



Bilgisayar Programcılığı

Veritabanı ve Yönetimi Sistemi I

Öğretim Görevlisi Mustafa ÇORUH

Vize Sınavı için Proje Ödevi

**SQL Script ile db_PMTP (Personel Maaşları Takip Programı)
Veritabanını oluşturma, tabloları yaratma ve verileri girme..**

Detaylar <http://webders.mustafacoruh.com/Pdfs/Ders/Proje-BLG117-VT1.pdf>
sitesindedir.



Personel Maaş Takip Program (PMTT) Tabloları

dbo_tbl_Personeller

| Pers_ID | Pers_Adi | Pers_Soyadi | Pers_Isim | Pers_DTarih | Pers_Ise_Gi | Pers_Isten | Pers_Adresi | Pers_Kenti | Pers_Ili | Pers_Il_Kod | Pers_Tel | Pers_... |
|---------|----------|-------------|---------------|-------------|-------------|------------|-----------------|-------------|-----------|-------------|------------|----------|
| 1 | Mustafa | Çoruh | Mustafa Çoruh | 01.01.1962 | 06.01.2016 | | Uzun Sok. No 1 | Kdz. Ereğli | Zonguldak | 67 | 3721112222 | 537111 |
| 2 | Buğra | Diniz | Buğra Diniz | 01.01.1998 | 07.01.2016 | | Kısa Sok. No 2 | Merkez | Bartın | 74 | 3721115555 | 535333 |
| 3 | Burak | Oğuz | Burak Oğuz | 01.01.1993 | 08.01.2016 | | Alt Sok. No 3 | Beylikdüzü | İstanbul | 34 | 3721114444 | 536222 |
| 4 | Ayşe | Meliha | Ayşe Meliha | 01.01.1968 | 10.01.2016 | | Yukarı Sok No 4 | Sincan | Ankara | 06 | 3721113333 | 536166 |

Kayıt: 1/4 Filtre Yok Ara

dbo_tbl_Bolumler

| Bolum_ID | Bolum_Adi | Bolum_Tel | Mudur_ID |
|----------|----------------|------------|----------|
| 1 | Bilişim Sistem | 3721112222 | 1 |
| 2 | Pazarlama | 3722223333 | 2 |
| 3 | Satış | 3723334444 | 3 |
| 4 | Muhasebe | 3724445555 | 4 |
| 5 | Finans | 3725556666 | 4 |
| 6 | Yönetim | 3726667777 | 4 |
| * | Yeni | | |

Kayıt: 1/6 Filtre Yok Ara

dbo_tbl_PersonelMaaslari

| Maas_ID | Pers_ID | Maas_Oden | Maas_Tutari | Maas_Komi | Ay_ID | Maas_Topla | Maas_Yili | Ekl... |
|---------|---------|------------|-------------|------------|-------|------------|-----------|--------|
| 1 | 1 | 01.07.2020 | 3.540,00 ₺ | 0,00 ₺ | 7 | 3.540,00 ₺ | 2020 | |
| 2 | 1 | 01.08.2020 | 3.540,00 ₺ | 0,00 ₺ | 8 | 3.540,00 ₺ | 2020 | |
| 3 | 1 | 01.09.2020 | 3.540,00 ₺ | 0,00 ₺ | 9 | 3.540,00 ₺ | 2020 | |
| 4 | 1 | 01.10.2020 | 3.540,00 ₺ | 0,00 ₺ | 10 | 3.540,00 ₺ | 2020 | |
| 5 | 1 | 01.11.2020 | 3.540,00 ₺ | 0,00 ₺ | 11 | 3.540,00 ₺ | 2020 | |
| 6 | 1 | 01.12.2020 | 3.540,00 ₺ | 0,00 ₺ | 12 | 3.540,00 ₺ | 2020 | |
| 7 | 2 | 01.11.2020 | 2.500,00 ₺ | 1.200,00 ₺ | 11 | 3.700,00 ₺ | 2020 | |
| 8 | 2 | 01.12.2020 | 2.500,00 ₺ | 1.000,00 ₺ | 12 | 3.500,00 ₺ | 2020 | |
| 9 | 3 | 01.12.2020 | 3.000,00 ₺ | 1.000,00 ₺ | 12 | 4.000,00 ₺ | 2020 | |
| 10 | 4 | 01.12.2020 | 3.500,00 ₺ | 0,00 ₺ | 12 | 3.500,00 ₺ | 2020 | |
| * | Yeni | | | | | | | |

Kayıt: 1/10 Filtre Yok Ara

dbo_tbl_Kategoriler

| Kategori_ID | Cinsiyet | Unvan | Kent_Adi | Il_Kodu | Il_Adi | Ulke | Ay_Adi | Kitap_Turu |
|-------------|----------|-----------------|------------|---------|-----------|-----------|---------|------------|
| 1 | Erkek | VT Yöneticisi | Merkez | 01 | Adana | Türkiye | Ocak | Bilişim |
| 2 | Kadın | Satış Elemanı | Kdz.Ereğli | 67 | Zonguldak | ABD | Şubat | Kentleşme |
| 3 | | Pazarlamacı | Alaplı | 34 | İstanbul | İngiltere | Mart | Bilim |
| 4 | | Ofis Yöneticisi | Beylikdüzü | 81 | Düzce | Kanada | Nisan | Öğretim |
| 5 | | CIO | Kızılay | 05 | Amasya | Meksika | Mayıs | Yönetim |
| 6 | | CEO | Sincan | 06 | Ankara | Almanya | Haziran | Veritabanı |

I. Hafta

Veritabanı Kavramları



Tablo-1.1: Veritabanları Terminolojisi
(Kroenke-Auer, 2012)

| İVT Tasarımcısı | Programcı | Kullanıcı |
|------------------------|------------------|----------------------|
| İlişki (Relation) | Dosya (File) | Tablo (Table) |
| Satır (Tuple-Row) | Kayıt (Record) | Satır-Dizeç (Row) |
| Nitelik (Attribute) | Alan (Field) | Sütun-Dikeç (Column) |



Tablo-1.2: Veritabanı Kavram ve Terimleri

| Terim | Kısaltma | Tanım |
|----------------------------------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Veri (Data) | | İşlenmemiş ham (raw) bilgiye denir. |
| Enformasyon (Information) | | Şekil ve içeriği belli bir kullanıma uygun manası olan ve aynı zamanda işlenmiş ve kodlanmış veriye denir. |
| Bilgi (Knowledge) | | Kodlanamayan veya yazılı hale getirilemeyen, sadece insan beyninde bir anlamı olan kavramları ifade eder. |
| Veritabanı (Database) | VT (DB) | Belirli bir konuda ve amaç doğrultusunda ilgili verilerin toplanıp işlendiği tümleşik kayıtlar kümesidir. Bir veritabanının amacı, insanların işlemleri takip etmelerine yardımcı olmaktır. Bir veritabanının gerçek gücü, verileri istenen şekilde gösterebilmesidir. |
| Veritabanı Yönetim Sistemi (Database Management System) | VTYS (DBMS) | Veritabanlarına erişimi sağlayan, yönetimini, güvenliğini ve denetimini gerçekleştiren ve verilerin işlenmesini, depolanmasını ve raporlanmasını sağlayan bilgisayar yazılımlarına denir. |
| Veritabanı Sistemi (Database System) | VTS (DBS) | Veritabanıyla ilgili veritabanı dosyalarına, Veritabanı Yönetim Sistemi (VTYS) yazılımlarına ve bunları organize eden Veritabanı çalışanlarının tümünü içeren sisteme denir. |



Tablo-1.2: Veritabanı Kavram ve Terimleri

| Terim | Kısaltma | Tanım |
|------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tablo (Table) | | İlişkisel veritabanı sistemlerinde, aynı özelliklere sahip veri kümelerinin saklandığı yapılara denir. Tablolar dikeçler (sütun) ve dizeçler (sıra)'den oluşur. |
| Kayıt-Satır-Dizeç (Record) | | Alanların birleşmesi ile oluşan her bir satırdaki veri kümesine kayıt (row, tuple) denir. |
| Alan-Sütun-Dikeç (Column) | | Veritabanında, tablolara yerleştirilen her kayıt için tanımlanmış olan her bir veri alanı veya niteliktir. Nitelikler işlevlerine göre farklı türlerde olabilir. |
| Anahtar Alan (Primary Key) | AA (PK) | Tablo içindeki her kaydı tek (unique) olarak tanımlamaya yarayan alan (Primary Key) veya alanlar gurubuna denir. |
| Referans Alan (Foreign Key) | RA (FK) | Diğer bir tabloya ait anahtar alana referans veren alana (Foreign key) denir. |
| Varsayılan Değer (Default) | VD (DF) | Herhangi bir sütuna/alana NULL değeri girmek yerine belirlenen değer girilmesini sağlayan VT nesnelidir. |



Tablo-1.2: Veritabanı Kavram ve Terimleri

| Terim | Kısaltma | Tanım |
|------------------------------------------------------------|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Dizin (Index) | DZ (IND) | Dizinler bir tablodaki kayıtlara, belirli bir alan verileri üzerinden daha hızlı erişimi sağlayan VT nesnelere dir. |
| Görünüm-Sorgu (View) | | Bir ya da birden fazla tablo ya da diğer bir görünümdeki verinin özelleştirilmiş gösterimini sağlayan kayıt altına alınmış sorgulardır. Görünümler veri içermezler verilerini kendilerine temel teşkil eden tablolardan alırlar. |
| Saklı Yordam-İşlem (Stored Procedure) | SY (SP) | VT’de herhangi bir işlemi veya işlemleri yerine getirmek için yazılan SQL kodlarını saklayan VT nesnelere dir. SY’lerin temel amacı tekrarlanan işlemlerin bir defa yazılarak VT içinde veya uygulama programlarıyla tekrar tekrar kullanılmasını sağlamaktır. |
| Tetikleyici (Trigger) | | Tetikleyiciler VT’de belirli bir işlemin hemen ardından otomatik olarak devreye giren işlem kodlarından oluşan VT nesnelere dir. Tetikleyici nesnesi, bir tablo üstünde tanımlanabilen ve veri üzerinde oluşan bir hareketten (Transactions) önce ve sonra çalışabilen SQL kodlarıdır. |
| Yapısal Sorgulama Dili (Structured Query Language), | YSD (SQL) | İlişkisel Veritabanlarındaki verinin depolanması, işlenmesi, değiştirilmesi, sorgulanması gibi işlemlerin yapılmasını sağlayan yapısal bir sorgu/bilgisayar dilidir. |
| Varlık-İlişki Diyagramı (Entity-Relation Diagram) | VİD (ERD) | Gerçek dünyadaki varlıkları, varlıktan tanımlayan nitelikleri ve bunlar arasındaki bağıntıların modellenmesini sağlayan diyagramlara denir. |



Tablo-1.2: Veritabanı Kavram ve Terimleri

| Terim | Kısaltma | Tanım |
|--------------------------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Veri Seti (Data set) | | Kayıtlardan oluşan kümeye denir. Bu küme, bir tablonun tamamı olabileceği gibi bir bölümü ya da birden fazla tablodan seçilmiş alanlardan oluşan bir kayıt kümesi ya da grubu da olabilir. |
| Üst Veri (Meta Data) | | Veritabanı ve nesnelere hakkındaki verilere denir. Örneğin tablo, görünüm, dizin, tetikleyici, nitelik, varsayılan değer isimleri ve bunların özellikleri hakkındaki veriler. |
| Normalizasyon | | VT oluşturacak verilerin belirli aşamalardan geçirilerek ilişkisel VT modeline dönüştürülme sürecine denir. 3. Normal Form (3NF) kabul edilen en alt seviyedir. |
| İşlemler (Transactions) | | Olayların ve hareketlerin temsilidir. |
| İlişki (Relationship) | | İVT'lerdeki tablolar, alanlar ve veriler arasındaki ilişkileri tanımlar. 1:1 (Bire-Bir), 1:N (Bire-Çok) ve N:M (Çoka-Çok) türleri vardır. |

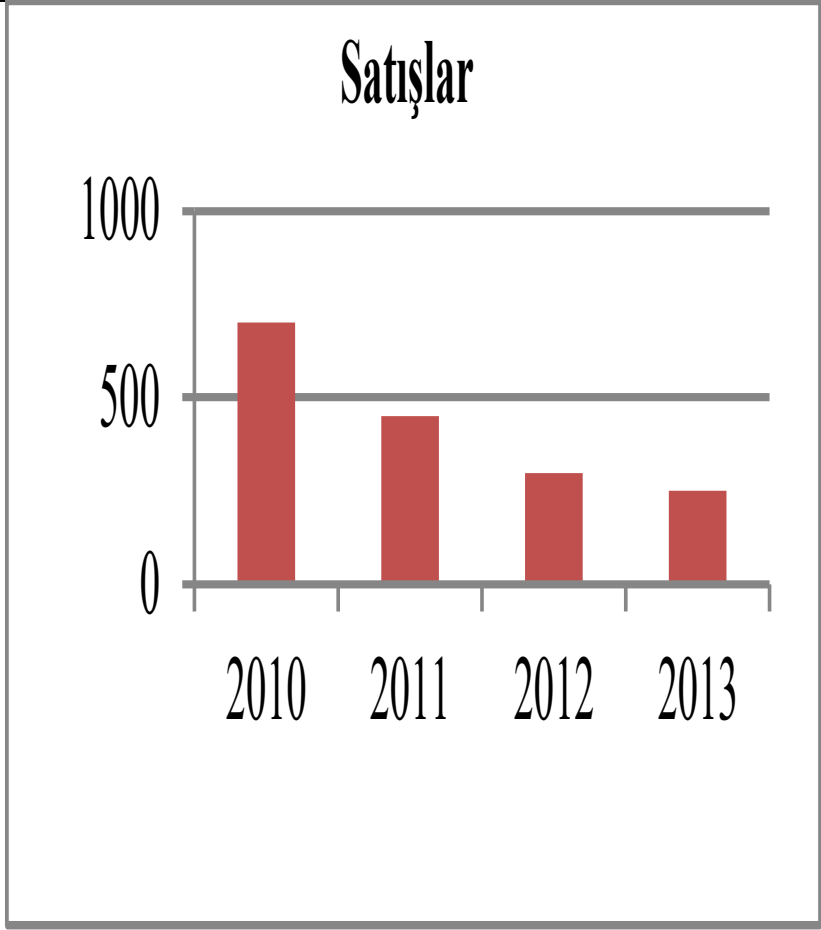


Veri, Enformasyon ve Bilgi

- Veri ve Enformasyon kodlanarak sayısal yazı haline getirilebilen ve dolayısıyla BT araçlarıyla işlenebilen kavramları, Bilgi ise kodlanamayan veya yazılı hale getirilemeyen, sadece insan beyinde bir anlamı olan kavramı ifade etmektedir
- Bilgi, insanın etrafındaki dünyayı şekillendirme, onu sınıflandırma, yaşamlarındaki belirsizliği azaltma ve belirli bir biçimde dünyayı yorumlamak için veri ve enformasyon yardımıyla beyinlerinde oluşturdukları anlamlardır (Beijerse, 1999).
- Uzmanlık, Larousse'nin yaptığı tanıma uygun olarak düşünme, araştırma, gözlem, deney, akıl yürütme sonucunda elde edilen düşünsel ürünün organize hale getirilmiş biçimi olarak tarif edilebilir (Tutar, 2010:53). Kısacası bilginin organize olmuş halidir.
- Bilginin felsefi manasını kavramak ise bilgeliktir. Ayrıca bilgelik aşamasına kadar doğru yapmak olgusu baskın iken bilgelik aşamasında artık doğru şeyi yapmak olgusu öne çıkmaktadır. Bilgelik, (Laudon, 2014:417) deyimiyile bilginin nerede, ne zaman ve nasıl uygulanacağını bilmektir.
- Bilginin değeri güncel olmayan veya yetersiz bilgiyle alınan kararların verdiği zararın ölçüsüyle ters orantılıdır. Veri toplama, enformasyon üretme ve bilgi yaratımı sürecinin bir maliyeti vardır.
- Elemanlar, bilgi açısından farklı kanalların oluşturulması konusunda çekimser davrandıklarında, bu, kendi bilgilerini paylaşma konusunda isteksiz oldukları anlamına gelir (Celep, 2003:106).
- Verinin BT araçlarıyla sayısallaştırılarak kodlanması insanla insan, makineyle makine ve insanla makine arasında kolayca iletişim kurulmasını sağlamaktadır. Enformasyonun bu şekilde kişiler, BT araçları ve hatta nesiller arasında kolayca aktarılmasıyla daha büyük toplumsal, finansal ve entelektüel değerlerin yaratılması sağlanmaktadır.
- Zaman içinde geliştirilen teknolojilerle (Uzman sistemler gibi) bugün insan beynindeki bazı örtük bilgiler kayıt altına alınarak enformasyon haline getirilebilmektedir. Bu işlemin son noktası insan beyinin kopyalanmasıyla sağlanabilecektir. Bu kopyalamayı sağlayacak teknolojiye “Bilgi Teknolojisi” denebilir.



