Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 «Программная инженерия» –

Системное и прикладное программное обеспечение

**Отчёт**

**По лабораторной работе №3**

по дисциплине «Базы данных»

**Вариант: NULL**

Выполнил:

Молчанов Фёдор Денисович

Группа: Р3113

Принял:

Николаев Владимир Вячеславович

г. Санкт-Петербург, 2024

Оглавление

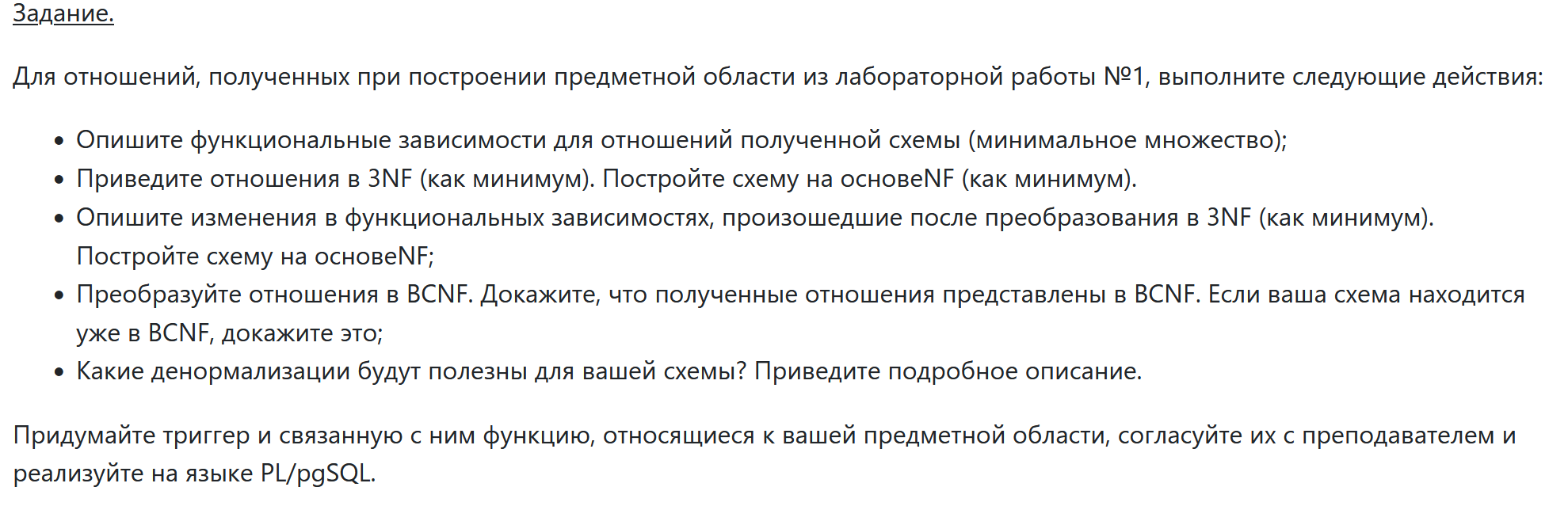
[Задание 3](#_Toc164952511)

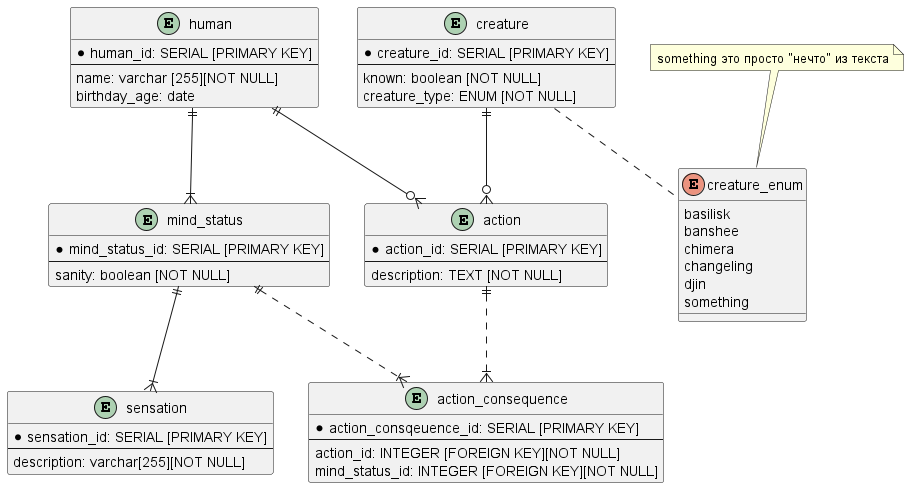
[Выполнение задания 3](#_Toc164952512)

[Triggers 5](#_Toc164952513)

[Вывод 5](#_Toc164952514)

# Задание





# Выполнение задания

Функциональные зависимости:

* human: human\_id -> (name, birthday\_age)
* Action: action\_id ->(description)
* Mind\_status: mind\_status\_id -> (sanity)
* Creature: creature\_id ->(known, creature\_type)
* Sensation: sensation\_id -> (description)
* Action\_consequence: (action\_id, mind\_status\_id) -> ()

1НФ: Отношение при котором на пересечении каждой строки и столбца — одно значение

2НФ: 1) отношение в 1НФ и 2) атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первчиного ключа отношения

3НФ — отношение в 1) 1НФ и 2НФ и 2) все атрибуты, которые не входят в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа.

Моя модель удовлетворяет 3НФ, т.к. все неключевые атрибуты зависят только от первичных ключей, и не содержат транзитивных зависимостей. А так как она удовлетворяет 3НФ, то удовлетворяет и 1НФ и 2НФ по определению.

НФБК — отношение в НФБК, когда для всех функциональных зависимостей отношения выполняется условие: детерминант — потенциальный ключ. Моя модель удовлетворяет НФБК, так как для всех функциональных зависимостей выполняется условие детерминант – потенциальный ключ.

Денормализация:

В моем случае чаще всего будут соединяться таблицы action и action\_consequence чтобы искать зависимости между действиями и их последствиями, поэтому в теории их можно соединить друг с другом.

# Triggers

--функция которая выкидывает Notice при удалении из таблицы Sensation

CREATE OR REPLACE FUNCITON Sensation\_delete() RETURNS TRIGGER AS $$

BEGIN

    RAISE NOTICE 'DELETE: Sensation named %', OLD.description;

    RETURN OLD;

END;

$$ LANGUAGE plpsql;

CREATE TRIGGER Sensation\_delete\_trigger

    BEFORE DELETE ON Sensation

    RETURN OLD;

END;

$$ LANGUAGE plpsql;

--функция которая выкидывает Notice при добавлении в таблцу Sensation

CREATE OR REPLACE FUNCITON Sensation\_insert() RETURNS TRIGGER AS $$

BEGIN

    RAISE NOTICE 'Insert: Sensation named %', NEW.description;

    RETURN NEW;

END;

$$ LANGUAGE plpsql;

CREATE TRIGGER Sensation\_insert\_trigger

    BEFORE DELETE ON Sensation

    RETURN OLD;

END;

$$ LANGUAGE plpsql;

# Вывод

В ходе выполнения лабораторной работы я узнал о понятиях нормализации и денормализации баз данных. Более подробно изучил реляционное представление, узнал эффективные способы нормализации.