Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра АСОІУ

**ЗВІТ**

про виконання комп’ютерного практикуму № 1

з дисципліни

“ Програмні додатки з використанням баз даних”

Варіант 6

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Cт.вик. Клименко О. М. |  | студент 3-го курсу  гр. ІП-51 ФІОТ  Зарічковий Олександр Анатолійович |

Київ – 2017

**ЗМІСТ:**

[1 Код програми 3](#_Toc476750393)

[2 Графіки Складностей алгоритмів 23](#_Toc476750394)

[3 Висновок 24](#_Toc476750395)

# Завдання

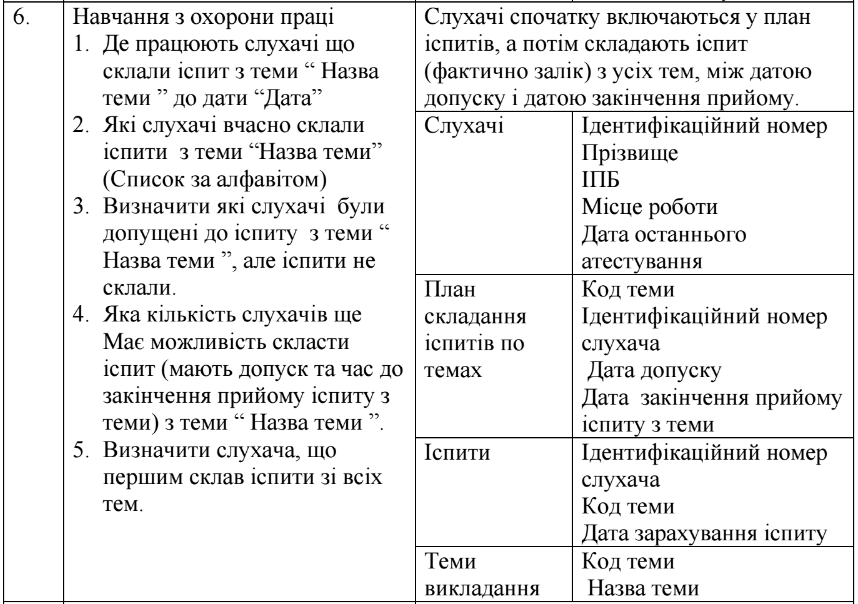
**Мета роботи**

Розвиток навику складання аналітичних запитів на основі текстових формулювань потреб користувачів.

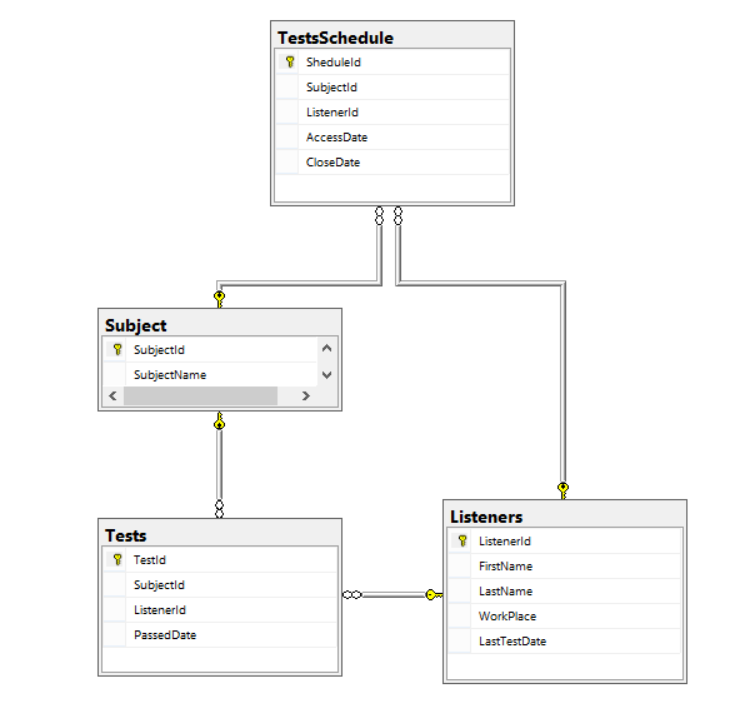
1. Створити базу даних по варіанту. Відобразити її в ER-діаграмі або у скріншоті схеми БД.
2. Написати 4 запити наступних типів:

* з використанням агрегуючих функцій та конструкції HAVING.
* з перетворенням типу даних результату запиту або формату дати
* з пошуком по фрагменту текстового поля. (Ех: знайти всі прізвища, що закінчуються на «ко»)
* зі вставкою будь-якого значення в поле результату запиту, що набуло значення NULL в результаті запиту (Ех: якщо прізвище «AUDI», що купив, - NULL, поставити символ «\_» оператор NVL або CASE ).

1. Виконати завдання 1-5 по варіанту:



# Схема бази даних



# Виконання Завдання

1. Завдання по написанню запитів
   1. Запит з використанням агрегуючих функцій та конструкції HAVING:  
      *Призначення*: Виводить ідентифікатори всіх студентів, які склали всі іспити та дати складання останнього іспиту.

