Министерство образования Республики Беларусь

Учреждение образования

Белорусский государственный университет  
информатики и радиоэлектроники

Кафедра экономической информатики

ОТЧЁТ  
по лабораторной работе № 6  
на тему «Манипулирование данными с помощью команд языка Transact-SQL»

Выполнил:

студент группы 710102

Цвирко А.Ю.

Проверил:

Бутов А.А.

Минск 2019

## Задание к работе

**1.** Создайте хранимую процедуру **pr\_КолебанияСпросаТоваров**, которая решает рассмотренную выше (см. раздел I) задачу определения наименования товара, по которому был наибольший или наименьший спрос за последние N дней. Эта процедура должна иметь два входных параметра (**@Интервал, @ТипРезультата)** и два выходных параметра (**@Имя**, **@Итог**). Если значение входного параметра **@ТипРезультата** равно 1, находится товар наибольшего спроса. Если же значение параметра равно 2 – находится товар наименьшего спроса.

**2.** Создайте хранимую процедуру **pr\_КлиентПоставщик\_СтранаИнтервал**, которая подсчитывает, сколько различных клиентов и различных поставщиков из указанной страны фигурирует в таблице **Заказ**, причем анализируются только те заказы, в которых значение поля **Дата заказа** попадает в указанный интервал дат. Эта процедура должна иметь три входных параметра (**@Страна**, **@НачалоИнтервала**, **@КонецИнтервала**) и два выходных параметра (**@ЧислоКлиентов**, **@ЧислоПоставщиков**). Если же значение параметра **@Страна** не будет указано (т.е. будет равно NULL), то подсчет клиентов и поставщиков должен вестись независимо от их национальной принадлежности.

**3.** Создайте хранимую процедуру **pr\_Товар\_СтранаВалютаИнтервал**, которая подсчитывает, сколько различных товаров в конкретной валюте было заказано клиентами из указанной страны, причем анализируются только те заказы, в которых значение поля **Дата заказа** попадает в заданный интервал дат. Эта процедура должна иметь четыре входных параметра (**@Страна**, **@Валюта**, **@НачалоИнтервала**, **@КонецИнтервала**) и один выходной параметр (**@ЧислоТоваров**). При этом расширьте возможности процедуры следующим образом:

- если значение параметра **@Страна** не будет указано (т.е. будет равно NULL), то подсчет товаров должен вестись независимо от национальной принадлежности клиента;

- если значение параметра **@Валюта** не будет указано (т.е. будет равно NULL), то подсчет товаров должен вестись применительно к национальной валюте (код валюты – BYR).

**4.** Создайте пользовательскую функцию **fn\_getЧислоДней\_вМесяце** типа Scalar, которая для конкретной даты возвращает число дней в месяце, который определяется этой датой (високосность года не учитывается). Эта функция должна иметь один входной параметр (**@Дата**).

**5.** Создайте пользовательскую функцию **fn\_getФИО\_вФормате** типа Scalar, которая на основе текстовой строки, содержащей фамилию, имя и отчество, формирует текстовую строку в одном из следующих форматов:

1) исходная строка переводится в верхний регистр;

2) исходная строка переводится в нижний регистр;

3) на верхнем регистре должны быть только первые буквы слов;

4) выводится только фамилия, а имя и отчество заменяются их первыми буквами с точкой.

Эта функция должна иметь два входных параметра (**@ФИО, @Формат**).

*Усложненный вариант*. Расширьте возможности функции таким образом, чтобы была допустима исходная строка (задаваемая параметром @ФИО), содержащая не один, а несколько пробелов между фамилией и именем или между именем и отчеством, а также допускающая наличие лидирующих пробелов перед фамилией.

**6.** Создайте пользовательскую функцию **fn\_getGroup\_НаименованиеВалюта** типа Inline Table-valued, которая возвращает таблицу со следующими столбцами:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Наименование*  *товара* | *Имя*  *валюты* | *Заказанное*  *кол-во* | *Стоимость в*  *валюте* | *Стоимость в*  *национальной валюте* |
| **. . .** | **. . .** | **. . .** | **. . .** | **. . .** |

Эта таблица должна отражать результат группировки данных по полям **Наименование** и **ИмяВалюты**. Для каждой такой группы подсчитывается итоговое количество заказанного товара и итоговая стоимость в валюте и в национальной валюте.

