

Университет ИТМО  
Факультет программной инженерии и компьютерной техники

## Информационные системы и базы данных

### Лабораторная работа №4

Выполнила: Машина Е.А.

Группа Р33113

Преподаватель: Николаев В.В.

Санкт-Петербург  
2020 г.

## Задание

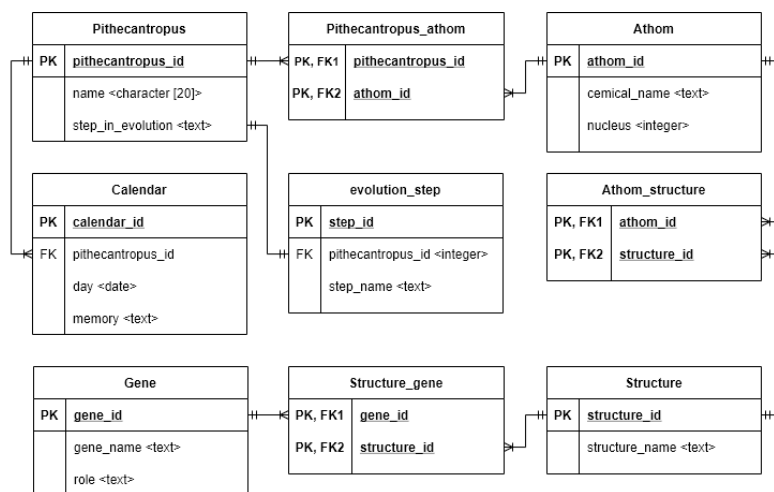
Для отношений, полученных при построении предметной области из лабораторной работы №1, выполните следующие действия:

- опишите функциональные зависимости для отношений полученной схемы (минимальное множество);
- приведите отношения в 3NF (как минимум). Постройте схему полученных отношений;
- опишите изменения в функциональных зависимостях, произошедшие после преобразования в 3NF.
- преобразуйте отношения в BCNF. Докажите, что полученные отношения представлены в BCNF;
- какие денормализации будут полезны для вашей схемы? Приведите подробное описание;

## Текст

Само по себе то, что он видел, не могло бы так повлиять на питекантропа, для этого нужно было еще усилить его способность к восприятию. За последние дни в жизни Смотрящего на Луну были пробелы; об этих периодах он ничего не мог бы вспомнить – именно тогда самые атомы его примитивного мозга перестраивались в новые структуры. Если он выживет, эти структуры будут увековечены – его гены передадут их грядущим поколениям.

## Выполнение



### 1. Функциональные зависимости:

- Pithecanthropus.pithecanthropus\_id -> Pithecanthropus.name;
- Pithecanthropus.pithecanthropus\_id -> Pithecanthropus.step\_in\_evolution;
- Calendar .calendar\_id -> Calendar.day;
- Calendar.calendar\_id -> Calendar.memory;
- Calendar.calendar\_id -> Calendar.pithecanthropus\_id;
- Evolution\_step.step\_id -> Evolution\_step.step\_name;
- Evolution\_step.step\_id -> Evolution\_step .pithecanthropus\_id;
- Atom.athom\_id -> Atom.chemical\_name;
- Atom.chemical\_name -> Atom\_nucleus;
- Structure.structure\_id -> Structure.name;
- Gene.gene\_id -> Gene.gene\_name;
- Gene.gene\_name -> Gene.role.

## 2. Преобразование в 3NF

**1NF:** отношение, на пересечении каждой строки и столбца — одно значение. Все мои таблицы удовлетворяют данным условиям.

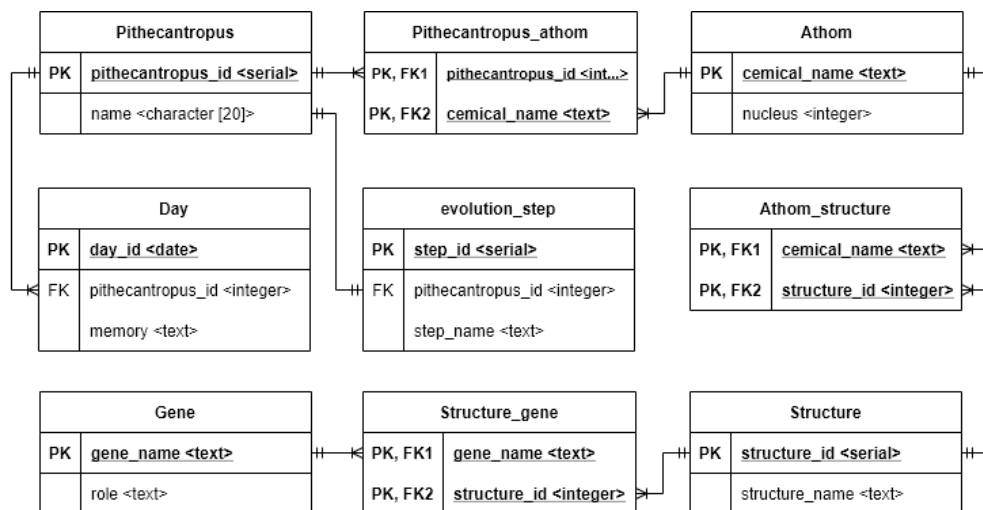
**2NF:** отношение в 1NF, каждый атрибут которого, отличный от атрибута первичного ключа, является полностью функционально независимым от любого потенциального ключа. То есть: нет частичных зависимостей от первичного и потенциальных ключей.