*Текст запиту*:

SELECT [ListenerId],

MAX([PassedDate]) AS [Worth\_date]

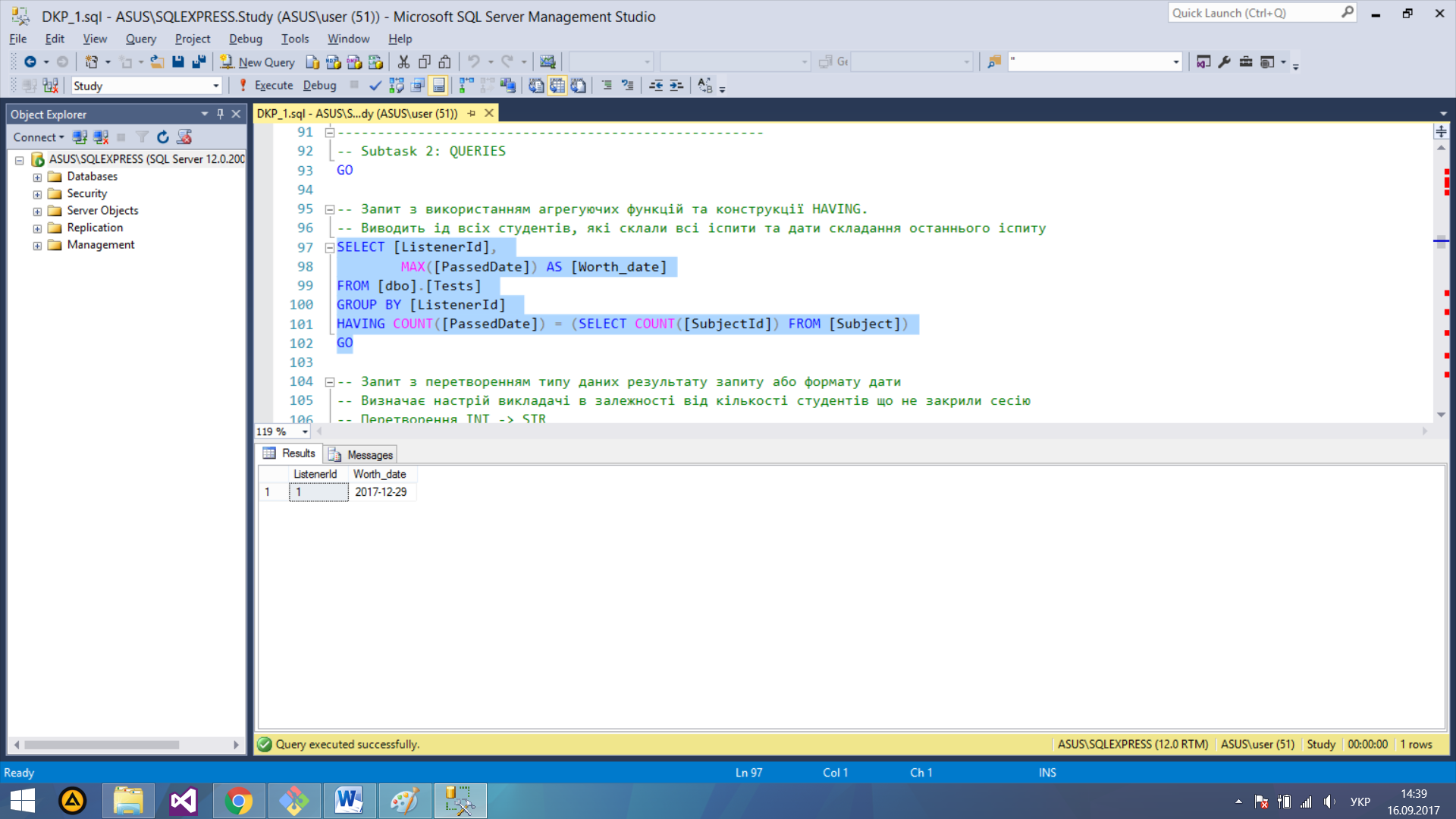
FROM [dbo].[Tests]

GROUP BY [ListenerId]

HAVING COUNT([PassedDate]) = (SELECT COUNT([SubjectId]) FROM [Subject])

GO

*Результат виконання*:



* 1. Запит з перетворенням типу даних результату запиту або формату дати:

*Призначення*: Визначає настрій викладачів в залежності від кількості студентів що не закрили сесію. Перетворення INT -> STR.

*Текст запиту*:

SELECT CASE COUNT(\*)

WHEN 0 THEN '=)'

WHEN 1 THEN ';)'

ELSE ':('

END AS [Teachers\_mood]

FROM [dbo].TestsSchedule AS [TS]

INNER JOIN [dbo].[Listeners] AS [L]

ON [TS].[ListenerId] = [L].[ListenerId]

INNER JOIN [dbo].[Tests] AS [T]

