

**ОТДЕЛЕНИЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ**

**СТРУКТУРНОЕ ПОДРАЗДЕЛЕНИЕ 10**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Допустить к защите** | | |
| Руководитель отделения | | | |
|  | | Селезнева А.В. | |
|  | | | 2023 год |

**КУРСОВАЯ РАБОТА**

|  |
| --- |
| Разработка информационной подсистемы для ресторана |
| |  | | --- | | **Специальность 09.02.07 Информационные системы и программирование**  код, название | |
| **КР.09.02.07.1.33.03** |

**(шифр работы)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент: |  |  |  |  | Ахтамов В.М. |
|  | (подпись) |  | (дата) |  | (Ф.И.О.) |
| Проверил преподаватель: |  |  |  |  | Зяблов Е.И. |
|  | (подпись) |  | (дата) |  | (Ф.И.О.) |
| Оценка: |  |  |  |  |  |
|  |  |  |  |  |  |

**Москва 2023г.**

Оглавление

[ВВЕДЕНИЕ 3](#_Toc132895907)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc132895908)

[1.1 Предметная область 5](#_Toc132895909)

[1.2 Базы данных 5](#_Toc132895910)

[1.2.1 СУБД 6](#_Toc132895911)

[1.2.1.1 SQL 6](#_Toc132895912)

[1.2.2 Модели данных 7](#_Toc132895913)

[1.2.2.1 Иерархическая модель данных 7](#_Toc132895914)

[1.2.2.2 Сетевая модель данных 8](#_Toc132895915)

[1.2.2.3 Реляционная модель данных 8](#_Toc132895916)

[1.3 Языки программирования 9](#_Toc132895917)

[1.3.1 Delphi 9](#_Toc132895918)

[1.3.2 C++ 10](#_Toc132895919)

[1.3.3 C# 10](#_Toc132895920)

[1.4 Среды разработки 10](#_Toc132895921)

[1.4.1 Rider 11](#_Toc132895922)

[1.4.2 Visual Studio Code (VS Code) 12](#_Toc132895923)

[1.4.3 Visual Studio 12](#_Toc132895924)

[1.4.4 Eclipse 13](#_Toc132895925)

[1.4.5 MonoDevelop 14](#_Toc132895926)

[1.4.6 Atom 14](#_Toc132895927)

[1.4.7 Vim 15](#_Toc132895928)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 17](#_Toc132895929)

[2.1 Создание базы данных 17](#_Toc132895930)

[2.1.1 Создание таблиц 17](#_Toc132895931)

[2.1.2 Изменение связей между таблицами 19](#_Toc132895932)

[2.1.3 Создание запросов 24](#_Toc132895933)

[ЗАКЛЮЧЕНИЕ 25](#_Toc132895934)

[СПИСОК ЛИТЕРАТУРЫ 26](#_Toc132895935)

# ВВЕДЕНИЕ

Развитие различных сфер человеческой деятельности на современном этапе невозможно без широкого применения вычислительной техники и создания информационных систем различного направления. Обработка информации в подобных системах стала самостоятельным научно-техническим направлением.

После этапа построения информационной модели начинается проектирование системы. На этом этапе производится выбор технологических решений, на основе которых будет построена информационная система.

Информация в современном мире превратилась в один из наиболее важных ресурсов, а информационные системы (ИС) стали необходимым инструментом практически во всех сферах деятельности. В реальных условиях проектирование - это поиск способа, который удовлетворяет требованиям функциональности системы средствами имеющихся технологий с учетом заданных ограничений.

Разнообразие задач, решаемых с помощью ИС, привело к появлению множества разнотипных систем, отличающихся принципами построения и заложенными в них правилами обработки информации.

