

Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина «Модели данных и системы управления базами данных»

«К ЗАЩИТЕ ДОПУСТИТЬ»

Руководитель курсового проекта
ассистент

_____._____.2023 А.В. Давыдчик

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему:

«Веб-приложение для начинающих барменов»

БГУИР КП 1-40 04 01 023 ПЗ

Выполнил студент группы 053504
Сергеев Евгений Геннадьевич

(подпись студента)

Курсовой проект представлен на
проверку _____._____.2023

(подпись студента)

Минск 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
1 МОДЕЛИРОВАНИЕ ДАННЫХ	6
1.1 Типы моделей данных	6
1.2 Методы моделирования данных	9
1.3 Процесс моделирования данных.....	10
1.4 Системы управления базами данных	11
2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ.....	12
2.1 <i>PostgreSQL</i>	12
3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ.....	14
4 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ	Error! Bookmark not defined.
5 АРХИТЕКТУРА ВЕБ-ПРИЛОЖЕНИЯ	Error! Bookmark not defined.
ЗАКЛЮЧЕНИЕ	Error! Bookmark not defined.
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	Error! Bookmark not defined.

ВВЕДЕНИЕ

В современном цифровом мире, в котором доступность информации и скорость обработки данных становятся все более важными, внедрение эффективных баз данных становится ключевым компонентом многих веб-приложений. Базы данных служат фундаментальным инструментом для организации информации и обеспечения доступа к ней во всех сферах деятельности, начиная от предприятий и исследовательских организаций и заканчивая личным использованием.

Одним из живых примеров приложений, требующих эффективного хранения и управления информацией, являются системы бронирования билетов в кинотеатрах. Киноиндустрия, как и многие другие, вынуждена адаптироваться к современным требованиям пользователей и рынка, предоставляя возможность онлайн-бронирования и приобретения билетов через интернет. Возможность выбора сеанса, места и оплаты билета через интернет упрощает жизнь многих людей.

Создание и поддержание таких систем требует не только удобного пользовательского интерфейса, но и мощной инфраструктуры для обработки и хранения данных о фильмах, сеансах, свободных местах и бронированиях. Для эффективного управления этой информацией используются базы данных, которые позволяют хранить, организовывать и обеспечивать быстрый доступ к данным.

В данной курсовой работе будет спроектирована и реализована база данных для системы бронирования билетов в кинотеатрах, организованы таблицы для хранения информации, а также как обеспечена безопасность и производительность при работе с данными. Также в данной работе будет создано веб-приложение, использующее спроектированную базу данных и предоставляющее необходимый функционал для работы с ней.

1 МОДЕЛИРОВАНИЕ ДАННЫХ

Моделирование данных – это процесс создания визуального представления или чертежа, определяющего системы сбора и управления информацией в любой организации. Этот план или модель данных помогает различным заинтересованным сторонам, таким как аналитики данных, ученые и инженеры, создать единое представление о данных организации. Модель описывает, какие данные собирает компания, взаимосвязь между различными наборами данных и методы, которые будут использоваться для хранения и анализа данных.

Модели данных строятся на основе бизнес-потребностей. Правила и требования к модели данных определяются заранее на основе обратной связи с бизнесом, поэтому их можно включить в разработку новой системы или адаптировать к существующей.

1.1 Типы моделей данных

Разработка баз данных и информационных систем начинается с высокого уровня абстракции и с каждым шагом становится все точнее и конкретнее. В зависимости от степени абстракции модели данных можно разделить на три категории. Процесс начинается с концептуальной модели, переходит к логической модели и завершается физической моделью.

Концептуальные модели данных, которые также называются моделями предметной области, описывают общую картину: что будет содержать система, как она будет организована и какие бизнес-правила будут задействованы. Концептуальные модели обычно создаются в процессе сбора исходных требований к проекту. Как правило, они включают классы сущностей (вещи, которые бизнесу важно представить в модели данных), их характеристики и ограничения, отношения между сущностями, требования к безопасности и целостности данных. Любые обозначения обычно просты.