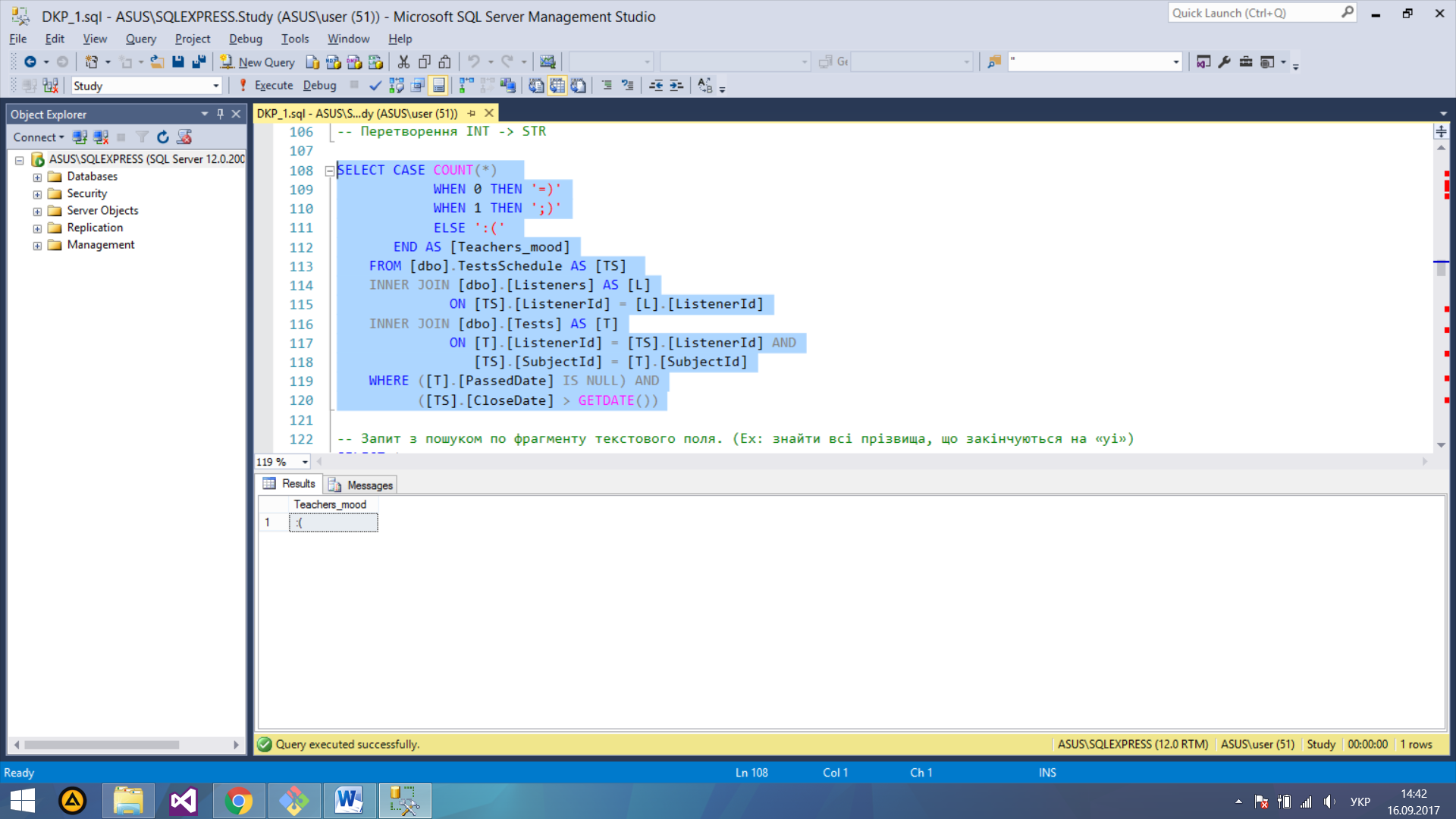
ON [T].[ListenerId] = [TS].[ListenerId] AND

[TS].[SubjectId] = [T].[SubjectId]

WHERE ([T].[PassedDate] IS NULL) AND

([TS].[CloseDate] > GETDATE())

*Результат виконання*:



* 1. Запит з пошуком по фрагменту текстового поля. (Ех: знайти всі прізвища, що закінчуються на «ко»):

*Призначення*: знайти всі прізвища, що закінчуються на «yi».

*Текст запиту*:

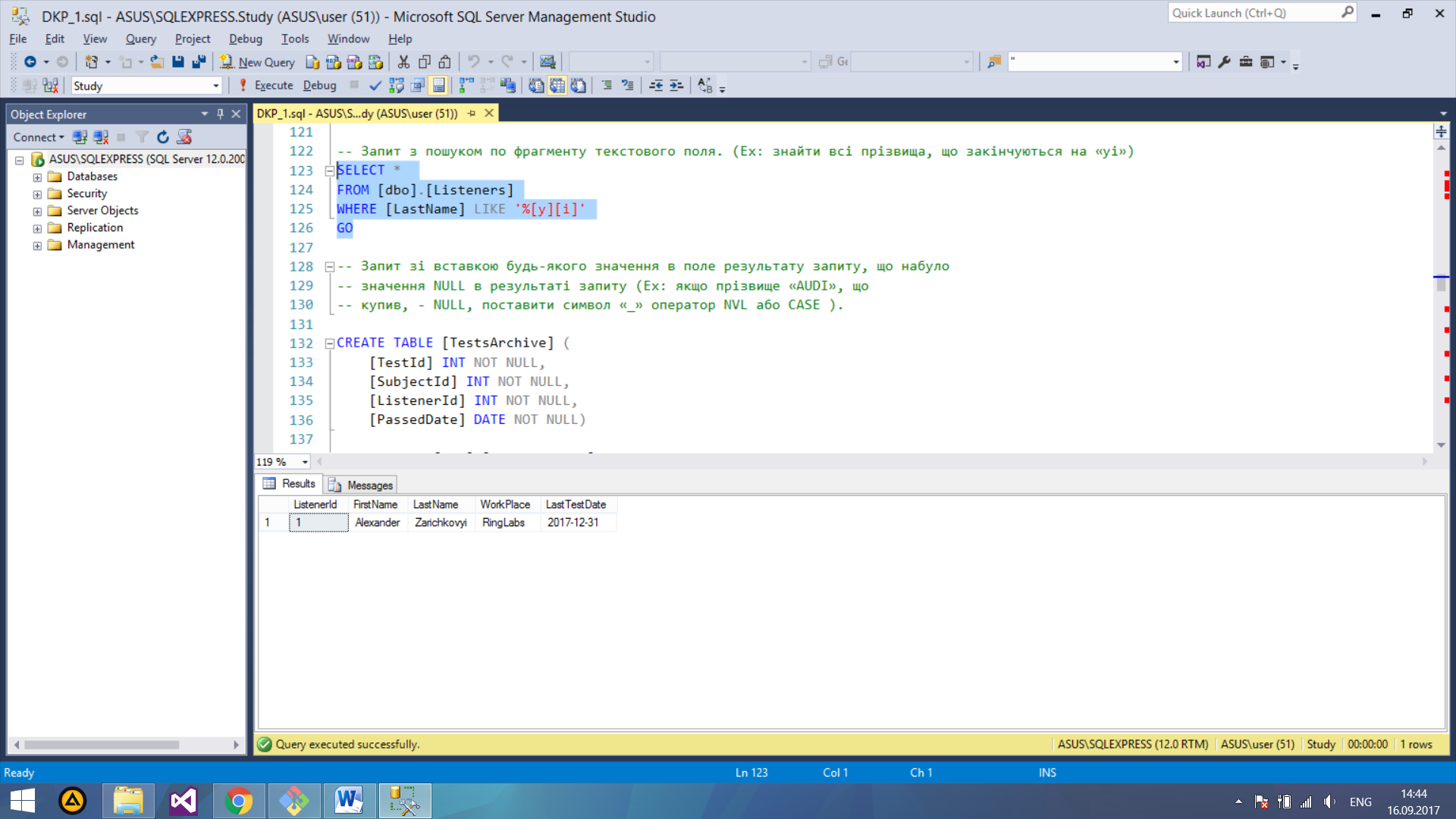
SELECT \*

FROM [dbo].[Listeners]

WHERE [LastName] LIKE '%[y][i]'

GO

*Результат виконання:*



* 1. Запит зі вставкою будь-якого значення в поле результату запиту, що набуло значення NULL в результаті запиту (Ех: якщо прізвище «AUDI», що купив, - NULL, поставити символ «\_» оператор NVL або CASE ):

*Призначення*: копіювання таблиці Tests в архівну таблицю TestsArchive з заміною усіх нульвих значень дат на фіксовану дату 01.01.1800.

*Текст запиту*:

CREATE TABLE [TestsArchive] (

[TestId] INT NOT NULL,

[SubjectId] INT NOT NULL,

[ListenerId] INT NOT NULL,

[PassedDate] DATE NOT NULL)

INSERT INTO [dbo].[TestsArchive]

SELECT [TestId]

,[SubjectId]

,[ListenerId]

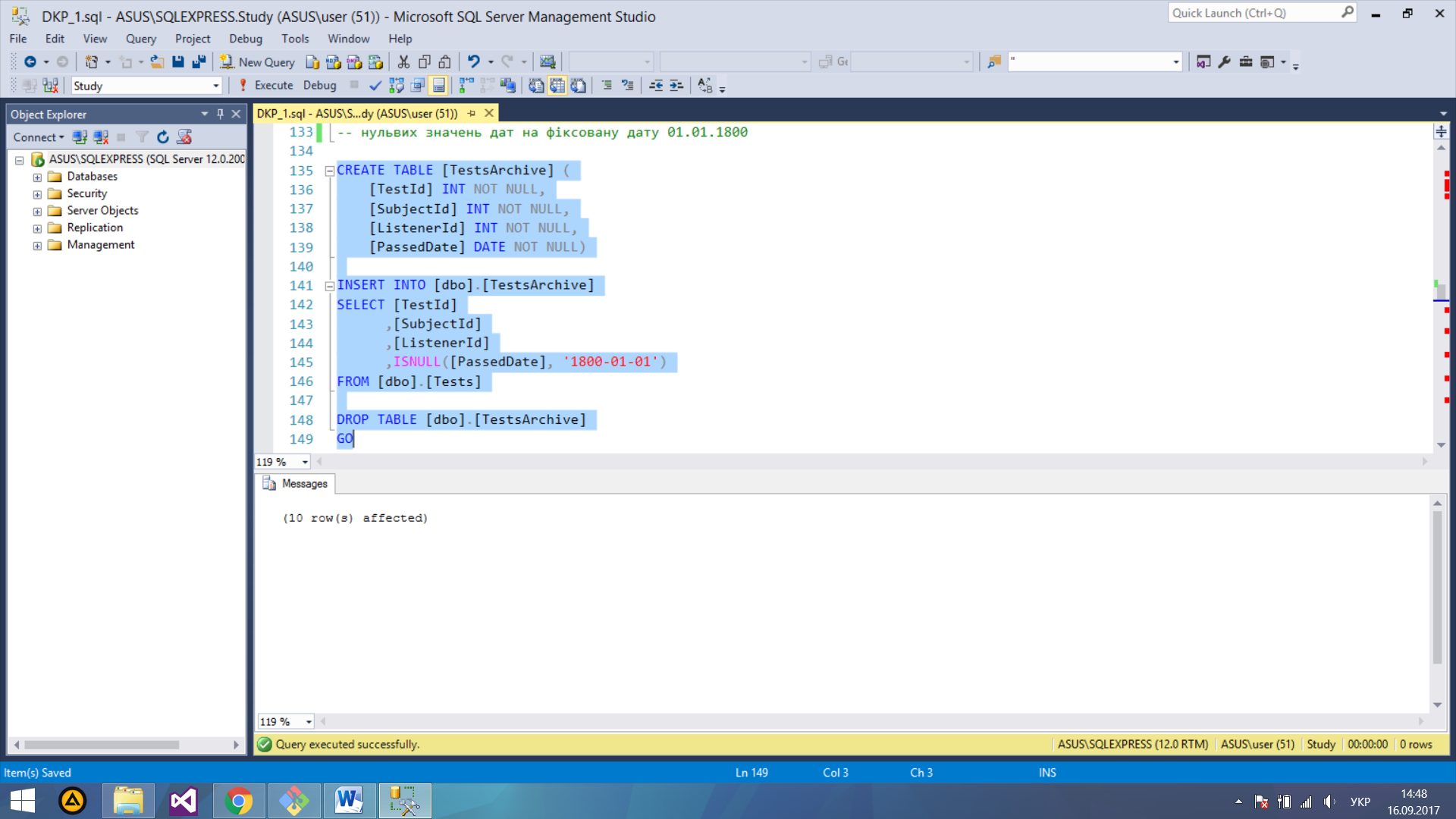
,ISNULL([PassedDate], '1800-01-01')

