



# Mühendislikte Veri Tabanları Dersi Uygulamaları *(MS-Access/SQL)*

---

*İstanbul Teknik Üniversitesi, İnşaat Fakültesi,  
Geomatik Mühendisliği Bölümü*

Prof. Dr. Nebiye MUSAOĞLU  
Doç. Dr. Şinasi KAYA  
Doç. Dr. Elif SERTEL  
Yrd. Doç. Filiz Bektaş BALÇIK  
Araş. Gör. Bahadır ÇELİK  
Araş. Gör. F. Başak SARIYILMAZ  
Araş. Gör. Özge DÖNMEZ



## İçindekiler Tablosu

KAPAK SAYFASI .....	0
<b>1. GİRİŞ</b> .....	3
1.1 Ön Bilgiler .....	3
1.2 Veri Tabanı Dosyası Oluşturma .....	4
<b>2. UYGULAMA</b> .....	6
2.1 SQL Komutları ile Tablo Oluşturma .....	6
2.2 Tasarım Görünümü ile Tabloların Oluşturulması .....	8
2.3 Tablolar Arasında İlişkilerin Oluşturulması .....	9
2.4 Sorgular .....	10
2.5 Diğer Sorgulama Örnekleri .....	13

# 1. GİRİŞ

## 1.1 Ön Bilgiler

**Veri tabanı**, verilerin belirli bir amaca yönelik olarak düzenlenerek bir araya getirilmiş biçimidir. Microsoft access, veritabanlarını yaratmak ve yönetmek için kullanılan bir ilişkisel veri tabanı yönetim sistemi yazılımıdır.

Access veritabanı yönetim sisteminde veriler;

- sorgulanabilir,
- süzülebilir,
- sıralanabilir,
- işlenebilir,
- raporlar hazırlanabilir ve daha pek çok işlem yerine getirilebilir.

Veri tabanı, en geniş anlamıyla; birbiriyle ilişkili verilerin tekrara yer vermeden, çok amaçlı kullanımına olanak sağlayacak şekilde depolanması olarak tanımlanabilir.

Veri tabanlarını yaratmayı ve kullanmayı sağlayan programlar topluluğuna “veri tabanı sistemi” ya da “veri tabanı yönetim sistemi – database management system (DBMS)” denir.

Access’in veri tabanını nasıl analiz ettiği incelenecek olursa,

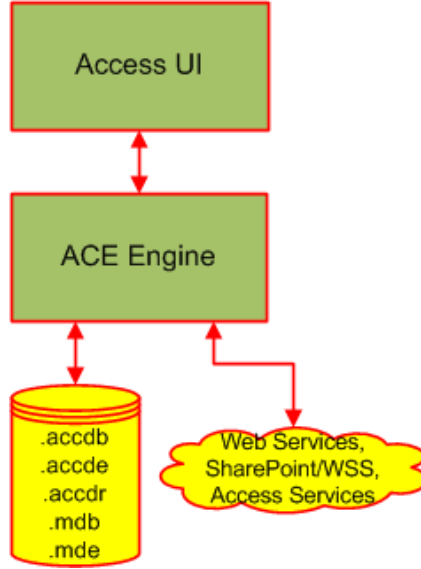
- Veri Tabanı
- Tablo (İlişkiler, Anahtarlar)
- Alan
- Veri türü (Metin, Sayı, Tarih, Para Birimi, Ek, Link vb)
- Değer

Microsoft Access 2010 için bazı veri türleri için alan boyutları;

- **Bayt** 0 ile 255 arasındaki tamsayılar için kullanın. Depolama gereksinimi 1 bayttır.
- **Tamsayı** -32.768 ile 32.767 arasındaki tamsayılar için kullanın. Depolama gereksinimi 2 bayttır.
- **Uzun Tamsayı** — -2.147.483.648 ile 2.147.483.647 arasındaki tamsayılar için kullanın. Depolama gereksinimi 4 bayttır.
- **Tek** -3,4 x 1038 ile 3,4 x 1038 arasında kalan ve en çok yedi anlamlı basamağı olan sayısal kayan nokta değerleri için kullanın. Depolama gereksinimi 4 bayttır.
- **Çift** -1,797 x 10308 ile 1,797 x 10308 arasında kalan ve en çok on beş anlamlı basamağı olan sayısal kayan nokta değerleri için kullanın. Depolama gereksinimi 8 bayttır.
- **Ondalık** -9,999... x 1027 ile 9,999... x 1027 arasındaki sayısal değerler için kullanın. Depolama gereksinimi 12 bayttır.
- **Text (Metin)** Belirli sayıda alfabetik karakter girişi yapılabilen veri tipidir. Maksimum 255 karakter girişi yapılabilir. Bu alana sembol girişi de yapılabilir.
- **Memo (Not)** Text tipinin yeterli olmadığı durumlarda kullanılır. 0 ile 32000 karakter alabilir. 2 KB (kilobyte) yer kaplar.
- **Currency (Para)** -2.147.483.648 ile +2.147.483.647 arasında 15 basamağa kadar hassasiyette ondalık sayılar girilebilir girilebilir.

**Kaynak:** <http://office.microsoft.com/tr-tr/access-help/HA010341783.aspx#DateTime>

Microsoft Access kavramsal çalışma prensibi;



**Kaynak:** [http://msdn.microsoft.com/en-us/library/office/ff965871\(v=office.14\).aspx#DataProgrammingWithAccess2010\\_Where](http://msdn.microsoft.com/en-us/library/office/ff965871(v=office.14).aspx#DataProgrammingWithAccess2010_Where)

Access 2007 öncesi - Microsoft Joint Engine Technology (JET)

Access 2007 ve sonrası - Microsoft Access Engine (ACE)

## 1.2 Veri Tabanı Dosyası Oluşturma

Yeni veritabanı dosyasının oluşturulması MS Access programı çalıştırılarak öncelikle dosyanın bilgisayarda hangi dizin altına ve ne isimle kaydedileceğine karar verilir ve kayıt işlemi ile dosya oluşturulur.

