МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное

учреждение высшего образования

«Казанский национальный исследовательский технический

университет им. А.Н. Туполева-КАИ»

(КНИТУ-КАИ)

|  |
| --- |
| Отделение СПО ИКТЗИ, КИТ |

*(наименование института (факультета)*

**КУРСОВАЯ РАБОТА**

по дисциплине «Основы проектирования баз данных»

тема «Проектирование баз данных для фитнес-зала»

|  |  |  |  |
| --- | --- | --- | --- |
| Выполнил: | обучающийся группы 4238 | |  |
|  | Каримов Б. И. | |  |
|  | *(ФИО)* | | *(подпись)* |
| Проверил: | преподаватель | Шумилкин А.О. |  |
|  | *(должность)* | *(Ф.И.О.)* | *(подпись)* |
| Курсовая работа защищена с оценкой: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | |
| Дата защиты «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г. | | | |

Казань 2024г.

**АННОТАЦИЯ**

Курсовая работа 55 страниц, 61 рисунок, 8 таблиц, 10 источников литературы.

В данном курсовом проекте будет спроектирована база данных для работников фитнесс клуба.

Для создания базы данных была использована стандартная среда администрирования баз данных выбранной СУБД MS SQL Server – Среда администрирования MS SQL Server Management.

База данных «Фитнесс клуб» включает таблицы, представления, триггеры и хранимые процедуры.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc167220844)

[ГЛАВА 1. ОСНОВНАЯ ЧАСТЬ 4](#_Toc167220845)

[1.1. Концептуальное проектирование 5](#_Toc167220846)

[1.2. Обоснование выбора СУБД 9](#_Toc167220847)

[1.3. Даталогическое проектирование 11](#_Toc167220848)

[1.3.1. Преобразование концептуальной модели в реляционную модель11](#_Toc167220849)

[1.3.2. Нормализация базы данных 12](#_Toc167220850)

[1.4. Создание БД в выбранной СУБД 17](#_Toc167220851)

[1.5. Поддержка целостности данных 18](#_Toc167220852)

[1.5.1. Декларативная поддержка ограничений целостности 18](#_Toc167220853)

[1.5.2. Процедурная поддержка ограничений целостности 30](#_Toc167220854)

[1.6. Реализация операций над данными 46](#_Toc167220855)

[ГЛАВА 2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 48](#_Toc167220856)

[ЗАКЛЮЧЕНИЕ 52](#_Toc167220857)

[СПИСОК ЛИТЕРАТУРЫ 53](#_Toc167220858)

# ВВЕДЕНИЕ

Фитнес клуб представляет собой учреждение, содержащее физические ценности общества и обеспечивающее использование этих ценностей для целей здоровья, спорта и личностного развития. Фитнес клуб играет важную роль в поддержании здоровья, формировании спортивных навыков и достижении физической формы у посетителей.   
 В данном курсовом проекте спроектируем базу данных для работников фитнесс клуба.

**Объектом исследования** является процесс разработки базы данных, а предметом курсовой работы является разработка БД «Фитнес клуб».

**Цель курсовой работы** – создание БД «Фитнес клуб».

Для достижения поставленной цели необходимо выполнить следующие **задачи**:

1. Ознакомиться с литературой по данной теме.
2. Проанализировать предметную область.
3. Построить ER-модель.
4. Построить логическую модель БД.
5. Построить физическую модель БД.
6. Создать объекты БД (таблицы, запросы).
7. Определить условия целостности БД, разработать триггеры и хранимые процедуры.
8. Разработать запросы к базе данных.

БД «Фитнес клуб» облегчит работу сотрудникам фитнес клуба. В частности, позволит улучшить контроль, работу с информацией (о абонементах, членов клуба т.д.), уменьшить время обслуживания членов клуба, увеличить качество обслуживания членов клуба, уменьшит нагрузку на сотрудников фитнес клуба.

# ГЛАВА 1. ОСНОВНАЯ ЧАСТЬ

## Концептуальное проектирование

Фитнес клуб — это учреждение, которое организует занятия спортом, фитнесом и здоровым образом жизни для своих посетителей. Фитнес клубы систематически занимаются проведением тренировок, хранением спортивного оборудования, пропагандой здорового образа жизни и предоставлением доступа к тренировочным программам. Они являются общедоступным местом для занятий спортом и основной базой для улучшения физической формы и самосовершенствования.

Деятельность в фитнес клубе связана с учетом большого количества операций, множество тренировок и посетителей серьезно замедляют работу инструкторов. Сложность поиска нужного тренировочного плана или оборудования занимает длительное время и целиком опирается на компетентность сотрудников клуба.

В фитнес клубе ведется база посетителей. Для ведения тренировочных планов, организации поиска требуемых упражнений и статистики клуба в базе должны храниться сведения, большая часть которых размещаются в аннотированных карточках тренировок.

В клуб приходит много оборудования от различных производителей. Каждому устройству в клубе присваивают номер, а затем размещают на различные зоны. При поступлении оборудования, учитывают такие данные как: номер поступившего оборудования, название устройства, производитель

Предусматривается автоматизация учета пользователей и тренировок, записи новых посетителей, а также хранения информации об имеющемся в наличии оборудовании и их производителях.

Каждый член фитнес клуба имеет следующие параметры: ФИО, электронную почту, номер абонемента

Каждому члену клуба присваивается уникальный номер.

В случае получения оборудования в фитнес клубе остается вкладыш, в котором указана дата получения, дата возврата и номер члена клуба.

При возврате оборудования во вкладыше отмечается срок возврата. При нанесении повреждения оборудованию, член клуба получает предупреждение. Когда предупреждений у члена клуба накапливается больше определенного предела, он лишается правом пользования клубом на определенный срок.

При утере оборудования или воровства член клуба лишается правом пользования клубом на определенный срок, независимо от количества предупреждений.

В фитнес клубе есть ограничения: нельзя пользоваться большим количеством оборудования.

Администрация фитнес клуба должна иметь возможность получать сведения о должниках - членах клуба, которые нарушили правила использования оборудования; сведения о наиболее популярных тренажерах, т.е. наиболее часто используемых тренажерах.

Основная форма документооборота фитнес клуба - карточки учета посетителей и тренировок. Однако использование бумажных документов имеет свои недостатки, такие как возможность порчи, утери или неудобства хранения.

Для решения этих проблем в фитнес клубах все чаще переходят на электронный документооборот. Электронные записи позволяют быстро вводить информацию, легко осуществлять поиск по заданным критериям, корректировать данные, генерировать отчеты и многое другое.

Автоматизация работы фитнес клуба помогает сотрудникам избавиться от рутины и обеспечивает более быстрое, полное и качественное обслуживание клиентов. Работа фитнесс-клуба включает в себя множество повторяющихся задач, таких как:

1. Учет посетителей.

2. Учет тренировок.

3. Продажа абонементов

4. Учет оборудования

В базе данных должны храниться сведения о пользователях фитнес-клуба, тренировках, абонементах, оборудовании и его производителях. Для каждого оборудования в базе данных должны быть данные об производителе, модели, годе выпуска и количестве экземпляров, имеющихся в каждом зале фитнес-клуба, а также идентификационный номер и дата присвоения его клиенту. Сведения о клиентах фитнес клуба должны включать ФИО, электронную почту и номер абонемента. Клиенты могут использовать определенные залы и могут записываться на тренировки или отписываться от них. Фитнес клуб имеет несколько тренировочных залов, которые характеризуются номером, названием и вместимостью. Клуб может получать новое оборудование и списывать старое. Идентификационный номер оборудования может изменяться в результате обновления базы данных.

На основе анализа предметной области были выделены следующие сущности:

­ оборудование – хранение информации о доступном оборудовании в фитнес-клубе;

­ клиенты – хранение данных о клиентах клуба;

­ классификация – классификация оборудования;

­ покупка абонемента – получение пользователем купленного им абонемента;

­ производители – данные о производителях оборудования для тренировок;

­ оборудование в залах - информация об оборудовании в тренировочных залах.

­ пользователи – данные о пользователях, которые имеют доступ к информации о деятельности фитнес-клуба. В таблице 1 представлен список атрибутов каждой выделенной сущности

Таблица 1 – Перечень атрибутов выделенных сущностей

|  |  |  |
| --- | --- | --- |
| Сущность | Атрибут | Описание |
| 1 | 2 | 3 |
| Пользователь | Код пользователя | Уникальный код пользователя |
| ФИО | Фамилия, имя и отчество |
| Электронная почта | Адрес электронной почты |
| Роль | Роль пользователя в системе |
| Код абонемента | Код абонемента купленного пользователем |
| Абонемент | Код абонемента | Уникальный код абонемента |
| Название | Полное название абонемента |
| Описание | Описание входящих услуг |
| Цена | Цена покупки абонемента |
| Тренировки | Код тренировки | Уникальный код тренировки |
| Наименование | Наименование тренировки |
| Описание | Описание тренировки |
| Время | Время, необходимое для тренировки |
| Оборудование | Код оборудования | Уникальный код оборудования |
| Производитель | Имя производителя |
| Модель | Модель оборудования |
| Количество | Количество экземпляров |

На рисунке 1 представлена ER-диаграмма «сущность-связь».

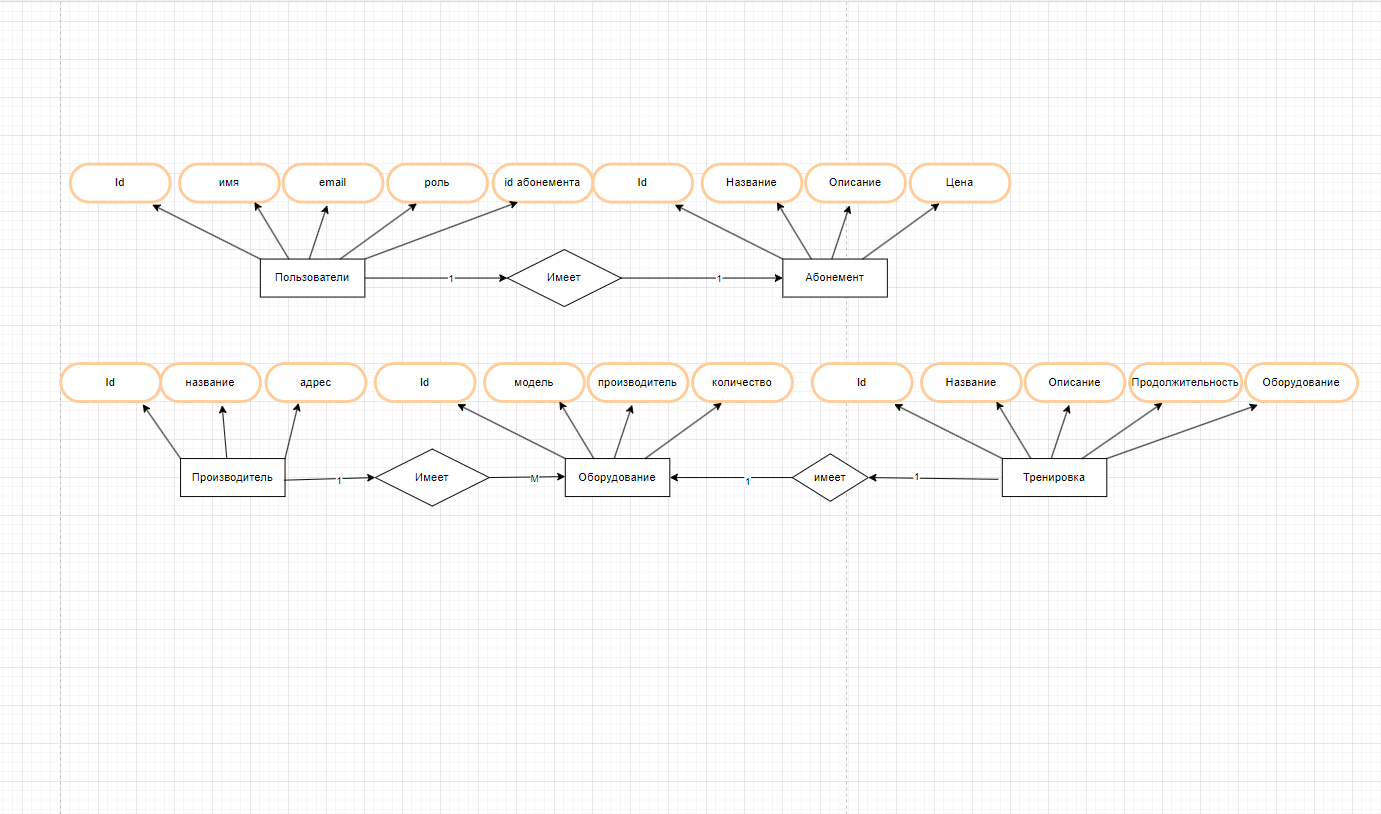


Рисунок 1 - ER-диаграмма «сущность-связь»

Между выделенными сущностями созданы следующие связи:

* 1:М – производитель может иметь много оборудования
* 1:1 – один пользователь может иметь один абонемент;
* 1:1 – одна тренировка может иметь один тренажер;

## 1.2. Обоснование выбора СУБД

*СУБД (система управления базами данных)* – это информационная среда для создания баз данных и управления хранимыми в них сведениями об объектах реального мира.

