

## İNTERNET TABANLI PROGRAMLAMA- 12.ders

### TEMEL SQL KULLANIMLARI

Örnekleri uygulamak için aşağıdaki gibi bir altyapı hazırlayalım.

BİLGİLERİ GETİR		
Column0	Column1	Column2
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc
abc	abc	abc

```
using System.Data.OleDb;
```

### SELECT

VT den bilgileri getirmemizi sağlar.

```
string Sorgu = "SELECT * FROM Ogrenciler";
```

BİLGİLERİ GETİR							
ID	BolumID	OgrNo	Ad	Soyad	DT	Vize	Final
12	1	123	Ali	Su	11.09.1988 00:00:00	20	56
13	1	124	Oya	Ay	11.11.1988 00:00:00	56	23
14	5	125	Can	Ak	21.10.1990 00:00:00	78	45
15	2	126	Ümit	Mutlu	19.09.1991 00:00:00	23	56
16	2	127	Emre	Ara	12.03.1991 00:00:00	45	78
17	3	128	Ebru	Polat	21.03.1990 00:00:00	56	89
18	2	129	Onur	Doğan	24.04.1993 00:00:00	34	99
19	3	140	Murat	Çakır	01.09.1990 00:00:00	22	88
20	1	155	Merve	Şan	08.09.1992 00:00:00	34	67

### SELECT (Alan Adlarına Göre Getirme)

```
string Sorgu = "SELECT Ad,Soyad,Vize,Final FROM Ogrenciler";
```

BİLGİLERİ GETİR			
Ad	Soyad	Vize	Final
Ali	Su	20	56
Oya	Ay	56	23
Can	Ak	78	45
Ümit	Mutlu	23	56
Emre	Ara	45	78
Ebru	Polat	56	89
Onur	Doğan	34	99
Murat	Çakır	22	88
Merve	Şan	34	67

## SELECT DISTINCT

Tekil olarak bilgileri getirir. Aynı isimde iki kişi varsa bir tane getirir.

```
string Sorgu = "SELECT DISTINCT Ad FROM Ogrenciler ";
```

BİLGİLERİ GETİR	
Ad	
Ali	
Can	
Ebru	
Murat	
Onur	
Oya	
Ümit	

Ad
Ali
Oya
Can
Ümit
Ali
Ebru
Onur
Murat
Ali

```
string Sorgu = "SELECT DISTINCT Ad, Soyad FROM Ogrenciler ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Ali	Şan
Can	Ak
Murat	Çakır
Onur	Doğan
Oya	Polat
Oya	Şan
Ümit	Mutlu

Ad	Soyad
Ali	Çakır
Oya	Şan
Can	Ak
Ümit	Mutlu
Ali	Çakır
Oya	Polat
Onur	Doğan
Murat	Çakır
Ali	Şan

## SELECT, ORDER BY

Bilgileri sıralayarak getirir. ASC kullanılırsa (A-Z) ye sıralar. DESC kullanılırsa (Z-A) sıralar.

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler ORDER BY Ad ASC ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Alişan	Şan
Aliya	Yılmaz
Ayşe	Çakır
Can	Ak
Murat	Çakır
Onur	Doğan
Oya	Polat
Ümit	Mutlu

Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Aliya	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Alişan	Şan

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler ORDER BY Soyad DESC ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Aliya	Yılmaz
Alişan	Şan
Oya	Polat
Ümit	Mutlu
Onur	Doğan
Murat	Çakır
Ayşe	Çakır
Ali	Çakır
Can	Ak

## SELECT, TOP

Belli sayıda kayıdı getirmek için kullanılır. TOP 3 yazılırsa 3 tane kayıt getir. TOP 50 PERCENT yazılırsa kayıtların %50 sini getirir.