Чтобы привести к 2NF — убрать частичные зависимости от первичного ключа:

- удалить частично-зависимые атрибуты
- новое отношение: удаленные атрибуты + копия детерминанты

Все таблицы уже удовлетворяют условиям 2NF, а следовательно, и в данном случае преобразований не требуется.

**3NF:** отношение в 1NF и 2NF, при этом не имеет атрибутов, которые не входят в первичный ключ и находятся в транзитивной функциональной зависимости от первичного ключа. (Должны избавиться от атрибутов, которые зависят не от первичного ключа). Таких атрибутов в моей БД тоже нет. Соответственно и тут преобразований не требуется.



## 3. Определение 3NF не охватывает отношения со следующими условиями:

1. Отношение имеет два или более потенциальных ключа.
2. Оба ключа являются сложными.
3. Потенциальные ключи отношения перекрываются, то есть имеют хотя бы один общий атрибут.

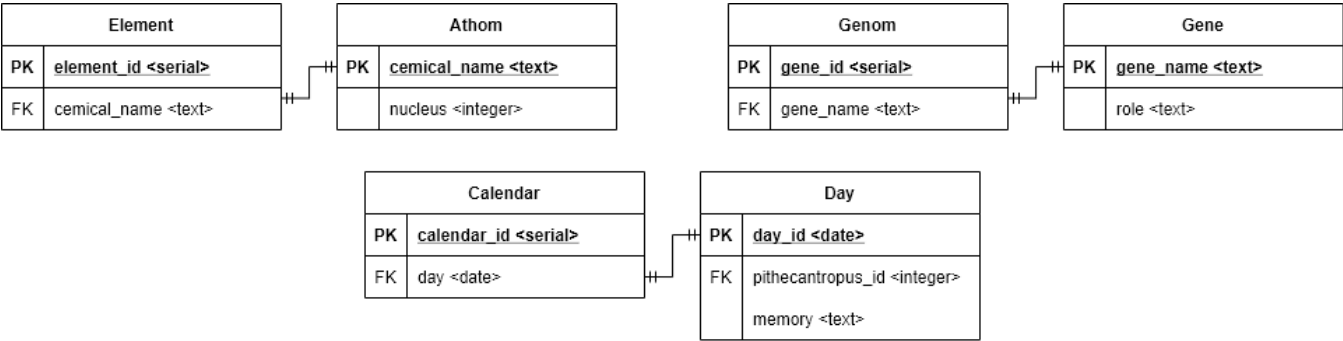
Для отношений, не удовлетворяющих хотя бы одному из этих условий, 3NF и НФБК эквивалентны. В частности, для отношения с простыми ключами разница между 3NF и НФБК отсутствует.

То есть в моем случае, 3NF=BCNF, так как все ключи - одинарные. Оправданные пути деморализации.

## Изменения (денормализация):

Изменены таблицы Athom, Gene и Calendar (теперь это логичнее называть Day) до этого заряд ядра транзитивно зависел от первичного ключа (athom\_id), название и роль гена зависели от первичного ключа (gene\_id), аналогично memory и date зависели от calendar\_id.

Я разделила их на два отношения (на примере атома): Атом (**id** и название) и Элемент (**название**, число протонов я ядре). Затем я поняла, что не имеет смысла отдельно хранить id с названием и удалила эту таблицу вообще. Аналогично с днём и геном.



## Вывод

В результате выполнения данной лабораторной работы я познакомилась с понятиями «функциональной зависимости», нормализации и денормализации.

И пришла к выводу, что при денормализации важно сохранить баланс между повышением скорости работы базы и увеличением риска появления противоречивых данных, между облегчением жизни программистам, пишущим Select'ы, и усложнением задачи тех, кто обеспечивает наполнение базы и обновление данных. Поэтому проводить денормализацию базы надо очень аккуратно, очень выборочно, только там, где без этого никак не обойтись. Если заранее нельзя подсчитать плюсы и минусы денормализации, то изначально необходимо реализовать модель с нормализованными таблицами, и лишь затем, для оптимизации проблемных запросов проводить денормализацию.