Выбор СУБД определяется многими факторами, но главный из них – возможность работы с построенной моделью данных. Поэтому одной из важнейших характеристик является тип модели (иерархический, сетевой, реляционный), который поддерживается СУБД. База данных для решения поставленной задачи является реляционной. Помимо модели данных важным показателем является стоимость лицензии для разработки базы данных и стоимость поддержки данной СУБД. Также необходимо обращать внимание на минимальные технические требования для оперативной и эффективной работы СУБД.

В настоящее время на рынке представлен огромный выбор СУБД и каждая СУБД «заточена» под определенный круг задач, которые она может выполнять. Проведем сравнительный анализ СУБД для разработки информационной системы складского учета.

Для сравнения выберем такие СУБД, как:

* MS Access;
* MySQL;
* MS SQL Server.

Сравнительный анализ СУБД представлен в таблице 2.

Таблица 2 - Сравнение СУБД

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии оценки | MS Access | MySQL | MS SQL Server |
| 1 | 2 | 3 | 4 |
| Размер базы данных | | | |
| несколько Мб | + | + |  |
| до 100Мб | + | + |  |
| несколько ГБ |  | + | + |
| 100Гб и более |  |  | + |
| Количество одновременных пользователей, работающих с базой данных напрямую | | | |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| 1 пользователь | + | + |  |
| до 10 | + | + |  |
| более 10 |  | + | + |
| более 100 |  | + | + |
| более 1000 |  |  | + |
| Стоимость лицензии | | | |
| бесплатно (только лицензия на пакет) | + | + |  |
| дешево |  |  |  |
| дорого |  |  | + |
| Платформа на которой способна работать СУБД | | | |
| Windows | + |  | + |
| Unix/Linux |  |  |  |
| Windows+Linux |  | + |  |
| Кластеры |  |  | + |
| Встроенный язык программирования | | | |
| Microsoft | + |  | + |
| Borland | + |  | + |
| Языки для Linux |  |  | + |
| Тип программы | | | |
| Локальное приложение | + |  |  |
| WEB |  | + |  |
| Сложная система |  |  | + |
| Защита данных | | | |
| низкая |  |  |  |
| слабая | + |  |  |
| сильная |  | + | + |
| Мощность языка SQL и возможности СУБД | | | |
| очень слабые |  |  |  |
| слабые | + |  |  |
| развитые |  | + |  |
| мощные |  |  | + |
| Требования к техническим характеристикам ПК | | | |
| неприхотливые | + | + |  |
| чувствительные |  |  |  |
| необходимы мощные сервера |  |  | + |
| Возможные способы доступа к данным | | | |
| ODBC |  |  |  |
| OLEDB/ADO | + | + |  |
| DAO | + |  |  |

Продолжение таблицы 2

|  |  |  |  |
| --- | --- | --- | --- |
| BDE |  |  |  |
| Другое | + | + | + |

Как видно из представленного анализа рассмотренные СУБД могут быть использованы для разработки базы данных автоматизации работы библиотеки. Окончательный выбор был сделан в пользу MS SQL Server, так как данная СУБД обладает высокими показателями производительности, способна хранить большие объемы информации без потери быстродействия при реализации доступа к данным из внешних источников.

## 1.3. Даталогическое проектирование

### 1.3.1. Преобразование концептуальной модели в реляционную модель

При переходе от концептуальной модели предметной области к схеме БД (например, реляционной) используются три различных подхода, которые подробно описаны в списке использованных источников.

При первом подходе преобразование осуществляется вручную.

Второй подход основан на автоматизированной компиляции концептуальной модели предметной области в схему БД, в результате которой создается реляционная база данных в третьей нормальной форме.

В рамках курсового проекта будем использовать первый подход. При этом сущности модели «сущность-связь» соответствует отношение (таблица) реляционной БД. Связи между сущностями типа «один – к – одному (1:1)» и «один – ко – многим (1:М)» соответствуют связям между отношениями. Атрибуты, которые идентифицируют, определяют или моделируют сущности в модели «сущность-связь» соответствуют атрибутам отношений.

На рисунке 2 представлена схема функциональных связей.

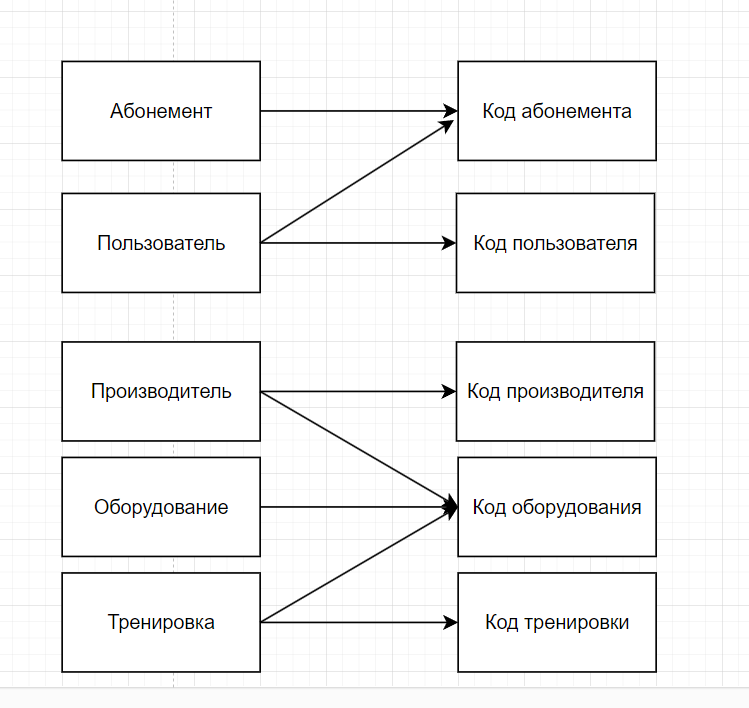


Рисунок 2 - Схема функциональных связей

При разработке реляционной базы данных для фитнес-клуба были проанализированы функциональные зависимости между атрибутами всех отношений и выбраны идентифицирующие атрибуты, которые в реляционной модели данных используются в качестве первичных ключей:

- Код абонемента;

- Код пользователя;

- Код тренировки;

- Код оборудования;

- Код производителя

### 1.3.2. Нормализация базы данных

Изначально разрабатываемая БД представляла собой логически не организованную структуру данных, представленную в таблице 3.

Таблица 3 - Ненормализованная база данных

|  |
| --- |
| Код пользователя |
| ФИО пользователя |
| Email пользователя |
| Роль пользователя |
| Код абонемента пользователя |
| Код абонемента |
| Наименование абонемента |
| Описание абонемента |
| Стоимость |
| Код оборудования |
| Модель |
| Производитель |
| Количество |
| Код производителя |
| Наименование производителя |
| Адрес производителя |
| Код тренировки |
| Название тренировки |
| Описание тренировки |
| Продолжительность тренировки |
| Оборудование для тренировки |

Все рассматриваемые отношения в реляционном подходе находятся в 1НФ, которые предполагают, что элементы доменов отношений не является множествами (атомарными).

**1 НФ.** На этом этапе выделены таблицы «Билеты», «Артисты», «Животные», «Покупатели», «Представление», «Номер». Данные сущности представлены в таблице 4.

Таблица 4 - 1НФ базы данных

|  |  |  |  |
| --- | --- | --- | --- |
| Тренировки | Абонеме нты | Пользователи | Оборудование |
| Код тренировки | Код абонемента | Код пользователя | Код оборудования |
| Описание тренировки | Описание абонемента | ФИО | Модель |
| Длительность тренировки | Стоимость | Email | Производитель |
|  |  | Почта |  |

Отношения находятся во 2НФ, если отношения находятся в 1НФ, и каждый не первичный атрибут формально полно зависит от каждого возможного ключа.

**2 НФ.** Добавлена таблица «Производитель», что продемонстрированно в таблице 5. Некоторые столбцы разделены на несколько атрибутов.

Таблица 5- 2НФ базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тренировки | Абонеме нты | Пользователи | Оборудование | Производител |
| Код тренировки | Код абонемента | Код пользователя | Код оборудования | Код производителя |
| Наименование | Наименование | ФИО | Модель | Наименование |
| Описание тренировки | Описание | Email | Производитель | Адрес |
| Длительность | Стоимость | Код абонемента |  |  |
| Оборудование |  | Роль |  |  |

Отношения находятся в 3НФ, если оно находиться в 2НФ и в нем нет зависимости атрибутов от возможных ключей (устранение данных, не зависящих от первичного ключа).

**3НФ.** На этом этапе выделено еще одна таблица для связи представлений и номера. (таблица 6). Таким образом, БД «Фитнес-клуб» приведена к 3НФ.

Таблица 6 - 3НФ базы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Тренировки | Абонеме нты | Пользователи | Оборудование | Производител |
| Код тренировки | Код абонемента | Код пользователя | Код оборудования | Код производителя |
| Наименование | Наименование | ФИО | Модель | Наименование |
| Описание тренировки | Описание | Email | Производитель | Адрес |
| Длительность | Стоимость | Код абонемента |  |  |
| Оборудование |  | Роль |  |  |

На рисунке 3 представлена логическая модель базы данных, отражающая связи таблиц БД, разработанная с помощью Draw Io.

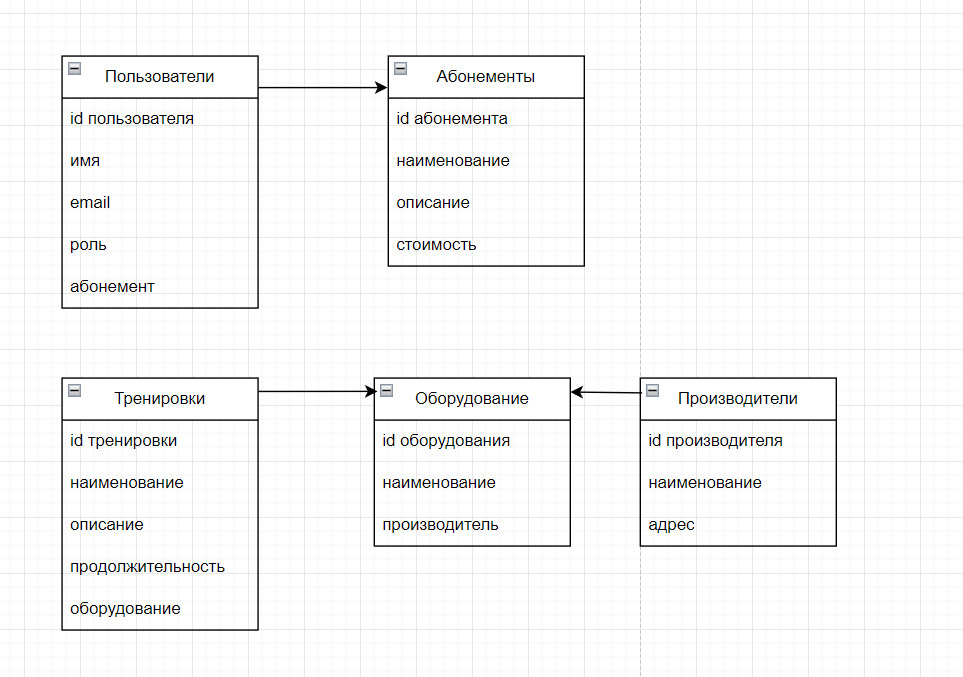


Рисунок 3 – Логическая модель базы данных

**1.3.3. Определение характеристик атрибутов**

Описание даталогической модели БД представлено в таблице 7.

