Exercise 3

March 10, 2022

Data Handling: Databases Prof. Dr. Michael Grossniklaus
Spring Semester 2022 Database and Information Systems
University of St. Gallen University of Konstanz

First Steps in PostgreSQL

In this exercise, we will set up a simple database for an online auction website. The database has the following logical schema.

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

Items.start denotes the starting price of the auction, whereas **Items**.end denotes the end date (and time) of the corresponding auction. **Bids**.amount is the price the referenced user is willing to pay for the referenced item. **Bids**.placed is the date (and time) when the corresponding bid was placed in the auction.

Task 3.1: Creating Your First Database in PostgreSQL

Using pgAdmin, create a new database called "eRummage" in your PostgreSQL installation. Do do so, first navigate to the Databases entry on the left-hand side of the application window. Right-click on the entry and select Create → Database.... You should see the following window.
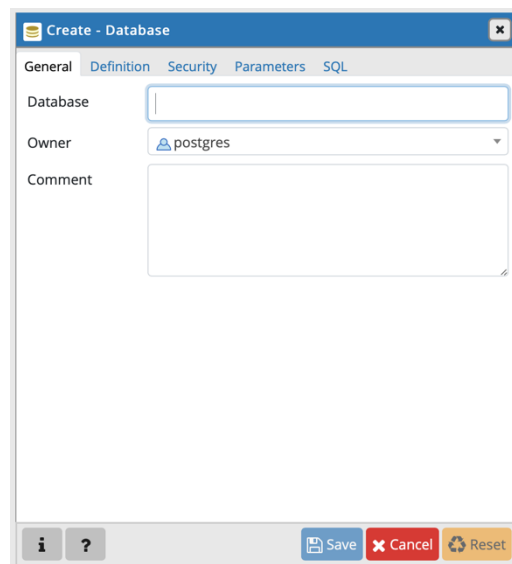

Enter the database name and click Save. For the time being, we do not change the Owner of the database. The Comment field is optional. Use it if you feel inspired to do so! Your new database should now appear under the Databases entry on the left-hand side of the window. Note, if your does not appear, you might need to refresh the tree, by right-clicking on the Database entry and selecting Refresh....

Task 3.2: Create the Schema of the Online Auction Database

In order to create the schema of your online auction database in pgAdmin, you need to open a so-called Query Tool. First, make sure that the database "eRummage" is selected in the browser tree on the left-hand side of the window. Then select Tools → Query Tool from the menu at the top of the window.
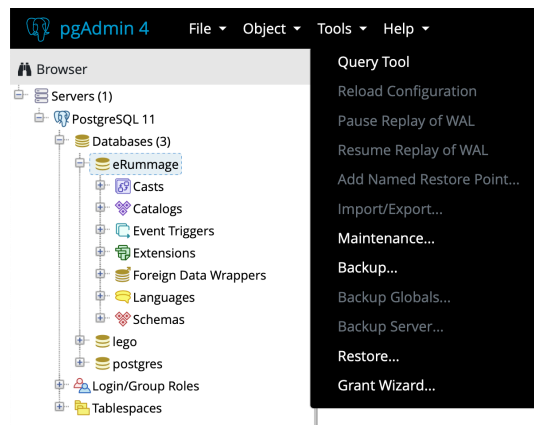
A new sub-window will open on the right-hand side of the main application window. Make sure that it says eRummage on postgres@PostgreSQL (or similar) just above the text entry.
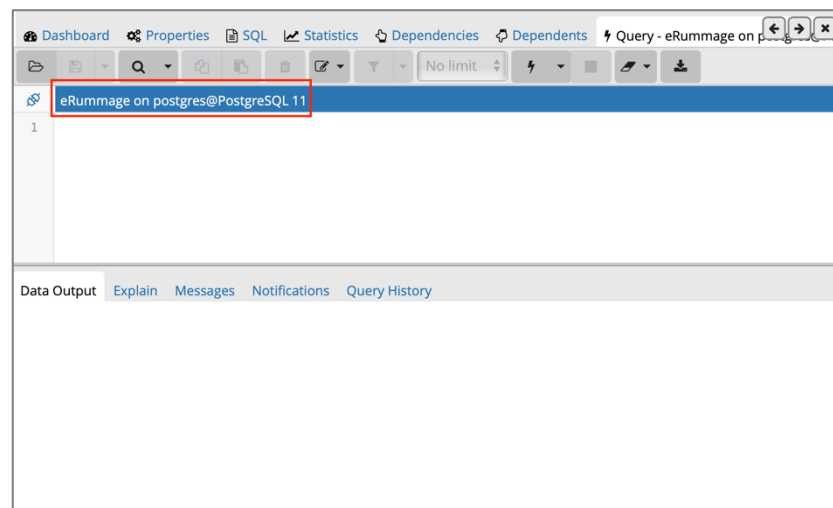
Now you are ready to create the schema of your database! Enter the corresponding SQL statements in the text entry and execute them by clicking the lighning symbol in the tool bar. Note you can execute individual SQL statements (rather than the entire file) by highlighting them before clicking the lightning symbol. Also, keep in mind that pgAdmin is a web application. Therefore, it is recommended to save your code regularly and, in particular, before executing it.

The PostgreSQL 11 documentation can be found online at https://www.postgresql.org/docs/11/. For this exercise, you will need to study the following parts.

> II. The SQL Language
>> 5. Data Definition
>>> 5.1. Table Basics
>>> 5.3. Constraints (in particular the sections on Primary and Foreign Keys)
>> 8. Data Types

Task 3.3: Inserting Data into the Online Auction Database

Once the database schema is in place, you can put some data into your database. This can also be achieved using the Query Tool. Create a database instance that contains two users who have a "bidding war" over one auction item.

For this exercise, you will need to look at the following parts of the PostgreSQL documentation.

> II. The SQL Language
> > 6. Data Manipulation
> > > 6.1. Inserting Data
> > > 6.3. Deleting Data

What happens if you try to insert a bid for an auction item that does not exist. What happens if you try to delete a user who bid on an aution?

Task 3.4: Refining the Database Schema

Auctions follow all kinds of rules that, ideally, are enfored by the database system through appropriate constraints. Which of the following rules can you enfore with the techniques you have learned so far.

1. It should not be possible to register two user accounts with the same e-mail address.
2. It should not be possible to place a bid for an item after the auction ended.
3. It should not be possible to place a bid with a lower amount than the current maximum bid.

If you can enfore one of these constraints, try to change the schema accordinly. The commands to change an existing schema are described in the following parts of the PostgreSQL documentation.

> II. The SQL Language
> > 5. Data Definition
> > > 5.5. Modifying Tables (in particular the section on Adding a Constraint)