

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»**



ITMO UNIVERSITY

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №3
по дисциплине
«БАЗЫ ДАННЫХ»:

Вариант 1383

Выполнил:

Ястребов-Амирханов Алекси

ГРУППА: Р3132

ПРЕПОДАВАТЕЛЬ: Афанасьев Дмитрий Борисович

Санкт-Петербург,
2024

Оглавление

Задание	2
Выполнение	3
Исходная модель	3
Функциональные зависимости	4
Анализ зависимостей.....	5
1NF	5
2NF	5
3NF	5
НФБК	6
Денормализация	6
Хранение агрегированных данных:	6
Дублирование данных для избежания операций JOIN:.....	6
Создание функции.....	6
Создание триггера.....	7
Вывод	7

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 основеNF (как минимум).

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

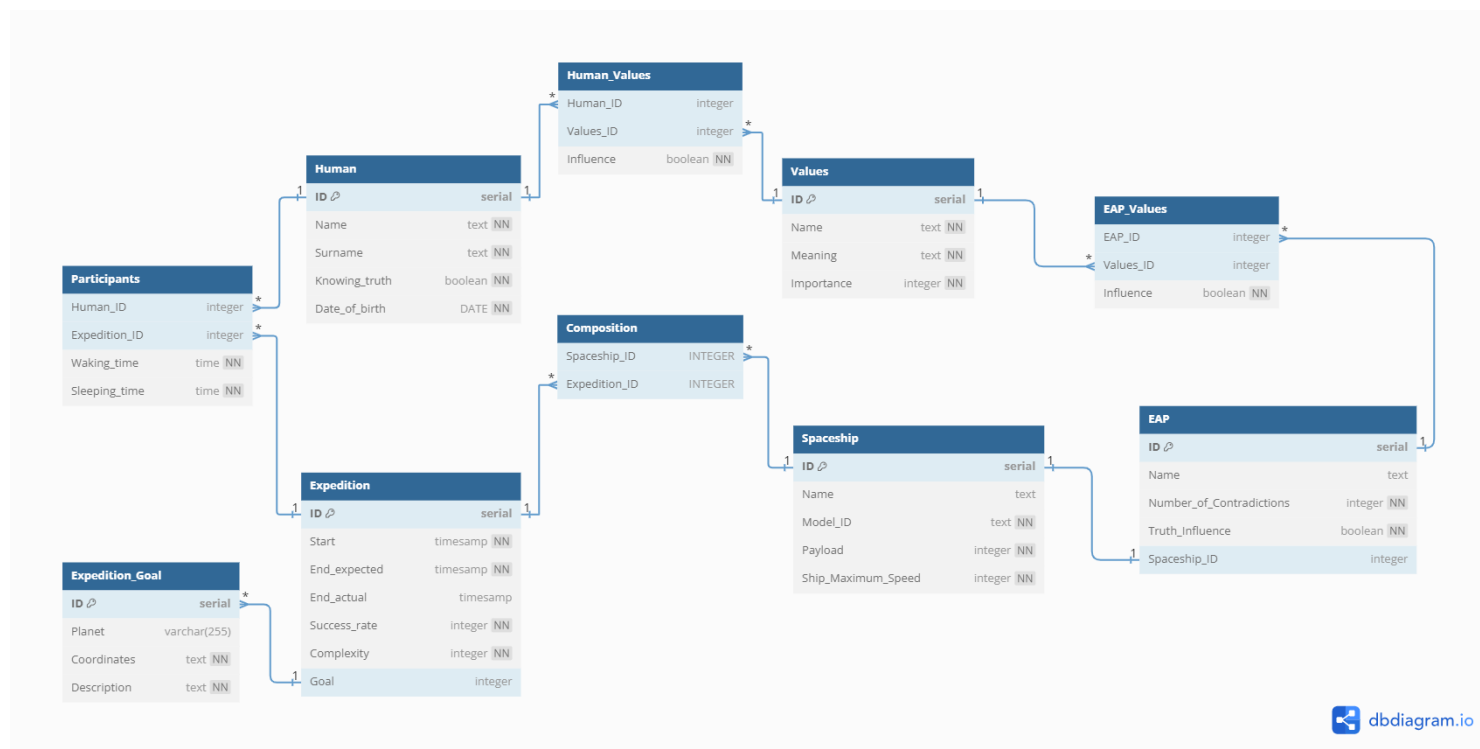
Преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF. Если ваша схема находится уже в BCNF, докажите это;

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.

