

Университет ИТМО, факультет ПИиКТ

Лабораторная работа №1

Вариант 852

Дисциплина: Информационные системы и базы данных

Выполнили: Чангалиди Антон

Группа: Р33113

Преподаватель:  
Гаврилов Антон Валерьевич

г. Санкт-Петербург

2020 г.

# ЛАБОРАТОРНАЯ РАБОТА № 1

## Задание

- На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
- Составить инфологическую модель.
- Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
- Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
- Заполнить созданные таблицы тестовыми данными.

Словно лосось, преодолевающий водопад, он за считанные секунды поднялся по электрической реке, соединяющей Ио с Юпитером, и достиг корабля, который принес его сюда с родной планеты. Тот казался карликом рядом с произведением технической мысли великой цивилизации.

## Описание предметной области

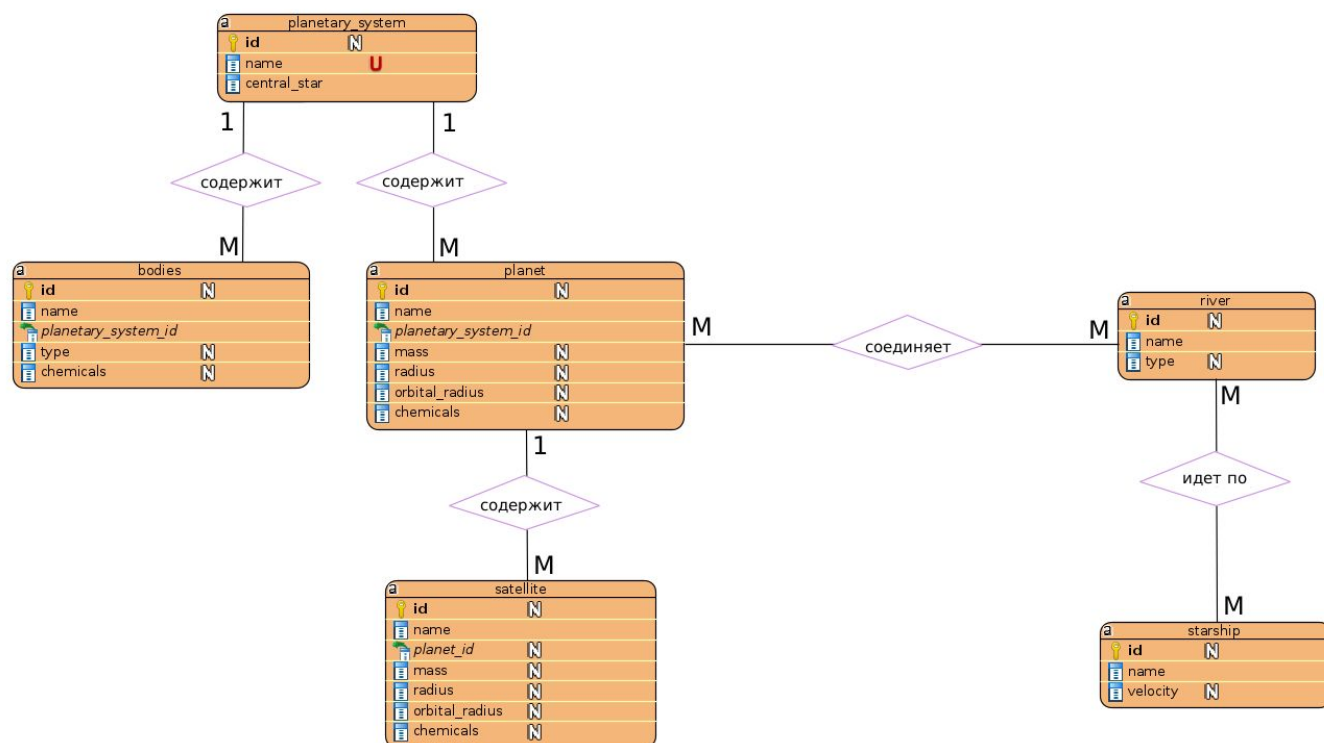
Во вселенной существует много разных ПЛАНЕТАРНЫХ СИСТЕМ, у которых есть одна центральная звезда, ПЛАНЕТЫ (у которых есть СПУТНИКИ), а также другие объекты (астероиды, метеороиды, кометы и космическая пыль). У каждой из планет есть физические характеристики (масса, радиус, радиус орбиты) и химические (состоит из камня/газа).

Планеты (как внутри одной системы, так и между разными системами) соединены МАГИЧЕСКИМИ РЕКАМИ, которые могут быть разной природы. По этим рекам курсируют быстрые и не очень КОСМИЧЕСКИЕ КОРАБЛИ.

