**РОССИЙСКАЯ ФЕДЕРАЦИЯ**

**ХАНТЫ-МАНСИЙСКИЙ АВТОНОМНЫЙ ОКРУГ-ЮГРА**



**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ**

**«СУРГУТСКИЙ ИНСТИТУТ ЭКОНОМИКИ, УПРАВЛЕНИЯ И ПРАВА»**

КУРСОВАЯ РАБОТА

по дисциплине «Технология разработки и защиты баз данных»

на тему «Разработка модели реляционной базы данных, для автомобильного гаража»

Выполнил: Таращук Максим Павлович

студент(ка) группы ИС 23/11

Проверил: Преподаватель профессиональных дисциплин

Колмыков М.В.\_\_\_\_\_\_\_\_\_\_\_

Сургут, 2024

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 2 |
| ГЛАВА I ТЕОРЕТИЧЕСКАЯ ЧАСТЬ: | 3 |
| 1.1 Основные положения теории баз данных, хранилищ данных, баз знаний | 3 |
| 1.1.1 Типы моделей данных: иерархическая, сетевая, реляционная и объектно ‒ ориентированная модели данных | 3 |
| 1.1.2 Свойства базы данных | 4 |
| 1.1.3 Основные принципы построения хранилищ данных и баз знаний | 6 |
| 1.2 Анализ предметной области проектирования | 7 |
| 1.2.1 Выбор приложения на чем будет реализована база дынных | 8 |
| 1.2.2 Описание основных сущностей и атрибутов, связей между ними | 10 |
| 1.2.3 Выделение ключевых сущностей, их атрибутов и связей | 11 |
| ГЛАВА II. ПРАКТИЧЕСКАЯ ЧАСТЬ: | 13 |
| 2.1 Определение сущностей логической модели БД | 13 |
| 2.1.1 Построение логической БД | 18 |
| 2.2 Построение физической модели БД | 18 |
| 2.2.1 Запросы для БД | 19 |
| 3 ЗАКЛЮЧЕНИЕ | 24 |
| 4 СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ | 25 |

ВВЕДЕНИЕ

Связь между парикмахерской и реляционной базой данных может быть весьма полезной для эффективного управления информацией о клиентах, услугах, расписании работы мастеров и других аспектах работы парикмахерской. Для повышения эффективности сервиса и улучшения опыта клиентов, используется реляционная база данных. Это помогает хранить информацию о клиентах, их предпочтениях, истории посещений, а также планировать расписание работы мастеров.

Данная курсовая включает в себя: основы теории баз данных; основные понятия и определения; модели данных: иерархическая, сетевая и реляционная; дальнейшее развитие способов организации данных; пост реляционные модели данных; атрибуты и ключи; нормализация отношений; проектирование баз данных; основные принципы проектирования; описание баз данных; логическая и физическая структура баз данных.

Цель этой курсовой работы состоит в разработке и реализации базы данных для парикмахерской, которая будет эффективно управлять всеми аспектами ее деятельности. Она будет включать в себя структуру данных для хранения информации о клиентах, работниках, услугах, расписании работы, финансах и других важных аспектах работы парикмахерской. В ходе этой работы мы будем изучать различные аспекты проектирования баз данных, такие как определение сущностей и их атрибутов, установление связей между таблицами, разработка оптимальной структуры таблиц и создание SQL запросов для эффективного взаимодействия с данными.

ГЛАВА I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Основные положения теории баз данных, хранилищ данных, баз знаний.

База данных ‒ это особым образом организованные и хранимые в электронном виде данные, а так же это совокупность взаимосвязанных хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их эффективное использование для одного или нескольких приложений (задач).

Цель создания баз данных ‒ построение такой системы данных, которая бы не зависела от программного обеспечения, применяемых технических средств и физического расположения данных. При проектировании базы данных предполагается многоцелевое её использование.

База данных в простейшем случае представляется в виде системы двумерных таблиц.

Схема данных ‒ это описание логической структуры данных, специфицированное на языке описания данных и обрабатываемое системой управления базами данных (СУБД).

Схема пользователя ‒ это зафиксированный для конкретного пользователя один вариант порядка полей таблицы.

Система управления базами данных (СУБД) ‒ это программное обеспечение, контролирующее организацию, хранение, целостность, внесение изменений, чтение и безопасность информации в базе данных.

Информационная системапредставляет собой систему, реализующую автоматизированный сбор, обработку и манипулирование данными и включающая технические средства обработки данных, программное обеспечение и обслуживающий персонал.

