

Федеральное государственное автономное образовательное учреждение высшего
образования
«Санкт-Петербургский национальный исследовательский университет
ИТМО»

Институт ПИИКТ

Дисциплина: Базы данных

Лабораторная работа №3

Выполнил: Садовой Григорий Владимирович

Группа: Р3107

Преподаватель: : Бострикова

Дарья Константиновна

2023г.

Оглавление

Задание	3
Выполнение	4
Исходная модель	4
Функциональные зависимости	4
Анализ зависимостей	5
1NF	Ошибка! Закладка не определена.
2NF	Ошибка! Закладка не определена.
3NF	Ошибка! Закладка не определена.
НФБК	5
Нормализованная модель:	Ошибка! Закладка не определена.
Денормализации	7
Объединение связанных таблиц	7
Добавление избыточных атрибутов	7
Функция	7
Триггер	9
Вывод	10

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;

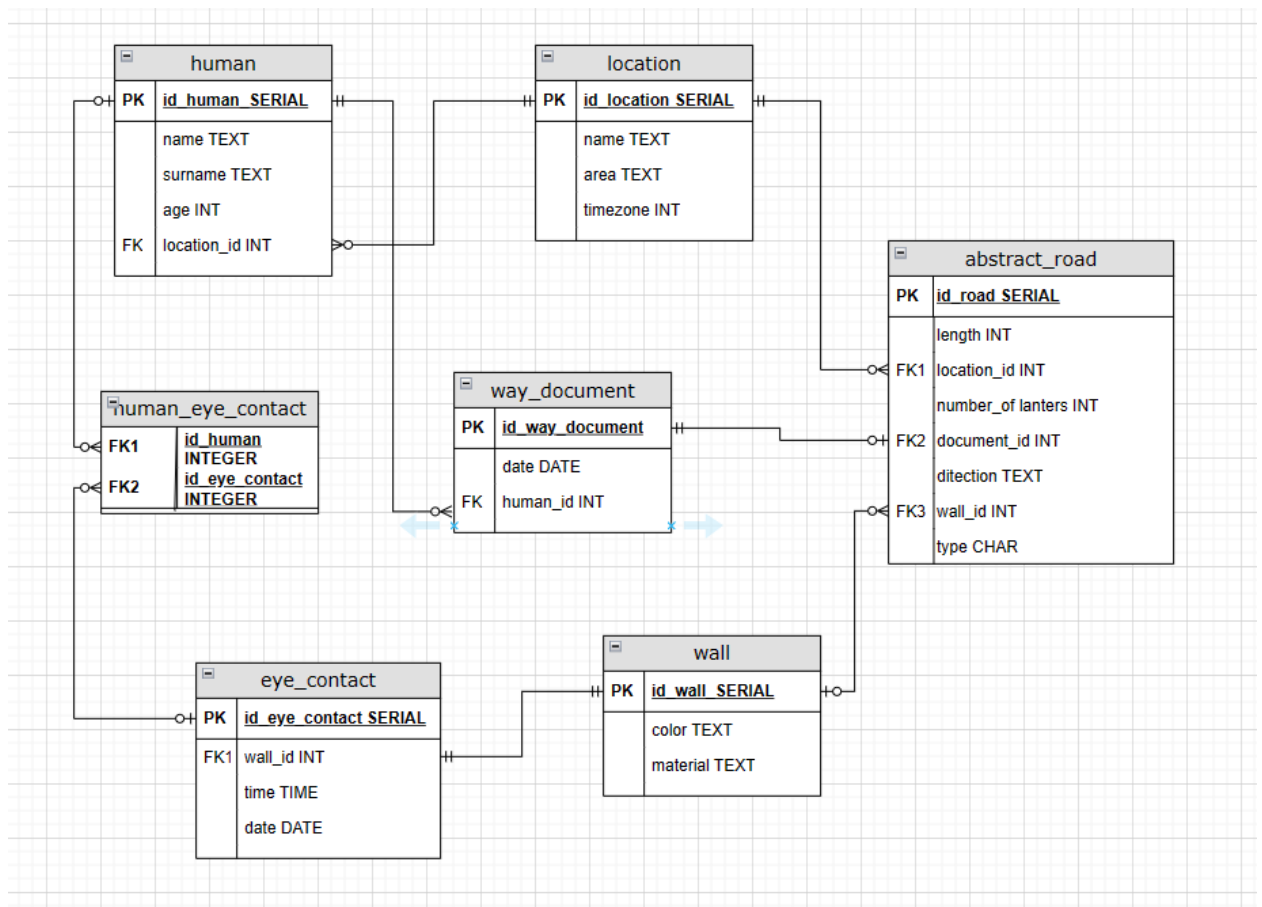
Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если ваша схема находится уже в BCNF, докажите это;

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.

