**Приложение А. Листинг.**

MainForm.cs

**using System;**

**using System.Collections.Generic;**

**using System.ComponentModel;**

**using System.Data;**

**using System.Data.SqlClient;**

**using System.Drawing;**

**using System.Linq;**

**using System.Text;**

**using System.Threading.Tasks;**

**using System.Windows.Forms;**

**namespace DB**

**{**

**public partial class MainForm : Form**

**{**

**public const string connectionString = "Data Source=ORANGE\\MSSQLEXPRESS2017;Initial Catalog=AutoParts;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False";**

**bool textBoxSearchActiveStorehouse = false;**

**bool textBoxSearchActiveParts = false;**

**bool textBoxSearchActiveProviders = false;**

**bool textBoxSearchActivePriceview = false;**

**//записки**

**//https://www.flaticon.com/free-icon/plus\_128575 иконка "назад"**

**//42880b - зеленые иконки**

**//1568A3 синие**

**//db342d красные**

**//E8C538 желтые**

**public MainForm()**

**{**

**InitializeComponent();**

**}**

**private void MainForm\_Load(object sender, EventArgs e)**

**{**

**try**

**{**

**this.bridge\_providers\_partsTableAdapter.Fill(this.autoPartsDataSet.bridge\_providers\_parts);**

**} catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из таблицы bridge\_providers\_parts", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**try**

**{**

**this.dealsTableAdapter.Fill(this.autoPartsDataSet.deals);**

**} catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из таблицы deals", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**try**

**{**

**this.partsTableAdapter.Fill(this.autoPartsDataSet.parts);**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из таблицы parts", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**try**

**{**

**this.providersTableAdapter.Fill(this.autoPartsDataSet.providers);**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из таблицы providers", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**try**

**{**

**this.storehouseTableAdapter.Fill(this.autoPartsDataSet.storehouse);**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления storehouse", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**try**

**{**

**this.priceviewTableAdapter.Fill(this.autoPartsDataSet.priceview);**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления priceview", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**labelRowCountStorehouse.Text += dataGridViewStorehouse.RowCount.ToString();**

**labelRowCountParts.Text = "Количество записей: " + dataGridViewParts.RowCount.ToString();**

**labelRowCountProviders.Text = "Количество записей: " + dataGridViewProviders.RowCount.ToString();**

**labelRowCountPriceview.Text = "Количество записей: " + dataGridViewPriceview.RowCount.ToString();**

**}**

**public void refreshAfterInsertStorehouse()**

**{**

**dataGridViewStorehouse.ClearSelection();**

**this.storehouseTableAdapter.Fill(this.autoPartsDataSet.storehouse);**

**dataGridViewStorehouse.FirstDisplayedScrollingRowIndex = dataGridViewStorehouse.RowCount - 1;**

**labelRowCountStorehouse.Text = "Количество записей: " + dataGridViewStorehouse.RowCount.ToString();**

**}**

**public void refreshAfterDeleteStorehouse()**

**{**

**dataGridViewStorehouse.ClearSelection();**

**this.storehouseTableAdapter.Fill(this.autoPartsDataSet.storehouse);**

**labelRowCountStorehouse.Text = "Количество записей: " + dataGridViewStorehouse.RowCount.ToString();**

**}**

**public void refreshAfterInsertParts()**

**{**

**dataGridViewParts.ClearSelection();**

**this.partsTableAdapter.Fill(this.autoPartsDataSet.parts);**

**dataGridViewParts.FirstDisplayedScrollingRowIndex = dataGridViewParts.RowCount - 1;**

**labelRowCountParts.Text = "Количество записей: " + dataGridViewParts.RowCount.ToString();**

**}**

**public void refreshAfterInsertParts(AddDealsForm updateForm)**

**{**

**dataGridViewParts.ClearSelection();**

**this.partsTableAdapter.Fill(this.autoPartsDataSet.parts);**

**dataGridViewParts.FirstDisplayedScrollingRowIndex = dataGridViewParts.RowCount - 1;**

**labelRowCountParts.Text = "Количество записей: " + dataGridViewParts.RowCount.ToString();**

**foreach (DataRow row in this.autoPartsDataSet.parts.Select())**

**{**

**updateForm.textBoxTitle.AutoCompleteCustomSource.Add(**

**row.ItemArray[0].ToString());**

**updateForm.textBoxManufacturer.AutoCompleteCustomSource.Add(**

**row.ItemArray[2].ToString());**

**}**

**}**

**public void refreshAfterDeleteParts()**

**{**

**dataGridViewParts.ClearSelection();**

**this.partsTableAdapter.Fill(this.autoPartsDataSet.parts);**

**labelRowCountParts.Text = "Количество записей: " + dataGridViewParts.RowCount.ToString();**

**}**

**public void refreshAfterInsertProviders()**

**{**

**dataGridViewProviders.ClearSelection();**

**this.providersTableAdapter.Fill(this.autoPartsDataSet.providers);**

**dataGridViewProviders.FirstDisplayedScrollingRowIndex = dataGridViewProviders.RowCount - 1;**

**labelRowCountProviders.Text = "Количество записей: " + dataGridViewProviders.RowCount.ToString();**

**}**

**public void refreshAfterInsertProviders(AddDealsForm updateForm)**

**{**

**dataGridViewProviders.ClearSelection();**

**this.providersTableAdapter.Fill(this.autoPartsDataSet.providers);**

**dataGridViewProviders.FirstDisplayedScrollingRowIndex = dataGridViewProviders.RowCount - 1;**

**labelRowCountProviders.Text = "Количество записей: " + dataGridViewProviders.RowCount.ToString();**

**updateForm.comboBoxProvider.Items.Clear();**

**foreach (DataRow row in this.autoPartsDataSet.providers.Select())**

**{**

**updateForm.comboBoxProvider.Items.Add(**

**row.ItemArray[0].ToString());**

**}**

**}**

**public void refreshAfterDeleteProviders()**

**{**

**dataGridViewProviders.ClearSelection();**

**this.providersTableAdapter.Fill(this.autoPartsDataSet.providers);**

**labelRowCountProviders.Text = "Количество записей: " + dataGridViewProviders.RowCount.ToString();**

**}**

**public void refreshAfterDeletePriceview()**

**{**

**dataGridViewPriceview.ClearSelection();**

**this.priceviewTableAdapter.Fill(this.autoPartsDataSet.priceview);**

**labelRowCountPriceview.Text = "Количество записей: " + dataGridViewPriceview.RowCount.ToString();**

**}**

**//Storehouse & Deals**

**private void SearchStorehouse(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter && textBoxSearchStorehouse.Text != "")**

**{**

**bool getEntryInRow = false;**

**foreach (DataGridViewRow row in dataGridViewStorehouse.Rows)**

**{**

**row.Visible = true;**

**}**

**dataGridViewStorehouse.ClearSelection();**

**dataGridViewStorehouse.CurrentCell = null;**

**if (comboboxSearchByStorehouse.Text == "Все")**

**{**

**for (int i = 0; i < dataGridViewStorehouse.RowCount; i++)**

**{**

**dataGridViewStorehouse.Rows[i].Selected = false;**

**for (int j = 0; j < dataGridViewStorehouse.ColumnCount; j++)**

**{**

**if (dataGridViewStorehouse.Rows[i].Cells[j].Value != null)**

**{**

**if (dataGridViewStorehouse.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchStorehouse.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewStorehouse.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewStorehouse.Rows[i].Selected = true;**

