*Факультет программной инженерии и компьютерной техники*

*Направление подготовки: 09.03.04 – Программная инженерия,*

*Системное и прикладное программное обеспечение*

**Лабораторная работа по №3**

по дисциплине

«Базы данных»

**Вариант №3139349**

Выполнил:

Ткачев Денис Владимирович

Группа P3111

Преподаватели:

Харитонова Анастасия Евгеньевна

Оглавление

[Задание 2](#_Toc87370215)

[Ход работы 3](#_Toc87370216)

[Текст исходной программы 3](#_Toc87370217)

[Описание программы 3](#_Toc87370218)

[Трассировка 4](#_Toc87370219)

[Вариант программы с меньшим количеством команд 4](#_Toc87370220)

[Трассировка с данными числами 5](#_Toc87370221)

[Вывод 5](#_Toc87370222)

[Список литературы 5](#_Toc87370223)

## 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 основе NF;
* 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.

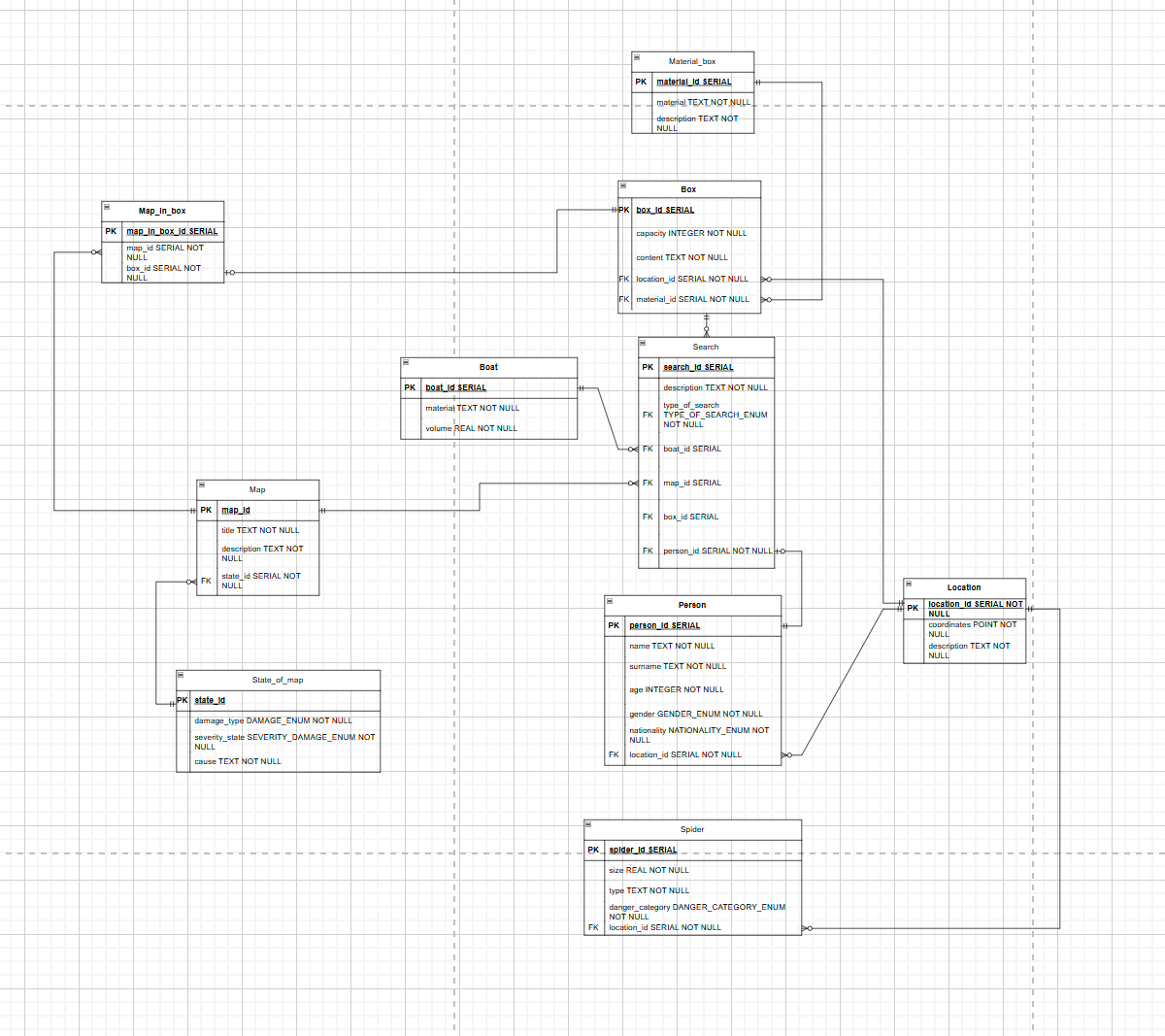
Придумайте триггер и связанную с ним функцию, относящиеся к вашей предметной области, согласуйте их с преподавателем и реализуйте на языке PL/pgSQL.

