Tierheimverband e.V.   Anfahrt   Helfer   Tiere

Aktuelles

Momentane Situation durch Corona

Die momentane Situation betrifft uns alle, doch gemeinsam schaffen wir das. Da unsere Pfleger und Pflegerinnen dauerhaft im Einsatz sind um unsere Vierbeiner zu versorgen, versuchen wir, unnötige Risiken zu minimieren.

Ab Juni können wir wieder Tiere mit TERMIN vermitteln. Interessenten müssen sich vorab bei unseren Pflegern melden und einen Termin vereinbaren. Wir bitten euch, nicht die ganze Familie mitzubringen und mit einer Mund - Nasen - Maske zum Termin zu erscheinen und weiterhin Abstand zu unseren Mitarbeitern einzuhalten.

Gassi gehen mit unseren Gassi - Gehern kann ebenfalls wieder stattfinden, allerdings auch nur nach voheriger Terminvereinbarung

Öffnungszeiten

Unsere vier Tierheime in Berlin, Düsseldorf, Mannheim und München öffen für Sie zu den hier genannten Zeiten.

Montag	8:00 - 12:00
Dienstag	8:00 - 12:00
Mittwoch	geschlossen
Donnerstag	12:00 - 17:00
Freitag	8:00 - 12:00

Database-Application "Animal-Shelter"

23.05.2020

—

Miriam Lorenz and Björn Bulkens
WWI-18-DSA

Requirement Specification

The project was to build a database for an association of animal shelters.

The association has to document some information about the shelter like the name, foundation-date, vet, and location.

Vets are recorded with their basic personal information and clinic address.

For organization tasks, every shelter has a manager who is documented with basic personal information and the shelter they manage.

To fund and support the shelters there are people donating money. Those donations are recorded through the combination of the amount, the donor, the shelter the money is donated to, and the time when the donation is made.

Each shelter is home to different animals, form different species and races.

Each animal has a name, gender, species and the time the association is already taking care of them.

There remains a record of already adopted animals and the people taking them in.

There are different suppliers of food, which can supply different types of food and to different shelters. Every food needs a short description and the supplier has to leave an IBAN and address.

If somebody wants to volunteer they can fill out an online form to register as an official supporter. If a Volunteer moves or does not want to help in the shelter any longer he/she can delete or update their entry. To declare those changes you need to provide a password.

As framework Flask will be used with the usage of flask_sqlalchemy as ORM to be independent of specific SQL-dialects.

The ORM will use SQLite instead of PostgreSQL for easier transfer of a working project and not need to worry about the SQL-server needed otherwise

All technical necessities, except python itself, to run the code are stored in a virtual environment, which comes in the folder with the source code.

ER Diagram (also provided as a separate file)

Miriam Lorenz & Björn Bulkens
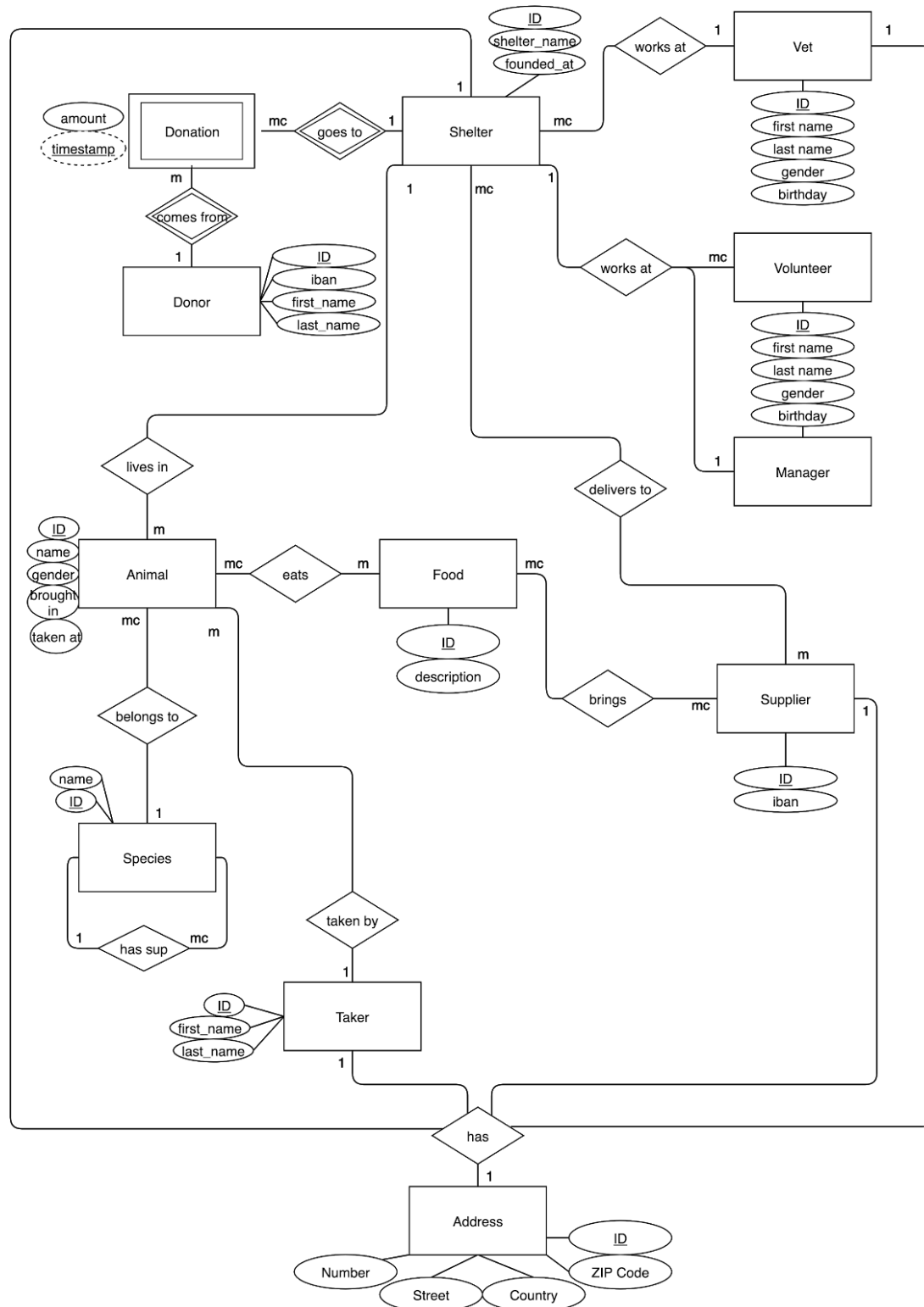
Normalization within the database

