Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

ПИиКТ

Лабораторная работа 1

по дисциплине

«Информационные системы и базы данных»

Выполнили: Студенты группы P33113

Мансуров Б.Б.

Преподаватель: Николаев В.В.

Санкт-Петербург

2020г

# Задание

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

Вариант 618

Два темных неподвижных круга - чем еще они могли быть? Они придавали каждому погружению элемент риска: Чудовище в любой момент могло покинуть свое укрытие и, распугивая рыбешку, ринуться за более крупной добычей. И никто не убедил бы Бобби и Дэвида, что осока не скрывает ничего более опасного, чем, скажем, украденный велосипед... Тем не менее дно Хрустального источника оставалось недостижимым.

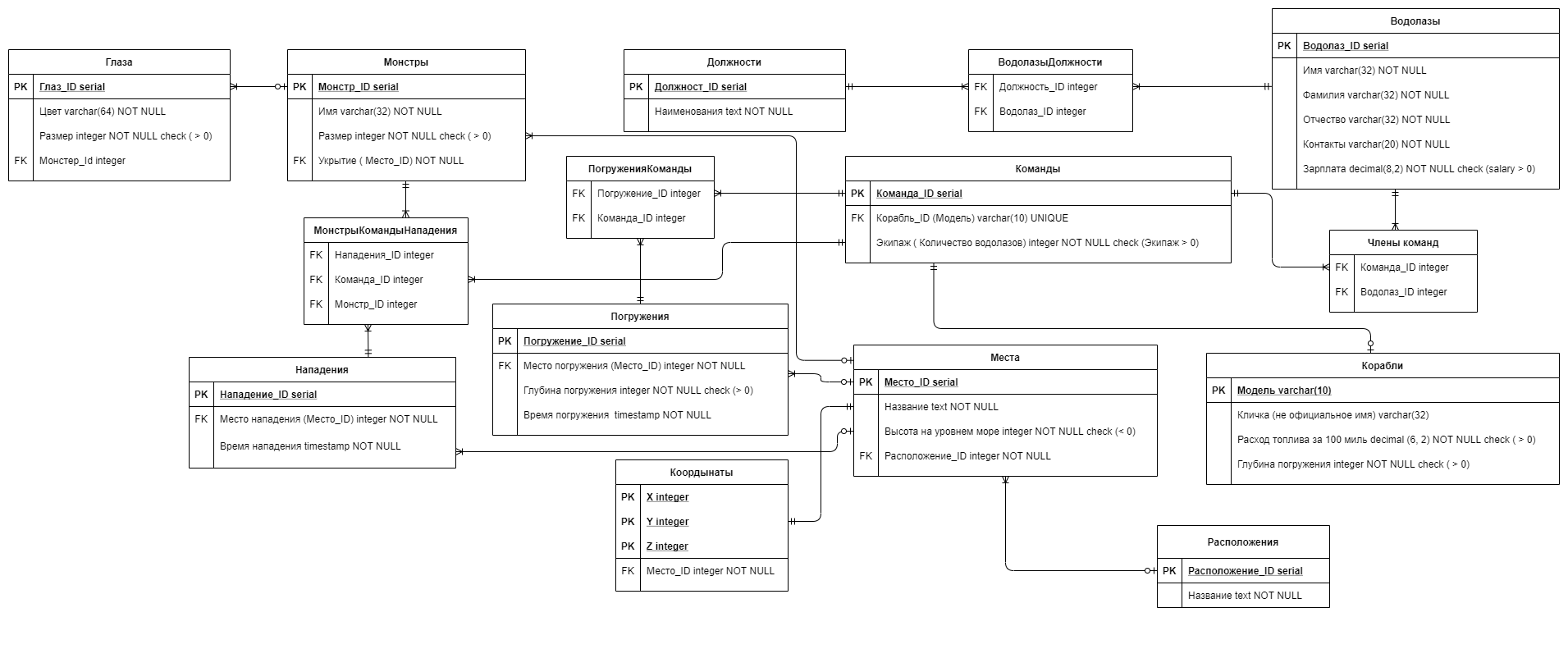
# Выполнение

Список сущностей и их классификация

1. Водолазы – стержневая сущность
2. Команда – ассоциативная сущность
3. Корабли –стержневая сущность
4. Монстры – ассоциативная сущность
5. Места – ассоциативная сущность
6. Расположения – стержневая сущность
7. Координаты – характеристическая сущность
8. Погружения – ассоциативная сущность
9. Должности – стержневая сущность
10. Глаза – характеристическая сущность
11. Нападения – ассоциативная сущность

# 

# Даталогическая модель



# Реализация

CREATE TABLE divers  
(  
 id\_diver serial,  
 name varchar(32) NOT NULL,  
 surname varchar(32) NOT NULL,  
 patronymic varchar(32) NOT NULL,  
 contacts varchar(20) NOT NULL,  
 salary decimal(8, 2) NOT NULL CHECK (salary > 0),  
 PRIMARY KEY (id\_diver)  
);  
  
CREATE TABLE positions  
(  
 id\_position serial,  
 name text NOT NULL,  
 PRIMARY KEY (id\_position)  
);  
  
CREATE TABLE div\_position  
(  
 id\_position integer,  
 id\_diver integer,  
 FOREIGN KEY (id\_position) REFERENCES positions (id\_position)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (id\_diver) REFERENCES divers (id\_diver)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_position, id\_diver)  
);  
  
CREATE TABLE ships  
(  
 model varchar(10),  
 name varchar(32),  
 fuel decimal(6, 2) NOT NULL CHECK (fuel > 0),  
 depth integer NOT NULL check (depth > 0),  
 PRIMARY KEY (model)  
);  
  
CREATE TABLE teams  
(  
 id\_team serial,  
 id\_ship varchar(10) NOT NULL UNIQUE,  
 crew integer NOT NULL CHECK (crew > 0),  
 PRIMARY KEY (id\_team),  
 FOREIGN KEY (id\_ship) REFERENCES ships (model)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE  
);  
  
CREATE TABLE team\_mem  
(  
 id\_team integer NOT NULL,  
 id\_diver integer NOT NULL,  
 FOREIGN KEY (id\_team) REFERENCES teams (id\_team)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (id\_diver) REFERENCES divers (id\_diver)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_team, id\_diver)  
);  
  
CREATE TABLE locations  
(  
 id\_location serial,  
 name text NOT NULL,  
 PRIMARY KEY (id\_location)  
);  
  
CREATE TABLE places  
(  
 id\_place serial,  
 name text NOT NULL,  
 altitude integer NOT NULL CHECK (altitude < 0),  
 id\_location integer NOT NULL,  
 FOREIGN KEY (id\_location) REFERENCES locations (id\_location)  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_place)  
);  
  
CREATE TABLE coordinates  
(  
 X integer,  
 Y integer,  
 Z integer,  
 id\_place integer NOT NULL UNIQUE,  
 FOREIGN KEY (id\_place) REFERENCES places (id\_place)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (X, Y, Z)  
);  
  
CREATE TABLE divings  
(  
 id\_diving serial,  
 id\_place integer NOT NULL,  
 depth integer NOT NULL check (depth > 0),  
 d\_time timestamp NOT NULL,  
 FOREIGN KEY (id\_place) REFERENCES places (id\_place)  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_diving)  
);  
  
CREATE TABLE divs\_team  
(  
 id\_diving integer NOT NULL,  
 id\_team integer NOT NULL,  
 FOREIGN KEY (id\_team) REFERENCES teams (id\_team)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (id\_diving) REFERENCES divings (id\_diving)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_team, id\_diving)  
);  
  
CREATE TABLE monsters  
(  
 id\_monster serial,  
 name varchar(32) NOT NULL,  
 size integer NOT NULL CHECK (size > 0),  
 id\_place integer NOT NULL,  
 FOREIGN KEY (id\_place) REFERENCES places (id\_place)  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_monster)  
);  
  
CREATE TABLE eyes  
(  
 id\_eye serial,  
 color varchar(32) NOT NULL,  
 size integer NOT NULL CHECK (size > 0),  
 id\_monster integer NOT NULL,  
 FOREIGN KEY (id\_monster) REFERENCES monsters (id\_monster)  
 ON UPDATE CASCADE  
 ON DELETE CASCADE,  
 PRIMARY KEY (id\_eye)  
);  
  
CREATE TABLE attacks  
(  
 id\_attack serial,  
 id\_place integer NOT NULL,  
 a\_time timestamp NOT NULL,  
 FOREIGN KEY (id\_place) REFERENCES places (id\_place)  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_attack)  
);  
  
CREATE TABLE monst\_attack\_on\_team  
(  
 id\_attack integer,  
 id\_team integer,  
 id\_monster integer,  
 FOREIGN KEY (id\_attack) REFERENCES attacks (id\_attack)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (id\_team) REFERENCES teams (id\_team)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 FOREIGN KEY (id\_monster) REFERENCES monsters (id\_monster)  
 ON DELETE CASCADE  
 ON UPDATE CASCADE,  
 PRIMARY KEY (id\_attack, id\_team, id\_monster)  
);

# Вывод:

При выполнение данной лабораторной работы были изучены сущности и их классификация, инфологическая модель, даталогическая модель, основы postgresql.