

Федеральное государственное автономное образовательное учреждение
высшего образования “Национальный исследовательский университет ИТМО”

Факультет Программной Инженерии И Компьютерной Техники

Лабораторная работа №3

Выполнила:

Абдуллаева София Улугбековна

Группа Р3108

Проверила:

Заболотняя Ольга Михайловна

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).
- 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 триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

Исходная предметная область

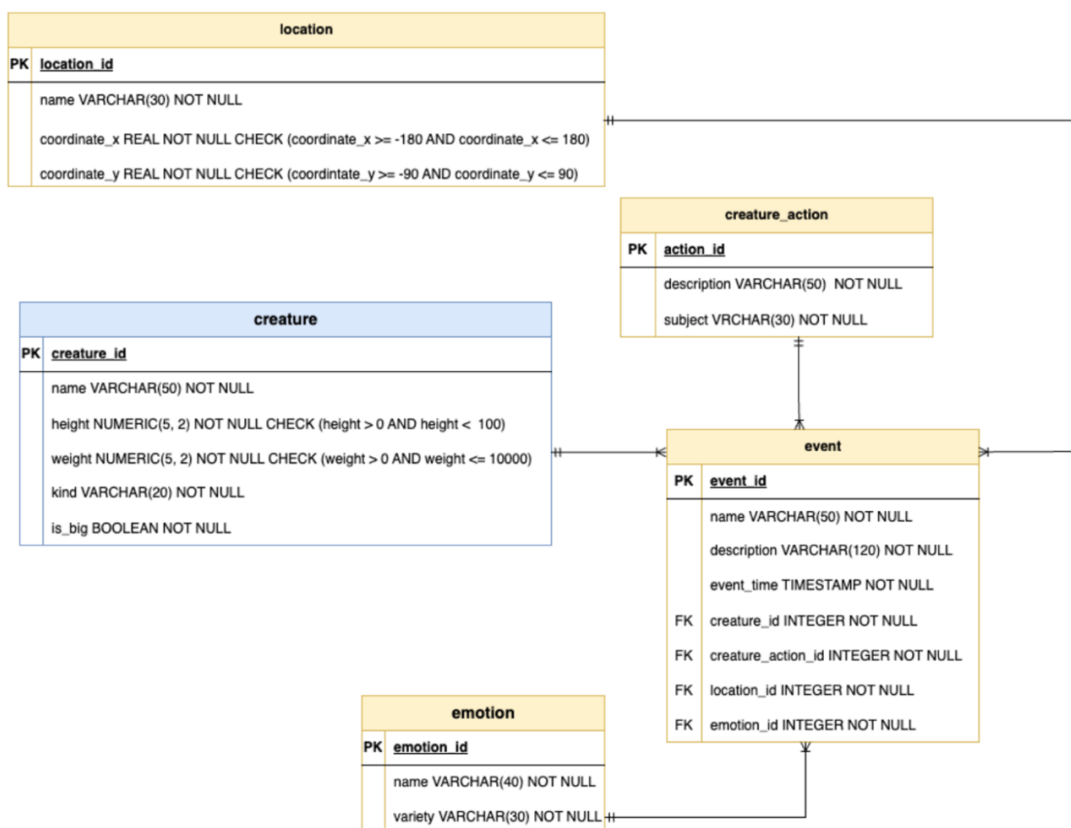
Введите вариант:

Описание предметной области, по которой должна быть построена доменная модель:

Впервые Смотрящий на Луну увидел Новый Камень в слабом свете нарождающегося дня, когда повел свою стаю на утренний водопой. Он почти забыл о всех ночных страхах – ведь после того необычайного звука ничего не случилось – и потому новый странный предмет не вызвал у него ни страха, ни ощущения опасности. Да в нем и не было ничего страшного.

Есть некоторые **существа**, они одного вида. Одно из них (Смотрящий на Луну) видит некоторый **предмет** (Новый Камень) **утром**, когда они меняют **локацию** (направляются к водопою). До этого оно испытывало негативную **эмоцию** (страх) **по ночам**. После некоторого **события** (необычайного звука) ничего не случилось, поэтому он уже не чувствует негативную **эмоцию** (страх, ощущение опасности) при виде этого предмета, но **заинтересован** им. В целом события могут быть у любого существа, также оно тоже может увидеть предметы.

