

РОССИЙСКАЯ ФЕДЕРАЦИЯ
ХАНТЫ-МАНСКИЙ 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 фитнеса зала»

Выполнил: Таращук Максим
Павлович
Студент 2 курса группы ИС
23/11
Проверил: Преподаватель
профессиональных дисциплин
Колмыков М.В. _____

Сургут, 2024

Содержание

ВВЕДЕНИЕ.....	3
ГЛАВА I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	4
1.1 Основные положения теории баз данных, хранилищ данных, баз знаний	4
1.2 Анализ предметной области проектирования.....	7
1.3. Разработка концептуальной модели данных.....	11
ГЛАВА II. ПРАКТИЧЕСКАЯ ЧАСТЬ.....	14
2.1 Определение сущностей логической модели БД.....	14
2.2 Построение физической модели БД.....	18
2.3 Запросы для БД.....	22
ЗАКЛЮЧЕНИЕ	26
Список использованных источников	27

Введение

В современном мире, где здоровый образ жизни становится все более популярным, фитнес залы играют ключевую роль в обеспечении населения возможностью вести активный и здоровый образ жизни. Однако, чтобы эффективно управлять такими заведениями, необходимо надежное и гибкое информационное обеспечение, которое позволит хранить, обрабатывать и анализировать данные о клиентах, тренировках, абонементах и многих других аспектах деятельности фитнес центра.

Разработка модели реляционной базы данных для фитнес-зала-это актуальная и многогранная задача, которая требует глубокого понимания принципов реляционных баз данных, а также специфики работы фитнес центров.

Целью данной курсовой работы является разработка модели реляционной базы данных, которая позволит эффективно управлять информацией в фитнес зале, обеспечивая:

Точность и актуальность данных: База данных должна хранить информацию о клиентах, тренерах, абонементах, занятиях и других сущностях в фитнес зале.

Гибкость и масштабируемость: Модель должна позволять легко добавлять новые данные и изменять существующие, а также адаптироваться к растущим потребностям фитнес центра.

Безопасность и конфиденциальность: важно обеспечить защиту персональных данных клиентов и предотвратить несанкционированный доступ к информации.

ГЛАВА I. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

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

База данных — это структурированная коллекция данных, хранящаяся в электронном виде, которая организована таким образом, чтобы минимизировать избыточность и обеспечить эффективное использование для одного или нескольких приложений.

Цель создания баз данных — разработка системы, которая не зависит от программного обеспечения, используемых технических средств и физического расположения данных. Проектирование баз данных предполагает их многоцелевое использование.

База данных, в самом простом виде, представляет собой набор двумерных таблиц.

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

Схема пользователя — это конкретный вариант порядка полей таблицы, зафиксированный для определенного пользователя.

Система управления базами данных (СУБД) — это программное обеспечение, отвечающее за организацию, хранение, целостность, изменение, чтение и безопасность информации в базе данных.

Информационная система — это система, автоматизирующая сбор, обработку и управление данными, включающая технические средства обработки данных, программное обеспечение и обслуживающий персонал.

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

В зависимости от способа хранения данных и организации связей между ними, базы данных делятся на несколько типов:

Иерархическая - данные организованы в древовидную структуру, где корневой элемент связан с элементами второго уровня иерархии.

Сетевая - похожа на иерархическую, но связи могут устанавливаться между любыми элементами графа.

Реляционная - данные организованы в виде таблиц, где каждая запись — это строка с уникальным ключом.

Объектно-ориентированная - данные хранятся в виде объектов, обладающих свойствами и методами.

Документа-ориентированная - данные организованы в виде документов, которые могут быть структурированными или неструктурированными.

Графовая - данные представлены в виде графа, где узлы — это объекты, а ребра — связи между ними.

Каждая модель имеет свои преимущества и недостатки, и выбор модели зависит от конкретных требований приложения и характера данных.

1.1.2 Свойства базы данных

Основные свойства баз данных:

Структурированность - данные организованы в определенной структуре, что позволяет эффективно хранить и управлять ими.

