

Міністерство освіти і науки України
Чернівецький національний університет
Імені Юрія Федьковича
Інститут фізико-технічних та комп'ютерних наук
Відділ комп'ютерних технологій
Кафедра математичних проблем управління і кібернетики

ЗВІТ

про виконання практичної роботи з дисципліни «Організація баз даних і знань»

Виконав студент
Курс ІІІ
Група 341
Викладач

Зелінський Д.Р.

Спіжавка Д.І.

Завдання

Варіант 9. Інформаційна система бібліотечного фонду

Бібліотечний фонд міста становлять бібліотеки, розташовані на території міста. Кожна бібліотека містить у собі абонементи й читальні зали. Користувачами бібліотек є різні категорії читачів: студенти, науковці, викладачі, школярі, робітники, пенсіонери й інші жителі міста. Кожна категорія читачів може мати непересічні характеристики – атрибути: для студентів ця назва навчального закладу, факультет, курс, номер групи, для науковця – назва організації, наукова тема й т.д. Кожен читач, будучи зареєстрованим в одній з бібліотек, має доступ до всього бібліотечного фонду міста.

Бібліотечний фонд (книги, журнали, газети, збірники статей, збірники віршів, дисертації, реферати, збірники доповідей і тез доповідей й ін.) розміщений у залах-сховищах різних бібліотек на певних місцях зберігання (номер залу, стелажа, полки) і ідентифікується номенклатурними номерами. При цьому існують різні правила щодо тих або інших видань: якісь підлягають тільки читанню в читальних залах бібліотек, для тих, що видаються, може бути встановлений різний строк видачі й т.д. Бібліотечний фонд може поповнюватися, а з часом відбувається його списання.

Твори авторів, що становлять бібліотечний фонд, також можна розділити на різні категорії, що характеризуються власним набором атрибутів: підручники, повісті, романи, статті, вірші, дисертації, реферати, тези доповідей і т.д.

Співробітники бібліотеки, що працюють у різних залах різних бібліотек, ведуть облік читачів, а також облік розміщення й видачі літератури

Види запитів в інформаційній системі:

- 1) Одержати список читачів із вказаними характеристиками: студенти вказаного навчального закладу або факультету, науковці.

- 2) Одержати інформацію про перелік читачів, на руках у яких перебуває вказана книга.
- 3) Одержати список видань, які протягом вказаного періоду одержував вказаний читач, вказуючи назви видань.
- 4) Одержати спичок видань, якими протягом вказаного періоду користувався вказаний читач із фонду бібліотеки.
- 5) Одержати список літератури, яка видана з вказаної полиці бібліотеки на даний момент.
- 6) Одержати список читачів, які протягом вказаного періоду часу зверталися до вказаного бібліотекаря.
- 7) Одержати інформацію про кількість читачів, яку обслуговував вказаний бібліотекар та загалом для всіх бібліотекарів.
- 8) Одержати список читачів із простроченим терміном здачі літератури.
- 9) Одержати список літератури, яка надішла до бібліотеки протягом вказаного періоду.
- 10) Одержати список літератури, яка була списана із бібліотеки протягом вказаного періоду.
- 11) Одержати список бібліотекарів, які працювали у вказаному читальному залі вказаної бібліотеки.
- 12) Одержати список читачів, які не відвідували бібліотеку протягом вказаного періоду, вказавши дату останнього відвідування бібліотеки.
- 13) Одержати список інвентарних номерів та назв книг із бібліотечного фонду, в яких знаходиться вказаний твір.
- 14) Одержати список інвентарних номерів та назв книг із бібліотечного фонду, в яких знаходяться твори вказаного автора.
- 15) Одержати список найпопулярніших творів.

Хід роботи

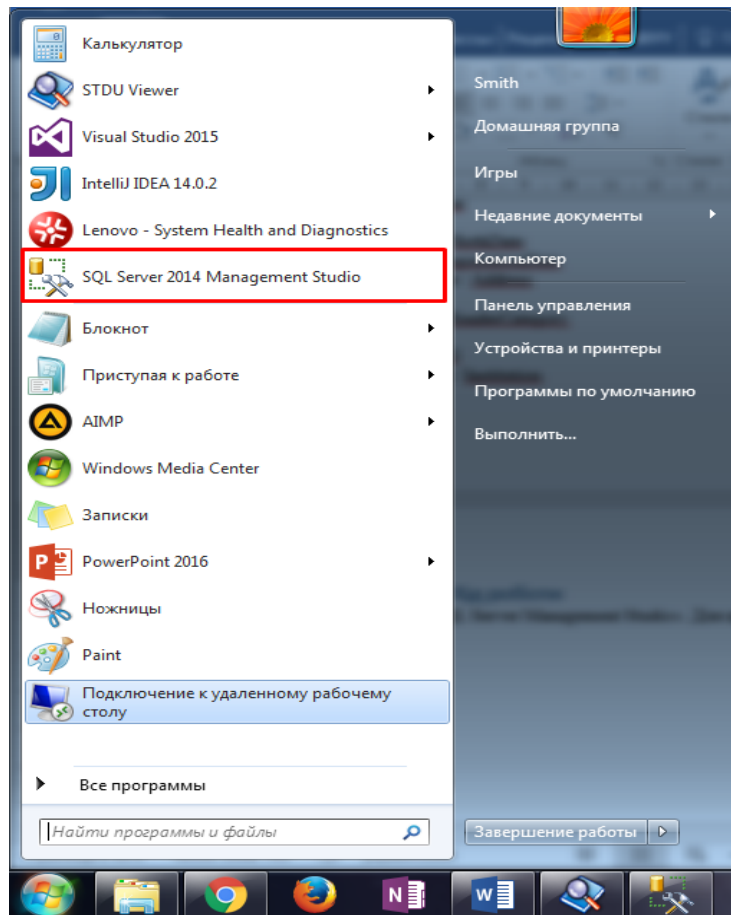
1. Проектування бази даних. Аналізуючи завдання, можна виділити такі сутності предметної області і їх атрибути:

- I. Бібліотека / Library
 - ID
 - Адреса / Address
 - телефон / Phone
 - директор / Director
- II. Викладач / Teacher
 - Id читача / ReaderID
 - Навчальний заклад / Institution
 - Факультет / Faculty
- III. Інші / Other
 - Id читача / ReaderID
 - Коментар / Comment
- IV. Історія / History
 - ID
 - Одиниця зберігання / Item
 - Кому видано / ReleaseTo
 - Ким видано / ReleaseBy
 - Дата видачі / ReleaseDate
 - Термін видачі / ReleaseTerm
- V. Категорія одиниці зберігання / ItemCategory
 - ID
 - Назва категорії / CategoryName
- VI. Науковець / Scientists
 - Id читача / Reader ID
 - Назва організації / Organization
 - Напрямок досліджень / Reserch
- VII. Одиниці зберігання / Item
 - Id
 - Назва / Name
 - автор / Author
 - Дата випуску / CreateDate
 - Категорія / Category
 - Бібліотека / Library
 - Сховище / Wareroom
 - Стелаж / Bench
 - В наявності / IsPresent
 - Лише для читання / ReadOnly
 - Останній користувач / LastReader
 - Дата купівлі / BuyingDate
 - Дата списання / DethDate
- VIII. Пенсіонер / Pensioner
 - Id читача / ReaderID
 - № пенсійного посвідчення / PensionerCode

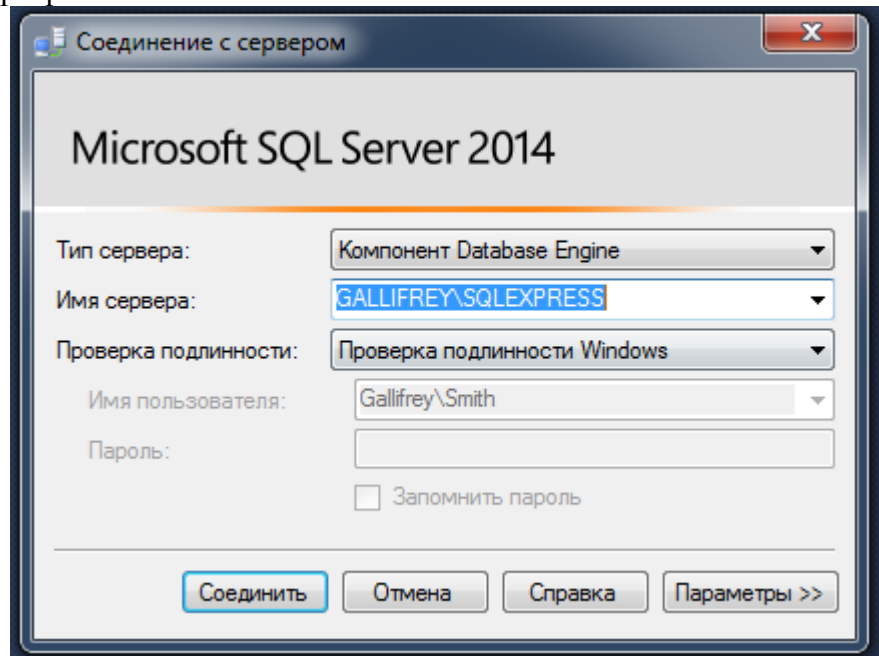
- IX. Робітник / Worker
 - Id читача / ReaderID
 - Підприємство / Factory
 - Посада / Occupation
- X. Співробітники / Employee
 - Id
 - Прізвище / Surname
 - Ім'я / Name
 - Дата народження / BirthDate
 - Бібліотека / Library
 - Посада / Occupation
 - Відповідальність / Responsibility
 - Оклад / Salary
 - Адреса проживання / Address
 - Телефон / Phone
- XI. Студент / Student
 - Id читача / ReaderID
 - Навчальний заклад / Institution
 - Факультет / Facult
 - Курс / Course
- XII. Сховище / Wareroom
 - Id
 - Назва сховища / Name
 - Бібліотека / Library
 - Відповідальний / Responsible
- XIII. Читальний зал / ReadingRoom
 - Id
 - Бібліотека / Library
 - Відповідальний / Responsible
- XIV. Читач / Reader
 - Id
 - Прізвище / Surname
 - Ім'я / Name
 - Дата народження / BirthDate
 - Дата реєстрації / RegistrationDate
 - Адреса проживання / Address
 - Телефон / Phone
- XV. Школяр / SchoolChild
 - Id читача / ReaderID
 - Навчальний заклад / Institution
 - Клас / Form

