

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерного проектирования
Кафедра проектирования информационно-компьютерных систем
Дисциплина «Структуры и базы данных»

«К ЗАЩИТЕ ДОПУСТИТЬ»

Руководитель курсового проекта
Магистр технических наук,
ассистент

_____. В.Ф.Алексеев
____.____.2022

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему:

**«БАЗА ДАННЫХ ДЛЯ ПОДДЕРЖКИ РАБОТЫ САЙТА ПО ПРОДАЖЕ
БИЛЕТОВ НА КУЛЬТУРНЫЕ МЕРОПРИЯТИЯ»**

БГУИР КП 1-39 03 02 058 ПЗ

Выполнил студент группы 013802
Швец Олег Александрович

(подпись студента)

Курсовой проект представлен на
проверку _____.2022

(подпись студента)

Минск 2022

РЕФЕРАТ

БГУИР КП 1-39 03 02 058 ПЗ

Швец, О.А. База данных для поддержки работы сайта по продаже билетов на культурные мероприятия: пояснительная записка к курсовому проекту / О.А. Швец – Минск: БГУИР, 2022. – 28 с.

Пояснительная записка 28 с., 10 рис., 10 источников, 4 приложения, 2 графических материала.

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ЕЁ ФОРМАЛИЗАЦИИ ДЛЯ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ, ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ ДЛЯ ОСНОВНОГО ВИДА ДЕЯТЕЛЬНОСТИ РАССМАТРИВАЕМОЙ ОБЛАСТИ, ПРИМЕНЕНИЕ РАЗРАБОТАННОЙ БАЗЫ ДАННЫХ

Цель проектирования: разработка WEB-приложения, проектирование эффективной и безопасной базы данных для поддержания работы сайта по продаже билетов на культурные мероприятия.

Методология проведения работы: В процессе решения поставленных задач использованы принципы системного подхода, теория проектирования базы данных, методы проектирования WEB -приложения.

Результаты работы: изучены способы хранения информации в базе данных, проведен анализ реляционной модели данных, разработано WEB-приложение для обеспечения работы сайта.

Область применения результатов: разработанная база данных и WEB-приложение в будущем, может использоваться для создания полноценного приложения для обеспечения комфортного использования пользователем сайта по продаже билетов.

ПЕРЕЧЕНЬ УСЛОВНЫХ ОБОЗНАЧЕНИЙ, СИМВОЛОВ И ТЕРМИНОВ

БГУИР	– Белорусский Государственный Университет Информатики и Радиоэлектроники;
БД(база данных)	– представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины;
КП	– курсовой проект;
ПЗ	– пояснительная Записка;
ПИКС	– кафедра проектирования информационно-компьютерных систем;
СУБД (система управления базами данных)	– совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных;
ФИО	– Фамилия, Имя, Отчество;
<i>ID(Identifier)</i>	– <u>UNIX</u> -утилита, выводящая информацию об указанном пользователе <i>USERNAME</i> или текущем пользователе, который запустил данную команду и не указал явного имя пользователя.
Веб-сервер	– сервер, принимающий <i>HTTP</i> -запросы от клиентов, обычно веб-браузеров, и выдающий им <i>HTTP</i> -ответы, вместе с <i>HTML</i> -страницей, изображением, файлом, медиа-поток или другими данными
Компетентность	– наличие знаний, опыта и навыков, нужных для эффективной деятельности в заданной предметной области
Нормальная форма	– свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных;
Среда выполнения	– вычислительное окружение, необходимое для выполнения компьютерной программы и доступное во время выполнения компьютерной программы;
<i>API (application programming interface)</i>	– описание способов (набор классов, процедур, функций, структур или констант), которыми одна компьютерная программа может взаимодействовать с другой программой;
<i>HTML (HyperText Markup Language)</i>	– стандартизированный язык разметки документов во Всемирной паутине;

<i>HTTP (HyperText Transfer Protocol)</i>	– протокол прикладного уровня передачи данных изначально – в виде гипертекстовых документов в формате <i>html</i> , в настоящий момент используется для передачи произвольных данных;
<i>IDE (Integrated development environment)</i>	– комплекс программных средств, используемый программистами для разработки программного обеспечения;
<i>IDEF</i>	– методология функционального моделирования (англ. <i>function modeling</i>) и графическая нотация, предназначенная для формализации и описания бизнес-процессов;
<i>SQL (structured query language)</i>	– язык структурированных запросов, декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных;
<i>Sybase ASE (Adaptive Server Enterprise)</i>	– реляционная система управления базами данных компании <i>SAP</i> , одна из СУБД, использующая в качестве основного процедурного <i>SQL</i> -расширения язык <i>Transact-SQL</i> ;
<i>URL (Uniform Resource Locator)</i>	– система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса;
<i>Usability</i>	– способность продукта быть понимаемым, изучаемым, используемым и привлекательным для пользователя в заданных условиях

СОДЕРЖАНИЕ

Введение	6
1 Анализ предметной области и ее формализация для проектирования базы данных	7
1.1 Описание предметной области	7
1.2 Анализ информационных потребностей пользователей и предварительное описание запросов	8
1.3 Определение требований и ограничений к базе данных с точки зрения предметной области	8
1.4 Постановка решаемой задачи	9
2 Проектирование базы данных для основного вида деятельности Рассматриваемой предметной области	10
2.1 Разработка инфологической модели предметной области базы данных ..	10
2.2 Выбор и обоснование используемых типов данных и ограничений (доменов)	13
2.3 Проектирование запросов к базе данных	14
2.4 Програмная реализация и документирование базы данных	16
3 Применение разработанной базы данных	17
3.1 Руководство пользователя	17
3.2 Руководство авторизованного пользователя	19
3.3 Реализация клиентских запросов	19
3.4 Обоснование и реализация механизма обеспечения безопасности и сохранности данных	20
Заключение	21
Список использованных источников	22
Приложение А (Антиплагиат)	23
Приложение Б (обязательное) Скрипт генерации БД	24
Приложение В (обязательное) Листинг программного кода	27
Приложение Г (обязательное) Ведомость курсового проекта	28

ВВЕДЕНИЕ

Культурные мероприятия являются неотъемлемой составляющей человеческой культуры уже много тысяч лет: начиная со сценок в пещерах и античным театром до сегодняшних дней – кинопоказы, музыкальные концерты и многое другое. С появлением интернета стало намного проще планировать свое время и покупать билеты на культурные мероприятия. Людям становится все более непривычно покупать билеты в кино, например, в кассах кинотеатра. Действительно, зачем тратить свое время на дорогу до кассы, если вся информация есть на смартфоне, ПК, ноутбуке.

