Частное учреждение профессионального образования
«Высшая школа предпринимательства»
(ЧУПО «ВШП»)

КУРСОВОЙ ПРОЕКТ

«База данных для бара “Кооператив”»

Выполнил:
студент 3-го курса специальности
09.02.07 «Информационные системы и программирование»
Матвеев Кирилл Дмитриевич
подпись: _____

Проверил:
преподаватель дисциплины,
преподаватель ЧУПО «ВШП»,
к.ф.н. Ткачев П.С.
оценка: _____
подпись: _____

Содержание

Тверь, 2024 г.

Введение

Актуальность темы

Базы данных играют важную роль в современном мире технологий, так как они позволяют сохранять, организовывать и обрабатывать различную информацию. Сегодня практически каждая организация использует базы данных для управления данными. Это актуально и для баров, где информация о клиентах, заказах, поставках и сотрудниках хранится и обрабатывается с помощью специализированных систем. Работа с базами данных становится важным навыком для успешного ведения бизнеса, а специалисты в этой области — всё более востребованными.

Цель работы

Цель данной работы заключается в разработке и реализации реляционной ба-зы данных для бара "Кооператив", обеспечивающей эффективное управление данными о клиентах, напитках, сотрудниках и заказах.

Задачи

Для достижения заданной мне цели, мне нужно выполнить следующие задачи:

- Обозначить требования к функционалу базы данных
- Выбрать инструменты и технологии для разработки
- Спроектировать базу данных на основе требований к функционалу
- Реализовать базу данных на основе спроектированной схемы
- Определить требования к базе данных, обеспечивающие целостность системы

1. Теоритические основы разработки базы данных

1.1. Понятие базы данных и её основные функции

База данных - это организованная структура, которая предназначена для хранения информации. В то время, когда происходило развитие термина баз данных, в них сохранялись исключительно информация, однако уже в наши дни многие системы управления базами данных позволяет размещать в своих структурах и данные, и программный код, с помощью которого совершается связь с пользователями или с другими программно-аппаратными комплексами. При этом данные не должны противоречить друг другу, быть целостными и не избыточными. База данных создается для сохранения и непосредственного доступа к информации, содержащей сведения об искомой предметной области. Прежде всего, целью использования информации из баз данных и сложностью информационных процессов, существующих в пределах предметной области в конкретных условиях. Основными функциями базы данных являются:

- Хранение данных: БД обеспечивает безопасное и надёжное хранение информации.
- Обработка данных: с помощью запросов и операций можно выполнять различные действия с данными, такие как поиск, сортировка, фильтрация и т. д.
- Анализ данных: базы данных позволяют проводить анализ данных, выявлять закономерности и тенденции, а также принимать обоснованные решения на основе полученной информации.

Система управления базами данных - это программный механизм, предназначенный для записи, поиска, сортировки, обработки и печати информации, содержащейся в базе данных. Иными словами, система управления базами данных - механизм, регулирующий работу самой базы данных.[14]

1.2. Типы баз данных

Существует множество типов баз данных, каждый из которых имеет свои особенности и преимущества в зависимости от поставленной задачи и требований проекта. В данной работе будут рассмотрены основные типы баз данных:

- Реляционные базы данных (РБД) — это наиболее распространённый тип баз данных. Они основаны на реляционной модели данных и используют таблицы для хранения информации. Реляционные базы данных имеют

множество преимуществ, таких как простота использования, надёжность и эффективность. Однако они также имеют некоторые недостатки, такие как сложность масштабирования и ограничения в производительности при работе с большими объёмами данных. [10]

- Объектно-ориентированные базы данных (ООБД) — этот тип баз данных основан на объектно-ориентированной модели данных. В ООБД данные представлены в виде объектов, а не таблиц. Это позволяет более точно моделировать сложные предметные области и обеспечивает более высокую гибкость при разработке приложений. Однако ООБД имеют более сложную структуру и требуют более высокой квалификации разработчиков. [2]
- Иерархические базы данных (ИБД) — в иерархических базах данных информация организована в виде древовидной структуры. ИБД используются для моделирования иерархических отношений между объектами, например, в системах управления проектами или в каталогах товаров. Иерархические базы данных просты в использовании и обеспечивают хорошую производительность при работе с небольшими объёмами данных, но они не подходят для сложных запросов и обработки больших объёмов данных. [8]
- Сетевые базы данных (СБД) — сетевые базы данных похожи на иерархические, но в них допускается наличие нескольких родительских узлов у одного дочернего узла. СБД используются для моделирования сложных отношений между объектами и обеспечивают большую гибкость по сравнению с иерархическими базами данных. Однако СБД сложнее в реализации и имеют более низкую производительность по сравнению с реляционными базами данных. [9]
- NoSQL базы данных — NoSQL базы данных представляют собой класс нереляционных баз данных, который включает в себя широкий спектр технологий, разработанных для обеспечения масштабируемости и гибкости. NoSQL базы данных не используют схемы и могут обрабатывать различные типы данных, включая документы, пары ключ-значение и графы. Они часто используются для обработки больших объёмов неструктурированных данных, таких как данные социальных сетей, журналов веб-сервера и т. д. [11]
- Графовые базы данных — графовые базы данных предназначены для работы с данными, представленными в виде графа. Графовые базы данных позволяют эффективно моделировать и обрабатывать отношения между объектами. Они используются в таких областях, как социальные сети, рекомендательные системы и анализ данных.[12]
- RDF (Resource Description Framework) — это стандарт представления данных, который используется для описания ресурсов в интернете. RDF

позволяет описывать ресурсы с помощью троек «субъект-предикат-объект», где субъект и объект являются ресурсами, а предикат описывает отношение между ними. RDF является основой для создания семантических веб-приложений, которые используют машинное понимание данных для предоставления более интеллектуальных услуг.[13]

1.3. Инструменты для разработки базы данных

Разработка баз данных включает в себя проектирование, создание, управление и оптимизацию баз данных. Для этих целей существует множество инструментов, каждый из которых предлагает определенные функции и возможности. В этом разделе будут рассмотрены основные инструменты для разработки баз данных.

Инструменты для разработки баз данных делятся на следующие категории:

- **Системы управления базами данных (СУБД)** - это программное обеспечение, предназначенное для создания, управления и модификации баз данных.
  Например, MySQL - распространенная реалиационная СУБД с открытым исходным кодом.

- **Инструменты моделирования данных:** помогают разработчикам создавать и визуализировать структуры баз данных.
  Например, Microsoft Visio - универсальный инструмент для создания диаграмм, включая диаграммы сущность-связь (ERD).

- **Инструменты управления базами данных:** предоставляют интерфейсы для администрирования и управления базами данных.
  Например, MySQL Workbench - инструмент для управления СУБД MySQL.

- **Инструменты разработки и интеграции:** облегчают процесс написания, тестирования и отладки SQL-кода и приложений, работающих с базами данных.
  Например, DBeaver - многофункциональный инструмент для работы с различными СУБД, поддерживающий множество форматов данных.

- **Инструменты мониторинга и оптимизации:** эти инструменты помогают отслеживать производительность баз данных и оптимизировать запросы.
  Например, Oracle AWR - инструмент для анализа производительности баз данных Oracle.

