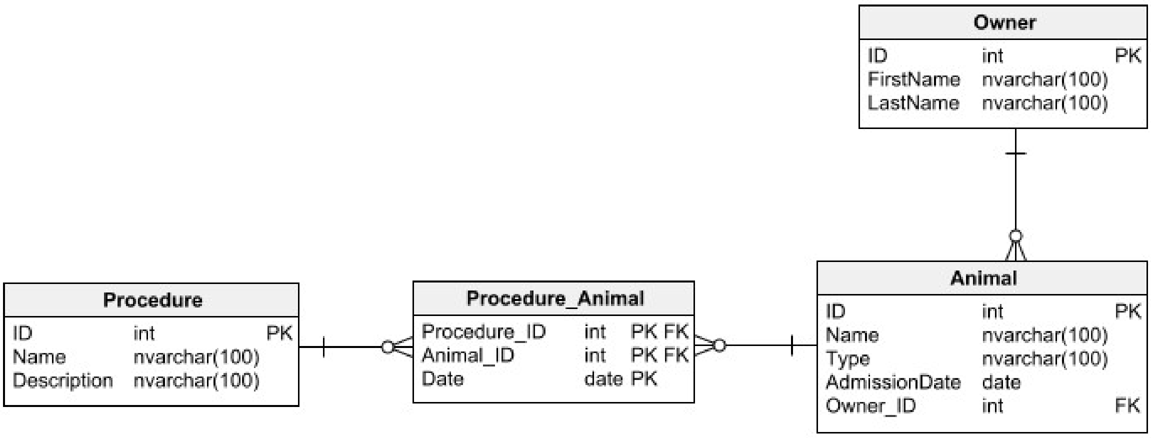
Example test 1

In this assignment you have to design a web API application that will support the database presented in the diagram. We are creating a service that supports a local animal clinic.

Database

The database we use is presented below. The script that creates the tables can be found in the file **example-test.sql**.



Owner

ID	int	PK
FirstName	nvarchar(100)	
LastName	nvarchar(100)	

Procedure

ID	int	PK
Name	nvarchar(100)	
Description	nvarchar(100)	

Procedure_Animal

Procedure_ID	int	PK FK
Animal_ID	int	PK FK
Date	date	PK

Animal

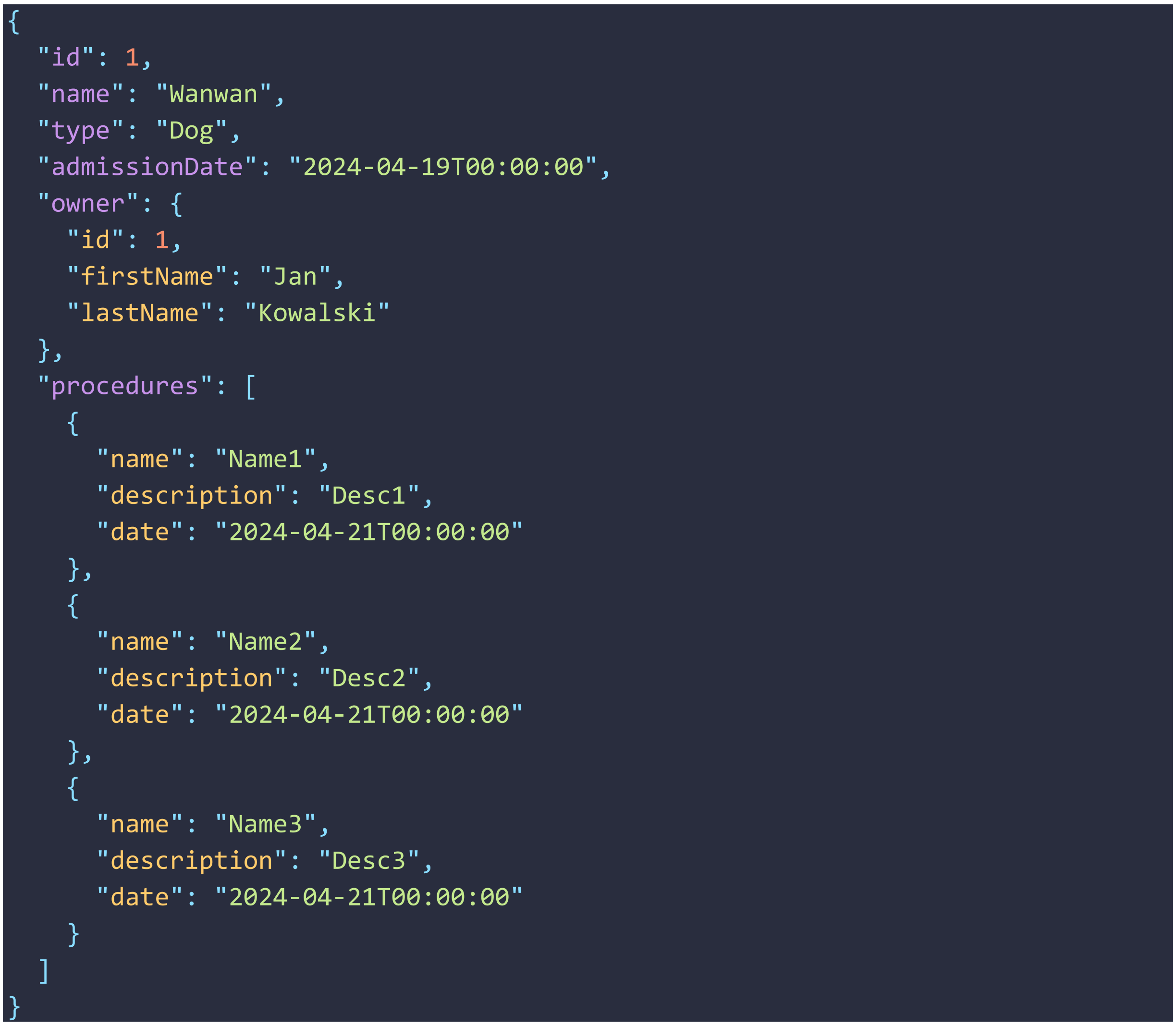
ID	int	PK
Name	nvarchar(100)	
Type	nvarchar(100)	
AdmissionDate	date	
Owner_ID	int	FK

