

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Базы Данных

Лабораторная работа № 1

Вариант 888

Выполнил студент: Маликов Глеб Игоревич

Группа № Р3124

Преподаватель: Королёва Юлия Александровна

г. Санкт-Петербург

2023

| | |
|-----------------------------|---|
| Задание | 3 |
| Описание текста | 4 |
| Инфологическая модель | 5 |
| Даталогическая модель..... | 6 |
| SQL код..... | 7 |
| Заполнение данных | 8 |
| Заключение | 9 |

Задание

На основе предложенной предметной области (текста) составить ее описание. Из полученного описания выделить сущности, их атрибуты и связи.

Как глупо было ожидать этого! Теперь он знал, что с таким же успехом можно разглядывать ветер или размышлять об истинной форме пламени. Потом его душу и тело охватила безмерная усталость, и Дэвид Боумен заснул в последний раз в своей жизни.

Составить инфологическую модель.

Составить даталогическую модель. При описании типов данных для атрибутов должны использоваться типы из СУБД PostgreSQL.

Реализовать даталогическую модель в PostgreSQL. При описании и реализации даталогической модели должны учитываться ограничения целостности, которые характерны для полученной предметной области.

Заполнить созданные таблицы тестовыми данными.

Описание текста

Сущности:

- Природный Элемент:
 - Элемент
 - Форма
 - Количество
 - Единица Измерения
- Персонаж:
 - Имя
 - Дата рождения
 - Дата смерти
 - Состояние
 - Жив
- Событие:
 - Описание

Ассоциация

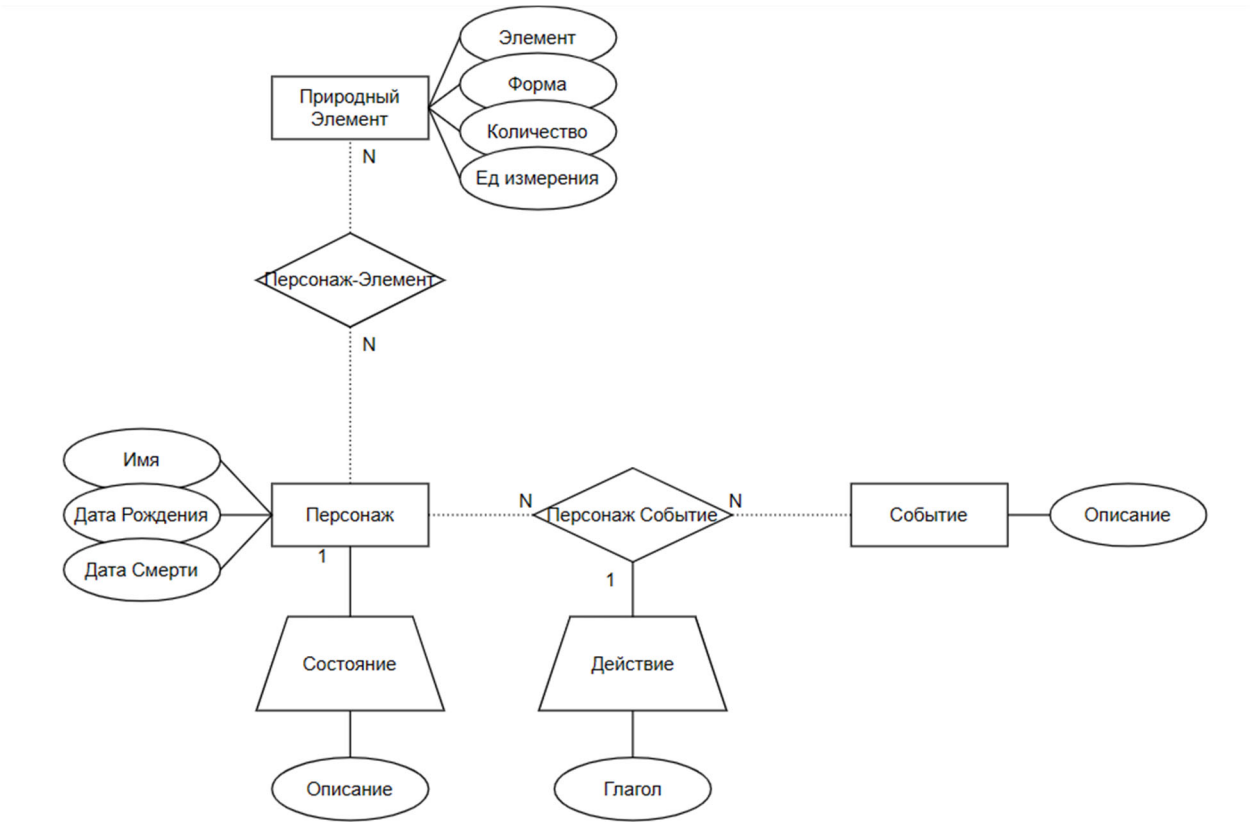
- Персонаж_Элемент:
 - Персонаж_ID
 - Элемент_ID
 - Действие_ID
- Персонаж_Событие:
 - Персонаж_ID
 - Событие_ID

