УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Информационные системы и базы данных»

Лабораторная работа №1

Студент

Бобрусь А.В.

P33091

Преподаватель

Харитонова А.Е.

Санкт-Петербург, 2023 г.

Задание

Лабораторная работа #1

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.

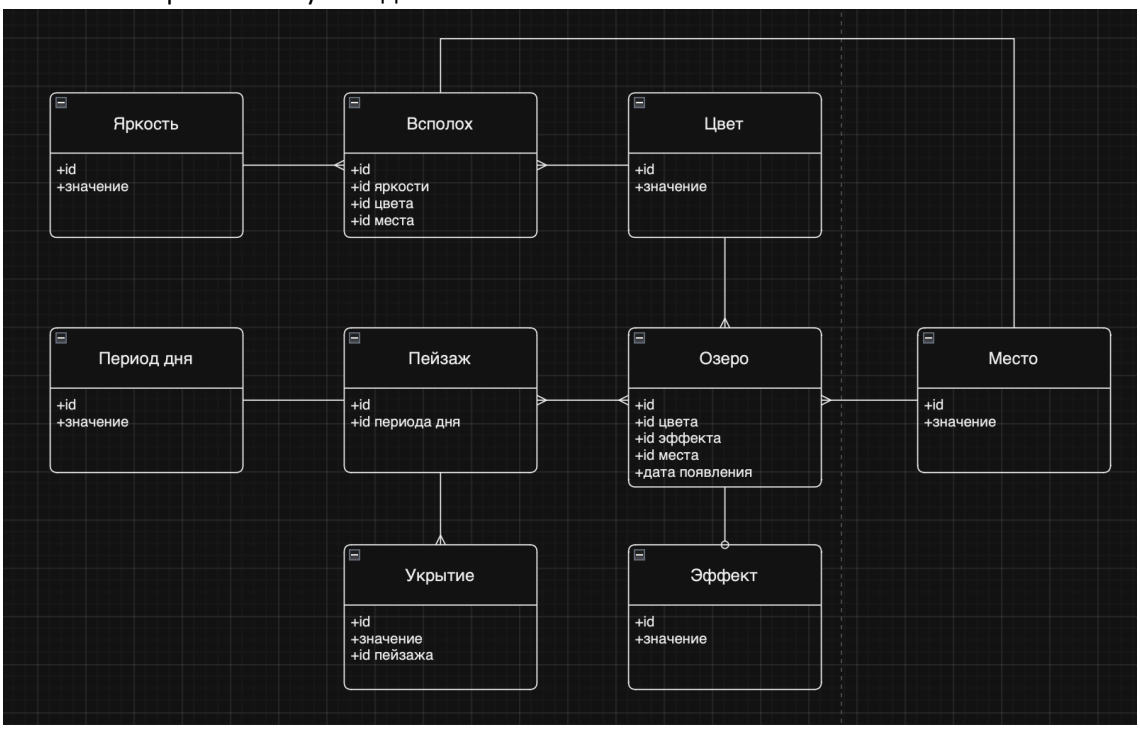
Вариант

Возле экватора вскрылось мерцающее синее озеро; всего несколько часов назад его не было и в помине. Вдоль кромки озера плясали яркие желтые всполохи - верный признак раскаленного натрия. Ночной пейзаж прикрывала призрачная паутина плазменного разряда - одного из обычных для Ио полярных сияний.