1.1.1 Типы моделей данных: иерархическая, сетевая, реляционная и объектно‒ориентированная модели данных

По способу хранения данных и организации связей между ними базы данных делятся на несколько видов:

‒ иерархическая;

‒ сетевая;

‒ реляционная;

‒ объектно ‒ ориентированная;

‒ документо ‒ ориентированная;

‒ графовая.

Иерархические данные объединены в древовидную структуру, в корневом элементе которой есть ссылки на узлы, расположенные на втором уровне иерархии.

Сетевая по способу записи элементов похожа на иерархическую, но связи можно строить между любыми узлами графа, а не только «сверху‒вниз».

Реляционные данные организуются в табличной форме, где каждая запись отдельная строка, имеющая уникальный признак, называемый ключом.

Объектно ‒ ориентированные сущности хранятся в виде совокупности объектов.

Гибридная. Попытка объединить реляционную модель и объектно‒ориентированный подход.

Документо ‒ ориентированная модель данных организована в виде документов, которые могут быть структурированными или неструктурированными. Каждый документ содержит данные в формате, таком как JSON или XML, и может иметь вложенные структуры.

Графовая модель данные представлены в виде графа, где узлы представляют объекты, а ребра ‒ их отношения. Эта модель хорошо подходит для моделирования связей и зависимостей между данными, например, в социальных сетях или сетях дорог.

Каждая из этих моделей имеет свои преимущества и недостатки, и выбор модели зависит от конкретных требований приложения и характера данных, которые необходимо хранить и обрабатывать.

1.1.2 Свойства базы данных

Основные свойства баз данных:

‒ структурированность,

‒ постоянство;

‒ доступность;

‒ целостность;

‒ безопасность;

‒ масштабируемость;

‒ производительность;

‒ распределенность;

‒ резервирование.

Структурированные данные в базе данных организованы в определенной структуре, которая позволяет эффективно хранить и управлять ими. Это может быть таблицы, схемы, графы и т.д.

Постоянство данных в базе данных сохраняются долгосрочно и остаются доступными после завершения текущей сессии или работы с приложением.

Доступность, база данных обеспечивает доступ к данным для пользователей и приложений, в соответствии с установленными правами доступа. Это позволяет пользователям получать нужную информацию в нужное время.

Целостность, база данных должна гарантировать целостность данных, то есть сохранять их точность и согласованность во всей системе, даже при одновременном доступе нескольких пользователей.

Безопасность, база данных обеспечивает защиту данных от несанкционированного доступа, изменений или утраты. Это включает в себя механизмы аутентификации, авторизации и шифрования.

Масштабируемость, БД должна быть способна масштабироваться, чтобы обрабатывать как малые, так и большие объемы данных, а также увеличивать производительность при необходимости.

Производительность, база данных должна обеспечивать быстрый доступ к данным и эффективную обработку запросов, чтобы удовлетворять требованиям пользователей по времени ответа.

Распределенность, современные базы данных могут быть распределенными, то есть данные могут храниться и обрабатываться на разных серверах или узлах, что обеспечивает отказоустойчивость и масштабируемость.

Резервирование и восстановление, база данных должна обладать механизмами резервного копирования и восстановления данных, чтобы предотвращать потерю информации в случае сбоев или аварий.

1.1.3 Основные принципы построения хранилищ данных и баз знаний

Основные принципы построения баз данных основаны на научных принципах, которые позволяют создавать высококачественные системы, соответствующие современным требованиям.

Из множества принципов создания баз данных, наиболее важными являются интеграция данных и централизация управления данными.

Оба принципа отражают суть баз данных. Интеграция данных является основой организации баз данных.

Централизация управления ‒ основой организации и функционирования систем управления базами данных (СУБД). Остальные принципы связаны с этими основными принципами.

Принцип интеграции данных заключается в объединении отдельных, несвязанных данных в единую базу данных, которая представляет собой единый информационный массив для пользователя и его прикладных программ. Следование этому принципу обеспечивает упрощение поиска и обработки взаимосвязанных данных, уменьшение избыточности данных и упрощение процесса ведения баз данных.

Интеграция данных способствует целостности и согласованности информации, повышая ее надежность и достоверность. Централизация управления данными обеспечивает единый контроль и администрирование всей базы данных, упрощая управление доступом, резервное копирование и восстановление.