Bu işlemten sonra, program kullanıcıya tablo oluşturma olanağını otomatik olarak sunar ve uygulayıcı kendi için uygun olan tablo oluşturma şeklini seçer.

Alan Adı	Veri Türü
Adı	Metin
Soyadı	Metin
Numara	Metin
Doğum Tarihi	Not
Ödeme Tutarı	Sayı
	Tarih/Saat
	Para Birimi
	Otomatik Sayı
	Evet/Hayır
	OLE Nesnesi
	Köprü
	Ek
	Hesaplanmış
	Arama Sihirbazı...

Uygulama		
Alan Adı	Veri Türü	
Adı	Metin	
Soyadı	Metin	
Numara	Sayı	100 ile 200 arasında olmalı
Doğum Tarihi	Tarih/Saat	
Ödeme Tutarı	Para Birimi	

Alan adı alfabetik veya alfa nümerik ise veri türünün metin olarak seçilmesi gerekmektedir. Örneğin alan adı kısmına isim yazdığımızda veri türü olarak metin seçilmesi gibi.

Genel	Arama
Alan Boyutu	20
Biçim	
Giriş Maskesi	
Resim Yazısı	
Varsayılan Değer	
Geçerlilik Kuralı	
Geçerlilik Metni	
Gerekli	Hayır
Sıfır Uzunluk İzni	Evet
Sıralı	Hayır
Unicode Sıkıştırma	Evet
IME Modu	Denetim Yok
IME Tümce Modu	Hiçbiri
Akıllı Etiketler	

Metin seçildiğinde tablo ekranının sol alt köşesinde yer alan kısmın doldurulması gerekmektedir.

İkinci alan adı olarak soyisim yazılır ve veri türü olarak metin seçilir. Üçüncü alan ismi olarak numara yazılır ve veri türü olarak sayı seçilir.

Uygulama		
Alan Adı	Veri Türü	
Kimlik	Otomatik Sayı	
Adı	Metin	
Soyadı	Metin	
Numara	Sayı	
Doğum Tarihi	Tarih/Saat	
Ödeme Tutarı	Para Birimi	

Genel	Arama
Alan Boyutu	Uzun Tamsayı
Biçim	Bayt
Ondalık Basamaklar	Tamsayı
Giriş Maskesi	Uzun Tamsayı
Resim Yazısı	Tek
Varsayılan Değer	Çift
Geçerlilik Kuralı	Yineleme Kimliği
Geçerlilik Metni	Ondalık
Gerekli	Hayır
Sıralı	Hayır
Akıllı Etiketler	
Metin Hizala	Genel

Seçilen sayı verisine ait özelliğe uygun olarak alan boyutu kısmında mevcut olan türlerden biri seçilir.

Alanlar yazıldıktan sonra tablo kapanırken tablo kayıt menüsü görülür.

Değişiklikleri kaydetmek için evet tıklanır. Tablonun ismi yazılır (uygulama) ve değişiklikler kayıt edilir.

Tüm Tablolar	⌵	⏪
Uygulama		⏩
Uygulama : Tablo		

## 2. UYGULAMA

Uygulama kapsamında, temel öğrenci işleri otomasyonunu sağlamak amacıyla öğrenci otomasyon sistemine ilişkin bir veritabanı tasarlanacaktır. Bu amaçla sistem için gerekli olan temel tablolar aşağıdaki şekilde planlanmıştır;

- 
- Fakülteler (Fakulteler)
  - Programlar (Programlar)
  - Öğretim Üyeleri (OgretimUyesi)
  - Öğrenciler (Ogrenci)
  - Dersler (Dersler)
  - Öğrencilerin Seçtiği Dersler (OgrenciDers)
  - Öğretim Üyelerinin Verdiği Dersler (OgretimUyesiDers)
- 

### 2.1 SQL Komutları ile Tablo Oluşturma

SQL, (Structured Query Language), (yapılandırılmış sorgu dili), temeli “ilişkisel cebir”e dayanan ve ilişkisel veritabanlarında saklanan verilerin, yönetilmesi, işlenmesi, sorgulanması için tasarlanmış bir veri tabanı sorgu dilidir.

SQL dili, içerdiği komutların işlevlerine göre 3 bileşenden oluşur;

- DDL (Data Definition Language): Veri tanımlama deyimleri. (Tablo ve Kullanıcı Tanımlama)
- DML (Data Manipulation Language) : Veri düzenleme dili. (Veri Girme, Güncelleme, Silme v.s)
- DCL (Data Control Language): Veri kontrol dili. (Kullanıcı yetkilendirme, yetki verme ve alma işlemleri (GRANT, REVOKE))

Microsoft Access 2010 yazılımı için DDL komutları;

- **ADD USER Statement (Microsoft Access SQL)**
- ALTER TABLE Statement (Microsoft Access SQL)
- **ALTER USER or DATABASE Statement (Microsoft Access SQL)**
- CONSTRAINT Clause (Microsoft Access SQL)
- CREATE INDEX Statement (Microsoft Access SQL)
- CREATE PROCEDURE Statement (Microsoft Access SQL)
- CREATE TABLE Statement (Microsoft Access SQL)
- **CREATE USER or GROUP Statement (Microsoft Access SQL)**
- CREATE VIEW Statement (Microsoft Access SQL)
- DROP Statement (Microsoft Access SQL)
- **DROP USER or GROUP Statement (Microsoft Access SQL)**
- **GRANT Statement (Microsoft Access SQL)**
- **REVOKE Statement (Microsoft Access SQL)**

**Kaynak:**

Microsoft Access 2010 SQL reference:

[http://msdn.microsoft.com/en-us/library/office/ff841694\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/office/ff841694(v=office.14).aspx)

Microsoft Access 2010 yazılımı için DML komutları;

