

# Lap-03

2015136052 박종한

컴퓨터공학부

a1pjh0515@koreatech.ac.kr

## 1 Introduction

### 1.1 Types of User-defined

#### Scalar Function

1. User-defined scalar functions return a single data value of the type defined in the RETURNS clause.
2. For an inline scalar function, there is no function body; the scalar value is the result of a single statement.
3. For a multi-statement scalar function, the function body, defined in a BEGIN...END block, contains a series of Transact-SQL statements that return the single value.

#### Table-Valued Functions

1. User-defined table-valued functions return a table data type.
2. For an inline table-valued function, there is no function body; the table is the result set of a single SELECT statement.

### 1.2 Creating scalar function

1. The CREATE FUNCTION statement allows you to create custom scalar functions that behave like the built-in scalar functions.
2. schema\_name is the name of the schema to which the user-defined function belongs.

3. `function_name` is the name of the user-defined function. Function names must comply with the rules for identifiers and must be unique within the database and to its schema.
4. `@parameter_name` is a parameter in the user-defined function.  
One or more parameters can be declared.  
A function can have a maximum of 2,100 parameters.  
The value of each declared parameter must be supplied by the user when the function is executed, unless a default for the parameter is defined.  
Specify a parameter name by using an at sign (@) as the first character.
5. `[ type_schema_name. ] parameter_data_type` is the parameter data type, and optionally the schema to which it belongs.
6. `[ =default ]` is a default value for the parameter. If a default value is defined, the function can be executed without specifying a value for that parameter.

### 1.3 Table-valued Functions

1. User-defined functions that return a table data type are referred to as table-valued functions.
2. These functions can be powerful alternatives to views.
3. A table-valued user-defined function can be used where table or view expressions are allowed in Transact-SQL queries.
4. While views are limited to a single SELECT statement, user-defined functions can contain additional statements that allow more powerful logic than is possible in views.
5. A table-valued user-defined function can also replace stored procedures that return a single result set.
6. The table returned by a user-defined function can be referenced in the FROM clause of a Transact-SQL statement, but stored procedures that return result sets cannot.
7. Table-valued functions may be divided into Inline Table-Valued Function and Multistatement Table-valued Function

## 2 Task

### 2.1 Task-01

Write scalar functions

1. A function that sums the prime numbers and non-prime numbers between (1 and a given number say (@rng)) and returns their difference2. create tables for Dream-Home database
2. (Pythagorean Triples) A right triangle can have sides whose lengths are all integers. The set of three integer values for the lengths of the sides of a right triangle is called a Pythagorean triple. The lengths of the three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Write a table valued function that stores the Pythagorean triples for side1, side2 and the hypotenuse in a table variable , all between given numbers say (@srange, @erange), then returns the table

#### 해결방법

1. A function that sums the prime numbers and non-prime numbers between (1 and a given number say (@rng)) and returns their difference

정수형 @rng을 매개변수로 받는 prime스칼라함수를 선언 하고 소수의 합을 저장 할 변수 @p\_num과 비소수의 합을 저장 할 변수 @n\_num을 선언한 후 0으로 초기화 한다.

후에 2로 초기화 된 @i 변수를 하나 씩 증가시키며 @i가 @rng값보다 작아 질 때까지 반복을 실시 한다. 반복을 하며 @i값이 소수라고 판단 되면 @p\_num에 @i값을 더해주고 소수가 아니라면 @n\_num에 @i값을 더해준다.

최종적으로 소수의 합과 비소수의 합의 차이를 반환하여 준다.

```

CREATE FUNCTION prime(@rng int)
RETURNS INT
AS
BEGIN
    DECLARE @p_sum INT -- 소수의 합을 저장 할 변수
    DECLARE @n_sum INT -- 비소수의 합을 저장 할 변수
    DECLARE @i INT
    DECLARE @j INT
    SET @p_sum = 0
    SET @n_sum = 0
    SET @i = 2

    WHILE 1 = 1 -- 무한루프
    BEGIN
        SET @j = 2
        WHILE @i % @j != 0 -- 나머지가 0이 아니면 반복
            SET @j = @j + 1

        IF @i = @j -- i와 j가 같으면 소수라고 판단
            SET @p_sum = @p_sum + @i -- 더해줌
        ELSE -- i와 j가 다르면 비소수라고 판단
            SET @n_sum = @n_sum + @i

        IF @i < @rng -- rng보다 작을 때
            SET @i = @i + 1 -- i값 1씩 증가
        ELSE -- rng값보다 크거나 같아지면
            BREAK -- 반복 종료
    END
    RETURN @n_sum - @p_sum + 1 -- 차이값 반환
END
GO

PRINT dbo.prime(10)