Принципы построения баз данных, такие как интеграция данных и централизация управления данными, обеспечивают основу для создания высококачественных систем, которые соответствует современными требованиям к надежности, эффективности и масштабируемости.

1.2 Анализ предметной области проектирования

Анализ предметной области проектирования для парикмахерской включает в себя изучение основных аспектов деятельности парикмахерской, потребностей клиентов и требований к функциональности программного обеспечения. Вот ключевые моменты, которые следует учитывать:

‒ услуги парикмахерской;

‒ расписание и запись;

‒ инвентарь и расходные материалы;

‒ учет услуг и финансов;

‒ клиентская база и управление клиентами;

‒ управление персоналом;

‒ безопасность и гигиена.

Парикмахерская может предоставлять различные услуги, такие как стрижка, укладка, окрашивание, маникюр, педикюр и другие. Важно проанализировать типы услуг, их частоту предоставления и стоимость.

Парикмахерская обычно работает по предварительной записи. Анализ расписания работы персонала и возможностей онлайн‒записи может помочь оптимизировать процесс управления клиентами и улучшить обслуживание.

Для предоставления услуг парикмахерам часто требуется использование различных инструментов, оборудования и расходных материалов. Важно проанализировать инвентарь и его управление, чтобы избежать нехватки или избытка материалов.

Система учета услуг и финансов позволяет отслеживать доходы, расходы, прибыль и другие финансовые показатели парикмахерской. Это важный аспект для эффективного управления бизнесом.

Ведение базы данных клиентов помогает отслеживать предпочтения, историю посещений, контактные данные и другую важную информацию. Это позволяет персонализировать обслуживание и улучшить взаимодействие с клиентами.

Для эффективной работы парикмахерской необходимо оптимальное управление персоналом, включая назначение рабочих графиков, контроль за производительностью и мотивацию сотрудников.

Парикмахерская должна соблюдать нормы безопасности и гигиены, чтобы обеспечить безопасность как для персонала, так и для клиентов. Это включает в себя соблюдение правил по стерилизации инструментов, использование защитных средств и т.д.

Анализ этих аспектов поможет определить требования к программному обеспечению для парикмахерской и разработать систему, которая будет эффективно поддерживать операции бизнеса.

1.2.1 Создание базы данных на SQLite.

SQLite компактная и самодостаточная реляционная СУБД (система управления базами данных), которая является частью стандартной библиотеки в большинстве языков программирования, включая C, C++, Python, PHP и многие другие. Она отличается от других СУБД, таких как MySQL или PostgreSQL, тем, что она не является серверной базой данных, а вместо этого представляет собой библиотеку, которая непосредственно встраивается в приложение.

Основные особенности SQLite:

‒ легковесность;

‒ надежность;

‒ кросс‒платформенность;

‒ поддержка большого объема данных.

Легковесность SQLite очень компактна и не требует установки отдельного сервера баз данных. Вся база данных хранится в одном файле, что делает ее удобной для интеграции в различные приложения.

Надежность обеспечивает надежное хранение данных с поддержкой транзакций и ACID‒свойств (атомарность, согласованность, изоляция, долговечность), что обеспечивает целостность данных.

Кросс‒платформенность поддерживается на различных операционных системах, включая Windows, macOS, Linux и другие.

Поддержка большого объема данных SQLite может управлять базами данных размером до нескольких терабайт, что делает его пригодным для широкого спектра задач.

Преимущества SQLite:

‒ легкость;

‒ независимость;

‒ эффективность;

‒ низкие требования к ресурсам;

‒ богатый набор функций;

‒ бесплатность и открытость.

Проста в интеграции в приложения благодаря своей компактности и отсутствию необходимости установки отдельного сервера баз данных. Его можно использовать в приложениях для хранения и управления данными без значительных затрат на конфигурацию и обслуживание.

Независимость от платформы SQLite, поддерживается на различных операционных системах, включая Windows, macOS, Linux, iOS и Android. Это делает его удобным выбором для кросс‒платформенных приложений.

Эффективность, обеспечивает высокую производительность и эффективное использование ресурсов, даже при работе с большими объемами данных. Он хорошо оптимизирован для быстрого доступа к данным и выполнения SQL‒запросов.

Низкие требования к ресурсам не требует больших объемов оперативной памяти или вычислительных мощностей для работы. Это делает его подходящим для использования на устройствах с ограниченными ресурсами, таких как мобильные устройства или встроенные системы.

