

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»

Кафедра информационных систем и программирования

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания по выполнению курсового проекта
для студентов всех форм обучения
направления подготовки 09.03.04 Программная инженерия

Краснодар
2019

Составитель: канд. техн. наук, доц., М.В. Янаева

УДК 681.31(031)

Проектирование информационных систем: методические указания по выполнению курсового проекта для студентов всех форм обучения направления подготовки 09.03.04 Программная инженерия / Сост.: М.В. Янаева; Кубан. гос. технол. ун-т. Каф. информационных систем и программирования. – Краснодар: 2019. – 41 с.

Методические указания содержат требования по выполнению курсового проекта, построению реляционной базы данных, обоснованию проектных решений, реализации типовых информационных запросов для заданной предметной области, реализации пользовательских приложений к базе данных средствами выбранной СУБД. При выполнении курсового проекта студент выполняет все этапы создания программного продукта: от постановки задачи до практической реализации, сопровождающейся инструкциями по его использованию.

Рецензенты: канд. техн. наук, доц. кафедры ИСП КубГТУ

В.А. Мурлина,

заведующий кафедрой «Компьютерных технологий и систем» ФГБОУ ВО КубГАУ им. академика И.Т. Трубилина,
проф. д-р.техн.наук, Лойко В.И.

Содержание

1	Нормативные ссылки	4
2	Требования к курсовому проекту	4
2.1	Общие положения	4
2.2	Средства программного обеспечения	6
2.3	Структура пояснительной записки.....	6
3	Пример реализации базы данных	9
4	Пример подключения базы данных с использованием С#	21
4.1	Пример подключения к MS SQL Server на С#.....	21
4.2	Создание простого приложения подключения к MS SQL Server с использованием визуальных средств проектирования	25
5	Варианты задания на курсовой проект	34
	Список рекомендуемой литературы.....	37
	Приложение А Форма титульного листа курсового проекта	38
	Приложение Б Форма задания на курсовой проект.....	39
	Приложение В Пример оформления реферата.....	40

1 Нормативные ссылки

В данных методических указаниях использованы ссылки на следующие нормативные документы:

- ГОСТ Р7.0.5-2008 СИБИД. Библиографическая ссылка. Общие требования и правила составления;
- ГОСТ Р1.5-2004. Стандарты национальные РФ. Правила построения, изложения, оформления и обозначения;
- ГОСТ 2.301-68 ЕСКД. Форматы;
- ГОСТ 7.82-2001 СИБИД. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления;
- ГОСТ 7.12-93 СИБИД. Библиографическая запись. Сокращения слов на русском языке. Общие требования и правила;
- ГОСТ 7.9-95 СИБИД. Реферат и аннотация. Общие требования.

2 Требования к курсовому проекту

2.1 Общие положения

Курс дисциплины «Проектирование информационных систем» предполагает выполнение студентами курсового проекта. Курсовой проект – серьезное учебно–научное исследование, предполагающее творческий подход студента к проработке его содержания и тщательность, грамотность оформления. Студент, выполняя курсовой проект, систематизирует, углубляет и закрепляет знания, полученные в процессе обучения.

Цель выполнения курсового проекта: применении студентами полученных теоретических знаний и практических навыков в проектировании и реализации реляционных баз данных в конкретной выбранной СУБД, а также проектирование приложения к базе данных средствами выбранной среды программирования.

В рамках выполнения курсового проекта предполагается:

- а) выполнение анализа процессов и структур данных конкретной предметной области;
- б) проектирование структуры базы данных;
- в) создание, заполнение, модификация, индексирование базы данных;
- г) построение desktop-го приложения;
- д) организация запросов различного назначения и типа;
- е) формирование необходимой отчетной документации.

Тематика курсовых проектов связана с проектированием базы данных и приложения для организации доступа к базе данных. В результате выполнения курсового проекта студент должен разработать приложение для работы с базой данных, реализующей, по крайней мере, пять типовых запросов к базе данных с использованием языка SQL (выборка, поиск, группировка, изменение данных, удаление).

Для достижения поставленной цели курсового проектирования студенту необходимо решить следующие задачи:

1. Выбрать тему курсового проекта согласно предложенным вариантам или в соответствии с научно – исследовательской работой, проводимой на кафедре.
2. Согласовать тему курсового проекта и составить план работ.
3. Выполнить проектирование базы данных.
4. Выполнить проектирование приложения.
5. Реализовать базу данных и приложение.
6. Оформить пояснительную записку к курсовому проекту.
7. Публично защитить результаты выполнения курсового проекта.

В пояснительную записку к курсовому проекту обязательно необходимо включать проектную часть, выполненную с использованием средств UML.

Результат выполнения курсового проекта оформляется в виде пояснительной записки. Общий объем пояснительной записки не должен превышать 40 листов, в том числе введение – не более 2 листов. К пояснительной записке прилагается носитель с созданными базами данных, отлаженными приложениями и иной необходимой информацией. Пояснительная записка к курсовому проекту должна содержать описание предметной области: основные процессы, наличие не менее 2–х входных документов и не менее 7–ми выходных документов (отчетов). Выходные данные должны формироваться при помощи запросов к базе данных не менее шести типов: выбор, поиск, расчеты, сортировка, группировка, добавление.

Изменение данных должно выполняться при помощи запросов: добавление, удаление, обновление, модификация.

Запросы в курсовом проекте должны быть как статическими, так и динамическими, формируемыми во время выполнения приложения, параметры которых определяются пользователем.

Записка иллюстрируется схемами, копиями экрана, выходными документами, листингом программного кода приложения.

В приложении необходимо предусмотреть возможность анализа данных в виде графиков или диаграмм на основе хранимых данных и результатов запроса. Копии форм анализа данных включаются в пояснительную записку.

2.2 Средства программного обеспечения

Рекомендуемые операционные системы: операционные системы семейства Windows, Linux.

Рекомендуемые среды проектирования: Star UML.

Рекомендуемые СУБД: MySQL, MS SQL Server.

Рекомендуемые инструментальные среды создания приложений: Microsoft Visual Studio.

2.3 Структура пояснительной записки

Пояснительная записка на листах формата А4 по ГОСТ 2.301 к курсовому проекту должна содержать следующие обязательные разделы:

- титульный лист (приложение А);
- задание на курсовой проект (приложение Б);
- реферат (приложение В), включающий количество страниц ПЗ, согласно ГОСТ 7.9 количество таблиц, рисунков, программ приложений, ключевые слова (прописными буквами), краткую характеристику и результаты работы;

- содержание;
- введение;
- основную часть:

- 1 Нормативные ссылки;
- 2 Описание предметной области
 - 2.1 Общее описание предметной области
 - 2.2 Назначение и цели создания ПО
 - 2.3 Требования к структуре ПО и базе данных
 - 2.4 Описание входных документов и сообщений
 - 2.5 Описание выходных документов и сообщений
 - 2.6 Список ограничений
- 3 Проектирование структуры базы данных
 - 3.1 Описание сущностей
 - 3.2 Построение концептуальной схемы базы данных
- 4 Реализация базы данных средствами выбранной СУБД
 - 4.1 Выбор СУБД
 - 4.2 Проектирование таблиц
 - 4.3 Построение ER- диаграммы
- 5 Проектирование структуры приложения
 - 5.1 Построение структурной схемы ПО
 - 5.2 Модель функционирования системы
 - 5.3 Диаграмма компонентов
 - 5.4 Диаграммы кооперации

- 5.5 Диаграмма размещения
- 5.6 Диаграмма состояний
- 5.7 Диаграммы потоков данных
- 5.8 Диаграмма классов
- 5.9 Диаграмма последовательности
- 5.10 Диаграмма прецедентов
- 6 Реализация приложения базы данных
 - 6.1 Выбор среды программирования
 - 6.2 Схема меню приложения
 - 6.3 Окна приложения или руководство пользователя
 - 6.4 Запросы к базе данных
 - 6.5 Функции и процедуры обработки данных
 - 6.6 Формирование отчетной документации
- 7 Тестирование программного продукта

- заключение;
- список использованных источников;
- приложения (листинг структуры файлов БД, листинг реализованных запросов, листинг компонент реализованных форм, отчетов, меню).

Все листы пояснительной записки следует сброшюровать и пронумеровать.

Далее приведем рекомендации по содержанию некоторых разделов пояснительной записки.

Раздел «*Введение*» включает краткое описание предметной области, цель проектирования и создания базы данных и приложения, предполагаемые средства проектирования базы данных и реализации, предполагаемые результаты и их применение в предметной области. Во введении должны содержаться сведения о наименовании системы, поставлены и определены задачи курсового проекта, выделены актуальность и практическая значимость решаемой задачи.

В разделе «*Описание предметной области*» необходимо провести подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами. Это описание должно корректно определять все взаимосвязи между объектами предметной области. Поскольку предметная область имеет сложную структуру и не упорядочена, необходимо ее ограничить: рассмотреть только определенную совокупность объектов и связей между ними.

В результате выполнения анализа предметной области следует:

- получить подробное описание информации об объектах предметной области, которая требуется для решения конкретных задач и которая должна храниться в БД;

– сформировать конкретные задачи, которые будут решены, с использованием данной базы данных с кратким описанием алгоритмов их решения в виде структурных блок-схем;

– выполнить описание выходных документов, которые должны генерироваться при работе системы;

– выполнить описание входных документов, которые служат основанием для заполнения данными БД.

Перед описанием документов необходимо привести описание основных процессов и построение диаграммы потоков данных или схемы документооборота.

Описание документа включает графическое изображение документа (отсканированная копия), описание его структуры и характеристик, нахождение в схеме документооборота предприятия, алгоритм и этапы его формирования, существующие ограничения на данные. Такое описание документов позволяет получить структуру сущностей и атрибутов предметной области. Но в первую очередь акцент при описании ставится на связи между объектами.

«*Концептуальная модель*» – концептуальное описание предметной области. Здесь должно быть дано описание выделяемых информационных объектов предметной области – сущностей и их атрибутов. Строится граф алгоритмической взаимосвязи показателей для устранения дублирования и возможной противоречивости. Поскольку будут использованы условные обозначения для определения алгоритмических зависимостей, приводится таблица условных обозначений, и дается подробное описание диаграмм и графа. При построении диаграмм «сущность – связь» выбирается одна из центральных сущностей, затем добавляются сущности и определяются связи между ними. В случае необходимости определяются атрибуты связи. Если система сложная, таких ER–диаграмм будет построено несколько. Таким образом, концептуальная модель включает в себя описание взаимосвязей между сущностями и ограничения, определенные для этих взаимоотношений. Модель не содержит ссылок и указаний на системы реализации базы данных и приложение.

В *заключении* должны быть четко сформулированы основные выводы и результаты проведенного исследования, намечены пути и цели дальнейшей работы над темой. Следует указать социальную ценность выполненного исследования и его практическую значимость.

Список использованных источников выполняется по ГОСТ Р 7.0.5, ГОСТ 7.82 и данным методическим указаниям.

3 Пример реализации базы данных

Рассмотрим пример создания базы данных средствами Microsoft SQL Server .

Следует заметить, что проектирование базы данных выполняется после получения концептуальной схемы базы данных и осуществления этапа нормализации.

Создание любой базы данных необходимо начать с запуска самого MS SQL Server, а именно SQL Server Management Studio (рисунок 3.1).