Постоянство - данные сохраняются долгосрочно и остаются доступными после завершения сессии работы с приложением.

Доступность - база данных обеспечивает доступ к данным для пользователей и приложений в соответствии с установленными правами доступа.

Целостность - база данных гарантирует точность и согласованность данных во всей системе, даже при одновременном доступе нескольких пользователей.

Безопасность - база данных обеспечивает защиту данных от несанкционированного доступа, изменений или утраты.

Масштабируемость - база данных должна быть способна обрабатывать как малые, так и большие объемы данных и увеличивать производительность при необходимости.

Производительность - база данных должна обеспечивать быстрый доступ к данным и эффективную обработку запросов.

Распределенность - современные базы данных могут быть распределенными, что обеспечивает отказоустойчивость и масштабируемость.

Резервирование - база данных должна иметь механизмы резервного копирования и восстановления данных.

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.

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. Это упрощает поиск и обработку взаимосвязанных данных, уменьшает избыточность данных и упрощает процесс ведения баз данных.

Принцип централизации управления данными обеспечивает единый контроль и администрирование всей базы данных, упрощая управление доступом, резервное копирование и восстановление.

Эти принципы обеспечивают основу для создания высококачественных систем, которые соответствуют современным требованиям к надежности, эффективности и масштабируемости.

1.1.4 Хранилища данных (DataWarehouses)

Хранилище данных — это централизованная база данных, предназначенная для хранения и анализа больших объемов, структурированных данных из различных источников. В отличие от операционных баз данных, которые оптимизированы для выполнения транзакций в реальном времени, хранилища данных оптимизированы для аналитических запросов и поддержки принятия решений.

Основные характеристики хранилищ данных:

Интеграция данных - хранилища данных объединяют данные из разных источников, включая операционные системы, системы CRM, финансовые системы и т.д.

Тематическая ориентированность - данные в хранилище данных организованы вокруг конкретных тем или областей деятельности, например, продажи, маркетинг, финансы.

Историчность - хранилища данных содержат исторические данные, что позволяет анализировать тенденции и изменения во времени.

Неизменность - данные в хранилище данных не изменяются, а только добавляются новые данные.

1.1.5 Базы знаний (KnowledgeBases)

База знаний — это система, которая хранит, организует и предоставляет доступ к информации, необходимой для принятия решений. В отличие от баз данных, которые хранят факты, базы знаний также содержат правила, процедуры, модели и другие типы знаний.

Основные характеристики баз знаний:

Семантическая организация - база знаний организована на основе смысла и взаимосвязей между данными.

Автоматизация вывода - база знаний может использовать правила и модели для автоматического вывода новых знаний из существующих данных.

Поддержка принятия решений - база знаний предоставляет информацию и знания, необходимые для принятия обоснованных решений.

1.2 Анализ предметной области проектирования

Анализ предметной области проектирования для фитнес клуба включает изучение основных аспектов деятельности, потребностей клиентов и требований к функциональности программного обеспечения.

Ключевые моменты, которые следует учитывать:

Услуги фитнес клуба: анализ предоставляемых услуг, таких как групповые занятия, персональные тренировки.

Расписание и запись: анализ расписания занятий, персональных тренировок и возможностей онлайн записи.

Инвентарь и оборудование: анализ инвентаря, включая тренажеры, спортивное оборудование и расходные материалы.

Учет услуг и финансов: система учета доходов, расходов, прибыли и других финансовых показателей.

Клиентская база и управление клиентами: ведение базы данных клиентов для отслеживания их активности, предпочтений, истории посещений и контактных данных.

Управление персоналом: оптимальное управление персоналом, включая назначение рабочих графиков, контроль за производительностью и мотивацию сотрудников.

Безопасность и гигиена: соблюдение норм безопасности и гигиены для обеспечения безопасности персонала и клиентов.

