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”

по темі

«Аналітичні запити»

Варіант 4

|  |  |  |
| --- | --- | --- |
| Прийняв: |  | Виконав: |
| Cт.вик. Клименко О. М. |  | студент 3-го курсу  гр. ІП-51 ФІОТ  Гладишко  Валентин  Юрійович |

Київ – 2017

**ЗМІСТ:**

[1 Завдання 3](#_Toc495230724)

[2 Виконання Завдання 5](#_Toc495230725)

# Завдання

Завдання 1:

Кожен студент згідно предметної області за варіантом повинен визначити таблицю\_1, що містить оперативні дані, та таблицю\_2, що містить довідник, що використовується в таблиці\_1, скласти наступні види запитів, попередньо описавши словами, що ці запити виконують (запит має бути адекватним, а не формальне застосування наведених службових слів):

1) Використовуючи count() (або будь-яку іншу агрегатну функцію), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

2) Використовуючи rank() або dense\_rank(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

3) Використовуючи sliding window (rows), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

4) Використовуючи sliding window (range) , partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

5) Самостійно розібратися, як застосовується функція lag(). Використовуючи lag(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

6) Самостійно розібратися, як застосовується функція lead(). Використовуючи lead(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.

Завдання 2 (повторення «Нормалізації» - матеріалу попереднього семестру):

Згідно Вашої предметної області, вибрати таблицю:

1) Змінити таблицю так, щоб вона не була у першій нормальній формі, навести цю таблицю у наповненому вигляді. Потім описати, які зміни треба зробити, щоб вона відповідала першій нормальній формі, навести цю таблицю у наповненому вигляді.

2) Описати, які зміни треба зробити, щоб таблиця відповідала другій нормальній формі, навести цю таблицю у наповненому вигляді.

3) Описати, які зміни треба зробити, щоб таблиця відповідала третій нормальній формі, навести цю таблицю у наповненому вигляді.

4) Описати, які зміни треба зробити, щоб таблиця відповідала нормальній формі Бойса-Кода, навести цю таблицю у наповненому вигляді.

**Преметна область**: Клієнти стоматологічної поліклініки

# Виконання Завдання

1. Завдання по написанню запитів
   1. Використовуючи count() (або будь-яку іншу агрегатну функцію), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції:  
      *Призначення*: Вивести по кожному виду лікування його прибуток Прибуток = Кількість клієнтів \* Ціна лікування.

*Текст запиту з використанням аналітичних функцій*:

SELECT DISTINCT [T].[TreatmentTypeID],

SUM([TP].[Price]) OVER (PARTITION BY [T].[TreatmentTypeID]) AS [Profit]

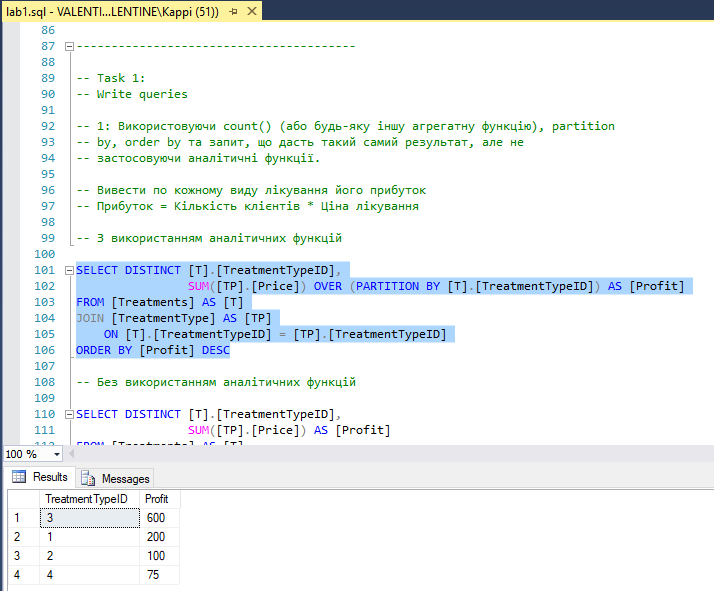
FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

ORDER BY [Profit] DESC

*Результат виконання з використанням аналітичних функцій*:



*Текст запиту без використання аналітичних функцій*:

SELECT DISTINCT [T].[TreatmentTypeID],

SUM([TP].[Price]) AS [Profit]

FROM [Treatments] AS [T]

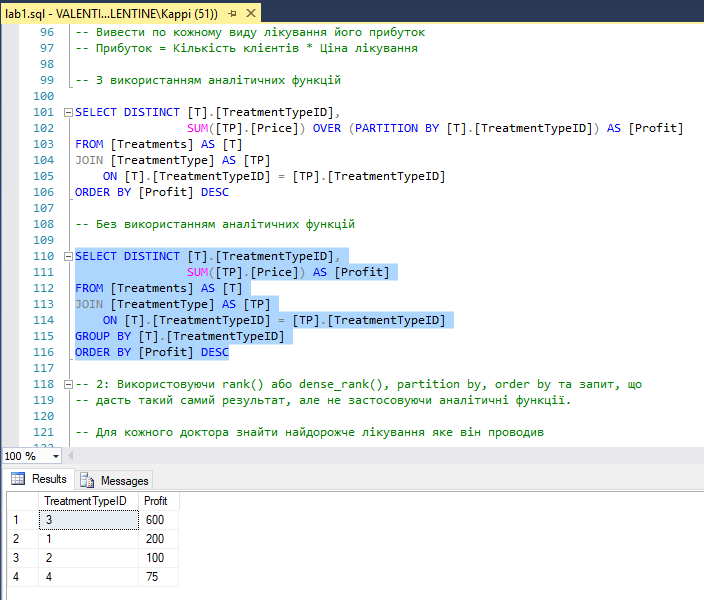
JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

GROUP BY [T].[TreatmentTypeID]

ORDER BY [Profit] DESC

*Результат виконання без використання аналітичних функцій*:



* 1. Використовуючи rank() або dense\_rank(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції:

*Призначення*: Для кожного доктора знайти найдорожче лікування яке він проводив.

