

# НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 - Программная инженерия

Дисциплина - «Базы данных»

## Лабораторная работа №3

### Вариант – ?

Студент – Мухсинов  
Сардорбек Пулатович

Преподаватель –

Инячина Д.А

Группа - Р3117

Санкт-Петербург, 2023г

# Задание:

Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

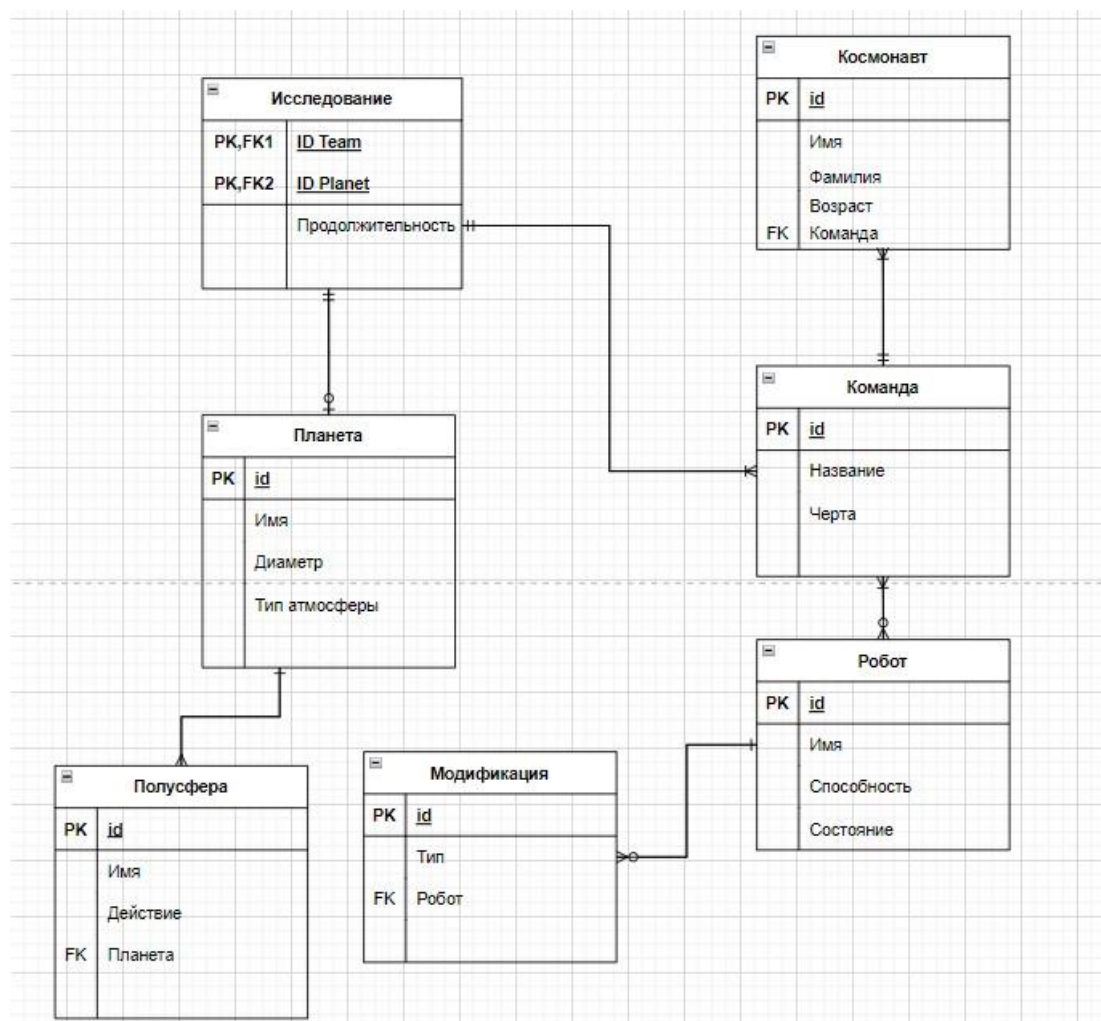
- опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
- приведите отношения в 3NF (как минимум). Постройте схему на основе 3NF (как минимум). Постройте схему на основе полученных отношений;
- опишите изменения в функциональных зависимостях, произошедшие после преобразования 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.

