Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное

образовательное учреждение высшего образования

«Комсомольский-на-Амуре государственный университет»

Факультет компьютерных технологий

Кафедра «ПУРИС»

ЛАБОРАТОРНАЯ РАБОТА №3

по дисциплине «Администрирование баз данных»

Студент группы 8ВТб-1 М.А. Иванов

Преподаватель А.Н Петрова

2021

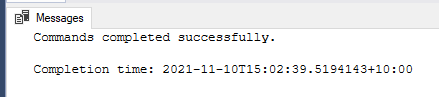
# Шифрование данных

Пример 12.1. Создание сертификата посредством инструкции CREATE CERTIFICATE.

Код запроса

|  |
| --- |
| USE sample;  CREATE MASTER KEY  ENCRYPTION BY PASSWORD = 'pls4w9d16!'  GO  CREATE CERTIFICATE cert01  WITH SUBJECT = 'Certificate for dbo'; |

Результат запроса

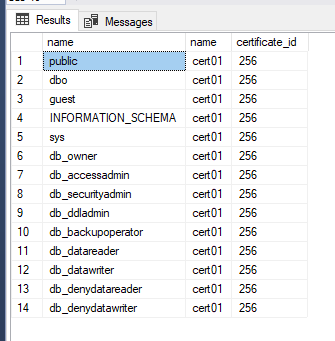


Пример 12.2. Получение информации о существующих сертификатах.

Код запроса

|  |
| --- |
| SELECT p.name, c.name, certificate\_id  FROM sys.database\_principals p, sys.certificates c  WHERE p.principal\_id = p.principal\_id |

Результат запроса



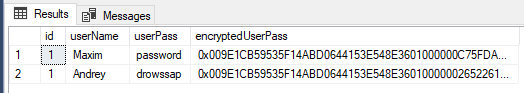
Задание 1.2 создать шифрование на уровне столбца одной из таблиц, продемонстрировать результат.

Шифрование столбца симметричным ключом:

Код запроса

|  |
| --- |
| create table users(  id int not null,  userName varchar(20),  userPass varchar(20),  encryptedUserPass varbinary(128))  go  insert into users (id, userName, userPass) values(1, 'Maxim', 'password')  insert into users (id, userName, userPass) values(1, 'Andrey', 'drowssap')  create symmetric key myKey  with algorithm = aes\_256  encryption by certificate cert01  go  open symmetric key myKey  decryption by certificate cert01  update users set encryptedUserPass = ENCRYPTBYKEY(key\_guid('myKey'), userPass)  close symmetric key myKey  go  select \* from users |

Результат запроса

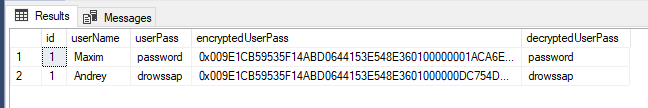


Дешифрование столбца симметричным ключом:

Код запроса

|  |
| --- |
| alter table users  add decryptedUserPass varchar(20)  go  open symmetric key myKey  decryption by certificate cert01  update users set decryptedUserPass = CONVERT(varchar, decryptByKey(encryptedUserPass))  close symmetric key myKey  go  select \* from users |

Результат запроса

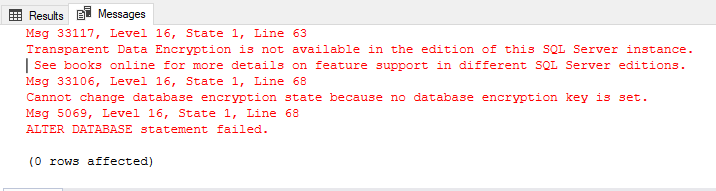


Задание 1.3 создать прозрачное шифрование данных.

Код запроса

|  |
| --- |
| use master  go  create master key encryption by password = 'mySecretKey'  go  create certificate myCertificate with subject = 'myCertificate'  go  use sample;  go  create database encryption key  with algorithm = aes\_128  encryption by server certificate myCertificate  go  alter database sample  set encryption on;  go  select \* from sys.dm\_database\_encryption\_keys  where encryption\_state = 3; |

Результат запроса



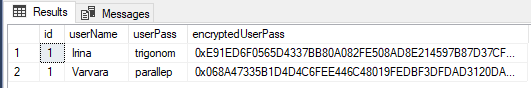
1.4 продемонстрировать создание и использование асимметричного ключа.

шифрование столбца асимметричным ключом:

Код запроса

|  |
| --- |
| create table asymUsers(  id int not null,  userName varchar(20),  userPass varchar(20),  encryptedUserPass varbinary(max))  go  insert into asymUsers (id, userName, userPass) values(1, 'Irina', 'trigonom')  insert into asymUsers (id, userName, userPass) values(1, 'Varvara', 'parallep')  create asymmetric key myAsymKey  with algorithm = RSA\_2048  encryption by password = 'asymPass'  go  update asymUsers set encryptedUserPass = encryptByAsymKey(asymKey\_ID('myAsymKey'), userPass)  select \* from asymUsers |

Результат запроса

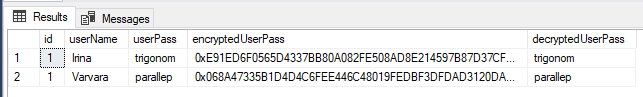


Дешифрование столбца асимметричным ключом:

Код запроса

|  |
| --- |
| alter table asymUsers  add decryptedUserPass varchar(20)  go  update asymUsers set decryptedUserPass =  CONVERT(varchar(max), decryptByAsymKey(asymKey\_ID('myAsymKey'), encryptedUserPass, N'asymPass'))  go  select \* from asymUsers |

Результат запроса



# Создание нового регистрационного имени, создание нового пользователя, управление разрешениями:

**3.1 Создать регистрационное имя и пользователя для следующих работников:**

- системный администратор;

- проектировщик БД;

- программист;

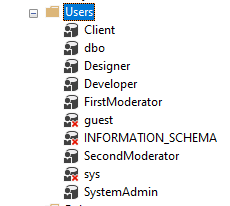
- модератор 1-го уровня. Он может изменять значения в основных справочниках, например, сотрудники, услуги и др., потому что соответствующий пользователь может увольнять и принимать на работу сотрудников, может назначать цены на услуги фирмы и менять список услуг.

- модератор 2-го уровня. Это оператор, работающий с клиентами. Он ведет списки клиентов, создает и ведет договора с клиентами и другие текущие документы.

- клиент, который имеет только право просмотра списка услуг и возможно данных по своему договору.

Код запроса

|  |
| --- |
| -- Создание логина  CREATE LOGIN SystemAdmin  WITH PASSWORD = '123';  GO  -- Создание пользователя бд по логину  CREATE USER SystemAdmin FOR LOGIN SystemAdmin;  GO  -- Создание логина  CREATE LOGIN Designer  WITH PASSWORD = '123';  GO  -- Создание пользователя бд по логину  CREATE USER Designer FOR LOGIN Designer;  GO  -- Создание логина  CREATE LOGIN Developer  WITH PASSWORD = '123';  GO  -- Создание пользователя бд по логину  CREATE USER Developer FOR LOGIN Developer;  GO  -- Создание логина  CREATE LOGIN FirstModerator  WITH PASSWORD = '123';  GO  -- Создание пользователя бд по логину  CREATE USER FirstModerator FOR LOGIN FirstModerator;  GO  -- Создание логина  CREATE LOGIN SecondModerator  WITH PASSWORD = '123';  GO  -- Создание пользователя бд по логину  CREATE USER SecondModerator FOR LOGIN SecondModerator;  GO  -- Создание логина  CREATE LOGIN Client  WITH PASSWORD = '123';  GO  -- Создание пользователя бд по логину  CREATE USER Client FOR LOGIN Client;  GO |



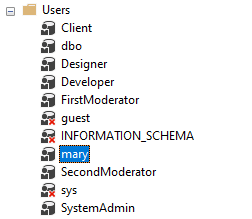
Результат запроса

3.2 Выполнить примеры

Пример 12.3. Создание нового регистрационного имени

Код запроса

|  |
| --- |
| USE Hotel;  CREATE LOGIN mary WITH PASSWORD = '123';  select \* from sys.server\_principals |



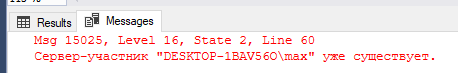


Результат запроса

Пример 12.4. Создание регистрационного имени, сопоставленного с учетной записью Windows

Код запроса

|  |
| --- |
| USE Hotel;  CREATE LOGIN [DESKTOP-1BAV56O\max] FROM WINDOWS;  select\*from sys.server\_principals |





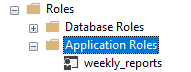
Результат запроса

3.3 Выполнить примеры

Пример 12.8. Создание роли приложения

Код запроса

|  |
| --- |
| USE Hotel;  CREATE APPLICATION ROLE weekly\_reports  WITH PASSWORD ='123',  DEFAULT\_SCHEMA = my\_schema; |

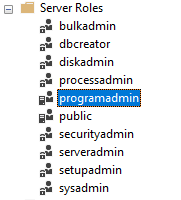


Результат запроса

Пример 12.9. Создание определяемой пользователем роли сервера и добавление в нее нового члена

Код запроса

|  |
| --- |
| USE Hotel;  GO  CREATE SERVER ROLE programadmin;  ALTER SERVER ROLE programadmin ADD MEMBER mary; |

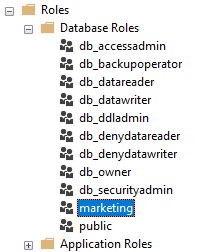


Результат запроса

Пример 12.10. Создание определяемой пользователем роли базы данных

Код запроса

|  |
| --- |
| USE Hotel;  CREATE ROLE marketing AUTHORIZATION peter;  GO  ALTER ROLE marketing ADD MEMBER peter;  ALTER ROLE marketing ADD MEMBER mary; |



Результат запроса

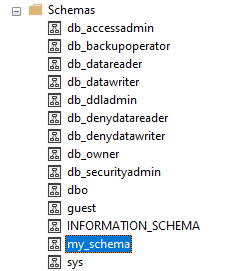
4 Создание схемы

4.1 Выполнить примеры 12.5 - 12.7.

Пример 12.5. Создание схемы и ее применение

Код запроса

|  |
| --- |
| USE Hotel;  GO  CREATE SCHEMA my\_schema AUTHORIZATION peter  GO  CREATE TABLE product  (product\_no CHAR(10) NOT NULL UNIQUE,  product\_name CHAR(20) NULL,  price MONEY NULL);  GO  CREATE VIEW product\_info  AS SELECT product\_no, product\_name  FROM product;  GO  GRANT SELECT TO mary;  DENY UPDATE TO mary; |

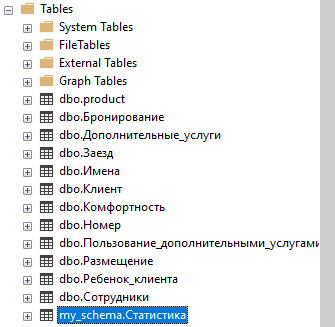


Результат запроса

Пример 12.6. Применение инструкции ALTER SCHEMA

Код запроса

|  |
| --- |
| USE Hotel;  ALTER SCHEMA my\_schema TRANSFER dbo.Статистика; |

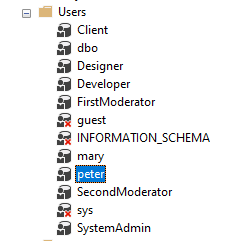


Результат запроса

Пример 12.7. Создание пользователя

Код запроса

|  |
| --- |
| USE Hotel;  CREATE USER peter FOR LOGIN [DESKTOP-1BAV56O\max];  CREATE USER mary FOR LOGIN mary WITH DEFAULT\_SCHEMA =  my\_schema; |

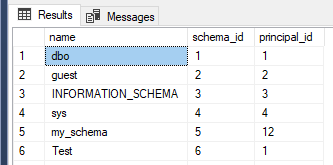


Результат запроса

4.2 Создать схему в своей БД и выделить на нее права доступа некоторому пользователю.

Код запроса

|  |
| --- |
| CREATE SCHEMA Test;  GO  -- Выдаем все права на схему Test пользователю Designer  GRANT CONTROL ON SCHEMA::Test TO Designer;  GO  SELECT \* FROM sys.schemas; |

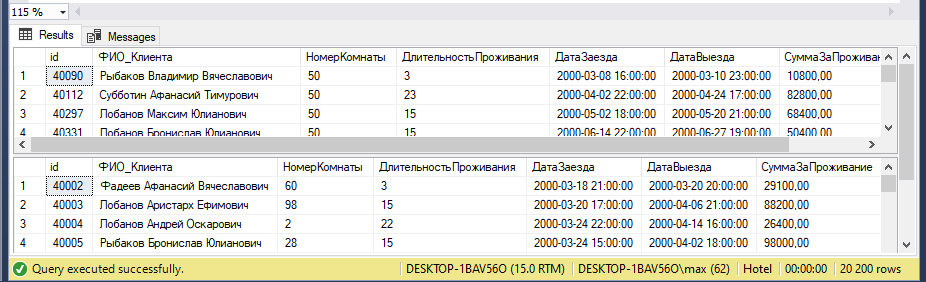


Результат запроса

4.3 Создать в схеме представление и функцию.

