

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 Р3131

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

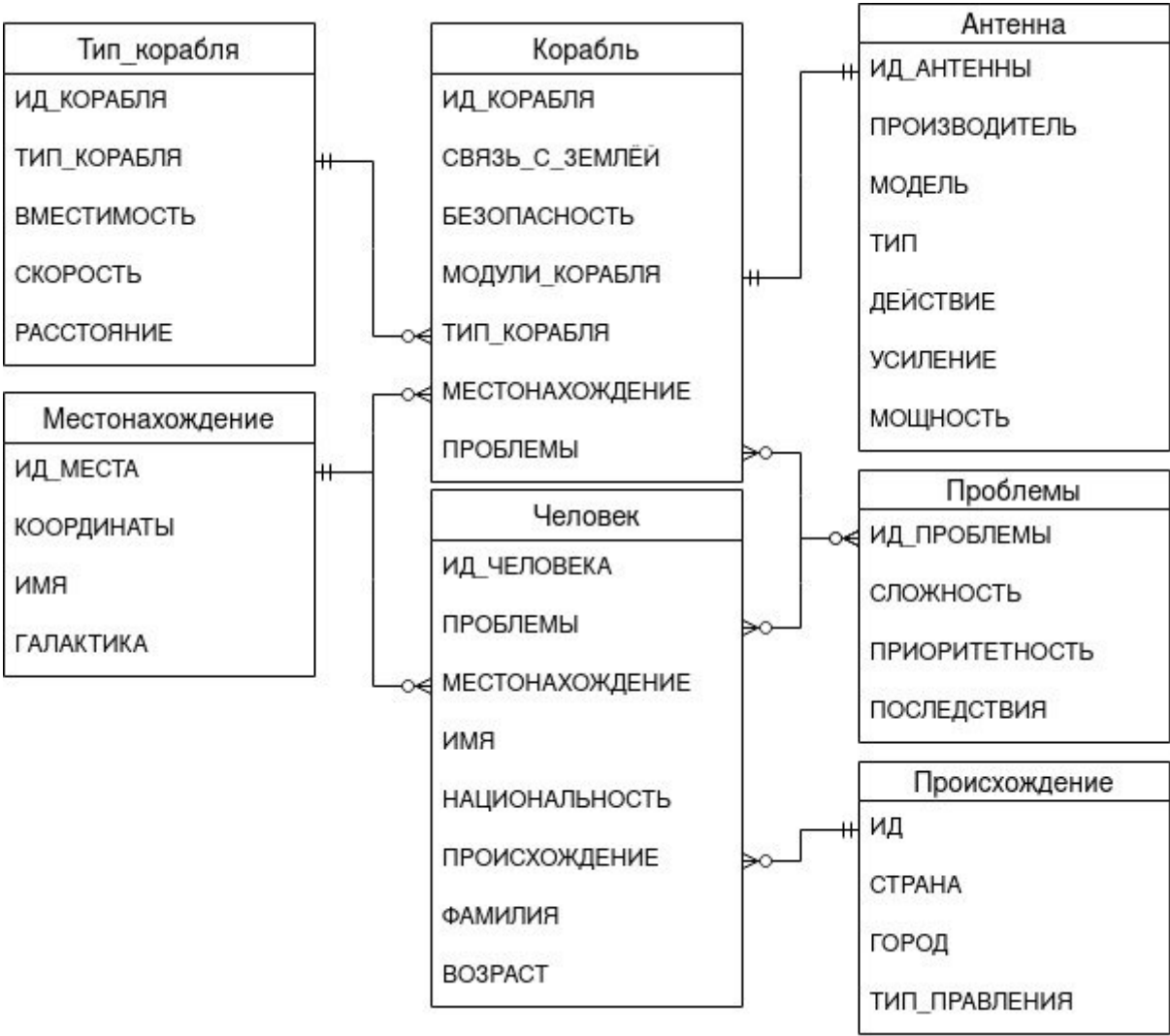
Ассоциации:

- *Место* – корабль-человек
- *Проблемы_корабля* – корабль-проблемы
- *Проблемы_человека* – человек-проблемы