Выполнение

Исходная модель



Функциональные зависимости

1. Expedition_Goal

Функциональные зависимости:

ID → (Planet, Description)

2. Expedition

Функциональные зависимости:

ID → (Start, End_expected, End_actual, Success_rate, Complexity, Goal)

3. Human

Функциональные зависимости:

ID → (Name, Surname, Knowing_truth, Date_of_birth)

4. Participants

Функциональные зависимости:

(Human_ID, Expedition_ID) → (Waking_time, Sleeping_time)

5. Values

Функциональные зависимости:

ID → (Name, Meaning, Importance)

6. Human_Values

Функциональные зависимости:

(Human_ID, Value_ID) → (Influence)

7. Spaceship

Функциональные зависимости:

ID → (Name, Model)

Model → (Payload, Ship_Maximum_Speed)

8. EAP

Функциональные зависимости:

ID → (Name, Number_of_Contradictions, Truth_Influence, Spaceship_ID)

9. EAP_Values

Функциональные зависимости:

(EAP_ID, Value_ID) → (Influence)

Анализ зависимостей

1NF

Отношение, на пересечении каждой строки и столбца — одно значение.
Представленная модель удовлетворяет первой нормальной форме

2NF

1) отношение в 1НФ

2) атрибуты, не входящие в первичный ключ, в полной функциональной зависимости от первичного ключа отношения.

