

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

Факультет программной инженерии и компьютерной техники

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ 3
БАЗЫ ДАННЫХ
ВАРИАНТ 328455

Студент: Пышкин Никита Сергеевич, Р3113

Преподаватель: Николаев Владимир Вячеславович

Санкт Петербург 2024

Содержание

Задание	3
Выполнение работы.....	4
Функциональные зависимости	4
Приведите отношения в 3NF	4
Приведите отношения в BCNF	4
Денормализация	5
Триггер и функция на языке PL/pgSQL.....	5
Заключение.....	7

Задание

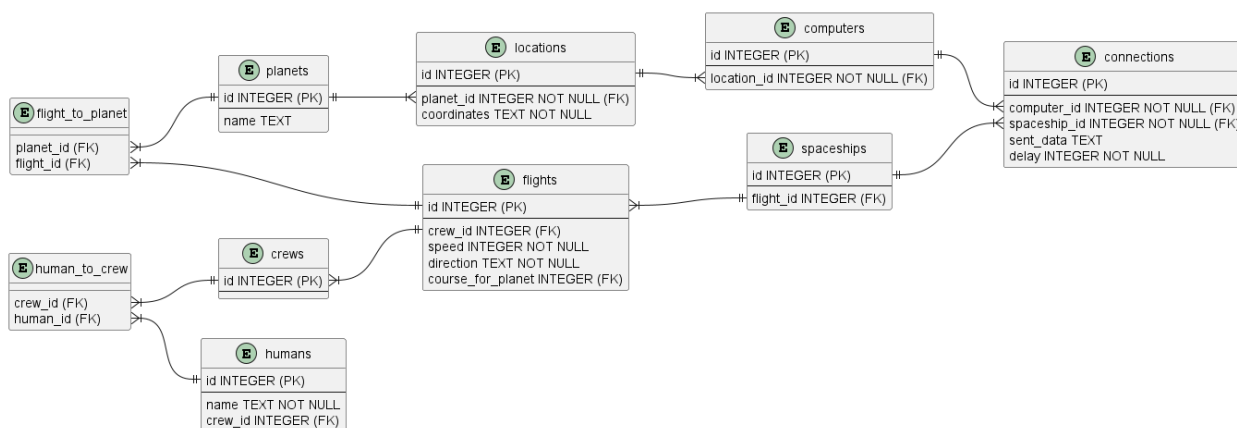
Лабораторная работа #3

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 работы

Функциональные зависимости

flight_to_planet: (planet_id, flight_id) \rightarrow ()

human_to_crew: (crew_id, human_id) \rightarrow ()

planets: id \rightarrow (name)

crews: id \rightarrow ()

humans: id \rightarrow (name, crew_id)

locations: id \rightarrow (planet_id, coordinates)

flights: id \rightarrow (crew_id, speed, direction, course_for_planet)

computers: id \rightarrow (location_id)

spaceships: id \rightarrow (flight_id)

connections: id \rightarrow (computer_id, spaceship_id, sent_data, delay)

Приведите отношения в 3NF

1NF: отношение находится в 1NF, если все его атрибуты содержат только атомарные значения. Моя модель удовлетворяет 1NF, т.к. все ее атрибуты атомарны и нет повторяющихся групп

2NF: отношение находится в 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа. Моя модель удовлетворяет 2NF, т.к. все неключевые атрибуты полностью функционально зависят от первичных ключей.

3NF: отношение находится в 3NF, если оно находится в 2NF и не содержит транзитивных зависимостей. Моя модель удовлетворяет 3NF, т.к. не содержит транзитивных зависимостей

Приведите отношения в BCNF

Отношение находится в BCNF, если для каждой функциональной зависимости $X \rightarrow Y$, X является суперключом. Моя модель удовлетворяет BCNF, т.к. для всех функциональных зависимостей X является суперключом.

Денормализация

В случае с моей моделью нет подходящих ситуаций для денормализации.

Триггер и функция на языке PL/pgSQL

Напишем триггер для ситуации с денормализованной формой, описанной выше

```
CREATE OR REPLACE FUNCTION
count_people_in_the_crew(target_crew_id INTEGER)
RETURNS INTEGER AS $count_people_in_the_crew$

DECLARE

    people_count INTEGER := 0;

BEGIN

    SELECT COUNT(*) INTO people_count FROM humans WHERE crew_id
= target_crew_id;

    RETURN people_count;

END;

$count_people_in_the_crew$ LANGUAGE plpgsql;


CREATE OR REPLACE FUNCTION update_crew_count()
RETURNS TRIGGER AS $update_crew_count$

BEGIN

    IF TG_OP = 'TRUNCATE' THEN

        UPDATE crews SET people_count = 0;

    END IF;

    IF TG_OP = 'DELETE' THEN

        UPDATE crews SET people_count =
count_people_in_the_crew(OLD.crew_id) WHERE crews.id =
OLD.crew_id;

    ELSE

        IF NEW.crew_id IS NOT NULL THEN
```

```
        UPDATE crews SET people_count =  
count_people_in_the_crew(NEW.crew_id) WHERE crews.id =  
NEW.crew_id;
```

```
    END IF;
```

```
END IF;
```

```
RETURN NEW;
```

```
END;
```

```
$update_crew_count$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER update_count  
AFTER INSERT OR UPDATE OR DELETE ON humans  
FOR EACH ROW  
EXECUTE FUNCTION update_crew_count();
```

```
CREATE TRIGGER truncate_count  
AFTER TRUNCATE ON humans  
EXECUTE FUNCTION update_crew_count();
```

Заключение

В ходе выполнения данной лабораторной работы я изучил 1NF, 2NF, 3NF, BCNF и узнал, почему денормализация может быть полезной. Познакомился с языком PL/pgSQL и триггерами.