Для разработки следует в лучшем случае выбрать инструмент из каждой категории, изходя из используемой СУБД и из личных предпочтений. Использование данных инструментов оптимизирует работу разработчика и сделает ее намного проще и качественнее.

1.4. Нормализация баз данных

Нормализация — это процесс организации структуры базы данных таким образом, чтобы минимизировать избыточность данных и обеспечить целостность информации. Нормализация позволяет улучшить производительность базы данных, упростить её обслуживание и обеспечить более эффективное использование ресурсов.

Существует несколько уровней нормализации, каждый из которых устраняет определённые виды избыточности. Чем выше уровень нормализации, тем больше таблиц требуется для представления данных, но при этом повышается целостность данных и уменьшается вероятность возникновения аномалий обновления. Уровни нормализации существуют следуюшие:

- **Первый уровень нормализации**, или 1НФ, является базовым уровнем нормализации реляционных баз данных. Он требует, чтобы все атрибуты были атомарными, то есть не содержали в себе других значений. Это означает, что каждый атрибут должен представлять собой одно значение, а не набор значений или список.
- **Второй уровень нормализации**, или 2НФ, является следующим шагом после достижения 1НФ. Он требует выполнения 1НФ и отсутствия частичной зависимости неключевых атрибутов от первичного ключа. Частичная зависимость возникает, когда неключевой атрибут зависит только от части первичного ключа. Для устранения частичной зависимости необходимо разбить таблицу на две или более таблиц.

- **Третий уровень нормализации**, или 3НФ, является самым строгим уровнем нормализации реляционных баз данных. Он требует выполнения 2НФ и отсутствия транзитивной зависимости неключевых атрибутов от первичного ключа. Транзитивная зависимость возникает, когда неключевой атрибут зависит от другого неключевого атрибута, который, в свою очередь, зависит от первичного ключа.

- **Нормальная форма Бойса-Кодда (BCNF)** — для любой функциональной зависимости X → A в отношении R атрибут A должен функционально

зависеть от каждого элемента в X (то есть X должен быть суперключом отношения R).

- **Четвёртый уровень (4NF)** — отношение находится в 4NF, если оно находится в НФБК и все нетривиальные многозначные зависимости фактически являются функциональными зависимостями от её потенциальных ключей.

- **Пятый уровень или нормальная форма проекции-соединения (5NF или PJ/NF)** — любая зависимость соединения в ней определяется потенциальными ключами отношения (ключами исходного отношения). Также отношение R соответствует 5NF тогда и только тогда, когда каждая нетривиальная зависимость соединения в R следует из существования некоторого потенциального ключа K для R.

- **Шестой уровень нормализации (6NF)** — также известен как Domain/Key Normal Form (DKNF) или идеальная нормальная форма. Это самый высокий уровень нормализации, который требует, чтобы каждый атрибут отношения зависел от первичного ключа этого отношения и не зависел функционально от любого другого атрибута. Другими словами, отношение находится в DKNF, когда оно уже находится в BCNF и каждый домен атрибутов полностью зависит от первичного ключа. Важно отметить, что шестой уровень нормализации редко используется на практике, так как он может привести к чрезмерной декомпозиции базы данных и усложнить её понимание и использование. Кроме того, многие системы управления базами данных не поддерживают этот уровень нормализации.

1.5. Связи между таблицами в базах данных

В базах данных (БД) связи между таблицами играют ключевую роль в обеспечении целостности данных, а также в упрощении процесса обработки информации. Связь между двумя таблицами представляет собой зависимость между данными, которые хранятся в этих таблицах. Существует несколько типов связей, каждый из которых имеет свои особенности и преимущества:

- **Один к одному**. В этом случае каждой записи в одной таблице соответствует только одна запись в другой таблице. Этот тип связи используется редко, так как он не позволяет эффективно использовать данные.

- **Один ко многим**. Это наиболее распространённый тип связи, который используется для моделирования отношений «один ко многим». Например, один клиент может иметь несколько заказов.
- **Многие ко многим**. Этот тип связи требует использования третьей таблицы, которая будет служить связующим звеном между двумя другими таблицами. Например, многие студенты могут изучать несколько предметов, и каждый предмет может изучаться многими студентами.
- **Самосвязь**. Этот тип связи не требует использования отдельных таблиц для реализации. При таком типе связи внешний ключ определенной таблицы ссылается на запись с этой же таблицы. Например, сущность пользователя хранит в себе внешний ключ сущности другого пригласившего на сайт пользователя,

Стоит отметить, что связи используются только в реалиационных БД. Другие типы БД не поддерживают данный функционал.

1.6. Безопасность баз данных

Безопасность базы данных (БД) — это совокупность мер и стратегий, направленных на защиту данных от несанкционированного доступа, утечек, повреждений и других угроз. В современных условиях обеспечения безопасности информации защита баз данных становится критически важной задачей.

Основные аспекты безопасности базы данных:

- **Аутентификация** — это процесс проверки подлинности пользователя или системы, пытающихся получить доступ к базе данных.
- **Авторизация** — это процесс определения прав и привилегий пользователя после аутентификации.
- **Ролевое управление доступом (RBAC):** Предоставляет доступ к данным на основе ролей, назначенных пользователям.
- **Политики доступа (ACL):** детализируют права доступа на уровне таблиц, строк или столбцов.
- **Шифрование:** защищает данные, делая их нечитаемыми для неавторизованных пользователей. Возможен вариант как шифровки данных, хранящиеся на диске, с использованием алгоритмов AES или RSA, так и шифровки данных, передаваемых между клиентом и сервером, с использованием SSL/TLS.

- **Управление уязвимостями:** регулярное обновление и исправление программного обеспечения базы данных для устранения известных уязвимостей. Важно своевременно устанавливать обновления безопасности.
- **Мониторинг и аудит:** Использование систем обнаружения вторжений (IDS) и ведение журналов аудита для выявления и анализа подозрительной активности.
- **Бэкапы и восстановление данных:** резервное копирование данных помогает восстановить базу данных в случае утраты или повреждения данных.
- **Управление сеансами и время жизни паролей:** контроль сеансов для отключения нежелательных пользователей.
- **Контроль доступа:** ограничение физического и логического доступа к серверам баз данных.

Применение этих мер позволяет значительно уменьшить риски утечки данных и обеспечить защиту информации в базе данных.

1.7 Проектирование схемы базы данных

Проектирование схемы базы данных представляет собой процесс создания структуры базы данных, включающий определение таблиц, столбцов, связей, индексов, ограничений и т.д.

Проектирование схемы базы данных включает следующие шаги:

- **Определение требований к базе данных:** анализ требований, определение целей и задач, которые должна решать база данных. Изучение потребностей пользователей и процессов, которые могут быть автоматизированы с помощью базы данных.

- **Определение сущностей и атрибутов:** создание таблиц (сущностей), создание необходимых по требованиям полей, в которых будет храниться нужная информация.

- **Выбор СУБД:** выбор подходящей СУБД для проектирования на основе выбранного типа базы данных.

- **Определение связей между сущностями:** Определение первичных и внешних ключей для обеспечения целостности и согласованности данных в базе данных.

