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 1

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Cт.вик. Клименко О. М. |  | студент 3-го курсу  гр. ІП-51 ФІОТ  Булатов Дмитро Єгорович |

Київ – 2017

**ЗМІСТ:**

[1 Завдання 3](#_Toc494139084)

[2 Схема бази даних 5](#_Toc494139085)

[3 Виконання Завдання 6](#_Toc494139086)

# Завдання

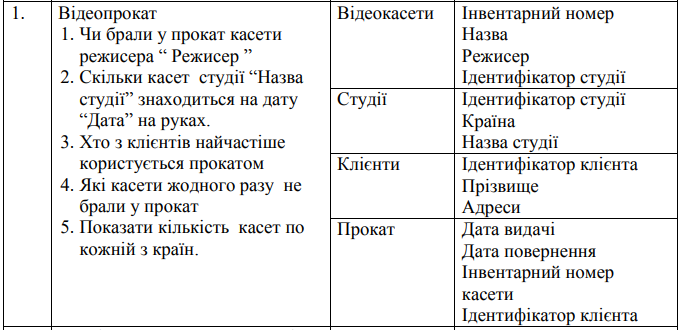
**Мета роботи**

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

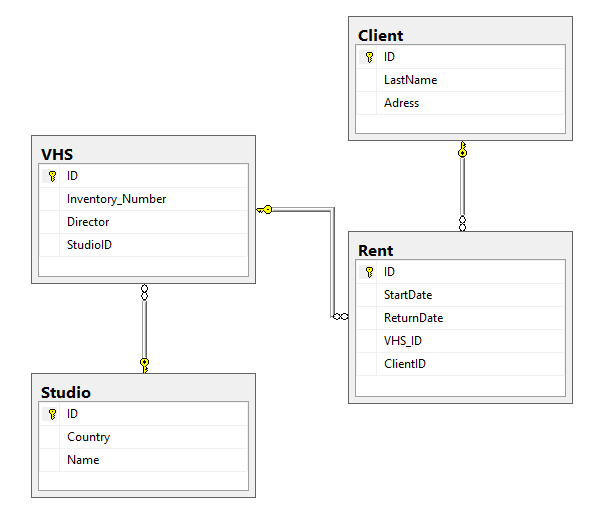
1. Створити базу даних по варіанту. Відобразити її в ER-діаграмі або у скріншоті схеми БД.
2. Написати 4 запити наступних типів:

* 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. Створення БД

CREATE DATABASE DB2

GO

use DB2

CREATE TABLE Studio(

ID int NOT NULL PRIMARY KEY IDENTITY(1,1),

Country char(2) NOT NULL,

Name varchar(40) NOT NULL

)

CREATE TABLE VHS(

ID int NOT NULL PRIMARY KEY IDENTITY(1,1),

Inventory\_Number int NOT NULL UNIQUE,

Director varchar(40),

StudioID int FOREIGN KEY REFERENCES Studio(ID)

)

CREATE TABLE Client(

ID int NOT NULL PRIMARY KEY IDENTITY(1,1),

LastName varchar(40),

Adress varchar(100)

)

CREATE TABLE Rent(

ID int NOT NULL PRIMARY KEY IDENTITY(1,1),

StartDate DateTime NOT NULL,

ReturnDate DateTime DEFAULT(NULL),

VHS\_ID int NOT NULL FOREIGN KEY REFERENCES VHS(ID),

ClientID int NOT NULL FOREIGN KEY REFERENCES Client(ID)

)

1. Завдання по написанню запитів
   1. Запит з використанням агрегуючих функцій та конструкції HAVING:  
      *Призначення*: Виводить ідентифікатори всіх користувачів, які хоч 1 раз орендували касету та кількість оренд.

