Министерство высшего образования и науки Российской Федерации

Национальный научно-исследовательский университет ИТМО

Факультет программной инженерии и компьютерной техники

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине

**«БАЗЫ ДАННЫХ».**

Выполнил:

Павленко И.Д.

Студент группы P3117.

Преподаватель:

Чупанов А.А.

Санкт-Петербург, 2025

Оглавление

[Текст задания 3](#_Toc164348110)

[Даталогическая модель (исходная) 4](#_Toc164348111)

[Функциональные зависимости (изначальные) 5](#_Toc164348112)

[Преобразование к 1НФ 5](#_Toc164348113)

[Преобразование к 2НФ 5](#_Toc164348114)

[Преобразование к 3НФ 6](#_Toc164348115)

[Преобразование к BCNF 6](#_Toc164348116)

[Денормализация 6](#_Toc164348117)

[Функциональные зависимости (после преобразований) 7](#_Toc164348118)

[Даталогическая модель (после преобразований) 8](#_Toc164348119)

[Триггер и функция 9](#_Toc164348120)

[Выводы по работе 10](#_Toc164348121)

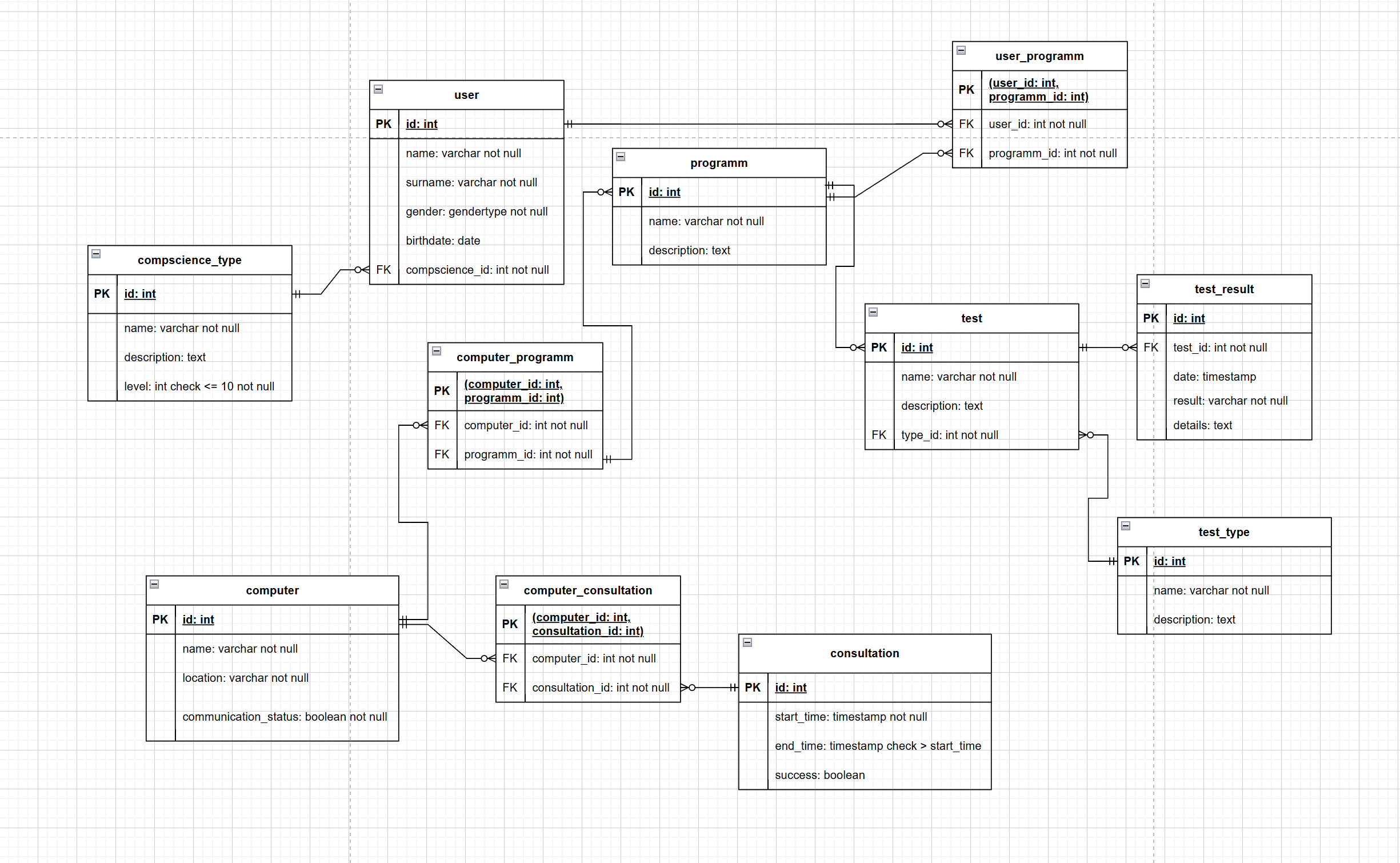
# Текст задания

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 NF (как минимум). Постройте схему на основе полученных отношений;
* опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе NF;
* преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
* 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.

# Даталогическая модель



# Функциональные зависимости (изначальные)

compscience\_type:

id → (name, description, level)

user:

id → (name, surname, gender, birthdate, compscience\_id)

programm:

id → (name, description)

test:

id → (name, description, type\_id)

test\_type:

id → (name, description)

test\_result:

id → (test\_id, date, result, details)

consultation:

id → (start\_time, end\_time, success)

computer:

id → (name, location, communication\_status)

# Преобразование к 1НФ

Отношение находится в 1NF, если все его атрибуты содержат только атомарные значения. Уже выполняется.

# Преобразование к 2НФ

Отношение находится во 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа. Уже выполняется, поскольку у всех первичных ключей нет подмножеств.