FROM [dbo].[Tests]

DROP TABLE [dbo].[TestsArchive]

GO

*Результат виконання:*



1. Завдання по написанню персональних запитів:
   1. *Призначення*: Де працюють слухачі що склали іспит з теми “ Назва теми ” до дати “Дата”.

*Текст запиту*:

CREATE PROCEDURE WORKPLACE\_LISTNERS\_PASSED\_SUBJECT

@subject\_id INT,

@pass\_date DATE

AS

SELECT [L].[FirstName],

[L].[LastName],

[L].[WorkPlace]

FROM [dbo].[Tests] AS [T]

INNER JOIN [dbo].[Listeners] AS [L]

ON [T].[ListenerId] = [L].[ListenerId]

WHERE [T].[SubjectId] = @subject\_id AND

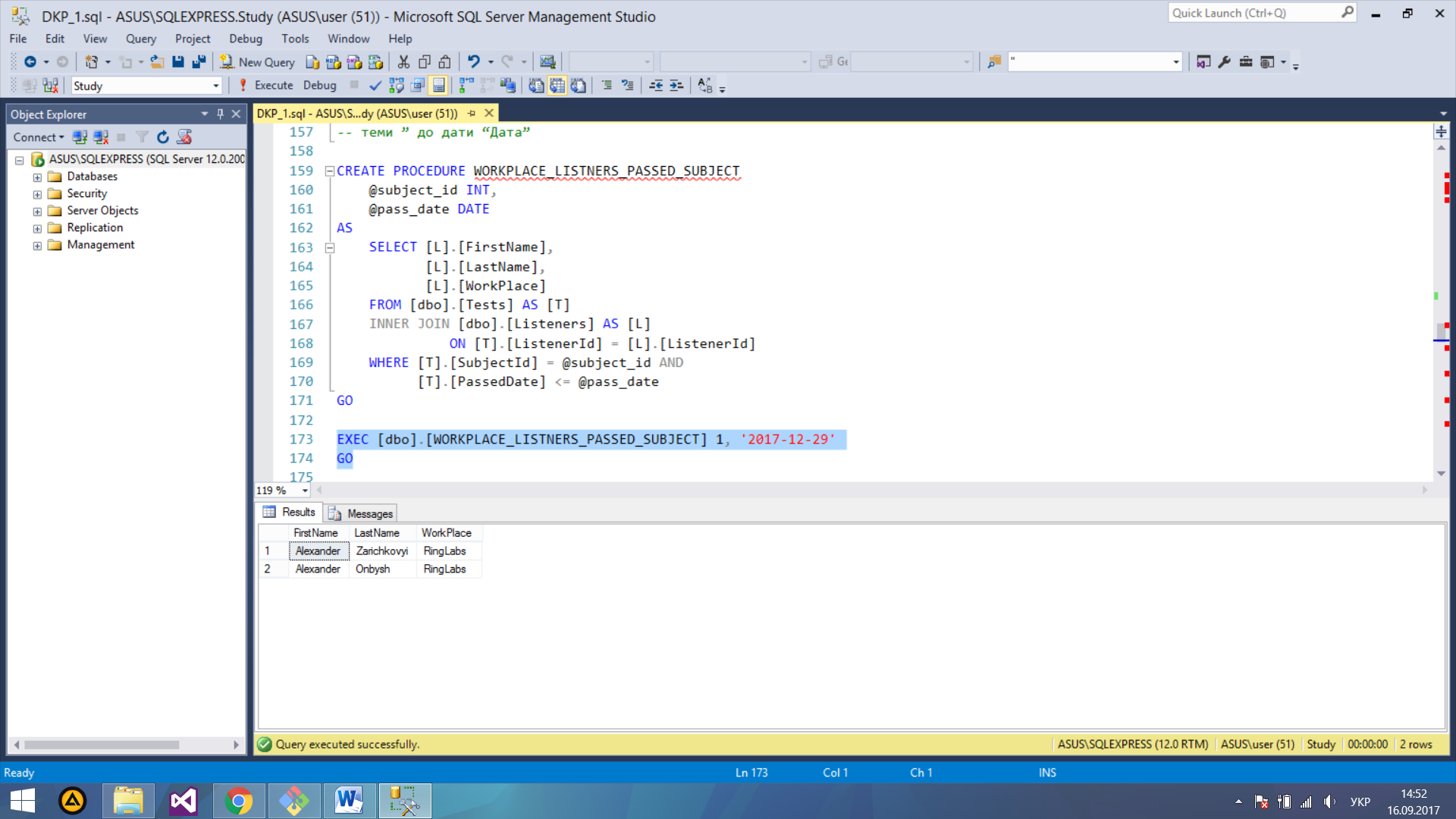
[T].[PassedDate] <= @pass\_date

GO

EXEC [dbo].[WORKPLACE\_LISTNERS\_PASSED\_SUBJECT] 1, '2017-12-29'