*Текст запиту*:

SELECT Client.ID, Count(Rent.ID) AS 'Rented'

FROM Client

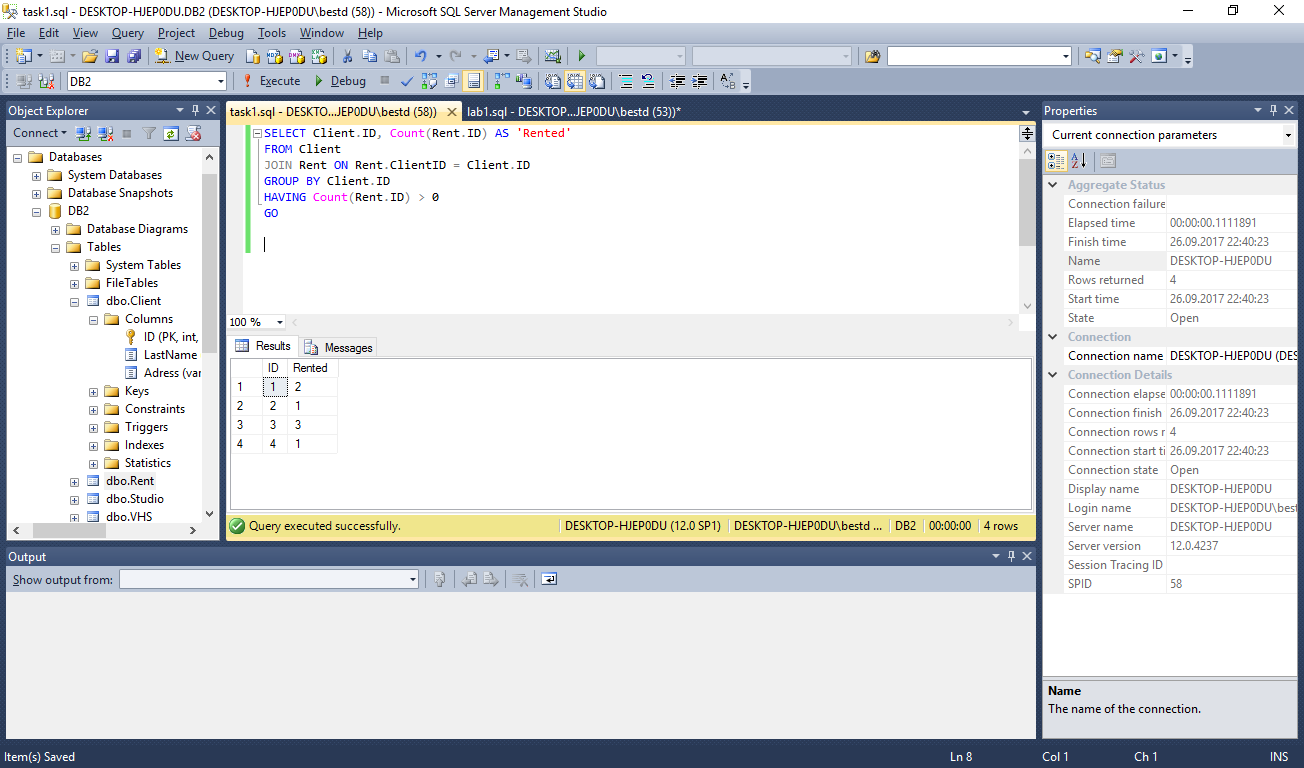
JOIN Rent ON Rent.ClientID = Client.ID

GROUP BY Client.ID

HAVING Count(Rent.ID) > 0

GO

*Результат виконання*:



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

*Призначення*: Визначає статус користувача

*Текст запиту*:

SELECT Client.ID, Count(Rent.ID) AS 'Rented',

CASE Count(Rent.ID)

WHEN 0 THEN 'Never rented anything'

WHEN 1 THEN 'Ordered once'

WHEN 2 THEN 'Ordered twice'

ELSE 'Regular customer'

END AS 'Customer status'

