Міністерство освіти та науки України

Національний технічний університет України “КПІ ім. Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

**ЗВІТ**

До залікової роботи

**з дисципліни**

**«Веб-орієнтована розробка програмного забезпечення»**

**Виконала:** студентка групи ІС-62

Путова Юлія Андріївна

**Перевірила:** доцент кафедри АСОІУ

Ліщук Катерина Ігорівна

Київ 2018

**Звіт**

**1. Завдання**

Варіант 2.

Домашня відеотека. В БД зберігається інформація про домашню відеотеку: фільми, актори, режисери.

Для фільмів необхідно зберігати:

- назву;

- імена акторів;

- дату виходу;

- країну, в якій випущений фільм.

Для акторів і режисерів необхідно зберігати:

- ПІБ;

- дату народження.

Завдання:

- Знайти всі фільми, що вийшли на екран у поточному і минулому році.

- Вивести інформацію про акторів, що знімалися в заданому фільмі.

- Вивести інформацію про акторів, що знімалися як мінімум в N фільмах.

- Вивести інформацію про акторів, які були режисерами хоча б одного з фільмів.

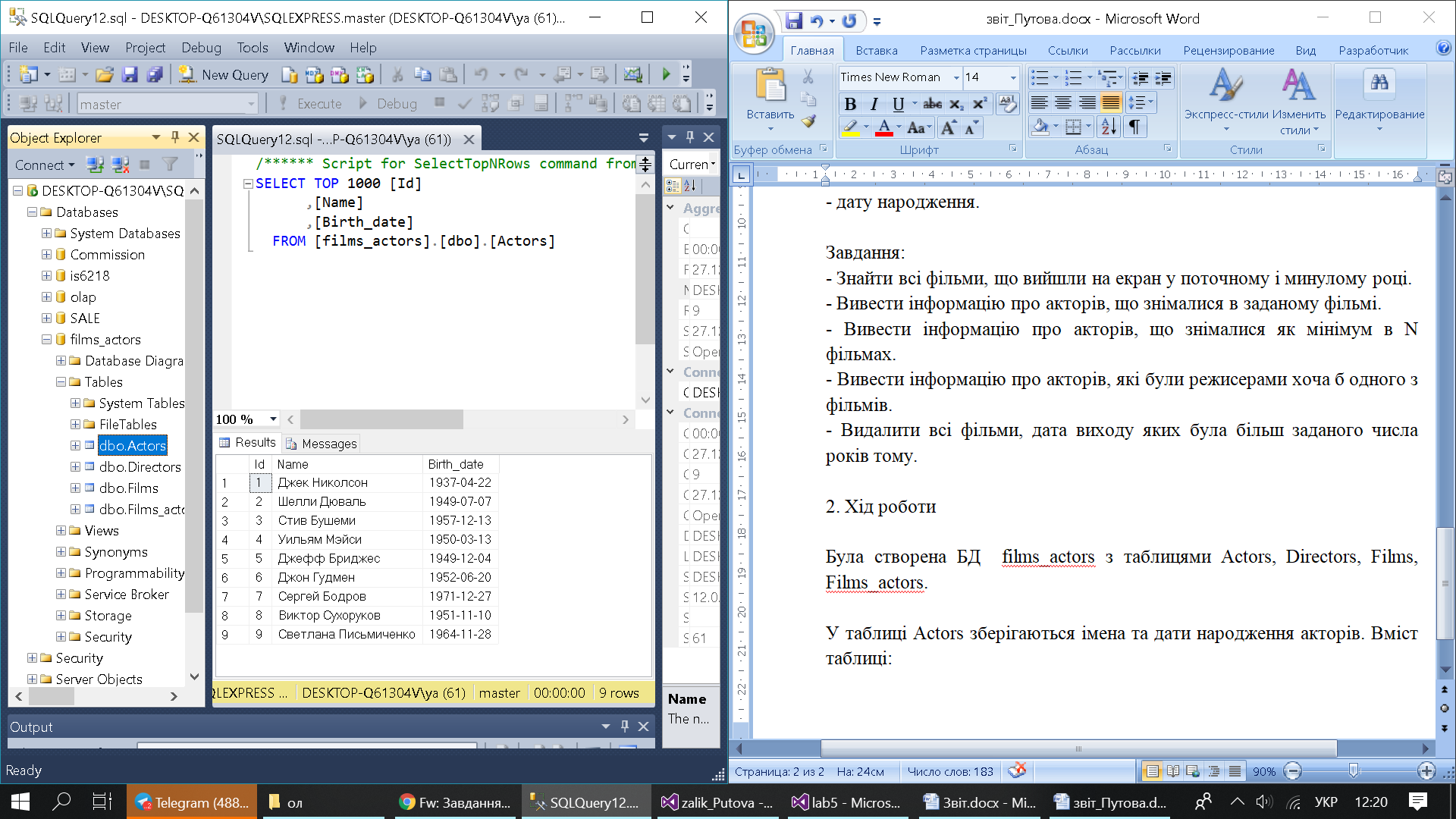
- Видалити всі фільми, дата виходу яких була більш заданого числа років тому.