Код запроса

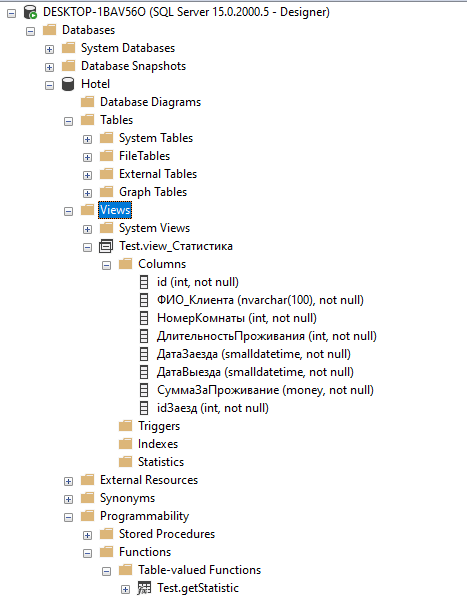
|  |
| --- |
| -- создаем представление на основе таблицы users  CREATE VIEW Test.view\_Статистика AS SELECT \* FROM Статистика  GO  -- создаем функцию  CREATE FUNCTION Test.getStatistic  (  @roomNum int -- Номер комнаты  )  RETURNS TABLE  AS  RETURN  (  -- Оператор SELECT, результат которого возвращает функция  SELECT \* from Test.view\_Статистика where НомерКомнаты=@roomNum  )  GO  -- Вызов функции  select \* from Test.getStatistic(50)  -- Вызов представления  select \* from Test.view\_Статистика |



Результат запроса

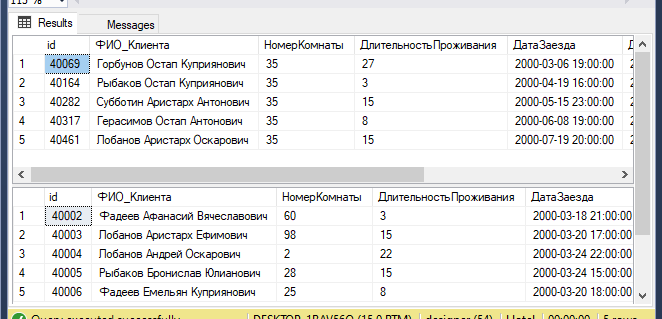
4.4 Продемонстрировать возможности пользователей с наличием прав доступа и без них.

В результате входа под учетной записью designer видим, что ей недоступно какое-либо взаимодействие с базой данных Hotel помимо содержимого схемы Test. Так как схеме Test принадлежат только представление view\_Статистика и функция getStatistic, designer может взаимодействовать только с ними.



Код запроса

|  |
| --- |
| select top 5 \* from Test.getStatistic(35)  select top 5 \* from Test.view\_Статистика |



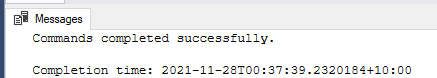
Результат запроса

5 Авторизация, выполнить задания 12.11 - 12.20

Пример 12.11. Использование разрешения CREATE

Код запроса

|  |
| --- |
| USE Hotel;  GRANT CREATE TABLE, CREATE PROCEDURE  TO peter, mary; |

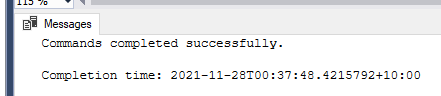


Результат запроса

Пример 12.12. Предоставление разрешения для создания определяемых пользователем функций

Код запроса

|  |
| --- |
| USE Hotel;  GRANT CREATE FUNCTION  TO mary; |

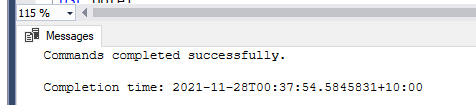


Результат запроса

Пример 12.13. Применение разрешения SELECT в инструкции GRANT

Код запроса

|  |
| --- |
| USE Hotel;  GRANT SELECT ON Статистика  TO peter, mary; |

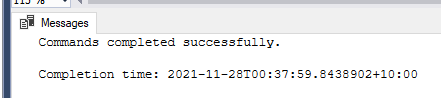


Результат запроса

Пример 12.14. Применение разрешения UPDATE в инструкции GRANT

Код запроса

|  |
| --- |
| USE Hotel;  GRANT UPDATE ON Статистика (ДлительностьПроживания, СуммаЗаПроживание)  TO peter; |

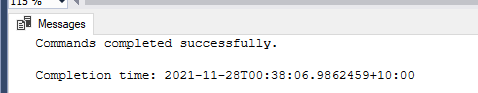


Результат запроса

Пример 12.15. Предоставление доступа для чтения метаданных

Код запроса

|  |
| --- |
| USE Hotel;  GRANT VIEW DEFINITION ON OBJECT::Статистика TO peter;  GRANT VIEW DEFINITION ON SCHEMA::dbo TO peter; |

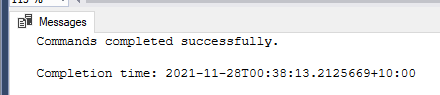


Результат запроса

Пример 12.16. Применение разрешения CONTROL

Код запроса

|  |
| --- |
| USE Hotel;  GRANT CONTROL ON DATABASE::Hotel TO peter; |

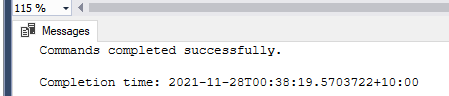


Результат запроса

Пример 12.17. Инструкция GRANT с предложением WITH GRANT OPTION

Код запроса

|  |
| --- |
| USE Hotel;  GRANT SELECT ON Статистика TO mary  WITH GRANT OPTION; |

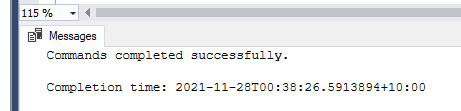


Результат запроса

Пример 12.18. Запрещение пользователю peter двух предоставленных ранее разрешений

Код запроса

|  |
| --- |
| USE Hotel;  DENY CREATE TABLE, CREATE PROCEDURE  TO peter; |

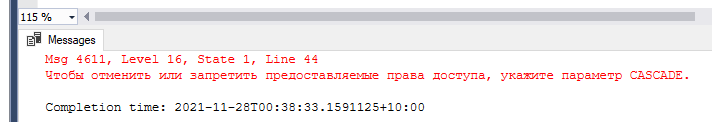


Результат запроса

Пример 12.19. Запрещение разрешений отдельным пользователям

Код запроса

|  |
| --- |
| USE Hotel;  GRANT SELECT ON Статистика  TO PUBLIC;  DENY SELECT ON Статистика  TO peter, mary; |

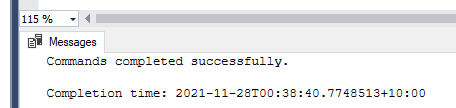


Результат запроса

Пример 12.20. Отмена разрешения на инструкцию SELECT

Код запроса

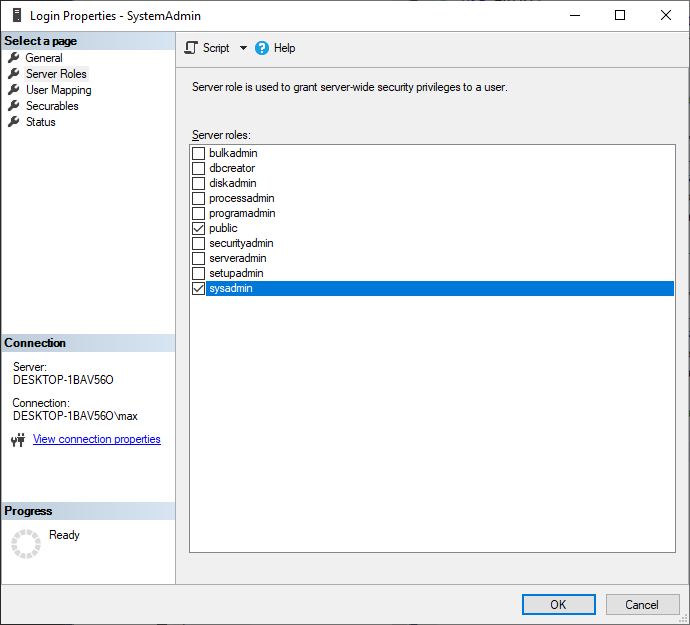
|  |
| --- |
| USE Hotel;  REVOKE SELECT ON Статистика FROM public; |



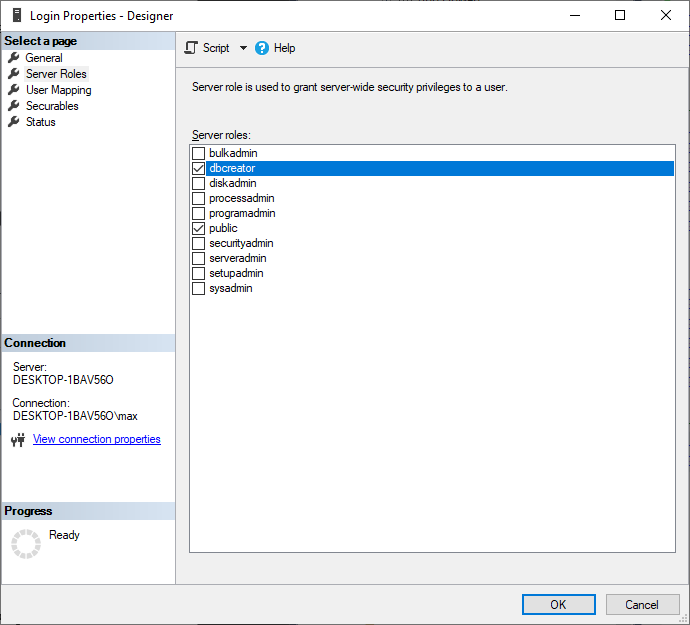
Результат запроса

5.1 Выделить права созданным пользователям в соответствии с описанными выше требованиями.

Выделение прав системному администратору. Серверная роль sysadmin.

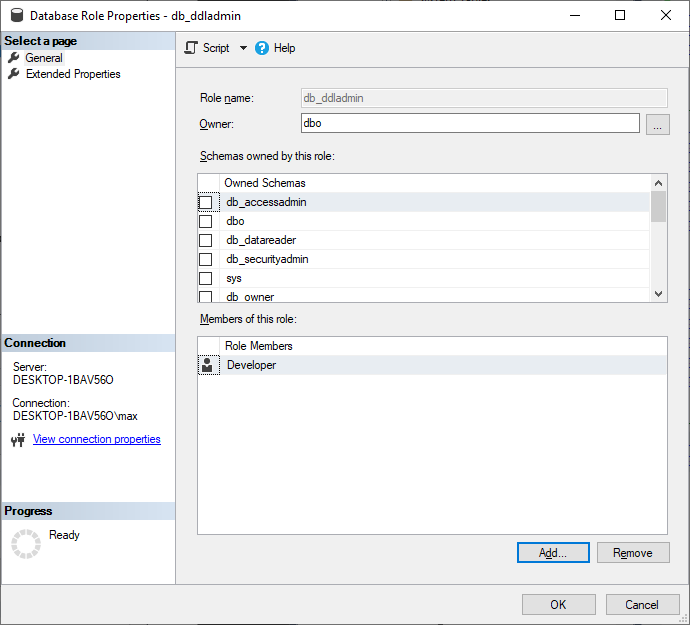


Выделение прав дизайнеру базы данных. Серверная роль dbcreator.



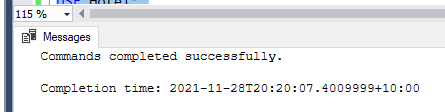
Выделение прав программисту. Роль db\_ddladmin.

Даём доступ на выполнение любых команд языка определения данных (DDL).