Таблица 7 – Описание даталогической модели БД

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Таблица | Поле | Тип поля | Размер поля | Допустимость  неопределенных значений |
| Пользователи | Код\_пользователя | Счетчик | - | NOT NULL |
| ФИО\_пользователя | Текстовый | 100 | NOT NULL |
| Email | Текстовый | 100 | NOT NULL |
| Код\_абонемента | Числовой | - | - |
| Роль | Текстовый | 20 | NOT NULL |
| Абонементы | Код\_абонемента | Счетчик | - | NOT NULL |
| Наименование | Текстовый | 50 | NOT NULL |
| Описание | Текстовый | 200 | NOT NULL |
| Стоимость | Числовой |  | NOT NULL |
| Тренировки | Код\_тренировки | Счетчик | - | NOT NULL |
| Наименование | Текстовый | 50 | NOT NULL |
| Описание | Текстовый | 256 | NOT NULL |
| Продолжительность | Текстовый | 50 | NOT NULL |
| Оборудование | Числовой |  | NOT NULL |
| Оборудование | Код\_оборудования | Счетчик | - | NOT NULL |
| Наименование | Текстовый | 100 | NOT NULL |
| Модель | Текстовый | 100 | NOT NULL |
| Производитель | Текстовый | 100 | NOT NULL |
| Производитель | Код\_производителя | Числовой |  | NOT NULL |
| Наименование | Текстовый | 255 | NOT NULL |
| Адрес | Текстовый | 255 | NOT NULL |

На рисунке 4 представлена даталогическая модель БД .

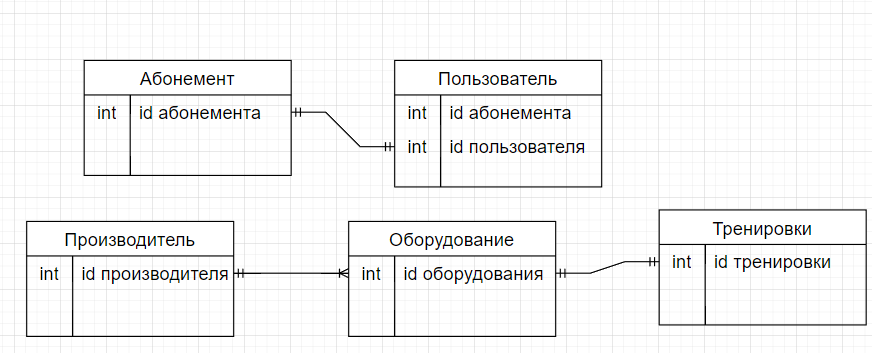


Рисунок 4 – Даталогическая модель БД

## 1.4. Создание БД в выбранной СУБД

Для создания базы данных была использована стандартная среда администрирования баз данных выбранной СУБД MS SQL Server – Среда администрирования MS SQL Server Management.

На рисунках 5-12 представлена структура созданных таблиц БД.

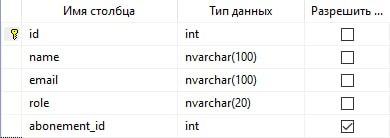


Рисунок 5 – Структура таблицы «Пользователь»

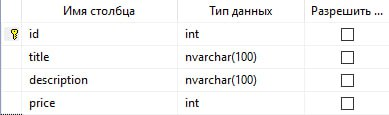


Рисунок 6 – Структура таблицы «Абонемент»

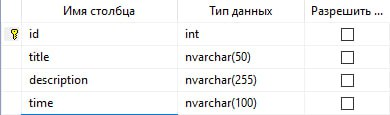


Рисунок 7 – Структура таблицы «Тренировка»

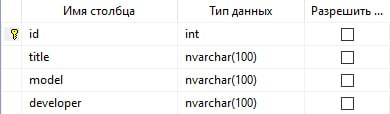


Рисунок 8 – Структура таблицы «Оборудование»

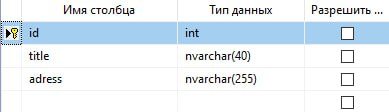


Рисунок 9 – Структура таблицы «Производитель»

На рисунке 10 представлена модель базы данных, отражающая индексы и связи таблиц БД.

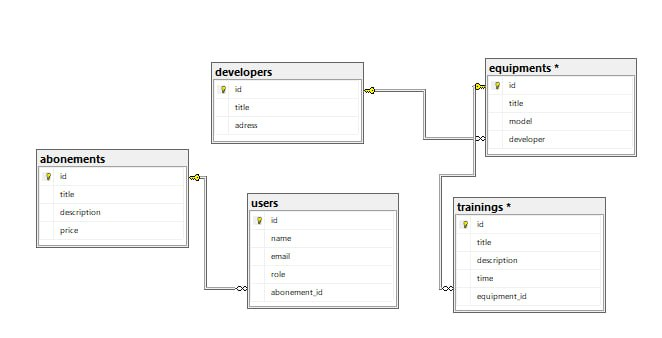


Рисунок 10 – Модель связей таблиц базы данных

## 1.5. Поддержка целостности данных

### 1.5.1. Декларативная поддержка ограничений целостности

Обеспечение целостности базы данных означает выполнение ряда ограничений, соблюдение которых необходимо для поддержания непротиворечивости хранимых данных. Среди ограничений целостности можно выделить ограничения диапазонов значений атрибутов отношений и структурные ограничения на кортежи отношений.

Первый тип ограничений предполагает контроль значений атрибутов отношений.

Структурные отношения фиксируют два требования целостности, которые должны поддерживаться реляционными СУБД: требование целостности сущностей и целостности ссылок. Каждому экземпляру сущности, представленному в отношении, соответствует только один его кортеж. Первое требование состоит в том, что любой кортеж отношения должен быть отличим от любого другого кортежа этого отношения, иными словами, любое отношение должно обладать первичным ключом.

Формулировка второго требования тесно связана с понятием внешнего ключа. Требование целостности по ссылкам состоит в том, что для каждого значения внешнего ключа родительской таблицы должна найтись строка в дочерней таблице с таким же значением первичного ключа.

В MS SQL Server поддержка ссылочной целостности выполняется с помощью настройки связей таблиц, например, в диаграмме связей*.*

В данном курсовом проекте ссылочная целостность формировалась с помощью настройки связей таблиц так, как показано на рисунке 11.

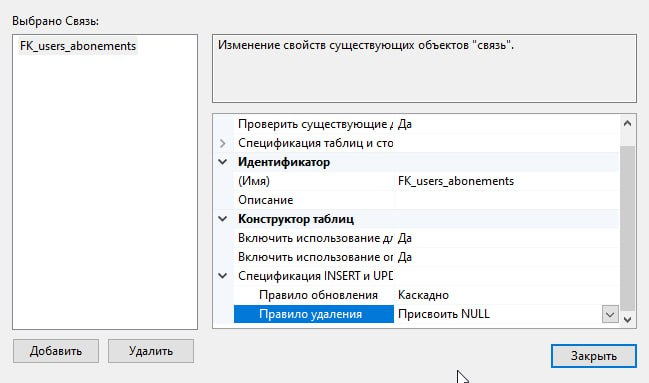


Рисунок 11 – Настройка связей таблиц БД для обеспечения целостности

Запросы добавления данных в таблицу представлены на рисунках 12-16.

Запросы изменения данных в таблице представлены на рисунках 17-21.

Запросы удаления данных из таблицы представлены на рисунках 22-26.

Листинг - 1

USE sport;

INSERT INTO users VALUES

('Владислав', 'busovrm4@gmail.com', 'user', 2),

('Александр', 'shum@yandex.ru', 'user', 2),

('Кристина', 'ivanovakristi0609@mail.ru', 'user', 3);

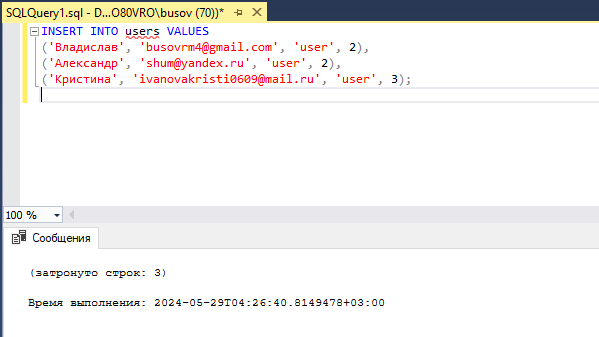


Рисунок 15 – Результат запроса на добавление новых полей «Пользователи»

Листинг - 2

USE sport;

INSERT INTO developers VALUES

('ProRun', 'Город Москва'),

('fiftyTors', 'Город Стамбул'),

('SevenOt', 'Город Казань'),

('TriplWors', 'Город Марий Эл'),

('Shumao', 'Город Казань');

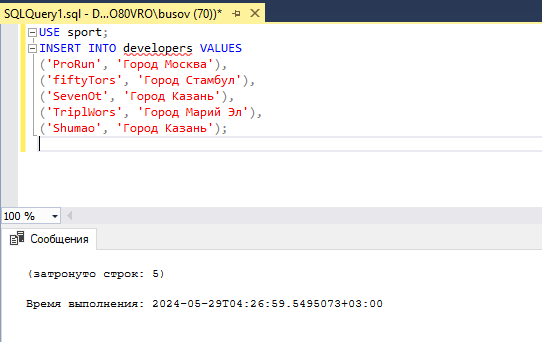


Рисунок 16 – Результат запроса на добавление новых полей «Производители»

Листинг - 3

USE sport;

INSERT INTO equipments VALUES

('Беговая дорожка', 'Walk 5000', 1),

('Гантели 5кг', 'GiGant5', 4),

('Шведская стенка', 'Shwed300', 2),

('Стойка для грифа', 'ST840', 3),

('Стенд', 'Gft94', 1);

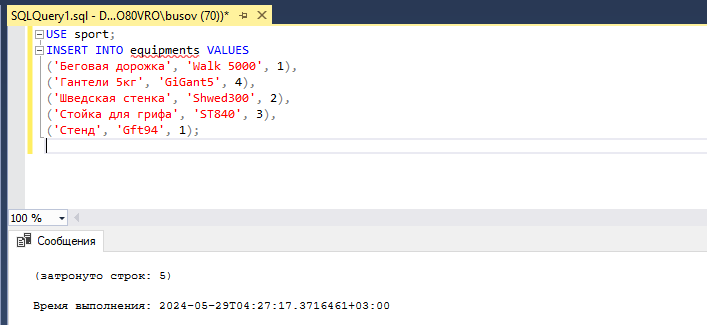


Рисунок 17 – Результат запроса на добавление новых полей «Оборудование»

Листинг - 4

USE sport;

INSERT INTO trainings VALUES

('Дельты', 'Подтягивания на шведской стенке', '30 раз за 5 подходов', 3),

('Бицепс', 'Бастрое разгибание и медленное сгибание руки с гантелей', '20 раз', 4),

('Трицепс', 'Быстрое сгибание и медленное разгибание руки с гантелей', '20 раз', 5),

('Икроножные', 'Бег на беговой дорожке со скоростью 10км\ч', '10 минут', 2),

('Широчайшие', 'При помощи гантелей необходимо разводить руки в разные стороны и сводить с периодичностью 2 секунды', '2 минуты', 1);

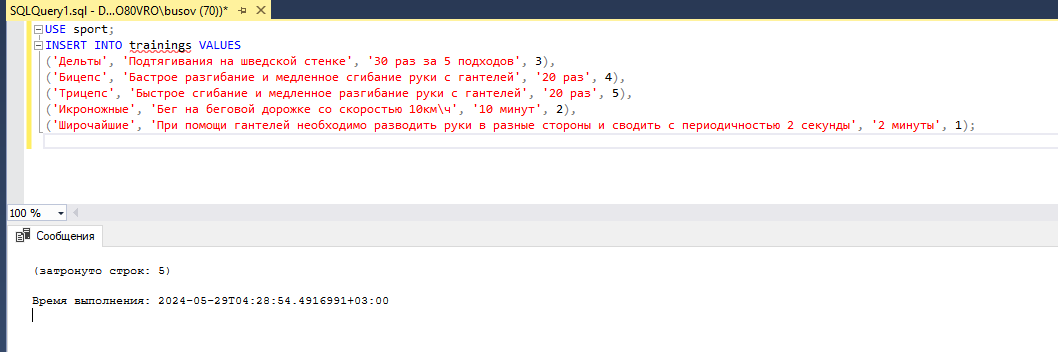


Рисунок 18 – Результат запроса на добавление новых полей «Тренировки»

Листинг - 5

USE sport;

INSERT INTO abonements

VALUES

(‘Бронзовый’, ‘Абонемент на месяц’, 2500),

(‘Серебряный’, ‘Абонемент на три месяца’, 5000),

(‘Платиновый’, ‘Абонемент на шесть месяцев’, 8500),

(‘Золотой’, ‘Абонемент на год’, 12000),

(‘Рубиновый’, ‘Абонемент на два года’, 15000);