Богатый набор функций SQLite поддерживает множество функций SQL, включая агрегатные функции, подзапросы, триггеры, представления и многое другое, что делает его мощным инструментом для работы с данными.

Бесплатность и открытость является бесплатным и открытым программным обеспечением, распространяемым под лицензией общественного домена (public domain), что позволяет свободно использовать его в любых проектах без ограничений.

1.2.2 Описание основных сущностей и атрибутов, связей между ними

Основные сущности, которые могут быть присутствовать в базе данных парикмахерской, могут включать в себя:

‒ клиенты;

‒ услуги;

‒ персонал;

‒ расписание;

‒ операции;

‒ инвентарь.

В таблице клиенты можно хранить информацию о клиентах парикмахерской, такую как их имя, контактные данные, предпочтения в отношении услуг, историю посещений и т.д.

Услуги, эта таблица может содержать информацию о различных услугах, предоставляемых парикмахерской, и их стоимость.

Персонал, в этой таблице можно хранить информацию о персонале парикмахерской, включая их имена, специализацию, контактные данные и график работы.

Расписание, здесь можно хранить информацию о расписании работы сотрудников и записях клиентов на определенные даты и время.

Операции, эта таблица может использоваться для отслеживания финансовых операций, таких как оплата услуг, расходы на инвентарь и другие расходы.

Инвентарь, здесь можно хранить информацию о инвентаре и расходных материалах парикмахерской, таких как шампуни, красители, расходные материалы и т.д.

Связи между сущностями могут быть следующими:

‒ каждый клиент может иметь несколько записей на услуги;

‒ каждый парикмахер может иметь много записей на услуги, в которых он участвует;

‒ услуги могут быть назначены на несколько записей клиентов.

Такая база данных позволит эффективно управлять информацией о клиентах, услугах и записях, а также облегчит планирование графика работы парикмахеров.

1.2.3 Выделение ключевых сущностей, их атрибутов и связей

Для выделения ключевых сущностей, их атрибутов и связей в базе данных используется метод семантического моделирования.

Реляционные модели данных подходят для моделирования предметных областей. Однако ограничения реляционной модели данных очевидны.

Модель не обеспечивает адекватных средств представления смысла данных.

Хотя процесс проектирования начинается с определения некоторых предметных объектов ("сущностей"), которые являются неотъемлемой частью приложения, и связей между этими сущностями, реляционная модель данных не предоставляет средств для разделения сущностей и связей.

Эти ограничения привели к появлению семантических (концептуальных, информационных) моделей данных. Разработанные семантические модели данных содержат структурную, операционную и целостную части, аналогичные реляционным моделям. Основная цель семантических моделей‒обеспечить возможность представления семантики данных.

В качестве альтернативы концептуальная схема автоматически компилируется в реляционную схему

Альтернативный вариант ‒ работа с базой данных в семантической модели, т. е. под управлением СУБД, основанной на семантической модели данных. (Третья возможность пока остается в области исследовательских и экспериментальных проектов).

Наиболее известным представителем класса семантических моделей в этой области является модель сущность ‒ связь (ER‒модель), ER ‒ модели обычно, обычно представляются в графической форме, либо с использованием оригинальной нотации, называемой ER‒диаграммой, либо с использованием других графических нотаций. На практике понятия ER ‒ модели и ER ‒ диаграммы часто не различаются, но были предложены другие графические нотации для визуализации ER‒модели Основными понятиями ER‒модели являются сущности, отношения и атрибуты (свойства).

ER-модель позволяет парикмахерской визуально представить свои данные и отношения между ними. Это помогает обеспечить целостность и точность данных, а также облегчает извлечение ценной информации для принятия решений.

ГЛАВА II. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Определение сущностей логической модели БД

В результате проведённого предметного анализа было выявлено что, данная база данных будет содержать следующие сущности:

− клиенты;

− залы;

− абонемент;

− тренажёрный зал;

− тренировка;

− Срок абонемента.

Для создания логической и физической схемы базы данных использовались инструменты Draw.io и DBdesigner. Данные инструменты позволили визуализировать концептуальную модель данных и преобразовать ее в физическую схему, учитывающую конкретную систему управления базами данных.