**break;**

**}**

**}**

**}**

**}**

**if (getEntryInRow == false) {**

**MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**return;**

**}**

**else**

**{**

**foreach (DataGridViewRow row in dataGridViewStorehouse.Rows)**

**{**

**if (!row.Selected) row.Visible = false;**

**}**

**dataGridViewStorehouse.ClearSelection();**

**for (int i = 0; i < dataGridViewStorehouse.RowCount; i++)**

**{**

**if (dataGridViewStorehouse.Rows[i].Visible)**

**{**

**for (int j = 0; j < dataGridViewStorehouse.ColumnCount; j++)**

**{**

**if (dataGridViewStorehouse.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchStorehouse.Text.ToLower()))**

**{**

**dataGridViewStorehouse.Rows[i].Cells[j].Selected = true;**

**}**

**}**

**}**

**}**

**}**

**}**

**else**

**{**

**int columnSearchBy;**

**switch (comboboxSearchByStorehouse.Text)**

**{**

**case "id":**

**columnSearchBy = 0;**

**break;**

**case "Название":**

**columnSearchBy = 1;**

**break;**

**case "Код":**

**columnSearchBy = 2;**

**break;**

**case "Цена":**

**columnSearchBy = 3;**

**break;**

**case "Количество":**

**columnSearchBy = 4;**

**break;**

**case "Производитель":**

**columnSearchBy = 5;**

**break;**

**case "Цена закупки":**

**columnSearchBy = 6;**

**break;**

**case "Дата закупки":**

**columnSearchBy = 7;**

**break;**

**case "Поставщик":**

**columnSearchBy = 8;**

**break;**

**default:**

**MessageBox.Show("Ошибка поиска! Выбранное поле отсутствует в таблице. Поиск будет произведен по названию.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**columnSearchBy = 1;**

**break;**

**}**

**for (int i = 0; i < dataGridViewStorehouse.RowCount; i++)**

**{**

**dataGridViewStorehouse.Rows[i].Selected = false;**

**if (dataGridViewStorehouse.Rows[i].Cells[columnSearchBy].Value != null)**

**{**

**if (dataGridViewStorehouse.Rows[i].Cells[columnSearchBy].Value.ToString().ToLower().Contains(textBoxSearchStorehouse.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewStorehouse.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewStorehouse.Rows[i].Cells[columnSearchBy].Selected = true;**

**}**

**}**

**}**

**if (getEntryInRow == false) MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**else if (e.KeyCode == Keys.Enter && textBoxSearchStorehouse.Text == "")**

**{**

**MessageBox.Show("Введите данные для поиска", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**private void buttonRefreshStorehouse\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**this.storehouseTableAdapter.Fill(this.autoPartsDataSet.storehouse);**

**foreach (DataGridViewColumn column in dataGridViewStorehouse.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**dataGridViewStorehouse.ClearSelection();**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления storehouse", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**labelRowCountStorehouse.Text = "Количество записей: " + dataGridViewStorehouse.RowCount.ToString();**

**}**

**private void buttonCleanStorehouse\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**if (dataGridViewStorehouse.RowCount > 0)**

**{**

**dataGridViewStorehouse.Sort(dataGridViewStorehouse.Columns[0], ListSortDirection.Ascending);**

**foreach (DataGridViewColumn column in dataGridViewStorehouse.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**foreach (DataGridViewRow row in dataGridViewStorehouse.Rows) row.Visible = true;**

**dataGridViewStorehouse.FirstDisplayedScrollingRowIndex = 0;**

**dataGridViewStorehouse.ClearSelection();**

**labelRowCountStorehouse.Text = "Количество записей: " + dataGridViewStorehouse.RowCount.ToString();**

**}**

**else**

**{**

**MessageBox.Show("Отсутствуют данные, сначала обновите таблицу.", "Ошибка загрузки данных", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**}**

**} catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления storehouse", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**private void comboBoxSearchByStorehouse\_KeyDown(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter) SearchStorehouse(sender, e);**

**}**

**private void textBoxSearchStorehouse\_Enter(object sender, EventArgs e)**

**{**

**if (textBoxSearchActiveStorehouse == false)**

**{**

**textBoxSearchStorehouse.Text = "";**

**textBoxSearchStorehouse.ForeColor = Color.Black;**

**textBoxSearchActiveStorehouse = true;**

**}**

**}**

**private void textBoxSearchStorehouse\_Leave(object sender, EventArgs e)**

**{**

**if (textBoxSearchStorehouse.Text == "")**

**{**

**textBoxSearchStorehouse.ForeColor = Color.Gray;**

**textBoxSearchStorehouse.Text = "Поиск🔍";**

**textBoxSearchActiveStorehouse = false;**

**} else**

**{**

**textBoxSearchActiveStorehouse = true;**

**}**

**}**

**private void textBoxSearchStorehouse\_KeyDown(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter)**

**{**

**SearchStorehouse(sender, e);**

**e.Handled = e.SuppressKeyPress = true;**

**}**

**}**

**private void buttonSearchClearStorehouse\_Click(object sender, EventArgs e)**

**{**

**textBoxSearchStorehouse.ForeColor = Color.Gray;**

**textBoxSearchStorehouse.Text = "Поиск🔍";**

**textBoxSearchActiveStorehouse = false;**

**}**

**private void buttonSelectRowStorehouse\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewStorehouse.SelectedCells;**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewStorehouse.Rows[cell.RowIndex].Selected = true;**

**}**

**}**

**private void buttonAddStorehouse\_Click(object sender, EventArgs e)**

**{**

**AddDealsForm addDealsForm = new AddDealsForm(this);**

**foreach (DataRow row in this.autoPartsDataSet.parts.Select())**

**{**

**addDealsForm.textBoxTitle.AutoCompleteCustomSource.Add(**

**row.ItemArray[0].ToString());**

**addDealsForm.textBoxManufacturer.AutoCompleteCustomSource.Add(**

**row.ItemArray[2].ToString());**

**}**

**foreach (DataRow row in this.autoPartsDataSet.providers.Select())**

**{**

**addDealsForm.comboBoxProvider.Items.Add(**

**row.ItemArray[0].ToString());**

**}**

**addDealsForm.Show();**

**}**

**private void buttonDeleteStorehouse\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewStorehouse.SelectedCells;**

**if (selectedCells.Count == 0)**

**{**

**MessageBox.Show("Не выбраны строки для удаления", "Ошибка удаления", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**return;**

**}**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewStorehouse.Rows[cell.RowIndex].Selected = true;**

**}**

**if (MessageBox.Show("Выбранные строки удалятся из базы данных. Продолжить?", "Удалить запись", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)**

**{**

**try**

**{**

**using (SqlConnection connection = new SqlConnection(connectionString))**

**{**

**connection.Open();**

**foreach (DataGridViewRow selectedRow in dataGridViewStorehouse.SelectedRows)**

**{**

**string expression = "delete from deals where id = " + selectedRow.Cells[0].Value;**

**SqlCommand request = new SqlCommand(expression, connection);**

**request.ExecuteNonQuery();**

**if (dataGridViewStorehouse.SelectedRows.Count <= 1) break;**

**}**

**this.refreshAfterDeleteStorehouse();**

**connection.Close();**

**}**

**} catch (SqlException err)**

**{**

**if (err.Number == 547)**

**{**

**MessageBox.Show("Невозможно удалить выбранную строку (строки) из таблицы deals, так как от нее зависимы другие таблицы. " +**

