Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики»

**ЛАБОРАТОРНАЯ РАБОТА № 2,3**

по дисциплине

“ Проектирование и реализация баз данных”

*Выполнил:*

Мохаджер Алиреза

Джафари Хоссаин

Студент группы K3240

*Преподаватель:*

Белов Александр Олегович

Говорова Марина Михайловна

Санкт-Петербург, 2025

# Цель работы: овладеть практическими навыками установки СУБД PostgreSQL и создания базы данных в pgadmin 4.

# 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

То же самое проделали и искусственный спутник Марса "Орбитер М-15", обегавший вокруг Марса дважды в сутки, и космический зонд, поднимавшийся в пространства, лежащие над плоскостью эклиптики, и даже искусственная комета э 5, уносившаяся в ледяные дали за Плутоном по орбите, до самой удаленной точки которой ей не долететь и за тысячу лет. Все их приборы зарегистрировали необычную вспышку энергии, и все они установленным порядком автоматически передали запись этих сигналов в хранилища информации на далекой Земле.

# Список сущностей

**Space Device:** Искусственный спутник или зонд, функционирующий в космосе и выполняющий задачи, такие как наблюдение, регистрация вспышек энергии и передача данных на Землю.

**Planet:** Планета, вокруг которой вращаются космические аппараты.

**Model:** Конкретная модель космического аппарата, определяемая типом и производителем.

**Productor:** Организация или страна, производившая модель космического аппарата.

**Owner:** Организация или страна, которой принадлежит космический аппарат.

**Launch:** Факт запуска космического аппарата: включает дату запуска, дату возвращения (если есть), координаты и информацию об успешности запуска.

**Recorder:** Устройство, установленное на космическом аппарате, предназначенное для регистрации данных (например, вспышек энергии).

**Burst:** Необычное энергетическое событие в космосе, зарегистрированное приборами (например, "вспышка энергии", упомянутая в тексте).

**Record Information of Burst:** Данные, зарегистрированные о вспышке с указанием записывающего устройства, времени и объема данных.

**Data Storage:** Устройство или система на Земле, куда поступают и хранятся данные, переданные с космических аппаратов.

**Saving Information in Storage:** Процесс или факт сохранения конкретной информации (включая время и объем) в хранилище данных.

Изображение выглядит как текст, диаграмма, План, Технический чертеж

Контент, сгенерированный ИИ, может содержать ошибки.Инфологическая модель

# Изображение выглядит как текст, диаграмма, План, Параллельный Контент, сгенерированный ИИ, может содержать ошибки.Даталогическая модель

# 

# Создание модели в PostgreSQL

-- Удаление таблиц, если существуют, с учетом зависимостей

DROP TABLE IF EXISTS record\_information\_of\_burst CASCADE;

DROP TABLE IF EXISTS recorder CASCADE;

DROP TABLE IF EXISTS saving\_information\_in\_storage CASCADE;

DROP TABLE IF EXISTS burst CASCADE;

DROP TABLE IF EXISTS launch CASCADE;

DROP TABLE IF EXISTS space\_device CASCADE;

DROP TABLE IF EXISTS data\_storage CASCADE;

DROP TABLE IF EXISTS model CASCADE;

DROP TABLE IF EXISTS planet CASCADE;

DROP TABLE IF EXISTS owner CASCADE;

DROP TABLE IF EXISTS productor CASCADE;

-- Таблица производителей моделей

CREATE TABLE productor (

id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

country VARCHAR(255) NOT NULL

);

-- Таблица моделей космических аппаратов

CREATE TABLE model (

id SERIAL PRIMARY KEY,

id\_productor INT NOT NULL REFERENCES productor(id) ON DELETE CASCADE,

name VARCHAR(255) NOT NULL,

type VARCHAR(100) NOT NULL CHECK (type IN ('satellite', 'probe', 'telescope', 'rover', 'station'))

);

-- Таблица владельцев аппаратов

CREATE TABLE owner (

id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL,

country VARCHAR(255) NOT NULL

);

-- Таблица планет

CREATE TABLE planet (

id SERIAL PRIMARY KEY,

name VARCHAR(255) NOT NULL UNIQUE,

coordinate VARCHAR(255) NOT NULL CHECK (coordinate ~ '^[A-Za-z0-9\-]+$')

);

-- Таблица космических аппаратов

CREATE TABLE space\_device (

id SERIAL PRIMARY KEY,

id\_model INT NOT NULL REFERENCES model(id) ON DELETE CASCADE,

id\_planet INT REFERENCES planet(id) ON DELETE SET NULL,

id\_owner INT NOT NULL REFERENCES owner(id) ON DELETE CASCADE,

is\_active BOOLEAN NOT NULL DEFAULT FALSE

);

-- Таблица запусков

CREATE TABLE launch (

id SERIAL PRIMARY KEY,

id\_space\_device INT NOT NULL REFERENCES space\_device(id) ON DELETE CASCADE,

coordinate VARCHAR(255) NOT NULL,

is\_successful BOOLEAN,

date\_launch DATE NOT NULL CHECK (date\_launch <= CURRENT\_DATE),

date\_return DATE CHECK (date\_return IS NULL OR date\_return >= date\_launch)

);

-- Таблица хранилищ данных

CREATE TABLE data\_storage (

id SERIAL PRIMARY KEY,

coordinate VARCHAR(255) NOT NULL CHECK (coordinate ~ '^[A-Za-z0-9\-]+$'),

volume INT NOT NULL CHECK (volume > 0),

is\_full BOOLEAN NOT NULL DEFAULT FALSE

);

