

## #13 Создание пользовательских функций

### Зачем делать эту лабораторную работу?

1. Чтобы научиться сохранять простые и сложные алгоритмы, реализованные на языке SQL как объекты базы данных.
2. Чтобы научиться оптимизировать код при многократном выполнении однотипных алгоритмов.
3. Чтобы понять механизмы работы различных типов функций в зависимости от структуры возвращаемых значений.
4. Научиться различать задачи, в которых для формирования результата необходимо применять запросы модификации данных.

### Что нужно делать?

1. Создать функцию, возвращающую скалярное значение.

С помощью инструкции Create Function создайте первую функцию из задания. Необходимый синтаксис инструкции создания scalar-функции следующий:

```
CREATE [ OR ALTER ] FUNCTION [ schema_name. ] function_name
( [ { @parameter_name [ AS ] [ type_schema_name. ] parameter_data_type [ NULL ]
  [ = default ] [ READONLY ] }
  [ ,...n ]
]
)
RETURNS return_data_type
[ AS ]
BEGIN
    function_body
    RETURN scalar_expression
END
[ ; ]
```

Пример функции, возвращающей возвращающую количество книг, находящихся в данный момент у заданного читателя:

```
Create function f1(@x int)
returns int
as
begin
Declare @k int
Select @k=Count(код_книги) from абонемент where код_читателя=@x and
отметка_o_возврате = 0
return @k
end
```

Вызов функции осуществляется инструкцией:

```
Select dbo.f1(1)
```

2. Создать функцию, возвращающую табличное значение.

С помощью инструкции Create Function создайте вторую функцию из задания. Необходимый синтаксис инструкции создания scalar-функции следующий:

```
CREATE [ OR ALTER ] FUNCTION [ schema_name. ] function_name
( [ { @parameter_name [ AS ] [ type_schema_name. ] parameter_data_type [ NULL ]
  [ = default ] [ READONLY ] }
  [ ,...n ]
]
)
RETURNS TABLE
[ AS ]
RETURN [ ( ] select_stmt [ ) ]
[ ; ]
```

Пример функции, возвращающей список книг, наиболее популярных в абонементе:

```
Create function f2()
returns table
as return (Select книги.код_книги, название
from книги inner join абонемент on
книги.код_книги=абонемент.код_книги
where книги.код_книги in (Select код_книги from
абонемент group by код_книги
Having COUNT(код_читателя)= ( Select max(количество) from
(Select код_книги, count(код_читателя) as количество
from абонемент group by код_книги) z1)))
```

Вызов функции осуществляется инструкцией:

```
Select * from dbo.f2()
```

### 3. Создать функцию, возвращающую табличное значение.

С помощью инструкции Create Function создайте третью функцию из задания.

Необходимый синтаксис инструкции создания scalar-функции следующий:

```
CREATE [ OR ALTER ] FUNCTION [ schema_name. ] function_name
( [ { @parameter_name [ AS ] [ type_schema_name. ] parameter_data_type [ NULL ]
  [ = default ] [ READONLY ] }
  [ ,...n ]
]
)
RETURNS @return_variable TABLE <table_type_definition>
[ AS ]
BEGIN
    function_body
RETURN
END
[ ; ]
```

Пример функции, формирующей ведомость закупок, в которой необходимо приобрести по две книги взамен каждой утерянной:

```
Create function f3()
returns @t table (Код_книги int, Количество int)
as begin
Insert @t Select Код_книги, count(код_книги)*2 from штрафы
where книга_утеряна=1
group by код_книги return
end
```

Вызов функции осуществляется инструкцией:

```
Select * from dbo.f3()
```

## Как узнать, что все выполнено?

Проверьте пункты в этом чек-листе:

- ☐ Разработан код трёх функций согласно условиям заданий
- ☐ Функции сохранены как объекты базы данных
- ☐ Функции возвращают корректные результаты при обращении
- ☐ Код запросов и алгоритмической части функций выполнен оптимальным образом

## Варианты заданий:

### 1 вариант:

1. Создать функцию, возвращающую количество преподавателей по заданному номеру кафедры.
2. Создать функцию, возвращающую самые загруженные аудитории.
3. Создать функцию, формирующую новое расписание учебных занятий, в котором необходимо оставить только одно занятие по каждой дисциплине для группы.

### 2 вариант:

1. Создать функцию, вычисляющую количество групп, которые сдавали зачеты по заданной дисциплине.
2. Создать функцию, возвращающую список студентов, получивших только отличные оценки.
3. Создать функцию, формирующую направления для студентов, не сдавших экзамены, к случайным преподавателям, ведущим эти дисциплины.

