

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «Санкт-Петербургский национальный исследовательский
университет ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ
ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА №1

по дисциплине «Информационные системы и базы данных»

Вариант № 367101

Выполнил:

Студент группы Р3116

Билошицкий Михаил Владимирович

Преподаватель:

Горбунов Михаил Витальевич

Санкт-Петербург, 2023

Содержание

Содержание	2
Задание.....	3
Текст задания	4
Описание предметной области.....	4
Список сущностей и их классификация.....	4
Инфологическая модель.....	5
Даталогическая модель	6
Реализация даталогической модели на SQL	7
Вывод.....	11

Задание

Лабораторная работа #1

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.

Для создания объектов базы данных у каждого студента есть своя схема. Название схемы соответствует имени пользователя в базе studs (sXXXXXX). Команда для подключения к базе studs:

```
psql -h pg -d studs
```

Каждый студент должен использовать свою схему при работе над лабораторной работой №1 (а также в рамках выполнения 2, 3 и 4 этапа курсовой работы).

Отчёт по лабораторной работе должен содержать:

1. Текст задания.
2. Описание предметной области.
3. Список сущностей и их классификацию (стержневая, ассоциация, характеристика).
4. Инфологическая модель (ER-диаграмма в расширенном виде - с атрибутами, ключами...).
5. Даталогическая модель (должна содержать типы атрибутов, вспомогательные таблицы для отображения связей "многие-ко-многим").
6. Реализация даталогической модели на SQL.
7. Выводы по работе.

Темы для подготовки к защите лабораторной работы:

1. Архитектура ANSI-SPARC
2. Модель "Сущность-Связь". Классификация сущностей. Виды связей. Ограничения целостности.
3. DDL
4. DML

Тем временем "Нина" после зыбного восьмичасового падения приземлилась на поверхность Большого Брата. Василий рассчитал массу объекта - та оказалась поразительно малой, всего девятьсот пятьдесят тысяч тонн. Таким образом, его плотность примерно равнялась плотности воздуха. Возможно, он был пустотелым; излюбленной темой дискуссий стало его предполагаемое содержимое.

Текст задания

Тем временем "Нина" после затяжного восьмичасового падения приземлилась на поверхность Большого Брата. Василий рассчитал массу объекта - та оказалась поразительно малой, всего девятьсот пятьдесят тысяч тонн. Таким образом, его плотность примерно равнялась плотности воздуха. Возможно, он был пустотелым; излюбленной темой дискуссий стало его предполагаемое содержимое.

Описание предметной области

У нас есть персонаж Нина, которая после падения, которое длилось 8 часов, приземляется на поверхность объекта Большой брат, массу и плотность которого смог рассчитать другой персонаж Василий, 950000 тонн и 1.3 кг/м^3 соответственно. Вскоре завелась дискуссия, в которой участвовали два персонажа и обсуждали предполагаемое содержимое объекта Большой брат.

Список сущностей и их классификация

Сущности:

Стержневые:

1. Персонаж
2. Объект
3. Название объекта

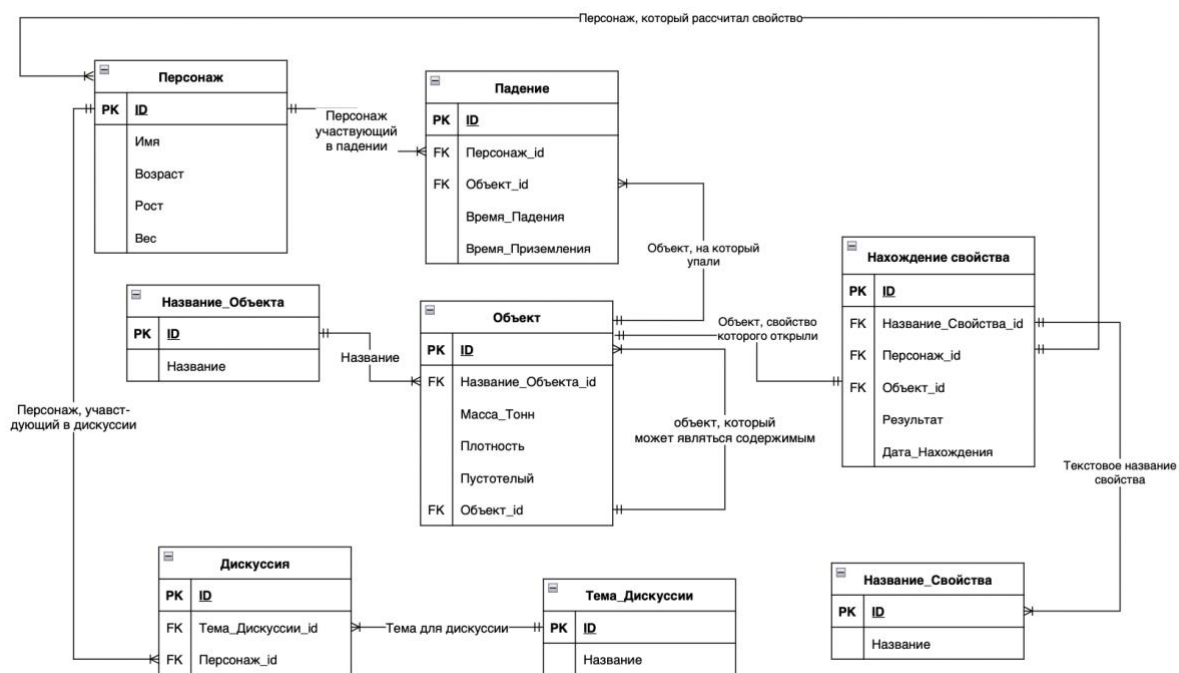
Ассоциативные:

1. Падение
2. Дискуссия
3. Нахождение свойства

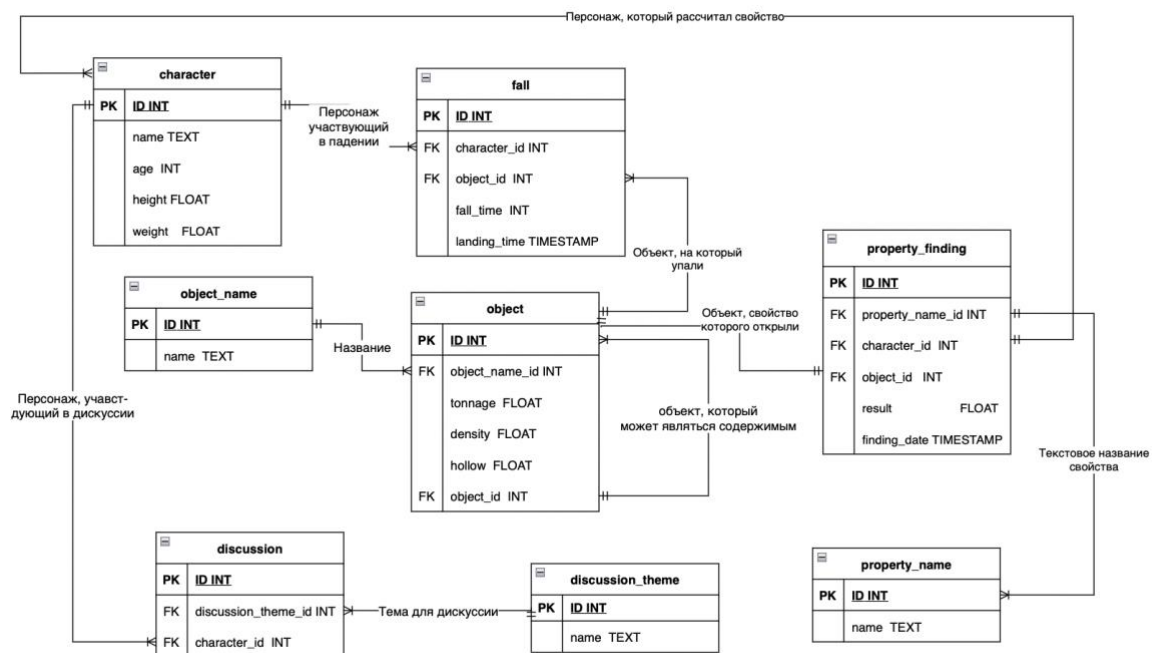
Характеристические:

1. Тема дискуссии
2. Название свойства

Инфологическая модель



Даталогическая модель



Реализация даталогической модели на SQL

```
DROP TABLE IF EXISTS "character" CASCADE;
DROP TABLE IF EXISTS "discussion" CASCADE;
DROP TABLE IF EXISTS "discussion_theme" CASCADE;
DROP TABLE IF EXISTS "fall" CASCADE;
DROP TABLE IF EXISTS "object" CASCADE;
DROP TABLE IF EXISTS "object_name" CASCADE;
DROP TABLE IF EXISTS "property_finding" CASCADE;
DROP TABLE IF EXISTS "property_name" CASCADE;

CREATE TABLE "character" (
    id SERIAL PRIMARY KEY,
    name text NOT NULL,
    age integer,
    height double precision,
    weight double precision);

CREATE TABLE "discussion" (
    id SERIAL PRIMARY KEY,
    discussion_theme_id integer NOT NULL,
    character_id integer NOT NULL);

CREATE TABLE "discussion_theme" (
    id SERIAL PRIMARY KEY,
    name text);

CREATE TABLE "fall" (
    id SERIAL PRIMARY KEY,
    character_id integer NOT NULL,
    object_id integer NOT NULL,
    fall_time integer NOT NULL,
    landing_time timestamp with time zone NOT NULL);

CREATE TABLE "object" (
    id SERIAL PRIMARY KEY,
    object_name_id integer NOT NULL,
    tonnage double precision,
    density double precision,
    hollow boolean NOT NULL,
```

```

    object_id integer);

CREATE TABLE "object_name" (
    id SERIAL PRIMARY KEY,
    name text NOT NULL);

CREATE TABLE "property_finding" (
    id SERIAL PRIMARY KEY,
    property_name_id integer NOT NULL,
    character_id integer NOT NULL,
    object_id integer NOT NULL,
    result double precision NOT NULL,
    finding_date timestamp with time zone NOT NULL);

CREATE TABLE "property_name" (
    id SERIAL PRIMARY KEY,
    name text NOT NULL);

ALTER TABLE ONLY "discussion"
    ADD CONSTRAINT discussion_character_id_fk FOREIGN KEY (character_id)
REFERENCES "character"(id) NOT VALID;
ALTER TABLE ONLY "discussion"
    ADD CONSTRAINT discussion_d_id_fk FOREIGN KEY (discussion_theme_id)
REFERENCES "discussion_theme"(id) NOT VALID;
ALTER TABLE ONLY "fall"
    ADD CONSTRAINT fall_character_id_fk FOREIGN KEY (character_id) REFERENCES
"character"(id) NOT VALID;
ALTER TABLE ONLY "fall"
    ADD CONSTRAINT fall_object_id_fk FOREIGN KEY (object_id) REFERENCES
"object"(id) NOT VALID;
ALTER TABLE ONLY "object"
    ADD CONSTRAINT object_name_id_fk FOREIGN KEY (object_name_id) REFERENCES
"object_name"(id) NOT VALID;
ALTER TABLE ONLY "object"
    ADD CONSTRAINT object_object_id_fk FOREIGN KEY (object_id) REFERENCES
"object"(id) NOT VALID;
ALTER TABLE ONLY "property_finding"
    ADD CONSTRAINT pf_character_id_fk FOREIGN KEY (character_id) REFERENCES
"character"(id) NOT VALID;

```



```

ALTER TABLE ONLY "property_finding"
    ADD CONSTRAINT pf_object_id_fk FOREIGN KEY (object_id) REFERENCES
"object"(id) NOT VALID;
ALTER TABLE ONLY "property_finding"
    ADD CONSTRAINT pf_property_name_id_fk FOREIGN KEY (property_name_id)
REFERENCES "property_name"(id) NOT VALID;

INSERT INTO "character"(name, age, height, weight) VALUES ('Нина', '19',
'160', '55'), ('Василий', '22', '178', '70');
INSERT INTO "discussion_theme"(name) VALUES ('Предполагаемое содержимое
объекта');
INSERT INTO "discussion"(discussion_theme_id, character_id) VALUES
((SELECT id FROM "discussion_theme" WHERE name='Предполагаемое содержимое
объекта'), (SELECT id FROM "character" WHERE name='Нина')),
((SELECT id FROM "discussion_theme" WHERE name='Предполагаемое содержимое
объекта'), (SELECT id FROM "character" WHERE name='Василий'));
INSERT INTO "object_name"(name) VALUES ('Большой брат'), ('Воздух');
INSERT INTO "object"(object_name_id, tonnage, density, hollow) VALUES
((SELECT id FROM "object_name" WHERE name='Воздух'), '1e-07', '1.2754',
'false' );
INSERT INTO "object"(object_name_id, tonnage, density, object_id, hollow)
VALUES
((SELECT id FROM "object_name" WHERE name='Большой брат'), '950000',
'1.3',
(SELECT id FROM "object" WHERE object_name_id=(SELECT id FROM
"object_name" WHERE name='Воздух')), 'true' );
INSERT INTO "fall"(character_id, object_id, fall_time, landing_time) VALUES
((SELECT id FROM "character" WHERE name='Нина'),
(SELECT id FROM "object" WHERE object_name_id=(SELECT id FROM
"object_name" WHERE name='Большой брат')),
'8', '2023-03-31 18:27:53.701886+03');
INSERT INTO "property_name"(name) VALUES ('масса'), ('плотность');
INSERT INTO "property_finding"(property_name_id, character_id, object_id,
result, finding_date) VALUES
((SELECT id FROM "property_name" WHERE name='масса'), (SELECT id FROM
"character" WHERE name='Василий'),
(SELECT id FROM "object" WHERE object_name_id=(SELECT id FROM
"object_name" WHERE name='Большой брат')),
'950000', '2023-03-30 18:27:53.701886+03');

```

```
INSERT INTO "property_finding"(property_name_id, character_id, object_id,
result, finding_date) VALUES
  ((SELECT id FROM "property_name" WHERE name='плотность'), (SELECT id FROM
"character" WHERE name='Василий'),
  (SELECT id FROM "object" WHERE object_name_id=(SELECT id FROM
"object_name" WHERE name='Большой брат')),
  '1.3', '2023-03-31 18:27:53.701886+03');
```

Вывод

В ходе проведения лабораторной работы я научился проектировать базы данных, описывая предметную область, выявляя сущности и формируя инфологическую и даталогическую ER – модели. Также закрепил навык реализации базы данных на физическом уровне на PostgreSQL. Полученные знания понадобятся нам в процессе дальнейшего обучения.