Пользовательская функция **fn\_getGroup\_НаименованиеВалюта** должна иметь два входных параметра (**@НачалоИнтервала**, **@КонецИнтервала**), поэтому при формировании результирующей таблицы необходимо учитывать только те строки из таблицы **Заказ**, в которых значение поля **Дата заказа** попадает в указанный параметрами интервал дат.

**7.** Создайте пользовательскую функцию **fn\_getTable\_СтоимостьНВ** типа Multi-statement Table-valued, которая возвращает таблицу со следующими столбцами:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *Номер* | *Дата заказа* | *Имя клиента* | *Наименование товара* | *Количество* | *Цена в НВ* | *Стоимость в НВ* |
| **. . .** | **. . .** | **. . .** | **. . .** | **. . .** | **. . .** | **. . .** |

Эта таблица строится в три этапа. Сначала создается таблица со столбцами, показанными выше, где столбец **Номер** является автоинкрементным первичным ключом, столбцы **Цена в НВ** (цена в национальной валюте) и **Стоимость в НВ** (стоимость в национальной валюте) являются вычисляемыми. Число строк этой таблицы будет равно числу строк в таблице **Заказ**.

На втором этапе подсчитывается средняя стоимость в национальной валюте всех заказанных товаров.

На третьем этапе из полученной таблицы удаляются все те строки, в которых значение столбца **Стоимость в НВ** будет меньше, чем подсчитанная на втором этапе средняя стоимость в национальной валюте. В результате будет получена таблица, которую и должна возвращать данная пользовательская функция. Эта функция не имеет ни одного входного параметра.

**Выполнение:**

use Склад\_214TSQL

go

-- 1 процедура

create procedure pr\_КолебанияСпросаТоваров

@Интервал int,

@ТипРезультата int,

@Имя varchar(50) output,

@Итог int output

as

if @ТипРезультата = 1

begin

select @Имя = Т.Наименование, @Итог = SUM(Количество)

from Заказ З

inner join Товар Т on З.КодТовара = Т.КодТовара

where ДатаЗаказа BETWEEN GetDate() - @Интервал AND GetDate()

group by Т.Наименование

order by SUM(Количество)

end

else

if @ТипРезультата = 2

begin

select @Имя = Т.Наименование, @Итог = SUM(Количество)

from Заказ З

inner join Товар Т on З.КодТовара = Т.КодТовара

where ДатаЗаказа BETWEEN GetDate() - @Интервал AND GetDate()

group by Т.Наименование

order by SUM(Количество) desc

end

go

declare @Имя varchar(50), @Итог int

exec pr\_КолебанияСпросаТоваров 20, 1, @Имя output, @Итог output

select @Имя as [Наименование товара], @Итог as [Спрос]

exec pr\_КолебанияСпросаТоваров 40, 2, @Имя output, @Итог output

select @Имя as [Наименование товара], @Итог as [Спрос]

go

-- 2 процедура

create procedure pr\_КлиентПоставщик\_СтранаИнтервал

@Страна varchar(20),

@НачалоИнтервала datetime,

@КонецИнтервала datetime,

@ЧислоКлиентов int output,

@ЧислоПоставщиков int output

as

if @Страна is not null

begin

select @ЧислоКлиентов = count(distinct З.КодКлиента)

from Заказ З

inner join Клиент К on З.КодКлиента = К.КодКлиента

inner join Регион Р on К.КодРегиона = Р.КодРегиона

where З.ДатаЗаказа <= @КонецИнтервала and З.ДатаЗаказа >= @НачалоИнтервала and @Страна = Р.Страна

select @ЧислоПоставщиков = count(distinct З.КодПоставщика)

from Заказ З

inner join Поставщик П on З.КодПоставщика = П.КодПоставщика

inner join Регион Р on П.КодРегиона = Р.КодРегиона

where ДатаЗаказа <= @КонецИнтервала and ДатаЗаказа >= @НачалоИнтервала and @Страна = Р.Страна

end

else

begin

select @ЧислоКлиентов = count(distinct З.КодКлиента)

from Заказ З

inner join Клиент К on З.КодКлиента = К.КодКлиента

inner join Регион Р on К.КодРегиона = Р.КодРегиона

where З.ДатаЗаказа <= @КонецИнтервала and З.ДатаЗаказа >= @НачалоИнтервала

select @ЧислоПоставщиков = count(distinct З.КодПоставщика)

