

Федеральное государственное автономное образовательное учреждение  
высшего образования «Национальный исследовательский университет  
ИТМО»

*Факультет программной инженерии и компьютерной  
техники*

*Направление подготовки: 09.03.01 - Информатика и  
вычислительная техника, Компьютерные системы и  
технологии*

*Дисциплина «Базы данных»*

**Отчет**  
**По лабораторной работе №1**  
**Вариант №52236**

Студент:

Миронов Иван Николаевич

Группа: Р3132

Преподаватель:

Афанасьев Дмитрий Борисович

Санкт-Петербург, 2024 год

## Оглавление

Текст задания.....	3
Описание предметной области .....	4
Список сущностей и их классификация .....	5
Модели.....	6
Инфологическая модель .....	6
Даталогическая модель.....	6
Реализация даталогической модели на SQL.....	7
Заключение: .....	10

## Текст задания

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.

Для создания объектов базы данных у каждого студента есть своя схема. Название схемы соответствует имени пользователя в базе studs (sXXXXXX). Команда для подключения к базе studs:

```
psql -h pg -d studs
```

Каждый студент должен использовать свою схему при работе над лабораторной работой №1 (а также в рамках выполнения 2, 3 и 4 этапа курсовой работы).

## Описание предметной области

В строгом смысле слова мы и специалистами-то в них не являемся. Меня, например, Мискатоникский университет направил в Антарктику как геолога: с помощью замечательной буровой установки, сконструированной профессором нашего же университета Фрэнком Х. Пэбоди, мы должны были добыть с большой глубины образцы почвы и пород. Не стремясь прослыть пионером в других областях науки, я тем не менее надеялся, что это новое механическое устройство поможет мне многое разведать и увидеть в ином свете.

## Список сущностей и их классификация

### Стержневые:

- Человек – имя, фамилия, надежда
- Машина – название, предназначение
- Университет – название
- Командировка – местоположение, университет, цель, даты начала и конца, статус

### Характеристические:

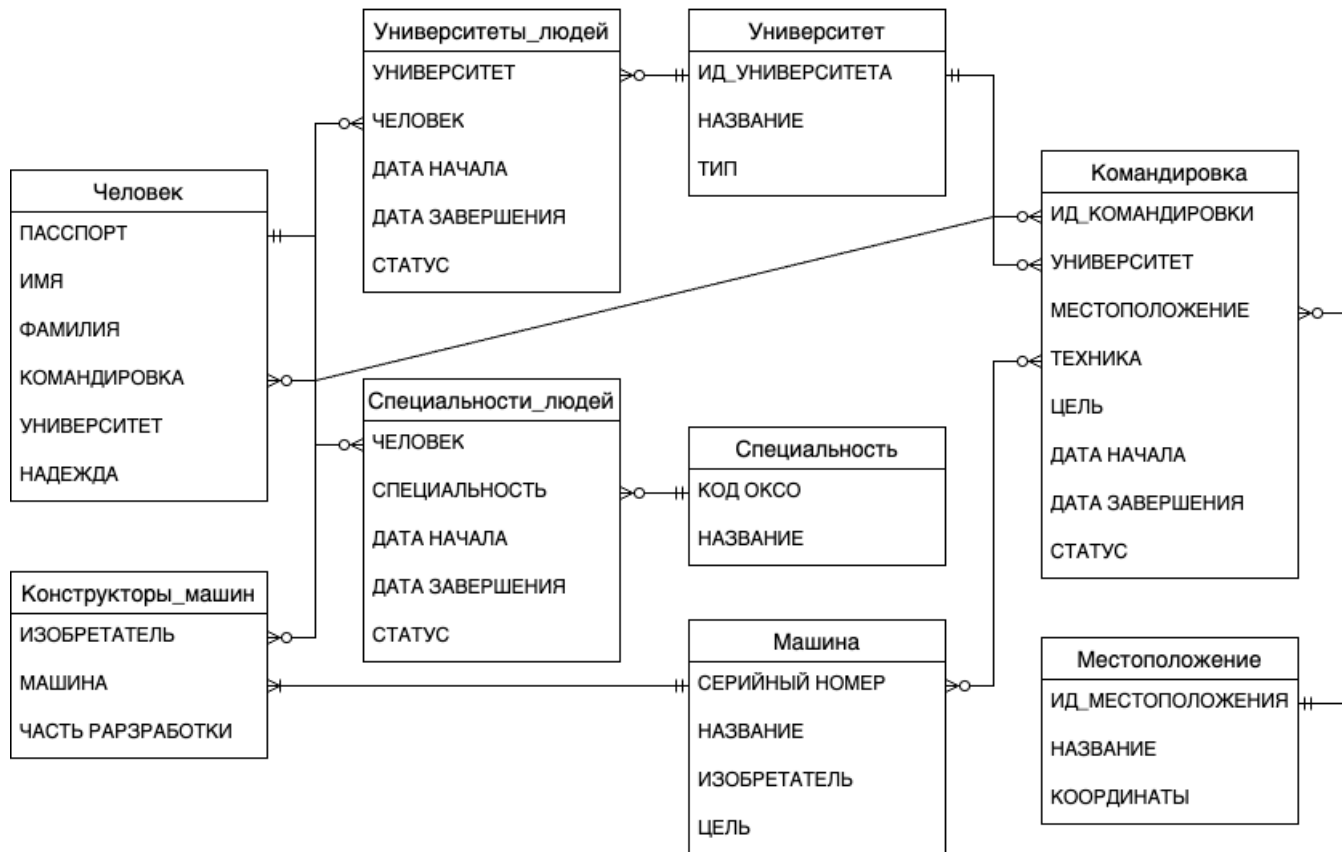
- Местоположение – название, координаты
- Специальность – название

