**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники

Кафедра Информатики и прикладной математики

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Базы Данных»

**Курсовая работа**

**«Хакатон»**

Группа P3117

Студент Царьков Григорий Иванович

Санкт-Петербург

2018 г.

1. Предметная область

Хакатон - соревнование по созданию идей и их прототипированию. Обычно он занимает 24 или 48 часов. Все люди на хакатоне делятся на **команды**, **менторов** и **жюри**. Ещё на хакатоне могут присутствовать зрители.

**Операторы** хакатона занимаются его организацией, поставкой питания, организационными моментами. Они не являются ни командами, ни менторами, ни жюри. Однако набором команд, менторов и жюри занимаются именно они.

**Тематика** хакатона всегда определена перед началом - например, AI, HR, Digital health. На некоторых хакатонах могут быть задания, которым надо строго следовать. Но обычно дается лишь общий вектор развития. Жюри **оценивает результаты** по таким критериям, как “инновационность”, “бизнес-составляющая”, и другим.

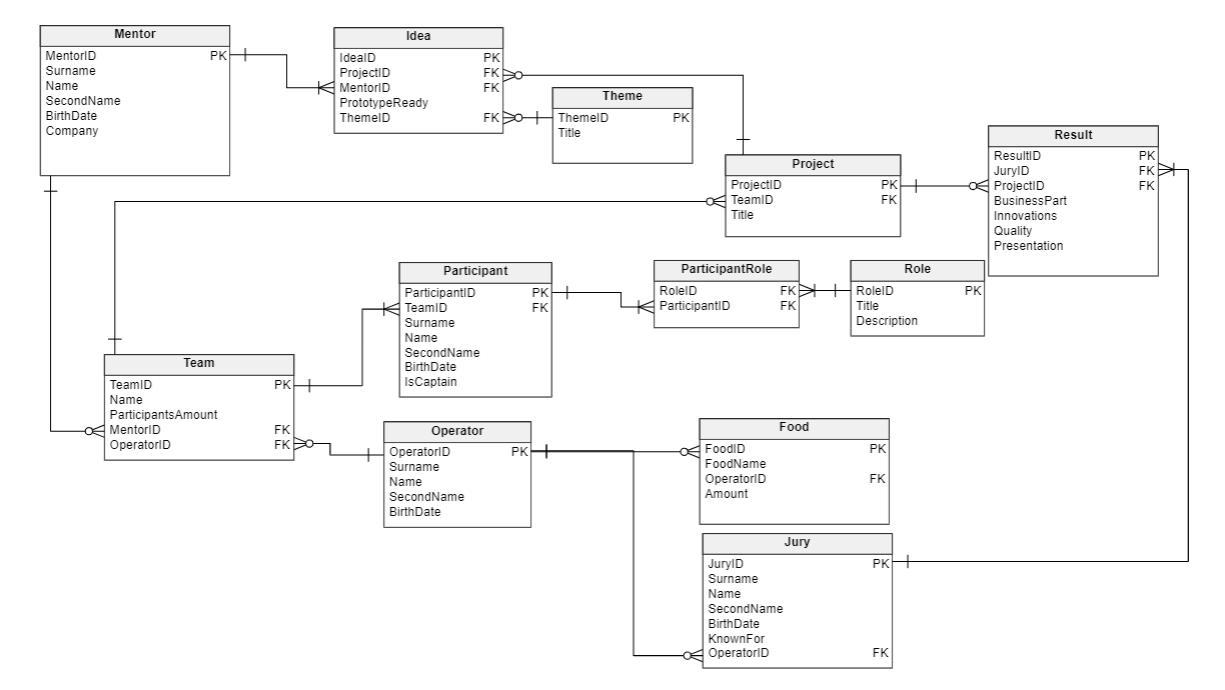
Менторы помогают командам с проработкой **идей**. Обычно они работают в крупных компаниях.

У каждой команды есть название, **члены**, капитан. А вот **проекта** изначально может и не быть. Но к концу хакатона он обязательно появится. Один проект может реализовывать сразу несколько идей