- DELETE Statement (Microsoft Access SQL)
- EXECUTE Statement (Microsoft Access SQL)
- INNER JOIN Operation (Microsoft Access SQL)
- INSERT INTO Statement (Microsoft Access SQL)
- LEFT JOIN, RIGHT JOIN Operations (Microsoft Access SQL)
- PARAMETERS Declaration (Microsoft Access SQL)
- PROCEDURE Clause (Microsoft Access SQL)
- SELECT Statement (Microsoft Access SQL)
- SELECT...INTO Statement (Microsoft Access SQL)
- SQL Subqueries (Microsoft Access SQL)
- TRANSACTION Statement (Microsoft Access SQL)
- TRANSFORM Statement (Microsoft Access SQL)
- UNION Operation (Microsoft Access SQL)
- UPDATE Statement (Microsoft Access SQL)
- WITH OWNERACCESS OPTION Declaration (Microsoft Access SQL)

**Kaynak:** Microsoft Access 2010 SQL reference:

[http://msdn.microsoft.com/en-us/library/office/ff841695\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/office/ff841695(v=office.14).aspx)

**SQL Standartları** (SQL-86, SQL-89, SQL-92, SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011)

SQL komutlarının Microsoft Access programına aktarılması için öncelikle SQL görünümüne geçilmesi gerekmektedir. Oluştur (Create) sekmesinden Sorgu Tasarımı (Query Design) özelliği seçildikten sonra, mevcut tablo olmadığı için açılan pencere kapatılır ve Tasarım (Design) sekmesinden ilk özellik olan SQL View butonu tıklanarak SQL sorgu penceresi açılır.



Oğrenci tablosunun oluşturulmasında kullanılacak SQL komutları;

```
CREATE TABLE Oğrenci (
```

```
oğrenciNo Long Primary Key,  
oğrenciProgramKodu Text(20),  
oğrenciAdi Text(70),  
oğrenciSoyadi Text(70),  
oğrenciDogumTarihi DateTime  
);
```

Create Table komutu ile istenilen isimde tablo oluşturuluyor.

Tabloya kaydedilmek istenilen alanlar ilgili özellikleri ile oluşturulabilir.

Bu komutların çalıştırılması ile Degerleyici tablosu oluşmuş olur. Şimdi ilgili tabloya bir öğrenci için veri girişi yapalım.

```
INSERT INTO Oğrenci  
VALUES (2014100, 'JDF', 'Alper', 'Esenli', '07.01.1989');
```



## 2.2 Tasarım Görünümü ile Tabloların Oluşturulması

- 1- Aşağıda verilen tablo verilerini kullanarak tasarım görünümünde tablo oluşturun.
- 2-Tabloların alan veri türleri yukarıda konu anlatımında bahsedilen şartlara bağlı kalınarak oluşturulmalıdır.

### Tabloların Tasarım Görünümü

#### Öğrenci Tablosu (Ogrenci)

Alan Adı	Veri Türü	Tanım
ogrenciNo	Sayı	Alan Boyutu (Uzun Tamsayı)
ogrenciProgramKodu	Metin	Alan Boyutu (20)
ogrenciAdi	Metin	Alan Boyutu (70)
ogrenciSoyadi	Metin	Alan Boyutu (70)
ogrenciDogumTarihi	Tarih/Saat	Biçim (Kısa Tarih)

#### Öğretim Üyesi Tablosu (OgretimUyesi)

Alan Adı	Veri Türü	Tanım
ogrUyeSicilNo	Sayı	Alan Boyutu (Uzun Tamsayı)
ogrUyeProgramKodu	Metin	Alan Boyutu (20)
ogrUyeUnvan	Metin	Alan Boyutu (20)
ogrUyeAdi	Metin	Alan Boyutu (70)
ogrUyeSoyadi	Metin	Alan Boyutu (70)

#### Fakülte Tablosu (Fakulteler)

Alan Adı	Veri Türü	Tanım
fakulteKodu	Sayı	Alan Boyutu (Tamsayı)
fakulteAdi	Metin	Alan Boyutu (255)

#### Programlar Tablosu (Programlar)

Alan Adı	Veri Türü	Tanım
programKodu	Metin	Alan Boyutu (20)
programFakulteKodu	Sayı	Alan Boyutu (Tamsayı)
programAdi	Metin	Alan Boyutu (255)

#### Dersler Tablosu (Dersler)

Alan Adı	Veri Türü	Tanım
dersCRN	Sayı	Alan Boyutu(Uzun Tamsayı)
dersProgramKodu	Metin	Alan Boyutu(20)
dersAdi	Metin	Alan Boyutu(255)
dersDonemi	Sayı	bayt(0,255), 1 = 1.dönem, 2=2.dönem
dersDerslik	Metin	Alan Boyutu(20)
dersGun	Metin	Alan Boyutu(50)
dersSaatBaslangic	Tarih/Saat	Biçim (Kısa Saat)
dersSaatBitis	Tarih/Saat	Biçim (Kısa Saat)
dersKapasite	Sayı	Alan Boyutu(Tamsayı)

## Öğretim Üyesinin Verdiği Dersler(OgretimUyesiDers)

Alan Adı	Veri Türü	Tanım
ogrUyeDersCRN	Sayı	Alan Boyutu(Uzun Tamsayı)
ogrUyeSicilNo	Sayı	Alan Boyutu(Uzun Tamsayı)

## Öğrencinin Aldığı Dersler (OgrenciDers)

Alan Adı	Veri Türü	Tanım
ogrenciNo	Sayı	Alan Boyutu(Uzun Tamsayı)
ogrenciDersCRN	Sayı	Alan Boyutu(Uzun Tamsayı)

