

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

Факультет программной инженерии и компьютерной техники

# Отчёт

по лабораторной работе №1

вариант 1298

Выполнил: Тимошкин Р. В., группа Р3131

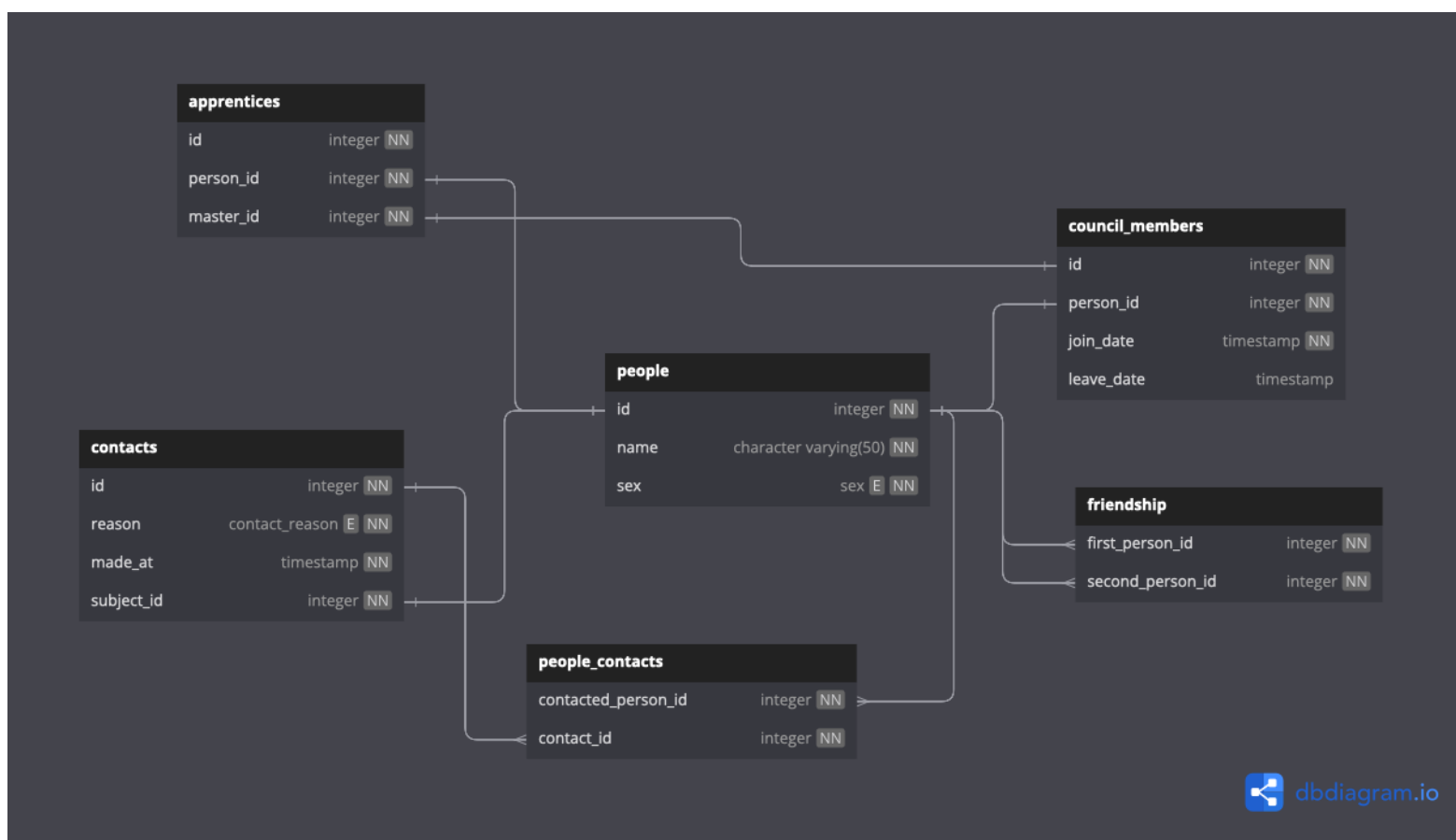
Преподаватель: Харитонов А. Е.

# 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 основе NF (как минимум).
- Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе NF;
- 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

- apprentices: id → (person\_id, master\_id)
- council\_members : id → (person\_id, join\_date, leave\_date)
- contacts: id → (reason, made\_at, subject\_id)
- people: id → (name, sex)
- friendship: (first\_person\_id, second\_person\_id) → ()

- people\_contacts: (contacted\_person\_id, contact\_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  $X \rightarrow Y$ ,  $X$  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 people и apprentices, если часто запрашиваются данные о человеке и его наставнике одновременно.

**Добавление избыточных атрибутов:** в некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, если часто запрашивается количество коллег отдельных людей, можно добавить атрибут friends\_count в таблицу people. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении или удалении людей.

## Функция на языке PL/pgSQL

```
CREATE OR REPLACE FUNCTION total_contacts(person1_id int, person2_id int)
    RETURNS int AS
$amount$
DECLARE
    amount          int := 0;
    first_second    int;
    second_first    int;
BEGIN
    IF person1_id IS NULL OR person2_id IS NULL THEN
        RAISE EXCEPTION 'Person id is null';
    END IF;

    SELECT COUNT(p.id)
    INTO first_second
    FROM contacts c
        JOIN people_contacts pc ON pc.contact_id = c.id
        JOIN people p ON pc.contacted_person_id = p.id
    WHERE c.subject_id = person1_id
        AND p.id = person2_id
    GROUP BY p.id;
    IF NOT FOUND THEN
        first_second := 0;
    END IF;

    SELECT COUNT(p.id)
    INTO second_first
    FROM contacts c
        JOIN people_contacts pc ON pc.contact_id = c.id
        JOIN people p ON pc.contacted_person_id = p.id
```

```

WHERE c.subject_id = person2_id
      AND p.id = person1_id
GROUP BY p.id;
IF NOT FOUND THEN
    second_first := 0;
END IF;

amount := first_second + second_first;

RETURN amount;
END;
$amount$ LANGUAGE plpgsql;

CREATE OR REPLACE FUNCTION update_total_contacts() RETURNS trigger AS
$update_total_contacts$
DECLARE
    subject_id    int;
    contacted_id  int;
BEGIN
    NEW.total_contacts := NULL;

    IF NEW.contact_id IS NULL THEN
        RAISE EXCEPTION 'Contact id is null';
    END IF;
    IF NEW.contacted_person_id IS NULL THEN
        RAISE EXCEPTION 'Contacted person id is null';
    END IF;

    SELECT c.subject_id INTO subject_id FROM contacts c WHERE c.id = NEW.contact_id;
    SELECT p.id INTO contacted_id FROM people p WHERE p.id = NEW.contacted_person_id;

    NEW.total_contacts := total_contacts(subject_id, contacted_id) + 1;
    RETURN NEW;
END;
$update_total_contacts$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER update_total_contacts_trigger
BEFORE INSERT OR UPDATE
ON people_contacts
FOR EACH ROW
EXECUTE FUNCTION update_total_contacts();

```

## Вывод

При выполнении лабораторной работы я познакомился с понятием нормализации и денормализации. Научился определять функциональные зависимости модели, а также анализировать последнюю на соответствие различным нормальным формам. Познакомился с процедурным языком PL/pgSQL. Изучил эффективные способы денормализации схемы базы данных и ситуации, в которых возможно их применение.