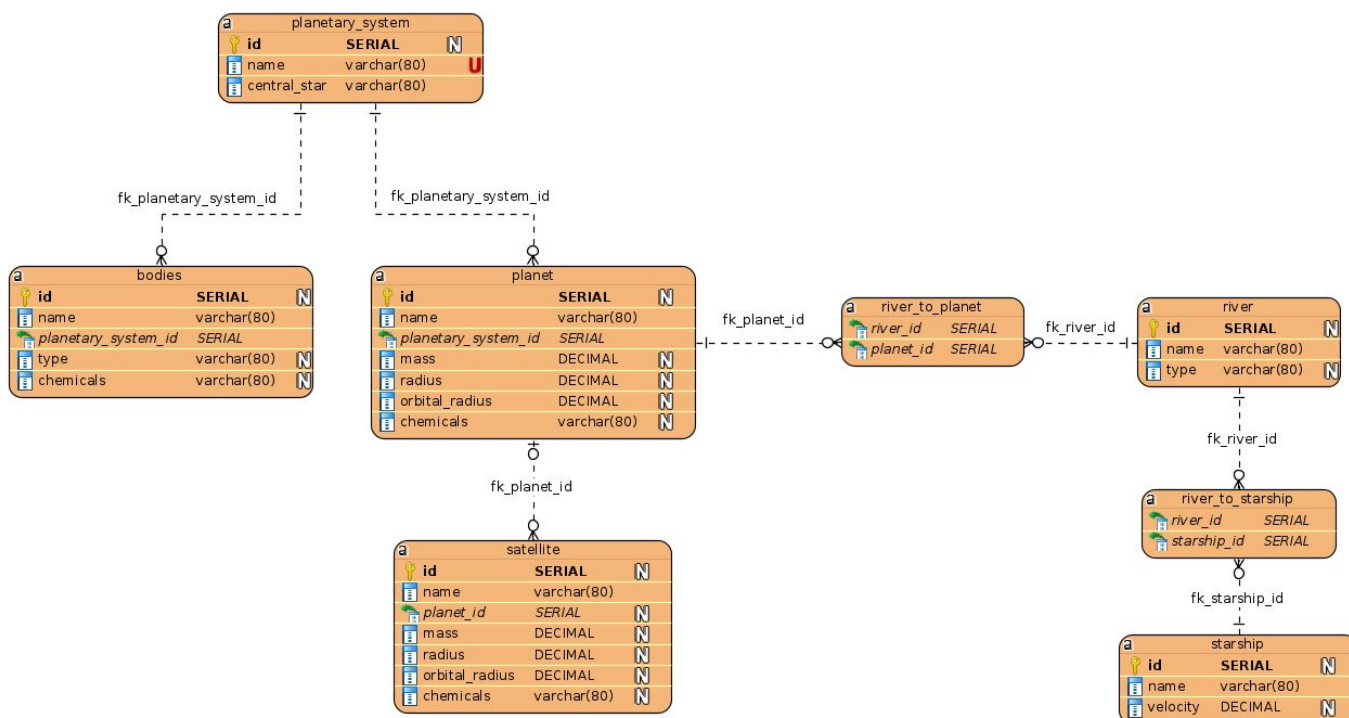
## Классификация сущностей:

- Планетарная система (Стержневая сущность)
- Планета (Стержневая сущность)
- Спутник (Стержневая сущность)
- (Другие) космические тела (Стержневая сущность)
- Реки (Стержневая сущность)
- Корабли (Стержневая сущность)
- “корабли\_реки” (Ассоциативная сущность)
- “планеты\_реки” (Ассоциативная сущность)