-- Таблица вспышек

CREATE TABLE burst (

id SERIAL PRIMARY KEY,

type VARCHAR(100) NOT NULL CHECK (type IN ('gamma', 'x-ray', 'radio', 'optical', 'particle')),

date TIMESTAMP NOT NULL CHECK (date <= CURRENT\_TIMESTAMP),

coordinate VARCHAR(255) NOT NULL

);

-- Таблица записывающих устройств

CREATE TABLE recorder (

id SERIAL PRIMARY KEY,

id\_space\_device INT NOT NULL REFERENCES space\_device(id) ON DELETE CASCADE,

id\_model INT NOT NULL REFERENCES model(id) ON DELETE CASCADE,

date\_production DATE NOT NULL CHECK (date\_production <= CURRENT\_DATE)

);

-- Таблица записи информации в хранилище

CREATE TABLE saving\_information\_in\_storage (

id SERIAL PRIMARY KEY,

id\_storage INT NOT NULL REFERENCES data\_storage(id) ON DELETE CASCADE,

id\_space\_device INT NOT NULL REFERENCES space\_device(id) ON DELETE CASCADE,

start\_time TIMESTAMP NOT NULL CHECK (start\_time <= CURRENT\_TIMESTAMP),

end\_time TIMESTAMP CHECK (end\_time IS NULL OR end\_time >= start\_time),

data\_volume INT NOT NULL CHECK (data\_volume > 0)

);

-- Таблица записи вспышек

CREATE TABLE record\_information\_of\_burst (

id SERIAL PRIMARY KEY,

id\_burst INT NOT NULL REFERENCES burst(id) ON DELETE CASCADE,

id\_recorder INT NOT NULL REFERENCES recorder(id) ON DELETE CASCADE,

id\_saving\_information INT NOT NULL REFERENCES saving\_information\_in\_storage(id) ON DELETE CASCADE,

recording\_date TIMESTAMP NOT NULL CHECK (recording\_date <= CURRENT\_TIMESTAMP),

data\_volume INT NOT NULL CHECK (data\_volume > 0)

);

------------------------------

-- Вставка данных (INSERT)

------------------------------

INSERT INTO productor (name, country) VALUES

('SpaceX', 'USA'),

('NASA', 'USA'),

('ESA', 'Europe'),

('Roscosmos', 'Russia');

INSERT INTO model (id\_productor, name, type) VALUES

(1, 'Starlink', 'satellite'),

(2, 'Voyager', 'probe'),

(3, 'Hubble', 'telescope'),

(4, 'Lunokhod', 'rover');

INSERT INTO owner (name, country) VALUES

('US Gov', 'USA'),

('EU Space', 'Europe'),

('Russian Fed', 'Russia'),

('Private Corp', 'USA');

INSERT INTO planet (name, coordinate) VALUES

('Earth', 'SOL-3'),

('Mars', 'SOL-4'),

('Moon', 'SOL-3-1'),

('Venus', 'SOL-2');

INSERT INTO space\_device (id\_model, id\_planet, id\_owner, is\_active) VALUES

(1, 1, 1, TRUE),

(2, NULL, 1, TRUE),

(3, NULL, 2, TRUE),

(4, 3, 3, FALSE);

INSERT INTO launch (id\_space\_device, coordinate, is\_successful, date\_launch, date\_return) VALUES

(1, '28.5618N-80.5774W', TRUE, '2020-01-01', NULL),

(2, '28.5618N-80.5774W', TRUE, '1977-09-05', NULL),

(3, '5.2397N-52.7688W', TRUE, '1990-04-24', NULL),

(4, '45.9650N-63.3050E', TRUE, '1970-11-10', '1970-11-17');

INSERT INTO data\_storage (coordinate, volume, is\_full) VALUES

('SOL-3-001', 1000, FALSE),

('SOL-3-002', 2000, TRUE),

('SOL-4-001', 500, FALSE),

('SOL-3-003', 1500, FALSE);

INSERT INTO burst (type, date, coordinate) VALUES

('gamma', '2022-01-01 12:00:00', 'RA14h20m'),

('x-ray', '2022-02-01 12:00:00', 'RA18h45m'),

('radio', '2022-03-01 12:00:00', 'RA22h10m'),

('optical', '2022-04-01 12:00:00', 'RA5h30m');

INSERT INTO recorder (id\_space\_device, id\_model, date\_production) VALUES

(2, 2, '1976-01-01'),

(3, 3, '1989-01-01'),

(1, 1, '2019-01-01'),

(4, 4, '1970-01-01');

INSERT INTO saving\_information\_in\_storage (id\_storage, id\_space\_device, start\_time, end\_time, data\_volume) VALUES

(1, 2, '2022-01-01 12:01:00', '2022-01-01 12:05:00', 100),

(2, 3, '2022-02-01 12:01:00', '2022-02-01 12:05:00', 200),

(3, 1, '2022-03-01 12:01:00', '2022-03-01 12:05:00', 150),

(4, 4, '2022-04-01 12:01:00', '2022-04-01 12:05:00', 180);

INSERT INTO record\_information\_of\_burst (id\_burst, id\_recorder, id\_saving\_information, recording\_date, data\_volume) VALUES

(1, 1, 1, '2022-01-01 12:01:30', 50),

(2, 2, 2, '2022-02-01 12:01:30', 80),

(3, 3, 3, '2022-03-01 12:01:30', 60),

(4, 4, 4, '2022-04-01 12:01:30', 70);

**Вывод**

Я познакомился с базой данных PostgreSQL. Освежил свои знания языка SQL.