ГУАП

КАФЕДРА № 53

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

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

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 3641 |  |  |  | М.Б.Фомин |
|  |  |  | 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. 8.2, создать в БД Курсоры, реализующие:

— реализовать ПЗ или ХП или Т, использующие курсоры (ISO Syntax);;

— продемонстрировать различия в работе статических и динамических курсоров, а также курсоров ***keyset*** (Transact-SQL Extended Syntax);

—используя директивы ***update*** и ***delete*** с условиями ***where current of <имя курсора>*** самостоятельно реализовать примеры изменения и удаления записей посредством курсора

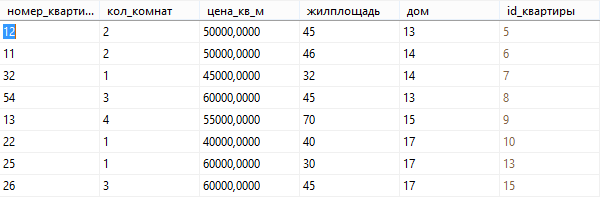
1. **Курсоры: (С использованием ХП, Триггеров, ПЗ)**

Создадим курсор, позволяющий реализовать:

1. *Произведем подсчет количества квартир для каждого из домов, которые есть в БД и выведем результат в отдельную таблицу «Колличество \_квартир\_у \_домов»*

Для данного курсора нам понадобится 1 таблица «квартира» (табл.1).

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



Запрос выполняется с помощью следующего скрипта:

use bd\_estate\_agency\_tr

go

drop proc Вычисл\_Колличество\_квартир\_у\_дома

go

create proc Вычисл\_Колличество\_квартир\_у\_дома

as

delete Количество\_квартир

insert into Количество\_квартир (id\_дома, кол\_кв\_у\_дома1 , кол\_кв\_у\_дома2 ,кол\_кв\_у\_дома3 ,кол\_кв\_у\_дома4 , кол\_кв\_у\_дома5)

select distinct квартира.дом, 0, 0, 0,0,0

from квартира

declare @дом smallint, @КоличествоКвар int

declare Вычисл\_12345 cursor for

select дом, count(дом) 'Количество\_кв\_в\_домах'

from квартира

group by дом

open Вычисл\_12345

fetch next from Вычисл\_12345 into @дом, @КоличествоКвар

while @@fetch\_status = 0 begin

if @дом = 13

update Количество\_квартир

set кол\_кв\_у\_дома1 = @КоличествоКвар

where id\_дома = @дом

if @дом = 14

update Количество\_квартир

set кол\_кв\_у\_дома2 = @КоличествоКвар

where id\_дома = @дом

if @дом = 15

update Количество\_квартир

set кол\_кв\_у\_дома3 = @КоличествоКвар

where id\_дома = @дом

if @дом = 17

update Количество\_квартир

set кол\_кв\_у\_дома4 = @КоличествоКвар

where id\_дома = @дом

fetch next from Вычисл\_12345 into @дом, @КоличествоКвар

end

close Вычисл\_12345

deallocate Вычисл\_12345

go

exec Вычисл\_Колличество\_квартир\_у\_дома

select \* from Количество\_квартир

go

Результат выполнения данного скрипта:



Рис.1.1. Результат подсчета в табл.

1. ***Формирование статистики***
2. *Использование курсора совместно с триггером для вставки, удаления,обновления значений и ведения статистики, для таблицы «квартира». (Если вставляется, удаляется квартира, автоматически изменяются значения таблицы «*Количество\_комнат*, которая хранит количество комнат для каждой квартиры).*

Запрос выполняется с помощью следующего скрипта:

drop trigger Изм\_Квартира\_8лаб

go

create trigger Изм\_Квартира\_8лаб

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

declare Вычисл\_12345\_i cursor for

select кол\_комнат

from inserted

open Вычисл\_12345\_i

fetch next from Вычисл\_12345\_i into @колКомнат

while @@fetch\_status = 0 begin

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

fetch next from Вычисл\_12345\_i into @колКомнат

end

close Вычисл\_12345\_i

deallocate Вычисл\_12345\_i

--delete

declare Вычисл\_12345\_d cursor for

select кол\_комнат

from deleted

open Вычисл\_12345\_d

fetch next from Вычисл\_12345\_d into @колКомнат

while @@fetch\_status = 0 begin

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

fetch next from Вычисл\_12345\_d into @колКомнат

end

close Вычисл\_12345\_d

deallocate Вычисл\_12345\_d

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

Результат выполнения данного скрипта:

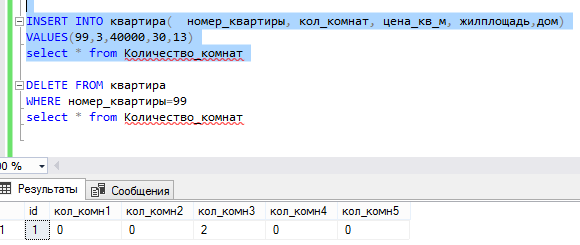


Рис.2.2.Вывод значений из табл количество комнат

1. **Изменение и удаление записей посредством курсора.**

***Динамический курсор***(позволяет изменять данные в курсоре) поддерживает данные в "живом" состоянии, но это требует затрат сетевых и программных ресурсов. При использовании *динамических курсоров* не создается полная копия исходных данных, а выполняется динамическая выборка из исходных таблиц только при обращении пользователя к тем или иным данным. На время выборки сервер блокирует строки, а все *изменения*, вносимые пользователем в полный результирующий набор *курсора*, будут видны в *курсоре*

1. *Динамичсекий курсор: Вставка значения в таблицу, если соблюдаются условия- используется динамический курсор.(Произведем вставку значения в табл. «агент по продажам», если если процент с продаж меньше 5 ,то ему присваивается статус стажёр*

Запрос выполняется с помощью следующего скрипта:

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real,

@телефон INT,

@статус VARCHAR(30)

DECLARE dynamic\_reg\_cursor CURSOR

DYNAMIC FOR

SELECT \*

FROM агент\_по\_продаже

OPEN dynamic\_reg\_cursor

IF @@CURSOR\_ROWS < 0

BEGIN

FETCH NEXT FROM dynamic\_reg\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

WHILE @@FETCH\_STATUS = 0

BEGIN

IF @процент <= 5

UPDATE [агент\_по\_продаже]

SET [статус] = 'стажёр'

WHERE CURRENT OF dynamic\_reg\_cursor

FETCH NEXT FROM dynamic\_reg\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

END

END

CLOSE dynamic\_reg\_cursor

DEALLOCATE dynamic\_reg\_cursor

SET NOCOUNT OFF

GO

select \* from агент\_по\_продаже

Результат выполнения данного скрипта:

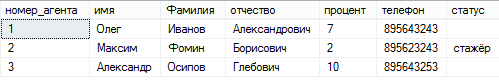


Рис.3.1. Изменения столбца «статус»

1. *Динамичсекий курсор(Удаление): Удаление значения из таблицы, если соблюдаются условия- используется динамический курсор.(Произведем удаление значения из табл. «агент по продаже», если количество работ режиссера равно 1 или в столбце Del стоит 1- результат работы триггера*

Запрос выполняется с помощью следующего скрипта:

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real,

@телефон INT,

@статус VARCHAR(30)

DECLARE dynamic\_reg\_cursor2 CURSOR

DYNAMIC FOR

SELECT \*

FROM агент\_по\_продаже

OPEN dynamic\_reg\_cursor2

IF @@CURSOR\_ROWS < 0

BEGIN

FETCH NEXT FROM dynamic\_reg\_cursor2

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

WHILE @@FETCH\_STATUS = 0

BEGIN

IF @процент <= 1

Delete from [агент\_по\_продаже]

WHERE CURRENT OF dynamic\_reg\_cursor2

FETCH NEXT FROM dynamic\_reg\_cursor2

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

END

END

CLOSE dynamic\_reg\_cursor2

DEALLOCATE dynamic\_reg\_cursor2

SET NOCOUNT OFF

GO

select \* from агент\_по\_продаже

Результат выполнения данного скрипта:

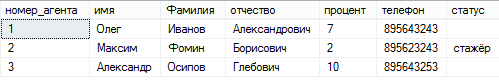


Рис.3.2. результат удаления

1. **Статический курсор:**

*Статический курсор(только для чтения)* делает как бы моментальный снимок данных, задаваемых оператором SELECT, и хранит их в базе данных tempdb. Он "не чувствует" изменений в структуре или в значениях данных, а поскольку любые модификации будут отражены только в копии, этот курсор всегда открывается в режиме "только чтение". Статические курсоры, однако, могут быть объявлены как последовательные или как прокручиваемые

Команда FETCH извлекает указанную строку из множества строк курсора

При указании SCROLL создается прокручиваемый курсор; обращаться к данным можно в любом порядке и в любом направлении.

При указании STATIC создается статический курсор.

При указании KEYSET создается ключевой курсор.

При указании DYNAMIC создается динамический курсор.

