ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| ХРАНИМЫЕ ПРОЦЕДУРЫ |
| 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. 6.2, создать в БД ХП, реализующие:

— вставку с пополнением справочников;

— удаление с очисткой;

— каскадное удаление;

— вычисление и возврат значения агрегатной функции;

— формирование статистики во временной таблице

**Результат выполнения работы:**

1. **Хранимые процедуры:**

Составим процедуры, позволяющие реализовать:

1. *вставку с пополнением справочников (вставляется информация о квартире, если указанный дом отсутствует в БД, новая запись добавляется в таблицу с перечнем домов);*

***Таблица 1.*** Данные таблицы «дом»:



***Таблица 2.*** Данные таблицы «квартира»:



Запрос выполняется с помощью следующей процедуры:

CREATE PROCEDURE AddFlat --Создаем процедуру добавляющую новую квартиру

(

--Входящие параметры

@id\_дома int,

@номер\_квартиры int,

@кол\_комнат int,

@цена\_кв\_м money,

@жилплощадь int

)

AS

SET IDENTITY\_INSERT дом ON;

IF not exists (SELECT \* FROM дом WHERE id\_дома=@id\_дома) --если дома для квартиры не существует ,создаём его

BEGIN

SELECT \* FROM дом

INSERT INTO дом (id\_дома, номер\_дома, улица, район, серия) VALUES (@id\_дома, '777', 1, 5,1)

SELECT \* FROM дом

END

begin

--Добавляем новую запись

SELECT \* FROM квартира

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

VALUES (@номер\_квартиры,@кол\_комнат, @цена\_кв\_м,@жилплощадь,@id\_дома)

--Возвращаем данные

SELECT \* FROM квартира

end

GO

EXECUTE AddFlat @id\_дома = 2, @номер\_квартиры = 4, @кол\_комнат= 4, @цена\_кв\_м = 40000,@жилплощадь = 77

drop procedure AddFlat

go

Результат выполнения данного скрипта:

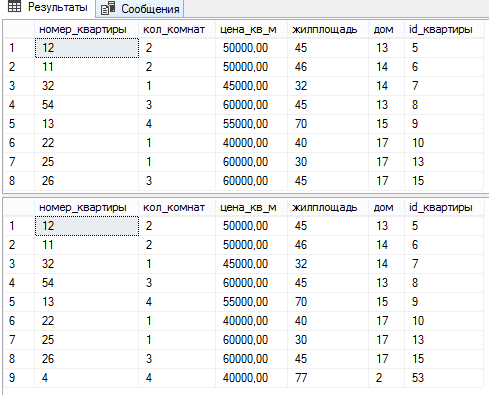


Рис.1.1. Вставка значения в таблицу справочник «квартира»

1. *удаление с очисткой справочников (удаляется информация о квартире , если у дома не существует квартир, запись этого дома удаляется из таблицы дом*

***Таблица 3.*** Данные таблицы «дом»:



***Таблица 4.*** Данные таблицы «квартира»:



Запрос выполняется с помощью следующего скрипта:

use bd\_estate\_agency\_tr

go

--Создаем процедуру

CREATE PROCEDURE del

(

--Входящие параметры

@id\_дома int,

@кол\_комнат int

)

AS

SET IDENTITY\_INSERT квартира ON;

SELECT \* FROM квартира join дом on квартира.дом = дом.id\_дома

DELETE FROM квартира --удаляем квартиру

WHERE дом=@id\_дома and кол\_комнат>@кол\_комнат

IF not exists (SELECT \* FROM квартира WHERE дом=@id\_дома) --если у дома больше нет квартир,удаляем его

BEGIN

DELETE FROM дом

WHERE id\_дома=@id\_дома

END

SELECT \* FROM квартира join дом on дом = дом.id\_дома

GO

EXECUTE del @id\_дома = 2,@кол\_комнат = 3

drop procedure del

go

Результат выполнения данного скрипта:

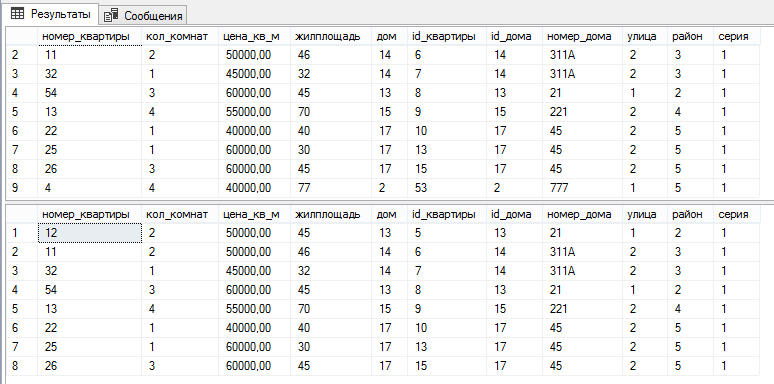


Рис.1.2.1. удаление значения из табл. «дом»