## 2.3 Tablolar Arasında İlişkilerin Oluşturulması

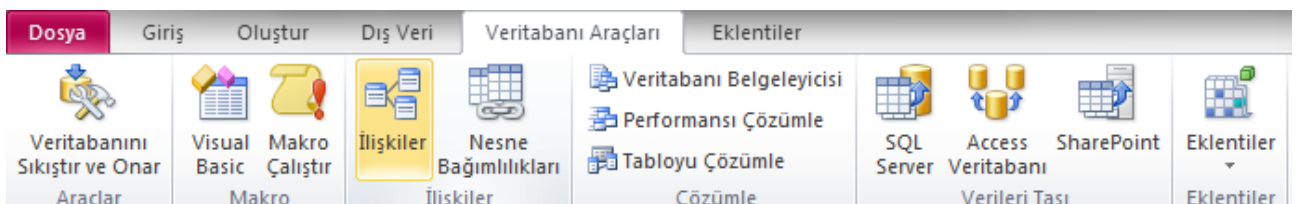
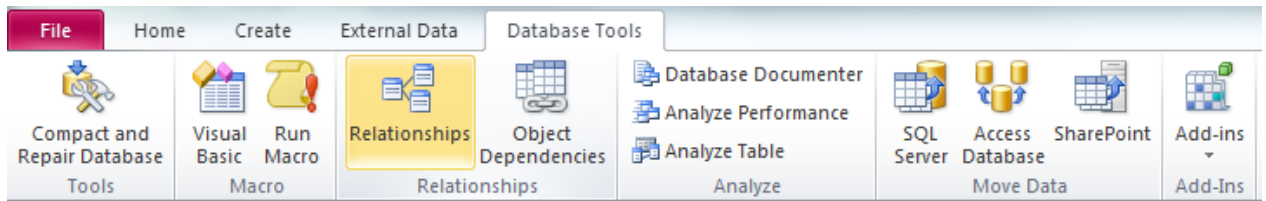
Birden fazla tablo içeren uygulamalarda sorgulama yapabilmek için tabloların ilişkilendirilmesi gerekmektedir. Veritabanı içindeki tabloları belirli alanlarına göre birbirleriyle ilişkilendirmek olasıdır. Bu sayede, bir tablodaki kayıtlar işlenirken, o kayıt ile ilişkili olan diğer tablodaki kayıtlarla da işlemler yapılabilecektir.

Tabloların ilişkilendirilmesi için;

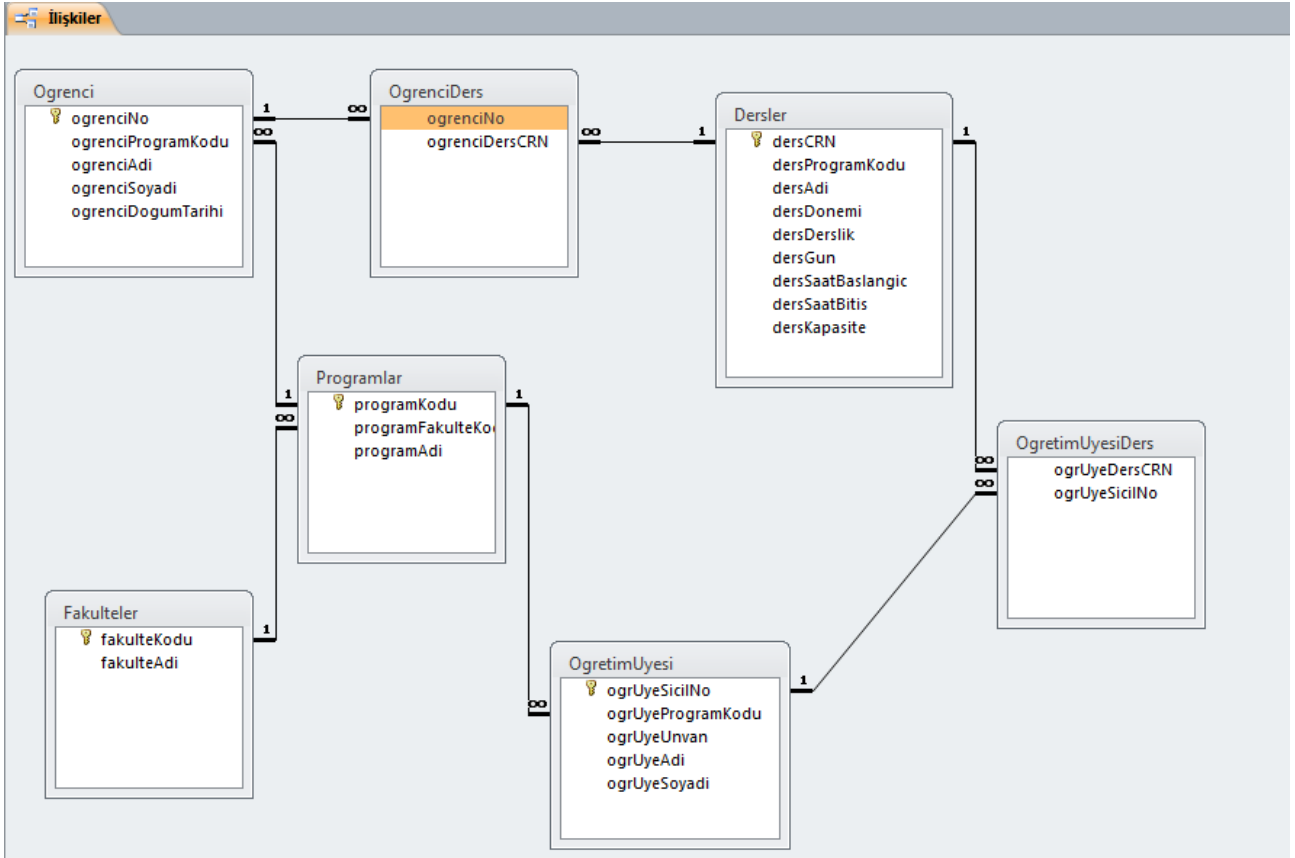
1. Aralarında ilişki kurulacak olan tabloların açık olmaması gerekmektedir.
2. Veritabanı araçları (DatabaseTools) menüsünden relationships (ilişkiler) sekmesi kullanılır.
3. Tabloyu göster penceresindeki tüm tablolar seçilir.
4. Tablolar arasında ilişki kurmak için, iki tabloda da ortak olan alan ismi belirlenir. Bir tablo üzerinde alan ismi seçilir ve o alan sürüklenerek diğer tablodaki aynı alan üzerine bırakılır.

Tablolar arası ilişkiler oluşturulurken (**check referential integrity**), (**Bilgi Tutarlılığını Zorla**) seçeneği seçilmelidir. Böylelikle, tablolar arası ilişkiler oluşturulduktan sonra, veri girişi anında, veri tabanı yönetim sistemi, bilgi tutarlılığını kontrol edecektir.

