МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ДНІПРОВСЬКА ПОЛІТЕХНІКА»

Факультет: Інформаційних технологій

Кафедра: Програмного забезпечення комп’ютерних систем

(повна назва кафедри)

КУРСОВА РОБОТА

з дисципліни: **ОРГАНІЗАЦІЯ БАЗ ДАНИХ І БАЗ ЗНАНЬ**

(назва дисципліни)

на тему: «**Розробка автоматизованої інформаційної системи»**

Студента 2 курсу, групи 122-19-1

Спеціальності «комп’ютерні науки»

Ященко Богдан Сергійович

(прізвище та ініціали)

Керівник \_доцент, к.т.н., доцент \_\_\_\_\_\_

\_\_\_Кабак Л.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кабак Л.В.

(підпис) (прізвище та ініціали)

м. Дніпро - 2021 рік

**З А В Д А Н Н Я**

**на курсову роботу**

**Студента групи 122-19-1**

**Ященко Богдан Сергійович**

Тема курсової роботи "Розробка автоматизованої інформаційно-пошукової системи для контролю виконання доручень"

Термін здачі закінченої роботи (10.05.2021)

Перелік питань для опрацювання:

1. Створити концептуальну модель предметної області згідно отриманого завдання.
2. Розробити на підставі концептуальної моделі розробити ER-діаграму бази даних.
3. Розробити SQL – Скрипт для створення бази даних (база даних крім таблиць повинна мати представлення, данні внесені у представлення повинні заноситись у таблицю за допомогою тригерів instead of)
4. На мові PL/SQL розробити тригери для генерації ключів.
5. Створити додаток використовуючи мову програмування C# та Oracle Net frame для роботи з БД (Додаток повинен здійснювати занесення, модифікацію та перегляд даних, проводити аналіз внесених даних за допомогою діаграм, та генерувати звіти).

Дата видачі завдання *\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Керівник курсової роботи доц. Кабак Л.В.

Завдання до виконання отримав Ященко Богдан Сергійович

Зміст

[Індивідуальне завдання на курсову роботу № 24 4](#_Toc71565594)

[Розробки логічної та фізичної моделі 5](#_Toc71565595)

[Створення Windows Forms програми 6](#_Toc71565596)

[SQL-скрипти для створення бази даних: 11](#_Toc71565597)

[SQL-скрипти для створення тригерів: 13](#_Toc71565598)

[Лістинг коду мовою C# для компілятора Visual C#: 15](#_Toc71565599)

[ЛІТЕРАТУРА 27](#_Toc71565600)

# Індивідуальне завдання на курсову роботу № 24

Створіть інформаційну систему з використанням бази даних для “Контролю виконання доручень” для будь-якої організації.

В якості вхідної інформації використовуються дані:

* порядковий номер доручення;
* назва доручення;
* зміст доручення;
* дата видачі доручення;
* строк виконання;
* дата фактичного виконання;
* виконавець;
* хто надав доручення.

Доручення може видавати керівник організації і керівники підрозділів.

Інформаційна система повинна забезпечити:

* введення і корегування інформації про доручення;
* перегляд доручень визначеною датою;
* щоденний друк доручень з поточною датою виконання (для керівника).

Розроблений додаток повинен створювати звіти за заданий період.

# Розробки логічної та фізичної моделі

Проектування бази даних починається з розробки логічної моделі. Було створено три таблиці: ASSIGNMENTS, EMPLOYEE, DEPARTMENT - призначення імен таблиць і їхніх атрибутів відображені у таблиці 1.1. Між таблицями встановлено взаємозв’язок "один до одного" у наступному вигляді: DEPARTMENT => EMPLOYEE => ASSIGNMENTS.

Таблиці 1.1 - Призначення імен таблиць і їхніх атрибутів

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DEPARTMENT (ВІДДІЛ) | | |  | | |
| № п/п | Найменування стовбців | Примітка | | Тип | Розмір |
| 1. | Dept\_id | Унікальний ключ відділу | | integer |  |
| 2. | Dept\_name | Найменування відділу | | nvarchar2 | 40 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EMPLOYEE (ПРАЦІВНИК) | | |  | | |
| № п/п | Найменування стовбців | Примітка | | Тип | Розмір |
| 1. | Emp\_id | Унікальний ключ працівника | | integer |  |
| 2. | First\_name | Ім’я | | nvarchar2 | 40 |
| 3. | Last\_name | Прізвище | | nvarchar2 | 40 |
| 4. | Job | Посада | | nvarchar2 | 40 |
| 5. | Salary | Заробітна плата | | integer |  |
| 6. | Hire\_date | Дата найму | | date |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ASSIGNMENTS (ДОРУЧЕННЯ) | | |  | | |
| № п/п | Найменування стовбців | Примітка | | Тип | Розмір |
| 1. | Assignments\_id | Унікальний ключ доручення | | integer |  |
| 2. | Assignments\_name | Найменування доручення | | nvarchar2 | 40 |
| 3. | Date\_of\_issue | Дата видачі доручення | | date |  |
| 4. | Deadline | Кінцевий термін виконання доручення | | date |  |
| 5. | Execution\_date | Дата виконання доручення | | date |  |
| 6. | Assignments\_content | Зміст доручення | | nvarchar2 | 150 |
| 7. | Executant | Виконавець | | nvarchar2 | 40 |
| 8. | Who\_gave\_the\_order | Хто надав доручення | | nvarchar2 | 40 |

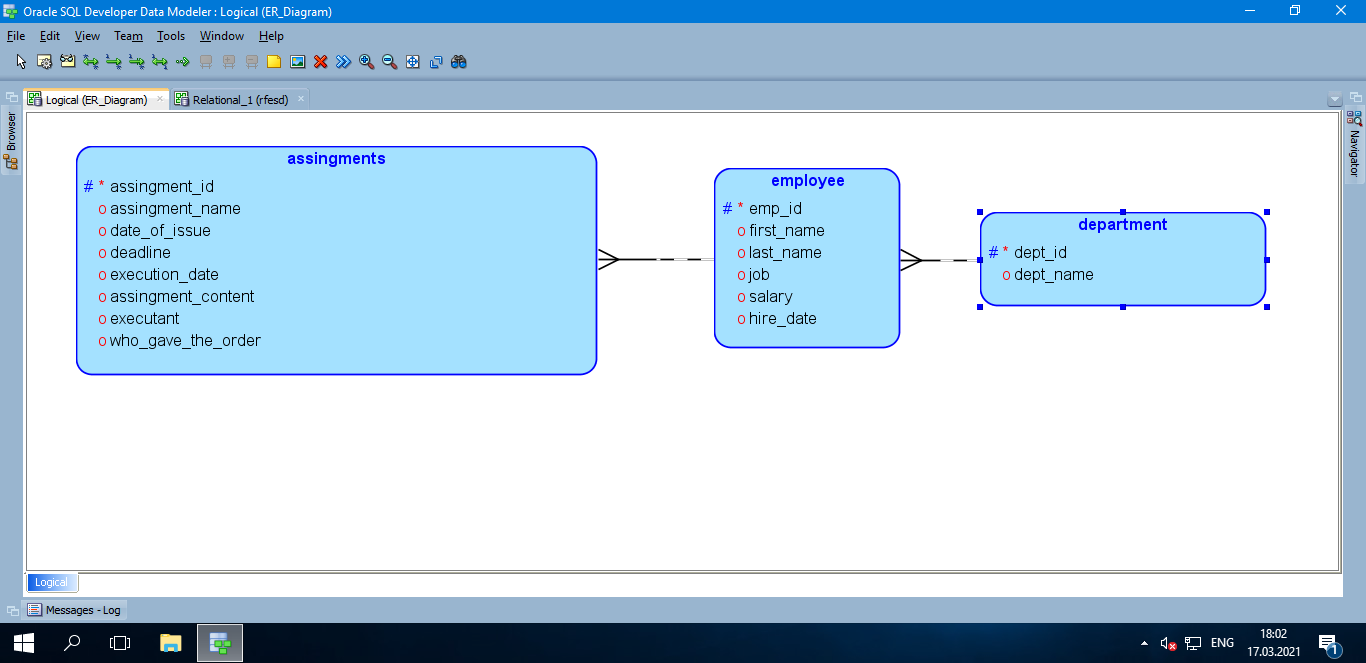


Рисунок 1.2 – Логічна модель

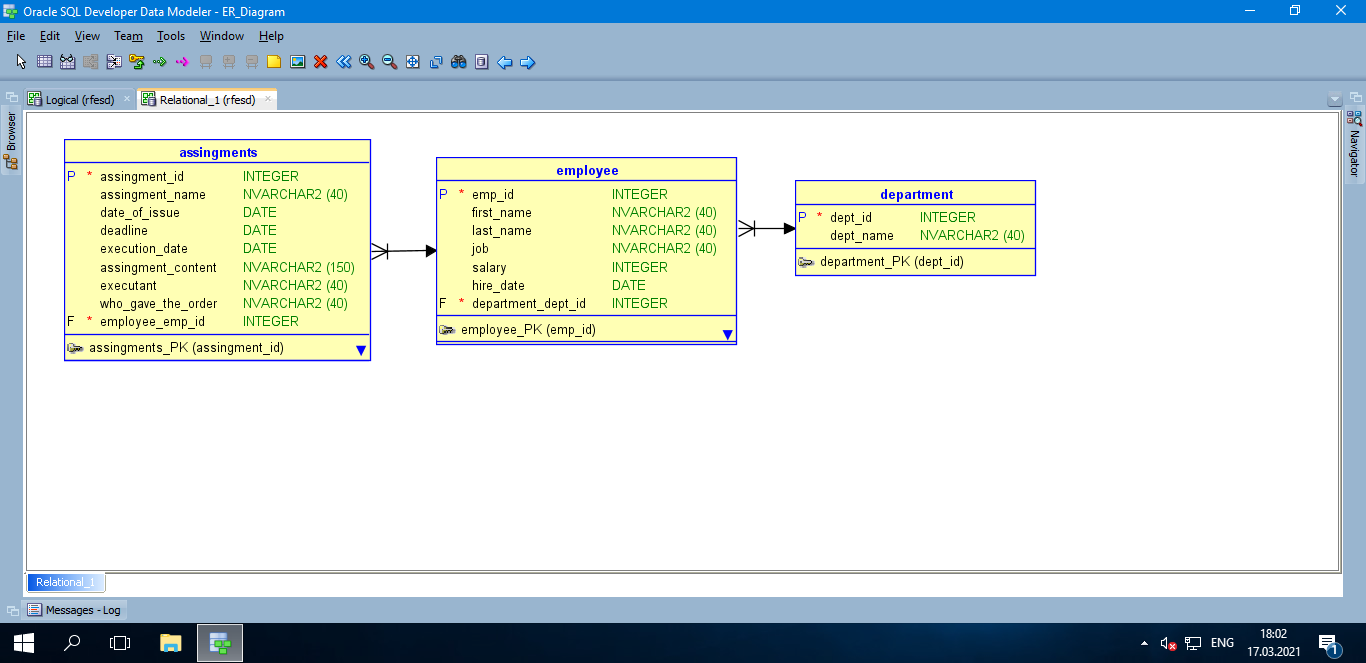


Рисунок 1.3 – Фізична модель

# Створення Windows Forms програми

Далі створюємо Windows Forms програму для роботи з базою даних. Програма складається з меню для вибору таблиці, форм для редагування кожної з таблиць, форми для генерації звітів та проведення аналізу внесених даних за допомогою діаграм, допоміжні форми для вибору періодів дат або певної дати.

