

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ	5
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	6
РЕФЕРАТ	7
ВВЕДЕНИЕ	8
1 Аналитическая часть	9
1.1 Анализ предметной области	9
1.2 Анализ известных решений	9
1.3 Формализация задачи	11
1.4 Формализация данных	11
1.5 Формализация ролей и типов	13
1.6 Анализ моделей баз данных	14
1.6.1 Дореляционная модель	15
1.6.2 Реляционная модель	16
1.6.3 Постреляционная модель	17
1.7 Вывод	17
2 Конструкторская часть	18
2.1 Диаграмма прецедентов	18
2.2 Описание сущностей базы данных	19
2.3 Ролевая модель	22
2.4 Алгоритм расчета цены посещения	22
2.4.1 Формализация мероприятия	22
2.4.2 Фундаментальная цена	24
2.5 Используемые триггеры	25

2.6	Архитектура приложения	25
2.6.1	Диаграмма потока данных	25
2.6.2	Диаграмма компонентов	25
2.6.3	Диаграмма классов	25
2.7	Вывод	25
3	Технологическая часть	26
3.1	Вывод	26
4	Исследовательская часть	27
	ЗАКЛЮЧЕНИЕ	28
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	29
	ПРИЛОЖЕНИЕ А	30

ОПРЕДЕЛЕНИЯ

В данной расчетно-пояснительной записке применяется следующая терминология:

ER-модель (*от англ. Entity-Relationship model*) – модель данных, позволяющая описывать концептуальные схемы предметной области [5].

Диаграмма прецедентов – диаграмма, описывающая, какой функционал разрабатываемой программной системы доступен каждой группе пользователей [9].

VIP-персона (*от англ. Very important person*) – человек, имеющий персональные привилегии, льготы из-за своего высокого статуса, популярности или капитала [6].

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В данной расчетно-пояснительной записке применяются следующие сокращения и обозначения:

БД – база данных

СУБД – система управления базами данных

РЕФЕРАТ

Расчетно-пояснительная записка 30 с., 4 рис., 15 таблиц, 27 источников, 1 приложение.

ШАХМАТЫ, КУБОК МИРА ПО ШАХМАТАМ, СТАВКИ, БАЗЫ ДАННЫХ, ЛОГИРОВАНИЕ, КЭШИРОВАНИЕ, POSTGRESQL, INFLUXDB, REDIS

Цель работы — разработка базы данных сыгранных на кубке мира шахматных партий.

В рамках курсовой работы была разработана база данных сыгранных на кубке мира шахматных партий и приложение к ней. Был проведен анализ предметной области, связанной с проведением кубка мира по шахматам и ставками на спорт. Были рассмотрены и сравнены существующие решения для хранения шахматных партий. Были сформулированы требования к проектируемым программному обеспечению и базе данных. Были рассмотрены системы управления базами данных на основе формализованной задачи. Были описаны сущности проектируемой базы данных и пользователи разрабатываемого приложения.

Были формализованы бизнес-правила приложения и спроектирована база данных. Были описаны ролевая модель и ограничения базы данных. Были разработаны схемы алгоритмов триггеров, необходимых для корректной работы системы. Была описана структура разрабатываемого приложения.

Были проанализированы и выбраны средства реализации приложения и базы данных. Были описаны триггеры, пользователи и ограничения целостности базы данных. Был разработан графический пользовательский интерфейс приложения.

Было проведено исследование, целью которого являлось определение зависимости среднего времени получения результата запроса на стороне фронта от параметра TTL кэша. По результатам измерений можно сделать вывод, что время получения результата запроса уменьшается по мере увеличения параметра TTL кэша, при этом при TTL большем 1000 мс наблюдается насыщение.

ВВЕДЕНИЕ

Целью данной курсовой работы является разработка приложения для внесения, хранения, изменения и обработки данных о мероприятиях.

Чтобы достичь данной цели, необходимо выполнить следующие задачи:

- проанализировать известные форматы представления информации и определить оптимальный вариант;
- проанализировать известные подходы к хранению информации и системы управления базой данных и выбрать наиболее подходящие;
- спроектировать базу данных, определив основные сущности и связи между ними;
- выбрать наиболее подходящие алгоритмы для достижения поставленной цели;
- спроектировать архитектуру и графический интерфейс приложения;
- выбрать средства реализации приложения и реализовать его;
- провести исследование зависимости цены посещения мероприятия от количества человек на нем и других.

1 Аналитическая часть

В этой части рассматриваются анализ предметной области, известных решений и моделей баз данных, формализации задачи и ролей.

1.1 Анализ предметной области

Организация мероприятий – это многогранный и трудоемкий процесс, который требует внимания к деталям и учета множества факторов. От выбора подходящей локации до составления списка гостей, от планирования бюджета до подготовки меню – каждый этап организации требует тщательной проработки. В ходе подготовки организаторы сталкиваются с рядом типичных вопросов, таких как: «Что необходимо приобрести для мероприятия?», «Какое количество гостей ожидается?» и «Какова стоимость участия?». Эти вопросы, хотя и кажутся простыми, но требуют значительных временных и организационных затрат, особенно если мероприятие масштабное или включает множество участников [1].