```
string Sorgu = "SELECT TOP 3 * FROM Ogrenciler ";
```

BİLGİLERİ GETİR							
ID	BolumID	OgrNo	Ad	Soyad	Vize	Final	DT
12	1	123	Ali	Çakır	98	56	11.09.1988 00:00:00
13	1	124	Ayşe	Çakır	100	23	11.11.1988 00:00:00
14	5	125	Can	Ak	78	45	21.10.1990 00:00:00

```
string Sorgu = "SELECT TOP 3 Ad, Soyad FROM Ogrenciler ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak

```
string Sorgu = "SELECT TOP 50 PERCENT Ad, Soyad FROM Ogrenciler ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Aliya	Yılmaz
Aliya	Yılmaz

Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Aliya	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Alişan	Şan

% 50 sini getirdi.

## WHERE

```
string Sorgu = "SELECT Ad,Soyad,Vize FROM Ogrenciler WHERE Vize>60";
```

BİLGİLERİ GETİR		
Ad	Soyad	Vize
Ali	Su	90
Can	Ak	78
Murat	Çakır	88

```
string Sorgu = "SELECT Ad,Soyad FROM Ogrenciler WHERE Soyad='Çakır'";
```

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Murat	Çakır

Where ifadesi aşağıdaki operatörlerle kullanılabilir.

Operator	Tanım
=	Eşit
<>	Eşit Değil
>	Büyük
<	Küçük
>=	Büyük yada Eşit
<=	Küçük yada Eşit
BETWEEN	Arasında
LIKE	Verilen bir desene Benzer
IN	Bir kolon için çok sayıda ihtimali belirlemek için

## WHERE, AND

Tüm ifadeler doğru (True) ise getirecektir.

```
string Sorgu = "SELECT Ad, Soyad, Vize FROM Ogrenciler WHERE Ad='Ali' AND Vize=100";
```

BİLGİLERİ GETİR		
Ad	Soyad	Vize
Ali	Ara	100
Ali	Şan	100

Ad	Soyad	Vize
Ali	Çakır	98
Oya	Ay	55
Can	Ak	78
Ümit	Mutlu	23
Ali	Ara	100
Ebru	Polat	56
Onur	Doğan	34
Murat	Çakır	18
Ali	Şan	100

## WHERE, OR

İfadelerden herhangi biri doğru (True) ise getirecektir.

```
string Sorgu = "SELECT Ad, Soyad, Vize FROM Ogrenciler WHERE Ad='Ali' OR Vize=100";
```

BİLGİLERİ GETİR		
Ad	Soyad	Vize
Ali	Çakır	98
Oya	Ay	100
Ali	Ara	56
Ali	Şan	100

Ad	Soyad	Vize
Ali	Çakır	98
Oya	Ay	100
Can	Ak	78
Ümit	Mutlu	23
Ali	Ara	56
Ebru	Polat	56
Onur	Doğan	34
Murat	Çakır	18
Ali	Şan	100

## WHERE, AND, OR

Parantez içerisinde kullanarak karmaşık sorgulamalar yapabiliriz.

```
string Sorgu = "SELECT BolumID, Ad, Soyad FROM Ogrenciler WHERE BolumID=1 AND (Ad='Ali' OR Ad='Oya')";
```

BİLGİLERİ GETİR		
BolumID	Ad	Soyad
1	Ali	Çakır
1	Oya	Şan
1	Ali	Şan

## WHERE, LIKE

Verilen desene benzeyen kayıtları getirir.

```
string Sorgu = "SELECT Ad,Soyad,Vize FROM Ogrenciler WHERE Soyad LIKE 'A%' ";
```

ID	TCKimlikNo	BolumID	Ad	Soyad	Yas	BasariOrtalamasi
1	111111	1	Aliye	SU	23	56,78
4	222222	1	Ayşe	EŞ	34	34,67
7	444444	1	Ata	İZ	22	34,67
9	666666	4	Ahmet	KARA	49	12,45
13	232322	3	Aliye	SU	17	34,67

BilgileriGetir

**Ek bilgi:** Eşittir ile (=) ile "LIKE" arasındaki fark; Eşittir olursa her iki tarafında tipi aynı olması gerekir. LIKE da ise böyle bir şart aranmaz.

