**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Дисциплина:** Базы данных

**Лабораторная работа №3**

**Вариант №1488**

Выполнила:

Хвостова И.Л

Группа:

P3124

Проверил:

Королёва Ю.А.

г.Санкт-Петербург

2023г.

**Оглавление**

[Задание 3](#_Toc135322768)

[Выполнение задания 4](#_Toc135322769)

[Реализация модели на SQL 7](#_Toc135322770)

# Задание

Для выполнения лабораторной работы №3 необходимо:

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.

# Выполнение задания

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

* Опишите функциональные зависимости для отношений полученной схемы (минимальное множество);

1. Отношение "Компания":
   1. company\_id -> name, genetic\_engineering, artificial\_animals\_creation, supercomputers\_quantity, amber\_reserves
2. Отношение "Вкладчики":
   1. investor\_id -> company\_id, name, country, contribution
3. Отношение "Остров":
   1. island\_id -> company\_id, amber\_id, name, location
4. Отношение "Суперкомпьютеры":
   1. supercomputer\_id -> company\_id, model, manufacturer, characteristics
5. Отношение "Искусственные копии животных":
   1. artificial\_animal\_id -> company\_id, dna\_id, supercomputer\_id, zoo\_id, animal\_species, characteristics, creation\_method
6. Отношение "Запасы янтаря":
   1. amber\_id -> company\_id, quantity, quality
7. Отношение "ДНК":
   1. dna\_id -> scientist\_id, amber\_id, source, characteristics, usage
8. Отношение "Зоопарки":
   1. zoo\_id -> name, location
9. Отношение "Дары зоопаркам":
   1. gift\_id -> company\_id, zoo\_id
10. Отношение "Ученые":
    1. scientist\_id -> company\_id, name, date\_of\_birth, gender, specialization, experience\_years

* Приведите отношения в 3NF (как минимум). Постройте схему на основеNF (как минимум).

3NF (Third Normal Form) - третья нормальная форма, является одним из уровней нормализации баз данных. Она определяет определенные требования к структуре данных в таблицах для устранения избыточности и зависимостей между атрибутами. Основные требования 3NF:

1. Все атрибуты в таблице должны быть атомарными. Это означает, что каждый атрибут должен содержать только одно значение.

2. Если в таблице есть неключевые атрибуты, они должны полностью зависеть от первичного ключа и не должны зависеть от других неключевых атрибутов. Все неключевые атрибуты должны зависеть только от первичного ключа.

3. В таблице не должно быть транзитивных зависимостей между неключевыми атрибутами. Это означает, что если атрибут A зависит от атрибута B, а атрибут B зависит от атрибута C, то атрибут A не должен зависеть от атрибута C.

Модель удовлетворяет требованиям.

Пример описания для первых трех сущностей

1. Отношение "Компания" (Company):
   * Каждый атрибут в отношении является атомарным.
   * Уникальный идентификатор (company\_id) является первичным ключом.
   * Все остальные атрибуты (name, genetic\_engineering, artificial\_animals\_creation, supercomputers\_quantity, amber\_reserves) зависят только от company\_id и не имеют повторяющихся групп атрибутов.
2. Отношение "Вкладчики" (Investor):
   * Каждый атрибут в отношении является атомарным.
   * Уникальный идентификатор (investor\_id) является первичным ключом.
   * Атрибут company\_id является внешним ключом, связывающим таблицу с отношением "Компания".
   * Все остальные атрибуты (name, country, contribution) зависят только от investor\_id и не имеют повторяющихся групп атрибутов.
3. Отношение "ДНК" (DNA):
   * Каждый атрибут в отношении является атомарным.
   * Уникальный идентификатор (dna\_id) является первичным ключом.
   * Атрибуты scientist\_id и amber\_id являются внешними ключами, связывающими таблицу с отношениями "Ученые" и "Запасы янтаря".
   * Все остальные атрибуты (source, characteristics, usage) зависят только от dna\_id и не имеют повторяющихся групп атрибутов.

* Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если ваша схема находится уже в BCNF, докажите это;

BCNF гарантирует, что каждый неключевой атрибут зависит только от первичного ключа, а не от любого другого неключевого атрибута в таблице.

Модель удовлетворяет требованиям.

* Какие денормализации будут полезны для вашей схемы? Приведите подробное описание.

Денормализация может быть полезной в предметной области для повышения производительности запросов, упрощения структуры модели и улучшения понятности данных. Ниже приведены несколько потенциальных денормализаций, которые могут быть полезны для данной схемы:

1. Добавление поля "supercomputers\_quantity" в таблицу "Остров":
   * Позволит избежать необходимости объединения с таблицей "Суперкомпьютеры" для получения количества суперкомпьютеров на острове.
   * Улучшит производительность запросов, связанных с информацией об островах и количестве суперкомпьютеров.

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

Для предметной области можно рассмотреть следующий триггер и связанную с ним функцию:

* Этот триггер обновляет количество суперкомпьютеров владельца компании при добавлении или удалении суперкомпьютера.
* Эта функция обновляет количество суперкомпьютеров у соответствующей компании.

# Реализация модели на SQL

-- Создание функции  
CREATE OR REPLACE FUNCTION *update\_supercomputers\_quantity*()  
 RETURNS TRIGGER AS  
$$  
BEGIN  
 UPDATE company  
 SET supercomputers\_quantity = (SELECT *COUNT*(\*)  
 FROM supercomputer  
 WHERE company\_id = NEW.company\_id)  
 WHERE company\_id = NEW.company\_id;  
  
 RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
-- Создание триггера  
CREATE TRIGGER update\_supercomputers\_quantity\_trigger  
 AFTER INSERT OR DELETE ON supercomputer  
 FOR EACH ROW  
EXECUTE FUNCTION *update\_supercomputers\_quantity*();

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание