

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

Лабораторная работа 1
по дисциплине
«Информационные системы и базы данных»

Выполнили: Студенты группы Р33113

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

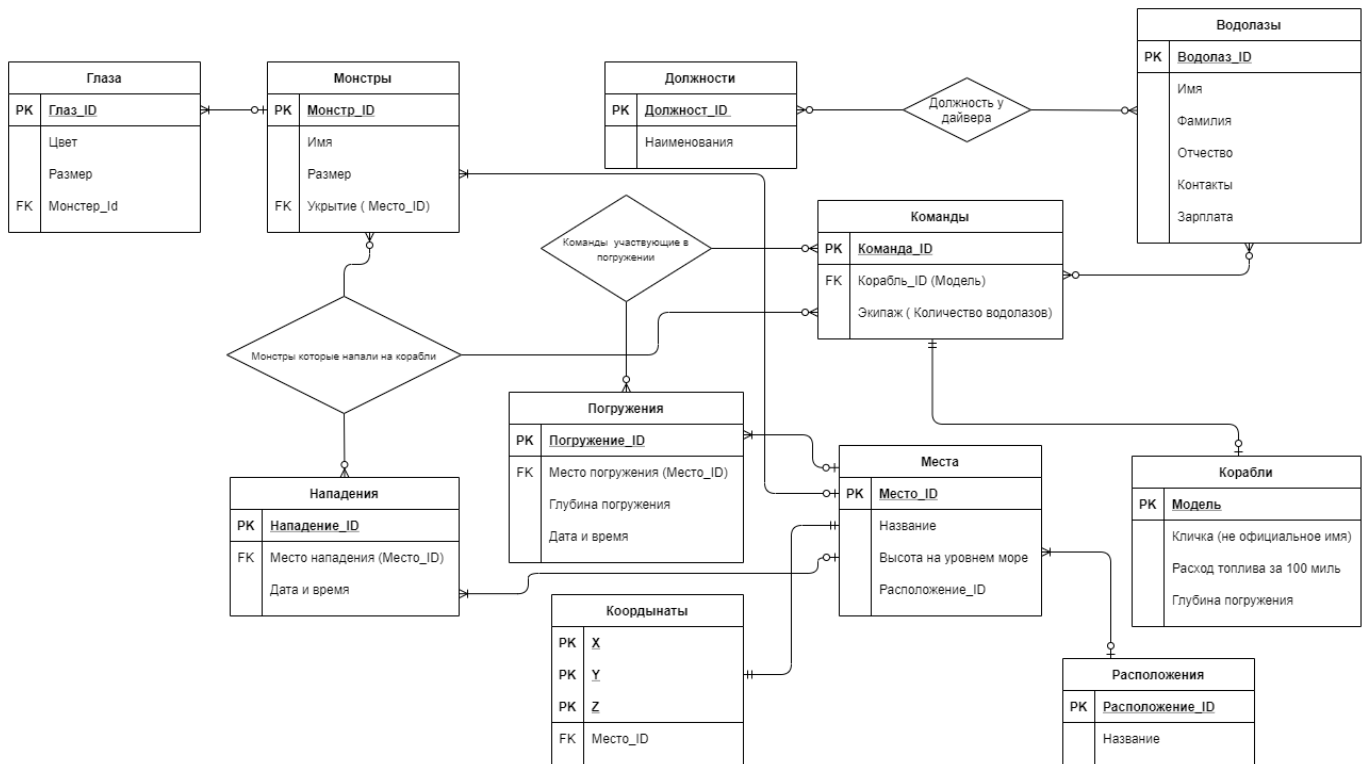
Вариант 618

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.

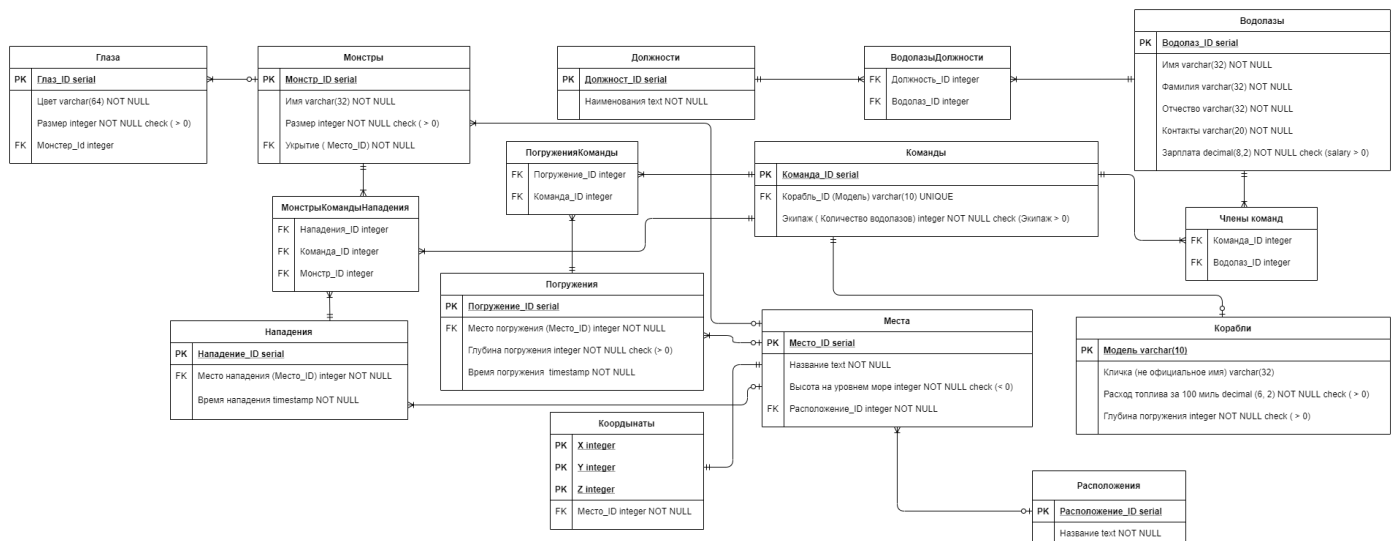
Выполнение

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 diversings
(
    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 divers_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 diversings (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 выполнении данной лабораторной работы были изучены сущности и их классификация, инфологическая модель, даталогическая модель, основы postgresql.