GO

*Результат виконання:*



* 1. *Призначення*: Які слухачі вчасно склали іспити з теми “Назва теми” (Список за алфавітом)

*Текст запиту*:

CREATE PROCEDURE PASSED\_STUDENTS

@subject\_id INT

AS

SELECT [L].[FirstName],

[L].[LastName]

FROM [dbo].TestsSchedule AS [TS]

INNER JOIN [dbo].[Listeners] AS [L]

ON [TS].[ListenerId] = [L].[ListenerId]

INNER JOIN [dbo].[Tests] AS [T]

ON [T].[ListenerId] = [TS].[ListenerId] AND

[TS].[SubjectId] = [T].[SubjectId]

WHERE [TS].[SubjectId] = @subject\_id AND

([T].[PassedDate] BETWEEN [TS].[AccessDate] AND [TS].[CloseDate])

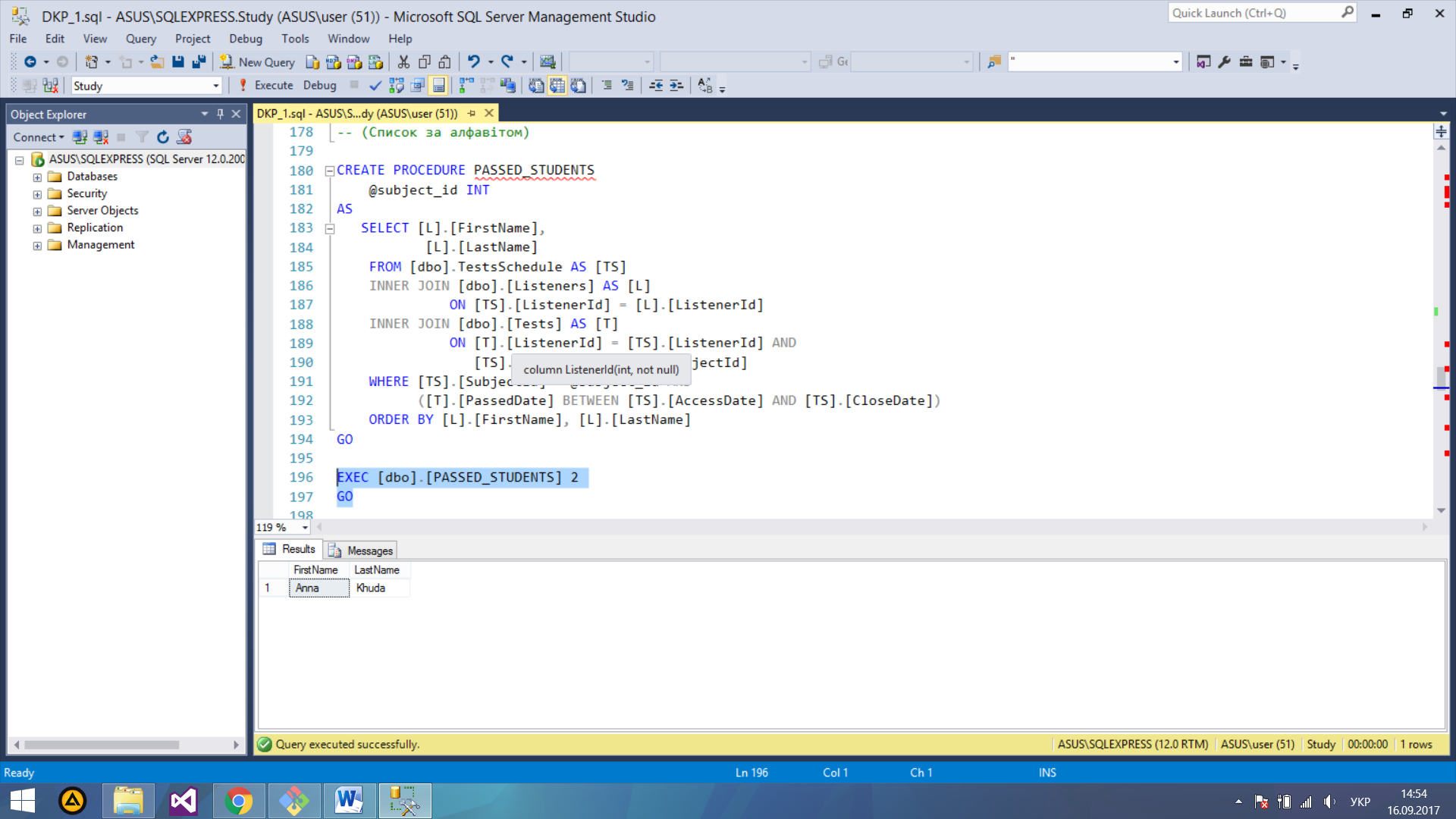
ORDER BY [L].[FirstName], [L].[LastName]

GO

EXEC [dbo].[PASSED\_STUDENTS] 2

GO

*Результат виконання:*



* 1. *Призначення*: Визначити які слухачі булидопущені до іспиту з теми “Назва теми ”, але іспити несклали.

