**Триггеры в SQL Server (SSMS)**

**Цель:**

Познакомить с понятием, типами и применением **триггеров**. Научиться создавать триггеры, использовать псевдотаблицы INSERTED, DELETED, блокировать действия и вести аудит изменений.

**🔹 Что такое триггер?**

**Триггер** — это объект базы данных, который **автоматически срабатывает** при выполнении определённого действия (INSERT, UPDATE, DELETE) над таблицей.

**🔹 Типы триггеров в SQL Server:**

| **Тип** | **Описание** |
| --- | --- |
| **AFTER** | Выполняется **после** действия (по умолчанию) |
| **INSTEAD OF** | Выполняется **вместо** действия (заменяет его) |
| **DDL-триггеры** | Срабатывают при изменении схемы (CREATE, ALTER) |

**🔹 Псевдотаблицы:**

| **Таблица** | **Доступна при** | **Содержит данные** |
| --- | --- | --- |
| INSERTED | INSERT/UPDATE | Новые значения |
| DELETED | DELETE/UPDATE | Старые значения |

**🔹 Пример AFTER INSERT триггера:**

sql

КопироватьРедактировать

CREATE TRIGGER trg\_AfterInsert\_User

ON Users

AFTER INSERT

AS

BEGIN

PRINT 'Добавлен новый

**Вариант 1: Аудит зарплаты**

**Таблицы:**

* Employees (Id, Name, Salary)
* SalaryLog (Id, EmpId, OldSalary, NewSalary, ChangeDate)

**Задания:**

1. Создайте AFTER UPDATE триггер, который сохраняет старую и новую зарплату в SalaryLog.
2. Если зарплата увеличивается более чем в 2 раза — запретить изменение.

**Вариант 2: Блокировка удаления**

**Таблицы:**

* Courses (Id, Title)
* Enrollments (Id, StudentId, CourseId)

**Задания:**

1. Создайте INSTEAD OF DELETE триггер на Courses, который:
   * Проверяет наличие студентов на курсе.
   * Запрещает удаление, если курс используется.

**Вариант 3: Логирование изменений**

**Таблицы:**

* Products (Id, Name, Price)
* PriceAudit (Id, ProductId, OldPrice, NewPrice, UpdatedAt)

**Задания:**

1. AFTER UPDATE триггер: логирует изменение цены в PriceAudit.
2. Добавить проверку: если цена уменьшилась — вывести сообщение Warning: Price drop.

**Вариант 4: Контроль удалений**

**Таблицы:**

* Books (Id, Title, IsArchived)
* DeleteLog (Id, BookId, DeletedAt)

**Задания:**

1. Запретить удаление книг, если IsArchived = 0.
2. Если удаление выполнено — добавить запись в DeleteLog.

**Вариант 5: История удаления студентов**

**Таблицы:**

* Students (Id, Name)
* DeletedStudents (Id, Name, DeletedAt)

**Задания:**

1. AFTER DELETE триггер, который переносит удалённые записи в DeletedStudents.

**SQL-шаблоны таблиц и данных**

**Employees + SalaryLog (Вариант 1)**

CREATE TABLE Employees (

Id INT IDENTITY PRIMARY KEY,

Name NVARCHAR(100),

Salary DECIMAL(10, 2)

);

GO

CREATE TABLE SalaryLog (

Id INT IDENTITY PRIMARY KEY,

EmpId INT,

OldSalary DECIMAL(10, 2),

NewSalary DECIMAL(10, 2),

ChangeDate DATETIME

);

GO

INSERT INTO Employees (Name, Salary)

VALUES ('Айгерим', 300000), ('Алихан', 500000);

GO

**Courses + Enrollments (Вариант 2)**

CREATE TABLE Courses (

Id INT IDENTITY PRIMARY KEY,

Title NVARCHAR(100)

);

GO

CREATE TABLE Enrollments (

Id INT IDENTITY PRIMARY KEY,

StudentId INT,

CourseId INT

);

GO

INSERT INTO Courses (Title) VALUES ('SQL'), ('C#');

INSERT INTO Enrollments (StudentId, CourseId) VALUES (1, 1);

GO

**Products + PriceAudit (Вариант 3)**

CREATE TABLE Products (

Id INT IDENTITY PRIMARY KEY,

Name NVARCHAR(100),

Price DECIMAL(10, 2)

);

GO

CREATE TABLE PriceAudit (

Id INT IDENTITY PRIMARY KEY,

ProductId INT,

OldPrice DECIMAL(10, 2),

NewPrice DECIMAL(10, 2),

UpdatedAt DATETIME

);

GO

INSERT INTO Products (Name, Price)

VALUES ('Ноутбук', 500000), ('Монитор', 200000);

GO

**Books + DeleteLog (Вариант 4)**

CREATE TABLE Books (

Id INT IDENTITY PRIMARY KEY,

Title NVARCHAR(100),

IsArchived BIT

);

GO

CREATE TABLE DeleteLog (

Id INT IDENTITY PRIMARY KEY,

BookId INT,

DeletedAt DATETIME

);

GO

INSERT INTO Books (Title, IsArchived)

VALUES ('SQL Server', 0), ('Python Guide', 1);

GO

**Students + DeletedStudents (Вариант 5)**

CREATE TABLE Students (

Id INT IDENTITY PRIMARY KEY,

Name NVARCHAR(100)

);

GO

CREATE TABLE DeletedStudents (

Id INT,

Name NVARCHAR(100),

DeletedAt DATETIME

);

GO

INSERT INTO Students (Name)

VALUES ('Нурсултан'), ('Алия');

GO