# Veritabanı Yönetim Sistemleri

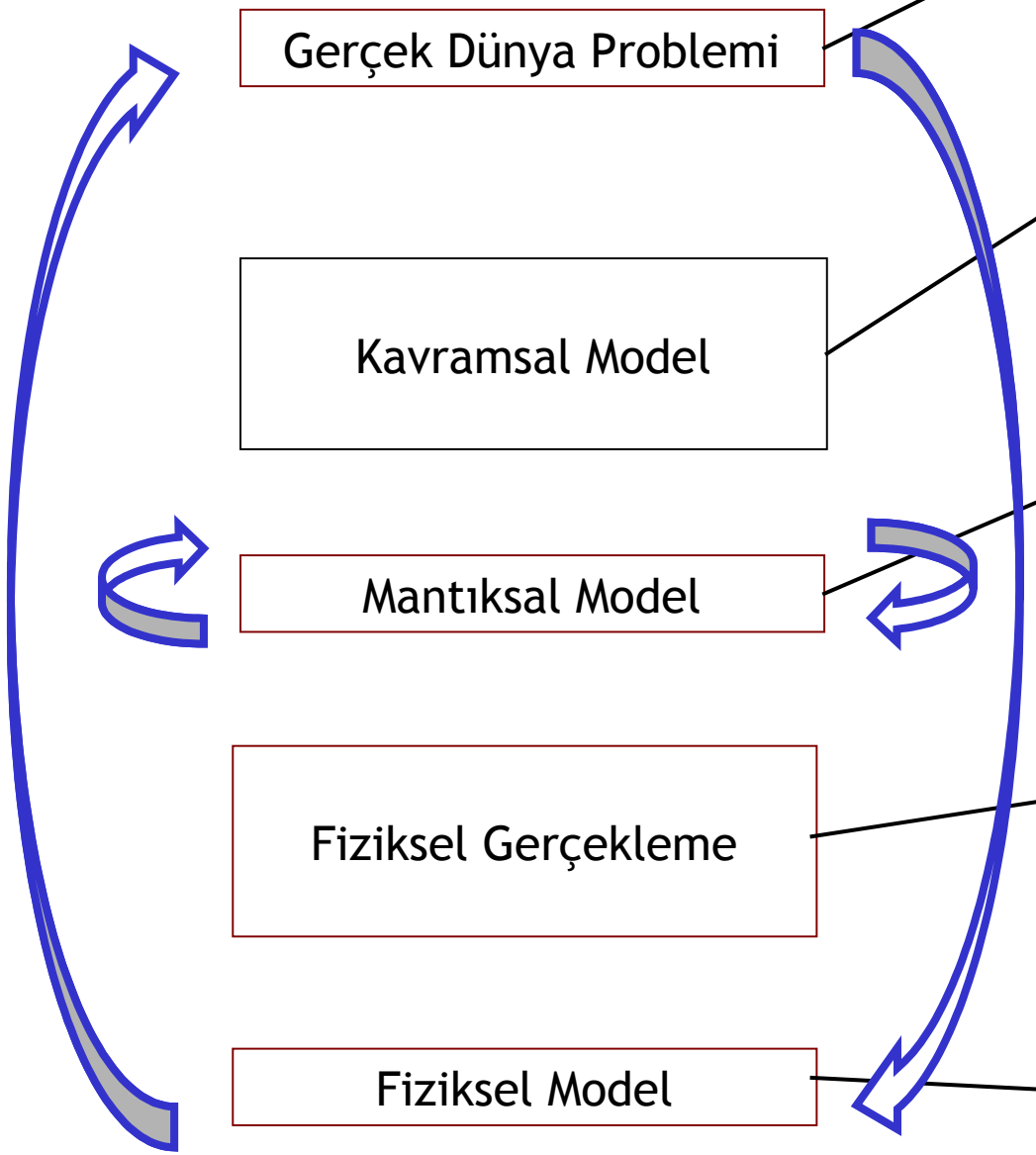
## (Veritabanı Kavramı)

## İş Kuralları ve Veri Modelleri

# Konular

- ✓ Veritabanı Tasarımı Yaşam Döngüsü
- ✓ Veri Modeli Nedir?
- ✓ Veri Modeli Temel Bileşenleri
- ✓ İş Kuralları (Business Rules)
- ✓ İş Kurallarını Veri Modeline Dönüştürme
- ✓ Veri Modellerinin Gelişimi
- ✓ Dosya Sistemi
- ✓ Hiyerarşik Model
- ✓ Ağ Modeli
- ✓ İlişkisel Veri Modeli
- ✓ Varlık Bağıntı Modeli
- ✓ Nesne Yönelimli Model
- ✓ Yeni Veri Modelleri
- ✓ Veri Soyutlama
- ✓ Kaynaklar

# Veritabanı Tasarımı Yaşam Döngüsü




Gereksinim analizi gerçekleştir: Problem senaryosundan, önemli noktalar ve ihtiyaçlar belirlenerek **iş kuralları** oluşturulur.

İş kurallarına bakılarak, donanım ve yazılımdan bağımsız olan kavramsal model tasarlanır.

Kavramsal model, VTYS'ne uygulanabilecek şekilde dönüştürülür/düzenlenir. Normalizasyon/Denormalizasyon uygulanır.

Bir VTYS kullanılarak mantıksal modelin uygulaması gerçekleştir. Örneğin; tabloların oluşturulması, kısıtları eklenmesi, yetkilendirmeler vb.

Gerçek verilerin saklandığı, VTYS tarafından yönetilen veritabanı yapısı. Donanım ve yazılım bağımlı.