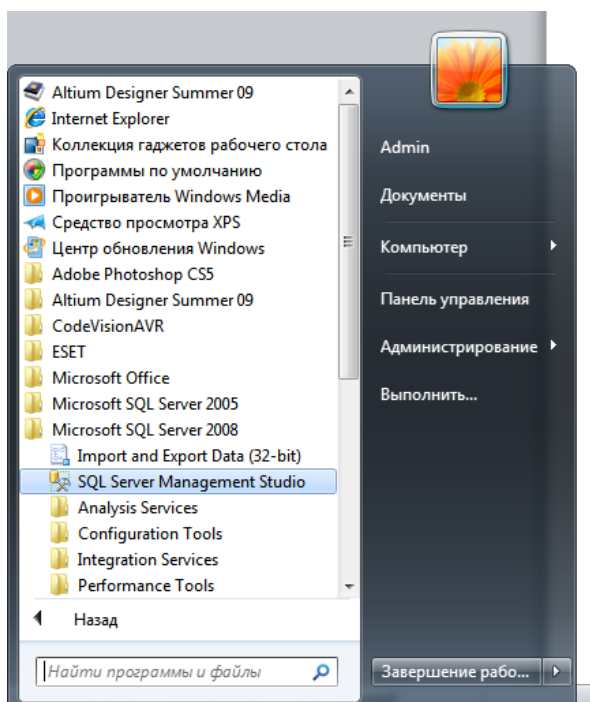


Рис. 3.1. Запуск SQL Server Management Studio

После запуска появляется диалоговое окно с выбором сервера, по умолчанию все поля заполнены данными, которые вводились при установке MS SQL Server, поэтому можно ничего не менять и просто нажать «Connect» (рисунок 3.2).

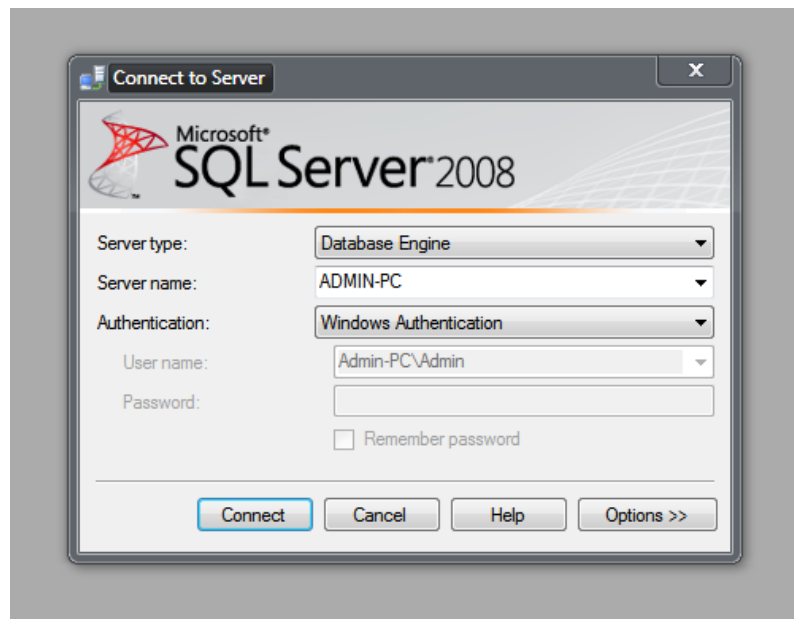


Рис. 3.2. Диалоговое окно подключения к серверу

После успешного подключения к серверу открывается окно Object Explorer (рисунок 3.3).

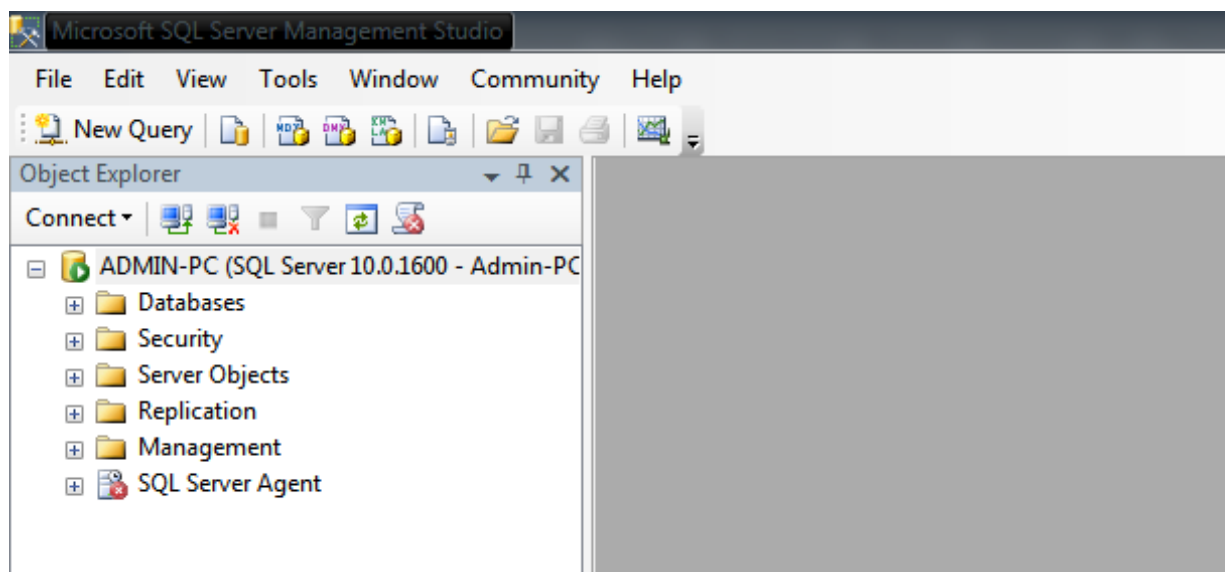


Рис.3.3. Окно Object Explorer

Для создания базы данных в окне Object Explorer выбираем объект «Databases» и пункт «New database...». Появится диалоговое окно создания новой базы данных (рисунок 3.4). Можно приступить к проектированию базы данных.

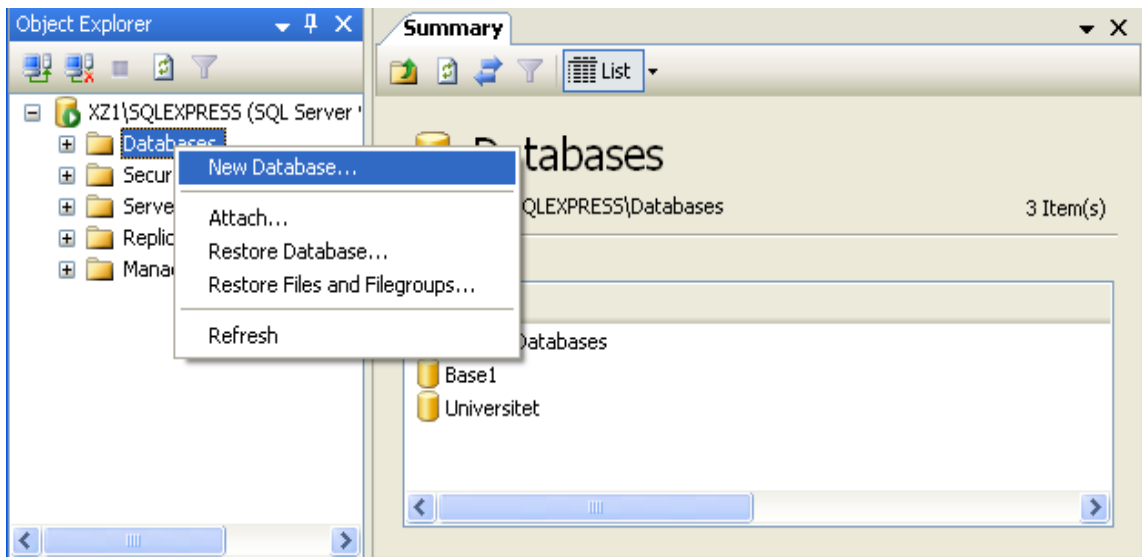


Рис. 3.4. Диалоговое окно создания новой базы данных

Вводим наименование базы данных в поле Database Name (рисунок 3.5).

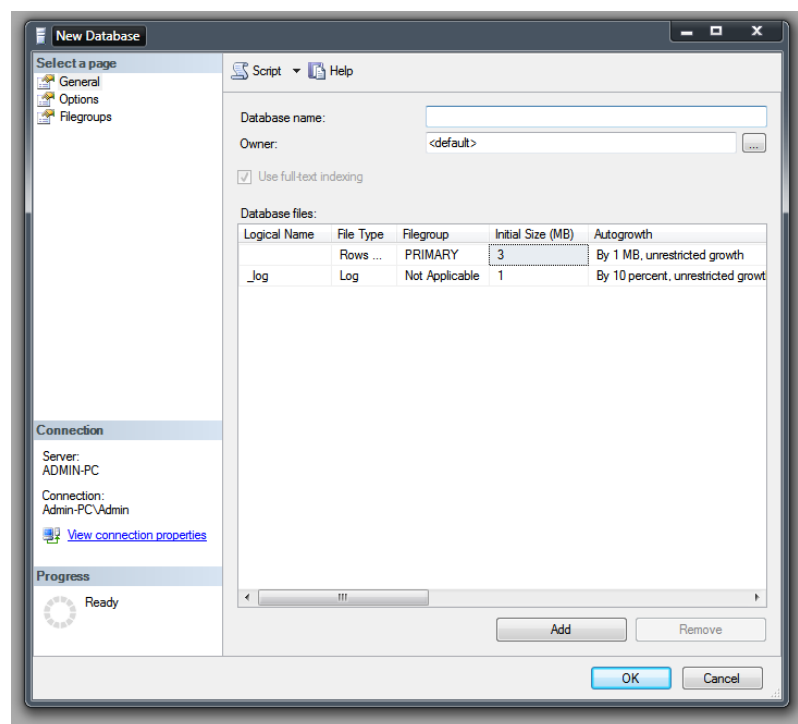


Рис. 3.5. Диалоговое окно New Database

В нашем примере будет проектироваться база данных автомобильного салона LadaService. После создания база данных отражается в дереве просмотра (рисунок 3.6):

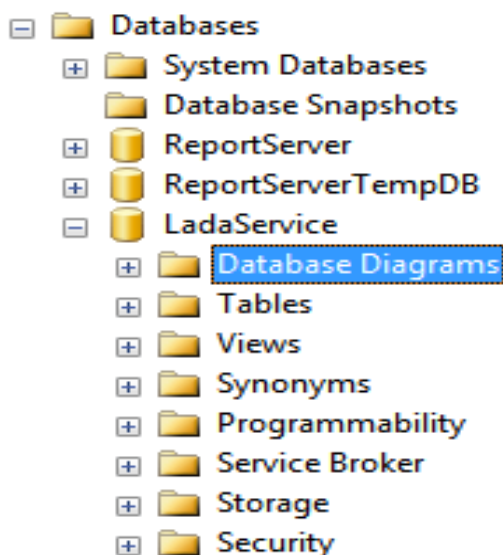


Рис. 3.6. Дерево просмотра

Для создания таблицы в окне «Object Explorer» выбираем созданную базу данных LadaService, и в раскрывшемся списке правой кнопкой мыши нажимаем на пункт «Tables», в котором находим и выделяем пункт «New table...» (рисунок 3.7).

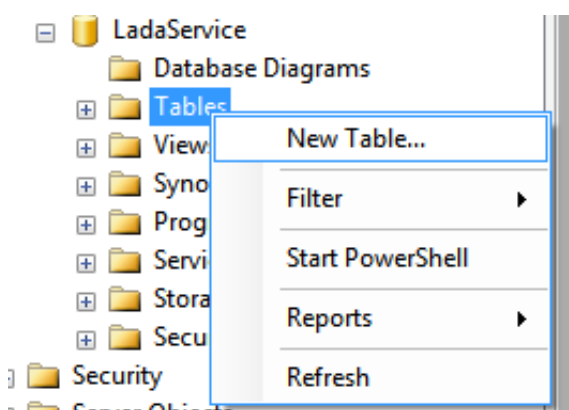


Рис. 3.7. Диалоговое окно создания таблицы БД

В рабочей области СУБД MS SQL Server откроется форма для создания таблицы. В столбце Column Name указывается название поля таблицы, в столбце Data Type – указывается тип столбца. В столбце Allow Nulls устанавливается флажок, предоставляющий возможность оставлять ячейку незаполненной (пустой) (рисунки 3.8, 3.9, 3.10).