Заинтересованные стороны и аналитики обычно создают концептуальную модель. Это простое диаграммное представление, которое не следует формальным правилам моделирования данных. Важно то, что оно помогает как техническим, так и нетехническим заинтересованным сторонам разделить общее видение и согласовать цель, объем и дизайн проекта по работе с данными. Концептуальные модели выступают в качестве моста между бизнес-правилами и лежащей в их основе физической системой

управления базами данных (СУБД). Схема концептуальной модели данных представлена на рисунке 1.

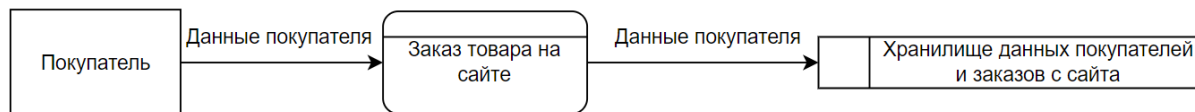


Рисунок 1 – Концептуальная модель

Логические модели данных отображают концептуальные классы данных на технических структурах данных. Они дают более подробную информацию о концепциях данных и их сложных отношениях, которые были определены в концептуальной модели данных, например:

- типы данных различных атрибутов (например, строка или число)
- взаимосвязи между объектами данных
- первичные атрибуты или ключевые поля в данных

Архитекторы данных и аналитики совместно работают над созданием логической модели. Они следуют одной из нескольких формальных систем моделирования данных для создания представления. Иногда гибкие команды могут пропустить этот шаг и перейти от концептуальных к физическим моделям напрямую. Однако эти модели полезны для проектирования больших баз данных, называемых хранилищами данных, и для проектирования систем автоматической отчетности.

Логические модели служат связующим звеном между концептуальной моделью данных и базовой технологией и языком баз данных, которые разработчики используют для создания базы данных. Однако они не зависят от технологии, и вы можете реализовать их на любом языке баз данных. Инженеры по данным и заинтересованные стороны обычно принимают технологические решения после создания логической модели данных. Логическая модель данных изображена на рисунке 2.

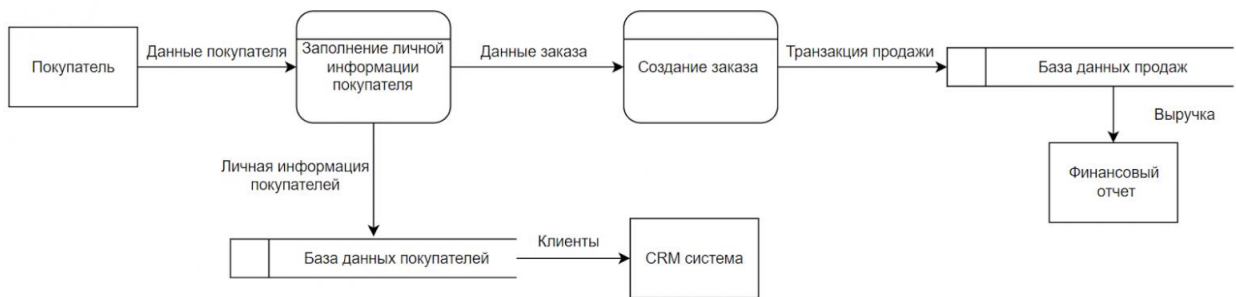


Рисунок 2 – Логическая модель

Физические модели данных представляют схему того, как данные будут храниться в базе. По сути, это наименее абстрактные из всех моделей. Они предлагают окончательный дизайн, который может быть реализован как реляционная база данных, включающая ассоциативные таблицы, которые иллюстрируют отношения между сущностями, а также первичные и внешние ключи для связи данных. Производительность базы данных, стратегия индексации, физическое хранилище и денормализация — важные параметры физической модели. Физическая модель данных выступает в качестве моста между логической моделью данных и конечной технологической реализацией. Физическая модель данных показана на рисунке 3.

