



آزمایشگاه پایگاه داده

جلسه سوم

انجام پرس جو های پیچیده تر – دستورات T-SQL

محمود فرجی و محمد جواد آکوچکیان

سال تحصیلی ۱۴۰۱-۱۴۰۲



دستور DELETE

این دستور برای حذف یک یا چند ردیف استفاده میشود. شکل کلی آن به صورت زیر است:

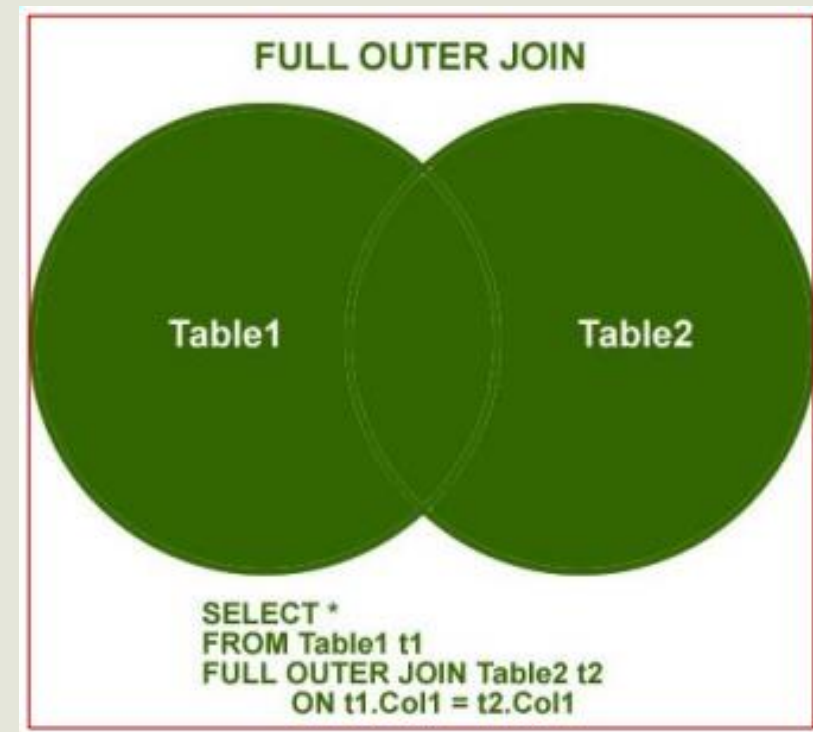
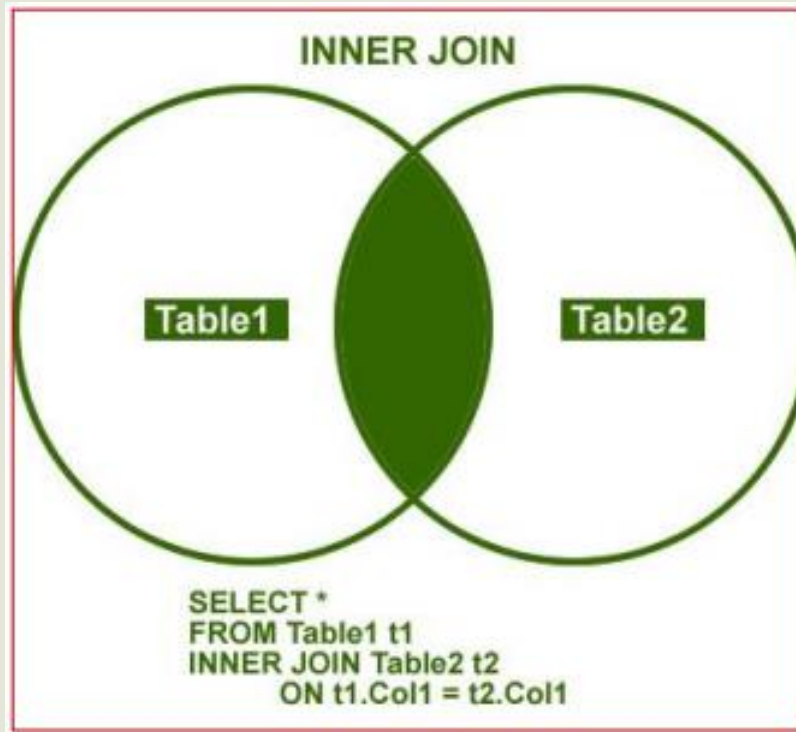
```
DELETE [FROM] table_or_view_name  
WHERE search_condition
```

```
DELETE FROM tblSailor  
WHERE sailor_rank<5
```

دستور **truncate** برای حذف تمامی ردیف های یک جدول استفاده میشود و نسبت به دستور **Delete** سرعت بیشتری دارد.



