Министерство высшего образования и науки Российской Федерации

Национальный научно-исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1  
по дисциплине  
**«Базы данных»**

Работу выполнил:  
Афанасьев Кирилл Александрович,  
Студент группы P3106

Преподаватель:  
Карапетян Эрик Акопович

Санкт-Петербург, 2023

Задание:

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 вот настал день, когда в Диаспаре не осталось ни единой живой души. Бодрствовал только Центральный Компьютер, повинующийся внесенным в него указаниям и контролирующий Хранилища Памяти, в которых спали мы все. Не осталось ни одного человека, который сохранил бы хоть какой-то контакт с прошлым... Таким вот образом в этот самый момент и начала свою поступь новая История...»

Локации имеют какое-то население. Население выражается живыми существами. Существа могут быть живыми/не живыми, а также они имеют закрепленную локацию. Локации могут быть за кем-то закреплены. В тексте происходят события, отмеченные временной пометкой. В событии могут участвовать сущности или локации или все сразу (или ничего из этого).

Список сущностей.

Стержневые:

* Сущность – id, имя
* Локация – id, имя, владелец (опционально)
* Событие – id, имя, время

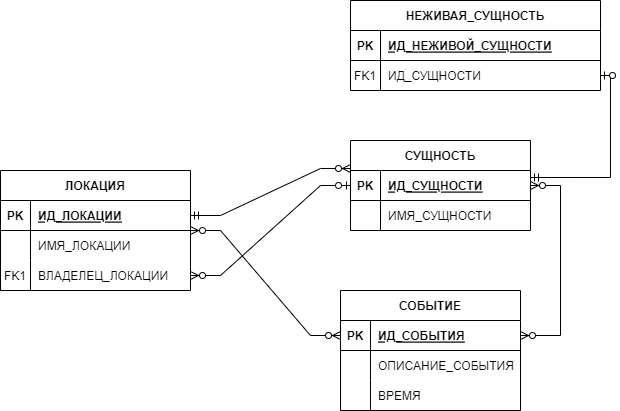
Ассоциации:

* Население – локация-сущность
* События – событие-локация-сущность

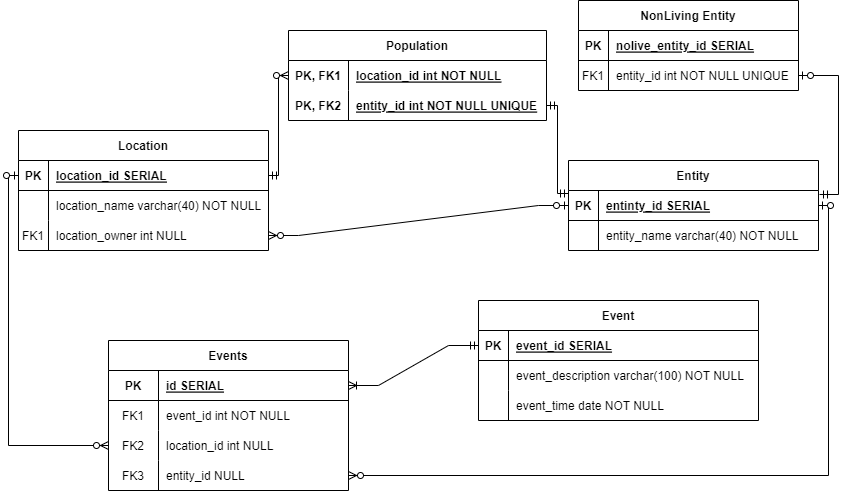
Характеристики:

* Неживые существа – id, id существа

Инфологическая модель:



Даталогическая модель:



Реализация на уровне PostgreSQL:

CREATE TABLE Entity(  
 entity\_id SERIAL PRIMARY KEY,  
 entity\_name varchar(40) NOT NULL  
);  
  
CREATE TABLE Location (  
 location\_id SERIAL PRIMARY KEY,  
 location\_name varchar(40) NOT NULL,  
 location\_owner int REFERENCES Entity(entity\_id) NULL  
);  
  
CREATE TABLE Population(  
 location\_id int REFERENCES Location(location\_id) ON DELETE CASCADE NOT NULL,  
 entity\_id int REFERENCES Entity(entity\_id) ON DELETE CASCADE NOT NULL UNIQUE,

PRIMARY KEY(location\_id, entity\_id)  
);  
  
CREATE TABLE NonLiving\_Entity(  
 noLive\_entity\_id SERIAL PRIMARY KEY,  
 entity\_id int REFERENCES Entity(entity\_id) NOT NULL UNIQUE  
);  
  