**"Сначала удалите связанные строки в них.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**if (dataGridViewStorehouse.SelectedRows.Count > 1) return;**

**}**

**else**

**{**

**MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**}**

**}**

**//Parts**

**private void SearchParts(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter && textBoxSearchParts.Text != "")**

**{**

**bool getEntryInRow = false;**

**foreach (DataGridViewRow row in dataGridViewParts.Rows)**

**{**

**row.Visible = true;**

**}**

**dataGridViewParts.ClearSelection();**

**dataGridViewParts.CurrentCell = null;**

**if (comboboxSearchByParts.Text == "Все")**

**{**

**for (int i = 0; i < dataGridViewParts.RowCount; i++)**

**{**

**dataGridViewParts.Rows[i].Selected = false;**

**for (int j = 0; j < dataGridViewParts.ColumnCount; j++)**

**{**

**if (dataGridViewParts.Rows[i].Cells[j].Value != null)**

**{**

**if (dataGridViewParts.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchParts.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewParts.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewParts.Rows[i].Selected = true;**

**break;**

**}**

**}**

**}**

**}**

**if (getEntryInRow == false)**

**{**

**MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**return;**

**}**

**else**

**{**

**foreach (DataGridViewRow row in dataGridViewParts.Rows)**

**{**

**if (!row.Selected) row.Visible = false;**

**}**

**dataGridViewParts.ClearSelection();**

**for (int i = 0; i < dataGridViewParts.RowCount; i++)**

**{**

**if (dataGridViewParts.Rows[i].Visible)**

**{**

**for (int j = 0; j < dataGridViewParts.ColumnCount; j++)**

**{**

**if (dataGridViewParts.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchParts.Text.ToLower()))**

**{**

**dataGridViewParts.Rows[i].Cells[j].Selected = true;**

**}**

**}**

**}**

**}**

**}**

**}**

**else**

**{**

**int columnSearchBy;**

**switch (comboboxSearchByParts.Text)**

**{**

**case "Название":**

**columnSearchBy = 0;**

**break;**

**case "Код":**

**columnSearchBy = 1;**

**break;**

**case "Производитель":**

**columnSearchBy = 2;**

**break;**

**case "Цена":**

**columnSearchBy = 3;**

**break;**

**default:**

**MessageBox.Show("Ошибка поиска! Выбранное поле отсутствует в таблице. Поиск будет произведен по названию.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**columnSearchBy = 1;**

**break;**

**}**

**for (int i = 0; i < dataGridViewParts.RowCount; i++)**

**{**

**dataGridViewParts.Rows[i].Selected = false;**

**if (dataGridViewParts.Rows[i].Cells[columnSearchBy].Value != null)**

**{**

**if (dataGridViewParts.Rows[i].Cells[columnSearchBy].Value.ToString().ToLower().Contains(textBoxSearchParts.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewParts.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewParts.Rows[i].Cells[columnSearchBy].Selected = true;**

**}**

**}**

**}**

**if (getEntryInRow == false) MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**else if (e.KeyCode == Keys.Enter && textBoxSearchParts.Text == "")**

**{**

**MessageBox.Show("Введите данные для поиска", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**private void buttonRefreshParts\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**this.partsTableAdapter.Fill(this.autoPartsDataSet.parts);**

**foreach (DataGridViewColumn column in dataGridViewParts.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**dataGridViewParts.ClearSelection();**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из таблицы Parts", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**labelRowCountParts.Text = "Количество записей: " + dataGridViewParts.RowCount.ToString();**

**}**

**private void buttonCleanParts\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**if (dataGridViewParts.RowCount > 0)**

**{**

**dataGridViewParts.Sort(dataGridViewParts.Columns[0], ListSortDirection.Ascending);**

**foreach (DataGridViewColumn column in dataGridViewParts.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**foreach (DataGridViewRow row in dataGridViewParts.Rows) row.Visible = true;**

**dataGridViewParts.FirstDisplayedScrollingRowIndex = 0;**

**dataGridViewParts.ClearSelection();**

**labelRowCountParts.Text = "Количество записей: " + dataGridViewParts.RowCount.ToString();**

**}**

**else**

**{**

**MessageBox.Show("Отсутствуют данные, сначала обновите таблицу.", "Ошибка загрузки данных", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**}**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления Parts", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**private void comboBoxSearchByParts\_KeyDown(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter) SearchParts(sender, e);**

**}**

**private void textBoxSearchParts\_Enter(object sender, EventArgs e)**

**{**

**if (textBoxSearchActiveParts == false)**

**{**

**textBoxSearchParts.Text = "";**

**textBoxSearchParts.ForeColor = Color.Black;**

**textBoxSearchActiveParts = true;**

**}**

**}**

**private void textBoxSearchParts\_Leave(object sender, EventArgs e)**

**{**

**if (textBoxSearchParts.Text == "")**

**{**

**textBoxSearchParts.ForeColor = Color.Gray;**

**textBoxSearchParts.Text = "Поиск🔍";**

**textBoxSearchActiveParts = false;**

**}**

**else**

**{**

**textBoxSearchActiveParts = true;**

**}**

**}**

**private void textBoxSearchParts\_KeyDown(object sender, KeyEventArgs e)**

**{**

**SearchParts(sender, e);**

**}**

**private void buttonSearchClearParts\_Click(object sender, EventArgs e)**

**{**

**textBoxSearchParts.ForeColor = Color.Gray;**

**textBoxSearchParts.Text = "Поиск🔍";**

**textBoxSearchActiveParts = false;**

**}**

**private void buttonSelectRowParts\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewParts.SelectedCells;**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewParts.Rows[cell.RowIndex].Selected = true;**

**}**

**}**

**private void buttonAddParts\_Click(object sender, EventArgs e)**

**{**

**AddPartsForm addPartsForm = new AddPartsForm(this);**

**addPartsForm.Show();**

**}**

**private void buttonDeleteParts\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewParts.SelectedCells;**

**if (selectedCells.Count == 0)**

**{**

**MessageBox.Show("Не выбраны строки для удаления", "Ошибка удаления", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**return;**

**}**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewParts.Rows[cell.RowIndex].Selected = true;**

**}**

**if (MessageBox.Show("Выбранные строки удалятся из базы данных. Продолжить? Будет затронута таблица истории цен для данной запчасти", "Удалить запись", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)**

**{**

**try**

**{**

**using (SqlConnection connection = new SqlConnection(connectionString))**

**{**

**connection.Open();**

**foreach (DataGridViewRow selectedRow in dataGridViewParts.SelectedRows)**

**{**

**for (int i = 0; i < dataGridViewStorehouse.Rows.Count; i++)**

**{**

**if (dataGridViewStorehouse.Rows[i].Cells[2].Value == selectedRow.Cells[1]) //deals.код == parts.код**

**{**

**MessageBox.Show("Невозможно у.далить выбранную строку (строки) из таблицы deals, так как от нее зависимы другие таблицы. " +**