Для здійснення пошуку в таблиці було додано додаткові SQL-скрипти в DataSet1.

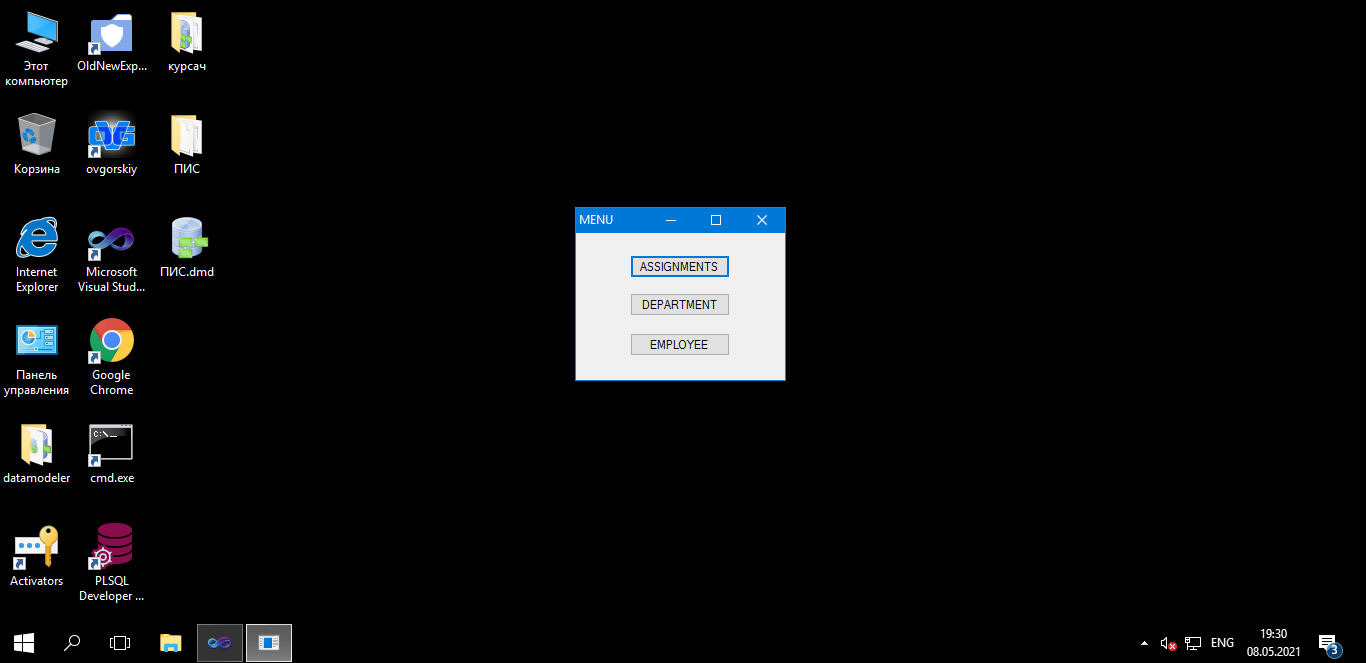


Рисунок 1.4 – Меню

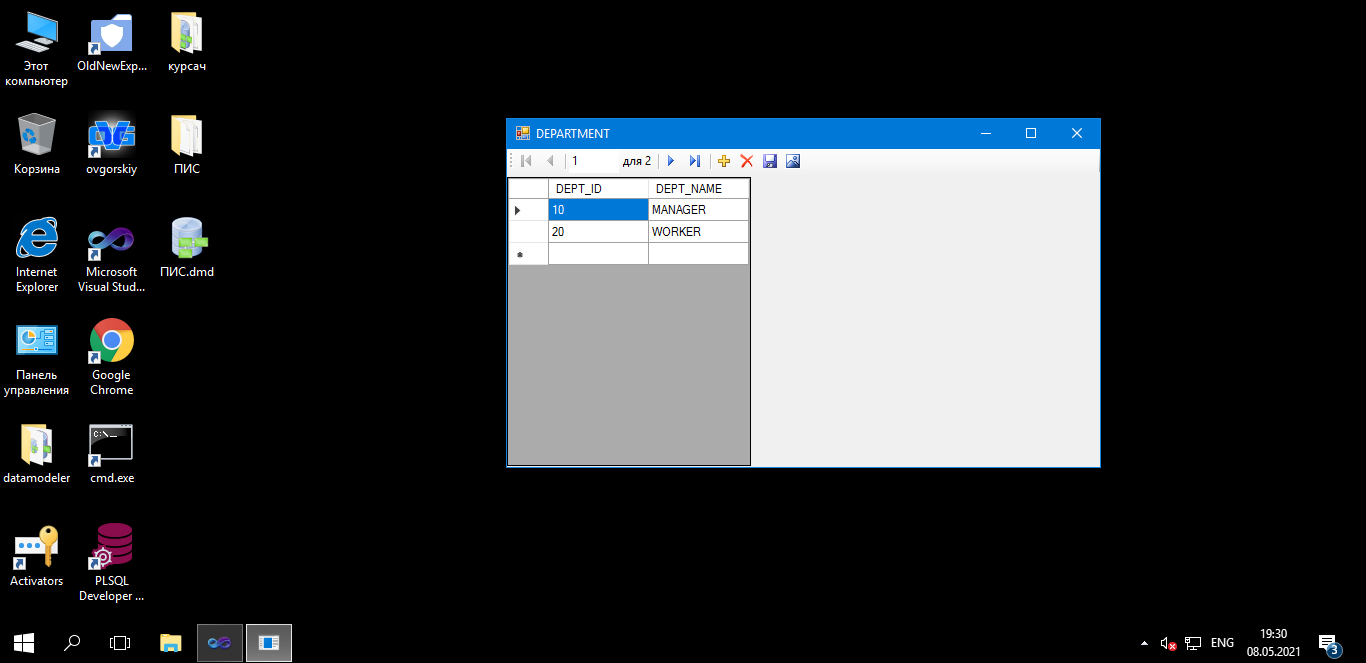


Рисунок 1.5 – Таблиця DEPARTMENT

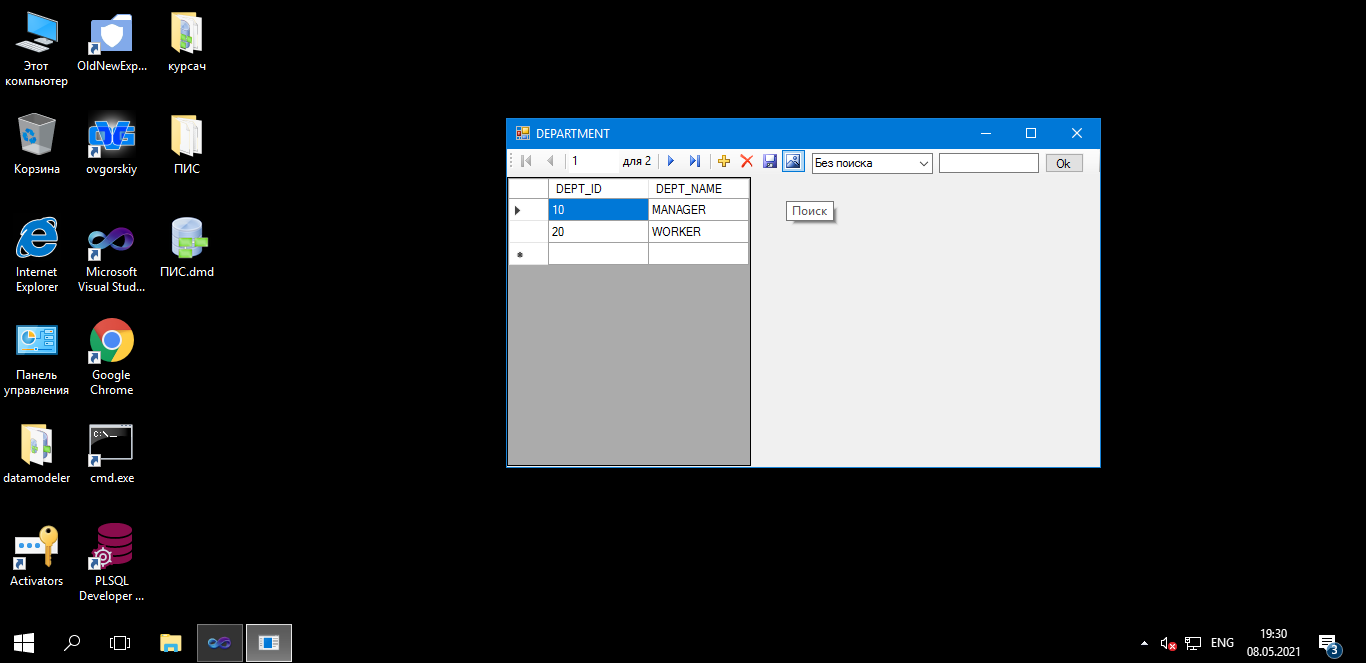


Рисунок 1.6 – Реалізація пошуку в таблиці DEPARTMENT

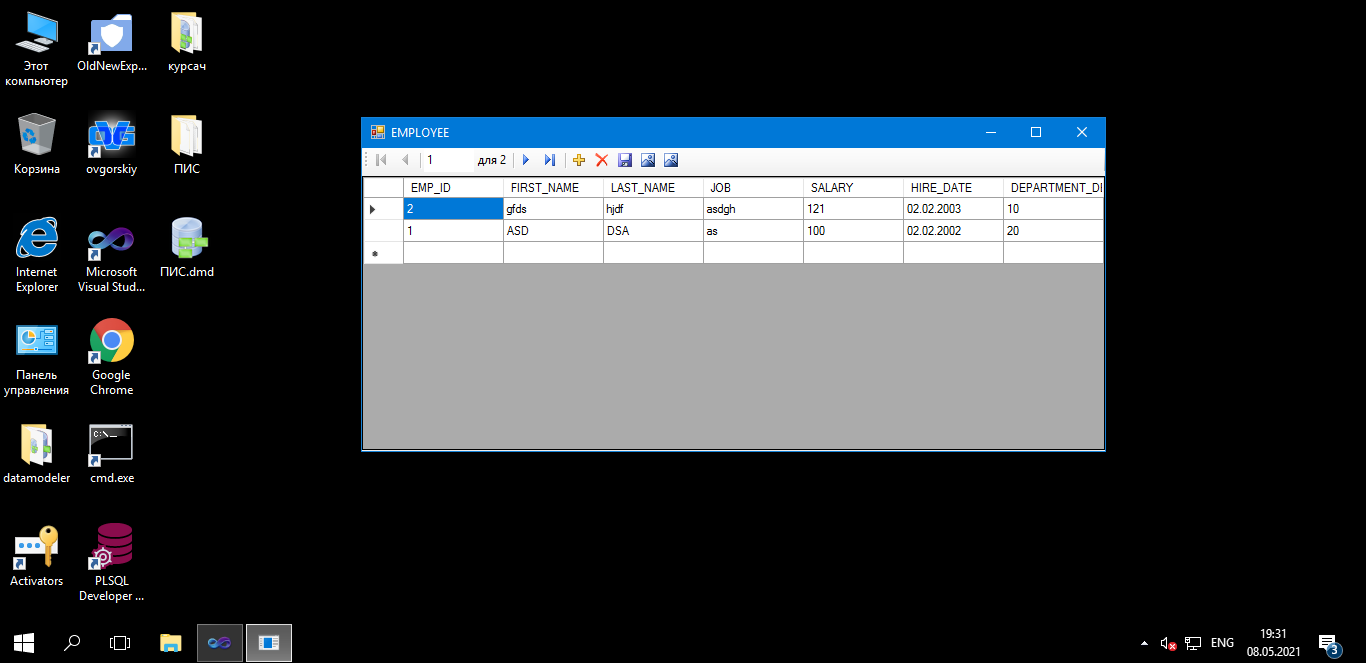


Рисунок 1.7 – Таблиця EMPLOYEE

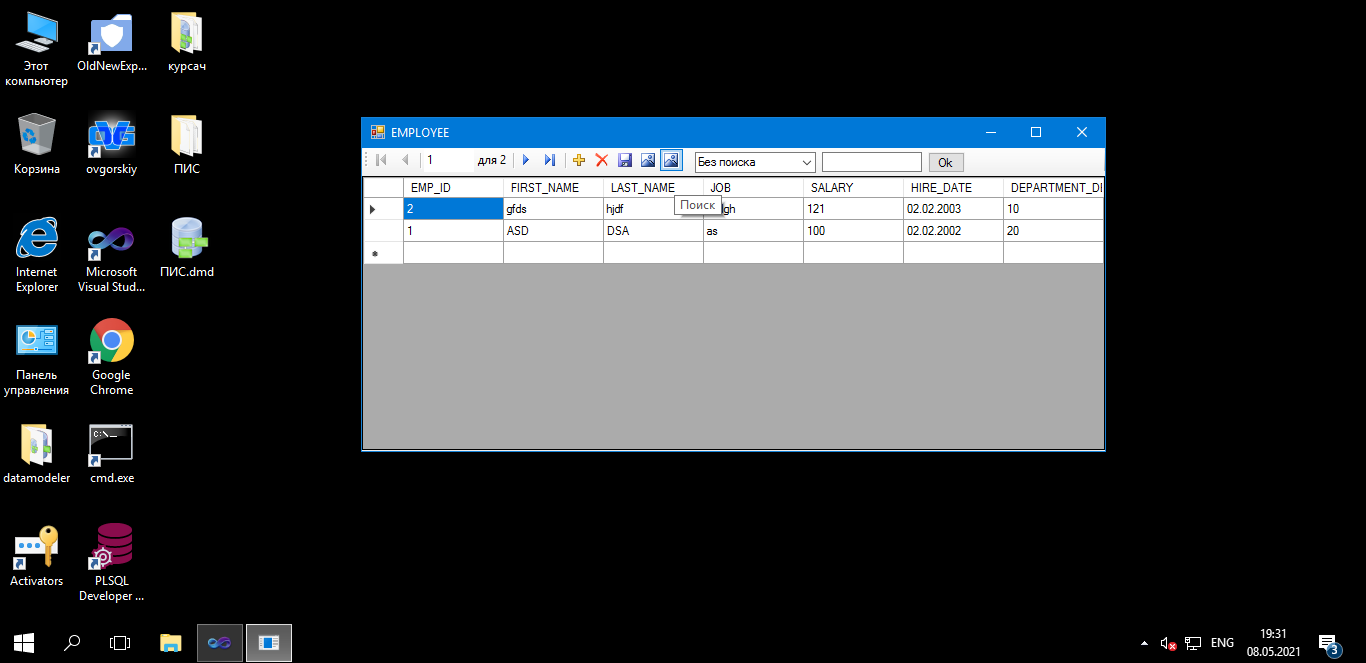


Рисунок 1.8 – Реалізація пошуку в таблиці EMPLOYEE

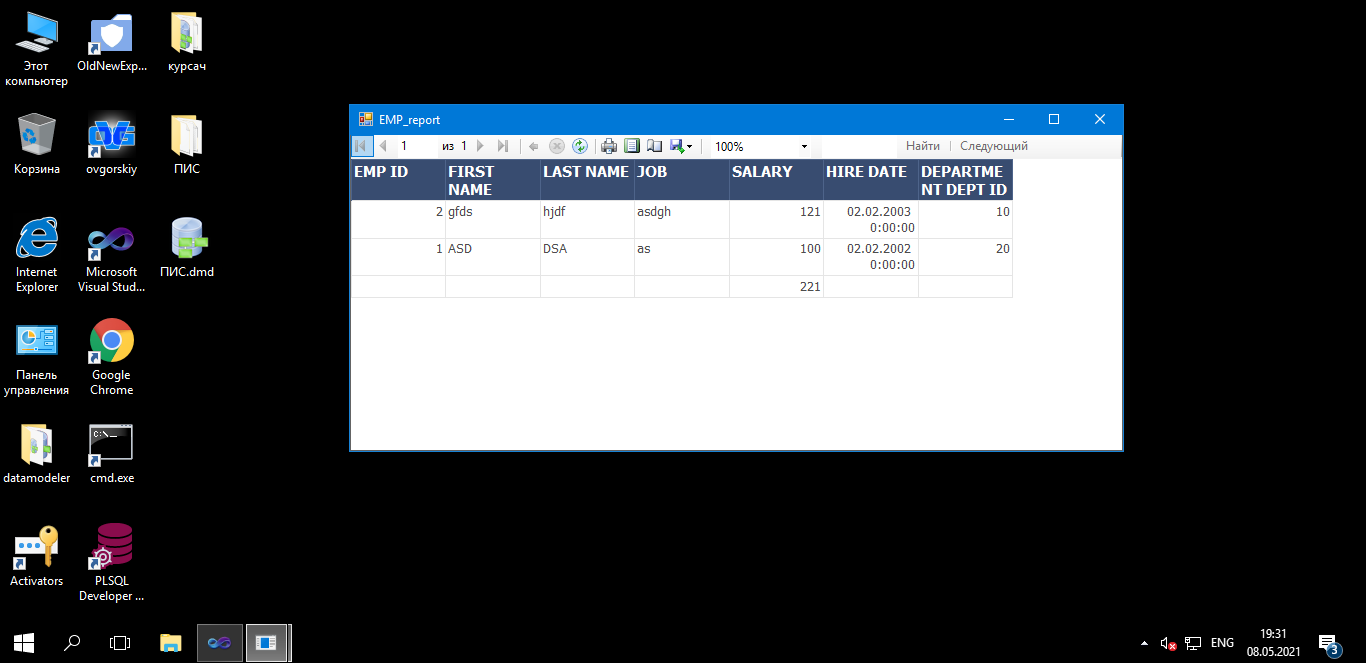


Рисунок 1.9 – Звіт таблиці EMPLOYEE

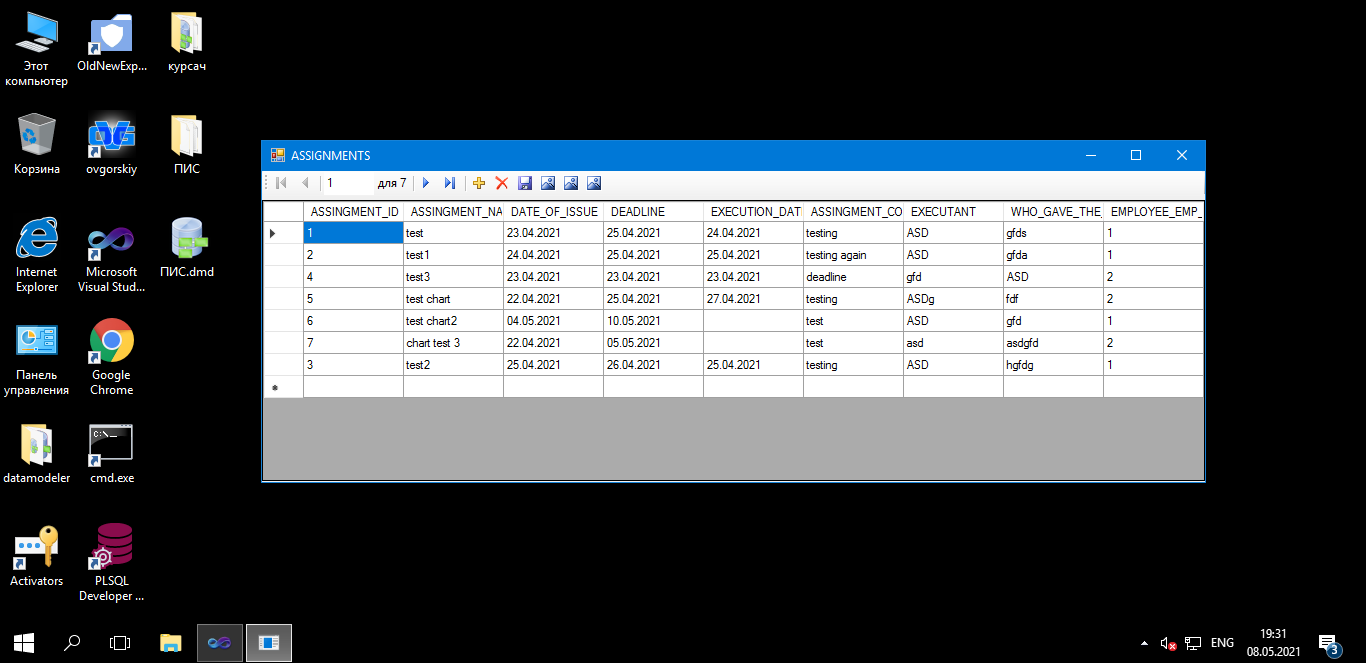


Рисунок 1.10 – Таблиця ASSIGNMENTS

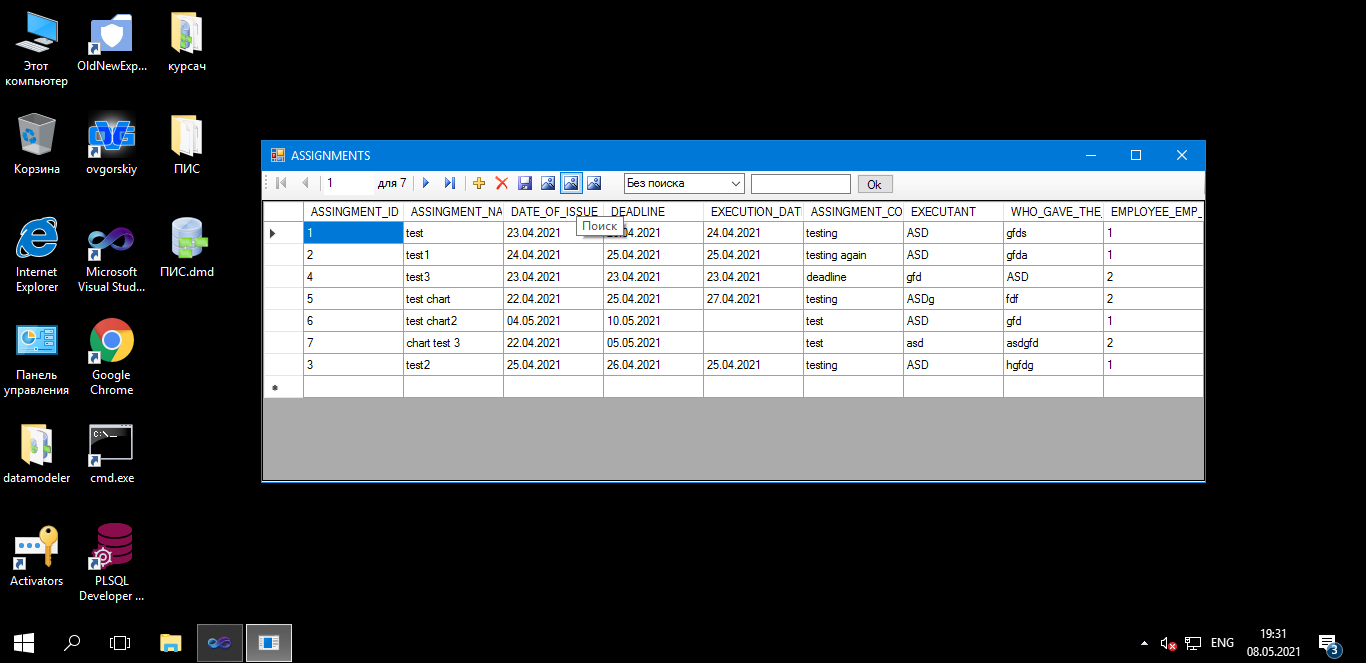


Рисунок 1.11 – Реалізація пошуку в таблиці ASSIGNMENTS

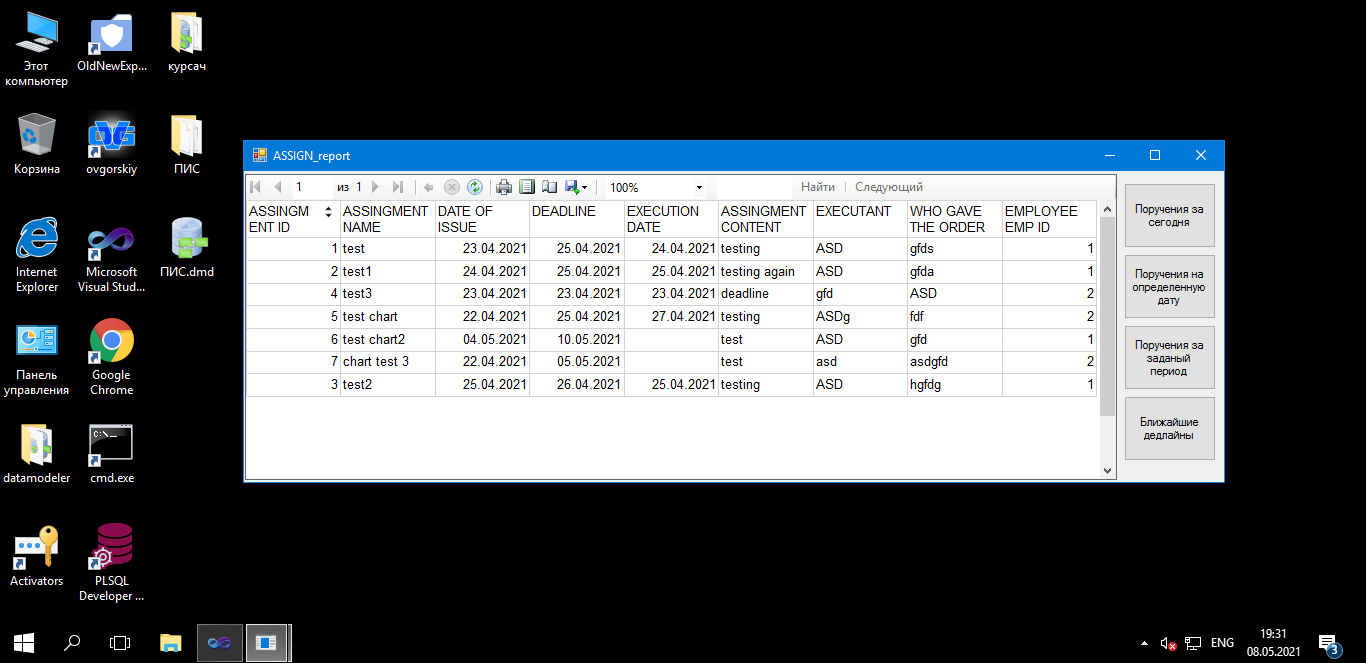


Рисунок 1.12 – Звіт таблиці ASSIGNMENTS

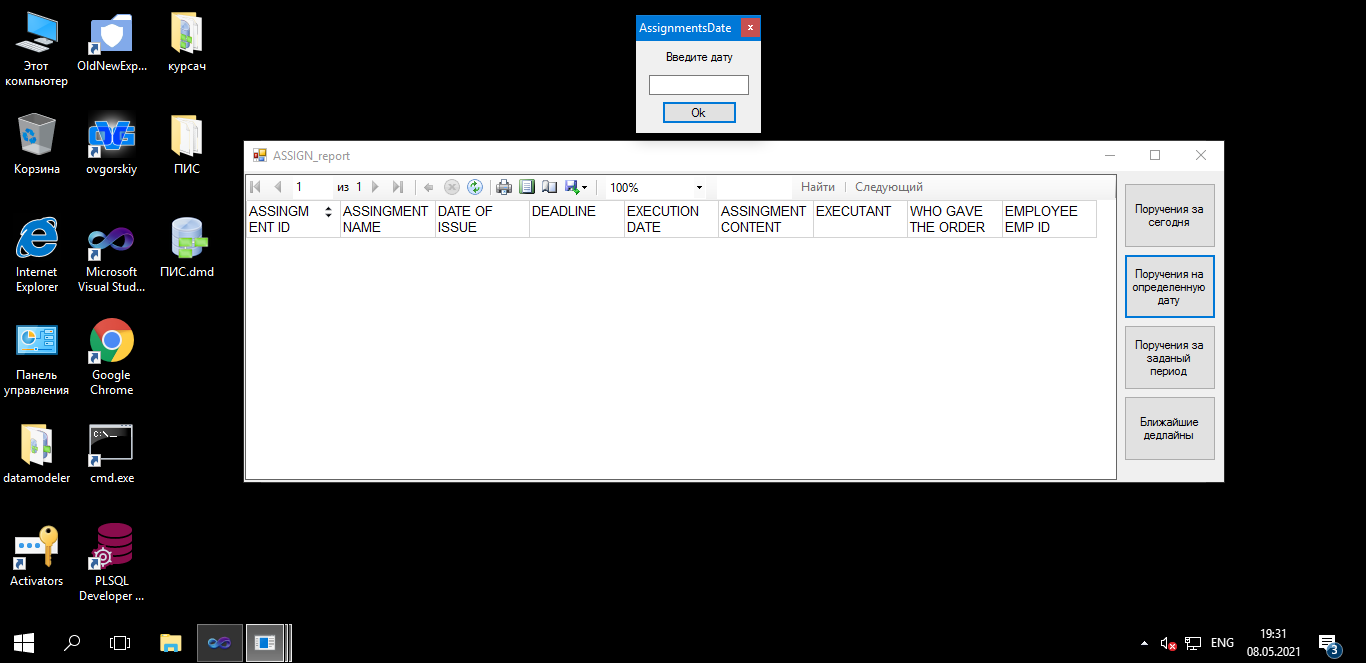


Рисунок 1.13 – Вибір дати для звіту таблиці ASSIGNMENTS

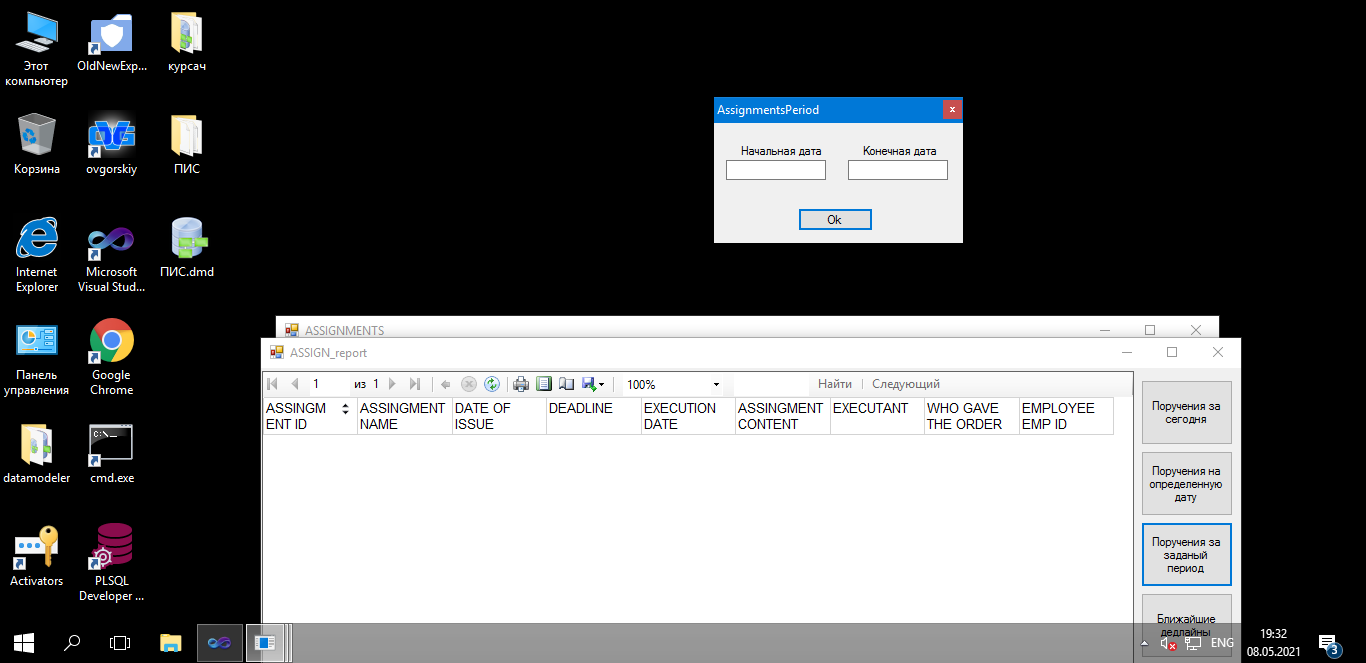


Рисунок 1.14 – Вибір періоду для звіту таблиці ASSIGNMENTS

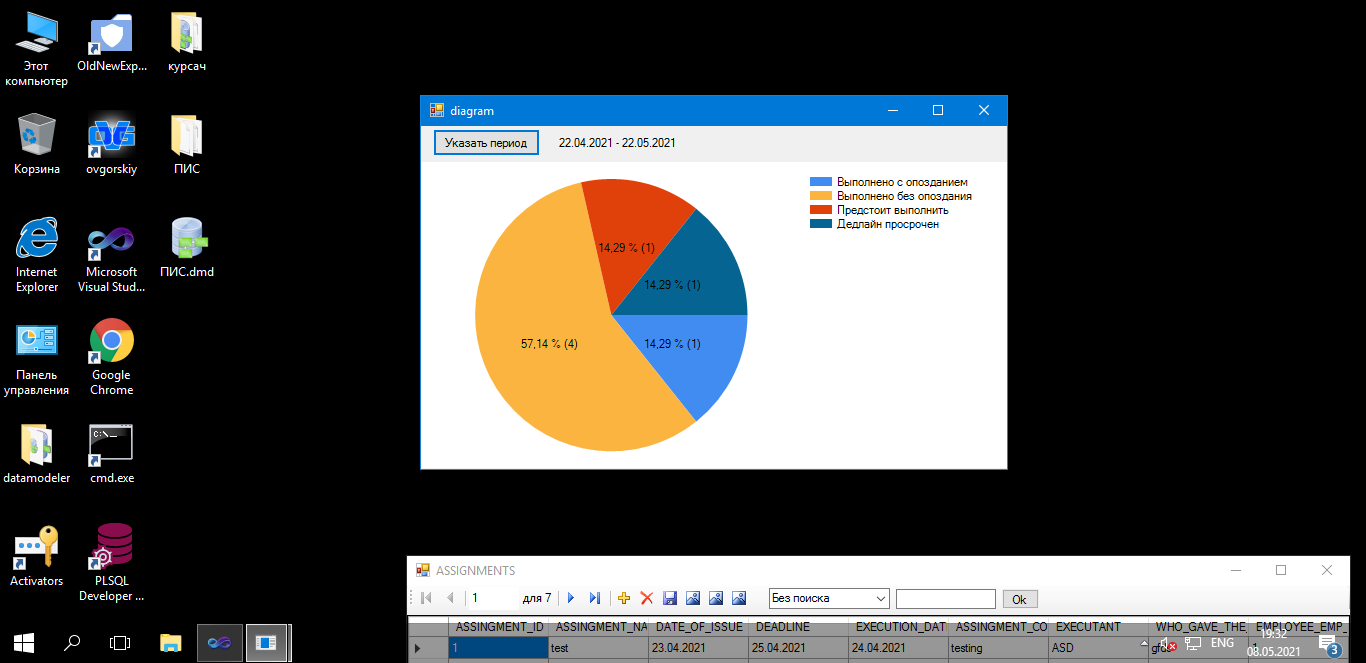


Рисунок 1.15 – Аналіз внесених даних в таблицю ASSIGNMENTS за допомогою діаграми