JOIN

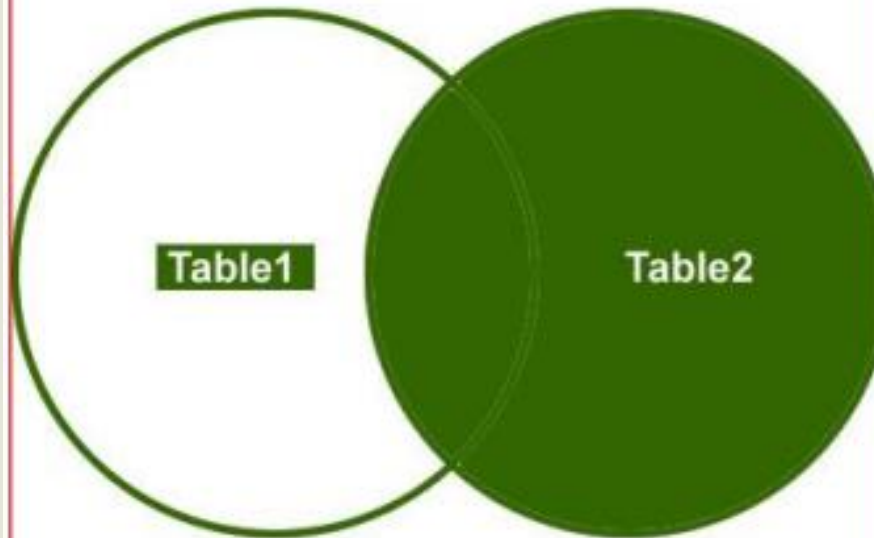




JOIN

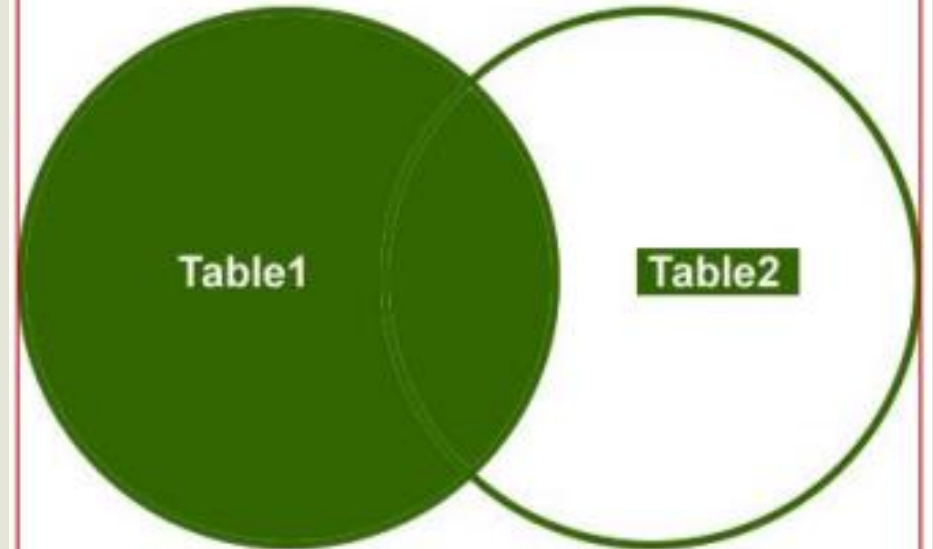


RIGHT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
RIGHT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

LEFT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
LEFT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```



TRANSACTION-SQL



یک زبان پرس و جو استاندارد است که برای واکنشی اطلاعات از آن استفاده میشود.
دارای قابلیت های بیشتری نسبت به SQL است:
- برنامه نویسی رویه ای
- تعریف متغیر محلی

زبان T-SQL امکان تعریف متغیرها برای استفاده در رویه های پرس و جو فراهم می آورد.

تعریف متغیر: یکی از اشیا در SQL Server است که برای حمل و نگهداری مقادیر مفرد به کار می روند.



TRANSACTION-SQL



برای تعریف متغیر از علامت @ استفاده میشود.
برای تعریف یک متغیر از قالب زیر استفاده میکنیم.

Declare @VariableName

برای مقدار دهی به متغیر ها از یکی از قالب های زیر استفاده می شود.