FROM Client

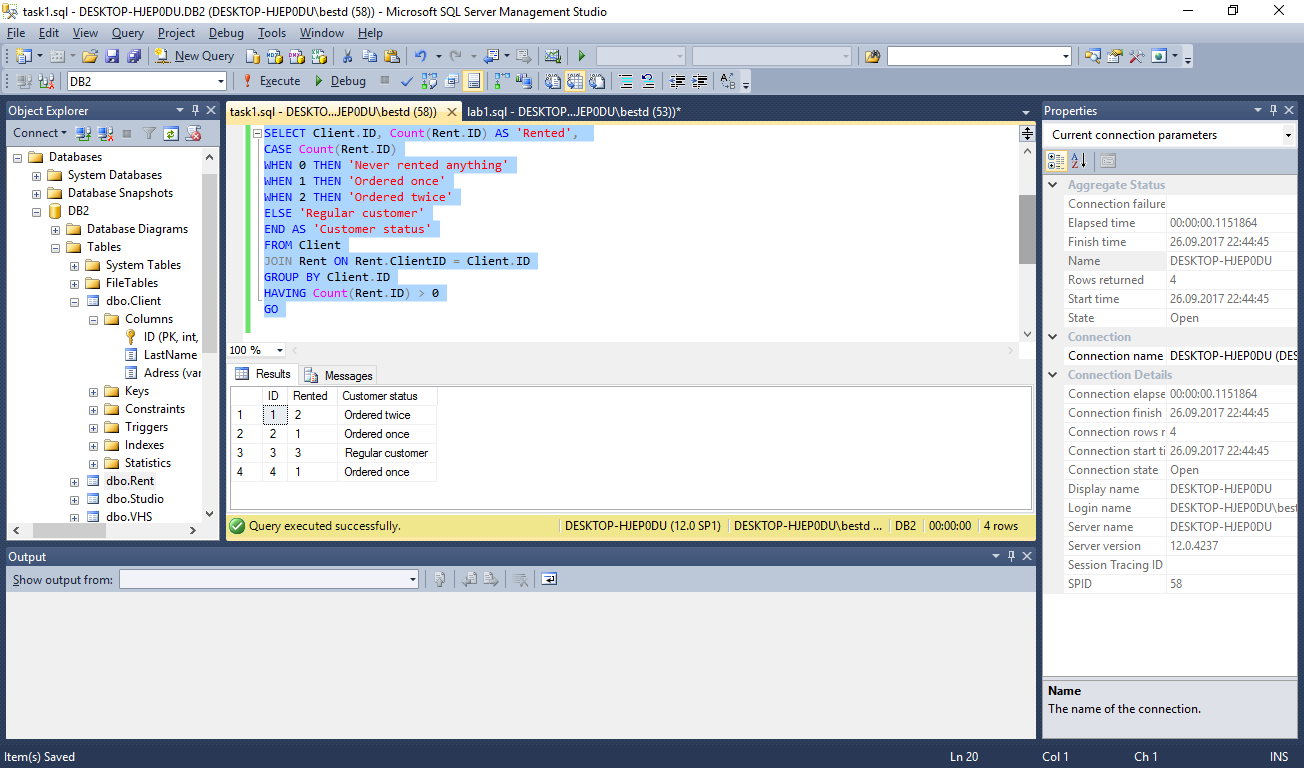
JOIN Rent ON Rent.ClientID = Client.ID

GROUP BY Client.ID

HAVING Count(Rent.ID) > 0

GO

*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».

