НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет Программной инженерии и компьютерной техники

Дисциплина Базы данных

Лабораторная работа № 1

Выполнил студент:

Лабор Тимофей Владимирович

Группа № P3125

Преподаватель:

Бострикова Дарья Константиновна

г. Санкт-Петербург

2024

Оглавление

[**Вариант:** 3](#_Toc162000964)

[**Задание:** 3](#_Toc162000965)

[**Отчет:** 3](#_Toc162000966)

[**Пояснение к предметной области:** 3](#_Toc162000967)

[**Список сущностей:** 4](#_Toc162000968)

[**Инфологическая модель:** 4](#_Toc162000969)

[**Даталогическеая модель:** 5](#_Toc162000970)

[**Реализация на SQL:** 5](#_Toc162000971)

[**Вывод:** 9](#_Toc162000972)

## **Вариант:**

2508

## **Задание:**

**Описание предметной области:**

Кромка плато оказалась едва ли не у самых их ног. Хилвар первым дошагал до нее, а спустя несколько секунд и Олвин, лишившись дара речи, уже стоял рядом. Он был ошеломлен, потому что оба они находились на краю вовсе не какого-то там плато, как им представлялось поначалу, но огромной чаши глубиной в полмили и диаметром мили в три. Поверхность впереди резко понижалась, плавно выравнивалась на дне этой огромной круглой долины и снова поднималась - все более и более круто - к противоположному краю. Самая низкая часть чаши была занята круглым озером, зеркало которого непрерывно трепетало, словно бы терзаемое непрекращающимся ветром.

**Для выполнения лабораторной работы №1 необходимо:**

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.

## **Отчет:**

### **Пояснение к предметной области:**

Описываются путешествующие люди, они находятся в какой-то локации и носят определенную одежду. Локация характеризуется названием, типом местности, почвы и погодой. У почвы есть поверхность, состав и высота. Одежда — это конкретный “костюм”, существующий в единственном экземпляре, он может быть сделан для некоторой цели, окрашен в некий цвет и находиться в каком-то состоянии. Также есть события, в которых может участвовать множество людей.

### **Список сущностей:**

Стрежневые:

* Человек
* Событие
* Локация

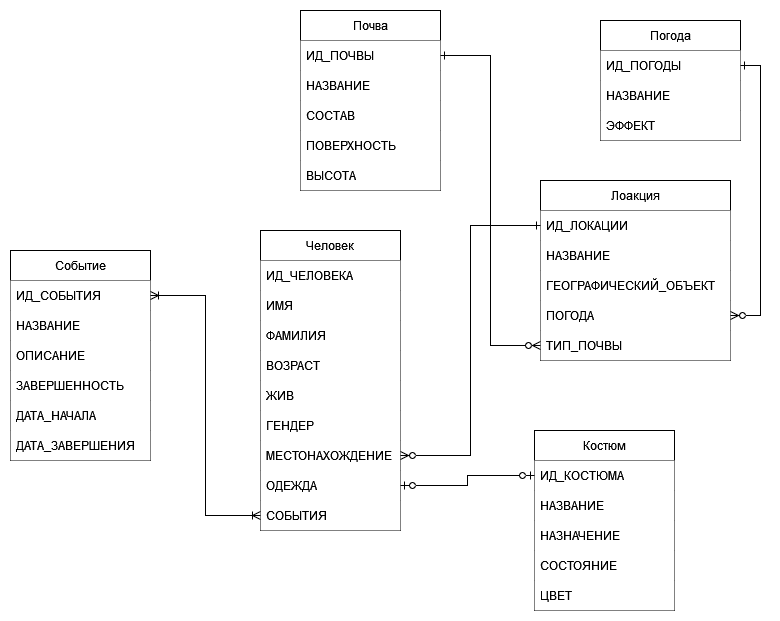
Ассоциации:

* Человек к событию

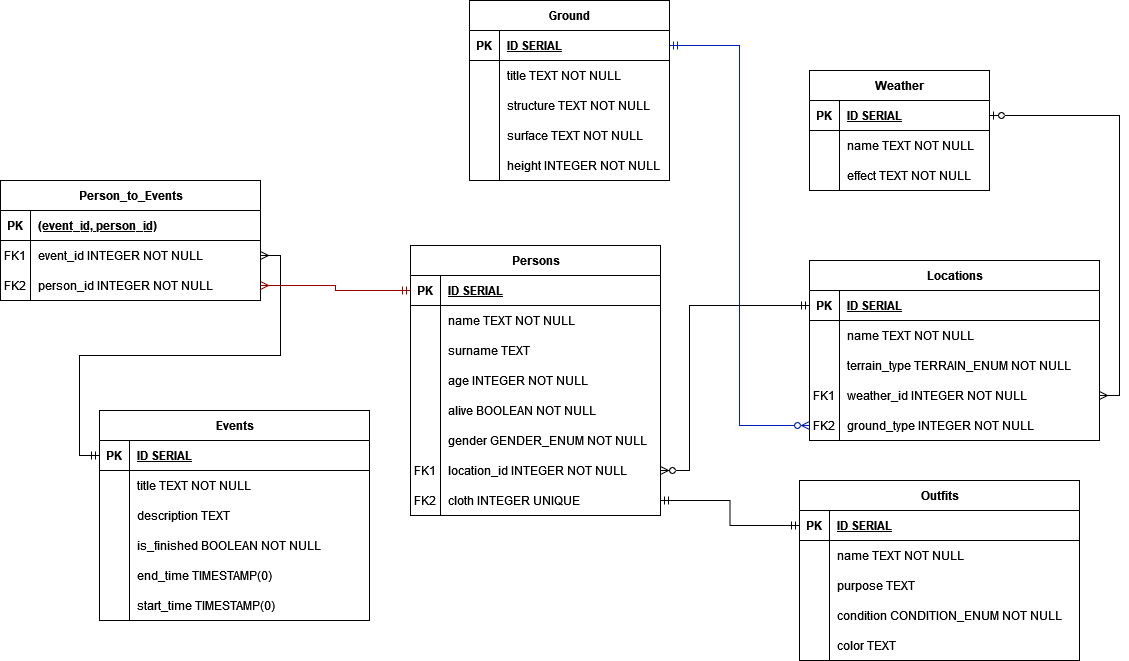
Характеристики:

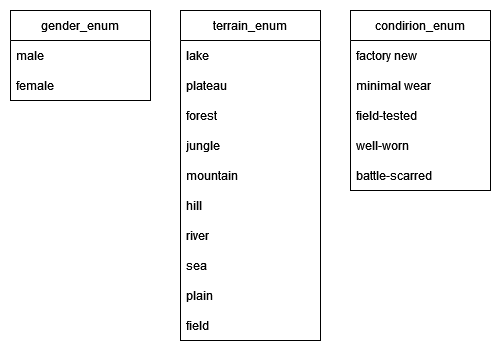
* Одежда
* Погода
* Почва

### **Инфологическая модель:**



### **Даталогическеая модель:**

****

****

### **Реализация на SQL:**

-- Создание таблиц

CREATE TYPE gender\_enum AS ENUM ('male', 'female');

CREATE TYPE condition\_enum AS ENUM ('factory new', 'minimal wear', 'field-tested', 'well-worn', 'battle-scarred');

CREATE TYPE terrain\_enum AS ENUM ('lake', 'plateau', 'mountain', 'hill', 'forest', 'jungle', 'river', 'sea', 'field', 'plain');

CREATE TABLE IF NOT EXISTS grounds (

id SERIAL PRIMARY KEY,

title TEXT NOT NULL,

structure TEXT NOT NULL,

surface TEXT NOT NULL,

height INTEGER NOT NULL

);

CREATE TABLE IF NOT EXISTS weather (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

effect TEXT NOT NULL

);