# İş Kuralları (Business Rules)

- ✓ Veritabanı oluşturulurken (varlık, nitelik, ilişki ve kısıtlar) iş kurallarına bakılır.
- ✓ Veritabanı tasarımı yapılacak organizasyon ile ilgili işleyiş, kural ya da yönetmeliğin özetlenmiş şekline iş kuralları denebilir (ihtiyaç listesine benzer).
- ✓ Örnek iş kuralları:
  - ✓ Bir müşteri çok sayıda sipariş verebilir.
  - ✓ Her müşterinin adı, soyadı, telefon numarası, vs. istenir.
  - ✓ Öğrenciler bir ara sınav ve bir yarıyıl sonu sınavına girerler.
- ✓ İş kurallarının kaynağı; yöneticiler, kural koyucular, ve yazılı dokümanlar olabilir.
- ✓ İş kurallarını oluşturmak için doğrudan uç kullanıcılarla görüşmek de oldukça etkili bir çözümdür.
- ✓ Veritabanı tasarımı açısından iş kurallarının önemi;
  - ✓ Kullanıcılar ile tasarımcılar arasındaki iletişimi sağlar.
  - ✓ Tasarımcının verinin doğasını, önemini ve kapsamını anlamasını sağlar.
  - ✓ Tasarımcının iş süreçlerini anlamasını sağlar.
  - ✓ Tasarımcının doğru bir veri modeli geliştirmesine yardım eder (veriler arası ilişkiler ve kısıtların kolayca belirlenmesini sağlar).
  - ✓ Kuruluşun veriye bakışını standart haline getirir.
- ✓ İş kuralları oluşturulduktan sonra, gerçekleştirilecek veritabanının modellenmesi aşamasına geçilir.

# Veri Modeli Nedir?

- ✓ Veri modeli: Karmaşık gerçek dünya veri yapılarının basit olarak gösterilmesi (genellikle grafiksel) için kullanılan araç.
- ✓ Veri tabanı tasarımcıları, uygulama programcıları ve uç kullanıcılar arasındaki iletişimi kolaylaştırır.
- ✓ Veri modelleri sayesinde veritabanı tasarımını gerçekleştirmek daha kolay olur.
- ✓ Veri modelleme yinelemeli (iterative) bir işlemdir. Önce basit model oluşturulur. Daha sonra ayrıntılar eklenir. En sonunda veritabanı tasarımında kullanılan şablon (blueprint) elde edilir.

# Veri Modeli Temel Bileşenleri

- ✓ **Varlık (Entity)** : Hakkında veri toplanan ve saklanan her şey (öğrenci, ders, personel vb.). Gerçek dünyadaki nesnelerin durumlarını ifade eder. Var olan ve benzerlerinden ayırt edilen her şey.

- ✓ **Varlık kümesi (Entity set):** Aynı türden benzer varlıkların oluşturduğu kümeye denir (Öğrenciler, Dersler vb.).

- ✓ **Nitelik (Attribute):** Varlığın sahip olduğu özellikler.

- ✓ **Bağıntı (Relationship):** Varlıklar arasındaki ilişkiyi ifade eder.
  - ✓ Bir-çok (One to Many **1:M**)

  Bir müşteri çok sayıda sipariş verebilir. Her sipariş yalnızca bir müşteri tarafından verilir.
  - ✓ Çok-çok (Many to Many **M:N**)

  Bir öğrenci çok sayıda ders alabilir. Her ders çok sayıda öğrenci tarafından alınabilir.
  - ✓ Bir-bir (One to One **1:1**)

  Bir mağaza bir personel tarafından yönetilir.

- ✓ **Kısıtlar (Constraints)**: Veri üzerindeki sınırlamalardır. Veri bütünlüğünün sağlanması açısından önemlidir. Örneğin;
  - ✓ Öğrenci notunun 0-100 arasında olması
  - ✓ TC Kimlik numarasının 11 karakter olması
  - ✓ Aynı ürünün birden fazla kayıt edilememesi

# Veri Modellerinin Gelişimi

Dosya Sistemi

Hiyerarşik Model

Ağ Modeli

İlişkisel veri modeli

Varlık Bağıntı modeli

Nesne Yönelimli Model

Yeni Veri Modelleri

# Dosya Sistemi

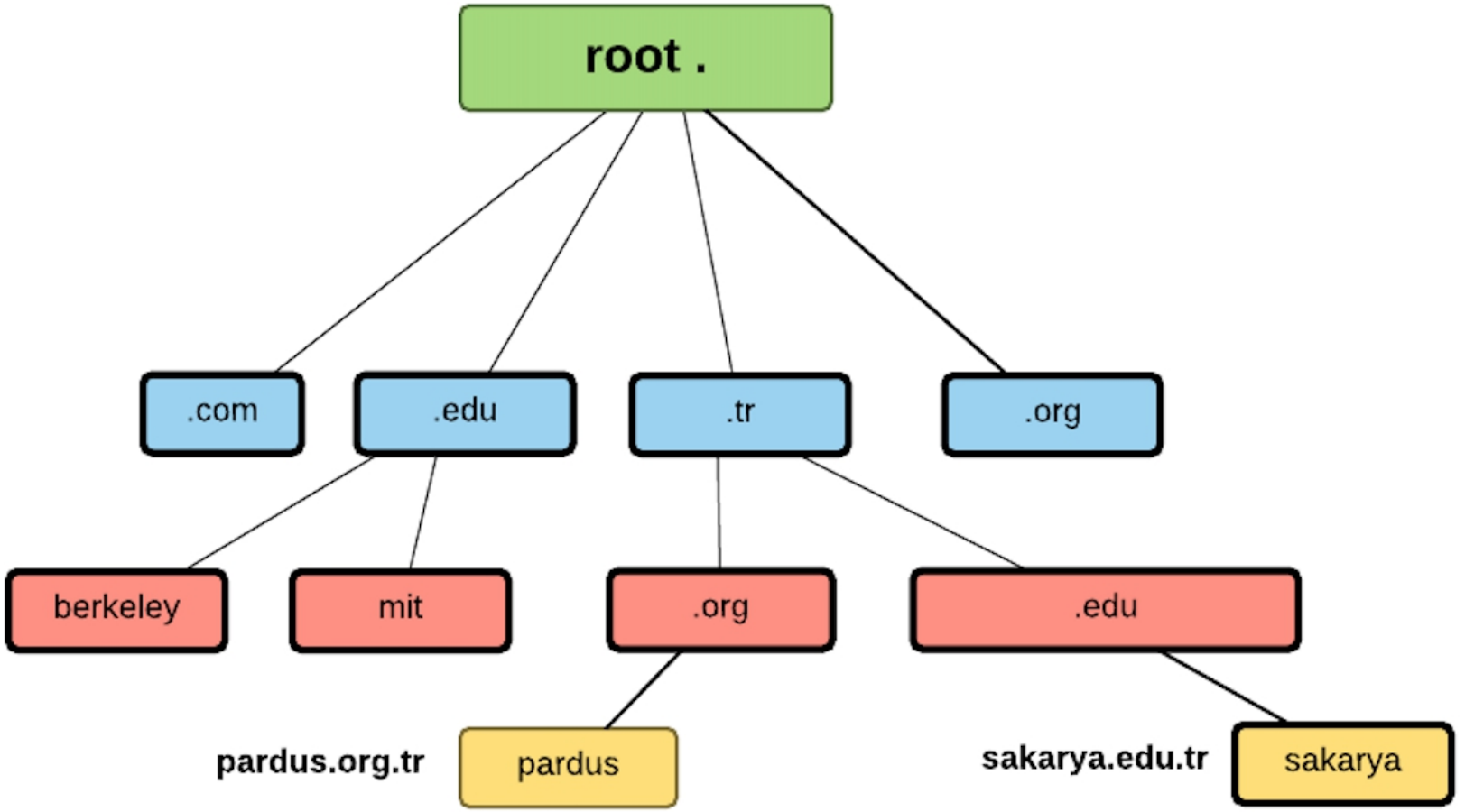
- ✓ 1960-1970 lerde çoğunlukla IBM Mainframe sistemlerde kullanılmıştır
- ✓ Dosyalar arasında ilişki yoktur…