```

Fig. 1. prime스칼라함수 작성

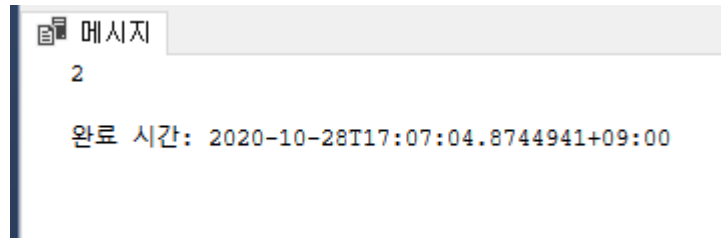
## 결과화면

메시지

21

완료 시간: 2020-10-28T17:04:21.1999218+09:00

Fig. 1-1. @rng = 10



**Fig. 1-2.** @rng = 8

### 해결방법

2. (Pythagorean Triples) A right triangle can have sides whose lengths are all integers. The set of three integer values for the lengths of the sides of a right triangle is called a Pythagorean triple. The lengths of the three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Write a table valued function that stores the Pythagorean triples for side1, side2 and the hypotenuse in a table variable , all between given numbers say (@srange, @erange), then returns the table

사용자가 지정한 범위 내에 있는 정수들 중 Pythagorean Triples를 만족하는 세변의 값을 테이블에 저장한다. 먼저 side1, side2, side3을 정수형 칼럼으로 지정한 뒤 삼중 반복문을 이용하여 각 변수(s1, s2, s3)의 값을 1씩 증가 시키며  $\text{power}(@s1, 2) + \text{power}(@s2, 2) = \text{power}(@s3, 2)$ 을 만족 하는 값을 찾을 때 마다 테이블에 INSERT해 준다.

최종적으로는 테이블을 리턴하고 SELECT문을 이용하여 저장된 3변의 정보를 확인 하였다.

```

CREATE FUNCTION dbo.tripleTable(@srange INT, @erange INT)
RETURNS @triTable table (id INT, side1 INT, side2 INT, side3 INT)
AS
BEGIN
    DECLARE @i INT = 1
    DECLARE @s1 INT = 1, @s2 INT = 1, @s3 INT = 1
    SET @s1 = @srange

    WHILE @s1 <= @erange
    BEGIN
        SET @s2 = @srange
        WHILE @s2 <= @erange
        BEGIN
            SET @s3 = @srange
            WHILE @s3 <= @erange
            BEGIN
                IF power(@s1, 2) + power(@s2, 2) = power(@s3, 2)
                BEGIN
                    INSERT INTO @triTable VALUES(@i, @s1, @s2, @s3)
                    SET @i += 1
                END
                SET @s3 += 1
            END
            SET @s2 += 1
        END
        SET @s1 += 1
    END

    RETURN
END
GO

SELECT *FROM dbo.tripleTable(1, 10)
DROP FUNCTION dbo.tripleTable

```

Fig. 2. TripleTable 테이블 함수 작성

## 결과화면

결과		메시지		
	id	side1	side2	side3
1	1	3	4	5
2	2	4	3	5
3	3	6	8	10
4	4	8	6	10

Fig. 2-1. Range(1, 10)

결과		메시지		
	id	side1	side2	side3
1	1	3	4	5
2	2	4	3	5
3	3	5	12	13
4	4	6	8	10
5	5	8	6	10
6	6	9	12	15
7	7	12	5	13
8	8	12	9	15

Fig. 2-2. Range(1, 15)

결과		메시지		
	id	side1	side2	side3
1	1	5	12	13
2	2	6	8	10
3	3	8	6	10
4	4	9	12	15
5	5	12	5	13
6	6	12	9	15

Fig. 2-3. Range(5, 15)

### 결과 및 분석

Fig. 1-1.에서는 1부터 10까지의 범위를 입력하여 결과값을 얻어냈다.

1부터 10사이의 정수 중 소수는 2, 3, 5, 7이 되겠고 비소수는 1, 4, 6, 8, 9, 10이다. 소수의 합( $2+3+5+7$ )은 17, 비소수의 합( $1+4+6+8+9+10$ )은 38이고 이들의 차이는 21로 결과값이 정상적으로 출력 되는 것을 확인 할 수 있다.

