notebook

May 29, 2023

[5]:
```
use TestDB1
GO

CREATE PROC PROC40
AS
BEGIN
    SELECT *
    FROM Countries
END

EXEC PROC40
```

[8]:
```
use TestDB1
GO

CREATE PROCEDURE PROC5
    @outputCount int OUTPUT
AS
BEGIN
    SELECT @outputCount = LEN(CAST([square] AS VARCHAR(MAX)))
    FROM Countries
END

DECLARE @countDigits int
EXEC PROC5 @outputCount = @countDigits OUTPUT
SELECT @countDigits
```

[9]:
```
use TestDB1
GO

CREATE PROCEDURE PROC006
AS
BEGIN
    CREATE TABLE TestTabl06 (
        name nvarchar(100),
        capital nvarchar(100),
        square float,
```

```
        population int,
        continent nvarchar(50)
    )

    INSERT INTO TestTabl06 (name, capital, square, population, continent)
    SELECT
                [name],
                [capital],
                [square],
                [population],
                continent
    FROM Countries
    WHERE [name] LIKE '%'
END

EXEC PROC006
```

[11]:
```
use TestDB1
GO

CREATE FUNCTION Fun4
(
        @maxArea float
)
RETURNS TABLE
AS
RETURN (
    SELECT
                [name],
                [capital],
                [square],
                [population],
                continent
    FROM Countries
    WHERE [square] < @maxArea
        )
GO

SELECT * FROM dbo.Fun4(1000000)
```

[15]:
```
use TestDB1
GO

CREATE FUNCTION Fun5()
RETURNS TABLE
AS
RETURN(
```

```
    SELECT [name] AS Country,
        [population] / [square] AS PopulationDensity
    FROM Countries
        )
GO

SELECT * FROM dbo.Fun5()
```

[18]:
```
use MusicProject
GO

CREATE PROCEDURE GetAuthorsBeforeDateCount
    @BirthdayMax date,
    @Count int OUTPUT
AS
BEGIN
    SELECT @Count = COUNT(*)
    FROM authors
    WHERE birthday <= @BirthdayMax
END

GO

DECLARE @AuthorsCount int
EXEC GetAuthorsBeforeDateCount @BirthdayMax = '1980-01-01', @Count =␣
 ↪@AuthorsCount OUTPUT
PRINT '    : ' + CAST(@AuthorsCount AS nvarchar(10));

GO;

use MusicProject
GO

CREATE PROCEDURE GetAlbumsByAuthor
    @AuthorID int
AS
BEGIN
    SELECT name, release
    FROM albums
    WHERE author = @AuthorID
END

GO

EXEC GetAlbumsByAuthor @AuthorID = 3

GO;
```

```
use MusicProject
GO

CREATE FUNCTION GetAuthorsBeforeDateCount2
(
    @BirthdayMax date
)
RETURNS int
AS
BEGIN
    DECLARE @Count int
    SELECT @Count = COUNT(*)
    FROM authors
    WHERE birthday <= @BirthdayMax
    RETURN @Count
END

GO

DECLARE @AuthorsCount int
SET @AuthorsCount = dbo.GetAuthorsBeforeDateCount2('1980-01-01')
PRINT '    : ' + CAST(@AuthorsCount AS nvarchar(10))

GO;

use MusicProject
GO

CREATE FUNCTION GetAlbumsByAuthor2
(
    @AuthorID int
)
RETURNS TABLE
AS
RETURN
(
    SELECT name, release
    FROM albums
    WHERE author = @AuthorID
)

GO

SELECT *
FROM dbo.GetAlbumsByAuthor2(6)
```

```
GO;

use MusicProject
GO

CREATE TRIGGER after_insert_albums
ON albums
AFTER INSERT
AS
BEGIN
    PRINT 'Trigger fired after insert'
END

GO

INSERT INTO albums (name, author, release)
VALUES ('This album does not exist', 3, GETDATE())

GO;

use MusicProject
GO

CREATE TRIGGER insert_item
ON items
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @album_id INT
    SELECT @album_id = album FROM inserted

    IF EXISTS (SELECT * FROM albums WHERE id = @album_id)
    BEGIN
        INSERT INTO items (album, collection, type, price)
        SELECT album, collection, type, price FROM inserted
    END
    ELSE
    BEGIN
        PRINT 'Error: No such album exists'
    END
END

GO

INSERT INTO items (album, collection, type, price)
VALUES (3, 3, 'CD', 50.00)
```

```
GO;
```