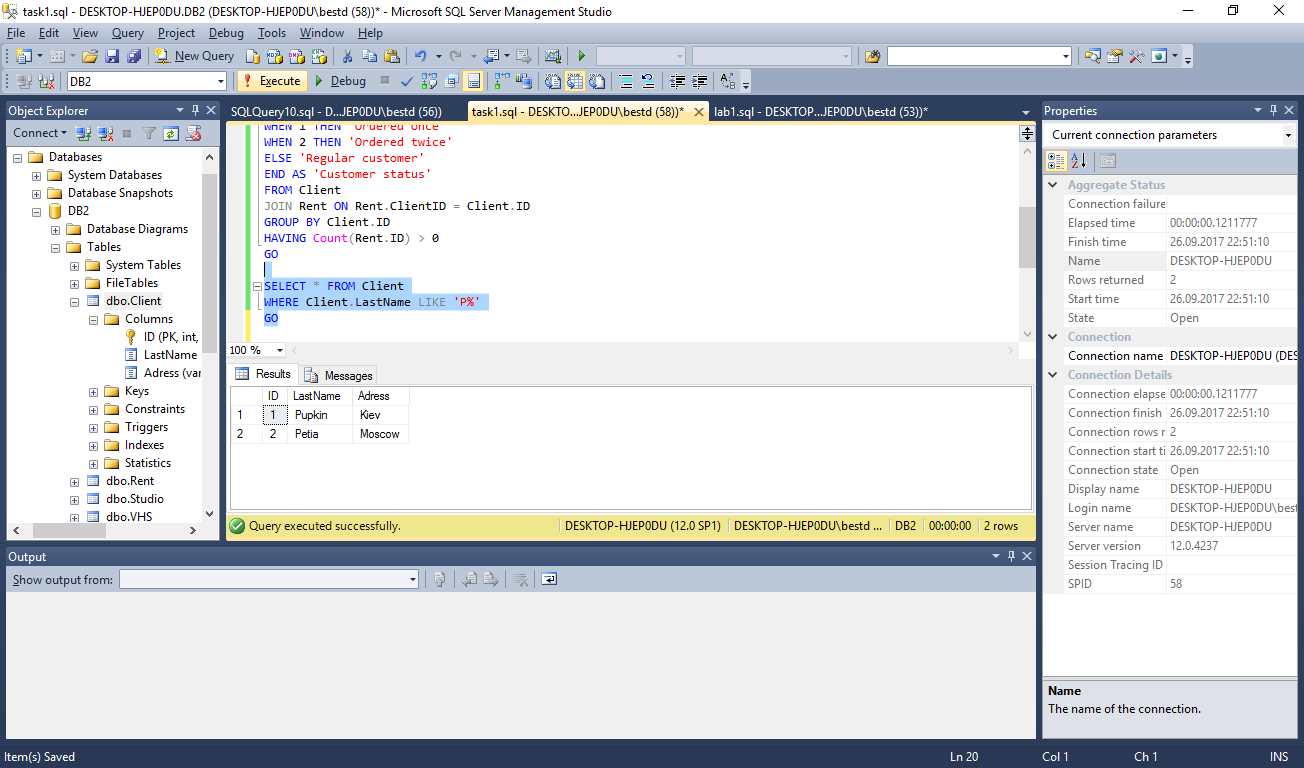
*Текст запиту*:

SELECT \* FROM Client

WHERE Client.LastName LIKE 'P%'

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*: Створює копію оренди касети та вказує що вона була щойно повернута.

*Текст запиту*:

INSERT INTO Rent(ClientID,VHS\_ID,StartDate,ReturnDate)

SELECT ClientID,VHS\_ID,StartDate,

CASE ReturnDate

WHEN NULL THEN GETDATE()

