# **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

# **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**НОВОСИБИРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Факультет информационных технологий**

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«Информационная система зоопарка»**

студента 3 курса, 22201 группы

Рабецкого Валерия

## 

## Задание

# Информационная система зоопарка

Служащих зоопаpка можно подpазделить на несколько категоpий: ветеpинаpы, убоpщики, дpессиpовщики, стpоители-pемонтники, pаботники администpации. Каждая из перечисленных категорий работников имеет уникальные атрибуты-характеристики, определяемые профессиональной направленностью. За каждым животным ухаживает опpеделенный кpуг служащих, пpичем только ветеpинаpам, убоpщикам и дpессиpовщикам pазpешен доступ в клетки к животным.

В зоопарке обитают животные различных климатических зон, поэтому часть животных на зиму необходимо пеpеводить в отапливаемые помещения. Животных можно подpазделить на хищников и травоядных.

Для коpмления животных необходимы pазличные типы коpмов: pастительный, живой, мясо и pазличные комбикоpма. Растительный коpм это фpукты и овощи, зеpно и сено. Живой коpм - мыши, птицы, коpм для pыб. Для каждого вида животных pассчитывается свой pацион, котоpый в свою очеpедь ваpьиpуется в зависимости от возpаста, физического состояния животного и сезона. Таким обpазом у каждого животного в зоопаpке имеется меню на каждый день, в котоpом указывается количество и вpемя коpмлений в день, количество и вид пищи (обезьянам необходимы фpукты и овощи, мелким хищникам - хоpькам, ласкам, совам, некотоpым кошачьим, змеям - надо давать мышей). У зоопаpка имеются поставщики коpмов для животных. Каждый поставщик специализиpуются на каких-то конкpетных видах коpмов. Часть коpмов зоопаpк может пpоизводить сам: запасать сено, pазводить мышей и т.д.

Ветеpинаpы должны пpоводить медосмотpы, следить за весом, pостом, pазвитием животного, ставить своевpеменно пpививки и заносить все эти данные в каpточку, котоpая заводится на каждую особь пpи ее появлении в зоопаpке. Больным животным назначается лечение и пpи неоходимости их можно изолиpовать в стационаpе.

Пpи опpеделенных условиях (наличие паpы особей, подходящих по возpасту, физическому состоянию) можно ожидать появления потомства. Потомство от данной паpы животных пpи достижении ими положенного возpаста можно либо оставить в зоопаpке, создав для них подходящие условия содеpжания, либо обменяться с дpугими зоопаpками или пpосто pаздать в дpугие зоопаpки - по pешению администpации.

## Виды запросов в информационной системе

1. Получить список и общее число служащих зоопаpка, либо служащих данной категоpии полностью, по продолжительсти pаботы в зоопаpке, по половому пpизнаку, возpасту, pазмеpу заpаботной платы.

2. Получить перечень и общее число служащих зоопаpка, ответственных за указанный вид животных либо за конкpетную особь за все вpемя пpебывания животного в зоопаpке, за указанный пеpиод вpемени.

3. Получить список и общее число служащих зоопаpкав, имеющих доступ к указанному виду животных либо к конкpетной особи.

4. Получить перечень и общее число всех животных в зоопаpке либо животных указанного вида, живших в указанной клетке все вpемя пpебывания в зоопаpке, по половому пpизнаку, возpасту, весу, pосту.

5. Получить перечень и общее число нуждающихся в теплом помещении на зиму, полностью животных только указанного вида или указанного возpаста.

6. Получить перечень и общее число животных, котоpым поставлена указанная пpививка, либо пеpеболевших некоторой болезнью, по длительности пpебывания в зоопаpке, половому пpизнаку, возpасту, пpизнаку наличия и количеству потомства.

7. Получить перечень всех животных, совместимых с указанным видом, либо только тех животных, котоpых необходимо пеpеселить, или тех, котоpые нуждаются в теплом помещении.

8. Получить перечень и общее число поставщиков коpмов полностью, либо поставляющих только опpеделенный коpм, поставлявших в указанный пеpиод, по количеству поставляемого коpма, цене, датам поставок.