При создании таблицы для последующей связи с другими таблицами необходимо задавать ключевые и индексные поля, но об этом речь пойдет немного позднее.

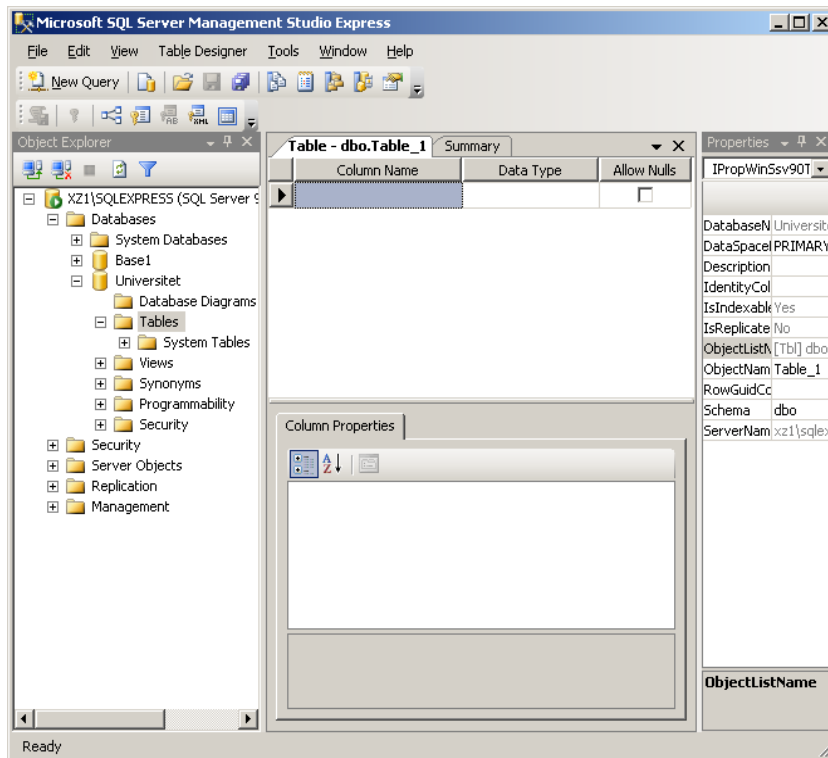


Рис. 3.8. Диалоговое окно Object Explorer

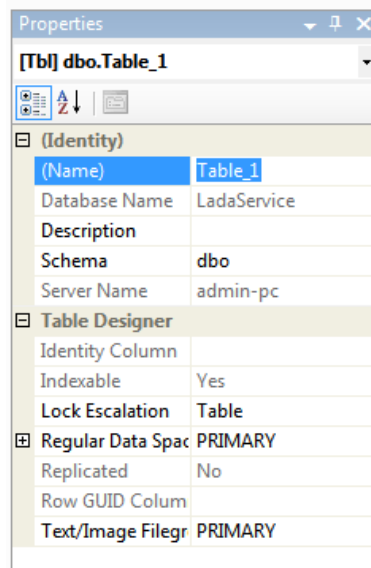


Рис. 3.9. Окно Properties

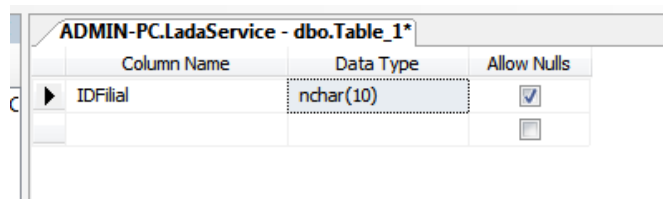


Рис. 3.10. Задание типа столбца таблицы БД

После ввода информации о всех столбцах таблицы нажимаем на кнопку «Сохранить» или **Ctrl + S** и вводим название таблицы. Таблица отразится в составе базы данных (рисунок 3.11).

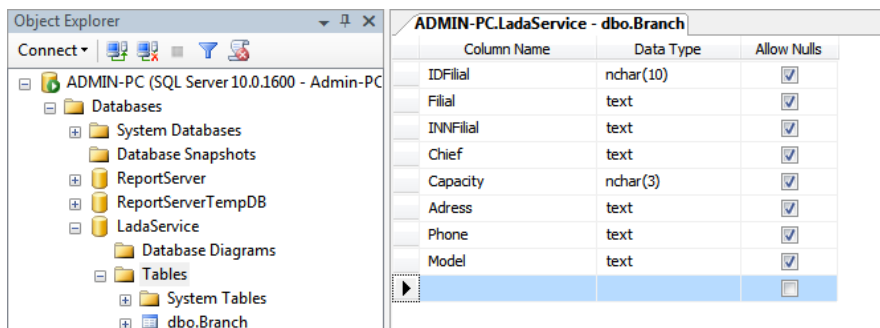


Рис. 3.11. Проектирование таблиц БД

Аналогичным образом проектируем все таблицы базы данных, в нашем примере их четыре (рисунок 3.12).

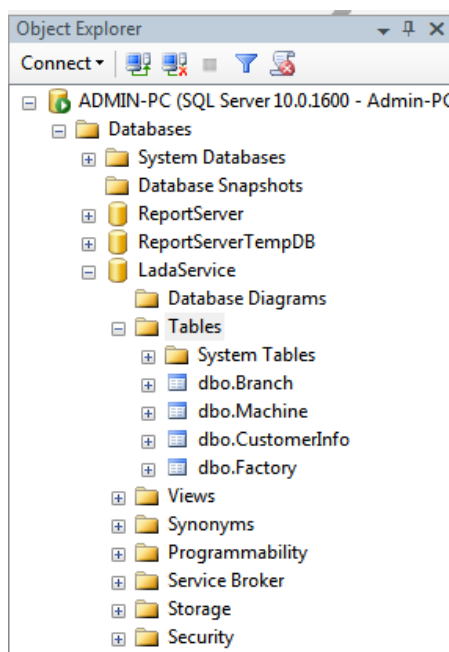


Рис. 3.12. Структура БД «LadaService»

Для заполнения любой таблицы в структуре БД необходимо щелкнуть правой кнопкой мыши на таблице, в которую вносятся данные, и выбрать пункт контекстного меню «Edit Top 200 Rows» (рисунок 3.13).

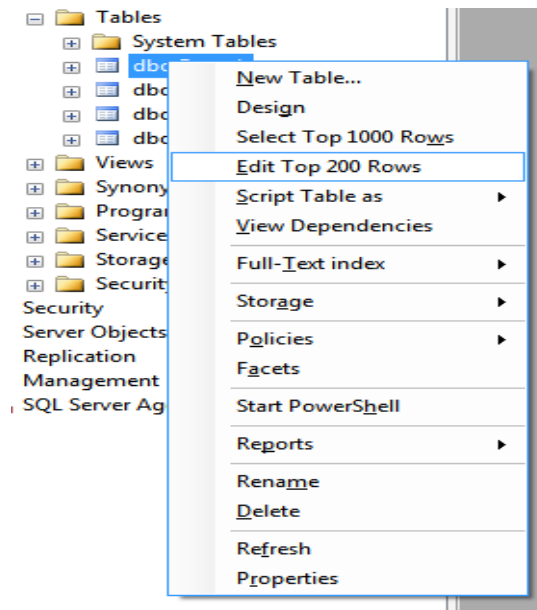


Рис. 3.13. Выбор режима редактирования таблицы

На экране появляется таблица с ранее определенными колонками. Для того чтобы заполнить строку таблицы нажимаем на ячейки с содержимым NULL и последовательно вводим информацию.

Пример заполненной таблицы БД «LadaService» представлен на рисунке 3.14.

IDFilial	Filial	INNFilial	Chief	Adress	Phone	Model	Capacity
0	Lada Center	234-234-4	Grigorin	Lenin Street	34-322-1	M-01	24
0	Lada Center	234-234-4	Grigorin	Lenin Street	34-322-1	M-02	24
0	Lada Center	234-234-4	Grigorin	Lenin Street	34-322-1	M-03	24
1	Lada Mini	322-858-2	Folov	Rigor Street	34-111-1	M-04m	6
2	Lada Priora	267-233-2	Rogkoff	Gordin Street	28-099-2	M-05	10
2	Lada Priora	267-233-2	Rogkoff	Gordin Street	28-099-2	M-06	10
3	Lada L	985-233-9	Nixto	Qwerty Street	98-322-4	M-07	2
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рис. 3.14. Пример заполнения таблицы БД «LadaService»

Аналогичным образом можно внести информацию во все таблицы базы данных.

Теперь коснемся вопроса задания ключевых полей базы данных. Назначать ключевые поля можно еще в процессе создания таблицы. Если же таблица сохранена и для нее необходимо задать ключевое поле, то ее следует заново открыть в режиме проектирования.

Для того чтобы назначить поле ключевым, необходимо в режиме проектирования таблицы щелкнуть правой кнопкой мыши на необходимом поле и выбрать пункт в контекстном меню Set Primary Key – тем самым назначая это поле ключевым (рисунок 3.15).

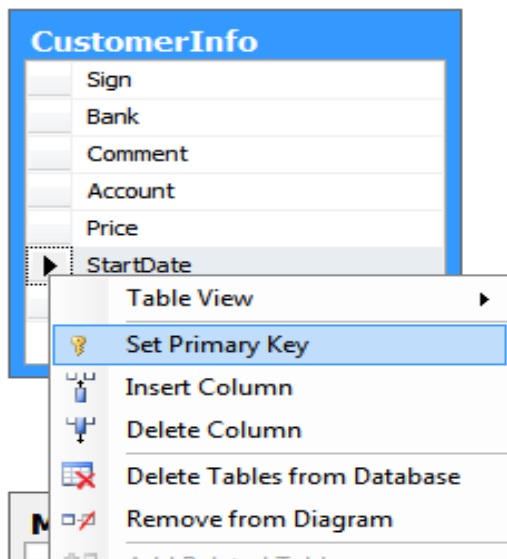


Рис. 3.15. Задание ключевого поля

Не все поля могут быть ключевыми, это зависит от типа и данных, содержащихся в таблице. Данные содержащиеся в данном столбце, должны быть уникальными.

После нажатия и успешного назначения (если полю разрешено быть ключом) слева от названия поля должен появиться ключик (рисунок 3.16).

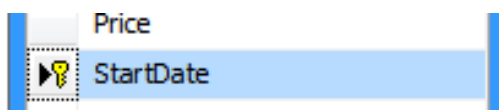


Рис. 3.16. Пример ключевого поля

Для создания связей между таблицами необходимо построить ER–диаграмму. Для этого необходимо щелкнуть правой кнопкой мыши на пункте Database Diagrams в дереве слева (рисунок 3.17).

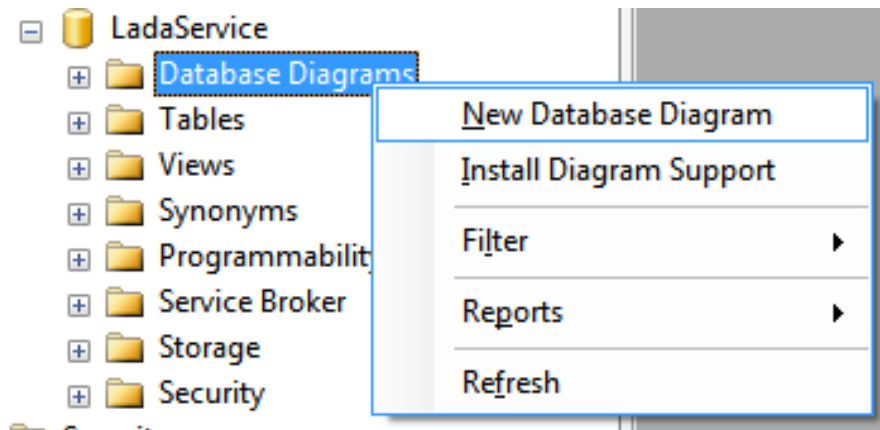


Рис. 3.17. Создание ER–диаграммы

В открывшемся диалоговом окне посредством кнопки «Add» осуществляем выбор необходимых нам таблиц, связи между которыми предстоит построить (рисунок 3.18).

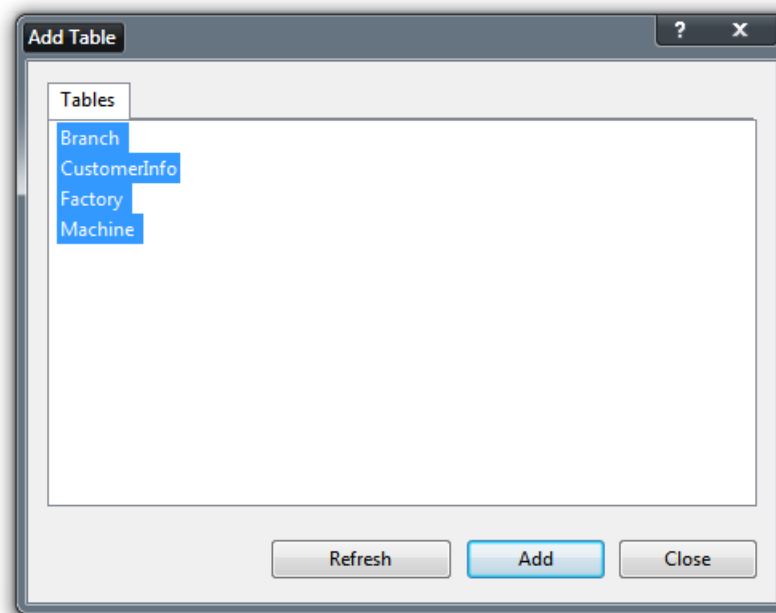


Рис. 3.18. Выбор таблиц для построения ER–диаграммы

Расставляем выбранные таблицы (путем перетаскивания за заголовки отдельно взятой таблицы) таким образом, чтобы связи между ними были более наглядны и отражали суть взаимосвязей таблиц в базе данных (рисунок 3.19).

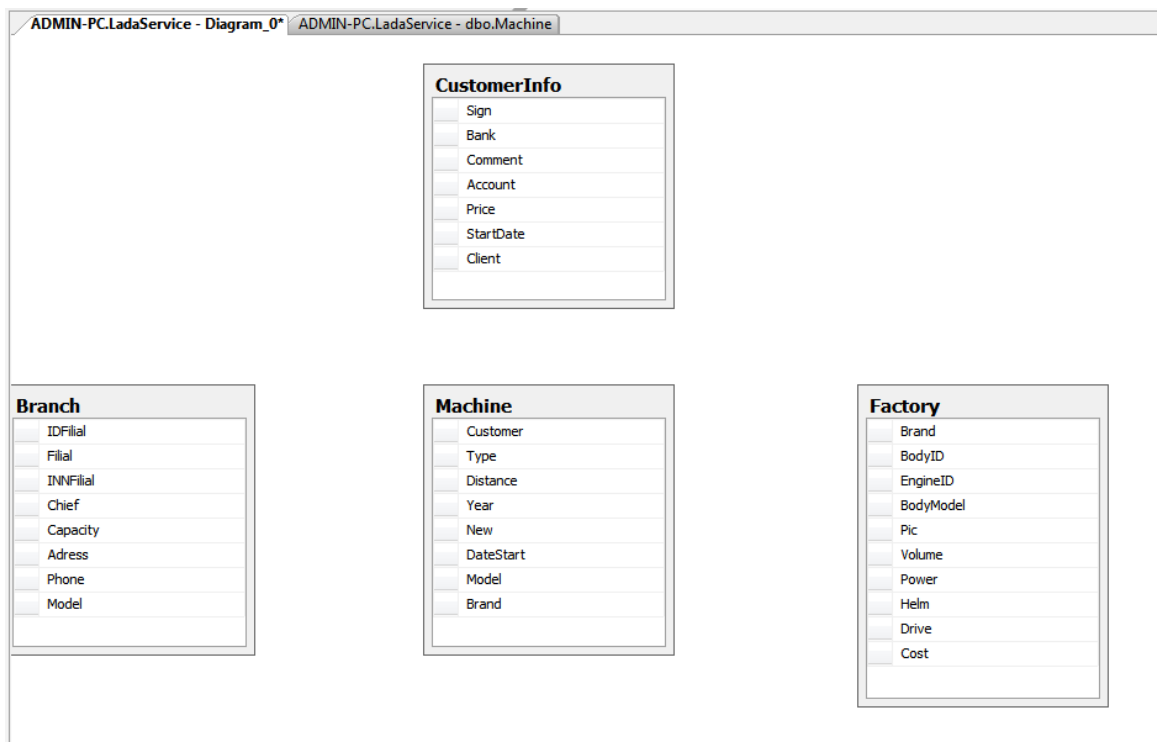


Рис. 3.19. Расположение таблиц базы данных

Теперь перейдем к процессу создания связей между таблицами.левой кнопкой мыши просто щелкаем на поле с ключом и, не отпуская кнопку мыши, тянем к ключевому или индексному полю таблицы, с которой необходимо установить связь (за мышью будет появляться след пунктирной линии). Процесс создания связей продемонстрирован на рисунке 3.20.

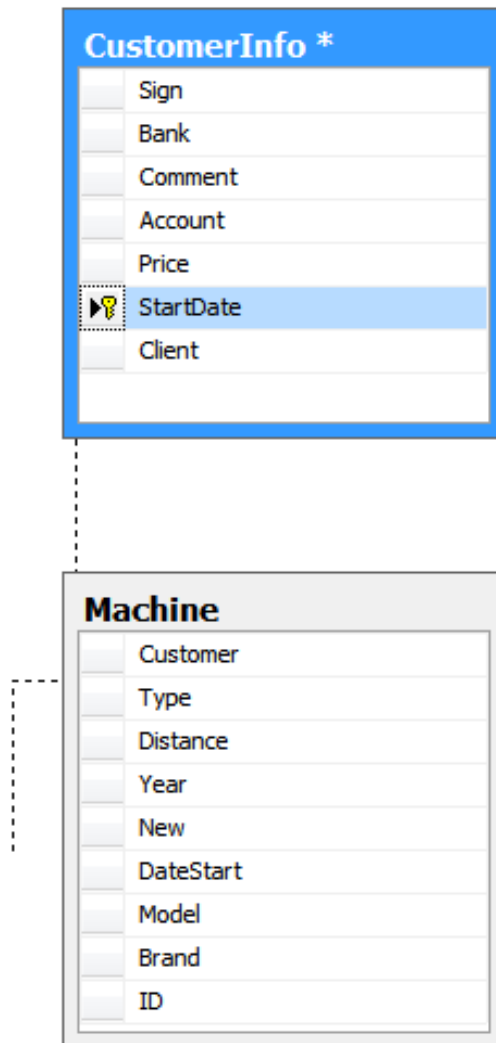


Рис. 3.20. Процесс установления связей между таблицами

После наведения на необходимое поле таблицы, с которой нужно установить связь, отпускаем левую кнопку мыши. На экране появляется диалоговое окно создания связи между полями. Вначале уточняем связующие поля, между которыми устанавливается связь, при необходимости вносим поправки, уточняем имя связи и задаем связь (рисунок 3.21).

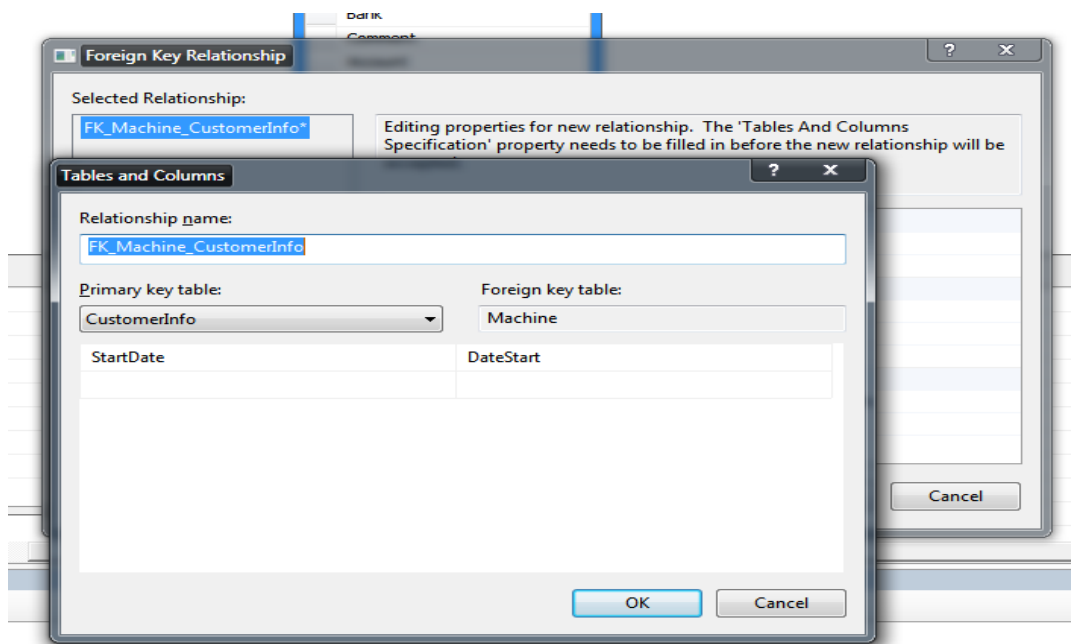


Рис. 3.21. Диалоговое окно задания связи между таблицами БД

После подтверждения создания связи между таблицами на диаграмме отражается линия связи (рисунок 3.22).

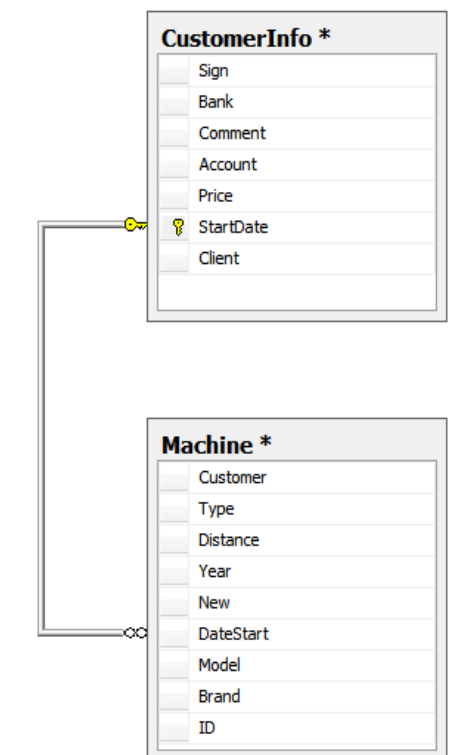


Рис. 3.22. Линия связи между таблицами БД

По аналогии создаем все виды связей между таблицами базы данных (рисунок 3.23).