Şekil-1.2: Veri, Enformasyon ve Bilgi'ye Örnekler (Çoruh, 2016:19)

| Veri | Enformasyon | Bilgi | | | | | | | | | | |
|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------|----------|------|-----|------|-----|------|-----|------|-----|----------------------------------------------|
| 2010 – 500 2011 – 450 2012 – 300 2013 – 250 |  <p>Satışlar</p> <p>1000 500 0</p> <p>2010 2011 2012 2013</p> <table border="1"><thead><tr><th>Yıl</th><th>Satışlar</th></tr></thead><tbody><tr><td>2010</td><td>500</td></tr><tr><td>2011</td><td>450</td></tr><tr><td>2012</td><td>300</td></tr><tr><td>2013</td><td>250</td></tr></tbody></table> | Yıl | Satışlar | 2010 | 500 | 2011 | 450 | 2012 | 300 | 2013 | 250 | Satışlarda düşüş var. Acilen önlem alınmalı. |
| Yıl | Satışlar | | | | | | | | | | | |
| 2010 | 500 | | | | | | | | | | | |
| 2011 | 450 | | | | | | | | | | | |
| 2012 | 300 | | | | | | | | | | | |
| 2013 | 250 | | | | | | | | | | | |
| 05091962 | 05.09.1962 | İşe almak için yaşlı | | | | | | | | | | |



Bilginin Çalışanlar İçin Özellikleri (Tutar, 2010:66)

- Bilgi çalışan için değerlidir.
- Bilgi kıttır (veri ve enformasyon ise boldur).
- Bilgi kolayca taklit ve kopya edilemez ve kullanılamaz, kişinin müsaadesine bağlıdır.
- Bilgi ikame edilemez, özel bir olay ve duruma bağlı olarak değeri vardır.



Sayısallaştırma ve Verileştirme

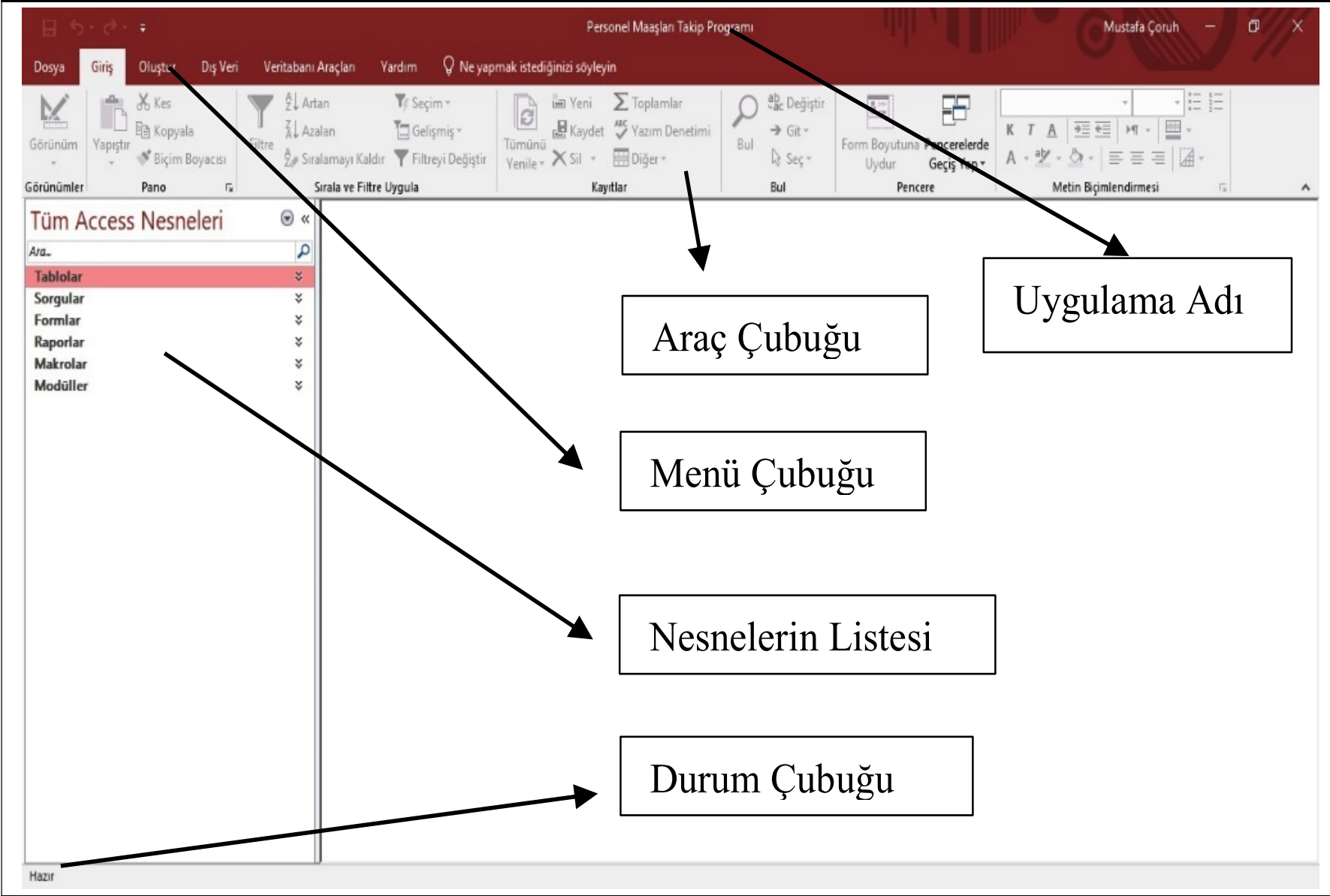
- Verileştirme, verinin çizelgelenebileceği, analiz edilebileceği ve yeni durumlarda yeniden kullanılabilir şekilde nicel bir formata (text dosyası) sokulmasıdır (Schönberger-Cukier, 2013:85).
- Burada tanımlanan verileştirme, özellikle analog olarak kaydedilmiş verinin (kâğıt üzerindeki resim veya yazı gibi) bilgisayarın kullanabileceği şekilde sıfır ve birden oluşan ikili koda dönüştürme süreci olan sayısallaştırmadan (dijitalleştirmeden) çok farklıdır.
- Sayısallaştırma, verinin 0 ve 1'ler yardımıyla kaydedilmesi iken verileştirme kaydedilmiş verilerin enformasyon ve bilgi üretiminde kullanılacak veya işlenecek şekilde sayısal olarak kaydedilmesi olarak düşünülebilir.
- 2011'de Science dergisinde rapor edilen bir çalışmada, iki yılı aşkın bir sürede, 84 ülkeden 2,4 milyon insandan gelen 509 milyon tweet'in analizi, insanların ruh durumlarının dünyada bir uçtan diğer uca bütün kültürlerde günlük ve haftalık benzer modeller izlediğini göstermiştir.
- Veri toplama ve analizi tüm toplumların öğrenmesi gereken zaruri bir ihtiyaç haline gelmektedir. Büyük veriyle, verinin değeri değişirken sayısal çağda veri, işlemleri destekleme rolünden ticareti yapılan malın kendisi haline gelmektedir. Verinin değeri, birincil kullanımından gelecekteki potansiyel kullanımlarına kaymaktadır (Schönberger-Cukier, 2013:103-106).
- Veri maddi şeylerden (yenen yemek, yanan bir mum) farklı olarak değeri kullanıldığında azalmamakta ve tekrar tekrar kullanılabilir. Veri, iktisatçıların 'rakip olmayan' mal dedikleri şeydir.



MS-Access Tanıtımı

- Access, Microsoft (MS) firmasının Office Profesyonel Yazılım paketi içerisinde yer alan Veritabanı Yönetim Sistemidir (VTYS).
- MS-Access'in çalışabilmesi için bilgisayarda Microsoft Windows İşletim sisteminin kurulmuş olması gerekir.
- Access Ticari olarak Office Profesyonel Programında yer alır. Office'in yüklenmesi esnasında tüm diğer Office uygulamaları gibi isteğe bağlı olarak yükleme yapılabilir. Kullanımı pek çok veritabanı yazılımına göre daha kolaydır.
- MS-Access VTYS, verileri birden çok ve birbirleriyle ilişkili tablolarda tutarak bunların kolayca sorgulanmasını sağlayan ilişkisel veritabanı oluşturur.
- Access, aynı zamanda bir uygulama program arayüzü de oluşturmayı sağlayan kişisel bir veritabanıdır.





Şekil-7.2: MS-Access VT Nesneleri



Uygulama

MS-Access ile Veritabanı
(VT) Oluřturma ve Öğrenci
Verilerini Tutacak bir Tablo
Oluřturma



II. Hafta

Veritabanlarına (VT) Giriş



Veritabanı Sistemi (VTS)

- Veritabanı
- Veritabanı Yönetim Sistemi (VTYS)
- Çalışanlar





Uygulama Programları

VTBT Yöneticileri (VT ve uygulama yazılım yönetimi ve güvenliği)



Programcılar (Program ve VT tasarımı)

Uygulama Programları

Kullanıcılar (Veri girişi, güncelleme ve rapor alma)

Çalışanlar

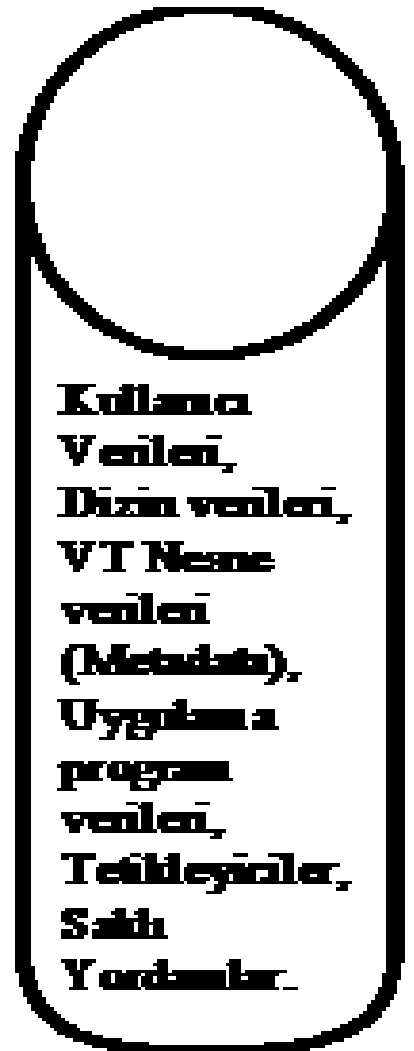
Tasarım Araçları
VT, Tablo, Dizin,
Kolon, Sınırlama,
Vaziyet Değer,
Sorgu, Trigger,
Stored procedure
Çalıştırma Araçları
(SQL),
Form, rapor ve
sorgu tasarımı
Araçları,
Programlama dilleri,
Yardımcı yazılımlar.

Çalıştırma Araçları
Form İşlemcisi,
Sorgu İşlemcisi,
Rapor İşlemcisi,
Program
Derleyicileri,
Yardımcı
programlar,
İnternet,
İletişim ve Ağ
Sistemleri.

V
T
Y
S

M
o
t
o
r
u

VTYS



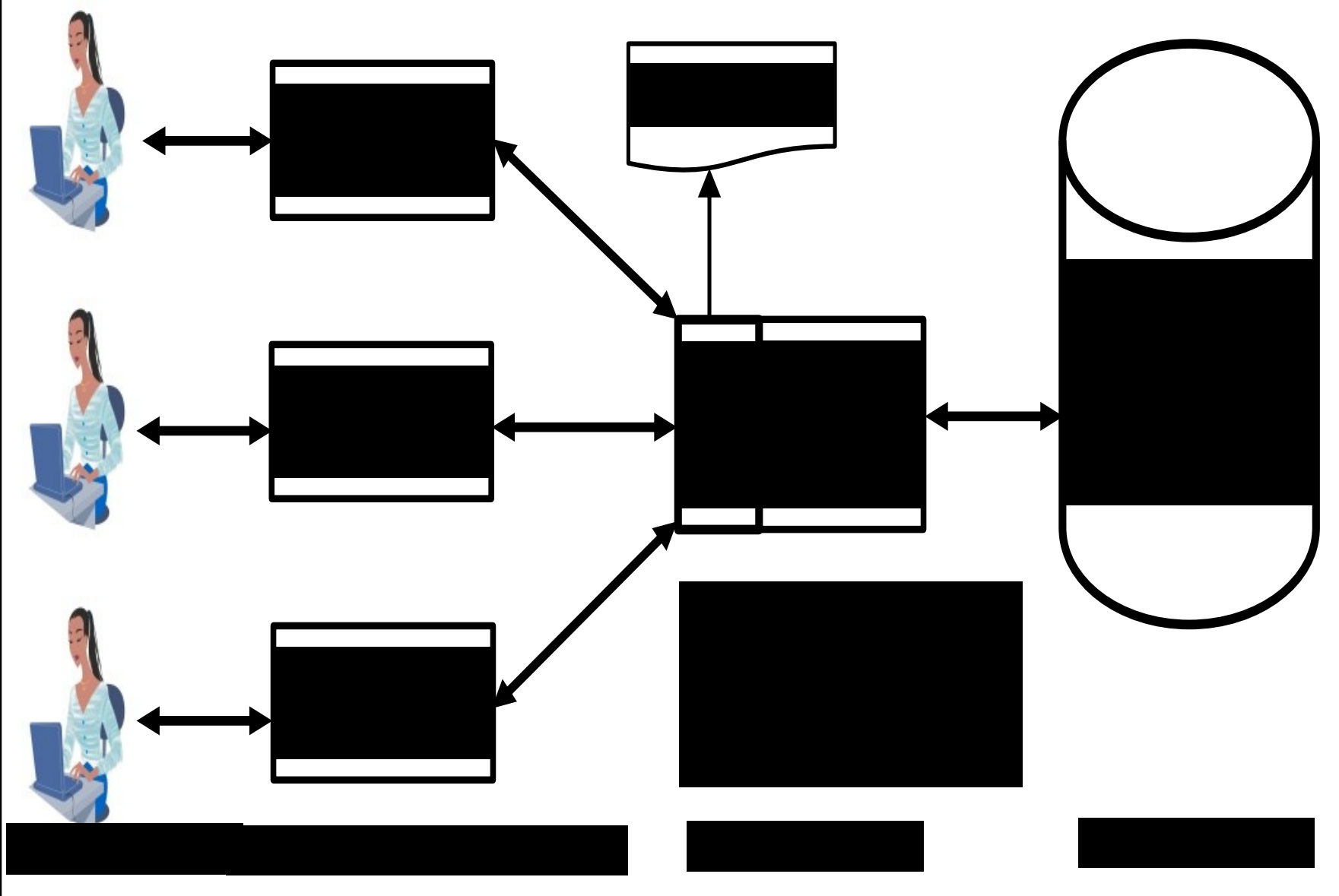
Kullanıcı Verileri,
Dizin verileri,
VT Nesne verileri
(Metadata),
Uygulama program verileri,
Tekilleştiriciler,
Sabit Yordamlar.

Veritabanı

Şekil-1.4: Veritabanı Sistem Elemanları (Kroenke, 2000, s. 11)



Şekil-1.5: VTYS'de Veri İşleme Süreci
(Kroenke, 2000:11 & Alp-Özdemir-Kilitçi, 2011:19)



Veritabanı (VT) Nedir?

- Veritabanı, belli bir konuda, birbiriyle ilişkili kayıtlar topluluğudur (Kroenke, 2000).
- Veritabanlarının temel fonksiyonu insanların olayları (satış veya stok hareketleri, öğrenci notlarını kaydetme, bankalardaki hesap hareketleri, muhasebe işlemleri gibi) takip etmesine, istedikleri şekilde sıralamasına, seçmesine ve güncellemesine yardım etmektir.
- VT'ler verilerin iniş çıkışına, gidişatına ve eğilimlerine bakarak bazı bilinmeyenleri azaltma yoluyla doğru karar verme oranını artırmak için kullanılmaktadır.
- VT'ler enformasyon ve bilgi üretimine yardım ederler.
- VT'lerin en önemli faydası kaydedilmiş verileri istenilen sıra ve şekilde kullanıcıya sunabilmesidir (Coruh, 2000).



Veritabanlarının Çıkış Sebebi

- Yapılan son hesaplar göstermektedir ki insanlığın elindeki kayıtlı Veri miktarı her yıl, Enformasyon miktarı ise her 10 yıl civarında (2-3 yılda olduğunu iddia edenlerde vardır) ikiye katlanmaktadır.
- İnsanın emrindeki bu veri ve Enformasyon birikimi Veritabanları sayesinde olmaktadır. Bu yüzden hızlı ve büyük bir yığın halinde biriken verilerin saklanması ve işlenmesi için VT ve VTYS yazılımları geliştirilmiştir.
- Sonuç olarak insanlığın elinde verinin çoğalması ve analiz ihtiyacı nedeniyle bunların kullanımını sağlayacak Veritabanları ortaya çıkmıştır.



Veritabanı İşleme Fonksiyonları

- Verinin Kaydedilmesi
- Verinin Transferi
- Verinin Depolanması
- Verinin Tekrar Çağrılması
- Verinin İşlenmesi
- Verinin Gösterilmesi



Veritabanlarının Bozulma Sebepleri (Coruh, 2000)

- Kurumlardaki Elektrik kesintisi,
- İşletim sisteminden düzgün çıkılmaması,
- Bilgisayar ağlarında oluşan hatalar,
- Virüsler,
- Dosya saklama anında aktif hale gelen mesajlar (screen savers, network mesajı gibi),
- Disk sürücüsündeki bozulmuş alanlar veya dosya atama tabloları,
- Korsan veya kasıtlı işlemler,
- VTYS ve İşletim Sistemi yazılımlarındaki hatalı kodlar (Bugs).