Для упрощения этого процесса были разработаны специализированные инструменты – планировщики мероприятий. Эти приложения предназначены для того, чтобы объединить все этапы организации мероприятия в единую систему, сделать процесс планирования более структурированным и прозрачным. Планировщик мероприятий позволяет организаторам:

- 1) систематизировать задачи – разбить процесс организации на этапы и подзадачи;
- 2) координировать участников – вести список гостей, учитывать их предпочтения и информировать о деталях мероприятия;
- 3) управлять бюджетом – учитывать расходы и планировать финансы, чтобы избежать непредвиденных затрат;
- 4) контролировать сроки – устанавливать дедлайны для каждой задачи и отслеживать их выполнение.

1.2 Анализ известных решений

Организация мероприятий всегда была важной и востребованной сферой, потому для упрощения этого процесса были разработаны специализированные

информационные системы, которые автоматизируют множество задач. Наиболее популярными являются:

- 1) Eventbrite – платформа для организации мероприятий, которая позволяет создавать страницы событий, продавать билеты онлайн, собирать данные о посетителях и управлять регистрациями [2];
- 2) Cvent – профессиональная платформа для организации мероприятий, которая предлагает комплексные решения для планирования, управления гостями, бюджетирования и аналитики [3];
- 3) Trello – инструмент для управления проектами и задачами, который можно адаптировать для планирования мероприятий [4].

Критерии сравнения известных решений и результаты их сравнительного анализа представлены в таблицах 1.1 и 1.2 соответственно.

Таблица 1.1 — Критерии сравнения известных решений

Критерий	Описание
Аккаунт	Возможность иметь аккаунт
Бесплатный доступ	Бесплатный доступ ко всем возможностям приложения
Привилегии участников	Возможность выдавать участникам роли с привилегиями
Рейтинг	Формирование оценки мероприятия по оставленным отзывам

Таблица 1.2 — Результаты сравнительного анализа известных решений

Критерий	Eventbrite	Cvent	Trello
Аккаунт	+	+	+
Бесплатный до- ступ	-	-	+
Привилегии участников	-	+	-
Рейтинг	+	+	-

Ни одно из рассматриваемых решений не обеспечивает пользователя всеми необходимыми функциями для организации мероприятий.

1.3 Формализация задачи

Необходимо спроектировать и реализовать базу данных, которая будет хранить данные о пользователях, мероприятиях и отзывах. Также требуется разработать приложение с функционалом для просмотра, добавления, редактирования и удаления информации.

1.4 Формализация данных

Исходя из анализа предметной области, можно выделить следующие ключевые группы данных:

- локация;
- мероприятие;
- день мероприятия;
- участник мероприятия;
- меню мероприятия;
- предмет меню;
- отзыв;
- пользователь.

Группы данных и сведения о них представлены в таблице 1.3

Таблица 1.3 — Группы данных и их сведения

Группа данных	Сведения
Локация	Название, описание, цена аренды, вместимость
Мероприятие	Название, описание, дата, количество участников, количество дней, локация, рейтинг
День мероприятия	Название, порядковый номер, описание, цена посещения, меню
Отзыв	Комментарий, рейтинг, участник мероприятия
Пользователь	Имя, телефон, гендер, роль, права доступа, пароль
Участник мероприятия	Имя, тип, факт оплаты
Меню мероприятия	Название, стоимость
Предмет меню	Название, тип, цена

ER-диаграмма сущностей в нотации Чена представлена на рисунке 1.1.