Хід роботи

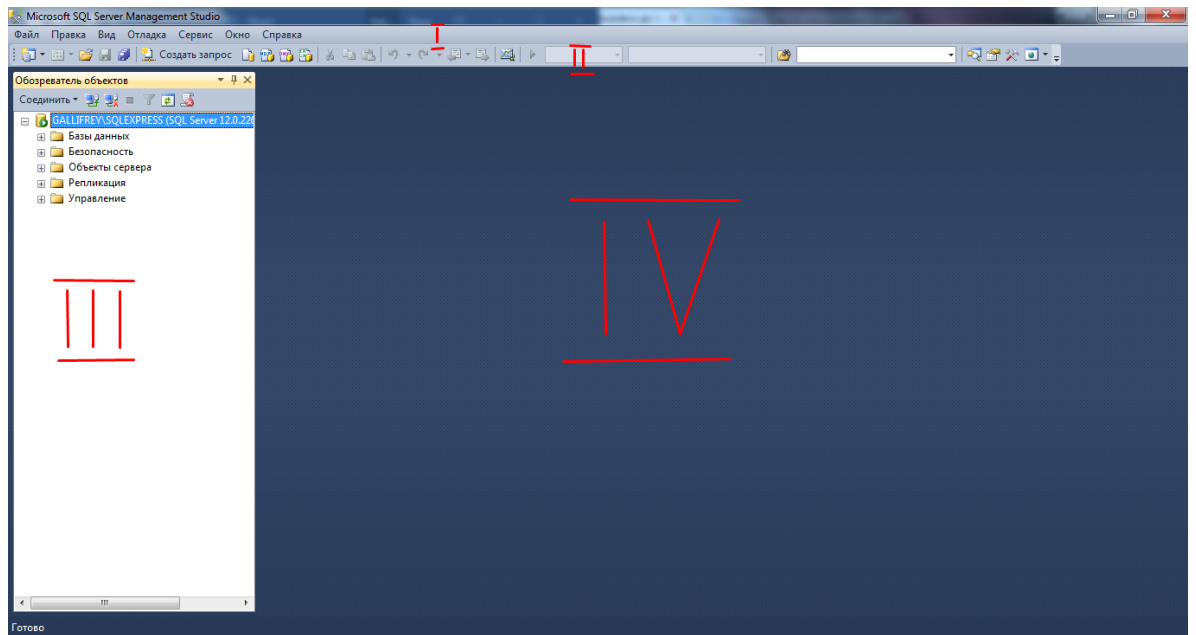
1. Запускаємо середовище розробки «SQL Server Management Studio». Для цього в меню «Пуск» вибираємо пункт «SQL Server 2014 Management Studio».



2. Після запуску середовища розробки з'явиться вікно підключення до сервера «З'єднання з сервером» .



3. Вибираємо сервер і нажимаємо «Соединить»
4. Після натиснення кнопки «З'єднати» з'явиться вікно середовища розробки «SQL Server Management Studio»:



5. Це вікно має наступну структуру:

I. Віконне меню - містить повний набір команд для управління сервером і виконання різних операцій.

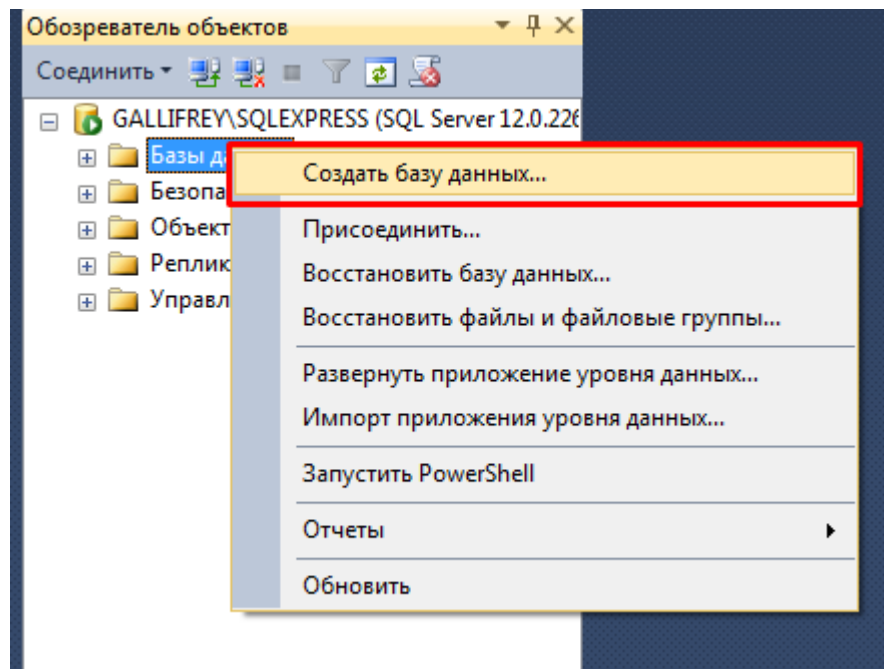
II. Панель інструментів - містить кнопки для виконання найбільш часто вироблюваних операцій. Зовнішній вигляд цієї панелі залежить від виконуваної операції.

III. Панель «Оглядач об'єктів» - оглядач об'єктів. Оглядач об'єктів - це панель з деревовидною структурою, що відображає усі об'єкти сервера, а також що дозволяє проводити різні операції, як з самим сервером, так і з БД. Оглядач об'єктів є основним інструментом для розробки БД.

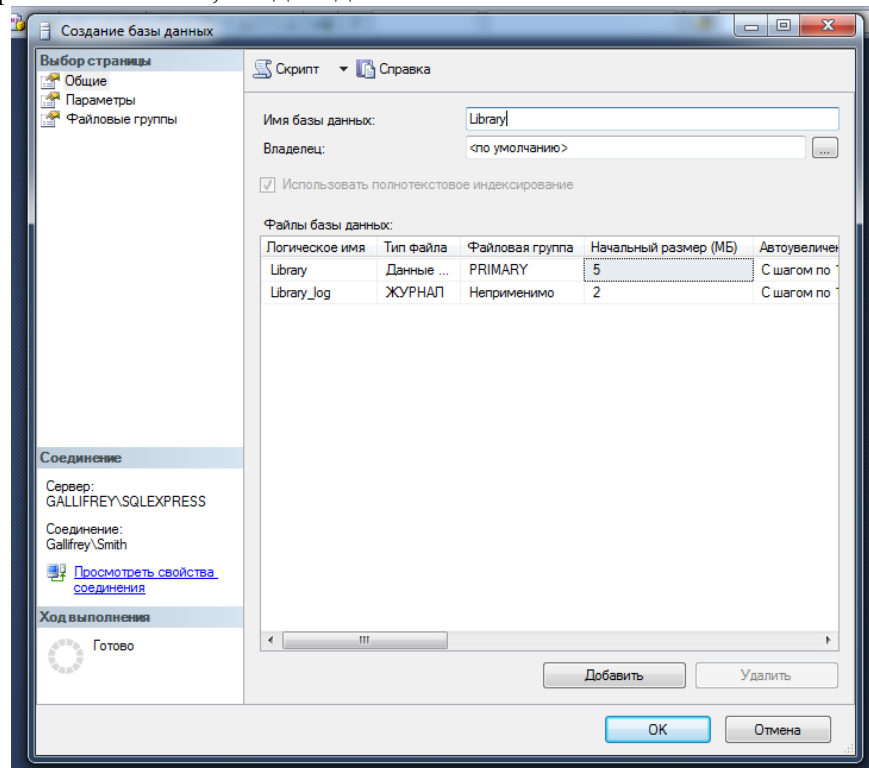
IV. Робоча область. У робочій області виконуються усі дії з БД, а також відображається її вміст.

