МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Национальный исследовательский университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И  
КОМПЬЮТЕРНОЙ ТЕХНИКИ

**ЛАБОРАТОРНАЯ РАБОТА №3**

по дисциплине

«Базы данных»

***Выполнил:***Студент группы P3107  
Пшеничников Артём Дмитриевич

***Преподаватель:***Байрамова Хумай Бахруз Кызы

**Содержание**

[Задание 3](#_gjdgxs)

[Функциональные зависимости 4](#_69g569z0kesh)

[Нормальные формы 5](#_alx52okj6kqx)

[Денормализация 6](#_luozjm6qpa1)

[Триггер 7](#_uorw5v7o0k7q)

[Заключение](#_3znysh7) 8

# **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;

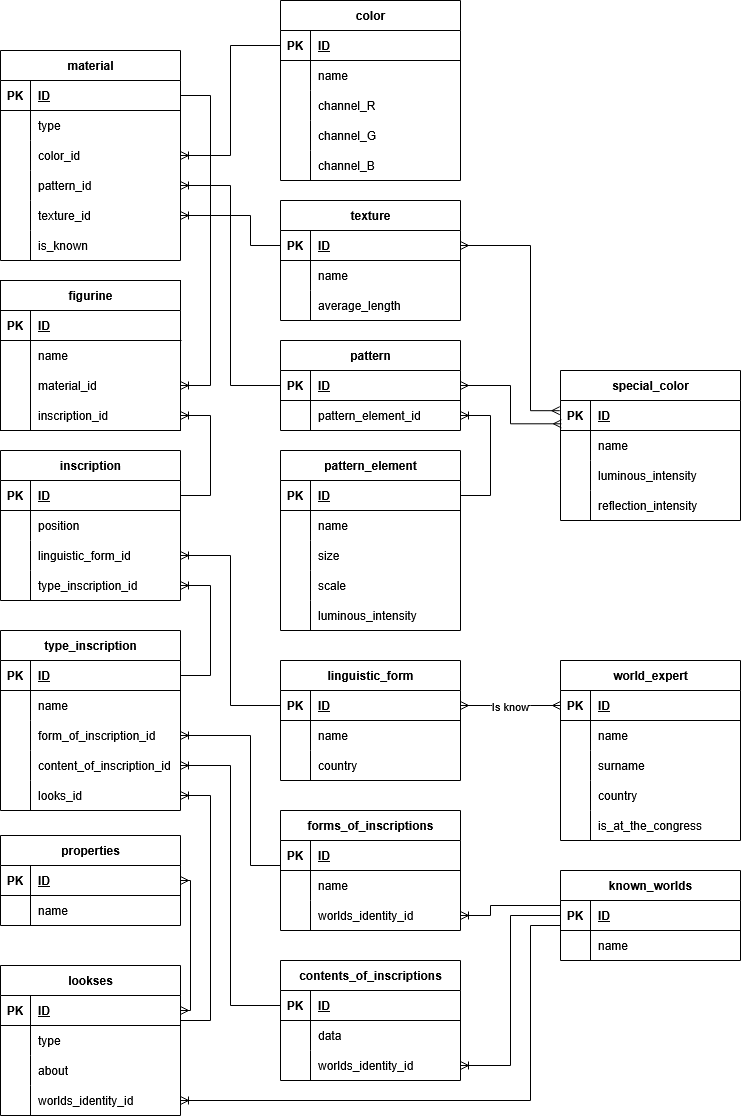
● Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если

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**

**Основные таблицы:**

1. **color**:  
   (id) → (name, channel\_R, channel\_G, channel\_B)
2. **pattern\_element**:  
   (id) → (name, size, scale, luminous\_intensity)
3. **pattern**:  
   (id) → (pattern\_element\_id)
4. **special\_color**:  
   (id) → (name, luminous\_intensity, reflection\_intensity)
5. **texture**:  
   (id) → (name, average\_lenght)
6. **material**:  
   (id) → (type, color\_id, pattern\_id, texture\_id, is\_known)
7. **linguistic\_form**:  
   (id) → (name, county)
8. **world\_expert**:  
   (id) → (name, surname, county, is\_at\_the\_congress)
9. **known\_worlds**:  
   (id) → (name)
10. **forms\_of\_inscriptions**:  
    (id) → (name, worlds\_identity\_id)
11. **contents\_of\_inscriptions**:  
    (id) → (data, worlds\_identity\_id)
12. **property**:  
    (id) → (name)
13. **lookses**:  
    (id) → (type, about, worlds\_identity\_id)
14. **type\_inscription**:  
    (id) → (name, form\_of\_inscriptions\_id, content\_of\_inscription\_id, looks\_id)
15. **inscription**:  
    (id) → (position, linguistic\_form\_id, type\_inscription\_id)
16. **figurine**:  
    (id) → (name, material\_id, inscription\_id)

**Связующие таблицы:**

1. **texture\_to\_special\_color**:  
   (texture\_id, special\_color\_id) → ()
2. **pattern\_to\_special\_color**:  
   (pattern\_id, special\_color\_id) → ()
3. **linguistic\_form\_to\_world\_expert**:  
   (linguistic\_form\_id, world\_expert\_id) → ()
4. **property\_to\_lookses**:  
   (property\_id, lookses\_id) → ()

