Nesneye Yönelik Programlama

HAFTA 13

T-Sql İşlemleri

- Select
  - Tablo/tablolardaki verileri getirmemizi sağlayan komuttur.
- İnsert
  - Bir tabloya veri eklememizi sağlayan bir komuttur.
- Update
  - Bir tablodaki herhangi bir veriyi güncellememizi sağlayan komuttur
- Delete
  - Bir tablodaki verileri silmemizi sağlayan komuttur.

SELECT

- Select * from table_Name
  - Employee tablosundaki verileri getiren sorgu:
    - Select * from tbl_Employee
- Select columnName,... From table_Name
  - Employee tablosundaki id ve isim bilgilerini getiren sorgu:
    - Select id, name FROM tbl_Employee

Top

- Select Top (count) * from table_Name
  - Employee tablosundaki ilk 3 çalışanın bilgilerini getiren sorgu:
    - Select top 3 * from tbl_Employee

Where

- Select * from table_Name Where Statemen(s)
  - Employee tablosunda id’si 2 olan çalışanın bilgilerini getiren sorgu:
    - Select * FROM tbl_Employee Where id=2
  - Employee tablosunda maaşı 3000’den yüksek olan çalışanları getiren sorgu:
    - Select * from tbl_Employee Where salary>3000
  - 2019 yılından bugüne işe başlayan çalışanları getiren sorgu:
    - select * from tbl_Employee where StartDate>='01.01.2019'

Koşul İfadeleri:=, >,<,>=,<=, <>, !=, LIKE

- Employee tablosunda id'si 2 olan çalışanın bilgilerini getiren sorgu:
  - Select * FROM tbl_Employee Where id=2
- Employee tablosunda maaşı 3000'den yüksek olan çalışanları getiren sorgu:
  - Select * from tbl_Employee Where salary>3000
- 2019 yılından bugüne işe başlayan çalışanları getiren sorgu:
  - Select * from tbl_Employee where StartDate>='01.01.2019'
- 2 numaralı departmanda çalışan çalışanları getiren sorgu:
  - Select * from tbl_Employee where department_Id!=2
  - Select * from tbl_Employee where department_Id<>2
- İsminin içinde mobil geçen projeleri getiren sorgu:
  - Select * from tbl_project where name Like 'mobil'
  - Select * from tbl_project where name Like '%mobil'
  - Select * from tbl_project where name Like 'mobil%'
  - Select * from tbl_project where name Like '%mobil%'

AND, OR, NOT

- SELECT columnName, ... FROM table_name WHERE condition1 AND condition2 AND condition3 ...;
  - İşe başlama tarihi 01.01.2019 ‘dan büyük eşit ve maaşı 3000’den yüksek olan çalışanlar:
    - select * from tbl_Employee where startDate>='01.01.2019‘ and salary>3000

- SELECT columnName, ... FROM tbl_name WHERE condition1 OR condition2
  - İşe başlama tarihi 01.01.2019 ‘dan büyük veya maaşı 3000’den yüksek olan çalışanlar:
    - select * from tbl_Employee where startDate>='01.01.2019' or salary>3000

- SELECT columnName, ... FROM tbl_name WHERE NOT Condition
  - Biten projelerin bilgileri:
    - Select * FROM tbl_project Where finishDate is NOT NULL

DISTINCT

- Select distinct columnName from table_Name

  - Üzerinde çalışılan projelerin kodları:
    - select distinct project_Id from tbl_ProjectEmployee

ORDER BY

- SELECT columnName, ... FROM table_name ORDER BY column1, column2, ... ASC|DESC
  - İşe başlama tarihe göre personellerin sıralı listesi
    - select * from tbl_Empolyee order by startDate
  - İşe başlama tarihi 01.01.2019'dan itibaren çalışanların isme göre artan sıralı gösterimi
    - select * from tbl_ Empolyee where startDate>='01.01.2019' order by name asc
  - İşe başlama tarihi 01.01.2019'dan itibaren çalışanların isme göre azalan sıralı gösterimi
    - select * from tbl_ Empolyee where startDate>='01.01.2019' order by name desc

GROUP BY

- SELECT columnName,... FROM table_name WHERE condition GROUP BY column_name(s) ORDER BY column_name(s);