*Текст запиту*:

CREATE PROCEDURE NOT\_PASSED\_STUDENTS

@subject\_id INT

AS

SELECT [L].[FirstName],

[L].[LastName]

FROM [dbo].TestsSchedule AS [TS]

INNER JOIN [dbo].[Listeners] AS [L]

ON [TS].[ListenerId] = [L].[ListenerId]

INNER JOIN [dbo].[Tests] AS [T]

ON [T].[ListenerId] = [TS].[ListenerId] AND

[TS].[SubjectId] = [T].[SubjectId]

WHERE [TS].[SubjectId] = @subject\_id AND

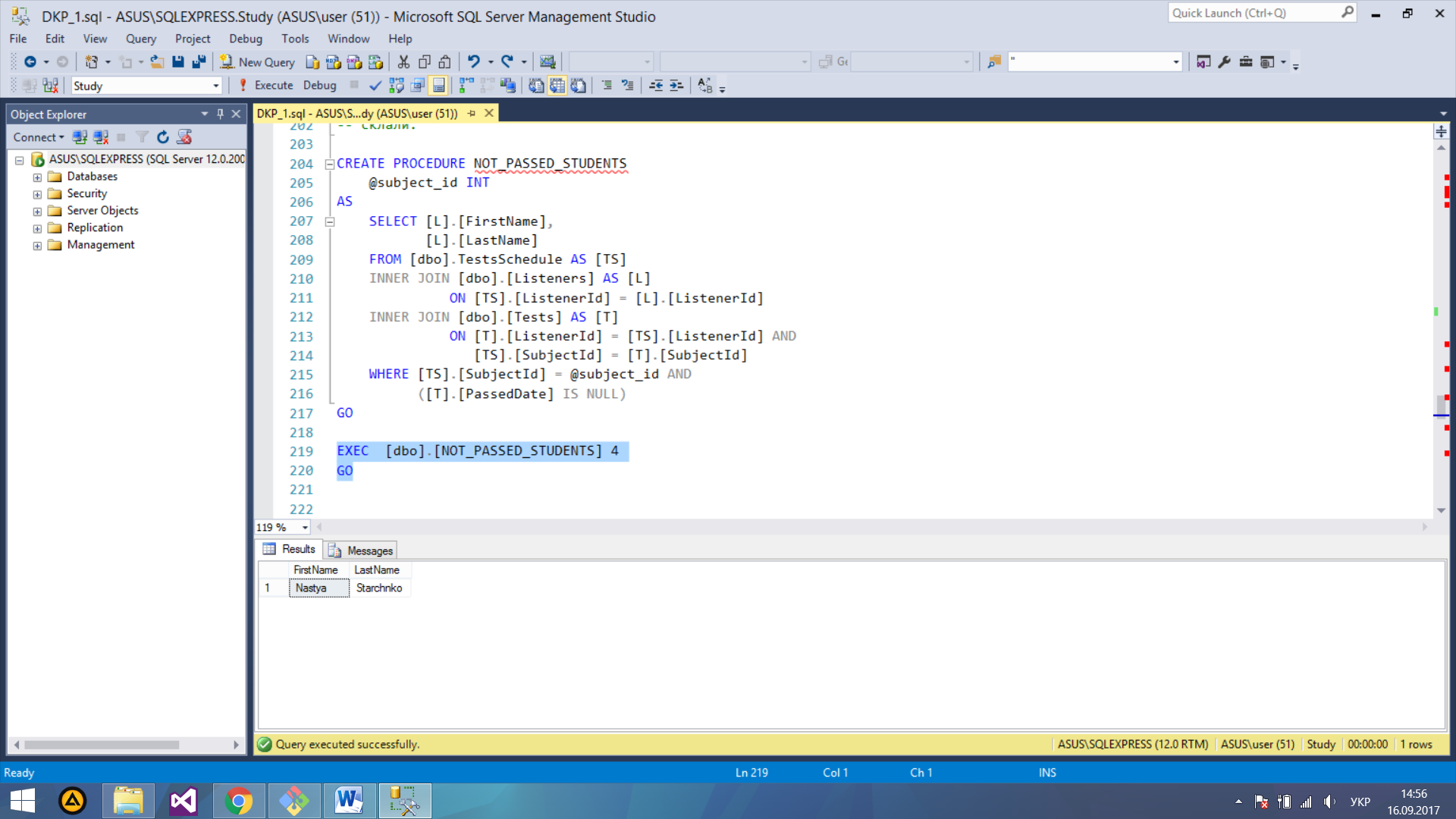
([T].[PassedDate] IS NULL)

GO

EXEC [dbo].[NOT\_PASSED\_STUDENTS] 4

GO

*Результат виконання:*



* 1. *Призначення*: Яка кількість слухачів ще має можливість скласти іспит (мають допуск та час до закінчення прийому іспиту з теми) з теми “ Назва теми ”.

