Федеральное государственное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Дисциплина «Базы данных»

**Лабораторная работа № 3**

Выполнил:

Смирнов Вадим

Проверил:

Миняев Илья Андреевич

Санкт-Петербург

2025

Оглавление

[Задание 3](#_Toc198537481)

[Функциональные зависимости 4](#_Toc198537482)

[Нормальные формы 4](#_Toc198537483)

[BCNF 4](#_Toc198537484)

[Денормализация 5](#_Toc198537485)

[Триггер 5](#_Toc198537486)

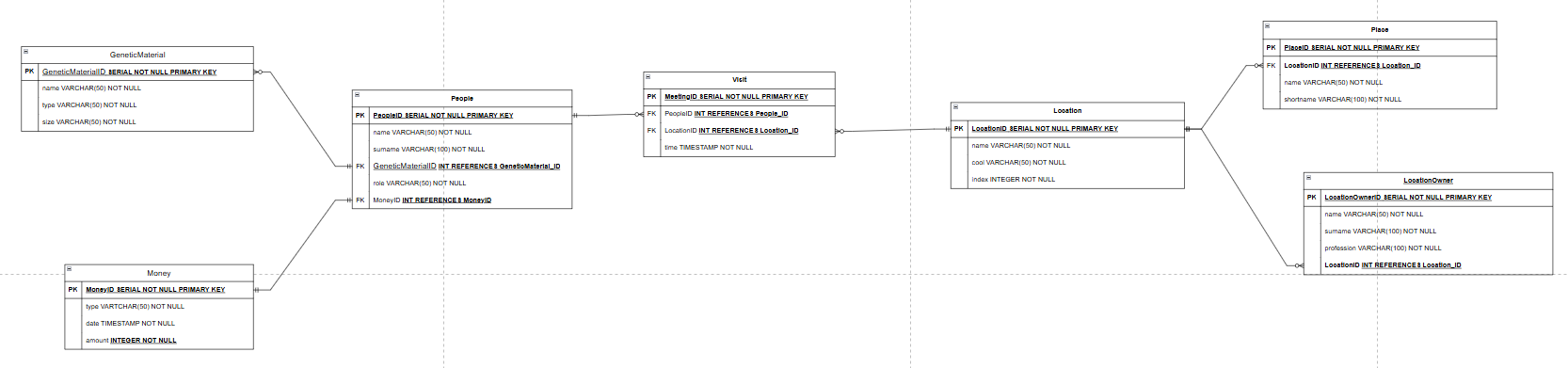
[Вывод 7](#_Toc198537487)

### 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.



### Функциональные зависимости

|  |
| --- |
| GeneticMaterial: GeneticMaterialID→ (name, type, size)  Money: MoneyID → (type, date, amount)  People: PeopleID → (name, surname, GeneticMaterialID, role, MoneyID)  Location: LocationID → (name, cool, index)  LocationOwner: LocationOwnerID → (name, surname, profession, LocationID)  Place: PlaceID → (LocationID, name, shortname)  Visit: VisitID → (name, from\_time, to\_time, employee\_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.  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 Location и Place, если часто запрашиваются данные локации и её расположении.  **Добавление избыточных атрибутов**: В некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, в таблицу LocationOwner можно добавить поле hard\_profession INTEGER. Это нарушает 2НФ, потому что это поле зависит от поля profession. |

### Триггер

Триггер проверяет перед обновлением записи в таблице Money, что сумма не превышает 100000 для людей с ролью "Сотрудник", иначе выбрасывает ошибку.

DROP TRIGGER IF EXISTS money\_limit\_trigger ON Money;

DROP FUNCTION IF EXISTS check\_money\_limit();

CREATE OR REPLACE FUNCTION check\_money\_limit()

RETURNS TRIGGER AS $$

DECLARE

    role\_limit VARCHAR(50);

BEGIN

    SELECT role INTO role\_limit

    FROM People

    WHERE MoneyID = NEW.MoneyID;

    IF NOT FOUND THEN

        RAISE EXCEPTION 'No person found for MoneyID %', NEW.MoneyID;

    END IF;

    IF role\_limit = 'Сотрудник' AND NEW.amount > 100000 THEN

        RAISE EXCEPTION 'Amount % exceeds limit for role Сотрудник', NEW.amount;

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER money\_limit\_trigger

BEFORE UPDATE ON Money

FOR EACH ROW

EXECUTE FUNCTION check\_money\_limit();

### Вывод

Во время выполнения лабораторной работы я познакомился с триггерами, нормализацией и денормализацией. Научился правильно это реализовывать и применять. Узнал подробнее про нормальные формы.