ELSE ReturnDate

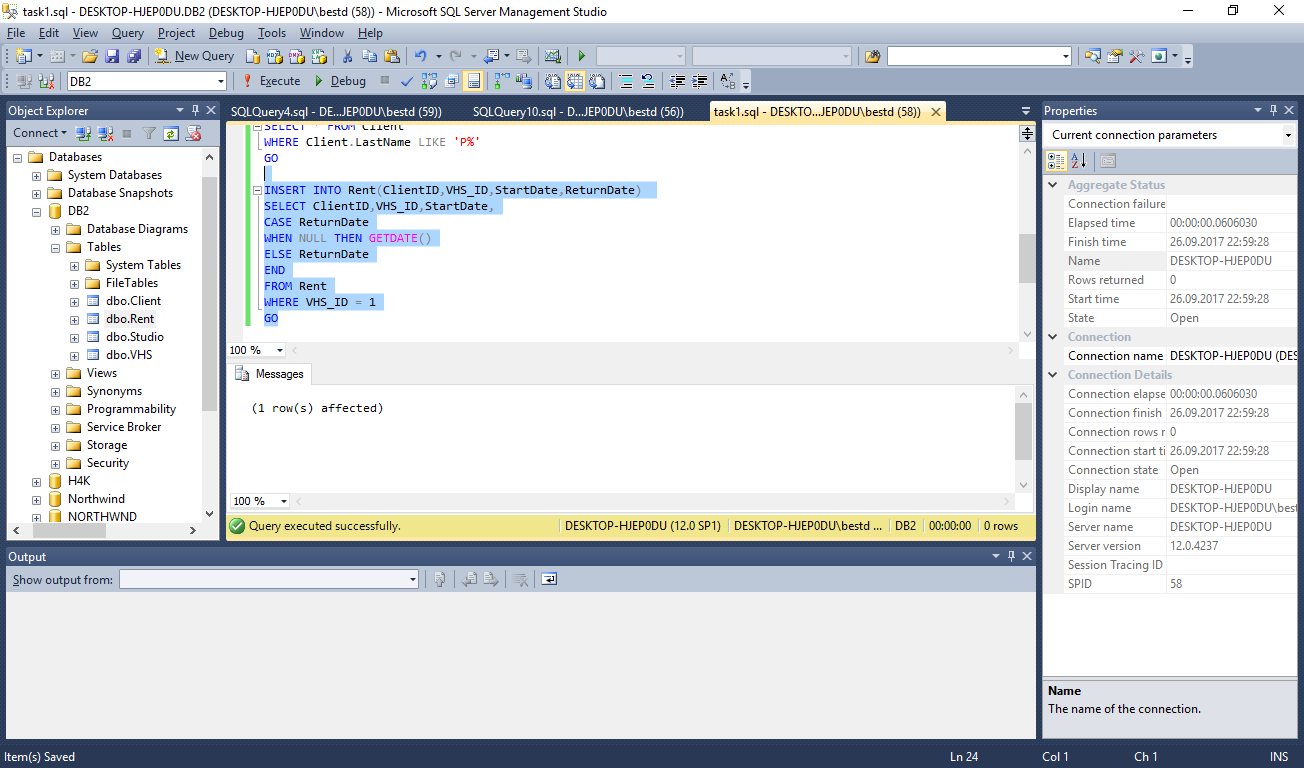
END

FROM Rent

WHERE VHS\_ID = 1

GO

*Результат виконання:*



1. Завдання по написанню персональних запитів:
   1. *Призначення*: Чи брали у прокат касети режисера “ Режисер ”

*Текст запиту*:

CREATE PROCEDURE dbo.GetDirectorRents @Director varchar(40)

AS

SELECT

CASE COUNT(\*)

WHEN 0 THEN 'No'

ELSE 'Yes'