- **Создание диаграммы (ERD):** визуально отображает сущности и их связи. Это помогает лучше понять структуру данных и связи между ними.

- **Нормализация данных: о**рганизации данных в таблицы для уменьшения избыточности и обеспечения целостности данных

- **Определение первичных и внешних ключей:** Первичные ключи уникально идентифицируют записи в таблице. Внешние ключи устанавливают связи между таблицами и обеспечивают целостность данных. Тем самым, данными методами мы обеспечиваем связи межу таблицами.

- **Создание физической схемы базы данных:** реализация логической схемы в конкретной СУБД. Она включает в себя создание таблиц, индексов, представлений и других объектов базы данных.

При проектировании схемы базы данных необходимо учитывать требования к производительности, безопасности и надежности. Важно обеспечить, чтобы база данных была масштабируемой и могла обрабатывать увеличивающиеся объемы данных без потери производительности.

1.8. Заключение первой главы

В заключение первой главы можно сделать вывод о важности теоретических основ разработки базы данных и её значении для эффективного управления и обслуживания клиентов.
Основные принципы и рекомендации по разработке базы данных включают:

- Анализ по требованиям к функционалу базы данных.
- Выбор типа базы данных
- Выбор инструментов для разработки базы данных по техническим требованиям и личным предпочтениям.
- Определние нормальной формы базы данных
- Проектирование схемы базы данных с учетом требований к производительности, безопасности и надежности.

Эти принципы и рекомендации являются основой для успешной разработки и внедрения базы данных, которая будет эффективно поддерживать деятельность бара "Кооператив" и обеспечивать высокое качество обслуживания клиентов.

2. Разработка базы данных для мессенджера

2.1. Анализ требований

Проектирование базы данных для бара “Кооператив” требует тщательного анализа функциональных и нефункциональных требований системы. В данном параграфе изложены основные требования к структуре базы данных, охватывающие ключевые аспекты хранения данных, обеспечения их целостности и производительности.

Клиенты:

Для управления и учета клиентов необходимо реализовать сущность клиента, которая будет включать данные, необходимые для идентификации и связи с клиентами. Эта сущность также должна быть связана с заказами для учета истории взаимодействий.

Требования, которые я выделил для проектирования данной сущности:

- **Уникальный идентификатор:** требуется для идентификации клиента системой.
- **Контактная информация:** включает имя, телефон и, при наличии, электронную почту для связи с клиентами.
- **Уникальность данных**: номера телефонов и адреса электронной почты должны быть уникальными для упрощения поиска клиентов и предотвращения дублирования.
- **История заказов:** связь с сущностью заказов позволяет отслеживать все взаимодействия клиента с заведением.

Сотрудники:
Сущность сотрудников предназначена для учета персонала и управления доступом к функционалу базы данных.

Требования, которые я выделил для проектирования данной сущности:

- **Уникальный идентификатор:** используется для связи сотрудников с заказами.
- **Роль:** определяет доступ сотрудника к функциям системы (например, администратор, бармен).
- **Данные авторизации:** логин и захешированный пароль необходимы для безопасности системы.
- **Уникальность логина:** логин должен быть уникальным для предотвращения конфликтов при авторизации.

Напитки:
Сущность напитков предназначена для управления ассортиментом и учетом наличия на складе.
Требования, которые я выделил для реализации данной сущности:

- **Уникальный идентификатор:** необходим для идентификации напитков в заказах.
- **Название и цена:** данные для отображения клиентам и расчета итоговой стоимости заказов.
- **Количество:** используется для управления складскими запасами.

Заказы:
Сущность заказов предназначена для учета всех операций, связанных с обслуживанием клиентов.
Требования, которые я выделил для реализации данной сущности:

- **Уникальный идентификатор:** требуется для работы с данными о заказах.
- **Клиенты и сотрудники:** связь с сущностями клиентов и сотрудников для учета, кто сделал и обслужил заказ.
- **Дата создания заказа:** фиксирует момент оформления заказа для анализа и отчетности.

2.2. Типовые запросы

Типовые запросы в SQL — это стандартные команды для выполнения часто используемых операций с базой данных, таких как выборка (SELECT), вставка (INSERT), обновление (UPDATE) и удаление (DELETE) данных. Они облегчают

управление и манипуляцию данными.

Запрос для получения всех заказов клиента:

```
SELECT o.id, o.order_date, d.title
AS drink_title, od.quantity AS
drink_quantity,
       (d.price * od.quantity) AS
total_price
FROM Orders o
JOIN Ordered_Drinks od ON o.id =
od.orders_id
JOIN Drinks d ON od.drinks_id = d.id
JOIN Customers c ON o.customers_id =
c.id
WHERE c.name = 'Иван';  -- Можно
заменить на нужного клиента
```

Запрос для получения списка всех напитков с их количеством на складе:

```
SELECT title, quantity, price
FROM Drinks
WHERE quantity > 0;  -- Показывать
только те напитки, которые есть в наличии
```

Запрос для получения общей суммы заказа по номеру заказа:

```
SELECT o.id, SUM(d.price *
od.quantity) AS total_price
FROM Orders o
JOIN Ordered_Drinks od ON o.id =
od.orders_id
JOIN Drinks d ON od.drinks_id =
d.id
WHERE o.id = 1  -- Замените на
нужный id заказа
GROUP BY o.id;
```

Запрос для получения списка заказов, выполненных барменом:

```
SELECT o.id, o.order_date, c.name AS
client_name, SUM(d.price *
od.quantity) AS total_price
FROM Orders o
JOIN Ordered_Drinks od ON o.id =
od.orders_id
JOIN Drinks d ON od.drinks_id = d.id
JOIN Customers c ON o.customers_id =
c.id
WHERE o.staff_id = (SELECT id FROM
Staff WHERE login = 'barmen')  --
Заменить на логин бармена
GROUP BY o.id, o.order_date,
client_name;
```

Запрос для обновления количества напитков на складе после продажи:

```
START TRANSACTION;
-- Уменьшение количества напитка на складе
UPDATE Drinks
SET quantity = quantity - 1  --
Например, продан 1 напиток
WHERE id = 1;  -- Замените на id нужного
напитка

-- Добавление информации в таблицу заказов
INSERT INTO Ordered_Drinks (orders_id,
drinks_id, quantity)
VALUES (1, 1, 2);  -- Замените на
актуальные данные

COMMIT;
```

2.3. Хранимые процедуры

Хранимая процедура — объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере. У них могут быть входные и выходные параметры и локальные переменные, в них могут производиться числовые вычисления и операции над символьными данными, результаты которых могут присваиваться переменным и параметрам. [1]

В своей базе данных я предусмотрел 1 хранимую процедуру для добавления заказа.

2.4. Триггеры

Триггеры в MySQL представляют собой специальные объекты базы данных, которые автоматически выполняются при определенных событиях или действиях с данными. Они позволяют программистам определить пользовательские действия, которые должны быть выполнены перед или после выполнения операции в базе данных. Иными словами, это определяемая пользователем SQL-команда, которая автоматически вызывается во время операций INSERT, DELETE или UPDATE. [1] В своей базе данных я буду использовать тригеры для обновления количества напитков в таблице напитков.