### 3 вариант:

1. Создать функцию, возвращающую случайный номер автомобиля, обсуживавшегося в заданном месяце.
2. Создать функцию, возвращающую автомобили, для которых каждый раз выполнялся только один вид работы.
3. Создать функцию, формирующую зарплатную ведомость для, механиков, определяющую выплаты по суммарной стоимости работ, из которых вычитается по 10% за каждый договор, сроком менее 10 дней.

### 4 вариант:

1. Создать функцию, возвращающую суммарное количество упаковок товара в заказе с заданным номером.
2. Создать функцию, возвращающую стоимости выполненных заказов за последний месяц.
3. Создать функцию, формирующую новые стоимости для компонентов: каждый новый заказ увеличивает стоимость на 3%.

### 5 вариант:

1. Создать функцию, возвращающую фамилию случайного арендатора, заключившего договор в заданном месяце.
2. Создать функцию, возвращающую не сданные в аренду объекты недвижимости.

3. Создать функцию, формирующую рейтинги объектов недвижимости: рейтинг представляет собой коэффициент, зависящий от количества сданных объектов с такой же площадью.

6 вариант:

1. Создать функцию, возвращающую количество выполненных перевозок для заданного водителя.
2. Создать функцию, возвращающую список автомобилей, выполнявших заказы на перевозку с наибольшим весом груза.
3. Создать функции, формирующую новый график работы водителей: для каждого заказа на перевозку груза весом более 3,5 тонн добавить еще один автомобиль.

7 вариант:

1. Создать функцию, возвращающую номер аудитории по заданным параметрам: код дисциплины, код экзаменатора, номер группы.
2. Создать функцию, возвращающую список групп, сдающих в течение сессии не больше двух экзаменов.
3. Создать функцию, формирующую новое расписание, в котором экзамены по иностранному языку принимаются по подгруппам.

8 вариант:

1. Создать функцию, возвращающую количество исполнителей по заданному номеру заказа и коду выполняемой работы.
2. Создать функцию, возвращающую стоимости выполненных заказов для заданного исполнителя.
3. Создать функцию, формирующую новые стоимости работ. Для всех работ с одинаковыми наименованиями уменьшить стоимость пропорционально их количеству.

9 вариант:

1. Создать функцию, возвращающую суммарное количество упаковок товара, реализованное заданным продавцом.
2. Создать функцию, возвращающую список покупателей, чаще всего совершающих покупки.
3. Создать функцию, возвращающую количество подкатегорий для каждой категории товара.

10 вариант:

1. Создать функцию, возвращающую суммарное количество требуемого товара по введенному наименованию.
2. Создать функцию, возвращающую список всех категорий и подкатегорий товаров для заданного ответственного сотрудника.
3. Создать функцию, формирующую новый список ответственных за категории. Необходимо перераспределить нагрузку так, чтобы каждый сотрудник отвечал только за одну категорию.

11 вариант:

1. Создать функцию, возвращающую количество препаратов, выписанное заданному пациенту на последнем приеме.
2. Создать функцию, возвращающую количество пациентов, принятых каждым из специалистов в заданный день

3. Создать функцию, формирующую график приема: всем пациентам, посетившим специалистов один раз назначить повторный прием через 2 недели.

12 вариант:

1. Создать функцию, возвращающую название случайной секции, запись в которую осуществлялась в заданном месяце.
2. Создать функцию, возвращающую список тренеров, задействованных в руководстве наибольшего количества секций.
3. Создать функцию, формирующую запись в секции таким образом, чтобы из всех секций, где число учеников превышает 30 перераспределить в одноименные секции.

13 вариант:

1. Создать функцию, возвращающую суммарное количество вакансий в бригадах по заданной специальности.
2. Создать функцию, возвращающую список бригад с количеством специалистов, имеющих высокую квалификацию.
3. Создать функцию, формирующую новый перечень квалификаций. Всем, рабочим, на специальности которых больше всего вакансий, изменить квалификацию на «высшую».

14 вариант:

1. Создать функцию, возвращающую время самой продолжительной работы на объекте для заданной бригады.
2. Создать функцию, возвращающую список объектов строительства, на которых задействована только одна бригада.
3. Создать функцию, формирующую новые сроки работы бригад на объектах. Для всех бригад, содержащих меньше 5 рабочих увеличить сроки строительства в два раза.

15 вариант:

1. Создать функцию, возвращающую количество поставщиков, с которыми работает заданный заказчик.
2. Создать функцию, возвращающую список заказчиков, в накладных на поставку которых все позиции товаров от разных поставщиков.
3. Создать функцию, формирующую новые накладные на поставку. Если возможно, для всех товаров одной накладной задайте одного поставщика.