**Національний університет біоресурсів і природокористування України**

**Факультет інформаційних технологій**

**Кафедра комп’ютерних наук**

Організація баз даних

**Тема:**  Розробка збережених процедур і тригерів у середовищі [**СУБД**](https://elearn.nubip.edu.ua/mod/glossary/showentry.php?eid=325240&displayformat=dictionary) на прикладі MS SQL Server

**Лабораторна робота №8**:

**Студент**: Возний Богдан Олександрович

**Дата виконання**: 16.03.2025

Київ 2025

**Файл 1 (Процедури)**

**Код процедур**

**короткий опис процедури**

### AddNewMedication

**Призначення:**  
Додає новий медикамент до таблиці Medication.

**Що робить:**

* Перевіряє, чи вже існує медикамент з таким іменем та виробником.
* Якщо такий є — виводить помилку "Цей медикамент уже існує."
* Інакше — додає новий запис із автоматично згенерованим id\_medication (через NEWID()).

**Код процедути**

-- ================================================

-- Template generated from Template Explorer using:

-- Create Procedure (New Menu).SQL

--

-- Use the Specify Values for Template Parameters

-- command (Ctrl-Shift-M) to fill in the parameter

-- values below.

--

-- This block of comments will not be included in

-- the definition of the procedure.

-- ================================================

use Helsi

go

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

-- =============================================

-- Author: <Author,,Name>

-- Create date: <Create Date,,>

-- Description: <Description,,>

-- =============================================

if object\_id('AddNewMedication','p') is not null

drop procedure AddNewMedication

go

CREATE PROCEDURE AddNewMedication

@name\_medication nVARCHAR(max),

@manufacturer nVARCHAR(max),

@expiration\_date DATE,

@description\_medication nVARCHAR(max)

AS

BEGIN

IF EXISTS (

SELECT 1 FROM Medication

WHERE name\_medication = @name\_medication AND manufacturer = @manufacturer

)

BEGIN

RAISERROR('Цей медикамент уже існує.', 16, 1)

RETURN

END

INSERT INTO Medication (id\_medication ,name\_medication, manufacturer, expiration\_date, description\_medication)

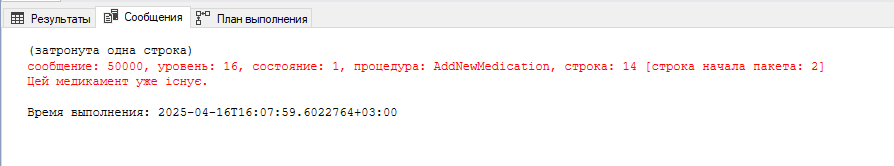
VALUES (CAST(NEWID() as char(10)), @name\_medication, @manufacturer, @expiration\_date, @description\_medication)

END

Go

**Скріншоти роботи процедури**







Exec AddNewMedication @name\_medication ='Парацетамол 1',@manufacturer = 'дарниця 1', @expiration\_date ='2029-03-19', @description\_medication = 'Від темпаратури'



**короткий опис процедури**

### proc\_UpdateInfoDepartment

**Призначення:**  
Оновлює дані про відділення (Department).

**Що робить:**

* За заданим @ID оновлює назву (name\_department) і опис (description\_department) відділення.

**Код процедути**

if object\_id('proc\_UpdateInfoDepartment','p') is not null

drop procedure proc\_UpdateInfoDepartment

go

create procedure proc\_UpdateInfoDepartment

@ID int = 0,

@valueName nvarchar(max) = '',

@valueDescription nvarchar(max) = ''

as

begin

set nocount on;

update Department

set name\_department = @valueName,

description\_department = @valueDescription

Where id\_department = @ID

end

go

**Скріншоти роботи процедури**





**короткий опис процедури**

### selectData\_Medication

**Призначення:**  
Вибірка медикаментів за виробником.

**Що робить:**

* Виводить усі записи з таблиці Medication, де значення поля manufacturer відповідає заданому параметру @value\_manufacturer.

**Код процедути**

if object\_id('selectData\_Medication','p') is not null

drop procedure selectData\_Medication

go

create procedure selectData\_Medication

@value\_manufacturer nvarchar(max) = ''

as

begin

set nocount on;

select \*

from Medication

where manufacturer = @value\_manufacturer

end

go

**Файл 2 (Тригери)**

**Код тригерів**

**короткий опис тригера**

**trg\_CheckMedicationExpiration**

**Призначення:**  
Перевіряє, чи не додається медикамент із вже протермінованою датою.

**Що робить:**  
Якщо в новому записі в таблиці Medication значення expiration\_date менше поточної дати (GETDATE()), то генерує помилку та скасовує транзакцію.

**Код Тригера**

if object\_id ('trg\_CheckMedicationExpiration','t' ) is not null

drop trigger trg\_CheckMedicationExpiration

go

create trigger trg\_CheckMedicationExpiration

on Medication

after insert

as

begin

if exists ( select 1 from inserted where expiration\_date < GETDATE() )

begin

RAISERROR('Неможливо додати медикамент з протермінованим терміном дії.', 16, 1)

rollback TRANSACTION

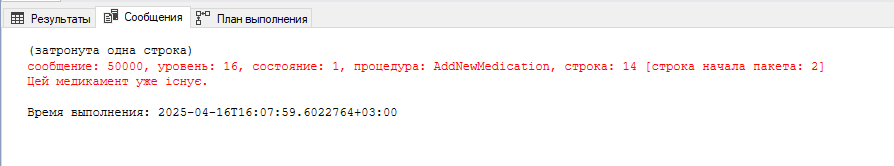
end

end

go

**Скріншоти роботи тригера**







Exec AddNewMedication @name\_medication ='Парацетамол 1',@manufacturer = 'дарниця 1', @expiration\_date ='2029-03-19', @description\_medication = 'Від темпаратури'



**короткий опис тригера**

**trg\_UpdateInfoDepartment**

**Призначення:**  
Перевіряє унікальність назв відділень у таблиці Department.

**Що робить:**  
При вставці або оновленні запису перевіряє, чи не існує вже відділення з такою ж назвою, але з іншим id\_department. Якщо таке є — викликає помилку та скасовує транзакцію.