Veritabanı Yönetim Sistemi (VTYS)

- Veritabanı Yönetim Sistemi (VTYS), veritabanlarına erişimi sağlayan, yönetim ve güvenliğini gerçekleştiren ve denetleyen yazılımlara denir.
- VTYS, bilgisayarın fiziksel hafızasındaki verileri çeşitli özelliklerine göre gruplandırıp şekillendirdikten sonra saklayan ve işleyen programlardır.
- VTYS, kaydedilen ve depolanan bu veriyi, özellikle ilişkisel VT'lerde SQL komutları ile kullanıcıların istekleri çerçevesinde işler, yeniden şekillendirir ve sorgulama yapılmasını sağlarlar.



Veritabanı Yönetim Sistem Mimari Türleri (Alp, Özdemir, & Kilitçi, 2011, s. 14)

- Yerel-Lokal (Single): Paradox, dBase, Access
- İstemci/Sunucu (Client/Server): Oracle, Sybase, MS SQL Server ve MySQL
- Dağıtık: DB2, SQL Server, Oracle



VTYS Yazılım Örnekleri

- MS-Access
- MS SQL Server
- IBM DB2
- Oracle
- MySQL
- Informix
- PostgreSQL
- Sybase



Uygulama

MS-Access'te db_PMTP Veritabanı (VT) ve tbl_Personeller, tbl_Bolumler, tbl_PersonelMaaslari, tbl_Aylar Tablolarını Oluşturma ve Veri Girme



III. Hafta

Veritabanı Türleri ve Tarihçesi



Veritabanı Türleri

- Dosya İşlem veritabanları,
 - Hiyerarşik (Hierarchical) model,
 - Ağ Modeli (Network Model),
 - İlişkisel (Relational) veritabanları,
 - Nesnel (Object Oriented) veritabanları,
 - Web Tabanlı (XML) veritabanları,
 - Çok Boyutlu (Multidimensional) veritabanları (veri ambarları da denmektedir),
 - Büyük Veri (Big Data) veritabanları,
 - SQL Olmayan (NoSQL) veritabanları.
-
- Günümüzde Veritabanı endüstrisine bakıldığı zaman asıl yoğunlaşılması (konsantre) gereken konunun veri işleme hızının veya performansının artırılması olduğu görülmektedir.



Tablo-1.11: Geleneksel Yaklaşımın Avantajları ve Sakıncaları

| Avantajları | Sakıncaları |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">○ Herhangi bir programda problem oluştuğunda diğer programlar çalışmaya devam eder.○ Veri yapısı kolayca anlaşılabilir.○ Veriler kolayca sınıflandırılır. | <ul style="list-style-type: none">○ Veri tekrarı ve veri tutarsızlığına sebep olabilir,○ Veri kolayca paylaşılamaz,○ Uygulamalardaki yeni gereksinimler yalnız uzman kişiler tarafından karşılanabilir veya gerekli değişiklikler yapılabilir,○ Veriye erişim ve istenen veriyi elde etmede güçlükler yaşanabilir,○ Karmaşık veri saklama yapıları ve erişim yöntemlerini bilme zorunluluğu vardır,○ Bütünlük (integrity) sorunları oluşturur,○ Güvenlik ve gizlilik sorunları oluşturur,○ Tasarım farklılıkları standart eksikliği oluşturur,○ Yedekleme, yeniden başlatma ve onarma gibi işletim sorunları oluşturur,○ Verilerin birden çok kullanıcı tarafından (özellikle de aynı anda) paylaşılmasına müsaade etmez. |



Tablo-1.12: VT Yaklaşımının Avantajları ve Sakıncaları

| Avantajları | Sakıncaları |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">○ Verileri kolayca çağırır ve birleştirebilir,○ Olan verilere dokunmadan yeni verilerin eklenmesini sağlar,○ Verilerin tekrarını azaltarak SD’de yer tasarrufu sağlar,○ Veriyi kullanacak arayüz programlarından bağımsızlaştırır,○ Veriyi anlaşılabilir ve kullanıcıya yakın hale getirir,○ Verilerin yönetimini ekonomik hale getirir,○ Verilere kolay ve güvenli ulaşım ve güncelleme sağlar,○ Merkezi bir güvenlik sistemini kolayca oluşturur,○ Güvenlik ve gizliliği istenilen düzeyde sağlar,○ Ortak dosyalar sayesinde veri tekrarını önler,○ Veriler üzerinde merkezi denetim ve tutarlık sağlar,○ Veri paylaşımını sağlar,○ Fiziksel yapı ve erişim yöntemi karmaşıklığını, çok katmanlı mimarileri sayesinde kullanıcılardan gizler,○ Her kullanıcıya yalnız ilgilendiği verileri, alışık olduğu şekilde, kolay ve anlaşılır yapılarda sunar,○ Sunulan tasarım ve geliştirme araçları ile uygulama yazılımı geliştirmeyi kolaylaştırır,○ Veri bütünlüğü için gerekli olanakları ve mekanizmaların kurulmasını sağlar,○ Yedekleme, yeniden başlatma, onarma gibi işletim sorunlarına kolay çözümler getirir,○ Bellek hacminin optimum kullanılmasını sağlar,○ Çoklu güncelleme sağlar,○ Herhangi bir veritabanına çeşitli programlama dillerini kullanarak erişilmesini sağlar. | <ul style="list-style-type: none">○ Veriler kolayca değiştirilebilir,○ VT bozulursa tüm programlar çalışamaz hale gelir,○ Çok iyi bir güvenlik sistemi gerektirir,○ Veritabanı sisteminin kurulumu ve bakımı pahalıdır,○ Veritabanı sistemi içinde, bazı bileşenler iyi tasarlanmadığı takdirde, sistem bütün olarak ciddi bir şekilde başarısızlığa uğrayabilir. |



İlişkisel Veritabanları (İVT)

- İlişkisel Veritabanı (İVT) bütünleşik, yapılandırılmış, karşılıklı ilişkileri tanımlanmış, bir uygulama amacıyla oluşturulmuş ve disk ortamında saklanan veriler kümesi olarak özetlenebilir.
- İVT, ortamı oluşturan varlıklar (tablolar) ve onlar arasındaki ilişkilerin veri modelidir. İVT'de Veritabanının yapısı kullanılan soyut Varlık-İlişki (Vi) modeline göre belirlenir.
- İVT'nin doğru ve tutarlı bir biçimde çalışması ve işlemleri yerine getirmesine veri bütünlüğü adı verilir.



Codd'a Göre İlişkisel VT'lerin Temel Özellikleri (Codd, 1970, s. 9)

- Veri satır ve sütunlardan oluşan iki boyutlu tablolarda (relation) saklanır.
- Tüm kayıtlar tektir. Satır ve sütünün kesiştiği yerler tek değer içermelidir.
- Sütün sıralaması ve Kayıtların (satırların) sırası önemsizdir.
- Sütunlara kullanıldıkları veri alanlarına göre mantıksal isimler verilir. Personel için Pers_Adı, Pers_Soyadi ve Pers_DYeri gibi.



Sütunlar (Nitelikler)

| Pers ID | Pers Adi | Bolum ID |
|---------|----------|----------|
| 10 | Mustafa | 1 |
| 20 | Meliha | 2 |
| 30 | Buğra | 3 |

Satırlar
(Kayıtlar)

Veri

Anahtar Alan
Sütunu

Referans Alan
Sütunu

Şekil-2.1: İlişkisel Veritabanında Tablo Yapısı



İVT Fonksiyonları

- Veri Sorgulama (Query),
- Veri Ekleme (Insert),
- Veri Güncelleme (Update),
- Veri Silme (Delete),
- Veri Listeleme (Select),
- Veri Yazdırma (Print),
- Veritabanı nesnelərini (Database Object) oluřturma, deęiřtirme ve silme (Create, Alter, Drop),
- Veritabanına ve veritabanı nesnelere eriřim saęlama,
- Veritabanı bütünlüęünü (integrity), güvenlięini (security) ve tutarlılıęını saęlama,
- Veri iřlemeyi kontrol etme,
- Programcılar için kolayca veri tanımlayabilme, veriye ulařım ve güncelleme saęlama,
- Veri yöneticileri için güvenlik, yedek alma (backup) ve yeniden yükleme (restore) gibi yönetim fonksiyonları saęlamadır.



Nesne Tabanlı Veritabanları

- Nesne Tabanlı Programlamada, bir nesne hem veriyi hem de verinin deęiştirilmesi için gerekli olan izlekleri içermektedir. Birimin özelliklerine ek olarak dięer birimlerle olan ilişkileri de içermektedir. Bu özelliklerin tek birimde birleřtirilerek depolanmasına sarmalama denmektedir. Böylece bir nesne farklı veri kümelerine yerleřtirilebilir.
- Bilgisayar Destekli Tasarım (CAD: Computer Aided Design) gibi yazılımlarda yazı, grafik ve sesin birlikte kullanılması gerektięinden nesneye dayalı modeller kullanılmaktadır. Coęrafi Biliřim Sistemleri (GIS-Geographic Information Systems) gibi karmařık, multimedya bileřenlerinin yoęun olarak kullanıldıęı sistemlerde nesneye dayalı VTYS'ler daha etkin kullanılabilir (Burmaoęlu & Őeřen, 2010, s. 305).



Nesne Tabanlı Programlamanın 4 Özelliđi

- Soyutlama (Abstraction)
- Sarmalama/ Paketleme (Encapsulation)
- Miras Alma/Kalıtım (Inheritance)
- Çok Biçimlilik (Polymorphism)



XML Veritabanları

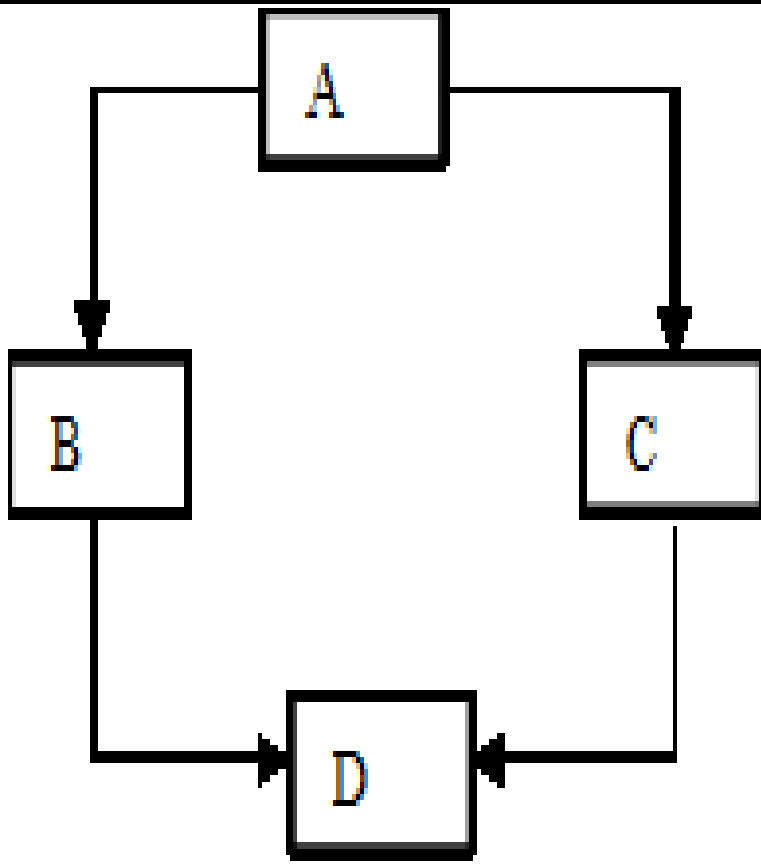
- Veritabanı işlemleri verilerin görüntülenebilmesi için doküman işlemlerine ihtiyaç duymakta ve doküman işlemleri de gösterecekleri verilerin saklanması için veritabanlarına ihtiyaç duymaktadırlar.
- Bu ihtiyacın giderilmesi XML teknolojileriyle sağlanmaktadır.
- XML veri, doküman yapısı ve gösterim arasındaki kesin ayrımları tanımlanmış bir teknolojidir. Bu amaçla DTD ve XLS standartlarını kullanır.
- XML standartları W3C tarafından belirlenip yayınlandığı için bir dünya standardı haline gelmiştir.



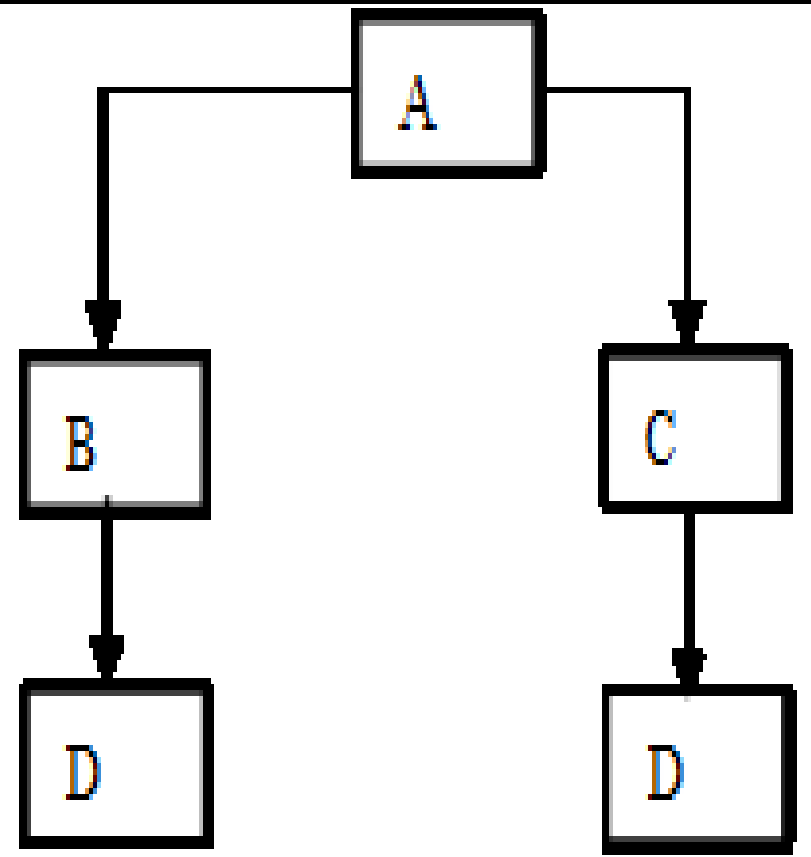
XML Veritabanlarının Özellikleri (Kroenke & Auer, 2012):

- Verilerin standart bir şekilde gösterilmesi,
- Verinin yapısı, verisi ve gerçekleşmesi arasındaki açık ayırım,
- Standart kodlama yapısının programla kontrol edilmesi,
- Herkes tarafından bilinen Standart bir XML dosya yapısının tanımlı olması.





İlişkisel VT



XML VT

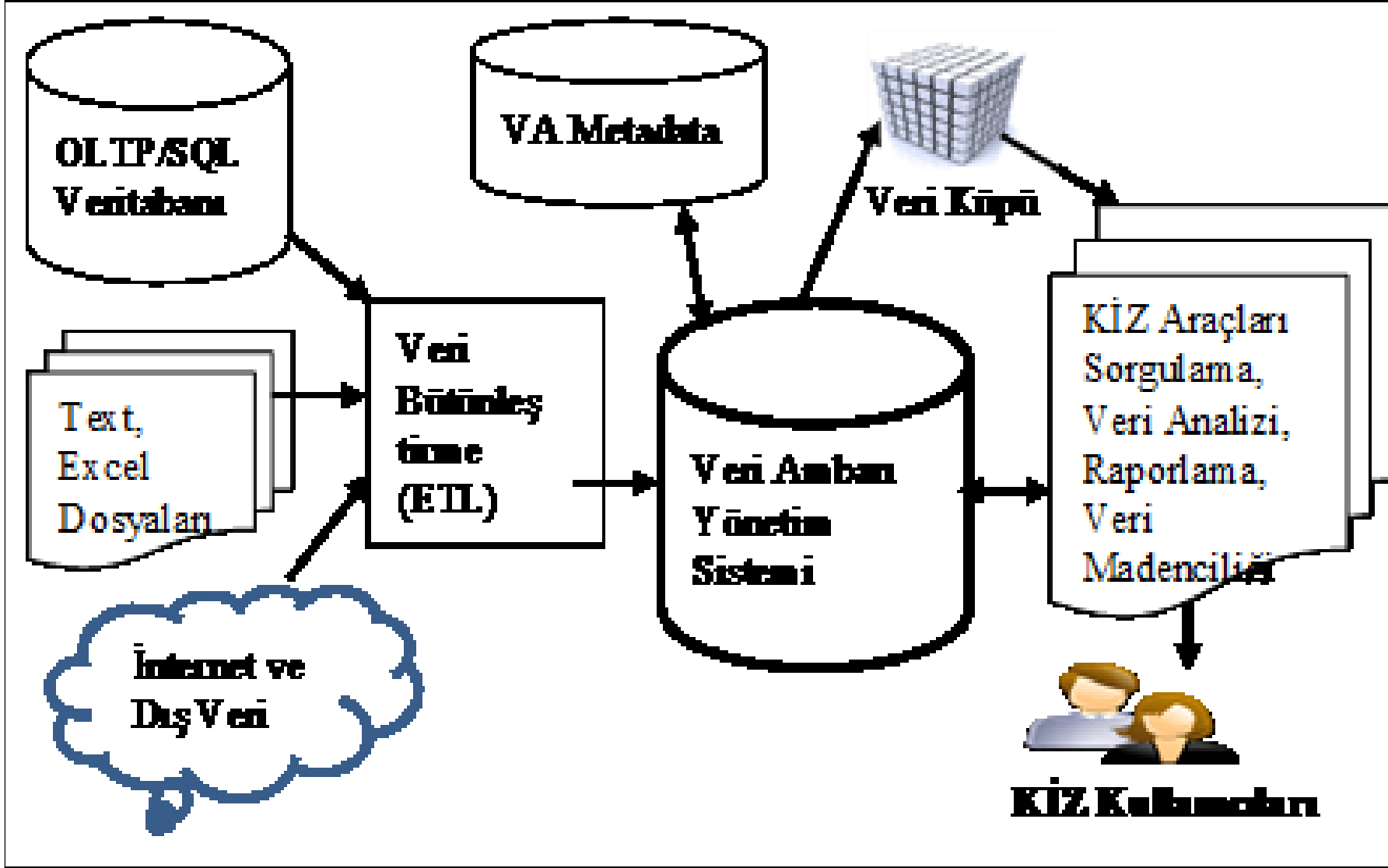
Şekil-2.3: İlişkisel VT ve XML Yapısında Veri (Nizam, 2016, s. 371)



Veri Ambarları (VA)

- Veri Ambarlarının (VA) temel amacı, dağıtık yapıda bulunan verilerin, veri kaynağı hangi biçimde olursa olsun, VA (veya KDS)'de bütünleştirilmesini sağlamaktır. Böylece, farklı sistemlerde, farklı biçimlerde tutulan verilere, tek bir sistem (VA) aracılığı ile bütünleşik olarak erişim sağlanır. Veriye sağlanan bu bütünleşik erişim, veri tekrarlarından ve farklı biçimlerde yapılan tanımlamalardan kaynaklanan, veriye hatalı bakışı da minimize eder.
- Veri ambarları veriye analitik bir bakış sağlayarak karar vericinin karar verme sürecini destekler. Böylece karar vericiler, operasyonel sistemlere müdahale etmeden ve operasyonel sistemlerdeki yapıları koruyarak, VA'ya aktarılan veriler üzerinden, analiz yapabilme imkânına kavuşurlar (Çağiltay & Tokdemir, 2010, s. 387).