Спрос на цифровые платформы растет с каждым годом, ведь в Республике Беларусь появляется все больше и больше культурных мероприятий. Как, например, Минский полумарафон, запись на который была реализована через интернет.

Все больше растет спрос на автоматизацию продажи билетов на культурные мероприятия. Также возникает необходимость хранения большого количества персональных данных пользователей, относящейся к организации культурных мероприятий и их последующему анализу.

Большую роль играет создание дружелюбного и удобного пользовательского интерфейса, ведь если человек, решивший приобрести билет на мероприятие не сможет разобраться в том, как это сделать, то прибыль не получит ни организатор, ни сервис по продаже билетов.

В настоящее время базы данных занимают лидирующее положение в области информационных технологий, они становятся неотъемлемой частью в любой сфере деятельности, а особенно в обработке большого объема информации.

Объектом исследования данной курсовой работы является обобщенная структура культурного мероприятия.

Цель исследования – изучение предметной области и разработка модели базы данных для обеспечения целостного функционирования системы сайта по продаже билетов на культурные мероприятия.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ЕЕ ФОРМАЛИЗАЦИЯ ДЛЯ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ

Для анализа предметной области необходимо разобраться, что является предметной областью. И на основе полученных данных начинать проектирование.

Предметная область – это часть реального мира, данные о котором необходимо отразить в базе данных. Исходя, из определения анализом предметной области состоит из ее подробного описания, выделения полезной информации и последующей формализации в данные [1].

1.1 Описание предметной области

В настоящее время наблюдается тенденция роста информатизации и компьютеризации общества, причем эта тенденция отмечается повсеместно – во всех странах мира, в разных областях профессиональной деятельности, в быту.

Сайт по продаже билетов – сервис, предоставляющий населению услуги развлекательного характера. Так как данная предметная область представляет собой сферу услуг, то основной её задачей является обеспечение максимальной скорости передачи информации клиенту. Можно предположить следующие задачи для разрабатываемого программного обеспечения:

- поиск нужного мероприятия;
- свободный доступ к информации;
- удобный и понятный для пользователя интерфейс приложения.

На основе детального изучения и анализа предметной области в дальнейшем будет спроектирована структура базы данных для сайта по продаже билетов на культурные мероприятия. Предметная область, содержит полную информацию о стоимости билетов на различные мероприятия, информацию о дате и времени проведения, информацию о всех мероприятиях, проходящих в определенный день, информацию о наличии свободных мест и т.д.

Предметная область содержит информацию о стоимости билетов на различные мероприятия, информацию о дате и времени проведения мероприятия, информацию о типах мест и т.д.

Пользователь, который хочет посетить мероприятие имеет две возможности: поиск по заданной дате и поиск конкретного мероприятия, например, фильма и сможет выбрать удобную для него дату. После этого пользователь выбирает удобное для него место в зале и приобретает билет.

1.2 Анализ информационных потребностей пользователей и предварительное описание запросов

В данной работе представлен план создания автоматизации системы покупки билетов на культурные мероприятия.

Так как мероприятия могут добавляться, а уже прошедшие удаляться, то должна быть создана возможность редактирования новых полей записи.

Для базы данных сайта по продаже билетов на культурные мероприятия выделены две группы пользователей: зарегистрированный пользователь с личным кабинетом и обычный пользователь без аккаунта. С учетом этого можно выделить ограничения для незарегистрированного пользователя:

- пользователь может просматривать информацию о мероприятиях, но не может приобретать билет;
- такой пользователь не может оставлять отзывы на мероприятия.
- Для удобного взаимодействия с базой данных будут описаны следующие запросы:
- зарегистрированный пользователь может купить билет, оставив свои данные;
- такой пользователь может просматривать все свои купленные билеты;
- зарегистрированный пользователь может оформить возврат билета;

С учетом вышеприведённых свойств формируются права у различных пользователей системы.

1.3 Определение требований и ограничений к базе данных с точки зрения предметной области

Исходя из выбранной предметной области были выявлены следующие требования:

- наименование мероприятий не должны повторяться;
- у билета должно быть указано место, дата и время начала мероприятия;
- у билета должна быть указана цена;
- у покупателя должен быть логин и пароль;
- логин и почта клиента должен быть уникальными;
- заказ должен содержать минимум один билет;

Учитывая все эти требования, работоспособность и логичность базы данных будет наилучшей.

1.4 Постановка решаемой задачи

Основной целью курсового проекта является проектирование базы данных для поддержки работы сайта по продаже билетов на культурные мероприятия. База данных должна быть надежной и защищенной, ведь придется хранить личную информацию пользователей.

Вся информация представляется в виде сопряженных между собой таблиц, разбитых на строки и столбцы. Каждая таблица должна иметь первичный ключ – поле, содержимое которого однозначно определяет запись в таблице и отличает ее от других [2]. Связь между двумя таблицами обычно образуется при добавлении в первую таблицу поля, содержащего значение первичного ключа второй таблицы.

В ходе проектирования курсового задания были выявлены следующие цели:

- способность удобной и быстрой покупки и просмотра билетов пользователем;
- удобная авторизация для пользователя;
- возможность удобного и быстрого администрирования базы данных;
- защита данных пользователя от несанкционированного или случайного доступа.

Для решения вышеперечисленных целей были сформулированы следующие задачи:

- разработка удобного и интуитивно понятного интерфейса;
- разработка базы данных, в которой не будет находиться избыточная информация, для её быстрого действия;
- для защиты данных пользователей необходимо определить права доступа в базе данных.

Таким образом будет спроектирована безотказная и удобная БД для обеспечения работы железнодорожного вокзала.

2 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ ДЛЯ ОСНОВНОГО ВИДА ДЕТЕЛЬНОСТИ РАССМАТРИВАЕМОЙ ПРЕДМЕТНОЙ ОБЛАСТИ

Проектирование базы данных – это самая сложная задача, связанная с созданием информационной системы. В результате решения этой задачи должны быть определены содержания базы данных, эффективный для всех ее будущих пользователей способ организации данных и инструментальные средства управления данными.