Таблица 1 ‒ «Клиент»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | Id | Primary key, autoincrement, integer | Уникальный идентификатор |
| 2 | full\_name | Varchar | ФИО клиента |
| 3 | hall\_id | Varchar | Идентификатор зала, куда куплен абонемент |
| 4 | begin\_ticket | Date | Начало абонемента |
| 5 | end\_ticket | Date | Конец абонемента |

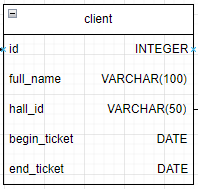


Рисунок 1 ‒ Таблица сущности клиента

Сущность клиент хранит информацию о клиентах фитнес зала, включая их данные, историю посещений, покупку абонемента. Эта информация используется для управления отношениями с клиентами, предоставления персонализированных услуг и повышения общего качества обслуживания.

Таблица 2 ‒ «Зал»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | id | Primary key, autoincrement, integer | Уникальный идентификатор |
| 2 | gym | Varchar | Зал с тренажёрами |
| 3 | dance | Varchar | Танцевальный зал |
| 4 | box | Varchar | Зал для занятия единоборствами |

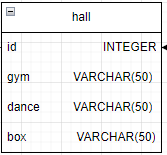


Рисунок 2 ‒ Таблица сущности зала

Сущность зала хранит информацию о наличии залов. Эта информация используется для знания используемых залов.

Таблица 3 ‒ «Фитнес комнат»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | id | Primary key, autoincrement, integer | Уникальный идентификатор |
| 2 | client\_id | Varchar | Идентификатор клиента, который купил абонемент |
| 3 | hall\_id | Varchar | Идентификатор зала, где было занятия |
| 4 | season\_ticket | Varchar | Абонент |
| 5 | coach | Varchar | Тренер |

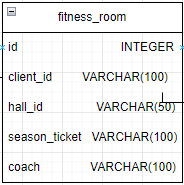


Рисунок 3 ‒ Таблица сущности фитнес комнат

Сущность фитнес комнат хранит информацию о различных услугах, предоставляемых фитнесом, включая их клиентов, нужно ли занятия с тренером. Эта информация отслеживания гостей, планирования встреч и предоставления клиентам информации о доступных вариантах.

Таблица 4 ‒ «Тренировка»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | id | Primary key, autoincrement, integer | Уникальный идентификатор |
| 4 | client\_id | Varchar | Идентификатор клиента, который занимался |
| 5 | begin\_coaching | Time | Время начала тренировки |
| 6 | end\_coaching | Time | Время конца тренировки |

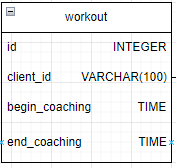


Рисунок 4 ‒ Таблица сущности тренировка

Сущность тренировки хранит информацию о клиенте, который пришёл, во сколько он начал тренировку и во сколько закончил.

Таблица 5 ‒ «Абонемент»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | id | Primary key, autoincrement, integer | Уникальный идентификатор |
| 2 | time\_ticket\_id | Date | Срок абонемента |
| 3 | hall\_id | Varchar | Уникальный идентификатор зала, на который купили абонемент |

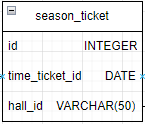
****

Рисунок 5 ‒ Таблица сущности абонемента

Сущность абонемента хранит информацию о том на сколько был куплен абонемент и в какой зал.

Таблица 6 ‒ «Срок абонемента»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Название | Тип | Описание |
| 1 | id | Primary key, autoincrement, integer | Уникальный идентификатор |
| 2 | one\_traning | Time | Одна тренировка |
| 3 | month | Time | Абонемент на месяц |
| 4 | year | Time | Абонемент на год |

Сущность инвентарь хранит информацию о расходных материалах и оборудовании, используемых в парикмахерской, включая их наименование, количество, поставщиков и стоимость. Эта информация используется для управления запасами, обеспечения наличия необходимых материалов, контроля расходов и оптимизации закупок.

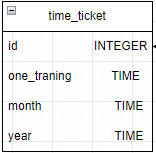


Рисунок 6 ‒ Таблица сущности срока абонемента

Сущность инвентарь хранит информацию о расходных материалах и оборудовании, используемых в парикмахерской, включая их наименование, количество, поставщиков и стоимость. Эта информация используется для управления запасами, обеспечения наличия необходимых материалов, контроля расходов и оптимизации закупок.