Bilgi tutarlılığının amacı, başı boş kayıtları önlemek ve artık var olmayan kayıtlara başvuruda bulunan kayıtların olmaması için başvuruları eşit durumda tutmaktır. Bilgi tutarlılığını zorunlu kılmak istediğinizde, bir tablo ilişkisi için etkinleştirirsiniz. Zorunlu kıldıktan sonra, Access, bu tablo ilişkisi için bilgi tutarlılığını ihlal edecek tüm işlemleri reddeder. (**Kaynak:** <http://office.microsoft.com/tr-tr/access-help/HA010072597.aspx#BM9> )



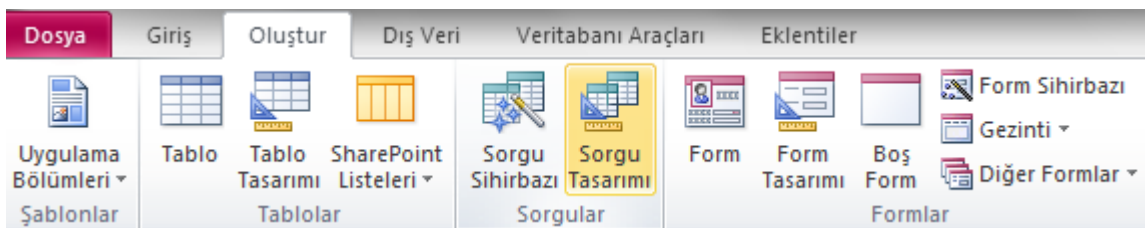
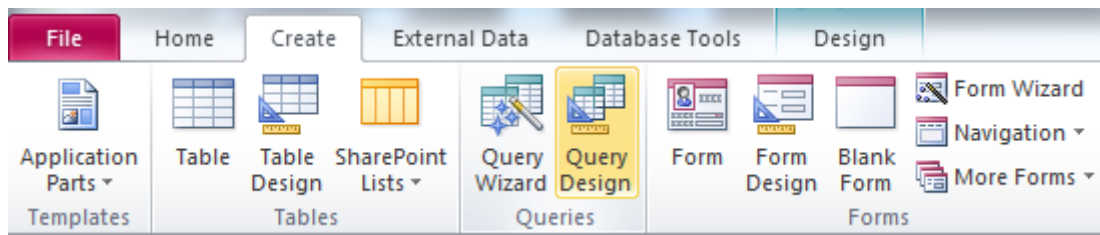
## Veri Tabanı İlişki Diagramı



## 2.4 Sorgular

### Access'te Sorgu Oluşturmak

Sorgu oluşturmak için Oluştur (Create) menüsü içerisindeki Sorgu (Query) seçenekleri kullanılır. Sorgu iki şekilde yaratılabilir. Tasarım görünümünde (Query Design) veya sihirbaz ile (Query Wizard), uygulamada detayların daha iyi anlaşılabilmesi için tasarım görünümünde sorgu yaratma seçeneği kullanılacaktır.



Sorgu tasarımı seçildikten sonra, “Tabloyu göster (show table)” menüsünden sorgu yapmak için kullanılacak olan tablolar seçilir.

**Sorgu 1 – “JDF” Kodlu Dersler:** Bu sorguyu oluşturmak için sadece “Dersler” tablosunun seçilmesi yeterli olacaktır. Dersler tablosundan **dersProgramKodu**, **dersCRN** ve **dersAdi** sütunları sorgu tasarımına ekleyelim. Özel koşul olarak, sadece açılan JDF kodlu dersleri aradığımız için **dersProgramKodu** özelliğine kriter olarak ‘JDF’ girişi yapalım.

Alan:	dersProgramKodu	dersCRN	dersAdi
Tablo:	Dersler	Dersler	Dersler
Sırala:			
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ölçüt:	"JDF"		
veya:			

Sorgu sekmesine sağ tıklayıp “kaydet” seçeneği yardımıle sorguyu istediğimiz bir isimde kaydedebiliriz. Sorguya “JDF Kodlu Dersler” adı verip kaydedelim. (Sorgulara isim verilirken noktalama işaretleri gibi özel karakter kullanılmamasına özen gösterilmelidir.)

Veri tabanı penceresinin sol kısmında, tablo isimlerin görüldüğü bölüme yeni yaratılan sorguda eklenmiş olacaktır. Sorgu sonuç çıktısı ise şu şekildedir;

dersProgramKodu	dersCRN	dersAdi
JDF	11192	Surveying II
JDF	11191	Surveying II
JDF	11193	Kartografyaya Giriş
JDF	11205	Jeodezik Astronomi
JDF	11208	Uzaktan Algılama
JDF	11212	Fotogrametri I
JDF	11231	Fiziksel Jeodezi
JDF	15820	Ölçme Bilgisi II

Sorguya ilişkin SQL;

```
SELECT Dersler.dersProgramKodu, Dersler.dersCRN, Dersler.dersAdi
FROM Dersler WHERE ((Dersler.dersProgramKodu)="JDF");
```

## Sorgu 2 – Sayma(Count,Group By,Having Yan Tümcesi) – İnşaat Mühendisliği (INS) Programındaki Öğretim Üyesi Sayısı

GROUP BY yan tümcesi içeren bir SELECT deyiminde hangi gruplandırılmış kayıtların görüntüleneceğini belirler. GROUP BY kayıtları birleştirdikten sonra, HAVING yan tümcesi, HAVING koşullarına uyan ve GROUP BY yan tümcesi ile gruplandırılmış kayıtları görüntüler.