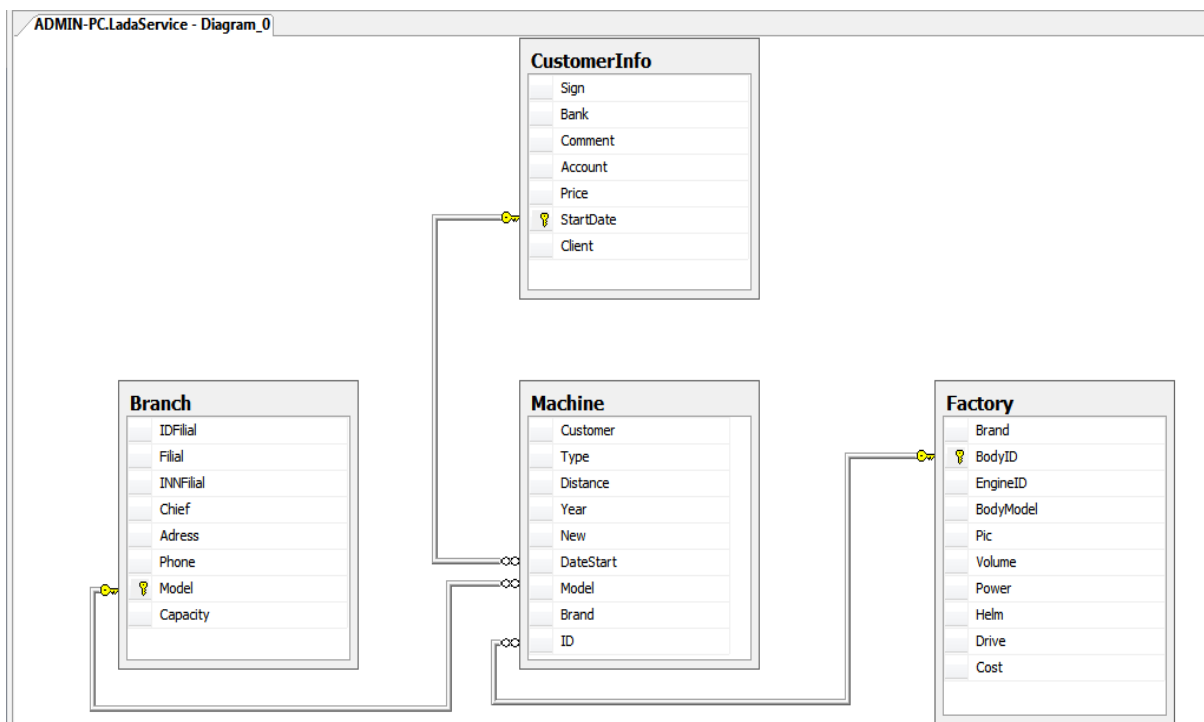


Рис. 3.23. Реляционная модель БД в виде ER–диаграммы

Сохраняем созданную диаграмму (путём нажатия Ctrl + S) – рисунок 3.24.

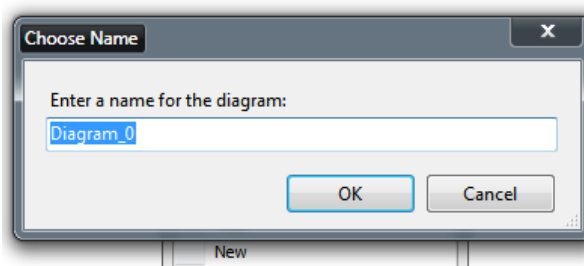


Рис. 3.24. Диалоговое окно задания имени ER–диаграммы

База данных спроектирована и готова к подключению к приложению.

4 Пример подключения базы данных с использованием C#

4.1 Пример подключения к MS SQL Server на C#

Приведем пример программного кода на C# подключения к MS SQL Server, выборки данных из какой-либо таблицы и возможного изменения данных в ней.

Создаем пустой, не оконный проект, и вводим следующий текст программы:

```

// Подключаем стандартные пространства имен
using System;

// Пространства имен для создания формы и объектов внутри формы
using System.Windows.Forms;
using System.Drawing;

// Пространства имен для оперирования с данными
using System.Data;

// Пространство имен для работы с базами данных MS SQL Server
using System.Data.SqlClient;

// Непосредственное описание формы
class DataInOutGrid : Form
{
    SqlDataAdapter dataAdapter; // добавляем дата-переменную для хранения и
    // временного редактирования данных из базы данных
    DataGridView dataGridView; // добавляем дата-переменную для
    // отображения из нее данных в компонент DataGridView
    static void Main() // Главная функция, с которой начинается выполнение
    // программы
    {
        // Создаем и запускаем форму
        Application.Run(new DataInOutGrid()); // вызываем функцию
        DataInOutGrid(); }
    public DataInOutGrid() // функция создания объектов, подключения к базе и
    // наполнения компонента DataGridView данными из базы данных
    {
        // Изменяем размеры формы окна
        this.Width = 900;
        this.Height = 550;

        // Указываем заголовок окна
        this.Text = " Пример подключения к SQL Server на C# ";

        // Далее добавляем элементы
        // Добавляем элемент DataGridView на форму
        dataGridView = new DataGridView();
        dataGridView.Width = 860;
        dataGridView.Height = 450;
        dataGridView.Location = new Point(10, 10);
        dataGridView.AutoSizeColumns();
        this.Controls.Add(dataGridView);

        // Добавляем командную кнопку
        Button buttonSave = new Button();

```

```

buttonSave.Location = new Point(340, 480);
buttonSave.Width = 220;
buttonSave.Text = " Сохранить изменения в базе данных ...";
buttonSave.Click += new System.EventHandler(ButtonSave_Click);
buttonSave.Parent = this;

// Формируем запрос к базе данных - делаем простую выборку из таблицы
Branch
string sql = "SELECT * FROM Branch";

// DataTable сохраняет данные в памяти как таблицу
DataTable dataTable = new DataTable();

// Подключаемся к базе данных LadaService (строка для SQL Server)
string connectionString = "data source = localhost; Initial Catalog =
LadaService;" + "Integrated Security = SSPI";
SqlConnection connection = new SqlConnection(connectionString);

//Открываем соединение
connection.Open();

//Создаем команду для запроса интересующих нас данных
SqlCommand sqlCommand = new SqlCommand(sql, connection);

//Создаем адаптер
// примечание : DataAdapter - посредник между базой данных и DataSet
(изменений вносимых пользователем)
dataAdapter = new SqlDataAdapter(sqlCommand);

//Создаем построитель команд
//Для адаптера становится доступной команда Update (возможны
добавление, удаление, корректировка строк таблицы)
SqlCommandBuilder commandBuilder =
new SqlCommandBuilder(dataAdapter);

// Данные из адаптера поступают в DataTable
dataAdapter.Fill(dataTable);

// Связываем данные с элементом DataGridView
dataGridView.DataSource = dataTable;

// Завершаем соединение с SQL Server
connection.Close(); }
void ButtonSave_Click(object sender, System.EventArgs args) //
Функция обработки нажатия кнопки
{ try // Пытаемся записать изменения в SQL Server

```

```

        {dataAdapter.Update((DataTable)dataGridView.DataSource); //
открываем соединение
        MessageBox.Show("Изменения в базе данных выполнены!",
"Уведомление о результатах", MessageBoxButtons.OK); // при успешном
соединении и обновлении БД в SQL Server выдаем сообщение }
        catch(Exception) // ловим ошибку при неудачном соединении
        { MessageBox.Show("Изменения в базе данных выполнить не
удалось!", "Уведомление о результатах", MessageBoxButtons.OK); // выводим
сообщение об ошибке сохранения данных
        }
    }
}

```

После успешной компиляции и запуска, программа сразу же попытается подключиться к SQL Server и считать данные с таблицы, после чего выведет их в компонент DataGridView, в противном случае, если при подключении к БД возникли ошибки, Visual Studio укажет на каком именно этапе подключения они возникли (рисунок 4.1)

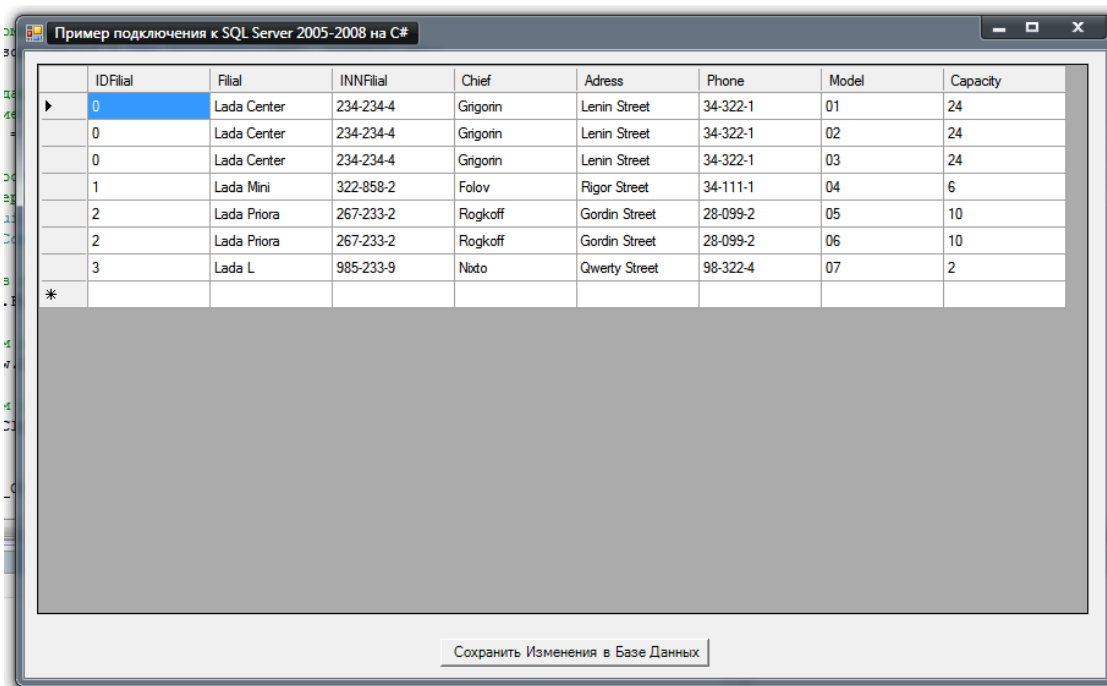


Рис. 4.1. Пример простого приложения подключения к БД

В приложении возможно редактирование и добавление данных в компоненте DataGridView. Сохраним изменения путем нажатия на кнопку «Сохранить изменения в базе данных», в подтверждение удачного или неудачного сохранения должно появиться диалоговое окно с соответствующим сообщением.

Простое приложение подключения к базе данных создано и вполне функционально.

4.2 Создание простого приложения подключения к MS SQL Server с использованием визуальных средств проектирования

Используя Form Designer в Visual Studio, для начала создаём новый проект (рисунок 4.2).

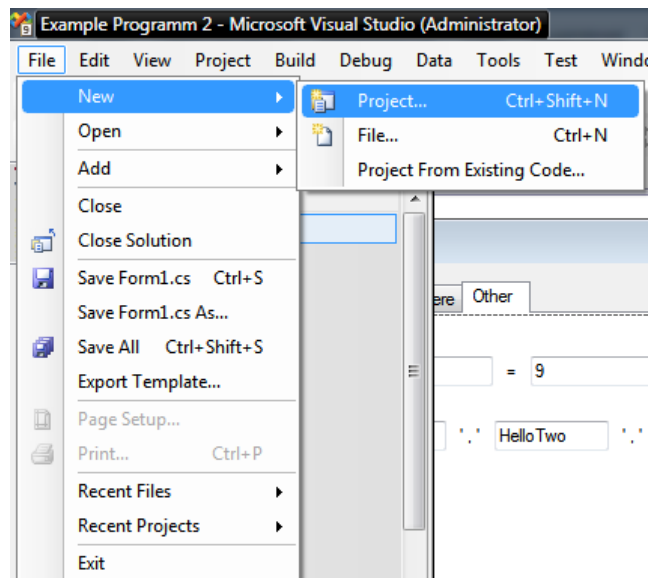


Рис. 4.2. Создание нового проекта

Указываем путь и имя проекта, не забывая выбрать проект Windows Forms Application (рисунок 4.3).