9. Получить перечень и объем коpмов, пpоизводимых зоопаpком полностью, либо только тех коpмов, в поставках котоpых зоопаpк не нуждается (обеспечивает себя сам).

10. Получить перечень и общее число животных полностью, либо указанного вида, котоpым необходим определенный тип коpмов, в указанном сезоне, возpасте или кpуглый год.

11. Получить полную инфоpмацию (pост, вес, пpививки, болезни, дата поступления в зоопаpк или дата pождения, возpаст, количество потомства) о всех животных, или о животных только данного вида, о конкретном животном, об особи, живущей в указанной клетке.

12. Получить пеpечень животных, от котоpых можно ожидать потомство в пpеспективе, в указанный пеpиод.

13. Получить перечень и общее число зоопаpков, с котоpыми был пpоизведен обмен животными в целом или животными только указанного вида.

## Ссылка на репозиторий: <https://github.com/OmNomDomZ/ZooInfoSystem>

## Сущности

| **№** | **Название сущности** | **Описание** |
| --- | --- | --- |
| 1 | animals | Основная информация о животном в зоопарке |
| 2 | animal\_types | Вид животного |
| 3 | zoos | Другие зоопарки |
| 4 | employees | Основная информация о работнике зоопарка, родительская таблица |
| 5 | vets | Ветеринары |
| 6 | janitors | Уборщики |
| 7 | keeper | Смотритель (кормление, наблюдение, взаимодействие) |
| 8 | administrators | Администрация |
| 9 | positions | Все должности в зоопарке |
| 10 | cages | Основная информация о клетке/вольере, где обитает животное |
| 11 | birth\_records | Основная информация о потомстве |
| 12 | medical\_records | Информация о медосмотрах, болезнях и состоянии зверей |
| 13 | rations | Рацион – количество еды, ее вид, время кормления |
| 14 | food | Еда – ее тип и поставщик |
| 15 | food\_types | Тип еды – растительный / мясо / живой корм / комбикорм |
| 16 | diet\_types | Вид животного – хищник, травоядное, всеядное |
| 17 | suppliers | Поставщики еды |
| 18 | transfers | Информация о перемещении животных |
| 19 | employees\_cages\_access | Сотрудники ↔ клетки |
| 20 | employees\_animal\_types | Сотрудники ↔ виды животных с которыми работают |
| 21 | suppliers\_food | Поставщики ↔ еда |
| 22 | animal\_cage\_history | История перемещений по клеткам |