**"Сначала удалите связанные строки в них.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**return;**

**}**

**}**

**string expression = "delete from priceHistory where parts\_article = " + selectedRow.Cells[1].Value.ToString();**

**SqlCommand request = new SqlCommand(expression, connection);**

**request.ExecuteNonQuery();**

**expression = "delete from parts where title = '" + selectedRow.Cells[0].Value.ToString() + "'";**

**request = new SqlCommand(expression, connection);**

**request.ExecuteNonQuery();**

**if (dataGridViewParts.SelectedRows.Count <= 1) break;**

**}**

**this.refreshAfterDeleteParts();**

**connection.Close();**

**}**

**} catch (SqlException err)**

**{**

**if (err.Number == 547)**

**{**

**MessageBox.Show("Невозможно удалить выбранную строку (строки) из таблицы deals, так как от нее зависимы другие таблицы. " +**

**"Сначала удалите связанные строки в них.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**if (dataGridViewStorehouse.SelectedRows.Count > 1) return;**

**}**

**else**

**{**

**MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**}**

**}**

**//Providers**

**private void SearchProviders(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter && textBoxSearchProviders.Text != "")**

**{**

**bool getEntryInRow = false;**

**foreach (DataGridViewRow row in dataGridViewProviders.Rows)**

**{**

**row.Visible = true;**

**}**

**dataGridViewProviders.ClearSelection();**

**dataGridViewProviders.CurrentCell = null;**

**if (comboboxSearchByProviders.Text == "Все")**

**{**

**for (int i = 0; i < dataGridViewProviders.RowCount; i++)**

**{**

**dataGridViewProviders.Rows[i].Selected = false;**

**for (int j = 0; j < dataGridViewProviders.ColumnCount; j++)**

**{**

**if (dataGridViewProviders.Rows[i].Cells[j].Value != null)**

**{**

**if (dataGridViewProviders.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchProviders.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewProviders.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewProviders.Rows[i].Selected = true;**

**break;**

**}**

**}**

**}**

**}**

**if (getEntryInRow == false)**

**{**

**MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**return;**

**}**

**else**

**{**

**foreach (DataGridViewRow row in dataGridViewProviders.Rows)**

**{**

**if (!row.Selected) row.Visible = false;**

**}**

**dataGridViewProviders.ClearSelection();**

**for (int i = 0; i < dataGridViewProviders.RowCount; i++)**

**{**

**if (dataGridViewProviders.Rows[i].Visible)**

**{**

**for (int j = 0; j < dataGridViewProviders.ColumnCount; j++)**

**{**

**if (dataGridViewProviders.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchProviders.Text.ToLower()))**

**{**

**dataGridViewProviders.Rows[i].Cells[j].Selected = true;**

**}**

**}**

**}**

**}**

**}**

**}**

**else**

**{**

**int columnSearchBy;**

**switch (comboboxSearchByProviders.Text)**

**{**

**case "Название":**

**columnSearchBy = 0;**

**break;**

**case "Адрес":**

**columnSearchBy = 1;**

**break;**

**case "Телефон":**

**columnSearchBy = 2;**

**break;**

**default:**

**MessageBox.Show("Ошибка поиска! Выбранное поле отсутствует в таблице. Поиск будет произведен по названию.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**columnSearchBy = 1;**

**break;**

**}**

**for (int i = 0; i < dataGridViewProviders.RowCount; i++)**

**{**

**dataGridViewProviders.Rows[i].Selected = false;**

**if (dataGridViewProviders.Rows[i].Cells[columnSearchBy].Value != null)**

**{**

**if (dataGridViewProviders.Rows[i].Cells[columnSearchBy].Value.ToString().ToLower().Contains(textBoxSearchProviders.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewProviders.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewProviders.Rows[i].Cells[columnSearchBy].Selected = true;**

**}**

**}**

**}**

**if (getEntryInRow == false) MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**else if (e.KeyCode == Keys.Enter && textBoxSearchProviders.Text == "")**

**{**

**MessageBox.Show("Введите данные для поиска", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**private void buttonRefreshProviders\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**this.providersTableAdapter.Fill(this.autoPartsDataSet.providers);**

**foreach (DataGridViewColumn column in dataGridViewProviders.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**dataGridViewProviders.ClearSelection();**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из таблицы providers", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**labelRowCountProviders.Text = "Количество записей: " + dataGridViewProviders.RowCount.ToString();**

**}**

**private void buttonCleanProviders\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**if (dataGridViewProviders.RowCount > 0)**

**{**

**dataGridViewProviders.Sort(dataGridViewProviders.Columns[0], ListSortDirection.Ascending);**

**foreach (DataGridViewColumn column in dataGridViewProviders.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**foreach (DataGridViewRow row in dataGridViewProviders.Rows) row.Visible = true;**

**dataGridViewProviders.FirstDisplayedScrollingRowIndex = 0;**

**dataGridViewProviders.ClearSelection();**

**labelRowCountProviders.Text = "Количество записей: " + dataGridViewProviders.RowCount.ToString();**

**}**

**else**

**{**

**MessageBox.Show("Отсутствуют данные, сначала обновите таблицу.", "Ошибка загрузки данных", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**}**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления Providers", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**private void comboBoxSearchByProviders\_KeyDown(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter) SearchProviders(sender, e);**

**}**

**private void textBoxSearchProviders\_Enter(object sender, EventArgs e)**

**{**

**if (textBoxSearchActiveProviders == false)**

**{**

**textBoxSearchProviders.Text = "";**

**textBoxSearchProviders.ForeColor = Color.Black;**

**textBoxSearchActiveProviders = true;**

**}**

**}**

**private void textBoxSearchProviders\_Leave(object sender, EventArgs e)**

**{**

**if (textBoxSearchProviders.Text == "")**

**{**

**textBoxSearchProviders.ForeColor = Color.Gray;**

**textBoxSearchProviders.Text = "Поиск🔍";**

**textBoxSearchActiveProviders = false;**

**}**

**else**

**{**

**textBoxSearchActiveProviders = true;**

**}**

**}**

**private void textBoxSearchProviders\_KeyDown(object sender, KeyEventArgs e)**

**{**

**SearchProviders(sender, e);**

**}**

**private void buttonSearchCleanProviders\_Click(object sender, EventArgs e)**

**{**

**textBoxSearchProviders.ForeColor = Color.Gray;**

**textBoxSearchProviders.Text = "Поиск🔍";**

**textBoxSearchActiveProviders = false;**

**}**

**private void buttonSelectRowProviders\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewProviders.SelectedCells;**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewProviders.Rows[cell.RowIndex].Selected = true;**

**}**

**}**

**private void buttonAddProviders\_Click(object sender, EventArgs e)**

**{**

**AddProvidersForm addProvidersForm = new AddProvidersForm(this);**

**addProvidersForm.Show();**

**}**

**private void buttonDeleteProviders\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewProviders.SelectedCells;**

**if (selectedCells.Count == 0)**

**{**

**MessageBox.Show("Не выбраны строки для удаления", "Ошибка удаления", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**return;**

**}**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewProviders.Rows[cell.RowIndex].Selected = true;**

**}**

**if (MessageBox.Show("Выбранные строки удалятся из базы данных. Продолжить? Будет затронута таблица запчастей. " +**

**"Если с ней связан выбранный поставщик (поставщики), удаление будет отменено.", "Удалить запись", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)**

