

Федеральное государственное автономное образовательное высшего образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники
Направление подготовки: 09.03.01 – Информатика и вычислительная техника

Отчет
по лабораторной работе №1
по дисциплине «Базы данных»

Вариант 310924

Выполнил:

Колмаков Дмитрий Владимирович, Р3131

Преподаватель:

Наумова Надежда Александровна

г. Санкт-Петербург, 2023 г.

Оглавление

Задание	3
Текст задания.....	3
Описание предметной области	3
Ход работы.....	4
Инфологическая модель.....	5
Даталогическая модель	5
Реализация БД на PostgreSQL.....	6
Вывод	9

Задание

Текст задания

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

Почему же тогда, подумал Дэвид, мы не сказали маме, куда идем, и дождались, пока папа уехал на очередной запуск? Всерьез он, конечно, не беспокоился: Бобби всегда знал, что делает. Наверно, здорово, когда тебе семнадцать и ты все знаешь. Вот только лучше бы он не тратил так много времени на эту глупую Бетти Шульц. Да, она красивая, но, черт возьми, она же девчонка! Улизнуть от нее сегодня едва удалось.

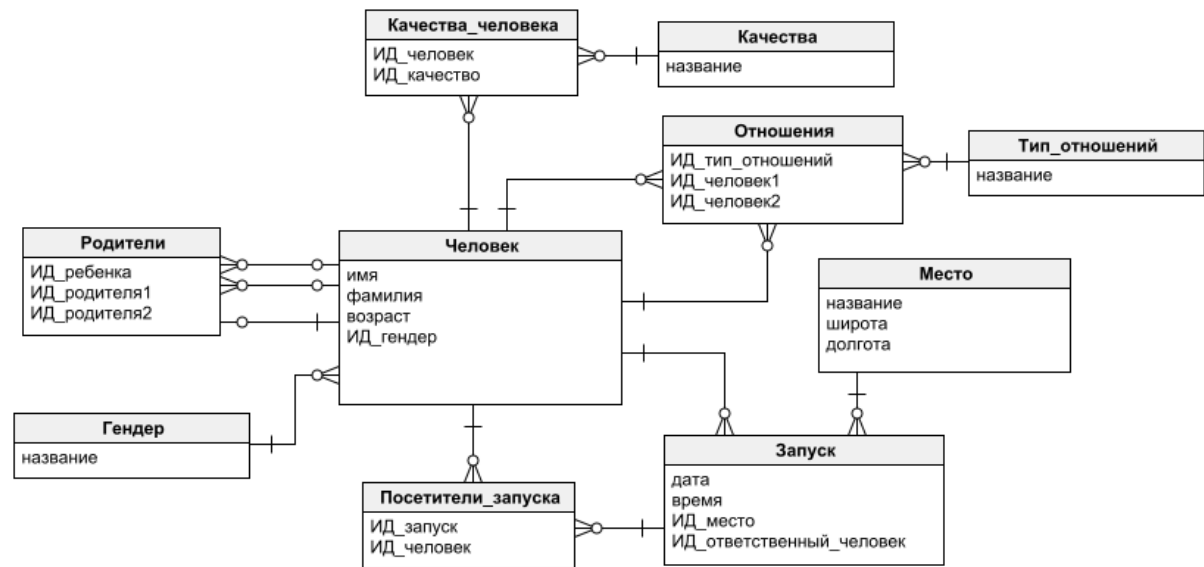
Ход работы

Есть люди: Дэвид, Бобби, Бэтти, мама и папа. Можем выделить сущность Человек. У нее есть имя, фамилия, возраст, гендер, их качества, а также два родителя - Человека. У одного Человека может быть один родитель-Человек, но у одного родителя-Человека может быть много детей-Человек связь, многие к одному. Выделим сущность Качества человека. У одного Человека может быть много Качеств, одно Качество может быть у многих Человек, значит связь многие ко многим. Выделим сущность Гендер. У одного Человека может быть один Гендер, один Гендер может быть у многих Человек, значит связь многие к одному. Также между людьми могут быть выстроены отношения: братья/сестры, дружеские или романтические. Выделим ассоциативную сущность Отношения между людьми, связь многие ко многим. У Отношений есть характеристика Тип отношений (братья/сестры, дружеские или романтические). Создадим сущность Тип отношений. Одни Отношения могут быть одного Типа, но один Тип может быть у многих Отношений, связь многие к одному. Можем выделить сущность Запуск (например, ракеты, корабля) с характеристиками место, дата, время, ответственный за запуск Человек (связь многие к одному). Для описания места выделим сущность Место с характеристиками название, широта, долгота. Связь Запуск-Место – многие к одному. Также у запуска могут быть посетители типа Человек, связь многие ко многим. Для этого выделим ассоциативную сущность Посетители запуска.