```
string Sorgu = "SELECT Ad,Soyad,Vize FROM Ogrenciler WHERE Vize = '34' ";
```

(hata verir, integer ve String eşittir ile karşılaştırılmaz)

```
string Sorgu = "SELECT Ad,Soyad,Vize FROM Ogrenciler WHERE Vize LIKE '34' ";
```

(hata vermez, integer ve string eşitlenmiş fakat arada Like var)

```
string Sorgu = "SELECT Ad,Soyad,Vize FROM Ogrenciler WHERE Vize = 34 ";
```

(hata vermez, her iki taraf integer)

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler WHERE Ad LIKE '%A%' ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Ali	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Ali	Şan

Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Ali	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Ali	Şan

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler WHERE Ad LIKE 'A__' ";
```

Not: A\_\_ şeklinde iki tane alt çizgi var.

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Ali	Yılmaz
Ali	Şan

Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Ali	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Alişan	Şan

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler WHERE Ad LIKE '%u_' ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Onur	Doğan

Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Ali	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Alişan	Şan

Belli harler ile başlayan soyisimleri getirirken [] içerisine harfleri yazıyoruz ve sonuna % işareti ekliyoruz.

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler WHERE Soyad LIKE '[aş]%' ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Can	Ak
Alişan	Şan

Ad	Soyad
Ali	Çakır
Ayşe	Çakır
Can	Ak
Ümit	Mutlu
Aliya	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Alişan	Şan

## WHERE, IN

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler WHERE Soyad IN ('Çakır', 'Polat', 'Şan') ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Ali	Çakır
Oya	Polat
Murat	Çakır
Alişan	Şan

Ad	Soyad
Ali	Çakır
Ayşe	Ekim
Can	Ak
Ümit	Mutlu
Aliya	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Alişan	Şan

Soyadı ('Çakır', 'Polat', 'Şan') olan kişileri getirdi.

## WHERE, BETWEEN ... AND

Bu ifademizi bizim belirleyeceğimiz aralığa göre kayıtlarımızı süzmek için kullanırız.

```
string Sorgu = "SELECT Ad,Soyad, Vize FROM Ogrenciler WHERE Vize BETWEEN 0 AND 60";
```

Aynı ifade aşağıdaki gibi kullanılabilir.

```
string Sorgu = "SELECT Ad,Soyad,Vize FROM Ogrenciler WHERE Vize >=0 AND Vize <=60";
```

BİLGİLERİ GETİR		
Ad	Soyad	Vize
Oya	Ay	60
Ümit	Mutlu	23
Emre	Ara	45
Ebru	Polat	56
Onur	Doğan	34
Merve	Şan	0



Aşağıdaki ifade Harf sırasına göre Can ile Oya arasındaki kişileri getirecektir.

```
string Sorgu = "SELECT Ad, Soyad FROM Ogrenciler WHERE Ad BETWEEN 'Can' AND 'Oya' ";
```

BİLGİLERİ GETİR	
Ad	Soyad
Can	Ak
İsa	Mutlu
Oya	Polat
Onur	Doğan
Murat	Çakır

Ad	Soyad
Ali	Çakır
Zeynep	Ekim
Can	Ak
İsa	Mutlu
Aliya	Yılmaz
Oya	Polat
Onur	Doğan
Murat	Çakır
Alişan	Şan

İsim sırasına göre Can ile Oya arasındaki kişileri getirdi.

## INSERT INTO

Bigileri kaydeder.

```
string Sorgu = "INSERT INTO Ogrenciler (Ad,Soyad, Yas) VALUES ('Lütfi','Kara',20)";
```

13	232322	3	Aliye	SU	17	34,67
14	777777	1	Hasan	Subaşı	21	12,45
15	565656	1	Hasan	Yas	22	12,45
16			Lütfi	Kara	20	
*	(New)					

## UPDATE

Bilgileri günceller

```
string Sorgu = "UPDATE Ogrenciler SET Ad='Murat' WHERE TckimlikNo=999999" ;
```

10	888888	4	İsmail	AYDIN	67	12,45
11	999999	5	Mustafa	TAŞ	18	56,78
12	132312	5	Sinan	Can	20	34,67

10	888888	4	İsmail	AYDIN	67	12,45
11	999999	5	Murat	TAŞ	18	56,78
12	132312	5	Sinan	Can	20	34,67