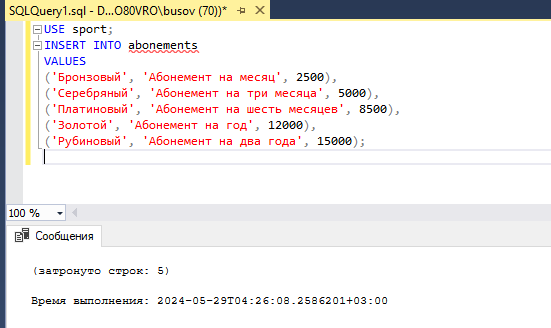


Рисунок 19 – Результат запроса на добавление новых полей «Абонементы»

Листинг - 6

USE sport;

SELECT \*

FROM users

WHERE

abonement\_id = 3;

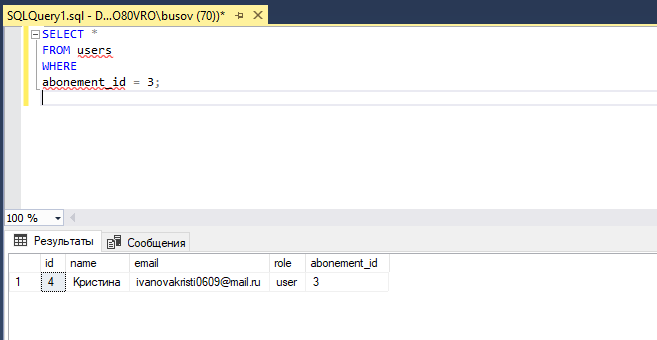


Рисунок 20 – Результат запроса на получение полей в таблице «Пользователи»

Листинг - 7

USE sport;

SELECT \*

FROM equipments

WHERE

Developer\_id = 2;

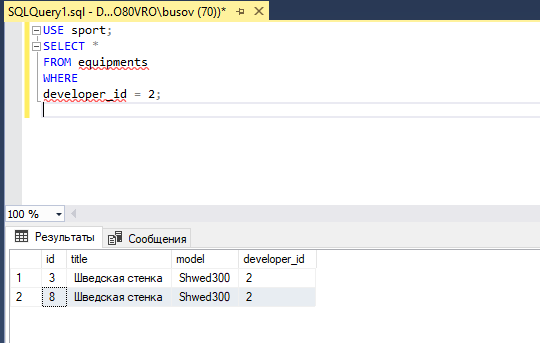


Рисунок 21 – Результат запроса на получение полей в таблице «Оборудование»

Листинг - 8

USE sport;

SELECT \*

FROM trainings

WHERE

equipment\_id = 1;

SELECT developer

FROM equipments

WHERE

developer = 1;

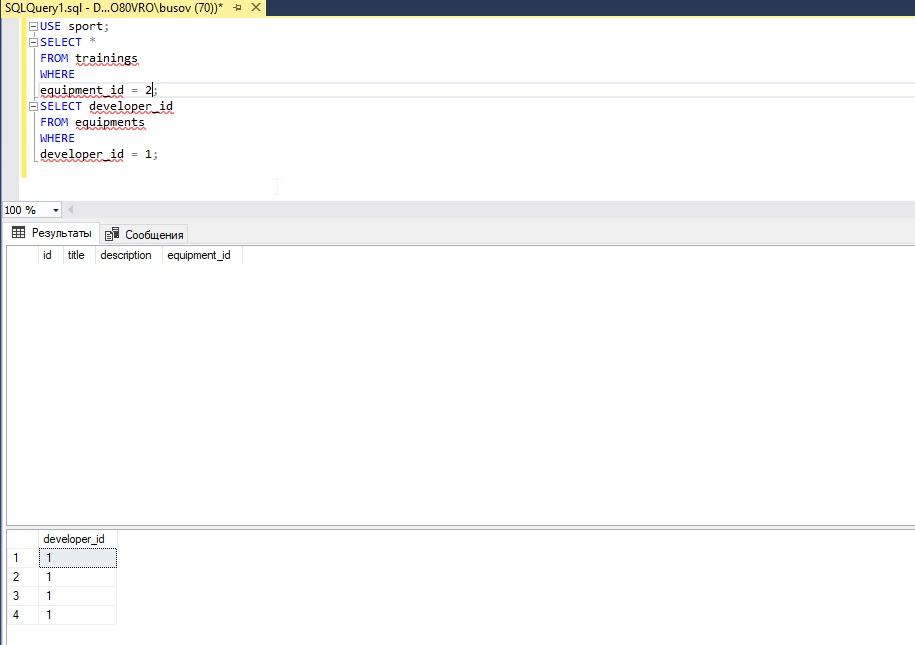


Рисунок 22 – Результат запроса на получение полей в таблице «Тренировки»

Листинг 9

USE sport;

UPDATE users

SET email = 'test@test.com'

WHERE

id = 3;



Рисунок 23 – Результат запроса на изменение поля в таблице «Пользователи»

Листинг - 10

USE sport;

UPDATE equipments

SET developer = 4

WHERE

title = 'Беговая дорожка';

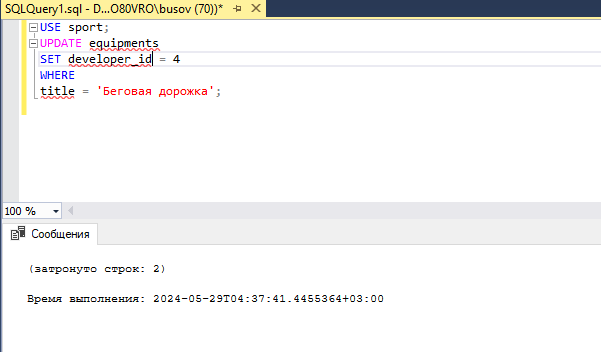


Рисунок 24 – Результат запроса на изменение поля в таблице «Оборудование»

Листинг - 11

USE sport;

DELETE

FROM trainings

WHERE

id = 5;

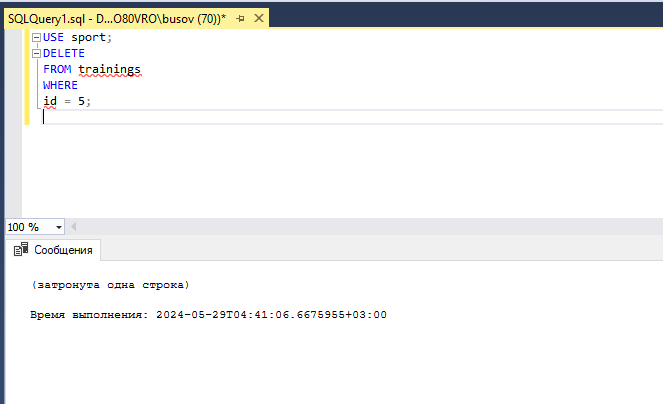


Рисунок 25 - Результат запроса на удаление полей из таблицы «Тренировки»

Листинг - 12

USE sport;

SELECT

trainings.id AS tr\_id,

trainings.title AS tr\_title,

trainings.description AS tr\_desc,

equipments.id as eq\_id,

equipments.title as eq\_title,

equipments.developer as eq\_dev

FROM trainings JOIN equipments ON trainings.equipment\_id = equipments.id;

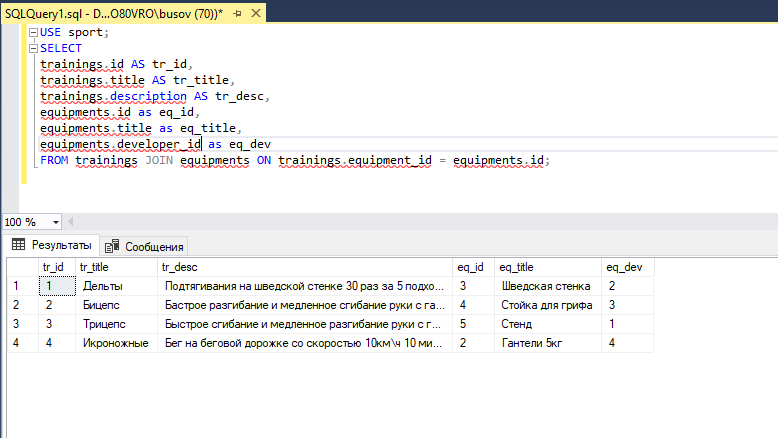


Рисунок 26 - Результат запроса на получение полей из таблицы «Оборудование»

Листинг - 13

USE sport;

DELETE FROM

developers

WHERE

id = 5;

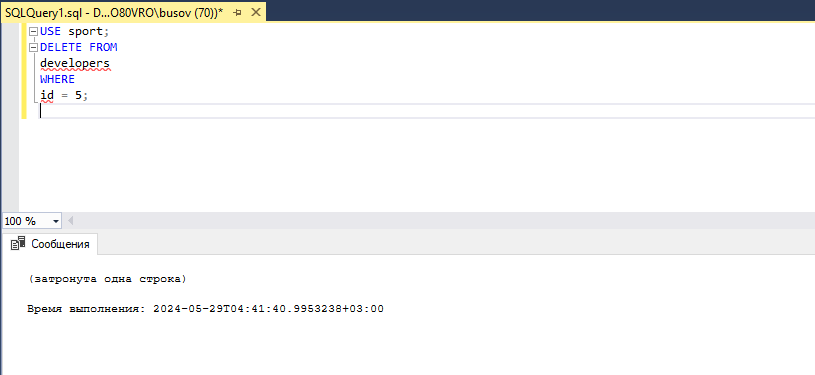


Рисунок 27 - Результат запроса на удаление полей из таблицы «Производители»

Листинг - 14

USE sport;

SELECT \*

FROM users

JOIN abonements ON

users.abonement\_id = abonements.id;

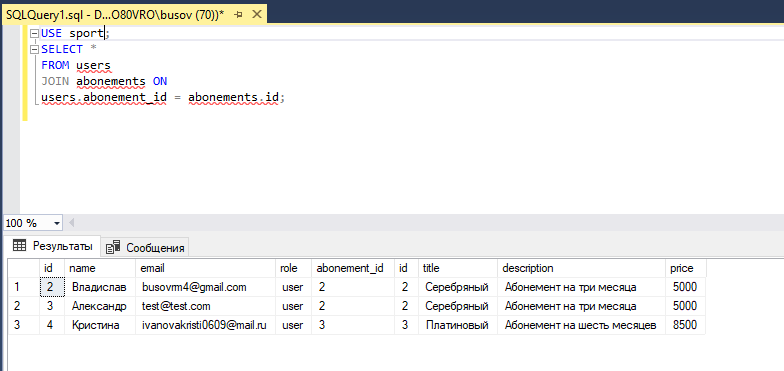


Рисунок 28 - Результат запроса на получение полей из таблицы «Пользователи»

Листинг - 15

USE sport;

UPDATE users

SET

abonement\_id = 3

WHERE

id = 2;

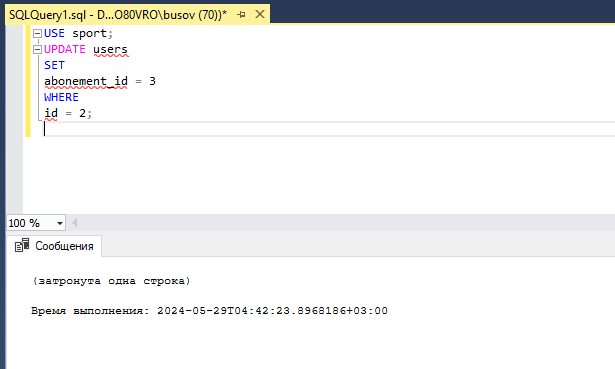


Рисунок 29 - Результат запроса на изменение полей в таблице «Пользователи»

### 1.5.2. Процедурная поддержка ограничений целостности

Триггеры представляют специальный тип хранимой процедуры, которая вызывается автоматически при выполнении определенного действия над таблицей или представлением, в частности, при добавлении, изменении или удалении данных, то есть при выполнении команд INSERT, UPDATE, DELETE.

Хранимые процедуры представляют набор инструкций, которые выполняются как единое целое. Тем самым хранимые процедуры позволяют упростить комплексные операции и вынести их в единый объект.

Также хранимые процедуры позволяют ограничить доступ к данным в таблицах и тем самым уменьшить вероятность преднамеренных или неосознанных нежелательных действий в отношении этих данных.