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

1.2.1 Создание базы данных на SQLite

SQLite — это компактная и самодостаточная реляционная СУБД, которая является частью стандартной библиотеки в большинстве языков программирования. Она отличается от других СУБД тем, что не является серверной базой данных, а вместо этого представляет собой библиотеку, которая непосредственно встраивается в приложение.

Основные особенности SQLite:

Легковесность: очень компактна и не требует установки отдельного сервера баз данных. Вся база данных хранится в одном файле.

Надежность: обеспечивает надежное хранение данных с поддержкой транзакций и ACID-свойств.

Кроссплатформенность: поддерживается на различных операционных системах.

Поддержка большого объема данных: может управлять базами данных размером до нескольких терабайт.

Преимущества SQLite:

Простота интеграции: легко интегрируется в приложения благодаря своей компактности и отсутствию необходимости установки отдельного сервера баз данных.

Независимость от платформы: поддерживается на различных операционных системах.

Эффективность: обеспечивает высокую производительность и эффективное использование ресурсов.

Низкие требования к ресурсам: не требует больших объемов оперативной памяти или вычислительных мощностей для работы.

Богатый набор функций: Поддерживает множество функций SQL.

Бесплатность и открытость: является бесплатным и открытым программным обеспечением.

1.2.2 Описание основных сущностей и атрибутов, связей между ними
Основные сущности, которые могут быть присутствовать в базе данных фитнес клуба:

Клиенты: хранение информации о клиентах, такой как имя, контактные данные, предпочтения в отношении услуг, история посещений.

Услуги: хранение информации о предоставляемых услугах и их стоимости.

Персонал: хранение информации о персонале, включая имена, специализацию, контактные данные и график работы.

Расписание: хранение информации о расписании занятий, персональных тренировок и записях клиентов на определенные даты и время.

Инвентарь: хранение информации о инвентаре и оборудовании.

Связи между сущностями:

Каждый клиент может иметь несколько записей на услуги.

Каждый тренер может иметь много записей на занятия, 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 расписания тренеров.

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 они имеют ограничения:

Не обеспечивают адекватных средств представления смысла данных.

Не предоставляют средств для разделения сущностей и связей.

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

Наиболее известным представителем класса семантических моделей является модель сущность-связь (ER-модель). ER-модели обычно представляются в графической форме, с использованием ER-диаграмм или других графических нотаций. Основными понятиями ER-модели являются сущности, отношения и атрибуты.

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

1.2.4 Анализ конкурентов и рынка

При анализе предметной области проектирования для фитнес клуба также важно проанализировать конкурентов и рынок. Это поможет определить

уникальные преимущества фитнес клуба и разработать программное обеспечение, которое будет соответствовать потребностям целевой аудитории.

Анализ конкурентов может включать:

Исследование предложений конкурентов: Какие услуги предлагают конкуренты? Какие у них цены? Какие у них преимущества и недостатки?

Анализ сайта и приложений конкурентов: Как конкуренты используют цифровые технологии для привлечения клиентов и управления бизнесом?

Опрос потенциальных клиентов: Что важно для клиентов при выборе фитнес клуба? Какие функции они хотели бы видеть в программном обеспечении?

1.2.5 Выбор технологий и инструментов

При выборе технологий и инструментов для разработки программного обеспечения для фитнес клуба необходимо учитывать следующие факторы:

Требования к функциональности: Какие функции должно выполнять программное обеспечение?

Масштабируемость: Будет ли программное обеспечение использоваться в одном фитнес клубе или планируется его масштабирование на несколько филиалов?

Бюджет: Каков бюджет на разработку программного обеспечения?

Опыт разработчиков: Какой опыт имеют разработчики в использовании выбранных технологий и инструментов?

1.3. Разработка концептуальной модели данных

Концептуальная модель данных — это высокоуровневое описание данных, которое фокусируется на сущностях и отношениях между ними, не привязываясь к конкретной реализации. Для разработки концептуальной модели данных для фитнес клуба можно использовать модель сущность-связь (ER-модель).