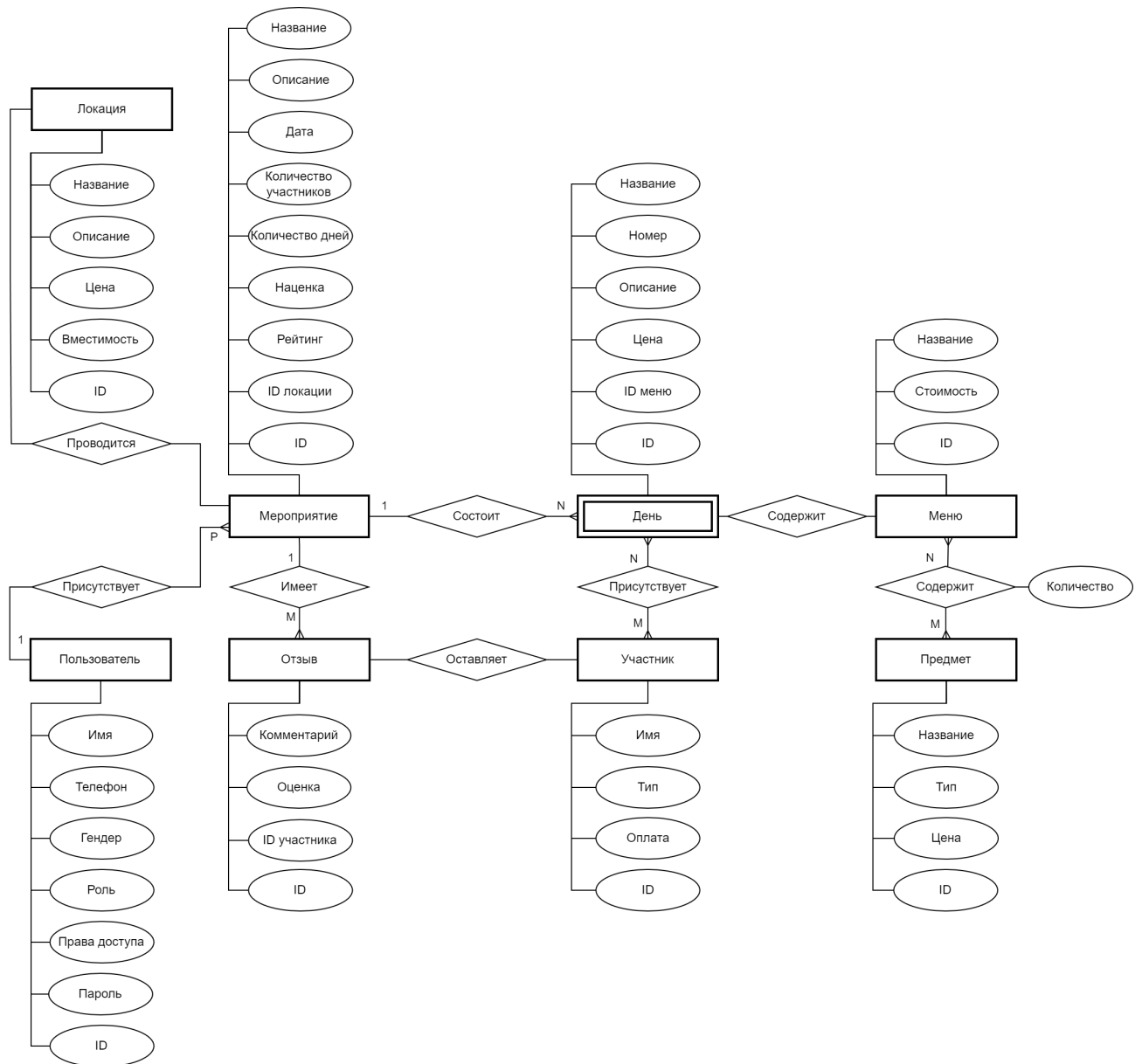


Рисунок 1.1 — ER-диаграмма сущностей в нотации Чена

1.5 Формализация ролей и типов

Множества значений перечисляемых сведений представлены в таблице 1.4.

Таблица 1.4 — Множества значений перечисляемых сведений

Группа данных	Сведение	Множество значений
Пользователь	Роль	Гость, зарегистрированный пользователь, администратор
	Гендер	Мужской, женский
Участник мероприятия	Тип	Простой участник, VIP, организатор
Предмет меню	Тип	Однодневный, многодневный

Роли пользователей, выделяемые в разрабатываемой системе, представлены в таблице 1.5.

Таблица 1.5 — Категории пользователей

Роль	Описание
Гость	Неавторизованный пользователь, который может регистрироваться, входить в систему и просматривать информацию о мероприятиях
Зарегистрированный пользователь	Может просматривать данные о мероприятиях, подавать и удалять заявки на участие, а также оставлять и удалять отзывы
Администратор	Обладает правами на просмотр, добавление и изменение данных о мероприятиях, пользователях и отзывах.

Типы участников, выделяемые в рамках мероприятия, представлены в таблице 1.6.

Таблица 1.6 — Типы участников

Тип	Описание
Простой участник	Не имеет специальных прав или привилегий
VIP-персона	Присутствует на мероприятии бесплатно
Организатор	Имеет доступ ко всей информации о мероприятии и присутствует на нём бесплатно

Типы предметов, выделяемые в рамках меню, представлены в таблице 1.7.

Таблица 1.7 — Типы предметов

Тип	Описание
Однодневный	Требуется только для одного дня мероприятия
Многодневный	Необходим на протяжении всего мероприятия

1.6 Анализ моделей баз данных

База данных – самодокументированное собрание интегрированных записей. Она служит основой для хранения, обработки и анализа информации, необходимой для функционирования бизнеса, научных исследований или других сфер деятельности. В зависимости от типа задач, базы данных могут быть классифицированы на два основных типа [7, 8]:

- 1) OLAP (*англ. Online Analytical Processing*) – технология для многомерного анализа больших объемов данных, используемая для быстрого выполнения сложных запросов;
- 2) OLTP (*англ. Online Transaction Processing*) – система для обработки транзакций в реальном времени, ориентированная на быструю запись и чтение небольших объемов данных.