Şekil-2.8: Veri Ambarı Elemanları ve Uygulama Süreci (Nizam, 2016, s. 377; Özkan, 2013, s. 31; Kroenke & Auer, 2012, s. 32)



Veri Ambarlarının 4 Temel Unsuru

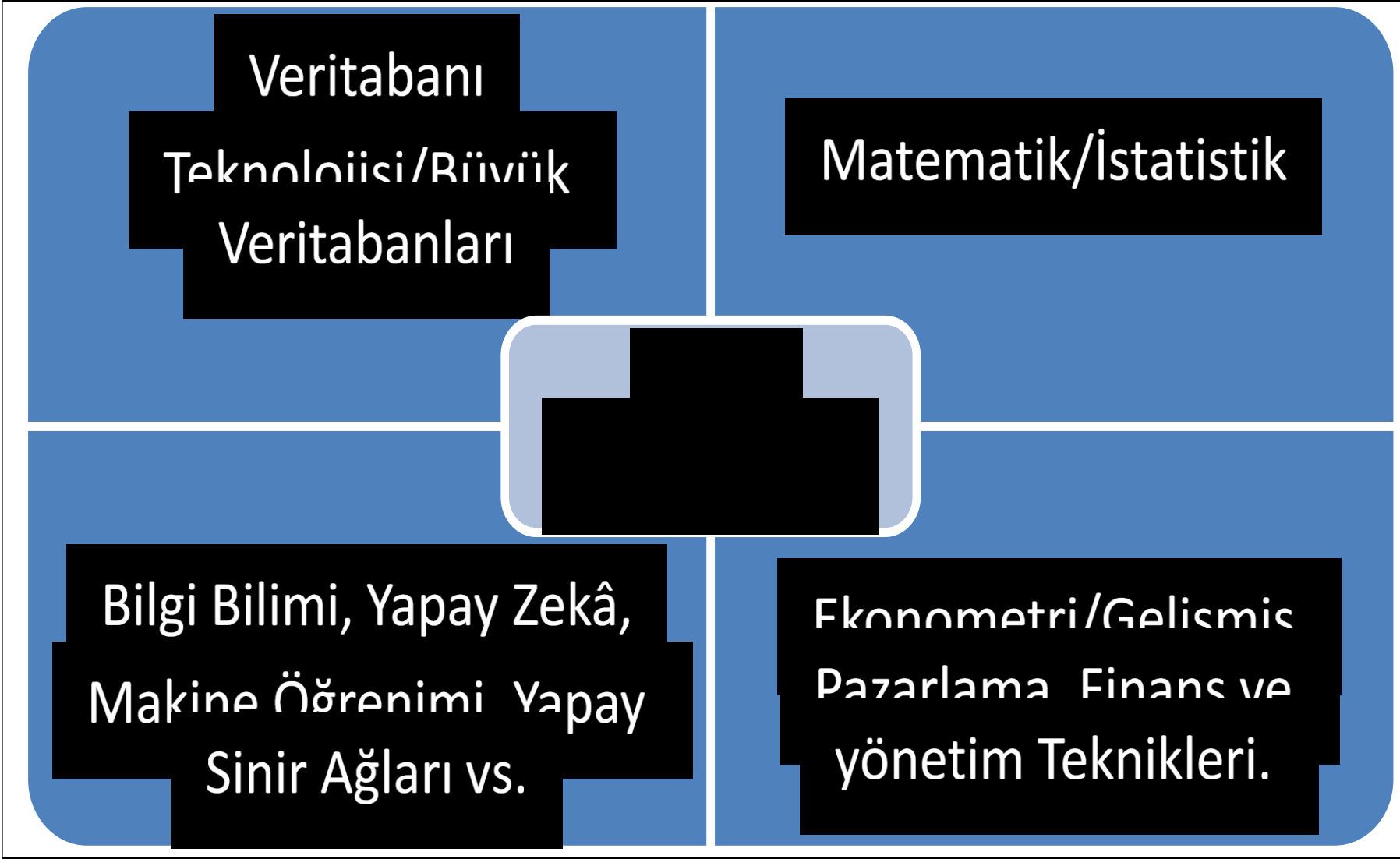
- **Hesaplama (Measures-facts)**
- **Boyutlar (Dimensions)**
- **Nitelikler (Attributes)**
- **Hiyerarşi (Hierarchies)**



Veri Madenciliđi (VM)

- Veri Madenciliđi (Data Mining-DM), VA'lardaki büyük veri kümeleri içerisindeki saklı desen ve ilişkilerin bulunması için yapılan analizlerin genel adıdır. Bunun için VM yazılımları kullanılır. Veri Madenciliđi (VM) geçmişini anlama, bugün ve gelecekte yapılacakları belirlemede yardımcı bir araçtır.
- Veriden Bilgi Keşfi (Knowledge Discovery from Data-KDD, Knowledge Discovery in Database-KDD) sürecinin çok önemli bir parçasıdır. VM işlemi için istatistik, makine öğrenmesi, yapay zekâ ve karar ağacı gibi yöntemler ayrı ayrı veya birlikte kullanılabilir (Nizam, 2016, s. 402).
- VM kullandığı yöntem ve teknikler sayesinde kurumlarda biriken veri içerisinde kurum için yararlı olanları bulup ortaya çıkarma işlemi olarak da tarif edilebilir (Özkan, 2013, s. 37).





Şekil-2.10: Veri Madenciliğinde Kullanılan Teknolojiler (Kroenke & Auer, 2012, s. 577; Tutar, 2010, s. 124)



Veri Madenciliđi İşlem Kategorileri

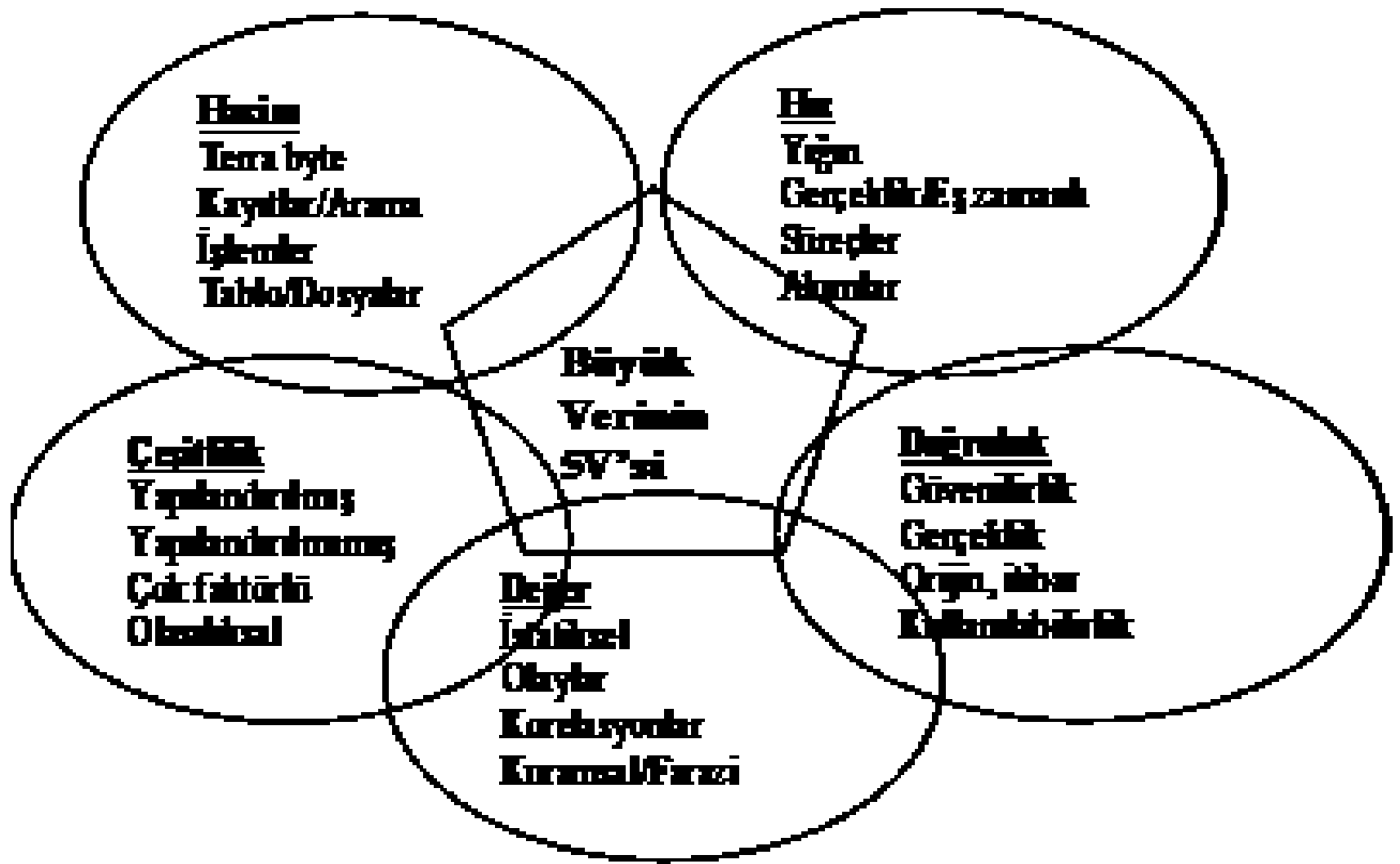
- Keşif: Veritabanında kayıtlı bulunan "veriler işlenir" ve bir model çerçevesinde kullanılacak biçimde formatlanır ve örüntüler keşfedilir.
- Tahmin modeli: Bu aşamada keşfedilen "modeller işlenir" ve modellere göre işlenen veriler geleceđin tahmini amacıyla kullanılır.
- Araştırma analizi: Örgütsel planların uygulanması ve karar süreçlerinde kullanılacak modellerin araştırılmasıdır. Hangi durumda hangi modelin kullanılacağına karar verilir.



Büyük Veri (BV)

- Büyük Veri (BV), geleneksel sistemler (İVT) tarafından etkin ve verimli şekilde saklanması ve analiz edilmesi güç, ham, yapısal olmayan veya yarı yapısal, çok büyük veri kümeleridir.
- Büyük Veri neden değil, ne hakkındadır. Bir olayın nedeni yerine ne hakkında olduğunun araştırılması daha çok fayda sağlayabilir (Schönberger & Cukier, 2013, s. 22).





Şekil-2.14: Büyük Veri Özellikleri (Zorlu, 2014) aktaran (Çelik S., 2018, s. 50)

Büyük Veri (BV) Özellikleri

- **Hacim (Volume):** BV hacim olarak çok büyüktür ve hızla üretilir.
- **Üretilme Hızı (Velocity):** BV birim zamanda çok hızlı bir şekilde üretilen veridir.
- **Çeşitlilik (Variety):** BV'de veri çeşitliliği çok fazladır. Metin, ses, video ve cihaz kayıtları gibi çok farklı yapısal olmayan veri türleri vardır.
- **Doğruluk (Veracity):** Büyük verinin içindeki verilerin doğruluk yüzdesi daha düşüktür.
- **Değer (Value):** Büyük verinin toplanıp işlenmesinden sonra işe yarar sonuçlar elde etmek için bu analizlerin bir "değer" üretmesi gerekmektedir.



NoSQL Veritabanları

- NoSQL, ilişkisel veritabanları haricindeki veritabanlarının genel adıdır. Tarihi 1980'lere uzanan bu terim nesne odaklı veya farklı saklama yapıları kullanan birçok veritabanı için kullanılmaktadır. Ancak günümüzde Büyük Veri ortamlarında verileri yönetmek için kullanılan veritabanlarını tanımlamaktadır (Nizam, 2016, s. 408).
- NoSQL daha çok sadece okuma amaçlı, normalizasyon ihtiyacı duyulmayan, başka verilerle birleştirme (JOIN) ihtiyacı olmayan ve sorgu ihtiyaçları için geliştirilen veritabanlarıdır. Bu sayede hızlı bir şekilde analiz ve sorgulama yapılabilmektedir.
- MySQL ve SQL Server'in yeni sürümlerinin NoSQL özellikleri kazanmaya başladığı görülmektedir.
- Günümüzde en çok kullanılan NoSQL veritabanları Hadoop, HBase, Apache Cassandra ve MangoDB'dir ki bunlar aynı zamanda BV veritabanları olarak da bilinmektedir.



Uygulama

db_PMTP Veritabanında
Sorgu (Query) Oluşturma
ve Verileri Sorgulama



IV. Hafta

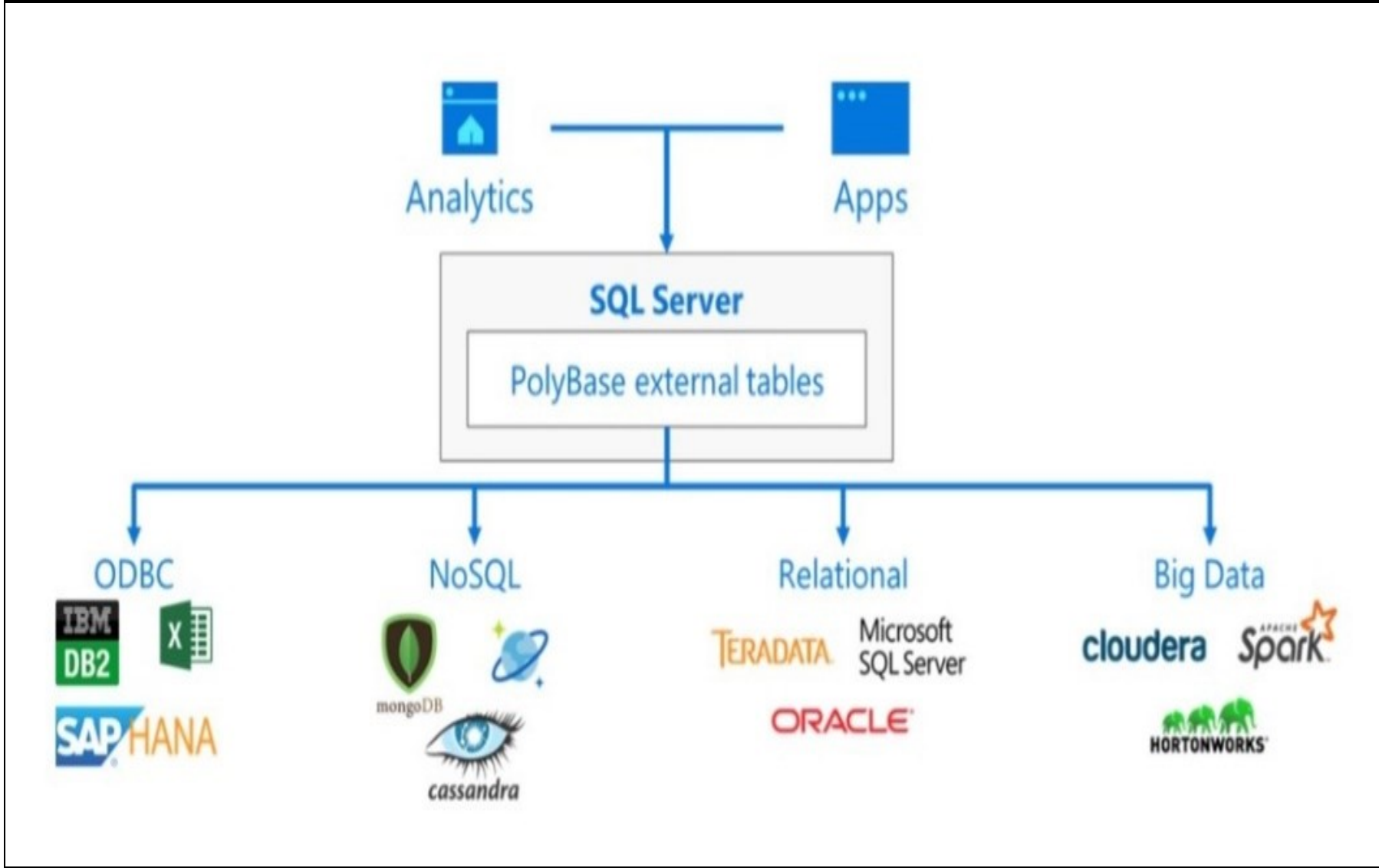
MS SQL Server VTYS Tanıtımı



SQL Server VTYS

- SQL Server, Microsoft tarafından geliştirilen ve pazarlanan İlişkisel Veritabanı Yönetim Sistemidir (İVTYS). İVTYS İngilizce olarak Relational Database Management System olarak adlandırılır ve kısaca RDBMS denir.
- Microsoft SQL Server'ın birçok farklı sürümleri vardır: SQL Server 2005, 2008, 2012, 2014, 2017 ve 2019. En son 2019'da yayınlanan SQL Server 2019 sürümü bu derste kullanılacaktır.
- SQL Server, 2017 sürümü ile birlikte Linux İşletim Sisteminde de çalışmaya başlamıştır. MS SQL Server 2019 sürümüyle birlikte PolyBase teknolojisi yardımıyla kendisini Veritabanlarıyla ilgili tüm alanlarda bir işlem merkezi olarak konumlandırmaya çalıştığı görülmektedir.
- SQL Serverla ilgili güncel bilgilere <https://www.microsoft.com/tr-tr/sql-server/> linkinden ulaşılabilir.





