Министерство науки и высшего образования Российской Федерации  
федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Нейротехнологии и программирование*

*Дисциплина «Информатика»*

Лабораторная работа по

Базам данных №3

Работу выполнил:

Мухин Никита Денисович

Группа:

P3123

Преподаватель:

Горбунов Михаил Витальевич

Санкт-Петербург,

2024

**Оглавление**

[**Текст задания** 3](#_Toc162053458)

[**Даталогическая модель** 4](#_Toc162053459)

[**Функциональные зависимости** 5](#_Toc162053460)

[**Преобразование в 3NF** 5](#_Toc162053461)

[**Преобразование в BCNF** 6](#_Toc162053462)

[**Денормализация** 6](#_Toc162053463)

[**Триггер и функция на PL/pgSQL** 6](#_Toc162053465)

* [**Триггер на добавление элементов** 6](#_Toc162053466)
* [**Триггер на удаление элементов** 6](#_Toc162053467)

[**Вывод** 7](#_Toc162053468)

## **Текст задания**

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.

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Исходная, нормализованная и денормализованная модели.
3. Ответы на вопросы, представленные в задании.
4. Функция и триггер на языке PL/pgSQL
5. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. Нормализация. Формы
2. Функциональные зависимости. Виды
3. Денормализация
4. Язык PL/pgSQL

## **Даталогическая модель**

A computer screen shot of a computer

Description automatically generated

# **Функциональные зависимости**

Для **Динозавров**:

* ID -> Вид, Размер, Голос\_ID, Состояние\_ID
* Голос\_ID -> Звук
* Состояние\_ID -> Состояние

Для **Людей**:

* ID -> Имя, Возраст

Для **Действий**:

* ID -> Дата\_и\_время, Описание, Динозавр\_ID, Человек\_ID

Для **Звука:**

* ID -> ID, Вид, Размер, Голос\_ID, Состояние\_ID

# **Преобразование в 3NF**

Для преобразования в 3NF необходимо убедится, что каждое отношение:

* Удовлетворяет функциональным зависимостям.
* Не содержит транзитивных зависимостей.

**Динозавры**:

* ID (PK)
* Вид
* Размер
* Голос\_ID (FK)

**Голоса\_Динозавров**:

* Голос\_ID (PK)
* Звук

**Состояния\_Динозавров**:

* Состояние\_ID (PK)
* Состояние

**Люди**:

* ID (PK)
* Имя
* Возраст

**Действия**:

* ID (PK)
* Дата\_и\_время
* Описание
* Динозавр\_ID (FK)
* Человек\_ID (FK)

# **Преобразование в BCNF**

Отношения в 3NF также удовлетворяют BCNF, так как все функциональные зависимости определяются только ключами каждого отношения.

# **Денормализация**

Денормализация может быть полезна для оптимизации запросов и улучшения производительности. Например, можно добавить поля в отношение **Динозавры** для часто запрашиваемых атрибутов, таких как звук и состояние.

### 

### **Триггер и функция на PL/pgSQL**

## **Триггер на добавление элементов**

CREATE OR REPLACE FUNCTION log\_insert\_trigger()  
RETURNS TRIGGER AS $$  
BEGIN  
RAISE NOTICE 'Добавлен элемент в таблицу % с ID %', TG\_TABLE\_NAME, NEW.ID;  
RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;

CREATE TRIGGER after\_insert\_trigger  
AFTER INSERT ON Динозавры  
FOR EACH ROW EXECUTE FUNCTION log\_insert\_trigger();

**Триггер на удаление элементов**

CREATE OR REPLACE FUNCTION log\_delete\_trigger()  
RETURNS TRIGGER AS $$  
BEGIN  
RAISE NOTICE 'Удален элемент из таблицы % с ID %', TG\_TABLE\_NAME, OLD.ID;  
RETURN OLD;  
END;  
$$ LANGUAGE plpgsql;

CREATE TRIGGER after\_delete\_trigger  
AFTER DELETE ON Динозавры  
FOR EACH ROW EXECUTE FUNCTION log\_delete\_trigger();

## **Вывод**

При выполнении лабораторной работы я познакомился с понятием нормализации и денормализации. Научился определять функциональные зависимости модели, а также анализировать последнюю на соответствие различным нормальным формам. Изучил эффективные способы денормализации схемы базы данных и ситуации, в которых возможно их применение.