**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет»**

**(ГГТУ)**

**Промышленно-экономический колледж**

**О Т Ч Ё Т**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**ПМ.01 Разработка модулей программного обеспечения для компьютерных систем**

(код, наименование ПМ)

**ПМ.11 Разработка, администрирование и защита баз данных**

(код, наименование ПМ)

|  |  |  |  |
| --- | --- | --- | --- |
| Студента (ки) | Родермель Артема Владимировича | | |
| Курс | 3 | группа | ИСП.21.2А |
|  |  |  |  |
| Специальность | 09.02.07 Информационные системы и программирование | | |
| Место практики | МОУ СОШ №26 | | |
| Период практики | 24.11.2023 г. – 07.12.2023 г. | | |
|  | 08.12.2023 г. – 21.12.2023 г. | | |
| Руководители практики | | | |
| от колледжа | Шалягина Наталья Сергеевна, Сафонова Марина Юрьевна | | |
|  | ФИО  Фиськова Елена Анатольевна | | |
| от организации | ФИО  Бабурин Антон Алексеевич | | |
|  | ФИО | | |

г. Орехово-Зуево, 2023

# СОДЕРЖАНИЕ

1. Ознакомление с целями и задачами практики.
2. Знакомство с рабочим местом и руководителями практики.
3. Анализ программного и аппаратного обеспечения организации.
4. Разработка программного продукта в соответствии с нуждами организации.
5. Создание базы данных в СУБД.
6. Выводы по практике.

# 1. Ознакомление с целями и задачами практики

Учебная практика проходила в МОУ СОШ №26. Продолжительность учебной практики – 4 недели, с 24 ноября по 21 декабря 2023 года.

Целями практики являются:

* приобретение первоначального практического опыта и первичных профессиональных умений по ПМ.01 и ПМ.11;
* закрепление, углубление знаний и приобретение навыков работы;
* улучшение качества профессиональной подготовки будущего специалиста.

Основными задачами учебной практики являлись:

* формирование у студентов знаний, умений и навыков, профессиональных компетенций, профессионально значимых личностных качеств;
* развитие профессионального интереса, формирование мотивационно-целостного отношения к профессиональной деятельности, готовности к выполнению профессиональных задач в соответствии с нормами морали, профессиональной этики и служебного этикета;
* адаптация студентов к самостоятельной профессиональной деятельности.

В ходе практики были освоены общие и профессиональные компетенции, а именно:

ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4 Выполнять тестирование программных модулей.

ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода.

ПК 11.1 Осуществлять сбор, обработку и анализ информации для проектирования баз данных.

ПК 11.2 Проектировать базу данных на основе анализа предметной области.

ПК 11.3 Разрабатывать объекты базы данных в соответствии с результатами анализа предметной области.

ПК 11.4 Реализовывать базу данных в конкретной системе управления базами данных.

ПК 11.5. Администрировать базы данных.

ПК 11.6 Защищать информацию в базе данных с использованием технологии защиты информации.

ОК 01 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК 02 Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности;

ОК 03 Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по финансовой грамотности в различных жизненных ситуациях;

ОК 04 Эффективно взаимодействовать и работать в коллективе и команде;

ОК 05 Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;

ОК 06 Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;

ОК 07 Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;

ОК 08 Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;

ОК 09 Пользоваться профессиональной документацией на государственном и иностранном языках.

В период учебной практики выполнены следующие работы:

* Создание программного модуля для учета оценок
* Создание базы данных для программного модуля
* Установка различного ПО

# 2. Знакомство с рабочим местом и руководителями практики

Практика проходила в МОУ СОШ №26 в Московской области, города Орехово-Зуево, улица Козлова дом 5, эта организация занимается обучением детей от начальной до средней школы. Здесь есть один компьютерный класс, где преподается информатика с 1 по 11 класс. Компьютерный класс имеет одноранговую локальную сеть. Практикой руководил директор школы Бабурин Антон Александрович.

# 3. Анализ программного и аппаратного обеспечения организации

На рабочем месте нам выдали 5 ноутбуков со следующими характеристиками:

* Процессор: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz
* Оперативная память: 4 Гб
* Видеокарта встроенная
* DVD-ROM

Программное обеспечение:

* Операционная система: Windows 10
* Архиватор: WinRar
* Антивирусная программа: Антивирус Касперского
* Коммуникационная программа: Yandex Browser
* Система программирования: VisualStudio
* Текстовый редактор: MS Word

# 4. Разработка программного продукта в соответствии с нуждами организации

За время практики был создан программный модуль учета клиентов и их абонементов для сотрудников данной организации. Этот программный модуль может использоваться для удобного хранения данных, а также для того чтобы вовремя сообщать клиенту о окончании абонемента. В учетной книжке можно будет регистрировать новых клиентов, присваивать им подписки, просматривать все варианты подписок, просматривать дату окончания подписки и удалять записи о присвоении привилегий.

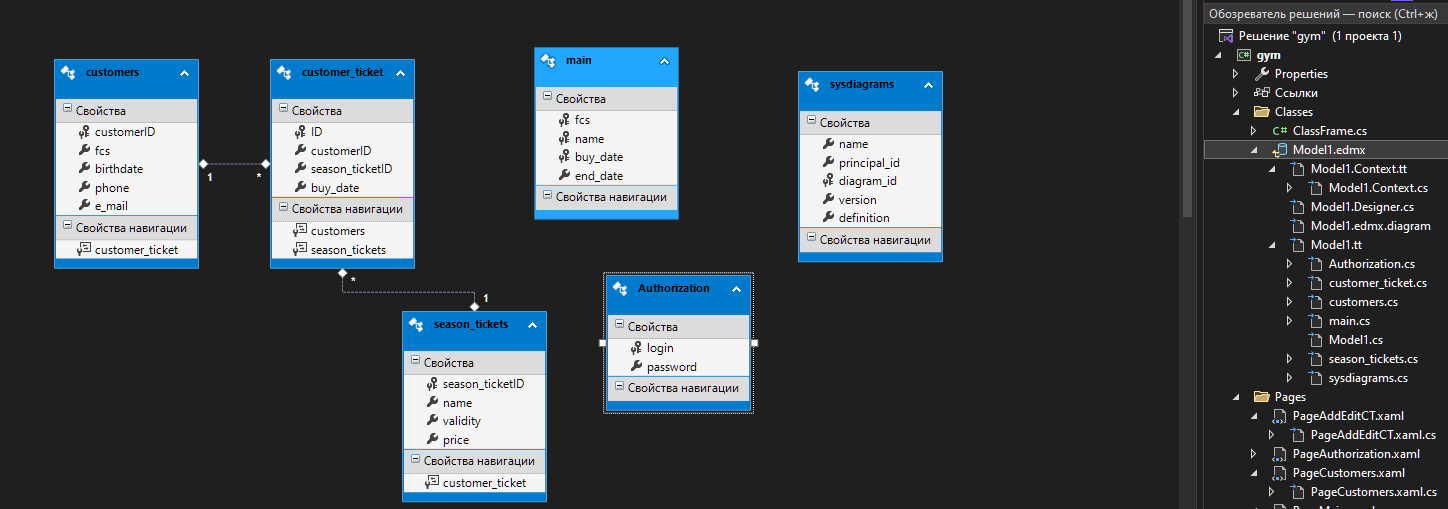
Разработка программного модуля была начата с создания основного экрана приложения, где будут находиться: просмотр присвоенных абонементов и дата их окончания, функция присвоения и удаления абонемента клиента, а также кнопки навигации по данным о клиентах и абонементах. Для работы приложения надо было создать базу данных, используя MS SQL Server ее создание выполнено в 3НФ, а также сделано представление (view) для расчета окончания подписки. Непосредственно разработка велась в Visual Studio, там был создан новый проект, а после создания сразу подключена база данных (рис. 1.).