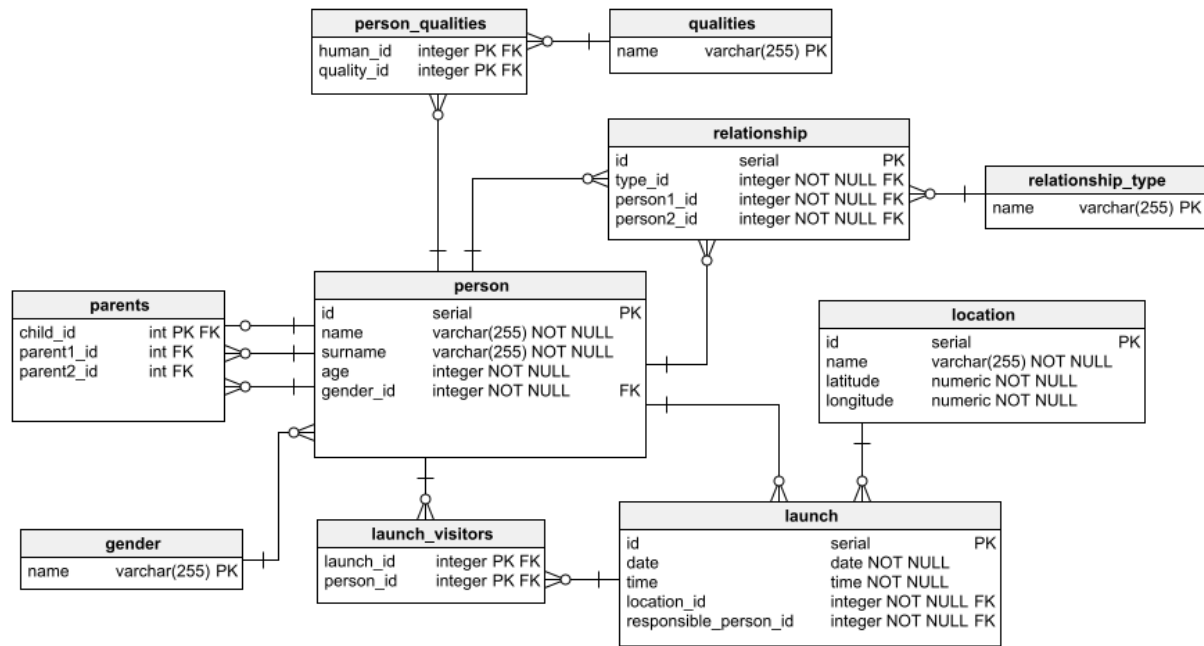
Сущности

- Стержневые
 - Человек (имя, фамилия, возраст, Гендер)
 - Родители (ребенок-Человек, родитель1-Человек, родитель2-Человек)
 - Запуск (дата, время, Место, ответственный-Человек)
 - Место (название, широта, долгота)
- Характеристики
 - Качества человека (название)
 - Гендер (название)
 - Тип отношений (название)
- Ассоциативные
 - Отношения (Тип отношений, Человек1, Человек2)
 - Качества человека (Человек, Качество)
 - Посетители запуска (Запуск, Человек)

