Exercise 6

March 31, 2022

Data Handling: Databases
Spring Semester 2022
University of St. Gallen

Prof. Dr. Michael Grossniklaus
Database and Information Systems
University of Konstanz

Data Definition and Manipulation in SQL

In this exercise, we return to the simple online auction database that we created in Exercise 3. Recall the following logical schema of the database.

```
Users(id, first, last, email)
Items(id, title, start, end)
Bids(user_id → Users, item_id → Items, amount, placed)
```

Task 6.1: "Weak Entities"

Assume that we want to extend our online auction database with the possibilities to store ratings for the registered users. In order to so, create the following table in your "eRummage" database.

```
Ratings(id, user_id → Users, rating, comment)
```

Of course, it makes little sense to store ratings for users in the database once they have deleted their account. Therefore, **`Ratings`** are conceptually weak entities with **`Users`** as the corresponding strong entity.

In SQL, the desired behaviour of weak entities can be implemented using cascading deletes. To test this behaviour, create a new user and together with two ratings. Then, delete the user and check whether the corresponding ratings have been deleted automatically.

Task 6.2: Constraints

In Exercise 3, we were not able to enforce the following two constraints automatically.

1. It should not be possible to place a bid for an item after the auction ended.
2. It should not be possible to place a bid with a lower amount than the current maximum bid.

Now, that we learned about **`TRIGGERS`** in SQL, we can finally alter our database schema implement this functionality as well. Adapt your schema and test your triggers by deliberately inserting tuples that violate the above constraints.