Основные шаги разработки концептуальной модели данных:

Идентификация сущностей: Определение основных объектов, которые будут представлены в базе данных. Для парикмахерской это могут быть:

- Клиенты
- Услуги
- Персонал
- Расписание
- Операции
- Инвентарь

Определение атрибутов сущностей: Описание свойств каждой сущности.

Например, для сущности "Клиенты" атрибутами могут быть:

- Имя
- Контактные данные
- Предпочтения в отношении услуг
- История посещений

Определение связей между сущностями: Описание отношений между сущностями. Например, связь между сущностями "Клиенты" и "Услуги" может быть "один ко многим".

1.3.1 Разработка логической модели данных

Логическая модель данных — это промежуточная модель между концептуальной моделью данных и физической моделью данных. Она определяет структуру данных, типы данных, связи между таблицами и ограничения целостности.

Основные шаги разработки логической модели данных:

Преобразование концептуальной модели в таблицы: каждая сущность концептуальной модели преобразуется в таблицу.

Определение атрибутов таблиц: каждый атрибут сущности преобразуется в столбец таблицы.

Определение первичных и внешних ключей: первичные ключи уникально идентифицируют каждую запись в таблице. Внешние ключи связывают таблицы между собой.

Определение ограничений целостности: ограничения целостности гарантируют точность и согласованность данных.

1.3.2 Разработка физической модели данных

Физическая модель данных — это конкретная реализация логической модели данных в выбранной СУБД. Она определяет физическое хранение данных, включая типы данных, индексы, хранимые процедуры и триггеры.

Основные шаги разработки физической модели данных:

Выбор СУБД: Выбор СУБД, которая наилучшим образом соответствует требованиям проекта.

Определение физических типов данных: Выбор физических типов данных, которые будут использоваться для хранения данных.

Создание индексов: Создание индексов для ускорения выполнения запросов.

Разработка хранимых процедур и триггеров: Разработка хранимых процедур и триггеров для автоматизации бизнес-логики.

ГЛАВА II. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Определение сущностей логической модели БД

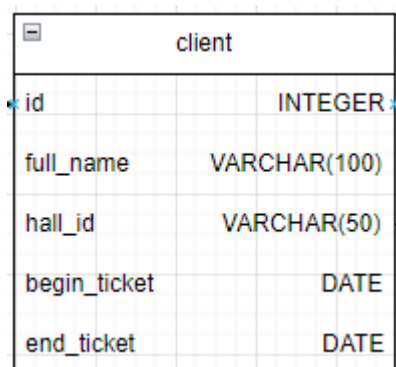
В результате проведённого предметного анализа было выявлено что, данная база данных будет содержать следующие сущности:

- 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.

Название	Тип	Описание
Id	Primary key, autoincrement, integer	Уникальный идентификатор
full_name	Varchar	ФИО клиента
hall_id	Varchar	Идентификатор зала, куда куплен абонемент
begin_ticket	Date	Начало абонемента
end_ticket	Date	Конец абонемента

Таблица 1 – «Клиент»



client	
id	INTEGER
full_name	VARCHAR(100)
hall_id	VARCHAR(50)
begin_ticket	DATE
end_ticket	DATE

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.

Название	Тип	Описание
id	Primary key, autoincrement, integer	Уникальный идентификатор
gym	Varchar	Зал с тренажёрами
dance	Varchar	Танцевальный зал
box	Varchar	Зал для занятия единоборствами

Таблица2 – «Зал»

hall	
id	INTEGER
gym	VARCHAR(50)
dance	VARCHAR(50)
box	VARCHAR(50)

Рисунок 2– Таблица сущности зала

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

Название	Тип	Описание
id	Primary key, autoincrement, integer	Уникальный идентификатор
client_id	Varchar	Идентификатор клиента, который купил абонемент
hall_id	Varchar	Идентификатор зала, где было занятия
season_ticket	Varchar	Абонемент
coach	Varchar	Тренер

Таблица3 – «Фитнес зала»

fitness_room		
*id		INTEGER *
client_id		VARCHAR(100)
hall_id		VARCHAR(50)
season_ticket		VARCHAR(100)
coach		VARCHAR(100)

Рисунок 3– Таблица сущности фитнес зала

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.