Şekil-6.1: MS SQL Server 2019 Çalışma Modeli (MSSQL, 2019)



Tablo-6.1: SQL Server Sistem Gereksinimleri

| | |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| İşletim Sistemi | Windows Server 2008 SP2 Windows Server 2008 R2 SP1 Windows Server 2012-2016 Windows 7 SP1 Windows 8 RTM Windows 10 Linux |
| Bellek (RAM) | 1 GB Kurulumu için yeterlidir. 4 GB kullanım için önerilir. |
| İşlemci | x86 Processor: 1.0 GHz - Pentium III (2019 desteklemiyor) x64 Processor: 1.4 GHz -AMD Athlon, Pentium IV, Core2Duo |
| Disk Alanı | Kurulum için: 6 GB. Kullanım için: 100 GB |
| .NET Framework | SQL Server 2019 için en az .NET Framework 4.6 |
| Management Studio | SQL Server Management Studio 18.0 |



SQLQuery2.sql - SONY-BILGISAYAR\SQLEXPRESS.master (Sony-Bilgisayar\Sony (52))* - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Object Explorer

Connect

SONY-BILGISAYAR\SQLEXPRESS (SQL Server 15.0.2070 - Sony-Bilgisayar\Sony)

Databases

- System Databases
- Database Snapshots
- db_Abstract
- db_ASPDers
- db_Education
- db_Insaat
- db_Mustafa
- db_Web
- dbTest
- ReportServer\$SQLEXPRESS
- ReportServer\$SQLEXPRESSTempDB

Security

- Server Objects
- Replication
- PolyBase
- Management
- XEvent Profiler

```
USE master
GO
CREATE DATABASE db_VTKitap
ON PRIMARY -- Veri dosyası
(NAME = N'db_VTKitap', FILENAME = N'D:\SQL2017DB\db_VTKitap.mdf', SIZE = 5120
LOG ON -- Log Dosyası
(NAME = N'db_VTKitap_log', FILENAME = N'D:\SQL2017DB\db_VTKitap_log.ldf', SIZE = 1024KB, FILEGROWTH = 10%)
GO

-- SQL Server default sürümünü belirleme (SQL Server 2017)
ALTER DATABASE db_VTKitap SET
GO
```

SQL Query Menü

SQL Parse (ayrıştırma) Butonu

Execution (Çalışma) Planı

VT Listesi

SQL Çalıştırma Butonu

Messages

Commands completed successfully.

Completion time: 2020-09-

Query executed successfully.

SONY-BILGISAYAR\SQLEXPRESS ... Sony-Bilgisayar\Sony (52) | master | 00:00:00 | 0 rows

Ready Ln 13 Col 1 Ch 1 INS

Şekil-6.7: T-SQL Komutları ile Veritabanı Oluşturma



SQL Server'da Kısa Yollar

- Bir sütuna NULL girmek için Ctrl+0 basılır.
- Sorgu sonuçları tab-delimited veya comma-separated formatlarında saklanabilir.
- Shift+F1 T-SQL'de context Sensitive yardım bilgileri getirir.
- Sorgu sonuçlarını LCD ekranlarda kolay okumak için Verdana veya Georgia 11-12 font kullanmak faydalıdır.
- Nesnelerin listesini yenilemek (Refresh) için F5 tuşuna basılır.
- Cntr+L, Query Analyzer ekranında sorgu planını görmeyi sağlar.
- F8 Object Explorer arayüzünü açar.
- F5 SQL Query'deki kodları çalıştırır.



SQL Server'da Nesneler Oluşturma

- SQL Server'da Tablo (Table) Oluşturma
- SQL Server'da Görüntü (View) Oluşturma
- SQL Server'da Dizin (Index) Oluşturma
- SQL Server'da Sınırlamalar (Constraints) Oluşturma



Uygulama

MS-SQLServer ile db_PMTP
Veritabanı (VT) ve
Tablolarını Oluşturma



V. Hafta

Veritabanı Tasarım
Konuları



Veritabanı Tasarımında Dikkat Edilecek Hususlar

- VT tasarımı ve modellemede süreç son kullanıcıda başlayıp yine son kullanıcıda bitmelidir.
- Son kullanıcının ihtiyaç olarak görmediği 'yenilikler' kullanıcılara anlatılarak onların desteği alınmalıdır yoksa boşuna bir zaman kaybı olarak kalabilir.
- VT geliştirme sürecinde gerçek dünyadaki varlık ve olaylarla ilgili birçok verinin takibi, saklanması ve yönetilmesi hedeflenir. Bu verilerin veritabanında (VT) saklanması için gerekli yapıların belirlenmesine veri modelleme süreci denir.
- VT tasarımı ve geliştirilmesi hem sanatsal hem de mühendislik yeteneklerinin bir arada kullanılmasını gerektirmektedir. Kullanıcı gereksinimlerini belirlemek ve organize etmek sanatsal yetenek, belirlenen tasarımların fiziksel olarak yaratılması ise mühendislik yeteneği gerektirmektedir (Kroenke-Auer, 2012).
- VT modelleri oluşturulurken genel olarak elemanların kullandığı formlara, raporlara ve veri dosyalarına bakılır.

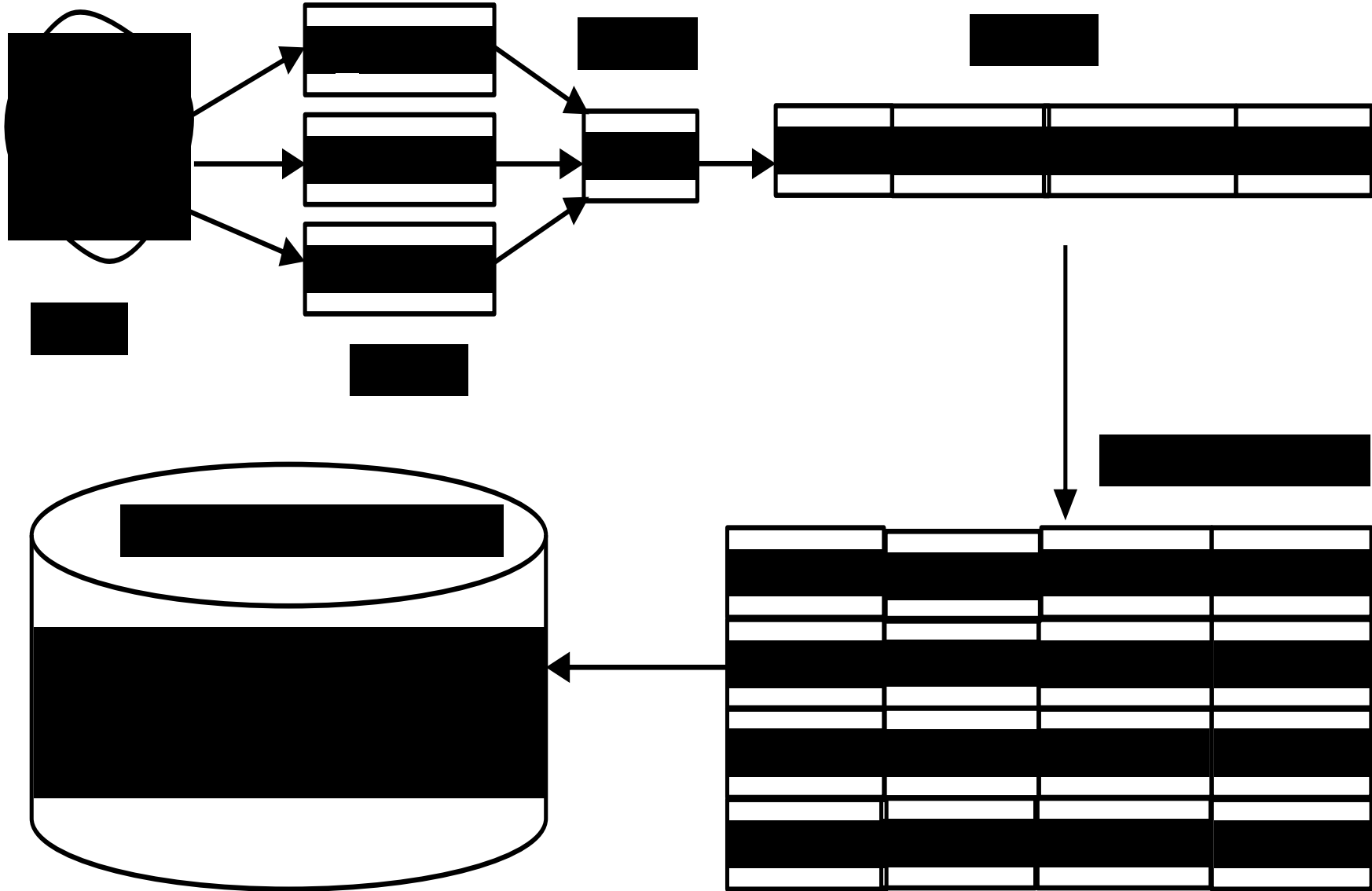


Veritabanı Tasarım Türleri




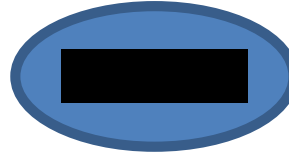


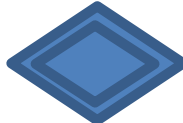
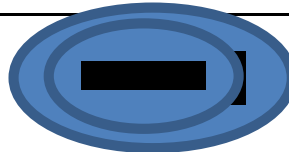
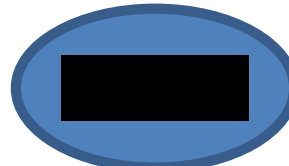




1. Hâlihazırda kullanılan elektronik tablolaama, text dosyaları veya VT raporlarının birleştirilerek VT oluşturulması. Burada VT'ye kâğıt üzerine kaydedilmiş olan verilerde sayısallaştırılarak katkı sağlayabilir.
2. Yeni tasarlanan Bilişim Sistemi projeleri nedeniyle tasarlanan VT'ler.
3. Eski VT'lerin yeni ihtiyaçlar doğrultusunda yeniden tasarımı.



Şekil-1.9: Veritabanı Oluşum Hiyerarşisi (Kul, 2013:155)

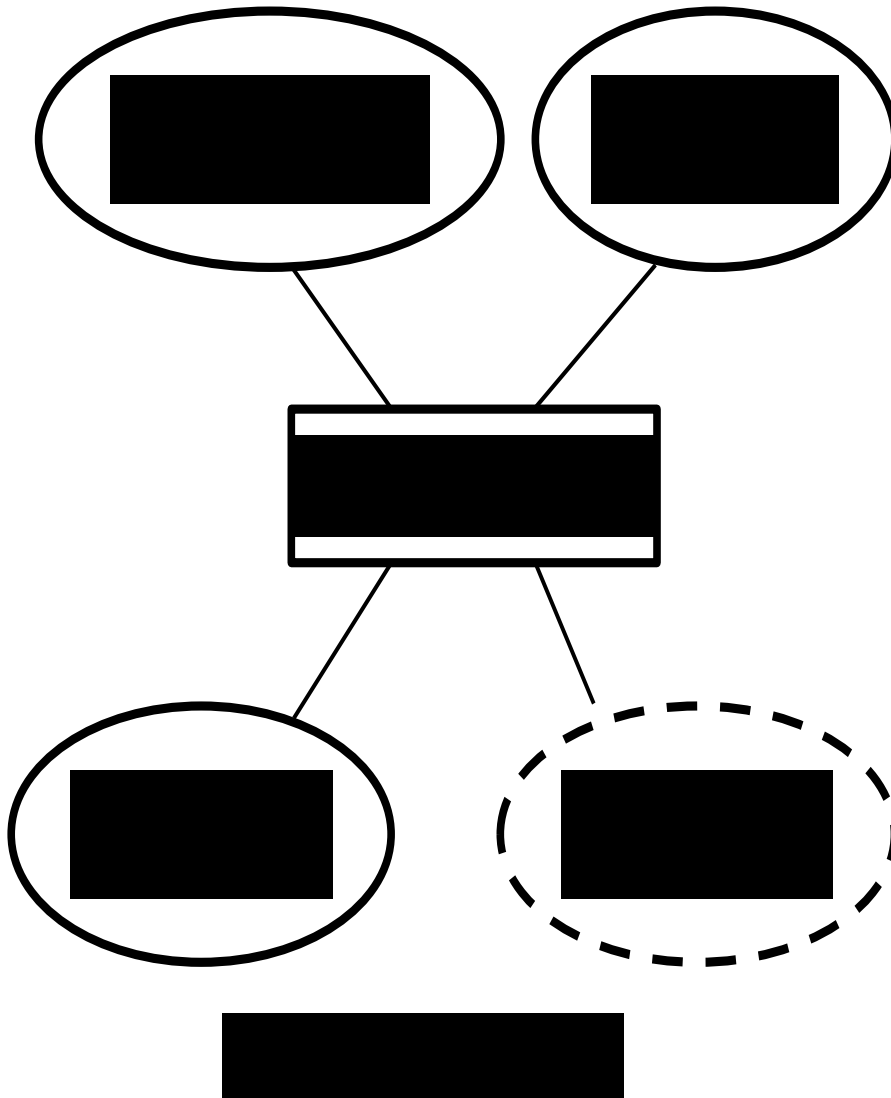


**Tablo-4.2: Varlık İlişki Diyagram Chen Model Gösterimi
(Çağıltay-Tokdemir, 2010:104)**

| | | | |
|-----------------------|-------------------------------------------------------------------------------------|-------------------------------------|---------------------------------------------------------------------------------------|
| Varlık |  | Nitelik |  |
| Zayıf Varlık |  | Birleşik Nitelik |  |
| Bağıntı |  | Türetilmiş Nitelik |  |
| Var olma Bağıntısı |  | Birden fazla değer alabilen Nitelik |  |
| | | Anahtar Nitelik |  |
| Çoklu Kısmen Katılım |  | Tekli Zorunlu Katılım |  |
| Çoklu Zorunlu Katılım |  | Tekli Kısmen Katılım |  |



Şekil-4.3: Varlık İlişki Diyagram (ViD) Modelleri



Tablo-4.5: Bir Varlık Sınıfı ve İki Örneđi

Tbl_Personeller Varlıđı:

Pers_ID

Pers_Adi

Pers_Soyadi

Pers_DTarihi

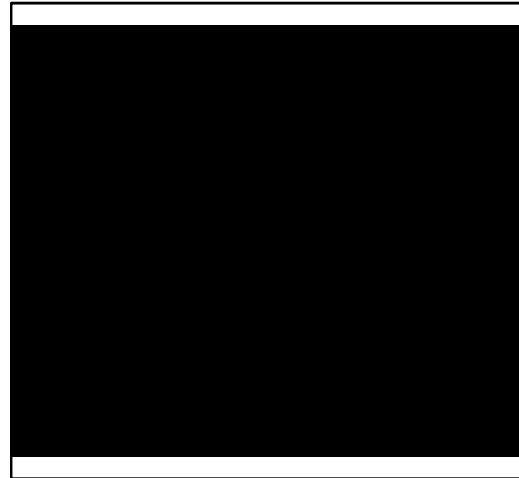
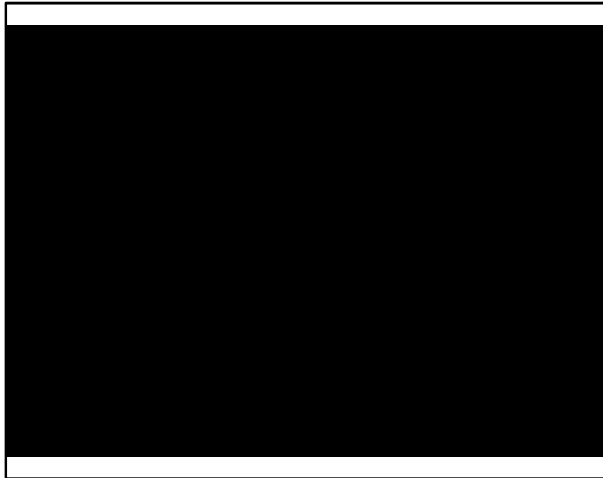
Pers_Ise_Giris_Tarihi

Pers_Cep

Pers_Adresi

Pers_Kenti

Tbl_Personeller Örneđleri:



Tablo-4.6: İlişkisel Veritabanlarında İlişki Türleri

| Tür | Açıklama | Örnek |
|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| Bire Bir (1:1) İlişki | Bir varlık kaydına karşılık ilişkili varlıkta yalnızca tek bir kayıt oluşabilir. Birebir ilişki karşılıklıdır. Karşılıklı iki varlıktan da birer kayıt olabilir. | Her çalışanın bir telefonu vardır. Her telefon yalnızca bir çalışana ait olabilir. |
| Bire Çok (1:N) İlişki | Bir varlık kaydına karşılık, ilişkili diğer varlıktaki kayıt sayısı N olabilir. N veri tabanı sınırları dâhilinde en büyük sayıdır. Müşteri ve adres arasındaki ilişki üzerinde 1:N bağlantısı vardır. | Bir müşteri birçok adrese sahiptir. Her adreste yalnızca bir müşteri tanımlanabilir. |
| Çoğa Çok (M:N) İlişki | İki varlık birbirinin karşılıklı olarak birden fazla kaydıyla eşleşebilir. Müşteri ve müşteri temsilcisi ilişkisinde olduğu gibi. | Bir müşteriyle tüm müşteri temsilcileri ilgilenebilir. Bir müşteri temsilcisi birden fazla müşteriyle çalışabilir. |



Normalizasyon ve Denormalizasyon

- Veritabanındaki anormal durumlar (Insert, Delete, Update), tasarlanan tablolar tek tek incelenerek ortaya çıkarılır ve düzeltilirler. Bu düzeltme işlemine normalizasyon denir.
- Normalizasyon işlemleri tablolar üzerinde uygulanır. Başka bir deyişle, normalizasyon işlemleri VT'nin bütününe değil VT'de bulunan tabloların incelenmesini sağlar. Normalizasyonun amacı VT tasarımlarındaki anormal durumları minimize etmektir.
- Tekrar eden veriyi kontrollü ve sistematik şekilde çağıran sorgular oluşturmak veya tablolara sütunlar eklemek performans artışı sağlayabilir. Bu işleme ise VT Denormalizasyonu denir.
- Denormalizasyonda temel kural; önce normalize et, sonra eğer performans problemi varsa denormalize et olmalıdır.



VT Tasarım Süreci

1. Varlık İlişki Diyagramıyla (VİD) ihtiyaçlar doğrultusunda oluşturulan Kavramsal Model (Analiz Aşaması),
2. İlişkisel VT modeliyle oluşturulan Mantıksal Model (Tasarım Aşaması),
3. SQL Veri Yapısı Tanımlama Dili (VYTD) kodlarıyla oluşturulan Fiziksel Model (Tasarım + Uygulama Aşaması).



Uygulama

**MS-SQLServer da db_PMTTP
Veritabanı Varlık İlişki
Diyagramının Oluşturulması**

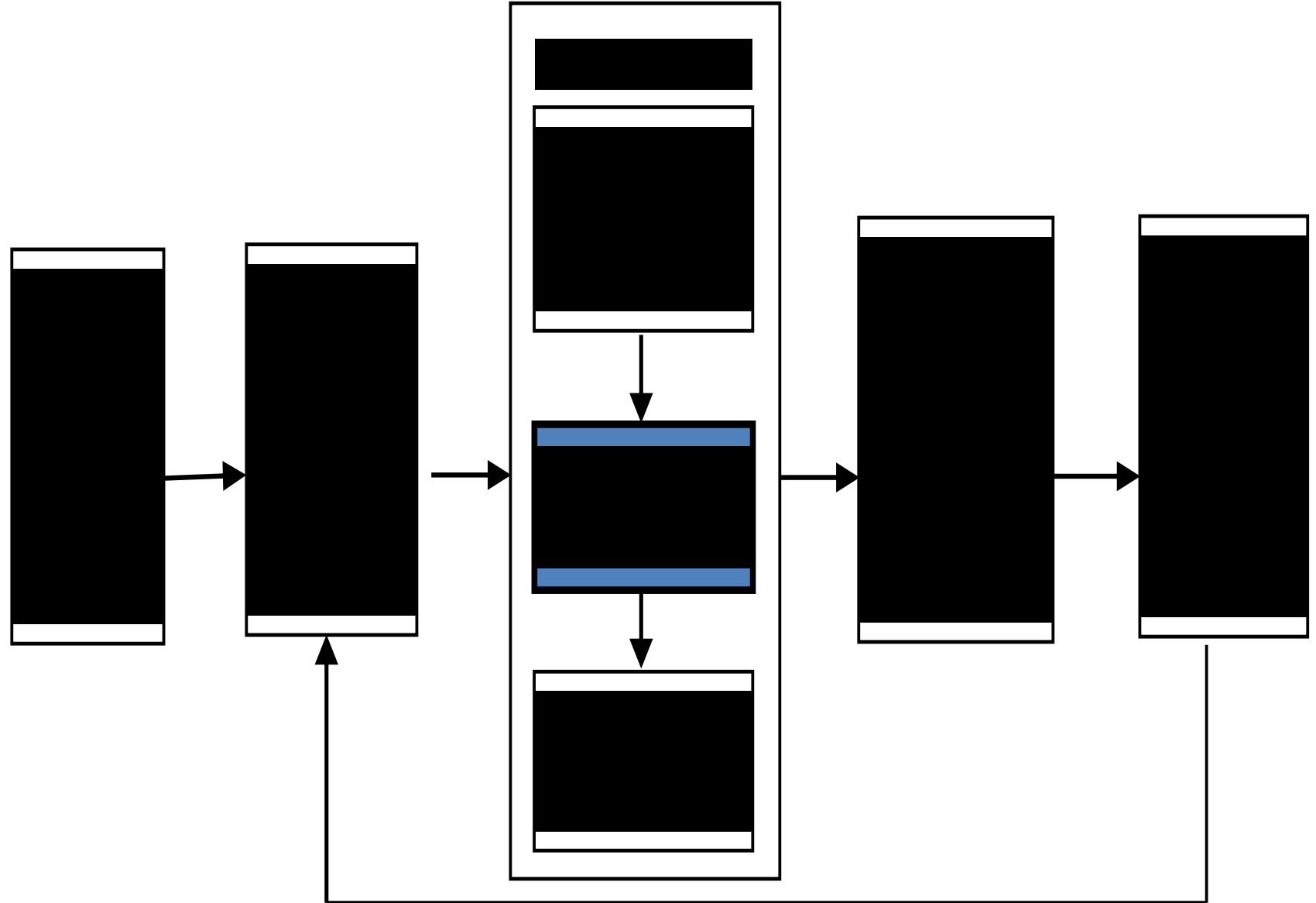


VI. Hafta

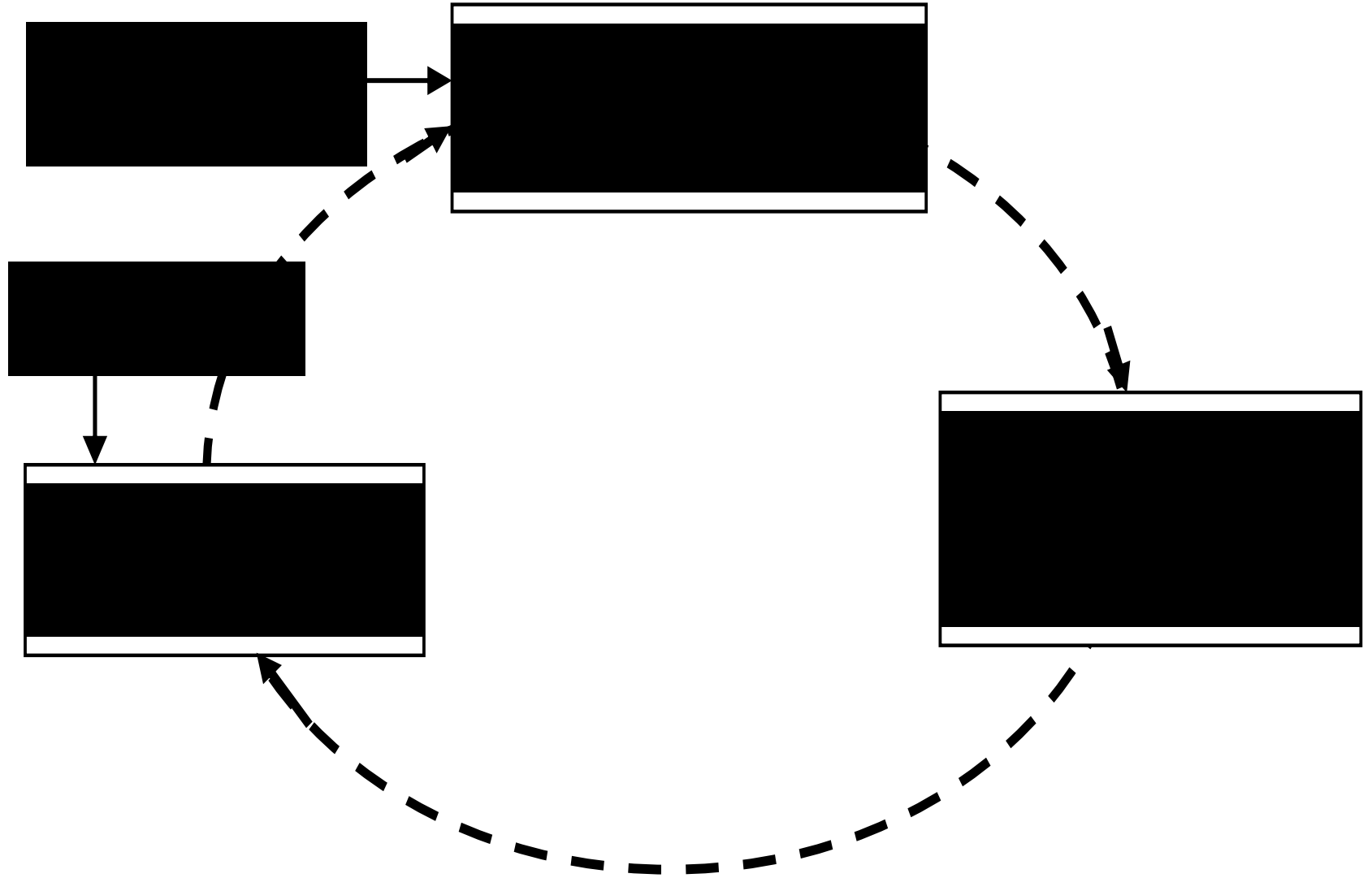
Veritabanı Tasarım Süreci



Şekil-4.1: Uygulama Yazılımı Geliştirme Yaşam Döngüsü (YGYD)



Şekil-4.2: VT Tasarım ve Geliştirme Döngüsü



Veritabanı Tasarım Sürecinin Uygulanması

- VT Tasarımı Kavramsal, Mantıksal ve Fiziksel tasarım aşamalarından oluşur.
- Önce Kavramsal model hazırlanır. Kavramsal Varlık-İlişki modelinden İlişkisel VT modeline (Mantıksal) geçiş yapılır.
- Mantıksal tasarım aşamasında Tablolar Veri tekrarını önleyecek şekilde küçük tablolara bölünerek normalize edilir. Performans artışı için verinin tekrar denormalize edilmesi de ilave bir aşama olarak düşünülebilir.
- Uygulama ise VT'nin fiziksel olarak belirli bir VTYS yardımıyla yaratıldığı, test edildiği ve veri girişlerinin yapıldığı son aşamadır.



Tablo-4.1: Veritabanı Tasarım Adımları (Coruh, 2000)

| Analiz | |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Gereksinim Analizi | Veritabanı amacını ve kullanıcı gereksinimlerini belirleme. |
| Tasarım | |
| Kavramsal Model | <ul style="list-style-type: none">○ Kavramsal model için Varlık-İlişki (Entity-Relation-ER), Semantic nesne veya UML diyagramlarından birini seçme. İlişkisel VTYS kullanılacağı için Varlık-İlişki diyagramı (VİD) seçilir.○ Veritabanında gereksinim duyulan varlıklar (tablolar) belirlenir.○ Tablolarda gereksinim duyulan nitelikler (alanlar) belirlenir.○ Her bir tablonun Anahtar alanları belirlenir.○ Kural, varsayılan değer ve sınırlamaları belirlenir.○ Referans Alanlar tespit edilerek tablolar arası ilişkiler ve türleri belirlenir.○ Varlık-İlişki Diyagramı (VİD) hazırlanır. |
| Mantıksal Model | İlişkisel (Relational), Network, Hierarchical, Object-Oriented veritabanı modellerinden birisi seçilir. Çok özel durumlar haricinde İVT modelleme seçilir ve buna uygun Mantıksal Tasarım hazırlanır. Bu aşamada tablolar normalize edilir ve aralarındaki 1:1, 1:N veya N:M ilişkileri son şekline getirilir. |
| Fiziksel Model | <ul style="list-style-type: none">○ Uygun bir İVTYS (Access, SQL Server, Oracle, Informix, DB2) seçilir.○ Seçilen İVTYS yardımıyla Veritabanı ve nesnelere fiziksel olarak oluşturulur.○ VT'ye başlangıç Verileri girilir ve güncellemeler yapılır. |
| Uygulama | |
| Test | Tasarlanan veritabanı test edilir ve Ortaya çıkan sorun ve hatalar uygulama yazılımı gelişmelerine paralel olarak düzeltilir. Kullanıcıların eğitimine başlanır. |
| Uygulama | Veritabanı sistemi sunucu bilgisayarlarına veya İnternet/İntranete sitesine yüklenerek kullanılmaya başlanır. |
| Güncelleme | Yeni ihtiyaçlar veya çıkan problemler doğrultusunda VT düzeltilir veya yeni eklemeler yapılır. |



Gereksinim Analizi

- Gereksinim analizi VT tasarımından çok Bilişim Sistemi (BS) uygulama yazılımı geliştirme sürecinin bir aşamasıdır.
- Geliştirilecek olan sistemin işlevselliği, sistemde tutulması gereken veriler, bunların birbirleri ile olan ilişkileri ve kullanıcıların operasyonel olarak gerçekleştirdikleri işlerin tüm detayları bu aşamada belirlenir.
- Bu gereksinimler aslında geliştirilecek olan BS'nin işlevselliğini belirlerken, sistemi destekleyecek olan VT'nin yapısını da belirler. VT açısından dikkat edilecek hususlar:
 - Veri tipleri ve grupları,
 - Veriler ile ilgili kurallar,
 - Veriler arasındaki ilişkiler,
 - Veriler üzerinde yapılması gereken işlemler.