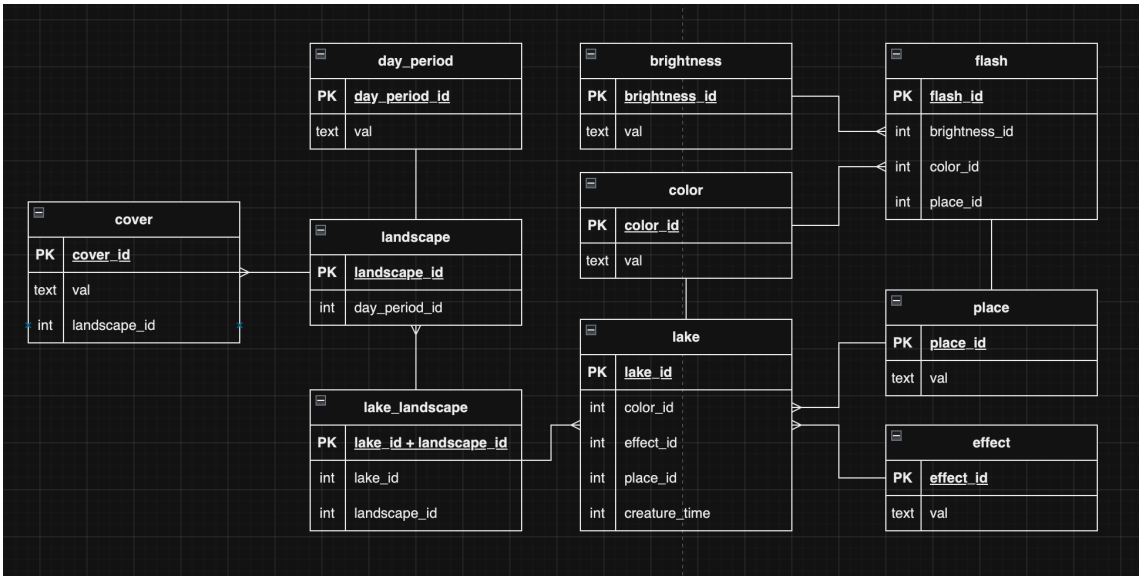
Ход работы

1) Исходя из варианта задания можем выделить следующие сущности.
    - Стержневые: всполох (имеет атрибуты id, id яркости, id цвета, id места), период дня (имеет атрибуты id, значение), пейзаж (имеет атрибуты id, id периода дня), озеро (имеет атрибуты id, id цвета, id эффекта, id места, дата), место (имеет атрибуты id, значение), укрытие (имеет атрибуты id, значение, id пейзажа).
    - Характеристические: яркость (имеет атрибуты id, значение), цвет (имеет атрибуты id, значение), эффект (имеет атрибуты id, значение).
    - Ассоциативные: пейзаж-озеро (синтетическая сущность для выражения отношения многие-ко-многим)

2) Составим инфологическую модель:



3) Составим даталогическую модель:



4) Создаем таблицы

```
DROP TABLE IF EXISTS brightness CASCADE;
DROP TABLE IF EXISTS color CASCADE;
DROP TABLE IF EXISTS place CASCADE;
DROP TABLE IF EXISTS flash CASCADE;
DROP TABLE IF EXISTS day_period CASCADE;
DROP TABLE IF EXISTS landscape CASCADE;
DROP TABLE IF EXISTS cover CASCADE;
DROP TABLE IF EXISTS effect CASCADE;
DROP TABLE IF EXISTS lake CASCADE;
DROP TABLE IF EXISTS lake_landscape CASCADE;


CREATE TABLE brightness(
```

```
        brightness_id serial PRIMARY KEY,
        val text NOT NULL
);

CREATE TABLE color(
        color_id serial PRIMARY KEY,
        val text NOT NULL
);

CREATE TABLE place(
        place_id serial PRIMARY KEY,
        val text NOT NULL
);

CREATE TABLE flash(
        flash_id serial PRIMARY KEY,
        brightness_id int REFERENCES brightness(brightness_id),
        color_id int REFERENCES color(color_id),
        place_id int REFERENCES place(place_id)
);

CREATE TABLE day_period(
        day_period_id serial PRIMARY KEY,
        val text NOT NULL
);

CREATE TABLE landscape(
        landscape_id serial PRIMARY KEY,
        day_period_id int REFERENCES day_period(day_period_id)
);

CREATE TABLE cover(
        cover_id serial PRIMARY KEY,
        val text NOT NULL,
        landscape_id int REFERENCES landscape(landscape_id)
);

CREATE TABLE effect(
        effect_id serial PRIMARY KEY,
        val text NOT NULL
);

CREATE TABLE lake(
        lake_id serial PRIMARY KEY,
        color_id int REFERENCES color(color_id),
        effect_id int REFERENCES effect(effect_id),
        place_id int REFERENCES place(place_id),
        creature_time time
);
```

```
CREATE TABLE lake_landscape(
        lake_id int REFERENCES lake(lake_id),
        landscape_id int REFERENCES landscape(landscape_id),
        PRIMARY KEY(lake_id, landscape_id)
);
```

5) Заполняем таблицы:

```
INSERT INTO color
VALUES
(1, 'blue'),
(2, 'yellow');

INSERT INTO brightness
VALUES
(1, 'bright'),
(2, 'faint');

INSERT INTO place
VALUES
(1, 'near the equator'),
(2, 'along the edge of the lake');

INSERT INTO day_period
VALUES
(1, 'morning'),
(2, 'day'),
(3, 'evening'),
(4, 'night');

INSERT INTO effect
VALUES
(1, 'glow');

INSERT INTO flash
VALUES
(1, 1, 2, 2),
(2, 1, 2, 2),
(3, 1, 2, 2);


INSERT INTO landscape
VALUES
(1, 4);

INSERT INTO cover
VALUES
(1, 'ghostly web of plasma discharge', 1);

INSERT INTO lake
VALUES
(1, 1, 1, 1, NOW());
```

```
INSERT INTO lake_landscape
VALUES
(1, 1);
```

Вывод

Ключевой момент, который стал для меня понятен во время выполнения лабораторной работы – как реализовывать отношение сущностей 'многие-ко-многим’. Также закрепил все изученные ранее знания о СУБД и SQL и применил их на практике.