Название	Тип	Описание
id	Primary key, autoincrement, integer	Уникальный идентификатор
client_id	Varchar	Идентификатор клиента, который занимался
begin_coaching	Time	Время начала тренировки
end_coaching	Time	Время конца тренировки

Таблица4 – «Тренировка»

workout		
id		INTEGER
client_id		VARCHAR(100)
begin_coaching		TIME
*end_coaching		TIME *

Рисунок 4– Таблица сущности тренировка

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

Название	Тип	Описание
id	Primarykey, autoincrement, integer	Уникальный идентификатор
time_ticket_id	Date	Срок абонеента
hall_id	Varchar	Уникальный идентификатор зала, на который купили абонемент

Таблица5 – «Абонемент»

season_ticket	
id	INTEGER
*time_ticket_id	DATE *
hall_id	VARCHAR(50)

Рисунок 5– Таблица сущности абонеента

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

Название	Тип	Описание
id	Primary key, autoincrement, integer	Уникальный идентификатор
one_traning	Time	Одна тренировка
month	Time	Абонемент на месяц
year	Time	Абонемент на год

Таблица6 – «Срок абонеента»

time_ticket	
id	INTEGER
one_traning	TIME
month	TIME
year	TIME

Рисунок 6 – Таблица сущности срока абонеента

Сущность срока абонеента хранит информацию о том какие абонементы присутствуют и что можно приобрести.

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:

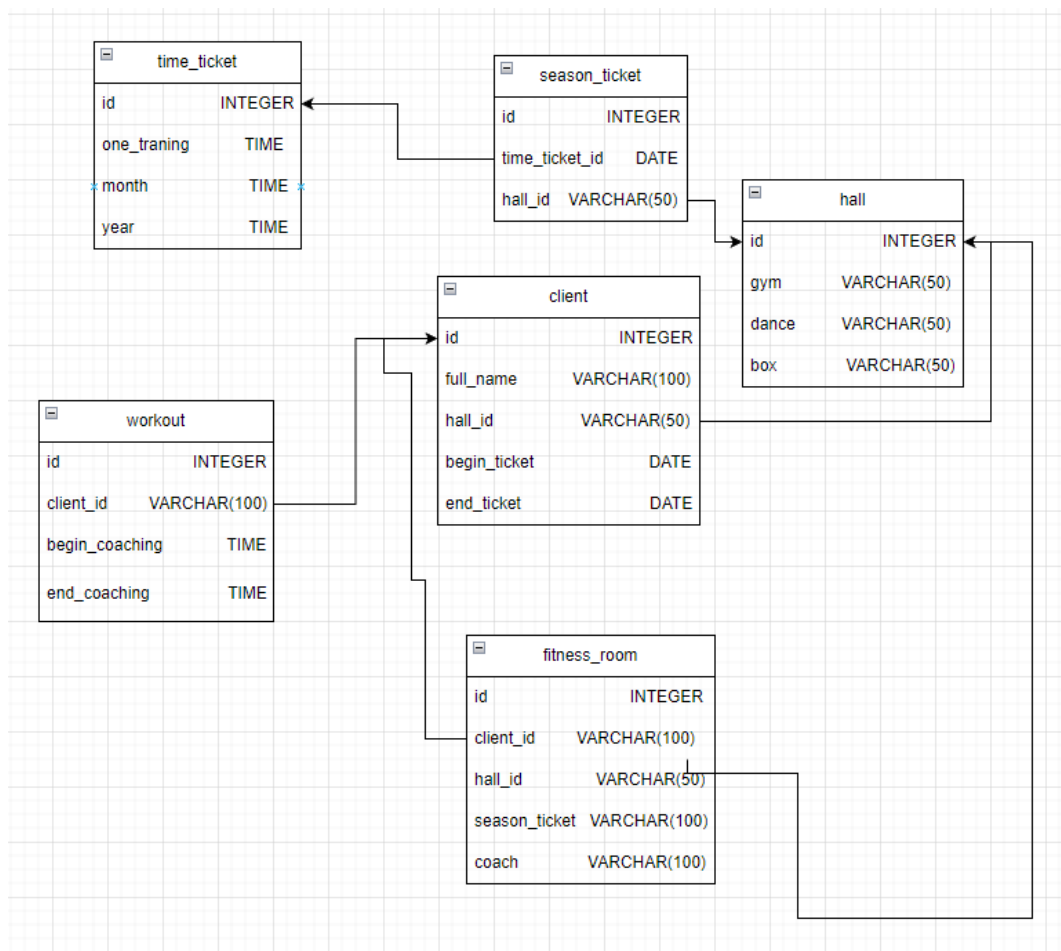


Рисунок 7—База данных

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:

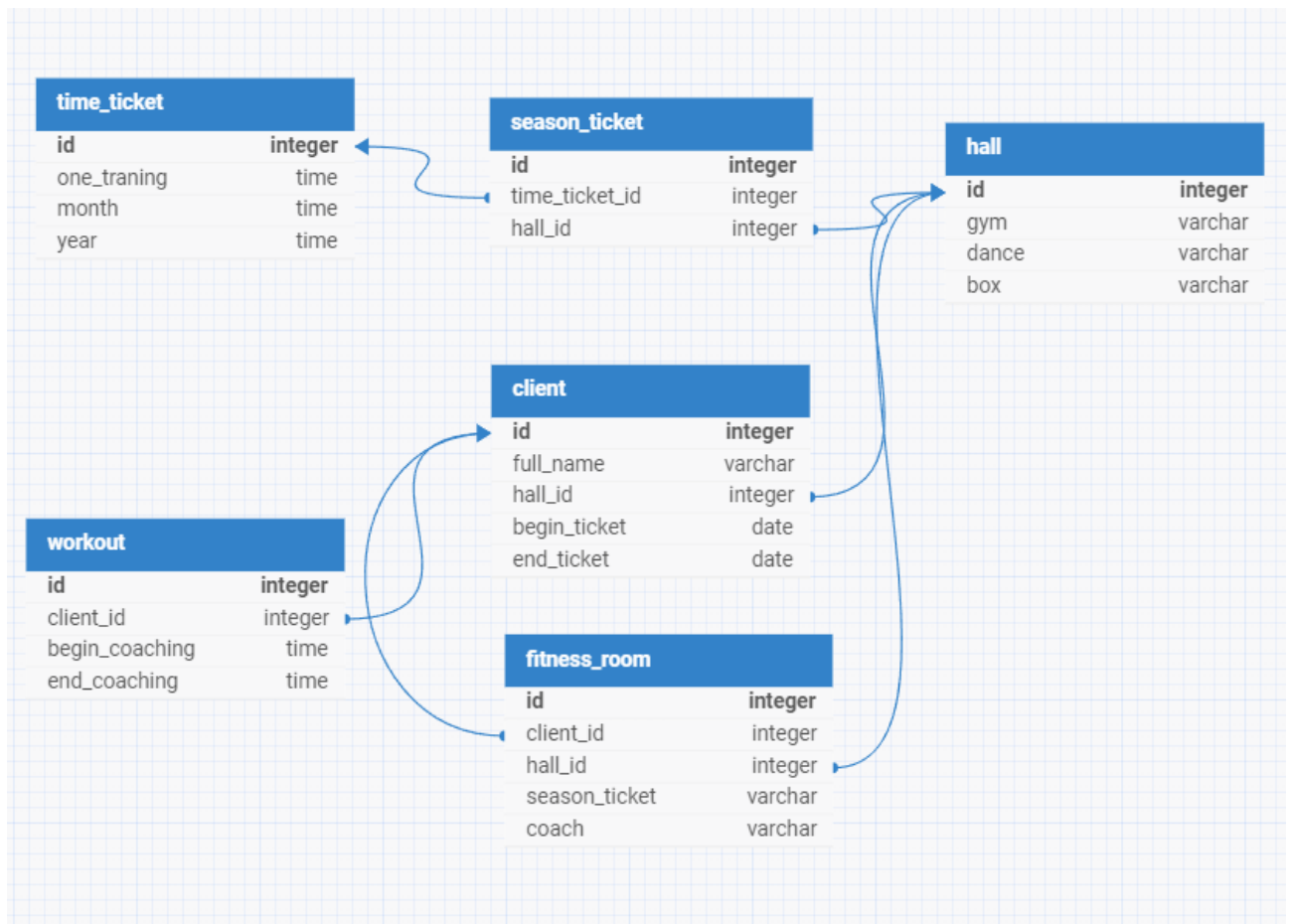


Рисунок8 – База данных на DBdesigner

2.2.1 Описание таблиц физической БД

Таблица: Клиент

1. id

Тип данных: INTEGER, PRIMARY KEY

Описание: Уникальный идентификатор присеваемый клиентам.

2. full_name

Тип данных: VARCHAR(150)

Описание: ФИО клиента.

3. hall_id

Тип данных: INTEGER

Описание: Служит для привязки клиента к конкретному залу.

4. begin_ticket

Тип данных: DATE

Описание: Дата начала действия абонемента.

5. end_ticket

Тип данных: DATE

Описание: Дата конца действия абонемента.

Таблица: Зал

1. id

Тип данных: INTEGER, PRIMARY KEY

Описание: Уникальный идентификатор присеваемый залам.

2. gym

Тип данных: VARCHAR(50)

Описание: Зал с тренажёрами.

3. dance

Тип данных: VARCHAR(50)

Описание: Танцевальный зал.

4. box

Тип данных: VARCHAR(50)

Описание: Зал для занятия единоборствами.

Таблица: Фитнес залам

1. id

Тип данных: INTEGER, PRIMARY KEY

Описание: Уникальный идентификатор присеваемый фитнес залам.

2. client_id

Тип данных: INTEGER

Описание: Служит для привязки клиента к конкретному фитнес залу.

3. hall_id

Тип данных: INTEGER

Описание: Служит для привязки зала к конкретному фитнес залу.

4. season_ticket

Тип данных: VARCHAR(100)

Описание: Абонемент на конкретный фитнес зал.

5. coach

Тип данных: VARCHAR(100)

Описание: Выбор занятия с тренером или без.

Таблица: Тренировка

1. id

Тип данных: INTEGER, PRIMARY KEY

Описание: Уникальный идентификатор присеваемый тренировкам.

2. client_id

Тип данных: INTEGER

Описание: Служит для привязки клиента к тренировке.

3. begin_coaching

Тип данных: TIME

Описание: Время начала тренировки.

4. end_coaching

Тип данных: TIME

Описание: Время конца тренировки.

Таблица: Абонемент

1. id

Тип данных: INTEGER, PRIMARY KEY

Описание: Уникальный идентификатор присеваемый абонементам.

2. time_ticket_id

Тип данных: INTEGER

Описание: Служит для привязки срока абонемента к абонементу.

3. hall_id

Тип данных: INTEGER

Описание: Служит для привязки зала к абонементу.

Таблица: Срок абонемент

1. id

Тип данных: INTEGER, PRIMARY KEY

Описание: Уникальный идентификатор присеваемый срокам абонемента.

2. one_training

Тип данных: TIME

Описание: Поход на одну тренировку.

3. month

Тип данных: TIME

Описание: Абонемент сроком на месяц.

4. year

Тип данных: TIME

Описание: Абонемент сроком на год.