1. *каскадное удаление (при наличии условия ссылочной целостности* ***no action*** *перед удалением записи о доме удаляются записи о всех квартирах этого дома)*

Запрос выполняется с помощью следующего скрипта:

--Создаем процедуру

CREATE PROCEDURE CasDel

(

--Входящие параметры

@id\_дом int

)

AS

SET IDENTITY\_INSERT дом ON;

SELECT \* FROM дом join квартира on дом.id\_дома = квартира.дом

DELETE FROM квартира --удаляем квартиру

WHERE дом= @id\_дом

DELETE FROM дом --удаляем дом

WHERE id\_дома= @id\_дом

SELECT \* FROM дом join квартира on дом.id\_дома = квартира.дом

GO

EXECUTE CasDel @id\_дом = 2

drop procedure CasDel

go

Результат выполнения данного скрипта:

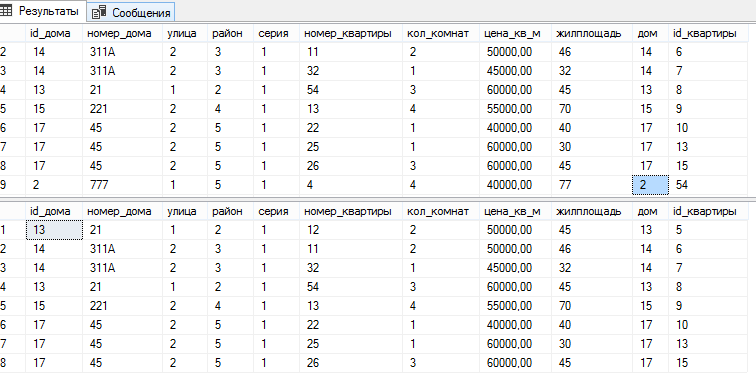


Рис.1.3.1. Каскадное удаление значения из таблицы «дом»

1. *вычисление и возврат значения агрегатной функции (на примере одного из запросов из задания):*

*Запрос: Найти* дома где площадь квартир больше средней

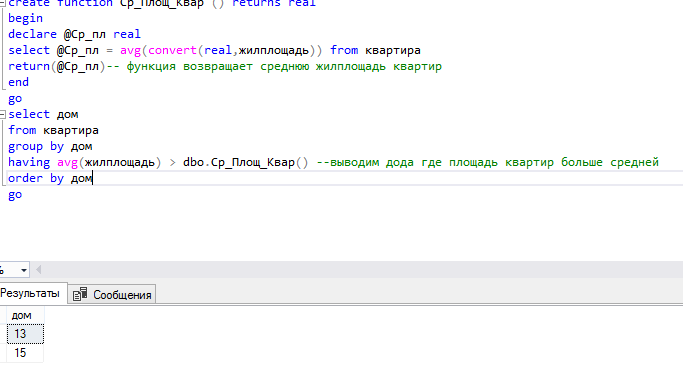


Рис.1.4. дома где площадь квартир больше средней

*формирование статистики во временной таблице (например, для рассматриваемой БД — хранящую данные о количестве квартир каждом доме)*

*Запрос: Создать временную таблицу, хранящую данные о количестве квартир каждом доме*

Запрос выполняется с помощью следующего скрипта:

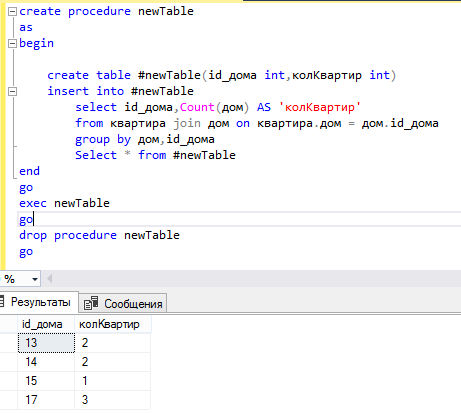


Рис.1.5.Формирование статистики во временной таблице «#newTable».

***Использование ПЗ и ХП (п.6.1):***

1. *Создадим таблицу со списком домов без продающихся квартир*

use bd\_estate\_agency\_tr

go

create procedure Дома\_Без\_прод\_квартир

as

if exists (select \* from дом left join квартира on дом.id\_дома = квартира.дом where квартира.дом is null)

BEGIN

CREATE TABLE домБезПродКвар

(

id\_дома int,

номер\_дома varchar(10),

улица int,

район int,

серия int

)

INSERT into домБезПродКвар( id\_дома,номер\_дома,улица,район,серия)

select дом.\*

from дом left join квартира on дом.id\_дома = квартира.дом

where квартира.дом is null

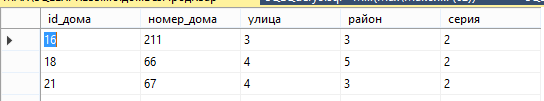
delete from домБезПродКвар

where серия = 1

END

else print 'домов без продающихся квартир не обнаружено'

go



1. Рис.1.6. создание таблицы *домов без продающихся квартир*
2. *Oператор цикла while:*