Инфологическая модель



Даталогическая модель



Реализация БД на PostgreSQL

```
DROP TABLE IF EXISTS gender CASCADE;
DROP TABLE IF EXISTS person CASCADE;
DROP TABLE IF EXISTS parents CASCADE;
DROP TABLE IF EXISTS qualities CASCADE;
DROP TABLE IF EXISTS person_qualities CASCADE;
DROP TABLE IF EXISTS relationship_type CASCADE;
DROP TABLE IF EXISTS relationship CASCADE;
DROP TABLE IF EXISTS location CASCADE;
DROP TABLE IF EXISTS launch CASCADE;
DROP TABLE IF EXISTS launch_visitors CASCADE;

CREATE TABLE gender (
    name VARCHAR(255) PRIMARY KEY
);

CREATE TABLE person (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    surname VARCHAR(255) NOT NULL,
    age INTEGER DEFAULT 0 CHECK (age >= 0),
    gender_id VARCHAR(255) REFERENCES gender(name) ON DELETE CASCADE NOT NULL
);

CREATE TABLE parents (
    child_id INTEGER PRIMARY KEY REFERENCES person(id) ON DELETE CASCADE,
    parent1_id INTEGER REFERENCES person(id) ON DELETE SET NULL,
    parent2_id INTEGER REFERENCES person(id) ON DELETE SET NULL,
    CHECK (parent1_id != parent2_id),
    CHECK (parent1_id != child_id),
    CHECK (parent2_id != child_id)
);

CREATE TABLE qualities (
    name VARCHAR(255) PRIMARY KEY
);

CREATE TABLE person_qualities (
    human_id INTEGER REFERENCES person(id) ON DELETE CASCADE NOT NULL,
    quality_id VARCHAR(255) REFERENCES qualities(name) ON DELETE CASCADE NOT NULL,
    PRIMARY KEY (human_id, quality_id)
);

CREATE TABLE relationship_type (
    name VARCHAR(255) PRIMARY KEY
);

CREATE TABLE relationship (
    id SERIAL PRIMARY KEY,
    type_id VARCHAR(255) REFERENCES relationship_type(name) ON DELETE CASCADE NOT NULL,
    person1_id INTEGER REFERENCES person(id) ON DELETE CASCADE NOT NULL,
    person2_id INTEGER REFERENCES person(id) ON DELETE CASCADE NOT NULL
);

CREATE TABLE location (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    latitude NUMERIC DEFAULT 0 CHECK (latitude >= -90 AND latitude <= 90),
    longitude NUMERIC DEFAULT 0 CHECK (longitude >= -180 AND longitude <= 180)
);
```

```

CREATE TABLE launch (
    id SERIAL PRIMARY KEY,
    date DATE NOT NULL,
    time TIME WITH TIME ZONE NOT NULL,
    location_id INTEGER REFERENCES location(id) NOT NULL,
    responsible_person_id INTEGER REFERENCES person(id) ON DELETE CASCADE NOT
NULL
);

CREATE TABLE launch_visitors (
    launch_id INTEGER REFERENCES launch(id) ON DELETE CASCADE NOT NULL,
    person_id INTEGER REFERENCES person(id) ON DELETE CASCADE NOT NULL,
    PRIMARY KEY (launch_id, person_id)
);

INSERT INTO gender VALUES ('male');
INSERT INTO gender VALUES ('female');
INSERT INTO gender VALUES ('other');

INSERT INTO person (name, surname, age, gender_id) VALUES ('Мама',
'Sмирнова', 40, 'female');
INSERT INTO person (name, surname, age, gender_id) VALUES ('Папа', 'Смирнов',
45, 'male');
INSERT INTO person (name, surname, age, gender_id) VALUES ('Дэвид',
'Sмирнов', 16, 'male');
INSERT INTO person (name, surname, age, gender_id) VALUES ('Бобби',
'Sмирнов', 17, 'male');
INSERT INTO person (name, surname, age, gender_id) VALUES ('Бэтти', 'Шульц',
17, 'female');
INSERT INTO person (name, surname, age, gender_id) VALUES ('Джон', 'До', 50,
'male');

INSERT INTO parents VALUES (3, 1, 2);
INSERT INTO parents VALUES (4, 1, 2);

INSERT INTO qualities VALUES ('уверенность');
INSERT INTO qualities VALUES ('глупость');
INSERT INTO qualities VALUES ('красота');

INSERT INTO person_qualities VALUES (4, 'уверенность');
INSERT INTO person_qualities VALUES (5, 'глупость');
INSERT INTO person_qualities VALUES (5, 'красота');

INSERT INTO relationship_type VALUES ('friend');
INSERT INTO relationship_type VALUES ('romantic');
INSERT INTO relationship_type VALUES ('marriage');

INSERT INTO relationship (type_id, person1_id, person2_id) VALUES
('marriage', 1, 2);
INSERT INTO relationship (type_id, person1_id, person2_id) VALUES
('romantic', 4, 5);
INSERT INTO relationship (type_id, person1_id, person2_id) VALUES ('friend',
3, 4);

INSERT INTO location (name, latitude, longitude) VALUES ('Байконур', 45.6167,
63.3167);

INSERT INTO launch (date, time, location_id, responsible_person_id) VALUES
('2023.03.16', '12:00', 1, 6);

INSERT INTO launch_visitors VALUES (1, 2);

```

```
INSERT INTO launch_visitors VALUES (1, 6);
```

```
ALTER TABLE launch DROP CONSTRAINT launch_location_id_fkey, ADD CONSTRAINT  
launch_location_id_fkey FOREIGN KEY (location_id) REFERENCES location(id) ON  
DELETE CASCADE;
```


Вывод

В ходе проведения лабораторной работы я научился проектировать базы данных, описывая предметную область, выявляя сущности и формируя инфологическую и даталогическую модели. Закрепил навык реализации базы данных на физическом уровне на PostgreSQL.