2. Хід роботи

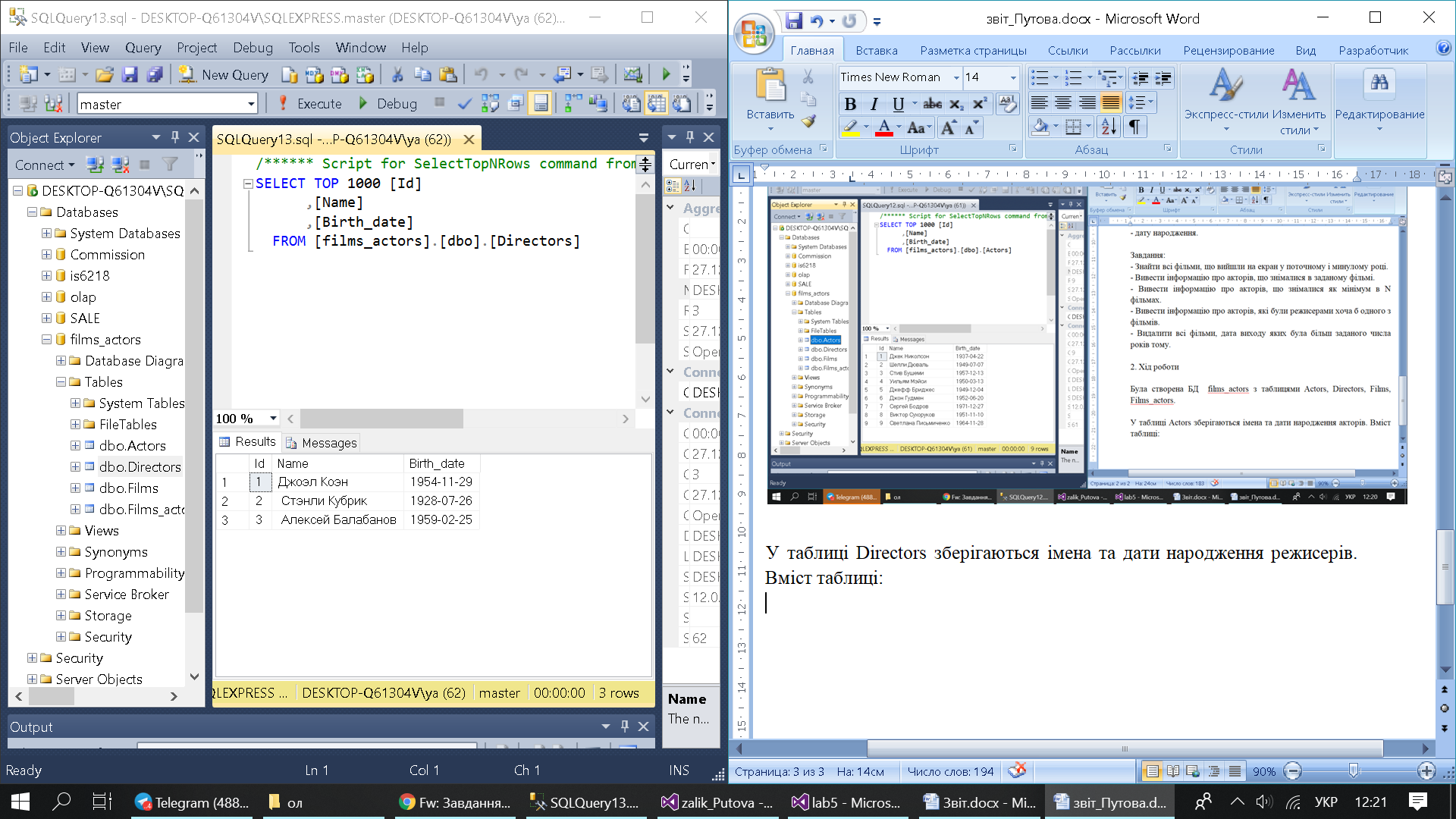
**Створення БД**

Була створена БД films\_actors з таблицями Actors, Directors, Films, Films\_actors.

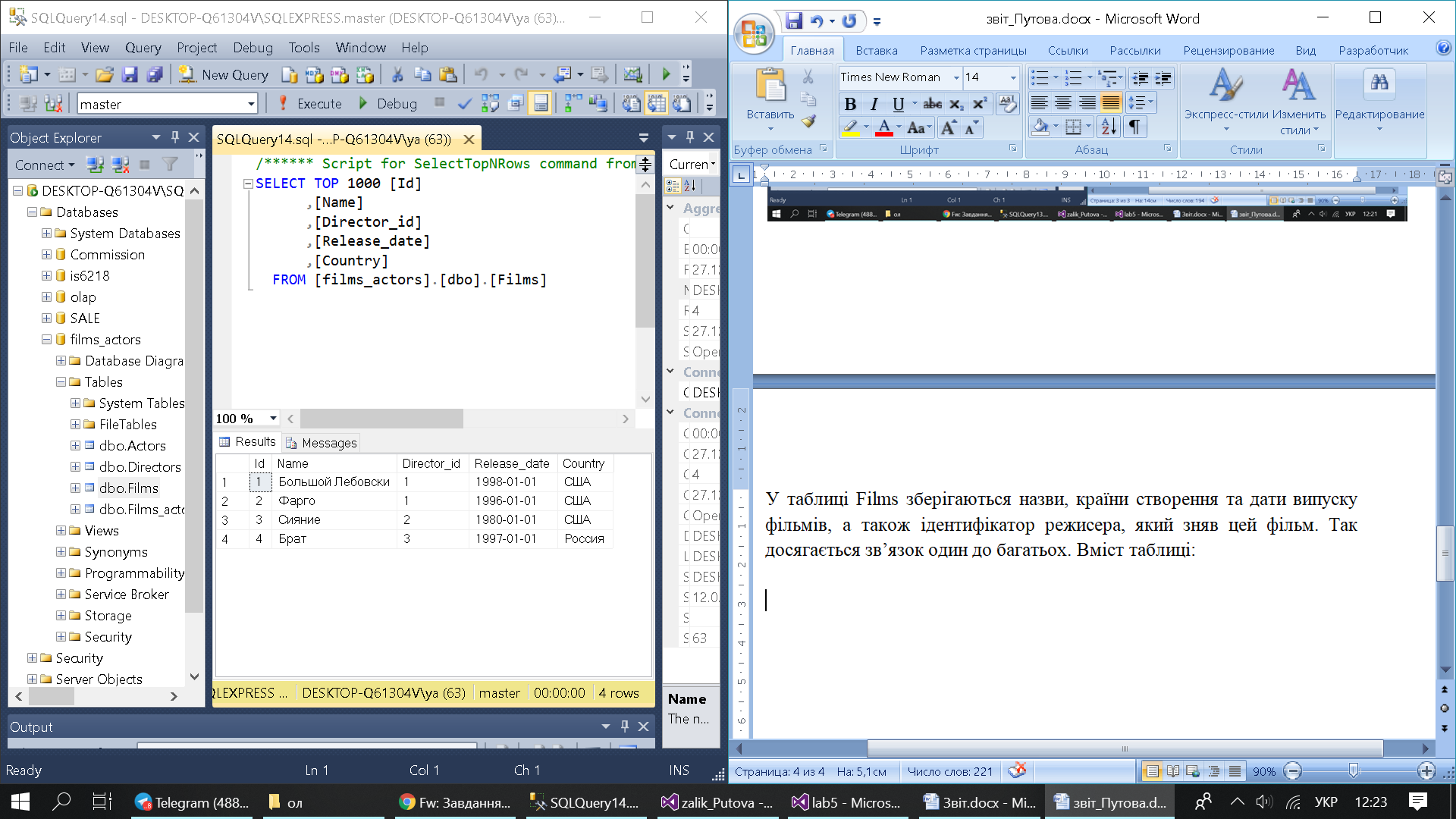
1) У таблиці Actors зберігаються імена та дати народження акторів. Вміст таблиці:



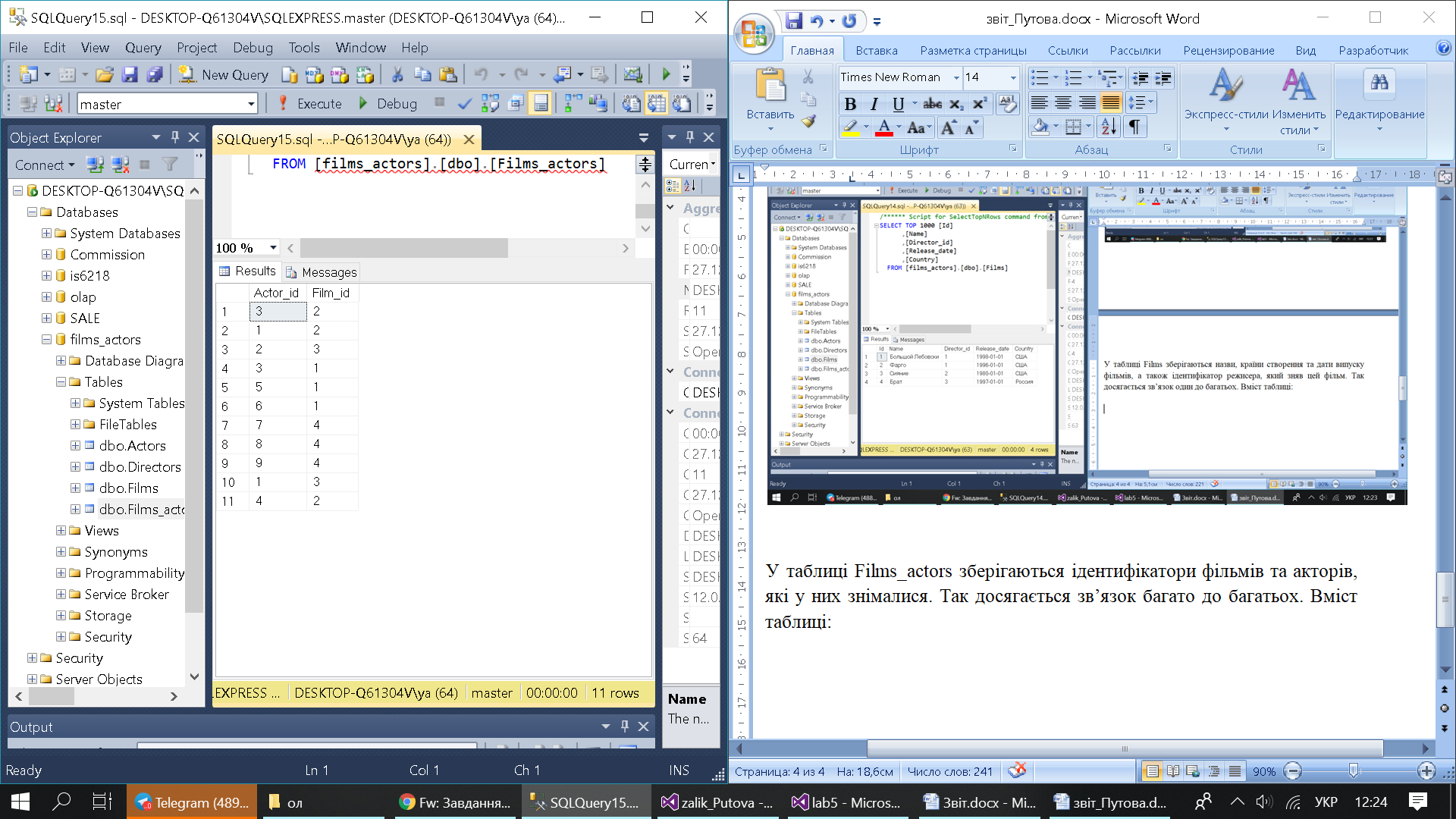
2) У таблиці Directors зберігаються імена та дати народження режисерів. Вміст таблиці:



3) У таблиці Films зберігаються назви, країни створення та дати випуску фільмів, а також ідентифікатор режисера, який зняв цей фільм. Так досягається зв’язок один до багатьох. Вміст таблиці:

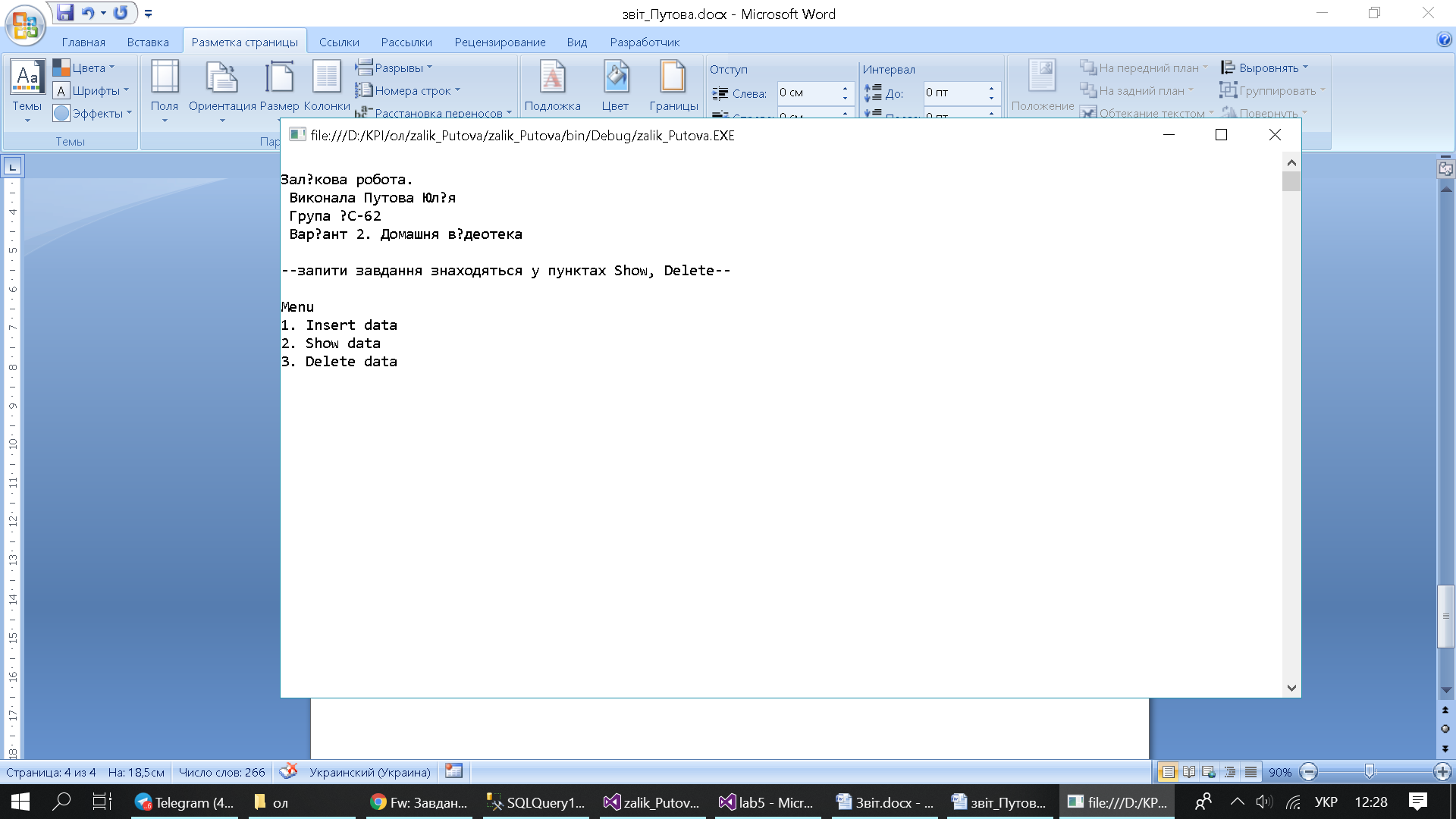


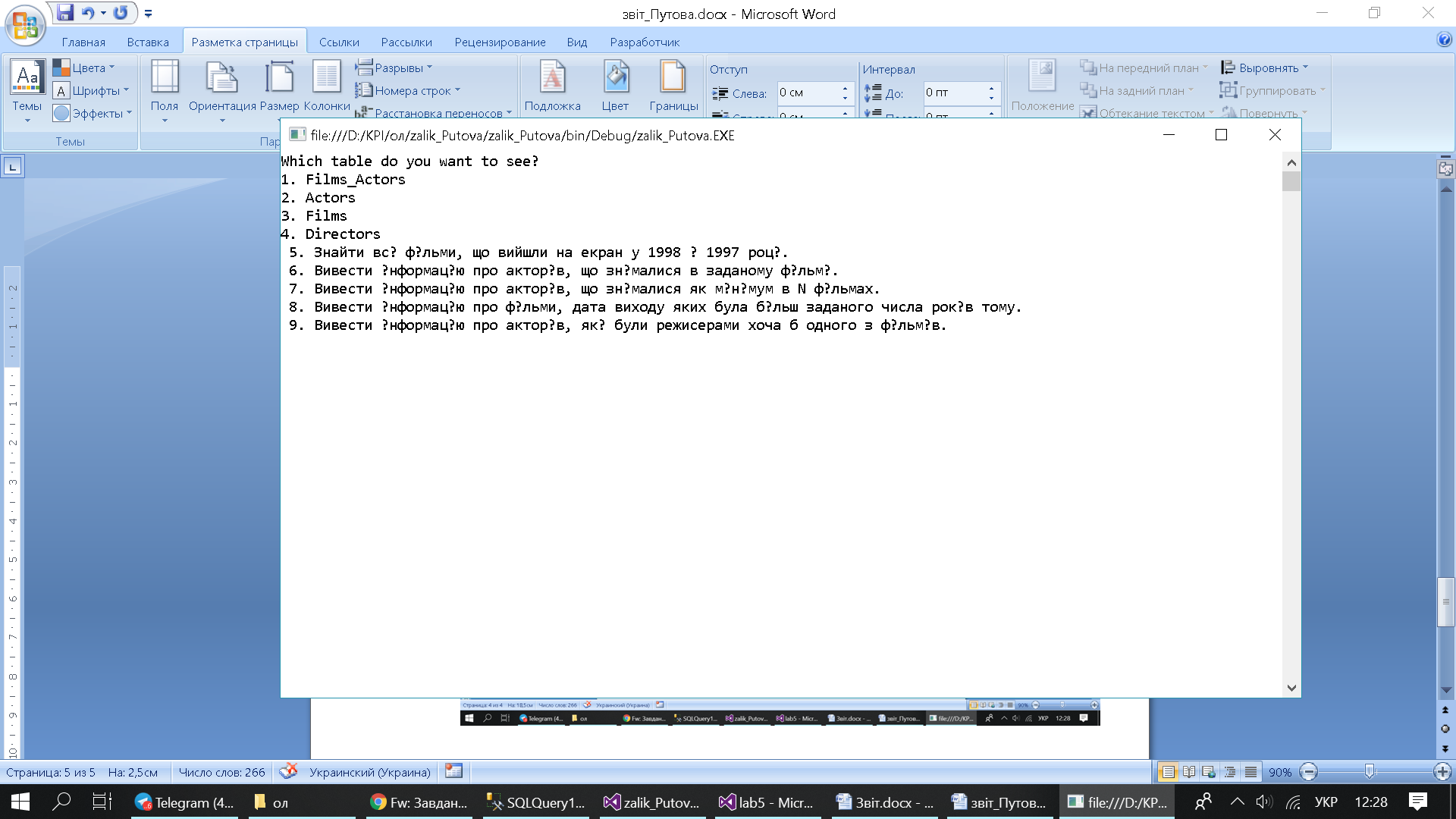
4) У таблиці Films\_actors зберігаються ідентифікатори фільмів та акторів, які у них знімалися. Так досягається зв’язок багато до багатьох. Вміст таблиці:



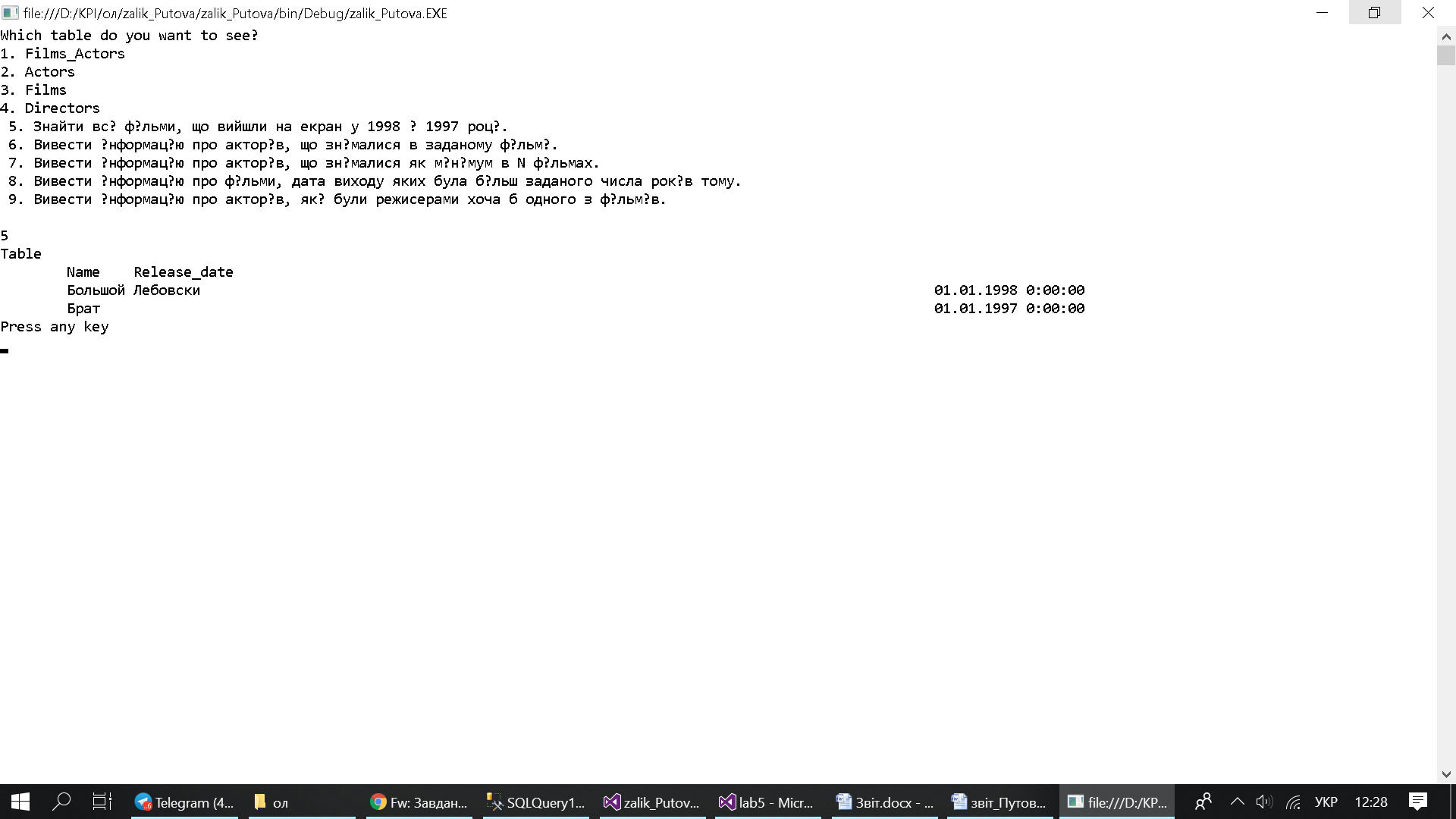
У створеному консольному застосуванні були створені класи Insert\_command, Delete\_command та Select\_command у яких зберігаються мтоди для виконання SQL запитів.