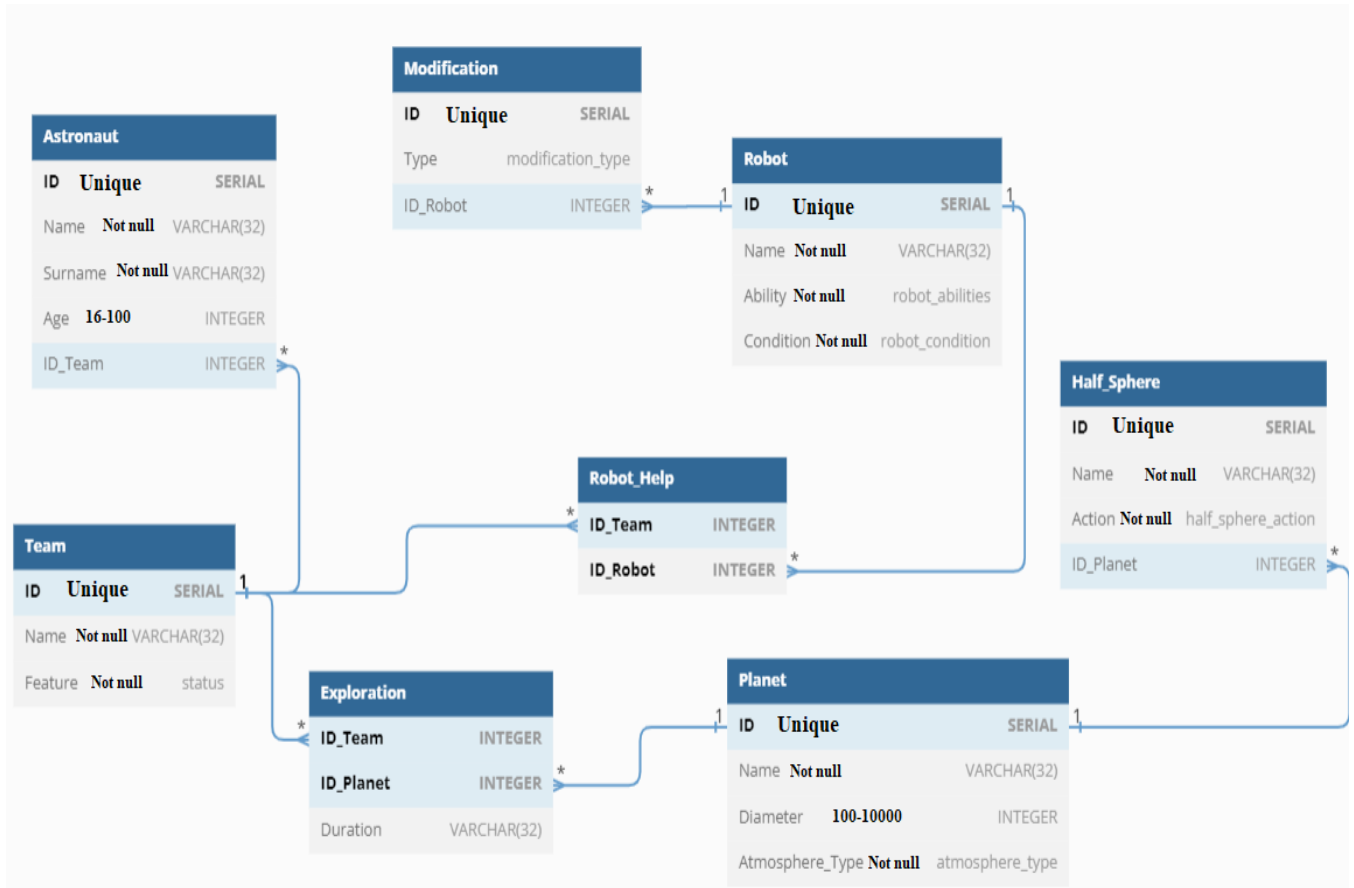
Какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

Придумайте функцию, связанную с вашей предметной областью, согласуйте ее с преподавателем и реализуйте на языке PL/pgSQL.