Система управления базами данных – приложение, обеспечивающее создание, хранение, обновление и поиск информации в базах данных [8].

Основные функции СУБД:

- управление данными во внешней памяти;
- управление буферами оперативной памяти;
- управление транзакциями;
- журнализация;
- поддержка языка или языковых пакетов;
- обеспечение целостности и безопасности базы данных.

Модель данных – это структурное представление элементов данных, их отношений и ограничений в системе управления базами данных. Системы управления базами данных классифицируются по модели данных, которая определяет их архитектуру, структуры данных и методы обработки.

Выделяются три категории моделей [7, 8]:

- дореляционные;
- реляционные;
- постреляционные.

1.6.1 Дореляционная модель

Дореляционные базы данных представляют собой ранние системы управления данными, которые широко использовались до появления реляционной модели. Эти системы, такие как иерархические и сетевые СУБД, были разработаны для организации и хранения данных в сложных структурах, которые отражали специфические взаимосвязи между объектами. Однако, несмотря на свою функциональность, они обладают рядом ограничений, которые делают их менее гибкими и удобными по сравнению с современными реляционными базами данных [7].

Иерархические базы данных используют дерево, где каждый элемент данных (узел) имеет строгую связь «родитель-потомок». Такая организация эффективна для данных, которые естественным образом поддаются иерархическому представлению, например, файловые системы или организационные структуры компаний. Однако главным недостатком иерархических моделей является их неэффективность: доступ к данным часто требует обхода всей структуры, что усложняет выполнение запросов.

Сетевые базы данных являются развитием иерархических моделей, устраняя некоторые их ограничения. В сетевых моделях данные организовываются в виде графа, где узлы могут иметь множественные связи друг с другом. Это позволяет более гибко представлять сложные взаимосвязи между объектами, что делает их подходящими для задач, где данные имеют множество пересекающихся связей. Однако, несмотря на большую универсальность, сетевые модели остаются сложными в проектировании и использовании.

Хотя дореляционные модели хорошо справляются с управлением памятью, их применение осложняется зависимостью от физической структуры данных, что делает процесс внесения изменений в базу данных более трудоёмким.

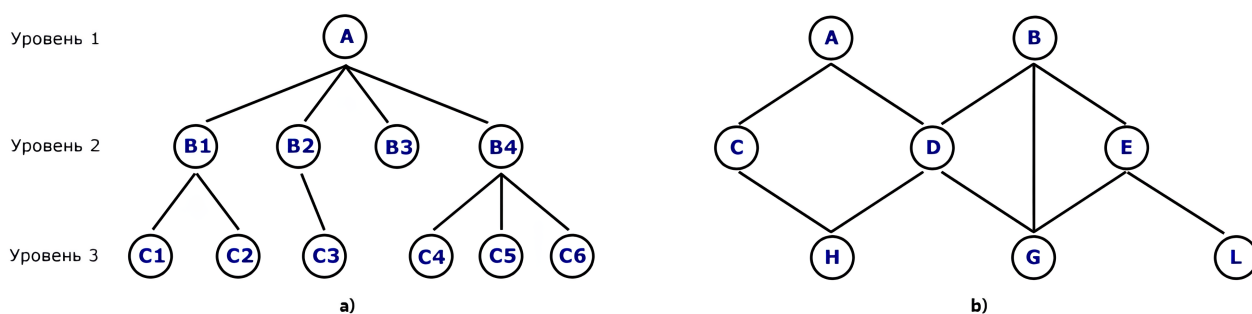


Рисунок 1.2 — Структуры дореляционных моделей: иерархическая (a), сетевая (b)

1.6.2 Реляционная модель

Реляционная модель данных включает в себя три части [8]:

- 1) структурная часть – данные представляются в виде таблиц, называемых отношениями. Каждая таблица состоит из строк (кортежей) и столбцов (атрибутов). Атрибуты имеют определённые домены – допустимые значения, которые они могут принимать. Ключи (первичные и внешние) обеспечивают уникальность записей и связь между таблицами. Схемы определяют структуру базы данных, включая таблицы, атрибуты и связи между ними;
- 2) целостная часть – определяет правила, которые гарантируют корректность и согласованность данных:
 - целостность сущностей: каждый кортеж в таблице должен иметь уникальный идентификатор (первичный ключ), что исключает дублирование записей;
 - целостность ссылок: внешние ключи в одной таблице должны соответствовать первичным ключам в другой, что обеспечивает согласованность данных между таблицами. Это требование поддерживается нормализацией отношений, которая устраняет избыточность данных.
- 3) манипуляционная часть – включает инструменты для работы с данными, такие как реляционная алгебра и реляционное исчисление. Реляционная алгебра основана на теории множеств и предоставляет набор операций (например, выборка, проекция, объединение, разность) для манипуля-

ции данными. Реляционное исчисление, основанное на логике предикатов, позволяет формулировать запросы на декларативном уровне.