Код запроса для выделения прав первому модератору

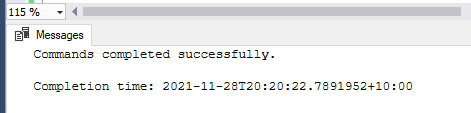
|  |
| --- |
| USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Сотрудники  TO FirstModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Дополнительные\_услуги  TO FirstModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Номер  TO FirstModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Размещение  TO FirstModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Комфортность  TO FirstModerator;  USE Hotel;  GRANT SELECT ON Бронирование  TO FirstModerator;  USE Hotel;  GRANT SELECT ON Заезд  TO FirstModerator;  USE Hotel;  GRANT SELECT ON Клиент  TO FirstModerator;  USE Hotel;  GRANT SELECT ON Пользование\_дополнительными\_услугами  TO FirstModerator;  USE Hotel;  GRANT SELECT ON Ребенок\_клиента  TO FirstModerator; |



Результат запроса

Код запроса для выделения прав второму модератору

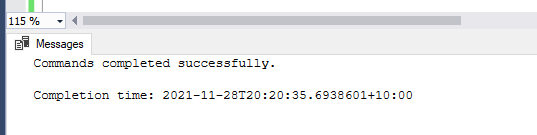
|  |
| --- |
| USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Бронирование  TO SecondModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Заезд  TO SecondModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Клиент  TO SecondModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Пользование\_дополнительными\_услугами  TO SecondModerator;  USE Hotel;  GRANT UPDATE, SELECT, INSERT, DELETE ON Ребенок\_клиента  TO SecondModerator;  USE Hotel;  GRANT SELECT ON Дополнительные\_услуги  TO SecondModerator;  USE Hotel;  GRANT SELECT ON Номер  TO SecondModerator;  USE Hotel;  GRANT SELECT ON Размещение  TO SecondModerator;  USE Hotel;  GRANT SELECT ON Комфортность  TO SecondModerator; |



Результат запроса

Код запроса для выделения прав клиенту

|  |
| --- |
| USE Hotel;  GRANT SELECT ON dbo.Номер TO Client;  GRANT SELECT ON dbo.Дополнительные\_услуги TO Client; |

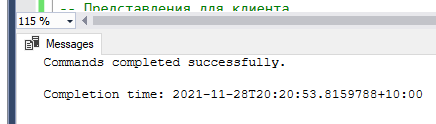


Результат запроса

5.2 Создать представление для ограничения доступа к данным и выделить права на него для соответствующих пользователей

Код запроса выделения прав на представления для первого модератора

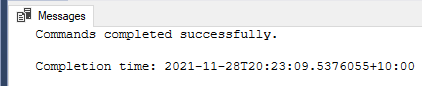
|  |
| --- |
| USE Hotel;  GRANT SELECT ON dbo.View\_Arrival TO FirstModerator;  GRANT SELECT ON dbo.View\_ClientChild TO FirstModerator;  GRANT SELECT ON dbo.View\_Reservation TO FirstModerator;  GRANT SELECT ON dbo.View\_Room TO FirstModerator;  GRANT SELECT ON dbo.View\_UsageService TO FirstModerator; |



Результат запроса

Код запроса выделения прав на представления для второго модератора

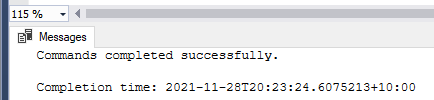
|  |
| --- |
| USE Hotel;  GRANT SELECT ON dbo.View\_Arrival TO SecondModerator;  GRANT SELECT ON dbo.View\_ClientChild TO SecondModerator;  GRANT SELECT ON dbo.View\_Reservation TO SecondModerator;  GRANT SELECT ON dbo.View\_Room TO SecondModerator;  GRANT SELECT ON dbo.View\_UsageService TO SecondModerator; |



Результат запроса

Код запроса выделения прав на представления для клиента

|  |
| --- |
| USE Hotel;  GRANT SELECT ON dbo.Room TO Client; |



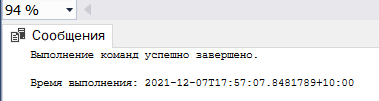
Результат запроса

6 Создание и управление ролью приложения:

6.1 Создать роль приложения для приложения, разработанного ранее на С#.

Код запроса

|  |
| --- |
| CREATE APPLICATION ROLE app  WITH PASSWORD = '123'  , DEFAULT\_SCHEMA = dbo;  GO  GRANT CONTROL ON DATABASE::Hotel TO app;  GO |



Результат запроса

Модифицировать приложение так, чтобы после авторизации пользователь обладал функционалом, которых соответствует его должности (руководитель фирмы, оператор).

Далее представлен модифицированный код приложения. Добавленные строки выделены желтым цветом.

Код файла Program.cs

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Threading.Tasks;  using System.Windows.Forms;  using System.Data.SqlClient;  namespace Hotel\_Database  {  static class Program  {  /// <summary>  /// Главная точка входа для приложения.  /// </summary>  [STAThread]  static void Main()  {  Application.EnableVisualStyles();  Application.SetCompatibleTextRenderingDefault(false);  Application.Run(new connection());  }  public static void openConnection(SqlConnection connection)  {  // Получаем данные из параметров  string strConn = Properties.Settings.Default.ConnStr.ToString();  bool isAppRole = Properties.Settings.Default.IsRoleApp;  string roleApp = Properties.Settings.Default.RoleApp.ToString();  // Создаем подключение  connection.ConnectionString = strConn;  connection.Open();    SqlCommand cmd = new SqlCommand(roleApp, connection);  // Если была выбрана авторизация по пользователю, то необходимо активировать роль приложения  if (isAppRole)  {  try  {  // Запускаем роль приложения  cmd.ExecuteNonQuery();  }  catch {  // если не удалось запустить роль приложения  emergencyRestart(connection, strConn);  }  }    cmd.Cancel();  }  static void emergencyRestart(SqlConnection connection, string strConn) {  connection.Close();  connection.ConnectionString = strConn;  connection.Open();  }  public static void openStartForm()  {  switch (Properties.Settings.Default.StartForm)  {  case 0:  {  Menu moderator1 = new Menu();  moderator1.ShowFirstDialog();  }  break;  case 1:  {  Menu moderator2 = new Menu();  moderator2.ShowSecondDialog();  }  break;  default:  {  MessageBox.Show(  "Невозможно перейти на предыдущю форму",  "Ошибка",  MessageBoxButtons.OK,  MessageBoxIcon.Information);  }  break;  }  }  }  } |

Код файла Settings.Designer.cs

|  |
| --- |
| //------------------------------------------------------------------------------  // <auto-generated>  // This code was generated by a tool.  // Runtime Version:4.0.30319.42000  //  // Changes to this file may cause incorrect behavior and will be lost if  // the code is regenerated.  // </auto-generated>  //------------------------------------------------------------------------------  namespace Hotel\_Database.Properties  {  [global::System.Runtime.CompilerServices.CompilerGeneratedAttribute()]  [global::System.CodeDom.Compiler.GeneratedCodeAttribute("Microsoft.VisualStudio.Editors.SettingsDesigner.SettingsSingleFileGenerator", "11.0.0.0")]  internal sealed partial class Settings : global::System.Configuration.ApplicationSettingsBase  {  private static Settings defaultInstance = ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new Settings())));  internal string ConnStr;  // Запрос на включение роли приложения  internal string RoleApp = "EXECUTE sys.sp\_setapprole 'app', '123';";  // Выбрана ли роль приложения  internal bool IsRoleApp = false;  // Стартовая форма при нажатии "Назад".  // 0 - полноценная, 1 - модератор второго уровня  internal int StartForm = 0;  public static Settings Default  {  get  {  return defaultInstance;  }  }  }  } |

Код файла advanced\_arrive\_adding

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class Form16 : Form  {  SqlConnection myConn = new SqlConnection();  public Form16()  {  InitializeComponent();  }  private void Form16\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  workerComboBox();  roomComboBox();  clientComboBox();  reservationComboBox();  }  private void workerComboBox()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_сотрудника from Сотрудники", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  comboBox1.DisplayMember = "ФИО\_сотрудника";  comboBox1.DataSource = dtbl;  }  private void roomComboBox()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select Номер\_комнаты from Номер", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  comboBox2.DisplayMember = "Номер\_комнаты";  comboBox2.DataSource = dtbl;  }  private void clientComboBox()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_клиента from Клиент", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  comboBox3.DisplayMember = "ФИО\_клиента";  comboBox3.DataSource = dtbl;  }  private void reservationComboBox()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select id\_Бронирование from Бронирование", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  // Добавляем пустое значение в ComboBox  dtbl.Rows.Add();  }  private void add\_Click(object sender, EventArgs e)  {  try  {  // Данные заезда  string nameStaff = comboBox1.Text;  string idRoom = comboBox2.Text.Split()[0];  string nameClient = comboBox3.Text;  string pricePerStay = textBox5.Text;  string priceForAddServe = textBox6.Text;  string fine = textBox7.Text;  string scheduledDateLeave = dateTimePicker1.Text + " " + dateTimePicker2.Text;  string dateArrival = dateTimePicker3.Text + " " + dateTimePicker4.Text;  string peopleNum = textBox11.Text;  // Данные клиента  string passportClient = textBox12.Text;  string dateClient = dateTimePicker7.Text;  string numberClient = textBox14.Text;  string countChildren = textBox15.Text;  // Остальные поля могут быть нулевыми  if (nameStaff.Trim() == "" || idRoom.Trim() == "" ||  nameClient.Trim() == "" || scheduledDateLeave.Trim() == "" ||  dateArrival.Trim() == "" || peopleNum.Trim() == "") throw new Exception();  string procedure = "execute add\_arrival @workerName = @p1, @clientName = @p2, @room = @p3, ";  // if (idReservation != "") procedure = procedure + "@idReservation = @p4, ";  if (pricePerStay != "") procedure = procedure + "@pricePerStay = @p5, ";  if (priceForAddServe != "") procedure = procedure + "@priceForAddServe = @p6, ";  if (fine != "") procedure = procedure + "@fine = @p7, ";  procedure = procedure + "@dateArrival = @p8, @scheduledDateLeave = @p9, ";  //if (dateLeave != "") procedure = procedure + "@dateLeave = @p10, ";  procedure = procedure + "@peopleNum = @p11, @clientPassport = @p12, " +  "@clientBirthday = @p13, @clientPhone = @p14, @clientChildCount = @p15";  // Создать команду для добавления  SqlCommand myComm = new SqlCommand(procedure, myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameStaff;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameClient;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = idRoom;  if (pricePerStay != "") {  myComm.Parameters.Add("@p5", SqlDbType.Money, 100);  myComm.Parameters["@p5"].Value = pricePerStay;  }  if (priceForAddServe != "") {  myComm.Parameters.Add("@p6", SqlDbType.Money, 100);  myComm.Parameters["@p6"].Value = priceForAddServe;  }  if (fine != "") {  myComm.Parameters.Add("@p7", SqlDbType.Money, 100);  myComm.Parameters["@p7"].Value = fine;  }  myComm.Parameters.Add("@p8", SqlDbType.SmallDateTime);  myComm.Parameters["@p8"].Value = dateArrival;  myComm.Parameters.Add("@p9", SqlDbType.SmallDateTime);  myComm.Parameters["@p9"].Value = scheduledDateLeave;  myComm.Parameters.Add("@p11", SqlDbType.NVarChar, 100);  myComm.Parameters["@p11"].Value = peopleNum;  myComm.Parameters.Add("@p12", SqlDbType.NVarChar, 100);  myComm.Parameters["@p12"].Value = passportClient;  myComm.Parameters.Add("@p13", SqlDbType.Date);  myComm.Parameters["@p13"].Value = dateClient;  myComm.Parameters.Add("@p14", SqlDbType.NVarChar, 100);  myComm.Parameters["@p14"].Value = numberClient;  myComm.Parameters.Add("@p15", SqlDbType.NVarChar, 100);  myComm.Parameters["@p15"].Value = countChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();  myConn.Close();  }  catch  {  myConn.Close();  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  }  } |

Код файла advanced\_reservation\_adding

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class advanced\_reservation\_adding : Form  {  SqlConnection myConn = new SqlConnection();  public advanced\_reservation\_adding()  {  InitializeComponent();  }  private void Form17\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  workerComboBox();  roomComboBox();  clientComboBox();  }  private void workerComboBox()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_сотрудника from Сотрудники", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  comboBox2.DisplayMember = "ФИО\_сотрудника";  comboBox2.DataSource = dtbl;  }  private void roomComboBox()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select Номер\_комнаты from Номер", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  comboBox1.DisplayMember = "Номер\_комнаты";  comboBox1.DataSource = dtbl;  }  private void clientComboBox()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_клиента from Клиент", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  comboBox3.DisplayMember = "ФИО\_клиента";  comboBox3.DataSource = dtbl;  }  private void add\_Click(object sender, EventArgs e)  {  try  {  // Данные бронирования  string roomNumber = comboBox1.Text.Split()[0];  string workerName = comboBox2.Text;  string clientName = comboBox3.Text;  string dateReservation = dateTimePicker1.Text + " " + dateTimePicker2.Text;  string scheduledDateArrival = dateTimePicker3.Text + " " + dateTimePicker4.Text;  string scheduledDateLeave = dateTimePicker5.Text + " " + dateTimePicker6.Text;  string peopleNum = textBox7.Text;  string reservationType = comboBox4.Text;  string reservaionPrice = textBox9.Text;  // Данные клиента  string passportClient = textBox12.Text;  string dateClient = dateTimePicker7.Text;  string numberClient = textBox14.Text;  string countChildren = textBox15.Text;  // надо фиксить - если запятой не будет, то исключение  // удаляем лишние знаки  // Остальные поля могут быть нулевыми  if (roomNumber.Trim() == "" || workerName.Trim() == "" ||  clientName.Trim() == "" || dateReservation.Trim() == "" ||  scheduledDateArrival.Trim() == "" || scheduledDateLeave.Trim() == "") throw new Exception();  // Если три последних поля оставить пустыми, то они установятся по умолчанию  string procedure = "execute add\_reservation @workerName = @p1, @clientName = @p2, @room = @p3, " +  "@dateReservation = @p4, @scheduledDateArrival = @p5, @scheduledDateLeave = @p6";  if (peopleNum != "") procedure = procedure + ", @peopleNum = @p7";  if (reservationType != "") procedure = procedure + ", @reservationType = @p8";  if (reservaionPrice != "") procedure = procedure + ", @reservaionPrice = @p9";  procedure = procedure + ", @clientPassport = @p10, @clientBirthday = @p11," +  " @clientPhone = @p12, @clientChildCount = @p13";  // Создать команду для добавления  SqlCommand myComm = new SqlCommand(procedure, myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = workerName;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = clientName;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = roomNumber;  myComm.Parameters.Add("@p4", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p4"].Value = dateReservation;  myComm.Parameters.Add("@p5", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p5"].Value = scheduledDateArrival;  myComm.Parameters.Add("@p6", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p6"].Value = scheduledDateLeave;  if (peopleNum != "") {  myComm.Parameters.Add("@p7", SqlDbType.NVarChar, 100);  myComm.Parameters["@p7"].Value = peopleNum;  }  if (reservationType != "") {  myComm.Parameters.Add("@p8", SqlDbType.NVarChar, 100);  myComm.Parameters["@p8"].Value = reservationType;  }  if (reservaionPrice != "") {  myComm.Parameters.Add("@p9", SqlDbType.Money, 100);  myComm.Parameters["@p9"].Value = reservaionPrice;  }  myComm.Parameters.Add("@p10", SqlDbType.NVarChar, 100);  myComm.Parameters["@p10"].Value = passportClient;  myComm.Parameters.Add("@p11", SqlDbType.Date);  myComm.Parameters["@p11"].Value = dateClient;  myComm.Parameters.Add("@p12", SqlDbType.NVarChar, 100);  myComm.Parameters["@p12"].Value = numberClient;  myComm.Parameters.Add("@p13", SqlDbType.NVarChar, 100);  myComm.Parameters["@p13"].Value = countChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  private void comboBox4\_SelectedIndexChanged(object sender, EventArgs e)  {  if (comboBox4.Text == "с предоплатой") textBox9.ReadOnly = false;  if (comboBox4.Text == "без предоплаты")  {  textBox9.ReadOnly = true;  textBox9.Text = "0,0000";  }  }  }  } |

