УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Информационный системы и базы данных»

Лабораторная работа №3

Вариант -666

Студент

Агнистова А.Ю.

P3125

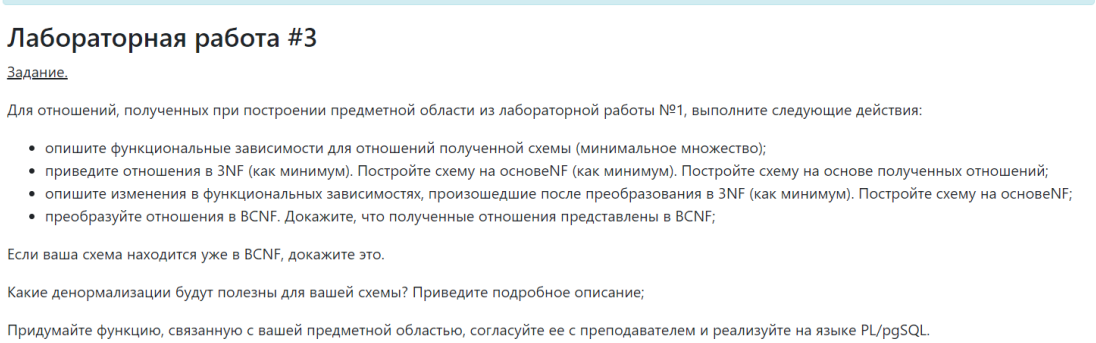
Преподаватель

Николаев В. В.

Санкт-Петербург, 2023 г.

Цель лабораторной работы: получить теоретические и практические навыки в работе с PostgreSQL, ознакомиться с нормализацией баз данных.

Описание задания



Лабораторная работа #3

Задание.

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

- опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
- приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум). Постройте схему на основе полученных отношений;
- опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основеNF;
- преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;

Если ваша схема находится уже в BCNF, докажите это.

Какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

Придумайте функцию, связанную с вашей предметной областью, согласуйте ее с преподавателем и реализуйте на языке PL/pgSQL.

Описание предметной области

Три миллиона лет! Вся до предела насыщенная людьми и событиями панорама Истории со всеми ее империями и королями, победами и трагедиями едва захватывала одну тысячную часть этого устрашающе огромного протяжения времени. Не только сам Человек, но и большинство животных, обитающих ныне на Земле, еще даже не существовали, когда эта загадочная черная глыба была погребена здесь, в самом приметном и ярко освещенном кратере Луны.

Список сущностей

Стержневые:

- *Object – name, description, type of object*
- *Location – name, description*
- *Action - description*

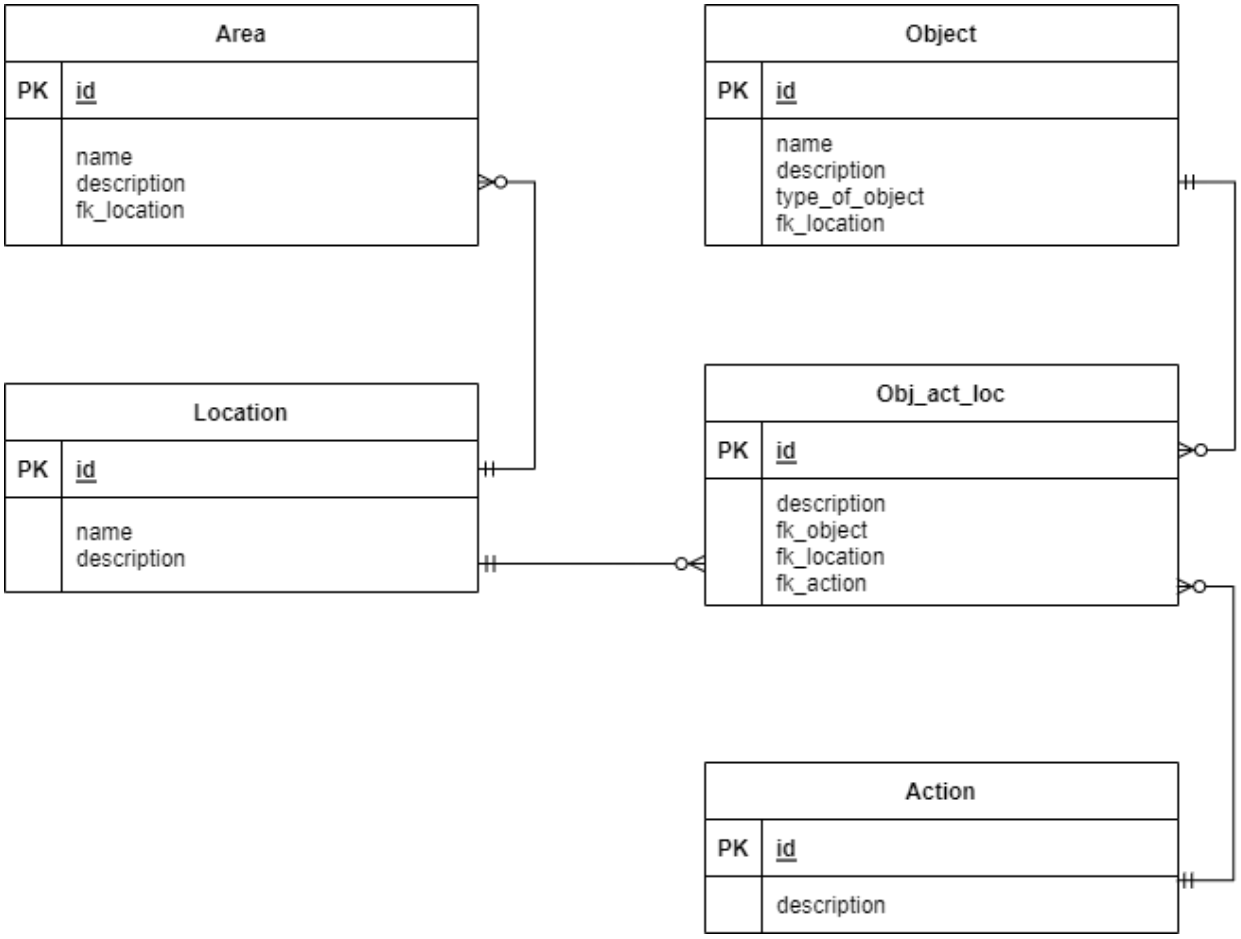
Характеристические:

- *Area – name, description, fk_location*

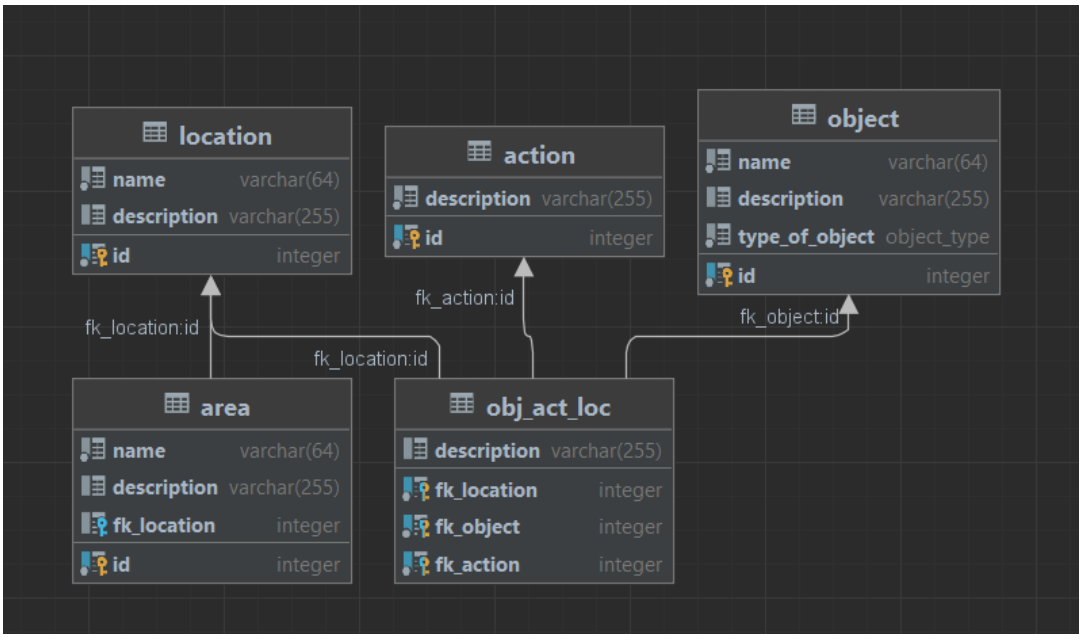
Ассоциативные*:*

- *Obj_act_loc – fk_location, fk_object, fk_action, description*

Инфологическая модель



Даталогическая модель



Функциональные зависимости для отношений схемы (минимальное множество):

- (area) id -> name
- (area) id -> description
- (area) id -> fk_location
- (location) id -> name
- (location) id -> description
- (object) id -> name
- (object) id -> description
- (object) id -> type_of_object
- (object) id -> fk_location
- (action) id -> description

Нормализация:

1 Нормальная форма (1НФ):

- на пересечении строки и столбца содержится только одно значение;
- наличие первичного ключа.

Это ключевое определение реляционной базы данных и при выполнении первой лабораторной работы это было учтено.