Напитки:
Вот пример работы моего тригерра:

```
DELIMITER $$

CREATE TRIGGER
UpdateDrinkStockAfterOrder
AFTER INSERT ON Ordered_Drinks
FOR EACH ROW
BEGIN
    -- Обновление количества
напитков в таблице drinks после
вставки в Ordered_Drinks
    UPDATE drinks
    SET quantity = quantity -
NEW.quantity
    WHERE id = NEW.drinks_id;
END $$

DELIMITER ;
```

2.5. Представления

Представления (Views) в MySQL — это виртуальные таблицы, которые основаны на результатах выполнения запроса SELECT. Они представляют собой логические структуры данных, которые могут быть использованы для упрощения и оптимизации запросов к базе данных.[1]

В моей базе данных я реализовал только 1 представление для вывода данных о заказах.

```
CREATE VIEW OrderDetails AS
SELECT
    c.name AS CustomerName,
    o.id AS OrderID,
    o.order_date AS OrderDate,
    d.title AS DrinkTitle,
    od.quantity AS DrinkQuantity,
    d.price AS DrinkPrice,
    (od.quantity * d.price) AS TotalPrice
FROM
    Customers c
JOIN
    Orders o ON c.id = o.customers_id
JOIN
    Ordered_Drinks od ON o.id =
od.orders_id
JOIN
    Drinks d ON od.drinks_id = d.id;
```

Данное представление объединяет данные о клиентах, заказах и напитках.

2.6. Пользовательские функции

Пользовательские функции (User-Defined Functions, UDF) в MySQL — это функции, создаваемые пользователями для выполнения специфических задач, которые не могут быть легко выполнены с помощью встроенных функций MySQL. UDF могут принимать параметры и возвращать скалярные значения (числа, строки, даты и т. д.). Они полезны для создания повторно используемых логических блоков, которые можно вызывать в SQL-запросах.

Для данной базы данных я реализовал одну пользовательскую функцию для вывода общей стоимости заказа с учетом id.

```
DELIMITER $$

CREATE
FUNCTION
GetTotalOrderP
rice(order_id
INT)
RETURNS
DECIMAL(10,2)
DETERMINISTIC
BEGIN
    DECLARE
total_price
DECIMAL(10,2);
```

2.6. Роли

Для обеспечения безопасности в базе данных, необходимо создать роли с определёнными правами и привилегиями.

В моей базе данных я реализовал только одну роль администратора. Данная роль разрешает пользователю просматривать, добавлять и редактировать работников а также работать с напитками и заказами.

[См. приложение 2]

2.7. Выбор инструментов и технологий для разработки базы данных

Для проектикрования, разработки и визуализации базы данных требуется выбрать определенные инструменты, без которых реализация базы данных невозможна, или которые значительно упростят и ускорят разработку.

Система управления базами данных (СУБД)

Так как я выбрал релиационный тип базы данных для своего проекта, я буду использовать СУБД MySQL. MySQL — это одна из самых популярных систем управления реляционными базами данных (СУБД) с открытым исходным кодом. Она была разработана и поддерживается компанией Oracle Corporation. MySQL используется в широком спектре приложений, включая веб-сайты, корпоративные системы и платформы электронной коммерции, благодаря своей надежности, производительности и простоте использования [4]. Именно на данной СУБД будут реализованы все. Ниже приведу отличительные черты MySQL, как положительные, так и отрицательные.

Достоинства MySQL:

- Открытый исходный код
- Высокая производительность
- Масштабируемость
- Широкая поддержка различных платформ
- Широкий спектров поддерживаемых языков программирования
- Поддержка транзакций
- Обширная документация

Недостатки MySQL**:**

- Ограниченная поддержка сложных запросов
- Ограниченная поддержка JSON и NoSQL
- Меньшая поддержка стандартов SQL

Инструмент для моделирования и работы с базой данных

В качестве инструмента для моделирования базы данных, а также для работы с ее данными я выбрал универсальный инструмент MySQL Workbench. MySQL Workbench — это интегрированная среда разработки (IDE) для работы с базами данных MySQL. Она предоставляет инструменты для проектирования схем баз данных, написания и выполнения SQL-запросов, администрирования серверов MySQL и создания отчетов. MySQL Workbench является официальным продуктом Oracle Corporation и доступна для различных операционных систем, включая Windows, macOS и Linux. У данного инструмента есть как и свои достоинства, так и свои недостатки.

Достоинства MySQL Workbench:

- Интуитивно понятный интерфейс
- Возможность проектирования схем данных
- Возможность администрирования серверов MySQL
- Возможность написания и выполнение SQL-запросов

- Возможность миграции данных
- Возможность генерации отчетов
- Поддержка различных операционных систем

Недостатки MySQL Workbench:

- Требовательность к ресурсам
- Проблемы с производительностью
- Ограниченные возможности для работы с большими данными
- Сложность при выполнении сложных операций
- Переодические ошибки и сбои
- Ограниченная поддержка командной строки

MySQL Workbench — это мощный и удобный инструмент для работы с базами данных MySQL, предоставляющий широкий спектр возможностей для проектирования, администрирования и анализа данных. Однако, его использование может быть ограничено из-за требовательности к ресурсам и некоторых проблем с производительностью. Несмотря на эти недостатки, MySQL Workbench остается популярным выбором среди разработчиков и администраторов баз данных благодаря своей функциональности и удобству использования. Именно поэтому мой выбор пал на данный инструмент.

2.8. Реализация базы данных

После проектирования базы данных следует ее реализовать по заданным ранее условиями. Для воссоздания сущностей я использовал ERD-диаграмму, в которой определил все нужные поля с нужными типами, все связи между

сущностями - первичные и вторичные ключи, индексы на идентификаторы каждой сущности. Для создания ERD-диаграммы я использовал, как я писал ранее, IDE MySQL Workbench.

Шаги по реализации ERD-диаграммы через MySQL Workbench:

- Открыть программу MySQLWorkbench.
- Открыть раздел "Моделирование" (Modeling) в MySQL Workbench и выбрать "Новый модуль ER" (New EER Model).
- Создать необходимые по требованиям сущности с требуемыми аттрибутами
- Добавить необходимые связи между сущностями.
- Обозначить индексы и ограничения на необходимые аттрибуты.
- Сгенерировать SQL-скрипт для создания таблиц на основе EDR, используя опцию "Database" -> "Forward Engineer..." в меню.
- Выполнить SQL-скрипт в редакторе MySQL Workbench

Таким образом, все нужные сущности с нужными связями, ограничениями и индексами были успешно созданы.

2.9. Заключение второй главы

Реализация базы данных для бара “Кооператив” с использованием MySQL, моделирования ERD в MySQL Workbench оказалась ключевым этапом в разработке данного проекта. Использование ERD позволило систематизировать и структурировать все необходимые сущности, их атрибуты и связи, что обеспечило логичное и эффективное проектирование базы данных.

Основные шаги включали создание таблиц для посетителей, напитков, заказов, работников и других сущностей, необходимых для функционирования базы данных. Каждая таблица была определена с учетом её роли в системе: хранение информации о клиентах и их заказов.

