Міністерство освіти і науки України Національний технічний університет  України «Київський політехнічний інститут

імені Ігоря Сікорського»

Кафедра автоматизації теплоенергетичних процесів

**Лабораторна робота №4**

з дисципліни «Проектування та розробка баз даних »

Технологія доступу до баз даних ADO.NET

Виконала:

Аксьонова Я.В.

Група: ТА-13, НН ІАТЕ

Перевірив:

Любицький С.В.

Київ – 2023

**3. Приклад створення підключення і виконання команди DML (INSERT,UPDATE або DELETE)**

var FilmName = FilmNameBox.Text;

var Duration = DurationBox.Text;

var Producer = ProducerBox.Text;

var Studio = StudioBox.Text;

var SeanseQ = SeanseQBox.Text;

var Genre = GenreBox.Text;

var IdFilm = IdFilmBox.Text;

string sqlExpression = $"INSERT INTO Film( FilmName, Duration, Producer, Studio, SeanceQuantity, Genre, FilmID)VALUES('{FilmName}','{Duration}', '{Producer}','{Studio}','{SeanseQ}', '{Genre}', '{IdFilm}')";

int number = 0;

SqlConnection connection = new SqlConnection(connectionString);

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

try

{

number = command.ExecuteNonQuery();

}

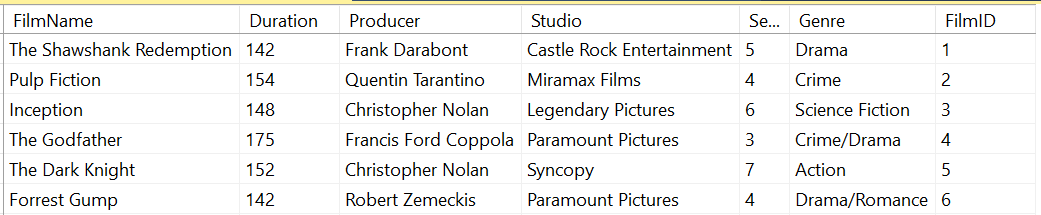
catch (Exception ex)

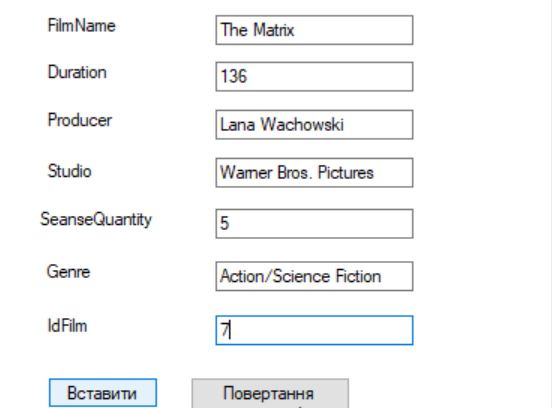
{

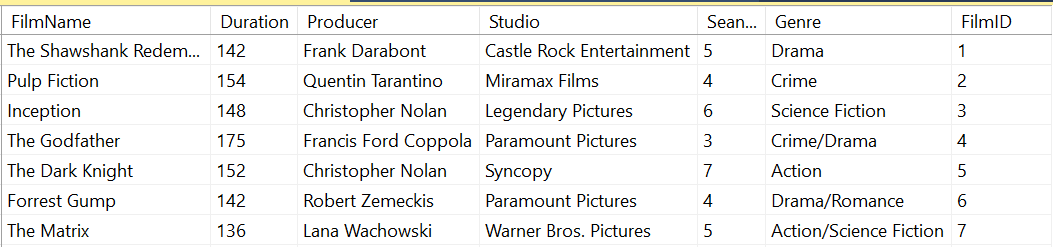
MessageBox.Show(ex.Message, "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

MessageBox.Show($"Додано об'єктів: {number} ", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

connection.Close();

