**Министерство науки и высшего образования Российской Федерации**

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

Факультет программной инженерии и компьютерной техники

**Дисциплина:** Базы данных

**Лабораторная работа №1**

**Вариант №1488**

Выполнила:

Хвостова И.Л

Группа:

P3124

Проверил:

Королёва Ю.А.

г.Санкт-Петербург

2023г.

**Оглавление**

[Задание 3](#_Toc137314732)

[Список сущностей 4](#_Toc137314733)

[Инфологическая модель 5](#_Toc137314734)

[Даталогическая модель 6](#_Toc137314735)

[Реализация модели на SQL 7](#_Toc137314736)

[Дополнительные запросы 9](#_Toc137314737)

# Задание

Для выполнения лабораторной работы №1 необходимо:

1. На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.
2. Составить инфологическую модель.
3. Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.
4. Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.
5. Заполнить созданные таблицы тестовыми данными.

**Описание предметной области, по которой должна быть построена доменная модель:**

Доджсон кратко изложил предысторию. Появление "ИнДжин" в 1983 году при помощи японских вкладчиков. Приобретение трех суперкомпьютеров "Крей Экс-эм-пи". Приобретение Isla Nublar в Коста-Рике. Запасы янтаря. Необычные дары зоопаркам по всему миру, начиная от Нью-Йорского зоологического общества, кончая Рантхапурским парком диких животных в Индии.

# Список сущностей

Стержневая:

1. Компания
2. Остров
3. Суперкомпьютеры
4. Запасы янтаря
5. Зоопарки
6. Ученые

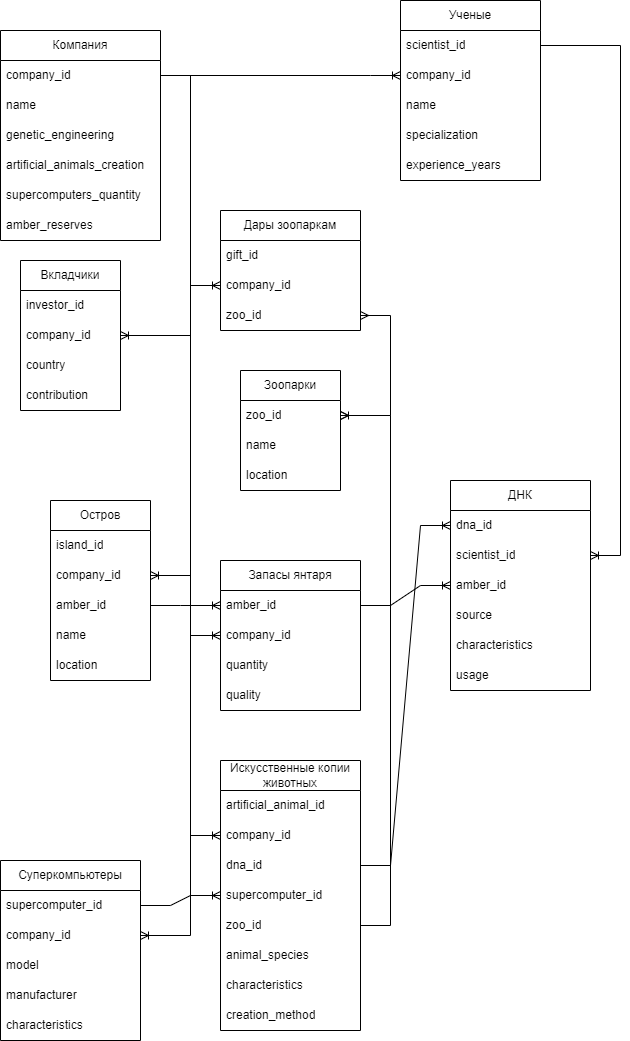
Ассоциация:

1. Вкладчики
2. Дары зоопаркам

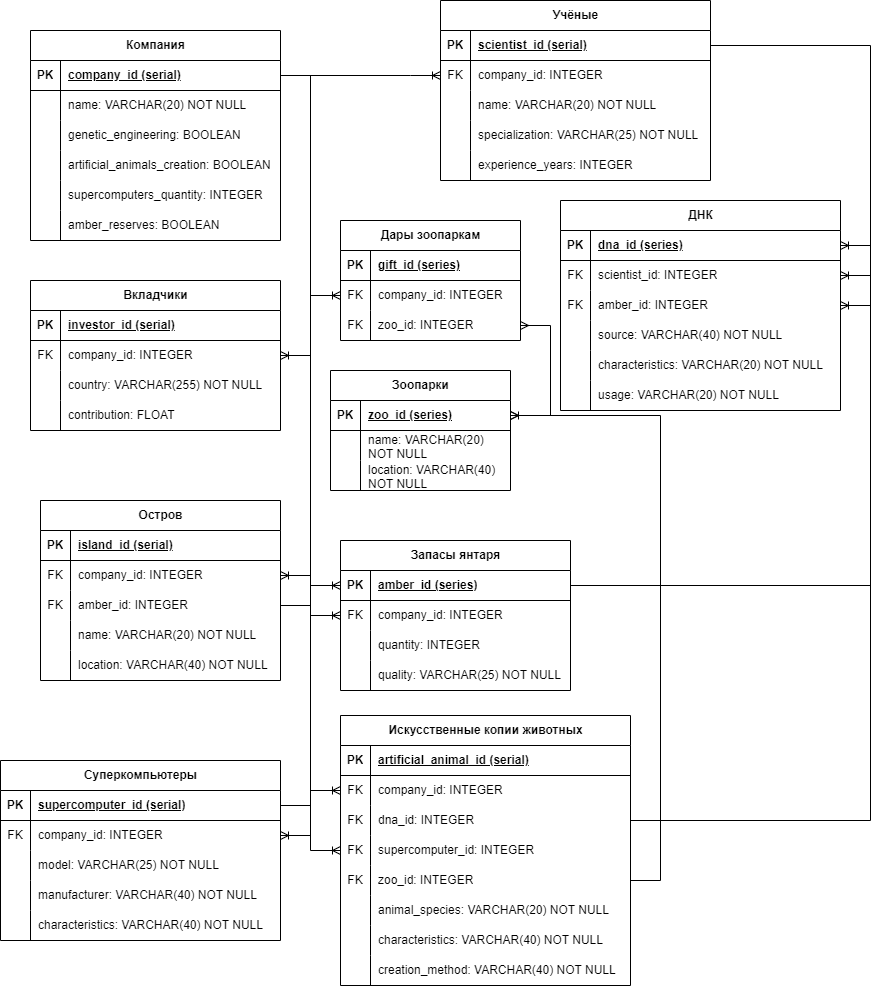
Характеристика:

1. Искусственные копии животных
2. ДНК

# Инфологическая модель



# Даталогическая модель



# Реализация модели на SQL

studs=> create table company(

studs(> company\_id SERIAL PRIMARY KEY,

studs(> name VARCHAR(20) not null,

studs(> quantity INTEGER,

studs(> artificial\_animals\_creation BOOLEAN,

studs(> supercomputers\_quantity INTEGER,

studs(> amber\_reserves BOOLEAN,

studs(> );

studs=> create table investor(

studs(> investor\_id SERIAL PRIMARY KEY,

studs(> company\_id INTEGER REFERENCES company(company\_id),

studs(> country VARCHAR(25) not null,

studs(> contribution FLOAT

studs(> );

studs=> create table amber(

studs(> amber\_id SERIAL PRIMARY KEY,

studs(> company\_id INTEGER REFERENCES company(company\_id),

studs(> quantity INTEGER,

studs(> quality VARCHAR(25) not null

studs(> );

studs=> create table island(

studs(> island\_id SERIAL PRIMARY KEY,

studs(> company\_id INTEGER REFERENCES company(company\_id),

studs(> amber\_id INTEGER REFERENCES amber(amber\_id),

studs(> name VARCHAR(20) not null,

studs(> location VARCHAR(40) not null

studs(> );