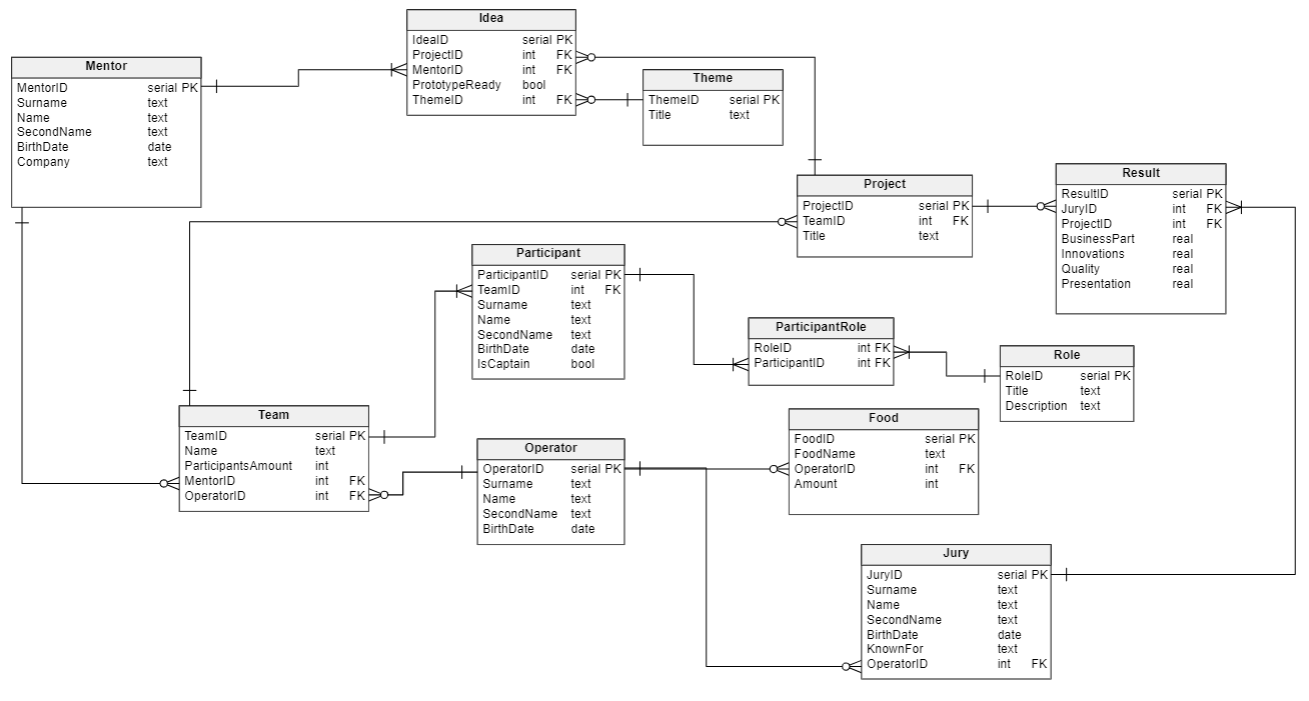
На хакатонах в командах может быть от 1 до 5, в редких случаях до 8 человек. Они распределены на роли - дизайнер, программист frontend, программист backend, человек, который показывает результат, специалист по custdev, капитан. Человек может одновременно занимать несколько ролей.

Важная часть хакатона - **питание** участников. Обычно в сутки умещается 3-4 приема пищи. Для операторов хакатона важно вести учет посещаемости, чтобы ресурсы не пропадали. Питаются не только команды, но и менторы, и жюри.

1. Инфологическая модель



1. Даталогическая модель



1. Реализация модели в СУБД

CREATE TABLE Operator (

OperatorID serial PRIMARY KEY,

Surname text NOT NULL,

Name text NOT NULL,

SecondName text,

BirthDate date

);

CREATE TABLE Mentor (

MentorID serial PRIMARY KEY,

Surname text NOT NULL,

Name text NOT NULL,

SecondName text,

BirthDate date,

Company text

);

CREATE TABLE Team (

TeamID serial PRIMARY KEY,

Name text NOT NULL,

ParticipantsAmount integer

CHECK (ParticipantsAmount>=1 and ParticipantsAmount<=8),

MentorID integer REFERENCES Mentor,

OperatorID integer REFERENCES Operator

);

CREATE TABLE Role (

RoleID serial PRIMARY KEY,

Title text NOT NULL,

Description text NOT NULL

);

CREATE TABLE Participant (

ParticipantID serial PRIMARY KEY,

TeamID integer REFERENCES Team,

Surname text NOT NULL,

Name text NOT NULL,

SecondName text,

BirthDate date,

IsCaptain boolean DEFAULT false

);

CREATE TABLE ParticipantRole (

RoleID integer REFERENCES Role,

ParticipantID integer REFERENCES Participant

);

CREATE TABLE Project (

ProjectID serial PRIMARY KEY,

TeamID integer REFERENCES Team,

Title text NOT NULL

);

CREATE TABLE Theme (

ThemeID serial PRIMARY KEY,

Title text NOT NULL

);

CREATE TABLE Idea (

IdeaID serial PRIMARY KEY,

ProjectID integer REFERENCES Project,

MentorID integer REFERENCES Mentor,

PrototypeReady bool,

ThemeID integer REFERENCES Theme

);

CREATE TABLE Jury (

JuryID serial PRIMARY KEY,

Surname text NOT NULL,

Name text NOT NULL,

SecondName text,

BirthDate date,

KnownFor text,

OperatorID integer REFERENCES Operator

);

CREATE TABLE Food (

FoodID serial PRIMARY KEY,

FoodName text NOT NULL,

OperatorID integer REFERENCES Operator,

Amount integer NOT NULL

);