И еще один важный аспект - производительность. Хранимые процедуры обычно выполняются быстрее, чем обычные SQL-инструкции. Все потому что код процедур компилируется один раз при первом ее запуске, а затем сохраняется в скомпилированной форме.

**Транзакция** – это команда или блок команд (инструкций), которые успешно завершаются как единое целое, при этом в базе данных все внесенные изменения фиксируются на постоянной основе, или отменяются, т.е. все изменения, внесенные любой командой, входящей в транзакцию, будут отменены. Другими словами, если одна команда или инструкция внутри транзакции завершилась с ошибкой, то все, что было отработано перед ней, также отменяется, даже если предыдущие команды завершились успешно.

Транзакции очень полезны и просто незаменимы в тех случаях, когда Вам необходимо реализовывать логику в базе данных Microsoft SQL Server, которая предполагает многошаговые операции.

Для организации процедурной поддержки ограничения целостности данных разработаны хранимые процедуры и триггеры. А также несколько транзакций.

Триггер на уведомление после попытку получить несуществующий абонемент представлен на рисунке 30.

Листинг - 16

|  |
| --- |
| CREATE or alter TRIGGER Целостность1  ON users  AFTER INSERT, UPDATE  AS  BEGIN  Declare @ab as int  SELECT @ab = i.id FROM INSERTED i  if @ab not in SELECT id FROM abonements  BEGIN  ROLLBACK TRAN  PRINT N'Такого абонемента не существует'  END  END  Go |

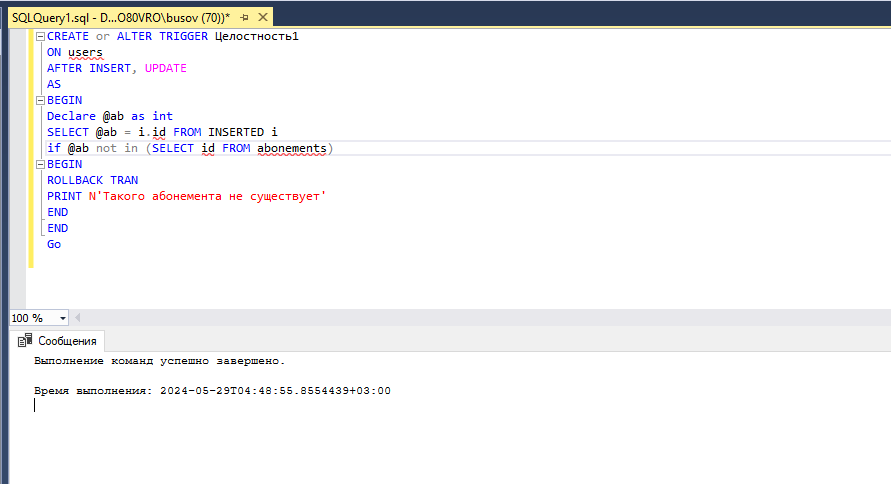


Рисунок 30 – «Триггер 1»

Триггер на обновление внешнего ключа пользователя при удалении абонемента представлен на рисунке 31.

Листинг - 17

CREATE or alter TRIGGER Целостность2

ON abonements

AFTER DELETE

AS

BEGIN

Declare @ab as int

SELECT @ab = i.id FROM DELETED i

BEGIN

UPDATE users SET abonement\_id = NULL WHERE abonement\_id = @ab

END

END

Go

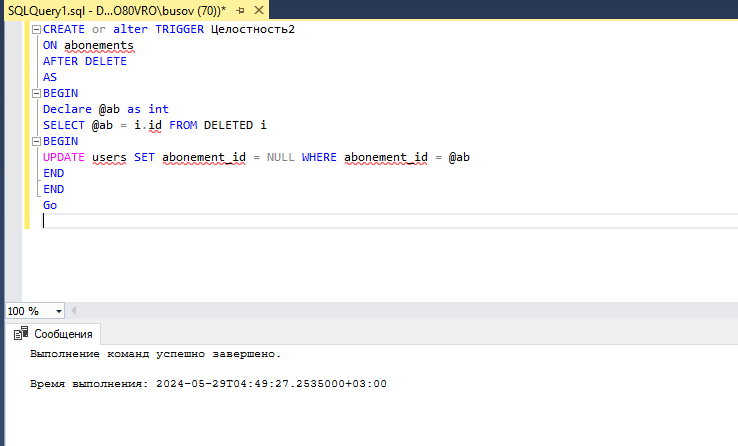


Рисунок 31 – «Триггер 2»

Триггер для удаления оборудования при удалении поставщика представлен на рисунке 32.

Листинг - 18

CREATE or ALTER TRIGGER Целостность3

ON equipments

AFTER DELETE

AS

BEGIN

Declare @dev as int

SELECT @dev=id FROM DELETED i

BEGIN

DELETE FROM equipments WHERE developer = @dev;

END

END

Go

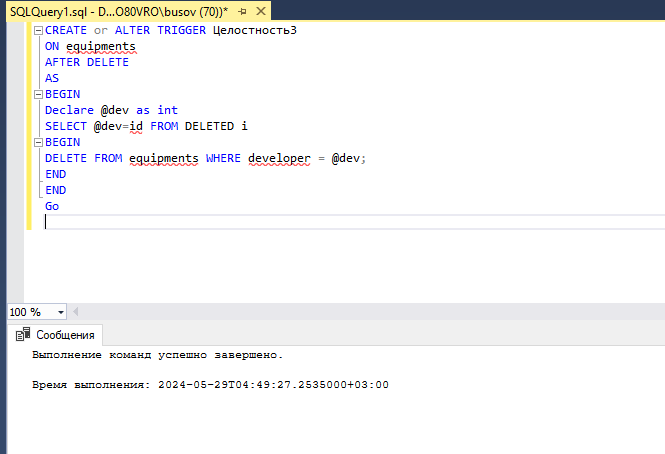


Рисунок 32 – Триггер 3

Триггер для уведомления и отката изменений при попытке обращения к несуществующему первичному ключу представлен на рисунке 33.

Листинг - 19

CREATE or ALTER TRIGGER Целостность4

ON equipments

AFTER INSERT, UPDATE AS

BEGIN

Declare @dev as int

SELECT @dev=i.developer FROM INSERTED i

if @dev not in (SELECT id FROM developers)

BEGIN

ROLLBACK TRAN

PRINT N'Неизвестный код производителя'

END

END

Go

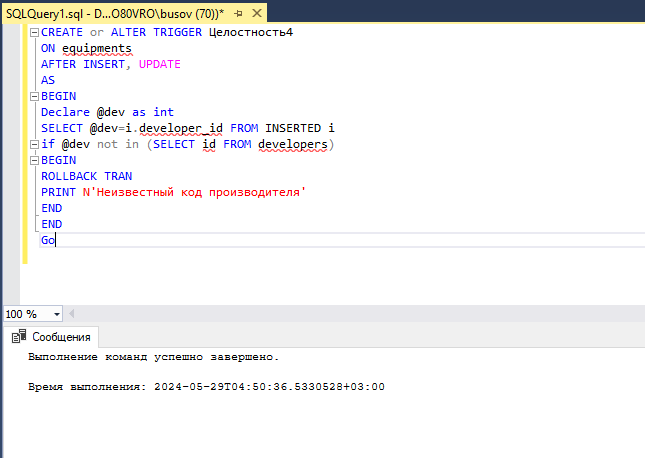


Рисунок 33 – Триггер 4

Триггер для проверки корректности данных при добавлении или обновлении записей в таблицу представления представлен на рисунке 34.

Листинг - 20

CREATE or ALTER TRIGGER Целостность5

ON equipments

AFTER DELETE

AS

BEGIN

Declare

@eq as int

SELECT

@eq = i.id FROM DELETED i

UPDATE trainings SET equipment\_id = NULL WHERE equipment\_id = @eq

END

Go

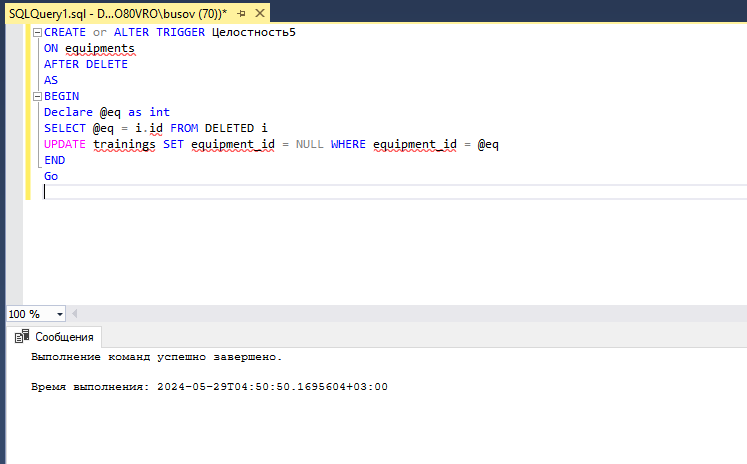


Рисунок 34 – Триггер 5

Также разработано несколько хранимых процедур. Процедура на показ оборудования по определенному первичному ключу представлена на рисунке 35.

Листинг - 21

CREATE PROCEDURE ТренажерДляТренировки

@train\_id as int

AS

BEGIN

SELECT \* FROM equipments WHERE id = (SELECT equipment\_id FROM trainings WHERE id = @train\_id)

END

GO

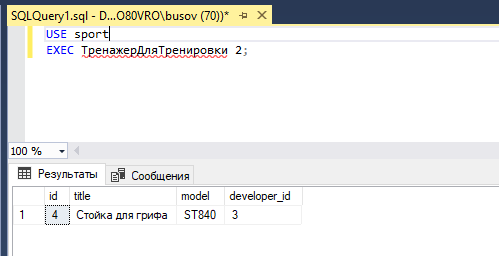


Рисунок 35 – Процедура 1

Хранимая процедура, которая показывает список клиентов с заданным абонементом представлена на рисунке 36.

Листинг - 22

CREATE PROCEDURE АбонементПользователя

@user\_id as int

AS

BEGIN

SELECT \* FROM abonements WHERE id = (SELECT abonement\_id FROM users WHERE id = @user\_id)

END

GO

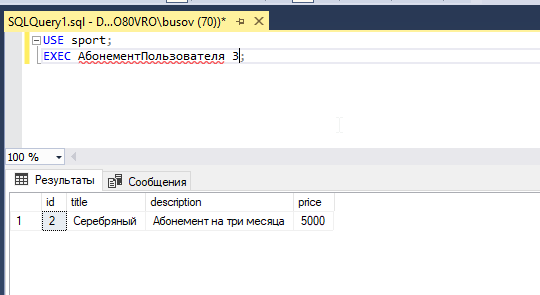


Рисунок 36 – Процедура 2

Хранимая процедура для добавления программы тренировки представлена на рисунке 37.

Листинг - 23

CREATE PROCEDURE ДобавлениеТренировки

@title as nvarchar(100),

@description as nvarchar(100),

@equip\_id as int

AS

BEGIN

INSERT INTO trainings(title, description, equipment\_id) VALUES (@title, @description, @equip\_id)

END

Go

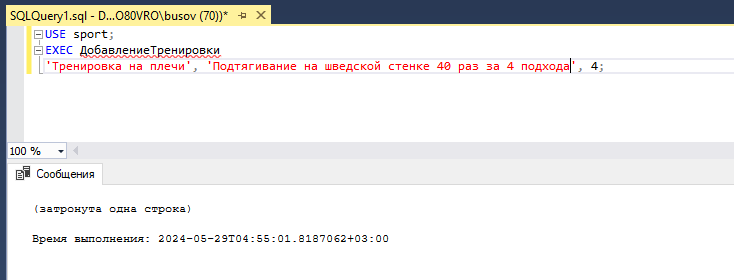


Рисунок 37 – Процедура 3

Хранимая процедура для добавления нового пользователя представлена на рисунке 38.

Листинг - 24

CREATE PROCEDURE ДобавлениеПользователя

@name as nvarchar(255),

@email as nvarchar(100)

AS

BEGIN

INSERT INTO users(name, email, role, abonement\_id) VALUES(@name, @email, 'User', NULL)

END

Go

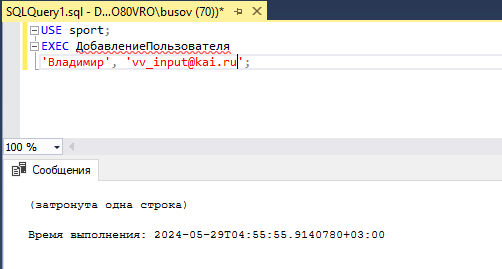


Рисунок 38 – Процедура 4

Хранимая процедура для получения производителей оборудования представлена на рисунке 39.

