

Описание предметной области "Кинологический клуб"

Пусть требуется разработать информационную систему для автоматизации учета собак и их владельцев в кинологическом клубе. Система должна предусматривать ведение базы данных пород собак, зарегистрированных владельцев и их питомцев, а также учет достижений и наград.

Структура данных (Документы на входе)

Отношение "Породы"

Каждая порода характеризуется следующими параметрами:

- Группа пород (терьеры, таксы и прочие)
- Название породы (уникальное)

Отношение "Владельцы"

На каждого владельца заносятся следующие сведения:

- Уникальный идентификатор
- Фамилия, имя, отчество
- Адрес проживания
- Контактный телефон

Отношение "Собаки"

Каждая собака, зарегистрированная в клубе, характеризуется следующими параметрами:

- Уникальный идентификатор (6-значный числовой код)
- Кличка (обязательное поле)
- Владелец (ссылка на таблицу "Владельцы")
- Дата рождения (обязательное поле)
- Пол (только 'м' или 'ж')
- Порода (ссылка на таблицу "Породы")
- Отец (ссылка на таблицу "Собаки", необязательное поле)
- Мать (ссылка на таблицу "Собаки", необязательное поле)
- Описание (до 300 символов)
- Дата смерти (необязательное поле)

Отношение "Награды"

Учет полученных наград:

- Собака (ссылка на таблицу "Собаки")
- Награда (описание награды)
- Дата получения награды

Предусмотреть следующие ограничения на информацию в системе:

1. Каждая собака должна быть зарегистрирована на конкретного владельца.
2. Пол собаки может принимать только значения 'м' или 'ж'.
3. Собака может иметь неограниченное количество наград.
4. Порода должна быть предварительно занесена в справочник пород.
5. Дата рождения является обязательной для регистрации собаки.
6. Дата смерти указывается только для умерших собак.

Пользователи системы и их задачи

С данной информационной системой должны работать следующие группы пользователей:

- Администраторы клуба;
- Владельцы собак;
- Гости клуба.

Администраторы клуба

При работе с системой администраторы должны иметь возможность решать следующие задачи:

1. Регистрировать новых владельцев в клубе.
2. Вносить в базу данных новые породы собак.
3. Регистрировать собак с полной информацией о родословной.
4. Вести учет наград и достижений собак.
5. Формировать отчеты и статистику по породам и владельцам.
6. Вести учет родословных связей между собаками.

Владельцы собак

Владельцы должны иметь возможность:

1. Просматривать информацию о всех (и своих) собаках.
2. Ознакамливаться с родословной своих питомцев.
3. Просматривать информацию о полученных наградах.
4. Получать информацию о предстоящих мероприятиях клуба.

Гости клуба

Гости должны иметь возможность:

1. Просматривать каталог пород, представленных в клубе.
2. Ознакамливаться с информацией о достижениях собак клуба.
3. Получать контактную информацию клуба.

Функциональные требования

Учет собак

1. Регистрация новых собак с обязательным указанием клички, даты рождения, пола и породы.
2. Ведение родословных связей между собаками.
3. Учет даты смерти для умерших собак.

Учет владельцев

1. Регистрация новых владельцев с контактной информацией.
2. Отслеживание количества собак у каждого владельца.

Учет наград

1. Регистрация наград с указанием даты получения.
2. Привязка наград к конкретным собакам.

Отчетность и аналитика

1. Формирование списков собак по различным критериям (пол, возраст, порода).
2. Статистика по породам (количество собак, даты рождения).
3. Выявление владельцев с двумя и более собаками.
4. Определение пород, не представленных в клубе.

Документы на выходе – статистика о мероприятиях и наградах.