2.1.1 Построение логической БД

На основе определения сущностей, можно построить логическую БД, я воспользовался программой Draw.io, и вот что у меня получилось:

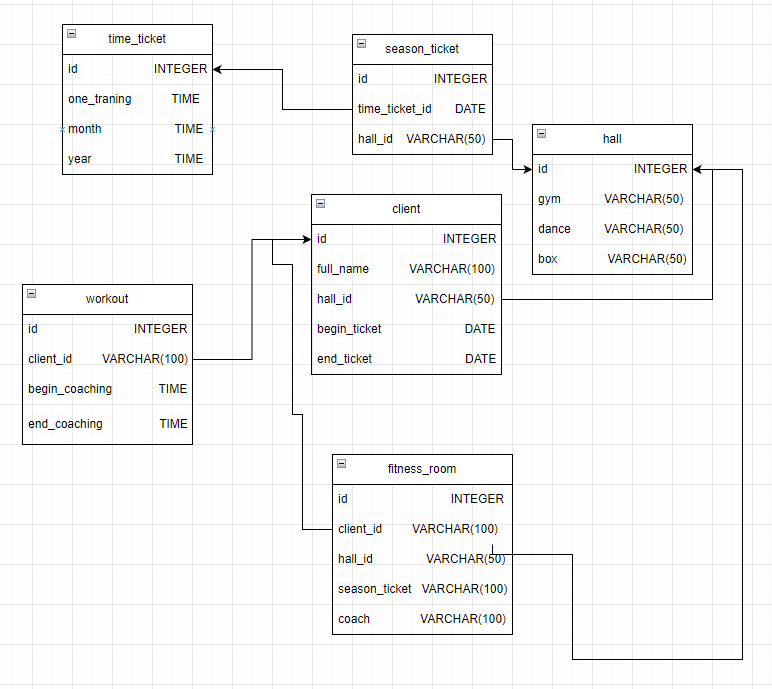


Рисунок 7 ‒ База данных

2.2 Построение физической модели БД

После создания логической БД, можно создать физическую БД, и реализовать связь между таблицами «Рисунок 7». Теперь же я воспользуюсь программой DBdesigner:

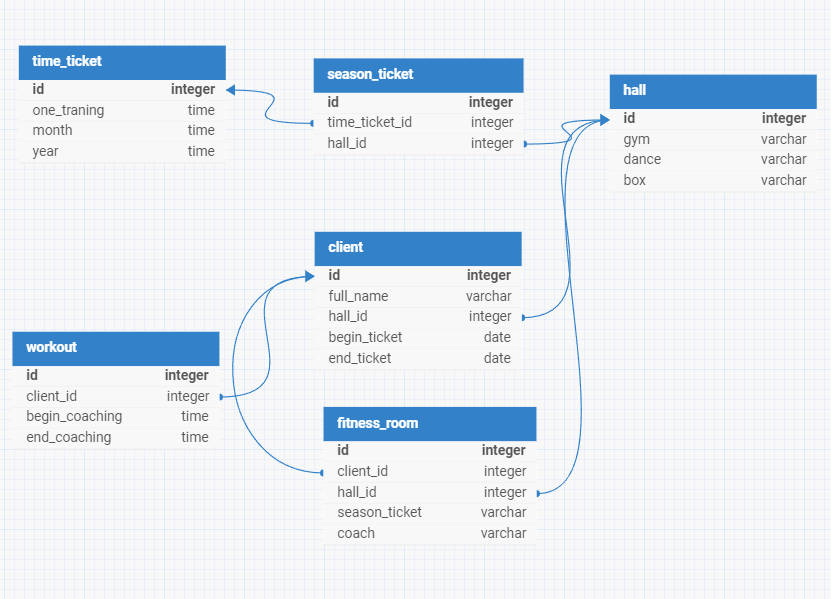


Рисунок 8 ‒ База данных на DBdesigner

2.2.1 Запросы для БД

Одним из основных типов запросов, используемых в базах данных, является запрос на выборку данных. Запросы на выборку позволяют извлекать конкретные данные из таблиц базы данных на основе определенных условий.

Запросы на выборку могут использоваться для различных целей, таких как:

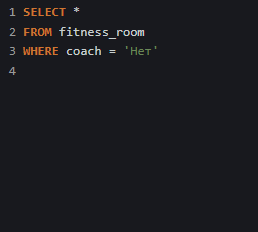
‒ получение списка клиентов;

‒начало и конец тренировки;

‒ вывод списка сотрудников, работающих в отделе парикмахерских услуг;

‒Выбор различных залов.

Используя запросы на выборку, администраторы могут спокойно пробивать информацию о клиентах и знать в какое время они приходят и уходят.



