МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 1

по дисциплине

‘Базы данных’

Вариант №310915

Выполнил:
Студент группы P3131
Дворкин Борис
Александрович

Преподаватель:
Наумова Надежда
Александровна

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2023

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:

«На вопрос Пула не очень-то легко было ответить. Они отрезаны от Земли. Собственно, само по себе это еще не угрожало безопасности корабля, и можно найти много способов восстановить связь. На худой конец - жестко зафиксировать антенну и наводить на Землю сам корабль. Задача чертовски трудная и на завершающем этапе полета доставила бы им кучу лишних хлопот, но это все же можно сделать, если все остальные попытки сорвутся.»

Речь идет о ситуации на борту корабля, который отрезан от Земли. На борту корабля есть люди, у которых есть национальность и происхождение. У корабля и людей есть относительное расположение в пространстве(координаты). Корабли бывают разных типов – космические и т.д. Чтобы восстановить связь с Землей, можно попробовать жестко зафиксировать антенну и наводить ее на Землю, но это достаточно сложная задача, которая может привести к лишним хлопотам на завершающем этапе полета. Сл-но, у людей есть проблемы. И у корабля есть «поломки» (всё troubles). У корабля есть модули. Антенна – модуль корабля.

Список сущностей:

Стержневые:

- *Корабль* – id, связь_с_землёй, безопасность, модули_корабля, местонахождение, тип_корабля, проблемы
- *Человек* – id, имя, фамилия, возраст, национальность, происхождение, проблемы, местонахождение

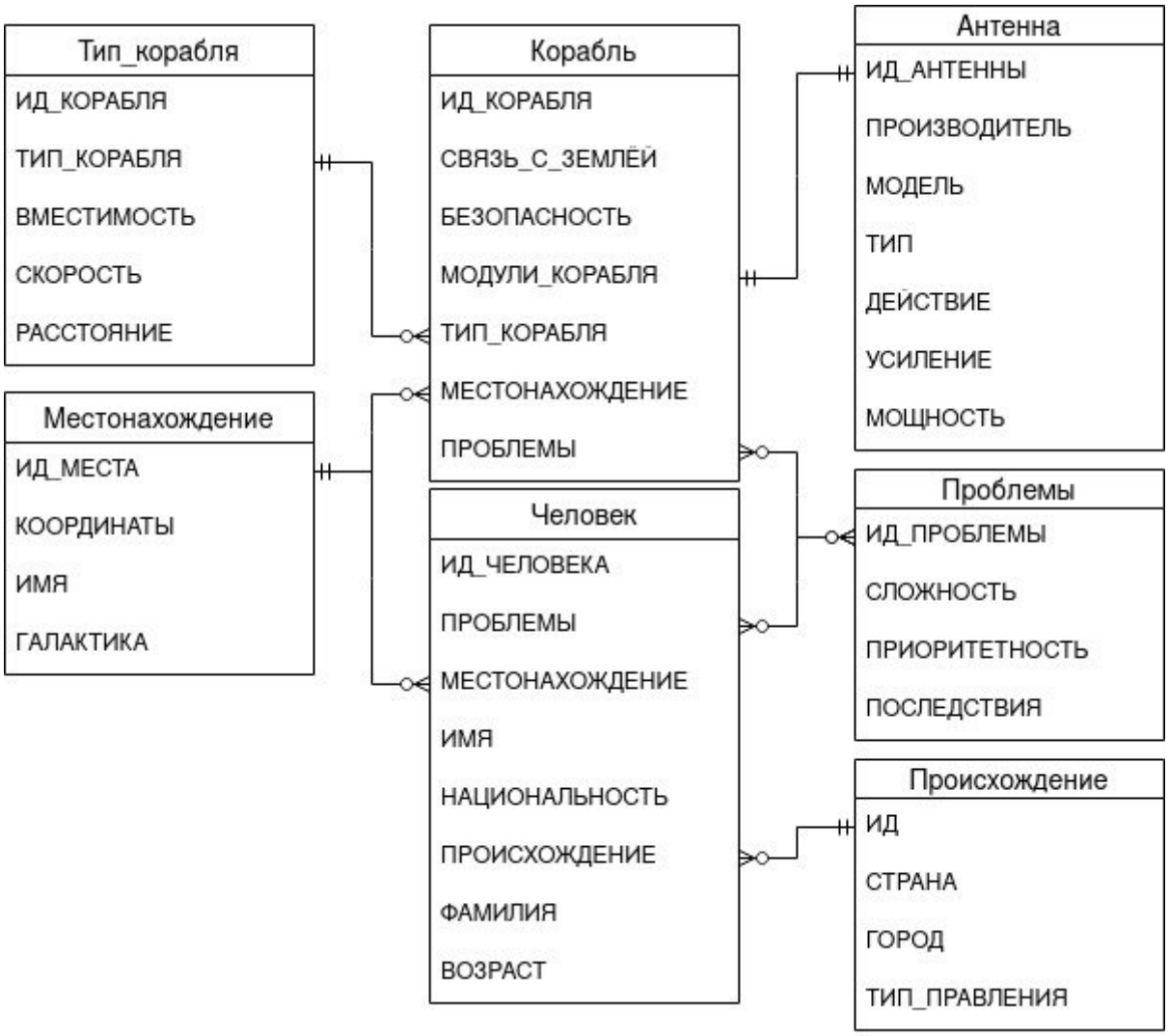
Ассоциации:

- *Место* – корабль-человек
- *Проблемы*_корабля – корабль-проблемы
- *Проблемы_человека* – человек-проблемы

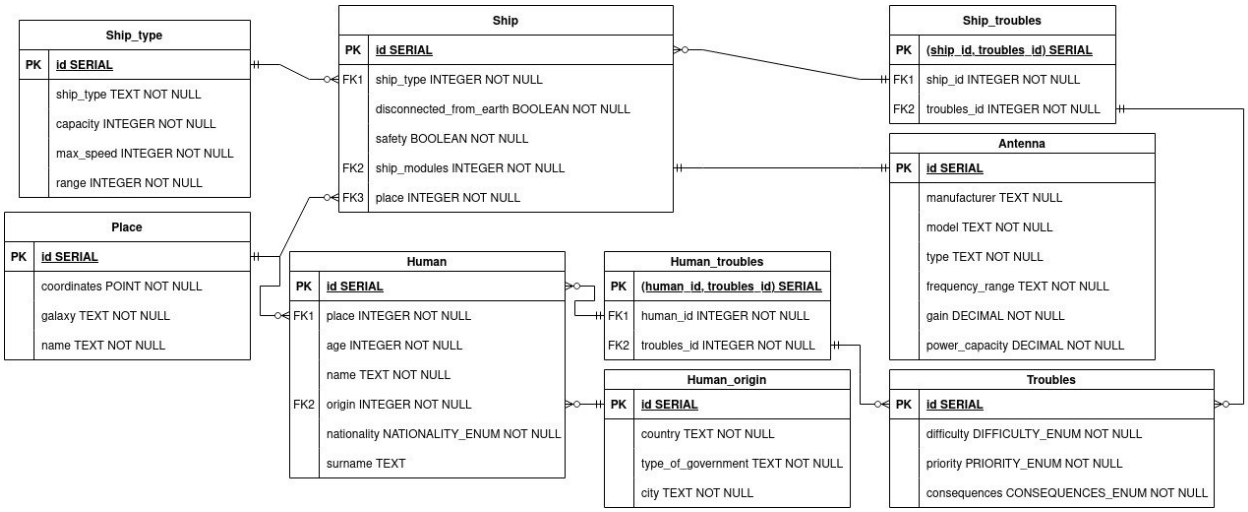
Характеристики:

- *Происхождение человека* – id, страна, город, тип_правления
- *Тип корабля* – id, тип корабля, вместимость, скорость, расстояние
- *Антенна* – id, производитель, модель, тип, действие, усиление, мощь

Инфологическая модель:



Даталогическая модель:



Реализация на уровне PostgreSQL:

```
-- dropping enum types
DROP TYPE IF EXISTS nationality_enum CASCADE;
DROP TYPE IF EXISTS priority_enum CASCADE;
DROP TYPE IF EXISTS difficulty_enum CASCADE;
DROP TYPE IF EXISTS consequences_enum CASCADE;
DROP TYPE IF EXISTS type_of_government_enum CASCADE;

-- dropping tables
DROP TABLE IF EXISTS human_troubles CASCADE;
DROP TABLE IF EXISTS ship_troubles CASCADE;
DROP TABLE IF EXISTS human CASCADE;
DROP TABLE IF EXISTS human_origin CASCADE;
DROP TABLE IF EXISTS ship CASCADE;
DROP TABLE IF EXISTS ship_type CASCADE;
DROP TABLE IF EXISTS place CASCADE;
DROP TABLE IF EXISTS troubles CASCADE;
DROP TABLE IF EXISTS antenna CASCADE;

-- dropping domains
DROP DOMAIN IF EXISTS positive_integer CASCADE;
DROP DOMAIN IF EXISTS positive_decimal CASCADE;
DROP DOMAIN IF EXISTS max_speed_constraint CASCADE;
DROP DOMAIN IF EXISTS range_constraint CASCADE;
DROP DOMAIN IF EXISTS power_capacity_constraint CASCADE;
DROP DOMAIN IF EXISTS gain_constraint CASCADE;
DROP DOMAIN IF EXISTS ship_capacity_constraint CASCADE;
DROP DOMAIN IF EXISTS age_constraint CASCADE;

-- creating enums if they're exists
CREATE TYPE nationality_enum AS ENUM ('American', 'British',
'Canadian', 'Chinese', 'French', 'German', 'Indian', 'Japanese', 'Russian',
'Spanish');
CREATE TYPE priority_enum AS ENUM ('high', 'medium', 'low');
CREATE TYPE difficulty_enum AS ENUM ('easy', 'moderate', 'hard');
CREATE TYPE consequences_enum AS ENUM ('minimal', 'moderate',
'severe', 'catastrophic');
CREATE TYPE type_of_government_enum AS ENUM ('democracy',
'monarchy', 'dictatorship', 'communism', 'socialism');

-- create domain's
CREATE DOMAIN positive_integer AS INTEGER
CHECK (VALUE > 0);
```

```
CREATE DOMAIN positive_decimal AS DECIMAL
CHECK (VALUE > 0);

CREATE DOMAIN max_speed_constraint AS positive_integer;
CREATE DOMAIN range_constraint AS positive_integer;
CREATE DOMAIN power_capacity_constraint AS positive_decimal;
CREATE DOMAIN gain_constraint AS positive_decimal;
CREATE DOMAIN ship_capacity_constraint AS positive_integer;
CREATE DOMAIN age_constraint AS positive_integer;

-- creating tables if they're exists
CREATE TABLE IF NOT EXISTS antenna (
    id SERIAL PRIMARY KEY,
    manufacturer TEXT NULL,
    model TEXT NOT NULL,
    type TEXT NOT NULL,
    frequency_range TEXT NOT NULL,
    gain gain_constraint NOT NULL,
    power_capacity power_capacity_constraint NOT NULL
);

CREATE TABLE IF NOT EXISTS troubles (
    id SERIAL PRIMARY KEY,
    difficulty difficulty_enum NOT NULL,
    consequences consequences_enum NOT NULL,
    priority priority_enum NOT NULL
);

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

CREATE TABLE IF NOT EXISTS ship_type (
    id SERIAL PRIMARY KEY,
    ship_type TEXT NOT NULL,
    ship_capacity ship_capacity_constraint NOT NULL,
    max_speed max_speed_constraint NOT NULL,
    range range_constraint NOT NULL
);

CREATE TABLE IF NOT EXISTS ship (
    id SERIAL PRIMARY KEY,
```

```
    disconnected_from_earth BOOLEAN NOT NULL,
    safety BOOLEAN NOT NULL,
    ship_modules INTEGER REFERENCES antenna(id) NOT NULL,
    place INTEGER REFERENCES place(id) NOT NULL,
    ship_type INTEGER REFERENCES ship_type(id) NOT NULL
);

CREATE TABLE IF NOT EXISTS human_origin (
    id SERIAL PRIMARY KEY,
    country TEXT NOT NULL,
    type_of_government type_of_government_enum NOT NULL,
    city TEXT NOT NULL
);

CREATE TABLE IF NOT EXISTS human (
    id SERIAL PRIMARY KEY,
    name TEXT NOT NULL,
    surname TEXT NOT NULL,
    age age_constraint NOT NULL,
    nationality nationality_enum NOT NULL,
    origin INTEGER REFERENCES human_origin(id) NOT NULL,
    place INTEGER REFERENCES place(id) NOT NULL
);

CREATE TABLE IF NOT EXISTS human_troubles (
    human_id INTEGER REFERENCES human(id) NOT NULL,
    troubles_id INTEGER REFERENCES troubles(id) NOT NULL,
    PRIMARY KEY (human_id, troubles_id)
);

CREATE TABLE IF NOT EXISTS ship_troubles (
    ship_id INTEGER REFERENCES ship(id) NOT NULL,
    troubles_id INTEGER REFERENCES troubles(id) NOT NULL,
    PRIMARY KEY (ship_id, troubles_id)
);
```

