

بنام خدا

پایگاه داده ۲

Structured Query Language (SQL)

بصیری

دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

خلاصه فصل

- چند تابع کاربردی پیش فرض (برخی منابع:
(<http://www.oracletutorial.com>, www.w3schools.com)
- نوشتن تابع
- نوشتن پراسیجر

جدول Products زیر را در نظر بگیرید

P_Id	ProductName	UnitPrice	UnitsInStock	UnitsOnOrder
1	Jarlsberg	10.45	16	15
2	Mascarpone	32.56	23	
3	Gorgonzola	15.67	9	20

اگر بخواهیم ارزش موجود در سفارش و انبار را داشته باشیم:

```
SELECT ProductName, UnitPrice * (UnitsInStock + UnitsOnOrder)  
FROM Products;
```

روش فوق چه مشکلی دارد؟

جایگذاری مقدار برای مقادیر نال

SQL Server ■

ISNULL(string1, replace_with) □

```
SELECT ProductName,  
       UnitPrice * (UnitsInStock + ISNULL(UnitsOnOrder, 0))  
FROM Products
```

Oracle ■

NVL(string1, replace_with) □

```
SELECT ProductName,  
       UnitPrice * (UnitsInStock + NVL(UnitsOnOrder, 0))  
FROM Products
```

یافتن رکورد قبلی و بعدی

■ SQL Server و Oracle یکی است

■ رکورد قبلی

```
LAG ( expression [, offset [, default] ] )  
OVER ( [ query_partition_clause ] order_by_clause )
```

■ رکورد بعدی

```
LEAD ( expression [, offset [, default] ] )  
OVER ( [ query_partition_clause ] order_by_clause )
```

جدول employees زیر را در نظر بگیرید

employee_number	last_name	first_name	salary	dept_id
12009	Sutherland	Barbara	54000	45
34974	Yates	Fred	80000	45
34987	Erickson	Neil	42000	45
45001	Parker	Sally	57500	30
75623	Gates	Steve	65000	30

```
SELECT dept_id, last_name, salary, LAG (salary,1) OVER (ORDER BY salary)
AS lower_salary FROM employees;
```



dept_id	last_name	salary	lower_salary
45	Erickson	42000	NULL
45	Sutherland	54000	42000
30	Parker	57500	54000
30	Gates	65000	57500
45	Yates	80000	65000

CASE Syntax

■ CASE

```
WHEN condition1 THEN result1  
WHEN condition2 THEN result2  
WHEN conditionN THEN resultN  
ELSE result  
END;
```

■ اولین شرط که برقرار باشد استفاده می شود.

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
2	10248	42	10
3	10248	72	5
4	10249	14	9
5	10249	51	40

- ```

SELECT OrderID, Quantity,
CASE WHEN Quantity > 30 THEN "The quantity is greater than 30"
 WHEN Quantity = 30 THEN "The quantity is 30"
 ELSE "The quantity is under 30"
END AS QuantityText
FROM OrderDetails;

```

| OrderID | Quantity | QuantityText                    |
|---------|----------|---------------------------------|
| 10248   | 12       | The quantity is under 30        |
| 10248   | 10       | The quantity is under 30        |
| 10248   | 5        | The quantity is under 30        |
| 10249   | 9        | The quantity is under 30        |
| 10249   | 40       | The quantity is greater than 30 |



# توابع مهم مربوط به رشته در SQL Server

| Function         | Example                      | Result  | Description                                                                       |
|------------------|------------------------------|---------|-----------------------------------------------------------------------------------|
| <b>CONCAT</b>    | CONCAT('WWW','com')          | WWW.com | Adds two or more strings together                                                 |
| <b>LEN</b>       | LEN('com')                   | 3       | Returns the length of a string                                                    |
| <b>LOWER</b>     | LOWER('SQL!')                | sql!    | Converts a string to lower-case                                                   |
| <b>LTRIM</b>     | LTRIM(' SQL !')              | SQL !   | Removes leading spaces from a string                                              |
| <b>PATINDEX</b>  | PATINDEX('%sc%', 'W3Sc.com') | 3       | Returns the position of a pattern in a string                                     |
| <b>REPLACE</b>   | REPLACE('TSQL', 'T', 'M')    | MSQL    | Replaces all occurrences of a substring within a string, with a new substring     |
| <b>RTRIM</b>     | RTRIM('SQL ')                | SQL     | Removes trailing spaces from a string                                             |
| <b>SUBSTRING</b> | SUBSTRING('SQL', 1, 2)       | SQ      | Extracts some characters from a string                                            |
| <b>TRIM</b>      | TRIM(' SQL ')                | SQL     | Removes leading and trailing spaces (or other specified characters) from a string |
| <b>UPPER</b>     | UPPER('sql')                 | SQL     | Converts a string to upper-case                                                   |
| <b>REPLICATE</b> | REPLICATE('SQ', 2)           | SQSQ    | Repeats a string a specified number of times                                      |
| <b>RIGHT</b>     | RIGHT('SQL', 2)              | QL      | Extracts a number of characters from a string (starting from right)               |
| <b>LEFT</b>      | LEFT('SQL', 2)               | SQ      | Extracts a number of characters from a string (starting from left)                |

