**Организация мероприятий — автоматизированная система**

**Введение**

В данной работе была разработана информационная система для организации мероприятий. Система позволяет отображать ближайшие мероприятия, регистрировать участников, управлять списком мероприятий и площадок через панель администратора.

**Цель проекта** — автоматизировать процесс записи на мероприятия и контроля за их заполненностью.

**2. Основная часть**

**2.1. Схема решения, описание процессов, информационные потоки**

Система состоит из следующих компонентов:

* Главная страница – показывает список ближайших мероприятий.
* Форма регистрации участника – пользователь вводит имя и email.
* Форма записи на мероприятие – после регистрации участник может выбрать мероприятие и записаться на него.
* Панель администратора – добавление, редактирование и удаление мероприятий и площадок, просмотр списка участников на каждое мероприятие.
* Личный кабинет участника – просмотр мероприятий, на которые он записан.

📥 Информационные потоки:

* Участник → Регистрация → База данных.
* Участник → Регистрация на мероприятие → Проверка заполненности → БД.
* Администратор → Управление мероприятиями / площадками → БД.
* Пользователь → Просмотр мероприятий → Получение из БД.

**2.2. Структура базы данных**

База данных состоит из следующих таблиц:

* participant (id, name, email)
* venue (id, name, address, capacity)
* event (id, name, description, age\_limit, start\_time, end\_time, venue\_id)
* registration (id, participant\_id, event\_id)

Связи:

Один участник может быть зарегистрирован на много мероприятий.

Одно мероприятие проводится только на одной площадке.

В одной площадке может проходить много мероприятий, но без пересечений по времени.

Все связи корректно реализованы с помощью внешних ключей. База данных приведена к 3 нормальной форме (3НФ): нет избыточности, дублирования и многозначных зависимостей.

**2.3. Описание алгоритма расчёта заполненности**

**Заполненность мероприятия** рассчитывается как:

заполненность = (число зарегистрированных участников / вместимость площадки) \* 100

**Алгоритм:**

* Получаем количество регистраций на мероприятие.
* Получаем вместимость площадки.
* Если заполненность < 100%, разрешаем регистрацию.
* Иначе выводим сообщение "Мест нет".

**Псевдокод:**

**capacity = get\_capacity(venue\_id)**

**current = count\_registered(event\_id)**

**if current < capacity:**

**allow\_registration()**

**else:**

**deny\_registration()**

**2.4. SQL-запросы**

Вывод всех мероприятий:

SELECT e.id, e.name, e.description, e.start\_time, v.name AS venue\_name, v.address

FROM event e

JOIN venue v ON e.venue\_id = v.id

ORDER BY e.start\_time;

Регистрация участника на мероприятие:

INSERT INTO registration (participant\_id, event\_id)

VALUES (%s, %s);

Проверка заполненности перед регистрацией:

SELECT COUNT(\*) FROM registration WHERE event\_id = %s;

SELECT capacity FROM venue WHERE id = (

SELECT venue\_id FROM event WHERE id = %s

);

Отмена участия:

DELETE FROM registration

WHERE participant\_id = %s AND event\_id = %s;

**3. Заключение**

Разработанная система полностью отражает требования задачи и реализует все основные функции: управление мероприятиями и площадками, регистрация участников, контроль за заполненностью.