Sözdizimi

```
SELECT alanlistesi
FROM tablo
WHERE seçimölçütleri
GROUP BY grupalanlistesi
[HAVING grupölçütleri]
```

HAVING yan tümcesi içeren bir SELECT deyiminin bölümleri şunlardır:

BÖLÜM	AÇIKLAMA
<i>alanlistesi</i>	Alan, diğer adı SQL toplam işlevi, seçim doğrulamaları (ALL, DISTINCT, DISTINCTROW veya TOP) veya diğer SELECT deyimi seçenekleri ile alınabilecek alan veya alanların adı.
<i>tablo</i>	Kayıtların alınacağı tablonun adı.
<i>seçimölçütleri</i>	Seçim ölçütleri. Deyim bir WHERE yan tümcesi içeriyorsa, Microsoft Access veritabanı altyapısı, kayıtlara WHERE koşullarını uyguladıktan sonra değerleri gruplandırır.
<i>grupalanlistesi</i>	Kayıtları gruplandırmak için kullanılan en çok 10 alanın adı. <i>Grupalanlistesi</i> içindeki alan adlarının sırası, grupta en üstten en alta doğru gruplandırma düzeylerini belirler.
<i>grupölçütleri</i>	Gruplandırılmış kayıtlardan hangilerinin görüntüleneceğini belirleyen ifade.

HAVING, hangi kayıtların seçileceğini belirten WHERE yan tümcesine benzer. Kayıtlar GROUP BY ile gruplandırıldıktan sonra HAVING hangi kayıtların görüntüleneceğini belirtir

**Kaynak:** <http://office.microsoft.com/tr-tr/access-help/HA001231483.aspx>

Bu sorguda, INS kodlu İnşaat Mühendisliği programındaki toplam öğretim üyesi sayısını yukarıda belirtilen açıklamalar doğrultusunda, öncelikle “Tasarım” görünümünde daha sonra SQL sorgusu yardımı ile elde etmeye çalışalım.

Sorguyu oluşturmak için, “OgretimUyesi” tablosunu eklememiz yeterli olacaktır. Bu tabloda, “Birincil Anahtar” olan “ogrUyeSicilNo” alanı ve “ogrUyeProgramKodu” alanları kullanılacaktır. Grublama işleminde “ogrUyeProgramKodu” alanı kullanılmalı ve kriter olarak program kodu “INS” girilmelidir. Öğretim üye sayısı için sayma işlemi ise “benzersiz alan” olan “ogrUyeSicilNo” alanı için “say”, “count” fonksiyonu kullanılmalıdır.

Sorgunun tasarım görünümündeki durumu ve sorgu çıktısı şu şekildedir;

Alan:	ogrUyeSicilNo	ogrUyeProgramKodu
Tablo:	OgretimUyesi	OgretimUyesi
Toplam:	Say	Grupla
Sırala:		
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ölçüt:		"INS"
veya:		

INS olan ogretim üyesi sayısı	
SayogrUyeSicilNo	ogrUyeProgramKodu
10	INS

Sorgu-2'ye ilişkin SQL;

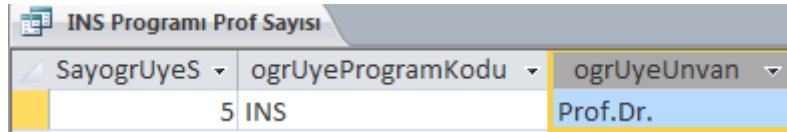
```
SELECT Count(OgretimUyesi.ogrUyeSicilNo) AS SayogrUyeSicilNo,  
OgretimUyesi.ogrUyeProgramKodu FROM OgretimUyesi  
GROUP BY OgretimUyesi.ogrUyeProgramKodu  
HAVING (((OgretimUyesi.ogrUyeProgramKodu)="INS"));
```

**Sorgu-2 (devam):** 2. Sorguya ek olarak İnşaat Müh. programındaki Prof. Olan öğretim üyelerinin sayısını bulmak istersek, bir önceki sorgunun "HAVING" yan tümcesi içinde ikinci bir kriter olarak "ogrUyeUnvan" alanını ekleyip kriter olarak, "Prof.Dr." giriş yapmamız yeterli olacaktır.

Alan:	ogrUyeSicilNo	ogrUyeProgramKodu	ogrUyeUnvan
Tablo:	OgretimUyesi	OgretimUyesi	OgretimUyesi
Toplam:	Say	Grupla	Grupla
Sırala:			
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ölçüt:		"INS"	"Prof.Dr."
veya:			

```
SELECT Count(OgretimUyesi.ogrUyeSicilNo) AS SayogrUyeSicilNo,  
OgretimUyesi.ogrUyeProgramKodu FROM OgretimUyesi  
GROUP BY OgretimUyesi.ogrUyeProgramKodu  
HAVING (((OgretimUyesi.ogrUyeProgramKodu)="INS") AND  
((OgretimUyesi.ogrUyeUnvan)="Prof.Dr."));
```

Sorgu Çıktısı;



SayogrUyeS	ogrUyeProgramKodu	ogrUyeUnvan
5	INS	Prof.Dr.

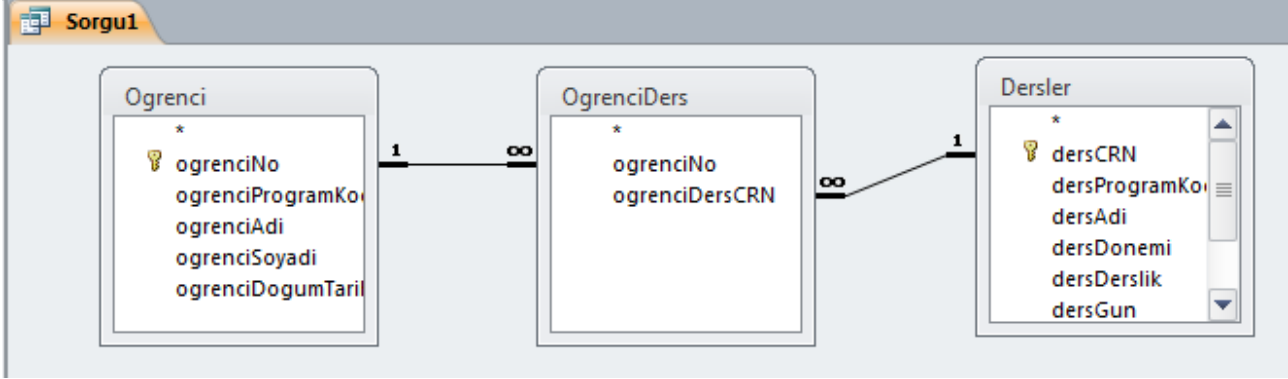
## 2.5 Diğer Sorgulama Örnekleri

### Bir Ölçüte Uygun Kayıtların Seçilmesi

Sorgu işlemlerinde tablo içinde yer alan bazı kayıtların, belirli bir ölçüte ya da bir başka deyişle bir koşula göre seçilerek görüntülenmesi söz konusu olabilir. Ölçütler tanımlanırken amacına uygun biçimde bazı özel işlemler kullanılabilir. Söz konusu işlemler aşağıdaki gibidir;