Физическая модель данных зависит от конкретной системы управления базами данных, фактически являясь отображением системного каталога. В физической модели содержится информация о всех объектах базы данных. Поскольку стандартов на объекты базах данных не существует, например, нет стандарта на типы данных, физическая модель зависит от конкретной реализации системы управления базами данных. Следовательно, одной и той же логической модели могут соответствовать несколько разных физических моделей. Если в логической модели не имеет значения, какой конкретно тип данных имеет атрибут, то в физической модели важно описать всю информацию о конкретных физических объектах - таблицах, колонках, индексах, процедурах и т. д.

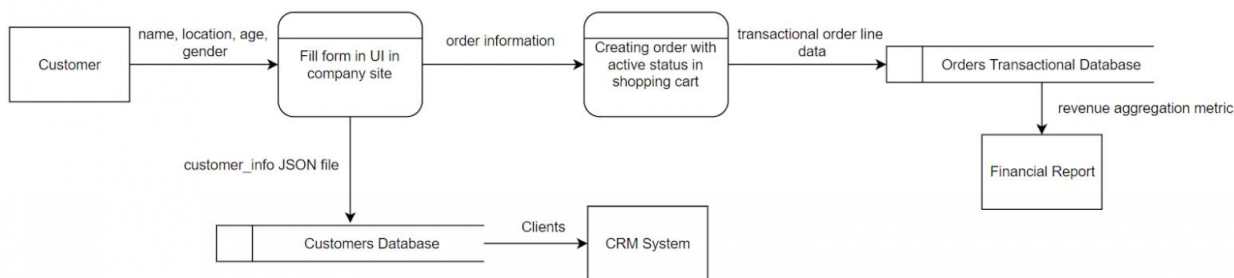


Рисунок 3 – Физическая модель данных

1.2 Методы моделирования данных

Методы моделирования данных — это различные методы, которые можно использовать для создания различных моделей данных. Эти подходы развивались с течением времени в результате инноваций в концепциях баз данных и управления данными. Далее представлены основные типы моделирования данных.

Иерархические модели данных представляют отношения «один ко многим» в древовидном формате. В модели этого типа каждая запись имеет единственный корень или родительский элемент, который сопоставляется с одной или несколькими дочерними таблицами. Эта модель была реализована в IBM Information Management System (IMS) в 1966 году и быстро нашла широкое применение, особенно в банковской сфере. Хотя этот подход менее эффективен, чем недавно разработанные модели баз данных, он все еще используется в системах расширяемого языка разметки (XML) и географических информационных системах (ГИС).

При иерархическом моделировании данных можно представить отношения между различными элементами данных в древовидном формате. Иерархические модели данных представляют собой отношения «один ко многим», когда родительские или корневые классы данных отображаются на несколько дочерних.

Иерархическое моделирование данных со временем превратилось в графовое. Графовые модели данных представляют отношения данных, в которых сущности рассматриваются одинаково. Сущности могут связываться друг с другом отношениями «один ко многим» или «многие ко многим» без понятия «родительский» или «дочерний».

Реляционные модели данных были предложены исследователем IBM Э. Ф. Коддом в 1970 году. Они до сих пор встречаются во многих реляционных базах данных, обычно используемых в корпоративных вычислениях.

Реляционное моделирование не требует детального понимания физических свойств используемого хранилища данных. В нем сегменты данных объединяются с помощью таблиц, что упрощает базу данных.

Моделирование данных «сущность-связь» (ER) использует формальные диаграммы для представления отношений между сущностями в базе данных. Архитекторы данных используют несколько инструментов ER-моделирования для представления данных.

В объектно-ориентированном программировании для хранения данных используются структуры данных, называемые объектами. Эти объекты данных являются программными абстракциями объектов реального мира. Например, в объектно-ориентированной модели данных автосалон будет иметь такие объекты данных, как «Клиенты», с такими атрибутами, как имя, адрес и номер телефона. Вы будете хранить данные о клиентах таким образом, чтобы каждый реальный клиент был представлен в виде объекта данных о клиенте.