*Запрос: Введем цикл увеличение процент с продаж пока есть агенты с процентом меньше 7 и максимальным процентом меньше 10*

WHILE ( SELECT AVG(процент) FROM агент\_по\_продаже) < 7

BEGIN

UPDATE агент\_по\_продаже

SET процент = процент +1;

SELECT MAX ( процент) FROM агент\_по\_продаже

IF ( SELECT MAX (процент) FROM агент\_по\_продаже) > 10

BREAK;

END

Результат выполнения данного скрипта:

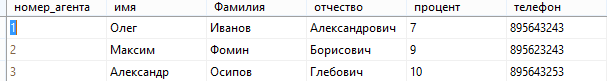


Рис.1.8.Цикл увеличения процента с продаж

**Выводы:**

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

***ПРИЛОЖЕНИЕ А.***

--Создаем процедуру

CREATE PROCEDURE AddFlat

(

--Входящие параметры

@id\_дома int,

@номер\_квартиры int,

@кол\_комнат int,

@цена\_кв\_м money,

@жилплощадь int

)

AS

SET IDENTITY\_INSERT дом ON;

IF not exists (SELECT \* FROM дом WHERE id\_дома=@id\_дома)

BEGIN

SELECT \* FROM дом

INSERT INTO дом (id\_дома, номер\_дома, улица, район, серия) VALUES (@id\_дома, '777', 1, 5,1)

SELECT \* FROM дом

END

else

begin

--Добавляем новую запись

SELECT \* FROM квартира

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

VALUES (@номер\_квартиры,@кол\_комнат, @цена\_кв\_м,@жилплощадь,@id\_дома)

--Возвращаем данные

SELECT \* FROM квартира

end

GO

EXECUTE AddFlat @id\_дома = 2, @номер\_квартиры = 4, @кол\_комнат= 4, @цена\_кв\_м = 40000,@жилплощадь = 77

drop procedure AddFlat

go

--Создаем процедуру

CREATE PROCEDURE CasDel

(

--Входящие параметры

@id\_дом int

)

AS

SET IDENTITY\_INSERT дом ON;

SELECT \* FROM дом join квартира on дом.id\_дома = квартира.дом

DELETE FROM квартира

WHERE дом= @id\_дом

DELETE FROM дом

WHERE id\_дома= @id\_дом

SELECT \* FROM дом join квартира on дом.id\_дома = квартира.дом

GO

EXECUTE CasDel @id\_дом = 2

drop procedure CasDel

go

use bd\_estate\_agency\_tr

go

--Создаем процедуру

CREATE PROCEDURE del

(

--Входящие параметры

@id\_дома int,

@кол\_комнат int

)

AS

SET IDENTITY\_INSERT квартира ON;

SELECT \* FROM квартира join дом on квартира.дом = дом.id\_дома

DELETE FROM квартира

WHERE дом=@id\_дома and кол\_комнат>@кол\_комнат

IF not exists (SELECT \* FROM квартира WHERE дом=@id\_дома)

BEGIN

DELETE FROM дом

WHERE id\_дома=@id\_дома

END

SELECT \* FROM квартира join дом on дом = дом.id\_дома

GO

EXECUTE del @id\_дома = 2,@кол\_комнат = 3

drop procedure del

go

create procedure newTable

as

begin

create table #newTable(id\_дома int,колКвартир int)

insert into #newTable

select id\_дома,Count(дом) AS 'колКвартир'

from квартира join дом on квартира.дом = дом.id\_дома

group by дом,id\_дома

Select \* from #newTable

end

go

exec newTable

go

drop procedure newTable

go

use bd\_estate\_agency\_tr

go

create procedure Дома\_Без\_прод\_квартир

as

if exists (select \* from дом left join квартира on дом.id\_дома = квартира.дом where квартира.дом is null)

BEGIN

CREATE TABLE домБезПродКвар

(

id\_дома int,

номер\_дома varchar(10),

улица int,

район int,

серия int

)

INSERT into домБезПродКвар( id\_дома,номер\_дома,улица,район,серия)

select дом.\*

from дом left join квартира on дом.id\_дома = квартира.дом

where квартира.дом is null

delete from домБезПродКвар

where серия = 1

END

else print 'домов без продающихся квартир не обнаружено'

go

create function Ср\_Площ\_Квар () returns real

begin

declare @Ср\_пл real

select @Ср\_пл = avg(convert(real,жилплощадь)) from квартира

return(@Ср\_пл)

end

go

select дом

from квартира

group by дом

having avg(жилплощадь) > dbo.Ср\_Площ\_Квар()

order by дом

go

WHILE ( SELECT AVG(процент) FROM агент\_по\_продаже) < 7

BEGIN

UPDATE агент\_по\_продаже

SET процент = процент +1;

SELECT MAX ( процент) FROM агент\_по\_продаже

IF ( SELECT MAX (процент) FROM агент\_по\_продаже) > 10

BREAK;

END