

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №1

По Базам Данных

Вариант 1965

Выполнил:

Ларионов Владислав Васильевич

Группа Р3109

Практик:

Воронина Дарья Сергеевна

Санкт-Петербург 2024

## Содержание

<b>Задание .....</b>	<b>3</b>
<b>Выполнение задания .....</b>	<b>5</b>
2.1 Список сущностей и их классификация .....	5
2.2 Инфологическая модель .....	6
2.3 Даталогическая модель .....	7
2.4 Реализация в PostgreSQL .....	8
<b>Вывод: .....</b>	<b>11</b>

## Задание

Для выполнения лабораторной работы №1 необходимо:

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.

Для создания объектов базы данных у каждого студента есть своя схема.

Название схемы соответствует имени пользователя в базе studs (sXXXXXX).

Команда для подключения к базе studs:

```
psql -h pg -d studs
```

Каждый студент должен использовать свою схему при работе над лабораторной работой №1 (а также в рамках выполнения 2, 3 и 4 этапа курсовой работы).

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Описание предметной области.
3. Список сущностей и их классификацию (стержневая, ассоциация, характеристика).
4. Инфологическая модель (ER-диаграмма в расширенном виде - с атрибутами, ключами...).
5. Даталогическая модель (должна содержать типы атрибутов, вспомогательные таблицы для отображения связей "многие-ко-многим").
6. Реализация даталогической модели на SQL.
7. Выводы по работе.

### **Предметная область:**

Лишь изредка вихри и порывы ураганного ветра нарушали строй облаков, протянувшихся параллельными рядами от горизонта до горизонта. Время от времени восходящие потоки более светлого газа раскрывали их пелену, открывая вид на темный край гигантской воронки, воздушного Мальстрема, низвергающегося в бездонные глубины Юпитера.

### **Описание предметной области:**

Действия происходят на планете, на ней существуют определенные явления (несколько явлений могут существовать на одной планете). Разные явления вызывают последствия, при этом несколько природных явлений могут вызвать одно и то же последствие, и одно природное явление может вызвать разные последствия.

# Выполнение задания

## 2.1 Список сущностей и их классификация

### **Стержни:**

phenomena (явления), consequences (последствия), natural\_scene (экшн-сцена природных явлений)

### **Ассоциации:**

phenomena\_consequences (последствия, вызываемые явлениями),  
phenomena\_in\_scene (явления в экшн-сцене)

### **Характеристики:**

planet (планета)