Модель отличается простотой, независимостью данных и удобством реализации, но имеет недостатки, такие как сложность описания иерархических связей.

1.6.3 Постреляционная модель

Классическая реляционная модель предполагает, что данные в полях таблиц должны быть неделимыми, что может ограничивать эффективность некоторых приложений. Постреляционная модель устраняет это ограничение, допуская многозначные поля и вложенные таблицы, что упрощает описание сложных структур. Основное преимущество постреляционной модели – возможность объединять несколько связанных таблиц в одну, что повышает наглядность и ускоряет обработку данных.

1.7 Вывод

В аналитической части работы были проведены анализы предметной области, известных решений и моделей баз данных, проведены формализации задачи, данных и ролей. Был сделан выбор в пользу реляционной модели базы данных.

2 Конструкторская часть

В этой части представляются требования к программе,

2.1 Диаграмма прецедентов

Диаграмма прецедентов представлена на рисунке 2.1.

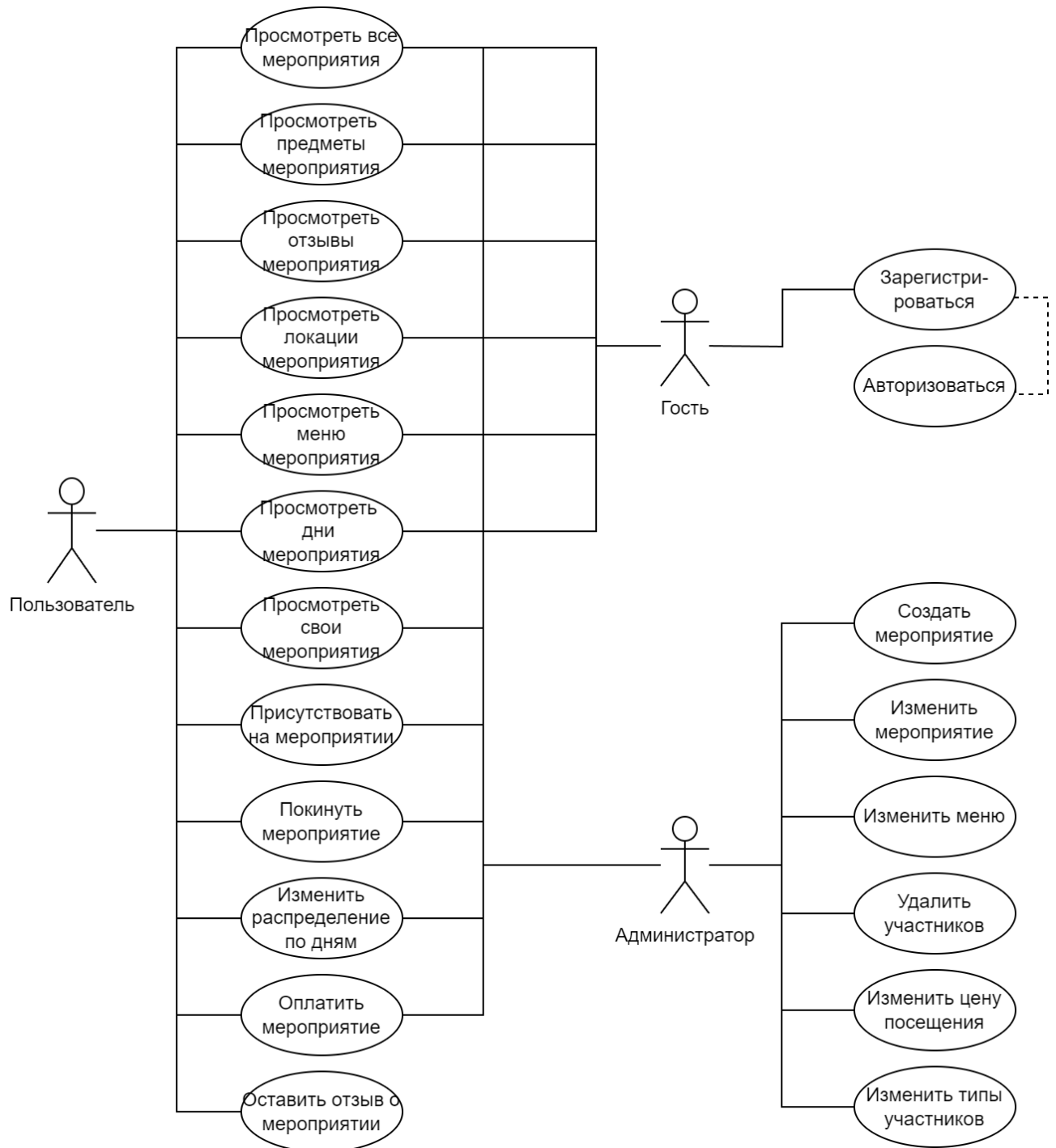


Рисунок 2.1 — Диаграмма прецедентов

2.2 Описание сущностей базы данных

На основе данных, представленных в таблице 1.3, можно определить таблицы, которые должны быть включены в базу данных:

- 1) locations – таблица локаций;
- 2) events – таблица мероприятий;
- 3) persons – таблица участников мероприятий;
- 4) days – таблица дней мероприятий;
- 5) menu – таблица меню дней мероприятий;
- 6) items – таблица предметов меню;
- 7) feedbacks – таблица отзывов участников;
- 8) users – таблица пользователей.