2.1 Разработка инфологической модели предметной области базы данных

В этом разделе будет способ создания полноценной базы данных, состоящей из 6 таблиц и способа их взаимодействия.

Проектирование базы данных – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы. В результате решения этой задачи должны быть определены содержание базы данных, эффективный для всех ее будущих пользователей способ организации данных и инструментальные средства управления данными [5].

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Храниться данные будут в диаграмме «сущность – связь», которая будет создана с помощью системы управления базами данных.

Связь – это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой или сама с собою [6].

Диаграмма «сущность – связь» (*ER*-диаграмма) позволяет графически представить все элементы информационной модели согласно простым, интуитивно понятным, но строго определенным правилам – нотациям.

В соответствии с индивидуальным заданием были определены требования к составу данных. На основе анализа предметной области выделены следующие сущности:

- сущность «Данные Клиента»;
- сущность «Тип Мероприятия»;
- сущность «Описание»;
- сущность «Мероприятие»;
- сущность «Билеты»;
- сущность «Оставить Отзыв».

ER-диаграмма представления базы данных приведена на рисунке 2.1.

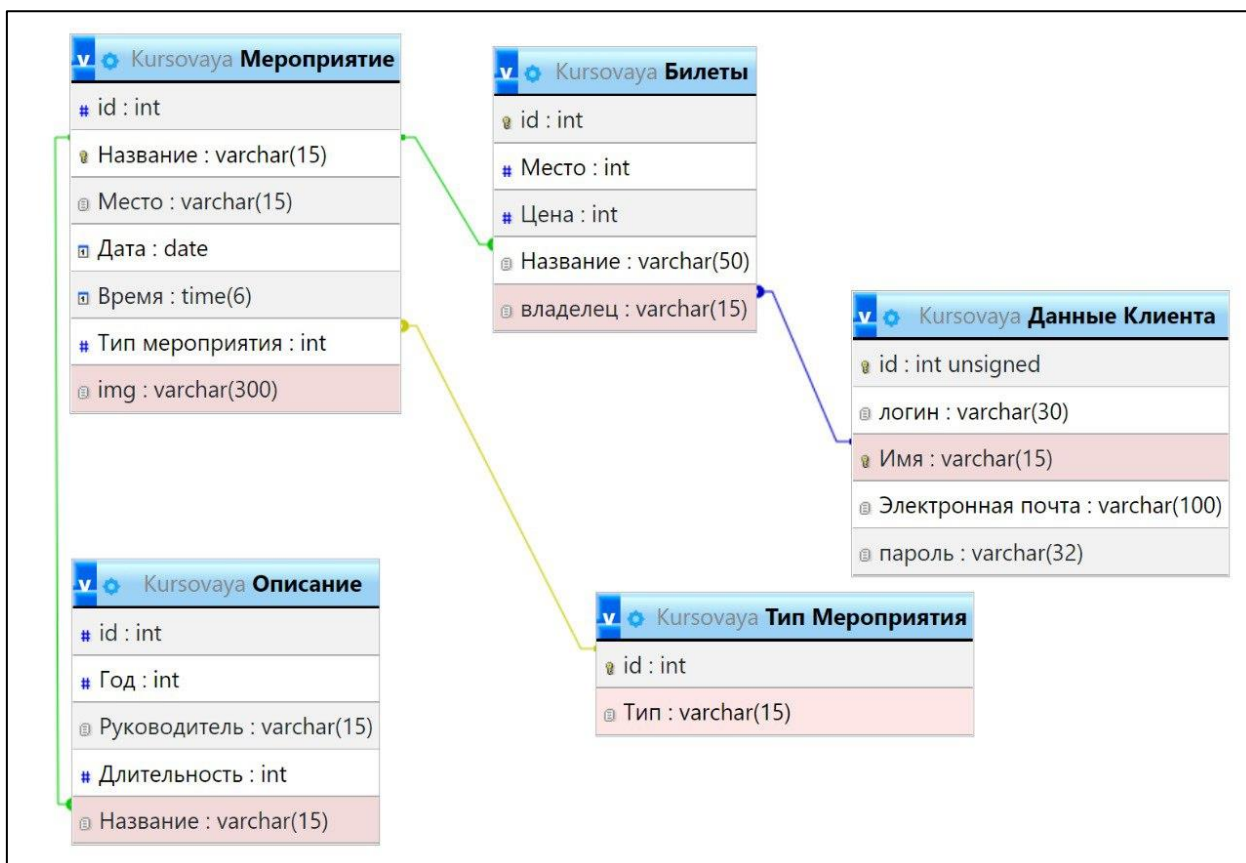


Рисунок 2.1 – ER-диаграмма базы данных

Первичный ключ представляет собой один из примеров уникального индекса и применяется для уникальной идентификации записей таблицы. Никакие две записи таблицы не могут иметь одинаковые значения первичного ключа. Первичный ключ обычно сокращенно обозначают как *РК*. Название первичного ключа должно быть уникальным и не совпадать ни с одним атрибутом других сущностей, а также со структурами запросов и названием привилегий.

В реляционных базах данных практически всегда разные таблицы логически связаны друг с другом. Первичные ключи как раз используются для однозначной организации такой связи.

Внешние ключи используются главным образом для проверки целостности данных, а не для объединения таблиц, как принято считать, однако в нашем случае внешний ключ будет применяться для связи между таблицами [7].

Внешние ключи – это как раз то, что делает реляционные базы реляционными, те связующие цепочки, которые связывают таблицы между собой. Они позволяют вам «разместить» покупателей в одной таблице, заказы – в другой, а товары из этих заказов – в третьей, таким образом, в базе

минимизируется избыточность данных. Чем меньше избыточных данных – тем больше у вас шансов сохранить целостность данных (две или более противоречащие друг другу записи – это всегда плохо). Название внешнего ключа одной сущности может реализации запросов. Если названия сущностей и атрибутов не подпадают под правило и на этапе проектирования запросов, то командная строка *MySQL* не определит ключ, который может совпадать с названием первичного название другой сущности, с которой идет связь, но лучше всего, чтобы атрибута внешнего ключа было уникальным и не совпадало ни с одним атрибутом других сущностей, а также со структурами запросов названием привилегий [8].