*Текст запиту з використанням аналітичних функцій*:

SELECT \*

FROM (SELECT DISTINCT [T].[DoctorID],

[TP].[Price],

RANK() OVER (PARTITION BY [T].[DoctorID]

ORDER BY [TP].[Price] DESC) AS [Max]

FROM [Treatments] AS [T]

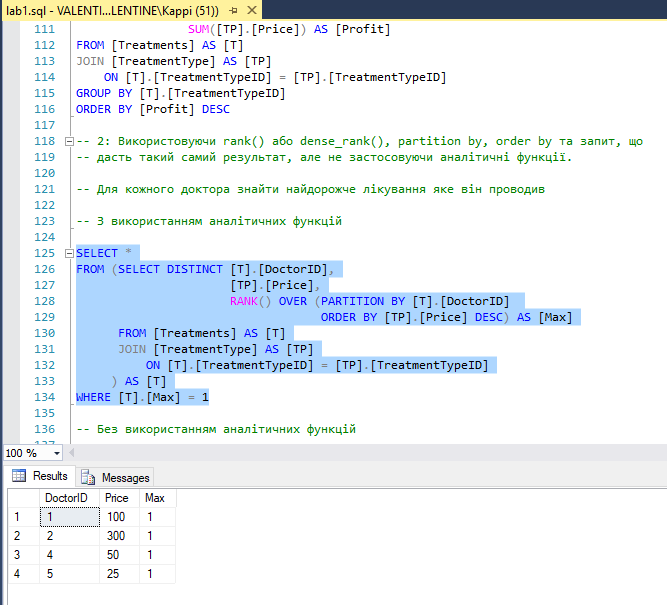
JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

) AS [T]

WHERE [T].[Max] = 1

*Результат виконання з використанням аналітичних функцій*:



*Текст запиту без використання аналітичних функцій*:

SELECT DISTINCT [T].[DoctorID],

MAX([TP].[Price]) AS [Max]

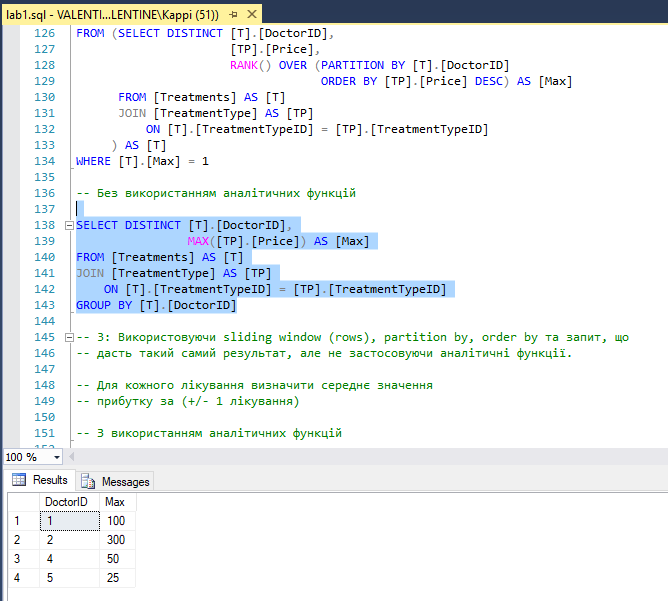
FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

GROUP BY [T].[DoctorID]

*Результат виконання без використання аналітичних функцій*:



* 1. Використовуючи sliding window (rows), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції

*Призначення*: Для кожного лікування визначити середнє значення прибутку за (+/- 1 лікування)

*Текст запиту з використанням аналітичних функцій*:

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price],

AVG([TP].[Price]) OVER(ORDER BY [T].[Date]

ROWS BETWEEN 1 PRECEDING AND 1 FOLLOWING) AS [AveragePrice]

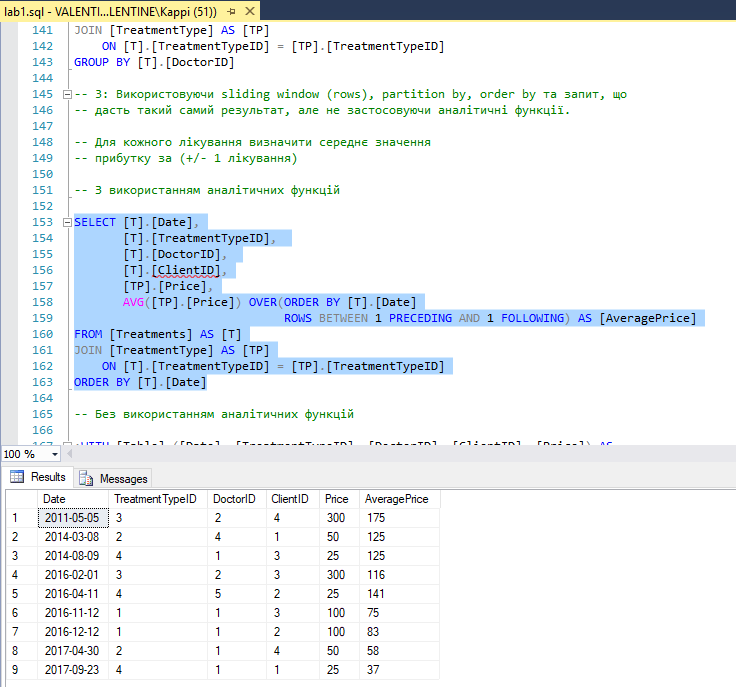
FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

ORDER BY [T].[Date]