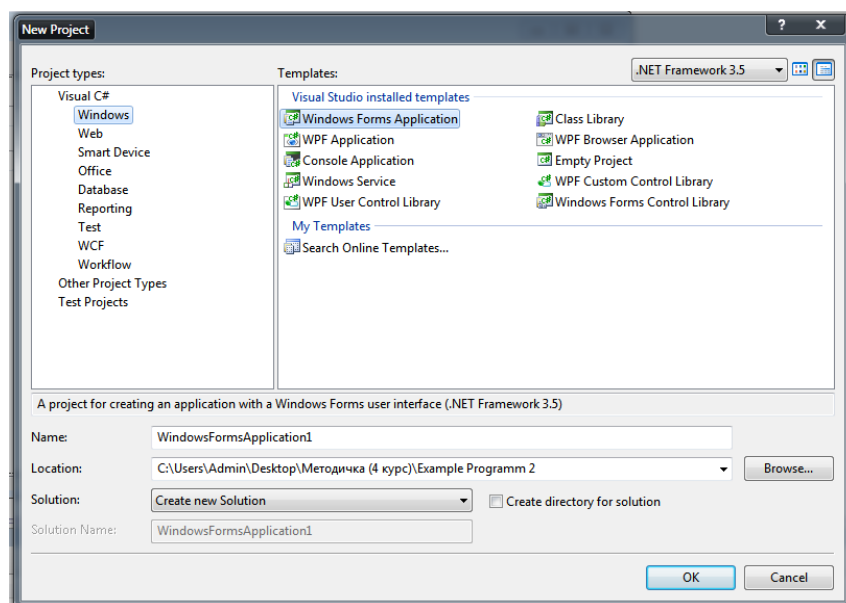


Рис. 4.3. Создание проекта Windows Forms Application

Получаем простую пустую форму приложения (рисунок 4.4).

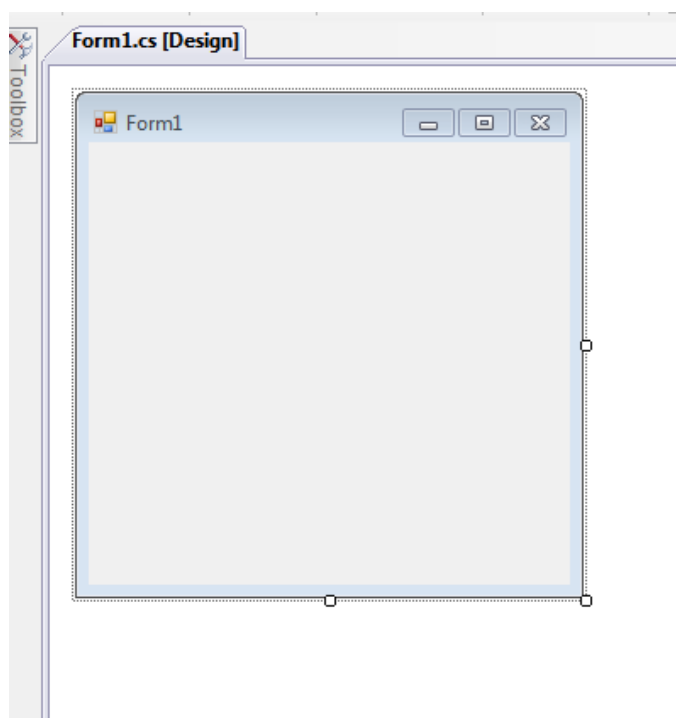


Рис. 4.4. Пустая форма приложения Windows Forms Application

Используя панель «Toolbox», добавляем на форму основные элементы интерфейса (рисунок 4.5):

- `tabControl` – элемент, предназначенный для размещения на форме вкладок;
- `dataGridView` – элемент, позволяющий отображать данные из БД в табличном виде;
- `Button` – элемент кнопка;
- `comboBox` – элемент-меню в виде списка для выбора таблицы;
- `textBox` – элемент поля редактирования, предназначенный для ввода данных;
- `Label` – элемент-метка, предназначенный для создания надписей на форме;
- `checkBox` – элемент, предназначенный для отметки опции посредством установки галочки.

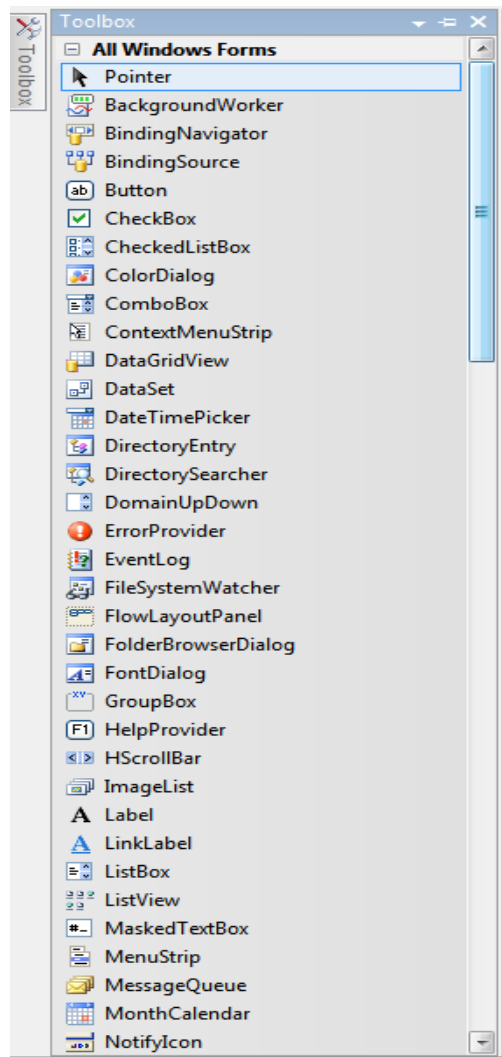


Рис. 4.5. Панель «Toolbox»

Для элемента `tabControl` из контекстного меню, вызываемого щелчком правой кнопки мыши на заголовке, выбираем пункт `Add Tab` и добавляем необходимое количество вкладок. В пункте `Properties` можно отдельно настроить каждую вкладку и задать ей имя (рисунок 4.6).

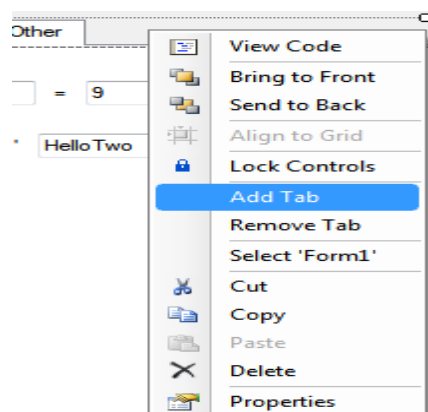


Рис. 4.6. Настройка элемента элемента `tabControl`

После добавления элемента `tabControl` и установки 4 вкладок, переходим к наполнению страниц, на первой размещаем элемент `dataGridView` и необходимые кнопки. На второй вкладке – элемент `dataGridView`, кнопку и один `comboBox`. На третьей вкладке – элементы `dataGridView`, `Button`, `comboBox`, несколько `Label`, а также несколько `textBox`. И на последней вкладке помещаем 2 кнопки, элементы `textBox`, метки и `CheckBox`. Пример окна проектирования одной из вкладок приведен на рисунке 4.7.

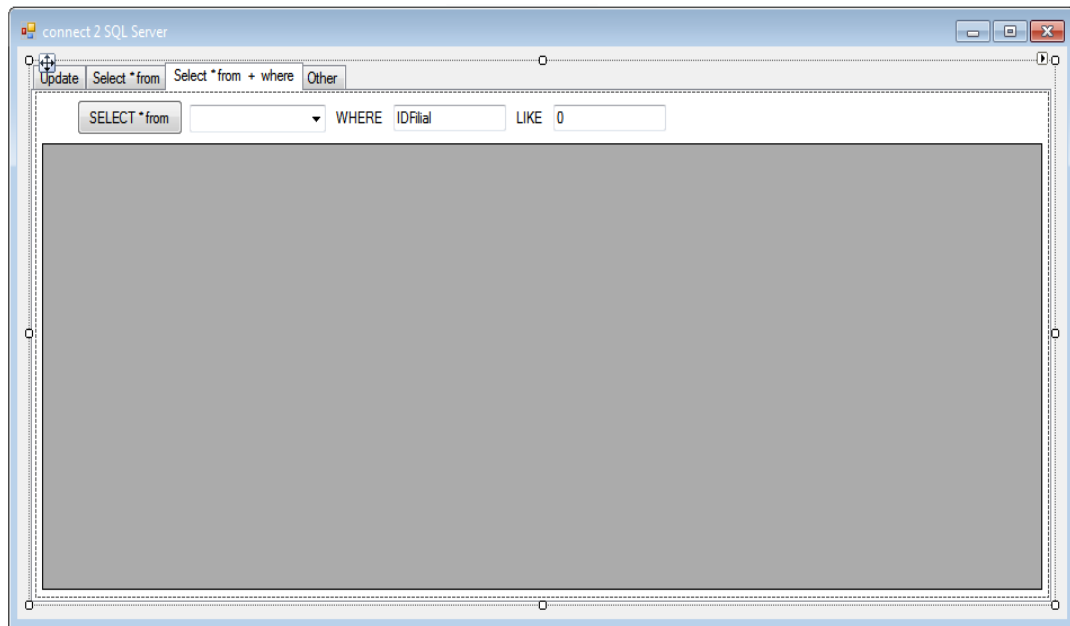


Рис. 4.7. Пример окна редактирования одной из вкладок

Для настройки того или иного элемента соответствующие настройки появляются при его выделении в окне `Properties` (рисунок 4.8).

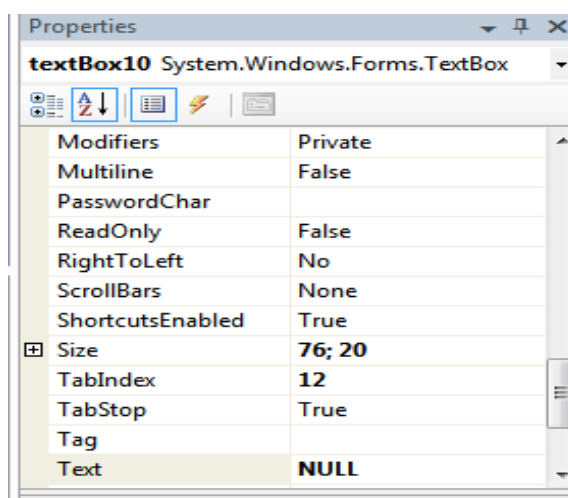


Рис. 4.8. Окно `Properties`

Для создания программного кода обработки спроектированной формы щелчком правой кнопки мыши на Form1.cs выбираем пункт View Code (рисунок 4.9).

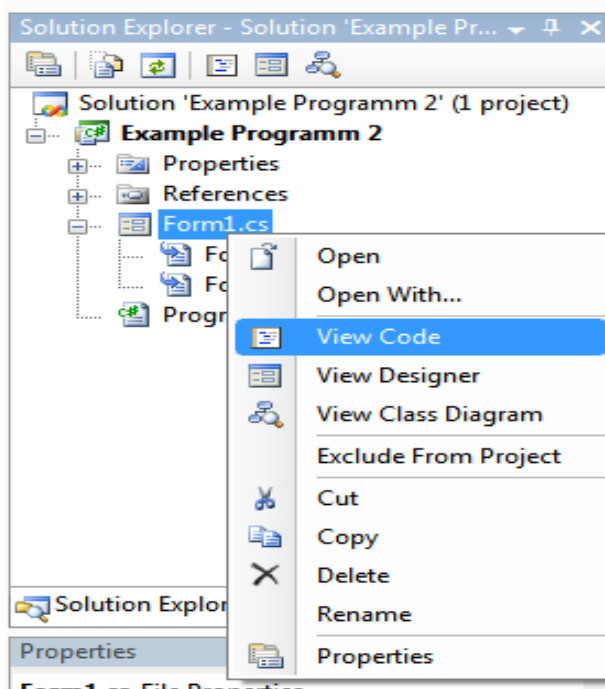


Рис. 4.9. Добавление программного кода обработки формы

Пример программного кода

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
namespace Example_Programm_2
{ public partial class Form1 : Form
    {SqlDataAdapter dataAdapter; // добавляем дата-переменную для хранения
и временного редактирования данных из базы данных
    SqlDataAdapter dataAdapter2;
    SqlDataAdapter dataAdapter3;
    public Form1()
    { InitializeComponent();
      SQLConn(); }
    private void button1_Click(object sender, EventArgs e)
```