Листинг - 25

CREATE PROCEDURE ПроизводительОборудования

@equip\_id as int

AS

BEGIN

SELECT \* FROM developers WHERE id = (SELECT developer\_id FROM equipments WHERE id = @equip\_id)

END

Go

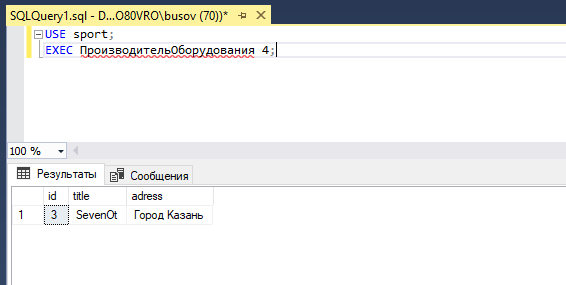


Рисунок 39 – Процедура 5

Представление для вывода информации о всех посетителях, которые купили абонементы, показано на рисунке 40.

Листинг – 26

CREATE or ALTER VIEW v1 AS

SELECT \* FROM users WHERE abonement\_id != NULL;

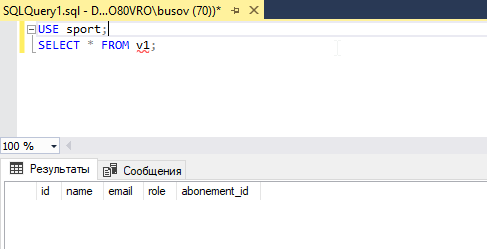


Рисунок 40 – Представление 1

Представление для вывода информации о тренировке и оборудовании показано на рисунке 41.

Листинг – 27

CREATE or ALTER VIEW v2 AS

SELECT

trainings.id AS tr\_id,

trainings.title AS tr\_title,

trainings.description AS tr\_desc,

equipments.id as eq\_id,

equipments.title as eq\_title,

equipments.developer as eq\_dev

FROM trainings JOIN equipments ON trainings.equipment\_id = equipments.id;

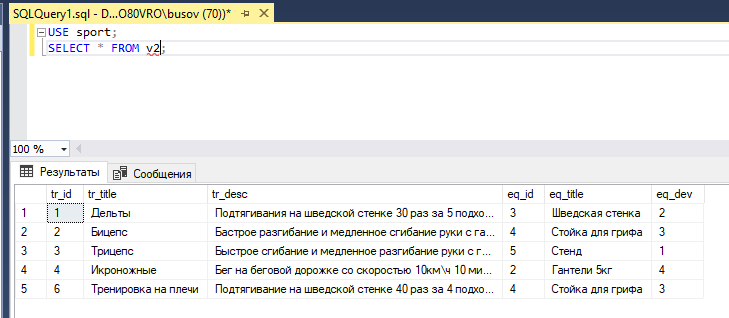


Рисунок 41 – Представление 41

Представление для вывода информации о всем оборудовании и его производителях показано на рисунке 42.

Листинг – 28

CREATE or ALTER VIEW v3 AS

SELECT

equipments.title as eq\_title,

equipments.model as eq\_model,

equipments.id as eq\_id,

developers.id as dev\_id,

developers.title as dev\_title,

developers.adress as dev\_adress

FROM equipments JOIN developers ON equipments.developer = developers.id;

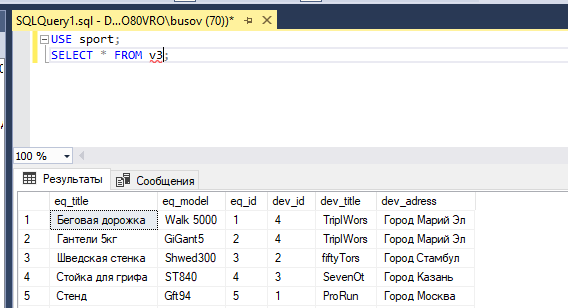


Рисунок 42 – Представление 3

Представление для вывода информации о всех пользователях и их абонементах показано на рисунке 43.

Листинг – 29

CREATE or ALTER VIEW v4 AS

SELECT

users.name as us\_name,

users.email as us\_email,

users.id as us\_id,

abonements.id as ab\_id,

abonements.title as ab\_title,

abonements.description as ab\_desc,

abonements.price as ab\_price

FROM users JOIN abonements ON users.abonement\_id = abonements.id;

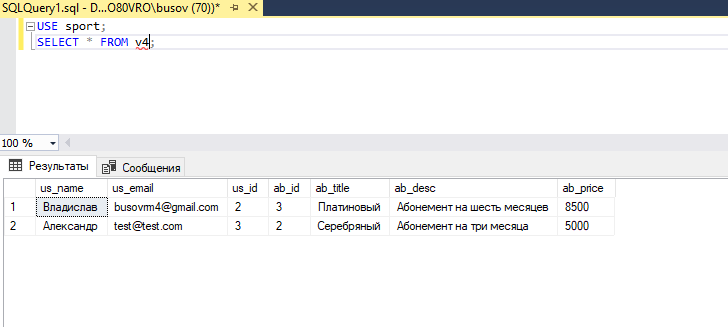


Рисунок 43 – Представление 4

Представление для вывода информации о всех производителях показано на рисунке 44.

Листинг - 30

CREATE or ALTER VIEW v5 AS

SELECT \* FROM developers;

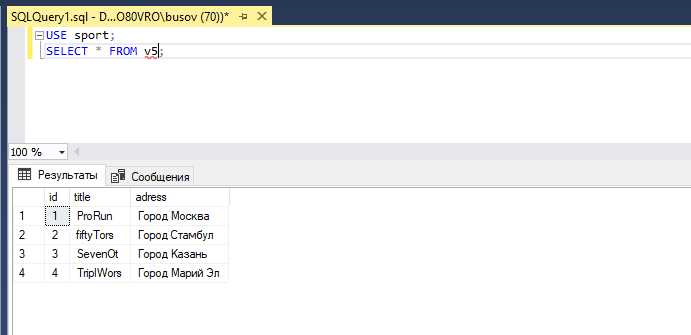


Рисунок 44 – Представление 5

Транзакция для обновления названия оборудования и оборудования для тренировки представлена на рисунке 45.

Листинг - 31

BEGIN TRANSACTION

UPDATE equipments SET title = 'Беговая дорожка' WHERE id = 2;

UPDATE trainings SET equipment\_id = 2 WHERE title = 'Бег';

COMMIT TRANSACTION

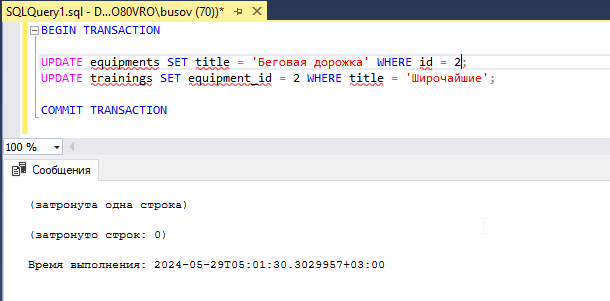


Рисунок 45 – Транзакция 1

Транзакция для обновления почты пользователя в таблице представлена на рисунке 46.

Листинг - 32

BEGIN TRANSACTION

SELECT \* FROM users;

UPDATE users SET email = 'test@test.test' WHERE id = 1;

SELECT \* FROM users;

COMMIT TRANSACTION

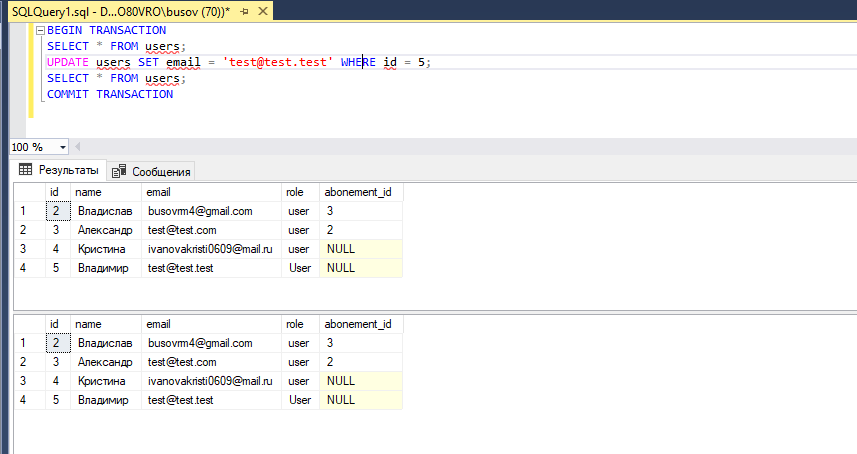


Рисунок 46 – Транзакция 2

Транзакция на обновление данных в абонемента показана на рисунке 47.

Листинг – 33

BEGIN TRANSACTION

UPDATE abonements SET title = 'new abonement' WHERE id = 1;

SELECT \* FROM abonements;

COMMIT TRANSACTION

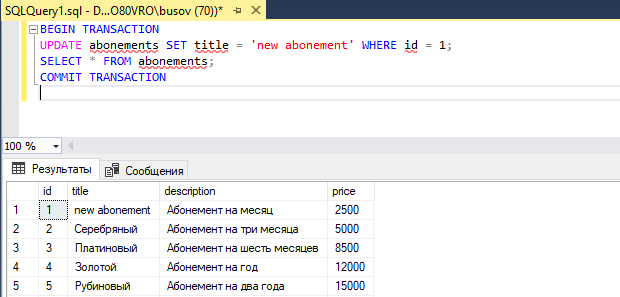


Рисунок 47 – Транзакция 3

## 1.6. Реализация операций над данными

В информационной системе учета работы фитнес-клуба созданы следующие запросы:

1. Запрос на общее количество абонементов, купленных пользователями. Функционал данного запроса представлен на рисунке 48.

Листинг - 34

SELECT COUNT(\*)

FROM users

WHERE

abonement\_id IS NOT NULL;

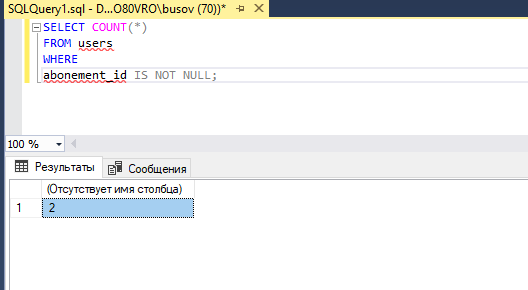


Рисунок 48 – Запрос в SQL

2. Запроса на показ типа тренировки и оборудования, необходимого для нее, представлен на рисунке 49.

Листинг - 35

SELECT \*

FROM trainings

JOIN equipments

ON

trainings.equipment\_id = equipments.id;

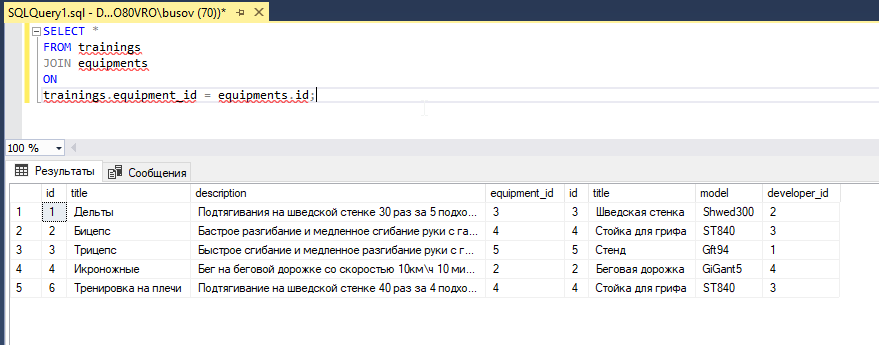


Рисунок 49 – Запрос в SQL

3. Найти абонемент, купленный максимальное количество раз. Функционал данного запроса представлен на рисунке 50.

Листинг - 36

SELECT title

FROM abonements

WHERE

id = (SELECT MAX(abonement\_id) FROM users);

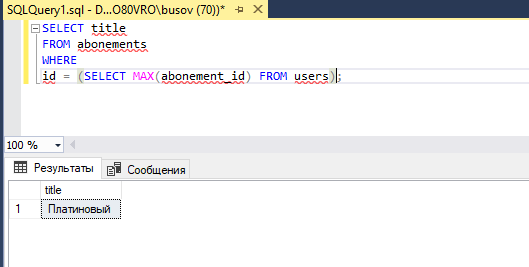


Рисунок 50 – Запрос в SQL

# ГЛАВА 2. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Данное приложение было воплощено в жизнь с помощью Visual Studio, используя шаблон ASP.NET CORE для создания пользовательского интерфейса. Для эффективной работы с базой данных был внедрен NuGet пакет Entity Framework Core и Entity Framework SQL Server, который обеспечил удобство и надежность взаимодействия с данными.