Процесс нормализации выполняется путем анализа отношений с учетом их первичного ключа и существующих функциональных зависимостей. Он включает ряд правил, которые используются для проверки отношений, чтобы вся база данных могла быть нормализована до желаемой степени нормализации. Если некоторое требование не удовлетворяется, то должна быть проведена декомпозиция отношения на отношения, каждое из которых удовлетворяет всем требованиям нормализации [3].

При работе с реляционными базами данных обязательным является удовлетворение только требованиям первой нормальной формы (1НФ). Все остальные формы нормализации могут использоваться по желанию проектировщиков. Однако рекомендуется выполнять нормализацию как минимум до 3НФ, чтобы получить хорошее качество отношений (приемлемую избыточность отношений).

Атрибут, значения которого атомарны (неделимы) называется простым атрибутом. Сложный атрибут получается соединением нескольких атомарных атрибутов, которые могут быть определены на одном или разных доменах (его также называют вектор или агрегат данных).

Ненормализованное отношение характеризуется таблицей, содержащей в одном или нескольких столбцах, повторяющиеся группы данных – сложные атрибуты (массив или вектор значений) [4].

Отношение, в котором на пересечении каждой строки и каждого столбца содержится атомарное (или единственное) значение, находится в 1НФ. При этом необходимо, чтобы отношение имело первичный ключ. Для преобразования ненормализованной таблицы в первую нормальную форму, следует найти в исходной таблице и устранить все повторяющиеся группы данных, используя декомпозицию этой таблицы.

Все атрибуты всех сущностей атомарны, у каждой сущности выделены первичные ключи, следовательно, отношения находятся в 1 НФ.

Вторая нормальная форма применяется к отношениям с составными ключами, т.е. к таким отношениям, первичный ключ которых состоит из двух или больше атрибутов. Отношение с первичным ключом на основе единственного атрибута всегда находится в 2НФ.

2.2 Выбор и обоснование используемых типов данных и ограничений (доменов)

В *MySQL* используется множество типов данных, которые можно разделить на шесть групп:

- целые;
- вещественные;
- строковые;
- бинарные;
- даты и времени;
- перечисления и множества.

В проектируемой базе данных были использованы:

- *INT* тип данных для хранения чисел;
- *DECIMAL* число с плавающей точкой, для хранения цены;
- *VARCHAR* текстовое поле переменной длины, для названий;
- *TEXT* представление информации строкового вида;
- *DATETIME* для хранения даты и время, используется для даты заказа.

Есть несколько правил, которые определяют отношения между таблицами:

- отношение 1:1 (один к одному): первичный ключ для одной из таблиц включен в качестве внешнего ключа в другой таблице;
- отношение 1:n (один ко многим): первичный ключ из таблицы 1 добавляется в качестве внешнего ключа в таблицу n;
- отношение n:m (многие ко многим): создается новая таблица (таблица связи), первичный ключ состоит из первичных ключей двух оригинальных таблиц.

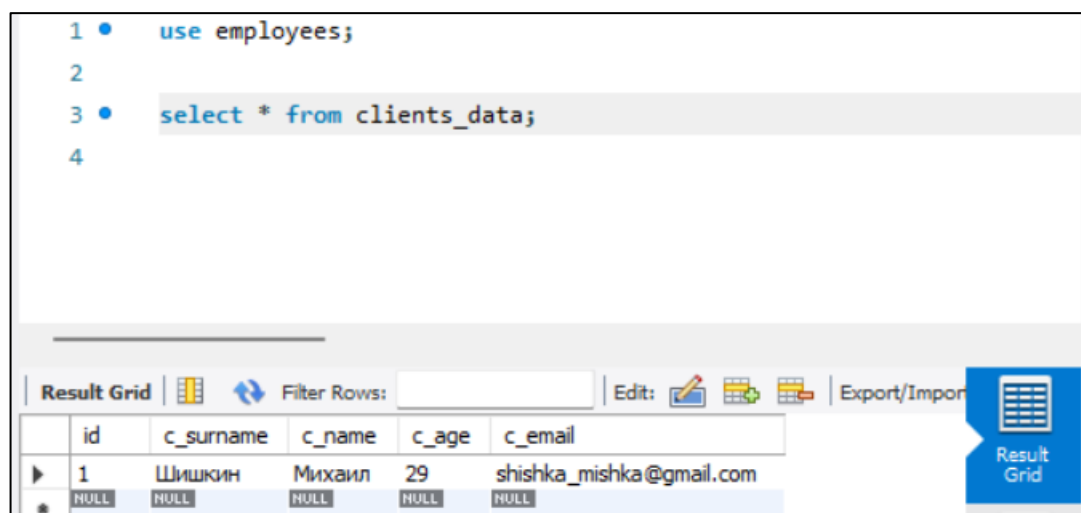
В следующем разделе рассматриваются запросы к базе данных от пользователей.

2.3 Проектирование запросов к базе данных

Запрос – это важнейший инструмент для извлечения информации из одной или нескольких таблиц базы данных. Посредством запроса можно вносить изменения в саму базу данных. Запрос может служить источником данных для форм, отчетов и страниц доступа к данным. Его результатом является новая таблица, которая может быть просмотрена, проанализирована, а затем сохранена или нет.

Запросы позволяют решать многие задачи, не прибегая к программированию. Например, представлять данные в агрегированном виде, производить вычисления над полями базы данных, группировать записи и находить для полей итоговые значения с помощью статических функций.

Оператор *select*, с помощью которой можно вывести все данные, содержащиеся в таблице клиентов. Результат выполнения данного оператора отображен на рисунке 2.2.



The screenshot shows a database query interface. At the top, a SQL query is entered in a text area:

```
1 • use employees;  
2  
3 • select * from clients_data;  
4
```

Below the query area is a toolbar with buttons for 'Result Grid', 'Filter Rows', 'Edit', and 'Export/Import'. Below the toolbar is a table displaying the results of the query. The table has five columns: 'id', 'c_surname', 'c_name', 'c_age', and 'c_email'. The first row shows data for a client with id 1, surname 'Шишкин', name 'Михаил', age 29, and email 'shishka_mishka@gmail.com'. The second row shows a row with all NULL values.

	id	c_surname	c_name	c_age	c_email
▶	1	Шишкин	Михаил	29	shishka_mishka@gmail.com
*	NULL	NULL	NULL	NULL	NULL

Рисунок 2.2 – Результат запроса *SELECT*

Для добавления информации в базу данных используется оператор *insert*. Результат представлен на рисунке 2.3.