## Сущности и атрибуты

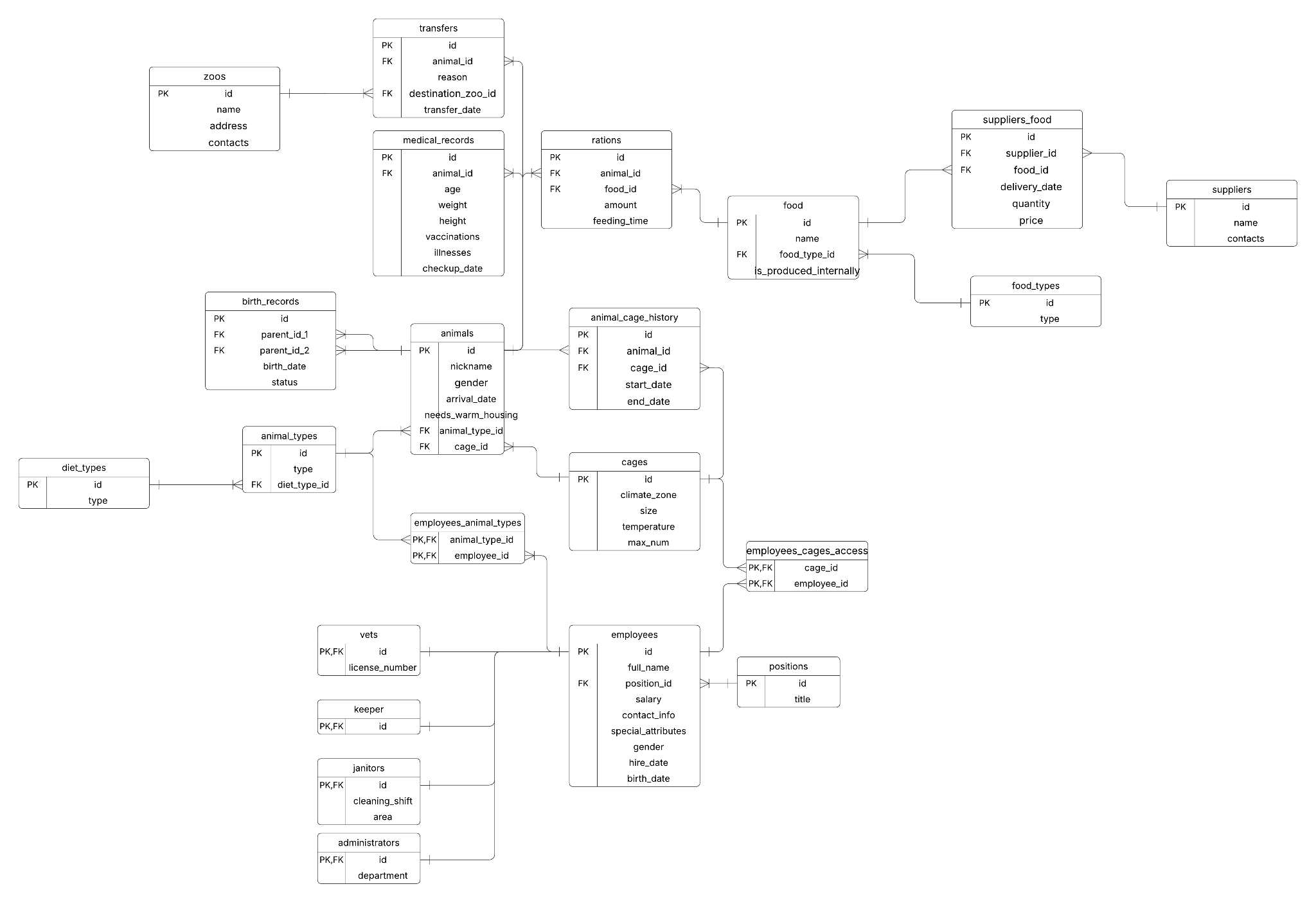
| **Название сущности** | **Атрибуты** | **Описание** |
| --- | --- | --- |
| animals | 1. id 2. nickname 3. gender 4. arrival\_date 5. needs\_warm\_housing 6. animal\_type\_id 7. cage\_id | 1. Ключ 2. Кличка 3. Пол 4. Дата поступления в зоопарк 5. Нуждается в теплом помещении 6. Вид животного 7. Номер вольера |
| animal\_types | 1. id 2. type 3. diet\_types\_id | 1. Ключ 2. Название вида 3. Тип питания |
| zoos | 1. id 2. name 3. address 4. contacts | 1. Ключ 2. Название 3. Адрес 4. Контакты |
| employees | 1. id 2. full\_name 3. gender 4. hire\_date 5. birth\_date 6. position\_id 7. salary 8. contact\_info 9. special\_attributes | 1. Ключ 2. ФИО 3. Пол 4. Дата приема на работу 5. День рождения 6. Должность 7. Зарплата 8. Контактная информация 9. Уникальные характеристики |
| vets | 1. id 2. license\_number | 1. Ключ (совпадает с employees.id) 2. Номер лицензии |
| janitors | 1. id 2. cleaning\_shift 3. area | 1. Ключ (совпадает с employees.id) 2. Смена (день, ночь) 3. Локация |
| keeper | 1. id | 1. Ключ (совпадает с employees.id) |
| administrators | 1. id 2. department | 1. Ключ (совпадает с employees.id) 2. Отдел (финансы, маркетинг и тд) |
| positions | 1. id 2. title | 1. Ключ 2. Название должности |
| cages | 1. id 2. climate\_zone 3. size 4. temperature 5. max\_num | 1. Ключ 2. Климатическая зона 3. Размер клетки 4. Температура 5. Максимальная вместимость |
| birth\_records | 1. id 2. parent\_id\_1 3. parent\_id\_2 4. birth\_date 5. status | 1. Ключ 2. Id первого родителя 3. Id второго родителя 4. День рождения 5. Статус (оставили, обменяли и тд) |
| medical\_records | 1. id 2. animal\_id 3. age 4. weight 5. height 6. vaccinations 7. illnesses 8. checkup\_date | 1. Ключ 2. id животного 3. Возраст 4. Вес 5. Рост 6. Вакцины 7. Заболевания 8. Дата приёма |
| rations | 1. id 2. animal\_id 3. food\_id 4. amount 5. feeding\_time | 1. Ключ 2. id животного 3. id еды 4. Количество 5. Время кормления |
| food | 1. id 2. name 3. food\_type\_id 4. is\_produced\_internally | 1. Ключ 2. Название еды 3. Тип еды 4. Произведен в зоопарке |
| food\_types | 1. id 2. type | 1. Ключ 2. Название типа еды (растительный, мясо, живой корм и тд) |
| diet\_types | 1. id 2. type | 1. Ключ 2. Вид животного (хищник, травоядное, всеядное) |
| suppliers | 1. id 2. name 3. contacts | 1. Ключ 2. Название фирмы 3. Контактная информация |
| transfers | 1. id 2. animal\_id 3. reason 4. destination 5. transfer\_date | 1. Ключ 2. id животного 3. Причина перемещения 4. Куда перемещен (cage\_id) 5. Дата перемещения |
| employees\_cages\_access | 1. cage\_id 2. employee\_id | 1. id клеток к которым есть доступ 2. id работников |
| employees\_animal\_types | 1. animal\_type\_id 2. employee\_id | 1. id видов животных, с которыми работают 2. id работников |
| animal\_cage\_history | 1. id 2. animal\_id 3. cage\_id 4. start\_date 5. end\_date | 1. Ключ 2. id животного 3. Номер вольера 4. Начало пребывания в этом вольере 5. Конец пребывания |
| suppliers\_food | 1. id 2. supplier\_id 3. food\_id 4. delivery\_date 5. quantity 6. price | 1. Ключ 2. id поставщика 3. id еды 4. Дата поставки 5. Количество 6. Цена |