CREATE TABLE Result (

ResultID serial NOT NULL,

JuryID integer REFERENCES Jury,

ProjectID integer REFERENCES Project,

BusinessPart real DEFAULT 0,

Innovations real DEFAULT 0,

Quality real DEFAULT 0,

Presentation real DEFAULT 0

);

1. Использование триггеров
2. Проверка целостности данных о дате рождения

CREATE FUNCTION check\_birthtime() RETURNS trigger AS $$

BEGIN

IF (NEW.BirthDate >= CURRENT\_DATE) THEN

RAISE EXCEPTION 'BirthDate is bigger than today’s date';

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER birthtime BEFORE INSERT OR UPDATE ON Operator

FOR EACH ROW EXECUTE PROCEDURE check\_birthtime();

1. Проверка целостности данных оценивания

CREATE OR REPLACE FUNCTION check\_positive() RETURNS trigger AS $$

BEGIN

IF (NEW.BusinessPart < 0) OR (NEW.Innovations < 0) OR (NEW.Quality < 0) OR (NEW.Presentation < 0) THEN

RAISE EXCEPTION 'result number % cannot have a negative amount of points',

NEW.ResultID;

END IF;

RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER positive BEFORE INSERT OR UPDATE ON Result

FOR EACH ROW EXECUTE PROCEDURE check\_positive();

1. Заполнение базы данными

INSERT INTO Operator VALUES (1, 'Ворошилов', 'Владимир', 'Яковлевич', '1930-12-18'), (2, 'Крюк', 'Борис', 'Александрович', '1966-08-18');

INSERT INTO Mentor VALUES(1, 'Цукерберг', 'Марк', 'Эллиот', '1984-05-14', 'Facebook');

INSERT INTO Team VALUES (1, 'Команда победителей', 6, 1, 1),

(2, 'Тибетские тушканчики', 6, 1, 2);

INSERT INTO Role VALUES (1, 'Дизайнер', 'Художник-конструктор'),

(2, 'Программист Frontend', 'Разработчик пользовательского интерфейса'),

(3, 'Программист Backend', 'Разработчик скрытой части приложения'),

(4, 'Специалист по custdev', 'Тестирование продукта на потенциальных потребителях'),

(5, 'Презентующий результат', 'Отчитывающийся за результаты работы');

INSERT INTO Participant VALUES (1, 1, 'Белозёров', 'Борис', 'Сергеевич', '1994-12-02', true),

(2, 1, 'Быкова', 'Ольга', 'Викторовна', '1994-09-21', false),

(3, 1, 'Любинская', 'Дарья', 'Олеговна', '1992-06-16', false),

(4, 1, 'Иоков', 'Антон', 'Викторович', '1994-08-14', false),

(5, 1, 'Старун', 'Никита', 'Сергеевич', '1992-10-25', false),

(6, 1, 'Галачян', 'Ким', NULL, '1988-08-11', false),

(7, 2, 'Панайотти', 'Дмитрий', 'Владимирович', '1987-05-15', false),

(8, 2, 'Самулев', 'Алексей', 'Антонович', '1978-03-21', false),

(9, 2, 'Марышев', 'Иван', 'Михайлович', '1987-12-23', false),

(10, 2, 'Лагутин', 'Денис', 'Ильич', '1986-10-13', false),

(11, 2, 'Филиппов', 'Юрий', 'Алексеевич', '1986-01-05', false),

(12, 2, 'Повышева', 'Алена', 'Алексеевна', '1987-12-23', true);

INSERT INTO ParticipantRole VALUES (1, 12), (5, 1), (5, 12), (2, 2), (3, 5),

(3, 7), (4, 3), (4, 8), (3, 4), (3, 6), (3, 9), (2, 10), (3, 11);

INSERT INTO Project VALUES (1, 1, 'Красно-черное дерево'),

(2, 2, 'Левосторонняя куча');

INSERT INTO Theme VALUES (1, '.NET Программирование');

INSERT INTO Idea VALUES (1, 1, 1, true, 1), (2, 2, 1, false, 1);

INSERT INTO Jury VALUES (1, 'Худяков', 'Андрей', 'Алексеевич', '1978-09-12', '', 1),

(2, 'Романов', 'Елисей', 'Святославович', '1990-11-20', '', 2),

(3, 'Гусман', 'Юлий', 'Соломонович', '1943-08-08', '', 1);

INSERT INTO Food VALUES (1, 'Кофе, л', 1, 300), (2, 'Печеньки, шт', 2, 1000);

INSERT INTO Result VALUES (1, 1, 1, 5, 6, 7, 8), (2, 1, 2, 3, 4, 2, 9),

(3, 2, 1, 7, 4, 6, 5), (4, 2, 2, 4, 6, 5.3, 8), (5, 3, 1, 7.2, 3, 5, 0),

(6, 3, 2, 6, 6, 6, 6);

1. Вывод

Выполняя данную курсовую работу, я более детально ознакомился с языком SQL и PostgreSQL в частности, а также сумел разработать свою собственную базу данных, имеющую прикладное предназначение.