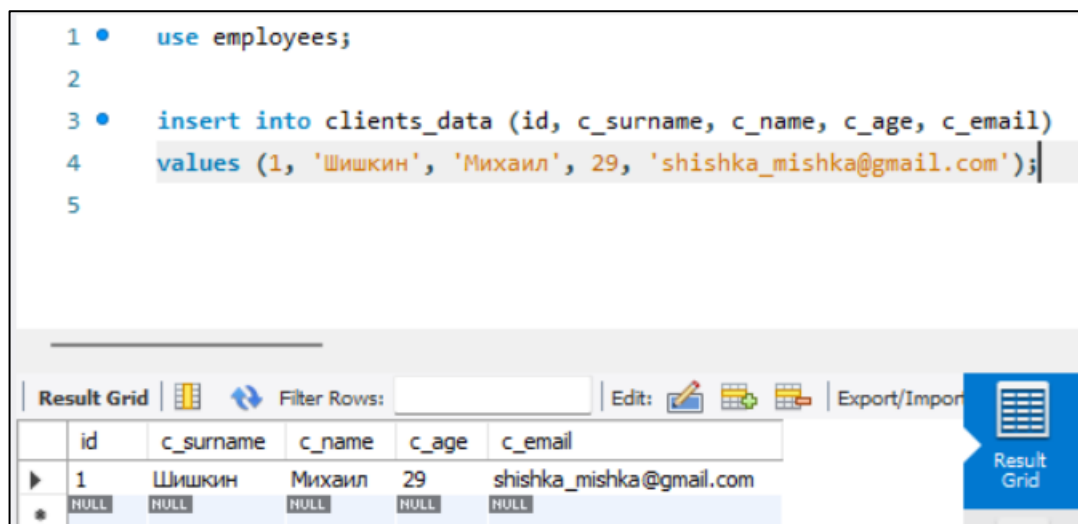


Рисунок 2.3 – Результат запроса *INSERT*

Для изменения информации в базе данных используется оператор *update*. Результат представлен на рисунке 2.4.

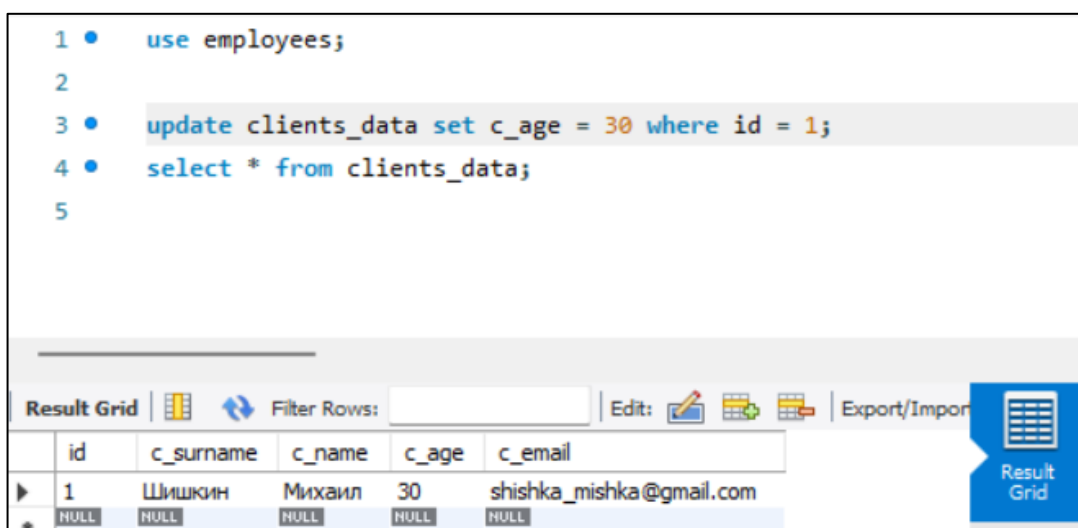


Рисунок 2.4 – Результат запроса *UPDATE*

Для удаления информации в базе данных используется оператор *delete*. Результат представлен на рисунке 2.5.

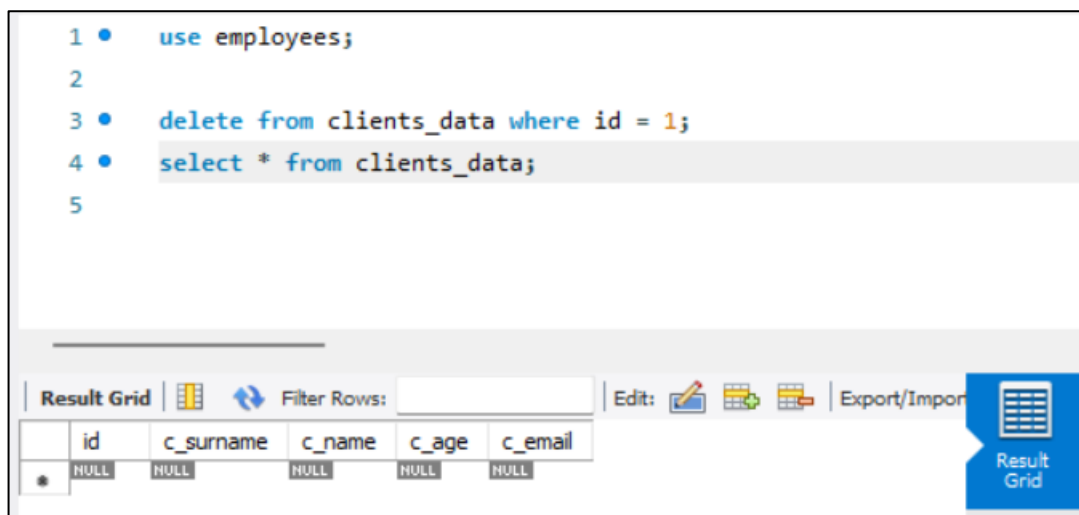


Рисунок 2.5 – Результат запроса *DELETE*

Таким образом, описаны основные запросы необходимые при разработке приложения.

2.4 Программная реализация и документирование базы данных

Будут использованы такие программные средства, как *Open Server Panel* и *Visual Studio Code 2022*.