END

FROM Rent

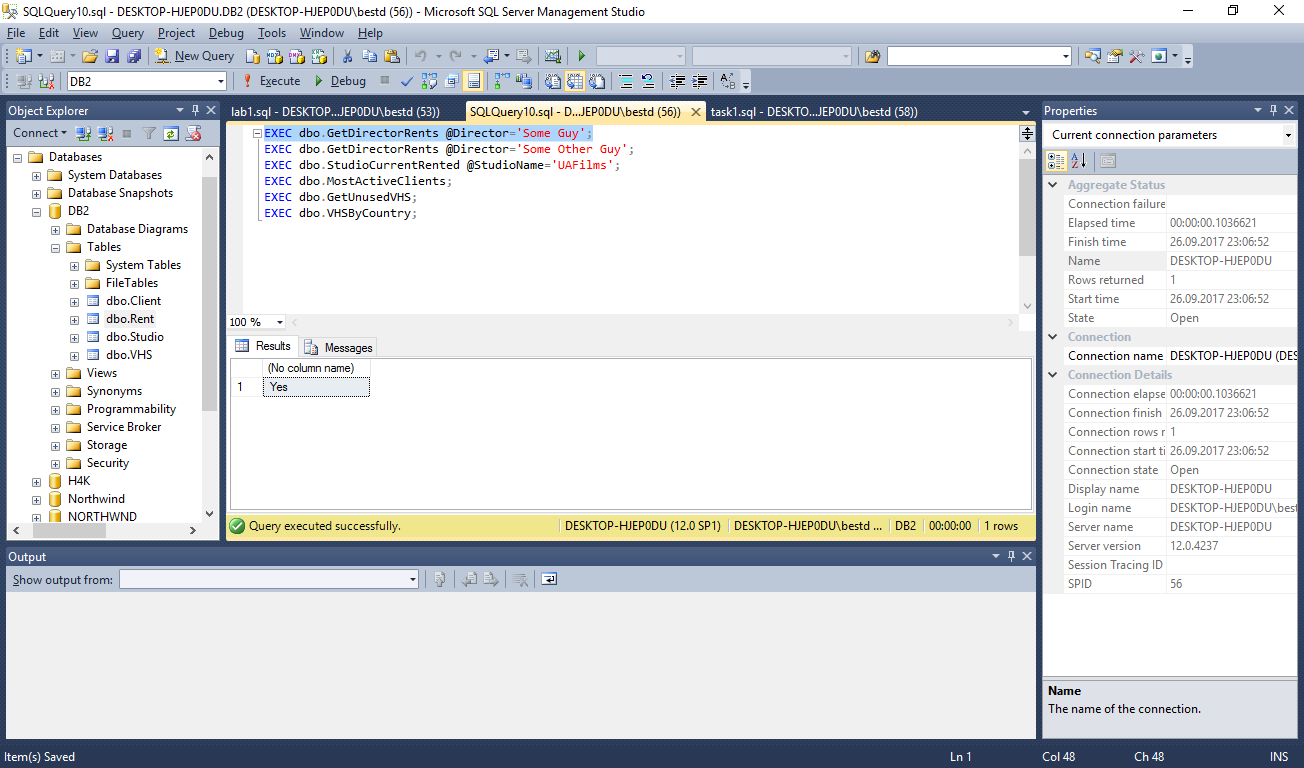
JOIN VHS ON Rent.VHS\_ID = VHS.ID

WHERE Director = @Director

GO

EXEC dbo.GetDirectorRents @Director='Some Guy';

*Результат виконання:*



* 1. *Призначення*: Скільки касет студії “Назва студії” знаходиться на дату “Дата” на руках.

*Текст запиту*:

CREATE PROCEDURE dbo.StudioCurrentRented @StudioName varchar(40)

AS

SELECT COUNT(Rent.ID) FROM Rent

JOIN VHS ON Rent.VHS\_ID = VHS.ID

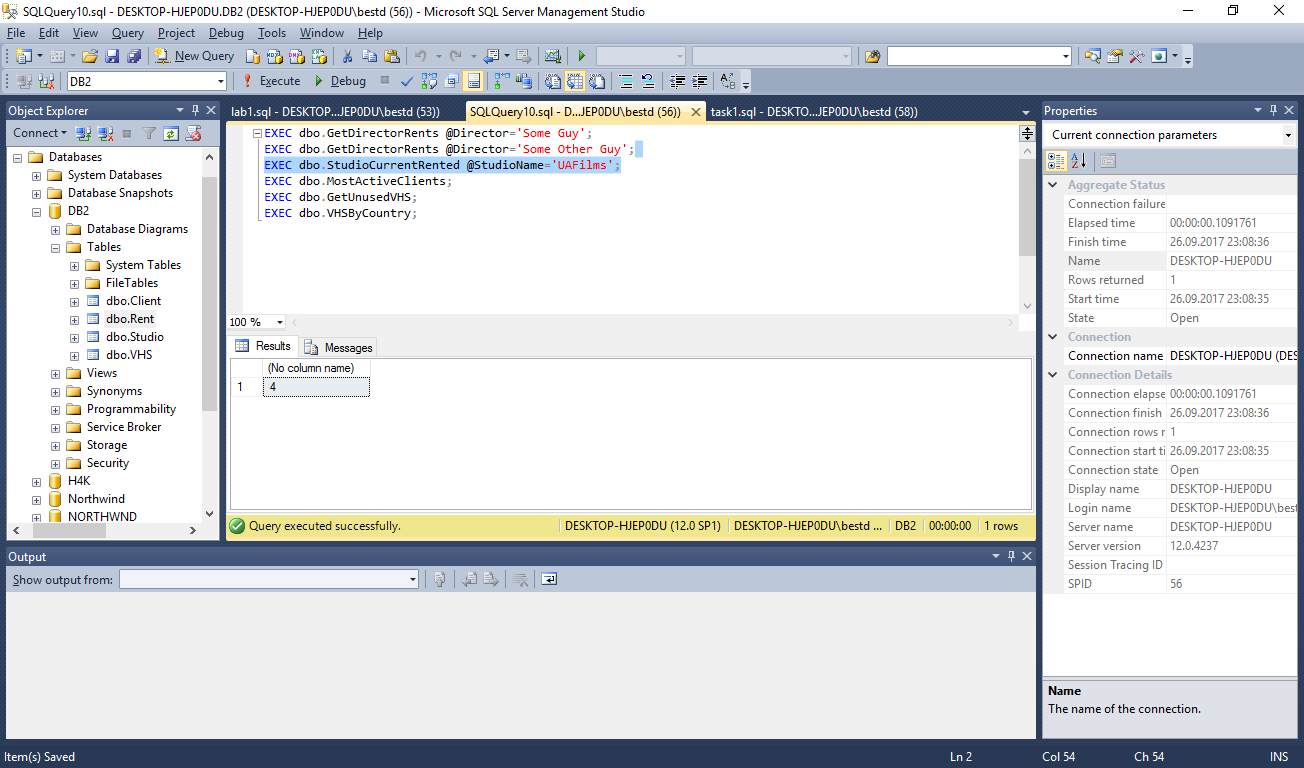
JOIN Studio ON VHS.StudioID = Studio.ID

WHERE Studio.Name LIKE @StudioName AND Rent.ReturnDate IS NULL

GO

EXEC dbo.StudioCurrentRented @StudioName='UAFilms';