Рис. 1. Подключенная модель базы данных

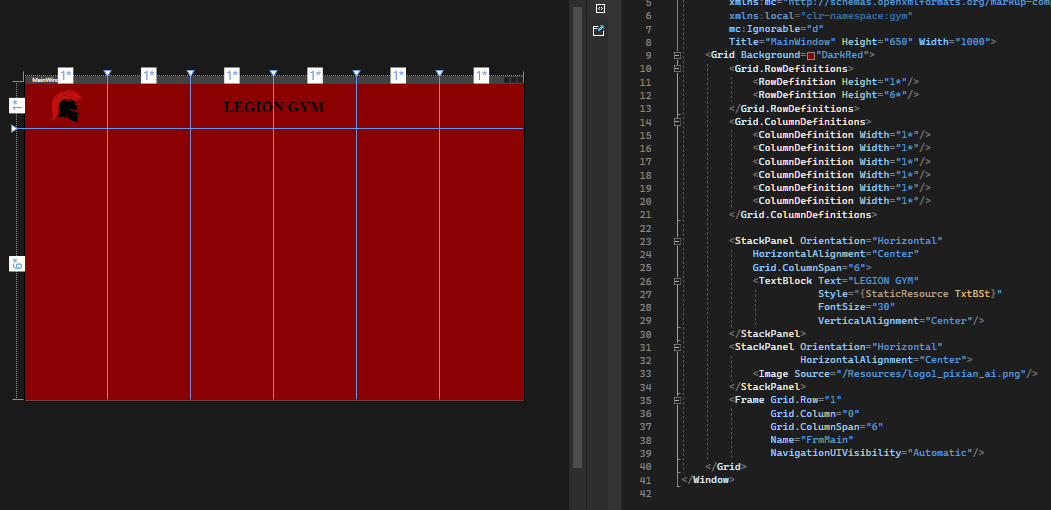
 Далее был оформлен главный лист (MainWindow) так, чтобы сверху на нем был логотип организации и ее название, а снизу были рабочие окна (рис. 2.).

Рис. 2. Главное окно и его код реализации

В логике главного окна было прописано открытие окна авторизации. Данное окно создано следующим образом – была добавлена папка Pages в ней создан элемент PageAuthorization, далее был оформлен внешний вид (рис. 3) и прописан код самой авторизации (рис. 4). После ввода логина и пароля в текстовые поля, приложение подключается к базе данных и проверяет такого пользователя, если он есть приложение проверяет пароль, если все значения введены верно, то нас переадресует на основное рабочее окно.

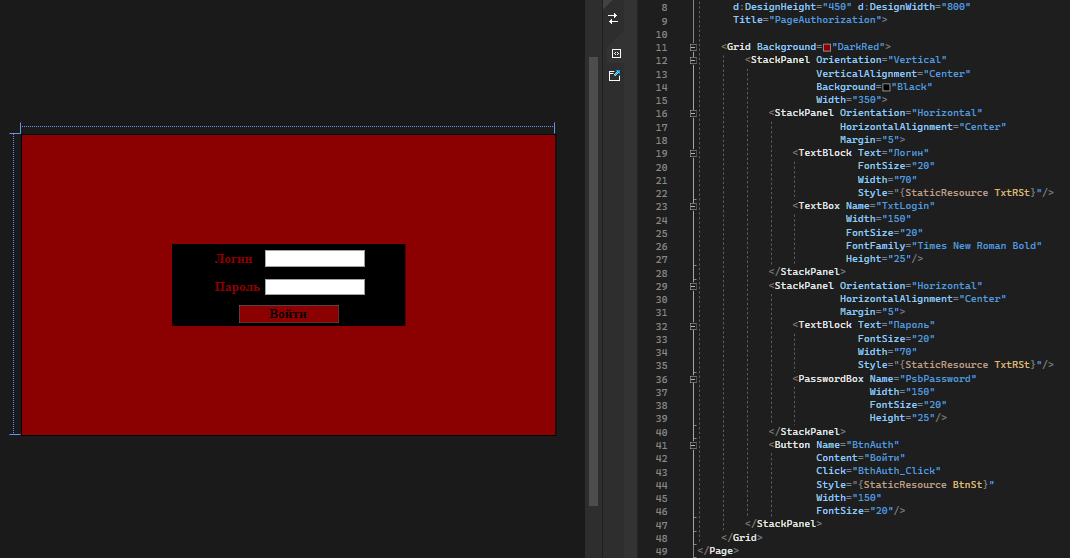
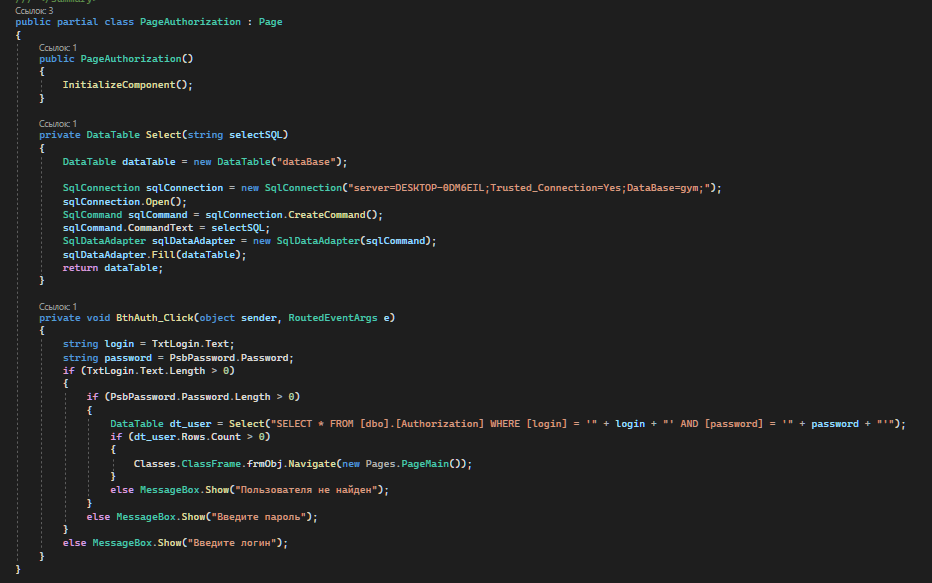
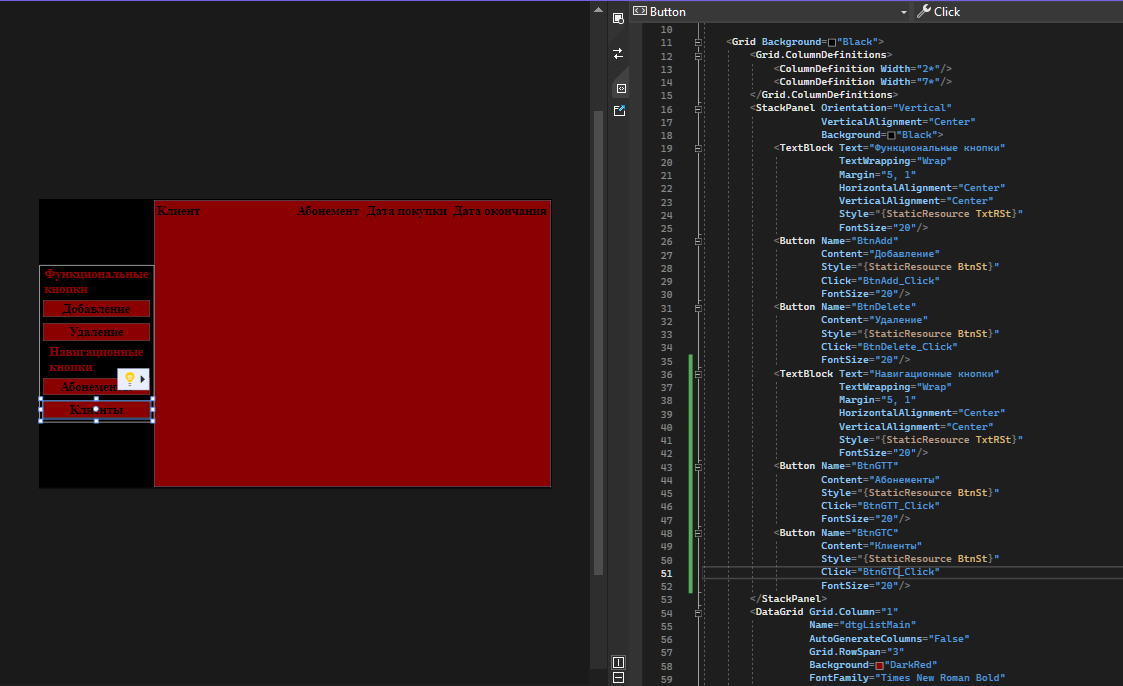
Рис. 3. Окно авторизации и его код реализации