Для успешной реализации проекта объект проектирования (ИС) должен быть прежде всего адекватно описан, должны быть построены полные и непротиворечивые функциональные и информационные модели ИС. Накопленный к настоящему времени опыт проектирования ИС показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов. Однако до недавнего времени проектирование ИС выполнялось в основном на интуитивном уровне с применением неформализованных методов, основанных на искусстве, практическом опыте, экспертных оценках и дорогостоящих экспериментальных проверках качества функционирования ИС. Кроме того, в процессе создания и функционирования ИС информационные потребности пользователей могут изменяться или уточняться, что еще более усложняет разработку и сопровождение таких систем. Процесс разработки таких ИСУ основывается на моделировании деятельности предприятия, описании организации и методов ведения их бизнеса, построении архитектуры системы и структуры баз данных, обосновании системы математических моделей и алгоритмов, реализации пользовательского интерфейса и выборе технических средств.

Цель курсовой работы: разработка информационной подсистемы ресторана.

Задача курсовой работы: выбор способа создания информационной системы.

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

# Предметная область

Ресторан – предприятие общественного питания с широким ассортиментом блюд сложного приготовления, включая заказные и фирменные; вина и крепких напитков, табачных и кондитерских изделий и повышенным уровнем обслуживания, в сочетании со стильным и оригинальным дизайном и интерьером помещения.

В данной курсовой работе под термином «ресторан» подразумевается любое заведение общественного питания. При этом, под предприятием общественного питания может пониматься также сеть заведений.

Список понятий предметной области:

ГП - готовая продукция

Заказчик - человек, оформивший заказ

Ресторан - любое предприятие общественного питания

ПОП - Предприятия общественного питания

# Базы данных

Данные – представление объектов реального мира и их свойств в формализованном виде, пригодном для хранения, передачи, интерпретации или обработки. В случае использования данных для уменьшения неопределенности знаний о каком-либо объекте данные превращаются в информацию.

База данных – совокупность данных, организованных по определенным правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, независимая от прикладных программ.

Ведение базы данных – деятельность по обновлению, восстановлению и перестройке структуры базы данных с целью обеспечения ее целостности, сохранности и эффективности использования.

# СУБД

Система управления базами данных – совокупность программ и языковых средств, предназначенных для управления данными в базе данных, ведения базы данных и обеспечения взаимодействия ее с прикладными программами.

# SQL

Язык структурированных запросов (SQL) – это язык программирования, предназначенный для управления и манипулирования базами данных, он используется для поиска и извлечения информации из огромных баз данных.

SQL отображается в реляционных базах данных, форме базы данных, полученной из реляционной модели Эдгара Ф. Кодда для хранения информации таким образом, чтобы она была доступной и идентифицируемой по отношению к другой информации. Язык изначально назывался SEQUEL (структурированный английский язык запросов) и был разработан Дональдом Чемберлином и Рэймондом Бойсом в IBM (1974 г.) SQL содержит четыре основные категории с двумя дополнительными командами, которые в основном используются для компьютерных баз данных. SQL - это язык четвертого поколения, его команды близки к человеческому языку. А вот языки программирования Java и C ++ - языки третьего поколения, которые абстрактны для неопытного глаза.

SQL (язык структурированных запросов) является наиболее широко используемым языком программирования для организации и извлечения данных из базы данных. Это позволяет нам выполнять все операции CRUD (создание, чтение, обновление и удаление) в базе данных.

В 1989 году Microsoft разработала собственную систему управления реляционными базами данных под названием Microsoft SQL Server. Он был разработан для поддержки ряда приложений для обработки транзакций, бизнес-аналитики и аналитики в ИТ. Существует несколько версий SQL Server с различными интеграциями инструментов управления данными и аналитики, а также с более новыми технологиями, такими как облачные вычисления и отчеты на мобильных устройствах. Oracle также предоставляет SQL, который можно использовать для доступа, определения и обслуживания данных. Его SQL-сервер может использоваться в сотрудничестве с Oracle, PHP, Java и другими языками программирования.

# Модели данных

Модель данных – совокупность правил порождения структур данных в БД, операций над ними, а также ограничений целостности, определяющих допустимые связи и значения данных, последовательность их изменения. База данных создается для достижения определенных целей исследования, и в зависимости от изменения или расширения целей модель БД может меняться.