# Преобразование к 3НФ

Отношение находится в 3NF, если оно находится во 2NF и не содержит транзитивных зависимостей. Уже выполняется.

# Преобразование к BCNF

Отношение находится в BCNF, если для каждой функциональной зависимости X →Y, X является потенциальным ключом. Уже выполняется.

# Денормализация

Объединение таблиц может ускорить обработку запросов. Например, можно рассмотреть объединение таблиц test\_result и test, если часто требуется одновременно знать, какой тест выполнен и его результат.

Также можно добавить несколько избыточных атрибутов, что может улучшить производительность запросов. Например, если часто запрашивается количество людей с определенным уровнем computer science, можно добавить в таблицу compscience\_type атрибут compscience\_count. Это позволит избежать подсчета при каждом запросе, но необходимо обновлять этот атрибут при добавлении и удалении пользователей.

# Триггер и функция

Триггер вызывает функцию, которая устанавливает communication\_status компьютеру, в зависимости от того, участвует ли он в активной консультации или нет.

create or replace function **update\_communication\_status**()  
 returns trigger as  
$$  
begin  
 update computer  
 set communication\_status = **exists**(select 1  
 from computer\_consultation cc  
 join consultation c on cc.consultation\_id = c.id  
 where cc.computer\_id = computer.id  
 and **NOW**() BETWEEN c.start\_time and c.end\_time)  
 where computer.id = OLD.computer\_id or computer.id = NEW.computer\_id;  
 return NEW;  
end;  
$$ language plpgsql;  
  
create trigger trigger\_comm\_status  
 after insert or update or delete  
 on computer\_consultation  
 for each row  
execute function **update\_communication\_status**();

# Выводы по работе

В ходе выполнения данной лабораторной работы я познакомился с понятием нормализации и денормализации, научился определять функциональные зависимости модели и анализировать модели на соответствие нормальным формам. Также научился писать собственные триггеры.