Код файла arrive\_information

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class arrive\_information : Form  {  SqlConnection myConn = new SqlConnection();  string idАrrival, nameStaff, idRoom, nameClient, idReservation,  pricePerStay, priceForAddServe, fine, scheduledDateLeave,  dateArrival, dateLeave, peopleNum;  public arrive\_information(string idАrrival, string nameStaff, string idRoom, string nameClient, string idReservation,  string pricePerStay, string priceForAddServe, string fine, string scheduledDateLeave,  string dateArrival, string dateLeave, string peopleNum)  {  this.idАrrival = idАrrival;  this.nameStaff = nameStaff;  this.idRoom = idRoom;  this.nameClient = nameClient;  this.idReservation = idReservation;  this.pricePerStay = pricePerStay;  this.priceForAddServe = priceForAddServe;  this.fine = fine;  this.scheduledDateLeave = scheduledDateLeave;  this.dateArrival = dateArrival;  this.dateLeave = dateLeave;  this.peopleNum = peopleNum;  InitializeComponent();  }  private void Form12\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Запускаем процедуру выборки данных  loadDataStaff();  loadDataRoom();  loadDataClient();  loadDataReservation();  // Устанавливаем значения полям  textBox1.Text = idАrrival;  textBox2.Text = nameStaff;  textBox3.Text = idRoom;  textBox4.Text = nameClient;  textBox5.Text = idReservation;  textBox6.Text = pricePerStay;  textBox7.Text = priceForAddServe;  textBox8.Text = fine;  textBox9.Text = scheduledDateLeave;  textBox10.Text = dateArrival;  textBox11.Text = dateLeave;  textBox12.Text = peopleNum;  }  private void loadDataStaff()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.Сотрудники where ФИО\_сотрудника = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameStaff;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView2.DataSource = dtbl;    }  private void loadDataReservation()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_Reservation where id\_Бронирование = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = idReservation;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView4.DataSource = dtbl;    }  private void loadDataClient()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.Клиент where ФИО\_клиента = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameClient;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView3.DataSource = dtbl;    }  private void loadDataRoom()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_Room where Номер\_комнаты = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = idRoom;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    }  }  } |

Код файла arrives\_list

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class arrives\_list : Form  {  SqlConnection myConn = new SqlConnection();  int indexSelectRow;  string idАrrival, nameStaff, idRoom, nameClient, idReservation,  pricePerStay, priceForAddServe, fine, scheduledDateLeave,  dateArrival, dateLeave, peopleNum;  public arrives\_list()  {  InitializeComponent();  }  private void Form11\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  nameStaffComboBox();  idRoomComboBox();  nameClientComboBox();  idReservationComboBox();  // Запускаем процедуру выборки данных  loadData();  }  private void addBig\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Form16 form16 = new Form16();  form16.ShowDialog();  // Перезапускаем форму для обновления данных  this.Hide();  arrives\_list form11 = new arrives\_list();  form11.ShowDialog();  this.Close();  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  idАrrival = dataGridView1[0, indexSelectRow].Value.ToString();  nameStaff = dataGridView1[1, indexSelectRow].Value.ToString();  idRoom = dataGridView1[2, indexSelectRow].Value.ToString();  nameClient = dataGridView1[3, indexSelectRow].Value.ToString();  idReservation = dataGridView1[4, indexSelectRow].Value.ToString();  pricePerStay = dataGridView1[5, indexSelectRow].Value.ToString();  priceForAddServe = dataGridView1[6, indexSelectRow].Value.ToString();  fine = dataGridView1[7, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[8, indexSelectRow].Value.ToString();  dateArrival = dataGridView1[9, indexSelectRow].Value.ToString();  dateLeave = dataGridView1[10, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[11, indexSelectRow].Value.ToString();  }  }  internal void OnlyRead()  {  addBig\_ToolStripMenuItem.Visible = false;  add\_ToolStripMenuItem.Visible = false;  delete\_ToolStripMenuItem.Visible = false;  update\_ToolStripMenuItem.Visible = false;  }  private void add\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  idАrrival = dataGridView1[0, indexSelectRow].Value.ToString();  nameStaff = dataGridView1[1, indexSelectRow].Value.ToString();  idRoom = dataGridView1[2, indexSelectRow].Value.ToString();  nameClient = dataGridView1[3, indexSelectRow].Value.ToString();  idReservation = dataGridView1[4, indexSelectRow].Value.ToString();  pricePerStay = dataGridView1[5, indexSelectRow].Value.ToString();  priceForAddServe = dataGridView1[6, indexSelectRow].Value.ToString();  fine = dataGridView1[7, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[8, indexSelectRow].Value.ToString();  dateArrival = dataGridView1[9, indexSelectRow].Value.ToString();  dateLeave = dataGridView1[10, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[11, indexSelectRow].Value.ToString();    string procedure = "execute add\_arrival ";  if (nameStaff != "") procedure = procedure + "@workerName = @p1, ";  if (nameClient != "") procedure = procedure + "@clientName = @p2, ";  if (idRoom != "") procedure = procedure + "@room = @p3, ";  if (idReservation != "") procedure = procedure + "@idReservation = @p4, ";  if (pricePerStay != "") procedure = procedure + "@pricePerStay = @p5, ";  if (priceForAddServe != "") procedure = procedure + "@priceForAddServe = @p6, ";  if (fine != "") procedure = procedure + "@fine = @p7, ";  procedure = procedure + "@dateArrival = @p8, ";  if (scheduledDateLeave != "") procedure = procedure + "@scheduledDateLeave = @p9, ";  if (dateLeave != "") procedure = procedure + "@dateLeave = @p10, ";  procedure = procedure + "@peopleNum = @p11";  // Создать команду для добавления  SqlCommand myComm = new SqlCommand(procedure, myConn);  // Создать параметр и передать в него значение текстового поля  if (nameStaff != "")  {  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameStaff;  }  if (nameClient != "")  {  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameClient;  }  if (idRoom != "")  {  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = idRoom;  }  if (idReservation != "")  {  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 100);  myComm.Parameters["@p4"].Value = idReservation;  }  if (pricePerStay != "")  {  myComm.Parameters.Add("@p5", SqlDbType.Money);  myComm.Parameters["@p5"].Value = pricePerStay;  }  if (priceForAddServe != "")  {  myComm.Parameters.Add("@p6", SqlDbType.Money);  myComm.Parameters["@p6"].Value = priceForAddServe;  }  if (fine != "")  {  myComm.Parameters.Add("@p7", SqlDbType.Money);  myComm.Parameters["@p7"].Value = fine;  }  myComm.Parameters.Add("@p8", SqlDbType.SmallDateTime);  myComm.Parameters["@p8"].Value = dateArrival;  if (scheduledDateLeave != "")  {  myComm.Parameters.Add("@p9", SqlDbType.SmallDateTime);  myComm.Parameters["@p9"].Value = scheduledDateLeave;  }  if (dateLeave != "")  {  myComm.Parameters.Add("@p10", SqlDbType.SmallDateTime);  myComm.Parameters["@p10"].Value = dateLeave;  }  if (peopleNum != "")  {  myComm.Parameters.Add("@p11", SqlDbType.NVarChar, 100);  myComm.Parameters["@p11"].Value = peopleNum;  }  else {  myComm.Parameters.Add("@p11", SqlDbType.NVarChar, 100);  myComm.Parameters["@p11"].Value = peopleNum;  }  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  private void delete\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {    if (idАrrival == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_arrival @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = idАrrival;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();    idRoomComboBox();  // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  idАrrival = dataGridView1[0, indexSelectRow].Value.ToString();  nameStaff = dataGridView1[1, indexSelectRow].Value.ToString();  idRoom = dataGridView1[2, indexSelectRow].Value.ToString();  nameClient = dataGridView1[3, indexSelectRow].Value.ToString();  idReservation = dataGridView1[4, indexSelectRow].Value.ToString();  pricePerStay = dataGridView1[5, indexSelectRow].Value.ToString();  priceForAddServe = dataGridView1[6, indexSelectRow].Value.ToString();  fine = dataGridView1[7, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[8, indexSelectRow].Value.ToString();  dateArrival = dataGridView1[9, indexSelectRow].Value.ToString();  dateLeave = dataGridView1[10, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[11, indexSelectRow].Value.ToString();    // Остальные поля могут быть нулевыми  if (nameStaff.Trim() == "" || idRoom.Trim() == "" ||  nameClient.Trim() == "" || scheduledDateLeave.Trim() == "" ||  dateArrival.Trim() == "" || peopleNum.Trim() == "") throw new Exception();  string procedure = "execute update\_arrival @workerName = @p1, @clientName = @p2, @room = @p3, ";  if (idReservation != "") procedure = procedure + "@idReservation = @p4, ";  if (pricePerStay != "") procedure = procedure + "@pricePerStay = @p5, ";  if (priceForAddServe != "") procedure = procedure + "@priceForAddServe = @p6, ";  if (fine != "") procedure = procedure + "@fine = @p7, ";  procedure = procedure + "@dateArrival = @p8, @scheduledDateLeave = @p9, ";  if (dateLeave != "") procedure = procedure + "@dateLeave = @p10, ";  procedure = procedure + "@peopleNum = @p11, @idArrival = @p12";  // Создать команду для добавления  SqlCommand myComm = new SqlCommand(procedure, myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameStaff;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameClient;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = idRoom;  if (idReservation != "")  {  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 100);  myComm.Parameters["@p4"].Value = idReservation;  }  if (pricePerStay != "")  {  myComm.Parameters.Add("@p5", SqlDbType.Money);  myComm.Parameters["@p5"].Value = pricePerStay;  }  if (priceForAddServe != "")  {  myComm.Parameters.Add("@p6", SqlDbType.Money);  myComm.Parameters["@p6"].Value = priceForAddServe;  }  if (fine != "")  {  myComm.Parameters.Add("@p7", SqlDbType.Money);  myComm.Parameters["@p7"].Value = fine;  }  myComm.Parameters.Add("@p8", SqlDbType.SmallDateTime);  myComm.Parameters["@p8"].Value = dateArrival;  myComm.Parameters.Add("@p9", SqlDbType.SmallDateTime);  myComm.Parameters["@p9"].Value = scheduledDateLeave;  if (dateLeave != "")  {  myComm.Parameters.Add("@p10", SqlDbType.SmallDateTime);  myComm.Parameters["@p10"].Value = dateLeave;  }  myComm.Parameters.Add("@p11", SqlDbType.NVarChar, 100);  myComm.Parameters["@p11"].Value = peopleNum;  myComm.Parameters.Add("@p12", SqlDbType.NVarChar, 100);  myComm.Parameters["@p12"].Value = idАrrival;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    idRoomComboBox();  // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void info\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  if (idАrrival.Trim() == "") throw new Exception();  arrive\_information form12 = new arrive\_information(idАrrival, nameStaff, idRoom, nameClient, idReservation,  pricePerStay, priceForAddServe, fine, scheduledDateLeave,  dateArrival, dateLeave, peopleNum);  form12.ShowDialog();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  "1. Возможно вы пытаетесь посмотреть информацию о несуществующих данных.", "Внимание!");  }  }  private void back\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Hide();  Program.openStartForm();  Close();  }  private void nameStaffComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_сотрудника from Сотрудники", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  name\_staff.DisplayMember = "ФИО\_сотрудника";  name\_staff.DataSource = dtbl;    }  private void idRoomComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select Номер\_комнаты from Номер", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  id\_room.DisplayMember = "Номер\_комнаты";  id\_room.DataSource = dtbl;    }  private void nameClientComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_клиента from Клиент", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  name\_client.DisplayMember = "ФИО\_клиента";  name\_client.DataSource = dtbl;    }  private void idReservationComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select id\_Бронирование from Бронирование", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  // Добавляем пустое значение в ComboBox  dtbl.Rows.Add();  id\_reservation.DisplayMember = "id\_Бронирование";  id\_reservation.DataSource = dtbl;    }  private void loadData()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_Arrival", myConn);  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  dtbl.Columns[3].AllowDBNull = true;  dtbl.Columns[4].AllowDBNull = true;  dtbl.Columns[5].AllowDBNull = true;  dtbl.Columns[6].AllowDBNull = true;  dtbl.Columns[7].AllowDBNull = true;  dtbl.Columns[8].AllowDBNull = true;  dtbl.Columns[9].AllowDBNull = true;  dtbl.Columns[10].AllowDBNull = true;  dtbl.Columns[11].AllowDBNull = true;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  // Если в таблице есть минимум одна запись  if (dataGridView1.Rows.Count > 1)  {  idАrrival = dataGridView1[0, indexSelectRow].Value.ToString();  nameStaff = dataGridView1[1, indexSelectRow].Value.ToString();  idRoom = dataGridView1[2, indexSelectRow].Value.ToString();  nameClient = dataGridView1[3, indexSelectRow].Value.ToString();  idReservation = dataGridView1[4, indexSelectRow].Value.ToString();  pricePerStay = dataGridView1[5, indexSelectRow].Value.ToString();  priceForAddServe = dataGridView1[6, indexSelectRow].Value.ToString();  fine = dataGridView1[7, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[8, indexSelectRow].Value.ToString();  dateArrival = dataGridView1[9, indexSelectRow].Value.ToString();  dateLeave = dataGridView1[10, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[11, indexSelectRow].Value.ToString();  }  }  }  } |