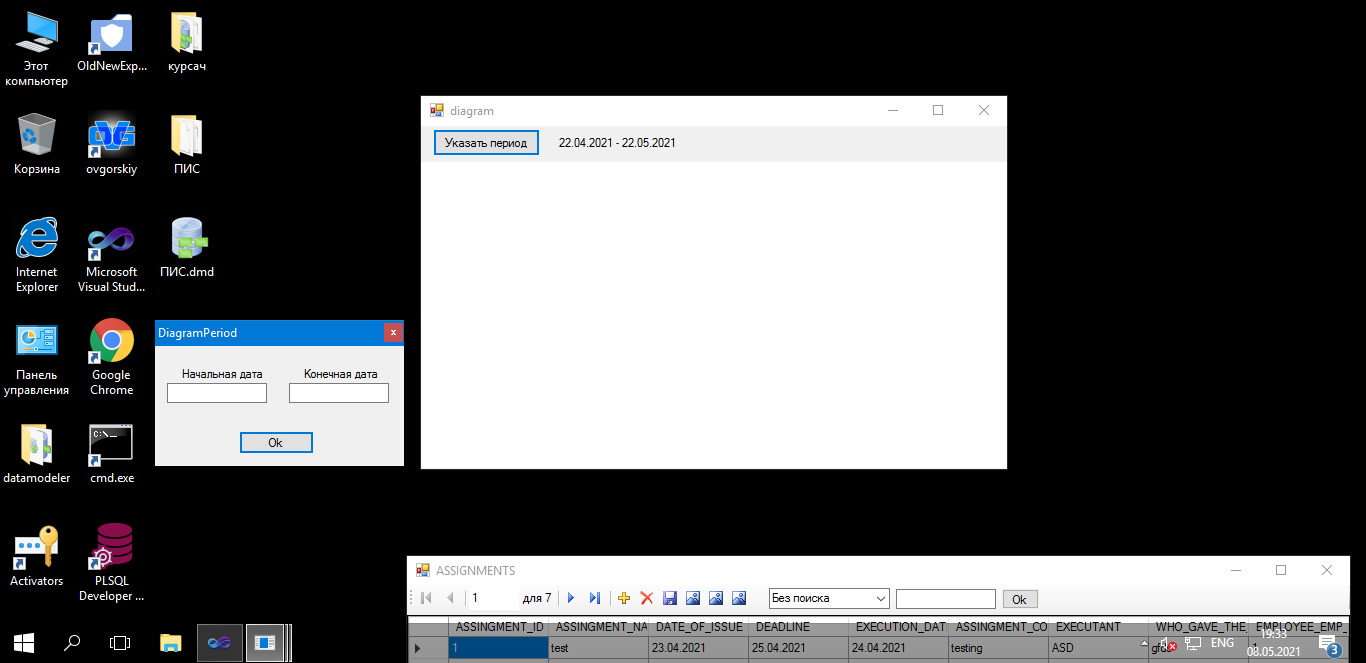


Рисунок 1.16 – Вибір періоду для діаграми таблиці ASSIGNMENTS

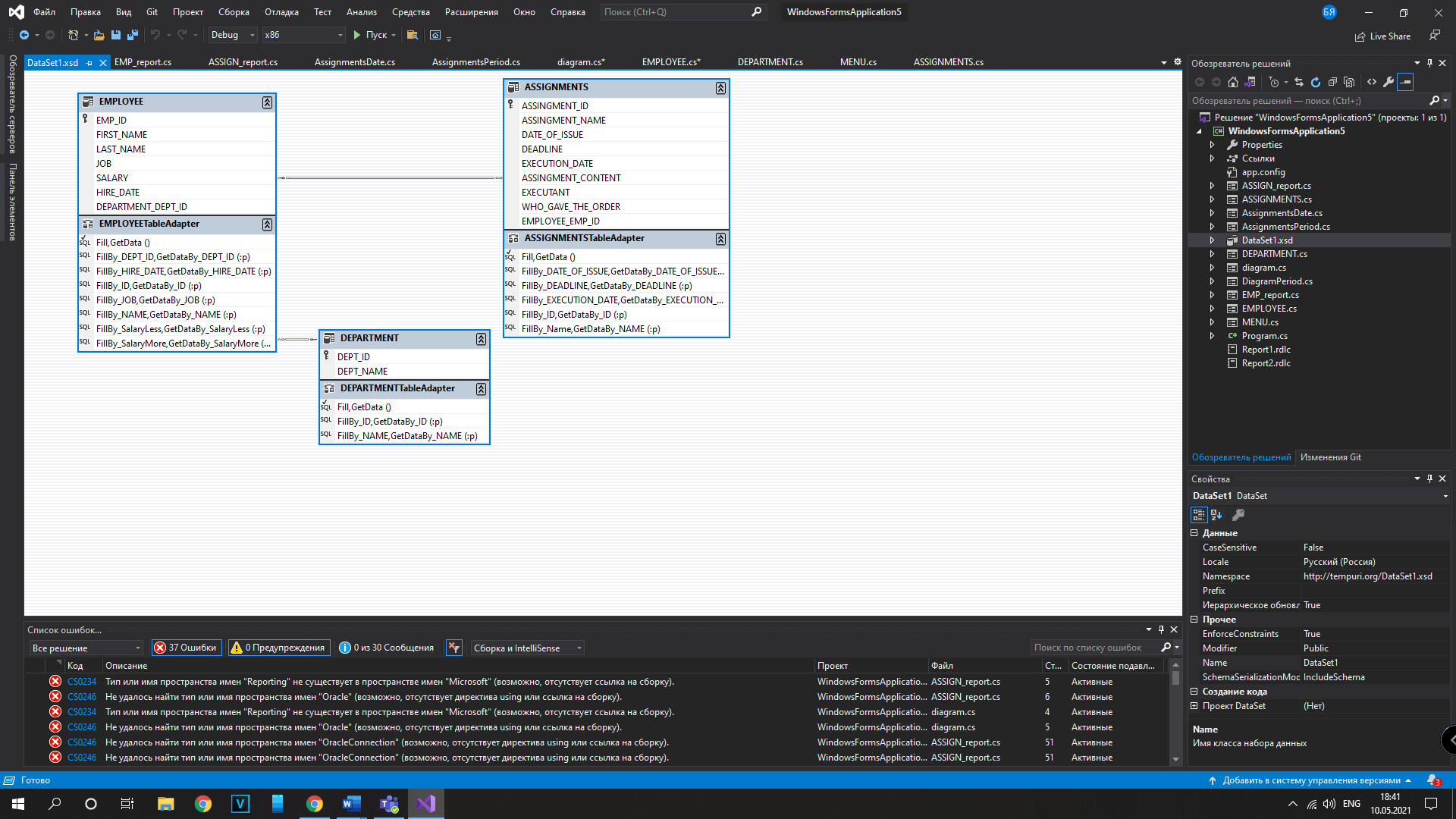


Рисунок 1.17 – DataSet1 з додатковими SQL-скриптами

# SQL-скрипти для створення бази даних:

-- Generated by Oracle SQL Developer Data Modeler 19.4.0.350.1424

-- at: 2021-03-17 18:03:12 EET

-- site: Oracle Database 11g

-- type: Oracle Database 11g

CREATE TABLE assingments (

assingment\_id INTEGER NOT NULL,

assingment\_name NVARCHAR2(40),

date\_of\_issue DATE,

deadline DATE,

execution\_date DATE,

assingment\_content NVARCHAR2(150),

executant NVARCHAR2(40),

who\_gave\_the\_order NVARCHAR2(40),

employee\_emp\_id INTEGER NOT NULL

);

ALTER TABLE assingments ADD CONSTRAINT assingments\_pk PRIMARY KEY ( assingment\_id );

CREATE TABLE department (

dept\_id INTEGER NOT NULL,

dept\_name NVARCHAR2(40)

);

ALTER TABLE department ADD CONSTRAINT department\_pk PRIMARY KEY ( dept\_id );

CREATE TABLE employee (

emp\_id INTEGER NOT NULL,

first\_name NVARCHAR2(40),

last\_name NVARCHAR2(40),

job NVARCHAR2(40),

salary INTEGER,

hire\_date DATE,

