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

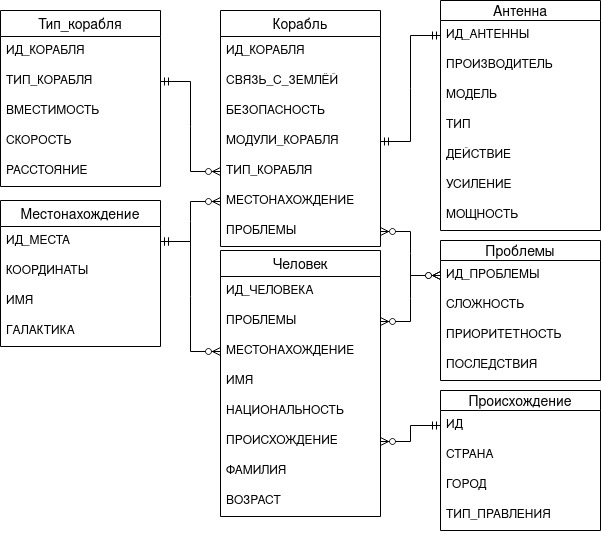
Ассоциации:

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

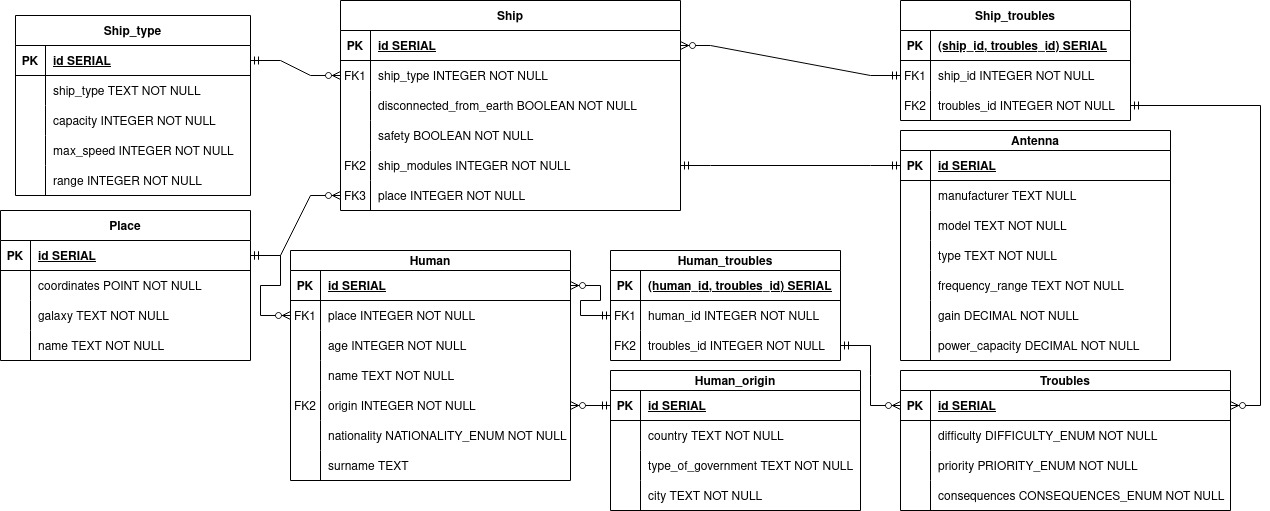
Характеристики:

* *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

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', 'republic', 'constitutional monarchy', 'parliamentary republic', 'parliamentary democracy', 'federal semi-presidential republic', 'federal presidential republic', 'federal parliamentary constitutional republic', 'absolute monarchy');

-- 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:**

-- dropping created previously sequences

DROP SEQUENCE troubles\_id\_sequence;

DROP SEQUENCE human\_id\_sequence;

-- creating sequences for handy inserting data into tables

CREATE SEQUENCE troubles\_id\_sequence START 1;

CREATE SEQUENCE human\_id\_sequence START 1;

-- insert data into antenna

INSERT INTO antenna (manufacturer, model, type, frequency\_range, gain, power\_capacity)