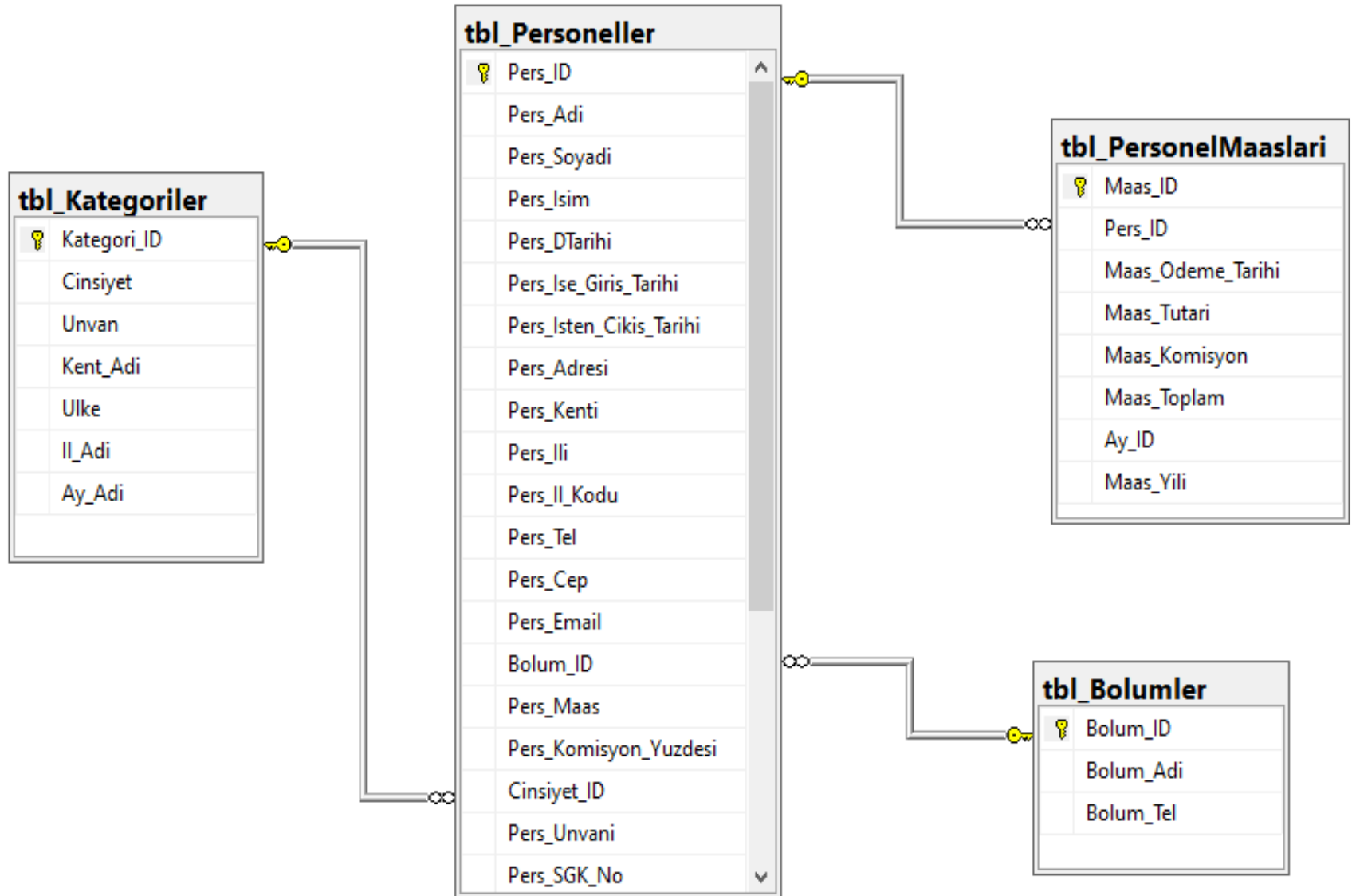


Kavramsal Tasarımda Dikkat Edilecek Hususlar (Çağltay-Tokdemir, 2010:103)

- Gereksinimler dikkatle incelenerek sistemde bulunması gereken varlıklar belirlenmeli,
- Varlıkların nitelikleri tanımlanmalı,
- Varlıkların Anahtar Alanları tanımlanmalı,
- Varlıklar arasındaki ilişkiler tanımlanmalı,
- Her ilişkiye ait eleman sayısı ve katılım kısıtları tanımlanmalı,
- Her bağıntıya tek ve anlamlı bir isim verilmeli,
- Tasarımlarda gereksiz ilişki kurmaktan kaçınılmalı,
- Verilere erişim için kullanılan yollar dikkatle hazırlanmalı ve aynı verilerin tekrarlı olarak sistemde tutulmasından kaçınılmalı,
- Tasarımlarda birden fazla alternatif geliştirilerek en uygun olan seçilmeli.



Kavramsal Tasarım (ViD)



Mantıksal Tasarıma Dönüşüm

- İlişkisel modele dönüşümünde ilk adım, varlıkların dönüşümüdür. Bu dönüşümde, varlığın isminde bir tablo oluşturulur (tbl_Personeller gibi) ve varlığın nitelikleri bu tabloya yerleştirilir. Burada tablonun Anahtar Alanının (AA) altı çizili olarak işaretlenir.
- Tablolar arasındaki ilişki türleri belirlenir. Örneğin Bire-bir (1:1), Bire-çok (1:N) veya çoğa çok (N:M).
- Bire-bir (1-1) ilişkilerin dönüşümü gerçekleştirilirken, bir tablonun Anahtar Alanı (AA) diğerine Referans Alan (RA) olarak taşınır.
- Bire-çok (1-N) ilişkilerin dönüşümü, ilişkiye tekli bağlantısı olan varlığın AA'sı, ilişkiye çoklu bağlantısı olan varlığa RA olarak taşınmasıyla gerçekleştirilir. Yani üst tablonun Anahtar Alanı, Referans Alan (RA) olarak alt tabloya taşınır. Bölüm_ID Personel tablosuna RA olarak yazılır.
- Çoğa-çok (M-N) tipindeki ilişkilerin ilişkisel VT modeline dönüşümünü gerçekleştirirken, bu türdeki her ilişki için yeni bir ara tablo oluşturulur. Bu tablonun AA'sı, ilişkiye bağlı olan varlıkların anahtar alanlarının birleştirilmesiyle elde edilir. Bu nitelikler yeni tablo için aynı zamanda birer RA anahtarı olarak tanımlanır. İlişkinin ayrıca kendisine ait nitelikleri varsa bu nitelikler de bu yeni oluşturulan tabloya taşınır.
- Birleşik alanlar belirlenir. Örneğin personel ismi ad ve soyad olarak iki alana ayrılır veya hesaplanmış alan olarak oluşturulur.



Tablo-4.14: İlişkisel VT (Mantıksal) Model Gösterimi

tbl_Personeller (**Pers ID**, Pers_Adi, Pers_Soyadi, Pers_Isim, Pers_Adresi, Pers_Kenti, Pers_ILi, Pers_IL_Kodu, Pers_Tel, Pers_Cep, *Bolum_ID*, Pers_Maas, Pers_Komisyon_Yuzdesi, Pers_Unvani, Pers_SGK_No, Pers_Aktif_Mi)

tbl_Bolumler (**Bolum ID**, Bolum_Adi, Bolum_Tel)

tbl_PersonelMaaslari (**Maas ID**, *Pers_ID*, Maas_Odeme_Tarihi, Maas_Tutari, Maas_Komisyon, Maas_Toplam, *Ay_ID*, Maas_Yili)

tbl_Aylar (**Ay ID**, Ay_Adi, Ay_Kisa_Adi)



Fiziksel Tasarım

- Fiziksel veri türü seçimi: Önceki aşamalarda genel olarak belirlenen veri türü VT'de en iyi performansı verecek şekilde ayarlanır. Örneğin bir VT'de tamsayılar için Int, Smallint, Money ve Float gibi birçok tür olabilir. Bunlardan uygulamaya en uygun olanı seçilir.
- Saklanacak verinin ilk büyüklüğü ve artma eğilimi: Bu tahmini veri büyüklükleri dikkate alınarak dosya organizasyonu ve dosya büyüklüğü seçilir. Örneğin ilk andaki kayıt sayısı ilk başta ayrılacak dosya alanını belirler. Artış eğilimi de fiziksel dosyanın nasıl büyüyeceğini belirlemede etkilidir.
- Kullanım sıklığı analizi: Sık kullanılan veriler kısmen veya tamamen veritabanı çalıştığında ana hafızada tutulabilir.
- Özel performans ihtiyaçlarının belirlenmesi: Kullanıcının sık sık sorguladığı alanlar belirlenerek bunlar üzerinde ne tür dizinler oluşturulacağına karar verilmelidir.
- Tabloda bölümlenme ihtiyacının belirlenmesi: Tablodaki verilerin yapısı dikkate alınarak tablo arka planda fiziksel dosyalara nasıl bölünüp ve nasıl saklanacağı belirlenmelidir.
- Saklama alanında gerekli iyileştirmeler: Veriler üzerinde sıkıştırma uygulanıp uygulanmayacağına seçilmesi gibi işlemleri içerir.
- Güvenlik ayarları: Tablo veya sütun tabanlı özel güvenlik gerektiren durumlar belirlenmelidir. Bu bilgi kullanılarak kullanıcıların hakları ve şifrelenecek sütunlar belirlenebilir.
- Veri kurtarma ihtiyacı: Verinin kritikliğine göre tablolara özel geri alma ayarları yapılabilir.



```
/**/*****/ tbl_Personeller Tablosunu yaratma ***/
CREATE TABLE [dbo].[tbl_Personeller](
    [Pers_ID] [int] IDENTITY(1,1) NOT NULL,
    [Pers_Adi] [nvarchar](50) NOT NULL,
    [Pers_Soyadi] [nvarchar](50) NOT NULL,
    [Pers_Isim] AS (([Pers_Adi]+' ')+[Pers_Soyadi]),
    [Pers_DTarihi] [Date] NULL,
    [Pers_Ise_Giris_Tarihi] [Date] NULL,
    [Pers_Isten_Cikis_Tarihi] [Date] NULL,
    [Pers_Adresi] [nvarchar](100) NULL,
    [Pers_Kenti] [nvarchar](20) NULL,
    [Pers_ILi] [nvarchar](20) NULL,
    [Pers_IL_Kodu] [char](2) NULL,
    [Pers_Tel] [char](10) NULL,
    [Pers_Cep] [char](10) NULL,
    [Bolum_ID] [int] NOT NULL,
    [Pers_Maas] [money]NULL,
    [Pers_Komisyon_Yuzdesi] [Float] NULL,
    [Pers_Cinsiyeti] [nchar](5) NULL
    [Pers_Unvani] [nvarchar](50) NULL,
    [Pers_SGK_No] [char](10) NULL,
    [Pers_Aktif_Mi] [bit] NOT NULL,
    CONSTRAINT [PK_tbl_Personeller_Pers_ID] PRIMARY KEY CLUSTERED
    ([Pers_ID] ASC) )
GO
```

```
/**/*****/ tbl_Personeller ile tbl_Bolumler Tablosu ilişkisini yaratma ***/
ALTER TABLE [dbo].[tbl_Personeller] WITH CHECK ADD CONSTRAINT
[FK_tbl_Personeller_tbl_Bolumler_Bolum_ID] FOREIGN KEY([Bolum_ID])
REFERENCES [dbo].[tbl_Bolumler] ([Bolum_ID])
GO
ALTER TABLE [dbo].[tbl_Personeller] CHECK CONSTRAINT
[FK_tbl_Personeller_tbl_Bolumler_Bolum_ID]
GO
```



VT Tasarım Sürecinin İVT Uygulanması

- Dünya'da en fazla kullanılan veritabanları olan ilişkisel (Relational) Veritabanlarının Fiziksel oluşumu aşağıdaki adımlar izlenerek yapılmasında fayda vardır:
- Veritabanı ve Dosyalarını oluşturma (CREATE Database),
- Tabloları oluşturma (CREATE Table),
- Nitelikleri (Alanları) oluşturma (Veri grubunun özellikleri),
- Her bir tablo için Anahtar Alanı (Primary Key-PK) belirleme (Bir satır veriye tek (unique) olarak tanımlayan sütun veya sütunlar),
- Referans Alanlar oluşturarak Tablolar arası ilişkiyi belirleme,
- Bilgi alanlarını (domain) belirleme (Bir alana yazılabilecek veri grubu),
- Kısıtlamaları veya sınırlamaları belirleme (Bir alana yazılabilecek verinin sınırlanması),
- İş mantığı ve kuralları belirleme,
- Form ve raporların oluşturulması,
- Güvenlik ve yönetim kurallarını belirleme,
- Tasarlanan sistemin test edilmesi,
- Veri girişi ve Uygulama.



Veritabanı Tasarım Kalite Unsurları (Piattini & Diaz, 2000) aktaran (Nizam, 2016, s. 158)

- **Tablo sayısı**
- **Sütun sayısı**
- **İlişki sayısı**
- **Hiyerarşilerin derinliği**



Uygulama

**MS-QLServer da db_PMTP
Veritabanı Tablolarının SQL
Scriptini Alma**



VII. Hafta

Veritabanında Normalizasyon



Normalizasyon ve Denormalizasyon

- İyi tasarlanmış bir veritabanı, içinde tekrarlı veri içermeyen, referans alanlarının doğru bir şekilde tanımlandığı, kullanıcı gereksinimlerine uygun ve sorguları hızlı bir şekilde karşılayan tasarımlardır.
- Veritabanındaki anormal durumlar (insert, delete, update), tasarlanan tablolar tek tek incelenerek ortaya çıkarılır ve düzeltilirler. Bu düzeltme işlemine normalizasyon denir.
- Normalizasyon işlemleri tablolar üzerinde uygulanır. Başka bir deyişle, normalizasyon işlemleri VT'nin bütününe değil VT'de bulunan tabloların incelenmesini sağlar. Normalizasyonun amacı VT tasarımlarındaki anormal durumları minimize etmektir.
- Tekrar eden veriyi kontrollü ve sistematik şekilde çağırın sorgular oluşturmak veya tablolara sütunlar eklemek performans artışı sağlayabilir. Bu işleme ise VT Denormalizasyonu denir.
- Denormalizasyonda temel kural; önce normalize et, sonra eğer performans problemi varsa denormalize et olmalıdır.
- Veri Ambarı (VA) ve OLAP sistemleri raporlama amaçlı çalışan sistemlerdir. Birçok açıdan Denormalizasyona dayalı çalışırlar. Raporlamayı kolaylaştırmak için küp isimli farklı yapılar da kullanılmaktadır. Alınabilecek raporlardaki birçok hesaplama önceden yapılarak küp içinde denormalize şekilde saklanır.
- ViD, ilişkisel VT'de mantıksal model olarak dönüştürüldüğünde tüm tablolar, sütunlar, sütunların NULL değer alıp almadığı, Anahtar ve Referans Alanlar ve tablolar arasındaki 1:1, 1:N veya N:N ilişki tipleri belirlenir.



Normal Formlar

- Birinci Normal Form (1NF): Bir ilişkisel tablonun 1NF'de olması için bir kayıttaki her sütün sadece tek bir mana ihtiva eden (atomik) veri içermeli ve kayıt tekrarı olmamalıdır.
- İkinci Normal Form (2NF): Bir tablonun 2NF'de olması için 1NF şartlarına ek olarak aday Anahtar Alan (AA) harici tüm alanlar, tüm aday AA'la tam fonksiyonel bağımlı olmalıdır. Başka bir deyişle diğer alanlar, aday AA'ların bazı alt kümelerine fonksiyonel bağımlı olmamalıdır.
- Üçüncü Normal Form (3NF): Bir tablonun 3NF'de olması için, 2NF şartlarına ek olarak: İlişkisel tabloda tüm alanlar Anahtar Alana doğrudan fonksiyonel bağımlı olmalıdır. Anahtar Alan haricindeki alanlar arasında hiç bir fonksiyonel bağımlılık olmamalıdır. Yani Bir tablo için, anahtar olmayan bir alan, anahtar olmayan başka hiçbir alana bağımlı olmamalıdır.
- Domain Key Normal Form (DK/NF): Herhangi bir tablodaki her bir veri değeri o tablonun Anahtar Alanlarının ve tanımlı veri kümelerinin mantıksal sonucuna bağlı olmalıdır. Her bir tablodaki her bir Alan sadece Anahtar Alana bağlı olmalıdır. Her bir tabloda sadece tek bir veri kümesi olmalıdır.



