

## Type

تعریف نوع داده ای جدید

Example1:

```
CREATE TYPE Sid FROM varchar(11) NOT NULL
CREATE TYPE sname FROM varchar(20)
CREATE TYPE grade FROM numeric(4,2)
CREATE TYPE address FROM text
CREATE TABLE student2(id sid,name sname,addr address)
```

نحوه استفاده از داده های تعریف شده

Example2:

```
create type StdID From char(4) Not Null
create type StfName From char(20) Not Null
create type Mark From Numeric(4,2) not Null
create type Mark2 From Numeric(4,2) Not Null
```

تعریف متغیر

```
declare @a Mark
declare @b Mark2
```

مقدار دهی متغیر

```
set @a=12.4
set @b=@c+3
```

نمایش متغیر

```
select @a
select @b
```

ساخت جدول

```
create table t1 (id StdID,name StdName, avg Mark)
```

## With

**With** : يك جدول موقتي است كه در Query مربوط به خودش استفاده مي شود.  
`Create With WithName[(Parametr1, Parametr2,...)] as`  
(  
    < `Select` Expression >  
)

---

Exmple1:

جدول موقتي زير نام و رشته دانشجوياني را كه درس هایشان را پاس کرده اند، را شامل مي شود.

```
With StuPass(StuName,StuMajor) as  
(  
    select Sname,major  
    From Term  
    Where Grade>=10  
)
```

Query زير با استفاده از جدول موقتي، نام دانشجويان IT كه دروس خود را پاس کرده اند را نشان مي دهد.

```
With StuPass(StuName,StuMajor) as  
(  
    select Sname,major  
    From Term  
    Where Grade>=10  
)  
select sname  
from StuPass  
where major='IT'
```

## Function

```
Create Function FuncName [(@Parametr1 Type,@Parametr2 Type,... )]  
Returns Table  
as  
Return  
(  
    < Select Expression >  
)  
FuncName (@Parametr1 Type,@Parametr2 Type,... ) //call Function
```

---

Example1:

ایجاد تابع اطلاعات دانشجو با فیلد رشته انتخابی

```
Create Function memberMajorSelection(@major varchar(10))  
returns table  
as  
return  
(  
    Select student.*  
    From person join student on person.pid=student.pid  
    where major=@major  
)  
select * from memberMajorSelection('Computer')
```

# Trigger

## Structure:

```
CREATE TRIGGER triggerName
ON { table | view }
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
AS {
    < SQL statement >
}
```

---

```
Example1: --Delete Student
create trigger upStu2 on t1 for Update
as
    select * from t1
```

```
Example2:
alter trigger upStu on t1 for Update
as
    update t1
    set avg=avg+1
    where id in (
        select id
        from inserted i)

    update t1
    set avg=avg-3
    where id in (
        select id
        from deleted i)

update t1
set avg=avg+1
where name='a'
```

## Transaction

در تراکنش، در صورت بوجود آمدن خطا در هنگام انجام عملیات، عملیات انجام شده قبل از خطا را اعمال نمی کند.

```
BEGIN { TRAN | TRANSACTION } [ transaction_name ]  
    < Expression >  
Commit {TRAN | TRANSACTION } [ transaction_name ]
```

---

Example1:

در مثال زیر به دلیل اینکه تعداد کارکنهای 31010 بیش از حد مجاز می باشد خطا رخ خواهد داد بنابراین عملیات قبلی اعمال نخواهد شد.

```
begin try  
    begin transaction  
        insert into t1 values ('1011', 'a', 10)  
        select max(avg)  
        from t1  
  
        insert into t1 values ('31010', 'a', 15)  
  
        select min(avg)  
        from t1  
    commit transaction  
end try  
begin catch  
    rollback  
end catch
```