*Результат виконання з використанням аналітичних функцій*:



*Текст запиту без використання аналітичних функцій*:

;WITH [Table] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price]

FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

),

[RowsTable] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price], [RowNumber]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

COUNT([T].[DoctorID]) AS [RowNumber]

FROM [Table] AS [T], [Table] AS [T2]

WHERE [T].[Date] >= [T2].[Date]

GROUP BY [T].[Date], [T].[TreatmentTypeID],

[T].[DoctorID], [T].[ClientID], [T].[Price]

)

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

AVG([T2].[Price]) AS [AveragePrice]

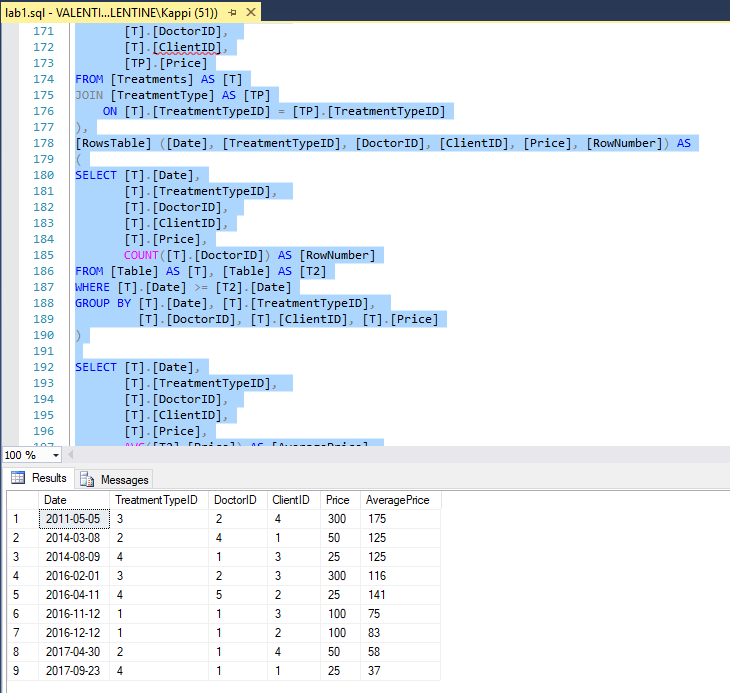
FROM [RowsTable] AS [T], [RowsTable] AS [T2]

WHERE [T2].[RowNumber] BETWEEN ([T].[RowNumber] - 1) AND ([T].[RowNumber] + 1)

GROUP BY [T].[Date], [T].[TreatmentTypeID],

[T].[DoctorID], [T].[ClientID], [T].[Price]

*Результат виконання без використання аналітичних функцій*:



* 1. Використовуючи sliding window (range) , partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції.:

*Призначення*: Для кожного лікування визначити середнє значення прибутку (поточне і всі попередні)

*Текст запиту з використанням аналітичних функцій*:

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price],

AVG([TP].[Price]) OVER(ORDER BY [T].[Date]

RANGE UNBOUNDED PRECEDING) AS [Average Price]

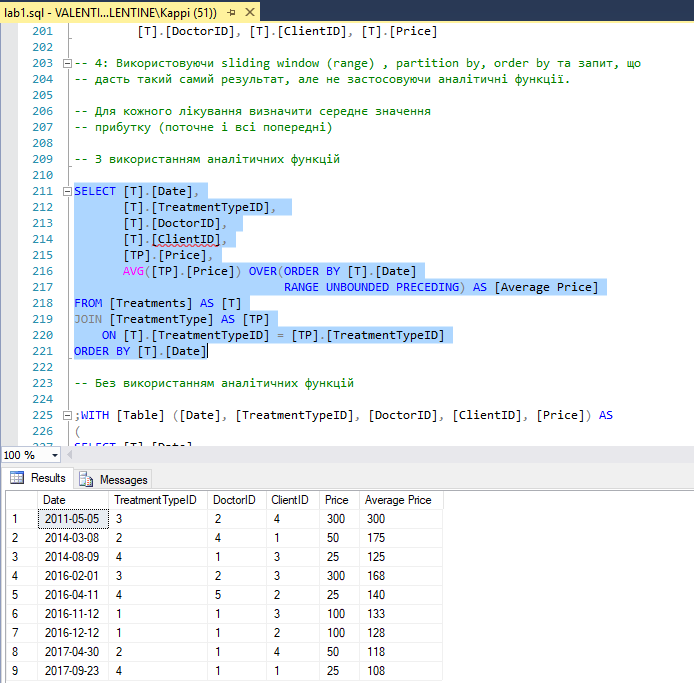
FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

ORDER BY [T].[Date]

*Результат виконання з використанням аналітичних функцій*:



*Текст запиту без використання аналітичних функцій*:

;WITH [Table] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price]

FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

),

[RowsTable] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price], [RowNumber]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

COUNT([T].[DoctorID]) AS [RowNumber]

FROM [Table] AS [T], [Table] AS [T2]

WHERE [T].[Date] >= [T2].[Date]

GROUP BY [T].[Date], [T].[TreatmentTypeID],

[T].[DoctorID], [T].[ClientID], [T].[Price]

)

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

AVG([T2].[Price]) AS [AveragePrice]