Select @VariableName = Value
Set @VariableName = Value

مثال استفاده از متغیرها

```
declare @student_name varchar(20);  
set @student_name='mary';  
select @student_name;
```

Results		Messages	
	(No column name)		
1	mary		

```
declare @name varchar(30);  
select @name= names from session3_DB.dbo.t1  
where studentid=8831033;  
  
select @name
```

	Column Name	Data Type	Allow Nulls
►	names	varchar(20)	<input checked="" type="checkbox"/>
	id	int	<input type="checkbox"/>
	familyname	varchar(30)	<input type="checkbox"/>

names	id	familyname	studentid
maryam	8	ghorbany	8831033
flor	9	nazary	9091312

	(No column name)
1	maryam

مثال استفاده از متغیرها

```
declare @name varchar(30);  
select @name= names from session3_DB.dbo.t1  
  
select @name;
```

	(No column name)
1	flor



TRANSACTION-SQL



زبان کنترل جریان بخشی از دستورات T-SQL را در بر میگیرد که جریان اجرای دستورات را در یک بلوک دستورات ، یک User Defined Function و یا یک Stored Procedure بر عهده میگیرد.

بدون استفاده از دستورات این زبان، دستورات زبان T-SQL به صورت ترتیبی و پشت سر هم اجرا می شوند. با استفاده از این دستورات امکان تعریف شرط، حلقه ، پرش از دستورات و نظایر آن فراهم می شود. کنترل جریان از دستورات زیر تشکیل شده است:

- | | |
|----------------------|------------------|
| ▪ BEGIN...END | IF...ELSE |
| ▪ WHILE | BREAK |
| ▪ RETURN | GOTO |
| ▪ WAITFOR | CONTINUE |



TRANSACTION-SQL



دستور IF ... ELSE :

عبارت شرطی If

کد دستورات مورد نظر

[ELSE] [کد دستورات مورد نظر]

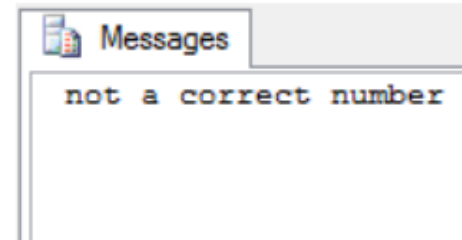
```
declare @number int;
```

```
set @number=12;
```

```
if @number=10
```

```
print '10';
```

```
else print 'not a correct number';
```





TRANSACTION-SQL



```
declare @number int;
```

```
set @number=12;
```

```
if @number=10
```

```
insert into
```

```
session3_DB.dbo.tblboat(boatname,boatcolor,boatrank)values('R4','black',13);
```

```
else insert into
```

```
session3_DB.dbo.tblboat(boatname,boatcolor,boatrank)values('R5','gray',9);
```

```
select *from session3_DB.dbo.tblboat;
```

	boatname	boatcolor	boatrank
1	R1	red	6
2	R2	blue	10
3	R3	Green	12
4	R5	gray	9

دستور IF ... ELSE :



TRANSACTION-SQL



While عبارت شرطی مضمون در حلقه

[Begin]

که دستورات مورد نظر

[End]

```
declare @number int;
```

```
set @number=100;
```

```
while @number>95
```

```
begin
```

```
    print @number;
```

```
    set @number=@number-1;
```

```
end
```

دستور WHILE:

```
100  
99  
98  
97  
96  
.
```




TRANSACTION-SQL



دستور CASE:

مقدار خاصی را بر اساس شرایط تعریف شده، در دستور select باز میگرداند.
دارای دو قالب زیر است:

حالت اول:

Case	عبارت		
	When	مقدار اولیه	Then
	...		
	[Else]		
End	[As	نام]	

حالت دوم:

Case			
	When	عبارت شرطی	Then
	...		
	[Else]		
End	[As	نام]	



TRANSACTION-SQL



boatname	boatcolor	boatrank
R1	red	6
R2	blue	10
R3	Green	12
R5	gray	9

مثال دستور CASE:

```
select boatcolor,bname=(case boatname  
    when 'R1' then 'AAAA'  
    when 'R2' then 'BBBB'  
    when 'R3' then 'CCCC'  
    when 'R5' then 'DDDD'  
    else 'others'  
end)
```

	boatcolor	bname
1	red	AAAA
2	blue	BBBB
3	Green	CCCC
4	gray	DDDD



TRANSACTION-SQL



WAITFOR {DELAY 'time' | TIME 'time'}

دستور WAITFOR:

```
--Pause for ten seconds  
WAITFOR delay '000:00:10';  
  
PRINT 'Done';  
  
--Pause until a certain time  
WAITFOR time '12:00:00';  
  
PRINT 'It is noon';
```