Развитие теории и практики проектирования и эксплуатации баз данных сопровождается интенсивным развитием моделей данных. Самой первой МД, которая использовалась для построения концептуальных схем была иерархическая модель. Вслед за ней появились сетевые модели. Потом ER-модели, и, как итог развития моделей, появились реляционные и постреляционные модели. Каждая из перечисленных моделей имеет свои достоинства и недостатки.

# Иерархическая модель данных

В основу иерархической МД положен тот факт, что данные предметной области могут объединяться в группы по наличию у них тех или иных признаков или иных общих свойств. Эти свойства выделяются в предметной области в виде абстрактных данных, и между ними устанавливаются иерархические связи.

Иерархическая модель данных (ИМД) – это модель, в которой абстрактные понятия находятся в отношении предшествования таким образом, что каждому понятию соответствует только один предшественник (родитель). Только одна часть, называемая корнем модели, не имеет предшественника.

Достоинство ИМД: данные, отражающие общие свойства совокупности конкретных данных не дублируются.

Недостаток ИМД связан с дублированием данных в случае их однотипности.

# Сетевая модель данных

Сетевая модель так же, как и иерархическая, обладает весьма выразительными свойствами.

Необходимость сетевой модели проявляется тогда, когда одни и те же конкретные данные в рамках одной и той же предметной области классифицируются не одной, а несколькими системами классификации, то есть предметная область разбита на части, связанные между собой бинарными связями.

Достоинство сетевых МД: данные имеют четкую структуру.

Недостатком сетевых МД является большое количество дополнительной информации о связях.

# Реляционная модель данных

Реляционная модель данных была предложена Э. Коддом и основана на понятии отношения (relation). Она является наиболее распространенной и практически все современные СУБД ориентированы на такое представление данных.

Реляционная модель данных (РМД) – это модель, в которой данные можно представить в виде отношений, изменяющихся во времени. Традиционно в реляционных системах отношением называют таблицу, кортежем – строку таблицы, а атрибутом – столбец. При этом атрибуты имеют уникальные имена в рамках одного отношения.

Достоинство реляционной модели заключается в простоте для понимания, наглядности и удобстве физической реализации на ЭВМ.

Недостатки реляционной модели данных: модель не допускает представления объектов со сложной структурой, поскольку в ее рамках возможно моделирование лишь с помощью двумерных таблиц. Данные об объектах содержатся, как правило, во многих таблицах, что значительно замедляет обработку данных.

# Языки программирования

На сегодняшний день существует огромное количество языков программирования и каждый из них создан для решения определенной категории задач. Выбор языка программирования является важным этапом проектирования и разработки информационной системы, от которого зависит множество факторов.

# Delphi

Delphi — это компилируемый, строго типизированный язык высокого уровня, который поддерживает структурированный и объектно-ориентированный дизайн. Основанный на Delphi, его преимущества включают легко читаемый код, быструю компиляцию и использование нескольких модульных файлов для модульного программирования.

# C++

Поддерживает такие парадигмы программирования как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование, обеспечивает модульность, раздельную компиляцию, обработку исключений, абстракцию данных, объявление типов (классов) объектов, виртуальные функции. Стандартная библиотека включает, в том числе, общеупотребительные контейнеры и алгоритмы. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. В сравнении с его предшественником – языком C, – наибольшее внимание уделено поддержке объектно-ориентированного и обобщённого программирования.

# C#

C# ("C Sharp") - это современный кроссплатформенный и с открытым исходным кодом язык программирования, созданный компанией Microsoft. Он входит в семейство языков программирования .NET.

Язык C# надежен и широко используется в корпоративном мире. Его синтаксис напоминает синтаксис языка Java, и он является объектно-ориентированным. На языке C# возможно разрабатывать код и запускать на компьютере с операционными системами macOS, Windows или Linux.

Язык C# также очень универсален. Его возможно использовать его для разработки широкого спектра приложений, включая настольные приложения, микросервисы, функции Azure, мобильные приложения и веб-интерфейсы. даже возможно запускать код C# в браузере, как и JavaScript, используя WebAssembly (WASM).