Даталогическая модель



Функциональные зависимости:

creature: creature_id -> name, height, weight, kind

location: location_id -> name, coordinate_x, coordinate_y

event: event_id -> name, event_time, creature_id, creature_action_id, location_id, emotion_id

creature_action: action_id -> description

emotion: emotion_id -> name, variety

Нормальные формы

1NF: отношение находится в 1NF, так как все его атрибуты содержат только атомарные значения и на пересечении строки и столбца – ровно 1 значение

2NF: отношение находится в 2NF, так как оно соответствует 1NF и все неключевые атрибуты находятся в полной функциональной зависимости от первичного ключа

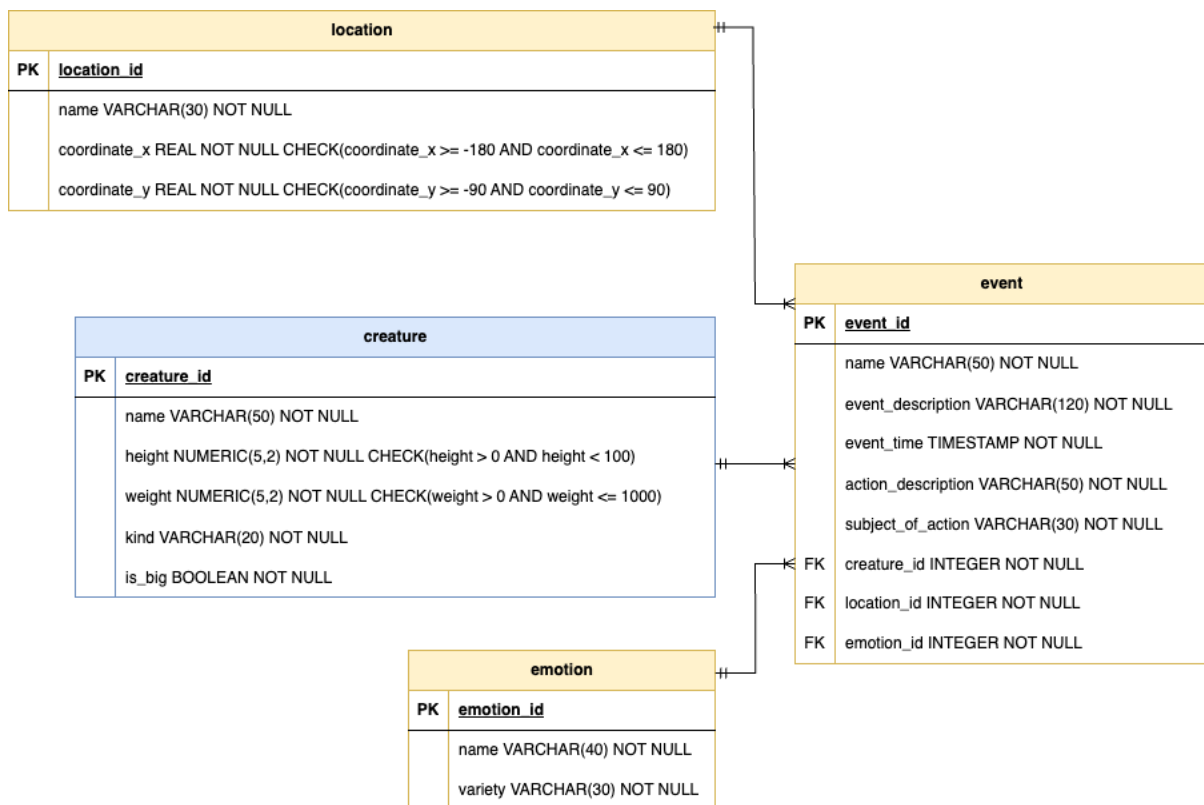
3NF: отношение находится в 3NF, так как оно соответствует 2NF и все неключевые атрибуты находятся в не транзитивной функциональной зависимости от первичного ключа

BCNF: отношение находится в BCNF, так как для каждой нетривиальной зависимости $A \rightarrow B$ A является суперключом. Получается, все ключевые атрибуты не зависят от неключевых атрибутов.