**Приклад виконання програми:**

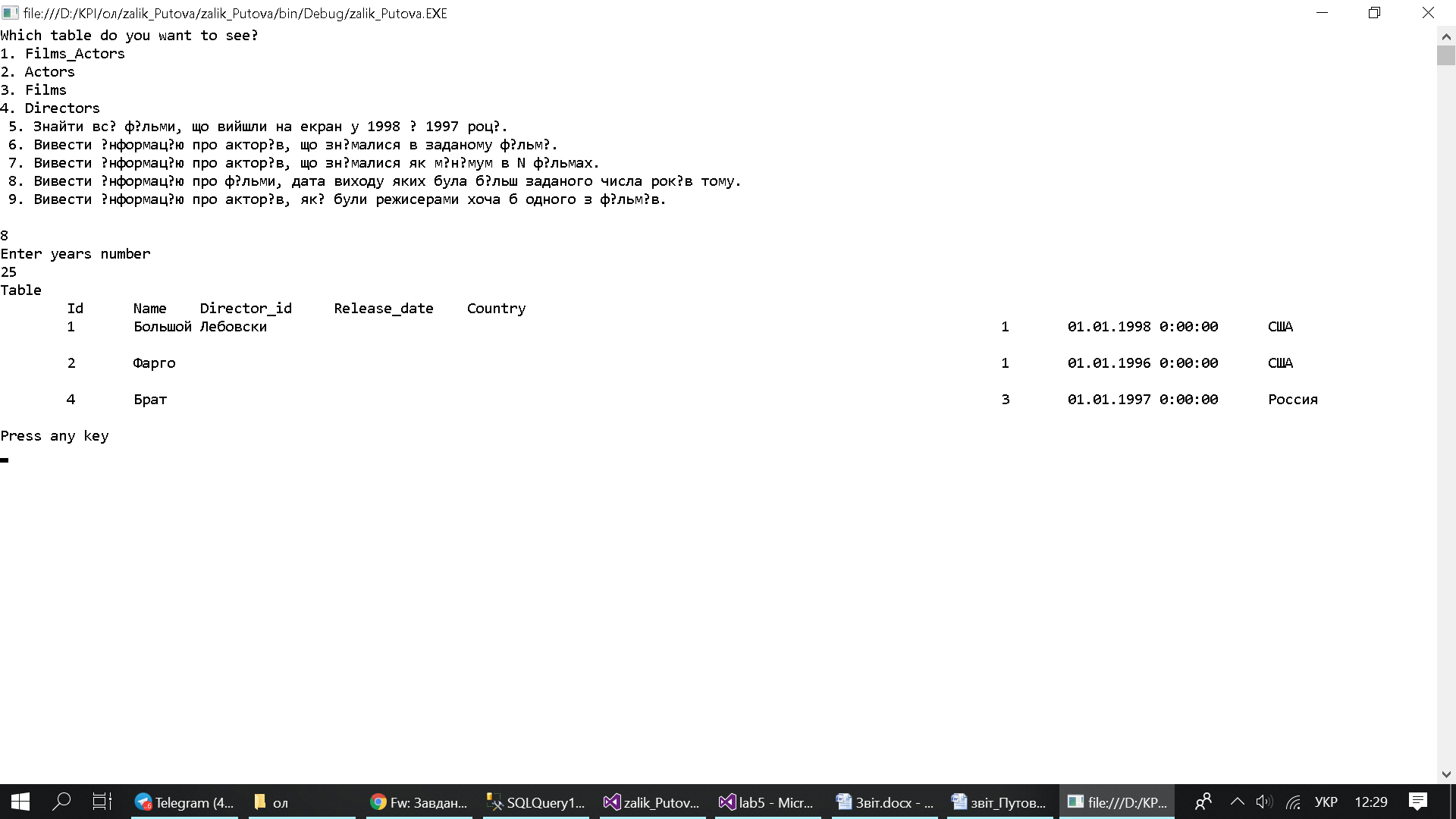




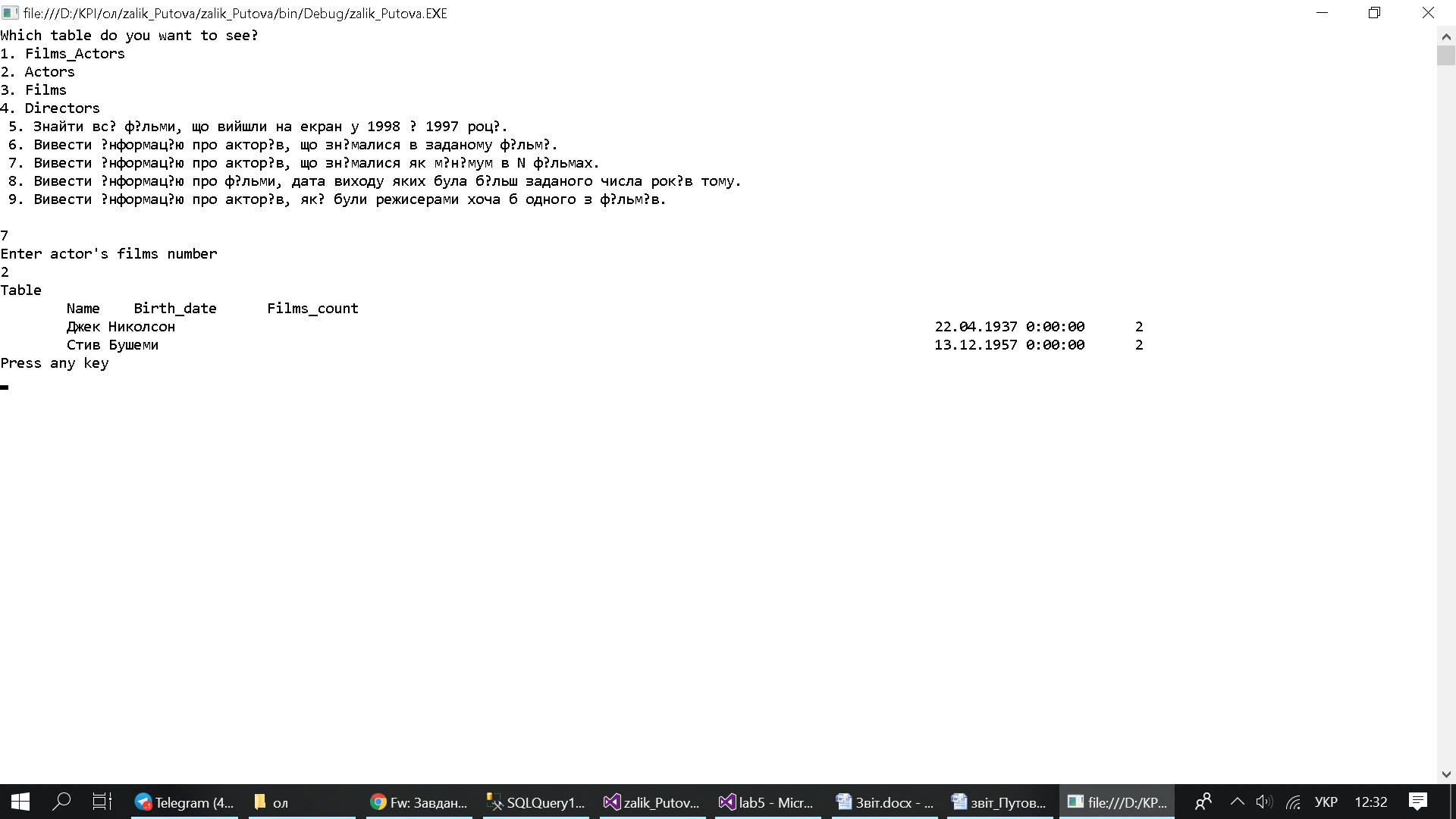
* Знайти всі фільми, що вийшли на екран у поточному і минулому році.



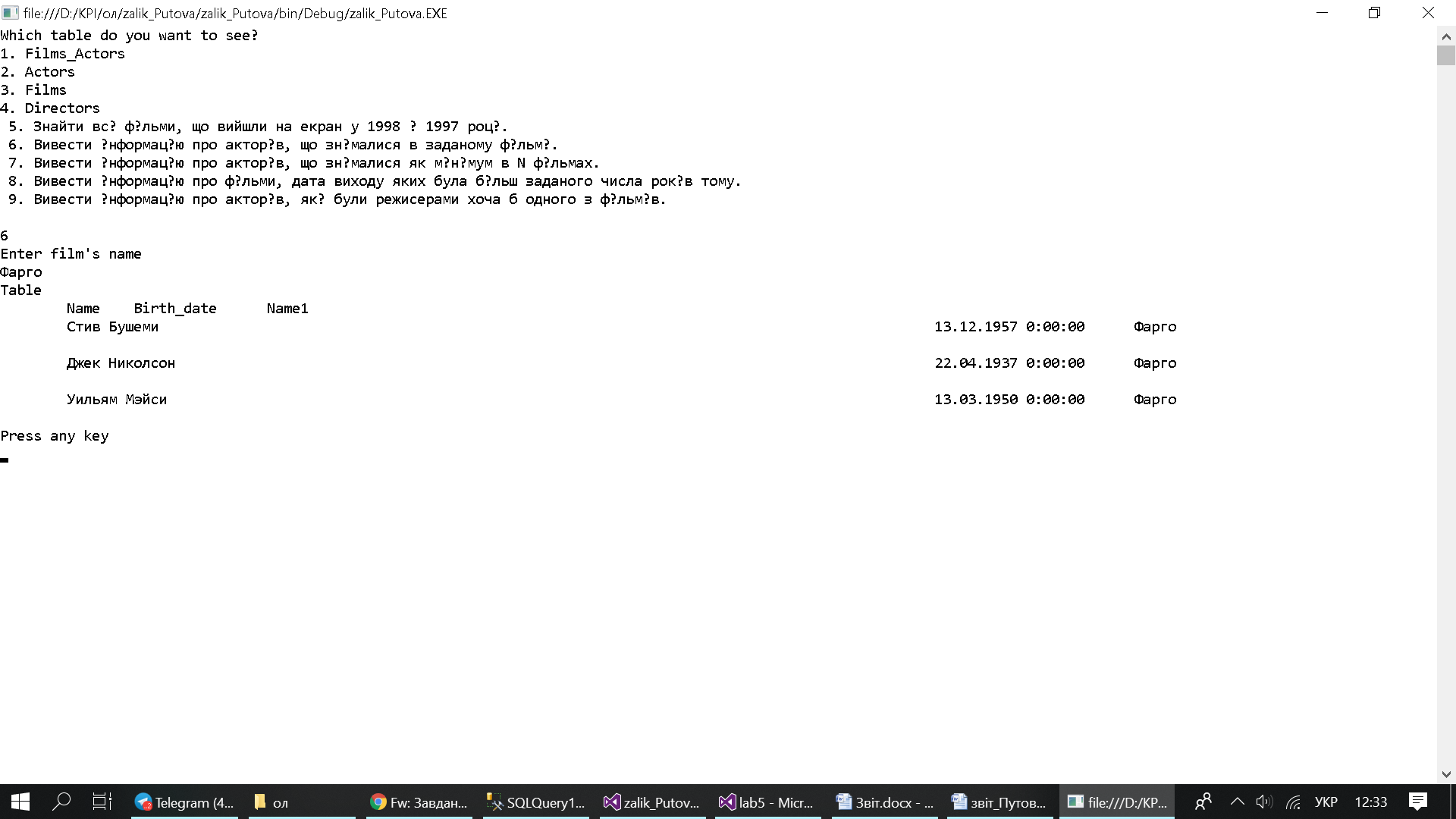
* Видалити всі фільми, дата виходу яких була більш заданого числа років тому.