****

****

//Оновлення клієнта(UPDATE)

var FilmName = FilmNameBox.Text;

var Duration = DurationBox.Text;

var Producer = ProducerBox.Text;

var Studio = StudioBox.Text;

var SeanseQ = SeanseQBox.Text;

var Genre = GenreBox.Text;

var IdFilm = IdFilmBox.Text;

string sqlExpression = $"UPDATE Film SET FilmName = '{FilmName}', Duration = '{Duration}', Producer = '{Producer}', Studio = '{Studio}', SeanceQuantity = '{SeanseQ}', Genre = '{Genre}' WHERE FilmID = '{IdFilm}'";

SqlConnection connection = new SqlConnection(connectionString);

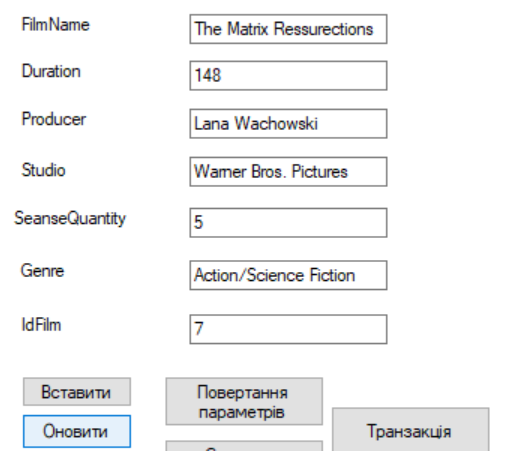
connection.Open();

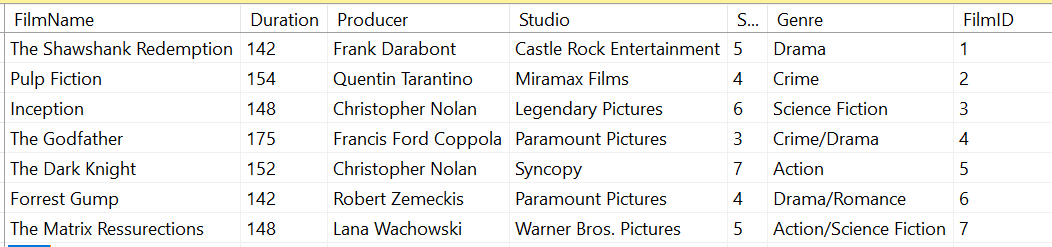
SqlCommand command = new SqlCommand(sqlExpression, connection);

int number = command.ExecuteNonQuery();

MessageBox.Show($"Оновлено об'єктів: {number} ", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

connection.Close();

****

****

//Видалення клієнта (DELETE)

var IdFilm = IdFilmBox.Text;

string sqlExpression = $"DELETE from Film WHERE FilmID = '{IdFilm}' ";

SqlConnection connection = new SqlConnection(connectionString);

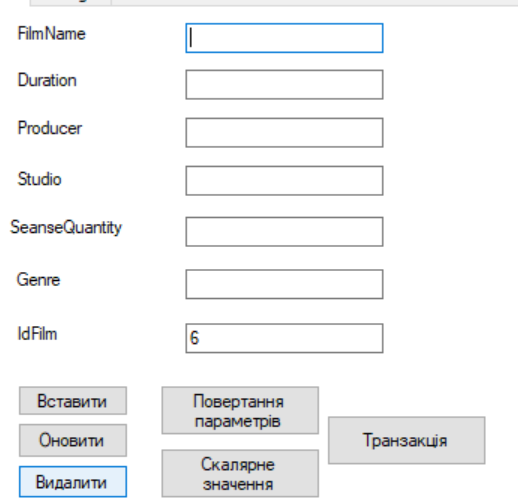
connection.Open();