Генерация SQL скриптов из ERD в MySQL Workbench упростила процесс создания базы данных, что позволило бы быстро перейти к фазе наполнения базы данных тестовыми данными и дальнейшему тестированию функциональности. Это

также способствовало удобству исследования структуры базы данных и её документированию для будущих изменений или оптимизаций.

Таким образом, процесс реализации базы данных для бара “Кооператив” с использованием MySQL, MySQL Workbench и моделирования ERD был успешно завершен, обеспечивая надежное и эффективное хранение данных, необходимых для работы приложения.

Заключение

Разработка базы данных для бара “Кооператив” представляла собой комплексный процесс, требующий глубокого понимания теоретических основ, тщательного анализа требований и правильного выбора инструментов. Использование MySQL в сочетании с MySQL Workbench для моделирования ERD оказалось эффективным подходом, обеспечивающим структурированное хранение и быстрый доступ к данным. Проектирование и реализация базы данных позволили создать надежную основу для функционирования всей логики в базе данных, гарантируя сохранность данных и эффективное выполнение запросов.

Заключение первой главы: Теоретические основы разработки базы данных

В первой главе курсовой работы были рассмотрены основные теоретические аспекты разработки баз данных. Изучены понятие базы данных и её функции, классификация типов баз данных, а также роль нормализации и связей между таблицами для обеспечения целостности данных. Особое внимание уделено вопросам безопасности и проектированию схемы баз данных. Этот теоретический фундамент сыграл важную роль в последующих этапах разработки базы данных для бара.

Заключение второй главы: Разработка базы данных для мессенджера

Во второй главе проведен анализ требований к базе данных бара, что включало определение основных сущностей и их атрибутов. Были выбраны инструменты и технологии для разработки базы данных, включая MySQL и MySQL Workbench для моделирования ERD. Проектирование базы данных включало создание таблиц, определение связей между ними и установку необходимых ограничений. Реализация базы данных была выполнена с учетом всех выявленных требований и анализа, обеспечивая функциональность и производительность системы. Этот этап завершился успешно благодаря систематическому подходу к разработке и тестированию функциональности базы данных.

Таким образом, курсовая работа по разработке базы данных для мессенджера позволила глубже понять и применить теоретические знания в практической среде, а также научилась использовать современные инструменты для проектирования и реализации баз данных. Полученный опыт будет полезен для дальнейшего профессионального развития в области баз данных и информационных технологий.

Список использованной литературы

Электронные ресурсы:

1. Кафедра ИТ – it.vshp.online – [Электронный ресурс] – Режим доступа: https://it.vshp.online
2. Объектно–ориентированная база данных – Wikipedia – [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Объектно–ориентированная_база_данных
3. Функциональные базы данных – Cyclowiki – [Электронный ресурс] – Режим доступа: https://cyclowiki.org/wiki/Функциональные_базы_данных
4. MySQL – Wikipedia – [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/MySQL
5. MySQL Преимущества и недостатки – Servergate – [Электронный ресурс] – Режим доступа: https://servergate.ru/articles/mysql-preimushchestva-i-nedostatki/
6. SQL: что это, в каких базах его используют и как работать с языком программирования – Skillbox Media – [Электронный ресурс] – Режим доступа: https://skillbox.ru/media/code/chto-takoe-sql-kak-ustroen-zachem-nuzhen-i-kak-s-nim-rabotat/
7. База данных – Wikipedia – [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/База_данных
8. Иерархическая база данных - Habr - [Электронный ресурс] - Режим доступа: https://habr.com/ru/articles/771676/
9. Сетевая модель данных - Wikipedia - [Электронный ресурс] - Режим доступа: https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85
10. Реляционная модель данных – Wikipedia – [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Реляционная_модель_данных
11. NoSQL – Wikipedia - [Электронный ресурс] - Режим доступа: https://ru.wikipedia.org/wiki/NoSQL
12. Графовая база данных - Wikipedia - [Электронный ресурс] - Режим доступа: https://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B0%D1%84%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85
13. Resource Description Framework – Wikipedia - [Электронный ресурс] - Режим доступа: https://en.wikipedia.org/wiki/Resource_Description_Framework

14. Система управления базами данных - Wikipedia - [Электронный ресурс] - Режим доступа: https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F_%D0%B1%D0%B0%D0%B7%D0%B0%D0%BC%D0%B8_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85

Приложение 1. Репозиторий GitHub

Ссылка на репозиторий:
https://github.com/Nasvaychik/-oursework

Приложение 2.

```
  -- Создание роли для админа
  CREATE ROLE IF NOT EXISTS `admin`;

  -- Разрешение на просмотр, добавление и редактирования работников.
  GRANT SELECT, INSERT, UPDATE ON `staff` TO `admin`;

  -- Разрешение на просмотр, добавление и удаление заказаных
  напитков.

  GRANT SELECT, INSERT, DELETE ON `ordered_drinks` TO `admin`;

  -- Разрешение на просмотр, добавление и удаление вложений для заказов.
  GRANT SELECT, INSERT, DELETE ON `orders` TO `admin`;

  -- Разрешение на все действия над напитками.
  GRANT ALL PRIVILEGES  ON `drinks` TO `admin`;

  -- Разрешение на просмотр, добавление и удаление клиентов.
  GRANT SELECT, INSERT, DELETE ON `customers` TO `admin`;

  -- Создание пользователя:
  CREATE USER IF NOT EXISTS 'admin'@'localhost' IDENTIFIED BY '1488';

  -- Назначение роли пользователю:
  GRANT `admin` TO 'admin'@'localhost';

  -- Устанавливается роль admin как роль по умолчанию для пользователя
  'admin':
  SET DEFAULT ROLE `admin` TO 'admin'@'localhost';

  Сохранение изменений
FLUSH PRIVILEGES
```

Уважаемый пользователь!
Обращаем ваше внимание, что система Антиплагиус отвечает на вопрос, является тот или иной фрагмент текста заимствованным или нет. Ответ на вопрос, является ли заимствованный фрагмент именно плагиатом, а не законной цитатой, система оставляет на ваше усмотрение.

Отчет о проверке № 9048265

Дата выгрузки: 2024-12-18 23:31:11
Пользователь: 89201805526@mail.ru, ID: 9048265

Отчет предоставлен сервисом «Антиплагиат» на сайте antiplagius.ru/

Информация о документе

№ документа: 9048265
Имя исходного файла: МДК.11.01 - Курсовой проект - Матвеев Кирилл ИСИП-3.docx
Размер файла: 0.12 МБ
Размер текста: 32228
Слов в тексте: 4655
Число предложений: 383

Информация об отчете

Дата: 2024-12-18 23:31:11 - Последний готовый отчет
Оценка оригинальности: 79%
Заимствования: 21%

79.39% 20.61%

Источники:

Доля в тексте	Ссылка
43.20%	https://scienceforum.ru/2018/article/2018004108
41.50%	https://revolution.allbest.ru/programming/01141222_0.html
33.30%	https://workspay.ru/work/132837/
19.10%	https://studfile.net/preview/16998229/
17.90%	https://prepod24.ru/readyworks/344045/
17.40%	https://www.turboreferat.ru/accounting/otchet-po-praktike-v-tc/2...
17.05%	https://appmaster.io/ru/blog/normalizatsiia-v-reliatsionnykh-baz...
15.20%	https://scienceforum.ru/2021/article/2018024159
13.20%	https://it.rfei.ru/course/~gk8r/~tjrlefRQ/~Mv0JyjU5
12.80%	https://www.yaneuch.ru/cat_22/kontrolnaya-rabota-po-informatike/...
12.30%	https://gb.ru/blog/chto-takoe-baza-dannyh/
12.10%	https://intuit.ru/studies/courses/3687/929/lecture/19324?page=4
11.60%	https://appmaster.io/ru/blog/modelirovanie-dannykh-v-rdbms
11.20%	https://na-journal.ru/1-2024-informacionnye-tekhnologii/8437-pos...
10.10%	https://www.decosystems.ru/normalizatsiya-bazy-dannykh-sql-2/
10.10%	https://sky.pro/wiki/sql/rabota-s-bazami-dannyh-osnovnye-zadachi...