**{**

**try**

**{**

**using (SqlConnection connection = new SqlConnection(connectionString))**

**{**

**connection.Open();**

**foreach (DataGridViewRow selectedRow in dataGridViewProviders.SelectedRows)**

**{**

**for (int i = 0; i < autoPartsDataSet.bridge\_providers\_parts.Rows.Count; i++)**

**{**

**if (autoPartsDataSet.bridge\_providers\_parts.Rows[i].ItemArray[0].ToString() == selectedRow.Cells[0].ToString()) //bridge\_priveders\_parts.providers\_title == parts.title**

**{**

**MessageBox.Show("Невозможно удалить выбранную строку (строки) из таблицы providers, так как от нее зависимы другие таблицы. " +**

**"Сначала удалите связанные строки в них.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**return;**

**}**

**}**

**string expression = "delete from providers where title = '" + selectedRow.Cells[0].Value.ToString() + "';";**

**SqlCommand request = new SqlCommand(expression, connection);**

**request.ExecuteNonQuery();**

**if (dataGridViewParts.SelectedRows.Count <= 1) break;**

**}**

**this.refreshAfterDeleteProviders();**

**connection.Close();**

**}**

**}**

**catch (SqlException err)**

**{**

**if (err.Number == 547)**

**{**

**MessageBox.Show("Невозможно удалить выбранную строку (строки) из таблицы deals, так как от нее зависимы другие таблицы. " +**

**"Сначала удалите связанные строки в них.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**if (dataGridViewStorehouse.SelectedRows.Count > 1) return;**

**}**

**else**

**{**

**MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**}**

**}**

**//priceview**

**private void SearchPriceview(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter && textBoxSearchPriceview.Text != "")**

**{**

**bool getEntryInRow = false;**

**foreach (DataGridViewRow row in dataGridViewPriceview.Rows)**

**{**

**row.Visible = true;**

**}**

**dataGridViewPriceview.ClearSelection();**

**dataGridViewPriceview.CurrentCell = null;**

**if (comboboxSearchByPriceview.Text == "Все")**

**{**

**for (int i = 0; i < dataGridViewPriceview.RowCount; i++)**

**{**

**dataGridViewPriceview.Rows[i].Selected = false;**

**for (int j = 0; j < dataGridViewPriceview.ColumnCount; j++)**

**{**

**if (dataGridViewPriceview.Rows[i].Cells[j].Value != null)**

**{**

**if (dataGridViewPriceview.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchPriceview.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewPriceview.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewPriceview.Rows[i].Selected = true;**

**break;**

**}**

**}**

**}**

**}**

**if (getEntryInRow == false)**

**{**

**MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**return;**

**}**

**else**

**{**

**foreach (DataGridViewRow row in dataGridViewPriceview.Rows)**

**{**

**if (!row.Selected) row.Visible = false;**

**}**

**dataGridViewPriceview.ClearSelection();**

**for (int i = 0; i < dataGridViewPriceview.RowCount; i++)**

**{**

**if (dataGridViewPriceview.Rows[i].Visible)**

**{**

**for (int j = 0; j < dataGridViewPriceview.ColumnCount; j++)**

**{**

**if (dataGridViewPriceview.Rows[i].Cells[j].Value.ToString().ToLower().Contains(textBoxSearchPriceview.Text.ToLower()))**

**{**

**dataGridViewPriceview.Rows[i].Cells[j].Selected = true;**

**}**

**}**

**}**

**}**

**}**

**}**

**else**

**{**

**int columnSearchBy;**

**switch (comboboxSearchByPriceview.Text)**

**{**

**case "id":**

**columnSearchBy = 0;**

**break;**

**case "Код детали":**

**columnSearchBy = 1;**

**break;**

**case "Название":**

**columnSearchBy = 2;**

**break;**

**case "Текущая цена":**

**columnSearchBy = 3;**

**break;**

**case "Старая цена":**

**columnSearchBy = 4;**

**break;**

**case "Действовала до":**

**columnSearchBy = 5;**

**break;**

**default:**

**MessageBox.Show("Ошибка поиска! Выбранное поле отсутствует в представлении. Поиск будет произведен по названию.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**columnSearchBy = 1;**

**break;**

**}**

**for (int i = 0; i < dataGridViewPriceview.RowCount; i++)**

**{**

**dataGridViewPriceview.Rows[i].Selected = false;**

**if (dataGridViewPriceview.Rows[i].Cells[columnSearchBy].Value != null)**

**{**

**if (dataGridViewPriceview.Rows[i].Cells[columnSearchBy].Value.ToString().ToLower().Contains(textBoxSearchPriceview.Text.ToLower()))**

**{**

**if (getEntryInRow == false)**

**{**

**getEntryInRow = true;**

**dataGridViewPriceview.FirstDisplayedScrollingRowIndex = i;**

**}**

**dataGridViewPriceview.Rows[i].Cells[columnSearchBy].Selected = true;**

**}**

**}**

**}**

**if (getEntryInRow == false) MessageBox.Show("Ничего не найдено, попробуйте изменить критерии поиска.", "Поиск", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**else if (e.KeyCode == Keys.Enter && textBoxSearchPriceview.Text == "")**

**{**

**MessageBox.Show("Введите данные для поиска", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Information);**

**}**

**}**

**private void buttonRefreshPriceview\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**this.priceviewTableAdapter.Fill(this.autoPartsDataSet.priceview);**

**foreach (DataGridViewColumn column in dataGridViewPriceview.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**dataGridViewPriceview.ClearSelection();**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления priceview", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**labelRowCountPriceview.Text = "Количество записей: " + dataGridViewPriceview.RowCount.ToString();**

**}**

**private void buttonCleanPriceview\_Click(object sender, EventArgs e)**

**{**

**try**

**{**

**if (dataGridViewPriceview.RowCount > 0)**

**{**

**dataGridViewPriceview.Sort(dataGridViewPriceview.Columns[0], ListSortDirection.Ascending);**

**foreach (DataGridViewColumn column in dataGridViewPriceview.Columns)**

**{**

**column.SortMode = DataGridViewColumnSortMode.NotSortable;**

**column.SortMode = DataGridViewColumnSortMode.Automatic;**

**}**

**foreach (DataGridViewRow row in dataGridViewPriceview.Rows) row.Visible = true;**

**dataGridViewPriceview.FirstDisplayedScrollingRowIndex = 0;**

**dataGridViewPriceview.ClearSelection();**

**labelRowCountPriceview.Text = "Количество записей: " + dataGridViewPriceview.RowCount.ToString();**

**}**

**else**

**{**

**MessageBox.Show("Отсутствуют данные, сначала обновите таблицу.", "Ошибка загрузки данных", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**}**

**}**

**catch (Exception err)**

**{**

**MessageBox.Show(err.ToString(), "Ошибка загрузки данных из представления Priceview", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**private void comboBoxSearchByPriceview\_KeyDown(object sender, KeyEventArgs e)**