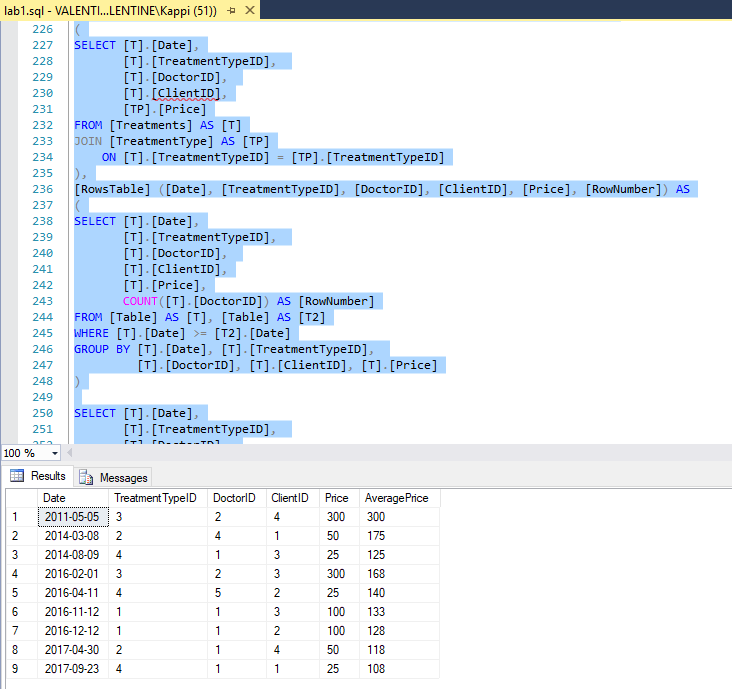
FROM [RowsTable] AS [T], [RowsTable] AS [T2]

WHERE [T2].[RowNumber] <= [T].[RowNumber]

GROUP BY [T].[Date], [T].[TreatmentTypeID],

[T].[DoctorID], [T].[ClientID], [T].[Price]

*Результат виконання без використання аналітичних функцій*:



* 1. Самостійно розібратися, як застосовується функція lag(). Використовуючи lag(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції:

*Призначення*: Для кожного лікування вивести вартість поточного та попереднього лікування

*Текст запиту з використанням аналітичних функцій*:

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price],

LAG([TP].[Price], 1 ,[TP].[Price]) OVER(ORDER BY [T].[Date]) AS [PreviousPrice]

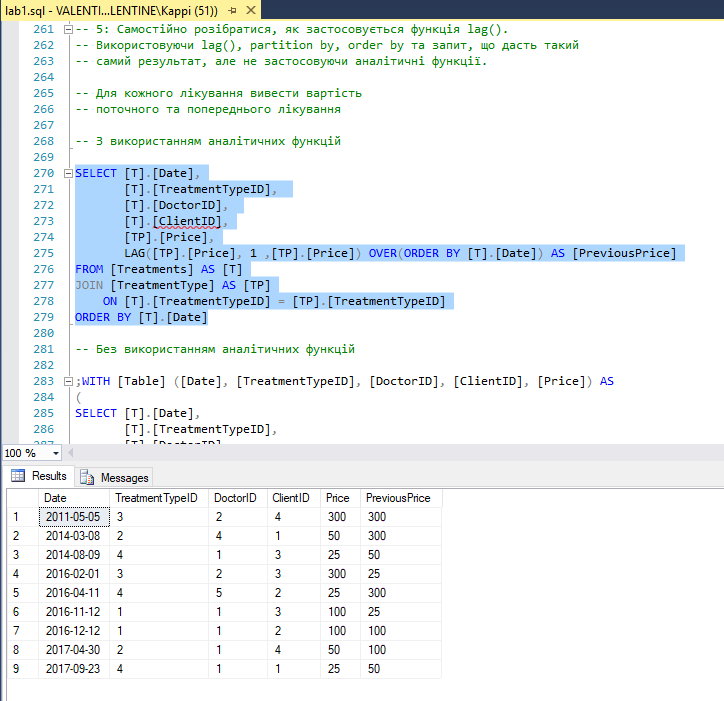
FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

ORDER BY [T].[Date]

*Результат виконання з використанням аналітичних функцій*:



*Текст запиту без використання аналітичних функцій*:

;WITH [Table] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price]

FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

),

[RowsTable] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price], [RowNumber]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

COUNT([T].[DoctorID]) AS [RowNumber]

FROM [Table] AS [T], [Table] AS [T2]

WHERE [T].[Date] >= [T2].[Date]

GROUP BY [T].[Date], [T].[TreatmentTypeID],

[T].[DoctorID], [T].[ClientID], [T].[Price]

)

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

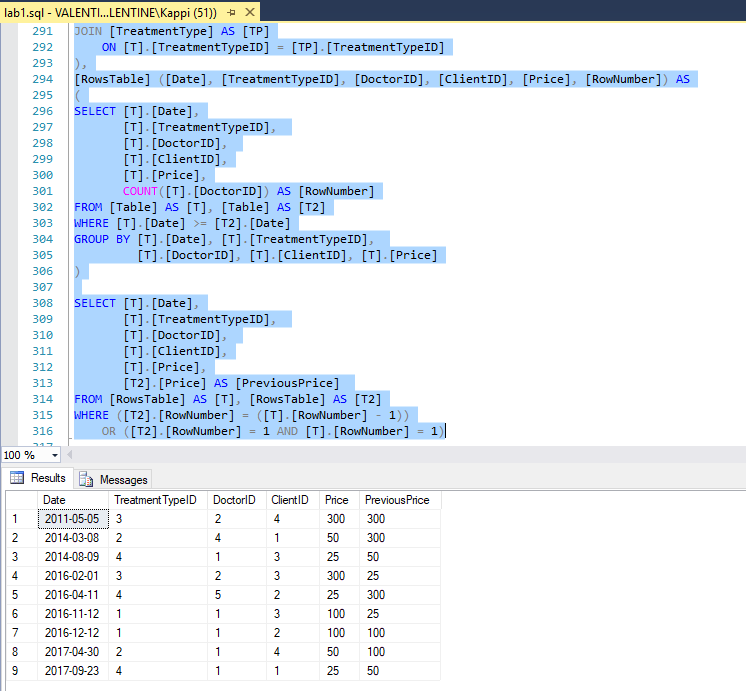
[T2].[Price] AS [PreviousPrice]