На основе информации о выбранной СУБД и ER-диаграммы на рисунке 1.1 можно определить структуры столбцов, их типы и ограничения для каждой таблицы, которые представлены в таблицах 2.1-2.8.

Таблица 2.1 — Информация о таблице локаций

Атрибут	Тип данных	Ограничения	Сведение
location_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор локации
name	VARCHAR(255)	NOT NULL	Название
description	TEXT	NOT NULL	Описание
price	NUMERIC	NOT NULL, CHECK (price >= 0)	Цена аренды на 1 день
capacity	INT	NOT NULL, CHECK (capacity >= 0)	Вместимость

Таблица 2.2 — Информация о таблице мероприятий

Атрибут	Тип данных	Ограничения	Сведение
event_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор мероприятия
name	VARCHAR(255)	NOT NULL	Название
description	TEXT	NOT NULL	Описание
date	DATE	NOT NULL	Дата
person_count	INT	NOT NULL, CHECK (person_count >= 0)	Количество участников
days_count	INT	NOT NULL, CHECK (days_count > 0)	Количество дней
percent	NUMERIC	NOT NULL, CHECK (percent >= 0)	Наценка на посещение в процентах
rating	NUMERIC	NOT NULL, CHECK (rating BETWEEN 0 AND 10)	Рейтинг

Таблица 2.3 — Информация о таблице дней мероприятий

Атрибут	Тип данных	Ограничения	Сведение
day_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор дня мероприятия
name	VARCHAR(255)	NOT NULL	Название
sequence_number	INT	NOT NULL, CHECK (sequence_number > 0)	Порядковый номер
description	TEXT	NOT NULL	Описание
price	NUMERIC	NOT NULL, CHECK (price >= 0)	Цена посещения

Таблица 2.4 — Информация о таблице участников мероприятий

Атрибут	Тип данных	Ограничения	Сведение
person_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор участника
name	VARCHAR(255)	NOT NULL	Имя
type	ENUM	NOT NULL	Тип
paid	BOOL	NOT NULL	Факт оплаты

Таблица 2.5 — Информация о таблице меню дней мероприятий

Атрибут	Тип данных	Ограничения	Сведение
menu_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор меню
name	VARCHAR(255)	NOT NULL	Название
cost	NUMERIC	NOT NULL, CHECK (cost >= 0)	Стоимость

Таблица 2.6 — Информация о таблице предметов меню

Атрибут	Тип данных	Ограничения	Сведение
item_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор предмета
name	VARCHAR(255)	NOT NULL	Название
type	ENUM	NOT NULL	Тип
price	NUMERIC	NOT NULL, CHECK (price >= 0)	Цена

Таблица 2.7 — Информация о таблице отзывов

Атрибут	Тип данных	Ограничения	Сведение
feedback_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор отзыва
event_id	UUID	NOT NULL, FOREIGN KEY	Идентификатор мероприятия
person_id	UUID	NOT NULL, FOREIGN KEY	Идентификатор участника
comment	TEXT	NOT NULL	Комментарий
rating	NUMERIC	NOT NULL, CHECK (rating BETWEEN 0 AND 10)	Рейтинг

Таблица 2.8 — Информация о таблице пользователей

Атрибут	Тип данных	Ограничения	Сведение
user_id	UUID	NOT NULL, PRIMARY KEY	Идентификатор пользователя
phone	VARCHAR(255)	NOT NULL	Телефон
gender	ENUM	NOT NULL	Гендер
password	VARCHAR(255)	NOT NULL	Пароль
role	ENUM	NOT NULL	Роль

Диаграмма базы данных представлена на рисунке 2.2.