department\_dept\_id INTEGER NOT NULL

);

ALTER TABLE employee ADD CONSTRAINT employee\_pk PRIMARY KEY ( emp\_id );

ALTER TABLE assingments

ADD CONSTRAINT assingments\_employee\_fk FOREIGN KEY ( employee\_emp\_id )

REFERENCES employee ( emp\_id );

ALTER TABLE employee

ADD CONSTRAINT employee\_department\_fk FOREIGN KEY ( department\_dept\_id )

REFERENCES department ( dept\_id );

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE 3

-- CREATE INDEX 0

-- ALTER TABLE 5

-- CREATE VIEW 0

-- ALTER VIEW 0

-- CREATE PACKAGE 0

-- CREATE PACKAGE BODY 0

-- CREATE PROCEDURE 0

-- CREATE FUNCTION 0

-- CREATE TRIGGER 0

-- ALTER TRIGGER 0

-- CREATE COLLECTION TYPE 0

-- CREATE STRUCTURED TYPE 0

-- CREATE STRUCTURED TYPE BODY 0

-- CREATE CLUSTER 0

-- CREATE CONTEXT 0

-- CREATE DATABASE 0

-- CREATE DIMENSION 0

-- CREATE DIRECTORY 0

-- CREATE DISK GROUP 0

-- CREATE ROLE 0

-- CREATE ROLLBACK SEGMENT 0

-- CREATE SEQUENCE 0

-- CREATE MATERIALIZED VIEW 0

-- CREATE MATERIALIZED VIEW LOG 0

-- CREATE SYNONYM 0

-- CREATE TABLESPACE 0

-- CREATE USER 0

--

-- DROP TABLESPACE 0

-- DROP DATABASE 0

--

-- REDACTION POLICY 0

--