| Ders Kodu | Ders Adı | Öğr.Türü | Dönem | Bölüm Adı |
|---|---|---|---|---|
| BSM207 | VERİ YAPILARI (B) | 1. Öğretim | 1 | BİLGİSAYAR MÜHENDİSLİĞİ PR. |
| BSM303 | VERİTABANI YÖNETİM SİSTEMLERİ (A) | 1. Öğretim | 1 | BİLGİSAYAR MÜHENDİSLİĞİ PR. |
| BSM207 | VERİ YAPILARI (B) | 2. Öğretim | 1 | BİLGİSAYAR MÜHENDİSLİĞİ PR. (İÖ) |
| BSM303 | VERİTABANI YÖNETİM SİSTEMLERİ (A) | 2. Öğretim | 1 | BİLGİSAYAR MÜHENDİSLİĞİ PR. (İÖ) |
| BSM303 | VERİTABANI YÖNETİM SİSTEMLERİ (?) | Uzaktan Eğitim | 1 | BİLGİSAYAR MÜHENDİSLİĞİ PR. (UZAKTAN EĞİTİM) |
| EBT514 | VERİTABANI TASARIM VE YÖNETİMİ (?) | Uzaktan Eğitim | 1 | BİLİŞİM TEKNOLOJİLERİ PR. (YL) (UZAKTAN EĞİTİM) |
| BSM829 | UZMANLIK ALANI (?) | 1. Öğretim | 1 | BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ PR. (YL) |
| BSM929 | UZMANLIK ALANI (?) | 1. Öğretim | 1 | BİLGİSAYAR VE BİLİŞİM MÜHENDİSLİĞİ PR. (DR) |
| BSM401 | BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI (F) | 1. Öğretim | 1 | BİLGİSAYAR MÜHENDİSLİĞİ PR. |
| BSM401 | BİLGİSAYAR MÜHENDİSLİĞİ TASARIMI (F) | 2. Öğretim | 1 | BİLGİSAYAR MÜHENDİSLİĞİ PR. (İÖ) |

Örnek dosya yapısı

# Hiyerarşik Model

1960'larda büyük miktardaki verileri yönetebilmek için geliştirilmiştir.
1969'da aya inen Apollo uzay mekiğinde kullanılmıştır.
Veriler ağaç yapısı şeklinde organize edilir.
Ebeveyn-çocuk (parent-child) arasında 1:M ilişkisi vardır. Kayıtların sadece 1 ebeveyn (parent) kaydı vardır.

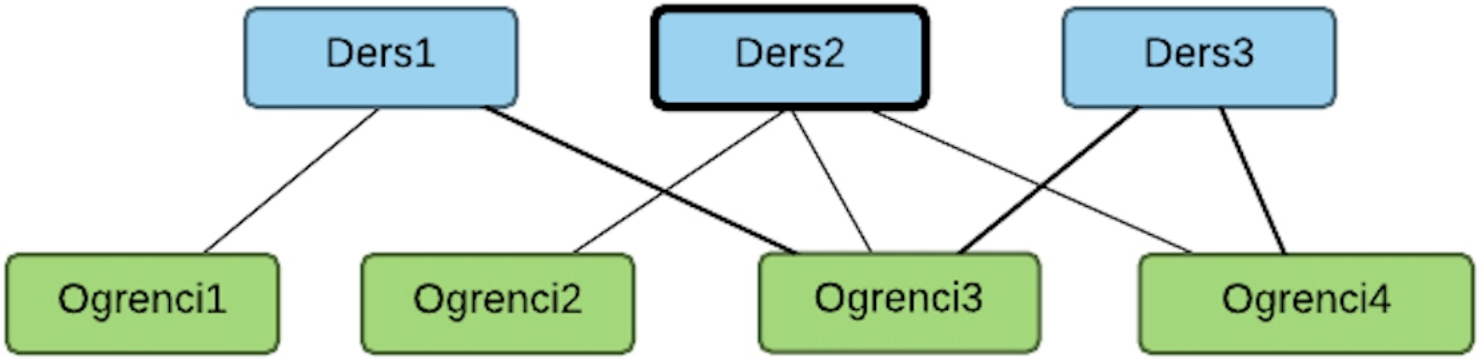


# Ağ Modeli

1970'lerde geliştirilmiştir. Veritabanı başarımını artırmak üzere daha karmaşık ilişkilere izin verilir
Hiyerarşik modelden farklı olarak kayıtların birden fazla ebeveyn (parent) kayıtları olabilir.
Ağ veri modeliyle birlikte ortaya çıkan ve hala kullanılan bazı kavramlar:

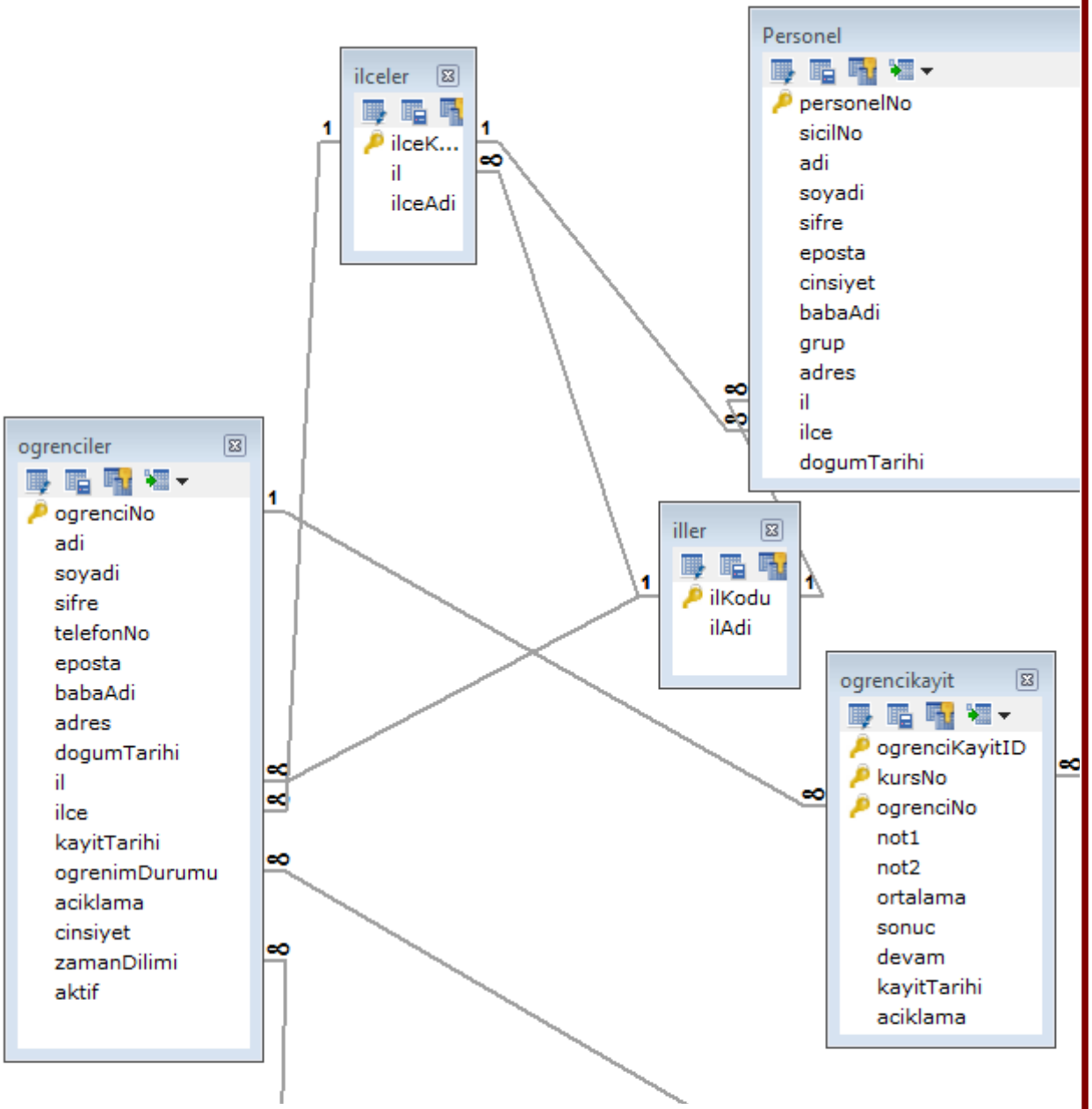
**Şema**: Tüm veritabanının, veritabanı yöneticisi tarafından görünen kavramsal organizasyonu.
**Alt şema**: Veritabanının istenen bilgiyi üreten uygulama programı tarafından görünen kısmı.
**Veri işleme dili (Data manipulation language, DML)** : Veritabanında bulunan verilerin, sorgulama işlemleri yapılarak güncellenmesi, yeni verilerin eklenmesi ve olan verilerin silinme işlemlerinin yapılmasını sağlayan dil.
**Veri tanımlama dili (Data definition language, DDL)**: Veri tabanında bulunan verilerin tip, yapı ve kısıtlamalarının tanımlanmasını sağlayan dil.

Çok basit sorgular için bile karmaşık program kodlarının kullanımını gerektirir.
**Ad hoc query:** Yazılımlarla birlikte gelmeyen kullanıcının kendi oluşturduğu sorgulara verilen isimdir.



# İlişkisel Veri Modeli

- ✓ 1970'de E. F. Codd tarafından ortaya atılmıştır (A Relational Model of Data for Large Shared Databanks, *Communications of the ACM, June 1970, pp. 377–387*).
- ✓ RDBMS (Relational DBMS) tarafından kullanılır.
- ✓ RDBMS'nin en önemli özelliklerinden biri ilişkisel modelin karmaşık yapısını kullanıcıdan gizlemesidir.
- ✓ Kullanıcı, ilişkisel modeli verileri içeren tablolardan oluşan bir yapı gibi görür.
- ✓ Tablolar birbirlerine ortak alanlarla bağlanırlar.
- ✓ İlişkisel şema, varlıklar, varlıkların nitelikleri ve aralarındaki bağlantıların gösteriminden oluşur.