Денормализованная модель

Денормализация нужна только в том случае, если мы хотим улучшить производительность запросов, например, с JOIN, она выполняется через объединение таблицы или нескольких таблиц в одну. Однако появляется избыточность данных, и повышается риск аномалий.

В моём случае можно объединить таблицу creature_action с event, так как событие существа связано с действием(существо выполняет действие во время некоторого события), и, соответственно, можно убрать из таблицы event FK и добавить атрибуты, которые были в creature_action.



Триггер

Назначение триггера: проверка правильности передаваемого командой INSERT/UPDATE значения TRUE или FALSE.

Триггерная функция: проверка, если существо имеет вид 'Leo', то происходит проверка поля is_big, оно может быть TRUE только, когда его $BMI = \text{weight}/(\text{height_in_meters})^2 > 400$, иначе FALSE. Если значение установлено неправильно, то вызывается исключение. В моём случае

данное исключение ловится, происходит откат к точке сохранения , и транзакция продолжается.

```
CREATE TABLE location(  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(30) NOT NULL,  
    coordinate_x REAL NOT NULL CHECK (coordinate_x >= -180 AND  
coordinate_x <= 180),  
    coordinate_y REAL NOT NULL CHECK (coordinate_y >= -90 AND  
coordinate_y <= 90));  
  
CREATE TABLE creature(  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    height NUMERIC(5, 2) NOT NULL CHECK (height > 0 AND height  
< 100),  
    weight NUMERIC(5, 2) NOT NULL CHECK (weight > 0 AND weight  
<= 10000),  
    kind VARCHAR(20) NOT NULL,  
    is_big BOOLEAN NOT NULL);  
  
CREATE TABLE emotion(  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(40) NOT NULL,  
    variety VARCHAR(30) NOT NULL);  
  
CREATE TABLE creature_action(  
    id SERIAL PRIMARY KEY,  
    description VARCHAR(50) NOT NULL,  
    subject VARCHAR(30) NOT NULL);  
  
CREATE TABLE event(  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(50) NOT NULL,  
    description VARCHAR(120) NOT NULL,  
    event_time TIMESTAMP NOT NULL,  
    creature_id INTEGER NOT NULL REFERENCES creature(id),  
    creature_action_id INTEGER NOT NULL REFERENCES  
creature_action(id),  
    location_id INTEGER NOT NULL REFERENCES location(id),  
    emotion_id INTEGER NOT NULL REFERENCES emotion(id));  
  
INSERT INTO location(name, coordinate_x, coordinate_y)  
VALUES ('watering hole', 10.51, -76.2), ('forest', 20.6,  
62.45), ('desert', 78.13, 12.345);  
  
INSERT INTO emotion(name, variety)  
VALUES('interest', 'positive'), ('fear', 'negative'),  
('delight', 'positive'), ('excitement', 'positive');
```

```

INSERT INTO creature_action(description, subject)
VALUES ('see', 'stone'), ('touch', 'stone'), ('admire',
'wonderful view'), ('search for a subject', 'ancient
artifact');
CREATE OR REPLACE FUNCTION is_creature_big_by_BMI()
    RETURNS TRIGGER
AS $$
DECLARE
    height_in_meters DOUBLE PRECISION;
    doubled_height DOUBLE PRECISION;
    bmi DOUBLE PRECISION;
    expected_is_big BOOLEAN;
BEGIN
    IF (NEW.kind = 'Leo')
    THEN
        height_in_meters := NEW.height / 100;
        doubled_height := height_in_meters * height_in_meters;
        bmi := NEW.weight / doubled_height;
        expected_is_big := (bmi > 400);
        RAISE NOTICE 'Триггер вызван для %, height: %, weight:
%, bmi: %',
            NEW.name, NEW.height, NEW.weight, bmi;
        IF NEW.is_big <> expected_is_big THEN
            RAISE EXCEPTION 'Значение поля is_big установлено
некорректно, ожидалось expected_is_big: %', expected_is_big;
        END IF;
    END IF;
    RETURN NEW;
END;
$$LANGUAGE plpgsql;

CREATE TRIGGER is_creature_big_by_BMI_trigger
    BEFORE INSERT OR UPDATE ON creature
    FOR EACH ROW
EXECUTE FUNCTION is_creature_big_by_BMI();

BEGIN;
INSERT INTO creature(name, height, weight, kind, is_big)
VALUES('Looking at the moon', 94.23, 635, 'Leo', TRUE);

SAVEPOINT sp1;
DO $$
    BEGIN
        INSERT INTO creature(name, height, weight, kind,
is_big)
        VALUES('Simba', 94, 520, 'Leo', FALSE);
        EXCEPTION WHEN OTHERS THEN
            RAISE NOTICE 'Ошибка при вставке Simba, %', SQLERRM;
        END$$;
ROLLBACK TO SAVEPOINT sp1;