TRANSACTION-SQL



دستور OUTPUT:

این عبارت به همراه دستورات تغییر اطلاعات به کار می رود مانند Delete,update,insert

```
INSERT|DELETE|UPDATE ...  
OUTPUT {DELETED | INSERTED | from_table_name} . { * | column_name } [ ,...n ]  
INTO { @table_variable | output_table } [WHERE ...]
```

دو جدول **deleted** و **inserted** وجود دارند که دارای ساختار مشابه جدول اصلی هستند و می توان تمام و یا بعضی از فیلدها را انتخاب کرد. جدول **inserted** حاوی مقادیر جدید و به روز شده است و جدول **deleted** مقادیر قبلی را دارد.



TRANSACTION-SQL



مثال دستور OUTPUT:

میخواهیم به قایق هایی که رتبه کم تر از ۱۰ دارند دو رتبه اضافه کنیم و سپس لیستی از رتبه های فعلی و قبلی رکورد های تغییر کرده را برگردانیم:

	boatname	boatcolor	boatrank
	R1	red	4
	R2	blue	8
	R3	Green	12

```
declare @temp table(
```

```
nameb varchar(20),
```

```
color varchar(20),
```

```
old_rank int,
```

```
new_rank int);
```

```
update session3_DB.dbo.tblBoat
```

```
set boatrank=boatrank+2
```

```
output inserted.boatname,inserted.boatcolor,deleted.boatrank,inserted.boatrank
```

```
into @temp
```

```
where boatrank<10;
```

```
select *from @temp;
```

	nameb	color	old_rank	new_rank
1	R1	red	4	6
2	R2	blue	8	10



دستور کار

۱- جدول زیر را با استفاده از دستورات SQL ایجاد کرده و داده های داده شده را در آن وارد کنید. (P-ID) از نوع auto increment بوده و زوج (نام ، نام خانوادگی) برابر کلید اصلی هستند.

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Tom	Vingvn 23	Stavanger

جدول persons

الف) افراد را برحسب نام خانوادگی به ترتیب صعودی مرتب کنید.

ب) در یک transaction فیلد جدید شماره تلفن را از نوع nvarchar اضافه کنید. شماره تلفن ها حتما باید با پیش شماره ی ۰۰۱ شروع شوند(تعریف قید) و شماره تلفن های دلخواهی را به ۴ داده ی درون جدول نسبت دهید(طول شماره تلفن های اضافه شده برابر باشد).

پ) با استفاده از دستور case نام، نام خانوادگی و آدرس افراد را نشان دهید به طوری که هر آدرس فعلی(نام خیابان)، نام میدان و شماره ی پلاک را شامل شود(این موارد را به دلخواه وارد کنید).

ت) داده ی زیر را با شماره تلفن دلخواه وارد جدول کرده و ۳ فیلد اول جدول را به گونه ای نمایش دهید که برحسب نامشان به صورت صعودی مرتب شده اند.(در یک transaction)

7	Tjessem	Jakob	Nissestien 67	Sandnes
---	---------	-------	---------------	---------

توجه داشته باشید که P-ID داده ی جدید وارد شده برابر ۷ است.

ث) اشخاصی را که شهری زندگی آن ها با حرف "S" شروع می شود را پس از ده ثانیه مشخص کنید.

ج) روالی بنویسید که در آن متغیر temp از نوع int آخرین مقدار P_ID در جدول را بگیرد، و به آن تعداد عبارت "okay" چاپ شود.

چ) داده ی زیر با شماره تلفن ۰۰۱۱۲۳۴۵۶۷ را در نظر بگیرید، در صورتی که شماره تلفن آن از Tjessem کوچک تر است، P-ID آن را ۶ قرار داده و در غیر این صورت P-ID را ۸ قرار دهید.

taylor	Jackson	Nisseisten87	Sandnes
--------	---------	--------------	---------

۲- جدول زیر را ایجاد کرده و داده های زیر را وارد کنید (کلید اصلی = student_id)

name	student_id	grade
R1	8831047	12
R2	8831043	10
R3	8831031	15
R4	8831051	16
R5	8831012	11

می خواهیم به کسانی که نمره ی کمتر از ۱۵ دارند، ۲ نمره اضافه کنیم، به طوری که نمره ی جدید و نمره ی قدیمی آن ها نیز نمایش داده شود. (راهنمایی: استفاده از output)