## DELETE

Bilgileri siler

```
string Sorgu = "DELETE * FROM Ogrenciler WHERE TckimlikNo=777777";
```

13	232322	3	Aliye	SU	17	34,67
14	777777	1	Hasan	Subaşı	21	12,45
15	565656	1	Hasan	Yas	22	12,45

12	132312	5	Sinan	Can	20	34,67
13	232322	3	Aliye	SU	17	34,67
15	565656	1	Hasan	Yas	22	12,45
16			Lütfi	Kara	20	

## GROUP BY

Veritabanımızdaki listelediğimiz kayıtları GROUP BY komutu ile istediğimiz bir kategoride gruplayabiliriz. Örnekleri sırayla deneyin.

```
string Sorgu = "SELECT SUM(Vize) AS VizeToplami FROM Ogrenciler ";
```

BİLGİLERİ GETİR	
<b>VizeToplami</b>	
504	

```
string Sorgu = "SELECT BolumID, SUM(Vize) AS VizeToplami FROM Ogrenciler GROUP BY BolumID ";
```

BİLGİLERİ GETİR	
<b>BolumID</b>	<b>VizeToplami</b>
1	180
2	102
3	144
5	78

```
string Sorgu = "SELECT BolumID, SUM(Vize) FROM Ogrenciler GROUP BY BolumID";
```

Button	
<b>BolumID</b>	<b>Expr1001</b>
1	76
2	68
3	56
5	78

```
string Sorgu = "SELECT BolumID, SUM(Vize) AS VizeToplami FROM Ogrenciler WHERE BolumID=1 GROUP BY BolumID";
```

BİLGİLERİ GETİR	
BolumID	Vize Toplamı
1	180

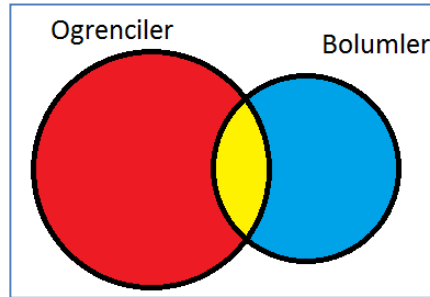
**Not:** Burada BolumID yerine BolumAdlarını getirmek gerekir. Fakat bu iki tabloyu birleştirmeyi gerektirir. Bu konu için InnerJoin konusuna bakınız.

## INNER JOIN

İki tablo arasında ortak bir sütun varsa bu sütun üzerinden iki tablodaki ortak bilgileri gösterir. Yani grafiğimizdeki sarı alanları getirecektir. Bilgiler görüntülenirken her iki tablodan da istediğimiz sütunu görüntüleyebiliriz. Örneğin aşağıdaki tablolarda Ad, Soyad Öğrenciler tablosundan, BolumAdi ise Bölümler tablosundan getirilmiştir.

```
string Sorgu = "SELECT Boluimler.BolumAdi, Ogrenciler.Ad, Ogrenciler.Soyad FROM Ogrenciler INNER JOIN Boluimler ON Ogrenciler.BolumID = Boluimler.ID";
```

BİLGİLERİ GETİR		
BolumAdi	Ad	Soyad
Mekatronik	Ali	Çakır
Mekatronik	Zeynep	Ekim
Metal	Can	Ak
Bilgisayar	İsa	Mutlu
Bilgisayar	Aliya	Yılmaz
Mekatronik	Oya	Polat
Bilgisayar	Onur	Doğan
Mekatronik	Alişan	Şan



ID	FakulteID	BolumAdi
1	1	Mekatronik
2	1	Bilgisayar
3	1	Elektronik
4	2	Tasarim
5	2	Metal
6	3	Tarih
7	3	Türk Dili

ID	BolumID	OgrNo	Ad	Soyad	Vize	Final
12	1	123	Ali	Çakır	98	56
13	1	124	Zeynep	Ekim	100	23
14	5	125	Can	Ak	78	45
15	2	126	İsa	Mutlu	23	56
16	2	127	Aliya	Yılmaz	56	78
17	1	128	Oya	Polat	56	89
18	2	129	Onur	Doğan	34	99
19	13	140	Murat	Çakır	18	88
20	1	155	Alişan	Şan	100	67