Fig. 2-1.에서는 1과 10사이의 정수 중 Pythagorean Triples를 만족하는 세변을 출력하였다. 각변의 값이 3, 4, 5일 때  $3^2 + 4^2 = 5^2$ 를 만족하므로 Pythagorean Triples 할 수 있겠고 6, 8, 10의 경우에도 이를 만족하므로 결과값이 정상적으로 출력 되었다.

## 느낀 점

t-sql로 스칼라함수와 테이블함수를 작성 해보았다. 이를 해결하며 스칼라함수와 테이블함수의 차이점을 알 수 있었고 t-sql의 기본적인 문법을 학습 할 수 있었다. 또한 함수를 이용하여 테이블에 접근해 원하는 트랜잭션을 수행하는 법을 배울 수 있어 뜻 깊었다.

## 2.2 Task-02

Write user-defined functions such as:

1. A function that returns the list of tables in your database.
2. A function that returns all Primary-keys and Foreign-Keys
3. A function that returns all user-defined triggers
4. A function that returns row-count for all user-defined tables in the database
5. A function that returns the list of all user defined functions.
6. A function that takes a table name as input and returns the column names with their types
7. A function that takes a function name as input and returns input parameters for that function with their types

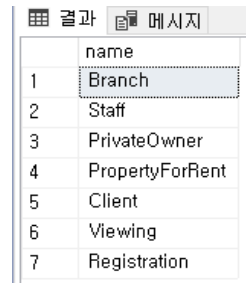
## 해결방법

sys.objects 등 다양한 테이블들을 이용하여 조건에 맞는 함수들을 구현하였다. 최종적으로 select문을 이용하여 결과를 출력한다.

## 결과

1. A function that returns the list of tables in your database.

```
create function dbo.getTableList()
returns table
as
return select name
from sys.objects
where type = 'U'
go
```



	name
1	Branch
2	Staff
3	PrivateOwner
4	PropertyForRent
5	Client
6	Viewing
7	Registration

Fig. 3-1



2. A function that returns all Primary-keys and Foreign-Keys

```
create function dbo.getPK_FK()
returns table
as
return select name
from sys.objects
where type = 'f' or type = 'pk'
go
```

	name		
1	branch_PK	9	client_PK
2	staff_PK	10	viewing_PK
3	Staff_Branch_FK	11	Viewing_Propty_FK
4	privateOwner_PK	12	Viewing_Client_FK
5	propertyForRent_PK	13	registration_PK
6	Property_Owner_FK	14	Regist_Client_FK
7	Property_Staff_FK	15	Regist_Branch_FK
8	Property_Branch_FK	16	Regist_Staff_FK

Fig 3- 2

3. A function that returns all user-defined triggers

```
create function dbo.gettriggers()
returns table
as
return select name
from sys.triggers
where type = 'tr'
go
```

	name
1	test

Fig. 3- 3

4. A function that returns row-count for all user-defined tables in the database

```
create function dbo.noofROW()
returns table
as
return select t.name, s.row_count
from sys.tables t, sys.dm_db_partition_stats s
where t.object_id = s.object_id and t.type_desc = 'USER_TABLE'
go
```

	name	row_count
1	Branch	6
2	Staff	6
3	PrivateOwner	4
4	PropertyForRent	6
5	Client	4
6	Viewing	5
7	Registration	4

Fig. 3- 4

5. A function that returns the list of all user defined functions.

```
create function dbo.gettriggers()
returns table
as
return select name
from sys.objects
where type = 'IF'
go
```

	name
1	gettriggers
2	getTablelist
3	getPK_FK
4	noofROW
5	getColumnsList
6	getParameters

Fig. 3- 5

6. A function that takes a table name as input and returns the column names with their types

```
create function dbo.getColumnsList(@tname as varchar(40))
returns table
as return
select SCHEMA_NAME(schema_id) as schema_name
,o.name as object_name, o.type, o.type_desc
,c.name as column_name
,TYPE_NAME(c.user_type_id) as column_type
,c.max_length

from sys.objects as o, sys.columns as c
where o.object_id = c.object_id and o.object_id = object_id(@tname)
go
```

	schema_name	object_name	type	type_desc	column_name	column_type	max_length
1	dbo	Branch	U	USER_TABLE	branchNo	char	4
2	dbo	Branch	U	USER_TABLE	street	varchar	25
3	dbo	Branch	U	USER_TABLE	city	varchar	15
4	dbo	Branch	U	USER_TABLE	postcode	varchar	8