## Инфологическая модель:



## Даталогическая модель:



## Функциональные зависимости:

- Team:  $id \rightarrow (Name, Feature)$
- Astronaut:  $id \rightarrow (Name, Surname, Age, ID\_Team)$
- Planet:  $id \rightarrow (Name, Diameter, atmosphere\_type)$
- Robot:  $id \rightarrow (Name, Ability, Condition)$
- Half\_Sphere:  $id \rightarrow (Name, Action, ID\_Planet)$
- Modification:  $id \rightarrow (Type, ID\_Robot)$
- Exploration:  $(ID\_Team, ID\_Planet) \rightarrow Duration$

## **Первая нормальная форма:**

В моей модели на пересечении каждой строки и столбца — **одно** значение и нет строк-дубликатов, следовательно, условие 1NF выполняется.

## **Вторая нормальная форма:**

Моя модель удовлетворяет 1NF и в ней все атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичных ключей, следовательно, она удовлетворяет 2NF.

## **Третья нормальная форма:**

Моя модель удовлетворяет 2NF и в ней нет зависимостей одних неключевых атрибутов от других, следовательно атрибуты не могут находиться в транзитивной функциональной зависимости от первичных ключей, следовательно, модель удовлетворяет 3NF.

## **Нормальная форма Бойса-Кодда:**

В моей модели все детерминанты являются потенциальными ключами, следовательно, модель удовлетворяет BCNF.

## **Денормализация:**

К примеру, можно рассмотреть объединение таблиц Робот и Модификация. Таким образом, сущность Робот получит дополнительный атрибут — Модификация. Объединение двух таблиц даст несущественный прирост производительности, но нарушит первую нормальную форму, так как один робот может иметь несколько модификаций.

К тому же можно добавить избыточные атрибуты, которые могут быть полезны. Например, в таблицу Robot можно добавить атрибут `modification_count`, который будет содержать количество модификаций конкретного робота. Это уменьшит количество операций подсчёта при каждом запросе, но придется обновлять значение данного атрибута при добавлении или удалении модификаций, что можно реализовать с помощью функций и триггеров.

## Функция:

```
CREATE OR REPLACE FUNCTION new_modification_function() RETURNS TRIGGER AS
$$
DECLARE
    robot_id int = NEW.id_robot;
BEGIN
    UPDATE robot
    SET modification_count = modification_count + 1
    WHERE id = robot_id;
    return NEW;
end;
$$LANGUAGE plpgsql;
CREATE OR REPLACE FUNCTION deleted_modification_function() RETURNS TRIGGER AS
$$
DECLARE
    robot_id int = OLD.id_robot;
BEGIN
    UPDATE robot
    SET modification_count = modification_count - 1
    WHERE id = robot_id;
    return OLD;
end;
$$LANGUAGE plpgsql;
CREATE OR REPLACE FUNCTION updated_modification_function() RETURNS TRIGGER AS
$$
DECLARE
    new_robot_id int = NEW.id_robot;
    old_robot_id int = OLD.id_robot;
BEGIN
    UPDATE robot
    SET modification_count = modification_count - 1
    WHERE id = old_robot_id;
    UPDATE robot
    SET modification_count = modification_count + 1
    WHERE id = new_robot_id;
    return NEW;
end;
$$LANGUAGE plpgsql;
CREATE TRIGGER new_modification_trigger AFTER INSERT ON modification FOR EACH
ROW EXECUTE PROCEDURE new_modification_function();
CREATE TRIGGER deleted_modification_trigger AFTER DELETE ON modification FOR
EACH ROW EXECUTE PROCEDURE deleted_modification_function();
CREATE TRIGGER updated_modification_trigger AFTER UPDATE ON modification FOR
EACH ROW EXECUTE PROCEDURE updated_modification_function;
```

**Вывод:**

В процессе выполнения лабораторной работы