Visual Studio Code был выбран по нескольким причинам:

- кросс-платформенность;
- поддержка множества языков;
- предоставление документации по различным языкам документации;
- встроенная интеграция Git;
- кастомизация;
- множество полезных функциональных команд
- отладка кода.

База данных и веб-приложение будут реализованы на таких языках, как *HTML5*, *CSS3*, *JavaScript*, *PHP* и *SQL*.

Проектируемая база данных логично задокументирована, так как название таблиц, атрибутов говорят об их назначении. Скрипт генерации базы данных представлен в приложении Б.

3 ПРИМЕНЕНИЕ РАЗРАБОТАННОЙ БАЗЫ ДАННЫХ

3.1 Руководство пользователя

При первом входе на сайт пользователю будет показана главная страница сервиса, с возможностью авторизации. Также будет выведена основная страница, где неавторизованный пользователь сможет ознакомиться с предстоящими мероприятиями (см. рисунок 3.1).

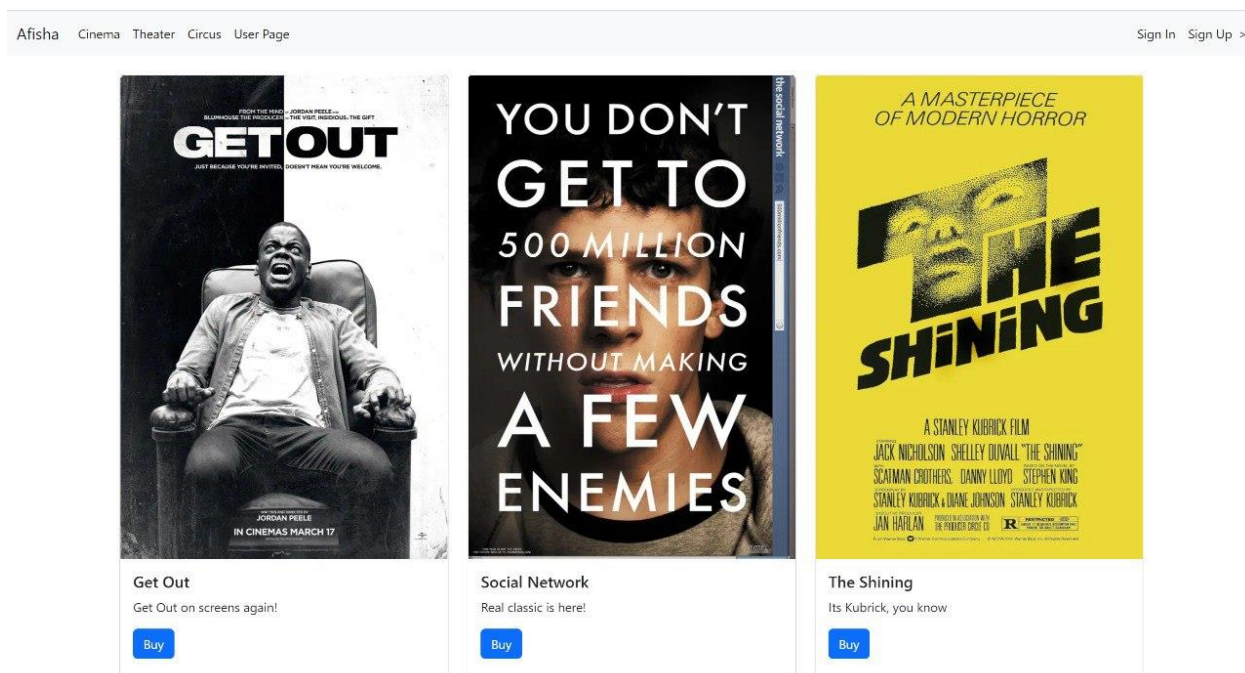


Рисунок 3.1 – Вывод главной страницы сайта

Так как существует разграничение прав пользователей, то и возможности у них различны. Рассмотрим функционал сервиса от имени неавторизованного пользователя. Обычный пользователь имеет возможность просмотреть всю актуальную информацию о предстоящих мероприятиях, а также совершить авторизацию или регистрацию для покупки билета. Окно авторизации показано на рисунке 3.2

Afisha Cinema Theater Circus User Page Sign In Sign Up >

Authorization

Login

Password

Sign In

афиша

Рисунок 3.2 – Окно авторизации

Если пользователь не имеет аккаунта на сервисе, он может зарегистрироваться, рисунок 3.3

Afisha Cinema Theater Circus User Page Sign In Sign Up >

Sign up

Login

☒ Your Name

Your Email

Password

Register

афиша

Рисунок 3.3 – Окно регистрации

Используя данное руководство пользователя, можно перейти к обзору функционала для авторизованного пользователя.


3.2 Руководство авторизованного пользователя

Авторизованный пользователь имеет возможность покупки билетов на разные мероприятия. Кликнув на постер мероприятия пользователь переходит на страницу с мероприятием, где описана вся основная информация: дата, время и т.д. Окно мероприятия показано на рисунке 3.4

The Shining

Horror

A MASTERPIECE
OF MODERN HORROR



A STANLEY KUBRICK FILM

STARRING
JACK NICHOLSON SHELLY DUVALL "THE SHINING"
SCATMAN CROTHERS, DANNY LLOYD, STEPHEN KING
PRODUCED BY
STANLEY KUBRICK & DIANE JOHNSON
DIRECTED BY
JAN HARLAN

PRODUCED IN ASSOCIATION WITH
THE PRODUCERS COMPANY

RESTRICTED
PARENTS STRONGLY CAUTIONED
SOME MATERIAL MAY BE INAPPROPRIATE FOR CHILDREN UNDER 17

Description

This movie is outrageous! You should definitely see it!

Screen

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

\$15

Info

Director: Stanley Kubrick
Year of production: 1980
Duration(min): 185
Place: Пионер
Date: 2022-12-25
Time: 23:00:00.000000

Рисунок 3.4 – Окно мероприятия

В окне мероприятия находится кнопка для покупки билета. Нажав на кнопку, откроется окно с выбором ряда и места. Так как пользователь авторизован и в базе данных имеются все необходимые данные о клиенте: электронная почта, ФИО, возраст, покупка билета оформляется моментально, а в личном кабинете у пользователя появляется информация о купленном билете.

Обобщая, данный сервис является классическим примером сайта по продаже билетов на культурные мероприятия. Этот сервис имеет предельно ясный функционал, поэтому пользователю не составит труда купить билеты, не прибегая к дополнительной помощи и не тратя лишнее время.