**Fig. 3- 6**

7. A function that takes a function name as input and returns input parameters for that function with their types

```
create function dbo.getParameters(@fname as varchar(30))
returns table
return
select SCHEMA_NAME(schema_id) as schema_name
,o.name as object_name, o.type, o.type_desc
,p.name as parameter_name
,TYPE_NAME(p.user_type_id) as parameter_type
,p.max_length
,p.precision

from sys.objects as o, sys.parameters as p
where o.object_id = p.object_id and o.object_id = object_id(@fname)
go
```

	schema_name	object_name	type	type_desc	parameter_name	parameter_type	max_length	precision
1	dbo	getColumnsList	IF	SQL_INLINE_TABLE_VALUED_FUNCTION	@tname	varchar	40	0

**Fig. 3- 7**

### 느낀 점

System 테이블을 이용하여 데이터베이스에 좀 더 간단하게 접근하여 여러가지 기능을 수행하는 법을 배울 수 있어 뜻 깊었고 처음 t-sql을 접했을 때 생소한 문법 때문에 어려웠었는데 이번 기회로 조금은 더 실력이 향상 된 것 같아 좋았다.

### 2.3 Task-03

Define at least five functions related to your database project application and implement them by writing user defined functions

### 해결방법

- function 1 : 대출가능 한 도서목록 출력
- function 2 : 대출일(roandate)에 따른 반납일(returndate) 설정
- function 3 : 반납일(returndate)에 따른 연체일 설정
- function 4 : 연체 된 도서목록 출력
- function 5 : 도서명으로 도서 검색

### 결과

function 1 : 대출가능 한 도서목록 출력

```
create function Loan_available()
returns table
return select
*from Book where (BSTATE = 0)
```

	BID	ISBN	BSTATE
1	1	142-5433-4324	0
2	4	142-5433-4327	0
3	6	142-5433-4354	0
4	7	142-5433-4375	0
5	8	142-5433-4314	0

Fig.4- 1

function 2 : 대출일(roandate)에 따른 반납일(returndate) 설정

```
create function setReturndate(@loandate date)
returns date
as
begin
return DATEADD(day, 7, @loandate)
end
go
```

메시지

2020-03-12

Fig.4- 2

function 3 : 반납일(returndate)에 따른 연체일 설정

```
create function setOverdue(@returndate date)
returns int
as
begin
return DATEDIFF(day, @returndate, sysdatetime())
end
go
```

메시지

25

Fig.4- 3

function 4 : 연체 된 도서목록 출력

```
create function dbo.getOverdueList()
returns table
as
return select * from Rental
where RRETURNDATE = ''
go
```

	RID	BID	UID	RROANDATE	RRETURNDATE
1	5	32	14	2020-10-22	1900-01-01
2	8	1	15	2020-10-21	1900-01-01
3	9	9	16	2020-10-19	1900-01-01

**Fig.4- 4**

function 5 : 도서명으로 도서 검색

```
create function dbo.searchBook(@bname varchar(20))
returns table
as
return select *from Bookinfo
where BNAME = @bname
go
```

```
select *from searchBook( 'DatabaseSystem' )
```

	ISBN	BNAME	BPUBLISHER	BAUTHOR	BDATE
1	0-321-52306-7	DatabaseSystem	koreatech	park	2015-05-15

**Fig.4- 5**

## 느낀 점

수업시간에 배운 내용을 바탕으로 텀프로젝트에서 수행 가능한 스칼라, 테이블 함수를 구현함으로써 내가 만든 데이터베이스가 점차 논리적이고 설계적이게 구현 되어가는 과정이 보람있었다.