```

```

INSERT INTO creature(name, height, weight, kind, is_big)
VALUES ('Shrek', 45.26, 700.04, 'Ogre', TRUE);

INSERT INTO creature(name, height, weight, kind, is_big)
VALUES ('Mike Wazowski', 23.55, 98.29, 'Monster', FALSE);
COMMIT;

INSERT INTO event(name, description, event_time, creature_id,
creature_action_id, location_id, emotion_id)
VALUES ('mysterious footprints', 'unusual footprints have
appeared in the forest leading deeper into the trees', '2024-
07-16 19:15:24', 3, 3, 2, 3),
      ('unusual sound', 'a weird subject near the watering
hole made sounds', '2024-05-13 06:30:25', 1, 1, 1, 1),
      ('desert mirage', 'a magnificent desert mirage rises
from the sands shimmering in the distance', '2024-06-15
13:50:10', 4, 4, 3, 4);

```

Вывод отладочной информации

```

[2025-05-05 21:29:29] completed in 88 ms
studs> BEGIN
[2025-05-05 21:29:29] completed in 54 ms
studs> INSERT INTO creature(name, height, weight, kind, is_big)
      VALUES('Looking at the moon', 94.23, 635, 'Leo', TRUE)
Триггер вызван для Looking at the moon, height: 94.23, weight: 635.00, bmi: 715.1470360888759
[2025-05-05 21:29:30] 1 row affected in 70 ms
studs> SAVEPOINT sp1
[2025-05-05 21:29:30] completed in 57 ms
studs> DO $$
      BEGIN
          INSERT INTO creature(name, height, weight, kind, is_big)
          VALUES('Simba', 94, 520, 'Leo', FALSE);
      EXCEPTION WHEN OTHERS THEN
          RAISE NOTICE 'Ошибка при вставке Simba, %', SQLERRM;
      END$$
Триггер вызван для Simba, height: 94.00, weight: 520.00, bmi: 588.5015844273428
Ошибка при вставке Simba, Значение поля is_big установлено некорректно, ожидалось expected_is_big: t
[2025-05-05 21:29:30] completed in 55 ms
studs> ROLLBACK TO SAVEPOINT sp1
[2025-05-05 21:29:30] completed in 61 ms
studs> INSERT INTO creature(name, height, weight, kind, is_big)
      VALUES ('Shrek', 45.26, 700.04, 'Ogre', TRUE)
[2025-05-05 21:29:30] 1 row affected in 61 ms
studs> INSERT INTO creature(name, height, weight, kind, is_big)
      VALUES ('Mike Wazowski', 23.55, 98.29, 'Monster', FALSE)
[2025-05-05 21:29:31] 1 row affected in 62 ms
studs> COMMIT

```

Вывод

В процессе выполнения лабораторной работы я узнала про функциональные зависимости в отношениях, аномалии операций с БД, нормальные формы, как приводить отношение в определённую нормальную форму, денормализацию, как пользоваться функциями, процедурами и создавать триггеры.