

Санкт-Петербургский Национальный Исследовательский Университет ИТМО

Факультет Программной Инженерии и Компьютерной Техники



Вариант №1295
Лабораторная работа №1
По дисциплине
Базы Данных

Выполнил студент группы Р3131:

Мироненко Артем

Преподаватель:

Харитоновна Анастасия Евгеньевна

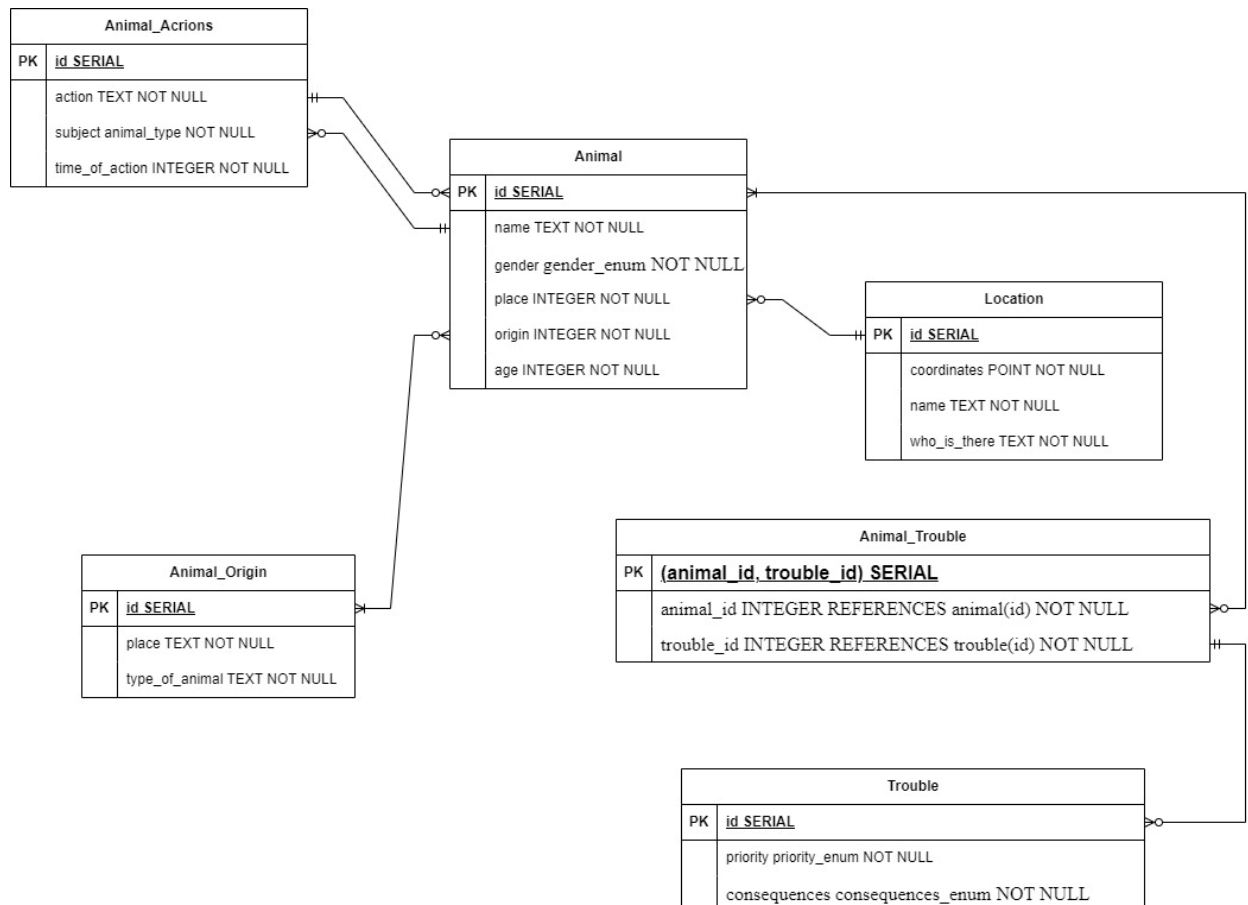
Санкт-Петербург 2023 г.

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 основе 3NF (как минимум).
- Опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF (как минимум). Постройте схему на основе 4NF;
- 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.



2. Функциональные зависимости

location:	id \rightarrow (name, coordinates, who_is_there)
animal:	id \rightarrow (name, gender, place, origin, age)
animal_actions:	id \rightarrow (action, subject, time_of_action)
animal_origin:	id \rightarrow (place, type_of_animal)
animal_trouble:	(animal_id, trouble_id) \rightarrow ()
trouble:	id \rightarrow (priority, consequences)

3. 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.

4. BCNF

Отношение находится в BCNF, если для каждой функциональной зависимости $X \rightarrow Y$, X является суперключом. Моя модель удовлетворяет BCNF, так как для всех функциональных зависимостей X является суперключом.

5. 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.

6. 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()
```

7. Триггер

Я добавил поле loc_traffic в таблицу location, которое отвечает за количество животных в локации и написал триггер 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;
```

8. Вывод

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