Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа **№3**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3112

Марьин Григорий Алексеевич

*Преподаватель*:

Кустарев Иван Павлович

Санкт-Петербург 2025 г.

1. **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 3NF (как минимум). Постройте схему на основе полученных отношений;
* 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 функцию, связанную с вашей предметной областью, согласуйте ее с преподавателем и реализуйте на языке PL/pgSQL.

****

1. **Функциональные зависимости**

|  |
| --- |
| **person:**  id → (name, surname, birth\_date, current\_location\_id)  **message:**  id → (text, sender\_id, receiver\_id, sent\_time)  **emotionalstatus**: id → (person\_id, emotion\_status)  **transport**: id → (name, type, current\_location\_id)  **movement**: id → (person\_id, transport\_id, to\_location\_id)  **location:** (name, type) → ()  **personloc:** (person\_id, loc\_id) → ()  **movementlocation:** (movement\_id, location\_id) → ()  **communication:** (message\_id, person\_id) → () |

1. **Нормальные формы**

|  |
| --- |
| 1NF: Отношение находится в 1NF, если все его атрибуты содержат только атомарные значения. Моя модель удовлетворяет 1NF, так как все атрибуты атомарны, и нет повторяющихся групп.  2NF: Отношение находится во 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 во 2NF и не содержит транзитивных зависимостей. Моя модель удовлетворяет 3NF, так как все неключевые атрибуты зависят только от первичных ключей, и не содержат транзитивных зависимостей. |

1. **BCNF**

|  |
| --- |
| Отношение находится в BCNF,  если для каждой функциональной зависимости X →Y,  X является первичным ключом. Моя модель удовлетворяет BCNF, так как  для всех функциональных зависимостей X является первичным ключом. |

1. **Денормализация**

|  |
| --- |
| **Объединение связанных таблиц**: в некоторых случаях, объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, можно рассмотреть объединение таблиц movement и transport, если часто запрашиваются данные о перемещениях и транспорте одновременно.  **Добавление избыточных атрибутов**: в некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, если часто запрашивается количество людей в одной локации, можно добавить атрибут people\_count в таблицу location. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении или удалении людей из локации. |

1. **Функция на языке PL/pgSQL**

Я разработаю триггер и функцию, которые будут автоматически записывать в таблицу сообщений (communication) уведомление, когда у человека резко меняется эмоциональный статус в таблице emotionalstatus

|  |
| --- |
| **CREATE OR REPLACE FUNCTION log\_emotional\_change()**  **RETURNS TRIGGER AS $$**  **DECLARE**  **person\_name VARCHAR(100);**  **message\_text TEXT;**  **BEGIN**  **-- Получаем имя человека из таблицы person**  **SELECT CONCAT(name, ' ', surname) INTO person\_name**  **FROM person**  **WHERE id = NEW.person\_id;**    **-- Формируем текст сообщения в зависимости от типа изменения**  **IF TG\_OP = 'INSERT' THEN**  **message\_text := person\_name || ' has new emotional status: ' || NEW.emotion\_status;**  **ELSIF TG\_OP = 'UPDATE' AND OLD.emotion\_status IS DISTINCT FROM NEW.emotion\_status THEN**  **message\_text := person\_name || ' changed emotional status from ' || OLD.emotion\_status ||**  **' to ' || NEW.emotion\_status;**  **ELSE**  **RETURN NEW; -- Ничего не делаем, если статус не изменился**  **END IF;**    **-- Вставляем уведомление в таблицу message**  **INSERT INTO message (content, sender\_id, receiver\_id, sent\_time)**  **VALUES (message\_text, NEW.person\_id, 4, NOW()); -- Отправляем системному пользователю (id=4)**    **RETURN NEW;**  **END;**  **$$ LANGUAGE plpgsql;**  CREATE TRIGGER emotional\_status\_change\_trigger  AFTER INSERT OR UPDATE ON emotionalstatus  FOR EACH ROW  EXECUTE FUNCTION log\_emotional\_change(); |

1. **Вывод**

При выполнении лабораторной работы я познакомился с понятием нормализации и денормализации. Научился определять функциональные зависимости модели, а также анализировать последнюю на соответствие различным нормальным формам. Познакомился с процедурным языком PL/pgSQL. Изучил эффективные способы денормализации схемы базы данных и ситуации, в которых возможно их применение.