МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

«Базы данных»

Вариант № 4582

***Выполнил:***Студент группы P3116  
 Григорьев Даниил Александрович

***Преподаватель:***Пименов Данила Дмитриевич

**Содержание**

[Задание 3](#_gjdgxs)

[Список сущностей и их классификация 3](#_69g569z0kesh)

[Инфологическая модель 4](#_alx52okj6kqx)

[Даталогическая модель 5](#_luozjm6qpa1)

[Реализация даталогической модели на языке SQL 6](#_uorw5v7o0k7q)

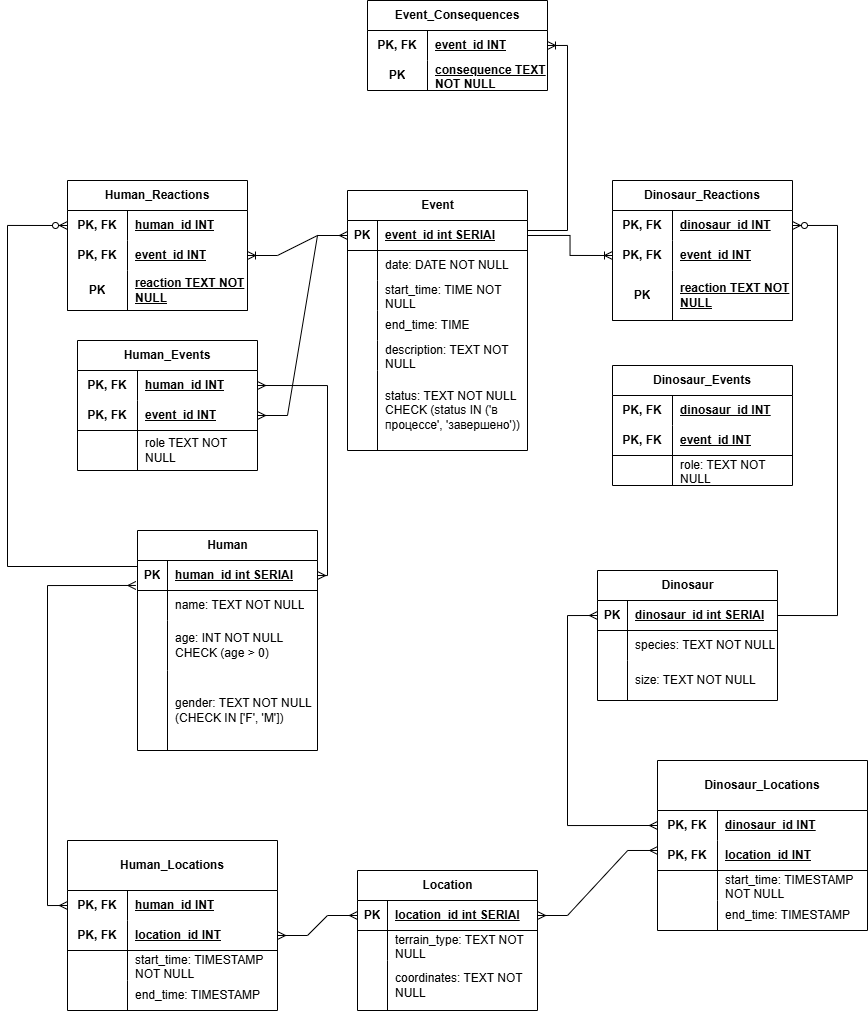
[Заключение](#_3znysh7) 8

# **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.



# **Функциональные зависимости**

Human: human\_id → name, age, gender

Dinosaur: dinosaur\_id → species, size

Event: event\_id → date, start\_time, end\_time, description, status

Location: location\_id → terrain\_type, coordinates

Human\_Reactions: (human\_id, event\_id, reaction) — ключ, нет неключевых атрибутов ⇒ нет нетривиальных ФЗ

Dinosaur\_Reactions: (dinosaur\_id, event\_id, reaction) - ключ, нет неключевых атрибутов ⇒ нет нетривиальных ФЗ

Event\_Consequences: (event\_id, consequence) — ключ, нет неключевых атрибутов ⇒ нет нетривиальных ФЗ

Human\_Events: (human\_id, event\_id) → role

Dinosaur\_Events: (dinosaur\_id, event\_id) → role

Human\_Locations: (human\_id, location\_id) → start\_time, end\_time

Dinosaur\_Locations: (dinosaur\_id, location\_id) → start\_time, end\_time

# 

**Нормальные формы**

* 1NF: Отношение находится в 1NF, если все его атрибуты содержат только атомарные значения, отсутствуют повторяющиеся группы и нет массивов. Мои отношения удовлетворяют 1NF, так как все атрибуты атомарны, нет повторяющихся групп и массивов
* 2NF: Отношение находится в 2NF, если оно находится в 1NF и все его неключевые атрибуты функционально зависят полностью от всего первичного ключа (если он составной). Моя модель удовлетворяет 2NF, так как все неключевые атрибуты зависят полностью функционально от всего первичного ключа, а не его части.
* 3NF: Отношение находится в 3NF, если оно находится в 2NF и все неключевые атрибуты зависят только от ключей — нет транзитивных зависимостей. Моя модель удовлетворяет 3NF, так как все ее неключевые атрибуты зависят только от ключей, нет транзитивных зависимостей.

BCNF

* Отношение находится в BCNF, если для каждой функциональной зависимости X → Y, X является суперключом. Моя модель удовлетворяет BCNF, так как для всех функциональных зависимостей X является суперключом.

**Денормализация**

*  **Объединение Human\_Events и Human\_Reactions**: в некоторых случаях если часто нужны и роль, и реакция человека в событии. Это сократит количество операций JOIN и уменьшит время обработки запросов. Это будет нарушать принцип 3NF: В объединённой таблице ключом станет (human\_id, event\_id, reaction), но role будет зависеть только от части ключа.

 **Денормализация Location и Human\_Locations**: если часто нужна информация о местности, где был человек вместе с временем пребывания человека. Это сократит количество JOIN, будет полезно для запросов по типу сколько времени люди проводили в определенном месте. Это будет нарушать принцип 3NF: данные о локации зависят только от location\_id, а не от составного ключа

**Триггер**

Если динозавр реагирует на событие “нападающий”, то реакции всех людей, участвующих в этом событии автоматически пометить, как “испуг”

CREATE OR REPLACE FUNCTION set\_human\_reactions\_scared()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.role = 'нападающий' THEN

        -- Удалим старые реакции для людей в этом событии

        DELETE FROM Human\_Reactions

        WHERE event\_id = NEW.event\_id

          AND human\_id IN (

              SELECT human\_id

              FROM Human\_Events

              WHERE event\_id = NEW.event\_id

          );

        -- Вставим новые реакции испуг

        INSERT INTO Human\_Reactions(human\_id, event\_id, reaction)

        SELECT human\_id, NEW.event\_id, 'испуг'

        FROM Human\_Events

        WHERE event\_id = NEW.event\_id;

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_set\_scared\_reaction

AFTER INSERT ON Dinosaur\_Events

FOR EACH ROW

EXECUTE FUNCTION set\_human\_reactions\_scared();