Анализ популярности пород в клубе

- Отчет по наградам и достижениям
- Прогноз развития пород в клубе

ER диаграмма

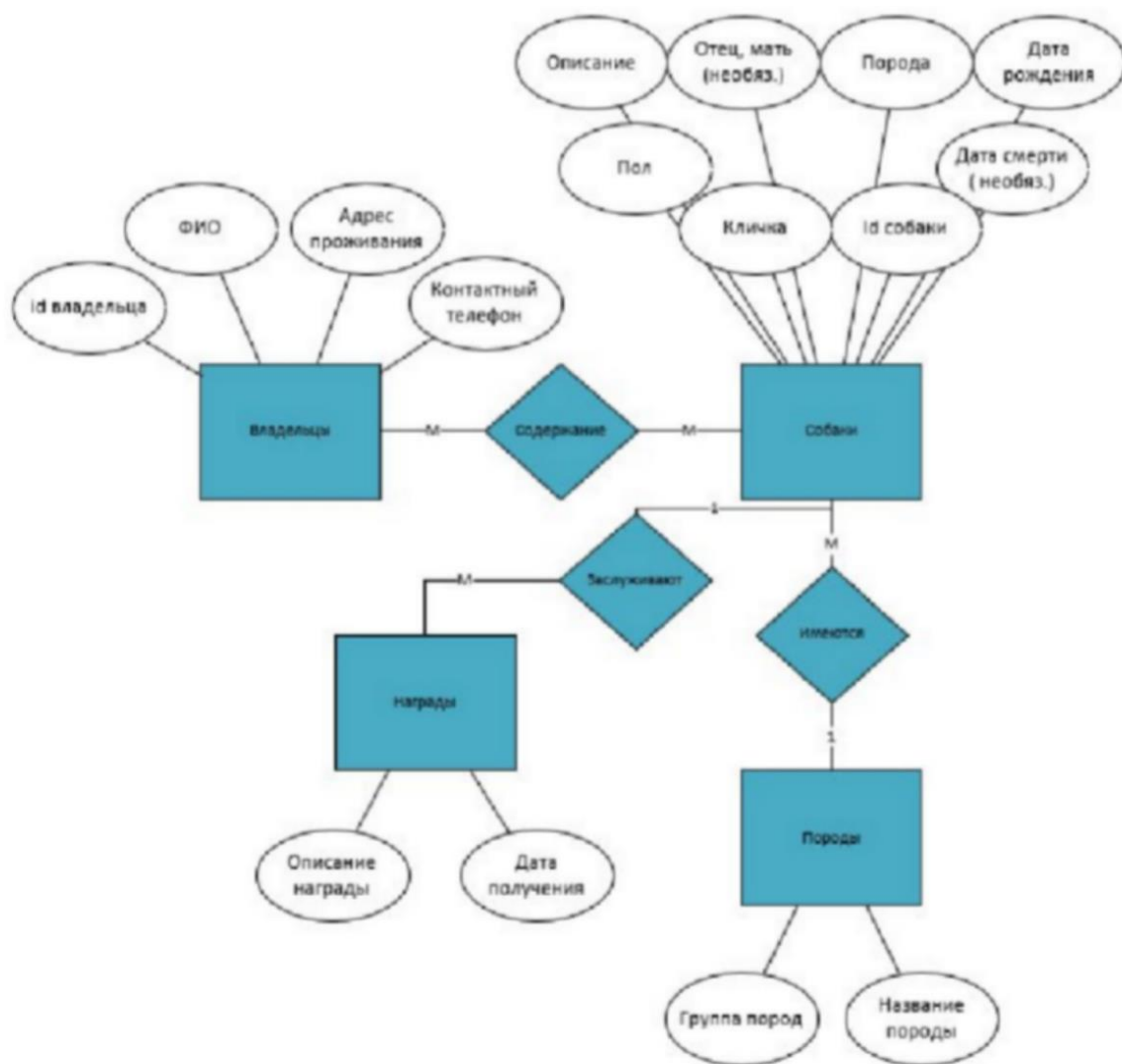
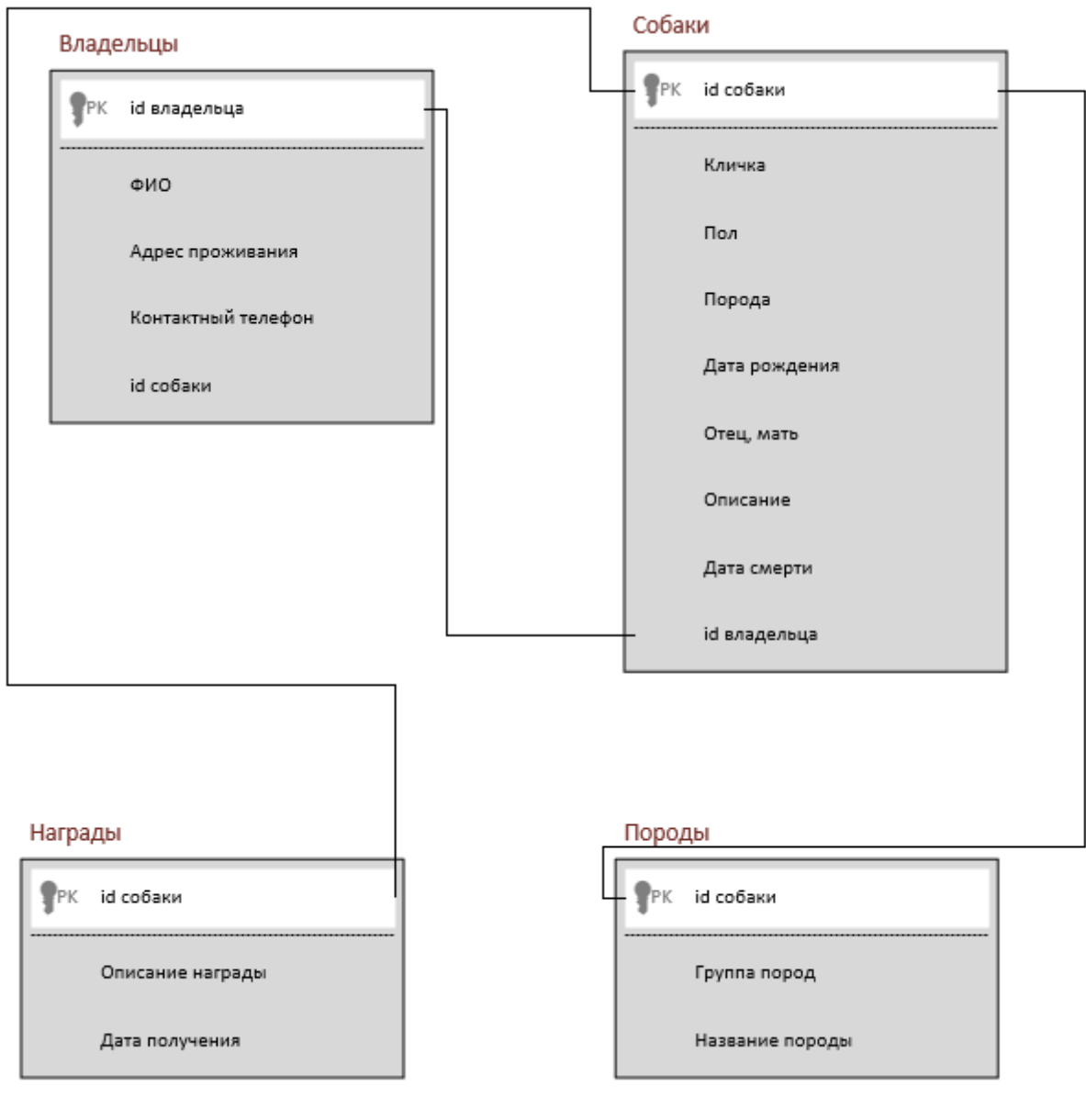


