Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО».

Факультет программной инженерии и компьютерной техники

Базы данных

Лабораторная работа №3

Вариант 5432

Выполнил: Свечников Константин Денисович, P3130

Проверил: Перцев Т. С., преподаватель практики

**Оглавление**

[Задание 2](#_Toc197795879)

[Ход работы 3](#_Toc197795880)

[Даталогическая модель (исходная) 3](#_Toc197795881)

[Функциональные значимости 3](#_Toc197795882)

[Преобразование к 1НФ 4](#_Toc197795883)

[Преобразование к 2НФ 4](#_Toc197795884)

[Преобразование к 3НФ 4](#_Toc197795885)

[Преобразование к НФБК 5](#_Toc197795886)

[Полезная денормализация 5](#_Toc197795887)

[Даталогическая модель (после денормализации) 5](#_Toc197795888)

[Заключение 6](#_Toc197795889)

Задание

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

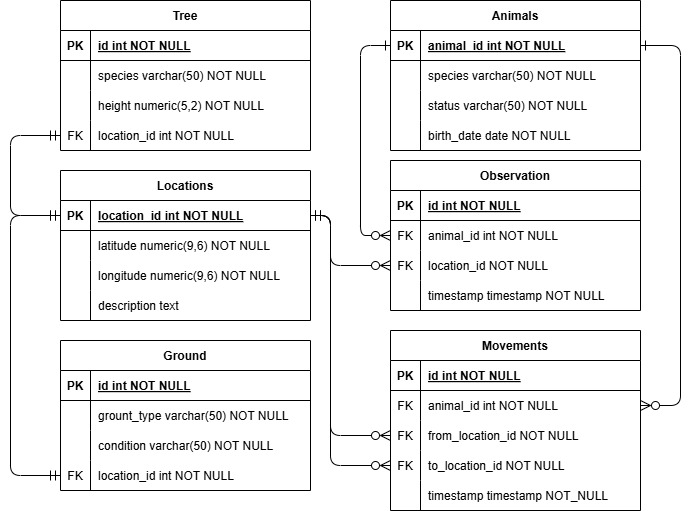
* Опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
* Приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум).
* Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основеNF;
* Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если ваша схема находится уже в BCNF, докажите это;
* Какие денормализации будут полезны для вашей схемы? Приведите подробное описание.

Придумайте триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

Ход работы

## 

## Даталогическая модель (исходная)



## Функциональные зависимости (исходные)

Tree:

* id → species
* id → height
* id → location\_id

Locations:

* location\_id → latitude
* location\_id → longitude
* location\_id → description

Ground:

* id → ground\_type
* id → condition
* id → location\_id

Animals:

* animal\_id → species
* animal\_id → status
* animal\_id → birth\_date

Observation:

* id → animal\_id
* id → location\_id
* id → timestamp
* timestamp → id

Movements:

* id → animal\_id
* id → from\_location\_id
* id → to\_location\_id
* id → timestamp
* timestamp → id

## Преобразование к 1НФ

Отношение, на пересечении каждой строки и столбца — одно значение выполняется, поэтому преобразований не требуется.

## Преобразование к 2НФ

Атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичного ключа отношения, поэтому преобразований не требуется.

## Преобразование к 3НФ

Все атрибуты, которые не входят в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа, поэтому преобразований не требуется.

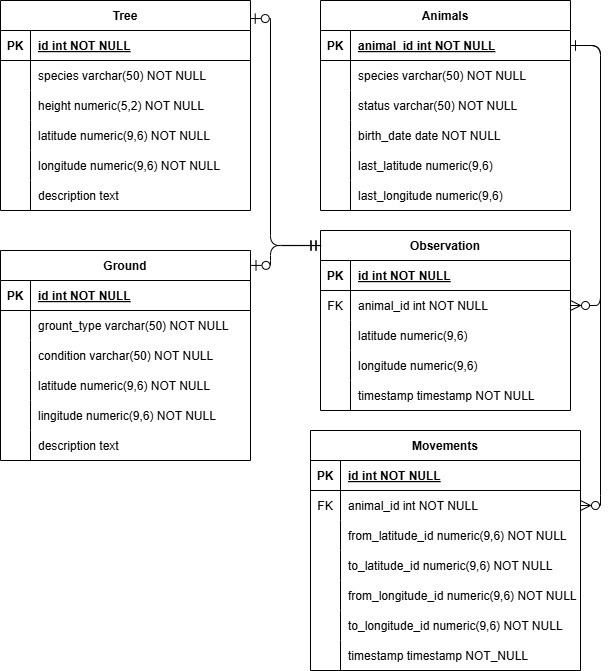
## Преобразование к НФБК

Не потребовалось, поскольку “ключевые” атрибуты в отношениях не зависят от “неключевых”. Например, по дате рождения (birth\_date) нельзя однозначно определить существо, по описанию локации ее координаты и т.д.

## Полезная денормализация

По сути, для увеличения скорости работы некоторых запросов можно избавиться от характеристической сущности Location, чтобы доступ к координатам получать напрямую из сущностей.

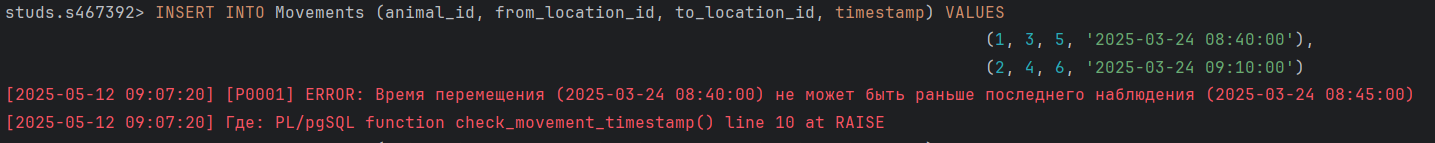
## Даталогическая модель (после денормализации)



## Функция и триггер

CREATE OR REPLACE FUNCTION *check\_movement\_timestamp*()  
 RETURNS TRIGGER AS $$  
DECLARE  
 last\_observation\_timestamp TIMESTAMP;  
BEGIN  
 SELECT *MAX*(timestamp) INTO last\_observation\_timestamp  
 FROM Observations  
 WHERE animal\_id = NEW.animal\_id;  
  
 IF last\_observation\_timestamp IS NOT NULL AND NEW.timestamp < last\_observation\_timestamp THEN  
 RAISE EXCEPTION 'Время перемещения (%) не может быть раньше последнего наблюдения (%)', NEW.timestamp, last\_observation\_timestamp;  
 END IF;  
  
 RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER validate\_movement\_time  
 BEFORE INSERT OR UPDATE ON Movements  
 FOR EACH ROW EXECUTE FUNCTION *check\_movement\_timestamp*();

Пример вывода триггера:



Заключение

В ходе выполнения данной лабораторной работы я узнал, что из себя представляет функциональная зависимость в базах данных, познакомился с сопутствующей терминологией. Также, я выяснил, как находить функциональные зависимости в отношениях и как приводить отношения к нормальным формам.