

Przykładowe kolokwium 2_2

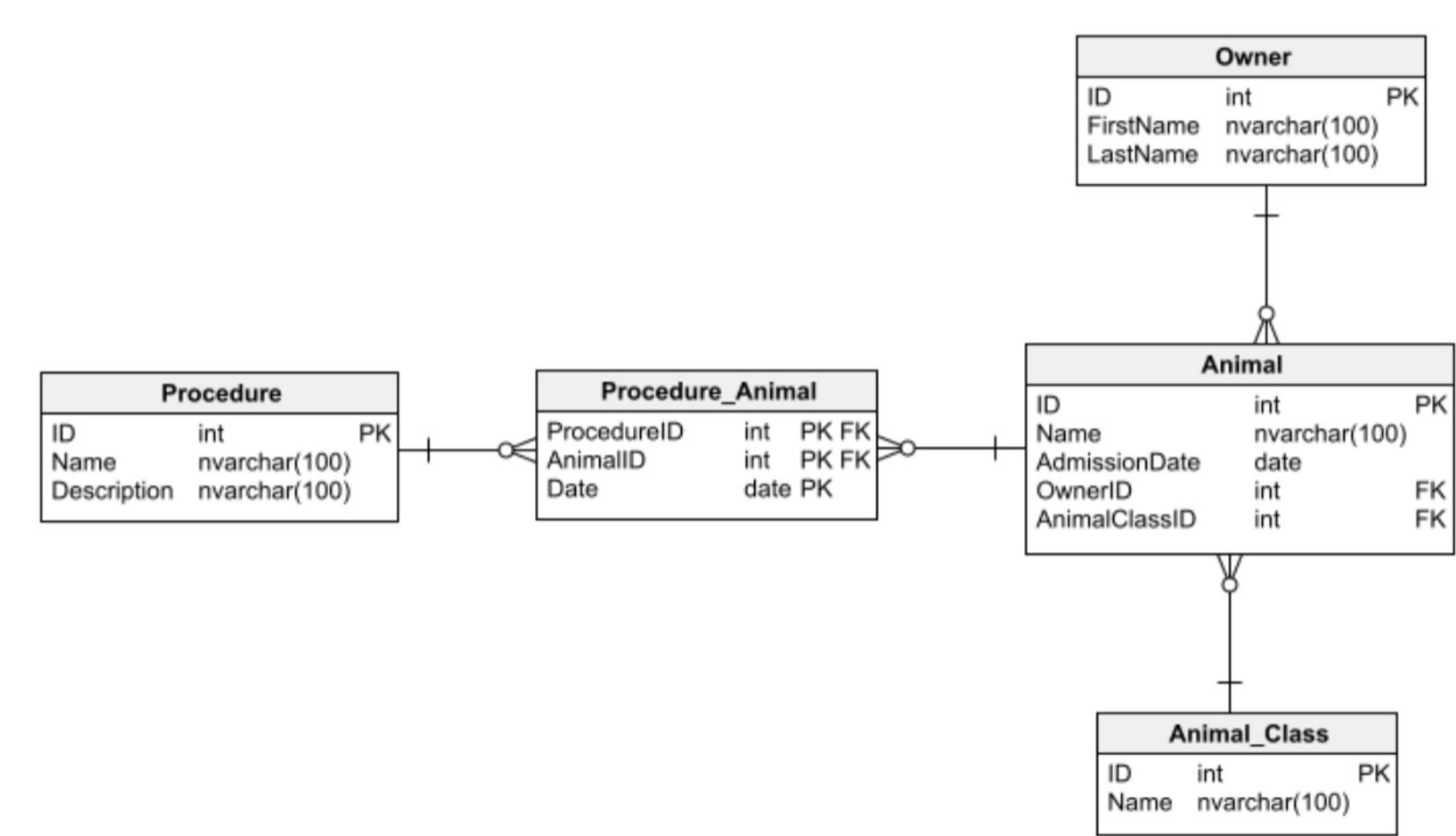
Zadania powinny zostać wykonane w projekcie Web API z wykorzystaniem frameworka ORM – EntityFrameworkCore.