Доля в тексте	Ссылка
9.30%	https://habr.com/ru/companies/otus/articles/755440/
9.20%	https://habr.com/ru/articles/254773/
8.90%	https://newsletter.radensa.ru/archives/2411
8.40%	https://nsportal.ru/npo-spo/informatika-i-vychislitelnaya-tekhni...
7.70%	https://selectel.ru/blog/tutorials/mysql-workbench-installation/

Информация о документе:

Частноеучреждениепрофессиональногообразования"Высшаяшколапредпринимательства"(ЧУПО"ВШП")КУСОВОЙПОЕКТ "База данных для бара "Кооператив"" Выполнил: студент 3-го курса специальности 09.02.07 "Информационные системы и программирование"МатвеевКириллДмитриевичподпись:Проверил:преподавательдисциплины,преподавательЧУПО"ВШП", к.ф.н.ТкачевП.С.оценка:подпись:СодержаниеТверь,2024г.Введение3Цельработы3Задачи31.Теоритическиеосновы разработкибазыданных41.1.Понятиебазыданныхиеёосновныефункции41.2.Типыбазданных41.3.Инструментыдля разработкибазыданных61.4.Нормализациябазданных71.5.Связимеждутаблицамивбазахданных81.6.Безопасностьбаз данных 9 1.7 Проектирование схемы базы данных 10 1.8. Заключение первой главы 11 2. азработка базы данных для мессенджера122.1.Анализтребований12Клиенты:12Сотрудники:13Напитки:13Заказы:132.2.Типовыезапросы132.3. Хранимыепроцедуры152.4.Триггеры17Напитки:172.5.Представления172.6.Пользовательскиефункции182.6.оли19 Системауправлениябазамиданных(СУБД)19Инструментдлямоделированияиработысбазойданных202.8.ealизациябазы данных21ШагипореализацииERD-диаграммычерезMySQLWorkbench:222.9.Заключениевторойглавы22Заключение24 Заключение первой главы: Теоретические основы разработки базы данных 24 Заключение второй главы: азработка базы данных для мессенджера 24 Список использованной литературы 25 Электронные ресурсы: 25 Приложение 1. епозиторий GitHub27Ссылканарепозиторий:27Приложение2.28ВведениеАктуальностьтемыБазыданныхиграютважнуюрольв современноммиретехнологий,таккаконипозволяютсохранять,организовыватьиобрабатыватьразличнуюинформацию. Сегодняпрактическикаждаяорганизацияиспользуетбазыданныхдляуправлениянными.Этоактуальноидлябаров,где информацияоклиентах,заказах,поставкахисотрудникаххранитсяиобрабатываетсяспомощьюспециализированныхсистем. аботасбазамиданныхстановитсяважнымнавыкомдляуспешноговедениябизнеса,аспециалистывэтойобласти-всёболее востребованными.ЦельработыЦельданнойработызаключаетсяврazработкеиреализацииреляционнойба-зыданныхдля бара"Кооператив",обеспечивающейэффективноеуправлениеданнымиоклиентах,напитках,сотрудникахизаказах.Задачи Длядостиженияzаданноймнецели,мненужновыполнитьследующиezадачи:*Обозначитьтребованиюк функционалубазы данных * Выбрать инструменты и технологии для разработки * Спроектировать базу данных на основе требований к функционалу * еализовать базу данных на основе спроектированной схемы * Определить требования к базе данных, обеспечивающие целостность системы 1. Теоритические основы разработки базы данных 1.1. Понятие базы данных и её основныефункции Базаданных-этоорганизованнаяструктура,котораяпредназначенадляхраненияинформации.Втовремя, когдапроисходилоразвитиетерминабазданных,внихсохранялисьисключительноинформация,однакоужевнашидни многиесистемыуправлениябазамиданныхпозволяетразмещатьвсвоихструктурахиданные,ипрограммныйкод,спомощью которого совершается связь с пользователями или с другими программно-аппаратными комплексами. При этом данные не должны противоречить друг другу, быть целостными и не избыточными. База данных создается для сохранения и непосредственного доступа к информации, содержащей сведения об искомой предметной области. Прежде всего, целью использованияинформацииизбазданныхисложностьюинформационныхпроцессов,существующихвпределахпредметной областив конкретныхусловиях.Основными функциями базыданныхявляются:*Хранениеданных:БД обеспечиваетбезопасное и надёжноехранениеинформации.*Обработка данных:с помощьюзапросови операцийможновыполнять различныедействия сданными,такиекакпоиск,сортировка,фильтрация и т.д.*Анализданных:базыданныхпозволяютпроводитьанализданных, выявлятьзакономерностиитенденции,атакжеприниматьобоснованные решениянаосновеполученнойинформации.Система управлениябазамиданных-этопрограммныймеханизм,предназначенный длязаписи, поиска,сортировки,обработки ипечати информации,содержащейсяв базеданных.Иными словами, системауправлениябазамиданных -механизм, регулирующий работусамой базыданных.[14]1.2.Типыбазданных Существуетмножествотипов базданных, каждыйиз которыхимеетсвои особенностиипреимуществавзависимостиотпоставленнойзадачиитребованийпроекта.Вданнойработебудутрассмотрены основныетипыбазданных:* еляционныебазыданных(БД)-этонаиболеераспространённыйтипбазданных.Они основаны нареляционноймоделиданныхииспользуюттаблицыдляхраненияинформации.еляционныебазыданныхимеютмножество преимуществ, таких как простота использования, надёжность и эффективность. Однако они также имеют некоторые недостатки,такие каксложность масштабированияи ограниченияв производительностиприработесбольшими объёмами данных. [10]*Объектно-ориентированныебазы данных(ООБД)-этот тип базданных основаннаобъектно-ориентированной моделиданных.ВООБДданныепредставленыввидеобъектов,анетаблиц.Этопозволяетболееточномоделироватьсложные предметныеобластииобеспечиваетболеевысокуюгибкостьприразработкеприложений.ОднакоООБДимеютболеесложную

