Отчёт по практической работе № 13 студента Телепова Д. А. группы ПКС-306

Тема: “Работа с представлениями ”

Цель: Научиться создавать простые представления

Выполнение работы

Эквиваленты запросов на SQL языке

1. Посчитать количество занятых штатных единиц по каждой должности каждого подразделения  
   SELECT кодОтдела, кодДолжности, COUNT(номер) AS [Кол-во занятых]

FROM dbo.Назачения

WHERE (датаУвольнения IS NULL)

GROUP BY кодОтдела, кодДолжности

1. Посчитать количество вакансий по каждому подразделению в виде: подразделение должность штатных единиц, из них занято, вакансий  
     
   SELECT dbo.Отделы.названиеОтдела, dbo.Должности.названиеДолжности, dbo.Штат.[Кол-во], dbo.[Кол-во занятых мест].[Кол-во занятых], dbo.Штат.кодОтдела, dbo.Штат.кодДолжности

FROM dbo.Должности INNER JOIN

dbo.Штат ON dbo.Должности.кодДолжности = dbo.Штат.кодДолжности INNER JOIN

dbo.Отделы ON dbo.Штат.кодОтдела = dbo.Отделы.кодОтдела LEFT OUTER JOIN

dbo.[Кол-во занятых мест] ON dbo.Штат.кодОтдела = dbo.[Кол-во занятых мест].кодОтдела AND dbo.Штат.кодДолжности = dbo.[Кол-во занятых мест].кодДолжности

GROUP BY dbo.Отделы.названиеОтдела, dbo.Должности.названиеДолжности, dbo.Штат.[Кол-во], dbo.Штат.кодОтдела, dbo.Штат.кодДолжности, dbo.[Кол-во занятых мест].[Кол-во занятых]  
  
потом меняем NULL на 0 и получаем готовую  
SELECT кодОтдела, названиеОтдела, кодДолжности, названиеДолжности, [Кол-во],

CASE

WHEN [Кол-во занятых] IS NULL

THEN 0

ELSE [Кол-во занятых]

END AS Занято, [Кол-во] - CASE WHEN [Кол-во занятых] IS NULL

THEN 0 ELSE [Кол-во занятых] END AS Вакансии

FROM dbo.[Всего, Занято (неполное)]

Код программы

Код главного окна (MainWindow)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Data.SqlClient;

using System.Data;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

List<string> SexList = new List<string>();

public MainWindow()

{

InitializeComponent();

SexList.Add("M");

SexList.Add("Ж");

SexCB.ItemsSource = SexList;

}

string ConString = @"Data Source=DESKTOP-ITGU7A9\SQLEXPRESS; Initial Catalog=РаботникиNEW; Integrated Security = True;";

SqlConnection connection = new SqlConnection();

List<string> TableNamesList = new List<string>();

List<string> DepartmensList = new List<string>();

List<string> PositionsList = new List<string>();

int tabCount = 0;

List<int> DepartmensCode = new List<int>();

List<int> PositionsCode = new List<int>();

int? DepartmentCode;

int? PositionCode;

List<int> newMemberNum = new List<int>();

int? memberNum;

string Bithday;

void Connecting()

{

connection.ConnectionString = ConString;

try

{

connection.Open();

MessageBox.Show("Подключение к Source DESKTOP-ITGU7A9\\SQLEXPRESS: УСПЕШНО \n" + "Подключение к бд: РаботникиNEW: УСПЕШНО\n", "Подключено", MessageBoxButton.OK);

}

catch (SqlException)

{

MessageBox.Show("Не удалось подключиться к Source: DESKTOP-ITGU7A9\\SQLEXPRESS к бд: РаботникиNEW","Ошибка",MessageBoxButton.OK);

}

}

void AddTabitem()

{

InputTableName tableNameWindow = new InputTableName();

tableNameWindow.Owner = this;

if (tableNameWindow.ShowDialog()==true)

{

if (tableNameWindow.TableName == null)

{

MessageBox.Show("Таблица не была выбрана", "Внимание!", MessageBoxButton.OK);

}

else

{

using (connection)

{

try

{

ConString = @"Data Source=DESKTOP-ITGU7A9\SQLEXPRESS; Initial Catalog=РаботникиNEW; Integrated Security = True;";

tabCount++;

DataGrid dataGrid = new DataGrid();

string CMDstring = $"SELECT \* FROM [{tableNameWindow.TableName}]";

SqlCommand command = new SqlCommand(CMDstring,connection);

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable dt = new DataTable();

adapter.Fill(dt);

dataGrid.ItemsSource = dt.DefaultView;