VALUES

('AntennaCorp', 'SuperAntenna', 'Dipole', '10 MHz - 1 GHz', 12.5, 100),

('AntennaWorks', 'HyperGain', 'Yagi', '1 GHz - 10 GHz', 16.2, 500),

('TechAntenna', 'UltraBeam', 'Parabolic', '2 GHz - 18 GHz', 20.1, 1000),

('AntennaCo', 'MegaHorn', 'Horn', '100 MHz - 6 GHz', 10.8, 200),

('AntennaCorp', 'OmniAntenna', 'Omni-Directional', '1 MHz - 100 MHz', 6.4, 50),

('AntennaWorks', 'SlimLine', 'Patch', '900 MHz - 2.4 GHz', 8.7, 300),

('TechAntenna', 'UltraFlex', 'Flexible', '500 MHz - 2 GHz', 9.3, 100),

('AntennaCo', 'GigaHorn', 'Horn', '10 GHz - 100 GHz', 12.1, 500),

('AntennaCorp', 'Helix', 'Helical', '300 MHz - 2.4 GHz', 7.6, 150),

('AntennaWorks', 'MegaPatch', 'Patch', '1.5 GHz - 5 GHz', 14.5, 400);

-- inserting data into troubles table

INSERT INTO troubles (difficulty, consequences, priority) VALUES

('easy', 'minimal', 'low'),

('moderate', 'moderate', 'medium'),

('hard', 'catastrophic', 'high');

-- insert data into place

INSERT INTO place (coordinates, galaxy, name)

VALUES

(POINT(40.7128, -74.0060), 'Milky Way', 'New York City'),

(POINT(51.5074, -0.1278), 'Milky Way', 'London'),

(POINT(43.6532, -79.3832), 'Milky Way', 'Toronto'),

(POINT(39.9042, 116.4074), 'Milky Way', 'Beijing'),

(POINT(48.8566, 2.3522), 'Milky Way', 'Paris'),

(POINT(52.5200, 13.4050), 'Milky Way', 'Berlin'),

(POINT(28.7041, 77.1025), 'Milky Way', 'New Delhi'),

(POINT(35.6895, 139.6917), 'Milky Way', 'Tokyo'),

(POINT(55.7558, 37.6173), 'Milky Way', 'Moscow'),

(POINT(40.4168, -3.7038), 'Milky Way', 'Madrid');

-- insert data into ship\_type

INSERT INTO ship\_type (ship\_type, ship\_capacity, max\_speed, range)

VALUES

('Fighter', 1, 200, 1000),

('Cruiser', 10, 150, 5000),

('Frigate', 20, 100, 10000),

('Destroyer', 50, 75, 15000),

('Transport', 100, 50, 20000);

-- insert 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, 2),

(TRUE, FALSE, 3, 3, 3);

-- inserting data into human\_origin table

INSERT INTO human\_origin (country, type\_of\_government, city) VALUES

('Mexico', 'democracy', 'Mexico City'),

('Italy', 'republic', 'Rome'),

('Japan', 'constitutional monarchy', 'Tokyo'),

('South Africa', 'parliamentary republic', 'Cape Town'),

('Canada', 'parliamentary democracy', 'Ottawa'),

('Australia', 'parliamentary democracy', 'Canberra'),

('Russia', 'federal semi-presidential republic', 'Moscow'),

('Brazil', 'federal presidential republic', 'Brasília'),

('India', 'federal parliamentary constitutional republic', 'New Delhi'),

('Saudi Arabia', 'absolute monarchy', 'Riyadh');

-- inserting data into human table

INSERT INTO human (name, surname, age, nationality, origin, place)

VALUES ('John', 'Doe', 35, 'American', 1, 1),

('Ivan', 'Ivanov', 28, 'Russian', 2, 2),

('Marie', 'Curie', 66, 'French', 3, 3),

('Alice', 'Johnson', 33, 'German', 4, 4);

-- Insert data into human\_troubles using the sequence

INSERT INTO human\_troubles (human\_id, troubles\_id)

SELECT nextval('human\_id\_sequence'), nextval('troubles\_id\_sequence')

FROM generate\_series(1, 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.