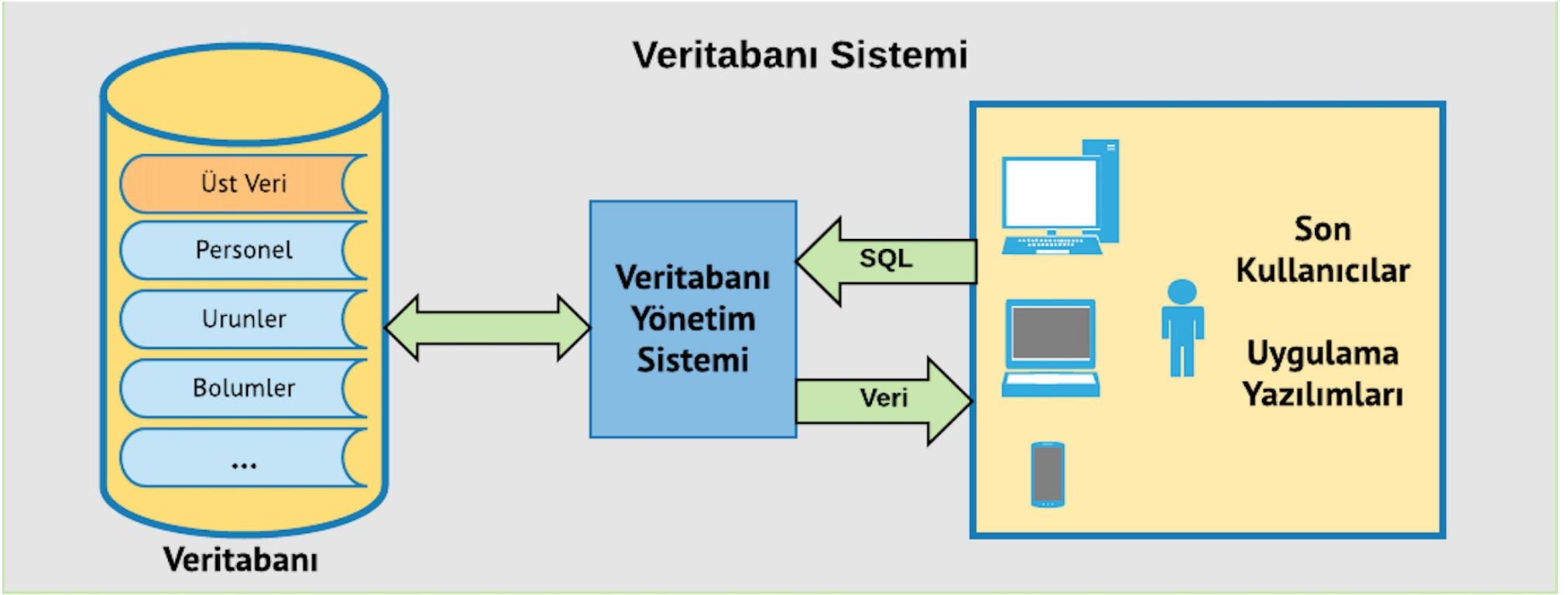


# İlişkisel Veri Modeli

İlişkisel veritabanı modelinin en güçlü yanlarından biri de verileri yönetmek için SQL dilinin kullanılıyor olmasıdır. SQL dili nasıl yapılması gerektiğini anlatmak yerine ne yapılması gerektiğinin ifade edildiği basit bir dildir. Bu nedenle, SQL kullanılarak veritabanlarının tasarımı ve yönetimi daha kolaydır.

İlişkisel bir veritabanı yönetim sistemi 3 temel bileşenden oluşur.

1. Verilerin saklandığı veritabanı
2. SQL komutlarını derleyerek istenenleri gerçekleştiren SQL Motoru (SQL Engine)
3. Kullanıcılarla iletişimi sağlayan arayüzler.

