Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

ПИИКТ, кафедра Вычислительной техники

Лабораторная работа №1

по дисциплине

«Информационные системы и базы данных»

Студент: Абульфатов Р.М. P33312

Преподаватель: Наумова. Н. А.

Санкт-Петербург

2023 г

**Текст варианта:**

У него не осталось сознательного воспоминания об увиденном, но этой ночью, когда он, понуро сгорбившись, сидел у входа в пещеру и чутким ухом ловил шумы окружающего мира, он впервые ощутил пока еще слабую щемящую боль от нового властного чувства. То была смутная неопределенная зависть, какая-то неудовлетворенность жизнью. Он понятия не имел, откуда взялось это чувство, а тем более - как его утолить, но недовольство закралось в его душу, и это было уже первым малым шагом к очеловечиванию.

**Описание предметной области:**

По приведенному тексту и остальному произведению, откуда взят данный текст, можно

составить следующую предметную область:

**Вожак** (у которого есть своя **стая**, отношение у стаи с вожаком 1:1), одноухий питекантроп, более тощий чем Смотрящий на Луну. Одноухий вожак первой стаи. **Соплеменник**, *имя* (атрибут) которого Смотрящий на Луну - вожак второй стаи. **Соплеменники** - члены **стаи** (M:1). Также **стаи** могут **передвигаться** из одного места в другое. **Стая** и **передвижение** - (M:1). Стая, под руководством Смотрящего на Луну, направилось по дороге на пастбище. Затем остановились внизу у ручья. Далее отправились в пещеру. Затем к этому вожаку пришли в голову умные мысли, и он понял, что заостренный камень можно использовать для охоты. И он удачно поохотился, однако ни он, ни сородичи сразу не поняли, что не нужно убивать добычу "до кровавого месива". Это осознание к вождю и его сородичам пришло со временем. Бородавочники – **вид** одного из **животных** (свиней). Отношение животного к виду (M:1). Смотрящий вперед, как вожак стаи, для того чтобы накормить стаю первым нанес удар тяжелым заостренным камнем длиной в ладонь молодому поросенку. Голодная стая столпилась вокруг вожака и добычи, восхищаясь поступком. Из данного фрагмента можно сделать вывод, что у нас есть сущность **оружие**. Оружие есть, как у **вожака** стаи, так и **соплеменника**. У **вожака**/**соплеменника** с **оружием** отношение (M:1). Также у **вожака** есть определенные полномочия (**Leader\_Action**), тут будет отношение (M:1). Смотрящий вперед, как вожак стаи, для того, чтобы накормить стаю первым нанес удар тяжелым заостренным камнем длиной в ладонь молодому поросенку. Голодная стая столпилась вокруг вожака и добычи, восхищаясь поступком. Как видно из данного предложения можно выделить сущность **действия стаи** и **стая** (M:1). Вожак руководит стаей, он ответственный за добычу, которой нужно накормить стаю, а также за безопасность стаи. Также у **стаи** есть **характеристика**, например, что она голодная (M:M). Первая стая - вожак Одноухий. Вторая стая - вожак Смотрящий на Луну. Соплеменники - действуют по указке вожака. Почитают и уважают вожака. Полностью доверяют свои жизни.

**Сущности:**

Flock (стержневая сущность) // стая

Leader (стержневая сущность) // вожак

Animal (стержневая сущность) // животное

Tribesman (ассоциативная сущность) // соплеменник

Weapon (ассоциативная сущность) // оружие

Movement\_Flock (ассоциативная сущность) // передвижение стаи

Leader\_Action (характеристическая) // действия лидера

Species (характеристическая) // вид животного

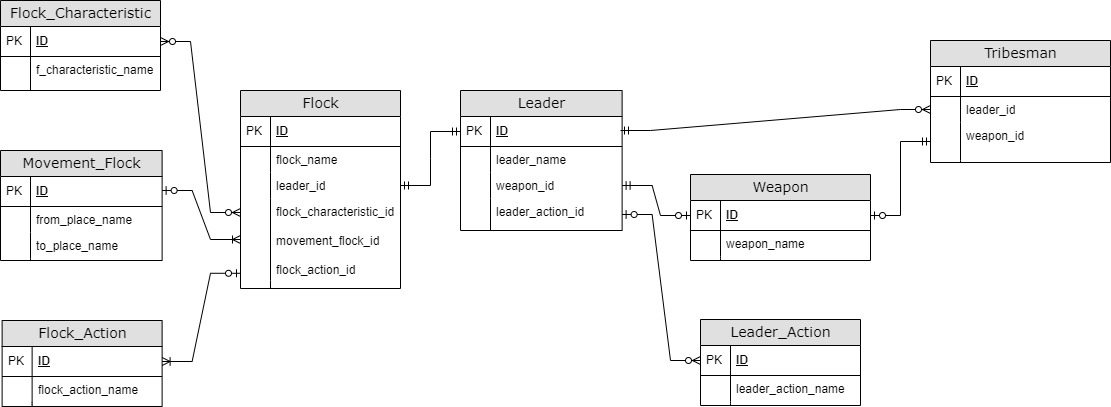
Flock\_Characteristic (характеристическая) // характеристика стаи