Характеристика

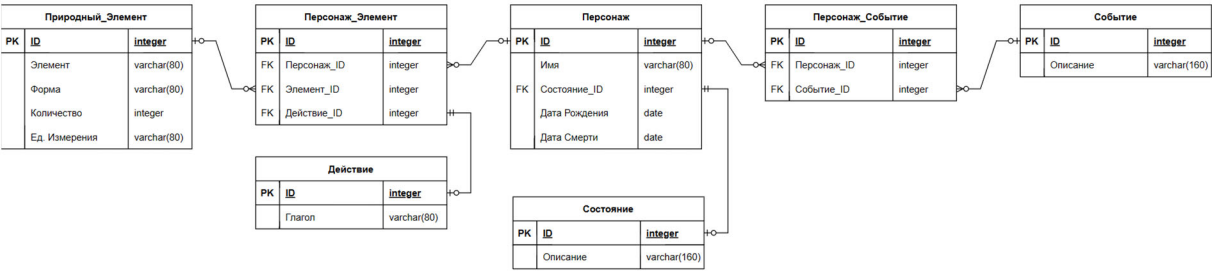
- Состояние:
 - Описание
- Действие:
 - Глагол

Модели

Инфологическая модель



Даталогическая модель



SQL код

create table Действие

```
(  
    id integer not null GENERATED ALWAYS AS IDENTITY primary key,  
    Глагол varchar(80) unique  
);
```

create table Состояние

```
(  
    id integer not null GENERATED ALWAYS AS IDENTITY primary key,  
    Описание varchar(160) unique  
);
```

create table Элемент

```
(  
    id integer not null GENERATED ALWAYS AS IDENTITY primary key,  
    Элемент varchar(80) not null,  
    Форма varchar(80),  
    Количество integer check (Количество > 0),  
    Единица_Измерения varchar(80)  
);
```

create table Персонаж

```
(  
    id integer not null GENERATED ALWAYS AS IDENTITY primary key,  
    Имя varchar(80) not null,  
    Состояние_ID INTEGER REFERENCES Состояние(id),  
    Дата_Рождения date,  
    Дата_Смерти date  
);
```

create table Событие

```
(  
    id integer not null GENERATED ALWAYS AS IDENTITY primary key,  
    Описание varchar(160) unique  
);
```

create table Персонаж_Событие

```
(  
    id integer not null GENERATED ALWAYS AS IDENTITY primary key,  
    Персонаж_ID integer references Персонаж(id),  
    Событие_ID integer references Событие(id),  
    UNIQUE (Персонаж_ID, Событие_ID)
```

```
);  
create table Персонаж_Элемент  
(  
    id integer not null GENERATED ALWAYS AS IDENTITY primary key,  
    Персонаж_ID integer references Персонаж(id),  
    Элемент_ID integer references Элемент(id),  
    Действие_ID integer references Действие(id)  
);
```