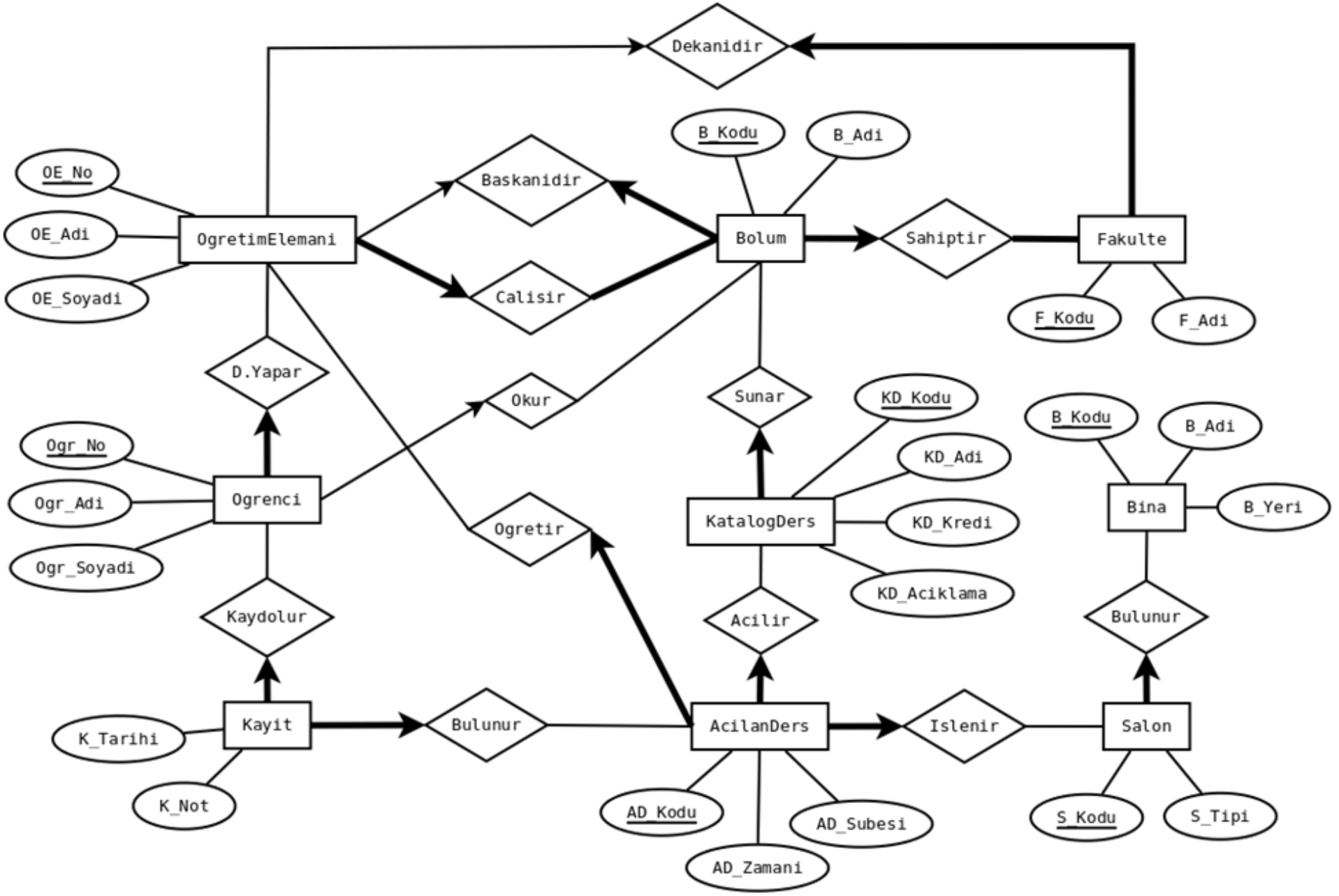


# Varlık Bağıntı Modeli

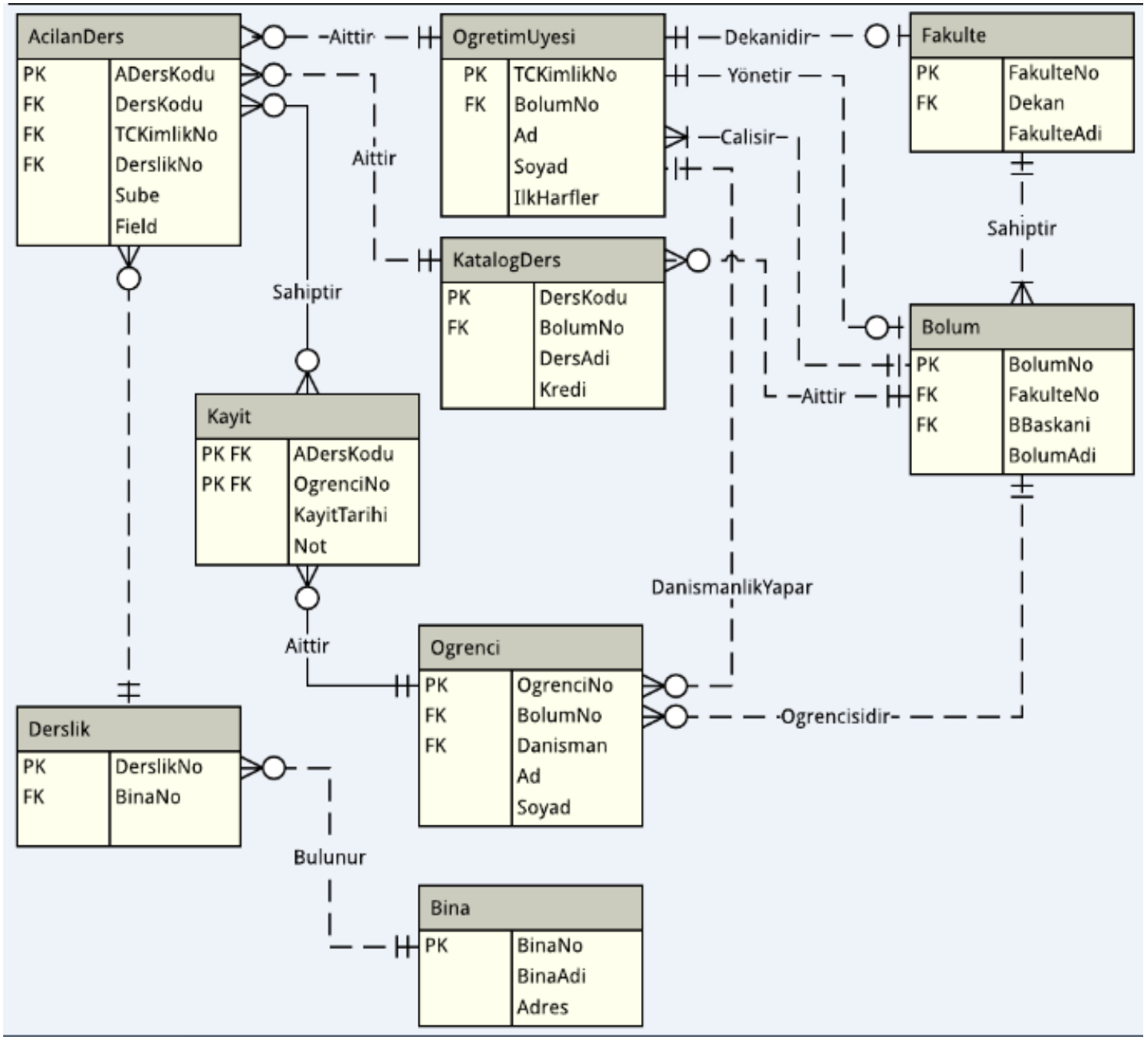
- ✓ İlişkisel Model daha önceki modellere göre çok daha kullanışlı olmasına rağmen veritabanı tasarımı için ilişkisel modelin grafiksel gösterimi olan varlık bağıntı veri modeli (**entity relationship (ER) model**) daha sık kullanılır.
- ✓ 1976'da Peter Chen tarafından önerilmiştir.
- ✓ İlişkisel veri modelinin tamamlayıcısı olduğu için kullanımı oldukça yaygınlaşmıştır.
- ✓ i) Chen notasyonu ve ii) Crow's Foot notasyonu sıkça kullanılan gösterim şekillerindendir.

Ders kapsamında Crow's Foot notasyonu kullanılacaktır.

# Varlık Bağıntı modeli
## Chen Gösterimi



# Varlık Bağıntı modeli
## Crow's Foot Gösterimi



# Nesne Yönelimli Model

- ✓ Nesne yönelimli programlama paradigmasından esinlenerek geliştirilen modeldir.
- ✓ Varlık bağıntı (VB - ER) modelindeki varlık (entity) bu modelde nesne olarak adlandırılır.
- ✓ Nesne hakkındaki bilgi, VB modelindeki niteliklere karşılık gelir.
- ✓ Varlık kümesi sınıf olarak adlandırılır.
- ✓ VB modelinden farklı olarak sınıflar üye fonksiyonlara da sahiptirler. Kisi ara, Ad listele vb.

| VB Modeli | İlişkisel Model | Nesne Yönelimli Model |
|---|---|---|
| Varlık | Kayıt/Satır | Nesne |
| Varlık Kümesi | Tablo | Sınıf |
| Nitelik | Kolon/Özellik | Değişken |
| - | S.Yordam/Fonksiyon | Yöntem |

# Yeni Veri Modelleri

- ✓ **Object/Relational Model**
  İlişkisel modelle nesne yönelimli modelin birleştirilmesi sonucu ortaya çıkmıştır.
- ✓ **XML**
  Çoğunlukla, farklı platformlar arası veri değişimi için kullanılan veri tanımlama standardıdır. Yapısal olmayan verileri tanımlamak için de kullanılır.
- ✓ **JSON**
  Çoğunlukla, farklı platformlar arası veri değişimi için kullanılan veri tanımlama standardıdır. Yapısal olmayan verileri tanımlamak için de kullanılır.
- ✓ **NoSQL**
  İlişkisel modelin yetersiz kaldığı büyük hacimli verilerin yönetimi için tercih edilir.