Современные корпоративные вычисления используют технологию хранилищ данных для хранения больших объемов данных для аналитики. Вы можете использовать проекты многомерного моделирования данных для их высокоскоростного хранения и извлечения из хранилища. Многомерные модели используют дублирование или избыточные данные и отдают приоритет производительности, а не использованию меньшего пространства для хранения данных.

1.3 Процесс моделирования данных

Моделирование данных начинается с договоренности о том, какие символы используются для представления данных, как размещаются модели и как передаются бизнес-требования. Это формализованный рабочий процесс, включающий ряд задач, которые должны выполняться итеративно. Сам процесс обычно выглядит так:

- 1 Определите сущности. На этом этапе идентифицируем объекты, события или концепции, представленные в наборе данных, который необходимо смоделировать. Каждая сущность должна быть целостной и логически отделенной от всех остальных.

- 2 Определите ключевые свойства каждой сущности. Каждый тип сущности можно отличить от всех остальных, поскольку он имеет одно или несколько уникальных свойств, называемых атрибутами. Например, сущность «клиент» может обладать такими атрибутами, как имя, фамилия, номер

телефона и т.д. Сущность «адрес» может включать название и номер улицы, город, страну и почтовый индекс.

3 Определите связи между сущностями. Самый ранний черновик модели данных будет определять характер отношений, которые каждая сущность имеет с другими. В приведенном выше примере каждый клиент «живет по» адресу. Если бы эта модель была расширена за счет включения сущности «заказы», каждый заказ также был бы отправлен на адрес. Эти отношения обычно документируются с помощью унифицированного языка моделирования (UML).

4 Полностью сопоставьте атрибуты с сущностями. Это гарантирует, что модель отражает то, как бизнес будет использовать данные. Широко используются несколько формальных шаблонов (паттернов) моделирования данных. Объектно-ориентированные разработчики часто применяют шаблоны для анализа или шаблоны проектирования, в то время как заинтересованные стороны из других областей бизнеса могут обратиться к другим паттернам.

5 Назначьте ключи по мере необходимости и определите степень нормализации. Нормализация — это метод организации моделей данных, в которых числовые идентификаторы (ключи) назначаются группам данных для установления связей между ними без повторения данных. Например, если каждому клиенту назначен ключ, этот ключ можно связать как с его адресом, так и с историей заказов, без необходимости повторять эту информацию в таблице с именами клиентов. Нормализация помогает уменьшить объем дискового пространства, необходимого для базы данных, но может сказываться на производительности запросов.

6 Завершите и проверьте модель данных. Моделирование данных — это итеративный процесс, который следует повторять и совершенствовать под потребности бизнеса.

1.4 Системы управления базами данных

Система управления базами данных (СУБД) — это программное обеспечение, предназначенное для хранения, извлечения и управления данными. Наиболее распространенной СУБД в системе баз данных предприятия является СУРБД. Полная форма СУБД — это система управления реляционными базами данных.

Согласно реляционной модели Э. Ф. Кодда, СУБД позволяет пользователям создавать, обновлять, управлять реляционной базой данных и взаимодействовать с ней, позволяя хранить данные в табличной форме.

Система управления реляционными базами данных организует данные в таблицах, которые могут быть связаны внутри в зависимости от общих данных. Это позволяет пользователю легко получить одну или несколько таблиц с помощью всего одного запроса. С другой стороны, в плоском файле данные хранятся в единой табличной структуре, что менее эффективно и требует больше места и памяти.

