

Университет ИТМО

Факультет программной инженерии и компьютерной техники
Образовательная программа системное и прикладное
программное обеспечение

Лабораторная работа №3
По дисциплине "Базы данных"

Выполнил студент группы Р3109
Евграфов Артём Андреевич
Проверила:
Воронина Дарья Сергеевна

Санкт-Петербург 2025

Содержание

1. Задание	2
2. Даталогическая модель	2
3. Функциональные зависимости	3
4. Преобразование к НФ	3
4.1. Преобразование к 1НФ	3
4.2. Преобразование к 2НФ	3
4.3. Преобразование к 3НФ	3
4.4. Преобразование к НФБК	3
5. Денормализация	4
6. Триггер	4
7. Вывод	4

1. 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 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.

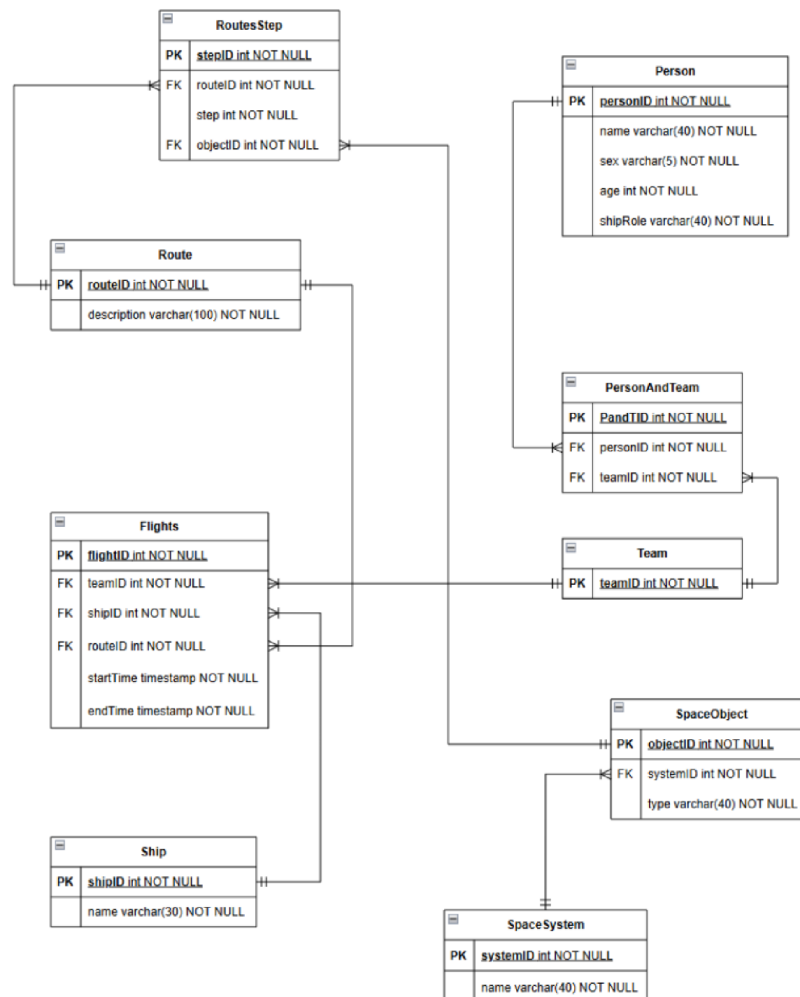
Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Исходная, нормализованная и денормализованная модели.
3. Ответы на вопросы, представленные в задании.
4. Функция и триггер на языке PL/pgSQL
5. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. Нормализация. Формы
2. Функциональные зависимости. Виды
3. Денормализация
4. Язык PL/pgSQL

2. Даталогическая модель



3. Функциональные зависимости

Person

personID \rightarrow name, sex, age, shipRole

Route

routeID \rightarrow description

RoutesStep

stepID \rightarrow routeID, step, objectID

Flights

flightID \rightarrow teamID, shipID, routeID, startTime, endTime

Ship

shipID \rightarrow name

PersonAndTeam

PandTID \rightarrow personID, teamID

Team

Функциональные зависимости отсутствуют

SpaceSystem

systemID \rightarrow name

SpaceObject

objectID \rightarrow systemID

4. Преобразование к НФ

4.1. Преобразование к 1НФ

Отношение находится в 1НФ, если на пересечении каждой строки и столбца одно ровно значение. Это условие уже выполняется для всех сущностей.

4.2. Преобразование к 2НФ

Все атрибуты в сущностях зависят от единственного атрибута *ID, следовательно, нет частичных зависимостей. Это условие уже выполняется для всех таблиц, т. к. у всех первичных ключей нет подмножеств, являющихся ПК, значит, все атрибуты в полной функциональной зависимости от первичных ключей, то есть все отношения находятся в 2НФ.

4.3. Преобразование к 3НФ

Отношение находится в 3НФ, если оно в 1НФ и 2НФ, а также все атрибуты, не входящие в первичный ключ, не находятся в транзитивной функциональной зависимости от первичного ключа. Это условие выполняется для всех сущностей, т.к. все атрибуты в функциональной зависимости от одного первичного ключа.

4.4. Преобразование к НФБК

Это требование к 3НФ говорит, что все атрибуты должны быть в функциональной зависимости от потенциального ключа. Это условие уже выполняется для всех таблиц.

5. Денормализация

Сущности можно денормализовать путем упрощения и соединения нескольких сущностей для того, чтобы упростить запросы и увеличить производительность за счет уменьшения JOIN-запросов. Например, можно не выносить SpaceSystem или Ship в отдельную сущность. В остальном денормализация излишня, поскольку она может привести к аномалиям.

6. Триггер

```
1 CREATE OR REPLACE FUNCTION check_flight_overlap() RETURNS TRIGGER AS $$
2 DECLARE
3     overlapping_flight INT;
4 BEGIN
5
6     SELECT COUNT(*) INTO overlapping_flight
7     FROM Flights
8     WHERE
9         shipID = NEW.shipID
10        AND (
11            (NEW.startTime BETWEEN startTime AND endTime) OR
12            (NEW.endTime BETWEEN startTime AND endTime) OR
13            (NEW.startTime <= startTime AND NEW.endTime >= endTime)
14        );
15
16     IF overlapping_flight > 0 THEN
17         RAISE EXCEPTION 'Пересечение временных интервалов для корабля %', NEW.shipID;
18     END IF;
19     RETURN NEW;
20 END;
21 $$ LANGUAGE plpgsql;
22
23 CREATE TRIGGER trg_flights_overlap
24 BEFORE INSERT OR UPDATE ON Flights
25 FOR EACH ROW EXECUTE FUNCTION check_flight_overlap();
```

7. Вывод

В ходе выполнения данной лабораторной работы я познакомился с нормальными формами, научился создавать триггеры и связанные с ними функции.