  
Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Інформаційно-комунікаційні системі:

Бази даних

Студент: Бичок Вадим Вячеславович

Робота: 8

Назва: СТВОРЕННАЯ І ВИКОРИСТАННЯ ТРИГЕРІВ.

СУБД: MSSQL

Використовуємо Базу Даних mssql для виконання SQL-запитів  
USE mssql;

Видаляємо вже створений тригер, якщо він існує.

DROP TRIGGER IF EXISTS DeletingBook;

Додаємо каскадне видалення, щоб коли видаляється книга, видалялися усі записи про неї.

ALTER TABLE ReadersTakingBook

DROP CONSTRAINT IF EXISTS BooksForeign;

ALTER TABLE ReadersTakingBook

ADD CONSTRAINT BooksForeign

FOREIGN KEY (BookID)

REFERENCES Books

ON DELETE CASCADE;

GO

Створюємо тригер, який буде замінювати стандартний принцип видалення, на мій бо тригер через FOR не коректно працює.

CREATE TRIGGER DeletingBook

ON Books

INSTEAD OF DELETE

AS

BEGIN

Виводимо що тригер спрацював.

PRINT 'Trigger executed';

З таблиці у яку заносяться рядки при видаленні з таблиці Books виводимо усе для перевірки.

SELECT \*

FROM deleted;

Пишемо умови після виконання яких тригер буде забороняти видалення з таблиці.

IF EXISTS

Для зменшення навантаження на базу даних, будемо повертати просто ‘TRUE’, замість якогось рядку. Вибираємо рядки з двох таблиць, ту в якій не має бути пустих значень, що означає що не повернули книгу, тому не можна її видалити з таблиці книг, і таблицю в якій у нас міститься рядки які ми хочемо видалити, якщо нам повернеться 'TRUE', тригер забороняє видалення, і повертає усі зміни, якщо не повернеться, то ідемо далі по коду.

(SELECT 'TRUE'

FROM deleted

INNER JOIN ReadersTakingBook

ON deleted.BookID = ReadersTakingBook.BookID

WHERE ReadersTakingBook.DateOfReceipt IS NULL)

BEGIN

RAISERROR('Can`t delete book if she isset in readers!!!', 16, 1)

ROLLBACK TRAN;

END

ELSE

Якщо попередня умова не виконалася, то прописуємо видалення з таблиці Books тих записів, які надійшли в таблицю deleted.

BEGIN

DELETE FROM Books

WHERE BookID IN (SELECT BookID FROM deleted)

END;

END;

Перевіряємо наші тригери

GO

SELECT \* FROM Books

WHERE BookID = 8;

SELECT \* FROM ReadersTakingBook

WHERE BookID = 8;

DELETE FROM Books

WHERE BookID = 8;  
Зображення, що містить текст, Шрифт, знімок екрана, алгебра

Автоматично згенерований опис  
Зображення, що містить текст, знімок екрана, Шрифт, програмне забезпечення

Автоматично згенерований опис

GO

Додатково, коли тригер не працював перевіряв чи є такі записи які не можна видаляти.

SELECT Name, Books.BookID, IssueDate, DateOfReceipt

FROM Books

INNER JOIN ReadersTakingBook

ON Books.BookID = ReadersTakingBook.BookID

WHERE ReadersTakingBook.DateOfReceipt IS NULL

AND Books.BookID = 8;

SELECT \*

FROM ReadersTakingBook

WHERE BookID = 8 AND DateOfReceipt IS NULL;

SELECT \* FROM Books

WHERE BookID = 8;

SELECT \* FROM ReadersTakingBook

WHERE BookID = 8;

SELECT \* FROM Books;

SELECT \* FROM ReadersTakingBook;

GO

DROP TRIGGER IF EXISTS UpdateBooks;

DROP TRIGGER IF EXISTS InsertBooks;

DROP VIEW IF EXISTS ViewRarities;

GO

CREATE VIEW ViewRarities

AS

SELECT

BookID,

Name,

Date,

Price \* POWER(1.05, YEAR(GETDATE()) - YEAR(Date)) AS RedemptionPrice

FROM Books;

GO

Створимо окремі тригери на запис і зміну, бо для кожного трохи різні принципи оновлення нашої таблиці раритетів.

Якщо до

CREATE TRIGGER InsertBooks

ON Books

FOR INSERT

AS

BEGIN

Дозволяємо вставляти явні значення у стовпець ідентифікаторів таблиці. Так як ми скопіювали стовпець у минулій лабі з автовизначенням стовпця BookID.

SET IDENTITY\_INSERT Rarities ON;

Для тригера на запис прописуємо дію запису до таблиці раритетів, і принци обрахунку ціни викупу.

INSERT INTO Rarities (BookID, Author, Name, Date, Price, IfNew, WhatAbout, RedemptionPrice)

SELECT i.BookID, i.Author, i.Name, i.Date, i.Price, i.IfNew, i.WhatAbout, i.Price \* POWER(1.05, YEAR(GETDATE()) - YEAR(i.Date))

FROM inserted i

WHERE i.Date < '1980-01-01';

END;

GO

CREATE TRIGGER UpdateBooks

ON Books

FOR UPDATE

AS

BEGIN

Для зміни прописуємо оновлення всіх стовпців, щоб не гратися з IF, і перераховуємо нашу ціну викупу.

UPDATE Rarities

SET Author = i.Author,

Name = i.Name,

Date = i.Date,

Price = i.Price,

IfNew = i.IfNew,

WhatAbout = i.WhatAbout,

RedemptionPrice = i.Price \* POWER(1.05, YEAR(GETDATE()) - YEAR(Rarities.Date))

FROM Rarities

INNER JOIN inserted i ON Rarities.BookID = i.BookID;

END;

GO

Перевіряємо наш тригер.

DELETE FROM Books WHERE Name = '0';

DELETE FROM Books WHERE Name = '1';

DELETE FROM Rarities WHERE Name = '0';

DELETE FROM Rarities WHERE Name = '1';

GO

INSERT Books (Author, Name, Date, Price, IfNew, WhatAbout)

VALUES ('0', '0', '1970-01-01', 10000, 'No', '000000000000000000000000');

INSERT Books (Author, Name, Date, Price, IfNew, WhatAbout)

VALUES ('1', '1', '1970-01-01', 10000, 'No', '000000000000000000000000');

SELECT \* FROM Books;

SELECT \* FROM Rarities