

Übung 6

Aufgabe 1

Gegeben sei das folgende Relationenmodell:

- User(Username, Email)
- likes(Username, PostID, Datum)
- Post(PostID, Text, verfasstVon, verfasstDatum)

(a) Erstellen Sie die Tabellen mit SQL. Beachten Sie dabei folgende Einschränkungen:

- Kein Attribut ist optional
- Es darf keine zwei User mit der gleichen Emailadresse geben
- Der Text eines Posts soll maximal 250 Zeichen beinhalten
- Die PostID soll automatisch generiert werden

(b) Fügen Sie bei der Tabelle User als zusätzliches Attribut ein Geburtsdatum hinzu. Ein User muss nicht zwangsläufig ein Geburtsdatum angeben.

(c) Wir beschließen, Likes über einen eigenen automatisch generierten Schlüssel zu identifizieren. Fügen Sie ein entsprechendes Attribut mit den korrekten Integritätsbedingungen in die Tabelle ein. Stellen Sie trotzdem sicher, dass ein User jeden Post nur einmal liken darf.

Aufgabe 2

Betrachten Sie das folgende Relationenmodell

- Kundin(AusweisNr, Name, FuehrerscheinNr)
- Auto(ID, Kennzeichen, Modell, Marke, AnzahlSitze, stehtIn)
- Autohaus(ID, Adresse)
- leiht(AusweisNr, AutoID, Datum)

Übersetzen Sie die folgenden Anfragen und Ausdrücke der Relationenalgebra in SQL Befehle.

- (a) Geben Sie alle Informationen zu Kund:innen aus.
- (b) Geben Sie Kennzeichen aller Autos von der Marke VW aus.
- (c) $\pi_{\text{Marke, Modell}}(\sigma_{\text{AnzahlSitze} \geq 5}(\text{Auto}))$
- (d) $\pi_{\text{Name}}(\sigma_{\text{Kundin.AusweisNr} = \text{leiht.AusweisNr} \wedge \text{Datum} = '2024-05-01'}(\text{Kundin} \times \text{leiht}))$
- (e) Geben Sie die AusweisNr und den Namen aller Kund:innen an, die im Autohaus mit der ID 5 ein Auto geliehen haben. Jeder Eintrag soll maximal einmal in der Ergebnistabelle auftauchen.

Aufgabe 3 *Bonus*

Erstellen Sie für die Relationen von Aufgabe 2 die zugehörigen Tabellen mit `create table`-Befehlen. Überlegen Sie sich sinnvolle Datentypen und Integritätsbedingungen.