-- ORDS DROP SCHEMA 0

-- ORDS ENABLE SCHEMA 0

-- ORDS ENABLE OBJECT 0

--

-- ERRORS 0

-- WARNINGS 0

# SQL-скрипти для створення тригерів:

CREATE SEQUENCE ASSIGNMENTS\_ASSIGNMENTS\_ID\_SEQ START WITH 1 NOCACHE ORDER;

CREATE SEQUENCE ASSIGNMENTS\_DEPARTMENT\_ID\_SEQ START WITH 1 NOCACHE ORDER;

CREATE SEQUENCE ASSIGNMENTS\_EMPLOYEE\_ID\_SEQ START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER ASSIGNMENTS\_ASSIGNMENTS\_id\_trg BEFORE

INSERT OR UPDATE ON ASSIGNMENTS

FOR EACH ROW

WHEN ( new.assingment\_id IS NULL )

BEGIN

:new.assingment\_id := ASSIGNMENTS\_ASSIGNMENTS\_id\_seq.nextval;

END;

/

CREATE OR REPLACE TRIGGER ASSIGNMENTS\_DEPARTMENT\_id\_trg BEFORE

INSERT OR UPDATE ON DEPARTMENT

FOR EACH ROW

WHEN ( new.DEPT\_id IS NULL )

BEGIN

:new.DEPT\_id := ASSIGNMENTS\_DEPARTMENT\_id\_seq.nextval;

END;

/

CREATE OR REPLACE TRIGGER ASSIGNMENTS\_EMPLOYEE\_id\_trg BEFORE

INSERT OR UPDATE ON EMPLOYEE

FOR EACH ROW

WHEN ( new.emp\_id IS NULL )

BEGIN

:new.emp\_id := ASSIGNMENTS\_EMPLOYEE\_id\_seq.nextval;

END;

/

# Лістинг коду мовою C# для компілятора Visual C#:

MENU

