

Beispiellösung zur 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 bei Erstellung eines Posts automatisch generiert werden

Lösungsvorschlag:

```
create table USER(  
  Username char(50) primary key,  
  Email varChar(255) not null unique);  
  
create table POST(  
  PostID int auto_increment primary key,  
  Text varchar(250) not null,  
  verfasstDatum date not null,  
  verfasstVon char(50) not null,  
  foreign key (verfasstVon) references USER (Username));  
  
create table LIKES(  
  Username char(50) not null,  
  PostID int not null,  
  Datum date not null,  
  primary key (Username, PostID),  
  foreign key (Username) references USER(Username),  
  foreign key (PostID) references POST (PostID));
```

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

Lösungsvorschlag:

```
alter table USER
add column Geburtsdatum date;
```

- (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.

Lösungsvorschlag:

```
alter table LIKES
drop primary key,
add column likeID int auto_increment primary key,
add unique (Username, PostID);
```

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.

Lösungsvorschlag:

```
select *  
from Kundin;
```

- (b) Geben Sie Kennzeichen aller Autos von der Marke VW aus.

Lösungsvorschlag:

```
select Kennzeichen  
from Auto  
where Marke='VW';
```

- (c) $\pi_{\text{Marke, Modell}}(\sigma_{\text{AnzahlSitze} \geq 5}(\text{Auto}))$

Lösungsvorschlag:

```
select distinct Marke, Model  
from Auto  
where AnzahlSitze >= 5;
```

- (d) $\pi_{\text{Name}}(\sigma_{\text{Kundin.AusweisNr}=\text{leiht.AusweisNr} \wedge \text{Datum}='2024-05-01'}(\text{Kundin} \times \text{leiht}))$

Lösungsvorschlag:

```
select distinct Name  
from Kundin, leiht  
where Kundin.AusweisNr = leiht.AusweisNr  
and Datum= '2024-05-01';
```

- (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.

Lösungsvorschlag:

```
select distinct Kundin.AusweisNr, Name  
from leiht, Auto, Kundin  
where AutoID = ID  
and leiht.AusweisNr = Kundin.AusweisNr  
and stehtIn = 5;
```

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.

Lösungsvorschlag:

```
create table Kundin(  
  AusweisNr char(10) primary key,  
  Name varchar(100) not null,  
  FuehrerscheinNr char(11) not null);
```

```
create table Autohaus(  
  ID int auto_increment primary key,  
  Adresse varchar(250) not null);
```

```
create table Auto(  
  ID int auto_increment primary key,  
  Kennzeichen char(12) not null unique,  
  Modell char(50) not null,  
  Marke char(50) not null,  
  stehtIn int,  
  foreign key (stehtIn) references Autohaus (ID));
```

```
create table leiht(  
  AusweisNr char(10) not null,  
  AutoID int not null,  
  Datum date not null,  
  primary key (AusweisNr, AutoID),  
  foreign key (AusweisNr) references Kundin(AusweisNr),  
  foreign key (AutoID) references Auto(ID));
```