**{**

**if (e.KeyCode == Keys.Enter) SearchPriceview(sender, e);**

**}**

**private void textBoxSearchPriceview\_Enter(object sender, EventArgs e)**

**{**

**if (textBoxSearchActivePriceview == false)**

**{**

**textBoxSearchPriceview.Text = "";**

**textBoxSearchPriceview.ForeColor = Color.Black;**

**textBoxSearchActivePriceview = true;**

**}**

**}**

**private void textBoxSearchPriceview\_Leave(object sender, EventArgs e)**

**{**

**if (textBoxSearchPriceview.Text == "")**

**{**

**textBoxSearchPriceview.ForeColor = Color.Gray;**

**textBoxSearchPriceview.Text = "Поиск🔍";**

**textBoxSearchActivePriceview = false;**

**}**

**else**

**{**

**textBoxSearchActivePriceview = true;**

**}**

**}**

**private void textBoxSearchPriceview\_KeyDown(object sender, KeyEventArgs e)**

**{**

**SearchPriceview(sender, e);**

**}**

**private void buttonSearchCleanPriceview\_Click(object sender, EventArgs e)**

**{**

**textBoxSearchPriceview.ForeColor = Color.Gray;**

**textBoxSearchPriceview.Text = "Поиск🔍";**

**textBoxSearchActivePriceview = false;**

**}**

**private void buttonSelectRowPriceview\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewPriceview.SelectedCells;**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewPriceview.Rows[cell.RowIndex].Selected = true;**

**}**

**}**

**private void buttonAddPriceview\_Click(object sender, EventArgs e)**

**{**

**if (dataGridViewPriceview.SelectedRows.Count > 1)**

**{**

**MessageBox.Show("Выберите только одну строку для изменения цены", "Ошибка выбора", MessageBoxButtons.OK, MessageBoxIcon.Warning);**

**return;**

**}**

**else**

**{**

**var selectedCells = dataGridViewPriceview.SelectedCells;**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewPriceview.Rows[cell.RowIndex].Selected = true;**

**}**

**}**

**AddPriceHistoryForm addPriceHistoryForm = new AddPriceHistoryForm(this);**

**addPriceHistoryForm.textBoxTitle.Text = dataGridViewPriceview.SelectedCells[2].Value.ToString();**

**if (dataGridViewPriceview.SelectedCells[3].ToString() != "")**

**addPriceHistoryForm.textBoxOldPrice.Text = dataGridViewPriceview.SelectedCells[3].Value.ToString();**

**foreach (DataRow row in this.autoPartsDataSet.priceview.Select())**

**{**

**addPriceHistoryForm.textBoxTitle.AutoCompleteCustomSource.Add(**

**row.ItemArray[2].ToString());**

**}**

**addPriceHistoryForm.Show();**

**addPriceHistoryForm.partsAutocompleteСhanged();**

**addPriceHistoryForm.isAllowedRecordRow();**

**}**

**private void buttonDeletePriceview\_Click(object sender, EventArgs e)**

**{**

**var selectedCells = dataGridViewPriceview.SelectedCells;**

**if (selectedCells.Count == 0)**

**{**

**MessageBox.Show("Не выбраны строки для удаления", "Ошибка удаления", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**return;**

**}**

**foreach (DataGridViewCell cell in selectedCells)**

**{**

**dataGridViewPriceview.Rows[cell.RowIndex].Selected = true;**

**}**

**if (MessageBox.Show("Выбранные строки удалятся из базы данных. Продолжить?", "Удалить запись", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)**

**{**

**try**

**{**

**using (SqlConnection connection = new SqlConnection(connectionString))**

**{**

**connection.Open();**

**foreach (DataGridViewRow selectedRow in dataGridViewPriceview.SelectedRows)**

**{**

**if (selectedRow.Cells[0].Value.ToString() == "")**

**{**

**MessageBox.Show("Для данной запчасти отсутствуют записи в таблице priceHistory, нечего удалять.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**return;**

**}**

**string expression = "delete from priceHistory where price\_id = " + selectedRow.Cells[0].Value.ToString() + ";";**

**SqlCommand request = new SqlCommand(expression, connection);**

**request.ExecuteNonQuery();**

**if (dataGridViewPriceview.SelectedRows.Count <= 1) break;**

**}**

**this.refreshAfterDeletePriceview();**

**connection.Close();**

**}**

**}**

**catch (SqlException err)**

**{**

**if (err.Number == 547)**

**{**

**MessageBox.Show("Невозможно удалить выбранную строку (строки) из представления priceview, так как от нее зависимы другие таблицы. " +**

**"Сначала удалите связанные строки в них.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**if (dataGridViewStorehouse.SelectedRows.Count > 1) return;**

**}**

**else**

**{**

**MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);**

**}**

**}**

**}**

**}**

**//additional functions**

**private void buttonRequest\_Click(object sender, EventArgs e)**

**{**

**var addRequest = new AddRequest(this);**

**addRequest.Show();**

**}**

**private void buttonToProvidersStorehouse\_Click(object sender, EventArgs e)**

**{**

**tabs.SelectTab(providersTab);**

**comboboxSearchByProviders.Text = "Название";**

**textBoxSearchProviders.Focus();**

**textBoxSearchProviders\_Enter(sender, e);**

**textBoxSearchProviders.Text = dataGridViewStorehouse.CurrentRow.Cells[8].Value.ToString();**

**SendKeys.Send("{ENTER}");**

**textBoxSearchProviders\_Leave(sender, e);**

**}**

**private void buttonPricehistoryStorehouse\_Click(object sender, EventArgs e)**

**{**

**tabs.SelectTab(pricehistoryTab);**

**comboboxSearchByPriceview.Text = "Все";**

**textBoxSearchPriceview.Focus();**

**textBoxSearchPriceview\_Enter(sender, e);**

**textBoxSearchPriceview.Text = dataGridViewStorehouse.CurrentRow.Cells[1].Value.ToString();**

**SendKeys.Send("{ENTER}");**

**textBoxSearchPriceview\_Leave(sender, e);**

**}**

**}**

**}**

AddDealsForm.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DB

{

public partial class AddDealsForm : Form

{

public const string connectionString = "Data Source=ORANGE\\MSSQLEXPRESS2017;Initial Catalog=AutoParts;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False";

public MainForm parentForm;

private static Dictionary<string, int> bridge\_id = new Dictionary<string, int> ();

public AddDealsForm(MainForm mainForm)

{

parentForm = mainForm;

InitializeComponent();

dateTimePicker.Value = System.DateTime.Now;

}

private void buttonSetCurrentDate\_Click(object sender, EventArgs e)

{

dateTimePicker.Value = System.DateTime.Now;

}

private void buttonCloseForm\_Click(object sender, EventArgs e)

{

this.Close();

}

private void AddForm\_Load(object sender, EventArgs e)

{

this.providersTableAdapter.Fill(this.autoPartsDataSet.providers);

this.dealsTableAdapter.Fill(this.autoPartsDataSet.deals);

this.bridge\_providers\_partsTableAdapter.Fill(this.autoPartsDataSet.bridge\_providers\_parts);

this.partsTableAdapter.Fill(this.autoPartsDataSet.parts);

}

public bool isAllowedRecordRow()

{

if (pictureTitleError.Visible == false && picturePartsCountError.Visible == false && pictureProviderError.Visible == false)

{

buttonRecordRow.Enabled = true;

return true;

}

else

{

buttonRecordRow.Enabled = false;

return false;

}

}

private void partsAutocomplete (object sender, EventArgs e)

{

if (textBoxTitle.Text == "")

{

pictureTitleError.Show();

pictureProviderError.Show();

return;

}

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand request;

request = new SqlCommand(

"select \* from parts where title = '" + textBoxTitle.Text + "';",

connection);

request.ExecuteNonQuery();

int partArticle = -1;

bridge\_id.Clear();

using (SqlDataReader reader = request.ExecuteReader())

{

if (reader.HasRows)