Endpoints

Design an endpoint that will return information about the animal with the given ID. Along with the animal's data, data about the owner and all procedures related to the animal should also be returned. If there is no animal with the given ID, the corresponding HTTP code should be returned.

Endpoint should respond to a query to the api/animals address, e.g.. HTTP GET http://localhost:5000/api/animals/1

Examples of the data that will be returned:



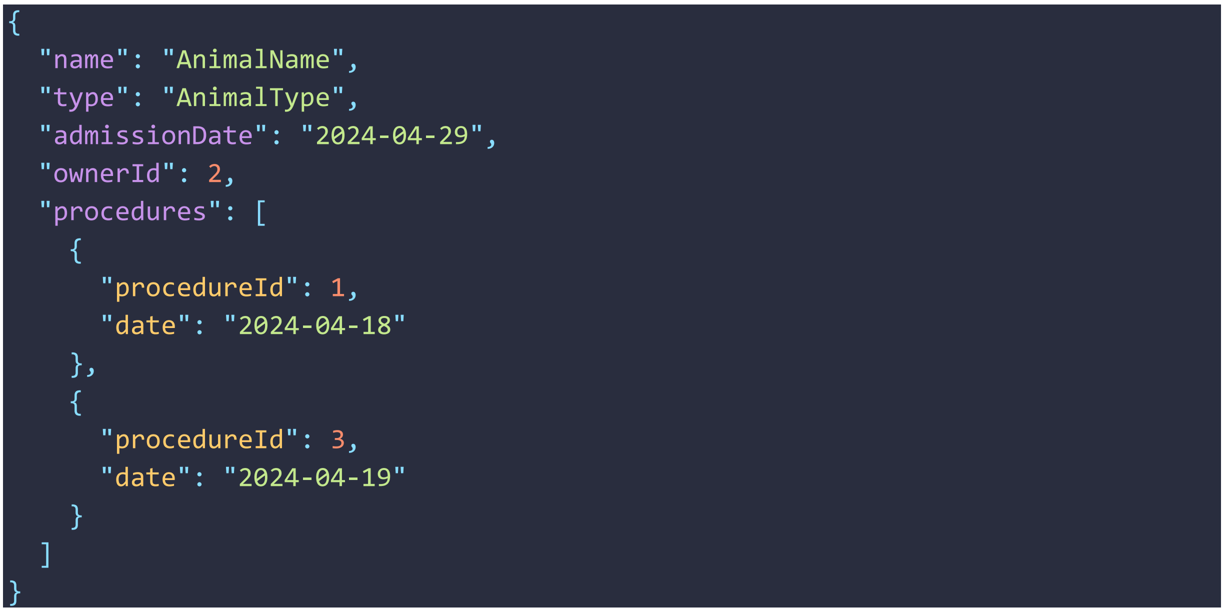
```
{
  "id": 1,
  "name": "Wanwan",
  "type": "Dog",
  "admissionDate": "2024-04-19T00:00:00",
  "owner": {
    "id": 1,
    "firstName": "Jan",
    "lastName": "Kowalski"
  },
  "procedures": [
    {
      "name": "Name1",
      "description": "Desc1",
      "date": "2024-04-21T00:00:00"
    },
    {
      "name": "Name2",
      "description": "Desc2",
      "date": "2024-04-21T00:00:00"
    },
    {
      "name": "Name3",
      "description": "Desc3",
      "date": "2024-04-21T00:00:00"
    }
  ]
}
```

Design an endpoint that will allow us to add the animal to the database. Endpoint should allow us to accept both the animal's data, as well as optionally a list of procedures it has undergone. In this case, we would like to add both the animal and the association table entries in a single transaction. Be sure to perform the described steps in a single transaction. In the database, all fields are required. Make sure that the owner and procedures exist in the database.

Endpoint should respond to a query to the api/animals/ address, e.g..

HTTP POST http://localhost:5000/api/animals/

Przykładowe dane jakie są przesłane wraz z żądaniem:



```json
{
  "name": "AnimalName",
  "type": "AnimalType",
  "admissionDate": "2024-04-29",
  "ownerId": 2,
  "procedures": [
    {
      "procedureId": 1,
      "date": "2024-04-18"
    },
    {
      "procedureId": 3,
      "date": "2024-04-19"
    }
  ]
}
```

Scoring

Each endpoint is calculated at **10 points**. That is, **20 points** can be earned for the whole.

Points may be deducted for:

- Incorrect or illegible variable names: **-2 pts**
- Incorrect HTTP codes: **-4 pts**
- Failure to separate database communication into a separate file/service/repository: **-10 pts**
- No dependency injection: **-8 pts**

Notes

- A program that does not compile - **0 pts**
- A program that is plagiarized – **2 as the end mark for the course**