**Дополнительные зависимости:**

* Для таблицы **material**:  
  (color\_id) → (name) (из таблицы color)  
  (pattern\_id) → (pattern\_element\_id) (из таблицы pattern)  
  (texture\_id) → (name, average\_lenght) (из таблицы texture)
* Для таблицы **type\_inscription**:  
  (form\_of\_inscriptions\_id) → (name, worlds\_identity\_id) (из forms\_of\_inscriptions)  
  (content\_of\_inscription\_id) → (data, worlds\_identity\_id) (из contents\_of\_inscriptions)  
  (looks\_id) → (type, about, worlds\_identity\_id) (из lookses)
* Для таблицы **inscription**:  
  (linguistic\_form\_id) → (name, county) (из linguistic\_form)  
  (type\_inscription\_id) → (name, form\_of\_inscriptions\_id, content\_of\_inscription\_id, looks\_id) (из type\_inscription)
* Для таблицы **figurine**:  
  (material\_id) → (type, color\_id, pattern\_id, texture\_id, is\_known) (из material)  
  (inscription\_id) → (position, linguistic\_form\_id, type\_inscription\_id) (из inscription)

# **Нормальные формы**

● 1NF: Отношение находится в 1NF, если все его атрибуты содержат только атомарные

значения и отсутствуют повторяющиеся группы. Мои отношения удовлетворяет 1NF,

так как все атрибуты атомарны, и нет повторяющихся групп.

● 2NF: Отношение находится в 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

она не содержит 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**

* Добавление вычисляемых полей. В моей схеме можно добавить в figurine поле material\_type, дублирующее material.type, если часто нужно получать материал фигурки. Не нужно делать join к таблице material.
* Объединение связанных таблиц. Если часто нужны данные inscription + linguistic\_form, можно добавить linguistic\_form.name в inscription.
* Создание агрегированных полей. Можно добавить в world\_expert поле known\_forms\_count, если требуется получать количество известных эксперту лингвистических форм

# **Триггер**

* Создаётся табличка топ-5 мировых экспертов
* По добавлению или обновлению в linguistic\_form\_to\_world\_experts нового эксперта:
  + Эксперт добавляется в топ, если он знает 2 или более лингвистических форм
  + При добавлении в топ эксперт автоматически приглашается на конгресс
  + Топ ограничен пятью экспертами, по мере добавления новых из топа удаляются эксперты с наименьшим числом форм и самой поздней активностью (активность - временнАя метка, ставится при изменении эксперта)

-- Таблица для отслеживания топовых экспертов  
CREATE TABLE top\_linguistic\_experts (  
 expert\_id INT PRIMARY KEY,  
 form\_count INT NOT NULL,  
 last\_activity TIMESTAMP NOT NULL  
);  
  
-- Триггерная функция для обновления статуса экспертов  
CREATE OR REPLACE FUNCTION update\_expert\_status()  
 RETURNS TRIGGER AS $$  
DECLARE  
 current\_form\_count INT;  
 expert\_in\_top BOOLEAN;  
BEGIN  
 SELECT COUNT(\*) INTO current\_form\_count -- Получаем количество форм известных эксперту  
 FROM linguistic\_form\_to\_world\_expert  
 WHERE world\_expert\_id = NEW.world\_expert\_id;  
  
 SELECT EXISTS ( -- Проверяем, есть ли эксперт в топе  
 SELECT 1 FROM top\_linguistic\_experts  
 WHERE expert\_id = NEW.world\_expert\_id  
 ) INTO expert\_in\_top;  
  
 IF current\_form\_count >= 2 AND NOT expert\_in\_top THEN -- Если эксперт знает 2+ форм и не в топе - добавляем  
 INSERT INTO top\_linguistic\_experts (expert\_id, form\_count, last\_activity)  
 VALUES (NEW.world\_expert\_id, current\_form\_count, NOW());  
  
 UPDATE world\_expert -- Если эксперт не на конгрессе - добавляем  
 SET is\_at\_the\_congress = TRUE  
 WHERE id = NEW.world\_expert\_id  
 AND is\_at\_the\_congress = FALSE;  
  
 RAISE NOTICE 'Эксперт id=% добавлен в топ и приглашен на конгресс', NEW.world\_expert\_id;  
  
 ELSIF expert\_in\_top THEN -- Если эксперт в топе просто обновляем счетчик  
 UPDATE top\_linguistic\_experts  
 SET form\_count = current\_form\_count, last\_activity = NOW()  
 WHERE expert\_id = NEW.world\_expert\_id;  
 END IF;  
  
 -- Если в топе больше 5 экспертов, сортируем по формам и времени активности и удаляем последнего  
 IF (SELECT COUNT(\*) FROM top\_linguistic\_experts) > 5 THEN  
 DELETE FROM top\_linguistic\_experts  
 WHERE expert\_id IN (  
 SELECT expert\_id FROM top\_linguistic\_experts  
 ORDER BY form\_count ASC, last\_activity ASC  
 LIMIT 1  
 );  
  
 RAISE NOTICE 'Эксперт удален из топа';  
 END IF;  
  
 RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
-- Триггер по вставке или обновлению  
CREATE TRIGGER trigger\_update\_expert\_status  
 AFTER INSERT OR UPDATE ON linguistic\_form\_to\_world\_expert  
 FOR EACH ROW  
EXECUTE FUNCTION update\_expert\_status();

# **Заключение**

В ходе лабораторной работы я изучил принципы нормализации и денормализации, проверил свою базу данных на BCNF и предложил варианты её денормализации, а также освоил работу с триггерами и написал свой.