У оглядачі об'єктів самі об'єкти знаходяться в теках. Щоб відкрити теку необхідно клацнути по знаку «+» зліва від зображення теки.

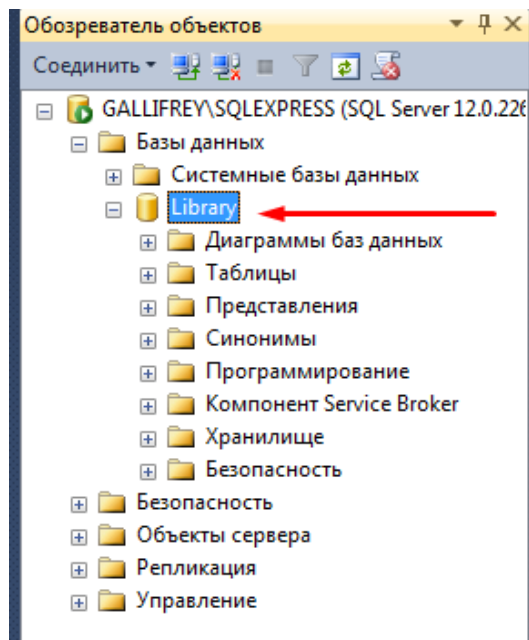
6. Створюємо нову базу даних згідно завдання: викликаємо контекстне меню на теці «Бази даних» - «Создать базу даних»



7. Відкривається вікно, вводим дані – натискаєм «ОК».



8. Після цього ми можемо побачити створену базу даних



Уся інформація у базі даних зберігається в таблицях, які є засобом зберігання даних. Таблиці складаються з рядків або записів і стовпців або полів. Кожне поле має три характеристики:

1. Ім'я поля - використовується для звернення до поля;
2. Значення поля - визначає інформацію, що зберігається в полі;
3. Тип даних поля - визначає, який вид інформації можна зберігати в полі.

У SQL сервер використовується наступні типи даних :

- **Двійкові типи даних**, які містять послідовності нулів і одиниць : 1) `binary(n)` — двійковий тип фіксованої довжини розміром в `n` байт, деп — значення від 1 до 8000; розмір при зберіганні складає `n` байт; 2) `varbinary(n)` - двійковий тип зі змінною довжиною, `n` може мати значення від 1 до 8000, `max` вказує, що максимальний розмір при зберіганні складає 23-1;

- **Цілочисельні типи даних** - типи даних для зберігання цілих чисел (у дужках вказаний діапазон значень типу даних) : `tinyint` (0.255), `smallint` (- 32768.+32767), `int` (- 231.+ (231-31)), `bigint` (- 263.+ (263-1));

- **Типи даних для зберігання чисел з плаваючою комою**: `real` займає в пам'яті 4 байти; `float(n)`, де `n` — ця кількість бітів, використовуваних для зберігання мантиси числа у форматі `float` при експоненціальному представленні, визначає точність даних і розмір для зберігання; значення параметра `n` повинне лежати в межах від 1 до 53; значенням за умовчанням для параметра `n` є 53;

- **Типи даних для зберігання чисел з фіксованою точністю і масштабом** : `decimal(p, s)` і `numeric(p, s)`, де `p` (точність) - максимальна кількість десяткових розрядів числа (як ліворуч, так і праворуч від десяткової коми). Точність має бути значенням в діапазоні від 1 до 38. За умовчанням це значення дорівнює 18. `s` (масштаб) - максимальна кількість десяткових розрядів числа праворуч від десяткової коми. Масштаб може набувати значення від 0 до `p` і може бути вказаний тільки спільно з точністю. За умовчанням масштаб набуває значення 0; тому $0 \leq s \leq p$;

- **Символьні типи даних** : `char(n)` - строковые дані фіксованої довжини не в Юнікоде, аргумент `n` визначає довжину рядка і повинен мати значення від 1 до 8000, розмір при зберіганні складає `n` байт; `varchar(n | max)` - строкові дані змінної довжини не в Юнікоде, аргумент `n` визначає довжину рядка і повинен мати значення від 1 до 8000,

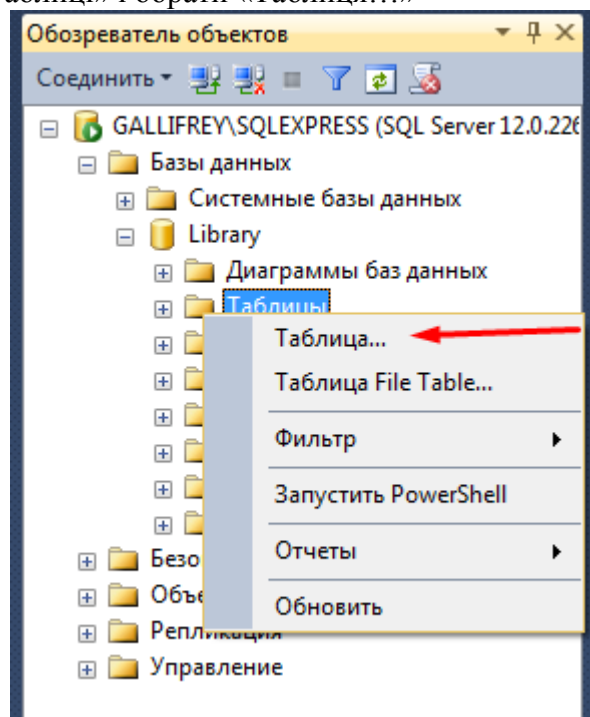
значення max вказує, що максимальний розмір при зберіганні складає 231-1 байт (2 ГБ); text - дані змінної довжини не в Юнікоде в кодовій сторінці сервера і з максимальною довжиною рядка 231-1;

- **Спеціальні типи даних** : bit - цілочисельний тип даних, який може набувати значень 1, 0 або NULL; image - тип даних для зберігання малюнка розміром до 2ГБ;

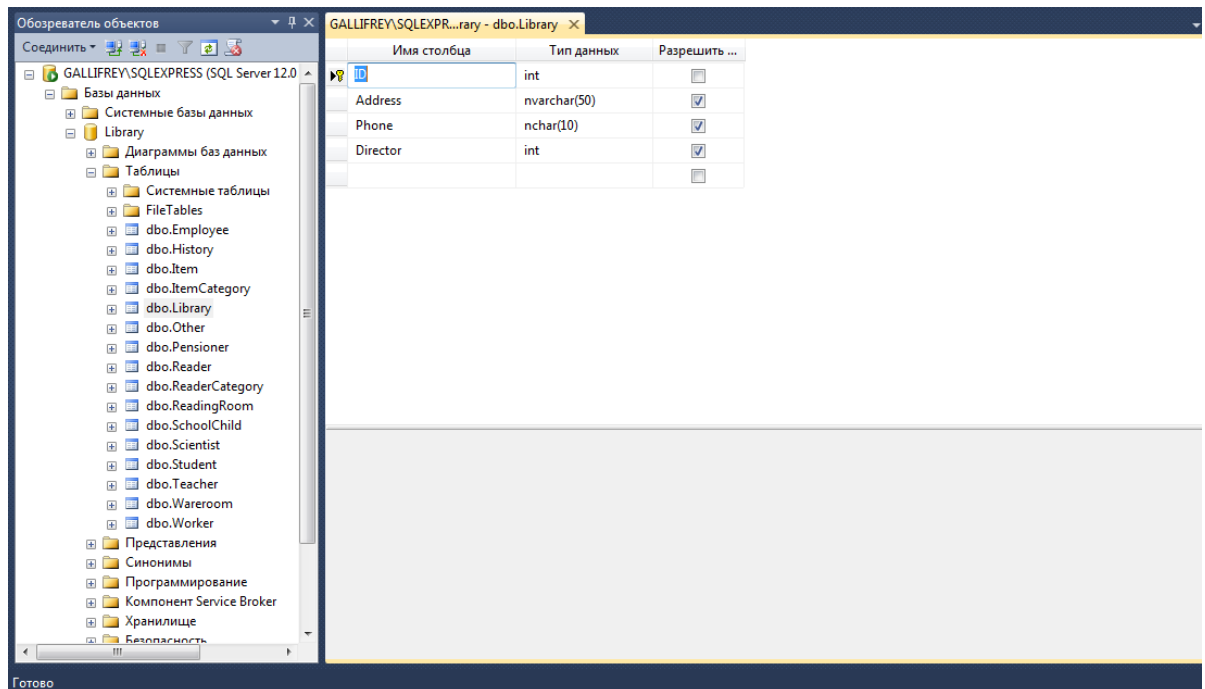
- **Типи даних дати і часу** : date (від 01.01.0001 до 31.12.9999); datetime (діапазон дати - від 01.01.1753 до 31.12.1999, діапазон часу - від 00:00:00 до 23:59:59, 997); smalldatetime (діапазон дати - від 01.01.1900 до 6.06.2079, діапазон часу - від 00:00:00 до 23:59:59); time (від 00:00:00.0000000 до 23:59:59.9999999);

- **Грошові типи даних** для зберігання фінансової інформації : money (8 байт) і smallmoney (4 байти) - типи даних, що представляють грошові (валютні) значення.

