

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

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3
Вариант 466495

Выполнил:
Лежнев Никита Сергеевич
Группа Р3112
Проверил:
Кустарев Иван Павлович

Санкт-Петербург 2025г.

Содержание:

Задание	3
Функциональные зависимости	3
Нормальные формы	4
BCNF	4
Денормализация	4
Триггер и связанная с ним функция	5

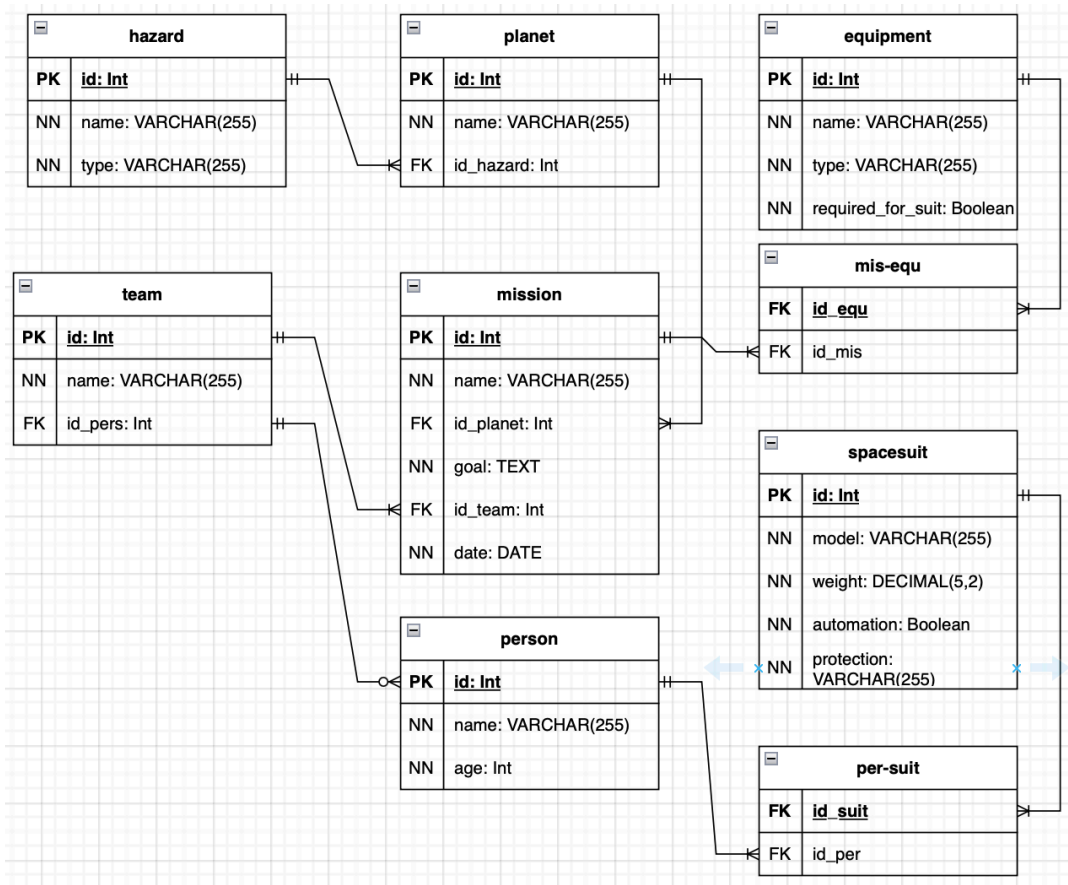
Задание

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 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.

Изначальная таблица:



Функциональные зависимости

hazard:

id → name, type

planet:

id → name, id_hazard

equipment:

id -> name, type, required_for_suit

team:

id -> name, id_pers

person:

id -> name, age

mission:

id -> name, id_planet, goal, id_team, date

spacesuit:

id -> model, weight, automation, protection

Нормальные формы

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.

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 зависят от первичных ключей.

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 $X \rightarrow Y$ выполняется условие, что X является суперключом.

Моя модель удовлетворяет **BCNF**, так как для всех её функциональных зависимостей X выступает в роли суперключа.

Денормализация

1. Рассмотрим таблицу mission:

Сейчас, чтобы получения названия планеты и типа опасности нужно соединять три таблицы: mission, planet и hazard. Это очень затратно и неэффективно, если такие запросы выполняются часто. Чтобы решить данную проблему можно добавить в mission поля planet_name и hazard_type. Благодаря этому данные о планете и её опасностях будут храниться прямо в записи миссии, что исключает необходимость JOIN. Однако я бы отметил минус: при изменении названия планеты или типа опасности нужно обновлять все связанные миссии. Чтобы не возникло проблем можно написать триггеры, которые будут как бы следить за изменениями в таблицах planet и hazard и синхронизировать данные.

2. Также рассмотрим таблицу person:

В этой таблице проблема может возникнуть при получении информации о скафандре, который использует человек. Чтобы получить информацию требуется соединить таблицы person, person_suit и spacesuit. Чтобы ускорить запросы, можно добавить в person поля suit_model и suit_protection. То есть не нужно писать JOIN, но потребует обновления данных при смене скафандра.

Триггер и связанная с ним функция

```
CREATE OR REPLACE FUNCTION validate_mission_date()
RETURNS TRIGGER AS $$
BEGIN
    IF NEW.date IS NOT NULL AND NEW.date < CURRENT_DATE THEN
        RAISE EXCEPTION 'Дата миссии "%" не может быть в прошлом!', NEW.date;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
-----
CREATE TRIGGER before_insert_or_update_mission
BEFORE INSERT OR UPDATE ON mission
FOR EACH ROW
EXECUTE FUNCTION validate_mission_date();
-----
INSERT INTO mission (name, id_planet, goal, date)
VALUES ('Марс-2025', 1, 'Исследование поверхности', '2025-05-20');

INSERT INTO mission (name, id_planet, goal, date)
VALUES ('Луна-2020', 2, 'Поиск воды', '2020-01-01');
```