Flock\_Action (характеристическая) // действия стаи

**Атрибуты к сущностям:**

* Flock
  + - flock\_id – уникальный идентификатор стаи;
    - flock\_name - название стаи;
    - leader\_id – айди вожака, которому принадлежит данная стая;
    - flock\_characteristic\_id – айди характеристики, определенной стаи;
    - movement\_flock\_id – айди передвижения, перечисление как двигалась стая.
    - flock\_action\_id – айди действия стаи, которая она совершает.
* Leader
  + - leader\_id – уникальный идентификатор вожака стаи;
    - leader\_name – имя вожака;
    - weapon\_id – айди оружия, которое может присутствовать у вожака.
    - leader\_action\_id – айди действия, которое совершает вожак.
* Tribesman
  + - tribesman\_id – уникальный идентификатор сородича стаи.
    - weapon\_id – айди оружия, который может быть у соплеменника.
    - leader\_id – айди вожака, который управляет данным соплеменником.
* Species
  + - species\_id – уникальный идентификатор вида животного,
    - species\_name – название вида,
* Animal
  + - animal\_id – уникальный идентификатор животного,
    - species\_id – айди вида, которому принадлежит данное животное,
    - animal\_name – название животного,
* Weapon
  + - weapon\_id – уникальный идентификатор оружия,
    - weapon\_name – название оружия,
* Leader\_Action
* leader\_action\_id – уникальный идентификатор действия лидера,
* leader\_action\_name – название действия,
* Movement\_Flock
  + - movement\_flock\_id – уникальный идентификатор движения стаи,
    - from\_place\_name – название начальной локации,
    - to\_place\_name – название конечной локации,
* Flock\_Characteristic
  + - * flock\_characteristic\_id – уникальный идентификатор характеристики стаи,
      * flock\_characteristic\_name – название характеристики.
* Flock\_Action
  + - * flock\_action\_id – уникальный идентификатор действия стаи,
      * flock\_action\_name – названия действия стаи,

**ER диаграмма:**

****

**Даталогическая модель:**

Изображение выглядит как текст, диаграмма, линия, Шрифт

Автоматически созданное описание

**Реализация даталогической модели в PostgreSQL:**

1. **creation.sql**

CREATE TABLE IF NOT EXISTS Flock\_Characteristic (

f\_characteristic\_id SERIAL PRIMARY KEY NOT NULL,

flock\_characteristic\_name VARCHAR(255) NOT NULL check ( length(flock\_characteristic\_name) > 0 )

);

CREATE TABLE IF NOT EXISTS Movement\_Flock (

movement\_flock\_id SERIAL PRIMARY KEY NOT NULL,

from\_place\_name VARCHAR(100) NOT NULL check ( length(from\_place\_name) > 0 ),

to\_place\_name VARCHAR(100) NOT NULL check ( length(to\_place\_name) > 0 )

);

CREATE TABLE IF NOT EXISTS Flock\_Action (

flock\_action\_id SERIAL PRIMARY KEY NOT NULL,

flock\_action\_name VARCHAR(255) NOT NULL check ( length(flock\_action\_name) > 0 )

);

CREATE TABLE IF NOT EXISTS Weapon (

weapon\_id SERIAL PRIMARY KEY NOT NULL,

weapon\_name VARCHAR(100) NOT NULL

);

CREATE TABLE IF NOT EXISTS Leader\_Action (

leader\_action\_id SERIAL PRIMARY KEY NOT NULL,

leader\_action\_name VARCHAR(255) NOT NULL

);

CREATE TABLE IF NOT EXISTS Leader (

leader\_id SERIAL PRIMARY KEY NOT NULL,

leader\_name VARCHAR(100) NOT NULL check ( length(leader\_name) > 0 ),

leader\_action\_id INT REFERENCES leader\_action (leader\_action\_id),

weapon\_id INT REFERENCES weapon (weapon\_id)

);

CREATE TABLE IF NOT EXISTS Tribesman (

tribesman\_id SERIAL PRIMARY KEY,

tribesman\_name VARCHAR(100),

weapon\_id INT REFERENCES weapon (weapon\_id),

leader\_id INT REFERENCES leader (leader\_id) NOT NULL

);

CREATE TABLE IF NOT EXISTS Flock (

flock\_id SERIAL PRIMARY KEY NOT NULL,

flock\_name VARCHAR(100) NOT NULL check ( length(flock\_name) > 0 ),

leader\_id INT REFERENCES leader (leader\_id) NOT NULL UNIQUE

);

CREATE TABLE IF NOT EXISTS Flock\_to\_Characteristic (

flock\_id INT REFERENCES flock(flock\_id) NOT NULL,

f\_characteristic\_id INT REFERENCES flock\_characteristic (f\_characteristic\_id) NOT NULL

);

1. **insertion.sql**

INSERT INTO Flock\_Characteristic(flock\_characteristic\_name)

VALUES ('голодная');

INSERT INTO Flock\_Characteristic(flock\_characteristic\_name)

VALUES ('удивленная');

INSERT INTO Flock\_Characteristic(flock\_characteristic\_name)

