SQL Server
Part: Function & Trigger

آزمایشگاه پایگاه داده
حمیدرضا رمضانی
h.ramezany72@gmail.com

Function

System Functions

Aggregate Functions	Configuration Functions	Cryptographic Functions	Cursor Functions
Date & Time Functions	Mathematical Functions	Metadata Function	Row Set Functions
Ranking Functions	Security Functions	String Functions	
System Statistical Functions		Text and Image Functions	

User Defined Function

- Aggregate Functions
- Scalar-Valued Functions
- Table-Valued Functions
  - Inline Functions
  - Multi Statement Functions

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

Function

توابع سیستمی توابعی هستند که در دسته بندی های گوناگون جهت مصارف گوناگون ساخته شده اند

GetDate	تاریخ و زمان جاری را از سرور بر می گرداند
Day	شماره روز را از تاریخ داده شده بر می گرداند
Month	شماره ماه را از تاریخ داده شده بر می گرداند
Year	شماره سال را از تاریخ داده شده بر می گرداند
DateName	نام متناظر با پارامتر داده شده و تاریخ داده شده را بر می گرداند
DatePart	بخش خواسته شده از تاریخ داده شده را بر می گرداند
DateAdd	تاریخ جدید بر اساس تاریخ داده شده و اختلاف داده شده را بر می‌گرداند
DateDiff	اختلاف دو تاریخ را بر اساس بخش خواسته شده بر می گرداند

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Function

برای ایجاد توابع User Defined به ترتیب زیر عمل می کنیم
خروجی این توابع یک و فقط یک مقدار خواهد بود . قالب کلی تعریف آن به صورت زیر است :

```
Create    Function  نام
                    (فهرست پارامترها)
Returns   نوع داده ای
As
          Begin
                ...
          End
```

چند نکته :

۱- توابع اسکالر ، امکان فراهم سازی خروجی به شکل یک Result Set را ندارند

۲- امکان فراخوانی بعضی دستورات در توابع مثل اجرای Stored Procedure وجود ندارد .

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Function

مثال :

```
USE G1Q1
GO
Create     Function    Sum(
                              @FirstNumber       Int ,
                              @SecondNumber    Int
                              )
           Returns       BigInt
As
   Begin
           Return     @FirstNumber  + @SecondNumber
   End
--Test
GO
Select dbo.Sum(10,20)
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Function

InLine Table Valued Functions

توابع InLine از لحاظ ساختاری مشابه View ها هستند ، با این تفاوت که به عنوان ورودی پارامتر می پذیرند .

قالب کلی تعریف آن به صورت زیر است :

```
Create    Function  نام
                    (فهرست پارامترها)
Returns   Table
As
          Return
                    (Select
                              .....
                    )
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Function

مثال :

```
CreateFunction   Quantity(   @ID        Int   )
   Returns Table
   As  Return
       (Select      Item.Title  As  Item ,      Color.Title  As  Color ,
       Inventory.Quantity  As            Quantity
              From          Item
                            Inner Join
                     Inventory
                            On     Item.ID          =    Inventory.Item_ID
                            Inner Join
                     Color
                            On     Color.ID    = Inventory.Color_ID
              Where
              Inventory.ID =   @ID
       )
-- Test
Select *    From      Quantity(3)
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Function

Multi Statement Table Valued Functions

توابع Multi Statement نیز دارای خروجی از نوع Result Set می‌باشند و معمولاً در From مورد استفاده قرار می‌گیرند.

قالب کلی تعریف آن به صورت زیر است:

```
Create    Function  نام
                        (فهرست پارامترها)
Returns  نام خروجی    Table       (فهرست ستونها)
As
      Begin
                        …
                        Return
      End
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Multi-statement Table Valued Functions

```
CREATE FUNCTION datesales2 (@deadline datetime)
RETURNS @table TABLE (
stor_id varchar(6) null,
ord_num varchar(8) null,
ord_date datetime null,
qty int, payterms varchar(20),
title_id varchar(6))
AS
BEGIN
INSERT @table
SELECT *
FROM sales
WHERE ord_date > @deadline
RETURN
END
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

تفاوت function و procedure

Functions	Stored Procedure
1) can be used with Select statement	1) have to use EXEC or EXECUTE
2) Not returning output parameter but returns Table variables	2) return output parameter
3) You can join UDF	3) can create table but won't return Table Variables
4) Cannot be used to change server configuration	4) you can not join SP
5) Cannot have transaction within function	5) can be used to change server configuration
	6) can have transaction within SP

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Trigger ها

زیر برنامه ای که بصورت خودکار و در هنگام رخ دادن رویداد خاصی ، اجرا می شود ، Trigger نامیده می شود

دو نوع Trigger در SQL Server وجود دارد :

۱- DDL Trigger

۲- DML Trigger

DDL Trigger ها اکثراً هنگامی اجرا می شوند ، که رویدادی سبب ایجاد تغییر در ساختار بانک اطلاعاتی یا اشیا در بانک اطلاعاتی شود .

DML Trigger ها در هنگام رخ دادن دستورات DML ، (Insert , Update , Delete) روی جداول و View ها اجرا می شوند . DML Trigger ها براساس زمان رخ داد خود به دسته های زیر تقسیم می شوند :

۱- Instead Of Trigger

۲- After Trigger

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

Instead of / After Triggers

Instead of Trigger همانطور که از نامش پیداست ، به جای هر یک از عملیات , Insert Update , Delete نشسته و در واقع در زمان وقوع هر یک از دستورات فوق ، به جای آن فعال می‌شود و عملیات مورد نظر را به شیوه خود انجام می دهد .

After Trigger ها زمانی وارد عمل می شوند ، که عملیات مورد نظر انجام شده و نیازی به اعمال آن توسط Trigger نیست . در این هنگام این نوع Trigger به منظور اعمال هدف خود ، اجرا می‌شود.

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

کاربرد Trigger

تعدادی از کاربردهای Trigger ها عبارتنداز :

۱- رسیدن به هدف جامعیت داده ها در جایی که Constraint ها کارایی لازم را ندارند .

۲- واقعه نگاری Logging

۳- کنترل و بومی سازی خطا ها در زمان اجرای دستورات DML

و ...

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

ایجاد Trigger ها

برای ایجاد Trigger از قالب زیر استفاده می شود :

Create Trigger *نام*

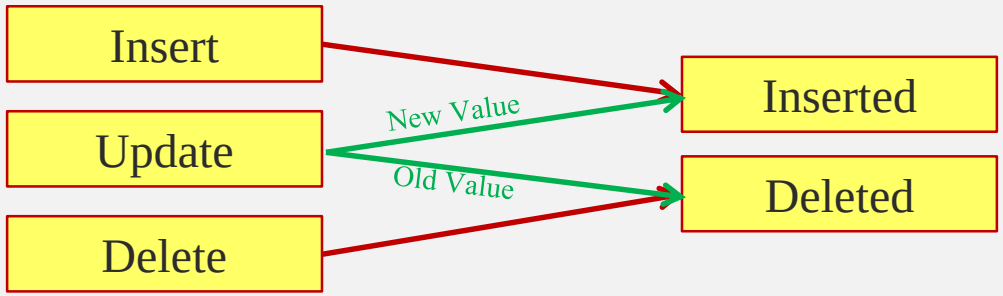
On *نام جدول یا View*

{After | Instead Of} [Insert] [,] [Update] [,] [Delete]

As Begin

End

در یک Trigger از دو جدول Inserted و Deleted برای دسترسی به رکوردهایی که تحت تاثیر قرار گرفته اند ، می توان استفاده نمود .



آزمایشگاه پایگاه داده
حمیدرضا رمضانی

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

مثال Trigger

```
CREATE TRIGGER [dbo].[scoretg] ON [dbo].[score]
INSTEAD OF INSERT AS
Begin
    DECLARE @g REAL;
    SELECT @g=inserted.grade FROM inserted;
    INSERT INTO score(grade,typeN) VALUES(@g,CASE WHEN
            @g<10 THEN 'B' else 'A' end)
End
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Create DML Trigger

```
CREATE TRIGGER [ schema_name . ]trigger_name
 ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME <method specifier [ ; ] > }
<dml_trigger_option> ::= [ ENCRYPTION ] [ EXECUTE AS Clause ]
<method_specifier> ::= assembly_name.class_name.method_name
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

Create DDL Trigger

•Trigger on a CREATE, ALTER, DROP, GRANT, DENY, REVOKE, or UPDATE STATISTICS statement (DDL Trigger)

```
CREATE TRIGGER trigger_name
ON { ALL SERVER | DATABASE }
[ WITH <ddl_trigger_option> [ ,...n ] ]
{ FOR | AFTER } { event_type | event_group } [ ,...n ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME < method specifier > [ ; ] }
<ddl_trigger_option> ::= [ ENCRYPTION ] [ EXECUTE AS Clause ]
<method_specifier> ::= assembly_name.class_name.method_name
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

ENABLE/DISABLE Trigger

- Enables a DML, DDL, or logon trigger.

ENABLE TRIGGER { [schema_name .] trigger_name [,...n] | ALL }

ON { object_name | DATABASE | ALL SERVER } [;]

- Disable Trigger

DISABLE TRIGGER { [schema_name .] trigger_name [,...n] | ALL }

ON { object_name | DATABASE | ALL SERVER } [;]

- **Example:**

```
USE G2T2;
GO
DISABLE TRIGGER Person.uAddress ON Person.Address;
GO
ENABLE Trigger Person.uAddress ON Person.Address;
GO
```

آزمایشگاه پایگاه داده
حمیدرضا رمضانی

SQL Aggregate Functions

SQL aggregate functions return a single value, calculated from values in a column.

Useful aggregate functions:

- AVG() - Returns the average value
- COUNT() - Returns the number of rows
- FIRST() - Returns the first value
- LAST() - Returns the last value
- MAX() - Returns the largest value
- MIN() - Returns the smallest value
- SUM() - Returns the sum