  - Hangi birimde kaç adet personel çalışmaktadır?
    - select department_Id, COUNT(id) AS NumberOfEmployee from tbl_Employee group by department_Id

Matematiksel T-Sql ifadeleri

- ABS: Mutlak değer
  - select ABS(-3)
- CEILING/FLOOR/ROUND: Yuvarlama işlemi
  - select CEILING(1.34)
  - select FLOOR(1.34)
  - select ROUND(1.34,1)
- RAND: [0,1] arasında rastgele değer üretir
  - select RAND()
- SQRT: Karekök alma
  - select SQRT(4)

Hesaplama Yapan T-Sql İfadeleri

- Count
  - select COUNT(id) from tbl_Employee
- Max
  - select MAX(id) from tbl_Employee
- Min
  - select min(id) from tbl_Employee

Metin işlemleri için T-Sql İfadeleri

- CHAR
  - select CHAR(65)
- CHARINDEX
  - select CHARINDEX('l','ali veli',1)
- LEFT
  - Select LEFT('ali veli',3)
- RIGHT
  - select RIGHT('ali veli',2)
- LEN
  - select LEN('ali veli')
- UPPER
  - select UPPER('ali veli‘)
- LOWER
  - select LOWER('ALİ VELİ')
- LTRIM
  - select LTRIM('   ali veli')
- RTRIM
  - select RTRIM('ali veli   ')
- REPLACE
  - select REPLACE('ali veli','veli',‘yılmaz')
- REVERSE
  - select REVERSE('ali')
- SUBSTRING
  - Select SUBSTRING('ali veli',3,2)

Tarih: T-Sql İfadeleri

- GETDATE()
  - select GETDATE()
  - select CONVERT(varchar,GETDATE(),105)
- DATEPART
  - select DATEPART(wk,getdate())/ select DATEPART(dy,getdate()) / select DATEPART(m,getdate())
- DAY/MONTH/YEAR
  - select DAY(getdate()) /select DAY(Convert(datetime,'18.12.2018',104))
  - select MONTH(GETDATE())
  - select YEAR(GETDATE())
- DATEADD
  - select DATEADD(day,3,GETDATE())
  - select DATEADD(MONTH,3,GETDATE())
  - select DATEADD(YEAR,3,GETDATE())

Tarih: T-Sql İfadeleri

- DATEDIFF
  - select DATEDIFF(day,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))
  - select DATEDIFF(MONTH,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))
  - select DATEDIFF(year,Convert(datetime,'16.10.2019',104),convert(datetime,'1.11.2019',104))
- DATENAME
  - SELECT DATENAME(M, '2017/08/25') AS DatePartString;

IN - NOT IN

- IN
  - İsmi “Ali”, “Veli”, “Deniz” olan personelin bilgileri:
    - select * from tbl_Employe where name in (‘Ali',‘Veli',‘Deniz')

- NOT IN
  - İsmi “Ali”, “Veli”, “Deniz” dışındaki personelin bilgileri:
    - select * from tbl_Employe where name not in (‘Ali',‘Veli',‘Deniz')

Çoklu Sorgulamalar

- IN
  - select * from tbl_Employee where department_Id in (select id from tbl_Department)
- EXISTS
  - En az bir adet proje yapılmış proje türleri:
  - select * from tbl_ProjectType where EXISTS(select ProjectType_Id from tbl_Project where ProjectType_Id=tbl_ProjectType.type_Id)

NOT EXISTS

- Hiç proje yapılmayan proje türleri:
- select * from tbl_ProjectType where NOT EXISTS(select ProjectType_Id from tbl_Project where ProjectType_Id=tbl_ProjectType.type_Id)

Çoklu Sorgular

- AS
- Çalışan bilgilerinin yanından bölüm isimlerinin yazılması:
  - select *, (select name from tbl_Department where id=Department_Id)AS DepartmentName from tbl_Employee

Çoklu Sorgular

- Klasik Join
- Inner Join
- Outer Join
- Cross Join

Klasik Join

- İç içe geçmiş sorgu=Klasik Join

- select * from tbl_Employee where department_Id in (select id from tbl_Department)

INNER JOIN

- Select * from table_Name1 [INNER] JOIN table_Name2
  ON tbl_Name1 .ColumnName= tbl_Name2.ColumnName

