Міністерство освіти і науки України Національний університет «Львівська політехніка»

Інститут прикладної математики та фундаментальних наук Кафедра прикладної математики

# Звіт

про виконання лабораторної роботи №4

з курсу «Організація баз даних та знань»

**Виконав:**

студент групи ПМ-21

Діяк Ігор

**Перевірив:**

кандидат наук

Пабирівський В. В.

## **Лабораторна робота № 4. “Тригери в СУБД Oracle”**

**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:

**Варіант 1**. В одній кімнаті проживає до трьох студентів. Студент не може мати одночасно позиченими книжок на суму більше ніж 100 грн.

**Створення полів UCR, DCR, ULC, DLC у таблицях:**

ALTER TABLE Students ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE DormRooms ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE Professors ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE Subjects ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE Groups ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE Hobbies ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE Books ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE BookLoans ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

ALTER TABLE Grades ADD

UCR NVARCHAR(100),

DCR DATETIME,

ULC NVARCHAR(100),

DLC DATETIME;

**Створення тригерів і модифікування для кожної таблиці:**

CREATE TRIGGER trg\_Students\_Insert

ON Students

AFTER INSERT

AS

BEGIN

UPDATE Students

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE StudentID IN (SELECT StudentID FROM inserted);

END;

CREATE TRIGGER trg\_Students\_Update

ON Students

AFTER UPDATE

AS

BEGIN

UPDATE Students

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE StudentID IN (SELECT StudentID FROM inserted);

END;

CREATE TRIGGER trg\_DormRooms\_Insert

ON DormRooms

AFTER INSERT

AS

BEGIN

UPDATE DormRooms

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE DormRoomID IN (SELECT DormRoomID FROM inserted);

END;

CREATE TRIGGER trg\_DormRooms\_Update

ON DormRooms

AFTER UPDATE

AS

BEGIN

UPDATE DormRooms

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE DormRoomID IN (SELECT DormRoomID FROM inserted);

END;

CREATE TRIGGER trg\_Professors\_Insert

ON Professors

AFTER INSERT

AS

BEGIN

UPDATE Professors

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE ProfessorID IN (SELECT ProfessorID FROM inserted);

END;

CREATE TRIGGER trg\_Professors\_Update

ON Professors

AFTER UPDATE

AS

BEGIN

UPDATE Professors

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE ProfessorID IN (SELECT ProfessorID FROM inserted);

END;

CREATE TRIGGER trg\_Subjects\_Insert

ON Subjects

AFTER INSERT

AS

BEGIN

UPDATE Subjects

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE SubjectID IN (SELECT SubjectID FROM inserted);

END;

CREATE TRIGGER trg\_Subjects\_Update

ON Subjects

AFTER UPDATE

AS

BEGIN

UPDATE Subjects

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE SubjectID IN (SELECT SubjectID FROM inserted);

END;

CREATE TRIGGER trg\_Groups\_Insert

ON Groups

AFTER INSERT

AS

BEGIN

UPDATE Groups

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE GroupID IN (SELECT GroupID FROM inserted);

END;

CREATE TRIGGER trg\_Groups\_Update

ON Groups

AFTER UPDATE

AS

BEGIN

UPDATE Groups

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE GroupID IN (SELECT GroupID FROM inserted);

END;

CREATE TRIGGER trg\_Hobbies\_Insert

ON Hobbies

AFTER INSERT

AS

BEGIN

UPDATE Hobbies

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE HobbyID IN (SELECT HobbyID FROM inserted);

END;

CREATE TRIGGER trg\_Hobbies\_Update

ON Hobbies

AFTER UPDATE

AS

BEGIN

UPDATE Hobbies

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE HobbyID IN (SELECT HobbyID FROM inserted);

END;

CREATE TRIGGER trg\_Books\_Insert

ON Books

AFTER INSERT

AS

BEGIN

UPDATE Books

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE BookID IN (SELECT BookID FROM inserted);

END;

CREATE TRIGGER trg\_Books\_Update

ON Books

AFTER UPDATE

AS

BEGIN

UPDATE Books

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE BookID IN (SELECT BookID FROM inserted);

END;

CREATE TRIGGER trg\_BookLoans\_Insert

ON BookLoans

AFTER INSERT

AS

BEGIN

UPDATE BookLoans

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE BookLoanID IN (SELECT BookLoanID FROM inserted);