структуруитребуютболеевысокой квалификациираз работчиков.[2]*Иерархические базыданных(ИБД)- в иерархических базахданныхинформацияорганизованаввидедревовидной структуры.ИБДиспользуютсядлямоделированияиерархических отношений между объектами, например, в системах управления проектами или в каталогах товаров. Иерархические базы данныхпростывиспользованииобеспечиваютхорошуюпроизводительностьприработе снебольшимиобъёмамиданных,но онинеподходят дляс ложныхзапросов и обработкибольших объёмовданных.[8]* Сетевые базыданных(СБД) - сетевыебазы данныхпохожинаиерархические,новнихдопускаетсяналичиенесколькихродительскихузловуодногодочернегоузла.СБД используютсядлямоделированиясложныхотношениймеждуобъектами иобеспечивают большуюгибкостьпо сравнению с иерархическимибазамиданных.ОднакоСБДсложнее вреализацииимеютболеенизкуюпроизводительностьпосравнениюс реляционнымибазамиданных.[9]*NoSQLбазыданных-NoSQLбазыданныхпредставляютсобойкласснереляционныхбаз данных,которыйвключаетвсебяширокийспектртехнологий,разработанныхдляобеспечениямасштабируемостиигибкости. NoSQLбазыданныхнеиспользуютсхемыимогутобрабатыватьразличныетипыданных,включая документы, пары ключ-значениеиграфы.Они частоиспользуются для обработкибольшихобъёмовнеструктурированныхданных,такихкакданные социальныхсетей,журналоввеб-сервераит.д.[11]*Графовыебазыданных-графовыебазы данныхпредназначеныдля работы с данными, представленными в виде графа. Графовые базы данных позволяют эффективно моделировать и обрабатывать отношения между объектами. Они используются в таких областях, как социальные сети, рекомендательные системы и анализ данных.[12] * RDF (Resource Description Framework) - это стандарт представления данных, который используется для описания ресурсов в интернете. RDF позволяет описывать ресурсы с помощью троек "субъект-предикат- объект",где субъектиобъектявляютсяресурсами,апредикатописываетотношениемеждуними.RDFявляетсяосновойдля создания семантических веб- приложений, которые используют машинное понимание данных для предоставления более интеллектуальных услуг.[13] 1.3. Инструменты для разработки базы данных азработка баз данных включает в себя проектирование, создание, управление и оптимизацию баз данных. Для этих целей существует множество инструментов, каждый из которых предлагает определенные функции и возможности. В этом разделе будут рассмотрены основные инструментыдляразработки базданных.Инструменты дляразработкибазданныхделятся наследующиекатегории:*Системы управлениябазамиданных(СУБД)- этопрограммное обеспечение,предназначенноедлясоздания, управленияи модификации баз данных. Например, MySQL - распространенная реалиационная СУБД с открытым исходным кодом. * Инструменты моделированияданных:помогаютразработчикамсоздавать ивизуализироватьструктурыбазданных.Например,Microsoft Visio - универсальныйинструментдлясозданиядиаграмм, включаядиаграммысущность-связь (ERD).*Инструментыуправления базамиданных: предоставляютинтерфейсыдляадминистрированияиуправлениябазамиданных.Например, MySQL Workbench - инструмент для управления СУБД MySQL. * Инструменты разработки и интеграции: облегчают процесс написания, тестирования иотладкиSQL-кода иприложений,работающих сбазамиданных.Например, DBeaver-многофункциональный инструментдляработы сразличными СУБД,поддерживающиймножество форматовданных.*Инструменты мониторингаи оптимизации:эти инструментыпомогаютотслеживатьпроизводительностьбазданных иоптимизировать запросы. Например, Oracle AWR- инструментдля анализапроизводительностибазданныхOracle.Дляразработкиследуетвлучшемслучаевыбрать инструмент из каждой категории, изходя из используемой СУБД и из личных предпочтений. Использование данных инструментовоптимизируетработуразработчикаисделаетеенамногопрощеикачественнее.1.4.Нормализациябазданных Нормализация-этопроцессорганизацииструктурыбазы данных таким образом,чтобы минимизироватьизбыточностьданных и обеспечить целостность информации. Нормализация позволяет улучшить производительность базы данных, упростить её обслуживание и обеспечить более эффективное использование ресурсов. Существует несколько уровней нормализации, каждый из которых устраняет определённые виды избыточности. Чем выше уровень нормализации, тем больше таблиц требуетсядляпредставленияданных,неприэтом повышаетсяцелостность данныхиуменьшаетсявероятность возникновения аномалий обновления. Уровни нормализации существуют следующие: * Первый уровень нормализации, или 1НФ, является базовым уровнем нормализации реляционных баз данных. Он требует, чтобы все атрибуты были атомарными, то есть не содержаливсебедругихзначений.Этоозначает,чтокаждыйатрибутдолженпредставлять собойоднозначение,а не набор значенийили список.*Второйуровеньнормализации,или2НФ,являетсяследующимшагомпоследостижения1НФ.Онтребует выполнения 1НФ и отсутствия частичной зависимости неключевых атрибутов от первичного ключа. Частичная зависимость возникает, когда неключевой атрибут зависит только от части первичного ключа. Для устранения частичной зависимости необходиморазбитьтаблицунадвеилиболеетаблиц.*Третийуровеньнормализации,или 3НФ, являетсясамымстрогим уровнем нормализации реляционных баз данных. Он требует выполнения 2НФ и отсутствия транзитивной зависимости неключевых атрибутов от первичного ключа. Транзитивная зависимость возникает, когда неключевой атрибут зависит от другогонеключевого атрибута,который,всвою очередь,зависитотпервичногоключа.*НормальнаяформаБойса-Кодда (BCNF) -□для□ любой□ функциональной□ зависимостиX□□→□ A в□□ отношении R□□ атрибут□ A должен□ □ функционально□ зависеть□ от□ каждого□ элемента в X(тоестьXдолжен быть суперключомотношенияR).*Четвёртый уровень(4NF) -отношение находится в 4NF,еслионо находитсявНФБКивсенетривиальныемногозначныезависимостифактическиявляются функциональными зависимостямиот её потенциальныхключей.*Пятыйуровеньилинормальнаяформапроекции-соединения(5NFилиPJ/NF)-любаязависимость соединенияв ней определяетсяпотенциальными ключами отношения(ключамиисходногоотношения).Также отношениеR соответствует5NFтогда итолькотогда,когдакаждаянетривиальнаязависимостьсоединениявRследуетизсуществования некоторогопотенциальногоключаKдляR.*Шестойуровеньнормализации(6NF)-такжеизвестенкакDomain/KeyNormalForm (DKNF)илиидеальнаянормальнаяформа.Этосамыйвысокий уровеньнормализации,который требует, чтобыкаждыйатрибут отношения зависелотпервичногоключа этогоотношенияинезависел функциональноотлюбогодругого атрибута.Другими словами, отношение находится в DKNF,когдаоно уже находится вBCNFикаждыйдоменатрибутов полностьюзависитот первичногоключа.Важноотметить,чтошестойуровеньнормализацииредкоиспользуетсянапрактике,таккаконможет привестикчрезмернойдекомпозициибазыданныхиусложнитьеёпониманиеииспользование.Крометого,многиесистемы управлениябазамиданныхнеподдерживаютэтотуровеньнормализации.1.5.Связимеждутаблицамивбазахданных Вбазах данных(БД)связимеждутаблицамииграютключевуюрольвобеспеченииицелостностиданных,атакжевупрощенииипроцесса обработкиинформации.Связьмеждудвумятаблицамипредставляетсобойзависимостьмеждуданными,которыехранятсяв этихтаблицах.Существуетнесколькотиповсвязей,каждыйизкоторыхимеетсвоиособенностиипреимущества:*Одинк