3.3 Реализация клиентских запросов

Приложение было разработано для покупки билетов на культурные мероприятия.

В приложении присутствуют все необходимые функции для покупателей.

Зарегистрированный пользователь может использовать все предложенные функции.

Незарегистрированный пользователь может лишь просмотреть все доступные мероприятия, но не может покупать билеты, т.к. не имеет личного кабинета на данном ресурсе.

3.4 Обоснование и реализация механизма обеспечения безопасности и сохранности данных

Механизмом обеспечения безопасности является протокол *HTTPS*. *HTTPS* расширение протокола *HTTP* для поддержки шифрования в целях повышения безопасности. Данные в протоколе *HTTPS* передаются поверх криптографических протоколов *TLS*.

HTTPS не является отдельным протоколом. Это обычный *HTTP*, работающий через зашифрованные транспортные механизмы *SSL* и *TLS*. Он обеспечивает защиту от атак, основанных на прослушивании сетевого соединения — от снифферских атак, при условии, что будут использоваться шифрующие средства и сертификат сервера проверен и ему доверяют. Криптографические протоколы, которые используются в *HTTPS*, имеют несколько уровней защиты:

- шифрование данных;
- сохранность данных;
- аутентификация.

Работа протокола *HTTPS* основана на том, что сервер и устройство пользователя выбирают единый секретный ключ, с помощью которого происходит шифрование данных. Этот ключ является уникальным и генерируется при каждом сеансе.

Для выполнения входа в приложение необходимо выполнить авторизацию. Для успешной авторизации нужно знать логин, описанный при регистрации и пароль.

Пароли пользователей, в свою очередь, защищены благодаря функции *md5*, которая позволяет создать хэш из данных. Таким образом, даже администратор базы данных не сможет просмотреть пароль пользователя, что обеспечивает превосходную защиту данных людей, зарегистрированных на данном веб-сервисе.

ЗАКЛЮЧЕНИЕ

В данной курсовой работе был разработан сервис по продаже билетов на культурные мероприятия, который полностью решает поставленную задачу и удовлетворяет всем поставленным требованиям. А также предложен план алгоритма по покупке билетов на мероприятия, с последующим отображением в личном кабинете пользователя. Реализована удобная система навигации по сайту, а именно фильтрация на основании типа мероприятия.

На основании выполненных задач по курсу курсового проектирования можно сказать, что были исследованы объект (базы данных) и субъект (клиент серверные приложения).

Теоретическая часть данной курсовой работы была реализована благодаря изучению информации представленной в:

- сети интернет;
- книгах;
- лекционных материалах;
- учебно-методических пособиях.

Разработанная программа значительно облегчает покупку билетов, она является удобной альтернативой обычным кассам, на которые люди тратят много времени.

Разработанная база данных соответствует всем требованиям предметной области:

- требование полноты и непротиворечивости данных;
- многократное использование данных;
- быстрый поиск и получение информации по запросам пользователей;
- простота обновления данных;
- защита данных от несанкционированного доступа.

В процессе написания курсового проекта были получены и закреплены на практике навыки проектирования баз данных, приобретены умения построения клиент-серверного приложения на базе архитектуры *MVC*, а также навыки программирования сайтов, с понятным и удобным интерфейсом, и полной функциональностью.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Формализация в данных. [Электронный ресурс]. – Режим доступа: https://universal_ru_en.academic.ru/2947197/формализация_данных – Дата доступа: 22.11.2022
- [2] Первичный ключ [Электронный ресурс]. – Режим доступа: <http://informatic.ugatu.ac.ru/lib/office/Proekt.htm> – Дата доступа: 22.11.2022
- [3] Запросы в базе данных [Электронный ресурс]. – Режим доступа: <https://support.microsoft.com/ru-ru/office/создание-простого-запроса-на-выборку-de8b1c8d-14e9-4b25-8e22-70888d54de59> – Дата доступа: 22.11.2022
- [4] Визуальное представление базы данных [Электронный ресурс]. – Режим доступа: <https://coderoad.ru/2441090/Визуальное-Представление-Схемы-Базы-Данных> – Дата доступа: 27.11.2022
- [5] Архитектура MVC [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/181772/> – Дата доступа: 27.11.2022
- [6] Пассивное представление MVC [Электронный ресурс]. – Режим доступа: <https://php.zone/oop-v-php-prodvintuyj-kurs/arhitektura-prilozheniya-i-pattern-mvc> Дата доступа: 27.11.2022
- [7] Авторизация пользователей [Электронный ресурс]. – Режим доступа: <https://cropas.by/seo-slovar/avtorizatsiya/> – Дата доступа: 27.11.2022
- [8] Авторизация пользователей [Электронный ресурс]. – Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Authentication> – Дата доступа: 13.04.2022
- [9] Механизм обеспечения безопасности [Электронный ресурс]. – Режим доступа: https://studme.org/126495/mechanizmy_obespecheniya_bezopasnost – Дата доступа: 13.04.2022
- [10] *MySQL Workbench* запросы [Электронный ресурс]. – Режим доступа: https://wiki.livid.pp.ru/students/dbms/labs/01_Workbench.html – Дата доступа: 13.04.2022

ПРИЛОЖЕНИЕ А
(обязательное)
Справка проверки на «Антиплагиат»

ПЕРЕМЕСТИТЬ

УДАЛИТЬ

ИСТОРИЯ ОТЧЕТОВ

Название

Дата загрузки

Оригинальность

PDF

записка_проверка

10 Дек 2022
18:39

88,76%

Рисунок А.1 – Проверка в системе «Антиплагиат»

ПРИЛОЖЕНИЕ Б

(обязательное)