1. *Использование статического курсора для вывода значений, хранящихся в таблице «агент по продаже»*

Запрос выполняется с помощью следующего скрипта:

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real

DECLARE static\_cursor CURSOR

STATIC FOR

SELECT [номер\_агента]

,[имя]

,[Фамилия]

,[отчество]

,[процент]

FROM агент\_по\_продаже

OPEN static\_cursor

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM static\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT 'ID = '+ CONVERT(VARCHAR(10), @номер\_агента)+', Имя = '+ @имя

+', Фамилия = '+ @Фамилия +', Отчество = '+ @отчество

+ ', процент = ' + CONVERT(VARCHAR(10),@процент)

FETCH NEXT FROM static\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

END

END

CLOSE static\_cursor

DEALLOCATE static\_cursor

SET NOCOUNT OFF

Результат выполнения данного скрипта:



Рис.4.1.Вывод значений таблицы «агент по продаже»

1. **Ключевые курсоры**

*Ключевой курсор* копирует в базу tempdb только те столбцы, которые уникально идентифицируют каждую строку. Чтобы иметь возможность объявить ключевой курсор, каждая таблица, входящая в определение оператора SELECT, должна иметь уникальный индекс, который задает копируемый набор – ключ.

Ключевые курсоры могут быть как модифицируемыми, так и иметь режим "только для чтения". Они также могут быть прокручиваемыми или последовательными. Курсор который пока открыт нельзя вставить данные, но удалять и изменять можно

1. *Использование курсора keyset для вывода значений, хранящихся в таблице «агент по продаже», организация задержки, а также на количество вывода информации.*

Запрос выполняется с помощью следующего скрипта:

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real

DECLARE keysets CURSOR

KEYSET FOR

SELECT [номер\_агента]

,[имя]

,[Фамилия]

,[отчество]

,[процент]

FROM агент\_по\_продаже

OPEN keysets

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM keysets

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

WHILE @@FETCH\_STATUS = 0

BEGIN

waitfor delay '00:00:03'

PRINT 'ID = '+ CONVERT(VARCHAR(10), @номер\_агента)+', Имя = '+ @имя

+', Фамилия = '+ @Фамилия +', Отчество = '+ @отчество

+ ', процент = ' + CONVERT(VARCHAR(10),@процент)

FETCH NEXT FROM keysets

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

PRINT @@FETCH\_STATUS

END

END

CLOSE keysets

DEALLOCATE keysets

SET NOCOUNT OFF

Результат выполнения данного скрипта:

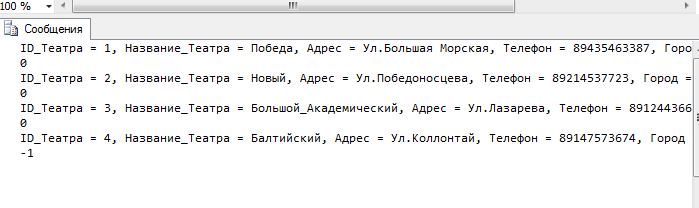


Рис.5.1.Вывод значений табл. «агент по продаже»

**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 курсоры требуемые в задании.

***ПРИЛОЖЕНИЕ А.*** Скрипт БД:

use bd\_estate\_agency\_tr

go

drop proc Вычисл\_Колличество\_квартир\_у\_дома

go

create proc Вычисл\_Колличество\_квартир\_у\_дома

as

delete Количество\_квартир

insert into Количество\_квартир (id\_дома, кол\_кв\_у\_дома1 , кол\_кв\_у\_дома2 ,кол\_кв\_у\_дома3 ,кол\_кв\_у\_дома4 , кол\_кв\_у\_дома5)

select distinct квартира.дом, 0, 0, 0,0,0

from квартира

declare @дом smallint, @КоличествоКвар int

declare Вычисл\_12345 cursor for

select дом, count(дом) 'Количество\_кв\_в\_домах'