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class MENU : Form

{

public MENU()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

ASSIGNMENTS f = new ASSIGNMENTS(); //Создаем экземпляр формы ASSIGNMENTS

this.Hide(); //Скрываем эту форму

f.Show(this); //Показываем форму ASSIGNMENTS и указываем владельцем эту форму

}

private void button2\_Click(object sender, EventArgs e)

{

DEPARTMENT f = new DEPARTMENT(); //Создаем экземпляр формы DEPARTMENT

this.Hide(); //Скрываем эту форму

f.Show(this); //Показываем форму DEPARTMENT и указываем владельцем эту форму

}

private void button3\_Click(object sender, EventArgs e)

{

EMPLOYEE f = new EMPLOYEE(); //Создаем экземпляр формы EMPLOYEE

this.Hide(); //Скрываем эту форму

f.Show(this); //Показываем форму EMPLOYEE и указываем владельцем эту форму

}

}

}

ASSIGNMENTS

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class ASSIGNMENTS : Form

{

public ASSIGNMENTS()

{

InitializeComponent();

this.FormClosed += new FormClosedEventHandler(ASSIGNMENTS\_FormClosed); //Добавляем обработчки событий при закрытии формы

comboBox1.Hide();

textBox1.Hide();

button1.Hide();

comboBox1.SelectedIndex = 0;