## 2.4 Task-04

Write store procedures for each table to

1. insert a record
2. delete a record
3. update a record

1. insert a record

### 해결방법

```
use Book
go

create proc sp_insertBookinfo @c1 varchar(20),
                             @c2 varchar(20),
                             @c3 varchar(20),
                             @c4 varchar(20),
                             @c5 date
as
begin
    insert into dbo.Bookinfo values (@c1, @c2, @c3, @c4, @c5)
end
```

Fig.5 insert procedure 생성

### 결과

```
exec dbo.sp_insertBookinfo '0-321-52306-10', 'TOEIC', 'koreatech', 'parkjonghan', '2020-10-30'

select * from BookINFO
```

Fig.5- 1 procedure 실행

	ISBN	BNAME	BPUBLISHER	BAUTHOR	BDATE
1	0-321-52306-7	DatabaseSystem	koreatech	park	2015-05-15
2	0-321-52306-8	hello world!!	koreatech	koo	2016-08-15
3	0-321-52306-9	c++programing	koreatech	kim	2016-04-15



	ISBN	BNAME	BPUBLISHER	BAUTHOR	BDATE
1	0-321-52306-10	TOEIC	koreatech	parkjonghan	2020-10-30
2	0-321-52306-7	DatabaseSystem	koreatech	park	2015-05-15
3	0-321-52306-8	hello world!!	koreatech	koo	2016-08-15
4	0-321-52306-9	c++programing	koreatech	kim	2016-04-15

Fig.5-2 Fig.5-1 실행결과(insert)

2. delete a record

#### 해결방법

```
USE [Book]
GO

create proc [dbo].[sp_deleteBookinfo] @c1 varchar(20)
as
begin
delete from dbo.BookINFO
where ISBN = @c1;
end
```

Fig.6 delete procedure 생성

#### 결과


```
use Book
go

exec dbo.sp_deleteBookinfo '0-321-52306-10'

select *from BookINFO
```

Fig. 6-1 procedure 실행

	ISBN	BNAME	BPUBLISHER	BAUTHOR	BDATE
1	0-321-52306-10	TOEIC	koreatech	parkjonghan	2020-10-30
2	0-321-52306-7	DatabaseSystem	koreatech	park	2015-05-15
3	0-321-52306-8	hello world!!	koreatech	koo	2016-08-15
4	0-321-52306-9	c++programing	koreatech	kim	2016-04-15



	ISBN	BNAME	BPUBLISHER	BAUTHOR	BDATE
1	0-321-52306-7	DatabaseSystem	koreatech	park	2015-05-15
2	0-321-52306-8	hello world!!	koreatech	koo	2016-08-15
3	0-321-52306-9	c++programing	koreatech	kim	2016-04-15

Fig. 6-2 Fig. 6-1 실행결과(delete)

## 3. update a record

## 해결방법

```

USE [Book]
GO
|
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create proc [dbo].[sp_updateBookinfo] @c1 varchar(20),
                                     @c2 varchar(20),
                                     @c3 varchar(20),
                                     @c4 varchar(20),
                                     @c5 date
as
begin
update dbo.BookINFO
set BNAME = @c2, BPUBLISHER = @c3, BAUTHOR = @c4, BDATE = @c5
where ISBN = @c1;
end

```

Fig. 7 update procedure 생성

## 결과

```

use Book
go
exec dbo.sp_updateBookinfo '0-321-52306-10', 'TOEIC', 'koreatech', 'parkjonghan', '2020-10-30'
select *from BookINFO

```

Fig. 7-1 update procedure 실행



	ISBN	BNAME	BPUBLISHER	BAUTHOR	BDATE
1	0-321-52306-7	DatabaseSystem	koreatech	park	2015-05-15
2	0-321-52306-8	hello world!!	koreatech	koo	2016-08-15
3	0-321-52306-9	c++programing	koreatech	kim	2016-04-15



	ISBN	BNAME	BPUBLISHER	BAUTHOR	BDATE
1	0-321-52306-7	DatabaseSystem	koreatech	park	2015-05-15
2	0-321-52306-8	hello world!!	koreatech	koo	2016-08-15
3	0-321-52306-9	TOEIC	koreatech	parkjonghan	2020-10-30

Fig. 7-1 Fig. 7-2 실행결과(update)

### 3 Conclusion

#### 느낀 점

이번 과제를 통해 t-sql의 전반적인 스칼라함수, 테이블함수의 작성법과 테이블이 서로 상호작용 하도록 함수를 구현하는 법을 배웠다. 처음에 t-sql문법이 많이 생소하여 어려움이 많았지만 강의자료와 교수님의 강의를 들으며 금방 익힐 수 있었고 내가 직접 만든 데이터베이스 내에도 여러가지 기능을 수행하는 함수들을 만듦으로써 데이터베이스의 완성도를 높일 수 있어서 매우 보람있는 과제 였다고 생각한다.