Код файла child\_information

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class child\_information : Form  {  SqlConnection myConn = new SqlConnection();  string idChildren, nameParent, nameChildren, dateChildren;  public child\_information(string idChildren, string nameParent, string nameChildren, string dateChildren)  {  this.idChildren = idChildren;  this.nameParent = nameParent;  this.nameChildren = nameChildren;  this.dateChildren = dateChildren;  InitializeComponent();  }  private void Form5\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Запускаем процедуру выборки данных  loadData1();  // Устанавливаем значения полям  textBox1.Text = idChildren;  textBox2.Text = nameParent;  textBox3.Text = nameChildren;  textBox4.Text = dateChildren;  }  private void loadData1()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.Клиент where ФИО\_клиента = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameParent;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    }  }  } |

Код файла child\_list

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class child\_list : Form  {  SqlConnection myConn = new SqlConnection();  int indexSelectRow;  string idChildren, nameParent, nameChildren, dateChildren;  public child\_list()  {  InitializeComponent();  }  private void Form4\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  loadDataComboBox();  // Запускаем процедуру выборки данных  loadData();  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameParent = dataGridView1[1, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[2, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[3, indexSelectRow].Value.ToString();  }  }  private void add\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameParent = dataGridView1[1, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[2, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[3, indexSelectRow].Value.ToString();    if (nameParent.Trim() == "" || nameChildren.Trim() == "" ||  dateChildren.Trim() == "") throw new Exception();  // Создать команду для добавления  SqlCommand myComm = new SqlCommand("execute add\_child @p1, @p2, @p3", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameParent;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameChildren;  myComm.Parameters.Add("@p3", SqlDbType.Date);  myComm.Parameters["@p3"].Value = dateChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  internal void OnlyRead()  {  add\_ToolStripMenuItem.Visible = false;  delete\_ToolStripMenuItem.Visible = false;  update\_ToolStripMenuItem.Visible = false;  }  private void delete\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {    if (idChildren == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_child @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 10);  myComm.Parameters["@p1"].Value = idChildren;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();    // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {    // Присваиваем переменным обновленные значения  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameParent = dataGridView1[1, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[2, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[3, indexSelectRow].Value.ToString();  if (nameParent.Trim() == "" || nameChildren.Trim() == "" ||  dateChildren.Trim() == "") throw new Exception();  // Создать команду для изменения  SqlCommand myComm = new SqlCommand("execute update\_child @p1, @p2, @p3, @p4", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.Int);  myComm.Parameters["@p1"].Value = idChildren;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameParent;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = nameChildren;  myComm.Parameters.Add("@p4", SqlDbType.Date);  myComm.Parameters["@p4"].Value = dateChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void info\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  if (idChildren.Trim() == "") throw new Exception();  child\_information form5 = new child\_information(idChildren, nameParent, nameChildren, dateChildren);  form5.ShowDialog();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  "1. Возможно вы пытаетесь посмотреть информацию о несуществующих данных.", "Внимание!");  }  }  private void back\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Hide();  Program.openStartForm();  Close();  }  private void loadDataComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_клиента from Клиент", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  name\_parent.DisplayMember = "ФИО\_клиента";  name\_parent.DataSource = dtbl;    }  private void loadData()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_ClientChild", myConn);  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  dtbl.Columns[3].AllowDBNull = true;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  // Если в таблице есть минимум одна запись  if (dataGridView1.Rows.Count > 1)  {  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameParent = dataGridView1[1, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[2, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[3, indexSelectRow].Value.ToString();  }  }  }  } |

Код файла client\_information

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class client\_information : Form  {  SqlConnection myConn = new SqlConnection();  int indexSelectRow;  string idClient, passportClient, nameClient, dateClient, numberClient, countChildren;  string idChildren, nameChildren, dateChildren;  public client\_information(string idClient, string passportClient, string nameClient, string dateClient, string numberClient, string countChildren)  {  this.idClient = idClient;  this.passportClient = passportClient;  this.nameClient = nameClient;  this.dateClient = dateClient;  this.numberClient = numberClient;  this.countChildren = countChildren;  InitializeComponent();  }  private void Form7\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Запускаем процедуру выборки данных  loadData();  // Устанавливаем значения полям  textBox1.Text = idClient;  textBox2.Text = passportClient;  textBox3.Text = nameClient;  textBox4.Text = dateClient;  textBox5.Text = numberClient;  textBox6.Text = countChildren;  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[1, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[2, indexSelectRow].Value.ToString();  }  }  private void add\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[1, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[2, indexSelectRow].Value.ToString();    if (nameClient.Trim() == "" || nameChildren.Trim() == "" ||  dateChildren.Trim() == "") throw new Exception();  // Создать команду для добавления  SqlCommand myComm = new SqlCommand("execute add\_child @p1, @p2, @p3", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameClient;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameChildren;  myComm.Parameters.Add("@p3", SqlDbType.Date);  myComm.Parameters["@p3"].Value = dateChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  private void delete\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {    if (idChildren == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_child @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 10);  myComm.Parameters["@p1"].Value = idChildren;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();    // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_Click(object sender, EventArgs e)  {  try  {      // Проверка на последнюю строчку  if (indexSelectRow >= dataGridView1.Rows.Count - 1) throw new Exception();  // Присваиваем переменным обновленные значения  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[1, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[2, indexSelectRow].Value.ToString();  if (nameClient.Trim() == "" || nameChildren.Trim() == "" ||  dateChildren.Trim() == "") throw new Exception();  // Создать команду для изменения  SqlCommand myComm = new SqlCommand("execute update\_child @p1, @p2, @p3, @p4", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.Int);  myComm.Parameters["@p1"].Value = idChildren;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameClient;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = nameChildren;  myComm.Parameters.Add("@p4", SqlDbType.Date);  myComm.Parameters["@p4"].Value = dateChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void loadData()  {    SqlCommand myComm = new SqlCommand("select id\_Ребенок\_клиента, ФИО\_ребенка, Дата\_рождения from dbo.Ребенок\_клиента where Клиент\_id\_Клиент = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = idClient;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  // Отключаем подстановку номеров  dtbl.Columns[0].AutoIncrement = false;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  // Если в таблице есть запись о ребенке клиента  if (dataGridView1.Rows.Count > 1)  {  idChildren = dataGridView1[0, indexSelectRow].Value.ToString();  nameChildren = dataGridView1[1, indexSelectRow].Value.ToString();  dateChildren = dataGridView1[2, indexSelectRow].Value.ToString();  }  }  }  } |

Код файл client\_list

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class client\_list : Form  {  SqlConnection myConn = new SqlConnection();  int indexSelectRow;  string idClient, passportClient, nameClient, dateClient, numberClient, countChildren;  public client\_list()  {  InitializeComponent();  }  private void Form6\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Запускаем процедуру выборки данных  loadData();  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  idClient = dataGridView1[0, indexSelectRow].Value.ToString();  passportClient = dataGridView1[1, indexSelectRow].Value.ToString();  nameClient = dataGridView1[2, indexSelectRow].Value.ToString();  dateClient = dataGridView1[3, indexSelectRow].Value.ToString();  numberClient = dataGridView1[4, indexSelectRow].Value.ToString();  countChildren = dataGridView1[5, indexSelectRow].Value.ToString();  }  }  private void add\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  idClient = dataGridView1[0, indexSelectRow].Value.ToString();  passportClient = dataGridView1[1, indexSelectRow].Value.ToString();  nameClient = dataGridView1[2, indexSelectRow].Value.ToString();  dateClient = dataGridView1[3, indexSelectRow].Value.ToString();  numberClient = dataGridView1[4, indexSelectRow].Value.ToString();  countChildren = dataGridView1[5, indexSelectRow].Value.ToString();    if (passportClient.Trim() == "" || nameClient.Trim() == "" ||  dateClient.Trim() == "" || numberClient.Trim() == "" ||  countChildren.Trim() == "") throw new Exception();  // Создать команду для добавления  SqlCommand myComm = new SqlCommand("execute add\_client @p1, @p2, @p3, @p4, @p5", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = passportClient;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = nameClient;  myComm.Parameters.Add("@p3", SqlDbType.Date);  myComm.Parameters["@p3"].Value = dateClient;  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 100);  myComm.Parameters["@p4"].Value = numberClient;  myComm.Parameters.Add("@p5", SqlDbType.NVarChar, 100);  myComm.Parameters["@p5"].Value = countChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  internal void OnlyRead()  {  add\_ToolStripMenuItem.Visible = false;  delete\_ToolStripMenuItem.Visible = false;  update\_ToolStripMenuItem.Visible = false;  }  private void delete\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {    if (idClient == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_client @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 10);  myComm.Parameters["@p1"].Value = idClient;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();    // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {    // Присваиваем переменным обновленные значения  idClient = dataGridView1[0, indexSelectRow].Value.ToString();  passportClient = dataGridView1[1, indexSelectRow].Value.ToString();  nameClient = dataGridView1[2, indexSelectRow].Value.ToString();  dateClient = dataGridView1[3, indexSelectRow].Value.ToString();  numberClient = dataGridView1[4, indexSelectRow].Value.ToString();  countChildren = dataGridView1[5, indexSelectRow].Value.ToString();  if (idClient.Trim() == "") throw new Exception();  // Создать команду для изменения  SqlCommand myComm = new SqlCommand("execute update\_client @p1, @p2, @p3, @p4, @p5, @p6", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.Int);  myComm.Parameters["@p1"].Value = idClient;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = passportClient;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = nameClient;  myComm.Parameters.Add("@p4", SqlDbType.Date);  myComm.Parameters["@p4"].Value = dateClient;  myComm.Parameters.Add("@p5", SqlDbType.NVarChar, 100);  myComm.Parameters["@p5"].Value = numberClient;  myComm.Parameters.Add("@p6", SqlDbType.NVarChar, 100);  myComm.Parameters["@p6"].Value = countChildren;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void info\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  if (idClient.Trim() == "") throw new Exception();  client\_information form7 = new client\_information(idClient, passportClient, nameClient, dateClient, numberClient, countChildren);  form7.ShowDialog();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  "1. Возможно вы пытаетесь посмотреть информацию о несуществующих данных.", "Внимание!");  }  }  private void back\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Hide();  Program.openStartForm();  Close();  }  private void loadData()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.Клиент", myConn);  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  dtbl.Columns[3].AllowDBNull = true;  dtbl.Columns[4].AllowDBNull = true;  dtbl.Columns[5].AllowDBNull = true;  // Отключаем подстановку номеров  dtbl.Columns[0].AutoIncrement = false;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  idClient = dataGridView1[0, indexSelectRow].Value.ToString();  passportClient = dataGridView1[1, indexSelectRow].Value.ToString();  nameClient = dataGridView1[2, indexSelectRow].Value.ToString();  dateClient = dataGridView1[3, indexSelectRow].Value.ToString();  numberClient = dataGridView1[4, indexSelectRow].Value.ToString();  countChildren = dataGridView1[5, indexSelectRow].Value.ToString();  }  }  } |

