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



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

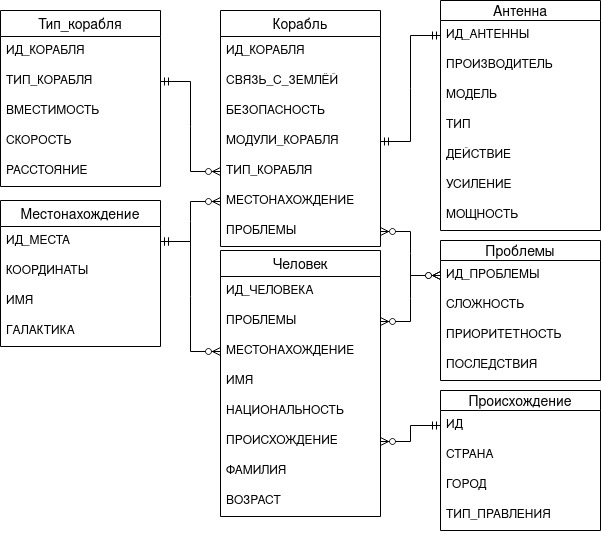
Ассоциации:

* *Место* – корабль-человек
* *Проблемы*\_корабля – корабль-проблемы
* *Проблемы\_человека* – человек-проблемы

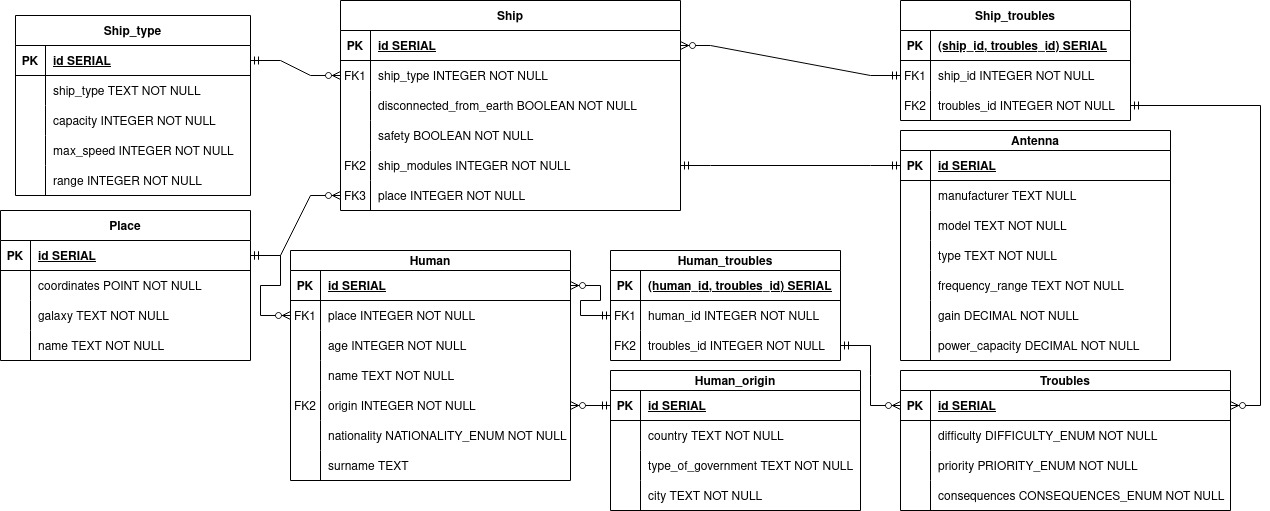
Характеристики:

* *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 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.