**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

**Отчет**

по лабораторной работе **№1**

по дисциплине «**Базы Данных**»

Автор:Ведерников Антон Владимирович

Факультет: ПИиКТ

Группа:P3123

Преподаватель: Королёва Ю.А



Санкт-Петербург, 2023

[Задание 3](#_Toc1)

[Описание предметной области 4](#_Toc2)

[Список сущностей 4](#_Toc3)

[Даталогическая модель 5](#_Toc4)

[Запросы 6](#_Toc5)

[SQL Реализация 7](#_Toc6)

[Вывод 8](#_Toc7)

# 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.

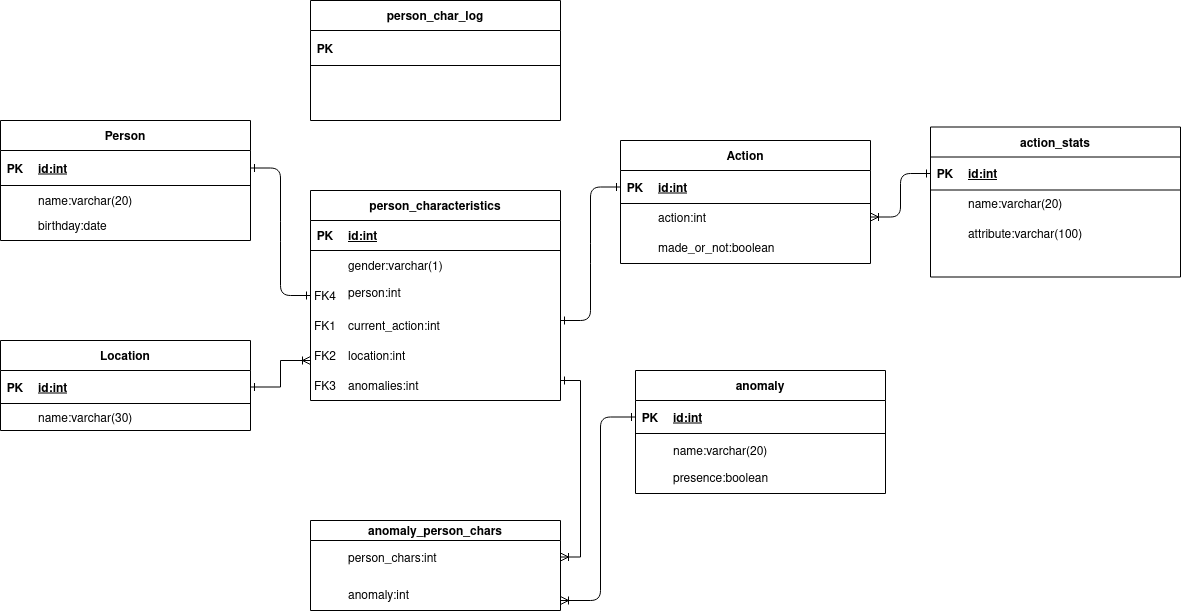
# Описание предметной области

*Он не обнаружил других следов разума на берегах лавового потока. Один раз наткнулся, однако, на жуткое подобие человека, плывущего кролем, - но у того не было ни глаз, ни ноздрей, лишь огромный беззубый рот, жадно поглощавший питание из воды, которая его окружала.*

# Список сущностей

1. Person - характеристика
2. Part\_of\_body - стержневая
3. Gender - стержневая
4. Action - стержневая
5. Location - стержневая

# Даталогическая модель



# Запросы

1. Имя человека + его аномалии

SELECT p.name, a.name

FROM person AS p

JOIN person\_characteristics AS pc ON p.id = pc.person

JOIN anomaly\_person\_chars AS apc ON pc.id = apc.person

JOIN anomaly AS a ON apc.anomaly = a.id;

2. Действие(название) и место где оно произошло

SELECT ast.name AS action\_name, loc.name AS location\_name

FROM action AS a

JOIN action\_stats AS ast ON a.action = ast.id

JOIN person\_characteristics AS pc ON a.id = pc.current\_action

JOIN location AS loc ON pc.location = loc.id;

3.имя человека и кол-во его аномалий

SELECT p.name, COUNT(apc.anomaly) AS anomaly\_count

FROM person AS p

LEFT JOIN person\_characteristics AS pc ON p.id = pc.person

LEFT JOIN anomaly\_person\_chars AS apc ON pc.id = apc.person

GROUP BY p.name;

4. Имена людей с кол-вом аномалий >2

SELECT p.name, COUNT(apc.anomaly) AS anomaly\_count

FROM person AS p

LEFT JOIN person\_characteristics AS pc ON p.id = pc.person

LEFT JOIN anomaly\_person\_chars AS apc ON pc.id = apc.person

GROUP BY p.name

HAVING COUNT(apc.anomaly) > 2;

5. 3. + если человек старше 20

SELECT p.name, COUNT(apc.anomaly) AS anomaly\_count

FROM person AS p

LEFT JOIN person\_characteristics AS pc ON p.id = pc.person

LEFT JOIN anomaly\_person\_chars AS apc ON pc.id = apc.person

WHERE p.birthday < current\_date - interval '20 years'

GROUP BY p.name;

6. 1. В колонку

SELECT p.name || E'\n' || string\_agg(a.name, E'\n') AS person\_and\_anomalies

FROM person AS p

LEFT JOIN person\_characteristics AS pc ON p.id = pc.person

LEFT JOIN anomaly\_person\_chars AS apc ON pc.id = apc.person

LEFT JOIN anomaly AS a ON apc.anomaly = a.id

GROUP BY p.name;

# SQL Реализация

create table if not exists location

(

id int primary key,

name varchar(30)

);

create table if not exists person

(

id int primary key,

name varchar(20),

birthday date

);

create table if not exists anomaly

(

id int primary key,

name varchar(20),

presence boolean

);

create table if not exists action\_stats

(

id int primary key,

name varchar(20),

attribute varchar(100)

);

create table if not exists action

(

id int primary key,

action int references action\_stats(id),

made\_or\_not boolean

);

create table if not exists person\_characteristics

(

id int primary key,

person int references person(id),

gender varchar(1) check (gender = 'M' or gender = 'F'),

current\_action int references action(id),

location int references location(id),

anomalies int references anomaly(id)

);

create table if not exists anomaly\_person\_chars

(

person int references person\_characteristics(id) on update cascade,

anomaly int references anomaly(id) on update cascade

);

# Вывод

В ходе данной лабораторной работы я изучил базовые аспекты создания

инфологической и даталогической моделей, базовый DML и DDL в

POSTGRESQL и ER-модели.