

گزارش آزمایش چهارم – آزمایشگاه پایگاه داده

تمرین ۱: رویه ای بنویسید که وضعیت نمره ای کلاس را با بررسی جدول `tblstudent` که شامل فیلدهای نام دانشجو (`stuName`)، شماره دانشجو (`stuID`) و نمره ی دانشجو (`stuGrade`) است به عنوان خروجی برگرداند. اگر تعداد دانشجویانی که نمره ی کم تر از ۱۰ گرفته اند، حداکثر یک نفر باشد وضعیت خوب (`GOOD`) است. اگر این تعداد دو یا سه نفر باشد وضعیت کلاس نرمال (`Normal`) است و در نهایت اگر این تعداد بیش تر بود، وضعیت کلاس بد (`Bad`) است. توجه: شماره ی دانشجویی کلید اصلی است. و `stuGrade` از نوع `real` است.

ابتدا جدول خواسته شده را تعریف میکنیم و داده ها را به آن اضافه میکنیم:

	Column Name	Data Type	Allow Nulls
	stuName	varchar(10)	<input checked="" type="checkbox"/>
🔑	stuID	nchar(7)	<input type="checkbox"/>
	stuGrade	real	<input checked="" type="checkbox"/>

	stuName	stuID	stuGrade
	Samin	9839039	19
	Hosna	9823010	20
	Mary	9831010	9
	Helen	9831011	7
▶*	NULL	NULL	NULL

با توجه به داده ها باید وضعیت نرمال شناسایی شود.

```

CREATE PROCEDURE spCalculateClassStatus
AS
BEGIN
    SELECT
        CASE
            WHEN COUNT(CASE WHEN stuGrade < 10 THEN 1 END) <= 1 THEN 'GOOD'
            WHEN COUNT(CASE WHEN stuGrade < 10 THEN 1 END) BETWEEN 2 AND 3 THEN 'Normal'
            ELSE 'Bad'
        END AS ClassStatus
    FROM tblstudent;
END

EXEC spCalculateClassStatus;
    
```

100 %

Results Messages

ClassStatus
1 Normal

تمرین ۴. روالی بنویسید که پارامتر `num` از نوع `int` را از ورودی بگیرد، با تعداد نمرات زیر ۱۰ از `tblStudent` مقایسه کند، اگر تعداد نمرات زیر ده از `num` کمتر بود، نمرات بین ۹ و ۱۰ را یک نمره اضافه کند، در غیر این صورت نمرات بین ۹.۵ تا ۱۰ را ۰.۵ اضافه کند.

ابتدا اطلاعات جدول را تغییر می‌دهیم تا نتیجه رویه شفاف تر شود:

NYX.Database4 - dbo.tblStudent			
	stuName	stuID	stuGrade
	Samin	9839039	19
	Mary	9839031	9
	Hosna	9823010	20
▶	Helene	9831031	9.25
	Ava	9823011	10
*	NULL	NULL	NULL