VALUES ('агрессивная');

INSERT INTO Flock\_Characteristic(flock\_characteristic\_name)

VALUES ('усталая');

INSERT INTO Flock\_Characteristic(flock\_characteristic\_name)

VALUES ('радостная');

INSERT INTO Movement\_Flock(from\_place\_name, to\_place\_name)

VALUES ('озеро', 'тропинка');

INSERT INTO Movement\_Flock(from\_place\_name, to\_place\_name)

VALUES ('тропинка', 'поле');

INSERT INTO Movement\_Flock(from\_place\_name, to\_place\_name)

VALUES ('поле', 'вход в пещеру');

INSERT INTO Movement\_Flock(from\_place\_name, to\_place\_name)

VALUES ('вход в пещеру', 'выход в пещеру');

INSERT INTO Movement\_Flock(from\_place\_name, to\_place\_name)

VALUES ('выход в пещеру', 'лес');

INSERT INTO Flock\_Action(flock\_action\_name)

VALUES ('удивляется');

INSERT INTO Flock\_Action(flock\_action\_name)

VALUES ('поедает');

INSERT INTO Flock\_Action(flock\_action\_name)

VALUES ('радуется');

INSERT INTO Flock\_Action(flock\_action\_name)

VALUES ('спит');

INSERT INTO Flock\_Action(flock\_action\_name)

VALUES ('атакует');

INSERT INTO Weapon(weapon\_name)

VALUES ('палка');

INSERT INTO Weapon(weapon\_name)

VALUES ('камень');

INSERT INTO Weapon(weapon\_name)

VALUES ('копье');

INSERT INTO Weapon(weapon\_name)

VALUES ('факел');

INSERT INTO Leader\_Action(leader\_action\_name)

VALUES ('направляет стаю');

INSERT INTO Leader\_Action(leader\_action\_name)

VALUES ('атакует кабана');

INSERT INTO Leader\_Action(leader\_action\_name)

VALUES ('следит за безопасностью стаи');

INSERT INTO Leader(leader\_name, leader\_action\_id, weapon\_id)

VALUES ('Лопоухий', 1, NULL);

INSERT INTO Leader(leader\_name, leader\_action\_id, weapon\_id)

VALUES ('Длинноухий', 2, 2);

INSERT INTO Leader(leader\_name, leader\_action\_id, weapon\_id)

VALUES ('Наблюдающий', 3, NULL);

INSERT INTO Leader(leader\_name, leader\_action\_id, weapon\_id)

VALUES ('Стражник', 3, 4);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES ('первый сородич', 3, 2);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES ('второй сородич', 1, 1);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES ('третий сородич', 3, 2);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 2);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 3);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 1);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 2);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 2);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 4);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 4);

INSERT INTO Tribesman(tribesman\_name, weapon\_id, leader\_id)

VALUES (NULL, NULL, 4);

INSERT INTO Flock(flock\_name, leader\_id)

VALUES('первая стая', 2);

INSERT INTO Flock(flock\_name, leader\_id)

VALUES('вторая стая', 3);

INSERT INTO Flock(flock\_name, leader\_id)

VALUES('третья стая', 4);

INSERT INTO Flock(flock\_name, leader\_id)

VALUES('четвертая стая', 1);

INSERT INTO Flock\_to\_Characteristic

SELECT 1, 3

WHERE

NOT EXISTS (

SELECT flock\_id, f\_characteristic\_id FROM Flock\_to\_Characteristic WHERE flock\_id = 1 and f\_characteristic\_id = 3

);

INSERT INTO Flock\_to\_Characteristic

SELECT 2, 4

WHERE

NOT EXISTS (

SELECT flock\_id, f\_characteristic\_id FROM Flock\_to\_Characteristic WHERE flock\_id = 2 and f\_characteristic\_id = 4

);

INSERT INTO Flock\_to\_Characteristic

SELECT 3, 1

WHERE

NOT EXISTS (

SELECT flock\_id, f\_characteristic\_id FROM Flock\_to\_Characteristic WHERE flock\_id = 3 and f\_characteristic\_id = 1

);

INSERT INTO Flock\_to\_Characteristic

SELECT 4, 5

WHERE

NOT EXISTS (

SELECT flock\_id, f\_characteristic\_id FROM Flock\_to\_Characteristic WHERE flock\_id = 4 and f\_characteristic\_id = 5

);

1. **drop.sql**

DROP TABLE Flock\_Characteristic CASCADE;

DROP TABLE Movement\_Flock CASCADE;

DROP TABLE Flock\_Action CASCADE;

DROP TABLE Flock\_to\_Characteristic CASCADE;

DROP TABLE Flock CASCADE;

DROP TABLE Leader CASCADE;

DROP TABLE Leader\_Action CASCADE;

DROP TABLE Weapon CASCADE;

DROP TABLE Tribesman CASCADE;

**Вывод:**

В ходе выполнения лабораторной работы я научился различать инфологическую и даталогическую модель данных, научился создавать таблицы в базе данных, добавлять данные в таблице и получать данные из таблиц посредством запросов. Я познакомился с DDL, DML и СУБД PostgreSQL.