

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Московский авиационный институт
(национальный исследовательский университет)

Институт № 8
Компьютерные науки и прикладная математика
Кафедра 806 «Вычислительная математика и программирование»

ЛАБОРАТОРНАЯ РАБОТА №1
по дисциплине «Базы Данных»
Вариант 6

Выполнил: студент группы М8О-310Б-20

Грубенко Максим Дмитриевич

(Фамилия, имя, отчество)

(подпись)

Приняла: Чумакова Екатерина Витальевна

(Фамилия, имя, отчество)

(подпись)

Оценка:

Дата:

Москва, 2022

1 Техническое задание

Для предложенной части реального мира (перечисленные сущности обязательно должны присутствовать) разработать концептуальную и логическую модели БД. Модель должна обязательно содержать указанные в варианте сущности, допускается добавление дополнительных сущностей при необходимости. Написать соответствующие sql-запросы на создание таблиц физической БД. Вариант выполняемого задания должен строго соответствовать номеру варианта в списке группы. Отчет должен содержать схемы концептуальной и логической моделей БД, описание атрибутов, доменов.

Вариант: программа, мероприятие, участник, организатор.

2 Решение

```
CREATE TABLE IF NOT EXISTS programs (  
    id SERIAL PRIMARY KEY,  
    program_name VARCHAR(50) NOT NULL,  
    description TEXT NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS events (  
    id SERIAL PRIMARY KEY,  
    event_name VARCHAR(70) NOT NULL,  
    start_time TIME NOT NULL,  
    duration INTEGER NOT NULL,  
    program_id INT NOT NULL REFERENCES programs(id)  
);  
  
CREATE TABLE IF NOT EXISTS participants (  
    id SERIAL PRIMARY KEY,  
    first_name VARCHAR(20) NOT NULL,  
    last_name VARCHAR(20) NOT NULL,  
    email VARCHAR(30) NOT NULL UNIQUE  
);  
  
CREATE TABLE IF NOT EXISTS participants_of_events (  
    participant_id INT REFERENCES participants(id) NOT NULL ,  
    event_id INT REFERENCES events(id) NOT NULL ,  
    PRIMARY KEY (participant_id, event_id)  
);  
  
CREATE TABLE IF NOT EXISTS companies (  
    id SERIAL PRIMARY KEY,  
    company_name VARCHAR(20) NOT NULL UNIQUE  
);
```

```

CREATE TABLE IF NOT EXISTS organizers (
    id SERIAL PRIMARY KEY,
    first_name VARCHAR(20) NOT NULL,
    last_name VARCHAR(20) NOT NULL,
    email VARCHAR(30) NOT NULL UNIQUE,
    company_id INT NOT NULL REFERENCES companies(id)
);

CREATE TABLE IF NOT EXISTS organizers_of_events (
    organizer_id INT REFERENCES organizers(id) NOT NULL ,
    event_id INT REFERENCES events(id) NOT NULL ,
    PRIMARY KEY (organizer_id, event_id)
);

```

В результате выполнения всех запросов получаем модель базы данных:

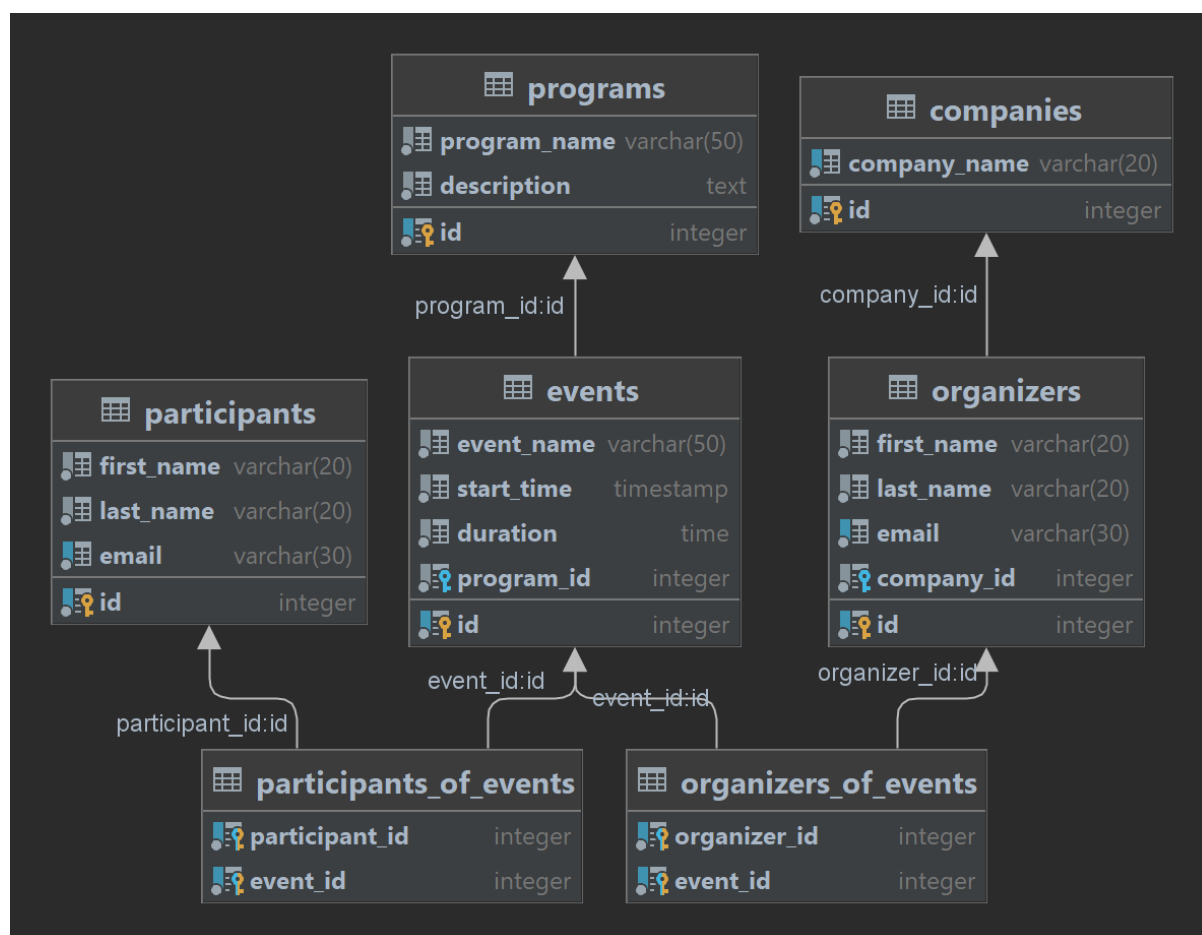


Рис. 1: ER-модель

Таблица **participants** хранит данные об участниках мероприятий. Помимо идентификатора в таблице присутствуют следующие атрибуты:

1. **first_name** (VARCHAR) - имя участника;

2. `last_name` (VARCHAR) - фамилия участника;
3. `email` (VARCHAR) - электронная почта участника.

Таблица `events` хранит о мероприятиях. Помимо идентификатора в таблице присутствуют следующие атрибуты:

1. `event_name` (VARCHAR) - название мероприятия;
2. `start_time` (TIMESTAMP) - время начала мероприятия;
3. `duration` (INTEGER) - длительность мероприятия, в минутах;
4. `program_id` (INTEGER FOREIGN KEY) - внешний ключ, связан с таблицей `programs`.

Таблица `programs` хранит данные о программах мероприятий. Помимо идентификатора в таблице присутствуют следующие атрибуты:

1. `program_name` (VARCHAR) - название программы мероприятий;
2. `description` (VARCHAR) - описание программы мероприятий.

Таблица `organizers` хранит данные об организаторах мероприятий. Помимо идентификатора в таблице присутствуют следующие атрибуты:

1. `first_name` (VARCHAR) - имя организатора;
2. `last_name` (VARCHAR) - фамилия организатора;
3. `email` (VARCHAR) - электронная почта организатора;
4. `company_id` (INTEGER FOREIGN KEY) - идентификатор компании, в которой работает организатор.

Таблица `companies` хранит данные о компании, в которую трудоустроены организаторы мероприятий. Содержит в себе идентификатор компании `id` (INTEGER PRIMARY KEY) и название компании `company_name` (VARCHAR).

Таблицы `participants_of_events` и `organizers_of_events` служат для организации связи many-to-many между таблицами `participants – events` и `organizers – events` соответственно.