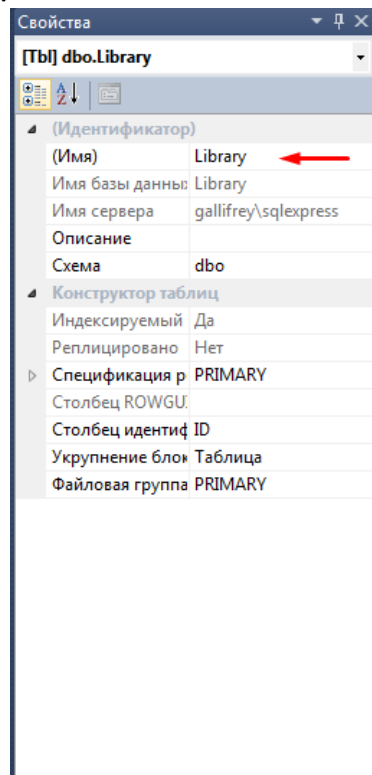
9. Додаєм таблиці до створеної бази даних. Для цього потрібно викликати контекстне меню на теці «Таблиці» і обрати «Таблица...»



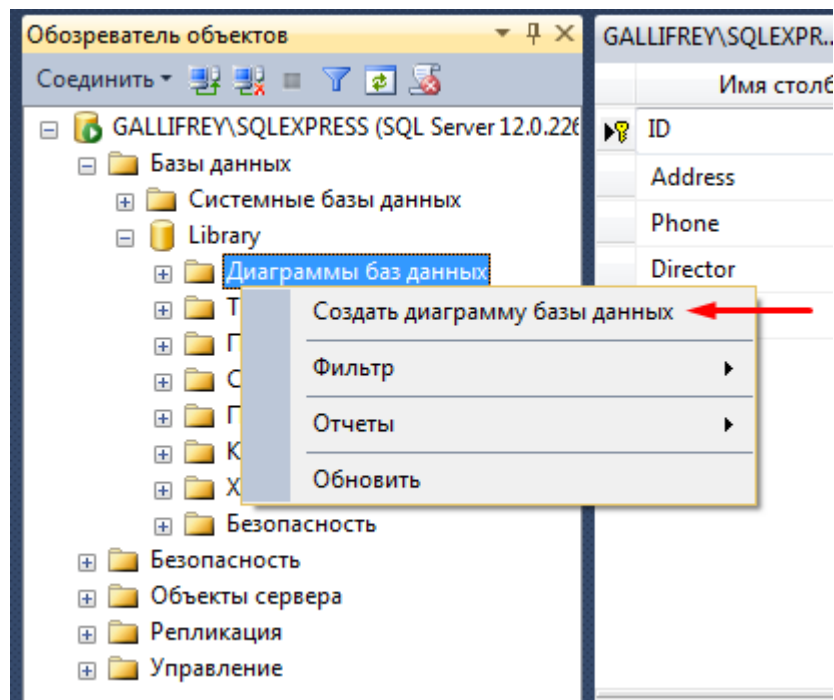
10. В правій частині відкриться конструктор таблиці



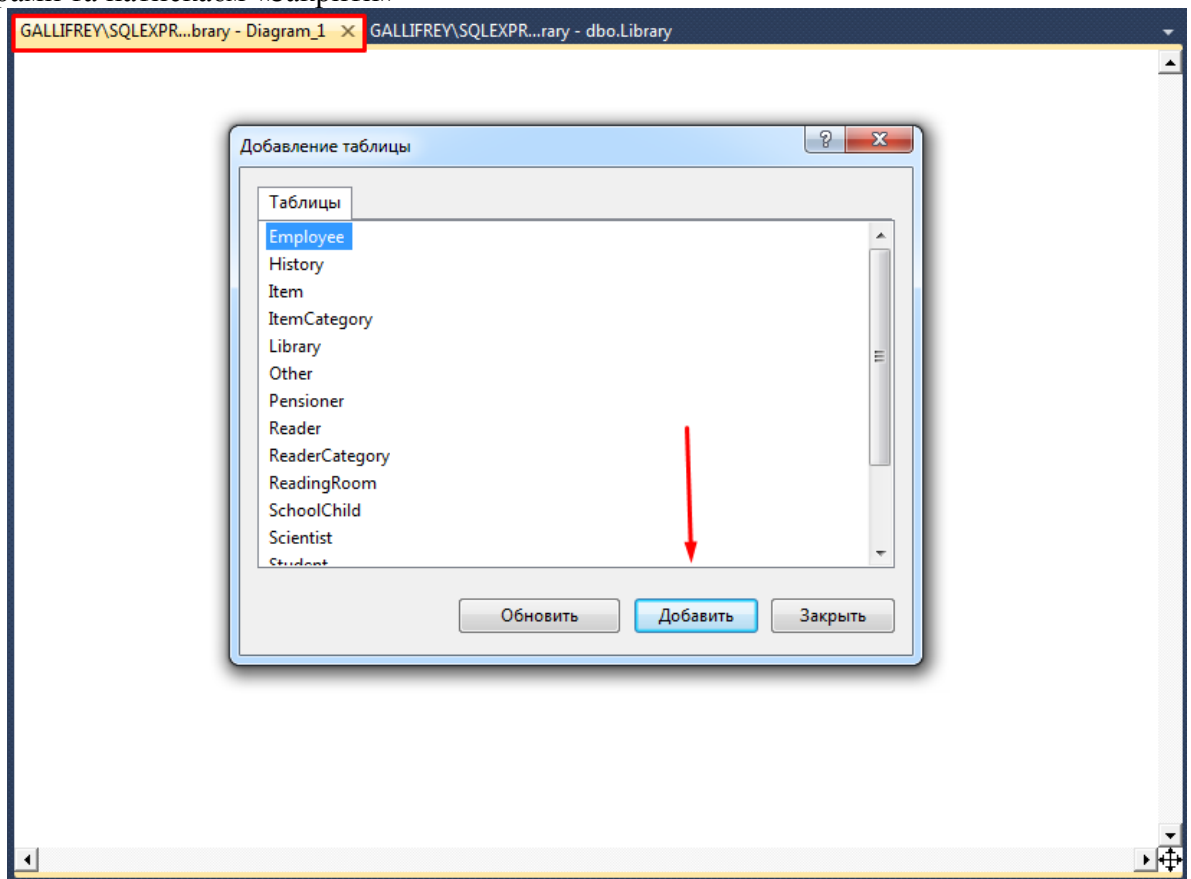
11. Заповнюєм поля згідно варіанту, задаючи імена, типи і ключові поля таблиці, та зберігаєм під відповідним іменем.



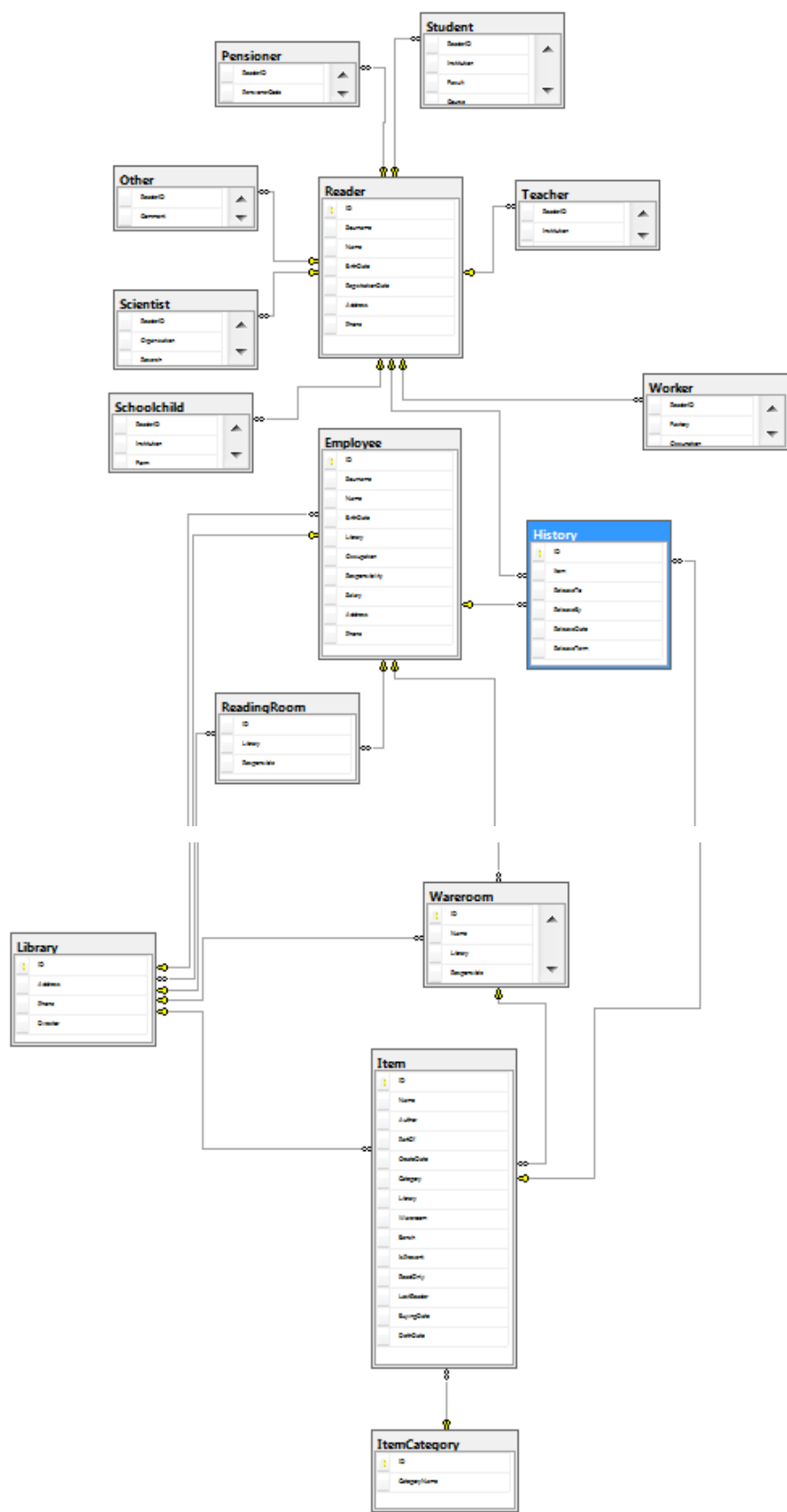
12. Після створення таблиць створюєм звязки між ними. Для цього викликаєм контекстне меню на теці «Діаграми баз даних» - та обираєм «Створити діаграму бази даних»



