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

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3

по дисциплине

Базы данных

Выполнила:

Рыженкова Алина Александровна

Группа Р3111

Санкт-Петербург 2025

Оглавление

[Задание 3](#_Toc196393351)

[Исходная модель 3](#_Toc196393352)

[Функциональные зависимости 3](#_Toc196393353)

[Нормальные формы 5](#_Toc196393354)

[1NF 5](#_Toc196393355)

[2NF 5](#_Toc196393356)

[3NF 5](#_Toc196393357)

[BCNF 6](#_Toc196393358)

[Денормализация 6](#_Toc196393359)

[Объединение таблиц 6](#_Toc196393360)

[Добавление избыточных атрибутов 6](#_Toc196393361)

[Создание агрегированных таблиц: 6](#_Toc196393362)

[Функция и триггер на языке PL/pgSQL 7](#_Toc196393363)

[Вывод 8](#_Toc196393364)

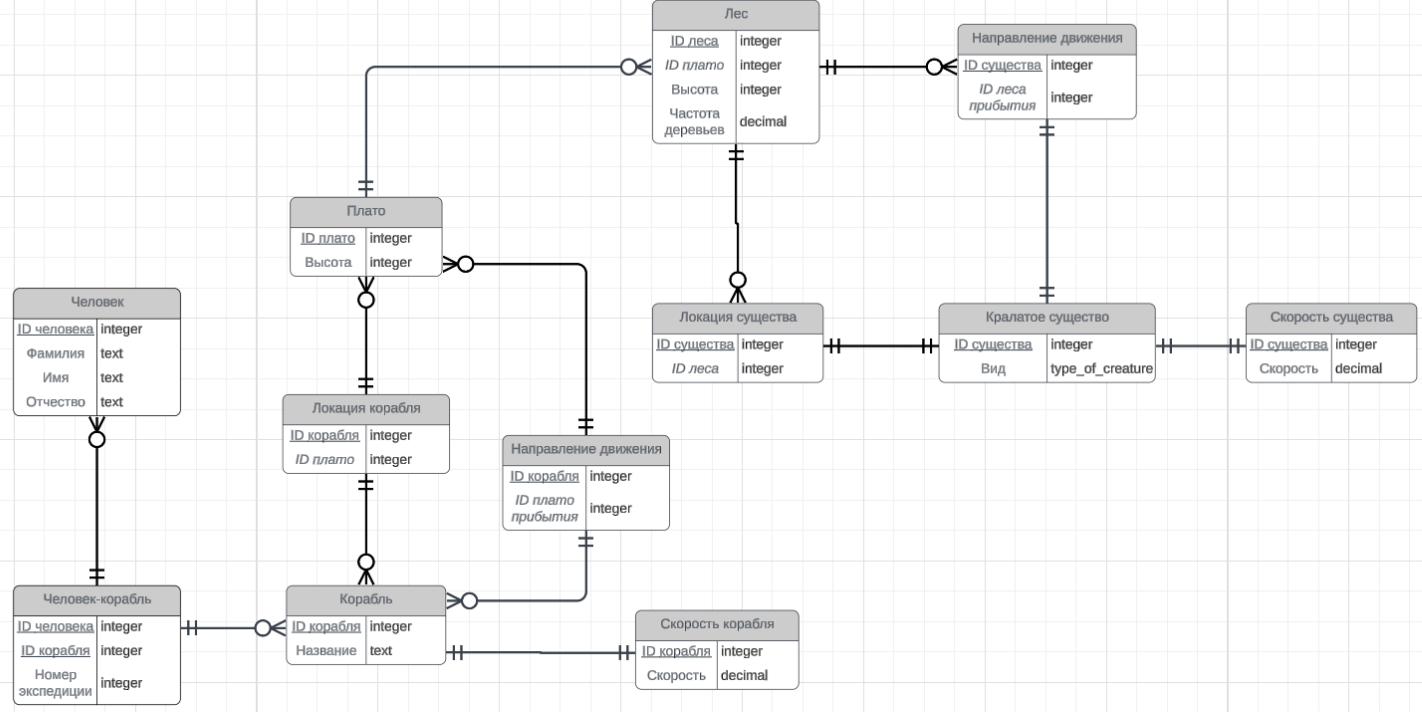
Задание

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.

Исходная модель



Функциональные зависимости

plateau: id → (height)

forest: id → (plateau\_id, height, trees\_frequency)

winged\_creature: id → (type)

creature\_speed: creature\_id→ (speed)

creature\_location: creature\_id→ (forest\_id)

creature\_direction\_of\_movement: creature\_id→ (arrival\_forest\_id)

ship: id → (name)

ship\_speed: ship\_id → (speed)

ship\_location: ship\_id → (arrival\_plateau\_id)

human: id → (name, surname, patronymic)

human\_ship: (human\_id, ship\_id, expedition\_number) → ()

Нормальные формы

1NF

Отношение находится в первой нормальной форме (1НФ) тогда и только тогда, когда

* в отношении нет повторяющихся групп (атрибутов с одинаковым смыслом)
* все атрибуты атомарны
* у отношения есть ключ

Моя модель находится в 1NF, поскольку нет повторяющихся групп, все атрибуты атомарны и у всех отношений есть ключ.