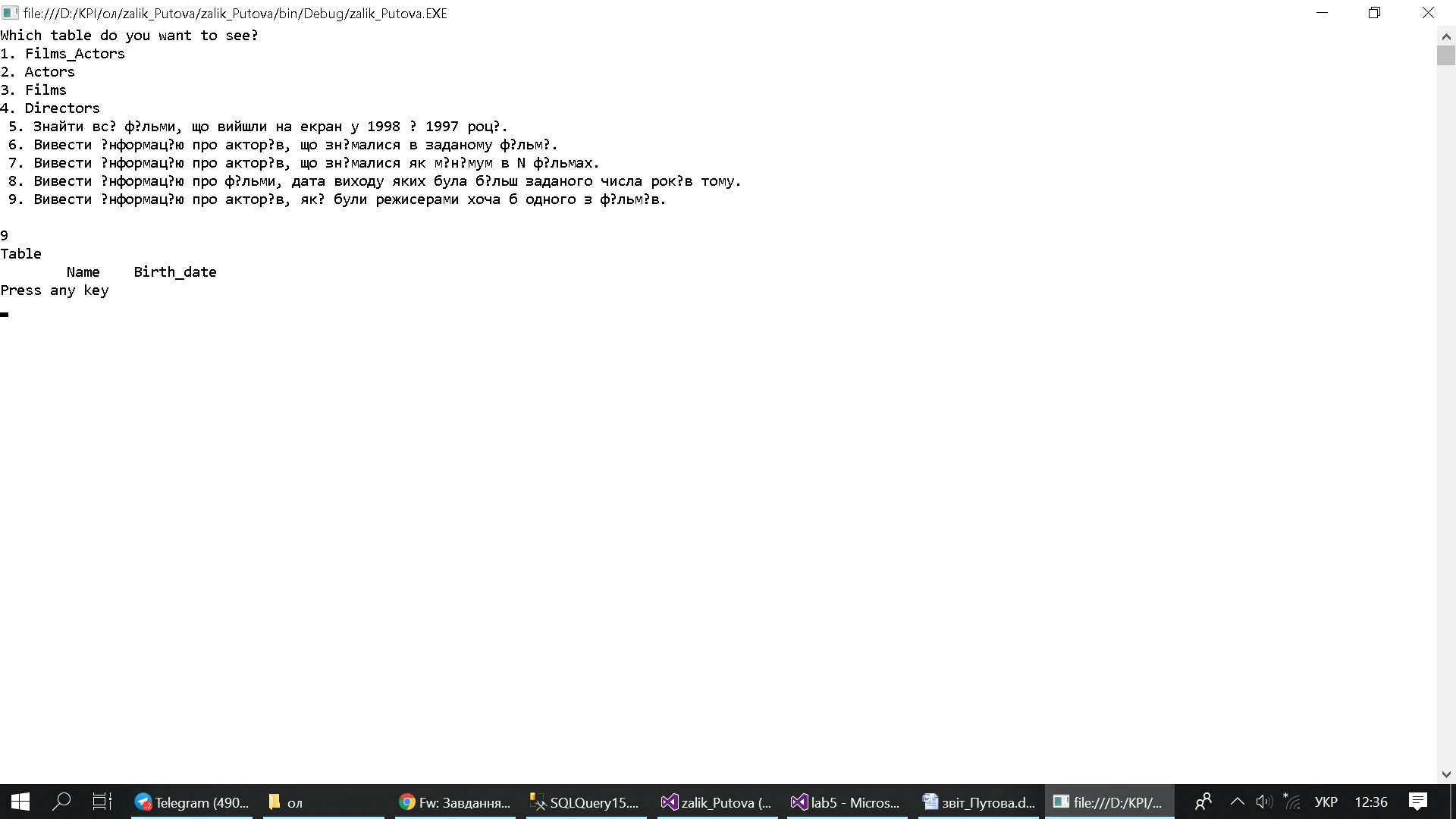
* Вивести інформацію про акторів, що знімалися як мінімум в N фільмах.



* Вивести інформацію про акторів, що знімалися в заданому фільмі.



* Вивести інформацію про акторів, які були режисерами хоча б одного з фільмів.