13. Відкривається нова діаграма і вікно додавання таблиць. Додаємо всі таблиці до діаграми та натискаємо «Закрити»



14. Відповідні зв'язки створюємо в результаті перетягування відповідних полів таблиць.



15. Реалізуємо заданні запити, та оформим їх як «Збережені процедури» з відповідними вхідними параметрами, необхідними для виконання запиту. Самі збережені процедури згодом можна буде викликати з допомогою звичайного запиту, передавши задані параметри. Це дозволить формувати та виконувати запити динамічно під час виконання програми. Нижче наведено скрипти створення збережених процедур, які реалізують задані запити:

I. /***** Object: StoredProcedure [dbo].[Q1] Script Date: 22.10.2016 2:41:39

*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[Q1]

@ReaderCategory nvarchar(50),

@Institution nvarchar(50)

AS

BEGIN

SET NOCOUNT ON;

IF (@ReaderCategory = N'Student')

SELECT dbo.Reader.Name, dbo.Reader.Surname, Student.Facult,
dbo.Student.Course

FROM dbo.Reader INNER JOIN
dbo.Student ON dbo.Reader.ID = dbo.Student.ReaderID

WHERE (dbo.Student.Institution = @Institution)

ELSE

IF (@ReaderCategory = N'Scientist')

SELECT dbo.Reader.Name, dbo.Reader.Surname,
dbo.Scientist.Organization, dbo.Scientist.Reserch

FROM dbo.Reader INNER JOIN
dbo.Scientist ON dbo.Reader.ID = dbo.Scientist.ReaderID

WHERE (dbo.Scientist.Institution = @Institution)

END

GO

II. /***** Object: StoredProcedure [dbo].[Q2] Script Date: 22.10.2016 2:41:39

*****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[Q2]

@name nvarchar(50),

@author nvarchar(50)

AS

BEGIN

SET NOCOUNT ON;

SELECT Reader.Surname, Reader.Name

FROM Reader INNER JOIN Item ON Item.LastReader = Reader.ID

WHERE Item.Name = @name AND Item.Author = @author AND
Item.IsPresent = 'FALSE'

```

END
GO
III. /***** Object: StoredProcedure [dbo].[Q3]   Script Date: 22.10.2016 2:41:39
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q3]
    @name nvarchar(50),
    @suorname nvarchar(50),
    @dateFrom date,
    @dateTo date

AS
BEGIN
    SET NOCOUNT ON;
    SELECT Reader.Sourname, Reader.Name, Item.Name, Item.Author
    FROM Reader INNER JOIN History ON History.ReleaseTo = Reader.ID
    INNER JOIN Item ON Item.id = History.Item
    WHERE Reader.Name = @name AND Reader.Sourname =
    @suorname AND (History.ReleaseDate BETWEEN @dateFrom AND
    @dateTo )
END
GO
IV. /***** Object: StoredProcedure [dbo].[Q4]   Script Date: 22.10.2016 2:41:39
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q4]
    @name nvarchar(50),
    @sourname nvarchar(50),
    @dateFrom date,
    @dateTo date

AS
BEGIN
    SET NOCOUNT ON;
    SELECT Reader.Sourname, Reader.Name, Item.Name, Item.Author
    FROM Reader INNER JOIN History ON History.ReleaseTo = Reader.ID
    INNER JOIN Item ON Item.id = History.Item
    WHERE Reader.Name = @name AND Reader.Sourname =
    @sourname AND Item.ReadOnly = 'TRUE' AND (History.ReleaseDate
    BETWEEN @dateFrom AND @dateTo )
END
GO
V. /***** Object: StoredProcedure [dbo].[Q5]   Script Date: 22.10.2016 2:41:39
*****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q5]
    @bench int
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Item.Name, Item.Author, Reader.Name, Reader.Surname as
'Reader'
    FROM Item JOIN READER ON Item.LastReader = Reader.ID
    WHERE Item.Bench = @bench AND Item.IsPresent = 'FALSE'
END
GO

```

VI. /***** Object: StoredProcedure [dbo].[Q6] Script Date: 22.10.2016 2:41:39
*****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q6]
    @name nvarchar(50),
    @surname nvarchar(50),
    @dateFrom date,
    @dateTo date
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Reader.Surname, Reader.Name
    FROM Reader INNER JOIN History ON History.ReleaseTo = Reader.ID
        INNER JOIN Employee ON Employee.id = History.ReleaseBy
        WHERE Employee.Name = @name AND Employee.Surname =
@surname AND (History.ReleaseDate BETWEEN @dateFrom AND
@dateTo )
END
GO

```

VII. /***** Object: StoredProcedure [dbo].[Q7] Script Date: 22.10.2016 2:41:39
*****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q7]
    @name nvarchar(50),
    @surname nvarchar(50)
AS
BEGIN
    SET NOCOUNT ON;

```



```

SELECT COUNT (History.ID) AS Amount
FROM HISTORY JOIN Employee ON History.ReleaseBy = Employee.ID
WHERE Employee.Surname = @surname AND Employee.Name =
@name
SELECT Employee.Surname,Employee.Name, COUNT (*) AS Amount
FROM HISTORY JOIN Employee ON History.ReleaseBy = Employee.ID
GROUP BY Employee.Surname, Employee.Name
END
GO

```

VIII. /***** Object: StoredProcedure [dbo].[Q8] Script Date: 22.10.2016 2:41:39
*****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q8]
AS
BEGIN
SET NOCOUNT ON;

SELECT Reader.Surname, Reader.Name, Item.Name, Item.Author,
MAX(History.ReleaseTerm) AS Term
FROM Reader
JOIN Item ON Reader.ID = Item.LastReader
JOIN History ON History.Item = Item.ID
WHERE Item.IsPresent = 'FALSE' AND History.ReleaseTerm <
GETDATE()
GROUP BY Reader.Surname, Reader.Name, Item.Name, Item.Author
END
GO

```

IX. /***** Object: StoredProcedure [dbo].[Q9] Script Date: 22.10.2016 2:41:39
*****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q9]
@dateFrom date,
@dateTo date
AS
BEGIN
SET NOCOUNT ON;
SELECT Item.Name, Item.Author
FROM Item
WHERE Item.BuyingDate BETWEEN @dateFrom AND @dateTo
END
GO

```

X. /***** Object: StoredProcedure [dbo].[Q10] Script Date: 22.10.2016 2:41:39
*****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q10]
    @dateFrom date,
    @dateTo date
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Item.Name, Item.Author
        FROM Item
        WHERE Item.DethDate BETWEEN @dateFrom AND @dateTo
END
GO

```

XI. /***** Object: StoredProcedure [dbo].[Q11] Script Date: 22.10.2016 2:41:39
 *****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q11]
    @lib int,
    @rroom int
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Employee.Sourname, Employee.Name
        FROM Employee JOIN ReadingRoom ON Employee.ID =
        ReadingRoom.Responsible
        WHERE ReadingRoom.Library = @lib AND ReadingRoom.ID = @rroom
END
GO

```

XII. /***** Object: StoredProcedure [dbo].[Q12] Script Date: 22.10.2016 2:41:39
 *****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q12]
    @dateFrom date,
    @dateTo date
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Reader.Sourname, Reader.Name, MAX(History.ReleaseDate) AS
        LasstVisit
        FROM Reader JOIN History ON Reader.ID = History.ReleaseTo
        WHERE History.ReleaseDate NOT BETWEEN @dateFrom AND @dateTo

```