Заполнение данных

```
INSERT INTO Действие (Глагол) VALUES ('Разглядывать');
INSERT INTO Действие (Глагол) VALUES ('Расмышлять');
INSERT INTO Действие (Глагол) VALUES ('Смотреть');
INSERT INTO Действие (Глагол) VALUES ('Есть');
INSERT INTO Действие (Глагол) VALUES ('Отдыхать');
INSERT INTO Действие (Глагол) VALUES ('Строить');
INSERT INTO Действие (Глагол) VALUES ('Пить');
```

```
INSERT INTO Состояние (Описание) VALUES ('Неизвестное');
INSERT INTO Состояние (Описание) VALUES ('Усталость');
INSERT INTO Состояние (Описание) VALUES ('Радость');
INSERT INTO Состояние (Описание) VALUES ('Страх');
INSERT INTO Состояние (Описание) VALUES ('Грусть');
```

```
INSERT INTO Элемент (Элемент, Форма, Количество, Единица_Измерения)
VALUES ('Ветер', 'Северо-Западный', 32, 'м/с');
INSERT INTO Элемент (Элемент, Форма, Количество, Единица_Измерения)
VALUES ('Пламя', 'Истинная', 500, 'С');
INSERT INTO Элемент (Элемент, Форма, Количество, Единица_Измерения)
VALUES ('Вода', 'Жидкая', 2, 'Литр');
INSERT INTO Элемент (Элемент, Форма, Количество, Единица_Измерения)
VALUES ('Земля', 'Твёрдая', 500, 'Kg');
INSERT INTO Элемент (Элемент, Форма, Количество, Единица_Измерения)
VALUES ('Железо', 'Твёрдое', 1000, 'Kg');
```

```
INSERT INTO Персонаж (Имя, Состояние_ID, Дата_Рождения, Дата_Смерти)
VALUES ('Дэвид Боумен', 1, '01/01/1970', null);
INSERT INTO Персонаж (Имя, Состояние_ID, Дата_Рождения, Дата_Смерти)
VALUES ('Глеб', 5, '11/08/2004', null);
INSERT INTO Персонаж (Имя, Состояние_ID, Дата_Рождения, Дата_Смерти)
VALUES ('Juan', 1, '05/09/1004', null);
INSERT INTO Персонаж (Имя, Состояние_ID, Дата_Рождения, Дата_Смерти)
VALUES ('Гарри Поттер', 3, '08/02/1904', null);
INSERT INTO Персонаж (Имя, Состояние_ID, Дата_Рождения, Дата_Смерти)
VALUES ('Пётр', 4, '08/02/1904', null);
```

```
INSERT INTO Событие (Описание) VALUES ('Глупость Ожидания');
```



```

INSERT INTO Событие (Описание) VALUES ('Знание о том что с таким же
успехом можно разглядывать ветер или размышлять об истинной форме пла-
мени');
INSERT INTO Событие (Описание) VALUES ('Испытывание безмерной уста-
лости');
INSERT INTO Событие (Описание) VALUES ('Засыпание в последний раз
жизни.');
```

```

INSERT INTO Событие (Описание) values ('Конец мира');
```

```

INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (1,
1);
INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (1,
2);
INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (1,
3);
INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (1,
4);
INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (2,
3);
```

```

INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (1,
5);
INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (2,
5);
INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (3,
5);
INSERT INTO Персонаж_Событие (Персонаж_ID, Событие_ID) VALUES (5,
5);
```

```

INSERT INTO Персонаж_Элемент (Персонаж_ID, Элемент_ID, Действие_ID)
VALUES (1, 1, 1);
INSERT INTO Персонаж_Элемент (Персонаж_ID, Элемент_ID, Действие_ID)
VALUES (1, 2, 2);
INSERT INTO Персонаж_Элемент (Персонаж_ID, Элемент_ID, Действие_ID)
VALUES (5, 4, 6);
INSERT INTO Персонаж_Элемент (Персонаж_ID, Элемент_ID, Действие_ID)
VALUES (5, 5, 6);
```

```
INSERT INTO Персонаж_Элемент (Персонаж_ID, Элемент_ID, Действие_ID)  
VALUES (2, 3, 7);
```

```
UPDATE Персонаж SET Состояние_ID = 2 WHERE id = 1;  
UPDATE Персонаж SET Дата_Смерти = now() WHERE Персонаж.id = 1;  
UPDATE Персонаж SET Дата_Смерти = now() WHERE Персонаж.id = 2;  
UPDATE Персонаж SET Дата_Смерти = now() WHERE Персонаж.id = 3;  
UPDATE Персонаж SET Дата_Смерти = null WHERE Персонаж.id = 5;
```

Запросы

- Вывод количества живых и мёртвых персонажей

```
SELECT COUNT(*) FROM Персонаж WHERE Дата_Смерти is not null  
UNION ALL SELECT COUNT(*) FROM Персонаж WHERE Дата_Смерти is  
null;
```

- Персонажи и количество элементов, с которым взаимодействует

```
SELECT Персонаж.*, count(*) FROM Персонаж_Элемент, Персонаж  
WHERE Персонаж_ID = Персонаж.id group by Имя, Состояние_ID,  
Дата_Рождения, Дата_Смерти, Персонаж.id;
```

- Персонажи и количество элементов, с которым взаимодействует
(больше 1 элемента)

```
SELECT Персонаж.*, count(*) FROM Персонаж_Элемент, Персонаж  
WHERE Персонаж_ID = Персонаж.id group by Имя, Состояние_ID,  
Дата_Рождения, Дата_Смерти, Персонаж.id having count(*) > 1;
```

- Персонажи и элементы, с которым взаимодействует (вывод назва-
ния элементов)

```
SELECT A.Имя, Элемент.Элемент FROM Элемент join (SELECT *  
FROM Персонаж JOIN Персонаж_Элемент on Персонаж.id = Персонаж_Эле-  
мент.Персонаж_ID) AS A ON Элемент_ID = Элемент.id;
```

- Вывод всех Персонажей и количество элементов, с которыми вза-
имодействуют (включая тех, кто не взаимодействуют ни с одним
элементом)

```
SELECT Персонаж.*, count(Персонаж_ID = Персонаж.id) FROM Персо-  
наж_Элемент RIGHT JOIN Персонаж ON Персонаж_ID = Персонаж.id group  
by Персонаж.id, Имя, Состояние_ID, Дата_Рождения, Дата_Смерти;
```

- Вывод персонажей старше 50 лет

```
SELECT * from Персонаж where (now() - Дата_Рождения) > '50 YEARS';
```

- Вывести всех персонажей и добавить надпись "взаимодействует с
пламенем" для тех, кто взаимодействуют с пламенем

```
SELECT Персонаж.Имя,  
CASE WHEN EXISTS (SELECT * FROM Персонаж_Элемент WHERE
```

```
Персонаж_Элемент.Персонаж_ID = Персонаж.id AND Персонаж_Эле-  
мент.Элемент_ID = 2)  
    THEN 'взаимодействует с пламенем'  
    ELSE "  
    END AS Взаимодействие_с_элементом  
FROM Персонаж;
```

Заключение

В ходе работы была разработана база данных по описанию текста. Были выделены сущности, их атрибуты и связи. Далее была добавлена информация из текста в таблицы и выполнены запросы для отображения данной информации. База данных реализована в сервере Helios на PostgreSQL.