

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Национальный исследовательский университет ИТМО»**

**Факультет Программной Инженерии и Компьютерной Техники**



**Лабораторная работа № 3**

**Вариант № 310922**

**Выполнил:**

Карташев Владимир Р3131

**Преподаватель:**

Наумова Надежда Александровна

г. Санкт-Петербург, 2023 г.

## Для выполнения лабораторной работы №1 необходимо:

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

## Описание предметной области из лабораторной работы №1:

Питекантропы, обитавшие в первобытном вельде, не обладали ни одним из этих свойств; поэтому они отнюдь не благоденствовали, а были, напротив, весьма близки к полному вымиранию. Около полусотни этих существ ютилось в нескольких пещерах на склоне сожженной солнцем долины; по дну ее протекал слабенький ручеек, питаемый снегами с гор, лежавших в трехстах километрах к северу. В особо засушливые годы ручеек исчезал совсем и племя сильно страдало от жажды.

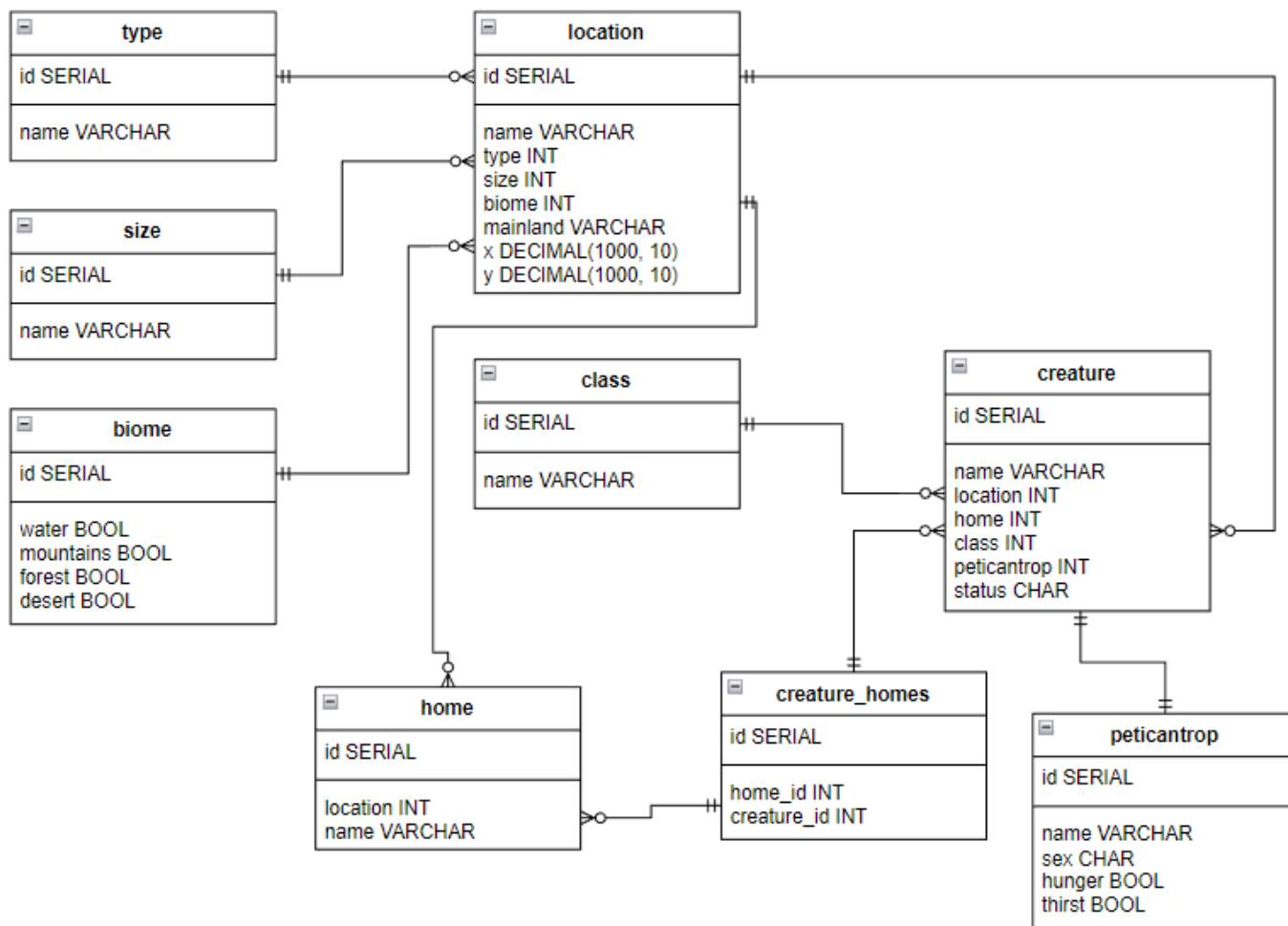
## Расширенная предметная область:

Засуха продолжалась десять миллионов лет, и царству ужасных ящеров уже давно пришел конец. Здесь, близ экватора, на материке, который позднее назовут Африкой, с новой яростью вспыхнула борьба за существование, и еще неясно было, кто выйдет из нее победителем. На этой бесплодной, иссушенной зноем земле благоденствовать или хотя бы просто выжить могли только маленькие, или ловкие, или свирепые.