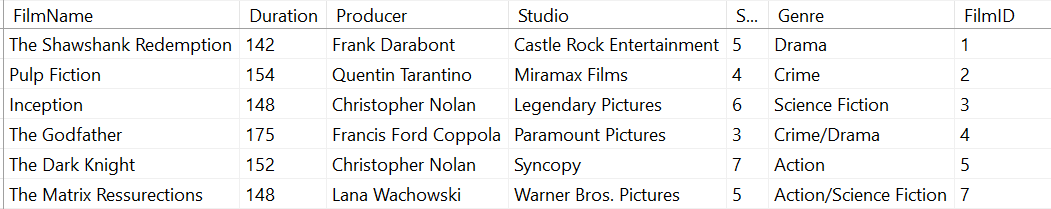
SqlCommand command = new SqlCommand(sqlExpression, connection);

int number = command.ExecuteNonQuery();

MessageBox.Show($"Видалено об'єктів: {number} ", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

connection.Close();





**4. Приклад виконання команди що повертає набір значень (DataReader)**

//виведення всіх фільмів

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

Console.WriteLine("{0} {1} {2} {3} {4} {5} {6}", reader.GetName(0), reader.GetName(1), reader.GetName(2), reader.GetName(3), reader.GetName(4), reader.GetName(5), reader.GetName(6));

while (reader.Read())

{

object FilmName = reader.GetValue(0);

object Duration = reader.GetValue(1);

object Producer = reader.GetValue(2);

object Studio = reader.GetValue(3);

object SeanceQuantity = reader.GetValue(4);

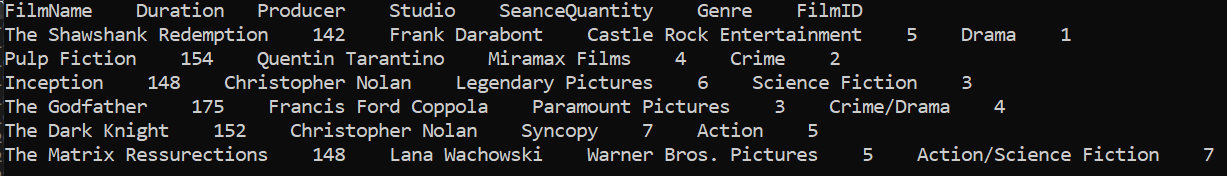
object Genre = reader.GetValue(5);

object FilmID = reader.GetValue(6);

Console.WriteLine("{0} {1} {2} {3} {4} {5} {6}", FilmName, Duration, Producer, Studio, SeanceQuantity, Genre, FilmID);

}

}

reader.Close();

**5. Приклад виконання команди що повертає скалярне значення.**

//команда , що повертає кількість фільмів та максимальну тривалість

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

string sqlExpression = "SELECT COUNT(\*) FROM Film";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlCommand command = new SqlCommand(sqlExpression, connection);

object count = command.ExecuteScalar();

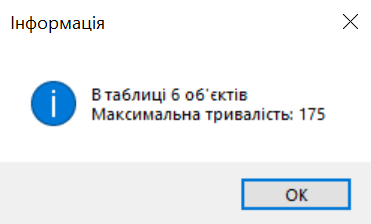
command.CommandText = "SELECT MAX(Duration) FROM Film";

object maxDuration = command.ExecuteScalar();

MessageBox.Show($"В таблиці {count} об'єктів\nМаксимальна тривалість: {maxDuration} ", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

connection.Close();

}

****

**6. Приклад виклику збереженої процедури з параметрами що повертаються.**

//команда , що повертає подробиці про фільм за його айді

connection.Open();

SqlCommand command = new SqlCommand("GetFilmByID", connection);

command.CommandType = CommandType.StoredProcedure;

command.Parameters.AddWithValue("@FilmId", IdFilmBox.Text);

command.Parameters.Add("@FilmName", SqlDbType.VarChar, 50).Direction = ParameterDirection.Output;

command.Parameters.Add("@Duration", SqlDbType.Int).Direction = ParameterDirection.Output;

command.Parameters.Add("@Producer", SqlDbType.VarChar, 50).Direction = ParameterDirection.Output;

command.Parameters.Add("@Studio", SqlDbType.VarChar, 50).Direction = ParameterDirection.Output;

command.Parameters.Add("@SeanceQuantity", SqlDbType.Int).Direction = ParameterDirection.Output;

command.Parameters.Add("@Genre", SqlDbType.VarChar, 50).Direction = ParameterDirection.Output;

command.ExecuteNonQuery();

var FilmName = command.Parameters["@FilmName"].Value.ToString();

var Duration = command.Parameters["@Duration"].Value.ToString();

var Producer = command.Parameters["@Producer"].Value.ToString();