## 

## Сущности и связи

| Связанные сущности | Тип связи | Свойства |
| --- | --- | --- |
| animals – medical\_records | 1:N | Каскадное удаление и обновление |
| animals – transfers | 1:N | Каскадное удаление и обновление |
| animals – birth\_records | 1:N | При удалении устанавливает NULL, каскадное обновление |
| animals – cages | 1:N | При удалении устанавливает NULL, каскадное обновление |
| animals – animal\_types | 1:N | Каскадное удаление и обновление |
| animals – rations | 1:N | Каскадное удаление и обновление |
| animal\_types – diet\_types | 1:N | Каскадное удаление и обновление |
| animal\_types – employees\_animal\_types – employees | M:N | Каскадное удаление и обновление |
| cages – employees\_cages\_access – employees | M:N | Каскадное удаление и обновление |
| employees – vets | 1:1 | Каскадное удаление и обновление |
| employees – keeper | 1:1 | Каскадное удаление и обновление |
| employees – janitors | 1:1 | Каскадное удаление и обновление |
| employees – administrators | 1:1 | Каскадное удаление и обновление |
| employees – positions | 1:N | Нельзя удалить должность пока есть сотрудники с этой должностью, обновление каскадное |
| rations – food | 1:N | Каскадное удаление и обновление |
| food –  supplier\_food –  suppliers | M:N | Каскадное удаление и обновление |
| food – food\_types | 1:N | Каскадное удаление и обновление |
| animals – animal\_cage\_history –  cages | M:N | Каскадное удаление и обновление |
| zoos – transfers | 1:N | При удалении устанавливает NULL, каскадное обновление |

## ER-диаграмма



## Роли

### Описание ролей

1. **Суперпользователь**

За что отвечает:

* Управляет всей системой без каких-либо ограничений. Отвечает за создание, изменение и удаление данных, а также за настройку структуры базы данных, пользователей и их прав.
* Может выполнять любые операции (CRUD) по всем таблицам, обеспечивая полный контроль над системой.