При открытии сайта отображается главный экран с покупки абонемента. Главная страница представлена на рисунке 53.

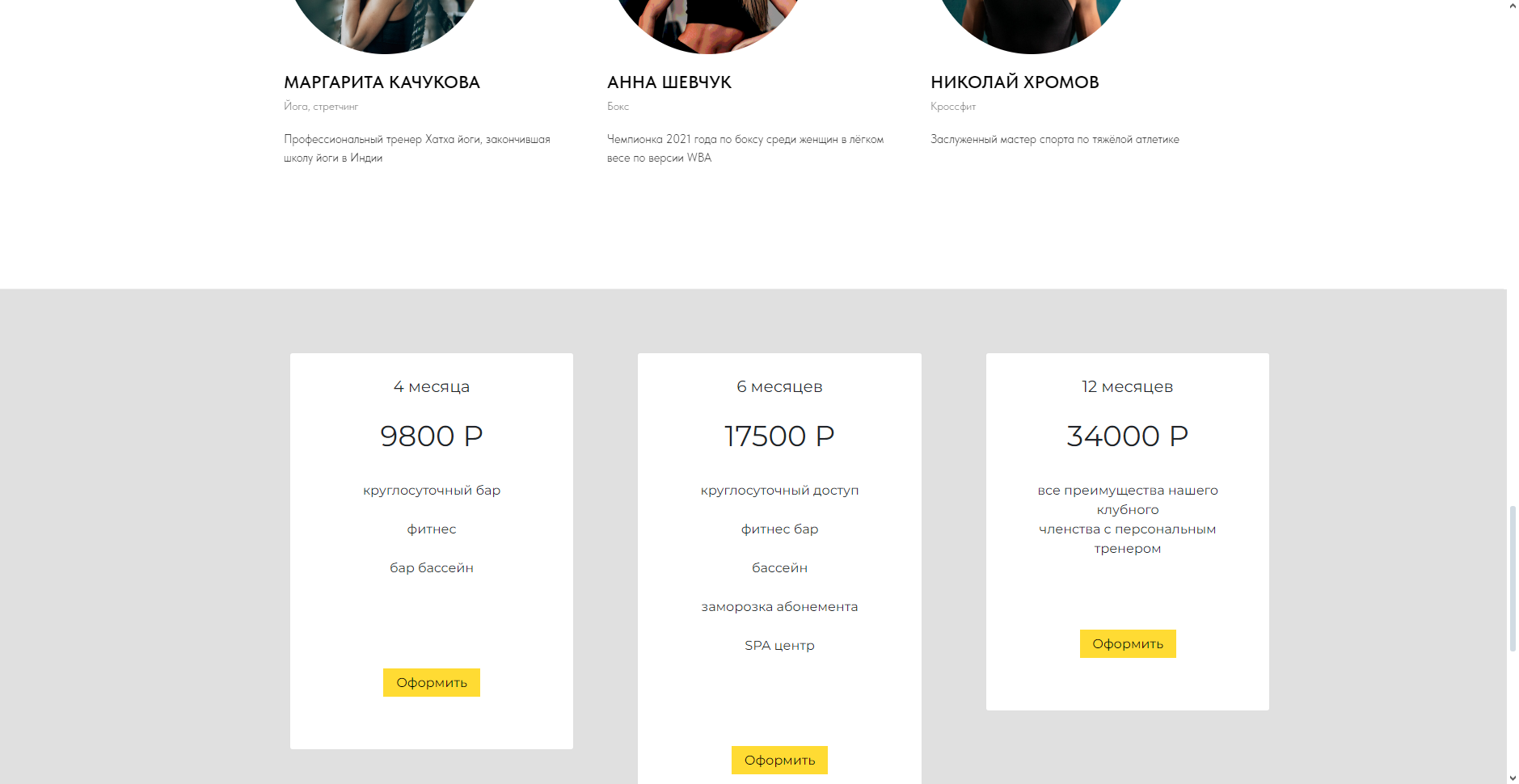


Рисунок 53 – Главная страница сайта

После открытия главной страницы сайта, администратор через раздел подвала может перейти в панель администратора. Вход осуществляется по запросу пароля и уникального логина администратора системы. После этого открывается страница со списком таблиц, показанная на рисунке 54.

При каждом вводе происходит проверка вводимых символов. Программа не позволит вводить буквы там, где должны быть числовые значения. И, наоборот. Там, где должны быть только буквы – не даст ввести цифры. Числовые значения также проверяются на целочисленность. При вводе дробного значения выходит окно с соответствующим сообщением.

Для удобства пользования сайтом, при вводе некоторых значений открывается выпадающий список, откуда мы можем выбрать уже ранее добавленные данные.

После каждого действия, произведенного с базой данных выводится окно с уведомлением о совершенном действии, их можно увидеть на рисунке 58.

По такому принципу были созданы все страницы с таблицами, функциональность которых показана на рисунках 54-61.

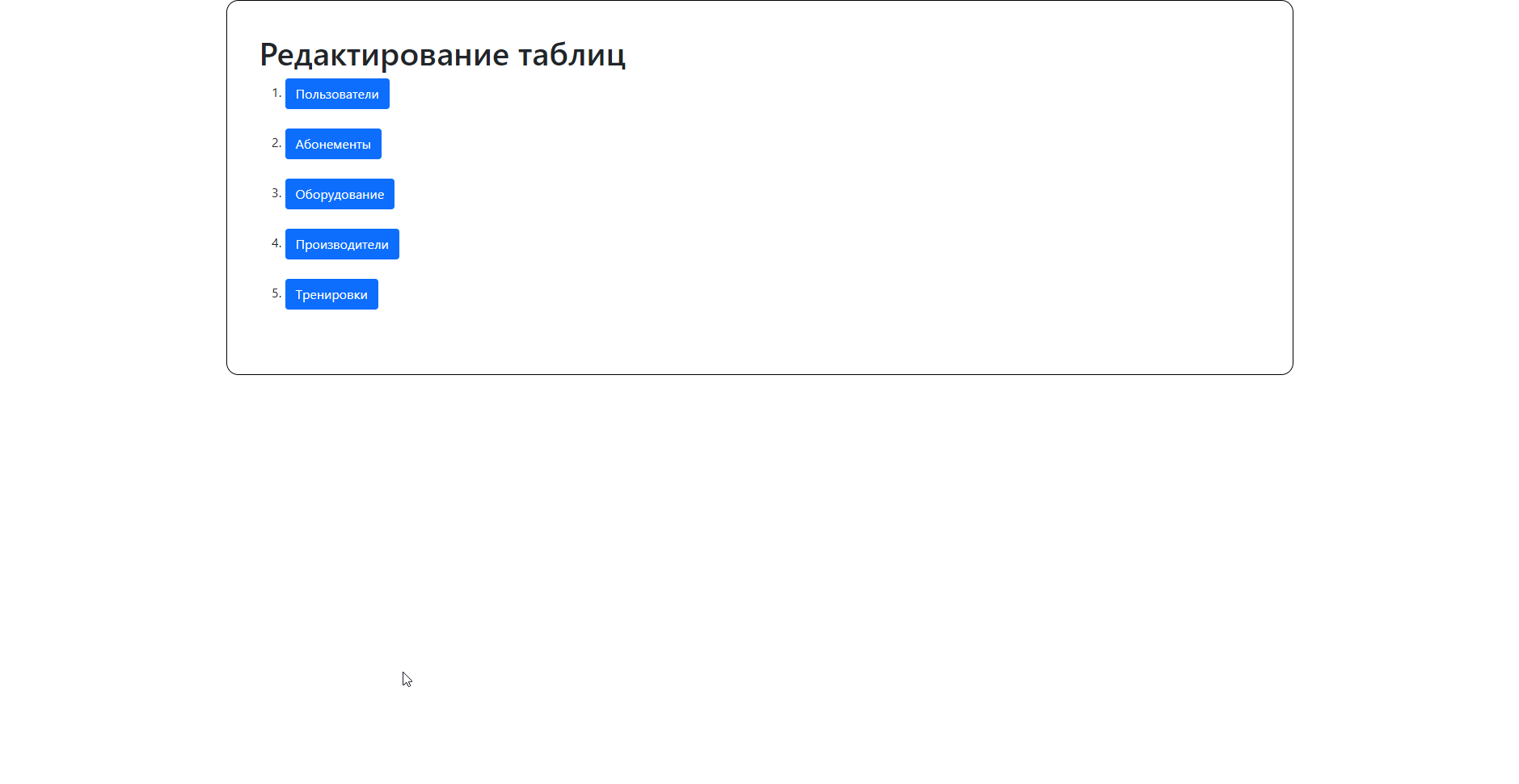


Рисунок 54 – Представление всех таблиц



Рисунок 55 – Редактирование таблицы «Пользователи»