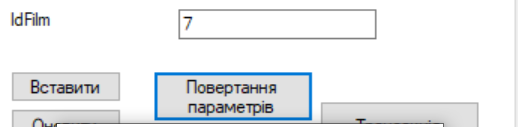
var Studio = command.Parameters["@Studio"].Value.ToString();

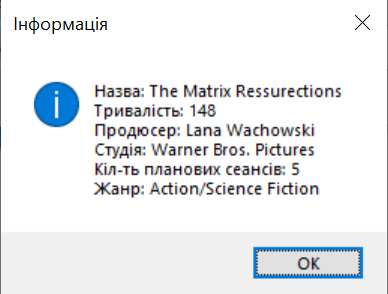
var SeanceQuantity = command.Parameters["@SeanceQuantity"].Value.ToString();

var Genre = command.Parameters["@Genre"].Value.ToString();

MessageBox.Show($"Назва: {FilmName}\nТривалість: {Duration}\nПродюсер: {Producer}\nCтудія: {Studio}\nКіл-ть планових сеансів: {SeanceQuantity}\nЖанр: {Genre}", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

connection.Close();



****

**7. Приклад виконання команд в транзакції.**

//транзакція яка добавляє 2-а фільми

connection.Open();

SqlTransaction transaction = connection.BeginTransaction();

SqlCommand command = connection.CreateCommand();

command.Transaction = transaction;

try

{

command.CommandText = $"INSERT INTO Film (FilmID, FilmName, Duration, Producer, Studio, SeanceQuantity, Genre)\r\nVALUES\r\n (8, 'Gone Girl', 149, 'David Fincher', '20th Century Fox', 4, 'Mystery/Thriller');";

command.ExecuteNonQuery();

command.CommandText = $"INSERT INTO Film (FilmID, FilmName, Duration, Producer, Studio, SeanceQuantity, Genre)\r\nVALUES\r\n (9, 'The Lion King', 118, 'Roger Allers, Rob Minkoff', 'Walt Disney Pictures', 6, 'Animation/Adventure');\r\n')";

command.ExecuteNonQuery();

transaction.Commit();

MessageBox.Show("Данні добавлені в таблицю", "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

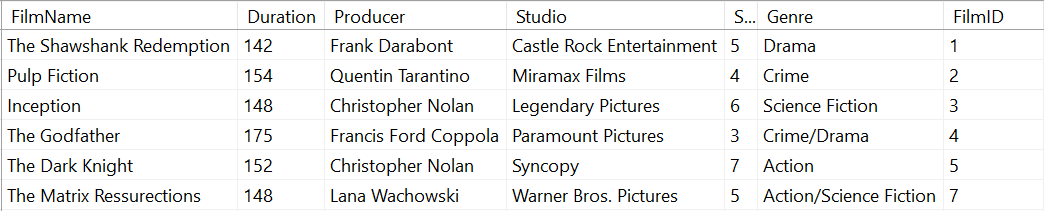
catch (Exception ex)

