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 P3108

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



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

A diagram of a function

Description automatically generated

## Функциональные зависимости:

**сreature:** 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 -> 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 = weight/(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

A computer screen shot of a program

Description automatically generated

## 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.