Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа **№3**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3110

Конкин Вадим

Вадимович

*Преподаватель*:

Николаев Владимир

Вячеславович

Санкт-Петербург 2023 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 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.

**Изображение выглядит как диаграмма, Технический чертеж, План, схематичный

Автоматически созданное описание**

1. **Функциональные зависимости**

|  |
| --- |
| Base:  id → name,  item\_id → name,  id → is\_opened,  item\_id → is\_opened,  id → item\_id,  id → place\_id,  item\_id → place\_id  Person:  id → name,  id → base\_id  Radiogram:  id → message,  id → sender\_id,  id → getter\_id  Action:  id → name  Material:  id → name  Tent:  id → is\_reinforced,  item\_id → is\_reinforced,  id → base\_id,  item\_id → base\_id  id → item\_id  Place:  id → name,  item\_id → name,  id → item\_id  Person\_to\_Place:  id → person\_id,  id → place\_id  Person\_to\_Action:  id → action\_id,  id → person\_id  Item\_to\_Action:  id → action\_id,  id → item\_id  Material\_to\_Tent:  id → material\_id,  id → tent\_id  После изменения:  Base:  id → base\_item\_id  Base\_Item:  id → name,  id → is\_opened,  id → place\_id  Person:  id → name,  id → base\_id  Radiogram:  id → message,  id → sender\_id,  id → getter\_id  Action:  id → name  Material:  id → name  Tent:  id → tent\_item\_id  Tent\_Item:  id → is\_reinforced,  id → base\_id,  Place:  id → place\_item\_id  Place\_Item:  id → name  Person\_to\_Place:  id → person\_id,  id → place\_id  Person\_to\_Action:  id → action\_id,  id → person\_id  Item\_to\_Action:  id → action\_id,  id → item\_id  Material\_to\_Tent:  id → material\_id,  id → tent\_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 суперключом. Моя модель удовлетворяет BCNF, так как  для всех функциональных зависимостей X является суперключом. |

1. **Денормализация**

|  |
| --- |
| **Объединение связанных таблиц**: В некоторых случаях, объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. Например, можно рассмотреть объединение таблиц Action\_to\_Item, Action, Action\_to\_Person, если человек часто совершает действия. Названия действий можно прописывать.  **Добавление избыточных атрибутов**: В некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, если часто запрашивается количество отправленных радиограм отдельных людей, можно добавить атрибут radiogram\_count в таблицу Person. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении радиограм. |

1. **Функция на языке PL/pgSQL**

Функция на языке PL/pgSQL.

|  |
| --- |
| CREATE OR REPLACE TRIGGER close\_base BEFORE INSERT ON radiogram  FOR EACH ROW EXECUTE FUNCTION close\_base();  CREATE OR REPLACE FUNCTION close\_base() RETURNS trigger AS $$  declare  who\_asked varchar(20);  who\_closed int;  base\_closed int;  opened boolean;  BEGIN  who\_closed = new.getter\_id;  SELECT base.id INTO base\_closed FROM base JOIN person ON person.base\_id = base.id  AND person.id = who\_closed;  IF new.message ILIKE 'Закрыть базу' then  SELECT base.is\_opened INTO opened From base where base.id = base\_closed;  IF (opened = false) then  RAISE EXCEPTION 'Base already closed';  end if;  UPDATE base SET is\_opened = false where base.id = base\_closed;  end if;  Return who\_asked;  END; $$ LANGUAGE plpgsql; |

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.