Код файла connection

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class connection : Form  {  public connection()  {  InitializeComponent();  }  private void Form1\_Load(object sender, EventArgs e)  {  cmbTypeAutor.Items.Add("Проверка подлинности Windows");  cmbTypeAutor.Items.Add("Проверка подлинности SQL Server");  txtUserName.Enabled = false;  txtPass.Enabled = false;  }  private void bttConnect\_Click(object sender, EventArgs e)  {  try  {  string StrConn;  if (cmbTypeAutor.SelectedIndex == 0)  {  StrConn = "Data Source =" + txtNameSrv.Text + ";Initial Catalog =" + txtNameDB.Text + ";Integrated Security = True";  }  else  {  StrConn = "Data Source =" + txtNameSrv.Text + ";Initial Catalog =" + txtNameDB.Text + ";User ID =" + txtUserName.Text + ";Password =" + txtPass.Text;  }  SqlConnection MainConn = new SqlConnection(StrConn); // Создание нового подключения на основе строки  MainConn.Open(); // Открытие подключения  // Проверка, установлено ли соединение с БД  if (MainConn.State == ConnectionState.Open)  {  // Если подключение прошло успешно, тосохраняем строку в параметры  Properties.Settings.Default.ConnStr = StrConn;  MainConn.Close();  if (txtUserName.Text == "SecondModerator")  {  Properties.Settings.Default.StartForm = 1;  }  // Переходим на следующую форму  Hide();  Program.openStartForm();  Close();  }  // Если подключение не установлено, то выводим сообщение  else  {  MessageBox.Show("Соединение с БД не установлено", "Внимание!");  }  }  // пустые поля, либо неправильные, либо на компьютере нет сервера  catch { MessageBox.Show("Невозможно подключиться к базе данных", "Внимание!"); }  }  private void cmbTypeAutor\_SelectionChangeCommitted(object sender, EventArgs e)  {  if (cmbTypeAutor.SelectedIndex == 0)  {  txtUserName.Text = "";  txtUserName.Enabled = false;  txtPass.Text = "";  txtPass.Enabled = false;  Properties.Settings.Default.IsRoleApp = false;  }  else  {  txtUserName.Enabled = true;  txtPass.Enabled = true;  Properties.Settings.Default.IsRoleApp = true;  }  }  }  } |

Код файла Menu

|  |
| --- |
| using System;  using System.Data;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class Menu : Form  {  public Menu()  {  InitializeComponent();  }  private void button1\_Click(object sender, EventArgs e)  {  int idNumberPlacement = 1;  Hide();  universal form3 = new universal(idNumberPlacement);  form3.ShowDialog();  Close();  }  private void button2\_Click(object sender, EventArgs e)  {  int idNumberComfort = 2;  Hide();  universal form3 = new universal(idNumberComfort);  form3.ShowDialog();  Close();  }  private void button3\_Click(object sender, EventArgs e)  {  int idNumberService = 3;  Hide();  universal form3 = new universal(idNumberService);  form3.ShowDialog();  Close();  }  private void button4\_Click(object sender, EventArgs e)  {  int idNumberStaff = 4;  Hide();  universal form3 = new universal(idNumberStaff);  form3.ShowDialog();  Close();  }  private void button5\_Click(object sender, EventArgs e)  {  Hide();  child\_list child\_list = new child\_list();  if (Properties.Settings.Default.StartForm == 0) child\_list.OnlyRead();  child\_list.ShowDialog();  Close();  }  private void button6\_Click(object sender, EventArgs e)  {  Hide();  client\_list client\_list = new client\_list();  if (Properties.Settings.Default.StartForm == 0) client\_list.OnlyRead();  client\_list.ShowDialog();  Close();  }  private void button7\_Click(object sender, EventArgs e)  {  Hide();  service\_using\_list service\_using\_list = new service\_using\_list();  if (Properties.Settings.Default.StartForm == 0) service\_using\_list.OnlyRead();  service\_using\_list.ShowDialog();  Close();  }  private void button8\_Click(object sender, EventArgs e)  {  Hide();  arrives\_list arrives\_list = new arrives\_list();  if (Properties.Settings.Default.StartForm == 0) arrives\_list.OnlyRead();  arrives\_list.ShowDialog();  Close();  }  private void button9\_Click(object sender, EventArgs e)  {  Hide();  room\_list form13 = new room\_list();  form13.ShowDialog();  Close();  }  private void button10\_Click(object sender, EventArgs e)  {  Hide();  reservation\_list reservation\_list = new reservation\_list();  if (Properties.Settings.Default.StartForm == 0) reservation\_list.OnlyRead();  reservation\_list.ShowDialog();  Close();  }  public void ShowFirstDialog()  {  ShowDialog();  }  public void ShowSecondDialog()  {  buttonRoom.Visible = false;  buttonService.Visible = false;  buttonWorker.Visible = false;  buttonComfort.Visible = false;  buttonPlacement.Visible = false;  Text = "Регистратор";  Height = 330;  ShowDialog();  }  }  } |

Код файла reservation\_information

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class reservation\_information : Form  {  SqlConnection myConn = new SqlConnection();  string idReservation, roomNumber, workerName, clientName,  dateReservation, scheduledDateArrival, scheduledDateLeave,  peopleNum, reservationType, reservaionPrice;  public reservation\_information(string idReservation, string roomNumber, string workerName, string clientName,  string dateReservation, string scheduledDateArrival, string scheduledDateLeave,  string peopleNum, string reservationType, string reservaionPrice)  {  this.idReservation = idReservation;  this.roomNumber = roomNumber;  this.workerName = workerName;  this.clientName = clientName;  this.dateReservation = dateReservation;  this.scheduledDateArrival = scheduledDateArrival;  this.scheduledDateLeave = scheduledDateLeave;  this.peopleNum = peopleNum;  this.reservationType = reservationType;  this.reservaionPrice = reservaionPrice;  InitializeComponent();  }  private void Form15\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Запускаем процедуру выборки данных  loadDataStaff();  loadDataRoom();  loadDataClient();  // Устанавливаем значения полям  textBox1.Text = idReservation;  textBox2.Text = roomNumber;  textBox3.Text = workerName;  textBox4.Text = clientName;  textBox5.Text = dateReservation;  textBox6.Text = scheduledDateArrival;  textBox7.Text = scheduledDateLeave;  textBox8.Text = peopleNum;  textBox9.Text = reservationType;  textBox10.Text = reservaionPrice;  }  private void loadDataClient()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.Клиент where ФИО\_клиента = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = clientName;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView3.DataSource = dtbl;    }  private void loadDataRoom()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_Room where Номер\_комнаты = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = roomNumber;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    }  private void loadDataStaff()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.Сотрудники where ФИО\_сотрудника = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = workerName;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView2.DataSource = dtbl;    }  }  } |

Код файла reservation\_list

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class reservation\_list : Form  {  SqlConnection myConn = new SqlConnection();  int indexSelectRow;  string idReservation, roomNumber, workerName, clientName,  dateReservation, scheduledDateArrival, scheduledDateLeave,  peopleNum, reservationType, reservaionPrice;  public reservation\_list()  {  InitializeComponent();  }  private void Form14\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  nameStaffComboBox();  idRoomComboBox();  nameClientComboBox();  // Запускаем процедуру выборки данных  loadData();  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  idReservation = dataGridView1[0, indexSelectRow].Value.ToString();  roomNumber = dataGridView1[1, indexSelectRow].Value.ToString();  workerName = dataGridView1[2, indexSelectRow].Value.ToString();  clientName = dataGridView1[3, indexSelectRow].Value.ToString();  dateReservation = dataGridView1[4, indexSelectRow].Value.ToString();  scheduledDateArrival = dataGridView1[5, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[6, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[7, indexSelectRow].Value.ToString();  reservationType = dataGridView1[8, indexSelectRow].Value.ToString();  reservaionPrice = dataGridView1[9, indexSelectRow].Value.ToString();  }  }  private void addBig\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  advanced\_reservation\_adding form17 = new advanced\_reservation\_adding();  form17.ShowDialog();  // Перезапускаем форму для обновления данных  this.Hide();  reservation\_list form14 = new reservation\_list();  form14.ShowDialog();  this.Close();  }  private void dataGridView1\_CellEnter(object sender, DataGridViewCellEventArgs e)  {  if (e.RowIndex != dataGridView1.Rows.Count - 1 && e.RowIndex != -1)  if (e.ColumnIndex == 9 || e.ColumnIndex == 8)  {  if (dataGridView1[8, e.RowIndex].Value.ToString() == "с предоплатой")  dataGridView1[9, e.RowIndex].ReadOnly = false;  if (dataGridView1[8, e.RowIndex].Value.ToString() == "без предоплаты")  {  dataGridView1[9, e.RowIndex].ReadOnly = true;  dataGridView1[9, e.RowIndex].Value = "0,0000";  }  }  }  private void add\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  idReservation = dataGridView1[0, indexSelectRow].Value.ToString();  roomNumber = dataGridView1[1, indexSelectRow].Value.ToString();  workerName = dataGridView1[2, indexSelectRow].Value.ToString();  clientName = dataGridView1[3, indexSelectRow].Value.ToString();  dateReservation = dataGridView1[4, indexSelectRow].Value.ToString();  scheduledDateArrival = dataGridView1[5, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[6, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[7, indexSelectRow].Value.ToString();  reservationType = dataGridView1[8, indexSelectRow].Value.ToString();  reservaionPrice = dataGridView1[9, indexSelectRow].Value.ToString();  // Остальные поля могут быть нулевыми  if (roomNumber.Trim() == "" || workerName.Trim() == "" ||  clientName.Trim() == "" || dateReservation.Trim() == "" ||  scheduledDateArrival.Trim() == "" || scheduledDateLeave.Trim() == "") throw new Exception();  // Если три последних поля оставить пустыми, то они установятся по умолчанию  string procedure = "execute add\_reservation @workerName = @p1, @clientName = @p2, @room = @p3, " +  "@dateReservation = @p4, @scheduledDateArrival = @p5, @scheduledDateLeave = @p6";  if (peopleNum != "") procedure = procedure + ", @peopleNum = @p7";  if (reservationType != "") procedure = procedure + ", @reservationType = @p8";  if (reservaionPrice != "") procedure = procedure + ", @reservaionPrice = @p9";  // Создать команду для добавления  SqlCommand myComm = new SqlCommand(procedure, myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = workerName;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = clientName;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = roomNumber;  myComm.Parameters.Add("@p4", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p4"].Value = dateReservation;  myComm.Parameters.Add("@p5", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p5"].Value = scheduledDateArrival;  myComm.Parameters.Add("@p6", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p6"].Value = scheduledDateLeave;  if (peopleNum != "") {  myComm.Parameters.Add("@p7", SqlDbType.NVarChar, 100);  myComm.Parameters["@p7"].Value = peopleNum;  }  if (reservationType != "") {  myComm.Parameters.Add("@p8", SqlDbType.NVarChar, 100);  myComm.Parameters["@p8"].Value = reservationType;  }  if (reservaionPrice != "") {  myComm.Parameters.Add("@p9", SqlDbType.Money, 100);  myComm.Parameters["@p9"].Value = reservaionPrice;  }  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();  //myConn.Close();  // Обновляем содержимое  loadData();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  private void delete\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {  //myConn.Open();  if (idReservation == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_reservation @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = idReservation;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();  // Обновляем содержимое  loadData();  }  }  catch  {  //myConn.Close();  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  idReservation = dataGridView1[0, indexSelectRow].Value.ToString();  roomNumber = dataGridView1[1, indexSelectRow].Value.ToString();  workerName = dataGridView1[2, indexSelectRow].Value.ToString();  clientName = dataGridView1[3, indexSelectRow].Value.ToString();  dateReservation = dataGridView1[4, indexSelectRow].Value.ToString();  scheduledDateArrival = dataGridView1[5, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[6, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[7, indexSelectRow].Value.ToString();  reservationType = dataGridView1[8, indexSelectRow].Value.ToString();  reservaionPrice = dataGridView1[9, indexSelectRow].Value.ToString();  // надо фиксить - если запятой не будет, то исключение  // удаляем лишние знаки  /\*  reservaionPrice = pricePerStay.Substring(0, pricePerStay.LastIndexOf(','));  \*/  //myConn.Open();  // Остальные поля могут быть нулевыми  if (roomNumber.Trim() == "" || workerName.Trim() == "" ||  clientName.Trim() == "" || dateReservation.Trim() == "" ||  scheduledDateArrival.Trim() == "" || scheduledDateLeave.Trim() == "") throw new Exception();  // Если три последних поля оставить пустыми, то они установятся по умолчанию  string procedure = "execute update\_reservation @idReservation = @p10, @workerName = @p1, @clientName = @p2, @room = @p3, " +  "@dateReservation = @p4, @scheduledDateArrival = @p5, @scheduledDateLeave = @p6";  if (peopleNum != "") procedure = procedure + ", @peopleNum = @p7";  if (reservationType != "") procedure = procedure + ", @reservationType = @p8";  if (reservaionPrice != "") procedure = procedure + ", @reservaionPrice = @p9";  // Создать команду для добавления  SqlCommand myComm = new SqlCommand(procedure, myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = workerName;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = clientName;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = roomNumber;  myComm.Parameters.Add("@p4", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p4"].Value = dateReservation;  myComm.Parameters.Add("@p5", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p5"].Value = scheduledDateArrival;  myComm.Parameters.Add("@p6", SqlDbType.SmallDateTime, 100);  myComm.Parameters["@p6"].Value = scheduledDateLeave;  if (peopleNum != "") {  myComm.Parameters.Add("@p7", SqlDbType.NVarChar, 100);  myComm.Parameters["@p7"].Value = peopleNum;  }  if (reservationType != "") {  myComm.Parameters.Add("@p8", SqlDbType.NVarChar, 100);  myComm.Parameters["@p8"].Value = reservationType;  }  if (reservaionPrice != "") {  myComm.Parameters.Add("@p9", SqlDbType.NVarChar, 100);  myComm.Parameters["@p9"].Value = reservaionPrice;  }  myComm.Parameters.Add("@p10", SqlDbType.NVarChar, 100);  myComm.Parameters["@p10"].Value = idReservation;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();  //myConn.Close();  // Обновляем содержимое  loadData();  }  catch  {  //myConn.Close();  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void info\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  if (idReservation.Trim() == "") throw new Exception();  reservation\_information form15 = new reservation\_information(idReservation, roomNumber, workerName, clientName,  dateReservation, scheduledDateArrival, scheduledDateLeave,  peopleNum, reservationType, reservaionPrice);  form15.ShowDialog();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  "1. Возможно вы пытаетесь посмотреть информацию о несуществующих данных.", "Внимание!");  }  }  private void back\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Hide();  Program.openStartForm();  Close();  }  private void nameStaffComboBox()  {  //myConn.Open();  SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_сотрудника from Сотрудники", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  worker\_name.DisplayMember = "ФИО\_сотрудника";  worker\_name.DataSource = dtbl;  //myConn.Close();  }  private void idRoomComboBox()  {  //myConn.Open();  SqlDataAdapter sqlDa = new SqlDataAdapter("select Номер\_комнаты from Номер", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  number\_room.DisplayMember = "Номер\_комнаты";  number\_room.DataSource = dtbl;  //myConn.Close();  }  private void nameClientComboBox()  {  //myConn.Open();  SqlDataAdapter sqlDa = new SqlDataAdapter("select ФИО\_клиента from Клиент", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  client\_name.DisplayMember = "ФИО\_клиента";  client\_name.DataSource = dtbl;  // myConn.Close();  }  private void loadData()  {  //myConn.Open();  SqlCommand myComm = new SqlCommand("select\*from dbo.View\_Reservation", myConn);  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;  //myConn.Close();  // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  dtbl.Columns[3].AllowDBNull = true;  dtbl.Columns[4].AllowDBNull = true;  dtbl.Columns[5].AllowDBNull = true;  dtbl.Columns[6].AllowDBNull = true;  dtbl.Columns[7].AllowDBNull = true;  dtbl.Columns[8].AllowDBNull = true;  dtbl.Columns[9].AllowDBNull = true;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  // Если в таблице есть минимум одна запись  if (dataGridView1.Rows.Count > 1)  {  idReservation = dataGridView1[0, indexSelectRow].Value.ToString();  roomNumber = dataGridView1[1, indexSelectRow].Value.ToString();  workerName = dataGridView1[2, indexSelectRow].Value.ToString();  clientName = dataGridView1[3, indexSelectRow].Value.ToString();  dateReservation = dataGridView1[4, indexSelectRow].Value.ToString();  scheduledDateArrival = dataGridView1[5, indexSelectRow].Value.ToString();  scheduledDateLeave = dataGridView1[6, indexSelectRow].Value.ToString();  peopleNum = dataGridView1[7, indexSelectRow].Value.ToString();  reservationType = dataGridView1[8, indexSelectRow].Value.ToString();  reservaionPrice = dataGridView1[9, indexSelectRow].Value.ToString();  }  }  public void OnlyRead()  {  addBig\_ToolStripMenuItem.Visible = false;  add\_ToolStripMenuItem.Visible = false;  delete\_ToolStripMenuItem.Visible = false;  update\_ToolStripMenuItem.Visible = false;  }  }  } |