## Описание предметной области

«Роясь в барахле. **Грант** не обнаружил никакой **лодки**, но зато обнаружил в металлическом **ящике** на стене комплект **карт**, которые от сырости покоробились и покрылись **плесенью**. Грант расстелил карты на **полу** и отбросил в сторону большого **паука**. Он рассматривал их очень долго.»

Из текста можно выделить такие сущности, как: Человек Грант (Person); Лодка; Ящик, в которой затем содержались карты. У карт есть состояние, повреждение (Покорбленные, покрытые плесенью). У ящика есть материал (металлический) из которого он сделан. Грант из изначально лодку, но затем находит ящик, в которой содержится комплект карт, которые покрылись плесенью. Расстелает карты на пол, который имеет цвет и материал. Попутно в это же время грант отбрасывает в сторону большого паука, у которого есть размер, вид, местонахождение.

## **Даталогическая модель**



## **Функциональные зависимости**

1. location: location\_id --> (coordinates, description)

2. spider: spider\_id --> (size, type, danger\_category, location\_id)

3. person: person\_id --> (name, surname, age, gender, nationality, location\_id)

4. search: search\_id --> (description, type\_of\_search, boat\_id, map\_id, box\_id, person\_id)

5. boat: boat\_id --> (material, volume)

6. map: map\_id --> (title, description, state\_id)

7. state\_of\_map: state\_id --> (damage\_type, severity\_state, cause)

8. box: box\_id --> (capacity, content, location\_id, material\_id)

9. material\_box: material\_id --> (material, description)

10. map\_in\_box: map\_in\_box\_id --> (map\_id, box\_id)

## **Нормальные формы**

1NF: Отношение находится в 1NF, если все его атрибуты содержат только атомарные значения. Моя модель удовлетворяет 1NF, так как все атрибуты атомарны, и нет повторяющихся групп.

2NF: Отношение находится во 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа. Моя модель удовлетворяет 2NF, так как все неключевые атрибуты полностью функционально зависят от первичных ключей. Нет частичных функциональных зависимостей.

3NF: Отношение находится в 3NF, если оно находится во 2NF и не содержит транзитивных зависимостей. Моя модель удовлетворяет 3NF, так как все неключевые атрибуты зависят только от первичных ключей, и не содержат транзитивных зависимостей.

## **BCNF**

Отношение находится в BCNF, если для каждой функциональной зависимости X →Y,

X является суперключом. Моя модель удовлетворяет BCNF, так как

для всех функциональных зависимостей X является суперключом.

## **Денормализация**

**Объединение связанных таблиц**: В некоторых случаях, объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, можно рассмотреть объединение таблиц box и material\_box, map и state\_of\_map, если часто запрашиваются данные о коробке и материал коробки или запрашивается карта и состояние карты.

**Добавление избыточных атрибутов**: В некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, если часто запрашивается количество карт отдельных коробок, можно добавить атрибут map\_count в таблицу box. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении или удалении карт в соответствующие коробки.

## **Функция и триггер на языке PL/pgSQL**

Триггер для вычисления среднего возраста всех людей

Функция и триггер на языке PL/pgSQL для вычисления среднего возраста всех людей

-- Триггер для вычисления среднего возраста всех людей

CREATE OR REPLACE FUNCTION *calculate\_global\_avg\_age\_trigger\_func*()  
RETURNS TRIGGER AS $$  
DECLARE  
 global\_avg\_age FLOAT;  
BEGIN  
 -- Вычисляем средний возраст всех людей  
 SELECT *AVG*(age) INTO global\_avg\_age  
 FROM person;  
  
 -- Выводим результат в консоль  
 RAISE NOTICE 'Добавлен человек с возрастом %. Средний возраст всех людей: % лет',  
 NEW.age,  
 *COALESCE*(global\_avg\_age::NUMERIC(10,2)::TEXT, 'нет данных');  
  
 RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
-- Создаем триггер  
CREATE TRIGGER person\_global\_avg\_age\_trigger  
AFTER INSERT ON person  
FOR EACH ROW  
EXECUTE FUNCTION *calculate\_global\_avg\_age\_trigger\_func*();  
  
INSERT INTO person (name, surname, age, gender, nationality, location\_id) VALUES ('Nikita', 'Ryz', 19, 'male', 'Russian',1);  
INSERT INTO person (name, surname, age, gender, nationality, location\_id) VALUES ('Grigoriy', 'Zizikov', 45, 'male', 'Russian',1);

## Вывод

При выполнении лабораторной работы я познакомился с понятием нормализации и денормализации. Научился определять функциональные зависимости модели, а также анализировать последнюю на соответствие различным нормальным формам. Познакомился с процедурным языком PL/pgSQL. Изучил эффективные способы денормализации схемы базы данных и ситуации, в которых возможно их применение.