2.3 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;

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

```
1 SELECT *  
2 FROM fitness_room  
3 WHERE coach = 'Нет'  
4
```

Рисунок 9 – Код для получения запроса

id	client_id	hall_id	begin_ticket	coach
1	Норман Гарри Анатольевич	Бокс	21:32:43	Нет
2	Скрамбл Мария Ивановна	Хип-Хоп	18:44:23	Нет

Рисунок 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.

```
1 SELECT *
2 FROM client
3 WHERE begin_ticket BETWEEN '2021-06-23' AND '2030-09-10';
```

id	full_name	hall_id	begin_ticket	end_ticket
1	Норман Гарри Анатольевич	Гиря	2022-04-26	2024-10-31
2	Иванов Николай Игоревич	Хип-Хоп	2023-07-18	2025-12-02

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

Вывод в таблице «client» с диапазоном дат.

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 2021-06-23и до2030-09-10. 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.

```
1 SELECT *
2 FROM workout
3
```

id	client_id	begin_coaching	end_coaching
1	Норман Гарри Анатольевич	21:32:43	23:04:46
2	Иванов Николай Игоревич	15:54:59	17:35:16
3	Скрамбл Мария Ивановна	10:10:12	11:48:10

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

Вывод информации в таблице «workout».

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 тренировок.

```
1 SELECT *
2 FROM season_ticket
3
```

id	time_ticket_id	hall_id
1	years	Бокс
2	years	Бокс
3	month	Бальные танцы

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 ходе выполнения курсовой работы была проведена комплексная работа по проектированию базы данных для фитнес клуба, включая анализ предметной области, разработку концептуальной, логической и физической моделей данных, а также реализацию базы данных с использованием СУБД SQLite.

Были выявлены ключевые аспекты деятельности фитнес клуба, такие как предоставляемые услуги, расписание работы, управление клиентами, персоналом.

Была создана концептуальная модель данных с использованием ER-диаграммы, включающая основные сущности: Клиенты, Фитнес комната, Тренировки, Залы, Абонемент и Количество тренировок.

Была разработана логическая модель данных, определяющая структуру таблиц, типы данных, связи между таблицами и ограничения целостности.

Была реализована физическая модель данных в СУБД SQLite, включая создание таблиц, индексов, хранимых процедур и триггеров.

Была создана база данных для фитнес клуба, наполненная начальными данными и настроенная для обеспечения безопасности и доступа.

Было проведено тестирование базы данных на целостность данных и производительность, а также выполнена оптимизация запросов и индексов.

В рамках практического задания была разработана логическая и физическая база данных для фитнес-зала. Были определены основные сущности, такие как Клиенты, Абонементы, Залы, Тренировки, Фитнес комната и Количество тренировок, и установлены связи между ними. Были созданы таблицы, индексы и ограничения целостности в СУБД SQLite, а также наполнены данными.

Выполнение курсовой работы позволило закрепить теоретические знания по проектированию баз данных и получить практические навыки работы с СУБД SQLite. Разработанная база данных для фитнес-зала является эффективным инструментом для управления информацией и поддержки принятия решений в бизнесе.

Список использованных источников

1. Дейт, К. Дж. "Введение в системы баз данных" (8-е издание). — Вильямс, 2006.
2. Гарсиа-Молина, Г., Ульман, Дж., Уидом, Дж. "Системы баз данных: полный курс" (3-е издание). — Вильямс, 2008.
3. Коннолли, К., Бегг, К. "Базы данных: проектирование, реализация и управление" (8-е издание). — Вильямс, 2011.
4. Силбер, Х. "Моделирование данных: подход к решению проблем управления информацией" (3-е издание). — Вильямс, 2005.
5. Официальный сайт: <https://www.sqlite.org/>
6. Документация: <https://www.sqlite.org/docs.html>
7. Coursera: курсы по базам данных от университетов, таких как Стэнфордский университет и Калифорнийский университет в Беркли.
8. KhanAcademy: бесплатные уроки по базам данных.
9. W3Schools: учебные материалы по SQL и базам данных
10. KimballGroup: ресурсы по проектированию хранилищ данных.
11. DataWarehousingInstitute (TDWI): исследования и ресурсы по хранилищам данных.
12. Stanford Encyclopedia of Philosophy: статьи по теории баз знаний.