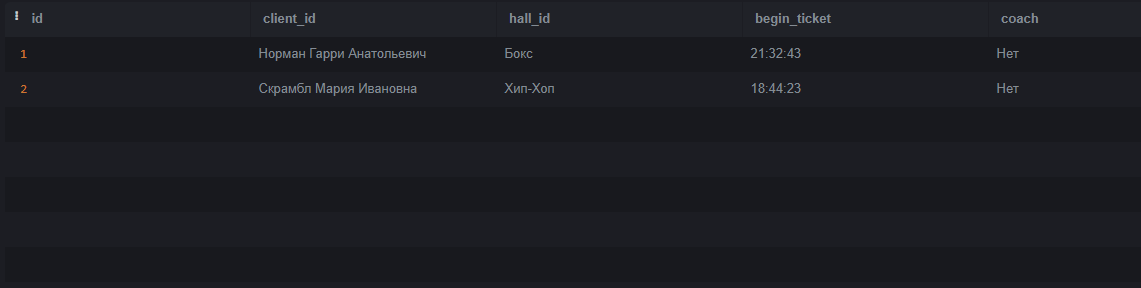


Рисунок 9,10 ‒ Результат запроса о получении имени пользователя

Вывод информации о клиенте в таблице «fitness\_room».

Запрос:

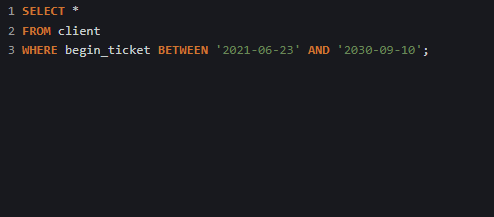
− select;

− from fitness\_room;

− where coach = ‘Нет’.

Этот запрос извлекает все строки из таблицы «fitness\_room», где значение столбца «coach» равно “Нет”. Он вернет всю информацию, хранящуюся в таблице для клиентов, которые отказались от тренера.

Условие WHERE фильтрует строки по столбцам, гарантируя, что будут возвращены только данные о клиентах, которые отказались от тренера.



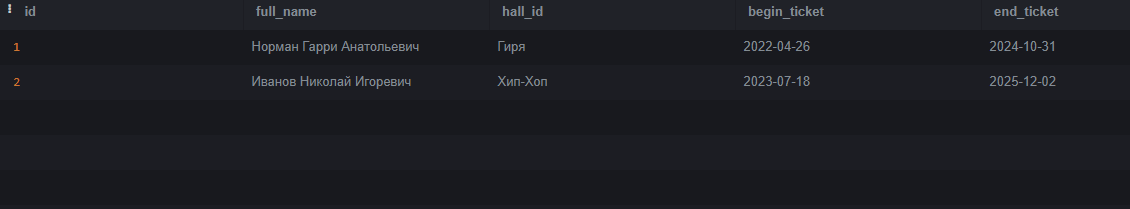


Рисунок 11,12 ‒ Результат запроса о выводе цены на товары

Вывод в таблице «client» цену закупок.

Запрос:

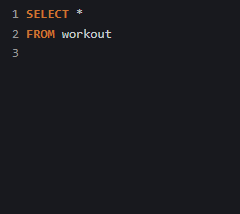
− select;

− from client;

−where begin\_ticket BETWEEN ‘2021-06-23’ AND ‘2030-09-10’.

Этот запрос извлекает все строки из таблицы «client», где значения столбца «begin\_ticket» в диапазоне от 2021-06-23 и до 2030-09-10. Он вернёт лишь те даты, которые находятся в этом диапазоне.

Условие WHERE фильтрует строки гарантируя, что будут возвращены только данные о тех датах, которые находятся в указанном диапазоне.



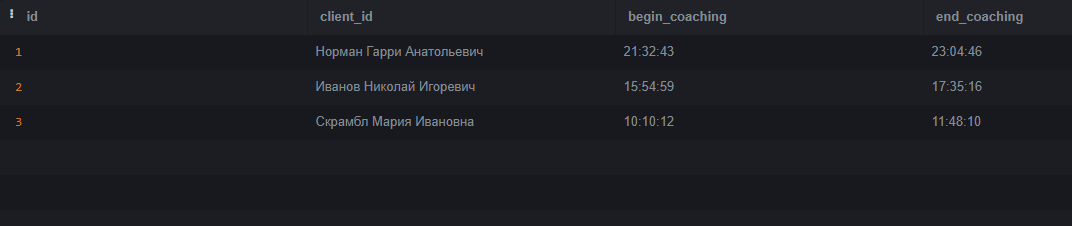


Рисунок 13,14 ‒ Результат запроса о выводе времени на запись

Вывод в таблице «workout» время и запись.