CREATE TABLE Event (  
 event\_id SERIAL PRIMARY KEY,  
 event\_description varchar(100) NOT NULL,  
 event\_time date NOT NULL  
);

CREATE TABLE Events (

id SERIAL PRIMARY KEY,

event\_id int REFERENCES Event(event\_id) ON DELETE CASCADE NOT NULL,

location\_id int REFERENCES Location(location\_id) ON DELETE SET NULL NULL,

entity\_id int REFERENCES Entity(entity\_id) ON DELETE SET NULL NULL

);

Заполнение тестовыми данными:

INSERT INTO s367837.entity (entity\_id, entity\_name) VALUES (DEFAULT, 'Центральный компьютер')

INSERT INTO s367837.entity (entity\_id, entity\_name) VALUES (DEFAULT, 'Персонаж\_Шестого\_Плана1')

INSERT INTO s367837.entity (entity\_id, entity\_name) VALUES (DEFAULT, 'Персонаж\_Шестого\_Плана2')

INSERT INTO s367837.entity (entity\_id, entity\_name) VALUES (DEFAULT, 'Персонаж\_Шестого\_Плана3')

INSERT INTO s367837.entity (entity\_id, entity\_name) VALUES (DEFAULT, 'Тестовый\_Персонаж')

INSERT INTO s367837.nonliving\_entity (nolive\_entity\_id, entity\_id) VALUES (DEFAULT, 1)

INSERT INTO s367837.nonliving\_entity (nolive\_entity\_id, entity\_id) VALUES (DEFAULT, 1)

[2023-02-26 04:26:28] [23505] ERROR: duplicate key value violates unique constraint "nonliving\_entity\_entity\_id\_key"

[2023-02-26 04:26:28] Подробности: Key (entity\_id)=(1) already exists.

INSERT INTO s367837.location (location\_id, location\_name, location\_owner) VALUES (DEFAULT, 'Диаспара', null)

INSERT INTO s367837.location (location\_id, location\_name, location\_owner) VALUES (DEFAULT, 'Ячейка\_Памяти1', 1)

INSERT INTO s367837.location (location\_id, location\_name, location\_owner) VALUES (DEFAULT, 'Ячейка\_Памяти2', 1)

INSERT INTO s367837.location (location\_id, location\_name, location\_owner) VALUES (DEFAULT, 'Ячейка\_Памяти3', 1)

INSERT INTO s367837.location (location\_id, location\_name, location\_owner) VALUES (DEFAULT, 'Ячейка\_Памяти4', 1)

INSERT INTO s367837.location (location\_id, location\_name, location\_owner) VALUES (DEFAULT, 'Ячейка\_Памяти5', 1)

INSERT INTO s367837.population (population\_id, location\_id, entity\_id) VALUES (1, 1)

INSERT INTO s367837.population (population\_id, location\_id, entity\_id) VALUES (2, 3)

INSERT INTO s367837.population (population\_id, location\_id, entity\_id) VALUES (3, 2)

INSERT INTO s367837.population (population\_id, location\_id, entity\_id) VALUES (1, 5)

INSERT INTO s367837.event (event\_id, event\_name, event\_time) VALUES (DEFAULT, 'В Диаспаре не осталось ни единой души', '2023-02-25')

INSERT INTO s367837.event (event\_id, event\_name, event\_time) VALUES (DEFAULT, 'Начала поступь новая История', '2023-02-26')

INSERT INTO s367837.events (e\_id, event\_id, location\_id, entity\_id) VALUES (DEFAULT, 1, 1, null)

INSERT INTO s367837.events (e\_id, event\_id, location\_id, entity\_id) VALUES (DEFAULT, 2, 1, 1)

INSERT INTO s367837.events (e\_id, event\_id, location\_id, entity\_id) VALUES (DEFAULT, 2, 2, 3)

INSERT INTO s367837.events (e\_id, event\_id, location\_id, entity\_id) VALUES (DEFAULT, 2, 3, 2)

INSERT INTO s367837.events (e\_id, event\_id, location\_id, entity\_id) VALUES (DEFAULT, 2, 4, null)

INSERT INTO s367837.events (e\_id, event\_id, location\_id, entity\_id) VALUES (DEFAULT, 2, null, 5)

INSERT INTO s367837.events (e\_id, event\_id, location\_id, entity\_id) VALUES (DEFAULT, 2, null, null)

Выводы:

Во время выполнения данной лабораторной работы я ознакомился с архитектурой построения ANSI-SPARC, базовым синтаксисом языка PostgreSQL, а результатом изучения стала созданная на внешнем сервере база данных с заполненными тестовыми значениями.