2NF

Отношение находится во второй нормальной форме (2НФ) тогда и только тогда, когда

* оно находится в первой нормальной форме
* все неключевые атрибуты функционально зависят от ключа целиком, но не от его части

Моя модель находится в 2NF, поскольку она находится в первой нормальной форме и все неключевые атрибуты [функционально зависят](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D0%B8:_%D0%B7%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5,_%D1%8D%D0%BA%D0%B2%D0%B8%D0%B2%D0%B0%D0%BB%D0%B5%D0%BD%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B8_%D0%BF%D1%80%D0%B0%D0%B2%D0%B8%D0%BB%D0%B0_%D0%B2%D1%8B%D0%B2%D0%BE%D0%B4%D0%B0) от ключа целиком.

3NF

Отношение находится в третьей нормальной форме (3НФ) тогда и только тогда, когда

* оно находится во второй нормальной форме
* неключевые атрибуты непосредственно (нетранзитивно) функционально зависят от ключей

Моя модель находится в 3NF, поскольку она находится во второй нормальной форме и все неключевые атрибуты зависят только от ключей.

Таким образом, моя схема уже находится в третьей нормальной форме.

BCNF

Отношение находится в нормальной форме Бойса-Кодда (НФБК) тогда и только тогда, когда для любой нетривиальной функциональной зависимости X→Y, X является надключом.

Моя модель находится в BCNF, поскольку она находится в третьей нормальной форме и никакие ключевые атрибуты не зависят от неключевых.

Таким образом, моя схема уже находится в BCNF.

Денормализация

Объединение таблиц

Объединение нескольких связанных таблиц в одну для уменьшения количества JOIN-операций. В моём случае, можно объединить таблицы winged\_creature, creature\_speed, creature\_location, creature\_direction\_of\_movement в одну таблицу. Это будет полезно, если часто требуется вспомогательная информация о крылатых существах, поскольку уменьшится количества объединений таблиц, и увеличится скорость выполнения запросов.

Аналогично, можно объединить таблицы ship, ship\_speed и ship\_location.

Добавление избыточных атрибутов

Введение дополнительных полей для хранения вычисляемых значений или значений из других таблиц. В моём случае, в отношение forest можно добавить атрибут count\_of\_creatures, в котором будет указано количество крылатых существ в лесу. На мой взгляд, это может быть полезно в случае, если существа редко меняют своё местоположение (то есть, количество существ в каждом лесу меняется редко), а количество крылатых существ в лесу требуется часто. Это может помочь увеличить скорость выполнения запросов.

Создание агрегированных таблиц:

Создание специальных таблиц для хранения агрегированных данных (например, сумм, средних значений) для ускорения отчетности и анализа. В моём случае, можно создать таблицу avg\_forest\_ height. На мой взгляд, это может быть полезно в случае, если средняя высота лесов меняется редко (то есть, в модель редко добавляются новые леса или меняется высота старых), но при этом часто требуется подсчитывать информацию о средней высоте лесов. Это может помочь увеличить скорость выполнения запросов.

Функция и триггер на языке PL/pgSQL

Я хочу написать функцию, которая будет определять, в каком лесу находится больше всего крылатых существ.

Таблица, в которую будут сохраняться данные:

CREATE TABLE populated\_forest (

id SERIAL PRIMARY KEY,

time TIMESTAMP,

forest\_id INT NOT NULL,

number\_of\_creatures positive\_integer

);

Функция:

CREATE FUNCTION count\_creatures\_in\_forests()

RETURNS trigger AS

$$

DECLARE forest INT;

number INT;

BEGIN

SELECT forest\_id, COUNT(\*) AS number

INTO forest, number

FROM creature\_location

GROUP BY forest\_id

ORDER BY number DESC

LIMIT 1;

INSERT INTO populated\_forest (time, forest\_id, number\_of\_creatures)

VALUES (NOW(), forest, number);

RETURN NEW;

END;

$$

LANGUAGE plpgsql;

Триггер:

CREATE TRIGGER update\_populated\_forest

AFTER INSERT OR UPDATE OR DELETE ON creature\_location

FOR EACH STATEMENT

EXECUTE FUNCTION count\_creatures\_in\_forests();

Вывод

При выполнении лабораторное работы я узнала о нормализации и денормализации, узнала, что такое функциональные зависимости, научилась писать функции на языке PL/pgSQL и использовать их в своей базе данных.