```

        AND Reader.ID NOT IN (
            SELECT Reader.ID
            FROM Reader JOIN History ON Reader.ID =
History.ReleaseTo
            WHERE History.ReleaseDate BETWEEN @dateFrom
AND @dateTo
        )
    GROUP BY Reader.Sourname, Reader.Name
END
GO

```

XIII. /***** Object: StoredProcedure [dbo].[Q13] Script Date: 22.10.2016 2:41:39
*****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q13]
    @name nvarchar(50),
    @auth nvarchar(50)
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Item.ID, Item.Name, Item.Author
    FROM Item
    WHERE Item.ID IN (
        SELECT Item.PartOf
        FROM Item
        WHERE Item.Name = @name AND Item.Author = @auth
    )
END
GO

```

XIV. /***** Object: StoredProcedure [dbo].[Q14] Script Date: 22.10.2016 2:41:39
*****/

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q14]
    @auth nvarchar(50)
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Item.ID, Item.Name, Item.Author
    FROM Item
    WHERE Item.ID IN (
        SELECT Item.PartOf
        FROM Item
        WHERE Item.Author = @auth
    )

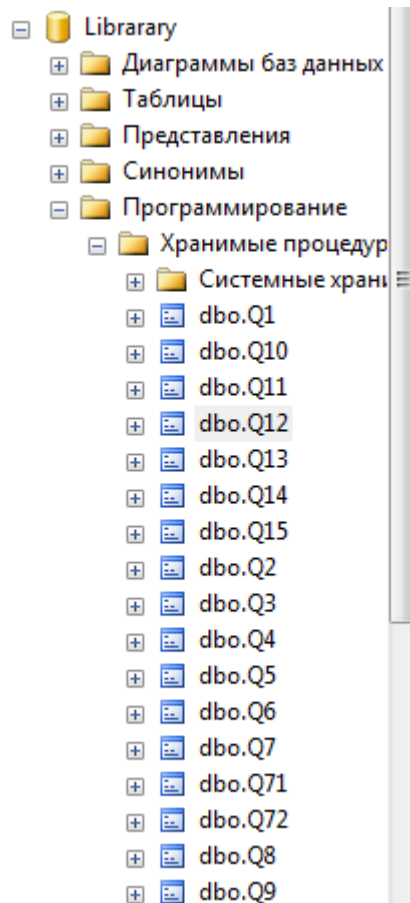
```

```

END
GO
XV. /***** Object: StoredProcedure [dbo].[Q15]   Script Date: 22.10.2016 2:41:39
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[Q15]
AS
BEGIN
    SET NOCOUNT ON;
    SELECT Item.Name, Item.Author, COUNT(History.Item) as Popularity
    FROM Item JOIN History ON Item.ID = History.Item
    GROUP BY Item.Name, Item.Author
    ORDER BY COUNT(History.Item) DESC
END
GO

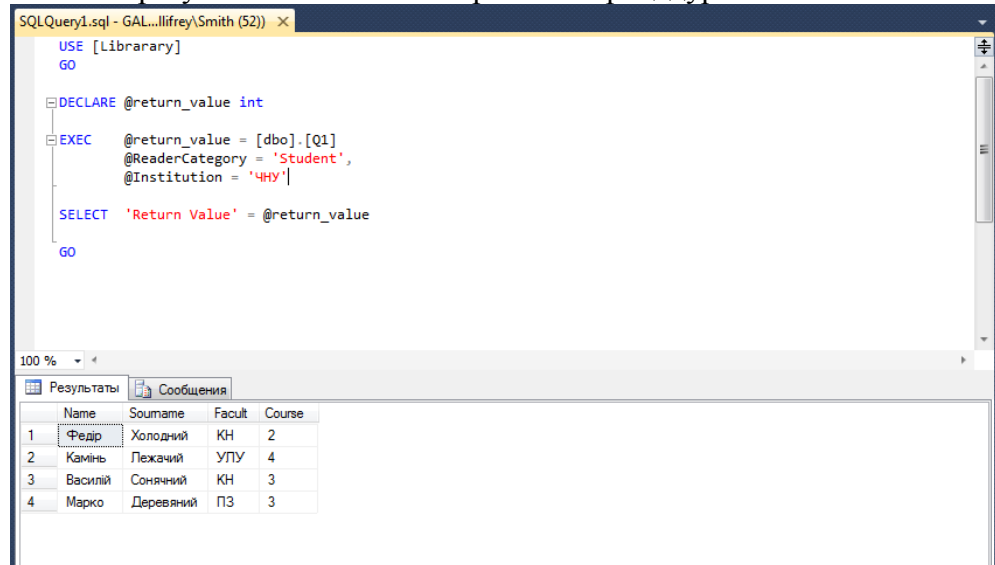
```

16. У результаті виконання створено 15 збережених процедур (по одній на кожен запит, додатково 7-мий запит розбито на дві процедури, необхідно буде далі для виклику через створену програму)



17. Перевіримо роботу збережених процедур, викликавши їх з певними параметрами

I. Виклик і результат виконання збереженої процедури №1



```
USE [Library]
GO

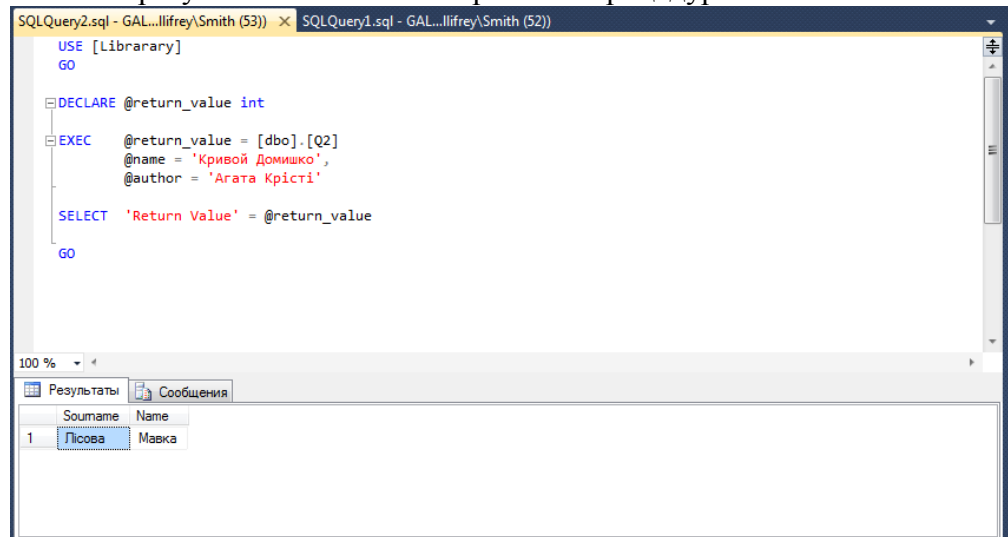
DECLARE @return_value int

EXEC @return_value = [dbo].[Q1]
    @ReaderCategory = 'Student',
    @Institution = 'ЧНУ'

SELECT 'Return Value' = @return_value
GO
```

	Name	Surname	Facult	Course
1	Федір	Холодний	КН	2
2	Камінь	Лежачий	УПУ	4
3	Василь	Сонячний	КН	3
4	Марко	Деревяний	ПЗ	3

II. Виклик і результат виконання збереженої процедури №2



```
USE [Library]
GO

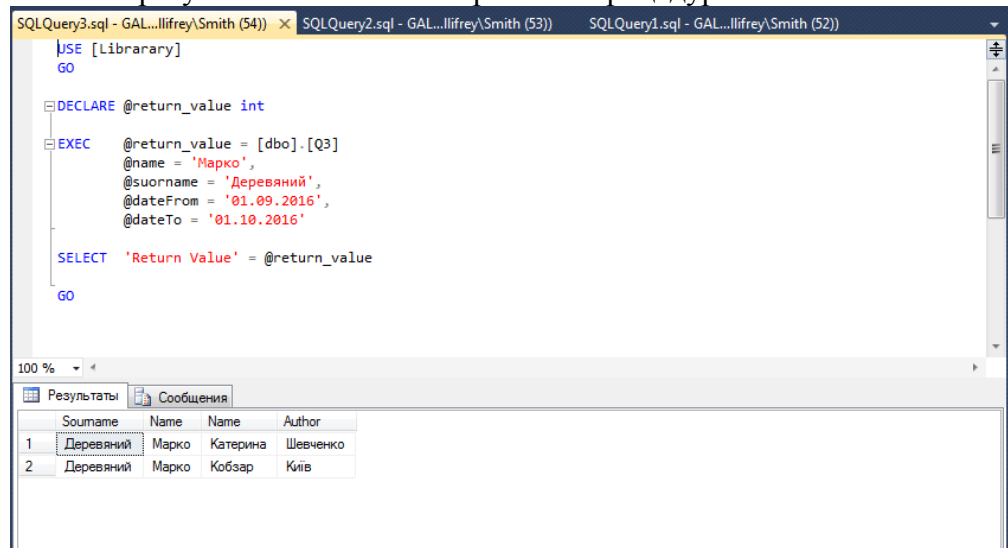
DECLARE @return_value int

EXEC @return_value = [dbo].[Q2]
    @name = 'Кривой Домашко',
    @author = 'Агата Крісті'

SELECT 'Return Value' = @return_value
GO
```

	Surname	Name
1	Пісова	Мавка

III. Виклик і результат виконання збереженої процедури №3