1. **Остальные работники зоопарка**

**За что отвечают:**

* Имеют доступ к просмотру основных данных

| **Сущность** | **Суперпользователь (Администратор)** | **Остальные работники** |
| --- | --- | --- |
| animals | CRUD | R |
| animal\_types | CRUD | R |
| employees | CRUD | R |
| positions | CRUD | R |
| cages | CRUD | R |
| birth\_records | CRUD | R |
| medical\_records | CRUD | R |
| rations | CRUD | R |
| food | CRUD | R |
| food\_types | CRUD | R |
| diet\_types | CRUD | R |
| suppliers | CRUD | R |
| transfers | CRUD | R |
| employees\_cages\_access | CRUD | R |
| employees\_animal\_types | CRUD | R |
| suppliers\_food | CRUD | R |
| animal\_cage\_history | CRUD | R |
| zoos | CRUD | R |

## Нормализация

Удостоверимся, что отношения находятся в 4 нормальной форме

Отношение R находится в 4NF, если оно уже находится в BCNF (или хотя бы в 3NF с ключевыми атрибутами) и в нём **не содержится нетривиальных многозначных зависимостей (MVD)**, за исключением тех, чей левый компонент — суперключ. Иначе говоря, никакие два набора атрибутов, описывающих «независимые» множества значений для одной и той же строки, не могут сосуществовать в одном и том же отношении, если только эти атрибуты не определяются полностью ключом таблицы.

| Таблица | PK | 1,2,3 нормальные формы | Функциональные зависимости от ключа | Проверка на многозначные зависимости |
| --- | --- | --- | --- | --- |
| animals | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: нет транзитивных | id → nickname, gender, arrival\_date, needs\_warm\_housing, animal\_type\_id, cage\_id | нет |
| animal\_types | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: нет транзитивных | id → type, diet\_type\_id | нет |
| employees | id | **1NF**: все атрибуты атомарны;  **2NF**: простой PK, нет части атрибутов, зависящих от части ключа;  **3NF**: нет зависимостей между неключевыми | id → full\_name, gender, hire\_date, birth\_date, position\_id, salary, contact\_info, special\_attributes | нет (все атрибуты функционально зависят от ключа) |
| positions | id | **1NF**: атомарные поля;  **2NF**: единственный PK, нет частичных зависимостей;  **3NF**: нет транзитивных зависимостей | id → title | нет(единственный зависимый атрибут) |
| cages | id | **NF**: атомарно;  **2NF**: простой PK;  **3NF**: capacity напрямую от id | id → animal\_type\_id, capacity | нет |
| birth\_records | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: нет транзитивных | id → child\_id, parent\_id\_1, parent\_id\_2, birth\_date, status | нет |
| medical\_records | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: нет транзитивных | id → animal\_id, birth\_date, weight, height, vaccinations, illnesses, checkup\_date | нет |
| rations | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: нет зависимостей между неключевыми | id → animal\_id, food\_id, amount, feeding\_time | нет |
| food | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: все атрибуты функционально от id | id → name, food\_type\_id, is\_produced\_internally | нет |
| food\_types | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: type напрямую от id | id → type | нет |
| diet\_types | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: type напрямую от id | id → type | нет |
| suppliers | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: contacts зависит только от id | id → name, contacts | нет |
| transfers | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: нет транзитивных | id → animal\_id, reason, destination\_zoo\_id, transfer\_date | нет |
| employees\_cages\_access | (employee\_id, cage\_id) | **NF**: атомарно;  **2NF**: каждый неключевой атрибут входит в PK;  **3NF**: нет транзитивных | PK → employee\_id, cage\_id | нет |
| employees\_animal\_types | (employee\_id, animal\_type\_id) | **1NF**: атомарно;  **2NF**: каждый неключевой атрибут входит в PK;  **3NF**: нет транзитивных | PK → employee\_id, animal\_type\_id | MVD тривиальны (левый+правый = все атрибуты) |
| suppliers\_food | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: все атрибуты напрямую от id | id → supplier\_id, food\_id, delivery\_date, quantity, price | MVD тривиальны (левый+правый = все атрибуты) |
| animal\_cage\_history | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: нет транзитивных | id → animal\_id, cage\_id, start\_date, end\_date | нет |
| zoos | id | **1NF**: атомарно;  **2NF**: простой PK;  **3NF**: address, contacts от id | id → name, address, contacts | нет |

