

112332МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине
«Базы данных»

Вариант № 31302

Выполнил:

Студент группы Р3131

Валиев Руслан Новруз оглы

Преподаватель:

Вербовой Александр Александрович

Содержание

Задание	3
Список сущностей и их классификация	3
Инфологическая модель	4
Даталогическая модель	5
Реализация даталогической модели на языке SQL	6
Заключение	9

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.

Введите вариант:

Описание предметной области, по которой должна быть построена доменная модель:

И здесь, в тридцати миллионах километров, мчались луны Юпитера - другие, намного меньшие. Это были просто летающие горы поперечником в десятки километров, но трасса корабля не подходила близко ни к одной из них. Корабельный радар с промежутками в несколько минут посылал в пространство импульсы энергии, подобные беззвучным грозовым разрядам, и не получал ни одного отраженного сигнала из ближайших зон - вокруг было пусто.

Список сущностей и их классификация

1) Стержневые:

- Корабли
- Маршруты
- Спутники

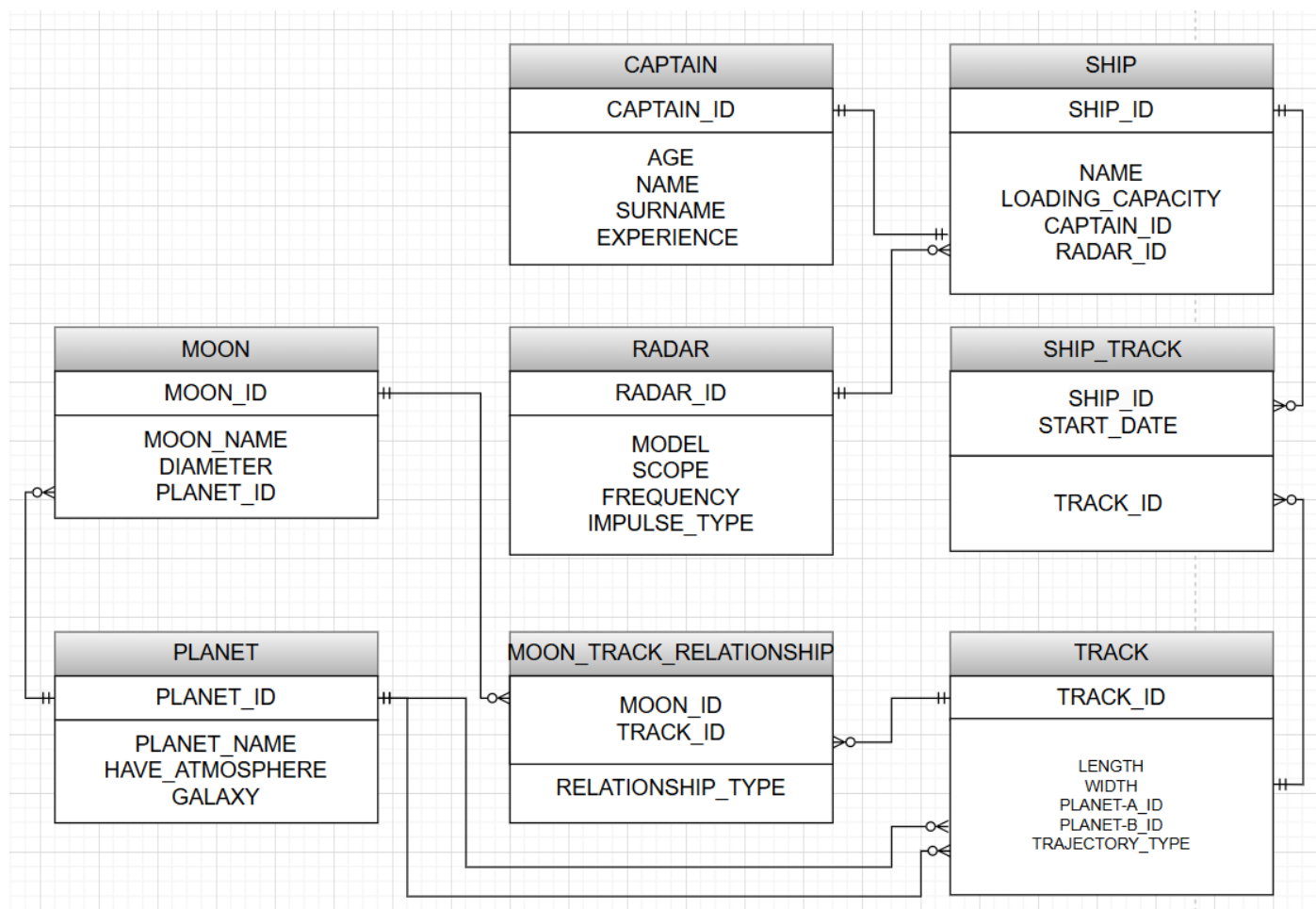
2) Характеристические:

- Капитаны
- Планеты
- Радары

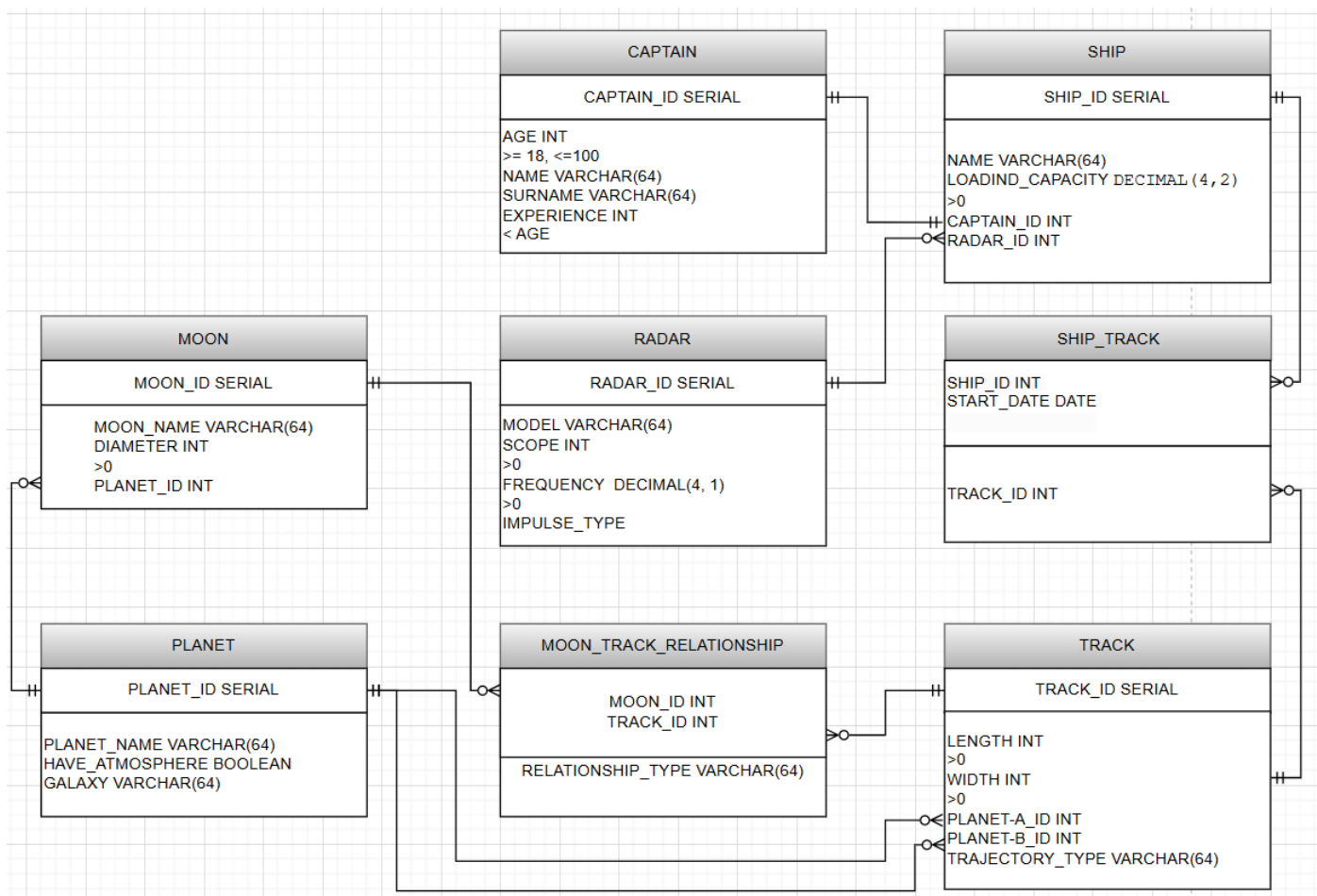
3) Ассоциативные:

- Связь кораблей и маршрутов
- Отношение между маршрутами и спутниками

Инфологическая модель



Датологическая модель



Реализация датологической модели на языке SQL

```
DROP TABLE IF EXISTS captain CASCADE;
DROP TABLE IF EXISTS ship CASCADE;
DROP TABLE IF EXISTS planet CASCADE;
DROP TABLE IF EXISTS moon CASCADE;
DROP TABLE IF EXISTS track CASCADE;
DROP TABLE IF EXISTS radar CASCADE;
DROP TABLE IF EXISTS moon_track_relationship CASCADE;
DROP TABLE IF EXISTS ship_track CASCADE;
```

```
CREATE TABLE radar (
  radar_id SERIAL PRIMARY KEY,
  model VARCHAR(64) NOT NULL,
  scope INT NOT NULL,
  frequency DECIMAL(4, 1) CHECK (frequency > 0) NOT NULL,
  impulse_type VARCHAR(64) NOT NULL
);
```

```
CREATE TABLE captain (
  captain_id SERIAL PRIMARY KEY,
  age INT CHECK (age >= 18 AND age <=100),
  name VARCHAR(64) NOT NULL,
  surname VARCHAR(64) NOT NULL,
  experience INT CHECK (experience <= age) NOT NULL
);
```

```
CREATE TABLE ship (
  ship_id SERIAL PRIMARY KEY,
  name VARCHAR(64) NOT NULL,
  loading_capacity DECIMAL(4, 2) NOT NULL,
  captain_id INT UNIQUE REFERENCES captain(captain_id) ON DELETE CASCADE,
  radar_id INT REFERENCES radar(radar_id) ON DELETE CASCADE
);
```

```
CREATE TABLE planet (
  planet_id SERIAL PRIMARY KEY,
  name VARCHAR(64) NOT NULL,
  have_atmosphere BOOLEAN,
  galaxy VARCHAR(64) NOT NULL
);
```

```
CREATE TABLE moon (
  moon_id SERIAL PRIMARY KEY,
  name VARCHAR(64) NOT NULL,
  diameter INT CHECK (diameter >0),
  planet_id INT REFERENCES planet(planet_id) ON DELETE CASCADE
);
```

```

CREATE TABLE track (
track_id SERIAL PRIMARY KEY,
length INT CHECK (length > 0),
width INT CHECK (width > 0),
planet_a_id INT REFERENCES planet(planet_id) ON DELETE CASCADE,
planet_b_id INT REFERENCES planet(planet_id) ON DELETE CASCADE,
trajectory_type VARCHAR(64) NOT NULL
);

CREATE TABLE moon_track_relationship (
moon_id INT REFERENCES moon(moon_id) ON DELETE CASCADE,
track_id INT REFERENCES track(track_id) ON DELETE CASCADE,
relationship_type VARCHAR(64) NOT NULL,
PRIMARY KEY (moon_id, track_id)
);

CREATE TABLE ship_track (
flight_date DATE NOT NULL,
ship_id INT REFERENCES ship(ship_id) ON DELETE CASCADE,
track_id INT REFERENCES track(track_id) ON DELETE CASCADE,
PRIMARY KEY (ship_id, flight_date)
);

INSERT INTO radar (model, scope, frequency, impulse_type) VALUES ('CR-12', 12332, 444.5,
'Электрические импульсы');
INSERT INTO radar (model, scope, frequency, impulse_type) VALUES ('CR-55', 72039, 978.8,
'Звуковые волны');

INSERT INTO captain (age, name, surname, experience) VALUES (45, 'Джек', 'Воробей', 20);
INSERT INTO captain (age, name, surname, experience) VALUES (66, 'Дейви', 'Джонс', 35);
INSERT INTO captain (age, name, surname, experience) VALUES (30, 'Уилл', 'Смит', 10);

INSERT INTO ship (name, loading_capacity, captain_id, radar_id) VALUES ('Черная
жемчужина', 35.70, 1, 1);
INSERT INTO ship (name, loading_capacity, captain_id, radar_id) VALUES ('Летучий
голандец', 45.69, 2, 2);

INSERT INTO planet (name, have_atmosphere, galaxy) VALUES ('Юпитер', TRUE, 'Млечный
путь');

INSERT INTO moon (name, diameter, planet_id) VALUES ('Ганимед', 5268, 1);

INSERT INTO track (length, width, planet_a_id, planet_b_id, trajectory_type) VALUES
(17000, 234, 1, 1, 'Крюк');
INSERT INTO track (length, width, planet_a_id, planet_b_id, trajectory_type) VALUES
(70000, 679, 1, 1, 'Прямая');

```

```
INSERT INTO moon_track_relationship (moon_id, track_id, relationship_type) VALUES (1, 1,  
'Маршрут проходит вблизи спутника');  
INSERT INTO moon_track_relationship (moon_id, track_id, relationship_type) VALUES (1, 2,  
'Находятся на большом расстоянии друг от спутника');  
  
INSERT INTO ship_track (flight_date, ship_id, track_id) VALUES ('2025-02-27', 1, 1);  
INSERT INTO ship_track (flight_date, ship_id, track_id) VALUES ('2025-03-01', 2, 2);
```


Заключение

В ходе лабораторной работы мы познакомились с архитектурой “ANSI-SPARK”, научились составлять инфологическую и даталогическую модель сущностей, по которым реализовали базу данных при помощи SQL.