

Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №408409  
Лабораторная работа №3  
По дисциплине:  
Базы Данных

Выполнил студент группы Р3130:

Гаврилин Олег Сергеевич

Преподаватель:

Шешуков Дмитрий Михайлович

Санкт-Петербург

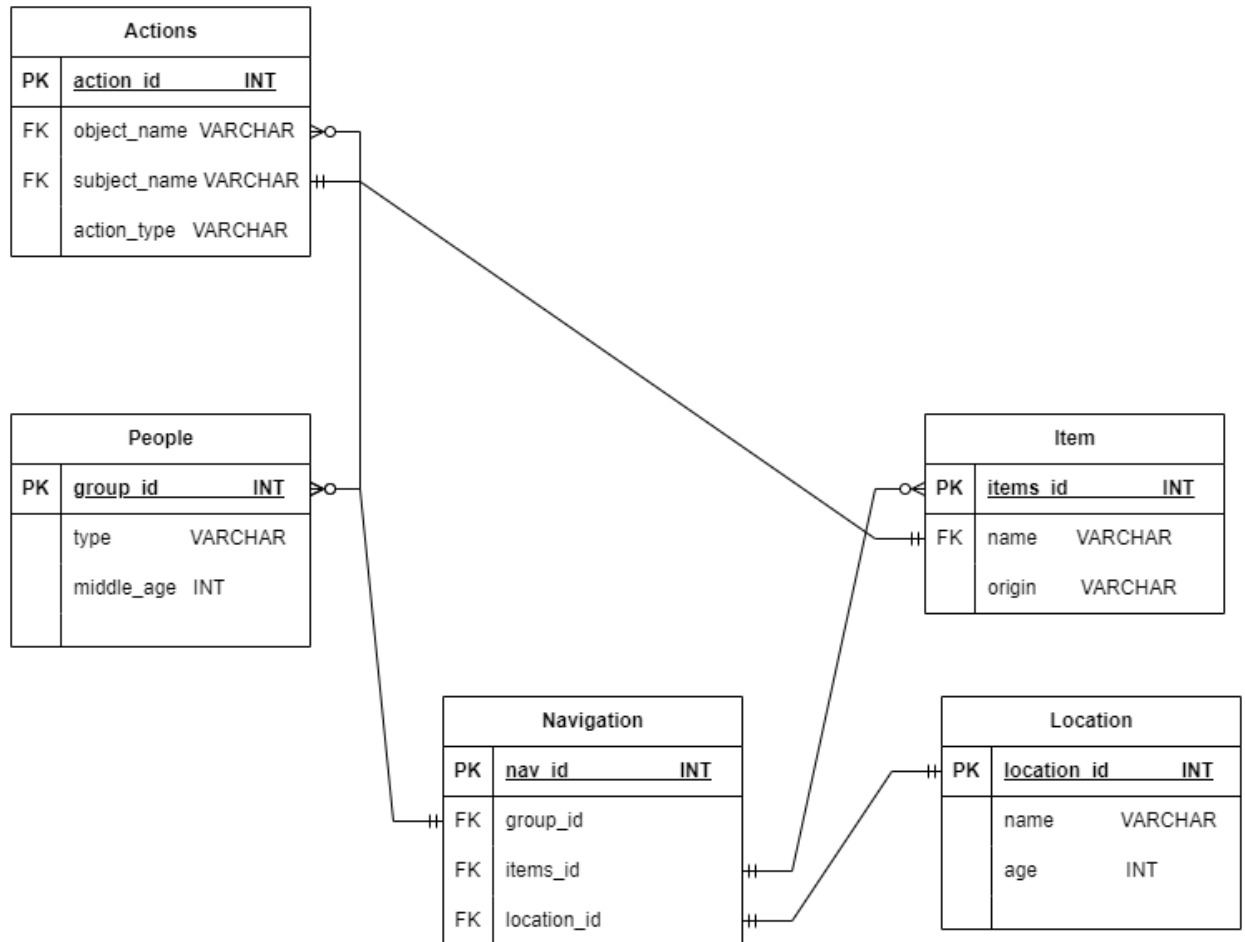
2024 г.

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

Actions: action\_id → object\_name, subject\_name, action\_type

People: group\_id → type, middle\_age

Navigation: nav\_id → group\_id, items\_id, location\_id

Item: items\_id → name, origin

Location: location\_id → name, age

## 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: Расширение третьей нормальной формы, в которой любой неключевой атрибут не зависит от других неключевых атрибутов.

## Денормализация

Денормализация — это процесс обратный нормализации, заключающийся в намеренном объединении данных из нескольких нормализованных таблиц в одну таблицу для улучшения производительности системы, особенно в операциях чтения данных. Основная цель денормализации — уменьшить количество соединений (joins) между таблицами при выполнении запросов, что может существенно повысить скорость выполнения запросов в базе данных.

## Примеры денормализации:

Например, если у вас есть таблица с информацией о продажах и вы хотите рассчитать общую сумму продаж для каждого заказа, вы можете добавить столбец для хранения этой общей суммы в каждой строке заказа, чтобы избежать вычислений на лету при запросах.

## Пример функции:

Функция обновляет поле “middle\_age” в таблице “People”

```
CREATE OR REPLACE FUNCTION update_middle_age()  
RETURNS TRIGGER AS $$  
BEGIN  
    UPDATE People  
    SET middle_age = (  
        SELECT AVG(age)  
        FROM Location  
        JOIN Navigation ON Navigation.location_id = Location.location_id  
        WHERE Navigation.group_id = NEW.group_id
```

```

    )

    WHERE group_id = NEW.group_id;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

```

## Триггер

Триггер вызывает выше описанную функцию при каждом добавлении новой записи в таблицу "Actions"

-- Создание функции для обновления среднего возраста

```

CREATE OR REPLACE FUNCTION update_middle_age()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE People
    SET middle_age = (
        SELECT AVG(Location.age)
        FROM Location
        JOIN Navigation ON Navigation.location_id = Location.location_id
        WHERE Navigation.group_id = (
            SELECT group_id
            FROM Navigation
            JOIN Actions ON Actions.object_name =
Navigation.group_id::varchar
            WHERE Actions.action_id = NEW.action_id
        )
    )
)

WHERE group_id = (
    SELECT group_id
    FROM Navigation
    JOIN Actions ON Actions.object_name = Navigation.group_id::varchar
    WHERE Actions.action_id = NEW.action_id
);

RETURN NEW;

END;

```

```
$$ LANGUAGE plpgsql;  
  
-- Создание триггера для таблицы Actions  
  
CREATE TRIGGER trigger_update_middle_age  
AFTER INSERT ON Actions  
FOR EACH ROW  
  
EXECUTE FUNCTION update_middle_age();
```

## 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.