from Заказ З

inner join Поставщик П on З.КодПоставщика = П.КодПоставщика

inner join Регион Р on П.КодРегиона = Р.КодРегиона

where ДатаЗаказа <= @КонецИнтервала and ДатаЗаказа >= @НачалоИнтервала

end

go

declare @ЧислоКлиентов int, @ЧислоПоставщиков int

exec pr\_КлиентПоставщик\_СтранаИнтервал 'Беларусь', "2019-03-10", "2019-05-10", @ЧислоКлиентов output, @ЧислоПоставщиков output

select @ЧислоКлиентов as [Число клинетов], @ЧислоПоставщиков as [Число поставщиков]

exec pr\_КлиентПоставщик\_СтранаИнтервал NULL, "2019-03-10", "2019-05-10", @ЧислоКлиентов output, @ЧислоПоставщиков output

select @ЧислоКлиентов as [Число клинетов], @ЧислоПоставщиков as [Число поставщиков]

go

-- 3 процедура

create procedure pr\_Товар\_СтранаВалютаИнтервал

@Страна varchar(20),

@Валюта char(3),

@НачалоИнтервала datetime,

@КонецИнтервала datetime,

@ЧислоТоваров int output

as

if @Страна is not null

if @Валюта is not null

begin

select @ЧислоТоваров = count(distinct З.КодТовара)

from Заказ З

inner join Клиент К on З.КодКлиента = К.КодКлиента

inner join Регион Р on К.КодРегиона = Р.КодРегиона

inner join Товар Т on З.КодТовара = Т.КодТовара

inner join Валюта В on Т.КодВалюты = В.КодВалюты

where З.ДатаЗаказа <= @КонецИнтервала and З.ДатаЗаказа >= @НачалоИнтервала and @Страна = Р.Страна and @Валюта = В.КодВалюты

end

else

begin

select @ЧислоТоваров = count(distinct З.КодТовара)

from Заказ З

inner join Клиент К on З.КодКлиента = К.КодКлиента

inner join Регион Р on К.КодРегиона = Р.КодРегиона

inner join Товар Т on З.КодТовара = Т.КодТовара

inner join Валюта В on Т.КодВалюты = В.КодВалюты

where З.ДатаЗаказа <= @КонецИнтервала and З.ДатаЗаказа >= @НачалоИнтервала and @Страна = Р.Страна and 'BYR' = В.КодВалюты

end

else

if @Валюта is not null

begin

select @ЧислоТоваров = count(distinct З.КодТовара)

from Заказ З

inner join Клиент К on З.КодКлиента = К.КодКлиента

inner join Регион Р on К.КодРегиона = Р.КодРегиона

inner join Товар Т on З.КодТовара = Т.КодТовара

inner join Валюта В on Т.КодВалюты = В.КодВалюты

where З.ДатаЗаказа <= @КонецИнтервала and З.ДатаЗаказа >= @НачалоИнтервала and @Валюта = В.КодВалюты

end

else

begin

select @ЧислоТоваров = count(distinct З.КодТовара)

from Заказ З

inner join Клиент К on З.КодКлиента = К.КодКлиента

inner join Регион Р on К.КодРегиона = Р.КодРегиона

inner join Товар Т on З.КодТовара = Т.КодТовара

inner join Валюта В on Т.КодВалюты = В.КодВалюты

where З.ДатаЗаказа <= @КонецИнтервала and З.ДатаЗаказа >= @НачалоИнтервала and 'BYR' = В.КодВалюты

end

go

declare @ЧислоТоваров int

exec pr\_Товар\_СтранаВалютаИнтервал 'Россия', 'USD', "2019-03-10", "2019-05-10", @ЧислоТоваров output

select @ЧислоТоваров as [Число Товаров]

exec pr\_Товар\_СтранаВалютаИнтервал NULL, NULL, "2019-03-10", "2019-05-10", @ЧислоТоваров output

select @ЧислоТоваров as [Число Товаров]

go

-- 4 функция типа Scalar

create function fn\_getЧислоДней\_вМесяце

(@Дата datetime)

returns int

begin

declare @ЧислоДней int

select @ЧислоДней = day(EOMONTH(@Дата))