Наиболее коммерчески доступной и общекорпоративной системой управления базами данных или системой управления реляционными базами данных, используемой сегодня, является язык структурированных запросов (база данных *SQL*) для доступа к базе данных. Другие широко используемые системы управления реляционными базами данных в компаниях включают базу данных *Oracle*, *MySQL*, *PostgreSQL* (реляционная база данных с открытым исходным кодом) и *Microsoft SQL Server*. Структуры РСУБД обычно используются для выполнения четырех основных операций: *CRUD* (создание, чтение, обновление и удаление), которые имеют решающее значение для поддержки согласованного управления данными.

2 ПЛАТФОРМА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 *PostgreSQL*

PostgreSQL, также известная как *Postgres*, является мощной и открытой системой управления реляционными базами данных, которая пользуется высокой популярностью в мире разработки приложений и анализа данных. Основанный на модели объектно-реляционной базы данных (*ORDBMS*), *PostgreSQL* обеспечивает надежное и гибкое хранение и обработку данных для разнообразных приложений.

PostgreSQL имеет открытый исходный код, что делает его привлекательным для широкого круга разработчиков и организаций. СУБД поддерживает мощный *SQL*, включая сложные запросы и агрегатные функции. Расширяемость *PostgreSQL* позволяет легко добавлять новые типы данных и функциональность.

СУБД обладает встроенной поддержкой *JSON*-данных, что полезно для работы с полуструктурированными данными. Она также гарантирует *ACID*-совместимость, обеспечивая надежность и целостность данных.

PostgreSQL поддерживает многопоточность и параллелизм для высокой производительности и масштабируемости. Он также предоставляет инструменты для настройки репликации и обеспечения отказоустойчивости.

Активное сообщество разработчиков и пользователей обеспечивает поддержку, обновления и регулярное выпуск новых версий. *PostgreSQL* – это мощная и гибкая СУБД, подходящая для различных задач от веб-приложений до анализа данных.

3 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ

В современном обществе эффективная регистрация актов гражданского состояния является неотъемлемой частью правовой системы. Регистрация рождений, браков и смертей играет важную роль в поддержании прав и обязанностей граждан, а также в статистическом анализе, планировании и принятии стратегических решений государством и обществом в целом. В этой связи разработка веб-приложения для регистрации актов гражданского состояния имеет важное значение, поскольку она может значительно улучшить процесс регистрации, обеспечить сохранность данных и улучшить доступность услуг для пользователей.

Существующая система регистрации актов гражданского состояния часто характеризуется рядом проблем, включающих в себя:

- 1 Недостаточную эффективность и длительность процесса регистрации. Низкую точность и надежность хранения данных.
- 2 Ограниченный доступ для пользователей к услугам регистрации.
- 3 Значительное количество бумажной работы и исчезновение документов.
- 4 Нецентрализованность и сложность связи между различными органами, занимающимися регистрацией.

Разработка веб-приложения для регистрации актов гражданского состояния позволит преодолеть эти проблемы и достичь следующих преимуществ:

Ускорение процесса регистрации: Автоматизация основных этапов регистрации и устранение необходимости личного присутствия позволит существенно сократить время ожидания и ускорить обработку запросов.

Веб-приложение позволит создать централизованную базу данных, где каждый акт будет сохранен безопасно и доступен для дальнейшего использования и обработки. Это снизит риск потери документов и обеспечит их долговременное хранение.

Веб-приложение будет доступно из любой точки сети интернет, что позволит пользователям регистрировать акты гражданского состояния в удобное время и без необходимости посещать офисы государственных учреждений.

Замена традиционных бумажных документов электронными форматами позволит снизить использование бумаги, упростить процесс архивации и защитить документы от физических повреждений или потерь.

Централизованная система регистрации позволит органам государственной власти и органам гражданского состояния связываться и обмениваться информацией более эффективно, минимизируя ошибки и повторную регистрацию.

Таким образом, разработка веб-приложения для регистрации актов гражданского состояния предоставит новые возможности для повышения эффективности, точности и доступности регистрационных услуг. Это сократит время, ресурсы и риски, связанные с обработкой актов гражданского состояния, и сделает процесс более удобным и простым для всех заинтересованных сторон.