## Tablo Adlarında Alias (Takma Ad, Kısaltma) kullanımı

Bazen bu tip kullanımlarda Tablo isimleri çok yer kaplayabilir (ör: Öğrenciler.Ad). Burada her sütun adını yazarken hangi tablodan geldiğini anlatmak için tablo adları yazıldı. Bu tablo adları için Alias (Takma Ad) kullanabiliriz. Bu amaçla AS ifadesini ekleyeceğiz. Sorguyu buna göre tekrar yazarsak şu şekilde olacaktır.

```
string Sorgu = "SELECT Blm.BolumAdi, Ogr.Ad, Ogr.Soyad FROM Ogrenciler AS Ogr INNER JOIN Boluimler AS Blm ON Ogr.BolumID = Blm.ID";
```

ifadesi aynı sonucu verecektir.

Üç Tabloyu Birleştirme Örneği:

```
string Sorgu = "SELECT MID(Fakulteler.FakulteAdi,1,3), Ogrenciler.ID, Boluimler.BolumAdi,
Ogrenciler.Ad, Ogrenciler.Soyad, Ogrenciler.Vize " +
"FROM ((Ogrenciler INNER JOIN Boluimler ON Ogrenciler.BolumID = Boluimler.BolumID ) " +
"INNER JOIN Fakulteler ON Fakulteler.FakulteID = Boluimler.BolumID)";
```

Inner Join ve Where Kullanımı Örneği

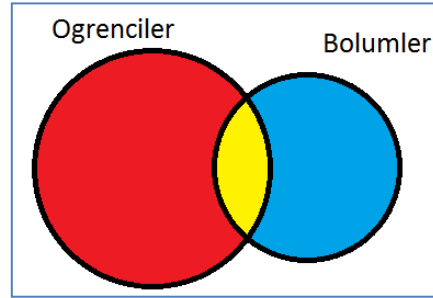
```
string Sorgu = "SELECT Fakulteler.FakulteAdi, Boluimler.BolumAdi FROM Boluimler " +
" INNER JOIN Fakulteler ON Boluimler.FakulteID = Fakulteler.FakulteID " +
" WHERE Boluimler.FakulteID = " + FakulteID3;
```

## LEFT JOIN

Inner Join kullanımında her iki tablodada ortak bilgiler var ise getirmişti. Bazen tabloların birinde diğer tablo ile ortak kayıt olmadığı bilgilerde olabilir. Bu bilgileride getirmek isteyebiliriz. FROM dan sonra Tablo isimlerini yazarken "Ogrenciler" tablosu solda, "Boluimler" tablosu ise sağda olduğu için LEFT JOIN kullanırsak Ogrenciler tablosundaki tüm bilgileri getirir. Yani Ogrenciler tablosunda bolum karşılığı olmayan kayıtlar var ise oda gelecektir. Grafiğimizdeki Kırmızı ve Sarı alanları getirecektir.

```
string Sorgu = "SELECT Blm.BolumAdi, Ogr.Ad, Ogr.Soyad FROM Ogrenciler AS Ogr LEFT
JOIN Boluimler AS Blm ON Ogr.BolumID = Blm.ID";
```

BİLGİLERİ GETİR		
BolumAdi	Ad	Soyad
Mekatronik	Ali	Çakır
Mekatronik	Zeynep	Ekim
Metal	Can	Ak
Bilgisayar	İsa	Mutlu
	Aliya	Yılmaz
Mekatronik	Oya	Polat
Bilgisayar	Onur	Doğan
	Murat	Çakır
Mekatronik	Alişan	Şan



ID	FakulteID	BolumAdi
1	1	Mekatronik
2	1	Bilgisayar
3	1	Elektronik
4	2	Tasarim
5	2	Metal
6	3	Tarih
7	3	Türk Dili
8	3	Coğrafya
* (New)		