**Лістинг програми:**

class Select\_command

{

string sqlExpression;

string connectionString;

public Select\_command(string connectString)

{

connectionString = connectString;

}

public void menu()

{

Console.Clear();

Console.WriteLine("Which table do you want to see?\n1. Films\_Actors\n2. Actors\n3. Films\n4. Directors\n "+

"5. Знайти всі фільми, що вийшли на екран у 1998 і 1997 році.\n "+

"6. Вивести інформацію про акторів, що знімалися в заданому фільмі.\n "+

"7. Вивести інформацію про акторів, що знімалися як мінімум в N фільмах.\n "+

"8. Вивести інформацію про фільми, дата виходу яких була більш заданого числа років тому.\n "+

"9. Вивести інформацію про акторів, які були режисерами хоча б одного з фільмів.\n ");

string answ = Console.ReadLine();

switch(answ)

{

case "1": sqlExpression = "SELECT \* FROM Films\_Actors"; selection(); break;

case "2": sqlExpression = "SELECT \* FROM Actors"; selection(); break;

case "3": sqlExpression = "SELECT \* FROM Films"; selection(); break;

case "4": sqlExpression = "SELECT \* FROM Directors"; selection(); break;

case "5": sqlExpression = "SELECT Films.Name, Films.Release\_date FROM Films WHERE year(Release\_date)=1997 OR year(Release\_date)=1998"; selection(); break;

case "6": Console.WriteLine("Enter film's name");

string name = Console.ReadLine();

sqlExpression = "SELECT Actors.Name, Actors.Birth\_date, Films.Name FROM Actors "

+" INNER JOIN Films\_actors ON Actors.Id=Films\_actors.Actor\_id"

+" INNER JOIN Films ON Films\_actors.Film\_id=Films.Id"

+" WHERE Films.Name="+"'"+name+"'"; selection(); break;

case "7": Console.WriteLine("Enter actor's films number");

string n = Console.ReadLine();

sqlExpression = "SELECT Actors.Name, Actors.Birth\_date, Count(Films.Name) AS Films\_count FROM Actors "

+" INNER JOIN Films\_actors ON Actors.Id=Films\_actors.Actor\_id "

+" INNER JOIN Films ON Films\_actors.Film\_id=Films.Id "

+" GROUP BY Actors.Name, Actors.Birth\_date"

+" HAVING Count(Films.Name)>="+n; selection(); break;

case "8": Console.WriteLine("Enter years number");

string m = Console.ReadLine();

sqlExpression = "SELECT \* FROM FILMS WHERE"+

" year(Release\_date)+"+m+">=2018"; selection(); break;

case "9": sqlExpression = "SELECT Actors.Name, Actors.Birth\_date FROM Actors "

+ " INNER JOIN Films\_actors ON Films\_actors.Actor\_id=Actors.Id "

+ " INNER JOIN Films ON Films\_actors.Film\_id=Films.id "

+ " INNER JOIN Directors ON Films.Director\_id=Directors.Id "

+ " WHERE Directors.Name=Actors.Name"; selection(); break;

default: break;

}

}

public void selection()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlDataAdapter adapter = new SqlDataAdapter(sqlExpression, connection);

DataSet ds = new DataSet();

adapter.Fill(ds);

foreach (DataTable dt in ds.Tables)

{

Console.WriteLine(dt.TableName);

foreach (DataColumn column in dt.Columns)

Console.Write("\t{0}", column.ColumnName);

Console.WriteLine();

foreach (DataRow row in dt.Rows)

{

var cells = row.ItemArray;

foreach (object cell in cells)

Console.Write("\t{0}", cell);

Console.WriteLine();

}

}

}

Console.WriteLine("Press any key");

Console.ReadLine();

}

}

class Delete\_command

{

string sqlExpression;

string connectionString;

public Delete\_command(string connectString)

{

connectionString = connectString;

}

public void menu()

{

Console.Clear();

Console.WriteLine("Which table do you want delete?\n0.Delete Row \n1. All data\n2.All data in Films\_Actors\n3.All data in Actors\n4.All data in Films\n5. All data in Directors\n6.Видалити всі фільми, дата виходу яких була більш заданого числа років тому.");

string answ = Console.ReadLine();

switch (answ)

{

case "0": DeleteRow(); break;

case "1": DeleteTestData(); break;

case "2": sqlExpression = "Delete from Films\_Actors";DeleteTable(sqlExpression); break;

case "3": sqlExpression = "Delete from Actors"; DeleteTable(sqlExpression); break;

case "4": sqlExpression = "Delete from Films"; DeleteTable(sqlExpression); break;

case "5": sqlExpression = "Delete from Directors"; DeleteTable(sqlExpression); break;

case "6": Console.WriteLine("Enter actor's films number");

string m = Console.ReadLine();

sqlExpression = "DELETE FROM FILMS WHERE" +

"year(Release\_date)+" + m + ">=2018"; DeleteTable(sqlExpression); break;

default: break;

}

}

public void DeleteRow()

{

Console.WriteLine("Enter Table name ");

string table = Console.ReadLine();

Console.WriteLine("Enter condition");

string cond = Console.ReadLine();

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

sqlExpression = String.Concat("Delete from ", table, " where ", cond);

SqlCommand command1 = new SqlCommand(sqlExpression, connection);

command1.ExecuteNonQuery();

}

}

public void DeleteTable(string sqlExpression)

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command1 = new SqlCommand(sqlExpression, connection);

command1.ExecuteNonQuery();

connection.Close();

};

}

public void DeleteTestData()

{

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

sqlExpression = String.Concat("Delete from Films\_Actors");

SqlCommand command1 = new SqlCommand(sqlExpression, connection);

command1.ExecuteNonQuery();

sqlExpression = String.Concat("Delete from Actors");

SqlCommand command2 = new SqlCommand(sqlExpression, connection);

command2.ExecuteNonQuery();

sqlExpression = String.Concat("Delete from Films");

SqlCommand command3 = new SqlCommand(sqlExpression, connection);

command3.ExecuteNonQuery();

sqlExpression = String.Concat("Delete from Directors");

SqlCommand command5 = new SqlCommand(sqlExpression, connection);

command5.ExecuteNonQuery();

connection.Close();

}

}

}

class Program

{

static void Main(string[] args)

{

string connectionString = @"Data Source=.\SQLEXPRESS;Initial Catalog=films\_actors;Integrated Security=True";

Select\_command select = new Select\_command(connectionString);

Insert\_command insert = new Insert\_command(connectionString);

Delete\_command delete = new Delete\_command(connectionString);

menu(select, insert, delete, connectionString);

}

static void menu(Select\_command select, Insert\_command insert, Delete\_command delete, string connectionString)

{

Console.Clear();

Console.WriteLine("\nЗалікова робота.\n Виконала Путова Юлія\n Група ІС-62\n Варіант 2. Домашня відеотека\n\n--запити завдання знаходяться у пунктах Show, Delete--\n");

Console.WriteLine("Menu\n1. Insert data\n2. Show data\n3. Delete data");

string answ = Console.ReadLine();

switch (answ)

{

case "1": insert.menu(); break;

case "2": select.menu(); break;

case "3": delete.menu(); break;

default: break;

}

menu(select, insert, delete, connectionString);

}

}