```
USE [Library]
GO

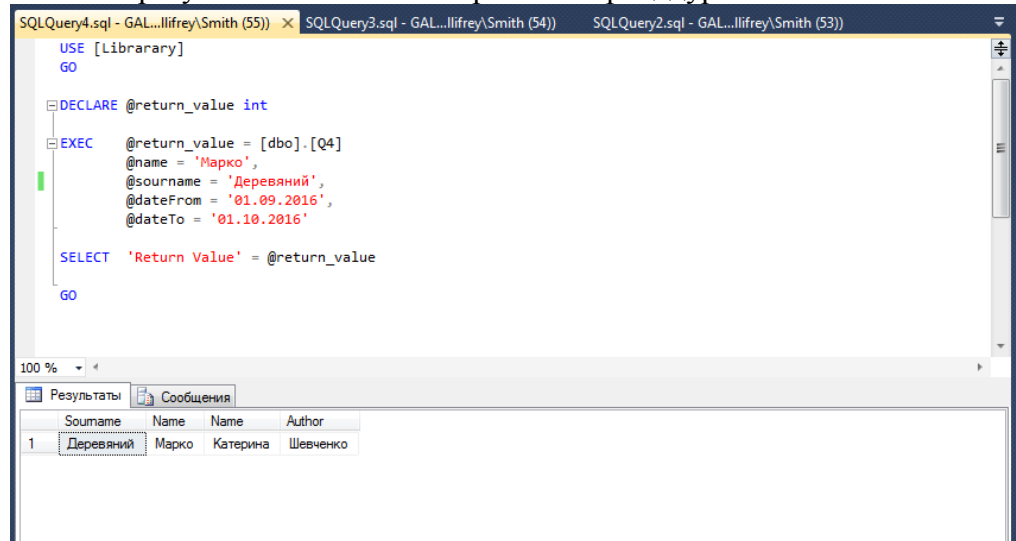
DECLARE @return_value int

EXEC @return_value = [dbo].[Q3]
    @name = 'Марко',
    @surname = 'Деревяний',
    @dateFrom = '01.09.2016',
    @dateTo = '01.10.2016'

SELECT 'Return Value' = @return_value
GO
```

	Surname	Name	Name	Author
1	Деревяний	Марко	Катерина	Шевченко
2	Деревяний	Марко	Кобзар	Кіів

IV. Виклик і результат виконання збереженої процедури №4



```
USE [Library]
GO

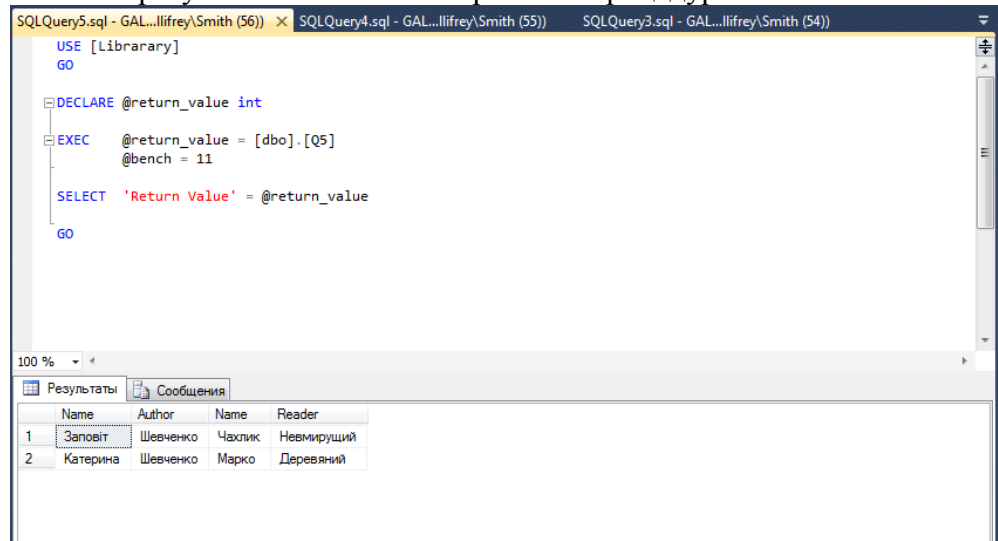
DECLARE @return_value int

EXEC @return_value = [dbo].[Q4]
    @name = 'Марко',
    @sourname = 'Деревяний',
    @dateFrom = '01.09.2016',
    @dateTo = '01.10.2016'

SELECT 'Return Value' = @return_value
GO
```

	Souname	Name	Name	Author
1	Деревяний	Марко	Катерина	Шевченко

V. Виклик і результат виконання збереженої процедури №5



```
USE [Library]
GO

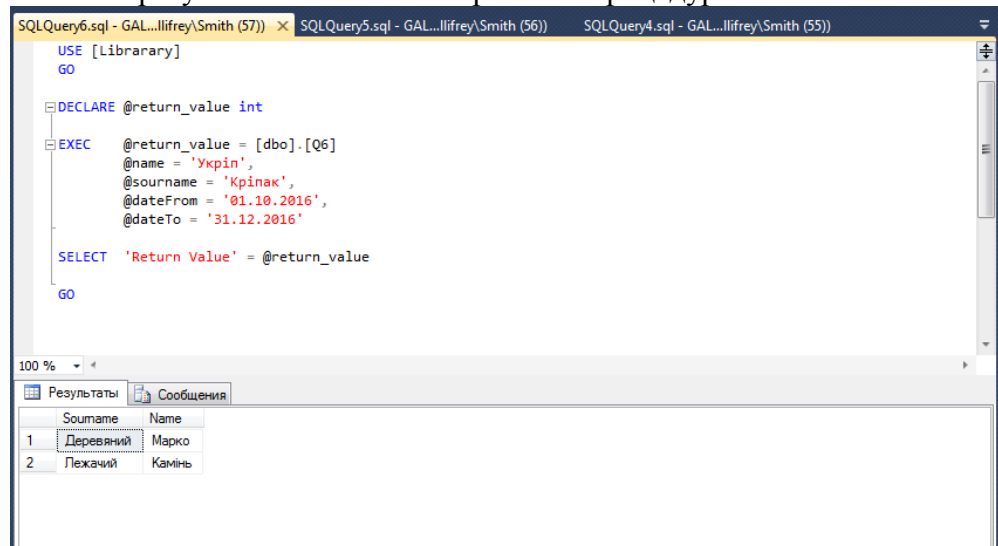
DECLARE @return_value int

EXEC @return_value = [dbo].[Q5]
    @bench = 11

SELECT 'Return Value' = @return_value
GO
```

	Name	Author	Name	Reader
1	Заповіт	Шевченко	Чухлик	Невмирущий
2	Катерина	Шевченко	Марко	Деревяний

VI. Виклик і результат виконання збереженої процедури №6



```
USE [Library]
GO

DECLARE @return_value int

EXEC @return_value = [dbo].[Q6]
    @name = 'Укпін',
    @sourname = 'Крпінак',
    @dateFrom = '01.10.2016',
    @dateTo = '31.12.2016'

SELECT 'Return Value' = @return_value
GO
```

	Souname	Name
1	Деревяний	Марко
2	Лежачий	Камінь

VII. Виклик і результат виконання збереженої процедури №7

```

USE [Library]
GO

DECLARE @return_value int

EXEC     @return_value = [dbo].[Q7]
        @name = 'Укпін',
        @sourname = 'Крпінак'

SELECT  'Return Value' = @return_value
GO
    
```

Amount
4

Sourname	Name	Amount
1	Комп'ютерний	Вірус
2	Домашній	Кіт
3	Фараон	Микита
4	Дворовий	Пес
5	Віконний	Петро
6	Крпінак	Укрпін

VIII. Виклик і результат виконання збереженої процедури №8

```

USE [Library]
GO

DECLARE @return_value int

EXEC     @return_value = [dbo].[Q8]

SELECT  'Return Value' = @return_value
GO
    
```

Sourname	Name	Name	Author	Term
1	Дерев'яний	Марко	Катерина Шевченко	2016-10-14
2	Лісова	Мавка	Кривий Домішко	Агата Крісті 2014-12-12

IX. Виклик і результат виконання збереженої процедури №9

```

USE [Library]
GO

DECLARE @return_value int

EXEC     @return_value = [dbo].[Q9]
        @dateFrom = '01.01.2008',
        @dateTo = '31.12.2008'

SELECT  'Return Value' = @return_value
GO
    
```

Name	Author
1	Як приручить дракона
2	Гаррі Поттер
3	Заповіт

X. Виклик і результат виконання збереженої процедури №10

```

USE [Library]
GO

DECLARE @return_value int

EXEC     @return_value = [dbo].[Q10]
        @dateFrom = '01.01.2016',
        @dateTo = '31.12.2016'

SELECT   'Return Value' = @return_value
GO
    
```

	Name	Author
1	Кривой Домишко	Агата Крісті
2	Основи програмування	Шкилдт
3	Кобзар	Київ
4	Гарри Поттер	Роулінг
5	Катерина	Шевченко
6	Кобзар	Шевченко

XI. Виклик і результат виконання збереженої процедури №11