TabsCntrl.Items.Add(new TabItem

{

Name = $"tabitem{tabCount.ToString()}",

Content = dataGrid,

Header = new TextBlock { Text = tableNameWindow.TableName + $" ({tabCount})" },

TabIndex = tabCount

}) ;

TableNamesList.Add(tableNameWindow.TableName);

tableNameWindow.TableName = null;

}

catch (SqlException)

{

MessageBox.Show("Была указана несуществующая таблица, или её имя было указано неверно", "Не удалось получить тиблицу",MessageBoxButton.OK);

}

}

connection.ConnectionString = ConString;

}

}

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

grid1.Visibility = Visibility.Hidden;

grid2.Visibility = Visibility.Visible;

Connecting();

string CMDstring = $"Select DISTINCT названиеОтдела FROM Вакансии WHERE dbo.Вакансии.Вакансии <> 0";

SqlCommand command = new SqlCommand(CMDstring, connection);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

DepartmensList.Add(reader[0].ToString());

}

reader.Close();

DepartmentCB.ItemsSource = DepartmensList;

}

private void AddTable\_Click(object sender, RoutedEventArgs e)

{

AddTabitem();

}

private void RemoveTables\_Click(object sender, RoutedEventArgs e)

{

TabsCntrl.Items.Clear();

tabCount=0;

TableNamesList.Clear();

}

private void UpdateTable\_Click(object sender, RoutedEventArgs e)

{

TabsCntrl.Items.Clear();

tabCount = 0;

foreach (var tableNamesInList in TableNamesList)

{

ConString = @"Data Source=DESKTOP-ITGU7A9\SQLEXPRESS; Initial Catalog=РаботникиNEW; Integrated Security = True;";

connection.ConnectionString = ConString;

tabCount++;

DataGrid dataGrid = new DataGrid();

string CMDstring = $"SELECT \* FROM [{tableNamesInList}]";

SqlCommand command = new SqlCommand(CMDstring, connection);

SqlDataAdapter adapter = new SqlDataAdapter(command);

DataTable dt = new DataTable();

adapter.Fill(dt);

dataGrid.ItemsSource = dt.DefaultView;

TabsCntrl.Items.Add(new TabItem

{

Name = $"tabitem{tabCount.ToString()}",

Content = dataGrid,

Header = new TextBlock { Text = tableNamesInList + $" ({tabCount})" },

TabIndex = tabCount

});

connection.ConnectionString = ConString;

}

}

private void AddMember\_Click(object sender, RoutedEventArgs e)

{

try

{

//Добавляем нового сотрудника в таблицу

string[] dateParts = BirthdayChooseCalendar.SelectedDate.ToString().Split('.');

Bithday = dateParts[2].Split(' ')[0] + "-" + dateParts[0] + "-" + dateParts[1] + " 00:00:00.000";

ConString = @"Data Source=DESKTOP-ITGU7A9\SQLEXPRESS; Initial Catalog=РаботникиNEW; Integrated Security = True;";

string CMDstring = $"INSERT INTO Сотрудники(Фамилия, Имя, Отчество, пол, датаРождения) VALUES ('{SurnameInputTB.Text}', '{NameInputTB.Text}', '{MiddleNameInputTB.Text}', '{SexCB.SelectedIndex.ToString()}', TRY\_CONVERT(datetime,'{Bithday}'))";

SqlCommand command = new SqlCommand(CMDstring, connection);

command.ExecuteNonQuery();

//получаем код должности, который выбрали при приеме

CMDstring = $"Select DISTINCT кодДолжности FROM Вакансии WHERE dbo.Вакансии.Вакансии <> 0 and кодОтдела = {DepartmentCode} and названиеДолжности = '{PositionCB.SelectedItem.ToString()}'";

command.CommandText = CMDstring;

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

PositionsCode.Add(Convert.ToInt32(reader[0].ToString()));

}

reader.Close();

PositionCode = PositionsCode.ToArray()[0];

//получаем номер нового сотрудника которого все еще нет в назначениях

CMDstring = $"SELECT номер FROM Сотрудники Where Фамилия = '{SurnameInputTB.Text}' AND Имя = '{NameInputTB.Text}' And Отчество = '{MiddleNameInputTB.Text}' AND номер NOT IN(SELECT номер FROM Назачения)";

command.CommandText = CMDstring;

reader = command.ExecuteReader();

while (reader.Read())

{

newMemberNum.Add(Convert.ToInt32(reader[0].ToString()));

}

reader.Close();

memberNum = newMemberNum.ToArray()[0];