FROM [RowsTable] AS [T], [RowsTable] AS [T2]

WHERE ([T2].[RowNumber] = ([T].[RowNumber] - 1))

OR ([T2].[RowNumber] = 1 AND [T].[RowNumber] = 1)

*Результат виконання без використання аналітичних функцій*:



* 1. Самостійно розібратися, як застосовується функція lead(). Використовуючи lead(), partition by, order by та запит, що дасть такий самий результат, але не застосовуючи аналітичні функції:

*Призначення*: Для кожного лікування вивести вартість поточного та наступного лікування

*Текст запиту з використанням аналітичних функцій*:

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price],

LEAD([TP].[Price], 1 ,[TP].[Price]) OVER(ORDER BY [T].[Date]) AS [NextPrice]

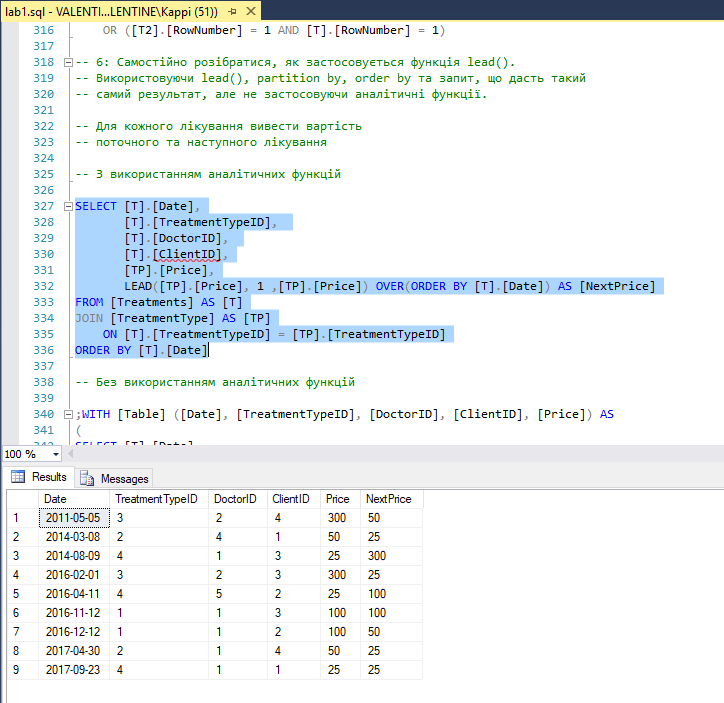
FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

ORDER BY [T].[Date]

*Результат виконання з використання аналітичних функцій*:



*Текст запиту без використанням аналітичних функцій*:

;WITH [Table] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[TP].[Price]

FROM [Treatments] AS [T]

JOIN [TreatmentType] AS [TP]

ON [T].[TreatmentTypeID] = [TP].[TreatmentTypeID]

),

[RowsTable] ([Date], [TreatmentTypeID], [DoctorID], [ClientID], [Price], [RowNumber]) AS

(

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

COUNT([T].[DoctorID]) AS [RowNumber]

FROM [Table] AS [T], [Table] AS [T2]

WHERE [T].[Date] >= [T2].[Date]

GROUP BY [T].[Date], [T].[TreatmentTypeID],

[T].[DoctorID], [T].[ClientID], [T].[Price]

),

[Rows] ([RowNumber]) AS

(

SELECT COUNT([T].[DoctorID]) AS [RowNumber]

FROM [Table] AS [T], [Table] AS [T2]

WHERE [T].[Date] >= [T2].[Date]

GROUP BY [T].[Date], [T].[TreatmentTypeID],

[T].[DoctorID], [T].[ClientID], [T].[Price]

)

SELECT [T].[Date],

[T].[TreatmentTypeID],

[T].[DoctorID],

[T].[ClientID],

[T].[Price],

[T2].[Price] AS [NextPrice]

FROM [RowsTable] AS [T], [RowsTable] AS [T2]

WHERE ([T2].[RowNumber] = ([T].[RowNumber] + 1))

OR ([T2].[RowNumber] =

(SELECT TOP(1) \* FROM [Rows]

ORDER BY [RowNumber] DESC)

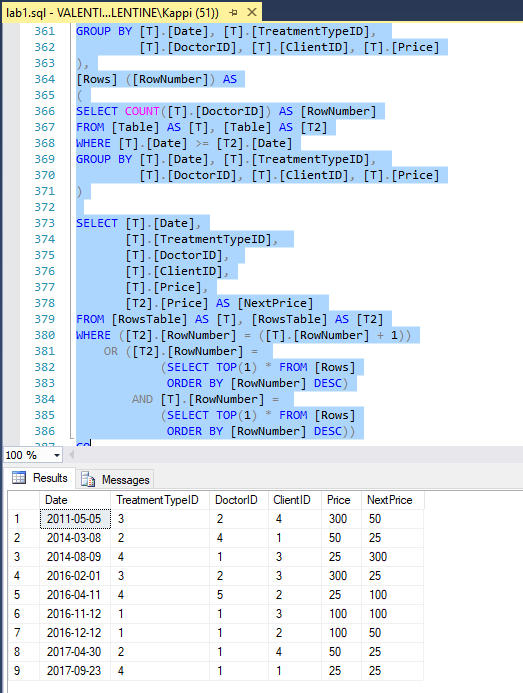
AND [T].[RowNumber] =

(SELECT TOP(1) \* FROM [Rows]

ORDER BY [RowNumber] DESC))

GO

*Результат виконання без використання аналітичних функцій*:



1. Завдання по нормалізації БД:
   1. Змінити таблицю так, щоб вона не була у першій нормальній формі, навести цю таблицю у наповненому вигляді. Потім описати, які зміни треба зробити, щоб вона відповідала першій нормальній формі, навести цю таблицю у наповненому вигляді.