Рис. 4. Логика окна авторизации

Главное рабочее окно реализовано следующим образом – в папке Pages была создана PageMain, было обдумано как будет выглядеть окно и воплощено в коде (рис. 5 и 6). На странице присутствуют: просмотр данных, добавление, удаление записи, переходы на страницы: абонементы и клиенты. Код для всех функций представлен на рисунке 7. Удаление работает следующим образом – после нажатия на кнопку пользователя переадресует на другое окно, на котором он выбирает одну или несколько записей, нажимет кнопку Удаление, после чего требует подтвеждение, если выбрать Да, то записи будут удалены с таблицы, а пользователь вернется на главное рабочее окно. Кнопка добавления переадресовывает пользователя на страницу добавления.

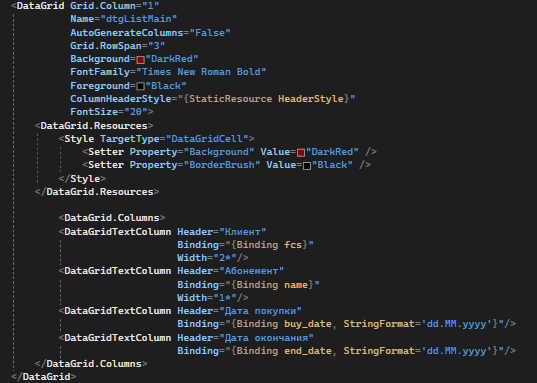
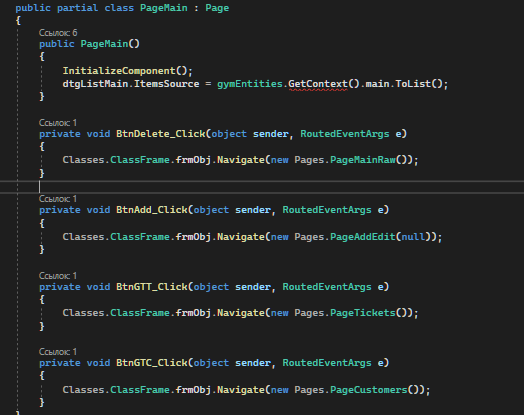
Рис. 5. Главное рабочее окно и его код реализации

Рис. 6. Продолжение кода рис. 5.

Рис. 7. Логика главного рабочего окна

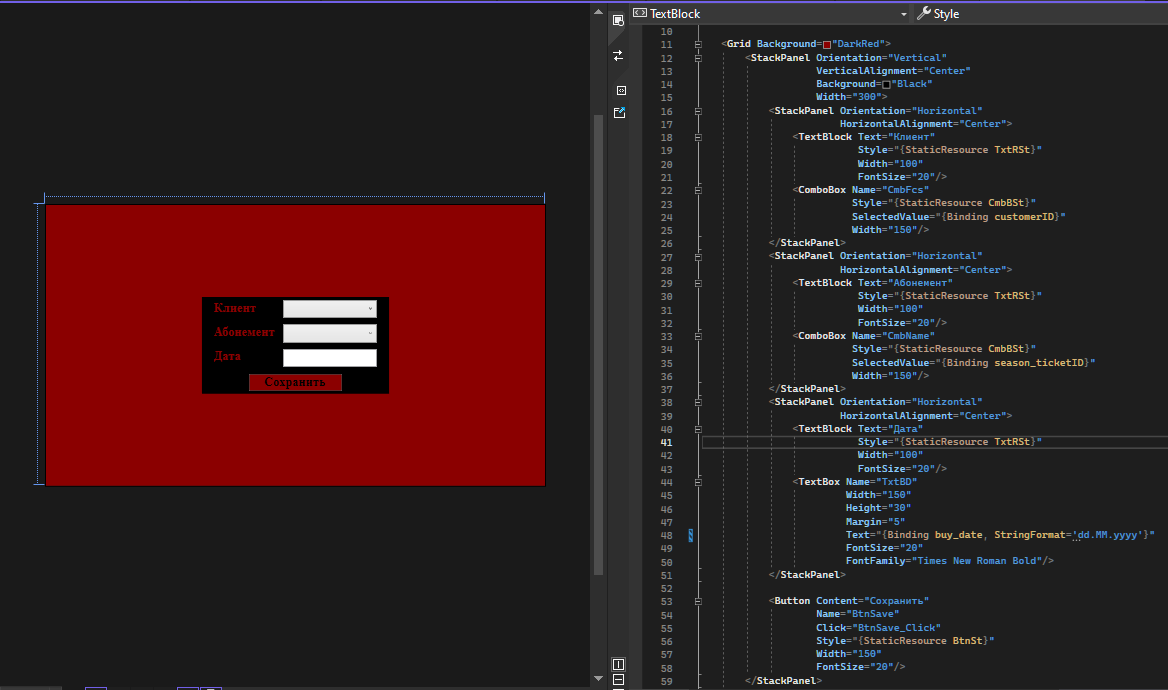
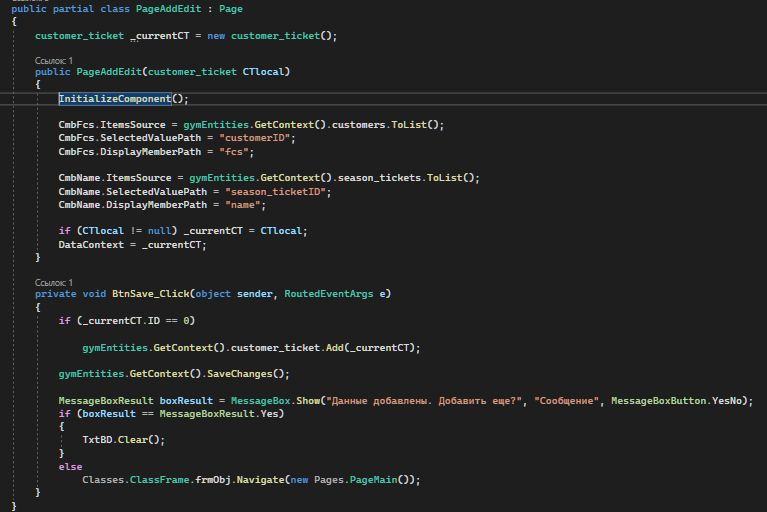
Страница PageAddEditCT также была создана в папке Pages, оформлен внешний вид (рис. 8), прописан код логики (рис. 9). Введенные данные добавляются в базу данных и отображаются в таблице. Для упрощения ввода был представлен выбор клиента и абонемента из выпадающих списков. При нажатии на кнопку сохранить, программный модуль берет введенные данные и вносит в базу данных, после чего пользователя переадресует на главное рабочее окно.