# توابع مهم مربوط به رشته در Oracle

| Function       | Example                            | Result            | Purpose                                                                                           |
|----------------|------------------------------------|-------------------|---------------------------------------------------------------------------------------------------|
| <b>CONCAT</b>  | CONCAT('A','BC')                   | 'ABC'             | Concatenate two strings and return the combined string                                            |
| <b>INSTR</b>   | INSTR('This is a playlist', 'is')  | 3                 | Search for a substring and return the location of the substring in a string                       |
| <b>LENGTH</b>  | LENGTH('ABC')                      | 3                 | Return the number of characters (or length) of a specified string                                 |
| <b>LOWER</b>   | LOWER('Abc')                       | 'abc'             | Return a string with all characters converted to lowercase                                        |
| <b>LPAD</b>    | LPAD('ABC',5,'*')                  | '**ABC'           | Return a string that is left-padded with the specified characters to a certain length.            |
| <b>LTRIM</b>   | LTRIM(' ABC ')                     | 'ABC '            | Remove spaces or other specified characters in a set from the left end of a string                |
| <b>REPLACE</b> | REPLACE('JACK AND JOND','J','BL'); | 'BLACK AND BLOND' | Replace all occurrences of a substring by another substring in a string                           |
| <b>RPAD</b>    | RPAD('ABC',5,'*')                  | 'ABC**'           | Return a string that is right-padded with the specified characters to a certain length.           |
| <b>RTRIM</b>   | RTRIM(' ABC ')                     | ' ABC'            | Remove all spaces or specified character in a set from the right end of a string                  |
| <b>SUBSTR</b>  | SUBSTR('Oracle Substring', 1, 6)   | 'Oracle'          | Extract a substring from a string                                                                 |
| <b>TRIM</b>    | TRIM(' ABC ')                      | 'ABC'             | Remove the space character or other specified characters either from the start or end of a string |
| <b>UPPER</b>   | UPPER('Abc')                       | 'ABC'             | Convert all characters in a specified string to uppercase                                         |

## خلاصه فصل

- چند تابع کاربردی پیش فرض (برخی منابع:  
(<http://www.oracletutorial.com>, [www.w3schools.com](http://www.w3schools.com))
- نوشتن تابع
- نوشتن پراسیجر

تابع اسکالر: این توابع تک خروجی هستند

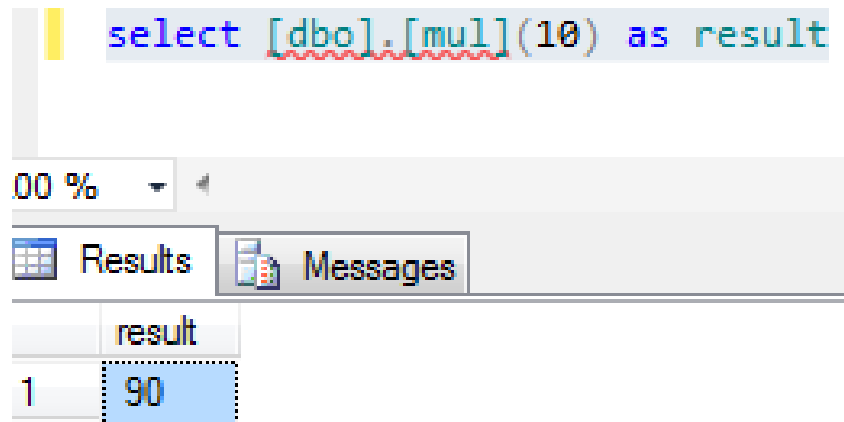
```
CREATE FUNCTION [owner_name.] function_name
 (@parameter_name [AS] data_type)
RETURNS data_type
AS
BEGIN

 function_body

 RETURN scalar_expression
END
```

