ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент |  |  |  | А.В. Бржезовский |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| ТРИГГЕРЫ |
| по курсу: МЕТОДЫ И СРЕДСТВА ПРОЕКТИРОВАНИЯ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3641 |  |  |  | Фомин.М.Б |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2019

**Задание: (Вариант №16 )**

Создайте базу данных для хранения следующих сведений: город, район, квартира, комната, площадь, цена, агент, продажа.

**Задачи:**

—По аналогии с примерами, приведенными в п. 7.2, создать в БД Триггеры, реализующие:

— триггеры каждого вида (***after***, ***instead of***) для каждой из операций (***insert***, ***update***, ***delete***), обеспечивающие активную целостность данных;

— триггер, реализующий вычисления или формирование статистики или ведение истории изменений в БД;

**Результат выполнения работы:**

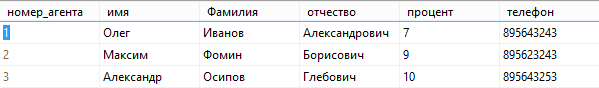
1. **Триггеры: (AFTER)**

Создадим триггеры, позволяющие реализовать:

1. *При изменении значений в таблице, не позволяющий вставить значение при условиях: (вставляется значение процента агенту по продаже, если процент превышает опр. значение, то триггер не позволить изменить значение);*

Для данного триггера нам понадобится 1 таблица «агент по продаже» (табл.1).

***Таблица 1.*** Данные таблицы «агент по продаже»:



Запрос выполняется с помощью следующего скрипта:

use bd\_estate\_agency\_tr

go

create trigger updAgent

on агент\_по\_продаже

after update

as

if (select max(процент) from агент\_по\_продаже) > 10

begin

print 'Максимальный процент с продаж агентам 10%'

rollback transaction

end

go

Результат выполнения данного скрипта:

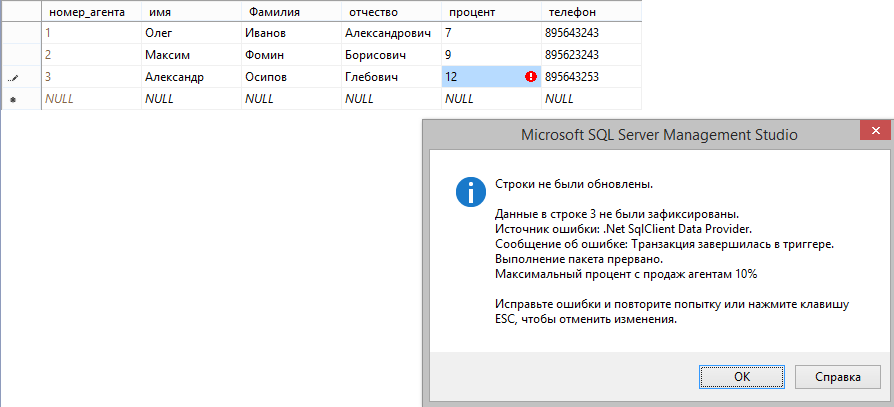


Рис.1.1. Результат выполнения

1. *Вставка значения в таблицу, и вывод с помощью триггера ответа о том какое значение было вставлено.*

Запрос выполняется с помощью следующего скрипта:

CREATE TABLE FlatOperation

(

Id INT IDENTITY PRIMARY KEY,

FlatId INT NOT NULL,

Operation VARCHAR(200) NOT NULL,

)

drop trigger InsertFlat

go

CREATE TRIGGER InsertFlat

ON Квартира

AFTER INSERT

AS

SET IDENTITY\_INSERT FlatOperation off;

INSERT INTO FlatOperation (FlatId, Operation)

SELECT id\_квартиры, 'Добавлена квартира ' + CAST(номер\_квартиры AS VARCHAR) + ' с количеством комнат ' + CAST(кол\_комнат AS VARCHAR) +

' в доме ' + CAST(дом AS VARCHAR) + ' по цене кв.м ' + CAST(цена\_кв\_м AS VARCHAR)

FROM INSERTED

INSERT INTO квартира( номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом)

VALUES(99,3,40000,30,13)

SELECT \* FROM FlatOperation

Результат выполнения данного скрипта:



Рис.1.2.1. Вставка значения в табл. «квартира» и вывод о нем информации

Результат выполнения данного скрипта:

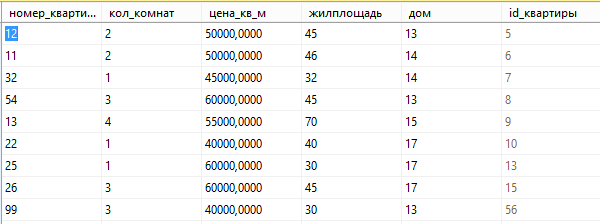


Рис.1.2.2. Результат выполнения

1. *Удаление значения из таблицы «квартира», и вывод с помощью триггера ответа о том какое значение было удалено.*