حالا اگر مقدار ورودی رویه ۴ باشد باید به این نمرات یکی اضافه شود:

```

CREATE PROCEDURE spUpdateGrades
    @num INT
AS
BEGIN
    DECLARE @count INT;
    SELECT @count = COUNT(*) FROM tblStudent WHERE stuGrade < 10;

    IF @count < @num
    BEGIN
        UPDATE tblStudent SET stuGrade = stuGrade + 1 WHERE stuGrade >= 9 AND stuGrade <= 10;
    END
    ELSE
    BEGIN
        UPDATE tblStudent SET stuGrade = stuGrade + 0.5 WHERE stuGrade >= 9.5 AND stuGrade <= 10;
    END

    SELECT * FROM tblStudent;
END

EXEC spUpdateGrades @num = 4;

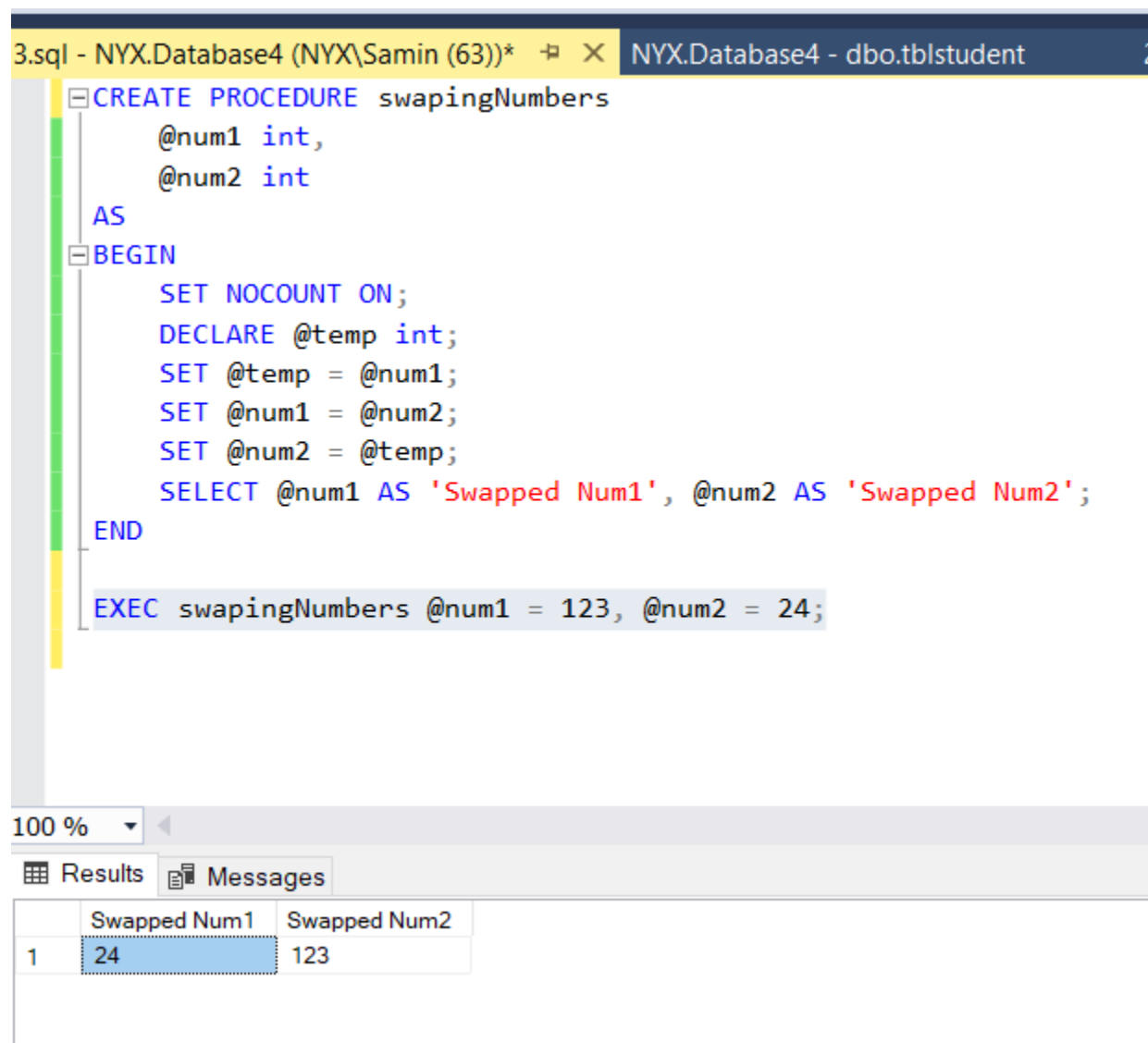
```

100 %

Results Messages

	stuName	stuID	stuGrade
1	Samin	9839039	19
2	Mary	9839031	10
3	Hosna	9823010	20
4	Helene	9831031	10.25
5	Ava	9823011	11

تمرین ۳. روالی بنویسید که دو عدد را swap کند. روال را با اعداد ۱۲۳ و ۲۴ امتحان کنید.