одному.Вэтомслучаекаждойзаписиводнойтаблицесоответствуеттолькоодназаписьвдругойтаблице.Этоттипсвязи используется редко, так как он не позволяет эффективно использовать данные. * Один ко многим. Это наиболее распространённыйтипсвязи,которыйиспользуетсядлямоделированияотношений"одинкомногим".Например,одинклиент можетиметьнесколькозаказов.*Многиекомногим.Этоттипсвязитребуетиспользованиятретьейтаблицы,котораябудет служитьсвязующимзвеномеждудвумядругимитаблицами.Например,многиестудентымогутизучатьнесколькопредметов, икаждыйпредметможетизучатьсямногимистудентами.*Самосвязь.Этоттипсвязинетребуетиспользованияотдельных таблицдляреализации.Притакомтипесвязивнешнийключопределеннойтаблицыссылаетсяназаписьсэтойжетаблицы. Например,сущностьпользователяхранитвсебевнешнийключсущностидругогопригласившегонасайтпользователя,Стоит отметить,чтосвязииспользуютсятольковреалиационныхБД.ДругиетипыБДнеподдерживаютданныйфункционал.1.6. БезопасностьбазданныхБезопасностьбазыданных(БД)-этосовокупностьмеристратегий,направленныхназащитуданных отнесанкционированногодоступа,утечек,повреждениийидругихугроз.Всовременныхусловияхобеспечениябезопасности информации защита баз данных становится критически важной задачей. Основные аспекты безопасности базы данных: * Аутентификация-этопроцесспроверкиподлинностипользователяилисистемы,пытающихсяполучитьдоступкбазеданных.* Авторизация - это процесс определения прав и привилегий пользователя после аутентификации. * олевое управление доступом(RBAC):Предоставляетдоступкданнымнаосноверолей,назначенныхпользователям.*Политикидоступа(ACL): детализируютправадоступанауровнетаблиц,строкилистолбцов.*Шифрование:защищаетданные,делаяихнечитаемыми для неавторизованных пользователей. Возможен вариант как шифровки данных, хранящиеся на диске, с использованием алгоритмовAESилиRSA,таки шифровки данных, передаваемыхмеждуклиентоми сервером,сиспользованиемSSL/TLS.* Управление уязвимостями: регулярное обновление и исправление программного обеспечения базы данных для устранения известных уязвимостей.Важносвоевременноустанавливатьобновлениябезопасности.*Мониторингиаудит:Использование системобнаруженияторжений(IDS)иведениежурналоваудитадлявыявленияианализаподозрительнойактивности.* Бэкапыивосстановлениеданных:резервноекопированиеданныхпомогаетвосстановитьбазуданныхвслучаеутратыили повреждения данных. * Управление сеансами и время жизни паролей: контроль сеансов для отключения нежелательных пользователей.*Контрольдоступа:ограничениефизическогоилогическогодоступаксерверамбазданных.Применениеэтих мер позволяет значительно уменьшить риски утечки данных и обеспечить защиту информации в базе данных. 1.7 ПроектированиесхемыбазыданныхПроектированиесхемыбазыданныхпредставляетсобойпроцесссозданияструктурыбазы данных,включающийопределениетаблиц,столбцов,связей,индексов,ограничений.д.Проектированиесхемыбазыданных включаетследующиешаги:*Определениетребованийкбазеданных:анализтребований,определениецелейизадач,которые должнарешатьбазаданных.Изучениепотребностейпользователейипроцессов,которыемогутбытьавтоматизированыс помощью базы данных. * Определение сущностей и атрибутов: создание таблиц (сущностей), создание необходимых по требованиям полей, в которых будет храниться нужная информация. * Выбор СУБД: выбор подходящей СУБД для проектированиянаосновевыбранноготипабазыданных.*Определениесвязеймеждусущностями:Определениепервичныхи внешних ключей для обеспечения целостности и согласованности данных в базе данных. * Создание диаграммы (ERD): визуально отображаетсущностии ихсвязи.Этопомогаетлучше понятьструктуруданных исвязимежду ними.*Нормализация данных:организацииданныхвтаблицыдляуменьшенияизбыточностиобеспеченияцелостностиданных*Определение первичныхивнешнихключей:Первичныеключиуникальноидентифицируютзаписивтаблице.Внешниеключиустанавливают связимеждутаблицами иобеспечиваютцелостностьданных. Темсамым,даннымиметодамимыобеспечиваемсвязимежу таблицами.*Созданиефизической схемыбазы данных:реализациялогической схемывконкретнойСУБД. Онавключаетв себя создание таблиц, индексов, представлений и других объектов базы данных. При проектировании схемы базы данных необходимоучитывать требованиякпроизводительности, безопасностиинадежности.Важнообеспечить,чтобыбазаданных была масштабируемой и могла обрабатывать увеличивающиеся объемы данных без потери производительности. 1.8. ЗаключениепервойглавыВзаключениепервойглавыможносделатьвыводо важноститеоретическихоснов разработкибазы данных и её значении для эффективного управления и обслуживания клиентов. Основные принципы и рекомендации по разработкебазыданныхвключают:*Анализпотребованиямкфункционалубазыданных.*Выбортипабазыданных*Выбор инструментовдляразработкибазыданныхпотехническимтребованиямиличнымпредпочтениям.*Определниенормальной формы базы данных * Проектирование схемы базы данных с учетом требований к производительности, безопасности и надежности.Этипринципыирекомендацииявляютсяосновойдляуспешнойразработкиивнедрениябазыданных,которая будетэффективноподдерживатьдеятельностьбара"Кооператив"иобеспечиватьвысокое качествообслуживанияклиентов.2. азработка базыданныхдля мессенджера2.1.Анализтребований Проектированиебазыданныхдлябара "Кооператив"требует тщательногоанализа функциональных инефункциональныхтребованийсистемы.Вданномпараграфеизложеныосновные требования к структуре базы данных, охватывающие ключевые аспекты хранения данных, обеспечения их целостности и производительности.Клиенты:Дляуправленияиучетаклиентовнеобходимореализоватьсущностьклиента,котораябудет включатьданные,необходимыедляидентификациииисвязисклиентами.Этасущностьтакжедолжнабытьсвязанасзаказами для учета истории взаимодействий. Требования, которые я выделил для проектирования данной сущности: * Уникальный идентификатор:требуетсядляидентификацииклиентасистемой.*Контактнаяинформация:включаетимя,телефони,при наличии,электроннуюпочтудлясвязисклиентами.*Уникальностьданных:номертелефоновиадресаэлектроннойпочты

должны быть уникальными для упрощения поиска клиентов и предотвращения дублирования. * История заказов: связь с сущностью заказов позволяет отслеживать все взаимодействия клиента с заведением. Сотрудники: Сущность сотрудников предназначена для учета персонала и упра