## 2.2 Инфологическая модель

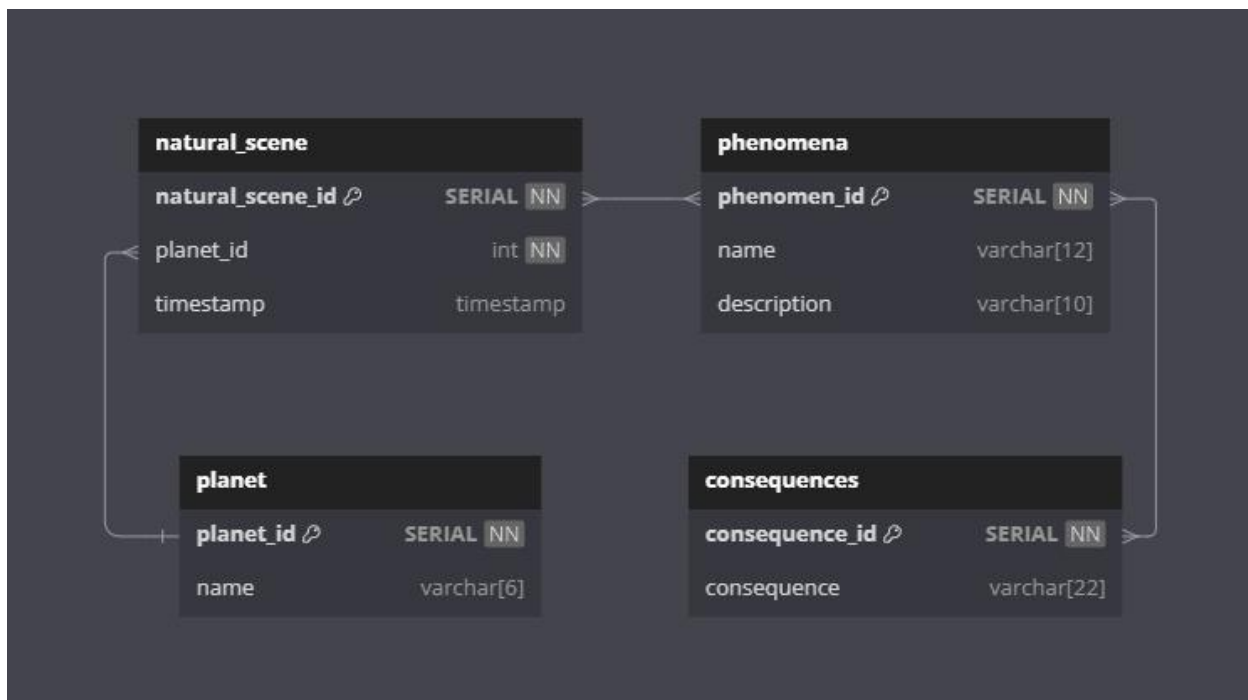


Рис. 1 – инфологическая модель

## 2.3 Даталогическая модель

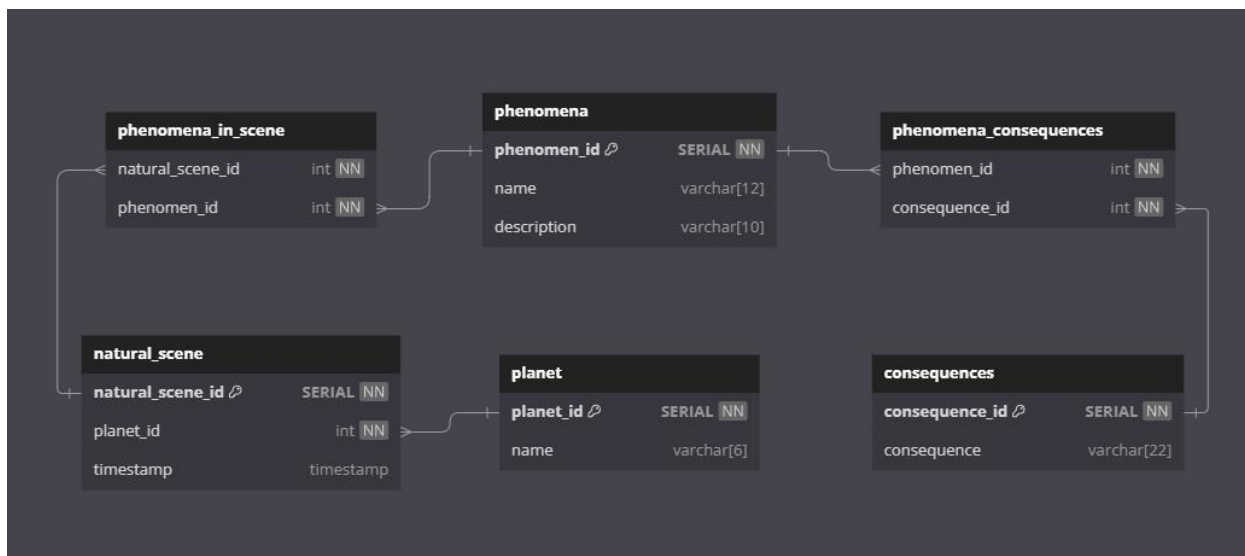


Рис. 2 – даталогическая модель

## 2.4 Реализация в PostgreSQL

```
DROP TABLE IF EXISTS phenomena_in_scene;
```

```
DROP TABLE IF EXISTS phenomena_consequences;
```

```
DROP TABLE IF EXISTS natural_scene;
```

```
DROP TABLE IF EXISTS phenomena;
```

```
DROP TABLE IF EXISTS consequences;
```

```
DROP TABLE IF EXISTS planet;
```

```
CREATE TABLE IF NOT EXISTS planet (
```

```
    planet_id SERIAL PRIMARY KEY,
```

```
    name VARCHAR(6) NOT NULL
```

```
);
```

```
CREATE TABLE IF NOT EXISTS natural_scene (
```

```
    natural_scene_id SERIAL PRIMARY KEY,
```

```
    planet_id INT NOT NULL,
```

```
    timestamp TIMESTAMP NOT NULL,
```

```
    FOREIGN KEY (planet_id) REFERENCES planet(planet_id) ON DELETE CASCADE
```

```
);
```

```
CREATE TABLE IF NOT EXISTS phenomena (
```

```
    phenomen_id SERIAL PRIMARY KEY,
```

```
    name VARCHAR(12) NOT NULL,
```

```
    description VARCHAR(10)
```

```
);
```

```
CREATE TABLE IF NOT EXISTS consequences (
```

```
    consequence_id SERIAL PRIMARY KEY,
```

```
    consequence VARCHAR(22) NOT NULL
```



);

```
CREATE TABLE IF NOT EXISTS phenomena_consequences (  
    phenomen_id INT NOT NULL,  
    consequence_id INT NOT NULL,  
    PRIMARY KEY (phenomen_id, consequence_id),  
    FOREIGN KEY (phenomen_id) REFERENCES phenomena(phenomen_id) ON DELETE  
    CASCADE,  
    FOREIGN KEY (consequence_id) REFERENCES consequences(consequence_id) ON  
    DELETE CASCADE  
);