The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows the definition of a stored procedure named 'swapingNumbers' in the 'NYX.Database4 - dbo.tblstudent' schema. The procedure takes two integer parameters, @num1 and @num2, and performs a swap using a temporary variable @temp. The bottom pane shows the execution results of the procedure, which returned two columns: 'Swapped Num1' and 'Swapped Num2'. The first row of results shows the values 24 and 123, respectively.

```
CREATE PROCEDURE swapingNumbers
    @num1 int,
    @num2 int
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @temp int;
    SET @temp = @num1;
    SET @num1 = @num2;
    SET @num2 = @temp;
    SELECT @num1 AS 'Swapped Num1', @num2 AS 'Swapped Num2';
END

EXEC swapingNumbers @num1 = 123, @num2 = 24;
```

	Swapped Num1	Swapped Num2
1	24	123

همانطور که مشاهده میشود عدد اول که در کد ۱۲۳ بوده با ۲۴ جابجا شده است.

تمرین ۴. تابعی بنویسید که نام یک دانشجو را به عنوان ورودی بگیرد و نمره‌ی آن دانشجوی خاص را برگرداند. دانشجویی با نام **ali** با شماره‌ی دانشجویی 9012345 و نمره‌ی ۱۴ را وارد جدول کنید. تابع نوشته شده را با 'ali' امتحان کنید.

اطلاعات دانشجوی خواسته شده را وارد جدول می‌کنیم:

	stuName	stuID	stuGrade
	Samin	9839039	19
	Mary	9839031	9
	Hosna	9823010	20
	Helene	9831031	9.25
	Ava	9823011	10
	Ali	9012345	14

```
CREATE FUNCTION getGradeByName
    (@name varchar(10))
RETURNS real
AS
BEGIN
    DECLARE @grade real;
    SELECT @grade = stuGrade FROM tblstudent WHERE stuName = @name;
    RETURN @grade;
END
```

حالا این تابع را تست می‌کنیم:

The screenshot shows a SQL query window with the following query:

```
SELECT dbo.getGradeByName('Ali') AS 'Ali Grade';
```


The query is executed, and the results are displayed in the 'Results' tab. The result is a single row with the value 14 for the column 'Ali Grade'.

	Ali Grade
1	14

همانطور که مشاهده میشود نمره دانشجوی خواسته شده نمایش داده شده است.

تمرین ۵. جدول زیر را وارد کرده و با استفاده از توابع رنکدهی گفته شده، بر اساس postal code رنک دهی کنید.

ابتدا جدول خواسته شده را ایجاد کرده و داده ها را وارد میکنیم:

	Column Name	Data Type	Allow Nulls
	FirstName	varchar(30)	<input checked="" type="checkbox"/>
	LastName	varchar(50)	<input type="checkbox"/>
	PostalCode	nchar(5)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

NYX.Database4 - dbo.tblpost		NYX.Database4 - db...ase	
	FirstName	LastName	PostalCode
	Micheal	Blythe	98027
	Linda	Mitchell	98027
	Jillian	Carson	98027
	Garret	Vargas	98027
	Tsvi	Reiter	98027
	Shu	Ito	98055
	Jose	Saraiva	98055
	David	Campbell	98055
	Tete	Mensa-Ann...	98055
	Lynn	Tsofilias	98055
	Rachel	Valdez	98055
	Jae	Pak	98055
	Ranjit	Varkry Chud...	98055
▶*	NULL	NULL	NULL

SELECT FirstName, LastName, PostalCode, RANK() OVER (PARTITION BY PostalCode ORDER BY LastName, FirstName) AS PostalCodeRank FROM tblpost;			
100 %			
Results	Messages		
FirstName	LastName	PostalCode	PostalCodeRank
1 Micheal	Blythe	98027	1
2 Jillian	Carson	98027	2
3 Linda	Mitchell	98027	3
4 Tsvi	Reiter	98027	4
5 Garret	Vargas	98027	5
6 David	Campbell	98055	1
7 Shu	Ito	98055	2
8 Tete	Mensa-Annan	98055	3
9 Jae	Pak	98055	4
10 Jose	Saraiva	98055	5
11 Lynn	Tsofilias	98055	6
12 Rachel	Valdez	98055	7
13 Ranjit	Varkry Chudukatil	98055	8

