Министерство 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**

Вариант №5744

Группа: P3112

Выполнил: Балин А. А.

Проверила: Лисицина В.В.

Оглавление

[Введение 3](#_Toc131036234)

[Текст задания 4](#_Toc131036235)

[Описание предметной области 5](#_Toc131036236)

[Инфологическая модель 7](#_Toc131036237)

[Даталогическая модель 8](#_Toc131036238)

[Заключение 9](#_Toc131036239)

[Список литературы 10](#_Toc131036240)

# Введение

В данной лабораторной работе я потренируюсь в выделении сущностей из предметной области (текста), создании связей между ними, напишу запросы для создания и заполнения таблиц в PostgreSQL.

# Текст задания

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

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.

Сущности

1. Стержневые:
2. Космический корабль
3. Луны Юпитера
4. Радар
5. Ассоциативные:
6. Импульс энергии – связывает космический корабль и радар, используется для обнаружения лун.
7. Зона луны – связывает луну и некоторую её окрестность, которая может быть исследована кораблём.

Атрибуты и ключи

Корабль:

* id (тип данных: SERIAL PRIMARY KEY)
* name (тип данных: VARCHAR(50))
* x, y, z coordinates (тип данных: BIGINT)

Луны Юпитера:

* id (тип данных: SERIAL PRIMARY KEY)
* name (тип данных: VARCHAR(50))
* diameter (тип данных: INTEGER)
* x, y, z coordinates (тип данных: BIGINT)

Радар:

* id (тип данных: SERIAL PRIMARY KEY)
* signal\_frequency (тип данных: DECIMAL)
* send signal\_frequency (тип данных: DECIMAL)

Зоны лун:

* id (тип данных: SERIAL PRIMARY KEY)
* moon\_id (тип данных: INTEGER)
* radius (тип данных: INTEGER)
* x, y, z center coordinates (тип данных: BIGINT)

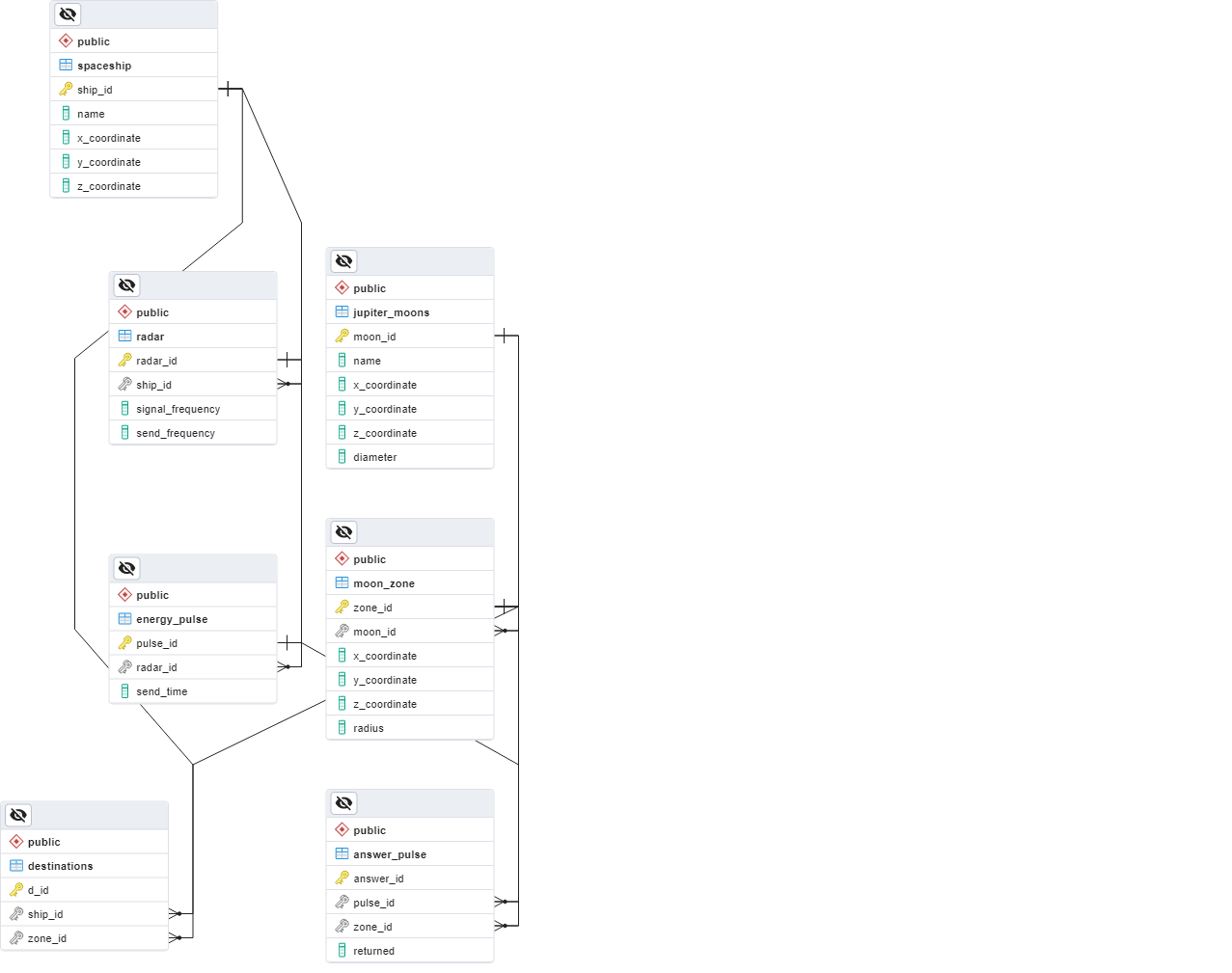
Импульсы энергии:

* id (тип данных: SERIAL PRIMARY KEY)
* sendtime (тип данных: TIMESTAMP)

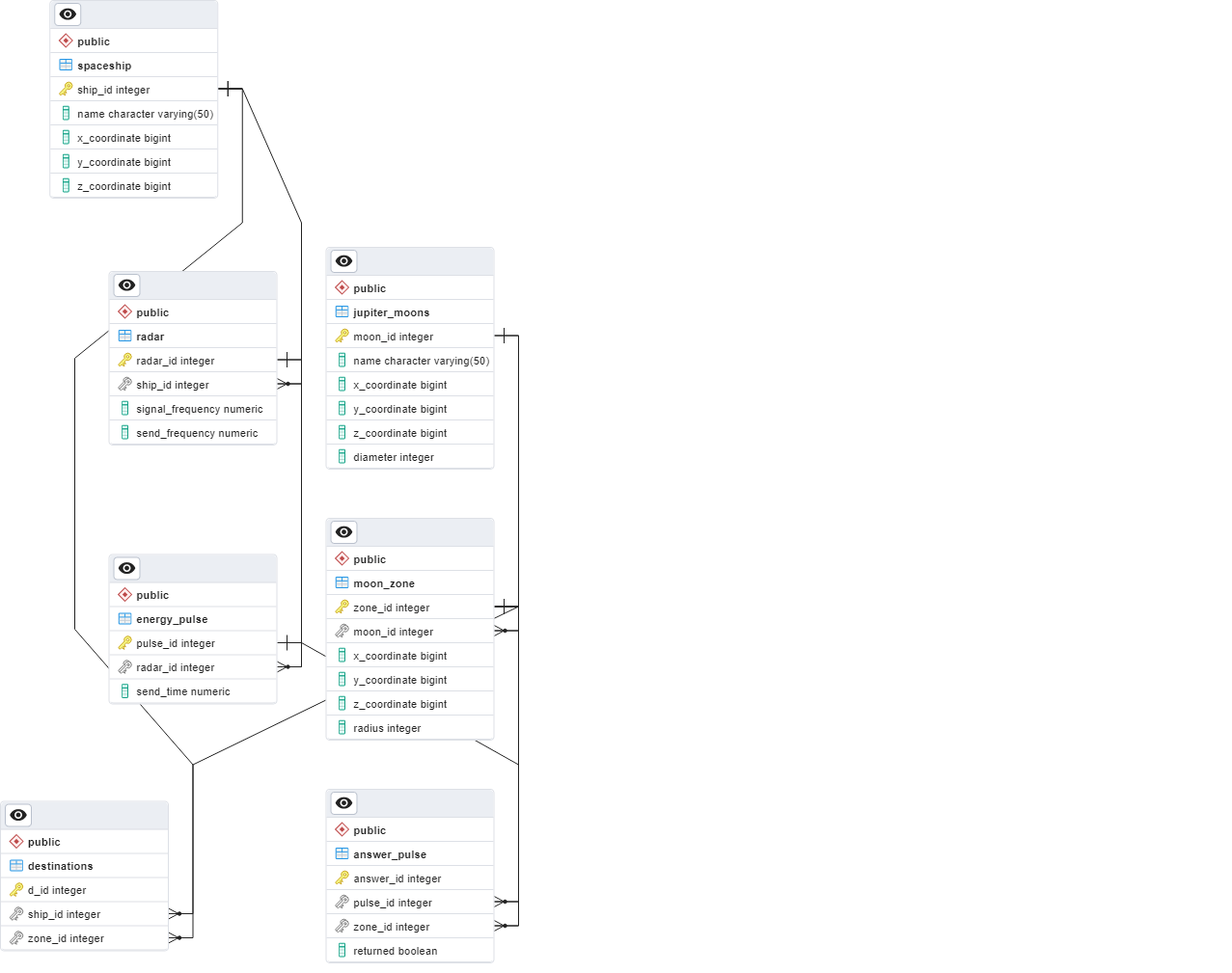
Ограничения целостности

* Атрибуты name в таблицах spaceship и jupiter\_moons не могут быть пустыми
* Атрибуты coordinates в таблицах spaceship, jupiter\_moons и moon\_zone могут принимать только уникальные значения
* Атрибуты send\_datetime в таблице energy\_pulse и center\_coordinates в таблице moon\_zone не могут быть пустыми
* Атрибуты spaceship\_id и radar\_id в таблице energy\_pulse и jupiter\_moon\_id в таблице moon\_zone должны ссылаться на существующие записи в таблицах spaceship, radar и jupiter\_moons соответственно

# Инфологическая модель



# Даталогическая модель



# Код

CREATE TABLE spaceship (

    ship\_id **SERIAL** **PRIMARY KEY**,

    name **VARCHAR**(50) NOT NULL,

    x\_coordinate **BIGINT**,

    y\_coordinate **BIGINT**,

    z\_coordinate **BIGINT**

);

CREATE TABLE radar (

    radar\_id **SERIAL** **PRIMARY KEY**,

    ship\_id **INTEGER** **REFERENCES** spaceship(ship\_id) **ON DELETE** RESTRICT,

    signal\_frequency **DECIMAL**,

    send\_frequency **DECIMAL**

);

CREATE TABLE energy\_pulse (

    pulse\_id **SERIAL** **PRIMARY KEY**,

    radar\_id **INTEGER** **DEFAULT** -1 **REFERENCES** radar(radar\_id)  **ON DELETE** SET **DEFAULT**,

    send\_time **DECIMAL**

);

CREATE TABLE cosmic\_type(type\_id **SERIAL** **PRIMARY KEY**, name **VARCHAR**(50));

CREATE TABLE cosmic\_body(

    obj\_id **SERIAL** **PRIMARY KEY**,

    name **VARCHAR**(50) NOT NULL,

    type\_id **INTEGER** **REFERENCES** cosmic\_type(type\_id) ON UPDATE RESTRICT,

    x\_coordinate **BIGINT**,

    y\_coordinate **BIGINT**,

    z\_coordinate **BIGINT**,

    diameter **INTEGER**

);