تابعی که  $n$  را دریافت کرده و  $n*(n-1)$  را خروجی می دهد.

```
create function [dbo].[mul] (@n int)
returns bigint
AS
begin
 declare @r bigint
 set @r=(@n-1)*@n
 return @r
end
```



The screenshot shows a SQL query window with the following text: `select [dbo].[mul](10) as result`. Below the query window, the 'Results' tab is active, displaying a table with one column named 'result' and one row containing the value '90'.

|   | result |
|---|--------|
| 1 | 90     |

تابعی که شماره یک دانشجو را گرفته و نام و نام خانوادگی وی را نمایش می دهد .

```
Create FUNCTION fn_Nameret(@s# int)
 RETURNS NVARCHAR(40)
AS
BEGIN
 DECLARE @ret_Value NVARCHAR(40);
 set @ret_Value = (SELECT Name + ',' +Family
 FROM STD WHERE S# =@S#)

 RETURN (@ret_Value)
End
```

توابع table-valued: یک جدول خروجی می دهند.

```
CREATE FUNCTION [owner_name.] function_name
 (@parameter_name [AS] data_type)
RETURNS table
AS
BEGIN
```

**function\_body**

```
 RETURN (select)
END
```

تفاوت اصلی این تعریف با توابع اسکالر در این است که نوع خروجی یک جدول تعریف شده است. محدودیتی که روی این نوع تعریف وجود دارد این است که خروجی تابع باید توسط یک دستور **select** ایجاد شود.

## سوال

- تابعی بنویسید که یک رشته به طول حداکثر ۲۰ و یک الگو به طول حداکثر ۳ دریافت نماید. این تابع باید تمام تکرارهای الگو را از رشته بزرگتر حذف کرده و رشته باقی مانده را در برگرداند. اگر با حذف یکباره دوباره رشته شامل آن الگو بود باید مجدد حذف شود.



## سوال

■ تابعی بنویسید که یک رشته به عنوان کد ملی دریافت کند و صحت کد ملی دریافت شده را بررسی و در خروجی برگرداند. مقدار ۱ به معنای درست بودن کد ملی وارد شده و مقدار صفر به معنای نادرست بودن آن است. از آدرس زیر می توانید نحوه صحت سنجی کد ملی را بیابید:

<http://www.aliarash.com/article/codemeli/codemeli.htm>

## خلاصه فصل

- چند تابع کاربردی پیش فرض (برخی منابع:  
(<http://www.oracletutorial.com>, [www.w3schools.com](http://www.w3schools.com))
- نوشتن تابع
- نوشتن پراسیجر



# Procedure

- A stored procedure is a set of Structured Query Language (SQL) statements with an assigned name, which are stored in a relational database management system as a group, so it can be reused and shared by multiple programs.
- Stored procedures can access or modify data in a database, but it is not tied to a specific database or object, which offers a number of advantages.



## **Benefits of using stored procedures**

- A stored procedure provides an important layer of security between the user interface and the database. It supports security through data access controls because end users may enter or change data, but do not write procedures.
- A stored procedure preserves data integrity because information is entered in a consistent manner. It improves productivity because statements in a stored procedure only must be written once.



## Benefits of using stored procedures (Cont.)

- Stored procedures offer advantages over embedding queries in a graphical user interface (GUI). Since stored procedures are modular, it is easier to troubleshoot when a problem arises in an application.
- Stored procedures are also tunable, which eliminates the need to modify the GUI source code to improve its performance. It's easier to code stored procedures than to build a query through a GUI.