*Результат виконання:*



* 1. *Призначення*: Хто з клієнтів найчастіше користується прокатом

*Текст запиту*:

CREATE PROCEDURE dbo.MostActiveClients

AS

SELECT LastName,COUNT(Rent.ID) FROM Client

JOIN Rent ON Rent.ClientID = Client.ID

GROUP BY Client.ID, LastName

HAVING COUNT(Rent.ID) = (

SELECT TOP(1) COUNT(Rent.ID) FROM Client

JOIN Rent ON Rent.ClientID = Client.ID

GROUP BY Client.ID

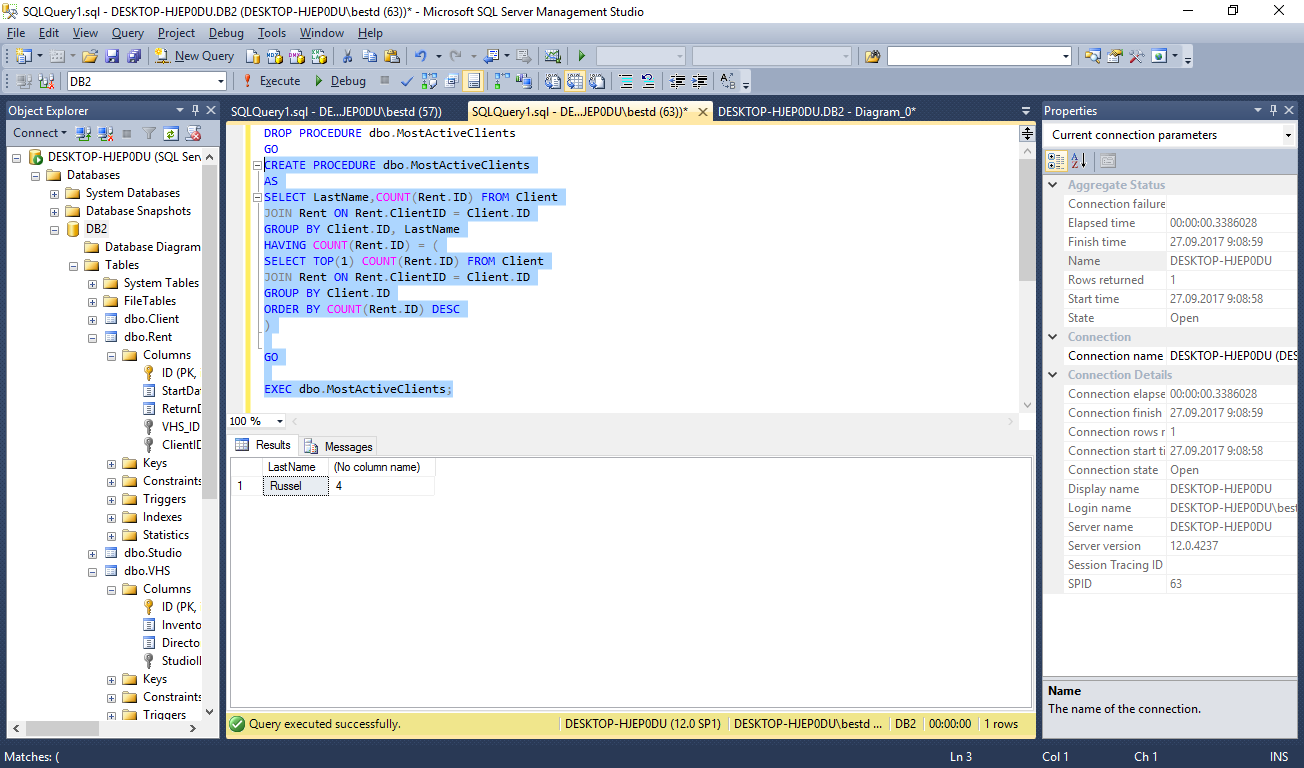
ORDER BY COUNT(Rent.ID) DESC

)