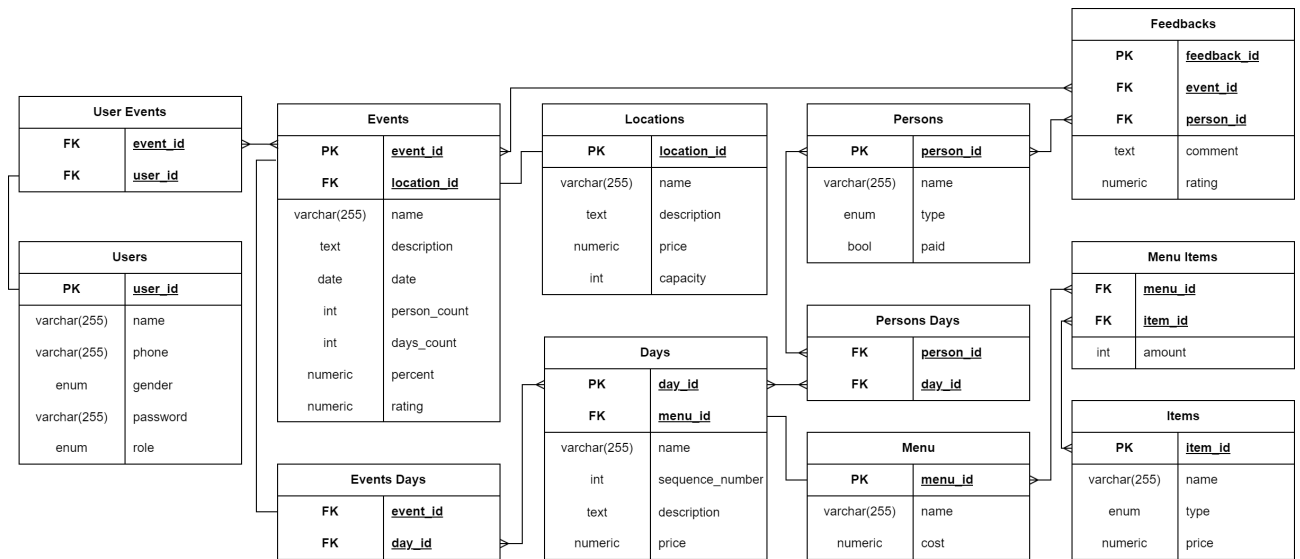


Рисунок 2.2 — Диаграмма базы данных

2.3 Ролевая модель

Ролевая модель в СУБД распределяет права доступа между ролями, ограничивая их действия для обеспечения безопасности:

- гость может выполнять SELECT для всех таблиц за исключением таблицы пользователей, INSERT для таблицы пользователей;
- авторизованный пользователь может выполнять SELECT для всех таблиц, INSERT и DELETE для таблицы мероприятий пользователей, UPDATE для таблицы пользователей;
- администратор имеет все права доступа.

2.4 Алгоритм расчета цены посещения

Мероприятие рассматривается как система, состоящая из участников, дней, предметов (например, еды), меню и других элементов.

2.4.1 Формализация мероприятия

Для формализации мероприятия воспользуемся теорией множеств:

- обозначим через $P = \{p_1, p_2, \dots, p_m\}$ множество всех участников мероприятия, где m – общее количество участников. Каждый элемент p_i представляет собой отдельного участника;

- обозначим через $D = \{d_1, d_2, \dots, d_n\}$ множество всех дней мероприятия, где n – общее количество дней. Каждый элемент d_i соответствует конкретному дню;
- обозначим через $O = \{o_1, o_2, \dots, o_k\}$ множество всех доступных предметов (например, блюд в меню или материалов для проведения активностей), где k – общее количество предметов;
- обозначим через $M = \{m_1, m_2, \dots, m_n\}$ множество меню, где каждое меню m_i является подмножеством множества предметов O , то есть $m_i \subseteq O$. Каждое меню соответствует конкретному дню мероприятия;
- обозначим через 2^D булеан множества D , то есть множество всех возможных подмножеств дней мероприятия;
- определим множество PD как множество упорядоченных пар (p_i, c_j) , где $p_i \in P$ – участник, а $c_j \in 2^D$ – подмножество дней, которые участник p_i планирует посетить. Таким образом, $PD = \{(p_i, c_j) \mid p_i \in P, c_j \in 2^D\}$;
- определим множество DM как множество упорядоченных пар (d_i, m_i) , где $d_i \in D$ – день мероприятия, а $m_i \in M$ – меню, соответствующее этому дню. Таким образом, $DM = \{(d_i, m_i) \mid d_i \in D, m_i \in M\}$;
- теперь можно определить мероприятие E как кортеж, состоящий из всех введенных множеств:

$$E = (P, D, O, M, PD, DM) \quad (2.1)$$

Для расчета стоимости мероприятия и его компонентов введем следующие функции:

- обозначим через $C_O : O \rightarrow \mathbb{R}$ функцию, которая каждому предмету $o_i \in O$ ставит в соответствие его стоимость $C_O(o_i)$;
- определим функцию стоимости меню $C_M : M \rightarrow \mathbb{R}$ как сумму стоимостей всех предметов, входящих в меню:

$$C_M(m) = \sum_{o_i \in m} C_O(o_i) \quad (2.2)$$

- определим функцию стоимости дня $C_D : D \rightarrow \mathbb{R}$ как стоимость меню,