}

void ASSIGNMENTS\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Owner.Show(); //Показываем форму владельца при закрытии

}

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

{

try

{

this.Validate();

this.aSSIGNMENTSBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.dataSet1);

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "dataSet1.ASSIGNMENTS". При необходимости она может быть перемещена или удалена.

this.aSSIGNMENTSTableAdapter.Fill(this.dataSet1.ASSIGNMENTS);

}

private void toolStripButton1\_Click(object sender, EventArgs e) //В toolStrip добавляем новую кнопку которая открывает форму с отчетом по этой таблице

{

ASSIGN\_report f = new ASSIGN\_report(); //Создаем экземпляр формы ASSIGN\_report

f.Show(this); //Показываем форму ASSIGN\_report и указываем владельцем эту форму

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

comboBox1.Show();

button1.Show();

textBox1.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (comboBox1.SelectedIndex == 0)

{

this.aSSIGNMENTSTableAdapter.Fill(this.dataSet1.ASSIGNMENTS);

}

if (comboBox1.SelectedIndex == 1)

{

this.aSSIGNMENTSTableAdapter.FillBy\_ID(this.dataSet1.ASSIGNMENTS, Decimal.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 2)

{

this.aSSIGNMENTSTableAdapter.FillBy\_Name(this.dataSet1.ASSIGNMENTS, textBox1.Text);

}

if (comboBox1.SelectedIndex == 3)

{

this.aSSIGNMENTSTableAdapter.FillBy\_DATE\_OF\_ISSUE(this.dataSet1.ASSIGNMENTS, DateTime.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 4)

{

this.aSSIGNMENTSTableAdapter.FillBy\_EXECUTION\_DATE(this.dataSet1.ASSIGNMENTS, DateTime.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 5)

{

this.aSSIGNMENTSTableAdapter.FillBy\_DEADLINE(this.dataSet1.ASSIGNMENTS, DateTime.Parse(textBox1.Text));

}

comboBox1.Hide();

textBox1.Hide();

button1.Hide();

textBox1.Text = "";

comboBox1.SelectedIndex = 0;

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

private void toolStripButton3\_Click(object sender, EventArgs e)

{

diagram f = new diagram(); //Создаем экземпляр формы diagram

f.Show(this); //Показываем форму diagram и указываем владельцем эту форму

}

}

}

DEPARTMENT

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class DEPARTMENT : Form

{

public DEPARTMENT()

{

InitializeComponent();

this.FormClosed += new FormClosedEventHandler(DEPARTMENT\_FormClosed); //Добавляем обработчки событий при закрытии формы

comboBox1.Hide();

textBox1.Hide();

button1.Hide();

comboBox1.SelectedIndex = 0;

}

void DEPARTMENT\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Owner.Show(); //Показываем форму владельца при закрытии

}

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

{

try

{

this.Validate();

this.dEPARTMENTBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.dataSet1);

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "dataSet1.DEPARTMENT". При необходимости она может быть перемещена или удалена.

this.dEPARTMENTTableAdapter.Fill(this.dataSet1.DEPARTMENT);

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (comboBox1.SelectedIndex == 0)

{

this.dEPARTMENTTableAdapter.Fill(this.dataSet1.DEPARTMENT);

}

if (comboBox1.SelectedIndex == 1)

{

this.dEPARTMENTTableAdapter.FillBy\_ID(this.dataSet1.DEPARTMENT, Decimal.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 2)

{

this.dEPARTMENTTableAdapter.FillBy\_NAME(this.dataSet1.DEPARTMENT, textBox1.Text);

}

comboBox1.Hide();

textBox1.Hide();

button1.Hide();

textBox1.Text = "";

comboBox1.SelectedIndex = 0;

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

private void toolStripButton1\_Click(object sender, EventArgs e)

{

comboBox1.Show();

button1.Show();

textBox1.Show();

}

}

}

EMPLOYEE

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class EMPLOYEE : Form

{

public EMPLOYEE()

{

InitializeComponent();

this.FormClosed += new FormClosedEventHandler(EMPLOYEE\_FormClosed); //Добавляем обработчки событий при закрытии формы

comboBox1.Hide();

textBox1.Hide();

button1.Hide();

comboBox1.SelectedIndex = 0;

}

void EMPLOYEE\_FormClosed(object sender, FormClosedEventArgs e)

{

this.Owner.Show(); //Показываем форму владельца при закрытии

}

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

{

try

{

this.Validate();

this.eMPLOYEEBindingSource.EndEdit();

this.tableAdapterManager.UpdateAll(this.dataSet1);

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

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

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "dataSet1.EMPLOYEE". При необходимости она может быть перемещена или удалена.

this.eMPLOYEETableAdapter.Fill(this.dataSet1.EMPLOYEE);

}

private void toolStripButton1\_Click(object sender, EventArgs e) //В toolStrip добавляем новую кнопку которая открывает форму с отчетом по этой таблице

{

EMP\_report f = new EMP\_report(); //Создаем экземпляр формы EMP\_report

f.Show(this); //Показываем форму EMP\_report и указываем владельцем эту форму

}

private void toolStripButton2\_Click(object sender, EventArgs e)

{

comboBox1.Show();

button1.Show();

textBox1.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

if (comboBox1.SelectedIndex == 0)

{

this.eMPLOYEETableAdapter.Fill(this.dataSet1.EMPLOYEE);

}

if (comboBox1.SelectedIndex == 1)

{

this.eMPLOYEETableAdapter.FillBy\_ID(this.dataSet1.EMPLOYEE, Decimal.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 2)

{

this.eMPLOYEETableAdapter.FillBy\_NAME(this.dataSet1.EMPLOYEE, textBox1.Text);

}

if (comboBox1.SelectedIndex == 3)

{

this.eMPLOYEETableAdapter.FillBy\_JOB(this.dataSet1.EMPLOYEE, textBox1.Text);

}

if (comboBox1.SelectedIndex == 4)

{

this.eMPLOYEETableAdapter.FillBy\_SalaryMore(this.dataSet1.EMPLOYEE, Decimal.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 5)

{

this.eMPLOYEETableAdapter.FillBy\_SalaryLess(this.dataSet1.EMPLOYEE, Decimal.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 6)

{

this.eMPLOYEETableAdapter.FillBy\_HIRE\_DATE(this.dataSet1.EMPLOYEE, DateTime.Parse(textBox1.Text));

}

if (comboBox1.SelectedIndex == 7)

{

this.eMPLOYEETableAdapter.FillBy\_DEPT\_ID(this.dataSet1.EMPLOYEE, Decimal.Parse(textBox1.Text));

}

comboBox1.Hide();

textBox1.Hide();

button1.Hide();

textBox1.Text = "";

comboBox1.SelectedIndex = 0;

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

}

}

Diagram

using System;

using System.Data;

using System.Windows.Forms;

using Microsoft.Reporting.WinForms;

using Oracle.DataAccess.Client;

namespace WindowsFormsApplication5

{

public partial class diagram : Form

{

private static string beginDate, endDate; //Переменные для создания отчета за период

/\*

Set и Get методы для переменных

\*/

public static string GetEndDate()

{

return endDate;

}

public static void SetBeginDate(string newDate)

{

beginDate = newDate;

}

public static string GetBeginDate()

{

return beginDate;

}

public static void SetEndDate(string newDate)

{

endDate = newDate;

}

public diagram()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

chart1.Series["s1"].Points.Clear();

DiagramPeriod f = new DiagramPeriod();

f.ShowDialog(this);

try

{

OracleConnection connection = new OracleConnection();

connection.ConnectionString = Properties.Settings.Default.ConnectionString;

connection.Open(); //Подключаемся к Базе Данных

OracleCommand cmd = connection.CreateCommand(); //Создаем SQL команду

chart1.Series["s1"].XValueType = System.Windows.Forms.DataVisualization.Charting.ChartValueType.String;

cmd.CommandText = "select COUNT(a.assingment\_id) from ASSIGNMENTS a where a.date\_of\_issue between to\_date('" + beginDate + "','dd.mm.yyyy') and to\_date('" + endDate + "','dd.mm.yyyy') and a.deadline < a.execution\_date"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дата находиться между указаных границ

if (Int32.Parse(cmd.ExecuteScalar().ToString()) > 0)

{

chart1.Series["s1"].Points.AddXY("Выполнено с опозданием", Int32.Parse(cmd.ExecuteScalar().ToString()));

}

cmd.CommandText = "select COUNT(a.assingment\_id) from ASSIGNMENTS a where a.date\_of\_issue between to\_date('" + beginDate + "','dd.mm.yyyy') and to\_date('" + endDate + "','dd.mm.yyyy') and a.deadline >= a.execution\_date"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дата находиться между указаных границ

if (Int32.Parse(cmd.ExecuteScalar().ToString()) > 0)

{

chart1.Series["s1"].Points.AddXY("Выполнено без опоздания", Int32.Parse(cmd.ExecuteScalar().ToString()));

}

cmd.CommandText = "select COUNT(a.assingment\_id) from ASSIGNMENTS a where a.date\_of\_issue between to\_date('" + beginDate + "','dd.mm.yyyy') and to\_date('" + endDate + "','dd.mm.yyyy') and a.execution\_date is null and a.deadline >= to\_date('" + System.DateTime.Now.ToString("dd.MM.yyyy") + "','dd.mm.yyyy')"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дата находиться между указаных границ

if (Int32.Parse(cmd.ExecuteScalar().ToString()) > 0)

{

chart1.Series["s1"].Points.AddXY("Предстоит выполнить", Int32.Parse(cmd.ExecuteScalar().ToString()));

}

cmd.CommandText = "select COUNT(a.assingment\_id) from ASSIGNMENTS a where a.date\_of\_issue between to\_date('" + beginDate + "','dd.mm.yyyy') and to\_date('" + endDate + "','dd.mm.yyyy') and a.execution\_date is null and a.deadline < to\_date('" + System.DateTime.Now.ToString("dd.MM.yyyy") + "','dd.mm.yyyy')"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дата находиться между указаных границ

if (Int32.Parse(cmd.ExecuteScalar().ToString()) > 0)

