

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

Факультет Программной Инженерии и Компьютерной Техники
Дисциплина: «Базы данных»

ОТЧЁТ

по лабораторной работе №3
Вариант -

Выполнил:

Студент группы Р3108

Грищенко Андрей Викторович

Проверил:

Афанасьев Дмитрий Борисович

Санкт-Петербург

2024 г.

1. Оглавление

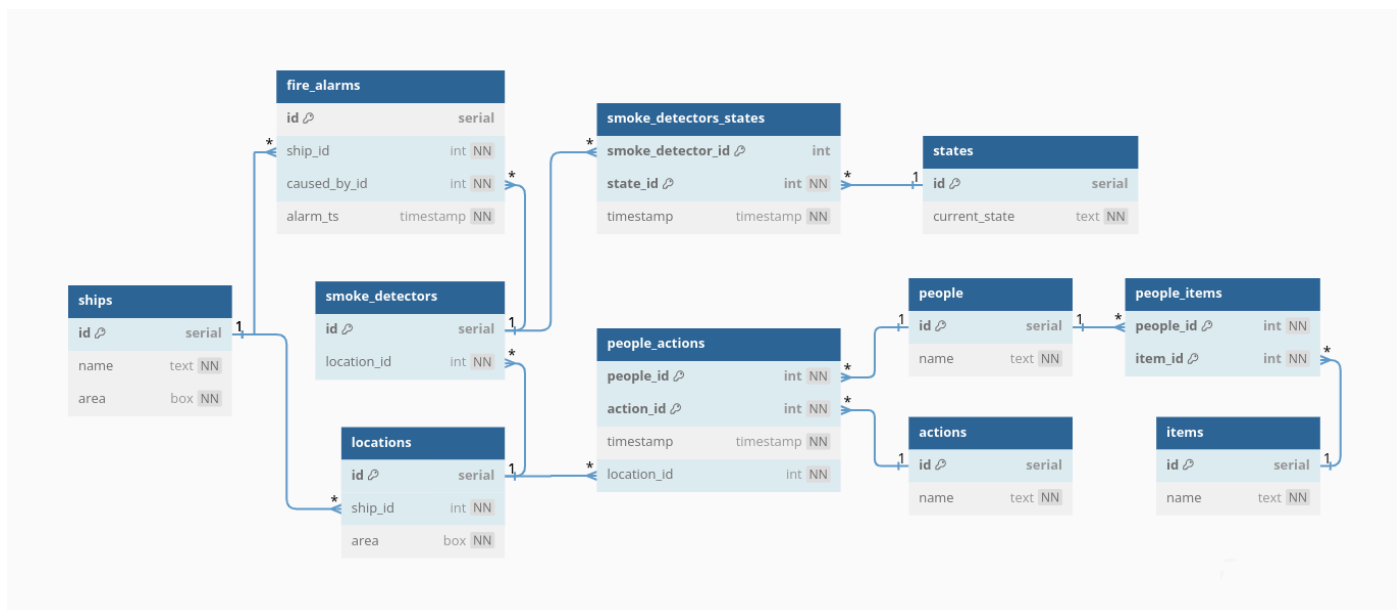
1. Оглавление.....	2
2. Текст задания.....	2
3. Функциональные зависимости.....	3
4. Нормализация.....	3
5. Возможная денормализация.....	4
6. Триггер и связанная с ним функция.....	4
7. Вывод.....	5

2. 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 4NF (как минимум).
- Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе 4NF;
- 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.



3. Функциональные зависимости

ships: (id) -> (name, area)

locations: (id) -> (ship_id, area)

actions: (id) -> (name)

states: (id) -> (current_state)

people: (id) -> (name)

people_actions: (people_id, action_id) -> (timestamp, location_id)

items: (id) -> (name)

people_items: (people_id, item_id) -> ()

smoke_detectors: (id) -> (location_id)

smoke_detectors_states: (smoke_detector_id, state_id) -> ()

fire_alarms: (id) -> (ship_id, caused_by_id, alarm_ts)

4. Нормализация

1NF (Первая нормальная форма):

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

Модель удовлетворяет 1NF, так как все атрибуты во всех таблицах являются атомарными, и нет повторяющихся групп атрибутов.

2NF (Вторая нормальная форма):

Отношение находится в 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа (без частичных зависимостей).

Модель удовлетворяет 2NF, так как все неключевые атрибуты во всех таблицах полностью функционально зависят от своих первичных ключей. Нет частичной зависимости от составного ключа.

3NF (Третья нормальная форма):

Отношение находится в 3NF, если оно находится в 2NF и не содержит транзитивных зависимостей.

Модель удовлетворяет 3NF, так как все неключевые атрибуты во всех таблицах полностью функционально зависят от своих первичных ключей.