create table supercomputer(

studs(> supercomputer\_id SERIAL PRIMARY KEY,

studs(> company\_id INTEGER REFERENCES company(company\_id),

studs(> model VARCHAR(25),

studs(> manufacturer VARCHAR(40) not null,

studs(> characteristics VARCHAR(40) not null

studs(> );

studs=> create table scientist(

studs(> scientist\_id SERIAL PRIMARY KEY,

studs(> company\_id INTEGER REFERENCES company(company\_id),

studs(> name VARCHAR(20) not null,

studs(> specialization VARCHAR(25) not null,

studs(> experience\_years INTEGER

studs(> );

studs=> create table dna(

studs(> dna\_id SERIAL PRIMARY KEY,

studs(> scientist\_id INTEGER REFERENCES scientist(scientist\_id),

studs(> amber\_id INTEGER REFERENCES amber(amber\_id),

studs(> source VARCHAR(40) not null,

studs(> characteristics VARCHAR(20) not null,

studs(> usage VARCHAR(20) not null

studs(> );

studs=> create table zoo(

studs(> zoo\_id SERIAL PRIMARY KEY,

studs(> name VARCHAR(20) not null,

studs(> location VARCHAR(40) not null

studs(> );

studs=> create table artificial\_animal(

studs(> artificial\_animal\_id SERIAL PRIMARY KEY,

studs(> company\_id INTEGER REFERENCES company(company\_id),

studs(> dna\_id INTEGER REFERENCES dna(dna\_id),

studs(> supercomputer\_id INTEGER REFERENCES supercomputer(supercomputer\_id),

studs(> zoo\_id INTEGER REFERENCES zoo(zoo\_id),

studs(> animal\_species VARCHAR(20) not null,

studs(> characteristics VARCHAR(40) not null,

studs(> creation\_method VARCHAR(40) not null

studs(> );

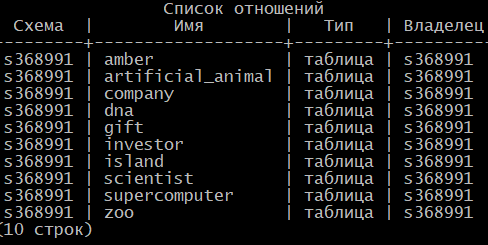
studs=> create table gift(

studs(> gift\_id SERIAL PRIMARY KEY,

studs(> company\_id INTEGER REFERENCES company(company\_id),

studs(> zoo\_id INTEGER REFERENCES zoo(zoo\_id)

studs(> );



# Дополнительные запросы

* **вывести ученых и каких животных они вывели**

SELECT u.name, a.animal\_species

FROM scientist u

JOIN dna ON u.scientist\_id = dna.scientist\_id

JOIN artificial\_animal a ON dna.dna\_id = a.dna\_id;

* **вывести вкладчика с наибольшим вложением и страну откуда он**

SELECT name, country, contribution

FROM investor

WHERE contribution = (SELECT MAX(contribution) FROM investor);

* **вывести страну которая вложилась больше всего**

SELECT country

FROM investor

GROUP BY country //групперуем вкладчиков по странам

ORDER BY SUM(contribution) DESC //подсчитает сумму вложений для каждой страны

LIMIT 1; // отсортирует результаты в порядке убывания суммы вложений и выберет первую страну

* **вывести только страну которая вложилась больше всего без суммы вклада**

SELECT c.name AS country\_name

FROM investor i

JOIN country c ON i.country\_id = c.country\_id

GROUP BY c.name

ORDER BY SUM(i.contribution) DESC

LIMIT 1;

* **вывести рейтинг вкладчиков с дополнительным столбиком об этом , страну откуда они и сумму их вложения по убыванию**

SELECT

investor.name,

investor.contribution,

country.name AS country,

RANK() OVER (ORDER BY investor.contribution DESC) AS rank,

SUM(investor.contribution) OVER (PARTITION BY country.name ORDER BY investor.contribution DESC) AS country\_contribution

FROM

investor

JOIN country ON investor.country\_id = country.country\_id

ORDER BY

rank ASC;