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 № 465635

***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**

1. На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
2. Составить инфологическую модель.
3. Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
4. Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
5. Заполнить созданные таблицы тестовыми данными.

# 

Внезапно вверху, на деревьях, раздался визг, и микроцератопсы в панике кинулись наутек, раскачивая ветки.

Огромная голова тиранозавра просунулась сквозь листву слева от путников.

Лекси в ужасе застонала.

Грант судорожно погреб к правому берегу, но река здесь была всего три метра шириной.

Тиранозавр запутался в густых зарослях.

Он ревел и вертел головой.

Затем голова скрылась.

Описание предметной области: Выживание людей в мире, где существуют динозавры

# **Список сущностей и их классификация**

1. Стержневые:

* Human: human\_id, name, age, gender
* Dinosaur: dinosaur\_id, species, size
* Events: event\_id, date, start\_time, end\_time, description
* Location: location\_id, terrain\_type, coordinates

1. Характеристические:

* Human\_Reactions: human\_id, event\_id, reaction\_id
* Dinosaur\_Reactions: dinosaur\_id, event\_id, reaction\_id
* Event\_consequenses: event\_id, consequence\_id

1. Ассоциативные:

* 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

Пояснение:

1. Стержневые:

* Creatures (существо (обобщение для людей и динозавров)): id, type, status
* Locations (места, где происходят события): id, terrain\_type, coordinates
* Event (события): id, date, start\_time, end\_time (время начала и конца события), description, status (к примеру: в процессе, завершено)

1. Характеристические:

* Human\_Attributes (атрибуты путников): creature\_id, name, age, gender
* Dinosaur\_Attributes(атрибуты динозавров): creature\_id, species, size
* Creature\_Reactions (реакции существа от события): id, creature\_id, event\_id, reaction (к примеру: страх, нападение, побег)
* Event\_Consequenses (последствия событий): id, event\_id, consequence (к примеру: человек ранен, динозавр убежал, ничего не произошло)

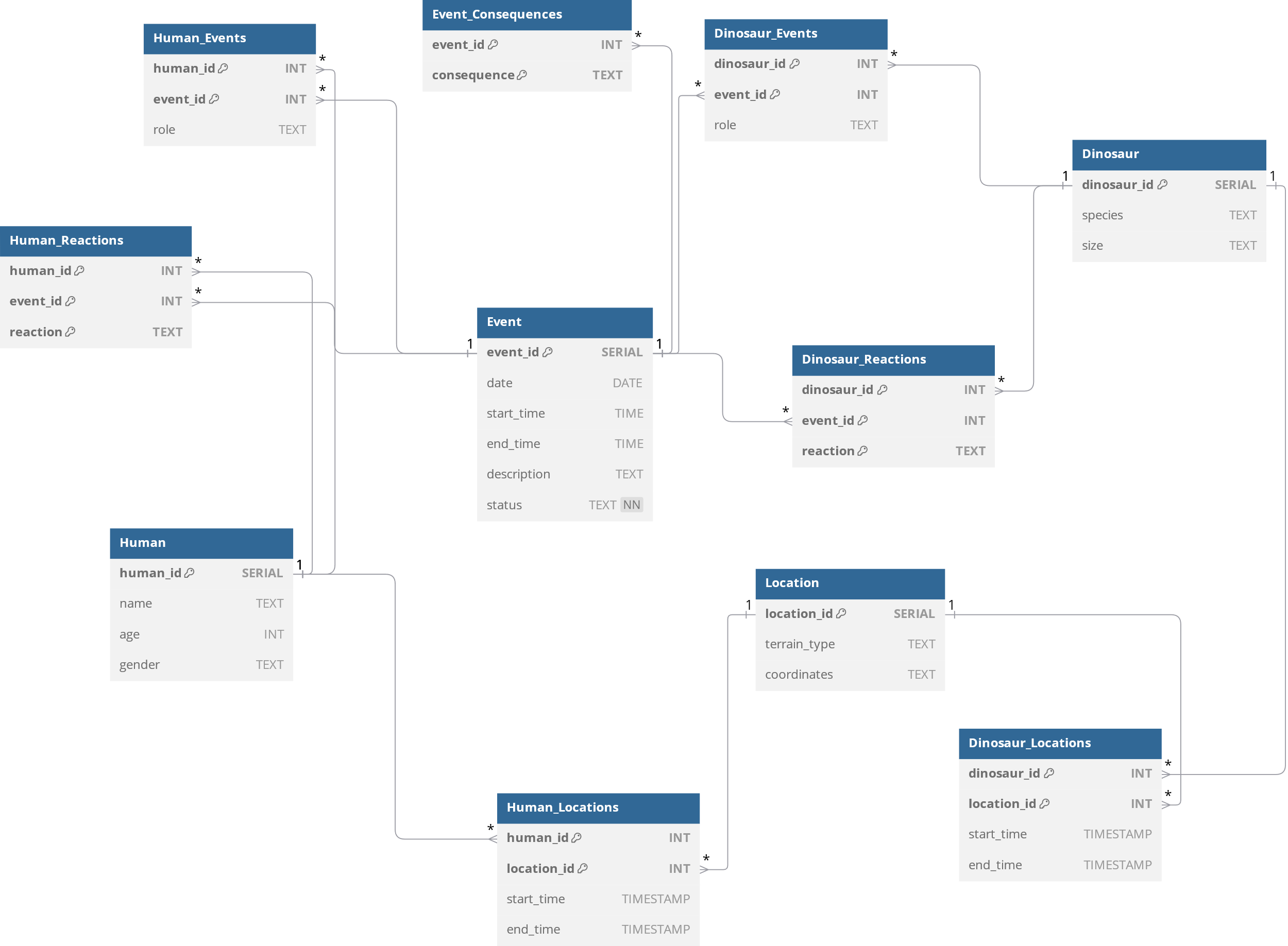
1. Ассоциативные:

* Creature\_Locations (нахождение определенного существа на определенной локации): id, start\_time (время входа в локацию), end\_time (время выхода (если есть)), creature\_id, location\_id
* Creature\_Events (присутствие определенного существа в определенном событии и какую роль играет): id, creature\_id, event\_id, role (к примеру: преследователь, жертва, наблюдатель)

# **Инфологическая модель**

# 

# **Даталогическая модель**



# **Реализация даталогической модели на языке SQL**

CREATE TABLE Human (

    human\_id SERIAL PRIMARY KEY,

    name TEXT,

    age INT,

    gender TEXT

);

CREATE TABLE Dinosaur (

    dinosaur\_id SERIAL PRIMARY KEY,

    species TEXT,

    size TEXT

);

CREATE TABLE Event (

    event\_id SERIAL PRIMARY KEY,

    date DATE,

    start\_time TIME,

    end\_time TIME,

    description TEXT,

    status TEXT CHECK (status IN ('в процессе', 'завершено')) NOT NULL

);

CREATE TABLE Location (

    location\_id SERIAL PRIMARY KEY,

    terrain\_type TEXT,

    coordinates TEXT

);

CREATE TABLE Human\_Reactions (

    human\_id INT,

    event\_id INT,

    reaction TEXT NOT NULL,

    PRIMARY KEY (human\_id, event\_id, reaction),

    FOREIGN KEY (human\_id) REFERENCES Human(human\_id),

    FOREIGN KEY (event\_id) REFERENCES Event(event\_id)

);

CREATE TABLE Dinosaur\_Reactions (

    dinosaur\_id INT,

    event\_id INT,

    reaction TEXT NOT NULL,

    PRIMARY KEY (dinosaur\_id, event\_id, reaction),

    FOREIGN KEY (dinosaur\_id) REFERENCES Dinosaur(dinosaur\_id),

    FOREIGN KEY (event\_id) REFERENCES Event(event\_id)

);

CREATE TABLE Event\_Consequences (

    event\_id INT,

    consequence TEXT NOT NULL,

    PRIMARY KEY (event\_id, consequence),

    FOREIGN KEY (event\_id) REFERENCES Event(event\_id)

);

CREATE TABLE Human\_Events (

    human\_id INT,

    event\_id INT,

    role TEXT,

    PRIMARY KEY (human\_id, event\_id),

    FOREIGN KEY (human\_id) REFERENCES Human(human\_id) ON DELETE CASCADE,

    FOREIGN KEY (event\_id) REFERENCES Event(event\_id) ON DELETE CASCADE

);

CREATE TABLE Dinosaur\_Events (

    dinosaur\_id INT,

    event\_id INT,

    role TEXT,

    PRIMARY KEY (dinosaur\_id, event\_id),

    FOREIGN KEY (dinosaur\_id) REFERENCES Dinosaur(dinosaur\_id) ON DELETE CASCADE,

    FOREIGN KEY (event\_id) REFERENCES Event(event\_id) ON DELETE CASCADE

);

CREATE TABLE Human\_Locations (

    human\_id INT,

    location\_id INT,

    start\_time TIMESTAMP,

    end\_time TIMESTAMP,

    PRIMARY KEY (human\_id, location\_id),

    FOREIGN KEY (human\_id) REFERENCES Human(human\_id) ON DELETE CASCADE,

    FOREIGN KEY (location\_id) REFERENCES Location(location\_id) ON DELETE CASCADE

);

CREATE TABLE Dinosaur\_Locations (

    dinosaur\_id INT,

    location\_id INT,

    start\_time TIMESTAMP,

    end\_time TIMESTAMP,

    PRIMARY KEY (dinosaur\_id, location\_id),

    FOREIGN KEY (dinosaur\_id) REFERENCES Dinosaur(dinosaur\_id) ON DELETE CASCADE,

    FOREIGN KEY (location\_id) REFERENCES Location(location\_id) ON DELETE CASCADE

);

# **Заключение**

В ходе лабораторной работы я познакомился с архитектурой “ANSI-SPARK”, научился составлять инфологическую и даталогическую модель сущностей, по которым реализовал базу данных при помощи SQL.