Рис. 8. Страница AddDeleteCT и ее код реализации

Рис. 9. Логика страницы AddDeleteCT

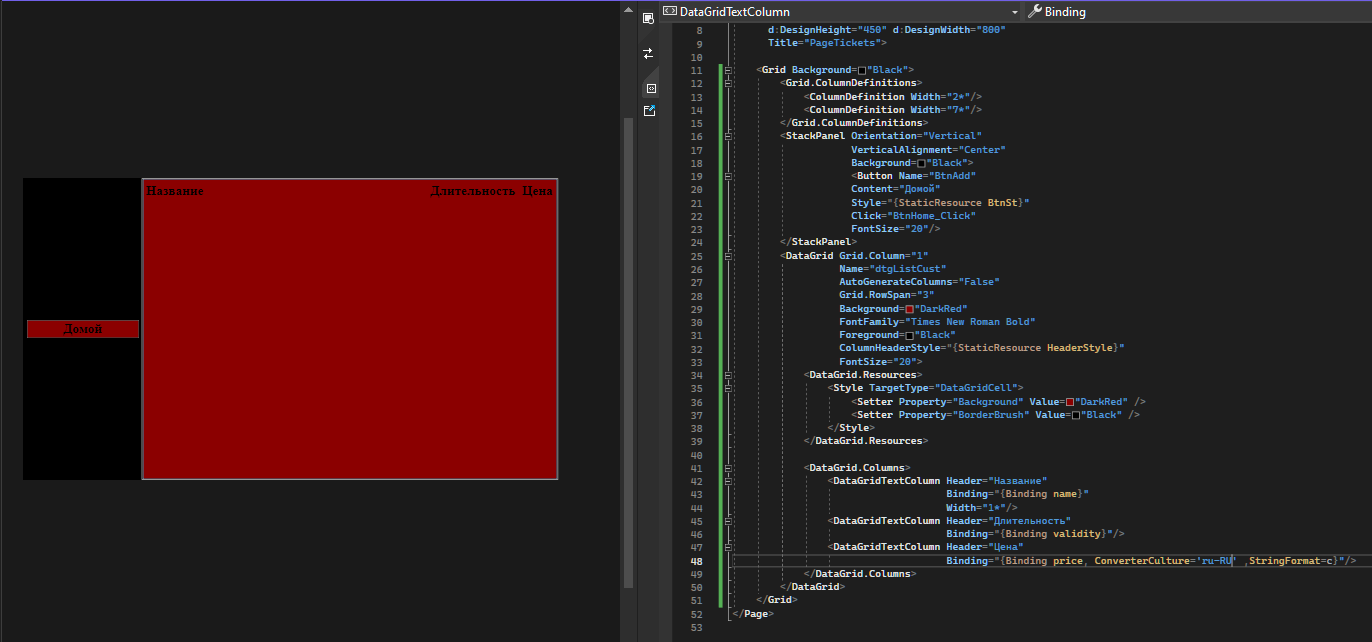
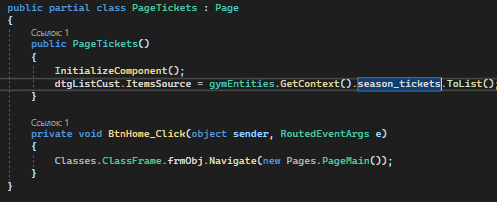
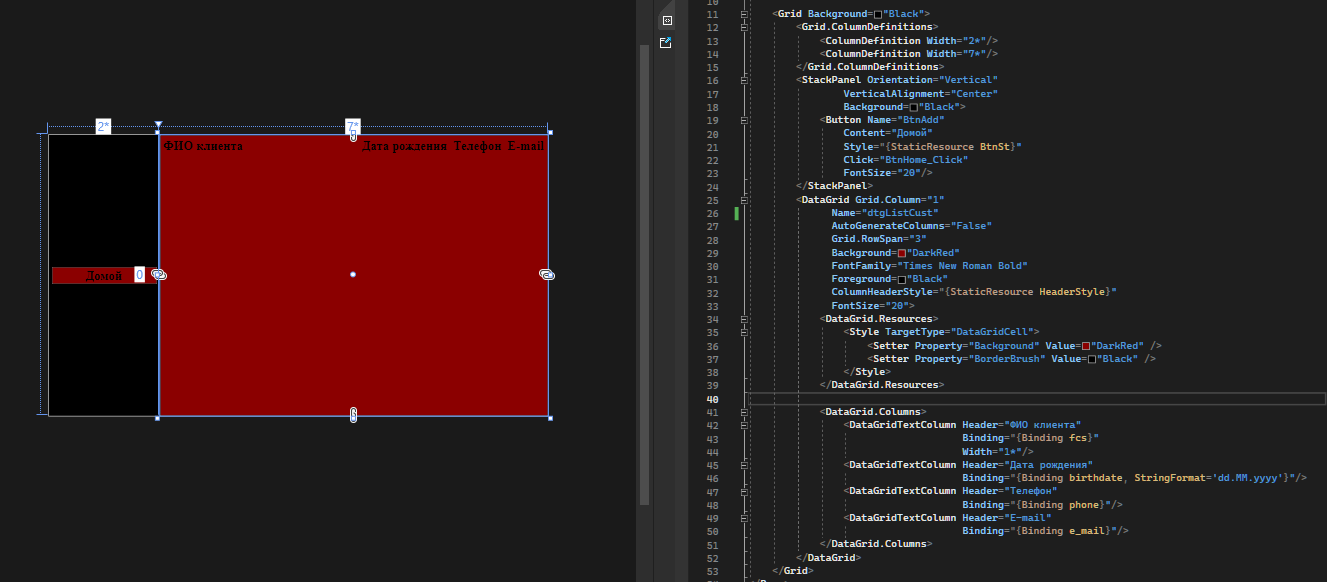
Как было сказано выше, на главном рабочем окне есть кнопки навигации на страницы PageTickets (рис. 10, 11) и PageCustomers (рис. 12, 13). Эти страницы оформлены как главная страница, за исключением кнопок, на этих страницах есть только 1 кнопка Домой, которая пересылает нас на главную страницу.

Рис. 10. Страница PageTickets и ее код реализации

Рис. 11. Логика страницы PageTickets

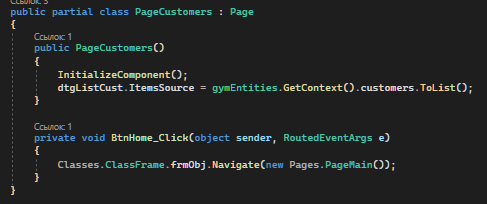
Рис. 12. Страница PageCustomers и ее код реализации

Рис. 13. Логика страницы PageCustomers

Также были созданы стили, используемые при реализации страниц, для упрощения оформления внешнего вида (рис.14).

Рис. 14. Стили

# 5. Создание базы данных в СУБД

За время практики была создана база данных для программного модуля, описанного выше. Создание было начато с разработки ее схемы с учетом 3НФ (рис.1).