Скрипт генерации базы данных

```
CREATE TABLE `Описание` (  
  `id` int NOT NULL,  
  `Год` int NOT NULL,  
  `Руководитель` varchar(15) NOT NULL,  
  `Длительность` int NOT NULL,  
  `Название` varchar(15) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
  
INSERT INTO `Описание` (`id`, `Год`, `Руководитель`,  
  `Длительность`, `Название`) VALUES  
(1, 2015, 'Jordan Pele', 135, 'Get Out'),  
(2, 2010, 'David Fincher', 165, 'Social Network'),  
(3, 1980, 'Stanley Kubrick', 185, 'The Shining'),  
(4, 1985, 'S. Spielberg', 124, 'Jurassic Park'),  
(5, 1995, 'Q. Tarantino', 123, 'Pulp Fiction'),  
(6, 1989, 'R. Zemeckis', 130, 'Back to Future'),  
(7, 2019, 'John Doe', 120, 'Weeping Game'),  
(8, 2020, 'James Sun', 140, 'Mac Beth'),  
(9, 2021, 'Anna Kittng', 190, 'Tigar'),  
(10, 2018, 'Maya Sito', 175, 'Man of LaManch'),  
(11, 2016, 'Sam Winchester', 170, 'Carmen'),  
(12, 2018, 'Elena Bor', 160, 'Aida'),  
(13, 1965, 'Ted Mosby', 90, 'Jumbo'),  
(14, 1987, 'John Wick', 85, 'Tiger Craziiness');  
  
CREATE TABLE `Билеты` (  
  `id` int NOT NULL,  
  `Место` int NOT NULL,  
  `Цена` int NOT NULL,  
  `Название` varchar(50) NOT NULL,  
  `владелец` varchar(15) CHARACTER SET utf8mb4 COLLATE  
utf8mb4_general_ci NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_0900_ai_ci;  
  
INSERT INTO `Билеты` (`id`, `Место`, `Цена`, `Название`,  
  `владелец`) VALUES  
(11, 0, 15, 'Get Out', 'loh'),  
(12, 0, 15, 'Get Out', 'loh'),  
(13, 0, 15, 'Get Out', 'loh'),  
(14, 0, 15, 'Get Out', 'loh'),  
(15, 16, 15, 'Get Out', 'loh'),  
(16, 1, 15, 'Get Out', 'loh'),  
(17, 0, 15, 'Get Out', 'loh'),  
(18, 3, 15, 'Get Out', 'loh');
```



```
CREATE TABLE `Данные Клиента` (
  `id` int UNSIGNED NOT NULL,
  `логин` varchar(30) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci NOT NULL,
  `Имя` varchar(15) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci NOT NULL,
  `Электронная почта` varchar(100) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci NOT NULL,
  `пароль` varchar(32) CHARACTER SET utf8mb3 COLLATE
utf8mb3_general_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `Данные Клиента` (`id`, `логин`, `Имя`, `Электронная
почта`, `пароль`) VALUES
(3, 'Vanyaааааа', 'Ivan', 'ivantsrevich@gmail.com',
'67a3b21b57bcfd9237c5be418e7d367a'),
(7, 'loh', 'loh', 'loh@gmail.com',
'3ff1ea09b981d88e7c8752b329a7702e'),
(4, 'tainipokupatel', 'Matvei', 'matvei@gmail.com',
'827ccb0eea8a706c4c34a16891f84e7b'),
(5, 'nikitaaa', 'Nikita', 'nikita@gmail.com',
'202cb962ac59075b964b07152d234b70'),
(1, 'olejaaa', 'oleg', 'olegsvec742@gmail.com', '123123'),
(2, 'senamar', 'Senya', 'senmar336@gmail.com', '1234');
```

```
CREATE TABLE `Тип Мероприятия` (
  `id` int NOT NULL,
  `Тип` varchar(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
COLLATE=utf8mb4_0900_ai_ci;
```

```
INSERT INTO `Тип Мероприятия` (`id`, `Тип`) VALUES
(1, 'Кино'),
(2, 'Театр'),
(3, 'Цирк'),
(4, 'Прочее');
```

```
ALTER TABLE `Мероприятие`
  ADD PRIMARY KEY (`Название`),
  ADD KEY `Тип мероприятия` (`Тип мероприятия`);
```

```
ALTER TABLE `Описание`
  ADD KEY `Название` (`Название`);
```

```
ALTER TABLE `Билеты`
  ADD PRIMARY KEY (`id`),
  ADD KEY `Название` (`Название`),
  ADD KEY `владелец` (`владелец`);
```

```
ALTER TABLE `Данные Клиента`
  ADD PRIMARY KEY (`Имя`),
  ADD UNIQUE KEY `id` (`id`);
ALTER TABLE `Тип Мероприятия`
  ADD PRIMARY KEY (`id`);
```

```

ALTER TABLE `Билеты`
  MODIFY `id` int NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;
ALTER TABLE `Данные Клиента`
  MODIFY `id` int UNSIGNED NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=8;
ALTER TABLE `Мероприятие`
  ADD CONSTRAINT `мероприятие_ibfk_1` FOREIGN KEY (`Тип
мероприятия`) REFERENCES `Тип Мероприятия` (`id`) ON DELETE
RESTRICT ON UPDATE RESTRICT;
ALTER TABLE `Описание`
  ADD CONSTRAINT `описание_ibfk_1` FOREIGN KEY (`Название`)
REFERENCES `Мероприятие` (`Название`) ON DELETE RESTRICT ON
UPDATE RESTRICT;
ALTER TABLE `Билеты`
  ADD CONSTRAINT `билеты_ibfk_1` FOREIGN KEY (`Название`)
REFERENCES `Мероприятие` (`Название`) ON DELETE RESTRICT ON
UPDATE RESTRICT,
  ADD CONSTRAINT `билеты_ibfk_2` FOREIGN KEY (`владелец`)
REFERENCES `Данные Клиента` (`Имя`) ON DELETE RESTRICT ON UPDATE
RESTRICT;
COMMIT;

```

ПРИЛОЖЕНИЕ В
(обязательное)
Листинг программного кода

```
<?php
error_reporting(E_ALL | E_STRICT);
ini_set('display_errors', TRUE);
ini_set('display_startup_errors', TRUE);
$login = filter_var(trim($_POST['login']));
$name = filter_var(trim($_POST['name']));
$email = filter_var(trim($_POST['email']),
FILTER_SANITIZE_EMAIL);
$password = filter_var(trim($_POST['password']));
$password = md5($password);

$mysql = new mysqli('localhost', 'root', '', 'Kurovaya');

$mysql->query("INSERT INTO `Данные Клиента` (`логин`, `Имя`,
`Электронная почта`, `пароль`)
VALUES('$login', '$name', '$email', '$password')");
$mysql->close();
header('Location: index.php');
?>
```

ПРИЛОЖЕНИЕ Г
(обязательное)
Ведомость курсового проекта