*Текст запиту*:

DROP TABLE [Treatments]

GO

CREATE TABLE [Treatments] (

[Date] DATE NOT NULL,

[TreatmentType] NVARCHAR(255) NOT NULL,

[Price] INT NOT NULL,

[DoctorLastName] NVARCHAR(255) NOT NULL,

[DoctorFirstName] NVARCHAR(255) NOT NULL,

[DoctorWork] DATE NOT NULL,

[ClientName] NVARCHAR(255) NOT NULL)

GO

INSERT INTO [Treatments] ([Date], [TreatmentType], [Price], [DoctorLastName], [DoctorFirstName], [DoctorWork], [ClientName])

VALUES ('2017-09-23', 'TreatmentType4', 50, 'Peter', 'Poroshenko', '2016-09-12', 'Valentine Gladyshko, Svetlana Reutskaya'),

('2016-12-12', 'TreatmentType1', 100, 'Peter', 'Poroshenko', '2016-09-12', 'Illya Volkov'),

('2014-08-09', 'TreatmentType4', 50, 'Peter', 'Poroshenko', '2016-09-12', 'Illya Volkov'),

('2017-04-30', 'TreatmentType2', 400, 'Victor', 'Frankenstein', '2012-01-14', 'Alexander Zarichkovyi, Svetlana Reutskaya'),

('2016-04-11', 'TreatmentType3', 250, 'Charles', 'Darwin', '2016-02-12', 'Valentine Gladyshko')

GO

SELECT \* FROM [Treatments]

GO

-- 1 NF

-- Добавити поле для ID не записувати кілька значень в одну колонку

DROP TABLE [Treatments]

GO

CREATE TABLE [Treatments] (

[TreatmentID] INT IDENTITY(1,1) NOT NULL,

[Date] DATE NOT NULL,

[TreatmentType] NVARCHAR(255) NOT NULL,

[Price] INT NOT NULL,

[DoctorLastName] NVARCHAR(255) NOT NULL,

[DoctorFirstName] NVARCHAR(255) NOT NULL,

[DoctorWork] DATE NOT NULL,

[ClientName] NVARCHAR(255) NOT NULL,

CONSTRAINT PK\_TreatmentID PRIMARY KEY CLUSTERED (TreatmentID))

GO

INSERT INTO [Treatments] ([Date], [TreatmentType], [Price], [DoctorLastName], [DoctorFirstName], [DoctorWork], [ClientName])

VALUES ('2017-09-23', 'TreatmentType4', 50, 'Peter', 'Poroshenko', '2016-09-12', 'Valentine Gladyshko'),

('2017-09-23', 'TreatmentType4', 50, 'Peter', 'Poroshenko', '2016-09-12', 'Svetlana Reutskaya'),

('2016-12-12', 'TreatmentType1', 100, 'Peter', 'Poroshenko', '2016-09-12', 'Illya Volkov'),

('2014-08-09', 'TreatmentType4', 50, 'Peter', 'Poroshenko', '2016-09-12', 'Illya Volkov'),

('2017-04-30', 'TreatmentType2', 400, 'Victor', 'Frankenstein', '2012-01-14', 'Alexander Zarichkovyi'),

('2017-04-30', 'TreatmentType2', 400, 'Victor', 'Frankenstein', '2012-01-14', 'Svetlana Reutskaya'),

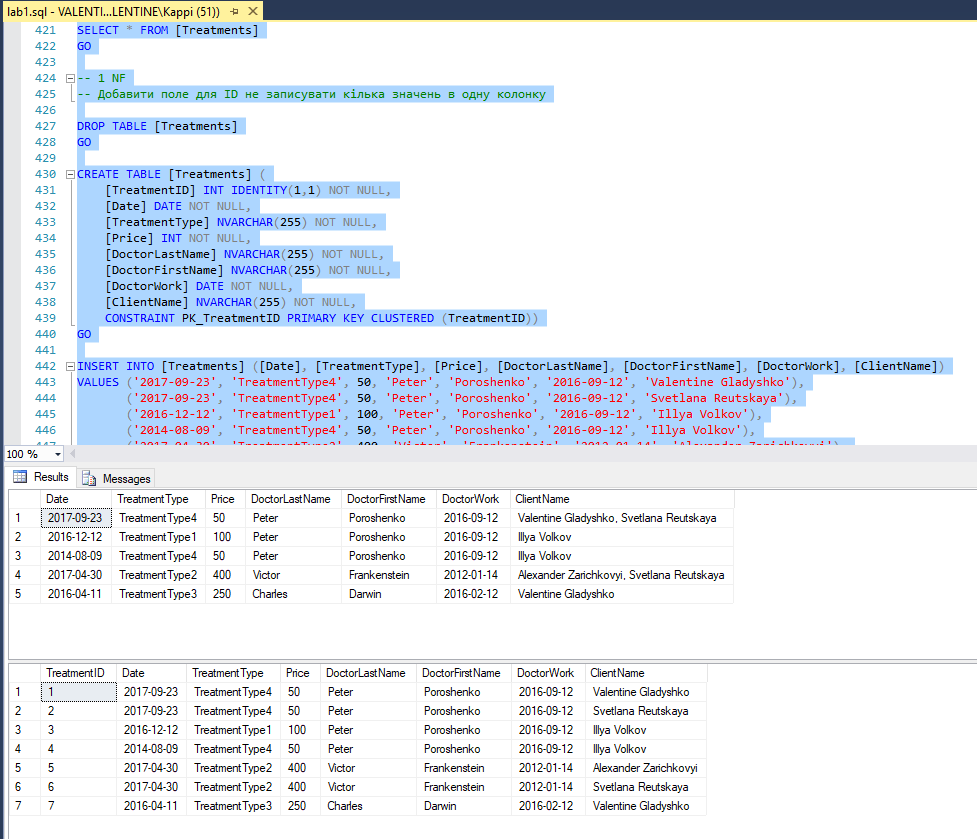
('2016-04-11', 'TreatmentType3', 250, 'Charles', 'Darwin', '2016-02-12', 'Valentine Gladyshko')