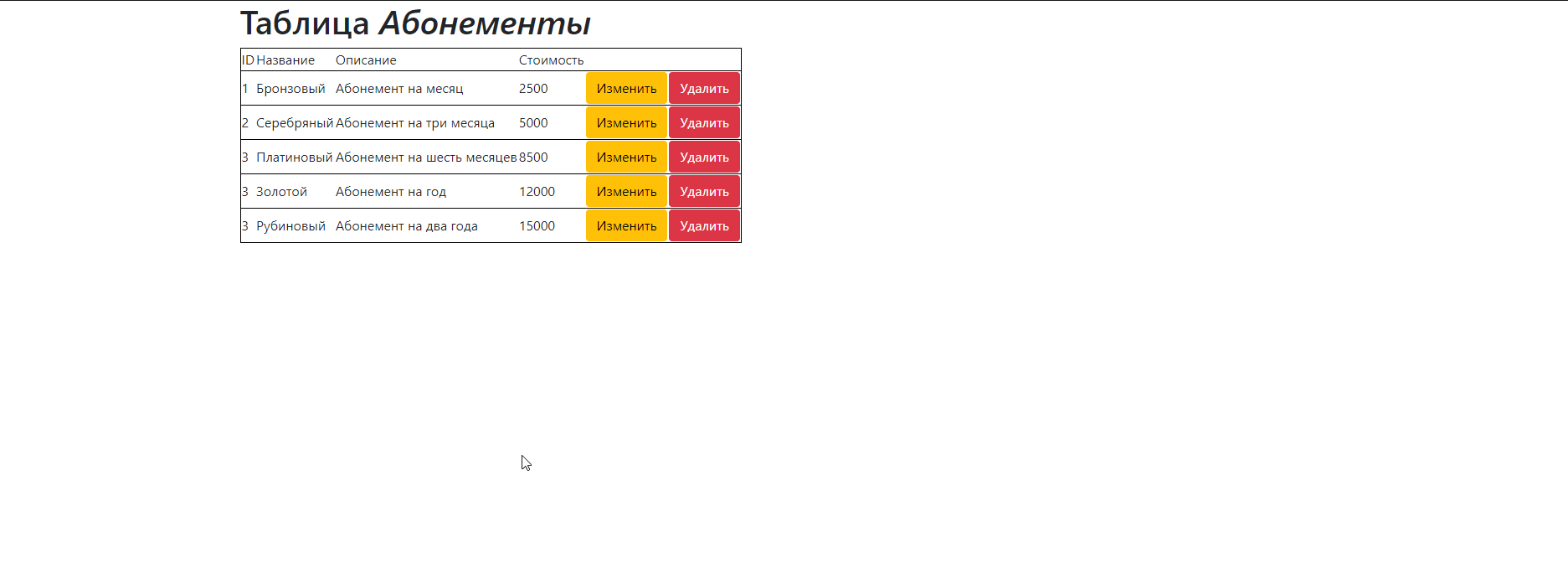


Рисунок 56 – Редактирование таблицы абонементы

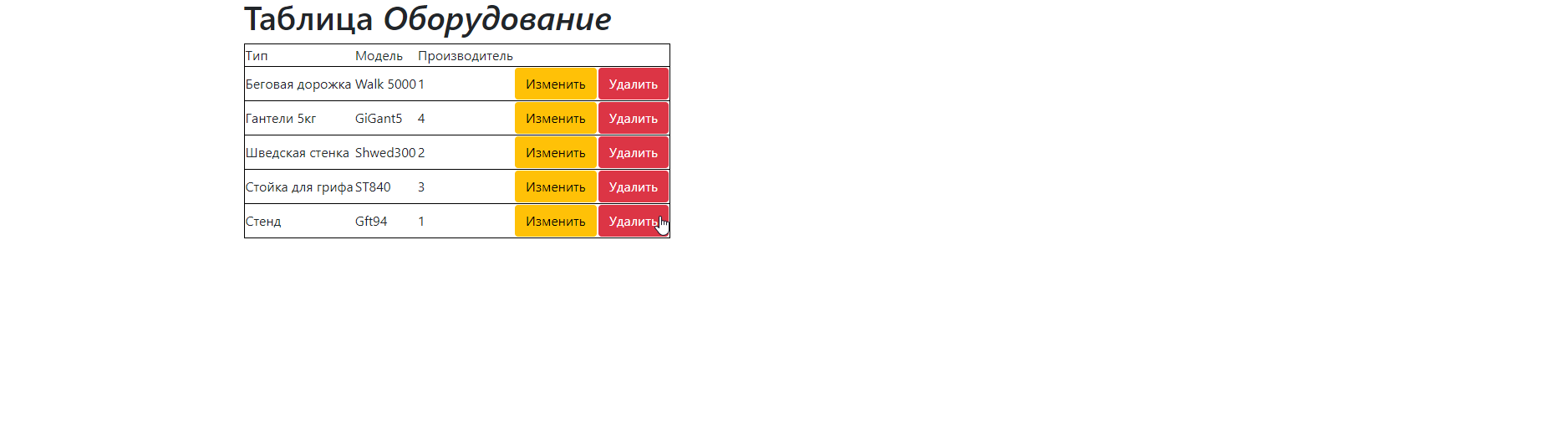


Рисунок 57 – Редактирование таблицы оборудование

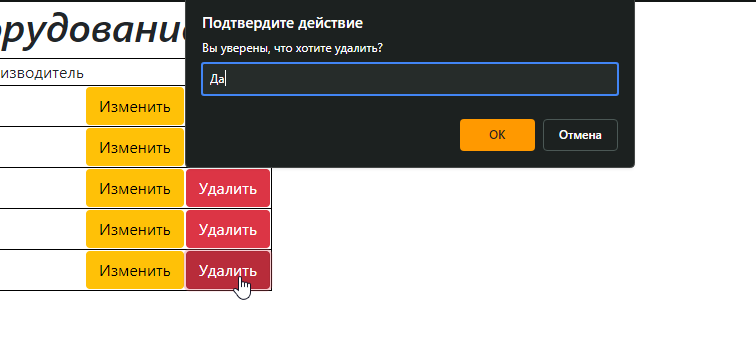


Рисунок 58 – Подтверждение удаления записи из базы данных

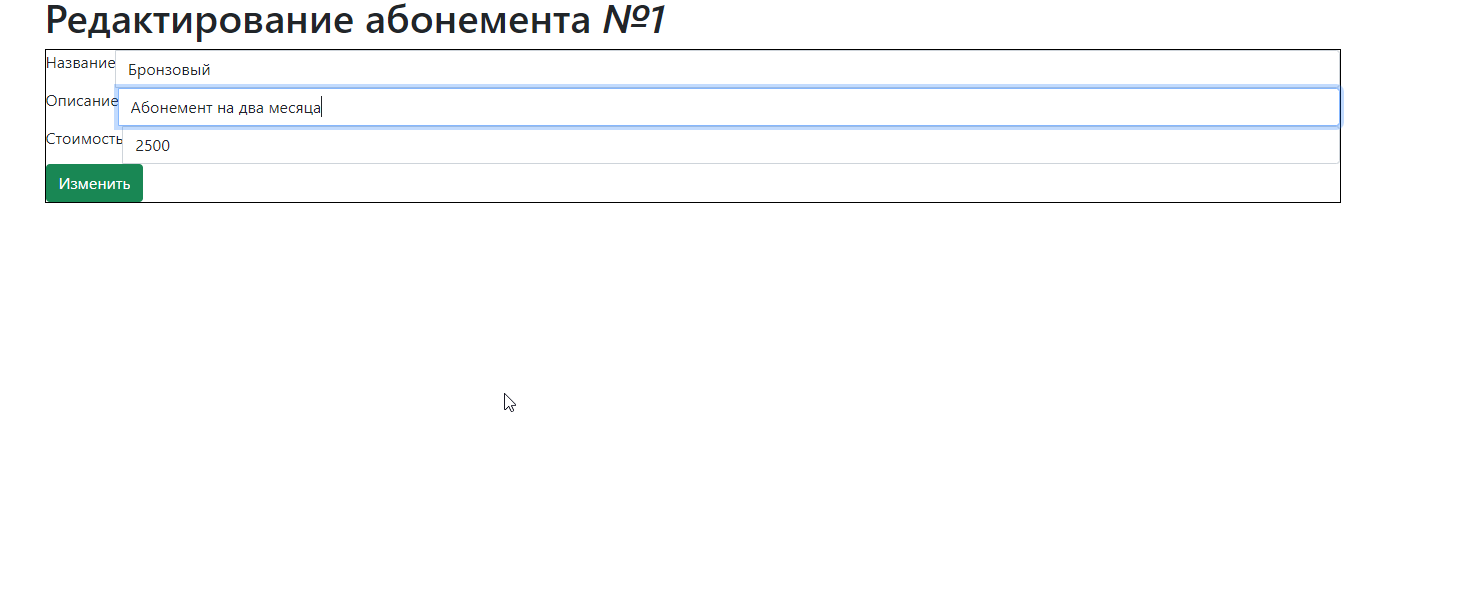


Рисунок 59 – Страница для изменения данных абонемента

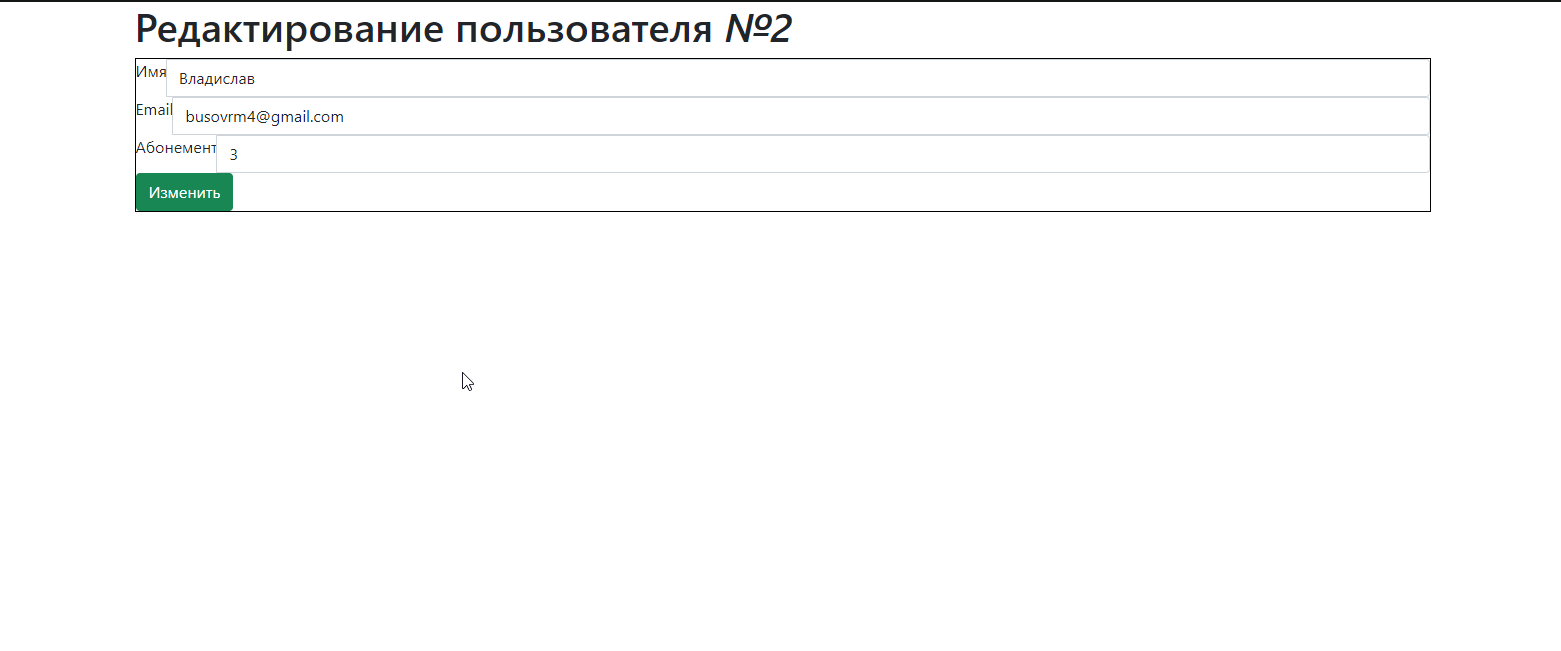


Рисунок 60 – Страница для изменения данных пользователя

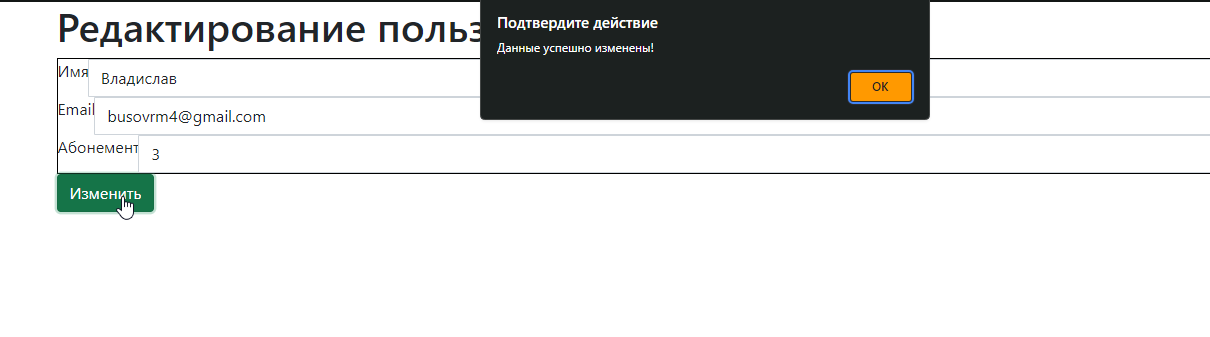


Рисунок 61 – Представление всех номеров

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы была создана база данных «Фитнес клуб», направленная на облегчения работы сотрудников клуба.

Исследована предметная область, рассмотрены особенности работы фитнес клуба и должностные обязанностей сотрудников.

Подробно описана постановка задачи курсового проекта. Определены основные моменты, которые должны быть учтены при проектировании базы данных.

Рассмотрен список выделенных сущностей и их атрибутов. Создана модель «Сущность-связь».

В состав базы данных входят таблицы, между которыми определена связь. Рассмотрены функциональные связи между атрибутами выделенных сущностями, проведена нормализация базы данных к 3НФ.

Создана инфологическая и даталогическая модели БД.

Для проектирования базы данных была выбрана СУБД MS SQL Server.

Созданы основные запросы к базе данных. Рассмотрены механизмы защиты данных и требования к техническому обеспечению.

Рассмотрена инструкция по использованию базы данных.

В ходе выполнения курсового проекта были приобретены устойчивые навыки проектирования баз данных и разработки запросов. Полученные знания могут быть использованы при разработке более серьезных коммерческих проектов.

# СПИСОК ЛИТЕРАТУРЫ

1. "Базы данных: концепции и реализация" (2006) автора Хеннинга Дельмера
2. "Database System Concepts" (2019) авторов Абрахама Сильбершатца, Генри Корнера и Сары Лорин
3. "Проектирование баз данных. Учебник для вузов" (2017) авторов Виктора Маркова и Александра Бабий
4. "Designing Data-Intensive Applications" (2017) автора Мартина Кляппмана
5. "SQL и реляционные технологии" (2016) автора Алексея Кузьменко
6. "Database Design for Mere Mortals" (2013) автора Майкла Херрингтона
7. "Beginning Database Design: From Novice to Professional" (2012) авторов Клэр Коффи и Шелли Джейкобс
8. "Data Modeling Essentials" (2005) автора Граэма В. Уттербека
9. "A Guide to the SQL Standard" (1996) авторов К. Дейт и Хью Дарвелл
10. "Data and Reality: A Timeless Perspective on Perceiving and Managing Information in Our Imprecise World" (2003) автора Уильяма Кент Прайса