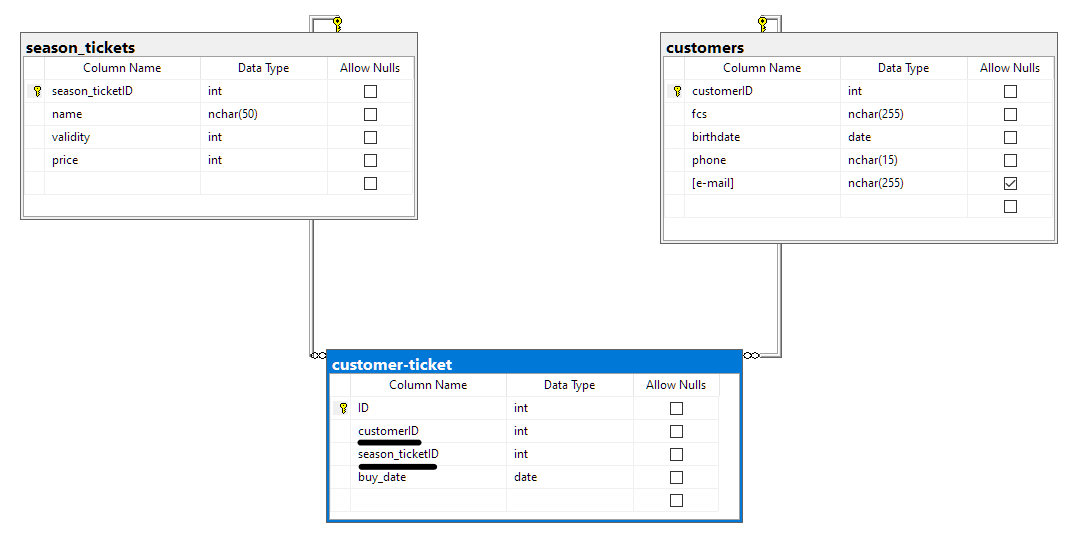
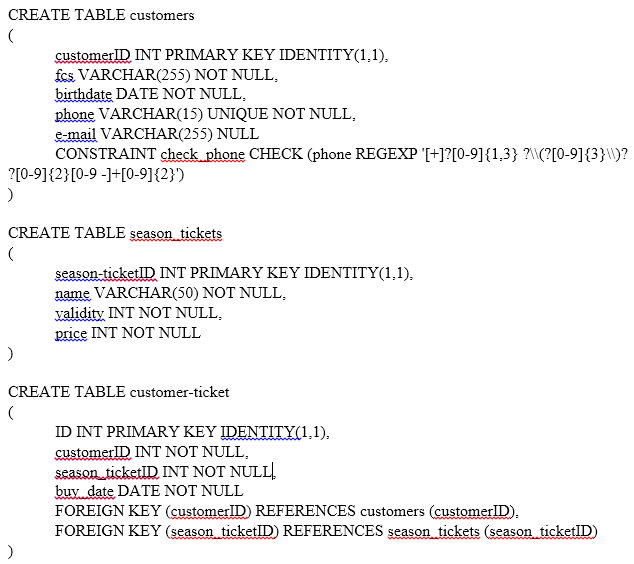
Для реализации базы данных была использована программа MS SQL Server, в ней были прописаны запросы для создания таблиц, описанные на рисунке 2.

Рис. 1. Схема базы данных

Рис. 2. SQL-запросы создания таблиц

При создании были задействованы или не задействованы следующие конструкции:

* PRIMARY KEY установлено на всех полях ID таблиц, также добавлен IDENTITY(1,1), который автоматически повышает значение на 1
* UNIQUE установлено в таблице customers в поле phone, т.к. у клиентов должны быть разные номера телефонов
* CHECK установлено в поле phone от customers, чтобы вводился только номер телефона
* FOREIGN KEY установлено в таблицах со внешними ключами (на диаграмме внешние ключи выделены черным подчеркиванием)
* ON DELETE не было задействовано, т.к. нет связей, в которых можно было бы использовать данную конструкцию
* ON UPDATE не было задействовано т.к. информация обновляется отдельно друг от друга
* DEFAULT не было использовано
* Все поля являются NOT NULL, кроме e-mail от customers, т.к. e-mail клиента необязателен

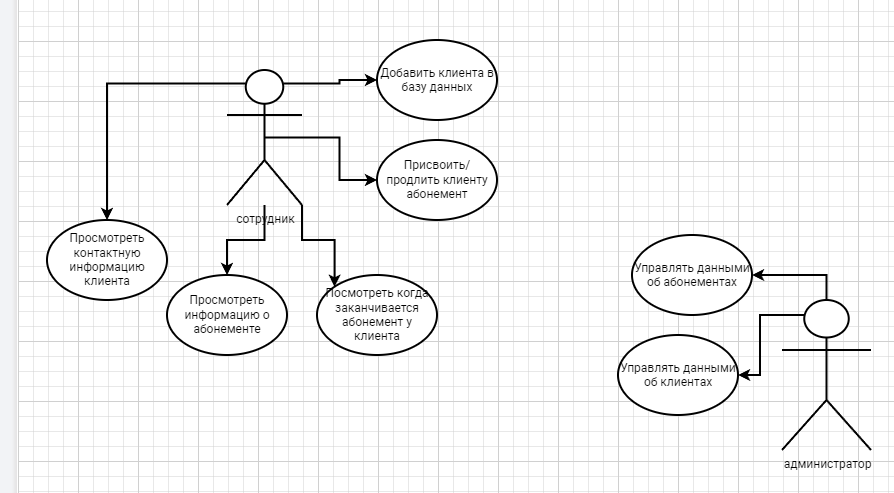
Также была спроектирована UML-диаграмма использования с точки зрения сотрудника и администратора (рис.3)

Рис. 3. UML-диаграмма использования базы данных

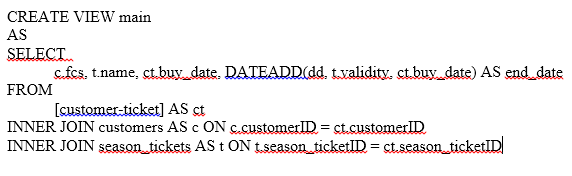
Для программного модуля также было реализовано представление (view), запрос реализации описан на рисунке 4.

Рис. 4. SQL-запрос на создание представления

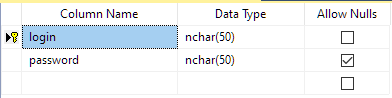
В программном модуле была использована авторизация по логину и паролю, поэтому была создана таблица с информацией об учетных записях (рис. 5).

Рис. 5. Таблица учетных записей

# 6. Выводы по практике

В ходе практики в МОУ СОШ №26 были изучены цели и задачи, а также проведено ознакомление с рабочим местом, расположенном в кабинете информатики и руководителем практики, директором учебного заведения Бабуриным Антоном Александровичем. Был проведен анализ программного и аппаратного обеспечения данной учебной организации. Был разработан программный модуль учета оценок (электронный журнал) для учителей. Также была создана база данных в СУБД, необходимая для корректной работы программного модуля.

Вывод: Задачи, поставленные в рамках практики были выполнены, цель была достигнута.

**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет»**

**(ГГТУ)**

**Промышленно-экономический колледж**

**ПРИЛОЖЕНИЯ К ОТЧЁТУ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

**ПМ.01 Разработка модулей программного обеспечения для компьютерных систем**

(код, наименование ПМ)

**ПМ.11 Разработка, администрирование и защита баз данных**

(код, наименование ПМ)

|  |  |  |  |
| --- | --- | --- | --- |
| Студента (ки) | Родермель Артема Владимировича | | |
| Курс | 3 | группа | ИСП.21.2А |
|  |  |  |  |
| Специальность | 09.02.07 Информационные системы и программирование | | |
| Место практики | МОУ СОШ №26 | | |
| Период практики | 24.11.2023 г. – 07.12.2023 г. | | |
|  | 08.12.2023 г. – 21.12.2023 г. | | |
| Руководители практики | | | |
| от колледжа | Шалягина Наталья Сергеевна, Сафонова Марина Юрьевна | | |
|  | ФИО  Смирнова Анастасия Николаевна | | |
| от организации | ФИО  Бабурин Антон Александрович | | |
|  | ФИО | | |

г. Орехово-Зуево, 2023

Ответы на вопросы:

1. База данных - совокупность данных, организованных по определённым правилам.