# Среды разработки

Разработчики программного обеспечения ежедневно используют интегрированные среды разработки (IDE) и текстовые редакторы. Наличие хорошего текстового редактора или IDE и умение эффективно его использовать жизненно важно для повышения производительности.

# Rider

Rider — это относительно новая IDE, запущенная в 2017 году. Компания JetBrains разрабатывает ее наряду со многими другими инструментами разработки программного обеспечения мирового класса. К ним относятся IntelliJ IDE и PhpStorm ReSharper.

Rider — одна из лучших IDE для C#, которую вы можете получить. Это мощная и легкая IDE, которая может похвастаться множеством улучшений рефакторинга кода и производительности. Среду разработки можно получить в рамках бесплатной пробной версии в течение 30 дней, после чего придется оплачивать ежемесячную или годовую подписку.

Более продвинутые подписки Rider включают инструменты разработки, такие как dotTrace, dotMemory и ReSharper. Базовая подписка стоит 150 долларов в год. 

Рисунок 1 – Годовая стоимость подписки среды разработки Rider

# Visual Studio Code (VS Code)

Visual Studio Code — это современный текстовый редактор с открытым исходным кодом от Microsoft. Он легкий и использует минимальные ресурсы на вашем ПК.

Одной из уникальных особенностей VS Code является его способность поддерживать несколько языков программирования. Есть поддержка Typescript и PHP, что является плюсом для full-stack разработчиков. VS Code также легко настраивается и расширяется через рынок его расширений. Основным расширением для разработки на C# является C# для Visual Studio Code от OmniSharp.



Рисунок 2 – Пример подсветки кода в VS Code разных языков программирования

# Visual Studio

Visual Studio — мощная IDE. Он может похвастаться широким спектром функций, таких как встроенные инструменты git, анализ и профилирование кода, управление пакетами NuGet и удаленная отладка. Неудивительно, что это одна из наиболее широко используемых IDE среди разработчиков C#.

Его можно использовать для разработки широкого спектра приложений от консольных программ до настольных приложений.

Visual Studio Community бесплатна и доступна в macOS и Windows. Для получения дополнительных функций и услуг поддержки необходимо приобрести бизнес-версию или корпоративную версию. 

Рисунок 3 – Cреда разработки Visual Studio

# Eclipse

Eclipse — надежная и мощная IDE, широко используемая разработчиками Java, но она также поддерживает другие языки, такие как C и C++.

Его функциональность можно легко расширить с помощью плагинов, доступных в Eclipse marketplace. Он также поставляется с инструментами анализа кода и отладки. Должен быть установлен плагин aCute для разработки на C# в Eclipse.

# MonoDevelop

MonoDevelop — еще одна надежная IDE. Он поддерживает не только C#, но и другие языки семейства Dotnet, такие как F# и Visual Basic.

Среда IDE может похвастаться широким набором функций, упрощающих разработчикам настройку сложных проектов или решений.

Используя MonoDevelop, можно легко переносить проекты C#, созданные с помощью Visual Studio, в другие операционные системы. Фактически Visual Studio для Mac основан на MonoDevelop.

MonoDevelop имеет открытый исходный код и доступен во всех основных операционных системах.

# Atom

Atom — это мощный и легкий текстовый редактор, поддерживаемый GitHub и его сообществом. Помимо языка C#, Atom поддерживает несколько других языков, таких как JavaScript и HTML.

Если вы знакомы с сочетаниями клавиш из других текстовых редакторов, таких как VS Code или Vim, вы можете легко перенести их в Atom. 

Рисунок 4 – среда разработки Atom

# Vim

Vim - это редактор терминала, существующий уже несколько десятилетий. Он используется для решения многих задач, от настройки системных файлов до написания скриптов и программирования полноценных приложений.

Редактор Vim надежный, легкий и очень настраиваемый. Vim также поставляется со встроенным учебником, который поможет освоить его использование в повседневной работе.

Заядлому пользователю Vim достаточно установить несколько инструментов, таких как OmniSharp, и он будет полностью готов к разработке на C# в этом мощном редакторе.

Vim имеет открытый исходный код и доступен для Windows, macOS и Linux. Большинство систем Linux и Unix поставляются с редактором Vim по умолчанию. 