```

        { // добавляем вопросительный диалог для кнопки выхода
          if ( MessageBox.Show(" Do u realy wanna exit ? ", " some message
..", MessageBoxButtons.OKCancel , MessageBoxIcon.Question ) ==
DialogResult.OK )
            { Close(); } }
private void SQLConn()
    { string sql = "SELECT * FROM Branch";
      string connectionString = "data source = localhost; Initial Catalog =
LadaService;" + "Integrated Security = SSPI";
      SqlConnection connection = new SqlConnection(connectionString);
      DataTable dataTable = new DataTable();
      SqlCommand sqlCommand = new SqlCommand(sql, connection);
      connection.Open();
      dataAdapter = new SqlDataAdapter(sqlCommand);
      SqlCommandBuilder      sqlCommandBuilder      =      new
SqlCommandBuilder(dataAdapter);
      dataAdapter.Fill(dataTable); // наполняем таблицу dataGridView1
      dataGridView1.DataSource = dataTable;
      connection.Close(); // забываем закрывать соединение
      comboBox1.Items.Add("Branch"); // добавляем имена таблиц в comboBox
      comboBox1.Items.Add("CustomerInfo");
      comboBox1.Items.Add("Factory");
      comboBox1.Items.Add("Machine");
      comboBox2.Items.Add("Branch");
      comboBox2.Items.Add("CustomerInfo");
      comboBox2.Items.Add("Factory");
      comboBox2.Items.Add("Machine"); }
private void button2_Click(object sender, EventArgs e)
    { try // Пытаемся записать изменения в SQL Server
      { // открываем соединение
        dataAdapter.Update((DataTable)dataGridView1.DataSource); }
      catch (Exception) // при ошибке выводим сообщение
        { MessageBox.Show("Changes Save False!", " some message..",
MessageBoxButtons.OK); } }
private void button3_Click(object sender, EventArgs e)
    { string sql = "SELECT * FROM " + comboBox1.Text; // выборка из
определённой таблицы, заданной из comboBox1
      string connectionString = "data source = localhost; Initial Catalog =
LadaService;" + "Integrated Security = SSPI";
      SqlConnection connection = new SqlConnection(connectionString);
      DataTable dataTable2 = new DataTable();
      SqlCommand sqlCommand = new SqlCommand(sql, connection);
      connection.Open();

```

```

        dataAdapter2 = new SqlDataAdapter(sqlCommand);
        SqlCommandBuilder      commandBuilder      =      new
SqlCommandBuilder(dataAdapter);
        dataAdapter2.Fill(dataTable2); // наполняем таблицу dataGridView2
        dataGridView2.DataSource = dataTable2;
        connection.Close(); }
private void button4_Click(object sender, EventArgs e)
{ string sql = "SELECT * FROM " + comboBox2.Text + " WHERE " +
textBox1.Text + " LIKE " + textBox2.Text;
    string connectionString = "data source = localhost; Initial Catalog =
LadaService;" + "Integrated Security = SSPI";
    SqlConnection connection = new SqlConnection(connectionString);
    DataTable dataTable3 = new DataTable();
    SqlCommand sqlCommand = new SqlCommand(sql, connection);
    connection.Open();
    dataAdapter3 = new SqlDataAdapter(sqlCommand);
    SqlCommandBuilder      commandBuilder      =      new
SqlCommandBuilder(dataAdapter);
    dataAdapter3.Fill(dataTable3); // наполняем таблицу dataGridView3
    dataGridView3.DataSource = dataTable3;
    connection.Close(); }
private void button5_Click(object sender, EventArgs e)
{ // удаление из таблицы Branch где textBox3 такой же как и в textBox4
    string connectionString = "data source = localhost; Initial Catalog =
LadaService;" + "Integrated Security = SSPI";
    SqlConnection connection = new SqlConnection(connectionString);
    SqlCommand myCommand = new SqlCommand ("DELETE Branch
WHERE " + textBox3.Text + " = " + textBox4.Text, connection);
    connection.Open();
    myCommand.ExecuteNonQuery(); // выполнение команды ТОЛЬКО после
открытия соединения
    connection.Close(); }
private void button6_Click(object sender, EventArgs e)
{ // вставляем в таблицу Branch строку с данными из textBox5 - textBox12
    string connectionString = "data source = localhost; Initial Catalog =
LadaService;" + "Integrated Security = SSPI";
    SqlConnection connection = new SqlConnection(connectionString);
    SqlCommand myCommand2 = new SqlCommand ("INSERT INTO
Branch Values (" + textBox5.Text + ", " + textBox6.Text + ", " +
textBox7.Text + ", " + textBox8.Text + ", " + textBox9.Text + ", " +
textBox10.Text + ", " + textBox11.Text + ", " + textBox12.Text + ")",
connection);
    try // пытаемся записать новую строку

```