return @ЧислоДней

end

go

SET DATEFORMAT DMY

select dbo.fn\_getЧислоДней\_вМесяце('05-06-2000') as [Число дней в месяце]

select dbo.fn\_getЧислоДней\_вМесяце('07-08-2000') as [Число дней в месяце]

select dbo.fn\_getЧислоДней\_вМесяце('04-09-2021') as [Число дней в месяце]

go

-- 5 функция

create function fn\_getФИО\_вФормате

(@ФИО varchar(100), @Формат int)

returns varchar(100)

begin

declare @ФорматированноеФИО varchar(100)

select @ФорматированноеФИО =

case @Формат

when 1 then lower(@ФИО)

when 2 then upper(@ФИО)

when 3 then upper(substring(parsename(replace(@ФИО, ' ', '.'), 3), 1, 1)) + lower(substring(parsename(replace(@ФИО, ' ', '.'), 3), 2, 100)) + ' '+

upper(substring(parsename(replace(@ФИО, ' ', '.'), 2), 1, 1)) + lower(substring(parsename(replace(@ФИО, ' ', '.'), 2), 2, 100)) + ' '+

upper(substring(parsename(replace(@ФИО, ' ', '.'), 1), 1, 1)) + lower(substring(parsename(replace(@ФИО, ' ', '.'), 1), 2, 100))

when 4 then upper(substring(parsename(replace(@ФИО, ' ', '.'), 3), 1, 1)) + lower(substring(parsename(replace(@ФИО, ' ', '.'), 3), 2, 100)) + ' '+

upper(substring(parsename(replace(@ФИО, ' ', '.'), 2), 1, 1)) + '. '+

upper(substring(parsename(replace(@ФИО, ' ', '.'), 1), 1, 1)) + '.'

end

return @ФорматированноеФИО

end

go

select dbo.fn\_getФИО\_вФормате('иванов иван иванович', 1) as [Форматированное полное имя]

select dbo.fn\_getФИО\_вФормате('иванов иван иванович', 2) as [Форматированное полное имя]

select dbo.fn\_getФИО\_вФормате('иванов иван иванович', 3) as [Форматированное полное имя]

select dbo.fn\_getФИО\_вФормате('иванов иван иванович', 4) as [Форматированное полное имя]

go

-- 6 функция

create function fn\_getGroup\_НаименованиеВалюта

(@НачалоИнтервала datetime, @КонецИнтервала datetime)

returns table

as return

select Т.Наименование as [Наименование товара], В.ИмяВалюты as [Имя валюты], sum(З.Количество) as [Заказанное кол-во], max(Т.Цена) \* sum(З.Количество) as [Стоимость в валюте], max(Т.Цена) \* max(В.КурсВалюты) \* sum(З.Количество) as [Стоимость в национальной валюте]

from Заказ З

inner join Товар Т on З.КодТовара = Т.КодТовара

inner join Валюта В on Т.КодВалюты = В.КодВалюты

where З.ДатаЗаказа <= @КонецИнтервала and З.ДатаЗаказа >= @НачалоИнтервала

group by Т.Наименование, В.ИмяВалюты

go

select \* from fn\_getGroup\_НаименованиеВалюта('2019-03-10', '2019-05-10')

go

-- 7 функция

create function fn\_getTable\_СтоимостьНВ

()

returns @Таблица table(

Номер int primary key identity(1,1),

ДатаЗаказа datetime,

ИмяКлиента varchar(40),

Наименование varchar(50),

Количество int,

ЦенаНВ money,

СтоиомостьНВ money)

begin

insert @Таблица(ДатаЗаказа, ИмяКлиента, Наименование, Количество, ЦенаНВ, СтоиомостьНВ)

select З.ДатаЗаказа, К.ИмяКлиента, Т.Наименование, З.Количество, В.КурсВалюты \* Т.Цена, В.КурсВалюты \* Т.Цена \* З.Количество

from Заказ З

inner join Клиент К on З.КодКлиента = К.КодКлиента

inner join Товар Т on З.КодТовара = Т.КодТовара

inner join Валюта В on Т.КодВалюты = В.КодВалюты

declare @AveragePrice int

select @AveragePrice = AVG(СтоиомостьНВ)

from @Таблица

delete from @Таблица

where СтоиомостьНВ < @AveragePrice

return

end

go

select \* from fn\_getTable\_СтоимостьНВ()

go