### Ассоциативные:

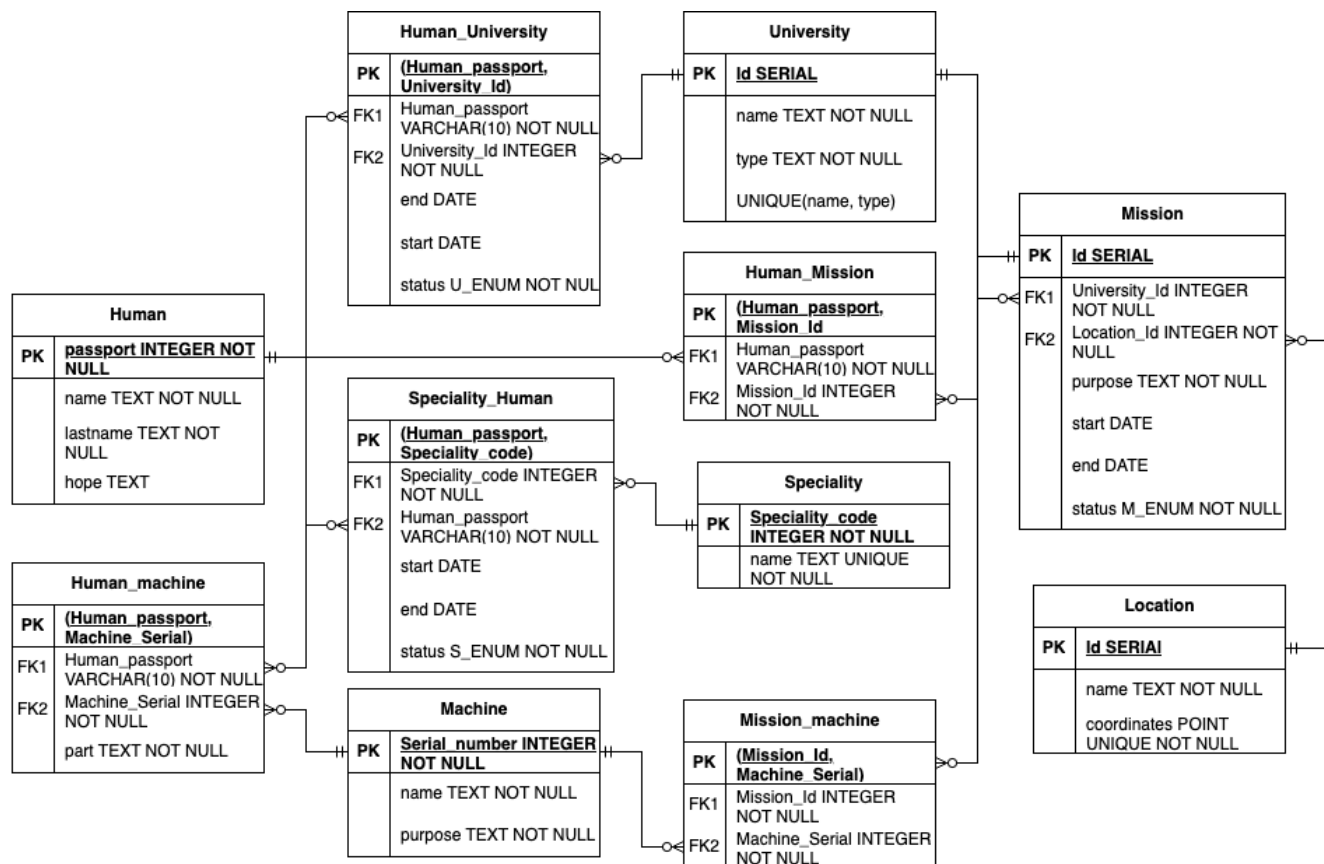
- Конструкторы машин – машина–человек, вклад в создание модели
- Университеты людей – университет-человек, даты начала и конца, статус
- Специальности людей – специальность-человек, даты начала и конца