Код файла room\_list

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class room\_list : Form  {  SqlConnection myConn = new SqlConnection();  int indexSelectRow;  string idRoom, comfortName, placementName, price, state;  public room\_list()  {  InitializeComponent();  }  private void Form13\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  loadDataComboBox1();  loadDataComboBox2();  // Запускаем процедуру выборки данных  loadData();  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  idRoom = dataGridView1[0, indexSelectRow].Value.ToString();  comfortName = dataGridView1[1, indexSelectRow].Value.ToString();  placementName = dataGridView1[2, indexSelectRow].Value.ToString();  price = dataGridView1[3, indexSelectRow].Value.ToString();  state = dataGridView1[4, indexSelectRow].Value.ToString();  }  }  private void add\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  idRoom = dataGridView1[0, indexSelectRow].Value.ToString();  comfortName = dataGridView1[1, indexSelectRow].Value.ToString();  placementName = dataGridView1[2, indexSelectRow].Value.ToString();  price = dataGridView1[3, indexSelectRow].Value.ToString();  state = dataGridView1[4, indexSelectRow].Value.ToString();  if (idRoom.Trim() == "")  {    if (comfortName.Trim() == "" || placementName.Trim() == "" ||  price.Trim() == "") throw new Exception();  // Создать команду для добавления  SqlCommand myComm = new SqlCommand("execute add\_room @p1, @p2, @p3, @p4", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 40);  myComm.Parameters["@p1"].Value = comfortName;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 40);  myComm.Parameters["@p2"].Value = placementName;  myComm.Parameters.Add("@p3", SqlDbType.Money, 100);  myComm.Parameters["@p3"].Value = price;  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 25);  myComm.Parameters["@p4"].Value = state;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  private void delete\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {    if (idRoom == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_room @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 10);  myComm.Parameters["@p1"].Value = idRoom;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();    // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {    // Присваиваем переменным обновленные значения  idRoom = dataGridView1[0, indexSelectRow].Value.ToString();  comfortName = dataGridView1[1, indexSelectRow].Value.ToString();  placementName = dataGridView1[2, indexSelectRow].Value.ToString();  price = dataGridView1[3, indexSelectRow].Value.ToString();  state = dataGridView1[4, indexSelectRow].Value.ToString();  if (idRoom.Trim() == "") throw new Exception();  // Создать команду для изменения  SqlCommand myComm = new SqlCommand("execute update\_room @p1, @p2, @p3, @p4, @p5", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.Int);  myComm.Parameters["@p1"].Value = idRoom;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = comfortName;  myComm.Parameters.Add("@p3", SqlDbType.NVarChar, 100);  myComm.Parameters["@p3"].Value = placementName;  myComm.Parameters.Add("@p4", SqlDbType.Money, 100);  myComm.Parameters["@p4"].Value = price;  myComm.Parameters.Add("@p5", SqlDbType.NVarChar, 100);  myComm.Parameters["@p5"].Value = state;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void info\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  }  private void back\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Hide();  Program.openStartForm();  Close();  }  private void loadDataComboBox1()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select Название AS Комфортность from Комфортность", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  comfort\_name.DisplayMember = "Комфортность";  comfort\_name.DataSource = dtbl;  }  private void loadDataComboBox2()  {  SqlDataAdapter sqlDa = new SqlDataAdapter("select Название AS Размещение from Размещение", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  placement\_name.DisplayMember = "Размещение";  placement\_name.DataSource = dtbl;  }  private void loadData()  {  SqlCommand myComm = new SqlCommand("select\*from dbo.View\_Room", myConn);  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  dtbl.Columns[3].AllowDBNull = true;  dtbl.Columns[4].AllowDBNull = true;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  // Если в таблице есть минимум одна запись  if (dataGridView1.Rows.Count > 1)  {  idRoom = dataGridView1[0, indexSelectRow].Value.ToString();  comfortName = dataGridView1[1, indexSelectRow].Value.ToString();  placementName = dataGridView1[2, indexSelectRow].Value.ToString();  price = dataGridView1[3, indexSelectRow].Value.ToString();  state = dataGridView1[4, indexSelectRow].Value.ToString();  }  }  }  } |

Код файл service\_information

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class service\_information : Form  {  SqlConnection myConn = new SqlConnection();  string idService, nameService, costService;  string dateService, idАrrival, usageCount;  int indexSelectRow;  public service\_information(string idService, string nameService, string costService)  {  this.idService = idService;  this.nameService = nameService;  this.costService = costService;  InitializeComponent();  }  private void add\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  dateService = dataGridView1[0, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[1, indexSelectRow].Value.ToString();  usageCount = dataGridView1[2, indexSelectRow].Value.ToString();    if (nameService.Trim() == "" || dateService.Trim() == "" ||  idАrrival.Trim() == "" || usageCount.Trim() == "") throw new Exception();  // Создать команду для добавления  SqlCommand myComm = new SqlCommand("execute add\_using\_service @p1, @p2, @p3, @p4", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = idАrrival;  myComm.Parameters.Add("@p3", SqlDbType.SmallDateTime);  myComm.Parameters["@p3"].Value = dateService;  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 100);  myComm.Parameters["@p4"].Value = usageCount;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  private void delete\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {    if (nameService == "" || dateService == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_using\_service @p1, @p2", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  myComm.Parameters.Add("@p2", SqlDbType.SmallDateTime);  myComm.Parameters["@p2"].Value = dateService;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();    // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_Click(object sender, EventArgs e)  {  try  {    // Проверка на последнюю строчку  if (indexSelectRow >= dataGridView1.Rows.Count - 1) throw new Exception();  // Запоминаем старые значение (составной ключ)  string oldNameService = nameService;  string oldDateService = dateService;  // Присваиваем переменным обновленные значения  dateService = dataGridView1[0, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[1, indexSelectRow].Value.ToString();  usageCount = dataGridView1[2, indexSelectRow].Value.ToString();  if (nameService.Trim() == "" || dateService.Trim() == "" ||  idАrrival.Trim() == "" || usageCount.Trim() == "") throw new Exception();  // Создать команду для изменения  SqlCommand myComm = new SqlCommand("execute update\_using\_service @p1, @p2, @p3, @p4, @p5, @p6", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = idАrrival;  myComm.Parameters.Add("@p3", SqlDbType.SmallDateTime);  myComm.Parameters["@p3"].Value = dateService;  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 100);  myComm.Parameters["@p4"].Value = usageCount;  myComm.Parameters.Add("@p5", SqlDbType.NVarChar, 100);  myComm.Parameters["@p5"].Value = oldNameService;  myComm.Parameters.Add("@p6", SqlDbType.SmallDateTime);  myComm.Parameters["@p6"].Value = oldDateService;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void Form8\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  loadData2ComboBox();  // Запускаем процедуру выборки данных  loadData();  // Устанавливаем значения полям  textBox1.Text = idService;  textBox2.Text = nameService;  textBox3.Text = costService;  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  dateService = dataGridView1[0, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[1, indexSelectRow].Value.ToString();  usageCount = dataGridView1[2, indexSelectRow].Value.ToString();  }  }  private void loadData2ComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select id\_Заезд from dbo.Заезд", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  id\_arrival.DisplayMember = "id\_Заезд";  id\_arrival.DataSource = dtbl;    }  private void loadData()  {    SqlCommand myComm = new SqlCommand("select Дата\_пользования, Заезд\_id\_Заезд, Количество\_пользований from dbo.View\_UsageService where Название = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  // Если в таблице есть минимум одна запись  if (dataGridView1.Rows.Count > 1)  {  dateService = dataGridView1[0, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[1, indexSelectRow].Value.ToString();  usageCount = dataGridView1[2, indexSelectRow].Value.ToString();  }  }  }  } |

Код файла service\_using\_information

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class service\_using\_information : Form  {  SqlConnection myConn = new SqlConnection();  string nameService, dateService, idАrrival, usageCount;  public service\_using\_information(string nameService, string dateService, string idАrrival, string usageCount)  {  this.nameService = nameService;  this.dateService = dateService;  this.idАrrival = idАrrival;  this.usageCount = usageCount;  InitializeComponent();  }  private void Form10\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Запускаем процедуру выборки данных  loadData1();  loadData2();  // Устанавливаем значения полям  textBox1.Text = nameService;  textBox2.Text = dateService;  textBox3.Text = idАrrival;  textBox4.Text = usageCount;  }  private void loadData1()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_UsageService WHERE Название = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    }  private void loadData2()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_Arrival WHERE id\_Заезд = @p1", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = idАrrival;  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView2.DataSource = dtbl;    }  }  } |

