgabe 2.5

Geben Sie für folgende Datenmanipulationen an den obigen Tabellen SQL-Statements an:

a) Auflisten aller *nr* in *person*, bei denen mindestens eines der restlichen Felder leer (NULL) ist;
b) Anhängen des Strings 'bla' an jeden *namen* in p*erson* (Hinweis: der Operator für String-Konkatenation lautet in SQL ||);
c) Auflisten der Tupel in *person*, bei denen die *strasse* keine Hausnummer hat (Hinweis: regex mit SIMILAR TO);
d) Auflisten aller *name*n und *vorname*n von *person*en zusammen mit dem *ort*, in dem sie wohnen;