

## Лабораторная работа №13

**Тема:**

Создание пользовательских функций.

**Цель работы:**

Получить навыки создания scalar, inline и multi-statement-функций.

**Теоретическая часть:**Инструкция создания пользовательской функции

Пользовательские функции - самостоятельные объекты базы данных, представляющие собой набор SQL-инструкций. Функции способны принимать и возвращать значения.

В MS SQL-Server предусмотрены следующие типы функций:

- 1) Scalar – функции возвращают значение базового типа данных. Значение вычисляется в теле функции BEGIN ... END;
- 2) Inline – функции, содержащие инструкцию SELECT и возвращающие значение типа TABLE;
- 3) Multi-statement – функции, возвращающие значение типа TABLE, формирующееся в теле функции запросами модификации данных INSERT, UPDATE и DELETE.

Синтаксис инструкции создания scalar-функции, следующий:

```
<создание_scalar_функции>::=
CREATE FUNCTION имя_функции
([@имя_параметра тип_данных [=default]][,...n]])
RETURNS тип_данных
[WITH {ENCRYPTION | SCHEMABINDING} [,...n]]
[AS]
BEGIN
SQL-инструкции
RETURN выражение
END
```

Синтаксис inline-функции:

```
<создание_inline_функции>::=
CREATE FUNCTION имя_функции
([@имя_параметра тип_данных [=default]][,...n]])
RETURNS TABLE
[WITH {ENCRYPTION | SCHEMABINDING} [,...n] ]
[AS]
RETURN [( ) инструкция_SELECT ( )]
```

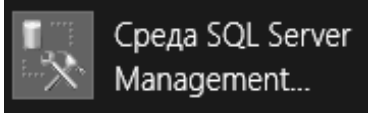
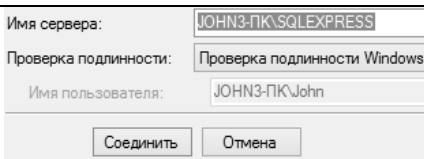
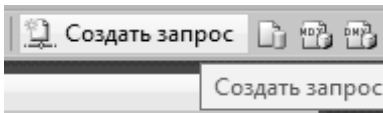
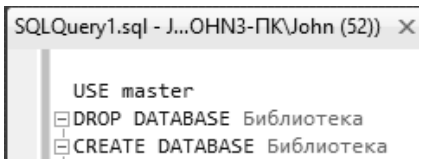
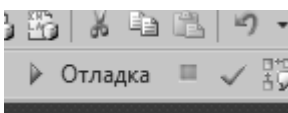
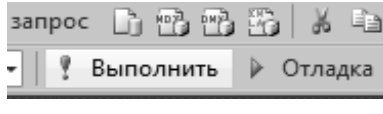
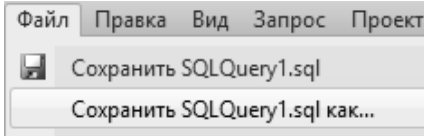
Синтаксис multi-statement-функции:

```
<создание_multi_statement_функции>::=
CREATE FUNCTION имя_функции
([@имя_параметра тип_данных [=default]] [,...n]])
RETURNS @имя_параметра TABLE <определение_таблицы>
[WITH {ENCRYPTION | SCHEMABINDING} [,...n] ]
[AS]
BEGIN
SQL_инструкции
RETURN END
```

**Задание:**

1. Подключиться к SQL-server.
2. Создать новый запрос к серверу.
3. Написать скрипт, реализующий создание трех пользовательских функций.
4. Вызвать функции и проверить корректность их работы.
5. Сохранить скрипт в sql-формате.
6. Показать работу преподавателю.

**Порядок выполнения:**

Запустите SQL Server Management Studio	Пуск → Все приложения → Среда SQL Management Studio	
Подключитесь к серверу	Выберите локальный сервер, установите проверку подлинности Windows и нажмите кнопку Соединить	
Создайте новый запрос	На панели Стандартная нажмите кнопку Создать запрос	
Напишите код скрипта	Напишите код скрипта согласно заданию, приведенному в вашем варианте	
Используйте отладчик	Для исправления семантических ошибок запустите скрипт в режиме отладки	
Выполните скрипт	На панели Стандартная нажмите кнопку выполнить, если необходимо произведите отладку скрипта	
Вызовите функции	Последовательно вызовите функции инструкцией Select, присваивая различные значения их параметрам	<pre>Select * from dbo.f1() Select код_товара from dbo.f2('12.01.14')</pre>
Сохраните скрипт в формате .sql	Выполните команду Файл → Сохранить запрос.sql и покажите работу преподавателю	

**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, обслуживавшегося 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 заданным 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.

16 вариант:

1. Создать функцию, возвращающую количество книг, находящихся в данный момент у заданного читателя.
2. Создать функцию, возвращающую список книг, наиболее популярных в абонементе.
3. Создать функцию, формирующую ведомость закупок: необходимо приобрести по две книги взамен каждой утерянной.

**Пример выполнения 16 варианта:**

```
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_о_возврате = 0
return @k
end

Select dbo.f1(1)

go
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)))
```

```
Select * from dbo.f2()  
go  
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  
  
Select * from dbo.f3()
```

**Контрольные вопросы:**

1. Укажите синтаксис инструкции Create Function.
2. Для чего используются встроенные функции?
3. Укажите отличительные особенности scalar, inline и multi-statement-функций.
4. Какие функции возвращают значение типа Table.
5. Каким образом формируется возвращаемое значение в multi-statement-функциях?
6. Перечислите запросы модификации данных.
7. Приведите отличия inline-функций от представлений.
8. Какую роль функции могут выполнять для обеспечения целостности данных?
9. Как передаются и возвращаются значения в функции?
10. Как вызвать функцию с помощью оператора Select?