Изменения не требуются

2 Нормальная форма (2НФ):

- отношения находятся в 1НФ;
- каждый атрибут, не входящий в первичный ключ, полностью функционально зависит от первичного ключа.

Чтобы привести к 2НФ необходимо убрать частичные зависимости от первичного ключа (т.е. в случае наличия составного первичного ключа, все атрибуты полностью зависят от ключа, а не только от его части).

В каждой таблице атрибуты не включенные в первичный ключ непосредственно зависят от него. Следовательно схема удовлетворяет условиям 2НФ.

- (area) id -> name
- (area) id -> description
- (area) id -> fk_location
- (location) id -> name
- (location) id -> description
- (object) id -> name
- (object) id -> description
- (object) id -> type_of_object
- (object) id -> fk_location

- (action) id -> description

Изменения не требуются

3 Нормальная форма (3НФ):

- отношения находятся в 2НФ;
- нет атрибутов, не входящих в первичный ключ, которые находятся в транзитивной зависимости от первичного ключа.
  Для того, чтобы доказать, что отношения находятся в 3НФ необходимо перебрать все возможные транзитивные зависимости. Учтем, что для отношений action и location перебор невозможен из-за количества полей, так как для транзитивной зависимости нужно минимум 3 элемента.
  Area:
- id -> name
- id -> description
- id -> fk_location

Атрибуты name, description, fk_location не зависят друг от друга, следовательно нет случая транзитивной зависимости, все атрибуты напрямую зависят от первичного ключа и только от него.

Object:

- id -> name
- id -> description
- id -> type_of_object
- id -> fk_location

Атрибуты name, description, type_of_object, fk_location не зависят друг от друга, следовательно нет случая транзитивной зависимости, все атрибуты напрямую зависят от первичного ключа и только от него.

Obj_act_loc:

- id -> description
- id -> fk_object
- id -> fk_location
- id -> fk_action

Атрибуты name, description, fk_object, fk_location, fk_action не зависят друг от друга, следовательно нет случая транзитивной зависимости, все атрибуты напрямую зависят от первичного ключа и только от него.

Можем сделать вывод, что нет ни одного отношения с транзитивными зависимостями, значит база данных находится в 3НФ.

Изменения не требуются

Нормальная форма Бойса-Кодда:

- отношения находятся в 3НФ;
- каждая нетривиальная и неприводимая слева функциональная зависимость обладает потенциальным ключом в качестве детерминанта.

A1 -> A2 , A2 -> может быть первичным ключом, A1 — обязательно ключ.

Ситуация, когда отношение будет находиться в 3НФ, но не в Бойса-Кодда, возникает, например, при условии, что отношение имеет два (или более) потенциальных ключа, которые являются составными, и между отдельными атрибутами таких ключей существует функциональная зависимость. Поскольку описанные зависимости не являются транзитивной, то такая ситуация под определение 3НФ не подпадает. На практике такие отношения встречаются достаточно редко, для всех прочих отношений 3НФ и Нормальная форма Бойса-Кодда эквивалентны.

Каждый детерминант каждой таблицы является потенциальным ключем:

- Location
  - id -> name, description
  - (name, description) – потенциальный ключ
- Action
  - id -> description
  - (description) – потенциальный ключ
- Area
  - Id -> name, description, fk_location
  - (name, description, fk_location) – потенциальный ключ
- Object
  - Id -> name, description,type_of_object, fk_location
  - (name, description,type_of_object, fk_location) – потенциальный ключ
- Obj_act_loc
  - Id -> description,fk_object, fk_location, fk_action
  - (description,fk_object, fk_location, fk_action) – потенциальный ключ

<u>Изменения не требуются</u>

Полезные денормализации:

По тексту «точное» местоположение есть только у Глыбы, а именно «кратер Луны», значит можно удалить таблицу area. Это не приведет к появлению новых ФЗ, а скорее обобщит данные в базе данных.

Функция:

Функция, которая ищет количество живых объектов на определенной локации.

```
CREATE OR REPLACE FUNCTION
count_animated_objects_on_location(location_name TEXT)
RETURNS INTEGER AS $$
DECLARE
      result INTEGER;
BEGIN
      SELECT COUNT(*) INTO result
      FROM Object
      INNER JOIN Obj_act_loc ON Object.id = Obj_act_loc.fk_object
      INNER JOIN Location ON Obj_act_loc.fk_location = Location.id
      WHERE Object.type_of_object = 'animate' AND Location.name =
location_name;
      RETURN result;
END;
$$ LANGUAGE plpgsql;
```

Выводы

При выполнении лабораторной работы я познакомилась с нормальными формами для нормализации базы данных, написала функцию, которая может быть полезной для моей базы данных.