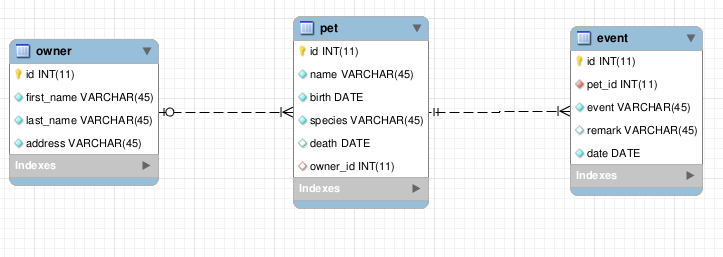
**REST and REST APIs**

General part

* Elaborate on some of the characteristics of REST, like: Stateless, Cacheable, Layered System, Uniform Interface etc.
* Explain the benefit(s) from having a backend that exposes all data to clients via a REST-API
* Elaborate on how JSON or XML supports communication between subsystems, even when the subsystems are implemented on diﬀerent platforms.

Practical part (requires the script-file: [**petHospitalSql**](https://github.com/Cphdat3sem2017f/StartcodeExercises/blob/master/REST/pethospital.zip)*+ gson dependency*)

The starting point for this exercise is a pet hospital example database.

The script setups the database with the three tables sketched in this diagram, and data for different animals, owners and events

**Getting Started:**

* Create a new NetBeans Web project as the starting point for this exercise
* Execute the script *pethospital.sql* from MySQL Workbench, to setup the database
* Use the NetBeans Wizard to create a set of matching Entity-classes

**Tasks (**All transfer of data between the API and Clients must be encapsulated in JSON)

1. Implement a simple facade class with a method that will return all Pet's, and demonstrate the method
2. Implement a Rest service to get the total Number of pets formatted like: {"petCount":4} (just call size() on the result from the method above)
3. Use the method from 1) to implement a REST service to get a json-list of all pets, with *id*, *name*, *birth, species* and the *first\_name* and *last\_name* of the owner

Implement as many as you have time for of the following REST endpoints

1. Get a list of all living pets.
2. Get a list of all pets that had an event on a given day
3. Create a new event for an existing pet

For each service you should: Define the API call (URI) and the (JSON) format used to transfer data back and forth between Client and Server.

Test the REST API using a browser for the GET methods and Postman for the POST method (or Rest Assured if you prefer).

If you have time: create an HTML page that could fetch and show data (using javascript or react) for some of the methods created above.