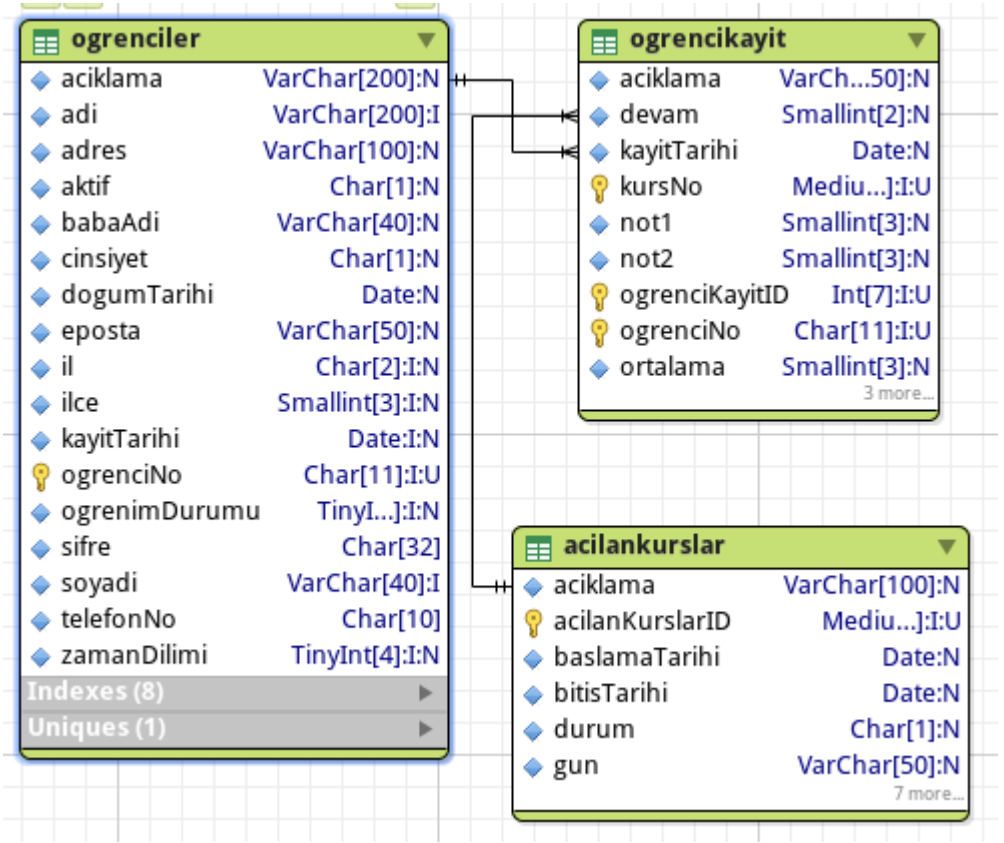
# Veri Soyutlama

**Veri modellerinin daha iyi anlaşılabilmesini sağlamak amacıyla ANSI-SPARC** (American National Standards Institute, Standards Planning and Requirements Committee), 1970'lerin başında, veri soyutlamanın 3 düzeyini tanımlamıştır.

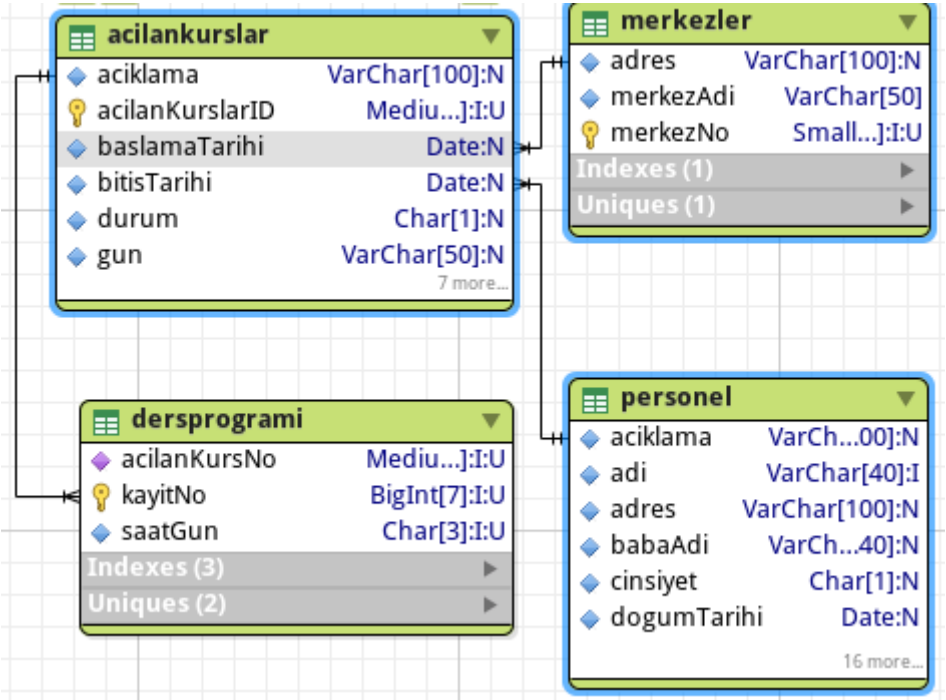
## 1. Harici Model (External Model)

Veritabanının uç kullanıcılar açısından görünen kısmı. Veritabanının sadece kullanıcıyla ilgili alt bölümlerini ifade eder.