# Инфологическая модель



# Даталогическая модель



## Создание таблиц:

```
CREATE TABLE planetary_system
(
    id SERIAL PRIMARY KEY,
    name VARCHAR(80) UNIQUE NOT NULL,
    central_star VARCHAR(80) NOT NULL
)
```

```
);
```

```
CREATE TABLE planet
```

```
(
    id                SERIAL PRIMARY KEY,
    name              VARCHAR(80)                NOT NULL,
    planetary_system_id SERIAL
        CONSTRAINT fk_planetary_system_id REFERENCES planetary_system (id) NOT NULL,
    mass              DECIMAL CHECK ( mass >= 0 ),
    radius            DECIMAL CHECK ( radius >= 0 ),
    orbital_radius    DECIMAL CHECK ( orbital_radius >= 0 ),
    chemicals         VARCHAR(80)

);
```

```
CREATE TABLE satellite
```

```
(
    id                SERIAL PRIMARY KEY,
    name              VARCHAR(80) NOT NULL,
    planet_id         SERIAL
        CONSTRAINT fk_planet_id REFERENCES planet (id),
    mass              DECIMAL CHECK ( mass >= 0 ),
    radius            DECIMAL CHECK ( radius >= 0 ),
    orbital_radius    DECIMAL CHECK ( orbital_radius >= 0 ),
    chemicals         VARCHAR(80)

);
```

```
CREATE TABLE bodies
```

```
(
    id                SERIAL PRIMARY KEY,
    name              VARCHAR(80)                NOT NULL,
    planetary_system_id SERIAL
        CONSTRAINT fk_planetary_system_id REFERENCES planetary_system (id) NOT NULL,
    type              VARCHAR(80),
    chemicals         VARCHAR(80)

);
```

```
CREATE TABLE river
```

```
(
    id    SERIAL PRIMARY KEY,
    name  VARCHAR(80) NOT NULL,
    type  VARCHAR(80)

);
```

```
CREATE TABLE river_to_planet
```

```
(
```

```

    river_id SERIAL
        CONSTRAINT fk_river_id REFERENCES river (id) NOT NULL,
    planet_id SERIAL
        CONSTRAINT fk_planet_id REFERENCES planet (id) NOT NULL
);

CREATE TABLE starship
(
    id SERIAL PRIMARY KEY,
    name VARCHAR(80) NOT NULL,
    velocity DECIMAL CHECK ( velocity >= 0 )
);

CREATE TABLE river_to_starship
(
    river_id SERIAL
        CONSTRAINT fk_river_id REFERENCES river (id) NOT NULL,
    starship_id SERIAL
        CONSTRAINT fk_starship_id REFERENCES starship (id) NOT NULL
);

```

## Заполнение таблиц:

```

INSERT INTO planetary_system (id, name, central_star) VALUES (0, 'Солнечная система',
'Sолнце');
INSERT INTO planetary_system (id, name, central_star) VALUES (1, 'Альфас системс', 'Альфа
центавра');
INSERT INTO bodies (id, name, planetary_system_id, type, chemicals) VALUES (0, 'Церера',
0, 'карликовая планета', NULL);
INSERT INTO bodies (id, name, planetary_system_id, type, chemicals) VALUES (1, 'Пояс
Кеплера', 0, 'Область', NULL);
INSERT INTO planet (id, name, planetary_system_id, mass, radius, orbital_radius,
chemicals) VALUES (0, 'Земля', 0, 5.972E24, 6000, NULL, 'КАМЕНЬ');
INSERT INTO planet (id, name, planetary_system_id, mass, radius, orbital_radius,
chemicals) VALUES (1, 'Сатурн', 0, NULL, NULL, NULL, 'ГАЗ');
INSERT INTO planet (id, name, planetary_system_id, mass, radius, orbital_radius,
chemicals) VALUES (2, 'Уникус', 1, NULL, NULL, NULL, 'ЭФИР');
INSERT INTO satellite (id, name, planet_id) VALUES (0, 'ЛУНА', 1);
INSERT INTO river (id, name, type) VALUES (0, 'КРЭЙЗИ', 'Электрическая');
INSERT INTO river (id, name, type) VALUES (1, 'МЕГА', 'Водяная');
INSERT INTO river_to_planet (river_id, planet_id) VALUES (0, 0);
INSERT INTO river_to_planet (river_id, planet_id) VALUES (0, 1);
INSERT INTO river_to_planet (river_id, planet_id) VALUES (1, 1);
INSERT INTO river_to_planet (river_id, planet_id) VALUES (1, 2);
INSERT INTO starship (id, name, velocity) VALUES (0, 'Карл Маркс', 10000);
INSERT INTO starship (id, name, velocity) VALUES (1, 'Викинг', 15000);
INSERT INTO starship (id, name, velocity) VALUES (2, 'Принцесса Виктория', 5000);

```

```
INSERT INTO river_to_starship (river_id, starship_id) VALUES (0, 1);  
INSERT INTO river_to_starship (river_id, starship_id) VALUES (0, 2);  
INSERT INTO river_to_starship (river_id, starship_id) VALUES (1, 0);
```

## Выводы

Интересная лабораторная работа - помогла закрепить знания и умения, сделанные во время подготовки к курсачу. Сделать почти три пункта курсача за одну лабу - easy:)

Что же входило в эту лабу? Проработка от художественного текста до создания и заполнения несложной БД, используя PostgreSQL.