Заполнение тестовыми значениями:

```
-- inserting data into antenna table
INSERT INTO antenna (manufacturer, model, type, frequency_range, gain, power_capacity) VALUES
('NASA', 'X1', 'parabolic', '8-12 GHz', 10.5, 2000.00),
('ESA', 'V2', 'dipole', '2-8 GHz', 7.2, 1500.00),
('JAXA', 'K3', 'helical', '4-10 GHz', 6.3, 1800.00);
-- inserting data into troubles table
INSERT INTO troubles (difficulty, consequences, priority) VALUES
('easy', 'minimal', 'low'),
('moderate', 'moderate', 'medium'),
('hard', 'catastrophic', 'high');
-- inserting data into place table
INSERT INTO place (coordinates, galaxy, name) VALUES
('(-35.6751, 174.3550)', 'Milky Way', 'New Zealand'),
('(51.5074, -0.1278)', 'Milky Way', 'London'),
('(37.7749, -122.4194)', 'Milky Way', 'San Francisco');
-- inserting data into ship_type table
INSERT INTO ship_type (ship_type, ship_capacity, max_speed, range)
VALUES
('Frigate', 50, 60, 1500),
('Destroyer', 70, 80, 2000),
('Cruiser', 100, 90, 2500);
-- inserting data into ship table
INSERT INTO ship (disconnected_from_earth, safety, ship_modules, place, ship_type) VALUES
(true, true, 1, 1, 1),
(false, false, 2, 2, 3),
(true, true, 3, 3, 2);
-- inserting data into human_origin table
INSERT INTO human_origin (country, type_of_government, city) VALUES
('USA', 'democracy', 'New York'),
('UK', 'monarchy', 'London'),
('India', 'democracy', 'New Delhi');
-- inserting data into human table
INSERT INTO human (name, surname, age, nationality, origin, place)
VALUES
('John', 'Doe', 30, 'American', 1, 1),
('Alice', 'Smith', 25, 'British', 2, 2),
('Ravi', 'Patel', 40, 'Indian', 3, 3);
```

```sql
-- inserting data into human_troubles table
INSERT INTO human_troubles (human_id, troubles_id) VALUES
(1, 1),
(2, 2),
(3, 3);
-- inserting data into ship_troubles table
INSERT INTO ship_troubles (ship_id, troubles_id) VALUES
(1, 1),
(2, 2),
(3, 3);
```

Вывод: во время выполнения лабораторной работы я ознакомился с архитектурой построения ANSI-SPARC и базовым синтаксисом PostgreSQL, научился создавать инфологические и даталогические диаграммы, enum’ы, ssh’а также создавать серверную базу данных и с ней взаимодействовать.