ID	BolumID	OgrNo	Ad	Soyad	Vize	Final	DT
12	1	123	Ali	Çakır	98	56	11.09.1988
13	1	124	Zeynep	Ekim	100	23	11.11.1988
14	5	125	Can	Ak	78	45	21.10.1990
15	2	126	İsa	Mutlu	23	56	19.09.1991
16	12	127	Aliya	Yılmaz	56	78	12.03.1991
17	1	128	Oya	Polat	56	89	21.03.1990
18	2	129	Onur	Doğan	34	99	24.04.1993
19	11	140	Murat	Çakır	18	88	01.09.1990
20	1	155	Alişan	Şan	100	67	08.09.1992
* (New)							

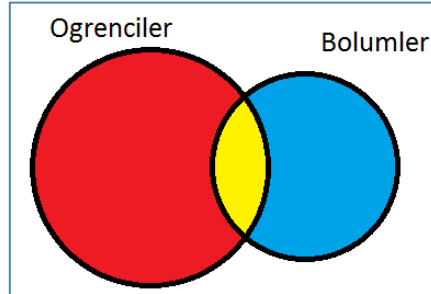
Dikkat edilirse Ogrenciler tablosunu olduğu gibi getirdi (Soldaki tablo olduğu için, Left Join) . Fakat bazı BolumID lerin karşılı Bolumler tabsounda yoktur. Burada 11 ve 12 ID li Bolumler, Bolumler tablosunda yoktur. Bu nedenle karşılar boş geldi.

## RIGHT JOIN

Aynı şekilde RIGT JOIN kullanırsa bu sefer Bolumler tablosu olduğu gibi gelir. Öğrenciler tablosunda karşılığı yok ise o kısımlar boş kalır. Grafikte sarı ve mavi alanları getirmiş olacaktır.

```
string Sorgu = "SELECT Blm.BolumAdi, Ogr.Ad, Ogr.Soyad FROM Ogrenciler AS Ogr RIGHT JOIN Bolumler AS Blm ON Ogr.BolumID = Blm.ID";
```

BİLGİLERİ GETİR		
BolumAdi	Ad	Soyad
Mekatronik	Ali	Çakır
Mekatronik	Zeynep	Ekim
Mekatronik	Oya	Polat
Mekatronik	Alişan	Şan
Bilgisayar	İsa	Mutlu
Bilgisayar	Onur	Doğan
Elektronik		
Tasarım		
Metal	Can	Ak
Tarih		
Türk Dili		
Coğrafya		



Bolumler tablosundaki tüm bilgileri getirdi. Fakat bazı bolumlerin karşılığı olan Öğrenciler olmadığı için bu kısımlar boş kaldı. Karşılı olan bölümler bulduğu öğrenci sayısı kadar tekrar etmiştir.

Not: Her iki tablodaki bilgileri karşılığı olsun olmasın getirmek için "Full Join" kullanılır fakat Access'de bu komut çalışmamıştır.

## SELECT, UNION

İki yada daha fazla SELECT ifadesini birleştirmek için kullanılır. Bunu aynı sütunlara sahip iki yada daha fazla tablodaki bilgileri altalta eklemek gibi de düşünebiliriz. Fakat burada her tablodaki (sorgu ifadesindeki) bilgileri Tekil olarak getirir. Aynı kayıt bir tablo içinde (sorgu içinde) iki defa geçiyorsa tekil olarak getirecektir.

```
string Sorgu = "SELECT BolumID, Ad,Soyad FROM Ogrenciler WHERE BolumID=1 UNION SELECT BolumID, Ad,Soyad FROM Ogrenciler WHERE BolumID=2";
```

BİLGİLERİ GETİR		
BolumID	Ad	Soyad
1	Ali	Su
1	Oya	Polat
1	Zeynep	Ekim
2	Ali	Su
2	Onur	Doğan

BolumID	OgrNo	Ad	Soyad
1	123	Ali	Su
1	124	Zeynep	Ekim
5	125	Can	Ak
2	126	Ali	Su
12	127	Aliya	Yılmaz
1	128	Oya	Polat
2	129	Onur	Doğan
11	140	Murat	Çakır
2	155	Ali	Su

Burada birinci SELECT kısmında (yani 1 numaralı bölümde) Ali Su adı bir kez geçiyor onu getirdi. İkinci SELECT kısmında (yani 2 numaralı bölümde) Ali Su ismi iki defa geçiyor fakat 1 kez yazdı. Burada ayrı SELECT sorgularını farklı tablolar mış gibi düşündük.