## Переход от ER-диаграммы к реляционной

| **Таблица** | **Поля** | **PK** | **FK** | **Ограничения** |
| --- | --- | --- | --- | --- |
| animals |  | id |  |  |
| animal\_types |  | id |  |  |
| employees | id (serial),  full\_name (varchar(100)), gender (varchar) ,  hire\_date (date),  birth\_date (date), position\_id (int),  salary (numeric), contact\_info (text), special\_attributes (jsonb DEFAULT '{}') | id | position\_id → positions(id) ON DELETE RESTRICT | full\_name, position\_id NOT NULL;  gender CHECK in ('мужской','женский');  hire\_date, birth\_date ≤ current\_date;  salary ≥ 0 |
| positions | id (serial),  title (varchar(50)) | id | - | title NOT NULL, UNIQUE |
| cages | id (serial),  animal\_type\_id (int), capacity (int) | id | animal\_type\_id → animal\_types(id) ON UPDATE CASCADE ON DELETE RESTRICT | capacity > 0 |
| birth\_records | id (serial),  child\_id (int),  parent\_id\_1 (int), parent\_id\_2 (int), birth\_date (date),  status (varchar(50)) | id | child\_id → animals(id) ON DELETE CASCADE;  parent\_id\_1 → animals(id);  parent\_id\_2 → animals(id) | birth\_date ≤ current\_date;  status IN('оставлен','обменен','отдан') |
| medical\_records | id (serial),  animal\_id (int),  birth\_date (date),  weight (numeric),  height (numeric), vaccinations (text), illnesses (text), checkup\_date (date) | id | animal\_id → animals(id) ON DELETE CASCADE | birth\_date, checkup\_date ≤ current\_date;  weight, height > 0 |
| rations | id (serial),  animal\_id (int),  food\_id (int),  amount (numeric), feeding\_time (time) | id | animal\_id → animals(id) ON DELETE CASCADE;  food\_id → food(id) ON DELETE CASCADE | amount > 0;  feeding\_time NOT NULL |
| food | id (serial),  name (varchar(100)), food\_type\_id (int), is\_produced\_internally (boolean DEFAULT false) | id | food\_type\_id → food\_types(id) ON DELETE CASCADE | name NOT NULL, UNIQUE |
| food\_types | id (serial),  type (varchar(50)) | id | - | type NOT NULL, UNIQUE |
| diet\_types | id (serial),  type (varchar(50)) | id | - | type NOT NULL, UNIQUE;  CHECK in('хищник','травоядное','всеядное') |
| suppliers | id (serial),  name (varchar(100)),  contacts (text) | id | - | name, contacts NOT NULL;  name UNIQUE |
| transfers | id (serial),  animal\_id (int),  reason (text), destination\_zoo\_id (int), transfer\_date (date) | id | animal\_id → animals(id) ON DELETE CASCADE;  destination\_zoo\_id → zoos(id) ON DELETE SET NULL | transfer\_date ≤ current\_date |
| employees\_cages\_access | employee\_id (int),  cage\_id (int) | (employee\_id, cage\_id) | employee\_id → employees(id) ON DELETE CASCADE;  cage\_id → cages(id) ON DELETE CASCADE | - |
| employees\_animal\_types | employee\_id (int), animal\_type\_id (int) | (employee\_id, animal\_type\_id) | employee\_id → employees(id) ON DELETE CASCADE;  animal\_type\_id → animal\_types(id) ON DELETE CASCADE | - |
| suppliers\_food | id (serial),  supplier\_id (int),  food\_id (int),  delivery\_date (date), quantity (numeric),  price (numeric) | id | supplier\_id → suppliers(id) ON DELETE CASCADE;  food\_id → food(id) ON DELETE CASCADE | delivery\_date ≤ current\_date;  quantity > 0;  price ≥ 0 |
| animal\_cage\_history | id (serial),  animal\_id (int),  cage\_id (int),  start\_date (date),  end\_date (date) | id | animal\_id → animals(id) ON DELETE CASCADE;  cage\_id → cages(id) ON DELETE CASCADE | start\_date NOT NULL |
| zoos | id (serial),  name (varchar(100)),  address (text),  contacts (text) | id | - | name NOT NULL, UNIQUE |

