

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

Факультет программной инженерии и компьютерной техники

Отчёт по лабораторной работе №3

Дисциплина «Базы данных»

Вариант 84711

Выполнил

Разгоняев Максим Витальевич

Группа Р3131

Проверил

Коновалов Арсений Антонович

Санкт-Петербург 2025

Оглавление

Текст задания.....	2
Даталогическая модель (исходная).....	3
Функциональные зависимости (изначальные)	3
Преобразование к 1NF	3
Преобразование к 2NF	3
Преобразование к 3NF	3
Преобразование к BCNF	4
Функциональные зависимости (после преобразований).....	4
Даталогическая модель (после преобразований)	4
Полезные денормализации	4
Даталогическая модель (после денормализации).....	5
Триггер и связанная с ним функция.....	5
Вывод.....	6

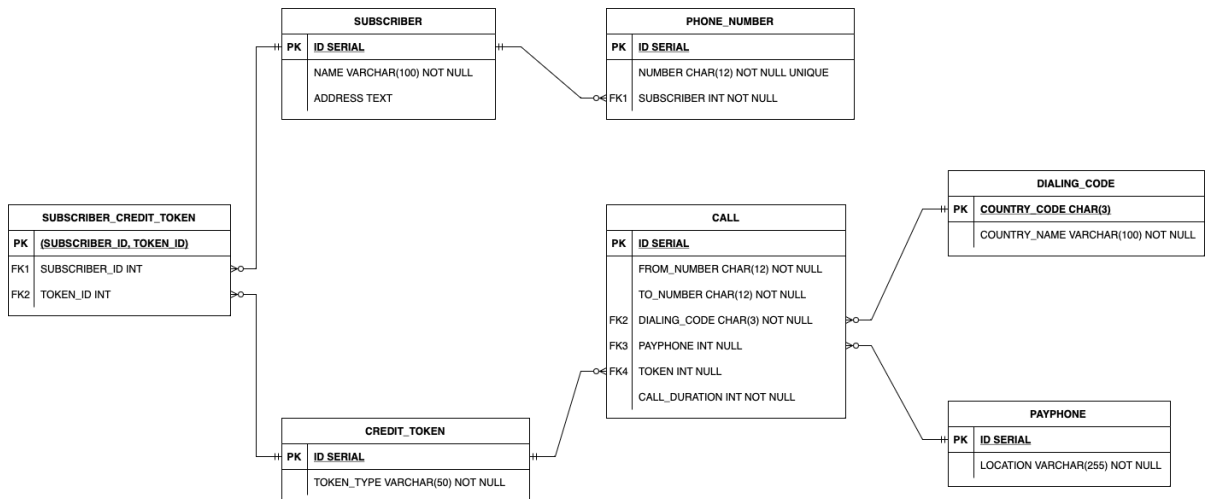
Текст задания

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.

Даталогическая модель (исходная)



Функциональные зависимости (изначальные)

SUBSCRIBER: ID → (NAME, ADDRESS)

PHONE_NUMBER: ID → (NUMBER, SUBSCRIBER), NUMBER → SUBSCRIBER

SUBSCRIBER_CREDIT_TOKEN: (SUBSCRIBER_ID, TOKEN_ID) → ()

CREDIT_TOKEN: ID → TOKEN_TYPE

CALL: ID → (TO_NUMBER, DIALING_CODE, PAYPHONE, TOKEN, CALL_DURATION)

DIALING_CODE: COUNTRY_CODE → COUNTRY_NAME

PAYPHONE: ID → LOCATION

Преобразование к 1NF

Не потребовалось, условие «на пересечении каждой строки и столбца – 1 значение» и так выполняется

Преобразование к 2NF

Не потребовалось, поскольку у всех первичных ключей нет подмножеств, а значит атрибуты всех отношений – в полной функциональной зависимости от соответствующих первичных ключей

Преобразование к 3NF

Не потребовалось, так как ни один неключевой атрибут не зависит от других неключевых атрибутов

Преобразование к BCNF

Нужно преобразовать таблицу PHONE_NUMBER, так как зависимость NUMBER → SUBSCRIBER не удовлетворяет BCNF, потому что NUMBER не является суперключом, только ID является primary key

Для того, чтобы привести таблицу к BCNF, сделаем NUMBER primary key, избавившись от id

Функциональные зависимости (после преобразований)

SUBSCRIBER: ID → (NAME, ADDRESS)

PHONE_NUMBER: NUMBER → SUBSCRIBER

SUBSCRIBER_CREDIT_TOKEN: (SUBSCRIBER_ID, TOKEN_ID) → ()