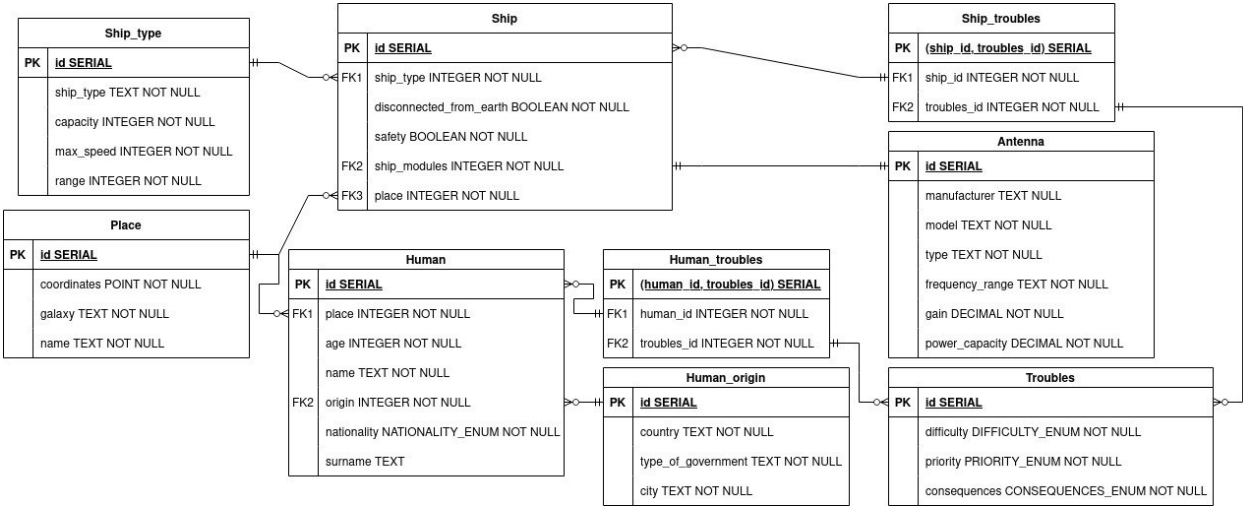
Характеристики:

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

Инфологическая модель:



Даталогическая модель:



Реализация на уровне PostgreSQL:

-- dropping enum types if they're exists

```
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;
```

-- dropping tables if they're exists

```
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;
```

-- 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');
```

-- 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 DECIMAL NOT NULL,  
    power_capacity DECIMAL 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,  
    capacity INTEGER NOT NULL,  
    max_speed INTEGER NOT NULL,  
    range INTEGER 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 TEXT 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 INTEGER 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:

```
INSERT INTO antenna (manufacturer, model, type, frequency_range, gain,
power_capacity) VALUES ('Company A', 'Model 1', 'Parabolic', '10-15 GHz',
25.4, 500);
INSERT INTO antenna (manufacturer, model, type, frequency_range, gain,
power_capacity) VALUES ('Company B', 'Model 2', 'Horn', '20-25 GHz',
30.2, 750);
INSERT INTO troubles (difficulty, consequences, priority) VALUES ('hard',
'severe', 'high');
INSERT INTO troubles (difficulty, consequences, priority) VALUES
('moderate', 'moderate', 'medium');
INSERT INTO place (coordinates, galaxy, name) VALUES (POINT(12.345,
-67.890), 'Milky Way', 'Earth');
INSERT INTO place (coordinates, galaxy, name) VALUES (POINT(23.456,
89.012), 'Andromeda', 'Planet X');
INSERT INTO ship_type (shipType, capacity, maxSpeed, range) VALUES
('Cruiser', 1500, 300, 6000);
INSERT INTO ship_type (shipType, capacity, maxSpeed, range) VALUES
('Fighter', 2, 800, 500);
INSERT INTO ship (disconnected_from_earth, safety, shipModules, place,
shipType) VALUES (TRUE, FALSE, 1, 1, 1);
INSERT INTO ship (disconnected_from_earth, safety, shipModules, place,
shipType) VALUES (FALSE, TRUE, 2, 2, 2);
INSERT INTO nationality (nationality) VALUES ('American');
INSERT INTO nationality (nationality) VALUES ('Russian');
INSERT INTO HumanOrigin (country, typeOfGovernment, City) VALUES
('United States', 'Presidential', 'New York');
INSERT INTO HumanOrigin (country, typeOfGovernment, City) VALUES
('Russia', 'Federal semi-presidential constitutional republic', 'Moscow');
INSERT INTO human (name, surname, age, nationality, origin, place)
VALUES ('John', 'Smith', 30, 'American', 1, 1);
INSERT INTO human (name, surname, age, nationality, origin, place)
VALUES ('Ivan', 'Ivanov', 35, 'Russian', 2, 2);
INSERT INTO human_troubles (human_id, troubles_id) VALUES (1, 1);
INSERT INTO human_troubles (human_id, troubles_id) VALUES (2, 2);
INSERT INTO ship_troubles (ship_id, troubles_id) VALUES (1, 1);
INSERT INTO ship_troubles (ship_id, troubles_id) VALUES (2, 2);
```

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.