- > Büyük
- < Küçük
- <> Eşit değil
- <= Küçük veya eşit
- >= Büyük veya eşit

**Sorgu 3 - (Birden Fazla Tablodan Veri Sorgulama) Belirli bir CRN kodlu dersi seçen öğrenciler:** Bu sorguda, “bire-çok” ilişkinin bulunduğu birden fazla tablodan istenilen kriterdeki verileri elde etmemeye yönelik sorgu tasarlanacaktır. Bu amaçla tasarladığımız veri tabanından “Ogrenci”, “OgrenciDers” ve “Dersler” tabloları kullanılacaktır. İlgili tablolar yeni sorgu oluşturmak amacıyla eklenir;



Belirlenmiş bir CRN olarak “11193” CRN koduna sahip dersi seçen öğrencileri bulmak için gerekli sorgu tasarımı ise şu şekildedir.

Alan:	dersCRN	ogrenciProgramKodu	ogrenciNo	ogrenciAdi	ogrenciSoyadi
Tablo:	Dersler	Ogrenci	Ogrenci	Ogrenci	Ogrenci
Sırala:					
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ölçüt:	11193				
veya:					

Sorgu Çıktısı;

dersCRN	ogrenciProgramKodu	ogrenciNo	ogrenciAdi	ogrenciSoyadi
11193	JDF	2014100	Alper	Esenli
11193	JDF	2014102	Nilgün	Çakır

Sorgu-3 SQL;

```
SELECT Dersler.dersCRN, Ogrenci.ogrenciProgramKodu, Ogrenci.ogrenciNo,
Ogrenci.ogrenciAdi, Ogrenci.ogrenciSoyadi
FROM Dersler INNER JOIN (Ogrenci INNER JOIN OgrenciDers ON
Ogrenci.ogrenciNo = OgrenciDers.ogrenciNo) ON Dersler.dersCRN =
OgrenciDers.ogrenciDersCRN
WHERE (((Dersler.dersCRN)=11193));
```

**Sorgu 4 - 09:00 ve 13:00 saatleri arasındaki dersler:** Bu sorguda istenen saatler arasındaki dersler için sorgulama yapılacaktır. Acces'te datetime veri türündeki alanlar için sorgulama yapılırken, kriter alanına girilecek değer, alan biçimine sadık kalınacak şekilde “#” işareti arasında yazılır.

**Sorgu1**

Dersler

- \* dersCRN
- dersProgramKodu
- dersAdi
- dersDonemi
- dersDerslik
- dersGun
- dersSaatBaslangic
- dersSaatBitis
- dersKapasite

Alan:	dersCRN	dersProgramKodu	dersAdi	dersSaatBaslangic	dersSaatBitis
Tablo:	Dersler	Dersler	Dersler	Dersler	Dersler
Sırala:					
Göster:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Ölçüt:				>#09:00:00#	<#13:00:00#
veya:					

Sorgunun sonucu aşağıdaki gibi olacaktır.

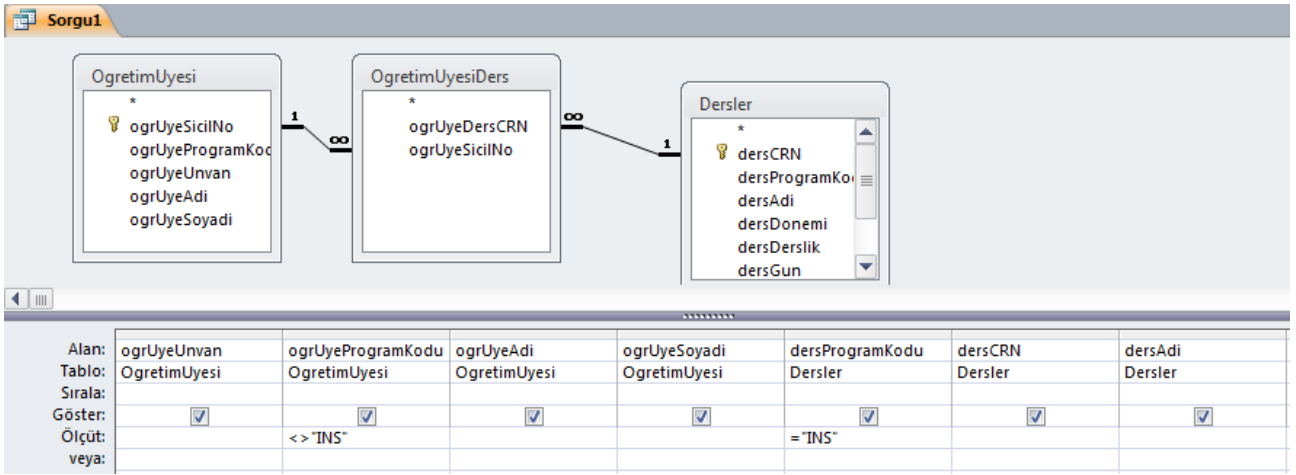
**Sorgu1**

dersCRN	dersProgramKodu	dersAdi	dersSaatBas	dersSaatBiti
11191	JDF	Surveying II	09:30	12:30
11231	JDF	Fiziksel Jeodezi	09:30	11:30
12272	INS	Karayolu Mühendisli	09:30	12:30
15820	JDF	Ölçme Bilgisi II	09:30	12:30
12321	INS	İnşaat Jeolojisi	09:30	12:30
12322	INS	İnşaat Jeolojisi	09:30	12:30
12240	INS	Yapı Statiği I	09:30	12:30
12253	INS	Yapı Statiği II	09:30	11:30
*				

### Sorgu 5 – Bölüm Dışı Ders Veren Öğretim Üyesi:

Herhangi bir programda görev yapan öğretim üyesinin bulunduğu program dışında verdiği ders olup olmadığını sorgulayabiliriz. Örneğin “INS” kodlu herhangi bir dersin bu programın haricinde başka bir programdan Öğretim üyesi tarafından verilir verilmediğini kontrol için “OgretimUyesi”, “OgretimUyesiDers” ve “Dersler” tablolarını kullanarak şu şekilde sorgulayabiliriz.