## نوشتن پراسیجر

```
CREATE PROCEDURE [owner_name.] procedure_name
 (@parameter_name [AS] data_type)
AS
BEGIN

 procedure_body

END
```

## مثال ١

```
1. CREATE PROCEDURE stpGetAllMembers
2. AS
3. BEGIN
4. -- Insert statements for procedure here
5. Select * from tblMembers
6. END
```

## مثال ٢

```
1. CREATE PROCEDURE stpGetMembersByCityName
2. -- Add the parameters for the stored procedure here
3. @CityName nvarchar(30)
4. AS
5. BEGIN
6. Select * From tblMembers
7. where MemberCity like '%' + @CityName + '%'
8.
9. END
```



## مثال ٣

1. **CREATE PROCEDURE** stpInsertMember
2. @MemberName **varchar**(50),
3. @MemberCity **varchar**(25),
4. @MemberPhone **varchar**(15)
5. **AS**
6. **BEGIN**
7.     **Insert into** tblMembers (MemberName,MemberCity,MemberPhone)  
       **Values** (@MemberName,@MemberCity, @MemberPhone)
8.     **END**

## مثال ٤

```
1. CREATE PROCEDURE stpUpdateMemberByID
2. @MemberID int,
3. @MemberName varchar(50),
4. @MemberCity varchar(25),
5. @MemberPhone varchar(15)
6.
7. AS
8. BEGIN
9. UPDATE tblMembers
10. Set MemberName = @MemberName,
11. MemberCity = @MemberCity,
12. MemberPhone = @MemberPhone
13. Where MemberID = @MemberID
14. END
```

## سوال

پراسیجری بنویسید که از جدول مشتریان بنام **Customer**، که به اشتباه برای یک مشتری ممکن است حاوی چند رکورد باشد، صرفاً اولین رکورد را بر اساس فیلد تاریخ ثبت (**RegDat**) در جدول **Customer** ذخیره کرده و رکوردهای اضافه را حذف نماید. شما می توانید از جداول **temp** برای نوشتن این پراسیجر استفاده کنید.

Customer

| Natcod | Name | Job | RegDat |
|--------|------|-----|--------|
|        |      |     |        |
|        |      |     |        |

## مثال ۵

بانکی را در نظر بگیرید که مشتریان در آن سپرده گذاری کرده اند و برای هر مشتری ممکن است تعدادی سپرده (Deposit) وجود داشته باشد. اطلاعات پایه ای سپرده مانند شماره سپرده و نوع ارز آن، شماره مشتری و... را در جدولی بنام `dimdeposit` ذخیره کرده ایم.

هر سپرده در یک تاریخ افتتاح شده و با واریز و برداشتهایی که مشتری روی حسابش انجام می دهد می تواند مانده سپرده اش را تغییر دهد. فرض کنید در یک جدول بنام `factdeposit` مانده تمام سپرده ها را در هر روز نگهداری می کنیم. بر اساس این اطلاعات میانگین موجودی هر سپرده را طی یکسال در جدولی ذخیره کنید.

## مثال ۵

```
CREATE OR REPLACE PROCEDURE "PROC1" is
currdate date;
begin
currdate := to_date('930101', 'yymmdd', 'nls_calendar=persian');
delete from temp1; delete from temp2;
delete from temp3; delete from Final;
commit;
insert into temp3
 select d.Depositkey,d.acurrcode,d.acurrdesc,
 0 active_date_coutn,
 0 avg_balance
from dimdeposit d;
```

## ادامه مثال ۵

```
while currrdate <= to_date('931229', 'yymmdd', 'nls_calendar=persian') loop

 insert into temp1

 select f.Depositkey, 1, f.bal

 from factdeposit f where f.effdate = currrdate;

commit;

insert into temp2

select p.Depositkey, p.acurrcode, p.acurrdesc,

 case when tmp.active_date_coutn is null then p.active_date_coutn

 else p.active_date_coutn + 1 end active_date_coutn,

 case when tmp.active_date_coutn is null then

 p.avg_balance

 Else ((p.active_date_coutn * p.avg_balance) + tmp.avg_balance) /

 (p.active_date_coutn + 1)

 end avg_balance

from temp3 p left outer join temp1 tmp

on tmp.Depositkey = p.Depositkey;
```

## ادامه مثال ۵

```
delete from temp3; commit;

insert into temp3

select * from temp2;

delete from temp1; delete from temp2; commit;

currdate := currdate + 1;

end loop;

insert into Final

select t.Depositkey, t.acurrcode,t.acurrdesc,t.active_date_coun,

t.avg_balance,

case when t.acurrcode = 'IRR' then t.avg_balance

else t.avg_balance * cur.acratebuy end avg_balance_moadel_riali

from temp3 t left join afcurrdate cur

on cur.acurrcode = t.acurrcode

and cur.acratedat =to_date('1393/12/28', 'yyyy/mm/dd', 'nls_calendar=persian');

commit;

end PROC1;
```