Банк данных - информационная система, состоящая из одной или нескольких баз данных и системы хранения, обработки и поиска информации в них.

Различия: нормализация есть только у базы данных, данные нельзя модифицировать в банке, так же у банка строгая упорядоченность данных

2. Система управления базами данных (СУБД) — это программное обеспечение, предназначенное для создания, управления, обновления и анализа баз данных.

Основные функции СУБД:

• Создание баз данных, изменение, удаление и объединение

• Хранение данных

• Защита данных от взлома и нежелательных изменений при помощи распределённого доступа

• Выгрузка и сортировка данных по заданным фильтрам при помощи SQL-запросов

• Поддержка целостности баз данных, резервное копирование и восстановление после сбоёв

Требования к СУБД:

• высокое быстродействие

• простота обновления данных

• независимость данных

• возможность многопользовательского использования данных

• безопасность данных

• стандартизация построения и эксплуатации БД

• адекватность отображения данных соответствующей предметной области

• дружелюбный интерфейс пользователя

Классификация СУБД: иерархические, сетевые, реляционные, объектно-ориентированные, объектно-реляционные

Примеры: MS SQL Server, MySQL, SQLite

3. СУБД состоит из следующий модулей:

• Модуль пользовательского интерфейса

• Модуль управления данными

• Модуль управления метаданными: отвечает за хранение и управление метаданными, такими как схемы баз данных, таблицы, индексы, ограничения и права доступа

• Модуль управления памятью

• Модуль управления транзакциями

• Модуль управления доступом

• Модуль управления резервным копированием и восстановлением

• Модуль управления параллелизмом: обеспечивает распределение работы между несколькими процессорами или узлами в распределенной системе

4. Настольные СУБД используются для сравнительно небольших задач. С учетом этого, указанные СУБД имеют относительно упрощенную архитектуру, в частности, функционируют в режиме файл-сервер, поддерживают не все возможные функции СУБД (например, не ведется журнал транзакций, отсутствует возможность автоматического восстановления базы данных после сбоев).

Архитектура с совместным использованием файлов получила название «файл/сервер». В данной архитектуре компьютеры объединены в сеть. СУБД на рабочей станции посылает запрос пользователя или прикладной программы к БД на сервере. Подсистема управления данными выбирает из БД необходимые для выполнения запроса таблицы целиком. Затем подсистема сетевой коммуникацию пересылает их на рабочую станцию, где СУБД выполняет запрос. При этом если таблицы имеют связанные с ними данные или индексы, то они также пересылаются по сети. В результате передается гораздо больший объем данных, чем реально нужно для выполнения запроса. Вследствие этого сеть сильно перегружается, что является недостатком архитектуры файл/сервер. Также минусом является трудность в обеспечении конфиденциальности информации.

5. Система данных – это набор связанных между собой данных, собранных вместе из одного источника. Существует несколько типов систем данных, среди которых можно выделить последовательную, секционированную, и с виртуальным методом доступа.

6. Многомерная СУБД - система управления базами данных, реализующая Ненормализованную Реляционную Форму, способную обрабатывать модели данных, адекватные представлениям реального мира и свободную от принципиальных общеизвестных недостатков, присущих традиционным СУБД на основе нормализованной реляционной формы

Измерение – последовательность значений одного из анализируемых параметров; для времени – последовательность дней, для параметра «регион» - список городов.

Ячейка - атомарная структура куба, соответствующая полному набору конкретный значений измерений.

Срез представляет собой фильтрацию данных по определенным значениям измерения.

7. Предметная область – это некоторая область управленческой деятельности с определенным набором функций и задач управления, для реализации которых создается и используется базы данных.

Объект - одна из логических единиц хранения информации в базе данных.

Используется для сущностей, которые присутствуют в реальной жизни, как некоторый предмет, существо или понятие, а не как совокупность каких-либо значений

Атрибут (элемент данных) – наименьшая единица структуры данных. Обычно каждому элементу при описании базы данных присваивается уникальное имя

Запись состоит из набора связанных полей, который можно сохранять, изменять и удалять как единое целое.

Кортеж - это одна запись в контексте реляционных баз данных

8. Проектирование базы данных — процесс создания схемы базы данных и определения необходимых ограничений целостности.

Требования к БД:

• высокое быстродействие

• простота обновления данных

• независимость данных

• возможность многопользовательского использования данных

• безопасность данных

• стандартизация построения и эксплуатации БД (фактически СУБД)

• адекватность отображения данных соответствующей предметной области

• дружелюбный интерфейс пользователя

9. Жизненный цикл БД:

1) Предварительное планирование базы данных

2) Проверка осуществимости предполагает подготовку отчетов по трем вопросам:

 есть ли технология

 имеются ли персонал

 окупится ли запланированная база данных

3) Определение требований. На этом этапе определяются:

 цели базы данных;

 информационные потребности различных структурных подразделений и их руководителей;

 требования к оборудованию:

 требования к программному обеспечению.

4) Концептуальное проектирование. На этом этапе создаются подробные модели пользовательских представлений данных предметной области

5) Логическое проектирование. На этом этапе осуществляется выбор типа модели данных

6) Физическое проектирование. На этом этапе логическая модель расширяется характеристиками, необходимыми для определения способов физического хранения базы данных, типа устройств для хранения, методов доступа к данным базы, требуемого объема памяти, правил сопровождения базы данных и др.

7) Оценка и поддержка базы данных

10. Модель Сущность-Связь — это модель данных, позволяющая описывать концептуальные схемы. Она предоставляет графическую нотацию, основанную на блоках и соединяющих их линиях, с помощью которых можно описывать объекты и отношения между ними какой-либо другой модели данных.

Сущность – множество экземпляров реальных или абстрактных объектов, обладающих общими атрибутами или характеристиками. (представляется как прямоугольник на диаграмме)

Связь – некоторая ассоциация между двумя сущностями, бывает:

 один-к-одному

 один-ко-многим

 много-ко-многим

(представляется как линия с обозначением связи 1 (оканчивается одинарной линией) или М (оканчивается пучком линий))

Класс принадлежности сущности - это характер участия сущности в связи. Различают обязательные и необязательные классы принадлежности сущности к связи. (представляется как квадратик на конце связи, обозначает обязательность)

11. Правила преобразования ER-диаграммы в реляционные таблицы:

При связи 1:1:

 Создайте отдельную таблицу для каждой сущности.

 В таблице каждой сущности добавьте первичный ключ, который будет ссылаться на другую таблицу.

 В таблице одной из сущностей добавьте внешний ключ, который будет ссылаться на первичный ключ другой сущности.

При связи 1:М:

 Создайте отдельные таблицы для каждой сущности.

 В таблице множественной сущности добавьте внешний ключ, который будет ссылаться на первичный ключ сущности с единственным вхождением.

При связи M:N:

 Создайте отдельные таблицы для каждой сущности.

 Создайте дополнительную таблицу для представления связи.

 В таблице-связи добавьте два внешних ключа, которые будут ссылаться на первичные ключи сущностей.

12. 1 к 1: адрес к клиенту

1 к М: клиент к заказам

М к М: заказы к товарам

13. Система автоматизированного проектирования - автоматизированная система, реализующая информационную технологию выполнения функций проектирования. Представляет собой организационно-техническую систему, предназначенную для автоматизации процесса проектирования.

