Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№1576**

Лабораторная работа **№3**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3115

Хромов Даниил

Тимофеевич

*Преподаватель*:

Горбунов Михаил

Витальевич

Николаев Владимир

Вячеславович

Санкт-Петербург 2023 г.

1. **Текст задания**

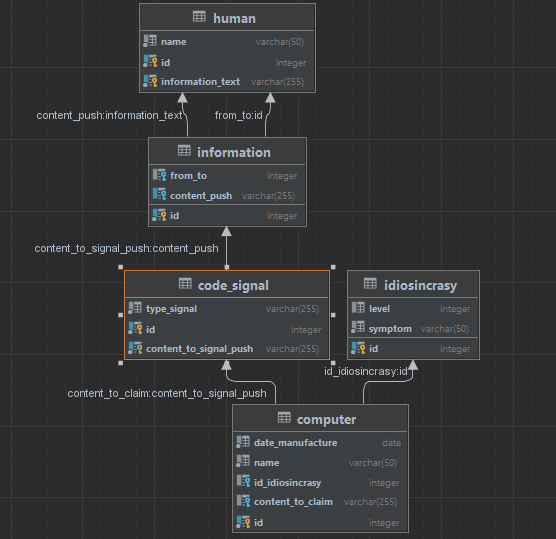
Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

* опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
* приведите отношения в 3NF (как минимум). Постройте схему на 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**

|  |
| --- |
| computer: id → (date\_manufacture, color)  human: id → (name, sex)  idiosyncrasy: id → (level, symptom)  information: id → (from\_to, content\_push)  code\_signal: id → (type\_signal, indifikator) |

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 people и employees, если часто запрашиваются данные о человеке и его работе одновременно.  **Добавление избыточных атрибутов**: В некоторых случаях добавление избыточных атрибутов может улучшить производительность запросов. Например, если часто запрашивается количество рюкзаков отдельных людей, можно добавить атрибут backpacks\_count в таблицу people. Это позволит избежать операций подсчета при каждом запросе, однако необходимо будет обновлять этот атрибут при добавлении или удалении рюкзаков. |

1. **Функция на языке PL/pgSQL**

Функция на языке PL/pgSQL для внесения даты последнего изменения.

|  |
| --- |
| **CREATE OR REPLACE FUNCTION update\_computer\_last\_modified()**  **RETURNS TRIGGER AS $$**  **BEGIN**  **NEW.last\_modified := NOW();**  **RETURN NEW;**  **END;**  **$$ LANGUAGE plpgsql;**  **CREATE TRIGGER computer\_update\_last\_modified**  **BEFORE UPDATE ON computer**  **FOR EACH ROW**  **EXECUTE FUNCTION update\_computer\_last\_modified();** |

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.