Рисунок 5 – Пример использования среды разработки Vim

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

* 1. Создание базы данных
     1. Создание таблиц

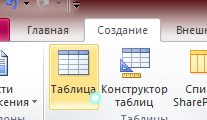


Рисунок 6 – создание таблицы

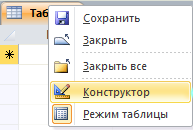


Рисунок 7 – переход в режим конструктора таблицы

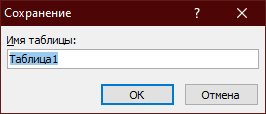


Рисунок 8 – окно изменения названия таблицы

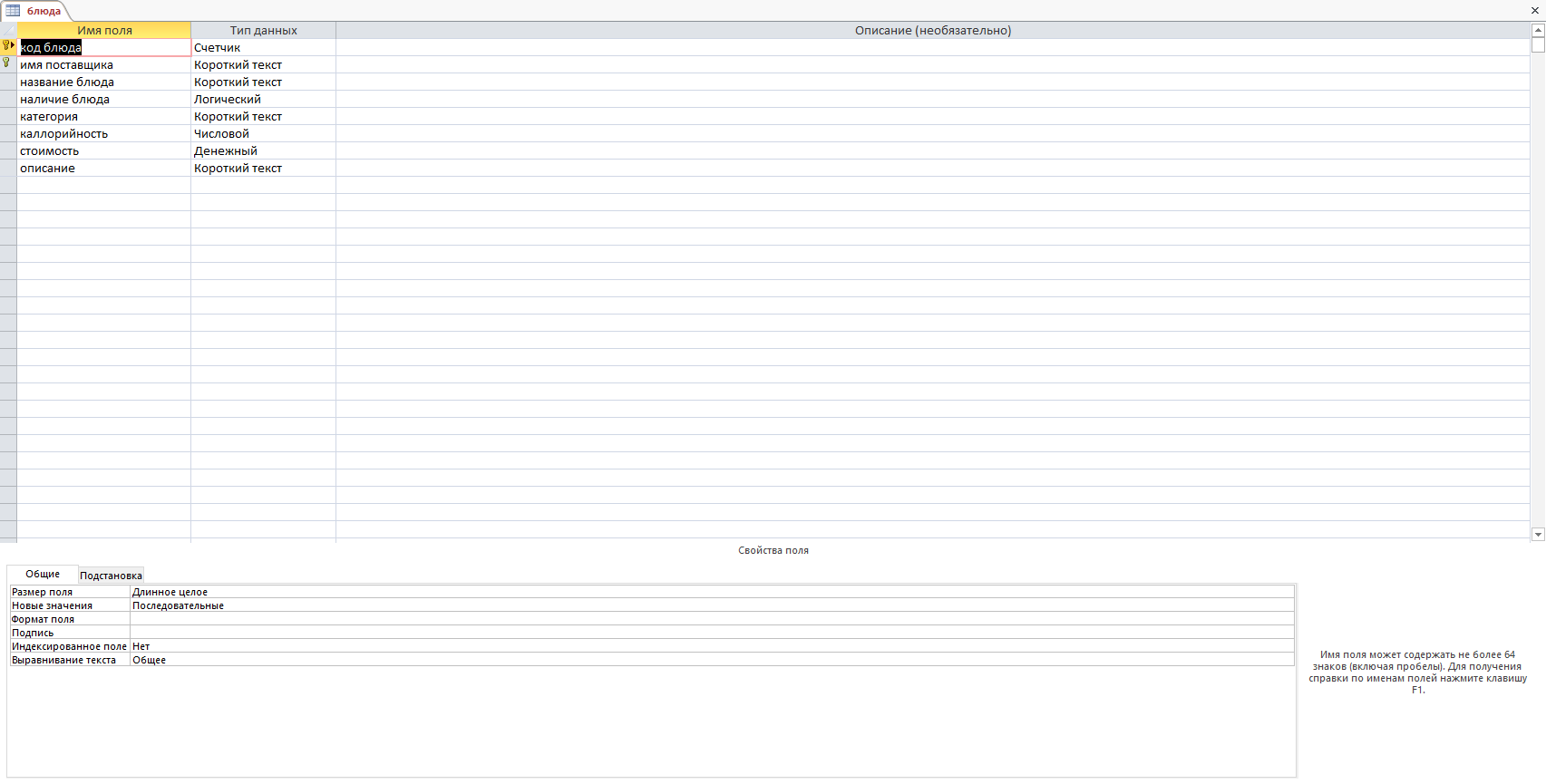


Рисунок 9 – окно конструктора создания таблицы

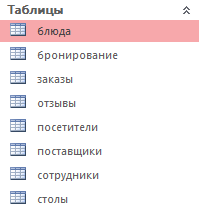


Рисунок 10 – созданные таблицы

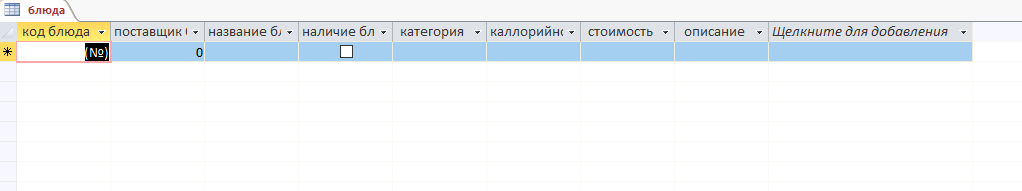


Рисунок 11 – просмотр данных в режиме таблицы

# Изменение связей между таблицами

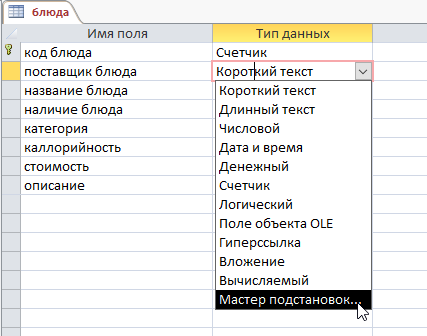


Рисунок 12 – изменение типа столбца