END;

CREATE TRIGGER trg\_BookLoans\_Update

ON BookLoans

AFTER UPDATE

AS

BEGIN

UPDATE BookLoans

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE BookLoanID IN (SELECT BookLoanID FROM inserted);

END;

CREATE TRIGGER trg\_Grades\_Insert

ON Grades

AFTER INSERT

AS

BEGIN

UPDATE Grades

SET UCR = SUSER\_SNAME(), DCR = GETDATE()

WHERE GradeID IN (SELECT GradeID FROM inserted);

END;

CREATE TRIGGER trg\_Grades\_Update

ON Grades

AFTER UPDATE

AS

BEGIN

UPDATE Grades

SET ULC = SUSER\_SNAME(), DLC = GETDATE()

WHERE GradeID IN (SELECT GradeID FROM inserted);

END;

Ці тригери відповідають за оновлення полям UCR (User Created) та DCR (Date Created) при вставці нових рядків у таблицю, а також за оновлення полями ULC (User Last Changed) та DLC (Date Last Changed) при зміні вже існуючих рядків у цій же таблиці.

**Написання тригерів відповідно з варіантом для перевірки обмежень цілісності:**

1. Тригер з обмеженням на проживання студентів в одній кімнаті:

CREATE TRIGGER trg\_CheckRoomCapacity

ON Students

AFTER INSERT, UPDATE

AS

BEGIN

IF EXISTS (SELECT 1 FROM Students

WHERE DormRoomID IN (SELECT DormRoomID FROM inserted)

GROUP BY DormRoomID

HAVING COUNT(StudentID) > 3)

BEGIN

RAISERROR ('В одній кімнаті не може проживати більше трьох студентів.', 16, 1);

ROLLBACK TRANSACTION;

END

END;

1. Тригер з обмеженням на позичення книг на суму не більше ніж 100 гривень:

CREATE TRIGGER trg\_CheckBookLoanAmount

ON BookLoans

AFTER INSERT, UPDATE

AS

BEGIN

-- Ініціалізуємо змінну для зберігання загальної вартості позичених книг

DECLARE @TotalAmount MONEY;

-- Перебираємо кожен запис, що був вставлений або оновлений

DECLARE @StudentID INT;

DECLARE StudentCursor CURSOR FOR

SELECT DISTINCT StudentID FROM inserted;

OPEN StudentCursor;

FETCH NEXT FROM StudentCursor INTO @StudentID;

WHILE @@FETCH\_STATUS = 0

BEGIN

-- Обчислюємо загальну вартість усіх книг, позичених студентом

SELECT @TotalAmount = ISNULL(SUM(b.Price), 0)

FROM Books b

INNER JOIN BookLoans bl ON b.BookID = bl.BookID

WHERE bl.StudentID = @StudentID;

-- Якщо сума вартості перевищує 100 грн, виводимо помилку і відкочуємо транзакцію

IF @TotalAmount > 100

BEGIN

RAISERROR ('Студент не може мати позичених книг на суму більше 100 грн.', 16, 1);

ROLLBACK TRANSACTION;

END

FETCH NEXT FROM StudentCursor INTO @StudentID;

END;

CLOSE StudentCursor;

DEALLOCATE StudentCursor;

END;

**Демонстрація роботи тригерів:**

-- Демонстрація роботи тригерів для заповнення полів UCR, DCR, ULC, DLC

-- Створення нового запису

INSERT INTO Groups (GroupName)

VALUES ('КН-34');

SELECT \* FROM Groups;



--Оновлення запису

UPDATE Groups

SET GroupName = 'КН-44'

WHERE GroupID = 1;

SELECT \* FROM Groups;



--Спроба видачі книги студенту коли борг перевищить 100 грн

INSERT INTO Books (Title, Author, Genre, Price)

VALUES

('Вступ до алгоритмів', 'Томас Кормен', 'Наука', 70.00),

('Історія України', 'Михайло Грушевський', 'Історія', 50.00);

--Помилки не повинно бути

INSERT INTO BookLoans (BookID, StudentID, LoanDate)

VALUES

(1, 1, '2023-10-01');

--Повинна бути помилка

INSERT INTO BookLoans (BookID, StudentID, LoanDate)

VALUES

(2, 1, '2023-10-01');