Выполнение

Исходная модель



Функциональные зависимости

human: id \rightarrow (name, surname, age, location_id)
location: id \rightarrow (name, area, timezone)
way_document: id \rightarrow (date, human_id),
wall: id \rightarrow (color, material)
eye_contact: id \rightarrow (wall_id, time, date)
abstract_road: id \rightarrow (length, location_id, number_of_lanterns, document_id, direction, wall_id, type)
human_eye_contact: (human_id, troubles_id) \rightarrow ()

Анализ зависимостей

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

- Нет дублирующихся строк
- Все атрибуты атомарны
- Нет повторяющихся атрибутов с одинаковым смыслом

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.

- Отношения находятся в 1NF
- Есть первичный ключ
- Все неключевые атрибуты функционально зависят от ключа целиком, но не от его части

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.

- Отношения находятся в 2NF
- Неключевые атрибуты напрямую зависят только от ПК, но не от других атрибутов

Моя модель удовлетворяет 3NF, так как все неключевые атрибуты зависят только от первичных ключей, и не содержат транзитивных зависимостей.

НФБК

Отношение в НФБК, когда для всех функциональных зависимостей отношения выполняется условие: детерминант — потенциальный ключ.

Рассматриваемая модель находится в нормальной форме Бойса-Кодда

Денормализации

Объединение связанных таблиц

В некоторых случаях, объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, можно рассмотреть объединение таблиц `abstract_road` и `way_document`, если часто запрашиваются данные о дороге и её происхождении одновременно.

Добавление избыточных атрибутов

В некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, если часто запрашивается количество людей в локации, можно добавить атрибут `human_count` в таблицу `location`. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении или удалении людей с локаций.

Функция

Функция на языке PL/pgSQL для поиска локации с минимальной загруженностью людьми.

```
CREATE OR REPLACE FUNCTION low_population() RETURNS integer AS $$
DECLARE
    min_human integer;
BEGIN
    IF (EXISTS (
        SELECT *
        FROM location
        LEFT JOIN human
        ON human.location_id = location.id
        WHERE human.id IS NULL
    )) THEN
        SELECT location.id INTO min_human
        FROM location
        LEFT JOIN human
        ON human.location_id = location.id
        WHERE human.id IS NULL
        LIMIT 1;
        RETURN min_human;
    ELSE
        SELECT location_id INTO min_human
        FROM human
        GROUP BY human.location_id
        ORDER BY COUNT(*)
        LIMIT 1;
        RETURN min_human;
    END IF;
END;
$$
LANGUAGE plpgsql;
```


Триггер

Я добавил поле `traffic_human` в таблицу `location`, которое отвечает за количество машин на дороге и написал для автоматического обновления этого атрибута триггер

```
CREATE OR REPLACE FUNCTION update_location_traffic()
RETURNS TRIGGER AS $$
BEGIN
    IF (TG_OP = 'INSERT') THEN
        UPDATE location
        SET traffic_human = traffic_human + 1
        WHERE id = NEW.location_id;
    ELSIF (TG_OP = 'DELETE') THEN
        UPDATE location
        SET traffic_human = traffic_human - 1
        WHERE id = OLD.location_id;
    ELSIF (TG_OP = 'UPDATE') THEN
        UPDATE location
        SET traffic_human = traffic_human - 1
        WHERE id = OLD.location_id;
        UPDATE location
        SET traffic_human = traffic_human + 1
        WHERE id = NEW.location_id;

    END IF;
    RETURN NULL;
END
$$
LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER update_location_traffic_trigger
AFTER INSERT OR DELETE OR UPDATE ON human
FOR EACH ROW
EXECUTE FUNCTION update_location_traffic();
```

Вывод

В процессе выполнения работы я узнал про нормализацию, привел мою модель к НФБК и доказал это, написал функцию и триггер.