## Модели

### Инфологическая модель



### Даталогическая модель



## Реализация даталогической модели на SQL

```
DROP TYPE IF EXISTS status_m CASCADE;  
DROP TYPE IF EXISTS status_u CASCADE;  
DROP TYPE IF EXISTS status_s CASCADE;
```

```
DROP TABLE IF EXISTS human CASCADE;  
DROP TABLE IF EXISTS machine CASCADE;  
DROP TABLE IF EXISTS university CASCADE;  
DROP TABLE IF EXISTS location CASCADE;  
DROP TABLE IF EXISTS mission CASCADE;  
DROP TABLE IF EXISTS mission_machine CASCADE;  
DROP TABLE IF EXISTS human_mission CASCADE;  
DROP TABLE IF EXISTS human_university CASCADE;  
DROP TABLE IF EXISTS human_machine CASCADE;  
DROP TABLE IF EXISTS speciality CASCADE;  
DROP TABLE IF EXISTS speciality_human;
```

```
CREATE TYPE status_m AS ENUM ('succeeded', 'failed', 'planned', 'continues');  
CREATE TYPE status_s AS ENUM ('working', 'fired', 'leaved');  
CREATE TYPE status_u AS ENUM ('expelled', 'succeeded', 'planned', 'studying');
```

```
CREATE TABLE IF NOT EXISTS human (  
    passport VARCHAR(10) NOT NULL PRIMARY KEY,  
    name TEXT NOT NULL,  
    lastname TEXT NOT NULL,  
    hope TEXT  
);
```

```
CREATE TABLE IF NOT EXISTS machine (  
    serial_number INTEGER PRIMARY KEY,  
    name TEXT NOT NULL,  
    purpose TEXT NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS university (  
    id SERIAL PRIMARY KEY, name TEXT NOT NULL,  
    type_u TEXT NOT NULL,  
    UNIQUE(name, type_u)  
);
```

```
CREATE TABLE IF NOT EXISTS location (  
    id SERIAL PRIMARY KEY,  
    name TEXT NOT NULL,  
    coordinates POINT NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS mission (  
    id SERIAL PRIMARY KEY,  
    location INTEGER REFERENCES location(id) NOT NULL,  
    university INTEGER REFERENCES university(id) NOT NULL,  
    purpose TEXT NOT NULL,  
    start_date DATE,  
    end_date DATE,  
    status status_m NOT NULL
```