{

chart1.Series["s1"].Points.AddXY("Дедлайн просрочен", Int32.Parse(cmd.ExecuteScalar().ToString()));

}

chart1.Series["s1"].Label = "#PERCENT{0.00 %} (#VALY)";

chart1.Series["s1"].LegendText = "#VALX";

label1.Text = beginDate + " - " + endDate;

connection.Close(); //Отключаемся от БД

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

}

}

AssignmentsPeriod

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class AssignmentsPeriod : Form

{

public AssignmentsPeriod()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

ASSIGN\_report.SetBeginDate(textBox1.Text); //Устанавлием нижнюю границу периода

ASSIGN\_report.SetEndDate(textBox2.Text); //Устанавлием верхнюю границу периода

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

this.Dispose(); //Высвобождаем ресурсы

}

}

}

AssignmentsDate

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class AssignmentsDate : Form

{

public AssignmentsDate()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

ASSIGN\_report.SetDate(textBox1.Text); //Устанавливаем дату для поиска указаний

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

this.Dispose(); //Высвобождаем ресурсы

}

}

}

ASSIGN\_report

using System;

using System.ComponentModel;

using System.Data;

using System.Windows.Forms;

using Microsoft.Reporting.WinForms;

using Oracle.DataAccess.Client;

namespace WindowsFormsApplication5

{

public partial class ASSIGN\_report : Form

{

private static string date, beginDate, endDate; //Переменные для создания отчета за определенную дату или за период

/\*

Set и Get методы для переменных

\*/

public static string GetDate()

{

return date;

}

public static void SetDate(string newDate)

{

date = newDate;

}

public static string GetEndDate()

{

return endDate;

}

public static void SetBeginDate(string newDate)

{

beginDate = newDate;

}

public static string GetBeginDate()

{

return beginDate;

}

public static void SetEndDate(string newDate)

{

endDate = newDate;

}

public ASSIGN\_report()

{

InitializeComponent();

reportViewer1.ReportRefresh += new CancelEventHandler(reportViewer1\_ReportRefresh); //Обработчик событий при нажатии на кнопку Refresh

}

void reportViewer1\_ReportRefresh(object sender, CancelEventArgs e)

{

try

{

OracleConnection connection = new OracleConnection();

connection.ConnectionString = Properties.Settings.Default.ConnectionString;

connection.Open(); //Подключаемся к Базе Данных

OracleCommand cmd = connection.CreateCommand(); //Создаем SQL команду

cmd.CommandText = "select \* from ASSIGNMENTS"; //Добавляем текст команды. Отбираем все элементы из таблицы

DataTable dt = new DataTable(); //Создаем переменную таблицы

OracleDataAdapter da = new OracleDataAdapter(cmd);

da.Fill(dt); //Заполняем таблицу

connection.Close(); //Отключаемся от БД

this.reportViewer1.LocalReport.DataSources.Clear(); //Удаляем все элементы из reportViewer1

this.reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("DataSet1", dt)); //Добавляем в reportViewer1 данные с таблицы dt

this.reportViewer1.RefreshReport(); //Обновляем reportViewer1

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

private void ASSIGN\_report\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "DataSet1.ASSIGNMENTS". При необходимости она может быть перемещена или удалена.

try

{

this.ASSIGNMENTSTableAdapter.Fill(this.DataSet1.ASSIGNMENTS);

this.reportViewer1.RefreshReport();

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

private void button1\_Click(object sender, EventArgs e)

{

try

{

OracleConnection connection = new OracleConnection();

connection.ConnectionString = Properties.Settings.Default.ConnectionString;

connection.Open(); //Подключаемся к Базе Данных

OracleCommand cmd = connection.CreateCommand(); //Создаем SQL команду

cmd.CommandText = "select \* from ASSIGNMENTS a where a.date\_of\_issue = to\_date('" + System.DateTime.Now.ToString("dd.MM.yyyy") + "','dd.mm.yyyy')"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дата равна сегодняшней

DataTable dt = new DataTable();

OracleDataAdapter da = new OracleDataAdapter(cmd);

da.Fill(dt); //Заполняем таблицу

connection.Close(); //Отключаемся от БД

this.reportViewer1.LocalReport.DataSources.Clear(); //Удаляем все элементы из reportViewer1

this.reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("DataSet1", dt)); //Добавляем в reportViewer1 данные с таблицы dt

this.reportViewer1.RefreshReport(); //Обновляем reportViewer1

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

private void button2\_Click(object sender, EventArgs e)

{

AssignmentsDate f = new AssignmentsDate();

f.ShowDialog(this);

try

{

OracleConnection connection = new OracleConnection();

connection.ConnectionString = Properties.Settings.Default.ConnectionString;

connection.Open(); //Подключаемся к Базе Данных

OracleCommand cmd = connection.CreateCommand(); //Создаем SQL команду

cmd.CommandText = "select \* from ASSIGNMENTS a where a.date\_of\_issue = to\_date('" + date + "','dd.mm.yyyy')"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дата равна указаной

DataTable dt = new DataTable();

OracleDataAdapter da = new OracleDataAdapter(cmd);

da.Fill(dt); //Заполняем таблицу

connection.Close(); //Отключаемся от БД

this.reportViewer1.LocalReport.DataSources.Clear(); //Удаляем все элементы из reportViewer1

this.reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("DataSet1", dt)); //Добавляем в reportViewer1 данные с таблицы dt

this.reportViewer1.RefreshReport(); //Обновляем reportViewer1

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

private void button3\_Click(object sender, EventArgs e)

{

AssignmentsPeriod f = new AssignmentsPeriod();

f.ShowDialog(this);

try

{

OracleConnection connection = new OracleConnection();

connection.ConnectionString = Properties.Settings.Default.ConnectionString;

connection.Open(); //Подключаемся к Базе Данных

OracleCommand cmd = connection.CreateCommand(); //Создаем SQL команду

cmd.CommandText = "select \* from ASSIGNMENTS a where a.date\_of\_issue between to\_date('" + beginDate + "','dd.mm.yyyy') and to\_date('" + endDate + "','dd.mm.yyyy')"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дата находиться между указаных границ

DataTable dt = new DataTable();

OracleDataAdapter da = new OracleDataAdapter(cmd);

da.Fill(dt); //Заполняем таблицу

connection.Close(); //Отключаемся от БД

this.reportViewer1.LocalReport.DataSources.Clear(); //Удаляем все элементы из reportViewer1

this.reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("DataSet1", dt)); //Добавляем в reportViewer1 данные с таблицы dt

this.reportViewer1.RefreshReport(); //Обновляем reportViewer1

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

private void button4\_Click(object sender, EventArgs e)

{

try

{

OracleConnection connection = new OracleConnection();

connection.ConnectionString = Properties.Settings.Default.ConnectionString;

connection.Open(); //Подключаемся к Базе Данных

OracleCommand cmd = connection.CreateCommand(); //Создаем SQL команду

cmd.CommandText = "select \* from ASSIGNMENTS a where a.deadline < to\_date('" + System.DateTime.Now.AddDays(7).ToString("dd.MM.yyyy") + "','dd.mm.yyyy') and a.deadline >= to\_date('" + System.DateTime.Now.ToString("dd.MM.yyyy") + "','dd.mm.yyyy')"; //Добавляем текст команды. Отбираем элементы из таблицы в которых дедлайн находиться в пределах 7 дней от сегодняшней даты

DataTable dt = new DataTable();

OracleDataAdapter da = new OracleDataAdapter(cmd);

da.Fill(dt); //Заполняем таблицу

connection.Close(); //Отключаемся от БД

this.reportViewer1.LocalReport.DataSources.Clear(); //Удаляем все элементы из reportViewer1

this.reportViewer1.LocalReport.DataSources.Add(new ReportDataSource("DataSet1", dt)); //Добавляем в reportViewer1 данные с таблицы dt

this.reportViewer1.RefreshReport(); //Обновляем reportViewer1

}

catch (Exception exc)

{

MessageBox.Show(exc.Message, "Error");

}

}

}

}

EMP\_report

using System;

using System.Windows.Forms;

namespace WindowsFormsApplication5

{

public partial class EMP\_report : Form

{

public EMP\_report()

{

InitializeComponent();

}

private void EMP\_report\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "DataSet1.EMPLOYEE". При необходимости она может быть перемещена или удалена.

this.EMPLOYEETableAdapter.Fill(this.DataSet1.EMPLOYEE);

this.reportViewer1.RefreshReport();

}

}

}

# ЛІТЕРАТУРА

1. https://docs.oracle.com/cd/B28359\_01/server.111/b28310/ds\_concepts001.htm#ADMIN12074 [Електронний ресурс]
2. <https://docs.oracle.com/cd/B19306_01/server.102/b14228/toc.htm> [Електронний ресурс]
3. Маклаков С.В. Моделирование бизнес-процессов с AllFusion PM. – 2-е изд., испр. и дополн. – М.: Издательство Диалог-МИФИ, 2008. – 224 с.
4. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. – М.: Вильямс, 2013. – 762 c.
5. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – 8-е издание. – М.: Вильямс, 2015. – 672 с.
6. Рик Гринвальд, Роберт Стаковьяк, Гэри Додж, Дэвид Кляйн, Бен Шапиро, Кристофер Дж. Челья Программирование баз данных Oracle для профессионалов = Professional Oracle Programming — М.: [«Диалектика»](http://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2007. — 784 с. — [ISBN 978-5-8459-1138-4](http://ru.wikipedia.org/wiki/%D0%A1%D0%BB%D1%83%D0%B6%D0%B5%D0%B1%D0%BD%D0%B0%D1%8F:BookSources/9785845911384).
7. Сервер Oracle 10G: Администрирование баз данных, Учебное пособие, Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.
8. Ms Visual Studio 2017 Режим електронного доступу –[https://www.microsoft.com/Ukraine/News/default.mspx].
9. Избачков Ю.С. Информационные системы: Учебник для вузов / Ю.С. Избачков, В.Н. Петров. – 2-е изд. – СПБ.: Питер, 2005. – 656 с.