GO

SELECT \* FROM [Treatments]

GO

*Результат виконання:*



* 1. Описати, які зміни треба зробити, щоб таблиця відповідала другій нормальній формі, навести цю таблицю у наповненому вигляді*.*

*Текст запиту*:

-- 2 NF

-- [Price] винести в окрему таблицю [TreatmentTypes]

DROP TABLE [Treatments]

GO

CREATE TABLE [TreatmentTypes] (

[TreatmentType] NVARCHAR(255) NOT NULL,

[Price] INT NOT NULL,

CONSTRAINT PK\_TreatmentType1 PRIMARY KEY CLUSTERED (TreatmentType))

GO

CREATE TABLE [Treatments] (

[TreatmentID] INT IDENTITY(1,1) NOT NULL,

[Date] DATE NOT NULL,

[TreatmentType] NVARCHAR(255) NOT NULL FOREIGN KEY REFERENCES [TreatmentTypes](TreatmentType),

[DoctorLastName] NVARCHAR(255) NOT NULL,

[DoctorFirstName] NVARCHAR(255) NOT NULL,

[DoctorWork] DATE NOT NULL,

[ClientName] NVARCHAR(255) NOT NULL,

CONSTRAINT PK\_TreatmentID PRIMARY KEY CLUSTERED (TreatmentID))

GO

INSERT INTO [TreatmentTypes] ([TreatmentType], [Price])

VALUES ('TreatmentType1', 100),

('TreatmentType2', 400),

('TreatmentType3', 250),

('TreatmentType4', 50)

GO

INSERT INTO [Treatments] ([Date], [TreatmentType], [DoctorLastName], [DoctorFirstName], [DoctorWork], [ClientName])

VALUES ('2017-09-23', 'TreatmentType4', 'Peter', 'Poroshenko', '2016-09-12', 'Valentine Gladyshko'),

('2017-09-23', 'TreatmentType4', 'Peter', 'Poroshenko', '2016-09-12', 'Svetlana Reutskaya'),

('2016-12-12', 'TreatmentType1', 'Peter', 'Poroshenko', '2016-09-12', 'Illya Volkov'),

('2014-08-09', 'TreatmentType4', 'Peter', 'Poroshenko', '2016-09-12', 'Illya Volkov'),

('2017-04-30', 'TreatmentType2', 'Victor', 'Frankenstein', '2012-01-14', 'Alexander Zarichkovyi'),

('2017-04-30', 'TreatmentType2', 'Victor', 'Frankenstein', '2012-01-14', 'Svetlana Reutskaya'),

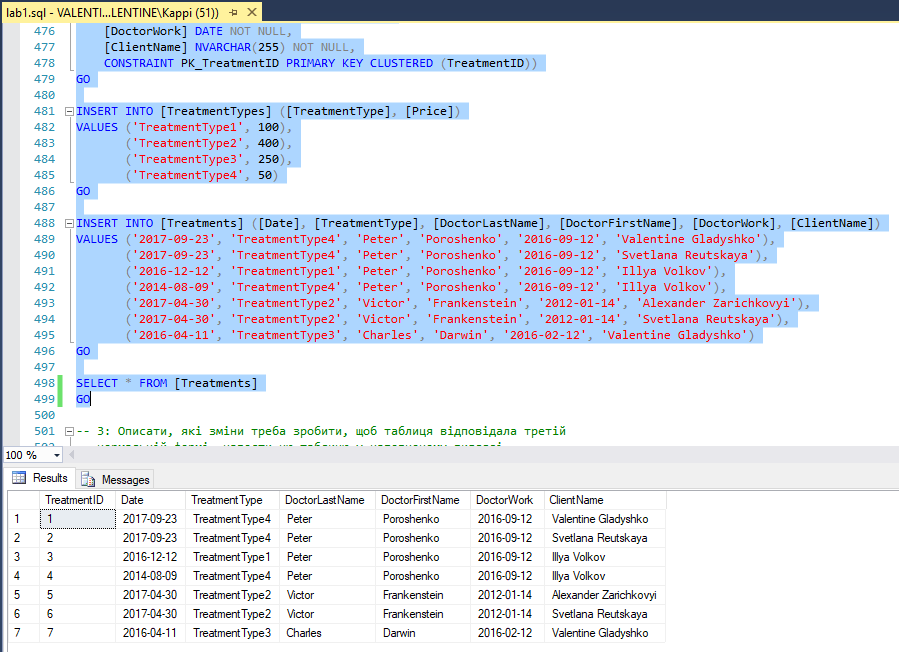
('2016-04-11', 'TreatmentType3', 'Charles', 'Darwin', '2016-02-12', 'Valentine Gladyshko')

GO

SELECT \* FROM [Treatments]

GO

*Результат виконання:*



* 1. Описати, які зміни треба зробити, щоб таблиця відповідала третій нормальній формі, навести цю таблицю у наповненому вигляді.