- select * tbl_Employee inner join tbl_Department ON tbl_Department.id= tbl_Employee .Department_Id

- select * tbl_Employee AS E inner join tbl_Department AS D ON D.id= E.Department_Id

OUTER JOIN

- LEFT
  - select * tbl_Employee LEFT join tbl_Department ON tbl_Department.id= tbl_Employee .Department_Id
- RIGHT
  - select * tbl_Employee RIGHT join tbl_Department ON tbl_Department.id= tbl_Employee .Department_Id
- FULL
  - select * tbl_Employee FULL join tbl_Department ON tbl_Department.id= tbl_Employee .Department_Id

CROSS JOIN

- SELECT * from table_Name1 CROSS JOIN table_Name2
- select * tbl_Employee CROSS join tbl_Department

TSQL JOIN TYPES

Created by Steve Stedman


```
SELECT *
  FROM Table1;

SELECT *
  FROM Table2;
```

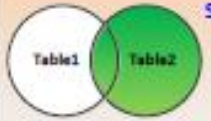
SELECT from two tables


```
SELECT *
  FROM Table1 t1
INNER JOIN Table2 t2
  ON t1.fk = t2.id;
```

INNER JOIN


```
SELECT *
  FROM Table1 t1
LEFT OUTER JOIN Table2 t2
  ON t1.fk = t2.id;
```

LEFT OUTER JOIN

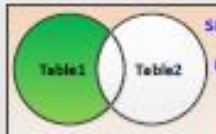



```
SELECT *
  FROM Table1 t1
RIGHT OUTER JOIN Table2 t2
  ON t1.fk = t2.id;
```

RIGHT OUTER JOIN


```
SELECT *
  FROM Table1 t1
WHERE EXISTS (SELECT 1
                FROM Table2 t2
               WHERE t1.fk = t2.id
              );
```

SEMI JOIN

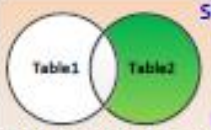



```
SELECT *
  FROM Table1 t1
WHERE NOT EXISTS (SELECT 1
                    FROM Table2 t2
                   WHERE t1.fk = t2.id
                  );
```

ANTI SEMI JOIN

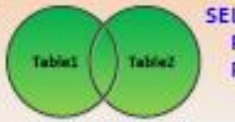

```
SELECT *
  FROM Table1 t1
LEFT OUTER JOIN Table2 t2
  ON t1.fk = t2.id
WHERE t2.id IS NULL;
```

LEFT OUTER JOIN with exclusion
– replacement for a NOT IN



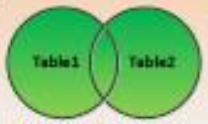

```
SELECT *
  FROM Table1 t1
RIGHT OUTER JOIN Table2 t2
  ON t1.fk = t2.id
WHERE t1.fk IS NULL;
```

RIGHT OUTER JOIN with exclusion
– replacement for a NOT IN



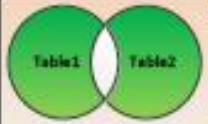

```
SELECT *
  FROM Table1 t1
FULL OUTER JOIN Table2 t2
  ON t1.fk = t2.id;
```

FULL OUTER JOIN




```
SELECT *
  FROM Table1 t1
CROSS JOIN Table2 t2;
```

CROSS JOIN, the Cartesian product




```
SELECT *
  FROM Table1 t1
  FULL OUTER JOIN Table2 t2
    ON t1.fk = t2.id
 WHERE t1.fk IS NULL
    OR t2.id IS NULL;
```

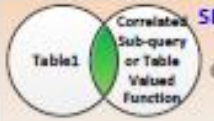
FULL OUTER JOIN with exclusion


```
SELECT *
  FROM Table1 t1
INNER JOIN Table2 t2
  ON t1.fk >= t2.id;
```

NON-EQUI INNER JOIN

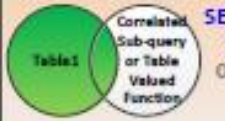
TSQL JOIN TYPES

Created by Steve Stedman



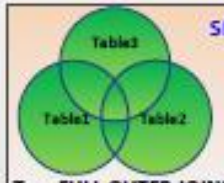

```
SELECT *
  FROM Table1 t1
CROSS APPLY
      [dbo].[someTVF](t1.fk)
      AS t;
```