CREATE TABLE IF NOT EXISTS outfits (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

purpose TEXT,

condition condition\_enum NOT NULL,

color TEXT

);

CREATE TABLE IF NOT EXISTS locations (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

terrain\_type terrain\_enum NOT NULL,

weather\_id INTEGER REFERENCES weather(id) NOT NULL,

ground\_type INTEGER REFERENCES grounds(id) NOT NULL

);

CREATE TABLE IF NOT EXISTS persons (

id SERIAL PRIMARY KEY,

name TEXT NOT NULL,

surname TEXT,

age INTEGER NOT NULL,

alive BOOLEAN NOT NULL,

gender gender\_enum NOT NULL,

location\_id INTEGER REFERENCES locations(id) NOT NULL,

cloth INTEGER REFERENCES outfits(id) NOT NULL,

CHECK (age > 0)

);

CREATE TABLE IF NOT EXISTS events (

id SERIAL PRIMARY KEY,

title TEXT NOT NULL,

description TEXT,

is\_finished BOOLEAN NOT NULL,

end\_time TIMESTAMP(0),

start\_time TIMESTAMP(0)

);

CREATE TABLE IF NOT EXISTS person\_to\_events (

event\_id INTEGER REFERENCES events(id) NOT NULL,

person\_id INTEGER REFERENCES persons(id) NOT NULL,

PRIMARY KEY (event\_id, person\_id)

);

--Вставка данных

INSERT INTO weather (name, effect)

VALUES

('дождь', 'влажность'),

('жара', 'сухость'),

('вьюга', 'плохая видимость'),

('туман', 'плохая видимость');

INSERT INTO outfits (name, purpose, condition, color)

VALUES

('комбенезон', 'стройка', 'field-tested', 'синий'),

('деловой костюм', 'офис', 'factory new', 'черный'),

('европейское женское платье', 'бал', 'factory new', 'красный'),

('камуфляж', 'охота', 'battle-scarred', NULL);

INSERT INTO grounds (title, structure, surface, height)

VALUES

('скалистая', 'гранит', 'гладкая', 1000),

('плодородная', 'чернозем', 'рыхлая', 250),

('каменистая', 'галька', 'зыбкая', 300),

('лесная', 'корни', 'неровная', 400),

('дно озера', 'ил', 'зыбкая', 500);

INSERT INTO locations (name, terrain\_type, weather\_id, ground\_type)

VALUES

('Подход к котловану', 'plateau', 2, 1),

('Дно огромной чаши', 'lake', 2, 4),

('Лес рядом с котлованом', 'forest', 1, 3),

('Замок Нойшванштайн', 'mountain', 4, 1);

INSERT INTO persons (name, surname, age, alive, gender, location\_id, cloth)

VALUES

('Хилвар', 'Сигурдсон', 30, true, 'male', 1, 1),

('Олвин', 'Олафсон', 31, true, 'male', 1, 4),

('Алиса', NULL, 20, true, 'female', 4, 3);

INSERT INTO events (title, description, is\_finished, end\_time, start\_time)

VALUES

('Открытие озера в котловане', 'Хилвар и Олвин нашли странную чашу с озером на дне', true, '1960-02-13 12:23:12', '1960-01-13 12:00:42'),

('Прыжок в кроличью нору', 'Девочка последоваала за кроликом и оказалось в другом мире.', true, '1890-06-13 13:43:13', '1890-06-13 13:10:02'),

('Исследование странной чаши', 'За открытием чаши следует ее исследование', false, NULL, '1960-02-13 12:23:12');

INSERT INTO person\_to\_events (event\_id, person\_id)

VALUES

(1, 1),

(1, 2),

(3, 1),

(3, 2),

(2, 3);

## **Вывод:**

В ходе выполнения лабораторной работы я узнал об архитектуре современных баз данных, о том, что такое сущности, связи и какие они бывают. Научился проектировать собственные базы данных, создавать инфологические и даталогические модели, а также строить по ним ER-диаграммы. Узнал основы синтаксиса языка PostgreSQL.