Питекантропы, обитавшие в первобытном вельде, не обладали ни одним из этих свойств, поэтому они отнюдь не благоденствовали, а были, напротив, весьма близки к полному вымиранию. Около полусотни этих существ ютилось в нескольких пещерах на склоне сожженной солнцем долины; по дну ее протекал слабенький ручеек, питаемый снегами с гор, лежавших в трехстах километрах к северу. В особо засушливые годы ручеек исчезал совсем и племя сильно страдало от жажды.

Питекантропы всегда голодали, а сейчас попросту умирали от голода. Когда первый слабый проблеск рассвета проник в пещеру, Смотрящий на Луну увидел, что его отец ночью умер. Собственно, он не знал, что Старик был его отцом, — такая связь одного существа с другим была совершенно недоступна его пониманию, но, глядя на иссохшее тело умершего, он ощутил смутное беспокойство — зародыш будущей человеческой скорби.

Даталогическая модель:



1. Опишите функциональные зависимости для отношений полученной схемы (минимальное множество).

type: **id** ⇒ (name)

size: **id** ⇒ (name)

biome: **id** ⇒ (water, mountains, forest, desert)

location: **id** ⇒ (name, type\_id, size\_id, biome\_id, mainland, x, y)

class: **id** ⇒ (name)

creature: **id** ⇒ (name, location\_id, home\_id, class\_id, peticantrop\_id, status)

home: **id** ⇒ (location\_id, name)

creature\_homes: **id** ⇒ (home\_id, creature\_id)

peticantrop: **id** ⇒ (name, sex, hunger, thirst)

2. Приведите отношения в **3NF** (как минимум). Постройте схему на основе **NF** (как минимум).

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

**Критерии 1NF:** Все строки должны быть различными. Все элементы внутри ячеек должны быть не списками, а атомарными (неделимыми). То есть, атомарный элемент нельзя разделить на части, которые могут использоваться в таблице независимо друг от друга.

Изначальная модель соответствует требованию **1NF**

**Критерии 2NF:** Таблица должна находиться в первой нормальной форме. Любое её поле, не входящее в состав первичного ключа, функционально полно зависит от первичного ключа.

Изначальная модель соответствует требованию **2NF**

**Критерии 3NF:** Таблица находится во второй нормальной форме. Любой её не ключевой атрибут функционально зависит только от первичного ключа.

Изначальная модель соответствует требованию **3NF**

3. Преобразуйте отношения в **BCNF**. Докажите, что полученные отношения представлены в **BCNF**. Если ваша схема находится уже в **BCNF**, докажите это;

**Критерии BCNF (нормальная форма Бойса-Кодда):** Таблица должна находиться в третьей нормальной форме. Здесь все как обычно, т.е. как и у всех остальных нормальных форм, первое требование заключается в том, чтобы таблица находилась в предыдущей нормальной форме, в данном случае в третьей нормальной форме;

Ключевые атрибуты составного ключа не должны зависеть от неключевых атрибутов (то есть  $X \Rightarrow Y$ )

Изначальная модель соответствует требованию **BCNF**

4. Какие денормализации будут полезны для вашей схемы? Приведите подробное описание.

**Денормализация:** Нормализация базы данных до 5 или 6 нормальной формы значительно снижает производительность, работа с такой базой будет неэффективна и неудобна, а управлять ей будет очень сложно.

Денормализация обычно проводится путем добавления избыточных данных в таблицу, т.е. тех данных, которые по требованиям той или иной нормальной формы должны выноситься в отдельную таблицу.

**Денормализация моей схемы:**

**Добавление избыточных данных:** Для того, чтобы ускорить операцию вывода, можно, например, добавить новое поле **biome\_id** в таблицу **home**. Это позволит не загружать при каждом обращении к таблице **home** большую таблицу **location**. Таким образом, мы сократим промежуток для вызова таблицы **biome**. Также можно построить новую таблицу с сохранением некоторых агрегатных функций (MIN, MAX, AVG и т.д.), что позволит нам быстрее обращаться к данным и сравнивать их. Но для этого придется постоянно обновлять. Также можно добавить в некоторые таблицы повторяющиеся данные, например: добавление поля **sex** в таблицу **creature** позволит брать данные для статистики намного быстрее, так как в данном случае нам не нужна будет обращаться к данным таблицы **peticantrop**.

**Объединение связанных таблиц:** можно объединить таблицы **class** и **peticantrop** с таблицей **creature**, для того, чтобы уменьшить расстояние между таблицами. Эту же операцию можно проделать с таблицами **type** и **size** с **location**.