from квартира

group by дом

open Вычисл\_12345

fetch next from Вычисл\_12345 into @дом, @КоличествоКвар

while @@fetch\_status = 0 begin

if @дом = 13

update Количество\_квартир

set кол\_кв\_у\_дома1 = @КоличествоКвар

where id\_дома = @дом

if @дом = 14

update Количество\_квартир

set кол\_кв\_у\_дома2 = @КоличествоКвар

where id\_дома = @дом

if @дом = 15

update Количество\_квартир

set кол\_кв\_у\_дома3 = @КоличествоКвар

where id\_дома = @дом

if @дом = 17

update Количество\_квартир

set кол\_кв\_у\_дома4 = @КоличествоКвар

where id\_дома = @дом

fetch next from Вычисл\_12345 into @дом, @КоличествоКвар

end

close Вычисл\_12345

deallocate Вычисл\_12345

go

exec Вычисл\_Колличество\_квартир\_у\_дома

select \* from Количество\_квартир

go

drop trigger Изм\_Квартира\_8лаб

go

create trigger Изм\_Квартира\_8лаб

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

declare Вычисл\_12345\_i cursor for

select кол\_комнат

from inserted

open Вычисл\_12345\_i

fetch next from Вычисл\_12345\_i into @колКомнат

while @@fetch\_status = 0 begin

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

fetch next from Вычисл\_12345\_i into @колКомнат

end

close Вычисл\_12345\_i

deallocate Вычисл\_12345\_i

--delete

declare Вычисл\_12345\_d cursor for

select кол\_комнат

from deleted

open Вычисл\_12345\_d

fetch next from Вычисл\_12345\_d into @колКомнат

while @@fetch\_status = 0 begin

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

fetch next from Вычисл\_12345\_d into @колКомнат

end

close Вычисл\_12345\_d

deallocate Вычисл\_12345\_d

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

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real,

@телефон INT,

@статус VARCHAR(30)

DECLARE dynamic\_reg\_cursor CURSOR

DYNAMIC FOR

SELECT \*

FROM агент\_по\_продаже

OPEN dynamic\_reg\_cursor

IF @@CURSOR\_ROWS < 0

BEGIN

FETCH NEXT FROM dynamic\_reg\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

WHILE @@FETCH\_STATUS = 0

BEGIN

IF @процент <= 5

UPDATE [агент\_по\_продаже]

SET [статус] = 'стажёр'

WHERE CURRENT OF dynamic\_reg\_cursor

FETCH NEXT FROM dynamic\_reg\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

END

END

CLOSE dynamic\_reg\_cursor

DEALLOCATE dynamic\_reg\_cursor

SET NOCOUNT OFF

GO

select \* from агент\_по\_продаже

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real,

@телефон INT,

@статус VARCHAR(30)

DECLARE dynamic\_reg\_cursor2 CURSOR

DYNAMIC FOR

SELECT \*

FROM агент\_по\_продаже

OPEN dynamic\_reg\_cursor2

IF @@CURSOR\_ROWS < 0

BEGIN

FETCH NEXT FROM dynamic\_reg\_cursor2

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

WHILE @@FETCH\_STATUS = 0

BEGIN

IF @процент <= 1

Delete from [агент\_по\_продаже]

WHERE CURRENT OF dynamic\_reg\_cursor2

FETCH NEXT FROM dynamic\_reg\_cursor2

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент, @телефон, @статус

END

END

CLOSE dynamic\_reg\_cursor2

DEALLOCATE dynamic\_reg\_cursor2

SET NOCOUNT OFF

GO

select \* from агент\_по\_продаже

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real

DECLARE static\_cursor CURSOR

STATIC FOR

SELECT [номер\_агента]

,[имя]

,[Фамилия]

,[отчество]

,[процент]

FROM агент\_по\_продаже

OPEN static\_cursor

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM static\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

WHILE @@FETCH\_STATUS = 0

BEGIN

PRINT 'ID = '+ CONVERT(VARCHAR(10), @номер\_агента)+', Имя = '+ @имя

+', Фамилия = '+ @Фамилия +', Отчество = '+ @отчество

+ ', процент = ' + CONVERT(VARCHAR(10),@процент)

FETCH NEXT FROM static\_cursor

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

END

END

CLOSE static\_cursor

DEALLOCATE static\_cursor

SET NOCOUNT OFF

SET NOCOUNT ON

DECLARE

@номер\_агента INT,

@имя VARCHAR(30),

@Фамилия VARCHAR(30),

@отчество VARCHAR(30),

@процент real

DECLARE keysets CURSOR

KEYSET FOR

SELECT [номер\_агента]

,[имя]

,[Фамилия]

,[отчество]

,[процент]

FROM агент\_по\_продаже

OPEN keysets

IF @@CURSOR\_ROWS > 0

BEGIN

FETCH NEXT FROM keysets

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

WHILE @@FETCH\_STATUS = 0

BEGIN

waitfor delay '00:00:03'

PRINT 'ID = '+ CONVERT(VARCHAR(10), @номер\_агента)+', Имя = '+ @имя

+', Фамилия = '+ @Фамилия +', Отчество = '+ @отчество

+ ', процент = ' + CONVERT(VARCHAR(10),@процент)

FETCH NEXT FROM keysets

INTO @номер\_агента, @имя, @Фамилия,

@отчество, @процент

PRINT @@FETCH\_STATUS

END

END

CLOSE keysets

DEALLOCATE keysets

SET NOCOUNT OFF