{

pictureTitleError.Hide();

while (reader.Read())

{

partArticle = reader.GetInt32(1);

textBoxManufacturer.Text = reader.GetString(2);

textBoxPrice.Text = reader.GetInt32(3).ToString();

}

}

else

{

pictureTitleError.Show();

comboBoxProvider.Text = "";

comboBoxProvider.Items.Clear();

pictureProviderError.Show();

textBoxManufacturer.Text = "";

textBoxPrice.Text = "";

}

}

if (partArticle != -1)

{

request = new SqlCommand(

"select \* from bridge\_providers\_parts where parts\_article = " + partArticle.ToString() + ";",

connection);

request.ExecuteNonQuery();

using (SqlDataReader reader = request.ExecuteReader())

{

if (reader.HasRows)

{

comboBoxProvider.Items.Clear();

while (reader.Read())

{

comboBoxProvider.Items.Add(reader.GetString(1));

bridge\_id.Add(reader.GetString(1), reader.GetInt32(0));

}

}

else

{

pictureProviderError.Show();

}

}

}

connection.Close();

isAllowedRecordRow();

}

}

private void buttonRecordRow\_Click(object sender, EventArgs e)

{

try

{

if (isAllowedRecordRow() && bridge\_id.Count > 0)

{

string expression = "insert into deals (deal\_date, bridge\_id, parts\_count) values ('" + dateTimePicker.Text.ToString() + "', " + bridge\_id[comboBoxProvider.Text] + ", " + textBoxCount.Text + ");";

//if (MessageBox.Show(expression, "SQL выражение", MessageBoxButtons.OKCancel, MessageBoxIcon.Information) == DialogResult.OK)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand request = new SqlCommand(expression, connection);

request.ExecuteNonQuery();

connection.Close();

}

}

}

else

{

MessageBox.Show("Не все ограничения удовлетворяют условиям добавления записи.", "Ошибка записи данных", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

} catch (Exception err)

{

MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

parentForm.refreshAfterInsertStorehouse();

MessageBox.Show("Запись успешно добавлена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void textBoxTitle\_Leave(object sender, EventArgs e)

{

partsAutocomplete(sender, e);

}

private void textBoxTitle\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

partsAutocomplete(sender, e);

}

}

private void textBoxCount\_TextChanged(object sender, EventArgs e)

{

foreach (char ch in textBoxCount.Text)

{

if (char.IsDigit(ch)) picturePartsCountError.Hide();

else

{

picturePartsCountError.Show();

break;

};

}

isAllowedRecordRow();

}

private void textBoxCount\_Leave(object sender, EventArgs e)

{

if (textBoxCount.Text != "")

{

try

{

int partsCount = Int32.Parse(textBoxCount.Text);

if (partsCount <= 0) picturePartsCountError.Show(); else picturePartsCountError.Hide();

isAllowedRecordRow();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

picturePartsCountError.Show();

isAllowedRecordRow();

}

}

}

private void comboBoxProvider\_TextChanged(object sender, EventArgs e)

{

if (comboBoxProvider.Text == "")

{

pictureProviderError.Show();

}

else pictureProviderError.Hide();

isAllowedRecordRow();

}

private void buttonAddPart\_Click(object sender, EventArgs e)

{

AddPartsForm addPartsForm = new AddPartsForm(this);

addPartsForm.Show();

}

private void buttonAddProvider\_Click(object sender, EventArgs e)

{

AddProvidersForm addProvidersForm = new AddProvidersForm(this);

addProvidersForm.Show();

}

}

}

AddPriceHistoryForm.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DB

{

public partial class AddPriceHistoryForm : Form

{

private const string connectionString = "Data Source=ORANGE\\MSSQLEXPRESS2017;Initial Catalog=AutoParts;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False";

private MainForm parentForm;

private int selectedPartArticle;

public AddPriceHistoryForm(MainForm mainForm)

{

parentForm = mainForm;

InitializeComponent();

dateTimePicker.Value = System.DateTime.Now;

partsAutocompleteСhanged();

}

private void buttonSetCurrentDate\_Click(object sender, EventArgs e)

{

dateTimePicker.Value = System.DateTime.Now;

isAllowedRecordRow();

}

private void buttonCloseForm\_Click(object sender, EventArgs e)

{

this.Close();

}

public bool isAllowedRecordRow()

{

if (pictureTitleError.Visible == false && pictureNewPriceError.Visible == false && pictureOldPriceError.Visible == false)

{

buttonRecordRow.Enabled = true;

return true;

}

else

{

buttonRecordRow.Enabled = false;

return false;

}

}

public void partsAutocompleteСhanged()

{

if (textBoxTitle.Text == "")

{

pictureTitleError.Show();

pictureOldPriceError.Show();

return;

}

foreach (DataRow row in parentForm.autoPartsDataSet.priceview.Rows)

{

if (row.ItemArray[2].ToString().ToLower().Contains(textBoxTitle.Text.ToLower()))

{

textBoxOldPrice.Text = row.ItemArray[3].ToString();

pictureTitleError.Hide();

pictureOldPriceError.Hide();

selectedPartArticle = int.Parse(row.ItemArray[1].ToString());

return;

}

}

MessageBox.Show("Не найдена цена для данной запчасти", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

pictureTitleError.Show();

pictureOldPriceError.Show();

isAllowedRecordRow();

}

private void textBoxTitle\_Leave(object sender, EventArgs e)

{

partsAutocompleteСhanged();

}

private void textBoxTitle\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

partsAutocompleteСhanged();

}

}

private void textBoxNewPrice\_TextChanged(object sender, EventArgs e)

{

foreach (char ch in textBoxNewPrice.Text)

{

if (char.IsDigit(ch)) pictureNewPriceError.Hide();

else

{

pictureNewPriceError.Show();

break;

};

}

isAllowedRecordRow();

}

private void textBoxNewPrice\_Leave(object sender, EventArgs e)

{

if (textBoxNewPrice.Text != "")

{

try

{

int partsCount = Int32.Parse(textBoxNewPrice.Text);

if (partsCount <= 0) pictureNewPriceError.Show(); else pictureNewPriceError.Hide();

isAllowedRecordRow();

}

catch (Exception err)

{

MessageBox.Show(err.Message, "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

pictureNewPriceError.Show();

isAllowedRecordRow();

}

}

}

private void buttonRecordRow\_Click(object sender, EventArgs e)

{

if (textBoxNewPrice.Text == textBoxOldPrice.Text)

{

MessageBox.Show("Текущая цена и введенная вами совпадают, запись отменена.", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

try

{

if (isAllowedRecordRow())

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand request = new SqlCommand("update parts set price = " + textBoxNewPrice.Text + " where article = " + selectedPartArticle.ToString(), connection);

request.ExecuteNonQuery();

connection.Close();

}

}

else

{

MessageBox.Show("Не все ограничения удовлетворяют условиям добавления записи.", "Ошибка записи данных", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (Exception err)

{

MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

parentForm.refreshAfterDeletePriceview();

MessageBox.Show("Запись успешно добавлена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

partsAutocompleteСhanged();

}

}

}

AddRequest.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DB

{

public partial class AddRequest : Form

{

private const string connectionString = "Data Source=ORANGE\\MSSQLEXPRESS2017;Initial Catalog=AutoParts;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False";

private MainForm parentForm;

static DataSet ds;

public AddRequest(MainForm mainForm)