**м**

if object\_id ('trg\_UpdateInfoDepartment','tr' ) is not null

drop trigger trg\_UpdateInfoDepartment

go

CREATE TRIGGER trg\_UpdateInfoDepartment

ON Department

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (

SELECT 1

FROM inserted i

JOIN Department d

ON i.name\_department = d.name\_department

WHERE d.id\_department != i.id\_department -- не співпадає ID

)

BEGIN

RAISERROR('Неможливо вставити або оновити запис: відділення з такою назвою вже існує.', 16, 1);

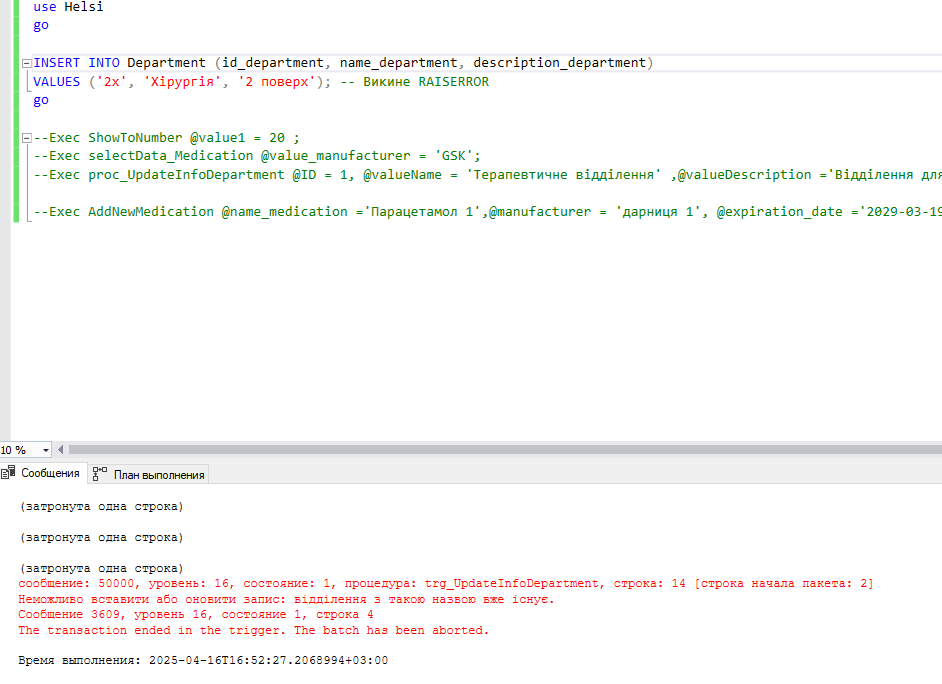
ROLLBACK TRANSACTION;

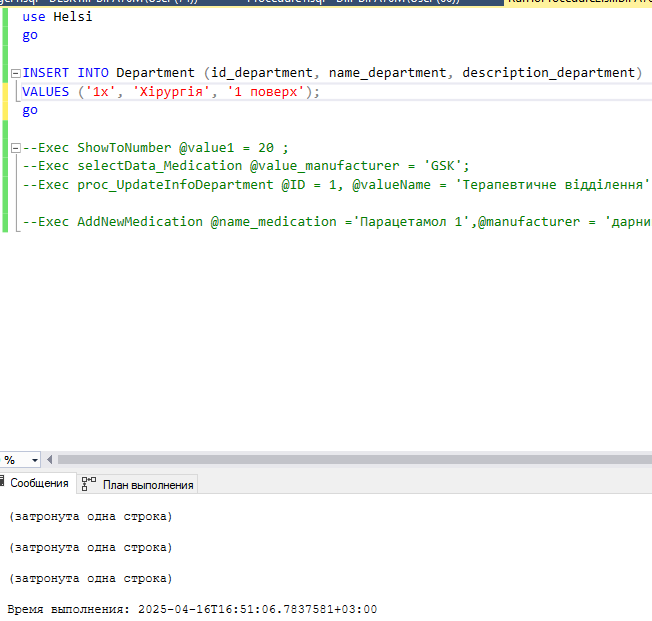
END

END

GO

**Скріншоти роботи тригера**





**короткий опис тригера**

**🔹 trg\_UpdateMedicalCardStatus**

**Призначення:**  
Автоматично оновлює статус медичної картки.

**Що робить:**  
Після вставки нового Episode змінює status\_card відповідної медичної картки (Medical\_card) на 'Активна'.

**Код Тригера**

if object\_id ('trg\_UpdateMedicalCardStatus','t' ) is not null

drop trigger trg\_UpdateMedicalCardStatus

go

create Trigger trg\_UpdateMedicalCardStatus

on Episode

after insert

as

begin

Update Medical\_card

set status\_card = 'Активна'

where id\_medical\_card in (select id\_medical\_card from inserted)

end

go