//добавляем в назначениях назначение на должность нового сотрудника

CMDstring = $"INSERT INTO Назачения(номер, кодОтдела, кодДолжности, датаПриема) VALUES({memberNum},{DepartmentCode},{PositionCode},GETDATE())";

command = new SqlCommand(CMDstring, connection);

command.ExecuteNonQuery();

//освобождаем поля и переменные для добавления следующего добавления

ConString = @"Data Source=DESKTOP-ITGU7A9\SQLEXPRESS; Initial Catalog=РаботникиNEW; Integrated Security = True;";

SurnameInputTB.Text = null;

NameInputTB.Text = null;

MiddleNameInputTB.Text = null;

SexCB.SelectedItem = null;

DepartmentCode = null;

PositionCode = null;

memberNum = null;

DepartmentCB.SelectedItem = null;

DepartmentCB.ItemsSource = null;

PositionCB.SelectedItem = null;

PositionCB.ItemsSource = null;

DepartmensList.Clear();

PositionsList.Clear();

DepartmensCode.Clear();

PositionsCode.Clear();

newMemberNum.Clear();

AddMember.IsEnabled = false;

DepartmentCB.IsEnabled = true;

CMDstring = $"Select DISTINCT названиеОтдела FROM Вакансии WHERE dbo.Вакансии.Вакансии <> 0";

command = new SqlCommand(CMDstring, connection);

reader = command.ExecuteReader();

while (reader.Read())

{

DepartmensList.Add(reader[0].ToString());

}

reader.Close();

DepartmentCB.ItemsSource = DepartmensList;

}

catch (SqlException)

{

MessageBox.Show("Не удалось ввести данные", "Ошибка ввода данных", MessageBoxButton.OK);

}

}

private void AcceptDepartment\_Click(object sender, RoutedEventArgs e)

{

if (DepartmentCB.SelectedItem!=null)

{

AddMember.IsEnabled = true;

ConString = @"Data Source=DESKTOP-ITGU7A9\SQLEXPRESS; Initial Catalog=РаботникиNEW; Integrated Security = True;";

string CMDstring = $"Select DISTINCT названиеДолжности FROM Вакансии WHERE dbo.Вакансии.Вакансии <> 0 AND названиеОтдела = '{DepartmentCB.SelectedItem.ToString()}'";

SqlCommand command = new SqlCommand(CMDstring, connection);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

PositionsList.Add(reader[0].ToString());

}

reader.Close();

PositionCB.ItemsSource = PositionsList;

DepartmentCB.IsEnabled = false;

CMDstring = $"Select DISTINCT кодОтдела FROM Вакансии WHERE dbo.Вакансии.Вакансии <> 0 and названиеОтдела = '{DepartmentCB.SelectedItem.ToString()}'";

command.CommandText = CMDstring;

reader = command.ExecuteReader();

while (reader.Read())

{

DepartmensCode.Add(Convert.ToInt32(reader[0].ToString()));

}

reader.Close();

DepartmentCode = DepartmensCode.ToArray()[0];

BithdayDateView.Content = BirthdayChooseCalendar.SelectedDate;

}

else

{

MessageBox.Show("Вы не выбрали отдел", "Ошибка подтверждения", MessageBoxButton.OK);

}

}

}

}

Код окна выбора таблицы (InputTableName)

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

using System.Data.SqlClient;

using System.Data;

namespace WpfApp1

{

/// <summary>

/// Логика взаимодействия для InputTableName.xaml

/// </summary>

public partial class InputTableName : Window

{

ComboBox Tables;

List<string> TableNamesList = new List<string>();

public InputTableName()

{

InitializeComponent();

Tables = TablesNames;

string ConString = @"Data Source=DESKTOP-ITGU7A9\SQLEXPRESS; Initial Catalog=РаботникиNEW; Integrated Security = True;";

SqlConnection connection = new SqlConnection(ConString);

connection.Open();

string CMDstring = $"SELECT name FROM sysobjects WHERE type = 'U' OR type = 'V'";

SqlCommand command = new SqlCommand(CMDstring, connection);

SqlDataReader reader = command.ExecuteReader();

while (reader.Read())

{

TableNamesList.Add(reader[0].ToString());

}

Tables.ItemsSource = TableNamesList;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

this.DialogResult = true;

}

public string TableName

{

get

{

if (Tables.SelectedItem==null)

{

return null;

}

else

{

return Tables.SelectedItem.ToString();

}

}

set

{

}

}

}

}

Ответы на контрольные вопросы

Выводы о полученных практических навыках

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_ оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ подпись

преподавателя \_\_\_\_\_\_\_\_\_\_\_замечание