CROSS APPLY




```
SELECT *
  FROM Table1 t1
OUTER APPLY
      [dbo].[someTVF](t1.fk)
       AS t;
```

OUTER APPLY

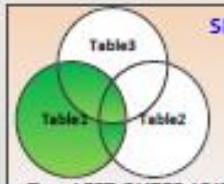



```
SELECT *
  FROM Table1 t1
FULL OUTER JOIN Table2 t2
  ON t1.fk = t2.id
FULL OUTER JOIN Table3 t3
  ON t1.fk_table3 = t3.id;
```

Two FULL OUTER JOINS

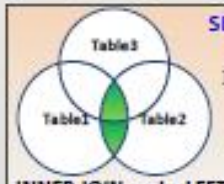

```
SELECT *
  FROM Table1 t1
INNER JOIN Table2 t2
  ON t1.fk = t2.id
INNER JOIN Table3 t3
  ON t1.fk_table3 = t3.id;
```

Two INNER JOINs




```
SELECT *
  FROM Table1 t1
LEFT OUTER JOIN Table2 t2
  ON t1.fk = t2.id
LEFT OUTER JOIN Table3 t3
  ON t1.fk_table3 = t3.id;
```

Two LEFT OUTER JOINS

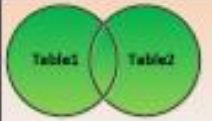



```
SELECT *
  FROM Table1 t1
INNER JOIN Table2 t2
  ON t1.fk = t2.id
LEFT OUTER JOIN Table3 t3
  ON t1.fk_table3 = t3.id;
```

INNER JOIN and a LEFT OUTER JOIN

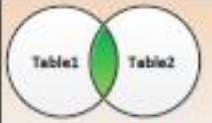

```
SELECT fk as id
  FROM Table1
EXCEPT
SELECT ID
  FROM Table2;
```

EXCEPT




```
SELECT fk as id
  FROM Table1
UNION
SELECT ID
  FROM Table2;
```

UNION




```
SELECT fk as id
  FROM Table1
INTERSECT
SELECT ID
  FROM Table2;
```

INTERSECT

Sample Schema

Table 1 (People)

	id	Name	fk	fk_table3
1	1	Steve	1	NULL
2	2	Aaron	3	NULL
3	3	Mary	2	NULL
4	4	Fred	1	NULL
5	5	Anne	5	NULL
6	6	Beth	5	1
7	7	Johnny	NULL	1
8	8	Karen	NULL	2

Table 2 (Favorite Colors)

	id	FavoriteColor
1	1	red
2	2	green
3	3	blue
4	4	pink
5	5	purple
6	6	mauve
7	7	orange
8	8	yellow
9	1	indigo

Table 3 (Favorite Foods)

	id	delelvalue
1	1	Pizza
2	2	Burger
3	3	Sushi

Note: Column names are very generic to simplify the sample queries.
Foreign keys are
Table1.fk -> Table2.id
Table2.fk_table3 -> Table3.id

UNION

- SELECT * FROM tbl_Name1

  UNION

  SELECT * FROM tbl_Name2

- select * from tbl_ProjectType

  union

  select * from tbl_Department

INSERT

- Insert into table_name (columnName,...) values (valuesofData,...)
  - insert into tbl_Department(id, name) values (1,‘Analiz’)
  - insert into tbl_Employee (name,surname,startDate) values (‘Ali', null,’03.05.2020’)

- Insert into table_name values (valuesofData,...)
  - insert into tbl_Department values (2,‘Tasarım’)
  - insert into tbl_Employee values (‘Ali',null,null, null,’03.05.2020’,2)

- Select columnName,... İNTO new_Tbl from tbl_name
  - select * into tbl_tmpDepartment from tbl_Department
  - select id,name into tbl_tmpDepartment from tbl_Department

UPDATE

- Update table_name Set

  columnName=newValue

  Where statement

- update tbl_Employee

  set Department_Id=1

  where ad='Ali'

DELETE

- DELETE FROM table_Name WHERE statement(s)
  - Delete from tbl_Employee where [id]=5
  - Delete from tbl_Project where name=‘Mobil Ağ Yönetimi‘

Örnek Sorgular