CREATE TABLE moon\_zone (

    zone\_id **SERIAL** **PRIMARY KEY**,

    moon\_id **INTEGER** **REFERENCES** cosmic\_body(obj\_id) **ON DELETE** SET NULL,

    x\_coordinate **BIGINT**,

    y\_coordinate **BIGINT**,

    z\_coordinate **BIGINT**,

    radius **INTEGER**

);

CREATE TABLE destination (

    d\_id **SERIAL** **PRIMARY KEY**,

    ship\_id **INTEGER** **REFERENCES** spaceship(ship\_id) ON UPDATE RESTRICT **ON DELETE** SET NULL,

    zone\_id **INTEGER** **REFERENCES** moon\_zone(zone\_id) **ON DELETE** SET NULL

);

CREATE TABLE answer\_pulse(

    answer\_id **SERIAL** **PRIMARY KEY**,

    pulse\_id **INTEGER** **REFERENCES** energy\_pulse(pulse\_id) **ON DELETE** NO ACTION,

    zone\_id **INTEGER** **REFERENCES** moon\_zone(zone\_id) **ON DELETE** SET NULL,

    returned **BOOLEAN**

);

INSERT INTO

    spaceship (name, x\_coordinate, y\_coordinate, z\_coordinate)

VALUES

    ('Discovery One', 1000000, 2000000, 3000000);

INSERT INTO

    spaceship (name, x\_coordinate, y\_coordinate, z\_coordinate)

VALUES

    ('Event Horizon', 500000, 1500000, 2500000);

INSERT INTO

    radar (ship\_id, signal\_frequency, send\_frequency)

VALUES

    (1, 5000.00, 100.00);

INSERT INTO

    radar (ship\_id, signal\_frequency, send\_frequency)

VALUES

    (2, 10000.00, 200.00);

INSERT INTO

    energy\_pulse (radar\_id, send\_time)

VALUES

    (1, 0.50);

INSERT INTO

    energy\_pulse (radar\_id, send\_time)

VALUES

    (2, 1.20);

INSERT INTO

    cosmic\_type (name)

VALUES

    ('Moon of Jupiter');

INSERT INTO

    cosmic\_type (name)

VALUES

    ('Asteroid');

INSERT INTO

    cosmic\_body (

        name,

        type\_id,

        x\_coordinate,

        y\_coordinate,

        z\_coordinate,

        diameter

    )

VALUES

    ('Io', 1, 20000000, 30000000, 40000000, 4000);

INSERT INTO

    cosmic\_body (

        name,

        type\_id,

        x\_coordinate,

        y\_coordinate,

        z\_coordinate,

        diameter

    )

VALUES

    ('Europa', 1, 25000000, 32000000, 41000000, 6000);

INSERT INTO

    moon\_zone (

        moon\_id,

        x\_coordinate,

        y\_coordinate,

        z\_coordinate,

        radius

    )

VALUES

    (1, 21000000, 31000000, 42000000, 20000);

INSERT INTO

    moon\_zone (

        moon\_id,

        x\_coordinate,

        y\_coordinate,

        z\_coordinate,

        radius

    )

VALUES

    (2, 26000000, 33000000, 43000000, 30000);

INSERT INTO

    destination (ship\_id, zone\_id)

VALUES

    (1, 1);

INSERT INTO

    destination (ship\_id, zone\_id)

VALUES

    (2, 2);

INSERT INTO

    answer\_pulse (pulse\_id, zone\_id, returned)

VALUES

    (1, 1, false);

INSERT INTO

    answer\_pulse (pulse\_id, zone\_id, returned)

VALUES

    (2, 2, true);

# Заключение

Я научился строить инфологические и даталогические диаграммы, разобрался в различиях между типами данных PostgreSQL.

# Список литературы

**PostgreSQL 9.6.24 Documentation** [В Интернете]. - https://www.postgresql.org/docs/9.6/.