

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 Р3106

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

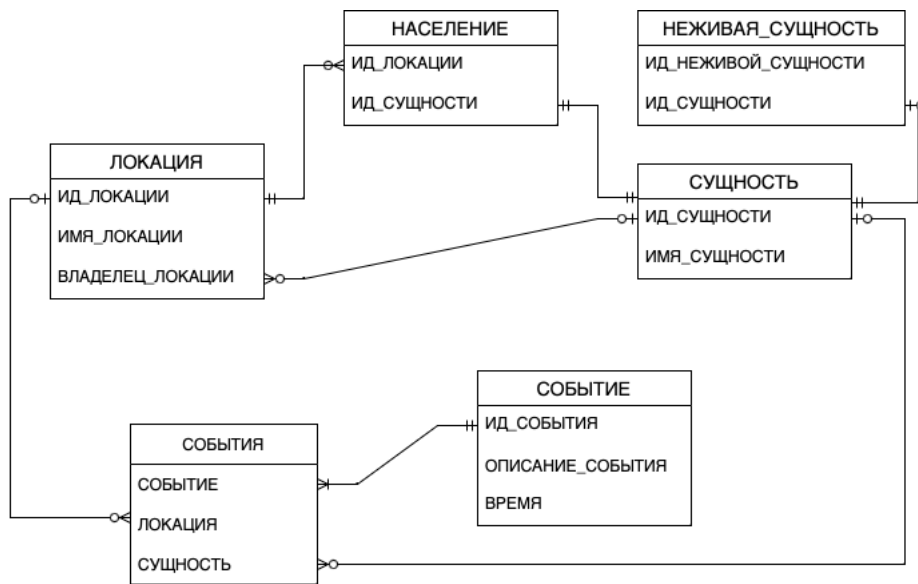
Ассоциации:

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

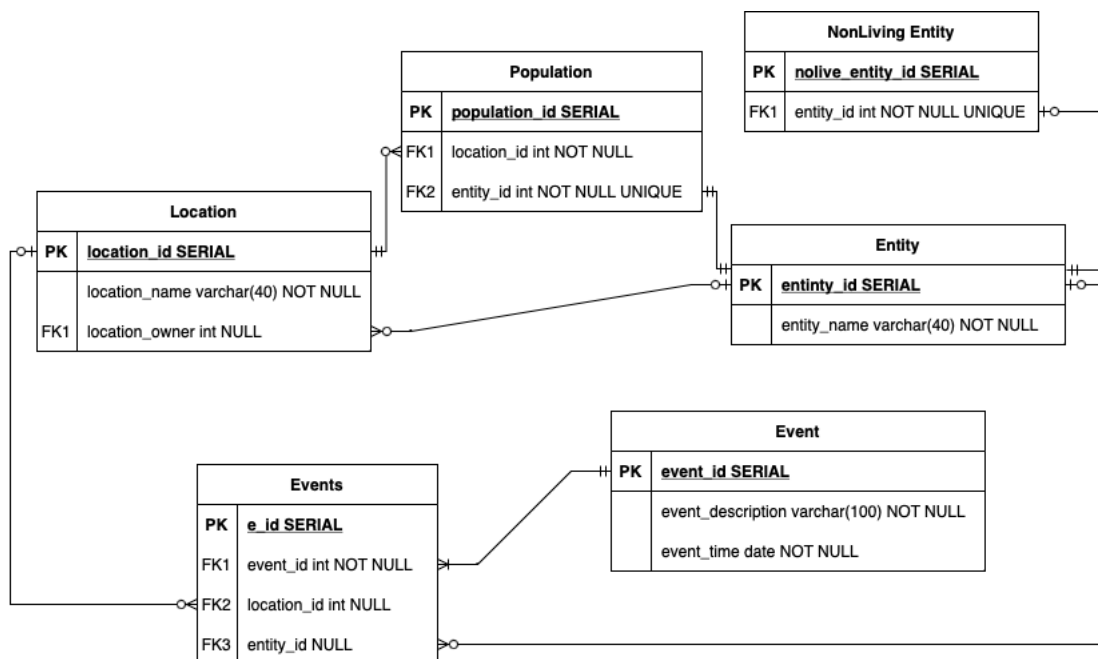
Характеристики:

- 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(
    population_id SERIAL PRIMARY KEY,
    location_id int REFERENCES Location(location_id) NOT NULL,
    entity_id int REFERENCES Entity(entity_id) NOT NULL UNIQUE
);

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(
    e_id SERIAL PRIMARY KEY,
    event_id int REFERENCES Event(event_id) NOT NULL,
    location_id int REFERENCES Location(location_id) NULL,
    entity_id int REFERENCES Entity(entity_id) 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, 'Дияспара', 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
(DEFAULT, 1, 1)
```

```
INSERT INTO s367837.population (population_id, location_id, entity_id) VALUES
(DEFAULT, 2, 3)
```

```
INSERT INTO s367837.population (population_id, location_id, entity_id) VALUES
(DEFAULT, 3, 2)
```

```
INSERT INTO s367837.population (population_id, location_id, entity_id) VALUES
(DEFAULT, 1, 5)
```

```
INSERT INTO s367837.event (event_id, event_name, event_time) VALUES (DEFAULT,
'В Дияспаре 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.