соответствующего этому дню:

$$C_D(d) = C_M(m), \text{ где } (d, m) \in DM \quad (2.3)$$

– определим функцию стоимости мероприятия $C_E : E \rightarrow \mathbb{R}$ как сумму стоимостей всех дней мероприятия:

$$C_E(E) = \sum_{d_i \in D} C_D(d_i) \quad (2.4)$$

Для дальнейших расчетов введем дополнительные множества, связанные со стоимостями:

- обозначим через $C = \{c_1, c_2, \dots, c_n\}$ множество стоимостей всех дней, где $c_i = C_D(d_i)$;
- обозначим через $A = \{a_1, a_2, \dots, a_n\}$ множество отношений стоимостей дней к минимальной стоимости. Каждый элемент a_i вычисляется по формуле:

$$a_i = \frac{c_i}{\min(C)}, \text{ где } c_i \in C. \quad (2.5)$$

Цена посещения дня мероприятия зависит от двух факторов: стоимости дня и количества участников, которые его посещают. Введем функцию цены посещения дня $P_D : D \rightarrow \mathbb{R}$.

Стоимость дня $C_D(d)$ может быть выражена через цену посещения $P_D(d)$ и количество участников, которые планируют посетить этот день:

$$C_D(d) = P_D(d) \cdot |\{p \mid (p, d) \in PD\}| \quad (2.6)$$

где $|\{p \mid (p, d) \in PD\}|$ – количество участников, посещающих день d .

2.4.2 Фундаментальная цена

Введем понятие фундаментальной цены P_0 , которая является константой. Фундаментальная цена – это базовая цена, которая используется для расчета цены посещения каждого дня мероприятия. Она определяется таким образом, чтобы общая выручка от участников покрывала все затраты на проведение мероприятия. Цена посещения дня d_i выражается через фундаментальную цену и

отношение стоимости a_i :

$$P_D(d_i) = a_i \cdot P_0, \text{ где } a_i \in A. \quad (2.7)$$

Если участник планирует посетить несколько дней $\{d_i, d_j, \dots\}$, то общая цена посещения вычисляется как сумма цен каждого дня:

$$P_D(\{d_i, d_j, \dots\}) = \sum_{d \in \{d_i, d_j, \dots\}} P_D(d) = (a_i + a_j + \dots) \cdot P_0 \quad (2.8)$$

Подставив выражение для $P_D(d_i)$ в уравнение стоимости дня, получим:

$$C_D(d) = a_i \cdot P_0 \cdot |\{p \mid (p, d) \in PD\}| \quad (2.9)$$

2.5 Используемые триггеры

2.6 Архитектура приложения

2.6.1 Диаграмма потока данных

2.6.2 Диаграмма компонентов

2.6.3 Диаграмма классов

2.7 Вывод

В конструкторской части работы были представлены требования к программе,

3 Технологическая часть

В данной части представляются выбор средств реализации и исходный код программы, описываются организация классов в программе и её интерфейс.

3.1 Вывод

В данной части были представлены выбор средств реализации и исходный код программы, описаны организация классов в программе и её интерфейс.

4 Исследовательская часть

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Румянцев Д., Франкель Н. Event-маркетинг. Все об организации и продвижении событий [Текст] / Румянцев Д., Франкель Н. — 1-е изд. — Санкт-Петербург: СПб, 2019 — 320 с.
2. Eventbrite [Электронный ресурс]. — URL: <https://www.eventbrite.com/> (дата обращения 15.03.2025)
3. Cvent [Электронный ресурс]. — URL: <https://www.cvent.com/> (дата обращения 15.03.2025).
4. Trello [Электронный ресурс]. — URL: <https://trello.com> (дата обращения 15.03.2025).
5. Peter Pin-Shan Chen. The Entity-Relationship Model — Toward a Unified View of Data / Peter Pin-Shan Chen [Электронный ресурс] // Association for Computing Machinery : [сайт]. — URL: http://portal.acm.org/ft_gateway.cfm?id=320440&type=pdf&coll=GUIDE&d1=GUIDE&CFID=72010946&CFTOKEN=54542835 (дата обращения: 17.03.2025).
6. Мюллер В. К., Англо-русский словарь. 53000 слов. [Текст] / В. К. Мюллер — 1-е изд. — Москва: Рус. яз., 1981 — 887 с.
7. Аврунев О. Е., Стасышин В. М. Модели баз данных: учебное пособие [Текст] / Аврунев О. Е., Стасышин В. М. — 1-е изд. — Новосибирск: НГТУ, 2018 — 123 с.
8. Дейт К. Дж., Введение в системы баз данных [Текст] / Дейт К. Дж. — 8-е изд. — Москва: Издательский дом «Вильямс», 2005 — 1328 с.
9. Мартин Фаулер Основы UML [Текст] / Мартин Фаулер — 3-е изд. — Санкт-Петербург: Символ-Плюс, 2004 — 192 с.

ПРИЛОЖЕНИЕ А