Для обеспечения информационной согласованности в САПР используются различные методы и техники:

 Единая модель данных

 Механизмы проверки целостности данных

 Механизмы обновления данных

 Централизованное управление данными

14. Независимость данных определяется как свойство СУБД, которое помогает вам изменять схему базы данных на одном уровне системы базы данных, не требуя изменения схемы на следующем более высоком уровне. Независимость данных помогает вам отделить данные от всех программ, которые их используют.

15. Концептуальное проектирование - сбор, анализ и редактирование требований к данным

Концептуальный уровень бд, связан с частным представлением данных группы пользователей в виде внешней схемы, объединяемых общностью используемой информации. Каждый конкретный пользователь работает с частью БД и представляет ее в виде внешней модели. Характеризуется разнообразием используемых моделей.

16. Логическое проектирование - преобразование требований к данным в структуры данных

Логический уровень бд, является обобщенным представлением данных всех пользователей в абстрактной форме. Используется три вида моделей: иерархические, сетевые и реляционные.

Сетевая модель является моделью объектов-связей, допускающей только бинарные связи “многие к одному” и использует для описания модель ориентированных графов.

Иерархическая модель является разновидностью сетевой, являющейся совокупностью деревьев.

Реляционная модель использует представление данных в виде таблиц, в основе лежит математическое понятие теоретико-множественного отношения, базируется на реляционной алгебре и теории отношений.

17. Физическое проектирование - определение особенностей хранения данных, методов доступа и т.д.

Физический уровень бд, связан со способом фактического хранения данных в физической памяти ЭВМ. Во многом определяется конкретным типом СУБД.

18. На уровне пользователя информация в базе данных (БД) представляется таким образом, чтобы быть понятной и удобной для использования

Одним из основных способов представления информации в БД на уровне пользователя являются формы или интерфейсы. Формы представляют структурированное отображение данных из БД. Они могут содержать различные элементы управления, такие как текстовые поля, выпадающие списки и кнопки, с помощью которых пользователь может вводить, изменять и просматривать данные.

Другим распространенным методом представления информации является отчетность. Отчеты позволяют пользователю получить структурированную информацию из БД в желаемом формате, как правило, в виде таблиц, графиков или диаграмм.

Кроме того, пользователь может взаимодействовать с БД через команды на языке запросов sql.

19. Иерархическая модель данных — это модель данных, где используется представление базы данных в виде древовидной структуры, состоящей из объектов различных уровней.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня. Такие объекты находятся в отношении предка к потомку, при этом возможна ситуация, когда объект-предок имеет несколько потомков, тогда как у объекта-потомка обязателен только один предок. Объекты, имеющие общего предка, называются близнецами.

Пример: школа – в самом верху директор – в его подчинении завучи – у завучей - преподаватели – преподаватели руководят параллелями классов – в каждом классе есть ученики

20. Сетевая модель данных — логическая модель данных, являющаяся расширением иерархического подхода, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных. В отличии от иерархической модели у потомка могут быть несколько предков.

Пример:

ОТДЕЛ имеет РУКОВОДИТЕЛЯ, ОТДЕЛ состоит из СЛУЖАЩИХ, РУКОВОДИТЕЛЬ является СЛУЖАЩИМ

21. Отношение - это таблица, состоящая из строк и столбцов и имеющая вверху строку, называемую заголовок отношения.

Строки таблицы - отношения называются кортежами, а столбцы атрибутами.

Домен в реляционной модели данных — тип данных, то есть множество допустимых значений

22. -Отличия иерархической и сетевой от реляционной:

• Иерархическая база данных должна быть закодирована внутри приложения, тогда как реляционные базы данных не зависят от приложения

• Иерархическая база данных хранит данные в виде родительских и дочерних узлов, образующих древовидную структуру, тогда как реляционная база данных хранит данные в строках и столбцах таблицы

• Иерархическая база данных реализует только отношения «один-один» и «один-ко-многим», тогда как реляционная база данных также реализует отношения «многие-ко-многим»

• Для извлечения данных в иерархической базе данных необходимо пройти все дерево, а язык запросов SQL используется в реляционных базах данных для извлечения данных

• Иерархическая база данных — это система баз данных первого поколения, а реляционная база данных — база данных второго поколения

23. В реляционных базах данных кортеж — это элемент отношения, строка таблицы; упорядоченный набор из N элементов.

Отношение - двумерная таблица, содержащая некоторые данные. Строки таких таблиц соответствуют записям, а столбцы - атрибутам.

24. Основные свойства отношений:

• кортежи не упорядочены;

• в отношении не может быть двух одинаковых кортежей;

• атрибуты не упорядочены;

• число кортежей в отношении конечно;

• любой атрибут отношения должен содержать данные одного типа (домена);

• все используемые типы данных простые;

• отношение не обладает метрическими свойствами, такими как ширина столбцов, число записей и т.д.

Ограничения отношений:

• Ограничение уникальности

• Ограничение внешнего ключа (заказ точно имеет своего клиента)

• Ограничение NOT NULL

• Ограничение проверки (18+)

25. Реляционная таблица представляет собой структурированное представление данных в виде таблицы, состоящей из строк и столбцов. Каждая строка таблицы представляет собой запись, а каждый столбец — атрибут.

Реляционная таблица обладает следующими свойствами:

• каждый элемент таблицы является одним элементом данных;

• каждый столбец обладает своим уникальным именем;

• одинаковые строки в таблице отсутствуют;

• все столбцы в таблице однородные;

• порядок следования строк и столбцов может быть произвольным.

Поле — столбец реляционной таблицы, который содержит конкретное свойство (атрибут) описываемого информационного объекта.

Запись — строка реляционной таблицы, которая соответствует значениям рассматриваемых атрибутов для каждого информационного объекта.

Первичный ключ – это поле, которое используется для обеспечения уникальности данных в таблице. Это означает, что значение в поле первичного ключа в каждой строке таблицы может быть уникальным.

Связь между таблицами производится благодаря внешнему ключу - это одно или несколько полей (атрибутов), которые являются первичными в другой таблице и значение которых заменяется значениями первичного ключа другой таблицы.

26. Реляционная целостность – это ограничения, которые предотвращают помещение в БД противоречивых данных.

• целостность отношений – первичный ключ не должен содержать отсутствующего значения;

• ссылочная целостность – значение внешнего ключа должно обязательно быть в качестве одного из значений первичного ключа.

27. Спецификой постреляционной модели является то, что она поддерживает множественные группы, называемые ассоциированными множественными полями. В постреляционной модели не накладываются требования на длину и количество полей в записях, что делает структуру таблиц более наглядной.

28. Объектно-ориентированная модель — тип модели для хранения данных, в которой данные представляются в качестве объекта.

Объекты – как отражение сущностей реального мира средствами проектирования БД. Объект характеризуется: – уникальным идентификатором; – состоянием; – поведением. Состояние объекта определяется множеством значений его атрибутов.

Классы – как объекты, схожие по поведению и другим свойствам.

Метод – процедура, производящая действия над свойствами объекта в случае наступления событий.

Наследование – распространение множества свойств и методов на всех потомков объекта.

Инкапсуляция – скрытие данных внутри объектов, доступ к данным осуществляется только через методы объектов.

Полиморфизм – способность объектов разных классов иметь методы с одинаковыми именами. Полиморфизм – способность объектов по-разному реагировать на одно и тоже событие в окружающем мире.

29. Объектно-реляционная модель – гибридная модель, совмещающая в себе возможности реляционных моделей с объективными свойствами данных. Отличительная особенность объектно-реляционной модели состоит в том, что она основана на стратегии реляционной модели.