## Перенос логического в физическое

1. Создание таблиц находится в

<https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/init/init_tables.sql>

1. Заполнение

<https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/init/data.sql>

1. Триггеры  
   <https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/init/triggers.sql>
2. Роли

<https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/init/roles.sql>

<https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/docker-compose.yml>

1. Сущности

<https://github.com/OmNomDomZ/ZooInfoSystem/tree/main/src/main/java/v/rabetsky/models/entities>

## Запросы

Готовые запросы из задания   
<https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/queries.sql>

Запросы для бэка находятся в классах DAO

<https://github.com/OmNomDomZ/ZooInfoSystem/tree/main/src/main/java/v/rabetsky/dao>

и также есть фильтры, для сортировки и распределения сущностей

<https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/filters.sql>

## Транзакции

**Код (Spring)**: @Transactional вокруг метода → Spring открывает/закрывает транзакцию, коммитит при успехе, откатывает при исключении.

**БД (Postgres/SQL)**: BEGIN…COMMIT/ROLLBACK → гарантирует атомарность SQL-операций и соблюдение изоляции.

[https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/init/transactions.sql](https://github.com/OmNomDomZ/ZooInfoSystem/blob/main/database/init/data.sql)

## Сессии

**Установка роли**

После логина в HttpSession сохраняется атрибут ROLE («ADMIN» или «READER»).

**Доступ в контроллерах**

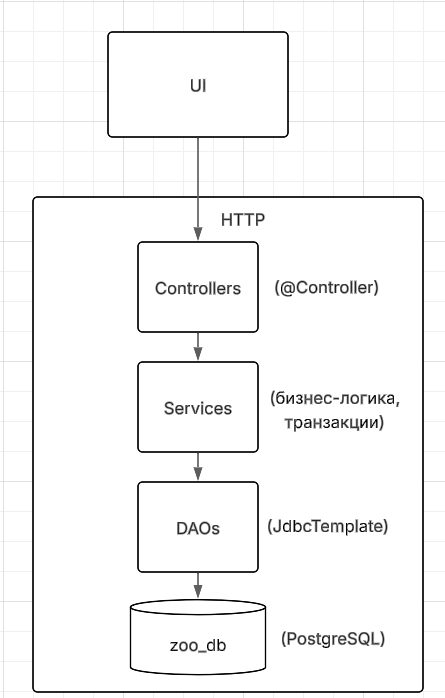
Интерсептор AdminInterceptor вытягивает текущую роль из RoleContext (который инициализируется из сессии) и блокирует недоступные методы.

**Жизненный цикл**

Роль хранится до конца HTTP-сессии; при logout сессия инвалидируется, а роль очищается.  
При повторном подключении в другой вкладки последняя нажатая роль перезапишет ROLE и будет действовать для всех вкладок (так как у обеих вкладок одна cookie)

У обеих вкладок одна cookie ⇒ один HttpSession. Последняя нажатая роль перезапишет ROLE и будет действовать для всех вкладок.

## Архитектура



## Реализация CRUD на бэке

CRUD реализован для сущностей:

1. animals
2. cages
3. employees
4. suppliers
5. medical\_records
6. suppliers
7. suppliers\_food

Также, для всех сущностей реализованы Create, Read, Delete