Sorgu çıktısı şu şekilde olacaktır;

**bolumDisiDersVerenOğretimUyesi**

ogrUyeUnva	ogrUyeProgramKodu	ogrUyeAdi	ogrUyeSoya	dersPrograml	dersCRN	dersAdi
Yrd.Doç.Dr	JEO	Erkan	Bozkurtoğlu	INS	12323	İnşaat Jeolojisi

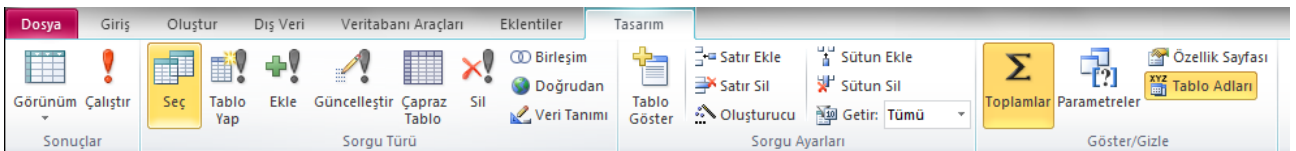
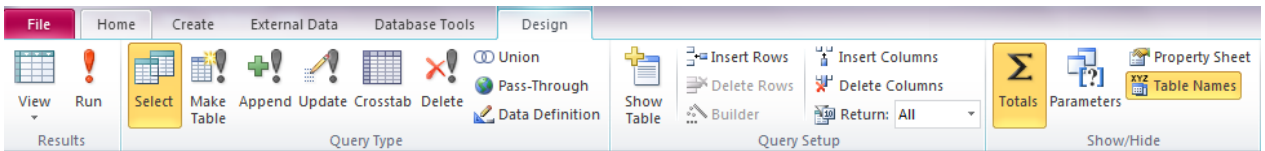
Sorgu-5 SQL;

```

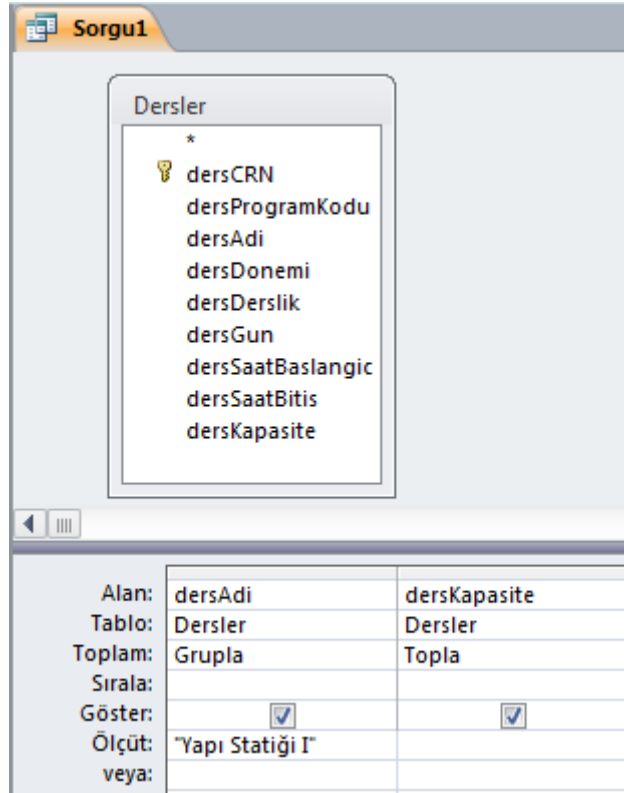
SELECT OğretimUyesi.ogrUyeUnvan, OğretimUyesi.ogrUyeProgramKodu,
OğretimUyesi.ogrUyeAdi, OğretimUyesi.ogrUyeSoyadi,
Dersler.dersProgramKodu, Dersler.dersCRN, Dersler.dersAdi
FROM Dersler INNER JOIN (OğretimUyesi INNER JOIN OğretimUyesiDers ON
OğretimUyesi.ogrUyeSicilNo = OğretimUyesiDers.ogrUyeSicilNo) ON
Dersler.dersCRN = OğretimUyesiDers.ogrUyeDersCRN
WHERE (((OğretimUyesi.ogrUyeProgramKodu)<>"INS") AND
((Dersler.dersProgramKodu)="INS"));

```

**Sorgu 6 :** Bir tablodaki sayısal alanların toplanması gerekebilir. Örneğin farklı öğretim üyeleri tarafından verilen “Yapı Statiği I” adlı dersin toplam kapasitesini öğrenmek isteyebiliriz. Sorgu için “Dersler” tablosu seçilir. Alan olarak “dersKapasite” seçilerek, Tasarım (Design) menüsü altında yer alan toplam ( $\Sigma$ ) butonu seçilir. Bu butonun seçilmesiyle alt kısımda total (toplam isimli) yeni bir menu oluşacaktır. Menünün içinde bulunan çeşitli seçeneklerden sum(Topla) seçilerek, toplam kapasite miktarı bulunabilir. Bu durumda aşağıdaki sonuç elde edilecektir.



Sorgu Tasarımı;



Sorgu Çıktısı;

dersAdi	TopladersKapasite
Yapı Statiği I	150

Sorgu-6 SQL;

---

```
SELECT Dersler.dersAdi, Sum(Dersler.dersKapasite) AS TopladersKapasite
FROM Dersler
GROUP BY Dersler.dersAdi
HAVING ((Dersler.dersAdi)="Yapı Statiği I");
```

---