Код файла service\_using\_list

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class service\_using\_list : Form  {  SqlConnection myConn = new SqlConnection();  int indexSelectRow;  string nameService, dateService, idАrrival, usageCount;  public service\_using\_list()  {  InitializeComponent();  }  private void Form9\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  // Заполняем объект типа ComboBox  loadData1ComboBox();  loadData2ComboBox();  // Запускаем процедуру выборки данных  loadData();  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  nameService = dataGridView1[0, indexSelectRow].Value.ToString();  dateService = dataGridView1[1, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[2, indexSelectRow].Value.ToString();  usageCount = dataGridView1[3, indexSelectRow].Value.ToString();  }  }  private void add\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Берем значения для добавления из предпоследней строчки (последняя - пустая)  indexSelectRow = dataGridView1.Rows.Count - 2;  nameService = dataGridView1[0, indexSelectRow].Value.ToString();  dateService = dataGridView1[1, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[2, indexSelectRow].Value.ToString();  usageCount = dataGridView1[3, indexSelectRow].Value.ToString();    if (nameService.Trim() == "" || dateService.Trim() == "" ||  idАrrival.Trim() == "" || usageCount.Trim() == "") throw new Exception();  // Создать команду для добавления  SqlCommand myComm = new SqlCommand("execute add\_using\_service @p1, @p2, @p3, @p4", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = idАrrival;  myComm.Parameters.Add("@p3", SqlDbType.SmallDateTime);  myComm.Parameters["@p3"].Value = dateService;  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 100);  myComm.Parameters["@p4"].Value = usageCount;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь добавить пустую строку.", "Внимание!");  }  }  internal void OnlyRead()  {  add\_ToolStripMenuItem.Visible = false;  delete\_ToolStripMenuItem.Visible = false;  update\_ToolStripMenuItem.Visible = false;  }  private void delete\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {    if (nameService == "" || dateService == "") throw new Exception();  // Создать команду для удаления  SqlCommand myComm = new SqlCommand("execute delete\_using\_service @p1, @p2", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  myComm.Parameters.Add("@p2", SqlDbType.SmallDateTime);  myComm.Parameters["@p2"].Value = dateService;  // Выполнить запрос на удаление без возвращения результата  myComm.ExecuteReader();    // Обновляем содержимое  loadData();  }  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь удалить пустую строку.", "Внимание!");  }  }  private void update\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {    // Запоминаем старые значение (составной ключ)  string oldNameService = nameService;  string oldDateService = dateService;  // Присваиваем переменным обновленные значения  nameService = dataGridView1[0, indexSelectRow].Value.ToString();  dateService = dataGridView1[1, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[2, indexSelectRow].Value.ToString();  usageCount = dataGridView1[3, indexSelectRow].Value.ToString();  if (nameService.Trim() == "" || dateService.Trim() == "" ||  idАrrival.Trim() == "" || usageCount.Trim() == "") throw new Exception();  // Создать команду для изменения  SqlCommand myComm = new SqlCommand("execute update\_using\_service @p1, @p2, @p3, @p4, @p5, @p6", myConn);  // Создать параметр и передать в него значение текстового поля  myComm.Parameters.Add("@p1", SqlDbType.NVarChar, 100);  myComm.Parameters["@p1"].Value = nameService;  myComm.Parameters.Add("@p2", SqlDbType.NVarChar, 100);  myComm.Parameters["@p2"].Value = idАrrival;  myComm.Parameters.Add("@p3", SqlDbType.SmallDateTime);  myComm.Parameters["@p3"].Value = dateService;  myComm.Parameters.Add("@p4", SqlDbType.NVarChar, 100);  myComm.Parameters["@p4"].Value = usageCount;  myComm.Parameters.Add("@p5", SqlDbType.NVarChar, 100);  myComm.Parameters["@p5"].Value = oldNameService;  myComm.Parameters.Add("@p6", SqlDbType.SmallDateTime);  myComm.Parameters["@p6"].Value = oldDateService;  // Выполнить запрос на изменение без возвращения результата  myComm.ExecuteNonQuery();    // Обновляем содержимое  loadData();  }  catch  {    MessageBox.Show("Ошибка. Возможное решение:\n\n " +  " 1. Возможно вы пытаетесь редактировать пустую строку.", "Внимание!");  }  }  private void info\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  if (nameService.Trim() == "") throw new Exception();  service\_using\_information form10 = new service\_using\_information(nameService, dateService, idАrrival, usageCount);  form10.ShowDialog();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  "1. Возможно вы пытаетесь посмотреть информацию о несуществующих данных.", "Внимание!");  }  }  private void back\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Hide();  Program.openStartForm();  Close();  }  private void loadData1ComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select Название from dbo.Дополнительные\_услуги", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  name\_service.DisplayMember = "Название";  name\_service.DataSource = dtbl;    }  private void loadData2ComboBox()  {    SqlDataAdapter sqlDa = new SqlDataAdapter("select id\_Заезд from dbo.Заезд", myConn);  DataTable dtbl = new DataTable();  sqlDa.Fill(dtbl);  id\_arrival.DisplayMember = "id\_Заезд";  id\_arrival.DataSource = dtbl;    }  private void loadData()  {    SqlCommand myComm = new SqlCommand("select\*from dbo.View\_UsageService", myConn);  SqlDataReader myReader = myComm.ExecuteReader();  DataTable dtbl = new DataTable(); dtbl.Load(myReader);  dataGridView1.DataSource = dtbl;    // Разрешаем нулевые значения (иначе событие DataError)  dtbl.Columns[0].AllowDBNull = true;  dtbl.Columns[1].AllowDBNull = true;  dtbl.Columns[2].AllowDBNull = true;  dtbl.Columns[3].AllowDBNull = true;  // Устанавливаем значения по умолчанию переменным (первая строка)  indexSelectRow = 0;  // Если в таблице есть минимум одна запись  if (dataGridView1.Rows.Count > 1)  {  nameService = dataGridView1[0, indexSelectRow].Value.ToString();  dateService = dataGridView1[1, indexSelectRow].Value.ToString();  idАrrival = dataGridView1[2, indexSelectRow].Value.ToString();  usageCount = dataGridView1[3, indexSelectRow].Value.ToString();  }  }  }  } |

Код файла universal

|  |
| --- |
| using System;  using System.Data;  using System.Data.SqlClient;  using System.Windows.Forms;  namespace Hotel\_Database  {  public partial class universal : Form  {  SqlConnection myConn = new SqlConnection();  SqlCommand myCommPlacement = new SqlCommand("select\*from dbo.Размещение");  SqlCommand myCommComfort = new SqlCommand("select\*from dbo.Комфортность");  SqlCommand myCommStaff = new SqlCommand("select\*from dbo.Сотрудники");  SqlCommand myCommService = new SqlCommand("select\*from dbo.Дополнительные\_услуги");  string idService, nameService, costService;  int indexSelectRow;  SqlDataAdapter sda = new SqlDataAdapter(); DataSet ds = new DataSet();  int idNumberTable;  public universal(int idNumberTable)  {  InitializeComponent();  this.idNumberTable = idNumberTable;  }  private void Form3\_Load(object sender, EventArgs e)  {  Program.openConnection(myConn);  try  {  switch (idNumberTable)  {  case 1:  {  // Устанавливаем заголовок формы  this.Text = "Список размещений";  info\_ToolStripMenuItem.Visible = false;  // Настраиваем стобец "Номер размещения"  id.HeaderText = "Номер размещения";  id.DataPropertyName = "id\_Размещение";  // Настраиваем стобец "Название"  name.HeaderText = "Название";  name.DataPropertyName = "Название";  // Избавляемся от ненужных столбцов  dataGridView1.Columns.Remove(passport);  dataGridView1.Columns.Remove(date);  dataGridView1.Columns.Remove(address);  dataGridView1.Columns.Remove(number);  dataGridView1.Columns.Remove(post);  dataGridView1.Columns.Remove(cost);  // Выборка создания и заполнения в DataSet таблицы  myCommPlacement.Connection = myConn;  sda.SelectCommand = myCommPlacement;  sda.Fill(ds, "Размещение");  dataGridView1.DataSource = ds.Tables["Размещение"];  }  break;  case 2:  {  // Устанавливаем заголовок формы  this.Text = "Список комфортностей";  info\_ToolStripMenuItem.Visible = false;  // Настраиваем стобец "Номер комфортности"  id.HeaderText = "Номер комфортности";  id.DataPropertyName = "id\_Комфортность";  // Настраиваем стобец "Название"  name.HeaderText = "Название";  name.DataPropertyName = "Название";  // Избавляемся от ненужных столбцов  dataGridView1.Columns.Remove(passport);  dataGridView1.Columns.Remove(date);  dataGridView1.Columns.Remove(address);  dataGridView1.Columns.Remove(number);  dataGridView1.Columns.Remove(post);  dataGridView1.Columns.Remove(cost);  // Выборка создания и заполнения в DataSet таблицы  myCommComfort.Connection = myConn;  sda.SelectCommand = myCommComfort;  sda.Fill(ds, "Комфортность");  dataGridView1.DataSource = ds.Tables["Комфортность"];  }  break;  case 3:  {  // Устанавливаем заголовок формы  this.Text = "Список дополнительных услуг";  // Настраиваем стобец "Номер услуги"  id.HeaderText = "Номер услуги";  id.DataPropertyName = "id\_Дополнительные\_услуги";  // Настраиваем стобец "Название"  name.HeaderText = "Название";  name.DataPropertyName = "Название";  // Настраиваем столбец "Цена"  cost.HeaderText = "Цена";  cost.DataPropertyName = "Цена";  // Избавляемся от ненужных столбцов  dataGridView1.Columns.Remove(passport);  dataGridView1.Columns.Remove(date);  dataGridView1.Columns.Remove(address);  dataGridView1.Columns.Remove(number);  dataGridView1.Columns.Remove(post);  // Выборка создания и заполнения в DataSet таблицы  myCommService.Connection = myConn;  sda.SelectCommand = myCommService;  sda.Fill(ds, "Услуга");  dataGridView1.DataSource = ds.Tables["Услуга"];  // Если пользователь нажмет просмотр информации без выбора записи  idService = dataGridView1[0, indexSelectRow].Value.ToString();  nameService = dataGridView1[1, indexSelectRow].Value.ToString();  costService = dataGridView1[2, indexSelectRow].Value.ToString();  }  break;  case 4:  {  // Устанавливаем заголовок формы  this.Text = "Список сотрудников";  info\_ToolStripMenuItem.Visible = false;  // Устанавливаем размер формы  this.Width = 1000;  this.Height = 500;  // Перемещаем форму в центр  this.CenterToScreen();  // Настраиваем стобец "Номер сотрудника"  id.HeaderText = "Номер сотрудника";  id.DataPropertyName = "id\_Сотрудники";  // Настраиваем стобец "ФИО"  name.HeaderText = "ФИО";  name.DataPropertyName = "ФИО\_Сотрудника";  // Настраиваем столбец "Паспорт"  passport.DataPropertyName = "Номер\_и\_серия\_паспорта";  passport.HeaderText = "Паспорт";  // Настраиваем столбец "Дата рождения"  date.HeaderText = "Дата рождения";  date.DataPropertyName = "Дата\_рождения";  // Настраиваем столбец "Адрес проживания"  address.HeaderText = "Адрес проживания";  address.DataPropertyName = "Адрес\_проживания";  // Настраиваем столбец "Номер телефона"  number.HeaderText = "Номер телефона";  number.DataPropertyName = "Номер\_телефона";  // Настраиваем столбец "Должность"  post.HeaderText = "Должность";  post.DataPropertyName = "Должность";  // Избавляемся от ненужных столбцов  dataGridView1.Columns.Remove(cost);  // Выборка создания и заполнения в DataSet таблицы  myCommStaff.Connection = myConn;  sda.SelectCommand = myCommStaff;  sda.Fill(ds, "Сотрудник");  dataGridView1.DataSource = ds.Tables["Сотрудник"];  }  break;  }  }  catch { }  dataGridView1.Refresh();  }  private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)  {  if (idNumberTable == 3)  {  // Для того, чтобы не обновлялись значения, если нажатия происходит в одной строке  if (e.RowIndex != indexSelectRow && e.RowIndex != -1)  {  indexSelectRow = e.RowIndex;  idService = dataGridView1[0, indexSelectRow].Value.ToString();  nameService = dataGridView1[1, indexSelectRow].Value.ToString();  costService = dataGridView1[2, indexSelectRow].Value.ToString();  }  }  }  private void add\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  switch (idNumberTable)  {  case 1:  ds.Tables["Размещение"].Rows.Add();  break;  case 2:  ds.Tables["Комфортность"].Rows.Add();  break;  case 3:  ds.Tables["Услуга"].Rows.Add();  break;  case 4:  ds.Tables["Сотрудник"].Rows.Add();  break;  }  }  private void delete\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  DialogResult result = MessageBox.Show("Данная информация будет удалена. Продолжить?", "Внимание!", MessageBoxButtons.YesNo);  if (result == DialogResult.Yes)  {  int rowIndex = dataGridView1.CurrentCell.RowIndex;  dataGridView1.Rows.RemoveAt(rowIndex);  }  }  catch { MessageBox.Show("Почему-то вызвалось исключение.", "Внимание!"); }  }  private void reloadTable(string tableName) {  sda.Update(ds.Tables[tableName]);  // очищаем  ds.Clear();  sda.Fill(ds, tableName);  }  private void save\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  // Создаем команды манипулирования данными  SqlCommandBuilder scb = new SqlCommandBuilder(sda);  scb.GetUpdateCommand(); scb.GetDeleteCommand(); scb.GetInsertCommand();  // Отправляем изменения в БД  switch (idNumberTable)  {  case 1:  reloadTable("Размещение");  break;  case 2:  reloadTable("Комфортность");  break;  case 3:  reloadTable("Услуга");  break;  case 4:  reloadTable("Сотрудник");  break;  }  }  catch { MessageBox.Show("Необходимо заполнить добавленную строку", "Внимание!"); }  }  private void info\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  try  {  if (idService.Trim() == "") throw new Exception();  service\_information form8 = new service\_information(idService, nameService, costService);  form8.ShowDialog();  }  catch  {  MessageBox.Show("Ошибка. Возможное решение:\n\n " +  "1. Возможно вы пытаетесь посмотреть информацию о несуществующих данных.", "Внимание!");  }  }  private void back\_ToolStripMenuItem\_Click(object sender, EventArgs e)  {  Hide();  Program.openStartForm();  Close();  }  }  } |