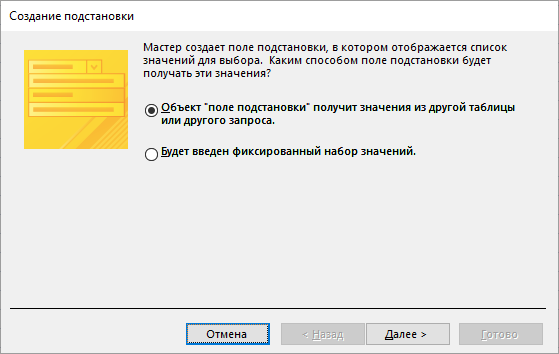


Рисунок 13 – окно создания подстановки

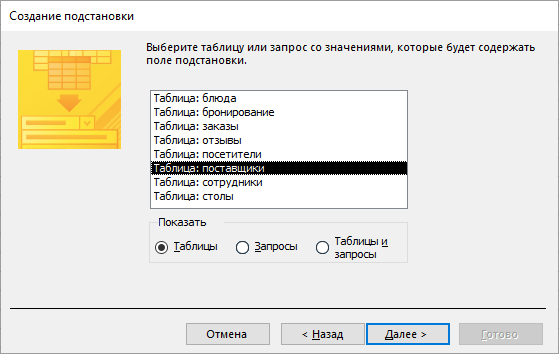


Рисунок 14 – таблица, из которой будут браться данные для подстановки

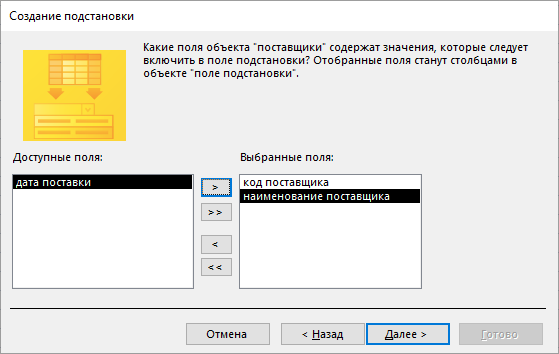


Рисунок 15 – выбранные столбцы с данными

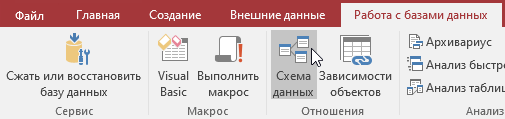


Рисунок 15 – создание связей между таблицами

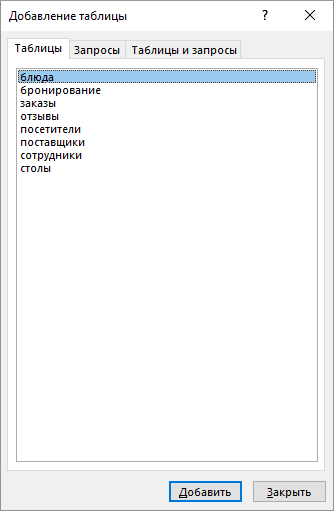


Рисунок 16 – окно выбора таблиц для настройки связей

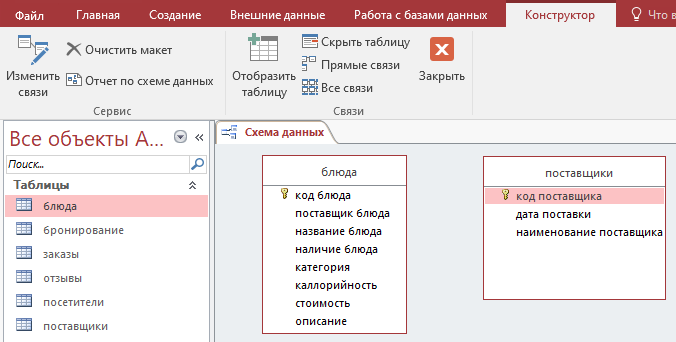


Рисунок 17 – выбранные таблицы

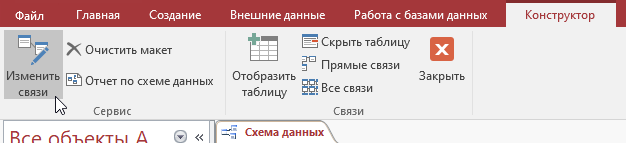


Рисунок 18 – кнопка изменения связей

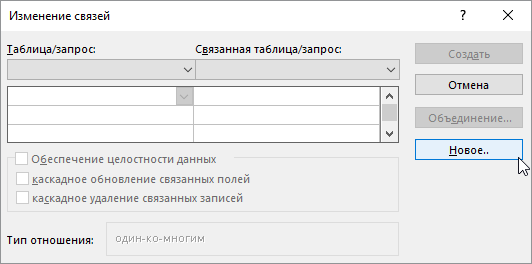


Рисунок 19 – создание новой связи в окне изменения связей

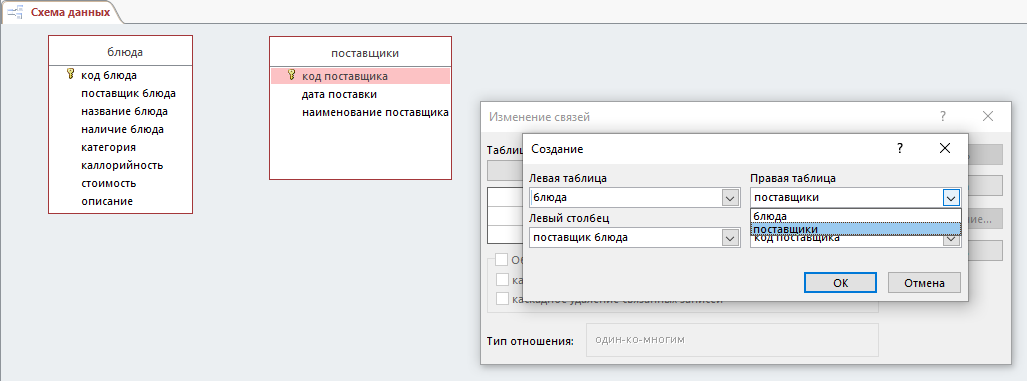


Рисунок 20 – окно создания связи с выбранными столбцами