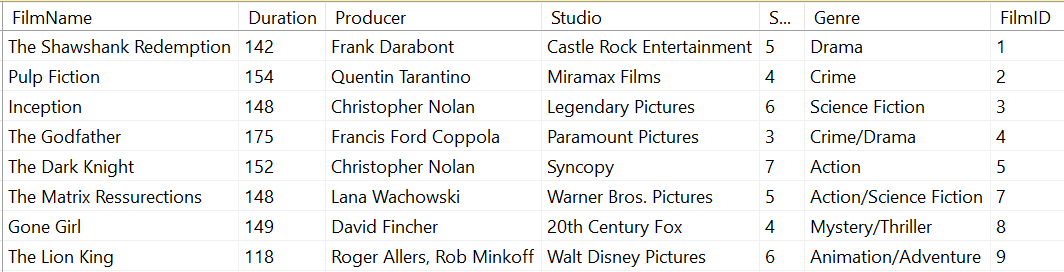
{

MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);

transaction.Rollback();

}

****

****

**8. Приклад роботи з DataSet, DataAdapter, DataGridView.**

string sqlExpression = "SELECT \* FROM Film";

using (SqlConnection connection = new SqlConnection(connectionString))

{

connection.Open();

SqlDataAdapter adapter = new SqlDataAdapter(sqlExpression, connection);

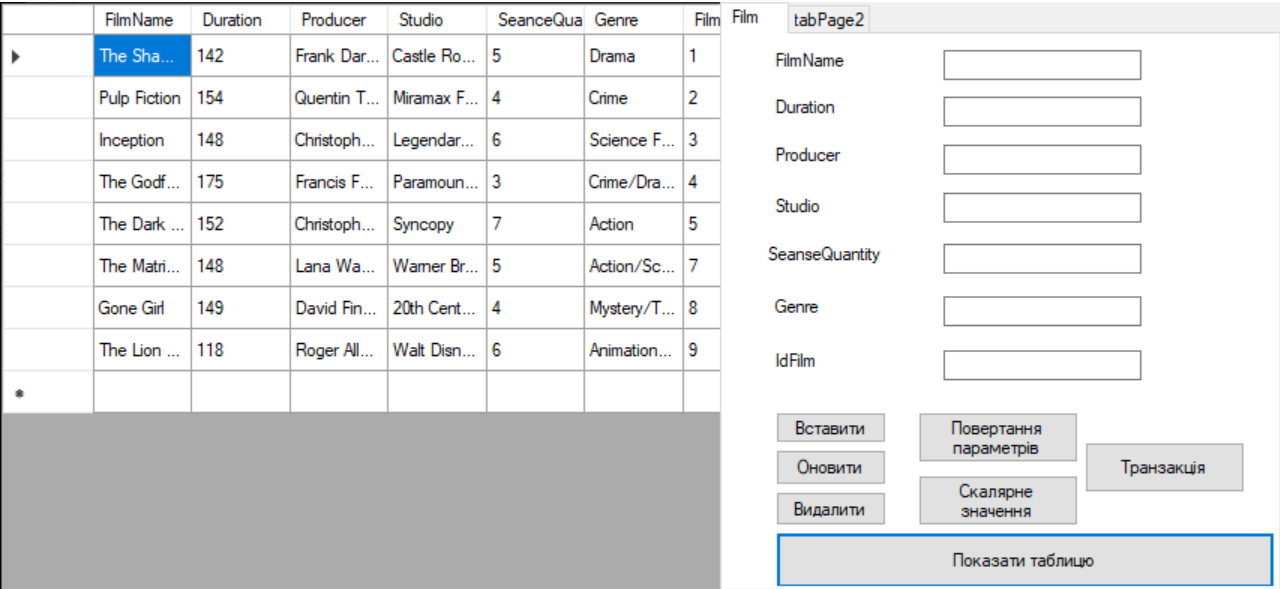
DataSet ds = new DataSet();

adapter.Fill(ds);

TableGridView.DataSource = ds.Tables[0];

connection.Close();

}



**Висновок :** під час виконання лабораторної роботи я ознайомилась із технологією доступу до баз даних ADO.net , а саме можливостям працювати з базою даних не тільки за допомогою SSMS . Розібралася з особливістю строки підключення , пул-підключенням , транзакціями та спеціальними класами для роботи з SQL базами даних.