GO

EXEC dbo.MostActiveClients;

*Результат виконання:*



* 1. *Призначення*: Які касети жодного разу не брали у прокат

*Текст запиту*:

CREATE PROCEDURE dbo.GetUnusedVHS

AS

SELECT \* FROM VHS

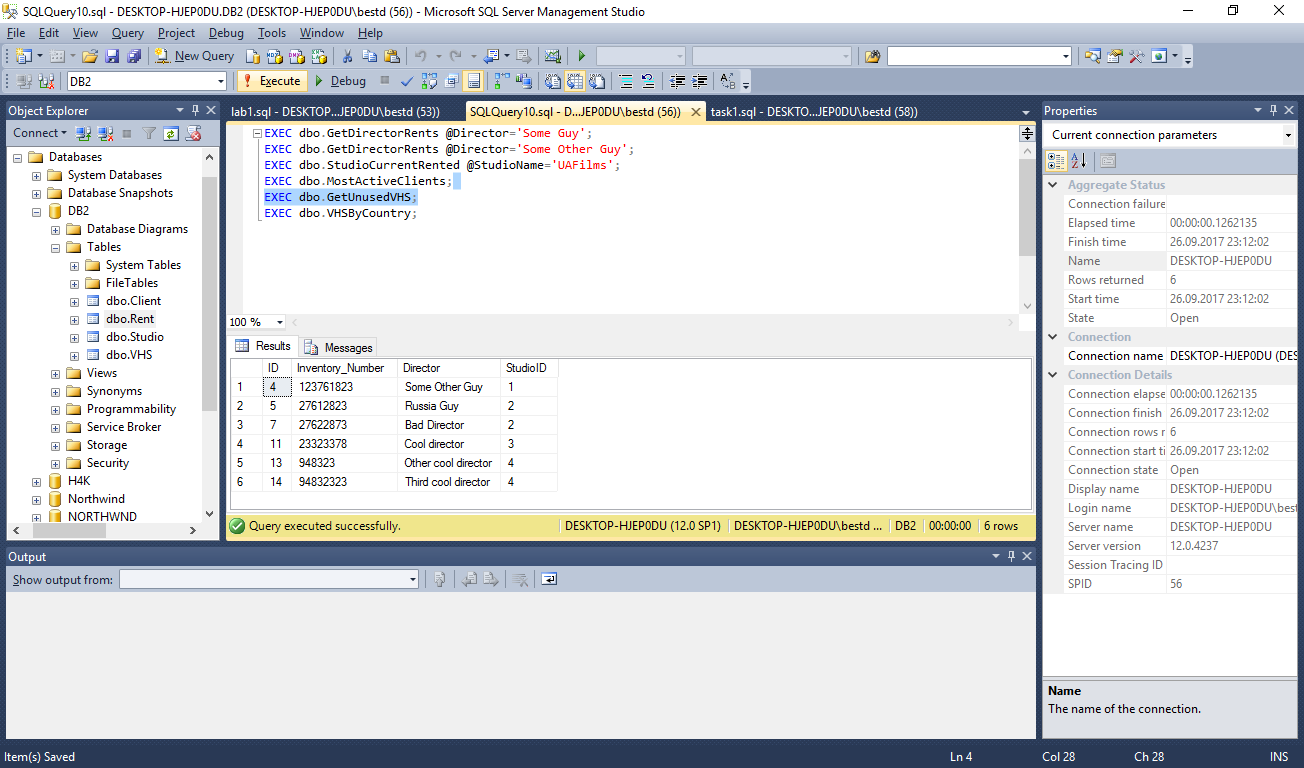
WHERE ID NOT IN

(SELECT VHS\_ID FROM Rent)

GO

EXEC dbo.GetUnusedVHS;

*Результат виконання:*



* 1. *Призначення*: Показати кількість касет по кожній з країн.

*Текст запиту*:

CREATE PROCEDURE dbo.VHSByCountry

AS

SELECT Country, Count(VHS.ID) FROM Studio

LEFT JOIN VHS ON VHS.StudioID = Studio.ID

GROUP BY Country

GO

EXEC dbo.VHSByCountry;

*Результат виконання:*