*Текст запиту*:

-- 3 NF

-- [DoctorLastName], [DoctorFirstName], [DoctorWork] винести в окрему таблицю [Doctor]

DROP TABLE [Treatments]

GO

CREATE TABLE [Doctor] (

[DoctorID] INT IDENTITY(1,1) NOT NULL,

[FirstName] NVARCHAR(255) NOT NULL,

[LastName] NVARCHAR(255) NOT NULL,

[Work] DATE,

CONSTRAINT PK\_Doctor1ID PRIMARY KEY CLUSTERED (DoctorID))

GO

CREATE TABLE [Treatments] (

[TreatmentID] INT IDENTITY(1,1) NOT NULL,

[Date] DATE NOT NULL,

[TreatmentType] NVARCHAR(255) NOT NULL FOREIGN KEY REFERENCES [TreatmentTypes](TreatmentType),

[DoctorID] INT NOT NULL FOREIGN KEY REFERENCES [Doctor]([DoctorID]),

[ClientName] NVARCHAR(255) NOT NULL,

CONSTRAINT PK\_TreatmentID PRIMARY KEY CLUSTERED (TreatmentID))

GO

INSERT INTO [Doctor] ([FirstName], [LastName], [Work])

VALUES ('Peter', 'Poroshenko', '2016-09-12'),

('Victor', 'Frankenstein', '2012-01-14'),

('Charles', 'Darwin', '2016-02-12')

GO

INSERT INTO [Treatments] ([Date], [TreatmentType], [DoctorID], [ClientName])

VALUES ('2017-09-23', 'TreatmentType4', 1, 'Valentine Gladyshko'),

('2017-09-23', 'TreatmentType4', 1, 'Svetlana Reutskaya'),

('2016-12-12', 'TreatmentType1', 1, 'Illya Volkov'),

('2014-08-09', 'TreatmentType4', 1, 'Illya Volkov'),

('2017-04-30', 'TreatmentType2', 2, 'Alexander Zarichkovyi'),

('2017-04-30', 'TreatmentType2', 2, 'Svetlana Reutskaya'),

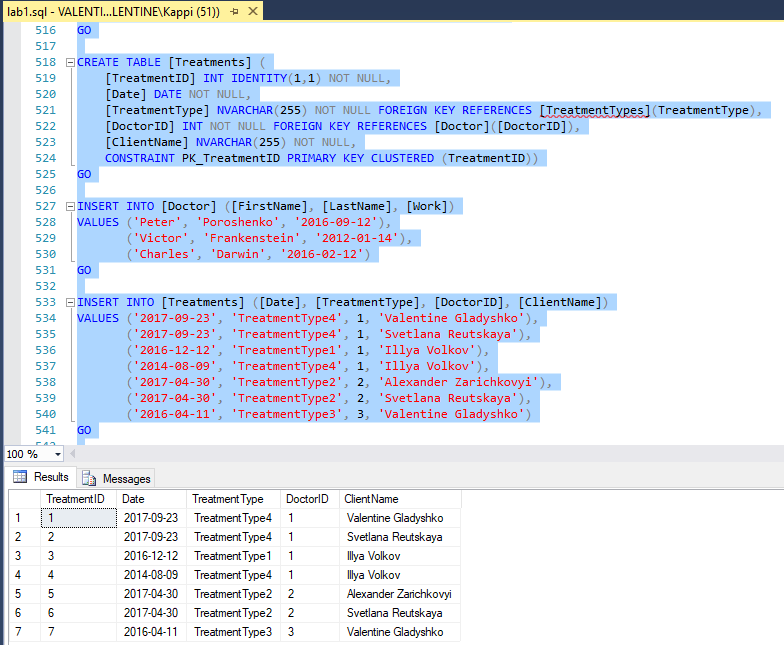
('2016-04-11', 'TreatmentType3', 3, 'Valentine Gladyshko')

GO

SELECT \* FROM [Treatments]

GO

*Результат виконання:*



* 1. Описати, які зміни треба зробити, щоб таблиця відповідала нормальній формі Бойса-Кода, навести цю таблицю у наповненому вигляді*.*

*Текст запиту*:

-- 3.5 NF

-- Винести [TreatmentType] в таблицю [TreatmentTypes]

DROP TABLE [Treatments]

GO

ALTER TABLE [dbo].[TreatmentTypes] DROP CONSTRAINT [PK\_TreatmentType1]

GO

DROP TABLE [dbo].[TreatmentTypes]

GO

CREATE TABLE [TreatmentTypes] (

[TreatmentTypeID] INT IDENTITY(1,1) NOT NULL,

[TreatmentType] NVARCHAR(255) NOT NULL,

[Price] INT NOT NULL,

CONSTRAINT PK\_TreatmentType1 PRIMARY KEY CLUSTERED (TreatmentTypeID))

GO

CREATE TABLE [Treatments] (

[TreatmentID] INT IDENTITY(1,1) NOT NULL,

[Date] DATE NOT NULL,

[TreatmentTypeID] INT NOT NULL FOREIGN KEY REFERENCES [TreatmentTypes](TreatmentTypeID),

[DoctorID] INT NOT NULL FOREIGN KEY REFERENCES [Doctor]([DoctorID]),

[ClientName] NVARCHAR(255) NOT NULL,

CONSTRAINT PK\_TreatmentID PRIMARY KEY CLUSTERED (TreatmentID))

GO

INSERT INTO [TreatmentTypes] ([TreatmentType], [Price])

VALUES ('TreatmentType1', 100),

('TreatmentType2', 400),

('TreatmentType3', 250),

('TreatmentType4', 50)

GO

INSERT INTO [Treatments] ([Date], [TreatmentTypeID], [DoctorID], [ClientName])

VALUES ('2017-09-23', 4, 1, 'Valentine Gladyshko'),

('2017-09-23', 4, 1, 'Svetlana Reutskaya'),

('2016-12-12', 1, 1, 'Illya Volkov'),

('2014-08-09', 4, 1, 'Illya Volkov'),

('2017-04-30', 2, 2, 'Alexander Zarichkovyi'),

('2017-04-30', 2, 2, 'Svetlana Reutskaya'),

('2016-04-11', 3, 3, 'Valentine Gladyshko')

GO

SELECT \* FROM [Treatments]

GO

*Результат виконання:*

