**Базы данных**

**Лабораторная работа №3**

**Вариант 11111**

Выполнил:

Полуянов Игорь Андреевич

Группа Р3110

Преподаватель:

Гаврилов А. В.

Оглавление

[Задание 3](#_Toc195179225)

[Функциональные зависимости 3](#_Toc195179226)

[Нормальные формы 4](#_Toc195179227)

[BCNF 4](#_Toc195179228)

[Денормализация 5](#_Toc195179229)

[Код программы 5](#_Toc195179230)

[Вывод 5](#_Toc195179231)

## 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 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.

Изображение выглядит как текст, визитная карточка, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

## Функциональные зависимости

* **Status**:
  + id → name
* **Classification**:
  + id → family, order\_name
* **Location**:
  + id → name, description
* **Organization**:
  + id → name, address
* **Species**:
  + id → name, status\_id, extinct\_since, classification\_id
  + id → name (через связь с таблицей Status)
  + id → family, order\_name (через связь с таблицей Classification)
* **Fossil**:
  + id → species\_id, description
  + id → name, status\_id, extinct\_since, classification\_id (через связь с таблицей Species)
* **Zoologist**:
  + id → name, organization\_id, contact\_details
  + id → name, address (через связь с таблицей Organization)
* **Discovery**:
  + id → fossil\_id, zoologist\_id, location\_id, discovery\_date, age, circumstances
  + id → species\_id, description (через связь с таблицей Fossil)
  + id → name, organization\_id, contact\_details (через связь с таблицей Zoologist)
  + id → name, description (через связь с таблицей Location)

## 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.

## 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.

**Пример:**

Species(

species\_id PK,

name,

status\_id → Status,

extinct\_since,

classification\_id → Classification

)

species\_id → name, status\_id, extinct\_since, classification\_id

Левая часть (species\_id) — **ключ**

Никаких других зависимостей, где неключевые атрибуты влияют на другие

## 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, сейчас, чтобы показать данные о виде нужно сделать такой **JOIN**:

SELECT s.name, st.name AS status, c.family, c.order\_name

FROM Species s

JOIN Status st ON s.status\_id = st.status\_id

JOIN Classification c ON s.classification\_id = c.classification\_id;

Если сделать **денормализацию**, можно кэшировать нужные данные прямо в Species:

ALTER TABLE Species

ADD COLUMN status\_name VARCHAR(50),

ADD COLUMN family VARCHAR(100),

ADD COLUMN order\_name VARCHAR(100);

## Код программы

[ITMO/DB/lab3/script.sql at main · PoluyanbIch/ITMO](https://github.com/PoluyanbIch/ITMO/blob/main/DB/lab2/script.sql)

## Вывод

При выполнении ЛР я узнал, что такое нормализация и денормализация. Научился определять функциональные зависимости, и анализировать модель на нормальные формы.