## SELECT, UNION ALL

Bu kullanımda iki tabloyu alt alta birleştirirken her aynı kayıtlardan varsa onların hepsini getirecektir.

BİLGİLERİ GETİR		
BolumID	Ad	Soyad
1	Ali	Su
1	Zeynep	Ekim
1	Oya	Polat
2	Ali	Su
2	Onur	Doğan
2	Ali	Su

BolumID	OgrNo	Ad	Soyad
1	123	Ali	Su
1	124	Zeynep	Ekim
5	125	Can	Ak
2	126	Ali	Su
12	127	Aliya	Yılmaz
1	128	Oya	Polat
2	129	Onur	Doğan
11	140	Murat	Çakır
2	155	Ali	Su

Burada ikinci sorgu kısmında (Select kısmına) ait tabloda iki tane Ali Su var. İkisini de getirdi.

## SELECT INTO

Bir tablodaki bilgileri başka bir tabloya aktarırken kullanılır. Daha çok tabloları yedeklemek (backup) için kullanılır. Başka bir veritabanındaki tablo içine de aktarılabilir.

```
string Sorgu = "SELECT * INTO Ogrenciler_Yedek FROM Ogrenciler";
```

ID	BolumID	OgrNo	Ad	Soyad	Vize	Final	DT
12	1	123	Ali	Su	98	56	11.09.1988
13	1	124	Zeynep	Ekim	100	23	11.11.1988
14	5	125	Can	Ak	78	45	21.10.1990
15	2	126	Ali	Su	23	56	19.09.1991
16	12	127	Aliya	Yılmaz	56	78	12.03.1991
17	1	128	Oya	Polat	56	89	21.03.1990
18	2	129	Onur	Doğan	34	99	24.04.1993
19	11	140	Murat	Çakır	18	88	01.09.1990
20	2	155	Ali	Su	100	67	08.09.1992
*	(New)						

Tablonun kopyasını başka bir veritabanı içerisine de atabiliriz. Tabii öncesinde yedekveritabanı oluşturulmuş olmalıdır. Bunun için başka sorgular da vardır.

## FONKSİYONLAR

### FORMAT()

Bilgileri görüntülerken belli formatlarda görüntülemek isteyebiliriz. Örneğin tarihleri görüntülerken saatleride yanında getiriyordu. Bunun yerine tarih sütununu belli bir formatta getirelim.

Önce eski halini bir görelim.

```
string Sorgu = "SELECT Ad,Soyad, DT FROM Ogrenciler";
```

BİLGİLERİ GETİR		
Ad	Soyad	DT
Ali	Su	11.09.1988 00:00:00
Zeynep	Ekim	11.11.1988 00:00:00
Can	Ak	21.10.1990 00:00:00
Ali	Su	19.09.1991 00:00:00
Aliya	Yılmaz	12.03.1991 00:00:00
Oya	Polat	21.03.1990 00:00:00
Onur	Doğan	24.04.1993 00:00:00
Murat	Çakır	01.09.1990 00:00:00
Ali	Su	08.09.1992 00:00:00

Şimdi Formatlı halini görelim.

```
string Sorgu = "SELECT Ad,Soyad, FORMAT(DT,'DD.MM.YYYY') FROM Ogrenciler";
```

**BİLGİLERİ GETİR**

Ad	Soyad	Expr1002
Ali	Su	11.09.1988
Zeynep	Ekim	11.11.1988
Can	Ak	21.10.1990
Ali	Su	19.09.1991
Aliya	Yılmaz	12.03.1991
Oya	Polat	21.03.1990
Onur	Doğan	24.04.1993
Murat	Çakır	01.09.1990
Ali	Su	08.09.1992

