

Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа №3
по дисциплине «Базы данных»
Вариант №3160

Выполнил:

Дядев Владислав Александрович

Группа Р3131

Проверил:

Вербовой А. А.

Санкт-Петербург

2025

Содержание

Задание.....	3
Порядок выполнения.....	4
Функциональные зависимости	4
Нормальные формы	4
Денормализация	5
Триггер	6
Вывод	7

Задание

Вариант №3160

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.

Порядок выполнения

Функциональные зависимости

- **sensient_creature:** (id) -> (type, description)
- **emotion:** (id) -> (type, intensity, creature_id)
- **event:** (id) -> (title, date, description)
- **natural_object:** (id) -> (type, characteristic, effect)
- **thought:** (id) -> (content, complexity, creature_id, reason_id)
- **reason:** (id) -> (type)
- **reason_emotion:** (id) -> (emotion_id)
- **reason_event:** (id) -> (event_id)
- **reason_object:** (id) -> (object_id)
- **setting:** (id) -> (object_id, event_id, effect_type)
- **participation:** (id) -> (creature_id, event_id, role)
- **reaction:** (id) -> (action_id, emotion_id)
- **action:** (id) -> (description, thought_id)

Нормальные формы

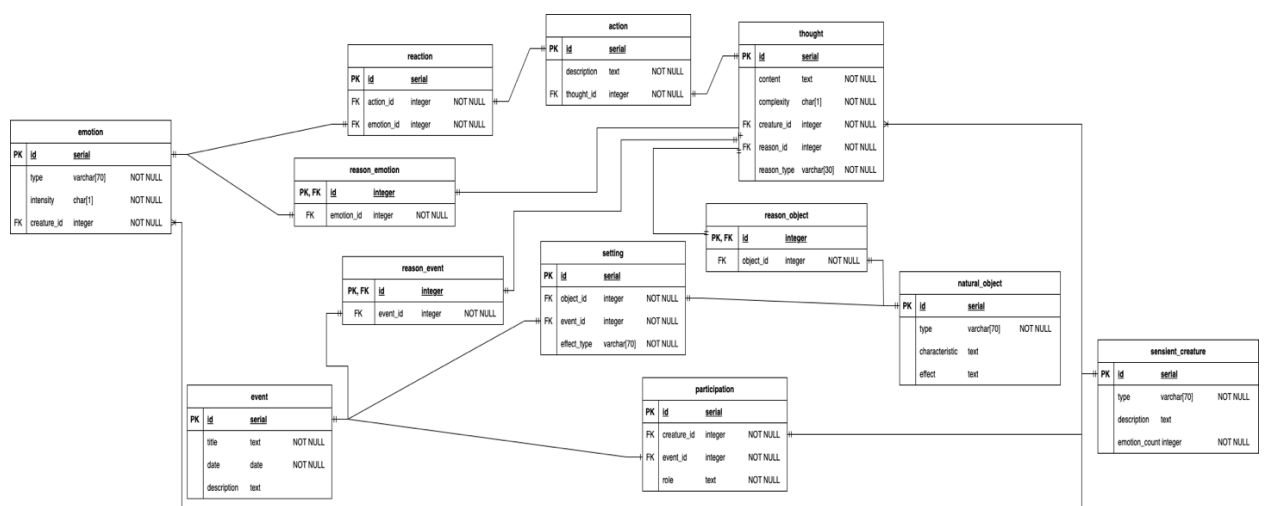
- **1NF:** Отношение находится в 1NF, если все его атрибуты содержат только атомарные значения и отсутствуют повторяющиеся группы. Мои отношения удовлетворяют 1NF, так как все атрибуты атомарны, и нет повторяющихся групп.
- **2NF:** Отношение находится в 2NF, если оно находится в 1NF и все его неключевые атрибуты полностью функционально зависят от первичного ключа. Моя модель удовлетворяет 2NF, так как 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 не удовлетворяет

3NF, так как все неключевые атрибуты напрямую зависят от первичных ключей.

- **BCNF:** Отношение находится в BCNF, если для каждой функциональной зависимости $X \rightarrow Y$, X является суперключом. Моя модель удовлетворяет BCNF, так как для всех функциональных зависимостей X является суперключом.

Денормализация

- **Добавление избыточных атрибутов:** в некоторых случаях можно улучшить производительность благодаря добавлению избыточных атрибутов. Например, можно добавить атрибут **emotion_count** в **sensient_creature**, чтобы получать количество чувств, которые существо испытывает прямо сейчас.
- **Объединение связанных таблиц:** в некоторых случаях объединение таблиц может уменьшить количество операций JOIN, то есть уменьшить время обработки запросов. В моей схеме можно рассмотреть объединение таблиц **thought** и **reason**, если часто необходимо узнать тип причины (**reason.type**), вызывающей мысль, без обращения к таблице причин.



Триггер

Когда у существа появляется эмоция с высокой интенсивностью (параметр “Н”), то автоматически создается событие, которое фиксирует факт сильного всплеска эмоций и связывает это событие с нужным существом.

Реализация на языке SQL представлена в репозитории на github:

<https://github.com/Alvas07/ITMO/blob/main/2%20DataBases/Lab3/trigger.sql>

Вывод

В ходе данной лабораторной работы я изучил понятия нормализации и денормализации. Научился работать с функциями, триггерами и писать собственные.