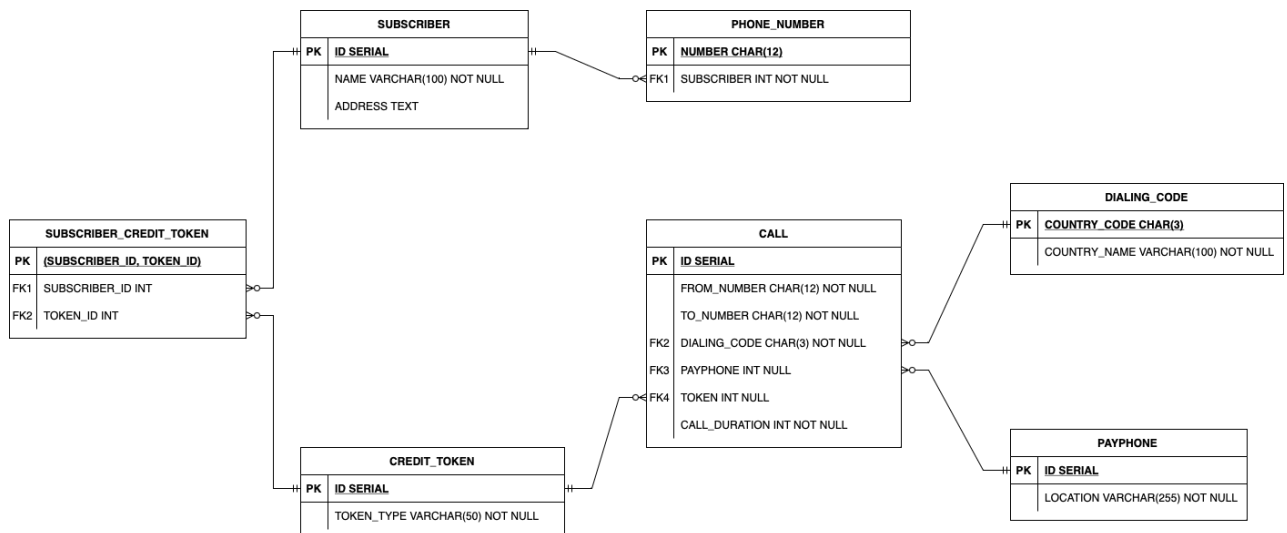
CREDIT_TOKEN: ID → TOKEN_TYPE

CALL: ID → (TO_NUMBER, DIALING_CODE, PAYPHONE, TOKEN, CALL_DURATION)

DIALING_CODE: COUNTRY_CODE → COUNTRY_NAME

PAYPHONE: ID → LOCATION

Даталогическая модель (после преобразований)

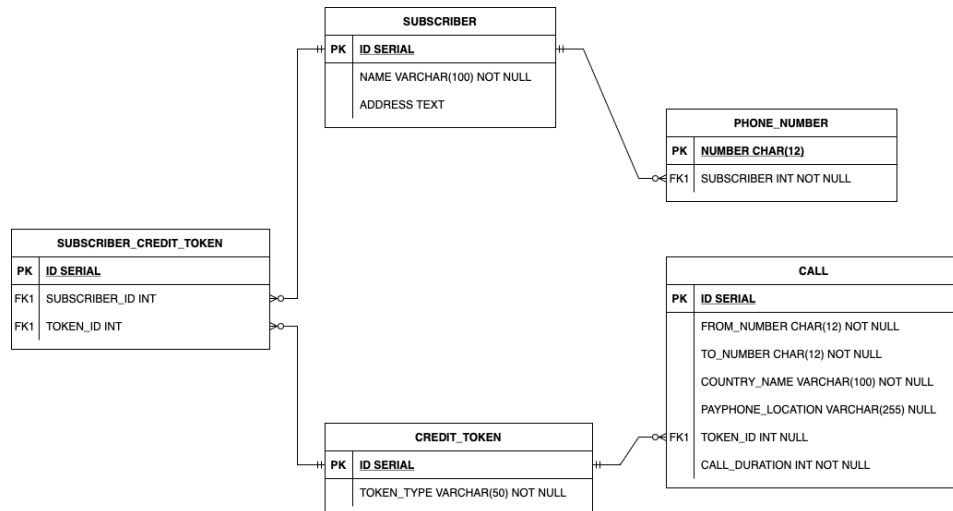


Полезные денормализации

Теоретически, можно увеличить скорость выполнения некоторых запросов к нашей бд за счёт сокращения количества сущностей, а именно — уничтожения некоторых

характеристических сущностей и добавления их атрибутов (кроме id, естественно) в стержневые сущности, но при этом увеличится избыточность данных

Даталогическая модель (после денормализации)



Триггер и связанная с ним функция

```

CREATE OR REPLACE FUNCTION validate_call_from_number_and_token()
RETURNS TRIGGER AS $$
BEGIN
    IF NOT EXISTS (
        SELECT 1
        FROM PHONE_NUMBER
        WHERE NUMBER = NEW.FROM_NUMBER
    ) THEN
        RAISE EXCEPTION 'Invalid FROM_NUMBER: % is not registered in PHONE_NUMBER', NEW.FROM_NUMBER;
    END IF;

    IF NEW.TOKEN IS NOT NULL THEN
        IF NOT EXISTS (
            SELECT 1
            FROM PHONE_NUMBER PN
            JOIN SUBSCRIBER_CREDIT_TOKEN SCT ON PN.SUBSCRIBER = SCT.SUBSCRIBER_ID
            WHERE PN.NUMBER = NEW.FROM_NUMBER AND SCT.TOKEN_ID = NEW.TOKEN
        ) THEN
            RAISE EXCEPTION 'Token ID % is not assigned to the subscriber for phone number %', NEW.TOKEN, NEW.FROM_NUMBER;
        END IF;
    END IF;

    RETURN NEW;
END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER check_call_from_number_and_token
BEFORE INSERT OR UPDATE ON CALL
FOR EACH ROW
EXECUTE FUNCTION validate_call_from_number_and_token();
    
```

Функция проверяет при заполнении или изменении полей в таблице CALL то, что номер, с которого звонят, зарегистрирован в таблице PHONE_NUMBER, а также проверяет,

чтобы токен, который используют для вызова, принадлежал именно номеру, который делает звонок

Вывод

При выполнении данной лабораторной работы я узнал, что из себя представляет функциональная зависимость в базах данных, познакомился с сопутствующей терминологией. Также я выяснил, как находить функциональные зависимости в отношениях и как приводить отношения к 1NF, 2NF, 3NF и BCNF. Изучил функции и триггеры и их синтаксис в PL/pgSQL