Burada tarihini listelediği başlık tam olarak anlaşılamadı. Burası için bir Alias (Takma isim) kullanalım.

```
string Sorgu = "SELECT Ad,Soyad, FORMAT(DT,'DD.MM.YYYY') AS Dogum_Tarihi FROM Ogrenciler";
```

**BİLGİLERİ GETİR**

Ad	Soyad	Dogum_Tarihi
Ali	Su	11.09.1988
Zeynep	Ekim	11.11.1988
Can	Ak	21.10.1990
Ali	Su	19.09.1991
Aliya	Yılmaz	12.03.1991
Oya	Polat	21.03.1990
Onur	Doğan	24.04.1993
Murat	Çakır	01.09.1990
Ali	Su	08.09.1992

**ROUND()**

Bu fonksiyon veritabanındaki küsüratlı sayıları görüntülerken yuvarlatarak göstermemizi sağlar.

Önce eski halini bir görelim.

```
string Sorgu = "SELECT Ad,Soyad, Ortalama FROM Ogrenciler";
```



**BİLGİLERİ GETİR**

Ad	Soyad	Ortalama
Ali	Su	64,3443543
Zeynep	Ekim	53,3454355
Can	Ak	56,9488757
Ali	Su	38,3773737
Aliya	Yılmaz	65,2388737
Oya	Polat	78,8484747
Onur	Doğan	67,8484747
Murat	Çakır	64,4477474
Ali	Su	72,9857575

Şimdi sayıları iki hane olacak şekilde yuvarlatalım.

```
string Sorgu = "SELECT Ad,Soyad, ROUND(Ortalama,2) FROM Ogrenciler";
```

**BİLGİLERİ GETİR**

Ad	Soyad	Expr1002
Ali	Su	64,34
Zeynep	Ekim	53,35
Can	Ak	56,95
Ali	Su	38,38
Aliya	Yılmaz	65,24
Oya	Polat	78,85
Onur	Doğan	67,85
Murat	Çakır	64,45
Ali	Su	72,99

**COUNT()**

Bir tabloda seçili olan alan içerisindeki değerleri saymak için COUNT deyimini kullanırız.

```
string Sorgu = "SELECT COUNT(Vize) AS Vize100 FROM Ogrenciler WHERE Vize=100";
```

**BİLGİLERİ GETİR**

Vize100
3

**MAX()**

Tabloda seçtiğimiz alanda en yüksek sayısal değeri almak için MAX komutunu kullanırız.

```
string Sorgu = "SELECT MAX(Final) AS FinalMaks FROM Ogrenciler";
```

BİLGİLERİ GETİR	
<b>FinalMaks</b>	
99	

```
string Sorgu = "SELECT MAX(Final) AS FinalMaksMekt FROM Ogrenciler WHERE BolumID=1";
```

BİLGİLERİ GETİR	
<b>FinalMaksMekt</b>	
67	

```
string Sorgu = "SELECT BolumID, MAX(Final) AS MaksNotlar FROM Ogrenciler GROUP BY BolumID";
```

BİLGİLERİ GETİR	
BolumID	MaksNotlar
1	67
2	99
3	89
5	45

## DİĞER FONKSİYONLAR

**AVG():** Sayıların ortalamasını verir

**SUM():** Sayıların toplamını verir

**NOW():** Sistem saatini getirir.

**UCASE():** Yazıları büyük harfe çevirir

**LCASE():** Yazıları küçük harfe çevirir.

**MID():** Yazıların ortasından belli bir karakteri okutmak için kullanılabilir.

**LEN():** Yazının boyutunu (kaç karakterden oluştuğunu) verir.

**FIRST():** İlk değeri verir.

**LAST():** Son değeri verir.

**MIN():** En küçük değeri verir.

## UYGULAMALAR

```
//Sorgu (Emir Listesi)
string Sorgu = "SELECT Ad, Soyad, Vize, Final, 'Ort=' & STR(Vize*0.4 + Final*0.6) AS Ortalama FROM Ogrenciler " +
"ORDER BY (Vize*0.4 + Final*0.6) ASC ";
```