**Öğrenci Ders Kaydı**

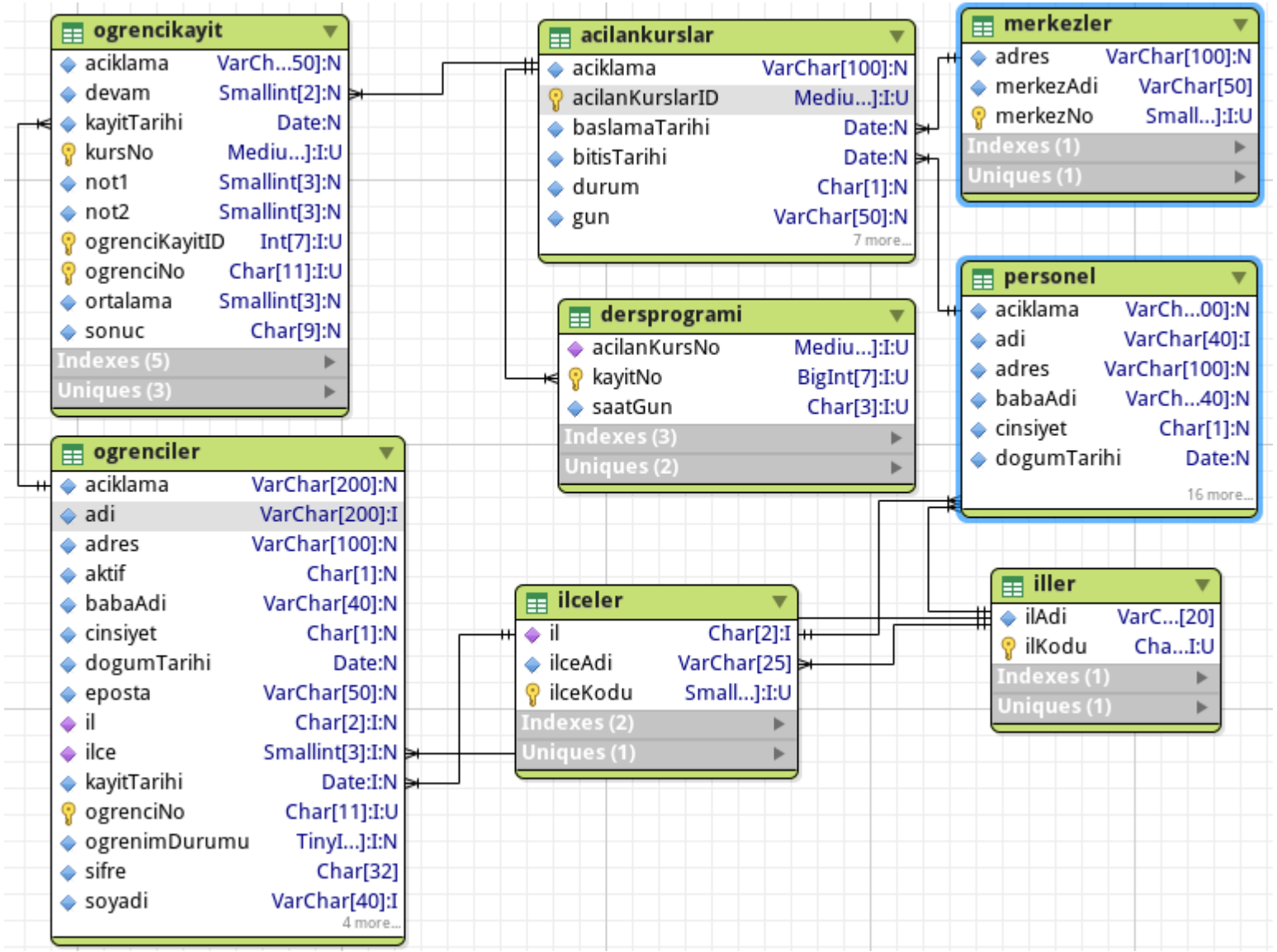



**Ders Programı**



# 2. Kavramsal Model (Conceptual Model)

- ✓ Veritabanının veritabanı tasarımcısı açısından görünen kısmı. Veritabanının tüm alt bölümlerini birleştirerek global olarak görünmesini sağlar.
- ✓ Varlık Bağıntı Diyagramı (VBD – ERD) ile gösterilir. Kullanılan yazılım (DBMS) ve donanımdan bağımsızdır. Donanım ya da yazılım değişikliği kavramsal model tasarımını etkilemez.
- ✓ Kavramsal model mantıksal görünüş olarak da kullanılır.



# 3. Dahili Model (Internal Model)

- ✓ Veritabanının, Veritabanı Yönetim Sistemi tarafından görünen kısmı.
- ✓ Dahili model = ilişkisel model
- ✓ Donanım bağımsız, yazılım bağımlı.

```
-- CREATE TABLE "dersprogrami" ----------------------------
CREATE TABLE `dersprogrami` (
        `kayitNo` BigInt( 7 ) UNSIGNED ZEROFILL AUTO_INCREMENT NOT NULL,
        `acilanKursNo` MediumInt( 6 ) UNSIGNED ZEROFILL NOT NULL UNIQUE,
        `saatGun` Char( 3 ) NOT NULL UNIQUE,
        PRIMARY KEY ( `kayitNo` ),
        CONSTRAINT `acilanKursNo` UNIQUE( `acilanKursNo`, `saatGun` ) )
ENGINE = InnoDB
AUTO_INCREMENT = 97;
-- ---------------------------------------------------------


-- CREATE TABLE "duyurular" -------------------------------
CREATE TABLE `duyurular` (
        `duyuruNo` Int( 7 ) UNSIGNED AUTO_INCREMENT NOT NULL,
        `personelNo` Char( 11 ) NOT NULL,
        `konu` VarChar( 50 ) NOT NULL,
        `eklemeTarihi` Date NOT NULL,
        `sonTarihi` Date NOT NULL,
        `icerik` VarChar( 10000 ) NULL,
        PRIMARY KEY ( `duyuruNo` ) )
ENGINE = InnoDB
AUTO_INCREMENT = 6;
-- ---------------------------------------------------------
```

# Kaynaklar


- ✓ Carlos Coronel, Steven Morris, and Peter Rob, Database Systems: Design, Implementation, and Management, Cengage Learning.