Запрос:

− select;

− from workout.

Этот запрос извлекает все строки из таблицы «workout». Он вернет всю информацию, хранящуюся в таблице, включая время и запись для каждого элемента расписания.

В данном случае запрос не использует условие WHERE, что означает, что он извлечет все строки из таблицы, не применяя никаких фильтров, и будут возвращены только данные тренировках.

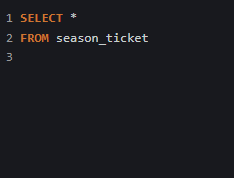




Рисунок 15,16 ‒ Результат запроса о выводе информации сотрудниках

Вывод информации о сотрудниках в таблице «season\_ticket»:

Запрос:

− select;

− from season\_ticket.

Этот запрос извлекает все строки из таблицы season\_ticket. Он вернет всю информацию, хранящуюся в таблице, для всех сотрудников.

В данном случае запрос не использует условие WHERE, что означает, что он извлечет все строки из таблицы, не применяя никаких фильтров, и будут возвращены только данные о абонементе.

ЗАКЛЮЧЕНИЕ

В процессе создания модели реляционной базы данных для парикмахерской были выявлены ключевые сущности и связи между ними. Это позволило разработать эффективный механизм хранения и управления данными, обеспечивающий быстрый доступ к информации о клиентах, услугах, персонале и других важных аспектах работы салона.

Созданная модель данных послужит основой для дальнейшего развития информационной системы парикмахерской. Она позволит автоматизировать процессы работы, повысить качество обслуживания клиентов и оптимизировать работу персонала.

Оптимизация работы персонала: Автоматизация рутинных задач позволит сотрудникам сосредоточиться на предоставлении качественных услуг клиентам, повышая общую эффективность работы.

Повышение удовлетворенности клиентов: Улучшение системы управления данными позволит парикмахерской предоставлять клиентам персонализированные услуги, учитывая их предпочтения и историю посещений.

Упрощение управления салоном: Централизация данных и автоматизированные процессы упростят управление парикмахерской, высвобождая время руководителей для стратегического планирования и развития бизнеса.

Внедрение базы данных, разработанной на основе выявленных сущностей и связей, позволит парикмахерской повысить свою эффективность, улучшить качество обслуживания и заложить основу для дальнейшего роста и развития, трансформируются в высокоэффективные и ориентированные на клиента предприятия. Они не просто улучшают свои операции, но и повышают планку для всей индустрии, устанавливая новые стандарты в области красоты и обслуживания.

Список использованных источников

1. А. И. Гайдов. Проектирование информационных систем. – М.: Издательский дом «Альпина Паблишер», 2017. – 416 с.
2. Архангельский А.Я. Программирование в Delphi 7. – М.: ООО "Бином Пресс", 2005. – 627 с.
3. Баженова И.Ю. «Основы проектирования приложений баз данных»,2009.
4. Голубев Д.А. "Реляционные базы данных: теория и практика". – М.: БХВ–Петербург, 2019.
5. К.Д. Кондратьев, С.И. Сидоренко, В.В. Карпов. Базы данных: учебное пособие. – М.: Инфра–М, 2019. – 432 с.
6. Карпов А.В. "Основы проектирования баз данных". – СПб.: Питер, 2018.
7. Кириллов В.В., Громов Г.Ю. «Введение в реляционные базы данных», 2009.
8. Кирин М. Н.”Учебник для продвинутых по Delphi 7 “. – М.: ООО "Бином–Пресс", 2003. – 496 с..
9. Климова Л.М. Delphi 7. Основы программирования. Решение типовых задач. Самоучитель. Издание третье. – М.: КУДИЙ–ОБРАЗ,2006. – 480 с.
10. Корняков В.Н. Программирование документов и приложений MS Office в Delphi. – 2005.
11. Кузнецов С.О. "Базы данных: учебное пособие". – М.: Издательский центр "Академия", 2019.
12. Л.С. Васильев. Основы проектирования баз данных. – СПб.: БХВ–Петербург, 2018. – 384 с.
13. Лукин А.В. "Реляционные базы данных: учебное пособие". – М.: Издательский центр "Академия", 2016.
14. Осипов Д.Л. Базы данных и Delphi. Теория и практика.: БХВ–Петербург, 2011.– 752 с.