

Übungsblatt 2

Übung 2.1 Geben Sie ein SQL-Script an, das die folgenden Tabellen anlegt:

Person	Plzort
<i>nr#</i>	<i>plz#</i>
<i>name</i>	<i>ort</i>
<i>vorname</i>	
<i>plz</i>	
<i>strasse</i>	

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
- die Kombination *vorname*, *name* soll eindeutig sein
- die *plz* darf nur aus Ziffern bestehen (Hinweis: *SIMILAR TO*)

Übung 2.2 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 ist
- b) Anhängen des Strings 'bla' an jeden *namen* in *person* (Hinweis: der Operator für String-Konkatenation lautet in SQL ||)
- c) Auflisten der Tupel in *person*, bei denen die Strasse keine Hausnummer hat (Hinweis: *SIMILAR TO*)

Übung 2.3 Mit dem *ALTER TABLE* Befehl können Spalten zu einer Tabelle hinzugefügt, entfernt oder umbenannt werden:

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

Das Ändern des Datentyps einer Spalte geht jedoch im allgemeinen nicht (warum?), bzw. nur über einen Umweg. Überlegen Sie, welcher Umweg das ist und geben Sie eine Folge von SQL-Statements an, mit dem der Datentyp der Spalte *person.plz* von *int* in *varchar(5)* geändert wird.

Übung 2.4 Finden Sie anhand der PostgreSQL-Dokumentation heraus, wie aus dem System Catalog von PostgreSQL alle Tabellen ermittelt werden können, die einem bestimmten Benutzer gehören.