);

```
CREATE TABLE IF NOT EXISTS speciality (  
    speciality_code INTEGER NOT NULL PRIMARY KEY,  
    name TEXT UNIQUE NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS speciality_human (  
    speciality_code INTEGER REFERENCES speciality(speciality_code) NOT NULL,  
    human_passport VARCHAR(10) REFERENCES human(passport) NOT NULL,  
    start_date DATE,  
    end_date DATE,  
    status status_s,  
    PRIMARY KEY (speciality_code, human_passport)  
);
```

```
CREATE TABLE IF NOT EXISTS mission_machine (  
    mission_id INTEGER REFERENCES mission(id) NOT NULL,  
    machine_num INTEGER REFERENCES machine(serial_number) NOT NULL,  
    PRIMARY KEY (mission_id, machine_num)  
);
```

```
CREATE TABLE IF NOT EXISTS human_mission (  
    mission_id INTEGER REFERENCES mission(id) NOT NULL,  
    human_passport VARCHAR(10) REFERENCES human(passport) NOT NULL,  
    PRIMARY KEY (mission_id, human_passport)  
);
```

```
CREATE TABLE IF NOT EXISTS human_university (  
    human_passport VARCHAR(10) REFERENCES human(passport) NOT NULL,  
    university_id INTEGER REFERENCES university(id) NOT NULL,  
    start_date DATE,  
    end_date DATE,  
    status status_u NOT NULL,  
    PRIMARY KEY (human_passport, university_id)  
);
```

```
CREATE TABLE IF NOT EXISTS human_machine (  
    human_passport VARCHAR(10) REFERENCES human(passport) NOT NULL,  
    machine_num INTEGER REFERENCES machine(serial_number) NOT NULL,  
    part TEXT NOT NULL,  
    PRIMARY KEY (human_passport, machine_num)  
);
```

```
INSERT INTO human (passport, name, lastname, hope) VALUES  
('8978126743', 'John', 'Johnson', 'mechanical device assistance');
```

```
INSERT INTO human (passport, name, lastname) VALUES  
('3278230917', 'Freddy', 'Ivanov'),  
('2387835297', 'Frank', 'Peabody'),  
('8237510842', 'Jim', 'Petrov');
```

```
INSERT INTO speciality (speciality_code, name) VALUES  
(3298, 'geologist'),  
(23875, 'physicist'),
```



(2387, 'programmer');

INSERT INTO machine (serial\_number, name, purpose) VALUES  
(8289, 'drilling rig', 'obtain soil samples'),  
(2839, 'construction\_crane', 'buid house');

INSERT INTO university (name, type\_u) VALUES  
(('Miskatonic University', 'humanytarian'),  
(('ITMO', 'technical'));

INSERT INTO location (name, coordinates) VALUES  
(('Antarctica', POINT(69.3, 65)),  
(('Moscow', POINT(55.7558, 37.6173)));

INSERT INTO speciality\_human (speciality\_code, human\_passport, start\_date, status)  
VALUES  
(3298, '8978126743', '2020-01-21', 'working');

INSERT INTO speciality\_human (speciality\_code, human\_passport, start\_date, end\_date,  
status) VALUES  
(2387, '8237510842', '2021-12-02', '2024-01-20', 'fired');

INSERT INTO mission (location, university, purpose, status) VALUES  
(1, 1, 'obtain soil samples', 'continues'),  
(2, 1, 'wash sneakers in machine', 'continues');

INSERT INTO human\_mission (human\_passport, mission\_id) VALUES  
(('8978126743', 1),  
(('3278230917', 2),  
(('2387835297', 2);

INSERT INTO human\_university (human\_passport, university\_id, start\_date, end\_date,  
status) VALUES  
(('8978126743', 1, '2021-05-19', '2028-01-10', 'studying');

INSERT INTO mission\_machine (mission\_id, machine\_num) VALUES  
(1, 8289),  
(2, 2839);

INSERT INTO human\_machine (human\_passport, machine\_num, part) VALUES  
(('2387835297', 8289, 'construction');

## Заключение:

В ходе выполнения лабораторной работы я ознакомился с архитектурой построения баз данных ANSI/SPARC, научился строить инфологические и даталогические модели, работать с СУБД PostgreSQL