#### **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. Изучил эффективные способы денормализации схемы базы данных и ситуации, в которых возможно их применение.

5. Написать функцию, которая выводит количество мужчин / женщин, живущих на определенном материке.

```
CREATE FUNCTION selectCountFromMainlend (SexParam VARCHAR, Mainland
VARCHAR)
RETURNS int AS
'SELECT COUNT(peticantrop.sex)
FROM peticantrop
LEFT JOIN creature ON peticantrop.id = creature.peticantrop
LEFT JOIN creature_homes ON creature.id = creature_homes.creature_id
LEFT JOIN home ON creature_homes.home_id = home.id
LEFT JOIN location ON home.location = location.id
WHERE Mainland = location.mainland AND SexParam = peticantrop.sex;'
LANGUAGE SQL;
```

```
studs=> select selectCountFromMainlend('Мужчина', 'Африка');
selectcountfrommainlend
-----
3
(1 строка)
```

6. Написать триггер, который выводит информацию об объекте, который был добавлен/изменен/удален в таблице **peticantrop**.

--удаление триггеров с нижеперечисленными названиями, если они уже существуют

```
DROP TRIGGER IF EXISTS peticantrop_insert_trigger ON PETICANTROP;  
DROP TRIGGER IF EXISTS peticantrop_update_trigger ON PETICANTROP;  
DROP TRIGGER IF EXISTS peticantrop_delete_trigger ON PETICANTROP;
```

--создание функции, вызываемой при добавлении нового объекта в таблицу peticantrop

```
CREATE OR REPLACE FUNCTION peticantrop_after_insert_function() RETURNS TRIGGER  
AS $$  
BEGIN  
    RAISE NOTICE 'INSERT: New object: id="%", name="%", sex="%", hunger="%", thirst="%",  
NEW.id, NEW.name, NEW.sex, NEW.hunger, NEW.thirst;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

--создание функции, вызываемой при обновлении объекта в таблице peticantrop

```
CREATE OR REPLACE FUNCTION peticantrop_after_update_function() RETURNS TRIGGER  
AS $$  
BEGIN  
    RAISE NOTICE 'UPDATE: Changed object: id="%", name="%", sex="%", hunger="%",  
thirst="%",  
NEW.id, NEW.name, NEW.sex, NEW.hunger, NEW.thirst;  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

--создание функции, вызываемой при удалении объекта из таблицы peticantrop

```
CREATE OR REPLACE FUNCTION peticantrop_before_delete_function() RETURNS  
TRIGGER AS $$  
BEGIN  
    RAISE NOTICE 'DELETE: Deleted object: id="%", name="%", sex="%", hunger="%",  
thirst="%", OLD.id, OLD.name, OLD.sex, OLD.hunger, OLD.thirst;  
    RETURN OLD;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER peticantrop_insert_trigger  
AFTER INSERT ON peticantrop  
FOR EACH ROW  
EXECUTE FUNCTION peticantrop_after_insert_function();
```

```
CREATE TRIGGER peticantrop_update_trigger
AFTER UPDATE ON peticantrop
FOR EACH ROW
EXECUTE FUNCTION peticantrop_after_update_function();
```

```
CREATE TRIGGER peticantrop_delete_trigger
BEFORE DELETE ON peticantrop
FOR EACH ROW
EXECUTE FUNCTION peticantrop_before_delete_function();
```

```
[s373440@helios ~/psqlscripts/lab3]$ psql -h pg -d studs
psql (14.2)
SSL-соединение (протокол: TLSv1.3, шифр: TLS_AES_256_GCM_SHA384, бит: 256, сжатие: выкл.)
Введите "help", чтобы получить справку.

studs=> \i trigger.sql
DROP TRIGGER
DROP TRIGGER
DROP TRIGGER
CREATE FUNCTION
CREATE FUNCTION
CREATE FUNCTION
CREATE TRIGGER
CREATE TRIGGER
CREATE TRIGGER
studs=> INSERT INTO peticantrop(name, sex, hunger, thirst)
VALUES ('Дядя-Яга', 'Мужчина', FALSE, FALSE);
NOTICE: INSERT: New object: id='10', name='Дядя-Яга', sex='Мужчина', hunger='f', thirst='f'
INSERT 0 1
studs=> UPDATE peticantrop
SET name = 'Тетя-Яга', sex = 'Женщина', hunger = TRUE, thirst = TRUE
WHERE id = 7;
NOTICE: UPDATE: Changed object: id='7', name='Тетя-Яга', sex='Женщина', hunger='t', thirst='t'
UPDATE 1
studs=> DELETE FROM peticantrop
Where id = 7;
NOTICE: DELETE: Deleted object: id='7', name='Тетя-Яга', sex='Женщина', hunger='t', thirst='t'
DELETE 1
studs=> █
```