Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант **№313101**

Лабораторная работа **№3**

По дисциплине

**Базы Данных**

*Выполнил*:

Студент группы P3131

Нодири Хисравхон

*Преподаватель*:

Наумова Надежда Александровна

Санкт-Петербург 2023 г.

1. **Текст задания**

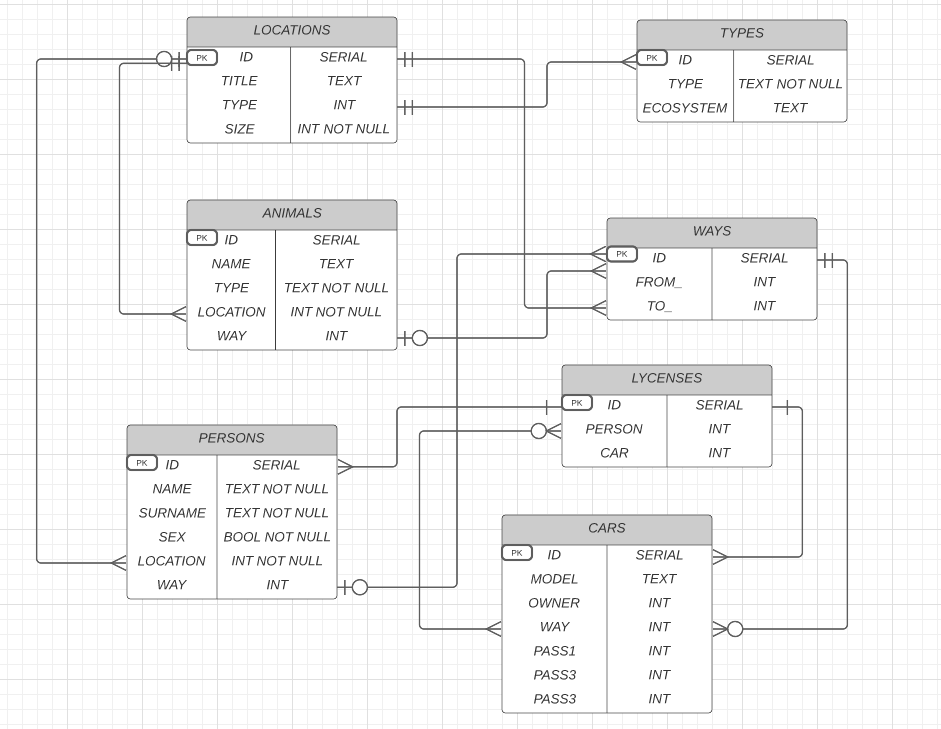
Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

* 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**

|  |
| --- |
| Locations: id → (title, type, size)  Types: id → (type, ecosystem)  Animals: id → (person\_id)  Ways: id → (from\_, to\_)  Lycenses: id → (person, car)  Persons: id → (name, surname, sex, location, way)  Cars: id → (mode, owner, way, pass1, pass2, pass3)  people\_actions: id → (action, object\_id, subject\_id, created\_at) |

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 автомобилей отдельных людей, можно добавить атрибут cars\_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 `listCars` разработана для извлечения списка уникальных моделей автомобилей и их количества в заданной локации, которыми владеют люди определенного пола.  -- Входные параметры:  -- `location\_title TEXT` - Название локации, в которой мы ищем автомобили. Функция вернет только автомобили в этой локации.  -- `person\_sex BOOLEAN` - Пол владельцев автомобилей. True означает мужчин, False - женщин. Функция вернет только автомобили, принадлежащие людям этого пола.  -- Функция возвращает таблицу с двумя столбцами:  -- `model TEXT`, который содержит модели автомобилей.  -- `count INT`, который содержит количество автомобилей каждой модели в указанной локации.  -- Каждая модель указана только один раз, даже если в локации больше одного такого автомобиля.  -- Если функция не найдет ни одного автомобиля, соответствующего критериям, она вызовет исключение с сообщением 'No cars found in % location', где % - название локации.  CREATE OR REPLACE FUNCTION listCars(location\_title TEXT, person\_sex BOOLEAN)  RETURNS TABLE (model TEXT, cars\_count BIGINT) AS $List$  BEGIN  RETURN QUERY  SELECT cars.model, COUNT(cars.id)  FROM persons  JOIN cars on persons.id = cars.owner  JOIN ways on cars.way = ways.id  JOIN locations on ways.to\_ = locations.id  WHERE locations.title = location\_title AND persons.sex = person\_sex  GROUP BY cars.model;    IF NOT FOUND THEN  RAISE EXCEPTION 'No cars found in % location', location\_title;  END IF;  END;  $List$ 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.