```

USE [Library]
GO

DECLARE @return_value int

EXEC     @return_value = [dbo].[Q11]
        @lib = 2,
        @rroom = 2

SELECT   'Return Value' = @return_value
GO
    
```

	Sourname	Name
1	Крпак	Україн
2	Домашній	Кіт
3	Філіппін	Філіп

XII. Виклик і результат виконання збереженої процедури №12

```

USE [Library]
GO

DECLARE @return_value int

EXEC     @return_value = [dbo].[Q12]
        @dateFrom = '01.09.2016',
        @dateTo = '01.10.2016'

SELECT   'Return Value' = @return_value
GO
    
```

	Sourname	Name	LastVisit
1	Сонячний	Василь	2016-12-11
2	Проточний	Василь	2016-10-10
3	Лежачий	Камінь	2016-11-09
4	Лісова	Мавка	2014-11-11
5	Шкільний	Петро	2016-08-08

XIII. Виклик і результат виконання збереженої процедури №13

```
USE [Library]
GO

DECLARE @return_value int

EXEC @return_value = [dbo].[Q13]
    @name = 'Заповіт',
    @auth = 'Шевченко'

SELECT 'Return Value' = @return_value
GO
```

ID	Name	Author
1	4	Кобзар

XIV. Виклик і результат виконання збереженої процедури №14

```
USE [Library]
GO

DECLARE @return_value int

EXEC @return_value = [dbo].[Q14]
    @auth = 'Шевченко'

SELECT 'Return Value' = @return_value
GO
```

ID	Name	Author
1	4	Кобзар
2	9	Кобзар

XV. Виклик і результат виконання збереженої процедури №15

```
USE [Library]
GO

DECLARE @return_value int

EXEC @return_value = [dbo].[Q15]

SELECT 'Return Value' = @return_value
GO
```

	Name	Author	Popularity
1	Гарри Поттер	Роулінг	3
2	Кривой Домишко	Агата Крісті	2
3	Кобзар	Кіів	2
4	Катерина	Шевченко	2
5	Как приручить дракона	Федько	1
6	Заповіт	Шевченко	1

18. Аналізуючи дані створеної бази і результати виклику запитів, впевнюємось, що вони працюють так як треба і дістають потрібні дані з створеної бази

19. Створимо програму, яка дозволить користувачу ввести параметри запиту, виконає його та виведе результат. Для її реалізації обираємо мову C# з використанням платформи .NetFramework в середовищі програмування Visual Studio 2015.

I. Створюємо новий проект. Додаємо до нього форму. Форма буде містити такі елементи для взаємодії з користувачем:

- **tabControl1** – графічний контейнер типу TabControl, який містить 15 вкладок (по одній на кожен запит). Кожна вкладка містить елементи:

1. **fillToolStrip** – складається з елементів типу Label, TextBox, Button – дозволяє користувачу ввести параметри запиту та виконати його.

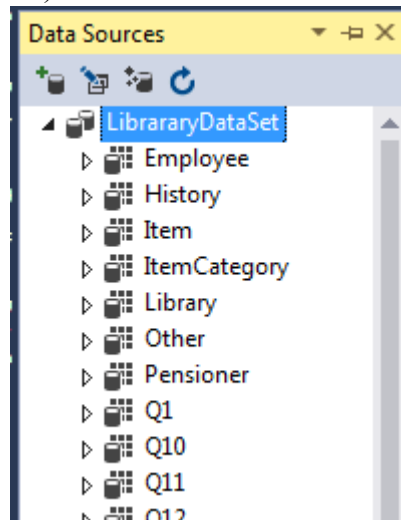
2. **DataGridView** – виводить результат запиту, має вигляд таблиці

- **Button1** – “Set Default Setting” задає стандартні параметри запитів (для полегшення демонстрації роботи програми)

	Name	Souname	Facult	Course	Organization	Reserch
*						

II. Приєднуємо базу даних **Library** (знаходиться на локальному сервері SQL). Для цього на панелі Data Source натискаєм Add New Data Source і, слідуючи майстру налаштування, прив'язуєм раніше створену базу даних Library з серверу SQL.

III. Після цього на панелі Data Source з'являться вибрані таблиці і збереженні процедури. (Додатково збережену процедуру №7 розіб'єм на дві по одній на кожен підзапит)



IV. Перетягуєм необхідні елементи і встановлюєм зв'язки, сукупне використання яких дозволить динамічно формувати запит, виконувати його, і виводити результат в відповідні таблиці.



V. Запускаєм програму і тестуєм. Програма успішно зв'язалась з базою даних, виконала запити з введеними користувачем параметрами, та вивела результати в відповідні таблиці DataGridView.

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

ReaderCategory: Student Institution: ЧНУ Fill

	Name	Souname	Facult	Course
►	Федір	Холодний	КН	2
	Камінь	Лежачий	УПУ	4
	Васи́лій	Сонячний	КН	3
	Марко	Дерев'яний	ПЗ	3
*				

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

name: Кривой Домишк author: Агата Крісті Fill

	Souname	Name
►	Пісова	Мавка
*		

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

name: Марко surname: Деревяний dateFrom: 01.09.2016 dateTo: 01.10.2016 Fill

	Souname	Name	Name1	Author
▶	Деревяний	Марко	Катерина	Шевченко
	Деревяний	Марко	Кобзар	Київ
*				

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

name: Марко surname: Деревяний dateFrom: 01.09.2016 dateTo: 01.10.2016 Fill

	Souname	Name	Name1	Author
▶	Деревяний	Марко	Катерина	Шевченко
*				

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

bench: 11

	Name	Author	Reader Name	Reader Souname
▶	Заповіт	Шевченко	Чохлик	Невмирущий
	Катерина	Шевченко	Марко	Деревяний
*				

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

name: Україн surname: Крпак dateFrom: 01.10.2016 dateTo: 31.12.2016

	Souname	Name
▶	Деревяний	Марко
	Лежачий	Камінь
*		

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

name: surname: Amount:

	Sourname	Name	Amount
▶	Комп'ютерний	Вірус <input type="button" value="Fill"/>	1
	Домашній	Кіт	1
	Фараон	Микита	1
	Дворовий	Пес	2
	Віконний	Петро	2
	Кріпак	Україн	4
*			

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

	Sourname	Name	Name1	Author	Term
▶	Дерев'яний	Марко	Катерина	Шевченко	14.10.2016
	Лежачий	Камінь	Как приручить дракона	Федько	12.12.2016
	Лісова	Мавка	Кривой Домишко	Агата Крісті	12.12.2014
	Невмирущий	Чохлик	Заповіт	Шевченко	12.12.2016
*					

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

dateFrom: 01.01.2008 dateTo: 31.12.2008 Fill

	Name	Author
▶	Как приручить дракона	Федько
	Гарри Поттер	Роулінг
	Заповіт	Шевченко
*		

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

dateFrom: 01.01.2016 dateTo: 31.12.2016 Fill

	Name	Author
▶	Кривой Домишко	Агата Крісті
	Основи програмування	Шилдт
	Кобзар	Київ
	Гарри Поттер	Роулінг
	Катерина	Шевченко
	Кобзар	Шевченко
*		

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

lib: 2 room: 2 Fill

	Soumame	Name
▶	Кріпак	Укрпін
	Домашній	Кіт
	Філіпишн	Філіп
*		

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

dateFrom: 01.09.2016 dateTo: 01.10.2016 Fill

	Name	Soumame	LastVisit
▶	Василій	Сонячний	11.12.2016
	Василь	Проточний	10.10.2016
	Камінь	Лежачий	09.11.2016
	Мавка	Лісова	11.11.2014
	Петро	Шкільний	08.08.2016
*			

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

name: Заповіт auth: Шевченко Fill

	ID	Name	Author
▶	4	Кобзар	Київ
*			

Set Default Setting

Form1

Query1 Query2 Query3 Query4 Query5 Query6 Query7 Query8 Query9 Query10 Query11 Query12 Query13 Query14 Query15

auth: Шевченко Fill

	ID	Name	Author
▶	4	Кобзар	Київ
	9	Кобзар	Шевченко
*			

Set Default Setting

	Name	Author	Popularity
▶	Гарри Поттер	Роулінг	3
	Кривой Домишко	Агата Крісті	2
	Кобзар	Київ	2
	Катерина	Шевченко	2
	Как приручить дракона	Федько	1
	Заповіт	Шевченко	1
*			

Set Default Setting

Програма успішно зв'язалась з базою даних, виконала запити з введеними користувачем параметрами, та вивела результати в відповідні таблиці DataGridView.

Висновки: У результаті виконання поставленого завдання я навчився створювати базу даних у середовищі MS Management Studio, проектувати її таблиці та зв'язки між ними, створювати різноманітні запити та збереженні процедури для маніпулювання таблицями та їх вмістимим. Також я навчився звертатися до бази даних на сервері SQL через засоби .NetFramework, та розробив програму, яка дозволяє користувачу задавати параметри запитів та виконувати їх через простий інтерфейс взаємодії, користувачу достатньо заповнити необхідні поля та натиснути кнопку, після чого програма виконає запит та виведе результат у вигляді таблиці.