Схема отношений



```
create database if not exists Kennelclub
character set utf8mb4
collate utf8mb4_0900_ai_ci;
```

```
create schema if not exists Kennelclub
default character set cp1251;
```

```
use kennelclub;
```

```
create table if not exists dogs(
    id_dog int not null auto_increment,
    nickname_dog varchar(20) not null,
    gender_dog varchar(10) not null,
    species_dog varchar(45) not null,
    date_birth_dog date not null,
    mom_dog varchar(20),
    dad_dog varchar(20),
    description varchar(300),
    date_death_dog date,
    id_owner int not null,
    primary key(id_dog),
    constraint fk_dogs_owners
        foreign key (id_owner)
        references owners(id_owner)
        on delete no action
        on update no action
);
```

```
create table if not exists owners(
    id_owner int not null auto_increment,
    fio_owner varchar(45) not null,
    address_owner varchar(45) not null,
    number_phone int,
    primary key (id_owner)
);
```

```
create table if not exists species(
    id_species int not null auto_increment,
    id_dog int not null,
    group_species varchar(45) not null,
    name_species varchar(45) not null,
    primary key (id_species),
    constraint fk_species_dogs
        foreign key (id_dog)
        references dogs(id_dog)
        on delete no action
        on update no action
);
```

```
create table if not exists awards(
    id_award int not null auto_increment,
    id_dog int not null,
    descriptions_awards varchar(300) not null,
    data_receipt_awards date,
    primary key (id_award),
    constraint fk_awards_dogs
        foreign key (id_dog)
        references dogs(id_dog)
        on delete no action
        on update no action
);
```