BCNF (Нормальная форма Бойса-Кодда)

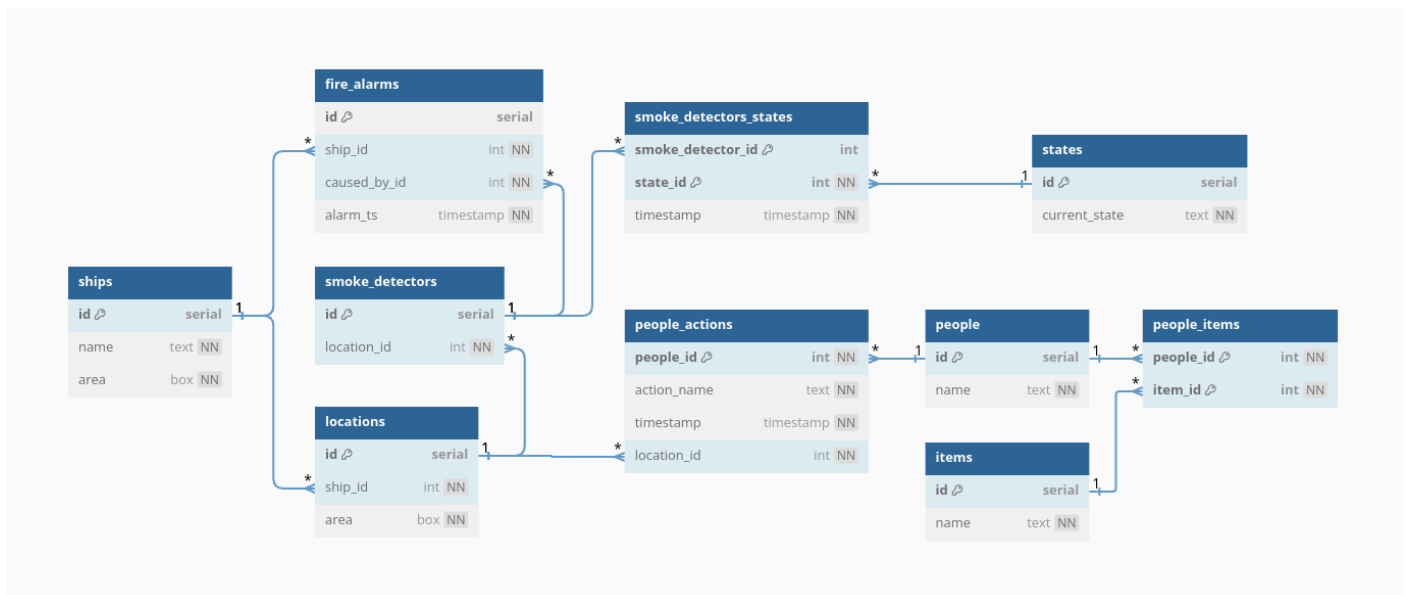
Отношение находится в BCNF, если оно находится в 3NF, и для каждой зависимости вида $X \rightarrow Y$, X является суперключом (т.е. может использоваться для однозначной идентификации строки) или Y является частью составного ключа.

Модель удовлетворяет BCNF, так как для всех функциональных зависимостей X является суперключом.

5. Возможная денормализация

Дублирование данных: иногда дублирование данных может улучшить производительность запросов, но я считаю что это может только создать аномалии, поэтому этого здесь не требуется

Объединение таблиц: иногда объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, можно объединить таблицы 'people_actions' и 'actions'



6. Триггер и связанная с ним функция

Функция будет автоматически создавать запись в таблице 'fire_alarms' когда в таблице 'smoke_detectors_states' появляется состояние "triggered".

```
CREATE OR REPLACE FUNCTION trigger_fire_alarm()
RETURNS TRIGGER AS $$
DECLARE
    ship_id int;
BEGIN
    IF (SELECT current_state FROM states WHERE id = NEW.state_id) = 'triggered' THEN
        SELECT l.ship_id INTO ship_id
        FROM locations l
        JOIN smoke_detectors sd ON l.id = sd.location_id
        WHERE sd.id = NEW.smoke_detector_id;

        INSERT INTO fire_alarms (ship_id, caused_by_id, alarm_ts)
        VALUES (ship_id, NEW.smoke_detector_id, NEW.timestamp);
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER after_smoke_detector_state_insert
AFTER INSERT ON smoke_detectors_states
FOR EACH ROW
EXECUTE FUNCTION trigger_fire_alarm();
```

7. Вывод

В результате проделанной лабораторной работы, я изучил работу функций и триггеров в PostgreSQL, узнал какие есть нормальные формы, а также привел модель к BCNF, реализовал триггер и связанную с ним функцию.