Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №1295

Лабораторная работа №1

По дисциплине

Базы Данных

Выполнил студент группы P3131:

Мироненеко Артем

Преподаватель:

Харитонова Анастасия Евгеньевна

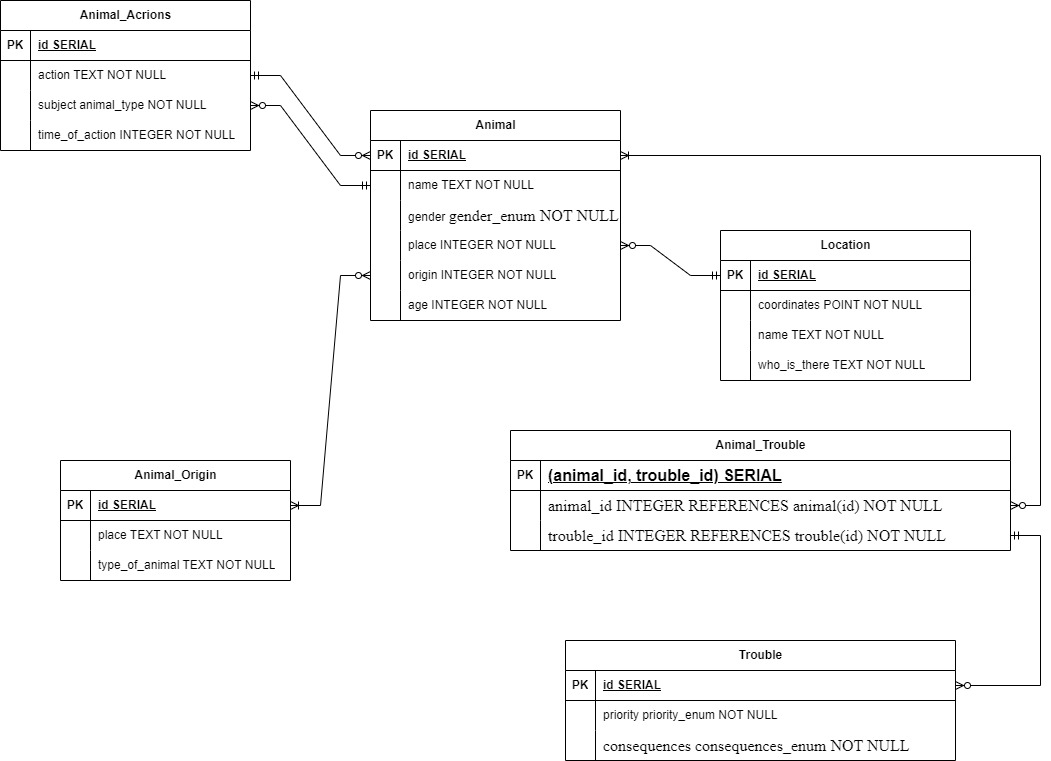
Санкт-Петербург 2023 г.

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.



1. **Функциональные зависимости**

location: id 🡪 (name, coordinates, who\_is\_there)

animal: id 🡪 (name, gender, place, origin, age)

animal\_actions: id 🡪 (action, subject, time\_of\_action)

animal\_origin: id 🡪 (place, type\_of\_animal)

animal\_trouble: (animal\_id, trouble\_id) 🡪 ()

trouble: id 🡪 (priority, consequences)

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,

Х является суперключом. Моя модель удовлетворяет BCNF, так как для всех функциональных зависимостей Х является суперключом.

1. **Денормализация**

**Объединение таблиц:**

Бывают ситуации, когда объединение таблиц может уменьшить количество операций JOIN и ускорить обработку запросов. У меня, например, можно рассмотреть объединение таблиц Animal и Animal\_Origin, если часто запрашиваются данные о животном и его происхождении одновременно.

**Добавление избыточных атрибутов:** В некоторых случаях добавление избыточных атрибутов способно повысить производительность запросов. Например, если часто количество проблем отдельного животного, можно добавить атрибут animal\_trouble\_count в таблицу Animal. Это должно помочь избежать операций подсчета при каждом запросе, но есть и небольшой минус, нужно будет обновлять этот атрибут при добавлении или удалении проблем.

1. **Функция на языке PL/pgSQL**

Функция нужна для поиска животного с минимальной численностью в стае

CREATE OR REPLACE FUNCTION *low\_number\_pack\_id*() RETURNS INTEGER AS  
$$  
DECLARE  
 min\_animal integer;  
BEGIN  
 IF (*EXISTS* (  
 SELECT \*  
 FROM animal  
 LEFT JOIN number\_in\_the\_pack  
 ON number\_in\_the\_pack.animal\_id = animal.id  
 WHERE number\_in\_the\_pack.id IS NULL  
 )) THEN  
 SELECT animal.id INTO min\_animal  
 FROM animal  
 LEFT JOIN number\_in\_the\_pack  
 ON number\_in\_the\_pack.animal\_id = animal.id  
 WHERE number\_in\_the\_pack.id IS NULL  
 LIMIT 1;  
 RETURN min\_animal;  
 ELSE  
 SELECT animal\_id INTO min\_animal  
 FROM number\_in\_the\_pack  
 GROUP BY number\_in\_the\_pack.animal\_id  
 ORDER BY *COUNT*(\*)  
 LIMIT 1;  
 RETURN min\_animal;  
 END IF;  
END;  
$$  
LANGUAGE plpgsql;  
SELECT \* FROM *low\_number\_pack\_id*()

1. **Триггер**

**Я добавил поле loc\_traffic в таблицу location, которое отвечает за количество животных в локации и написал тригер для автоматического обновления этого атрибута.**  
CREATE OR REPLACE FUNCTION *update\_location\_traffic*()  
RETURNS TRIGGER AS $$  
BEGIN  
 IF (TG\_OP = 'INSERT') THEN  
 UPDATE location  
 SET loc\_traffic = loc\_traffic + 1  
 WHERE id = NEW.location\_id;  
 ELSIF(TG\_OP = 'DELETE') THEN  
 UPDATE location  
 SET loc\_traffic = loc\_traffic - 1  
 WHERE id = OLD.location\_id;  
 ELSIF(TG\_OP = 'UPDATE') THEN  
 UPDATE location  
 SET loc\_traffic = loc\_traffic - 1  
 WHERE id = OLD.location\_id;  
 UPDATE location  
 SET loc\_traffic = loc\_traffic + 1  
 WHERE id = NEW.location\_id;  
 END IF;  
 RETURN NULL;  
END;  
$$  
LANGUAGE plpgsql;  
  
CREATE OR REPLACE TRIGGER update\_location\_traffic\_trigger  
 AFTER INSERT OR DELETE OR UPDATE ON location  
 FOR EACH ROW  
 EXECUTE FUNCTION *update\_location\_traffic*();  
  
SELECT \* FROM location;

1. **Вывод**

В процессе выполнения работы я узнал про нормализацию, привел мою модель к НФБК и доказал это, написал функцию и триггер.