تمرین ۶. تابعی بنویسید که زمان جاری را به عنوان ورودی بگیرد و و قسمت روز ۴ روز بعد را به عنوان خروجی برگرداند. (راهنمایی: استفاده از DATENAME).

```
CREATE FUNCTION GetDateAfterFourDays(@currentDate DATETIME)
RETURNS VARCHAR(10)
AS
BEGIN
    DECLARE @result VARCHAR(10)
    SET @result = DATENAME(weekday, DATEADD(day, 4, @currentDate))
    RETURN @result
END
```

```
SELECT dbo.GetDateAfterFourDays(GETDATE())
```

100 %

Results Messages

	(No column name)
1	Wednesday

تمرین ۷. آیا function ها را می توان با دستوراتی مانند insert/update/delete به کار برد(امتحان کنید و نتیجه را همراه با عکس ذکر کنید).

توابع برای تغییر دادن داده ها طراحی نشده اند بنابراین نمیتوانند همراه دستورات ذکر شده استفاده شوند و در صورت استفاده با خطا روبرو خواهیم شد:

فرض کنید جدولی به این صورت داریم:

NYX.Database4 - d...ase4 - dbo.myTable		7.sql - NYX.Database4 (NYX\Samin (56))
Column Name	Data Type	Allow Nulls
myColumn	int	<input type="checkbox"/>
		<input type="checkbox"/>

حالا میخواهیم یک تابع بنویسیم که مقدار ورودی بگیرد و به این جدول بیافزاید:

```

7.sql - NYX.Database4 (NYX\Samin (57))
CREATE FUNCTION myFunction (@param INT)
RETURNS INT
AS
BEGIN
    INSERT INTO myTable (myColumn) VALUES (@param)
    RETURN 1
END

```

00 %

Messages

Msg 443, Level 16, State 15, Procedure myFunction, Line 5 [Batch Start Line 0]
Invalid use of a side-effecting operator 'INSERT' within a function.

Completion time: 2023-04-22T21:39:45.2904960+03:30

همانطور که مشاهده میشود هنگام اجرای این تابع با خطا روبرو میشویم.

برای تست کردن آپدیت تابع دیگری مینویسیم که یک عدد گرفته و به مقدار سطری که مقدار ۱ دارد آن را می افزاید:

	myColumn
	1
	2
	3
▶*	NULL

```

CREATE FUNCTION updateMyTable
(
    @myValue INT
)
RETURNS INT
AS
BEGIN
    UPDATE myTable
    SET myColumn = myColumn + @myValue
    WHERE myColumn = 1
    RETURN @@ROWCOUNT
END

```

100 %

Messages

Msg 443, Level 16, State 15, Procedure updateMyTable, Line 8 [Batch Start Line 0]
Invalid use of a side-effecting operator 'UPDATE' within a function.

Completion time: 2023-04-22T22:06:01.1917842+03:30

همانطور که مشاهده میشود با ارور روبرو شدیم.

برای تست کردن دیلیت تابعی مینویسیم که یک ورودی گرفته و سطری که آن ورودی دارد را حذف میکند:

```
SQLQuery4.sql - NY...4 (NYX\Samin (65))* NYX.Database4 - dbo.myTable SC
CREATE FUNCTION dbo.DeleteRows
(
    @myValue INT
)
RETURNS INT
AS
BEGIN
    DELETE FROM myTable WHERE myColumn = @myValue;
    RETURN @@ROWCOUNT;
END;
```

00 %

Messages

Msg 443, Level 16, State 15, Procedure DeleteRows, Line 8 [Batch Start Line 0]
Invalid use of a side-effecting operator 'DELETE' within a function.

Completion time: 2023-04-22T22:08:39.6375661+03:30

همانطور که مشاهده میشود با ارور روبرو شدیم.