{

parentForm = mainForm;

InitializeComponent();

//var sqlDataAdapter = new SqlDataAdapter("select \* from parts", connectionString);

//dataGridViewRequest.DataSource = parentForm.autoPartsDataSet.parts;

}

private void buttonClose\_Click(object sender, EventArgs e)

{

this.Close();

}

private void buttonSend\_Click(object sender, EventArgs e)

{

try

{

if (textBoxRequest.Text.ToLower().Contains("select"))

{

using (var sqlDataAdapter = new SqlDataAdapter(textBoxRequest.Text, connectionString))

{

ds = new DataSet();

sqlDataAdapter.Fill(ds);

dataGridViewRequest.DataSource = ds.Tables[0];

ds.Dispose();

}

}

else

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand request = new SqlCommand(textBoxRequest.Text, connection);

request.ExecuteNonQuery();

connection.Close();

}

}

}

catch (Exception err)

{

MessageBox.Show("SQL запрос отклонен: ", err.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

MessageBox.Show("SQL запрос успешно выполнен", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

}

AddPartsForm.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DB

{

public partial class AddPartsForm : Form

{

public const string connectionString = "Data Source=ORANGE\\MSSQLEXPRESS2017;Initial Catalog=AutoParts;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False";

MainForm parentMainForm;

AddDealsForm parentAddDealsForm;

public AddPartsForm(MainForm mainForm)

{

parentMainForm = mainForm;

InitializeComponent();

}

public AddPartsForm(AddDealsForm addDealsForm)

{

parentAddDealsForm = addDealsForm;

InitializeComponent();

}

public bool isAllowedRecordRow()

{

if (pictureTitleError.Visible == false && pictureManufacturerError.Visible == false && picturePriceError.Visible == false)

{

buttonRecordRow.Enabled = true;

return true;

}

else

{

buttonRecordRow.Enabled = false;

return false;

}

}

private void KeyDownEnterSelectNextControl(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

e.Handled = e.SuppressKeyPress = true;

SendKeys.Send("{tab}");

}

}

private void textBoxTitle\_TextChanged\_Leave(object sender, EventArgs e)

{

if (textBoxTitle.Text == "") pictureTitleError.Show();

else pictureTitleError.Hide();

}

private void textBoxManufacturer\_TextChanged\_Leave(object sender, EventArgs e)

{

if (textBoxManufacturer.Text == "") pictureManufacturerError.Show();

else pictureManufacturerError.Hide();

}

private void textBoxPrice\_TextChanged(object sender, EventArgs e)

{

if (textBoxPrice.Text != "")

{

foreach (char ch in textBoxPrice.Text)

{

if (char.IsDigit(ch)) picturePriceError.Hide();

else

{

picturePriceError.Show();

break;

};

}

}

else picturePriceError.Show();

isAllowedRecordRow();

}

private void textBoxPrice\_Leave(object sender, EventArgs e)

{

if (textBoxPrice.Text != "")

picturePriceError.Hide();

else

{

picturePriceError.Show();

isAllowedRecordRow();

return;

}

try

{

int price = Int32.Parse(textBoxPrice.Text);

if (price <= 0) picturePriceError.Show(); else picturePriceError.Hide();

isAllowedRecordRow();

}

catch (Exception err)

{

MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

picturePriceError.Show();

isAllowedRecordRow();

}

}

private void buttonRecordRow\_Click(object sender, EventArgs e)

{

if (isAllowedRecordRow())

{

string expression = "insert into parts(title, manufacturer, price) values ('" + textBoxTitle.Text + "', '" + textBoxManufacturer.Text + "', " + textBoxPrice.Text + ");";

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand request = new SqlCommand(expression, connection);

request.ExecuteNonQuery();

connection.Close();

}

} catch (Exception err)

{

MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

MessageBox.Show("Запись успешно добавлена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

if (parentMainForm != null) parentMainForm.refreshAfterInsertParts();

else if (parentAddDealsForm != null)

{

parentAddDealsForm.parentForm.refreshAfterInsertParts(parentAddDealsForm);

parentAddDealsForm.Activate();

this.Close();

}

}

}

private void buttonClose\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

AddProvidersForm.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DB

{

public partial class AddProvidersForm : Form

{

public const string connectionString = "Data Source=ORANGE\\MSSQLEXPRESS2017;Initial Catalog=AutoParts;Integrated Security=True;Connect Timeout=30;Encrypt=False;TrustServerCertificate=False";

MainForm parentMainForm;

AddDealsForm parentAddDealsForm;

public AddProvidersForm(MainForm mainForm)

{

parentMainForm = mainForm;

InitializeComponent();

}

public AddProvidersForm(AddDealsForm addDealsForm)

{

parentAddDealsForm = addDealsForm;

InitializeComponent();

}

public bool isAllowedRecordRow()

{

if (pictureTitleError.Visible == false && pictureAddressError.Visible == false && picturePhoneError.Visible == false)

{

buttonRecordRow.Enabled = true;

return true;

}

else

{

buttonRecordRow.Enabled = false;

return false;

}

}

private void KeyDownEnterSelectNextControl(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

e.Handled = e.SuppressKeyPress = true;

SendKeys.Send("{tab}");

}

}

private void textBoxTitle\_TextChanged\_Leave(object sender, EventArgs e)

{

if (textBoxTitle.Text == "") pictureTitleError.Show();

else pictureTitleError.Hide();

isAllowedRecordRow();

}

private void textBoxAddress\_TextChanged\_Leave(object sender, EventArgs e)

{

if (textBoxAddress.Text == "") pictureAddressError.Show();

else pictureAddressError.Hide();

isAllowedRecordRow();

}

private void textBoxPhone\_TextChanged\_Leave(object sender, EventArgs e)

{

if (textBoxPhone.Text == "")

{

picturePhoneError.Hide();

return;

}

else

{

for (int i = 0; i < textBoxPhone.Text.Length; i++)

{

if (char.IsDigit(textBoxPhone.Text[i]))

{

continue;

}

else if (i == 0 && textBoxPhone.Text[i] == '+')

{

continue;

}

else

{

picturePhoneError.Show();

return;

}

}

picturePhoneError.Hide();

isAllowedRecordRow();

}

}

private void buttonRecordRow\_Click(object sender, EventArgs e)

{

if (isAllowedRecordRow())

{

string expression;

if (textBoxPhone.Text != "")

expression = "insert into providers(title, provider\_address, phone) values ('" + textBoxTitle.Text + "', '" + textBoxAddress.Text + "', '" + textBoxPhone.Text + "');";

else

expression = "insert into providers(title, provider\_address) values ('" + textBoxTitle.Text + "', '" + textBoxAddress.Text + "');";

try

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand request = new SqlCommand(expression, connection);

request.ExecuteNonQuery();

connection.Close();

}

}

catch (Exception err)

{

MessageBox.Show(err.ToString(), "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

MessageBox.Show("Запись успешно добавлена", "Уведомление", MessageBoxButtons.OK, MessageBoxIcon.Information);

if (parentMainForm != null) parentMainForm.refreshAfterInsertProviders();

else if (parentAddDealsForm != null)

{

parentAddDealsForm.parentForm.refreshAfterInsertProviders(parentAddDealsForm);

parentAddDealsForm.Activate();

this.Close();

}

}

}

private void buttonClose\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace DB

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new MainForm());

}

}

}