Запрос выполняется с помощью следующего скрипта:

drop trigger DelFlat

go

use bd\_estate\_agency\_tr

go

CREATE TRIGGER DelFlat

ON квартира

AFTER DELETE

AS

INSERT INTO FlatOperation (FlatId, Operation)

SELECT id\_квартиры, 'Удалена квартира ' + CAST(номер\_квартиры AS VARCHAR) + ' с количеством комнат ' + CAST(кол\_комнат AS VARCHAR) +

' в доме ' + CAST(дом AS VARCHAR) + ' по цене кв.м ' + CAST(цена\_кв\_м AS VARCHAR)

FROM DELETED

INSERT INTO квартира( номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом)

VALUES(99,3,40000,30,13)

DELETE FROM квартира

WHERE номер\_квартиры=99

SELECT \* FROM FlatOperation

go

Результат выполнения данного скрипта:



Рис.1.3. Удаление значения из таблицы «квартира»

1. **Триггеры:(INSTEAD OF)**

Триггер INSTEAD OF срабатывает вместо операции с данными. Он определяется в принципе также, как триггер AFTER, за тем исключением, что он может определяться только для одной операции - INSERT, DELETE или UPDATE. И также он может применяться как для таблиц, так и для представлений (триггер AFTER применяется только для таблиц).

1. *Использование триггера INSTEAD OF для замены операции удаления(Delete) для таблицы «тип дома». (Если удаляется какой-нибудь тип, то он не удаляется, эта операция заменяется триггером, в столбец Del, ставится «1», в строке типа который должен быть удален ):*

Запрос выполняется с помощью следующего скрипта:

drop trigger InsteadDel

go

CREATE TRIGGER InsteadDel

ON тип\_дома

INSTEAD OF DELETE

AS

UPDATE тип\_дома

SET Del = 1

WHERE серия in (SELECT серия FROM deleted)

go

INSERT INTO тип\_дома(серия, описание)

VALUES ('тс777','трёх-комнатная квартира')

DELETE FROM тип\_дома

WHERE серия='тс777';

SELECT \* FROM тип\_дома;

go

Результат выполнения данного скрипта:

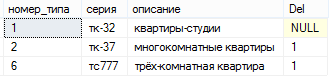


Рис.1.4.Колонка «*Del» в 1,* вместо операции удаления

1. *Использование триггера INSTEAD OF для замены операции вставки(Insert) для таблицы «квартира». (Если вставляется какая-нибудь запись в таблицу, то идет проверка, если в доме, количество квартир больше 5, то запись не вставляется).*

Запрос выполняется с помощью следующего скрипта:

drop trigger InsteadIns

go

use bd\_estate\_agency\_tr

go

create trigger InsteadIns

on квартира

instead of insert

as

if exists (select \* from квартира

where id\_квартиры in (select id\_квартиры from inserted)

group by id\_квартиры

having count(дом) >= 5)

print 'Слишком много квартир'

else

insert into квартира select номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом from inserted

go

1. ***Формирование статистики***
2. *Использование триггера для вставки, удаления,обновления значений и ведения статистики, для таблицы «квартира». (Если вставляется, удаляется квартира автоматически изменяются значения таблицы «Изм\_Квартира,).*

Запрос выполняется с помощью следующего скрипта:

use bd\_estate\_agency\_tr

go

drop trigger Изм\_Квартира

go

create trigger Изм\_Квартира

on квартира

for insert, update, delete

as

declare @ci int, @cd int

select @ci = count(\*) from inserted

select @cd = count(\*) from deleted

if @ci > 1 and @cd > 1 return -- инструкция затронула 1 строку

declare @колКомнат smallint

if @колКомнат = 1

update Количество\_комнат

set кол\_комн1 = кол\_комн1 + 1

if @колКомнат = 2

update Количество\_комнат

set кол\_комн2 = кол\_комн2 + 1

if @колКомнат = 3

update Количество\_комнат

set кол\_комн3 = кол\_комн3 + 1

if @колКомнат = 4

update Количество\_комнат

set кол\_комн4 = кол\_комн4 + 1

if @колКомнат = 5

update Количество\_комнат

set кол\_комн5 = кол\_комн5 + 1

if @колКомнат = 1

update Количество\_комнат

set кол\_комн1 = кол\_комн1 - 1

if @колКомнат = 2

update Количество\_комнат

set кол\_комн2 = кол\_комн2 - 1

if @колКомнат = 3

update Количество\_комнат

set кол\_комн3 = кол\_комн3 - 1

if @колКомнат = 4

update Количество\_комнат

set кол\_комн4 = кол\_комн4 - 1

if @колКомнат = 5

update Количество\_комнат

set кол\_комн5 = кол\_комн5 - 1

INSERT INTO квартира( номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом)

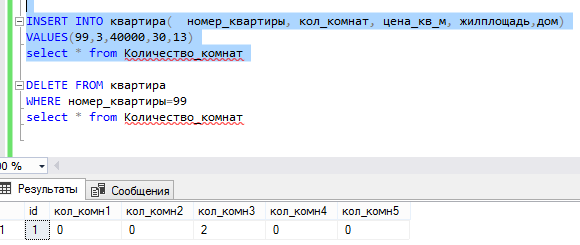
VALUES(99,3,40000,30,13)

select \* from Количество\_комнат

DELETE FROM квартира

WHERE номер\_квартиры=99

select \* from Количество\_комнат



**Выводы:**

В ходе выполнения данной работы была реализована база данных театра, в соответствии с требованиями заявленными в данной работе, а так же построена ее схема и написан скрипт, были написаны все триггеры требуемые в задании.

