**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Институт компьютерных технологий и защиты информации**

**Кафедра прикладной математики и информатики**

Лабораторная работа №6

По дисциплине: Программирование на Transact SQL

 Выполнил: студент группы 4311

                    Федотов А.Д., Червов В.А.

                                                                                      Проверил: Зайдулин С.С.

                  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

Казань, 2024 год

**Краткие теоретические сведения**

Триггер - специальный вид хранимой процедуры, которая выполняется, когда оператор INSERT, UPDATE, или DELETE изменяет данные в указанной таблице. Триггер может запрашивать данные другой таблицы и может включать сложные операторы Transact-SQL. Триггеры часто создаются для обеспечения ссылочной целостности или согласованности среди логически связанных данных в различных таблицах.

Существует две категории триггеров DML: AFTER и INSTEAD OF. Триггеры AFTER. Триггеры AFTER выполняются после действия операторов INSERT, UPDATE или DELETE. Определение триггера AFTER имеет то же самое определение, что и триггер FOR, который являлся единственным доступным типом в более ранних версиях Microsoft SQL Server. Вы можете определять триггеры AFTER только для таблиц. Триггеры INSTEAD OF. Триггеры INSTEAD OF выполняются вместо обычных триггерных действий (вставка, обновление, удаление). Триггеры INSTEAD OF также могут быть определены на представлениях с одной или более базисными таблицами, чтобы расширить действия операторов обновления, поддерживаемые представлениями.

Триггер INSERT - триггер, который выполняется всякий раз, когда оператор INSERT вставляет данные в таблицу или представление, для которого определен этот триггер.

Триггер DELETE – специальный вид хранимой процедуры, которая выполняется всякий раз, когда оператор DELETE удаляет данные из таблицы или представления, на котором этот триггер был определен.

Триггер UPDATE - триггер, который выполняется всякий раз, когда оператор INSERT изменяет данные в таблице или представлении, на котором определен этот триггер.

Триггер INSTEAD OF выполняется вместо обычного действия триггера. Триггеры INSTEAD OF могут также быть определены на представлениях с одной или более базовыми таблицами, где они могут расширить действия операторов обновления, поддерживаемые представлениями.

**Задание на лабораторную работу**

1. Изучить материалы файла «[Поддержка целостности данных и триггеры](https://bb.kai.ru:8443/bbcswebdav/pid-460297-dt-content-rid-4829411_1/xid-4829411_1)».
2. Ответить на контрольные вопросы.
3. Создать триггер для INSERT
4. Создать триггер для DELETE
5. Создать триггер для UPDATE
6. Создать триггер для INSTEAD OF

**Ход работы**

-- Создание базы данных "medicine"

CREATE DATABASE medicine

GO

-- Использование базы данных "medicine"

USE medicine

GO

-- Создание таблицы "Medicines" для хранения информации о лекарствах

CREATE TABLE Medicines (

id INT PRIMARY KEY,

name VARCHAR(50),

quantity INT,

price DECIMAL(10, 2)

);

-- Вставка данных в таблицу "Medicines"

INSERT INTO Medicines (id, name, quantity, price) VALUES

(1, 'Paracetamol', 100, 5.50),

(2, 'Amoxicillin', 50, 8.75),

(3, 'Ibuprofen', 75, 4.20);

-- Создание таблицы "Sales" для хранения информации о продажах

CREATE TABLE Sales (

id INT PRIMARY KEY,

medicine\_id INT,

quantity\_sold INT,

total\_price DECIMAL(10, 2)

);

-- Вставка данных в таблицу "Sales"

INSERT INTO Sales (id, medicine\_id, quantity\_sold, total\_price) VALUES

(1, 1, 10, 55.00),

(2, 2, 5, 43.75),

(3, 3, 15, 63.00);

-- Создание таблицы "Log\_Medicines" для хранения логов операций с таблицей "Medicines"

CREATE TABLE Log\_Medicines (

id INT IDENTITY PRIMARY KEY,

operation\_desc VARCHAR(100),

operation\_date DATE,

operation\_type VARCHAR(10),

quantity\_before INT,

quantity\_after INT

);

-- Выборка всех данных из таблицы "Medicines"

SELECT \* FROM Medicines;

-- Выборка всех данных из таблицы "Sales"

SELECT \* FROM Sales;

-- Выборка всех данных из таблицы "Log\_Medicines"

SELECT \* FROM Log\_Medicines;

-- Создание триггера "insert\_trigger" на таблице "Medicines" для логирования операций вставки

CREATE TRIGGER insert\_trigger

ON Medicines

AFTER INSERT

AS

BEGIN

INSERT INTO Log\_Medicines (operation\_type, operation\_desc, operation\_date, quantity\_before, quantity\_after)

SELECT 'INSERT', 'Вставка записи в таблицу Medicines', GETDATE(), NULL, I.quantity

FROM inserted I;

END;

GO

-- Создание триггера "delete\_trigger" на таблице "Medicines" для логирования операций удаления

CREATE TRIGGER delete\_trigger

ON Medicines

AFTER DELETE

AS

BEGIN

INSERT INTO Log\_Medicines (operation\_type, operation\_desc, operation\_date, quantity\_before, quantity\_after)

SELECT 'DELETE', 'Удаление записи из таблицы Medicines', GETDATE(), D.quantity, NULL

FROM deleted D;

END;

GO

-- Создание триггера "update\_trigger" на таблице "Medicines" для логирования операций обновления

CREATE TRIGGER update\_trigger

ON Medicines

AFTER UPDATE

AS

BEGIN

INSERT INTO Log\_Medicines (operation\_type, operation\_desc, operation\_date, quantity\_before, quantity\_after)

SELECT 'UPDATE', 'Обновление записи в таблице Medicines', GETDATE(), D.quantity, I.quantity

FROM deleted D

INNER JOIN inserted I ON D.id = I.id;

END;

GO

-- Создание триггера "insteadof\_trigger" на таблице "Sales" для перехвата операций вставки, обновления и удаления

CREATE TRIGGER insteadof\_trigger

ON Sales

INSTEAD OF INSERT, UPDATE, DELETE

AS

BEGIN

INSERT INTO Log\_Medicines (operation\_type, operation\_desc, operation\_date, quantity\_before, quantity\_after)

VALUES ('UNSPECIFI', 'Неопределенная операция на таблице Sales', GETDATE(), NULL, NULL);

END;

GO

-- Удаление записи из таблицы "Medicines" с id = 2

DELETE FROM Medicines WHERE id = 2;

-- Вставка новой записи в таблицу "Medicines"

INSERT INTO Medicines (id, name, quantity, price) VALUES (4, 'Aspirin', 50, 3.25);

-- Обновление количества в таблице "Medicines" для записи с id = 1

UPDATE Medicines SET quantity = 90 WHERE id = 1;

-- Удаление записи из таблицы "Sales" с id = 2

DELETE FROM Sales WHERE id = 2;

-- Повторная выборка всех данных из таблицы "Medicines"

SELECT \* FROM Medicines;

-- Повторная выборка всех данных из таблицы "Sales"

SELECT \* FROM Sales;

-- Повторная выборка всех данных из таблицы "Log\_Medicines"

SELECT \* FROM Log\_Medicines;

**Вывод**

В ходе выполнения лабораторной работы были получены практические навыки по созданию и использованию триггеров. А именно с такими категориями триггеров как AFTER и INSTEAD OF.

В целом, лабораторная работа позволила ознакомиться с основами создания и использования триггеров в базе данных, их настройкой и применением для решения практических задач.