Міністерство освіти і науки України

Національний університет “Львівська політехніка”

Інститут прикладної математики та фундаментальних наук   
Кафедра прикладної математики

Звіт

про виконання лабораторної роботи №4

“Підпрограми СУБД Oracle”

з курсу “Організація баз даних і знань”

Виконала:

Студентка групи ПМ-22

Карпин О.І.

Перевірив

Пабирівський В.В.

Львів-2024р.

**Завдання:**

1.У всіх таблицях створити поля UCR, DCR, ULC, DLC. Написати тригери які будуть заповнювати дані поля наступним чином: UCR – ім’я користувача, що створив даний запис; DCR – дата та час створення даного запису; ULC – ім’я користувача, що останнім змінив даний запис; DLC – дата та час останньої модифікації даного запису.

2.Створити сурогатний ключ для деякої таблиці, та написати тригер для обов’язкового заповнення цього поля послідовними значеннями.

3.Написати тригери для перевірки наступних обмежень цілісності: кожен виступаючий може брати участь в кількох секціях, але за один день він може виступати тільки в одній секції. В одному приміщенні не можуть проводитися одночасно засідання двох секцій.

Створила поля UCR, DCR, ULC, DLC та створила відповідні тригери для заповнення цих полів:

ALTER TABLE Conferences

ADD UCR NVARCHAR(50),

DCR DATETIME,

ULC NVARCHAR(50),

DLC DATETIME;

ALTER TABLE Equipment

ADD UCR NVARCHAR(50),

DCR DATETIME,

ULC NVARCHAR(50),

DLC DATETIME;

ALTER TABLE Perfomances

ADD UCR NVARCHAR(50),

DCR DATETIME,

ULC NVARCHAR(50),

DLC DATETIME;

ALTER TABLE Performers

ADD UCR NVARCHAR(50),

DCR DATETIME,

ULC NVARCHAR(50),

DLC DATETIME;

ALTER TABLE Section

ADD UCR NVARCHAR(50),

DCR DATETIME,

ULC NVARCHAR(50),

DLC DATETIME;

ALTER TABLE PerfomancesInSections

ADD UCR NVARCHAR(50),

DCR DATETIME,

ULC NVARCHAR(50),

DLC DATETIME;

AFTER INSERT, UPDATE

AS

BEGIN

IF NOT UPDATE(UCR) AND NOT UPDATE(DCR)

BEGIN

DECLARE @ID INT;

SELECT @ID = ID FROM INSERTED;

UPDATE Conferences

SET ULC = USER\_NAME(),

DLC = GETDATE()

WHERE ID = @ID;

END

SELECT @ID = ID FROM INSERTED;

UPDATE Conferences

SET UCR = USER\_NAME(),

DCR = GETDATE()

WHERE ID = @ID;

END;

AFTER INSERT, UPDATE

AS

BEGIN

IF NOT UPDATE(UCR) AND NOT UPDATE(DCR)

BEGIN

DECLARE @ID INT;

SELECT @ID = ID FROM INSERTED;

UPDATE Equipment

SET ULC = USER\_NAME(),

DLC = GETDATE()

WHERE ID = @ID;

END

SELECT @ID = ID FROM INSERTED;

UPDATE Equipment

SET UCR = USER\_NAME(),

DCR = GETDATE()

WHERE ID = @ID;

END;

AFTER INSERT, UPDATE

AS

BEGIN

IF NOT UPDATE(UCR) AND NOT UPDATE(DCR)

BEGIN

DECLARE @Topic NVARCHAR(50);

SELECT @Topic = Topic FROM INSERTED;

UPDATE Perfomances

SET ULC = USER\_NAME(),

DLC = GETDATE()

WHERE Topic = @Topic;

END

SELECT @Topic = Topic FROM INSERTED;

UPDATE Perfomances

SET UCR = USER\_NAME(),

DCR = GETDATE()

WHERE Topic = @Topic;

END;

AFTER INSERT, UPDATE

AS

BEGIN

IF NOT UPDATE(UCR) AND NOT UPDATE(DCR)

BEGIN

DECLARE @SectionName NVARCHAR(50);

SELECT @SectionName = SectionName FROM INSERTED;

UPDATE PerfomancesInSections

SET ULC = USER\_NAME(),

DLC = GETDATE()

WHERE SectionName = @SectionName;

END

AFTER INSERT, UPDATE

AS

BEGIN

IF NOT UPDATE(UCR) AND NOT UPDATE(DCR)

BEGIN

DECLARE @ID INT;

SELECT @ID = ID FROM INSERTED;

UPDATE Performers

SET ULC = USER\_NAME(),

DLC = GETDATE()

WHERE ID = @ID;

END

SELECT @ID = ID FROM INSERTED;

UPDATE Performers

SET UCR = USER\_NAME(),

DCR = GETDATE()