*Текст запиту*:

CREATE PROCEDURE ABLE\_TO\_PASS\_STUDENTS\_COUNT

@subject\_id INT

AS

SELECT COUNT(\*) AS [Quantity\_of\_students\_able\_to\_pass\_subject]

FROM [dbo].TestsSchedule AS [TS]

INNER JOIN [dbo].[Listeners] AS [L]

ON [TS].[ListenerId] = [L].[ListenerId]

INNER JOIN [dbo].[Tests] AS [T]

ON [T].[ListenerId] = [TS].[ListenerId] AND

[TS].[SubjectId] = [T].[SubjectId]

WHERE [TS].[SubjectId] = @subject\_id AND

([T].[PassedDate] IS NULL) AND

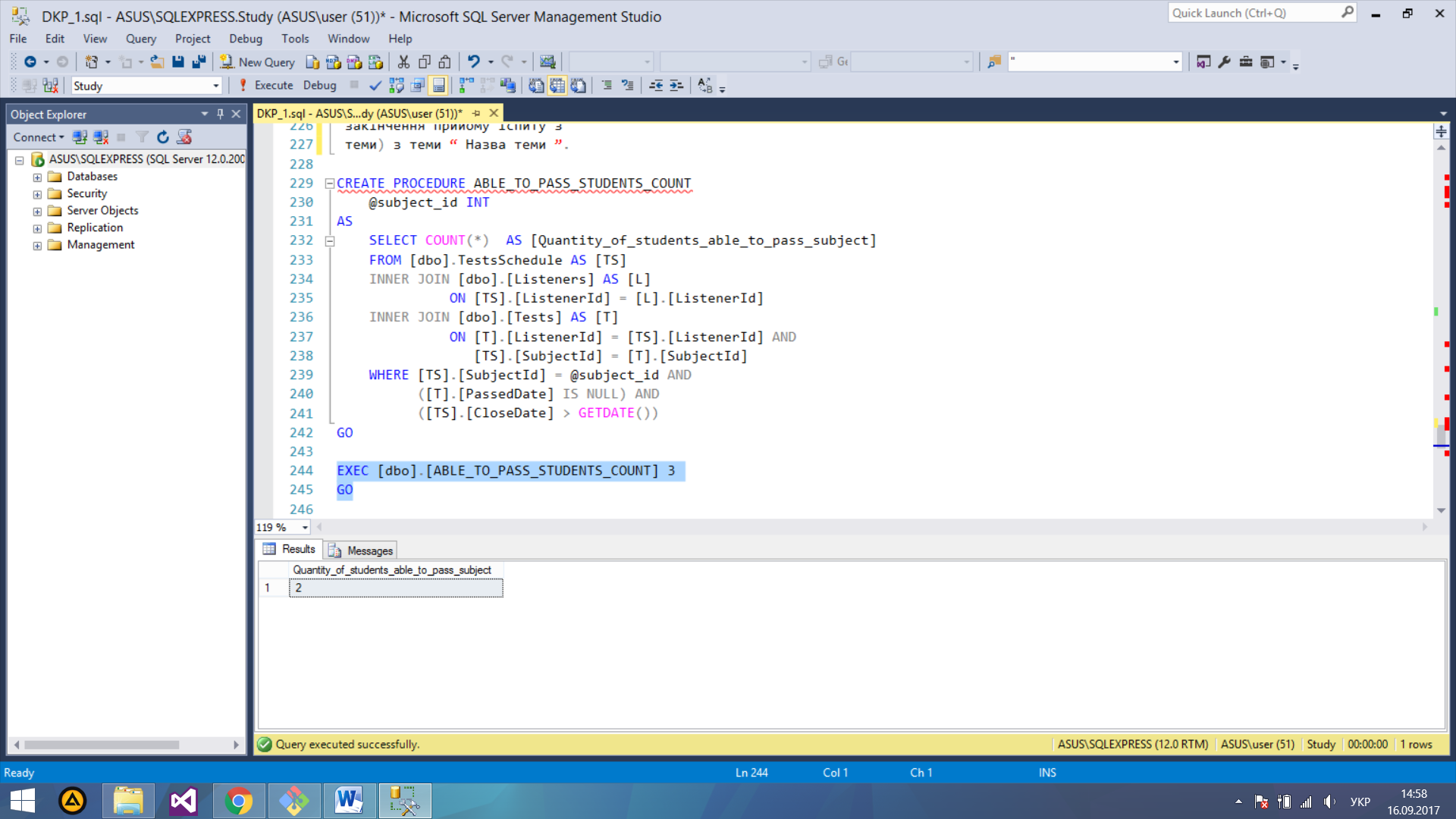
([TS].[CloseDate] > GETDATE())

GO

EXEC [dbo].[ABLE\_TO\_PASS\_STUDENTS\_COUNT] 3

GO

*Результат виконання:*



* 1. *Призначення*: Визначити слухача, що першим склав іспити зі всіх тем.

*Текст запиту*:

CREATE PROCEDURE BEST\_STUDENT

AS

SELECT TOP(1)

[L].[FirstName],

[L].[LastName]

FROM [dbo].[Listeners] as [L]

INNER JOIN (

SELECT [ListenerId],

MAX([PassedDate]) AS [Worth\_date],

COUNT([PassedDate]) AS [Subject\_count]

FROM [dbo].[Tests]

GROUP BY [ListenerId]

) AS [B]

ON [B].[ListenerId] = [L].[ListenerId]

WHERE [B].[Subject\_count] = (SELECT COUNT([SubjectId]) FROM [Subject])

ORDER BY [B].[Worth\_date]

GO

EXEC BEST\_STUDENT

GO

*Результат виконання:*