Baza danych

Poniżej zaprezentowany jest diagram bazy danych wykorzystywanej w poniższych zadaniach.

Kolokwium powinno zostać zrealizowane na bazie lokalnej lub uczelnianej przy pomocy Entity Framework w wersji Core.

Należy zadbać o dodanie przykładowych danych do odpowiednich tabel.



Końcówki

1. Zaprojektuj końcówkę, która będzie zwracać informację na temat zwierzęcia o podanym ID. Wraz z danymi zwierzęcia, powinny też być zwrócone dane na temat właściciela oraz przeprowadzonych procedur. Jeśli nie ma zwierzęcia o podanym identyfikatorze, powinien zostać zwrócony odpowiedni kod.

Końcówka powinna reagować na zapytanie pod adres `api/animals` np.

HTTP GET `http://localhost:5000/api/animals/1`

Przykładowe dane jakie zostaną zwrócone:

```
{
  "id": 1,
  "name": "AnimalName",
  "animalClass": "Mammal",
  "admissionDate": "2023-05-13",
  "owner": {
    "id": 3,
    "firstName": "Jan",
    "lastName": "Kowalski"
  },
  "procedures": [
    {
      "name": "Endoscopy",
      "date": "2023-05-14"
    },
    {
      "name": "Cholecystectomy",
      "date": "2023-05-15"
    }
  ]
}
```

2. Zaprojektuj końcówkę, która pozwoli nam dodać zwierzę do bazy danych. Końcówka powinna pozwolić na przyjęcie zarówno danych zwierzęcia, jak również opcjonalnie listy zabiegów jakim zostało poddane. W takim wypadku chcielibyśmy w ramach jednej transakcji dodać zarówno zwierzę, jak również wpisy do tabeli asocjacyjnej. Pamiętaj o wykonaniu opisanych czynności w ramach jednej transakcji. W bazie danych wszystkie pola są wymagane. Należy sprawdzić, czy właściciel, klasa zwierzęcia i procedury istnieją w bazie.

Końcówka powinna reagować na zapytanie pod adres `api/animals/` np.

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

Przykładowe dane jakie są przesłane wraz z żądaniem:

```
{
  "id": 1,
  "name": "AnimalName",
  "animalClass": "Bird",
  "admissionDate": "2023-04-13",
  "ownerID": 2,
  "procedures": [
    {
      "procedureID": 1,
      "date": "2023-05-14"
    },
    {
      "procedureID": 3,
      "date": "2023-05-15"
    }
  ]
}
```

Punktacja

Każda końcówka jest wyliczana na **10 punktów**. Czyli za całość można zdobyć **20 punktów**.

Punkty mogą być odjęte za:

- Niepoprawne lub nieczytelne nazwy zmiennych: **-2 pkt**
- Niepoprawne kody HTTP: **-4 pkt**
- Brak wydzielenia komunikacji bazy danych do oddzielnego pliku/serwisu/repozytorium: **-10 pkt**
- Brak wstrzykiwania zależności: **-8 pkt**

Uwagi

- Kod należy spushować na Github Classrooma. Rozwiązanie powinno zostać umieszczone na platformie **max. 5 min** po zakończeniu zajęć. Projekty umieszczone po czasie nie będą brane pod uwagę podczas sprawdzania (!)
- Kolokwium powinno być rozwiązane przy użyciu podejścia **Code-First**
- Rozwiązanie kolokwium bez tabel w bazie danych - **0 pkt**
- Rozwiązanie kolokwium bez przykładowych danych - **-50% punktów**
- Oddanie niekompilującego się kolokwium - **0 pkt**
- Wykrycie plagiatu - **2 z przedmiotu** bez możliwości poprawy