WHERE ID = @ID;

END;

SELECT @SectionName = SectionName FROM INSERTED;

UPDATE PerfomancesInSections

SET UCR = USER\_NAME(),

DCR = GETDATE()

WHERE SectionName = @SectionName;

END;

AFTER INSERT, UPDATE

AS

BEGIN

IF NOT UPDATE(UCR) AND NOT UPDATE(DCR)

BEGIN

DECLARE @Name NVARCHAR(50);

SELECT @Name = Name FROM INSERTED;

UPDATE Section

SET ULC = USER\_NAME(),

DLC = GETDATE()

WHERE Name = @Name;

END

SELECT @Name = Name FROM INSERTED;

UPDATE Section

SET UCR = USER\_NAME(),

DCR = GETDATE()

WHERE Name = @Name;

END;

INSTEAD OF INSERT

AS

BEGIN

SET IDENTITY\_INSERT Section ON;

DECLARE @NextSectionID INT;

SET @NextSectionID = COALESCE((SELECT MAX(SectionID) FROM Section), 0) + 1;

INSERT INTO Section (SectionID, Name, SequenceNumber, Presiding, Premises, Duration, Start, ConferenceID, UCR, DCR, ULC, DLC)

SELECT @NextSectionID, Name, SequenceNumber, Presiding, Premises, Duration, Start, ConferenceID, UCR, DCR, ULC, DLC

FROM INSERTED;

SET IDENTITY\_INSERT Section OFF;

END;

**Тригери для перевірки наступних обмежень цілісності: кожен виступаючий може брати участь в кількох секціях, але за один день він може виступати тільки в одній секції. В одному приміщенні не можуть проводитися одночасно засідання двох секцій:**

AFTER INSERT, UPDATE

AS

BEGIN

-- Перевірка обмеження для приміщень

IF EXISTS (

SELECT 1

FROM Section s1

JOIN Section s2 ON s1.Premises = s2.Premises

AND s1.Start < DATEADD(MINUTE, s2.Duration, s2.Start)

AND DATEADD(MINUTE, s1.Duration, s1.Start) > s2.Start

AND s1.Name <> s2.Name

WHERE (s1.Name IN (SELECT Name FROM inserted) OR s1.Name IN (SELECT Name FROM deleted))

)

BEGIN

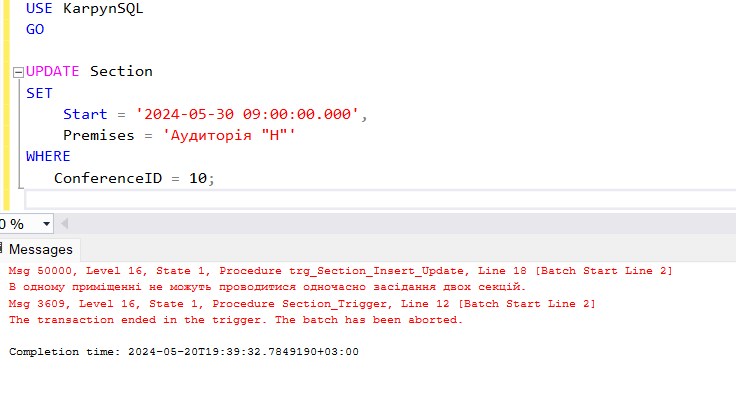
RAISERROR('В одному приміщенні не можуть проводитися одночасно засідання двох секцій.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END

END;

****

AFTER INSERT, UPDATE

AS

BEGIN

-- Перевірка обмеження для виступаючого

IF EXISTS (

SELECT 1

FROM Perfomances p1

JOIN Perfomances p2 ON p1.PerformerID = p2.PerformerID AND p1.Date = p2.Date AND p1.Topic <> p2.Topic

WHERE (p1.Topic IN (SELECT Topic FROM inserted) OR p1.Topic IN (SELECT Topic FROM deleted))

)

BEGIN

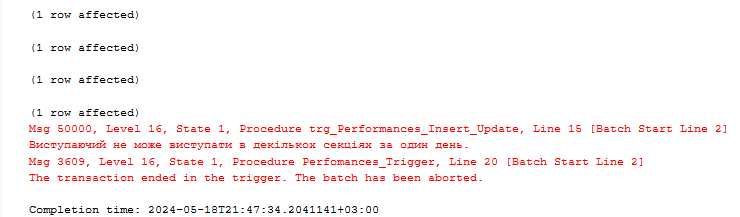
RAISERROR('Виступаючий не може виступати в декількох секціях за один день.', 16, 1);

ROLLBACK TRANSACTION;

RETURN;

END

END;

****

**Посилання на репозиторій GitHub:**

<https://github.com/Olesia-Karpyn/KarpynSQL.git>

**Висновок:** отже я написала тригери відповідно до завдань лабораторної роботи та перевірила їх роботу.