Достоинства: позволяет обрабатывать объекты со сложной структурой. Недостатки: отсутствие унифицированной теории; отсутствие формальной методологии в проектировании.

30. Многомерная модель данных - это метод, который используется для упорядочивания данных в базе данных наряду с хорошим расположением и сборкой содержимого в базе данных. Многомерная модель данных позволяет клиентам задавать аналитические вопросы, связанные с тенденциями рынка или бизнеса, в отличие от реляционных баз данных, которые позволяют клиентам получать доступ к данным в форме запросов.

Измерение – множество однотипных данных, образующих одну из границ гиперкуба.

Ячейка – часть данных, которая определяется путем определения одного элемента в каждом измерении многомерного массива. Ячейки гиперкуба могут быть пусты или полны.

31. Ключ отношения – это атрибут или комбинация атрибутов, которые уникально идентифицируют каждый кортеж в отношении.

Первичным ключом отношения называется атрибут отношения, однозначно идентифицирующий каждый из его кортежей.

32. Альтернативный ключ отношения - это атрибут (или группа атрибутов), несовпадающий с первичным ключом и уникально идентифицирующий экземпляр сущности.

Внешний ключ отношения - это набор атрибутов отношения, содержащий ссылки на потенциальный ключ другого отношения.

33. –В реляционной модели данных отношения между таблицами или сущностями устанавливаются при помощи ключей.

Взаимосвязи между таблицами в реляционной модели данных бывают разные типы: один к одному (1:1), один к многим (1:N), многие ко многим (N:N).

Взаимосвязь отношений в реляционной модели данных играет важную роль при проектировании баз данных, позволяя эффективно организовывать и структурировать данные так, чтобы они соответствовали реальным взаимосвязям между объектами и сущностями в предметной области.

34. Основные принципы реляционной алгебры:

• Корректность - операции должны давать корректный результат.

• Ассоциативность - порядок выполнения операций не имеет значения.

• Коммутативность - порядок операндов не важен.

• Дистрибутивность - позволяет расширять или сужать отношения.

• Идемпотентность - многократное выполнение не меняет результат.

• Замыкание - все операции дают результат в виде отношения.

• Универсальность - любые отношения могут быть получены из других.

35. Операция проекции - построение вертикального подмножества отношения, т.е. подмножества кортежей, получаемого выбором одних и исключением других атрибутов.

Операция селекции в БД означает выборку или извлечение данных из базы данных в соответствии с определенными критериями или условиями.

Результатом объединения отношений A и B будет отношение с тем же заголовком, что и у совместимых по типу отношений A и B, и телом, состоящим из кортежей, принадлежащих или A, или B, или обоим отношениям.

36. Операция пересечения в БД означает объединение данных из двух или более таблиц, которые имеют общее поле или условие. В результате пересечения получается таблица, содержащая строки, которые присутствуют в обеих исходных таблицах или запросах.

Операция разности в БД означает вычитание одного набора данных (отношения) из другого. Результатом является отношение, содержащее все строки из первого отношения, которых нет во втором отношении.

Операция соединения в БД - это процесс объединения двух или более таблиц на основе общего ключа или условия. В результате соединения получается таблица, которая содержит все комбинации строк из исходных таблиц, соответствующих указанному условию соединения.

37. Декартово произведение в БД - это операция, которая возвращает набор данных, состоящий из всех возможных комбинаций строк двух или более исходных таблиц.

Основные объекты БД:

• Таблица - это объект, предназначенный для хранения данных в виде записей (строк) и полей (столбцов).

• Представление — это виртуальная таблица, содержимое которой определяется запросом.

• Хранимые процедуры — это объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

• Триггер базы данных - это процедурный код, который автоматически выполняется в ответ на определенные события в определенной таблице или представлении базы данных.

• Индекс — объект базы данных, создаваемый с целью повышения производительности поиска данных.

38. Создание базы данных - это сложный процесс, который включает в себя определение требований, проектирование, реализацию, тестирование и ввод в эксплуатацию.

Правила создания таблиц:

• Каждая таблица должна иметь первичный ключ - уникальный идентификатор записи.

• Поля таблицы должны иметь корректный тип данных и достаточную длину для хранения информации.

• Необходимо определить ограничения на значения полей, например, уникальность, диапазон или обязательность ввода.

• Имена таблиц и полей должны быть осмысленными и отражать их содержимое.

• Следует избегать дублирования информации в разных таблицах, если это возможно.

• Рекомендуется использовать нормальные формы для структурирования данных, чтобы избежать избыточности и аномалий.

Способы создания таблиц:

• Использование языка SQL для написания скриптов создания и модификации таблиц

• Использование графических инструментов для работы с БД

39. Типы данных, используемые при создании таблиц:

• Текстовые данные (char, varchar, text)

• Числовые данные (int, float, decimal)

• Дата и время (date, time, datetime)

• Логические данные (bool, boolean)

• Большие объекты (blob, binary)

• Географические данные (point, linestring, polygon)

40. -Свойства атрибутов таблиц баз данных включают в себя:

• Имя атрибута

• Тип данных

• Размер поля

• Обязательность

• Уникальность

• Индекс - указывает, будет ли для данного поля создан индекс.

• Значение по умолчанию

• Правила валидации - набор правил, которые должны выполняться при вводе данных в поле.

Связь между таблицами можно установить, добавив в таблицу внешний ключ, ссылающийся на другую таблицу

41. Сущности - это основные объекты, которые представляют собой что-либо конкретное (таблицы).

Каждая сущность имеет свои уникальные атрибуты, которые описывают ее свойства. Например, сущность “клиент” может иметь атрибуты “имя”, “адрес”, “телефон” и т.д.

Идентификатор сущности - это уникальное значение, которое используется для идентификации конкретной сущности.

42. Виды взаимосвязей:

• 1:1: одна запись в одной таблице соответствует одной записи в другой таблице.

• 1:N: одна запись в одной таблице может иметь связь со многими записями в другой таблице.

• N:M: многие записи в одной таблице могут иметь связь со многими записями в другой таблице.

43. Нормализация — это процесс организации данных в базе данных. Целью нормализации является уменьшение потенциальной противоречивости хранимой в базе данных информации.

1НФ означает, что каждый атрибут в таблице должен быть атомарным, то есть не содержать несколько значений.

2НФ означает, что таблица находится в 2НФ, если она находится в 1НФ и каждый не ключевой атрибут зависит от первичного ключа.

3НФ означает, что таблица находится в 3НФ, если она находится во 2НФ и все не ключевые атрибуты взаимно независимы (атрибут не зависим от атрибута)

44. -Для преобразования логической схемы базы данных в физическую, необходимо выполнить следующие шаги:

• Определить требования к базе данных

• Разработать структуру таблиц, определить поля и их типы данных.

• Создать индексы

• Определить ограничения и правила валидации данных.

• Настроить параметры базы данных, такие как максимальный размер таблиц и максимальный размер страницы.

• Протестировать работу базы данных, выполнить оптимизацию производительности.

Нормализация — это процесс организации данных в базе данных.

1НФ означает, что каждый атрибут в таблице должен быть атомарным, то есть не содержать несколько значений.

2НФ означает, что таблица находится в 2НФ, если она находится в 1НФ и каждый не ключевой атрибут зависит от первичного ключа.

3НФ означает, что таблица находится в 3НФ, если она находится во 2НФ и все не ключевые атрибуты взаимно независимы (атрибут не зависим от атрибута)

45. Функциональная зависимость - это связь между атрибутами в реляционной модели данных. Она определяет, что значение одного атрибута однозначно определяется значениями других атрибутов.

2НФ означает, что таблица находится в 2НФ, если она находится в 1НФ и каждый не ключевой атрибут зависит от первичного ключа.