first normal form:

All information is stored as atomic values: all names are split in first and last name, amounts of money are always stored as a value in euros so there is no need to declare the type of money explicitly, etc.

All tables of automatically generated indexes as the primary key to ensure they are unique (except donations which is a weak table)

The order in which the data is stored does not matter because all columns are properly named and the row order is defined in queries explicitly

second normal form:

There should be no Partial Dependency: Each table only contains columns that are dependent on the primary key of their table.

third normal form:

There should be no transitive dependency: This normal form is not fully fulfilled because we did not put everything in a different table that could have been referenced in a separate table. Through already having an own table for addresses only we did not see the need to separate the name of a city from the rest of the table even though it can be uniquely identified by zip code, considering the size of the database.

And even though IBAN is a candidate key, it was not useful to bring the name of the donor into a separate table. (Would have been useful if we used more data that could be identified through the IBAN like Bank code and Bank account number)

As of our knowledge, this should be the only case were we (willingly) broke the third normal form. And we know that we usually would create 2 tables addresses (with everything it has now, except city) and a second table mapping zips and cities and then create a materialized view we can query over. We hope that you recognize that we this.

Indexing

We created two indexes:

- the species of animals because that's the separation factor between different animals in shelters

- the IBAN of donors because bookkeeping requires to document those regularly

For the rest, we used automatically generated IDs on almost every table, which has both up and downsides: The primary key index is slightly smaller because storage is combined with the data pages themselves. On the other hand, the data needs to be in order on the disk, which can complicate inserts and updates. On the other hand, auto-increment guarantees that all new rows go at the end of the data. But the upside outweighed the downsides considering the size of the project.

Query.sql file

We provided a separate file with a number of statements that can be run on the database.

We filled the database with data using a GUI because we could provide CSV-files, so we just provided a premade .db file to run the application on in the ZIP.

Creation.sql file

To recreate an empty replica of the database we provided a full-on initialization in SQL DDL including views and indexes.

Application

I. Link to the presentation of source code and the application

https://youtu.be/9MGHIfCuOHk

(uploaded as not listed youtube-video on my channel)

[The website has seen updates in the GUI since the recording was made, so make sure to check it out yourself]

II. The Source code is also provided in a zip-folder

III. To run the application the provided zip-folder contains a virtual environment with flask, bootstrap, and jinja2. Otherwise, just packages were also stored in the requirements.txt

IV. database.py is the python file to initialize the database and run the application itself

V. production.db is the database file which we worked with

VI. Queries are used in different forms to show different ways of doing similar things:

    A. Raw SQL-statements in text format

    B. Separate Flask-SQLalchemy queries, which are used in one table combined through python code in the jinja2 template

VII.   An update and a delete, as well as an insert functionality, have been implemented

VIII.   This project can also be found as a repository on GitHub (but we might work further on the project after the submission deadline, so use the provided ZIP for evaluation):

https://github.com/FatManWalking/Animal_Shelter.git