```

{ connection.Open();
  myCommand2.ExecuteNonQuery(); }
catch (Exception EX) // при ошибке записи новой строки выводим сообщение
{ if (checkBox1.Checked) MessageBox.Show("" + EX, " some
message..", MessageBoxButtons.OK); // подробное описание произошедшей ошибки
else MessageBox.Show("Changes Save False!"," some message..",
MessageBoxButtons.OK); }
connection.Close(); } } }

```

После успешного компилирования и запуска программа соединяется с БД и читает данные из таблицы Branch в DataGridView1 первой вкладки приложения, где можно осуществлять операции удаления, добавления, изменения строк и сохранять изменения посредством нажатия кнопки «Save Change» (рисунок 4.10).

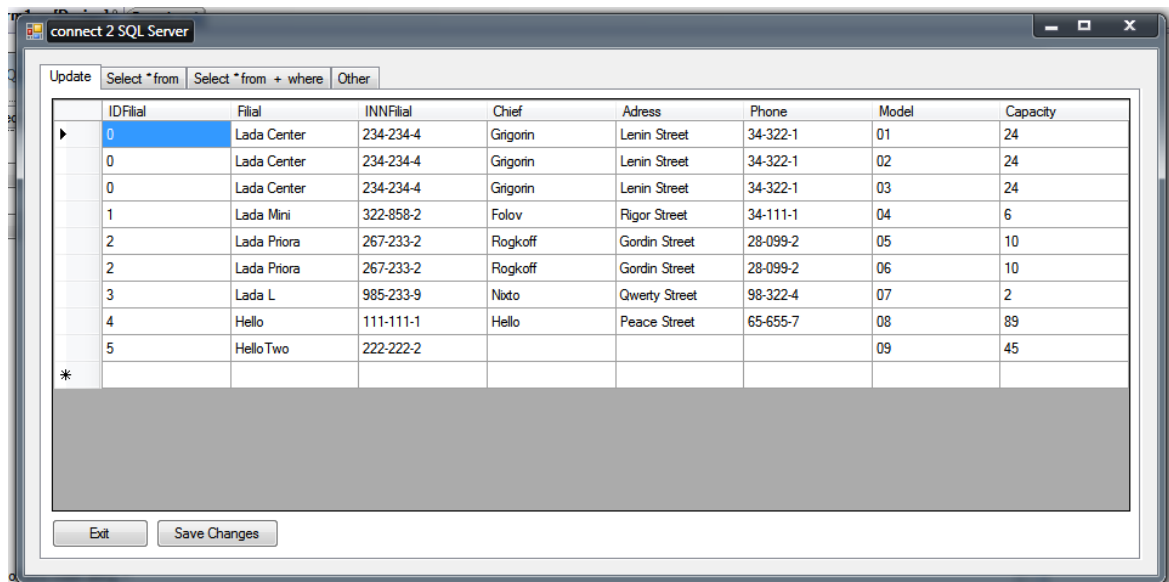


Рис. 4.10. Форма приложения

На второй вкладке основной формы возможно осуществлять выборку из определенной таблицы, выбирая её с помощью компонента comboBox и нажимая кнопку «SELECT * from» (рисунок 4.11).

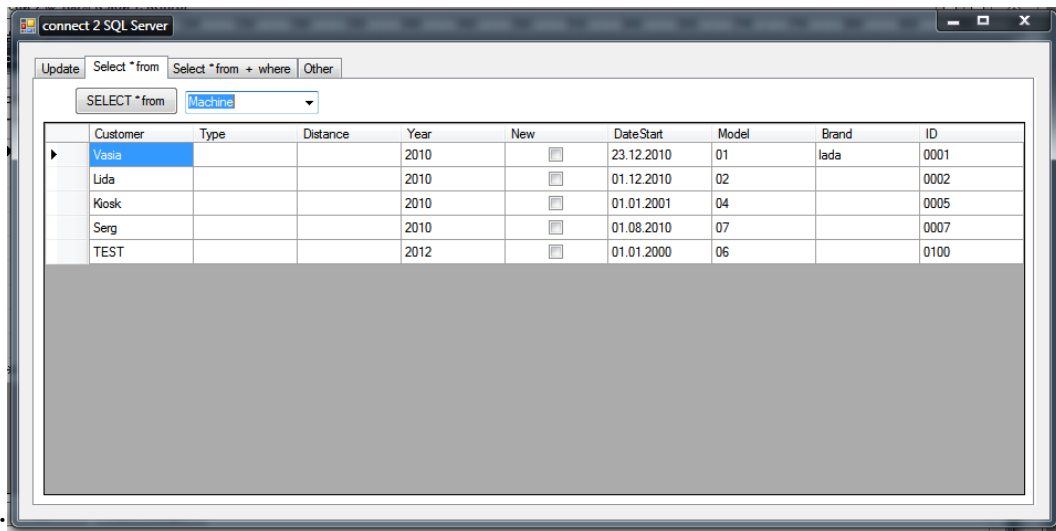


Рис. 4.11. Вкладка выборки данных

На третьей вкладке основной формы можно осуществить выборку из таблицы с параметром, заданным в компонентах `textBox1` и `textBox2` (рисунок 4.12).

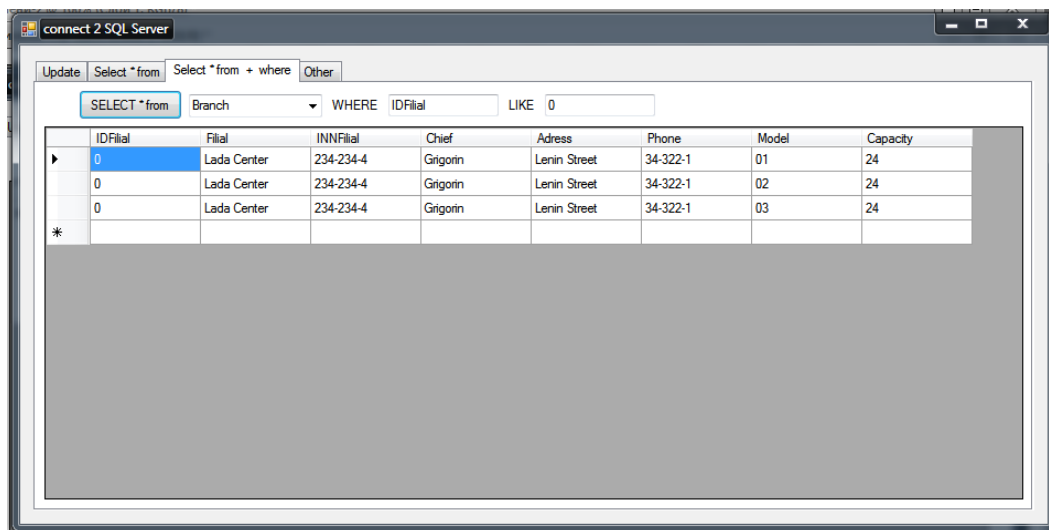


Рис. 4.12. Выборка данных с параметром

И последняя вкладка демонстрирует процесс добавления строки в таблицу `Branch`, а также позволяет осуществить удаление строки из таблицы `Branch` с указанными параметрами (рисунок 4.13). При ошибке добавления строки выводится соответствующее сообщение.

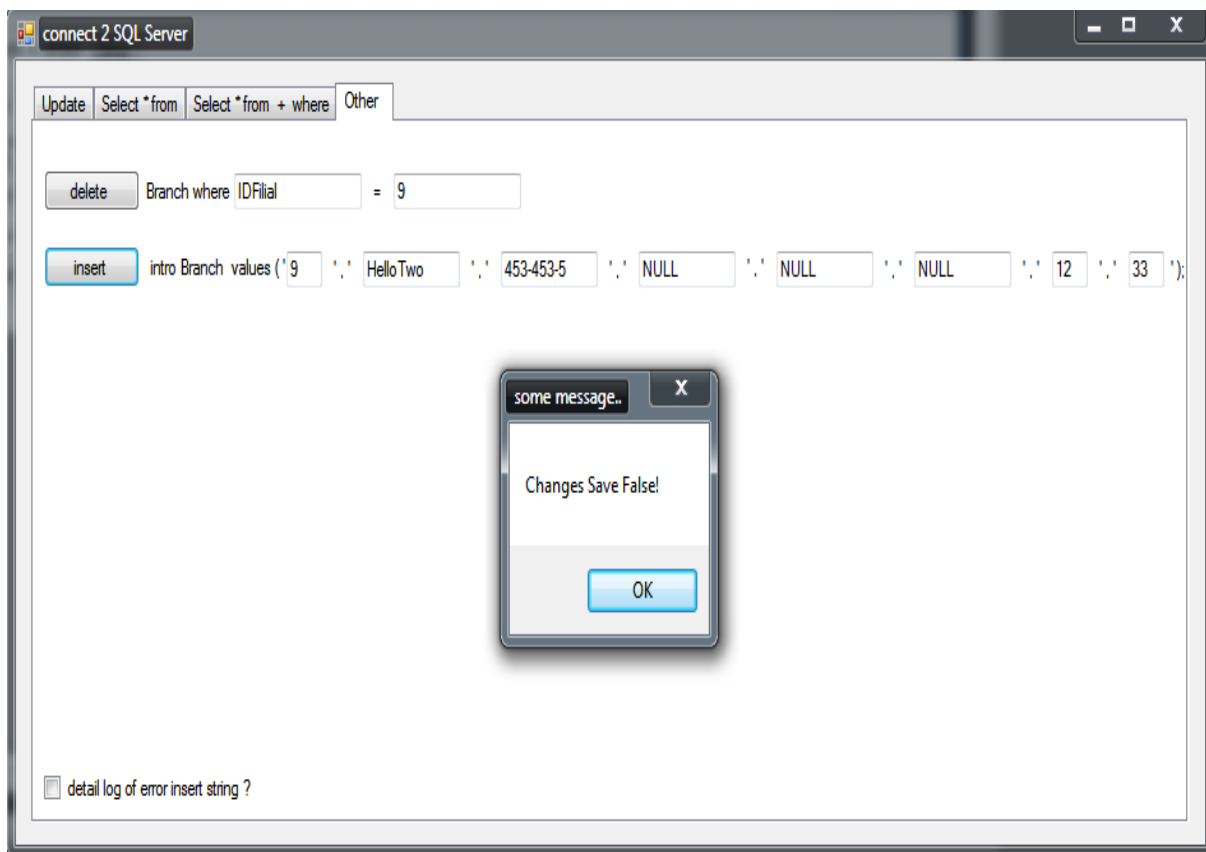


Рис. 4.13. Вкладка редактирования таблицы Branch

В рассмотренных примерах в разделах 4.1 и 4.2 реализованы простые приложения, интерфейс которых достаточно прост, что связано прежде всего с тем, что целью их рассмотрения были демонстрация подключения к базе данных и выполнение основных операций с данными. В курсовом проектировании необходимо продумать интерфейс каждого приложения и подойти к его дизайну с творческой точки зрения. Интерфейс должен быть функциональным, удобным и позволяющим осуществлять весь спектр операций по модификации, выборке данных и формированию необходимой отчетной документации.

5 Варианты задания на курсовой проект

Проанализировать предметную область, спроектировать и реализовать базу данных и приложение к ней согласно предложенным вариантам:

1. Информационная система «Автовокзал».
2. Информационная система «Надзорная деятельность подразделения МЧС».

3. Информационная система предприятия оптово–розничной торговли магазина бытовой химии.
4. Информационная система по нормированию и оплате труда персонала ремонтно–локомотивного депо.
5. Информационная система учета товарооборота кирпичного завода.
6. Информационная система учета складского товара.
7. Информационная система «Библиотека».
8. Информационная система оздоровительного комплекса.
9. Информационная система гостиничного бизнеса.
10. Информационная система туристического агентства.
11. Информационная система медицинского обслуживания населения «Поликлиника».
12. Информационная система «Школа XXI века».
13. Информационная система «Почтамт».
14. Информационная система «АРМ бухгалтера. Формирование расчётных листов сотрудников».
15. Информационная система «Экспедитор».
16. Информационная система «Рекламное агентство».
17. Информационная система «Фотоцентр».
18. Информационная система «Водоканал».
19. Информационная система «Автосервис».
20. Информационная система «ТСЖ».
21. Информационная система «Абитуриент».
22. Информационная система «Расчет заработной платы»
23. Информационная система «Профком».
24. Информационная система «Отдел кадров».
25. Информационная система «Страховая компания».
26. Информационная система «Электронный авиабилет».
27. Информационная система «Бронирование номерного фонда».
28. Информационная система «Магазин компьютерной техники».
29. Информационная система «Учет академической успеваемости в вузе».
30. Информационная система «Расписание занятий в вузе».
31. Информационная система «Аренда помещений».
32. Информационная система «Прокат бытовой техники».
33. Информационная система «Налоговая инспекция».
34. Информационная система «Склад».
35. Информационная система «Биржа труда».
36. Информационная система «Автобусные маршруты».
37. Информационная система «Железно–дорожные перевозки».
38. Информационная система «Записная книжка».
39. Информационная система «Автосервис».
40. Информационная система «Строительная компания».

41. Информационная система «Справочник абитуриента»
42. Информационная система «Вступительные экзамены».
43. Информационная система «Телепрограмма».
44. Информационная система «Риэлтерская контора».
45. Информационная система «Платежное поручение».

Список рекомендуемой литературы

1. Белоногов Г.Г., Новоселов А.П. Автоматизация процессов накопления, поиска и обобщения информации / - М.: Наука, 2017. - 256 с.
2. Ульман Дж. Основы систем баз данных / - М.: Финансы и статистика, 2017. - 292 с.
3. Пинягина О.В., Фукин И.А. Практикум по курсу «Базы данных»: учеб. пособие / Казан. (Приволж.) федер. ун-т. - Казань: Казанский университет, 2012. - 91 с.
4. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учеб. пособие / С.А. Мартишин [и др.] - М.: ИД ФОРУМ: НИЦ Инфра-М, 2012. - 160 с.
5. Пирогов В.Ю. Информационные системы и базы данных: организация и проектирование: учеб. пособие / - СПб.: БХВ-Петербург, 2009. - 528 с.
6. Дунаев В.В. Базы данных. Язык SQL для студента. - 2-е изд., доп. и перераб.- СПб.: БХВ-Петербург, 2007.
7. Агальцов В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: учебник / - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 272 с.
8. Агальцов В.П. Базы данных. В 2-х т. Т. 1. Локальные базы данных: учебник / - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 352 с.
9. Голицына О.Л., Максимов Н.В., Попов И.И. Базы данных: учебное пособие / - М.: Форум, 2012. - 400 с.
10. Карпова И.П. Базы данных: учеб. пособие / - СПб.: Питер, 2013. - 240 с.
11. Кузин А.В., Левонисова С.В. Базы данных: Учебное пособие для студ. высш. учеб. заведений / - М.: ИЦ Академия, 2012. - 320 с.
12. Советов Б.Я., Цехановский В.В., Чертовской В.Д. Базы данных: теория и практика: учебник для бакалавров / - М.: Юрайт, 2013. - 463 с.

Приложение А
Форма титульного листа курсового проекта

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО КубГТУ)

Институт _____

Кафедра _____

Направление подготовки _____

Профиль _____

КУРСОВОЙ ПРОЕКТ

по дисциплине _____
(наименование дисциплины)

на тему _____
(тема курсового проекта)

Выполнил (а) студент (ка) _____ курса _____ группы _____
(ф.и.о.)

Допущен к защите _____

Руководитель (нормоконтролер) работы _____

Защищен _____ Оценка _____
(дата)

Члены комиссии _____

(подпись, дата, расшифровка подписи)

Краснодар
20__

Приложение Б

Форма задания на курсовой проект

ФГБОУ ВПО «Кубанский государственный технологический университет»
(ФГБОУ ВО КубГТУ)

Институт _____

Кафедра _____

Направление подготовки _____

Профиль _____

УТВЕРЖДАЮ

Зав. кафедрой _____ М.В. Янаева.

«___» _____ 20__ г.

ЗАДАНИЕ

на курсовой проект

Студенту: _____ группы _____ курса
(Ф.И.О.) (№ группы и курса)

Тема проекта: _____

(утверждена указанием директора института № ___ от ___ 20__)

План проекта:

1. _____

2. _____

3. _____

Объем работы:

а) пояснительная записка _____ с.

б) иллюстративная часть _____ лист (-ов)

Рекомендуемая литература

1. _____

2. _____

3. _____

Срок выполнения: с «___» _____ по «___» _____ 20__ г.

Срок защиты: «___» _____ 20__ г.

Дата выдачи задания: «___» _____ 20__ г.

Дата сдачи проекта на кафедру: «___» _____ 20__ г.

Руководитель работы _____ Фамилия И.О.
(должность, подпись,)

Задание принял студент _____ Фамилия И.О.
(подпись)

Приложение В

Пример оформления реферата

Реферат

Курсовой проект: ... страниц, ... рисунка, ... таблиц, ... используемых источников, ... приложения, ... иллюстративная часть.

АВТОМАТИЗАЦИЯ УПРАВЛЕНИЯ ВЗАИМОДЕЙСТВИЯМИ С КЛИЕНТАМИ, CRM-СИСТЕМА, ERP-СИСТЕМА, MRP-СИСТЕМА, БАЗЫ ДАННЫХ, УПРАВЛЕНИЕ ПРОЕКТАМИ, ДОКУМЕНТООБОРОТ, ТОВАРООБОРОТ, СКЛАД, ФИЛИАЛ, МНОГОПОЛЬЗОВАТЕЛЬСКИЕ СИСТЕМЫ, УПРАВЛЕНИЕ АССОРТИМЕНТОМ, ДИАГРАММА КЛАССОВ, ДИАГРАММА ВЗАИМОДЕЙСТВИЯ ДАННЫХ, АВТОМАТИЗИРОВАННАЯ СИСТЕМА, АВТОМАТИЗАЦИЯ МАЛЫХ ПРЕДПРИЯТИЙ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ.

Объектом автоматизации является деятельность салона сотовой связи (до 20 автоматизированных рабочих мест) для организации работы с клиентами сферы розничной торговли и бизнеса.

Целью работы является увеличение эффективности работы с клиентами за счёт обеспечения на техническом уровне следующих условий: стандартизация процессов работы с клиентами; наличие единого хранилища информации; синхронизированность управления множественными каналами взаимодействия; моделирование систем управления запасами, доставкой, оперативным и бухгалтерским учетом; единая система безопасности; обеспечения инструментов анализа динамики продаж; интегрирование различных видов деятельности предприятия; обеспечение взаимодействия между организациями.

Основные полученные результаты:

- проведён анализ ключевых процессов работы с клиентами, позволяющих автоматизацию в краткосрочной перспективе.
- спроектирован и разработан пакет прикладных программ для автоматизации предприятия;

Об эффективности внедрения можно судить по изменению таких показателей, как: время обработки обращения клиента, время заказа и приемки товара на головной склад и/или удаленный филиал, время на подготовку отчётов о динамике продаж и динамике клиентской базы, а также анализ продаж всего представленного ассортимента.

ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

Методические указания

Внутри кафедральное издание

Составитель

Янаева Марина Викторовна