***ПРИЛОЖЕНИЕ А.*** Скрипт БД:

drop trigger updAgent

go

use bd\_estate\_agency\_tr

go

create trigger updAgent

on агент\_по\_продаже

after update

as

if (select max(процент) from агент\_по\_продаже) > 10

begin

print 'Максимальный процент с продаж агентам 10%'

rollback transaction

end

go

drop trigger DelFlat

go

use bd\_estate\_agency\_tr

go

CREATE TRIGGER DelFlat

ON квартира

AFTER DELETE

AS

INSERT INTO FlatOperation (FlatId, Operation)

SELECT id\_квартиры, 'Удалена квартира ' + CAST(номер\_квартиры AS VARCHAR) + ' с количеством комнат ' + CAST(кол\_комнат AS VARCHAR) +

' в доме ' + CAST(дом AS VARCHAR) + ' по цене кв.м ' + CAST(цена\_кв\_м AS VARCHAR)

FROM DELETED

INSERT INTO квартира( номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом)

VALUES(99,3,40000,30,13)

DELETE FROM квартира

WHERE номер\_квартиры=99

SELECT \* FROM FlatOperation

Go

CREATE TABLE FlatOperation

(

Id INT IDENTITY PRIMARY KEY,

FlatId INT NOT NULL,

Operation VARCHAR(200) NOT NULL,

)

drop trigger InsertFlat

go

CREATE TRIGGER InsertFlat

ON Квартира

AFTER INSERT

AS

SET IDENTITY\_INSERT FlatOperation off;

INSERT INTO FlatOperation (FlatId, Operation)

SELECT id\_квартиры, 'Добавлена квартира ' + CAST(номер\_квартиры AS VARCHAR) + ' с количеством комнат ' + CAST(кол\_комнат AS VARCHAR) +

' в доме ' + CAST(дом AS VARCHAR) + ' по цене кв.м ' + CAST(цена\_кв\_м AS VARCHAR)

FROM INSERTED

INSERT INTO квартира( номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом)

VALUES(99,3,40000,30,13)

SELECT \* FROM FlatOperation

drop trigger InsteadDel

go

CREATE TRIGGER InsteadDel

ON тип\_дома

INSTEAD OF DELETE

AS

UPDATE тип\_дома

SET Del = 1

WHERE серия in (SELECT серия FROM deleted)

go

INSERT INTO тип\_дома(серия, описание)

VALUES ('тс777','трёх-комнатная квартира')

DELETE FROM тип\_дома

WHERE серия='тс777';

SELECT \* FROM тип\_дома;

Go

drop trigger InsteadIns

go

use bd\_estate\_agency\_tr

go

create trigger InsteadIns

on квартира

instead of insert

as

if exists (select \* from квартира

where id\_квартиры in (select id\_квартиры from inserted)

group by id\_квартиры

having count(дом) >= 5)

print 'Слишком много квартир'

else

insert into квартира select номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом from inserted

go

drop trigger Изм\_Квартира

go

create trigger Изм\_Квартира

on квартира

for insert, update, delete

as

declare @ci int, @cd int

select @ci = count(\*) from inserted

select @cd = count(\*) from deleted

if @ci > 1 and @cd > 1 return -- инструкция затронула 1 строку

declare @колКомнат smallint

if @колКомнат = 1

update Количество\_комнат

set кол\_комн1 = кол\_комн1 + 1

if @колКомнат = 2

update Количество\_комнат

set кол\_комн2 = кол\_комн2 + 1

if @колКомнат = 3

update Количество\_комнат

set кол\_комн3 = кол\_комн3 + 1

if @колКомнат = 4

update Количество\_комнат

set кол\_комн4 = кол\_комн4 + 1

if @колКомнат = 5

update Количество\_комнат

set кол\_комн5 = кол\_комн5 + 1

if @колКомнат = 1

update Количество\_комнат

set кол\_комн1 = кол\_комн1 - 1

if @колКомнат = 2

update Количество\_комнат

set кол\_комн2 = кол\_комн2 - 1

if @колКомнат = 3

update Количество\_комнат

set кол\_комн3 = кол\_комн3 - 1

if @колКомнат = 4

update Количество\_комнат

set кол\_комн4 = кол\_комн4 - 1

if @колКомнат = 5

update Количество\_комнат

set кол\_комн5 = кол\_комн5 - 1

INSERT INTO квартира( номер\_квартиры, кол\_комнат, цена\_кв\_м, жилплощадь,дом)

VALUES(99,3,40000,30,13)

select \* from Количество\_комнат

DELETE FROM квартира

WHERE номер\_квартиры=99

select \* from Количество\_комнат