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

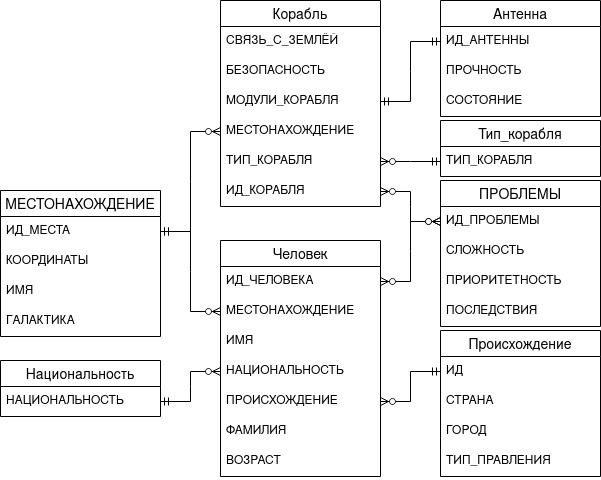
Ассоциации:

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

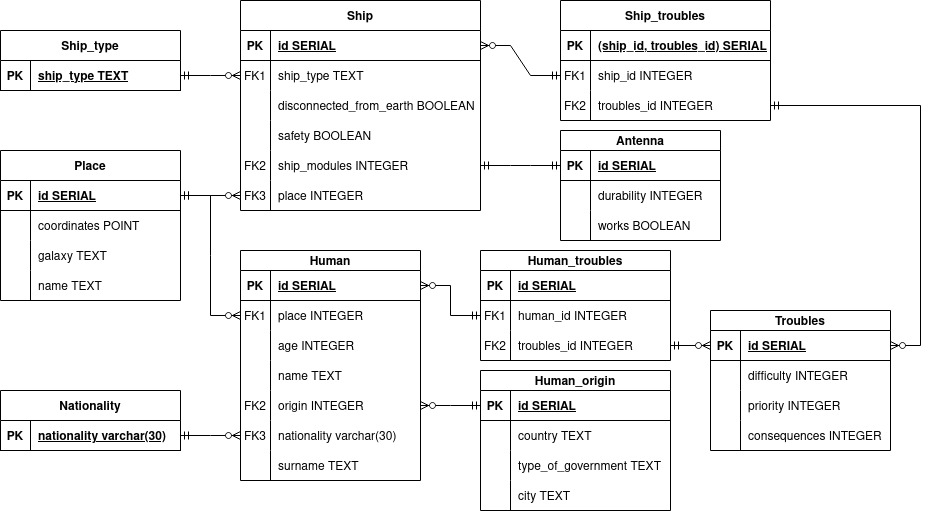
Характеристики:

* *Происхождение человека* – id, страна, город, тип\_правления
* *Национальность* – национальность
* *Тип корабля* – тип корабля
* *Антенна* – id, прочность, состояние

**Инфологическая модель:**



**Даталогическая модель:**



**Реализация на уровне PostgreSQL:**

CREATE TABLE antenna (

id SERIAL PRIMARY KEY,

durability INTEGER,

works BOOLEAN

);

CREATE TABLE troubles (

id SERIAL PRIMARY KEY,

difficulty INTEGER,

consequences INTEGER,

priority INTEGER

);

CREATE TABLE place (

id SERIAL PRIMARY KEY,

coordinates POINT,

galaxy TEXT,

name TEXT

);

CREATE TABLE shipType (

shipType TEXT PRIMARY KEY

);

CREATE TABLE ship (

id SERIAL PRIMARY KEY,

disconnected\_from\_earth BOOLEAN,

safety BOOLEAN,

shipModules INTEGER REFERENCES antenna,

place INTEGER REFERENCES place,

shipType TEXT REFERENCES shipType

);

CREATE TABLE nationality (

nationality varchar(30) PRIMARY KEY

);

CREATE TABLE HumanOrigin (

id SERIAL PRIMARY KEY,

country TEXT,

typeOfGovernment TEXT,

City TEXT

);

CREATE TABLE human (

id SERIAL PRIMARY KEY,

name TEXT,

surname TEXT,

age INTEGER,

nationality TEXT REFERENCES nationality,

origin INTEGER REFERENCES HumanOrigin,

place INTEGER REFERENCES place

);

CREATE TABLE human\_troubles (

human\_id INTEGER REFERENCES human(id),

troubles\_id INTEGER REFERENCES troubles(id),

PRIMARY KEY (human\_id, troubles\_id)

);

CREATE TABLE ship\_troubles (

ship\_id INTEGER REFERENCES ship(id),

troubles\_id INTEGER REFERENCES troubles(id),

PRIMARY KEY (ship\_id, troubles\_id)

);

**Заполнение тестовыми значениями:**

INSERT INTO antenna (durability, works)

VALUES (85, true),

(70, false),

(90, true);

INSERT INTO troubles (difficulty, consequences, priority)

VALUES (4, 20, 3),

(6, 30, 2),

(8, 40, 1);

INSERT INTO place (coordinates, galaxy, name)

VALUES ('(10,20)', 'Milky Way', 'Earth'),

('(30,40)', 'Andromeda', 'Planet X'),

('(50,60)', 'Triangulum', 'Planet Y');

INSERT INTO shipType (shipType)

VALUES ('Cargo'),

('Passenger'),

('Military');

INSERT INTO ship (disconnected\_from\_earth, safety, shipModules, place, shipType)

VALUES (true, true, 1, 1, 'Cargo'),

(false, false, 2, 2, 'Passenger'),

(true, true, 3, 3, 'Military');

INSERT INTO nationality (nationality)

VALUES ('American'),

('Russian'),

('Chinese');

INSERT INTO HumanOrigin (country, typeOfGovernment, City)

VALUES ('USA', 'Republic', 'New York'),

('Russia', 'Federal semi-presidential constitutional republic', 'Moscow'),

('China', 'Single-party socialist state', 'Beijing');

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

VALUES ('John', 'Doe', 25, 'American', 1, 1),

('Ivan', 'Ivanov', 35, 'Russian', 2, 2),

('Li', 'Wang', 45, 'Chinese', 3, 3);

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

VALUES (1, 2),

(2, 1),

(3, 3);

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

VALUES (1, 3),

(2, 2),

(3, 1);

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