Представленная модель удовлетворяет второй нормальной форме

3NF

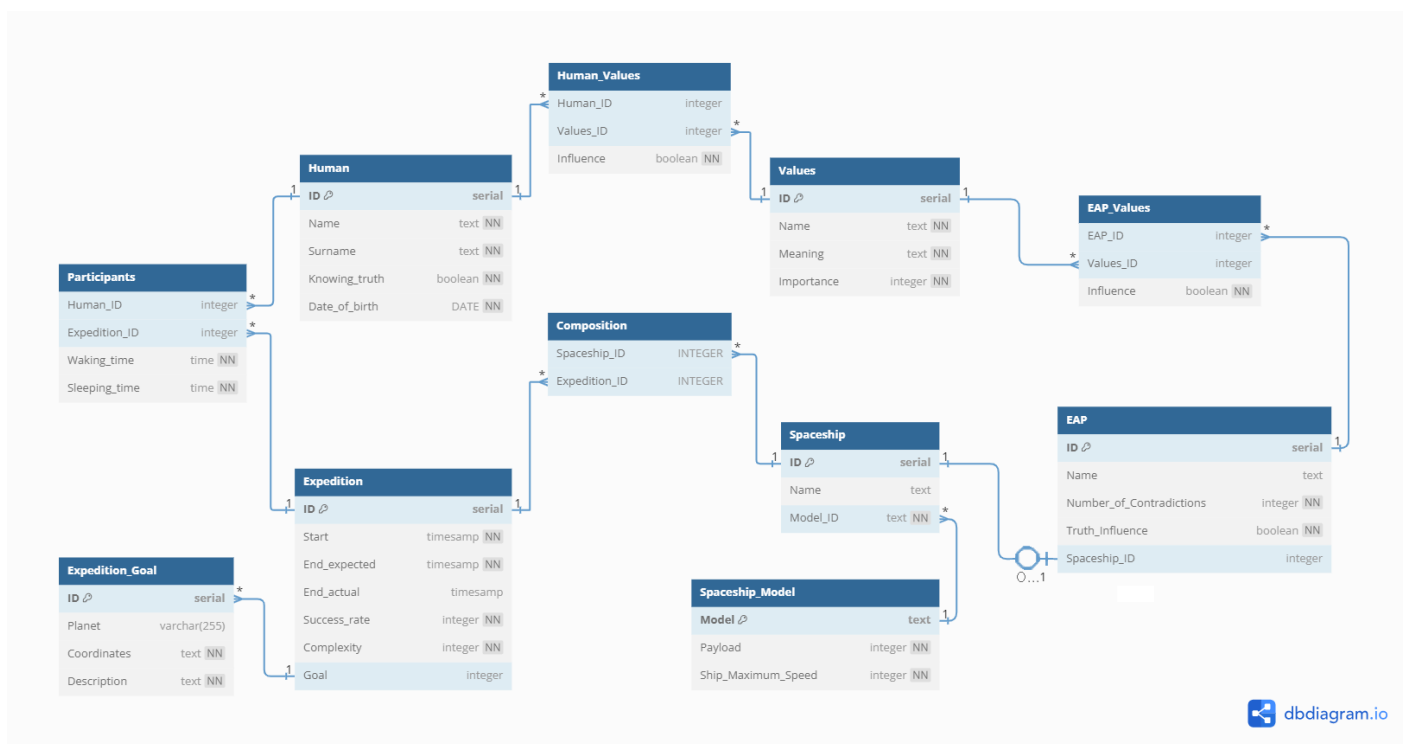
Отношение в

1) 1НФ и 2НФ

2) все атрибуты, которые не входят в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа.

Таблица Spaceship не удовлетворяет третьей нормальной форме

Новая модель:



1. Spaceship

Функциональные зависимости:

ID → (Name, Model)

2. Spaceship_Model

Функциональные зависимости:

Model → (Payload, Ship_Maximum_Speed)

НФБК

Отношение в НФБК, когда для всех функциональных зависимостей отношения выполняется условие: детерминант — потенциальный ключ.
Рассматриваемая модель находится в нормальной форме Бойса-Кодда

Денормализация

Хранение агрегированных данных:

Вместо использования вычислений (например, среднего значения, суммы или количества) каждый раз при выполнении запросов, можно создать дополнительные столбцы в таблицах, которые будут содержать предварительно рассчитанные значения. Например, можно добавить столбец Total_Participants в таблицу Expedition, который будет содержать общее количество участников каждой экспедиции.

```
ALTER TABLE Expedition ADD COLUMN Total_Participants INTEGER;
```

Дублирование данных для избежания операций JOIN:

В некоторых случаях может быть полезно дублировать данные из одной таблицы в другую, чтобы избежать операций JOIN при выполнении запросов.

Например, если часто требуется информация о космических кораблях и их моделях вместе с экспедициями, можно денормализовать данные, добавив необходимые атрибуты из таблиц Spaceship и Spaceship_Model в таблицу Expedition. (Например: Spaceship_Name, Spaceship_Model, Payload, Ship_Maximum_Speed)

```
ALTER TABLE Expedition  
ADD COLUMN Spaceship_Name TEXT,  
ADD COLUMN Spaceship_Model TEXT,  
ADD COLUMN Spaceship_Payload INTEGER,  
ADD COLUMN Spaceship_Max_Speed INTEGER;
```

Создание функции

Нам нужно автоматически обновлять поле Total_Participants в таблице Expedition каждый раз, когда добавляется или удаляется запись в таблице Participants.

```
CREATE OR REPLACE FUNCTION update_total_participants()  
RETURNS TRIGGER AS $$  
BEGIN  
    -- Обновление количества участников в экспедиции  
    UPDATE Expedition  
    SET Total_Participants = (  
        SELECT COUNT(*)  
        FROM Participants  
        WHERE Expedition_ID = NEW.Expedition_ID  
    )  
    WHERE ID = NEW.Expedition_ID;  
  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

Создание триггера

```
CREATE TRIGGER participants_after_insert_delete  
AFTER INSERT OR DELETE ON Participants  
FOR EACH ROW  
EXECUTE FUNCTION update_total_participants();
```

Этот триггер и функция обеспечат автоматическое обновление поля Total_Participants в таблице Expedition при добавлении или удалении участников.

Вывод

В процессе выполнения работы я узнал про нормализацию, привел мою модель к НФБК. Написал функцию и триггер.