```

```
CREATE TABLE IF NOT EXISTS phenomena_in_scene (  
    natural_scene_id INT NOT NULL,  
    phenomen_id INT NOT NULL,  
    PRIMARY KEY (natural_scene_id, phenomen_id),  
    FOREIGN KEY (natural_scene_id) REFERENCES natural_scene(natural_scene_id) ON  
    DELETE CASCADE,  
    FOREIGN KEY (phenomen_id) REFERENCES phenomena(phenomen_id) ON DELETE  
    CASCADE  
);
```

```
INSERT INTO planet (name) VALUES  
( 'Юпитер');
```

```
INSERT INTO phenomena (name, description) VALUES  
( 'вихри', NULL),  
( 'порывы ветра', 'ураганные'),  
( 'потoki газа', 'восходящие'),  
( 'воронка', 'гигантская');
```

```
INSERT INTO consequences (consequence) VALUES
```

```
('нарушать строй облаков'),
```

```
('раскрывать пелену'),
```

```
('открывать вид'),
```

```
('низвергаться в глубины');
```

```
INSERT INTO natural_scene (planet_id, timestamp) VALUES
```

```
(1, '2025-03-03 12:00:00'),
```

```
(1, '2025-03-03 14:00:00');
```

```
INSERT INTO phenomena_consequences (phenomen_id, consequence_id) VALUES
```

```
(1, 1),
```

```
(2, 2),
```

```
(3, 3),
```

```
(4, 4);
```

```
INSERT INTO phenomena_in_scene (natural_scene_id, phenomen_id) VALUES
```

```
(1, 1),
```

```
(1, 2),
```

```
(2, 3),
```

```
(2, 4);
```

```
SELECT * from natural_scene;
```

```
SELECT * from planet;
```

```
SELECT * from phenomena_in_scene;
```

```
SELECT * from phenomena;
```

```
SELECT * from phenomena_consequences;
```

```
SELECT * from consequences;
```

## **Вывод:**

В ходе выполнения данной лабораторной работы я научился создавать инфологические и даталогические модели, научился создавать базы данных в PostgreSQL.