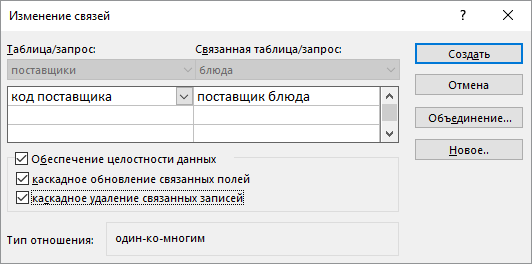


Рисунок 21 – изменение параметров данных в таблицах

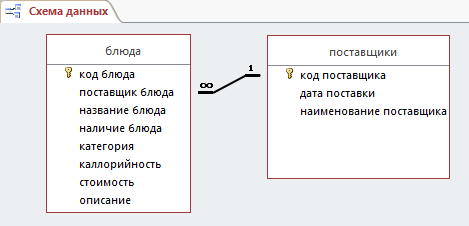


Рисунок 22 – созданная связи выбранных таблиц

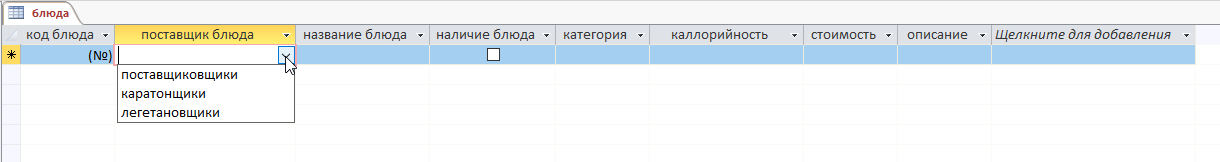


Рисунок 23 – подстановка в столбце при создании записи в таблице

# Создание запросов

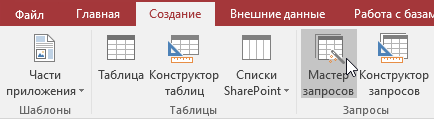


Рисунок № – кнопка для создания запросов

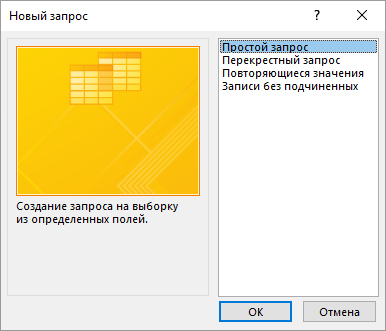


Рисунок № – окно создания запросов

# ЗАКЛЮЧЕНИЕ

# СПИСОК ЛИТЕРАТУРЫ

Федорова, Г. Н. Разработка, внедрение и адаптация программного обеспечения отраслевой направленности : учебное пособие / Г. Н. Федорова. — Москва : КУРС : ИНФРА-М, 2021. — 336 с.

Голицына, О. Л. Базы данных : учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. — 4-е изд., перераб. и доп. — Москва : ФОРУМ : ИНФРА-М, 2020. — 400 с.

Гагарина, Л. Г. Технология разработки программного обеспечения: учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Сидорова-Виснадул ; под ред. проф. Л. Г. Гагариной. - Электрон. текстовые дан. - Москва: ФОРУМ: ИНФРА-М, 2019. - 400 с.

Зубкова, Т. М. Технология разработки программного обеспечения : учебное пособие для СПО / Т. М. Зубкова. — Саратов : Профобразование, 2019. — 468 c.

Грекул, В. И. Управление внедрением информационных систем : учебное пособие для СПО / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. — Саратов : Профобразование, 2021. — 277 c.

Попова-Коварцева, Д. А. Основы проектирования баз данных: учеб. пособие / Д.А. Попова-Коварцева, Е.В. Сопченко. – Самара: Издательство Самарского университета, 2019. –112 с.: ил.