Tablo-4.8: tbl_Kitaplar_0NF Tablosu

tbl_Kitaplar_0NF

| Kitap_Adi | Baski | Kategori | Yazar | Yayin_Kenti | Yayin_Yili | Yayinci | Yayinci_Tel |
|--------------------------------------------|----------|--------------------|---------------|-------------|------------|--------------------|--------------|
| Akıllı Kentler Çağı | 1. Baskı | Bilişim, Kentleşme | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| Bilişim Teknolojileri Ekonomisi ve Toplumu | 3. Baskı | Bilişim, Bilim | Mustafa Çoruh | Ankara | 2015 | Post Yayıncılık | 372-111-2222 |
| BT Destekli Öğretim | 1. Baskı | Bilişim, Öğretim | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| İşletmelerde BS Yönetimi | 1. Baskı | Bilişim, Yönetim | Buğra Diniz | İzmir | 2017 | Yeni Yayıncılık | 232-555-6666 |
| Veritabanları Tasarımı | 2. Baskı | Veritabanı | Burak Oğuz | İzmir | 2003 | Yeni Yayıncılık | 232-555-6666 |
| Veritabanları Teorisi | 1. Baskı | Veritabanı | Ali Nizam | İstanbul | 2013 | Papatya Yayıncılık | 212-333-4444 |

Record: 1 of 6 No Filter Search



Tbl_Kitaplar_0NF Kavramsal (ViD) Modeli

tbl_Kitaplar_0NF

Kitap_Adi

Baski

Kategori

Yazar

Yayin_Kenti

Yayin_Yili

Yayinci

Yayinci_Tel



Tbl_Kitaplar_0NF Mantıksal (İVT) Modeli

- Tbl_Kitaplar_0NF (Kitap_Adi, Baski, Kategori, Yazar, Yayin_Kenti, Yayin_Yili, Yayıncı, Yayin_Tel)



Tbl_Kitaplar_0NF Fiziksel (SQL) Modeli

- CREATE TABLE Tbl_Kitaplar_1NF (
 - Kitap_Adi nvarchar (50),
 - Baski nvarchar (10),
 - Kategori nvarchar (20),
 - Yazar nvarchar (50),
 - Yayin_Kenti nvarchar (20),
 - Yayin_Yili int,
 - Yayıncı nvarchar (50),
 - Yayin_Tel nvarchar (15))



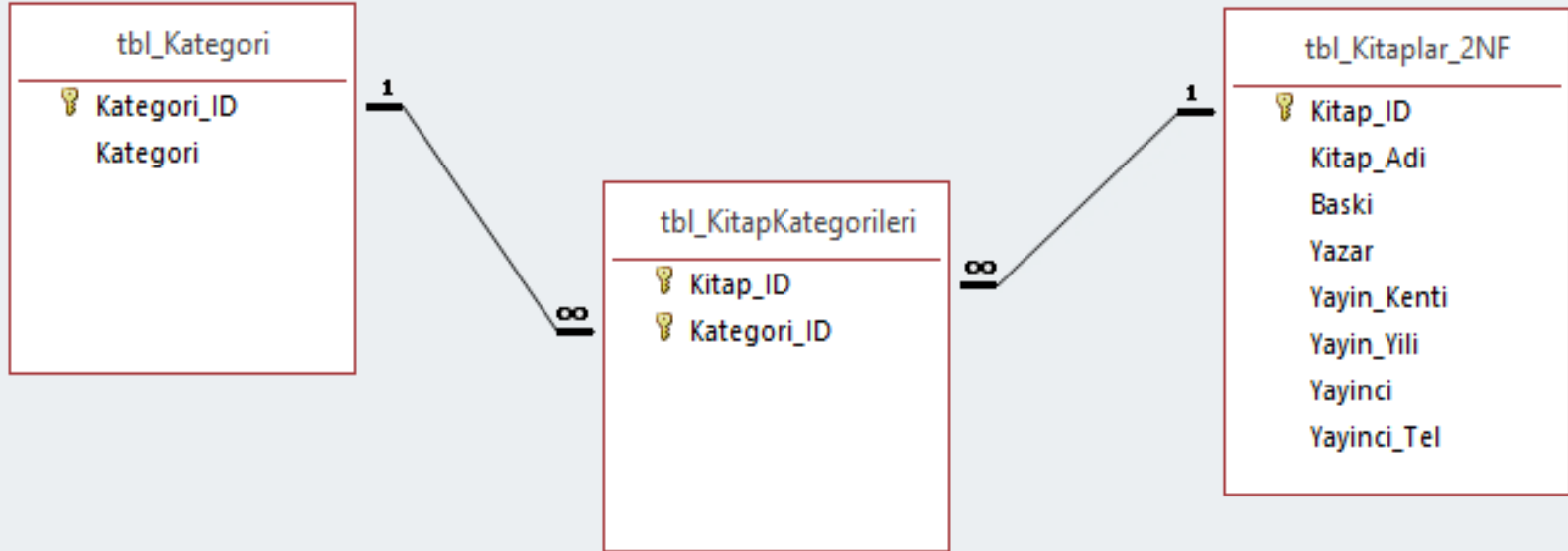
Tablo-4.8.1: tbl_Kitaplar_1NF Tablosu

| Kitap_Adi | Baski | Kategori | Yazar | Yayin_Kenti | Yayin_Yili | Yayinci | Yayinci_Tel |
|--------------------------------------------|----------|------------|---------------|-------------|------------|--------------------|--------------|
| Akıllı Kentler Çağı | 1. Baskı | Bilişim | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| Akıllı Kentler Çağı | 1. Baskı | Kentleşme | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| Bilişim Teknolojileri Ekonomisi ve Toplumu | 3. Baskı | Bilişim | Mustafa Çoruh | Ankara | 2015 | Post Yayıncılık | 372-111-2222 |
| Bilişim Teknolojileri Ekonomisi ve Toplumu | 3. Baskı | Bilim | Mustafa Çoruh | Ankara | 2015 | Post Yayıncılık | 372-111-2222 |
| BT Destekli Öğrenim | 1. Baskı | Bilişim | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| BT Destekli Öğrenim | 1. Baskı | Öğretim | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| İşletmelerde BS Yönetimi | 1. Baskı | Bilişim | Buğra Diniz | İzmir | 2017 | Yeni Yayıncılık | 232-555-6666 |
| İşletmelerde BS Yönetimi | 1. Baskı | Yönetim | Buğra Diniz | İzmir | 2017 | Yeni Yayıncılık | 232-555-6666 |
| Veritabanları Tasarımı | 2. Baskı | Veritabanı | Burak Oğuz | İzmir | 2003 | Yeni Yayıncılık | 232-555-6666 |
| Veritabanları Teorisi | 1. Baskı | Veritabanı | Ali Nizam | İstanbul | 2013 | Papatya Yayıncılık | 212-333-4444 |

Record: 1 of 10 | No Filter | Search



Tbl_Kitaplar_2NF ViD Modeli



Tablo-4.9: Kitaplar 2NF Tablolarının Mantıksal (İVT) Modeli

tbl_Kitaplar_2NF (Kitap ID, Kitap_Adi, Baski, Yazar, Yayin_Kenti, Yayin_Yili, Yayıncı, Yayıncı_Tel)

tbl_Kategoriler (Kategori ID, Kategori)

tbl_KitapKategorileri (Kitap ID, Kategori ID)



Tbl_Kitaplar_2NF Fiziksel Model

- CREATE TABLE [dbo].[tbl_Kitaplar_2NF](
 - [Kitap_ID] [int] NOT NULL,
 - [Kitap_Adi] [nvarchar](50) NOT NULL,
 - [Baski] [nvarchar](50) NULL,
 - [Yazar] [nvarchar](50) NULL,
 - [Yayin_Kenti] [nvarchar](50) NULL,
 - [Yayin_Yili] [int] NULL,
 - [Yayinci] [nvarchar](50) NULL,
 - [Yayinci_Tel] [nvarchar](50) NULL,
 - CONSTRAINT [PK_tbl_Kitaplar_2NF] PRIMARY KEY CLUSTERED ([Kitap_ID] ASC))

- CREATE TABLE [dbo].[tbl_Kategoriler](
 - [Kategori_ID] [int] IDENTITY(1,1) NOT NULL,
 - [Kategori] [nvarchar](50) NULL,
 - CONSTRAINT [PK_tbl_Kategoriler_Kategori_ID] PRIMARY KEY CLUSTERED ([Kategori_ID] ASC))

- CREATE TABLE [dbo].[tbl_KitapKategorileri](
 - [Kategoriler_ID] [int] NOT NULL,
 - [Kitap_ID] [int] NOT NULL,
 - CONSTRAINT [PK_tbl_KitapKategorileri_Kitap_ID_Kategori_ID] PRIMARY KEY CLUSTERED ([Kategori_ID], [Kitap_ID], ASC))



Tablo-4.8.2: Kitaplar 2NF Tabloları

| Kitap_ID | Kitap_Adi | Baski | Yazar | Yayin_Kenti | Yayin_Yili | Yayinci | Yayinci_Tel |
|----------|--------------------------------------------|----------|---------------|-------------|------------|--------------------|--------------|
| 1 | Akıllı Kentler Çağı | 1. Baskı | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| 2 | Bilişim Teknolojileri Ekonomisi ve Toplumu | 3. Baskı | Mustafa Çoruh | Ankara | 2015 | Post Yayıncılık | 372-111-2222 |
| 3 | BT Destekli Öğrenim | 1. Baskı | Mustafa Çoruh | Ankara | 2016 | Post Yayıncılık | 372-111-2222 |
| 4 | İşletmelerde BS Yönetimi | 1. Baskı | Buğra Diniz | İzmir | 2017 | Yeni Yayıncılık | 232-555-6666 |
| 5 | Veritabanları Tasarımı | 2. Baskı | Burak Oğuz | İzmir | 2003 | Yeni Yayıncılık | 232-555-6666 |
| 6 | Veritabanları Teorisi | 1. Baskı | Ali Nizam | İstanbul | 2013 | Papatya Yayıncılık | 212-333-4444 |

Record: 1 of 6

| Kategori_ID | Kategori |
|-------------|------------|
| 1 | Bilişim |
| 2 | Kentleşme |
| 3 | Bilim |
| 4 | Öğretim |
| 5 | Yönetim |
| 6 | Veritabanı |
| * | (New) |

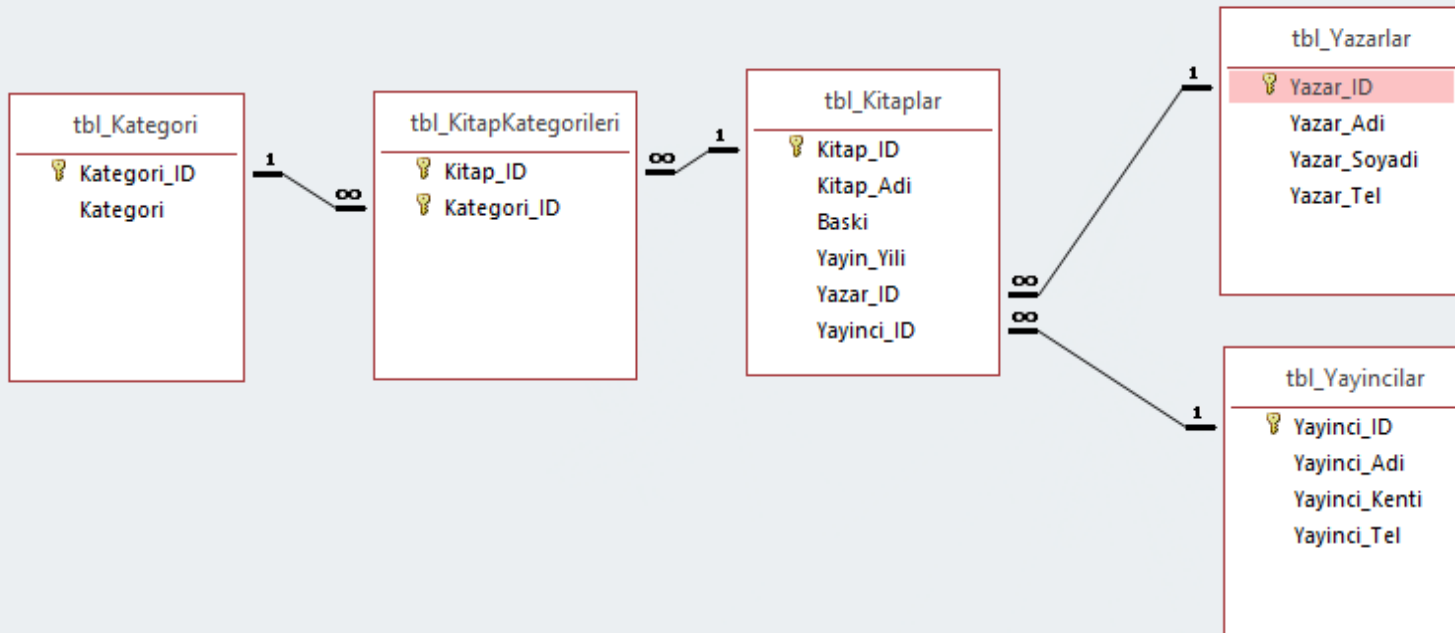
Record: 1 of 6

| Kitap_ID | Kategori_ID |
|----------|-------------|
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |
| 2 | 3 |
| 3 | 1 |
| 3 | 4 |
| 4 | 1 |
| 4 | 5 |
| 5 | 6 |
| 6 | 6 |
| * | |

Record: 3 of 10



Tbl_Kitaplar_3NF ViD Modeli



Tablo-4.10: Books 3NF Tablolarının Mantıksal (İVT) Modeli

tbl_Kitaplar (**Kitap ID**, Kitap_Adi, Baski, Yayin_Yili, *Yazar_ID*, *Yayinci_ID*)
tbl_KitapYayincilari (**Yayinci ID**, Yayinci_Adi, Yayinci_Kenti, Yayinci_Tel)
tbl_KitapYazarlari (**Yazar ID**, Yazar_Adi, Yazar_Soyadi, Yazar_Tel)
tbl_Kategoriler (**Kategori ID**, Kategori)
tbl_KitapKategorileri (**Kitap ID**, **Kategori ID**)

Açıklamalar:

Her kitap bir yazar tarafından yazılmış ve bir yayıncı tarafından basılmıştır.

Her kitap Yayinci_Kentin'de yayınlanmıştır.

Yazar_Adi + Yazar_Soyadi raporlarda birleştirilecektir.

Yazar, Yayıncı ve Kategori verileri, uygulama kullanıcıları tarafından kullanılmadan önce VTY tarafından ilgili tablolara girilmelidir.

Tüm ID alanları, İVTYS tarafından doldurulmuş otomatik veri giriş alanlarıdır.



Tbl_Kitaplar_3NF Fiziksel Model

- CREATE TABLE [dbo].[tbl_Kitaplar](
 - [Kitap_ID] [int] NOT NULL,
 - [Kitap_Adi] [nvarchar](50) NOT NULL,
 - [Baski] [nvarchar](50) NULL,
 - [Yazar_ID] [int] NULL,
 - [Yayin_Yili] [int] NULL,
 - [Yayinci_ID] [int] NULL,
 - CONSTRAINT [PK_tbl_Kitaplar] PRIMARY KEY CLUSTERED ([Kitap_ID] ASC))

- CREATE TABLE [dbo].[tbl_Yazarlar](
 - [Yazar_ID] [int] IDENTITY(1,1) NOT NULL,
 - [Yazar_Adi] [nvarchar](50) NULL,
 - [Yazar_Soyadi] [nvarchar](50) NULL,
 - [Yazar_Tel] [nvarchar](12) NULL,
 - CONSTRAINT [PK_tbl_Yazarlar_Yazar_ID] PRIMARY KEY CLUSTERED ([Yazar_ID] ASC))

- CREATE TABLE [dbo].[tbl_Yayincilar](
 - [Yayinci_ID] [int] IDENTITY(1,1) NOT NULL,
 - [Yayinci_Adi] [nvarchar](50) NULL,
 - [Yayinci_Tel] [nvarchar](50) NULL,
 - [Yayinci_Kenti] [nvarchar](50) NULL,
 - CONSTRAINT [PK_tbl_Yayincilar_Yayinci_ID] PRIMARY KEY CLUSTERED ([Yayinci_ID], ASC))



Tablo-4.8.3: Kitaplar 3NF Tabloları

| Kitap_ID | Kitap_Adi | Baski | Yayin_Yili | Yazar_ID | Yayinci_ID |
|----------|--------------------------|----------|------------|----------|------------|
| 1 | Bilişim Kentleri | 1. Baskı | 2016 | 1 | 1 |
| 2 | Bilişim Teknolojileri | 3. Baskı | 2015 | 1 | 1 |
| 3 | BT Destekli Öğrenim | 1. Baskı | 2016 | 1 | 1 |
| 4 | İşletmelerde BT Yönetimi | 1. Baskı | 2017 | 2 | 2 |
| 5 | Veritabanları Tasarımı | 2. Baskı | 2003 | 3 | 2 |
| 6 | Veritabanları Teorisi | 1. Baskı | 2013 | 4 | 3 |
| * (New) | | | | | |

| Kategori_ID | Kategori |
|-------------|------------|
| 1 | Bilişim |
| 2 | Kentleşme |
| 3 | Bilim |
| 4 | Öğretim |
| 5 | Yönetim |
| 6 | Veritabanı |
| * (New) | |

| Yayinci_ID | Yayinci_Adi | Yayinci_Kenti | Yayinci_Tel |
|------------|--------------------|---------------|--------------|
| 1 | Post Yayıncılık | Ankara | 372-111-2222 |
| 2 | Yeni Yayıncılık | İzmir | 232-555-6666 |
| 3 | Papatya Yayıncılık | İstanbul | 212-333-4444 |
| * (New) | | | |

| Kitap_ID | Kategori_ID |
|----------|-------------|
| 1 | 1 |
| 1 | 2 |
| 2 | 1 |
| 2 | 3 |
| 3 | 1 |
| 3 | 4 |
| 4 | 1 |
| 4 | 5 |
| 5 | 6 |
| 6 | 6 |
| * (New) | |

| Yazar_ID | Yazar_Adi | Yazar_Soyadi | Yazar_Tel |
|----------|-----------|--------------|--------------|
| 1 | Mustafa | Çoruh | 372-111-2222 |
| 2 | Buğra | Diniz | 232-555-6666 |
| 3 | Burak | Oğuz | 232-555-6666 |
| 4 | Ali | Nizam | 212-333-4444 |
| * (New) | | | |

Uygulama

**MS-QLServer veya MS-Access
ile Normalizasyon Tablolarını
Oluřturma**



VIII. Hafta

Vize Sınavı



IX. Hafta

Yapısal Sorgulama Dili - Structured Query Language (SQL)



SQL Özellikleri

- SQL insanların İlişkisel VT sistemleri ile konuşmasını sağlayan özel bir dildir.
- SQL bilgisayarda kullanılan bir dildir; ancak C++, C#, VB.NET gibi bir programlama dili değildir.
- Döngü, karar, kontrol, dallanma ve parametre tanımı gibi programlama komutları Saklı Yordamlar (Stored Procedures), Tetikleyiciler (Triggers) ve Fonksiyonlar gibi veritabanı nesnelere içinde kullanılmaktadır.
- SQL standartlarında çevrim oluşturan komutlar (LOOP vb.), kontrol ve dallanma komutları (IF, ELSE, CASE, GOTO vb.) yoktur.
- İVT'de verinin işlenmesi, sorgulanması, yönetimi, veri yapısının oluşturulması, güvenlik tanımları ve hareket yönetimi SQL komutlarıyla gerçekleştirilir.
- SQL veri üzerindeki işlemlerin fiziksel saklama yapısından bağımsız olarak yürütülmesini sağlar.
- SQL Nesne, XML, NoSQL gibi yapıların eklenmesiyle gelişimine melez bir dil olarak devam etmektedir.



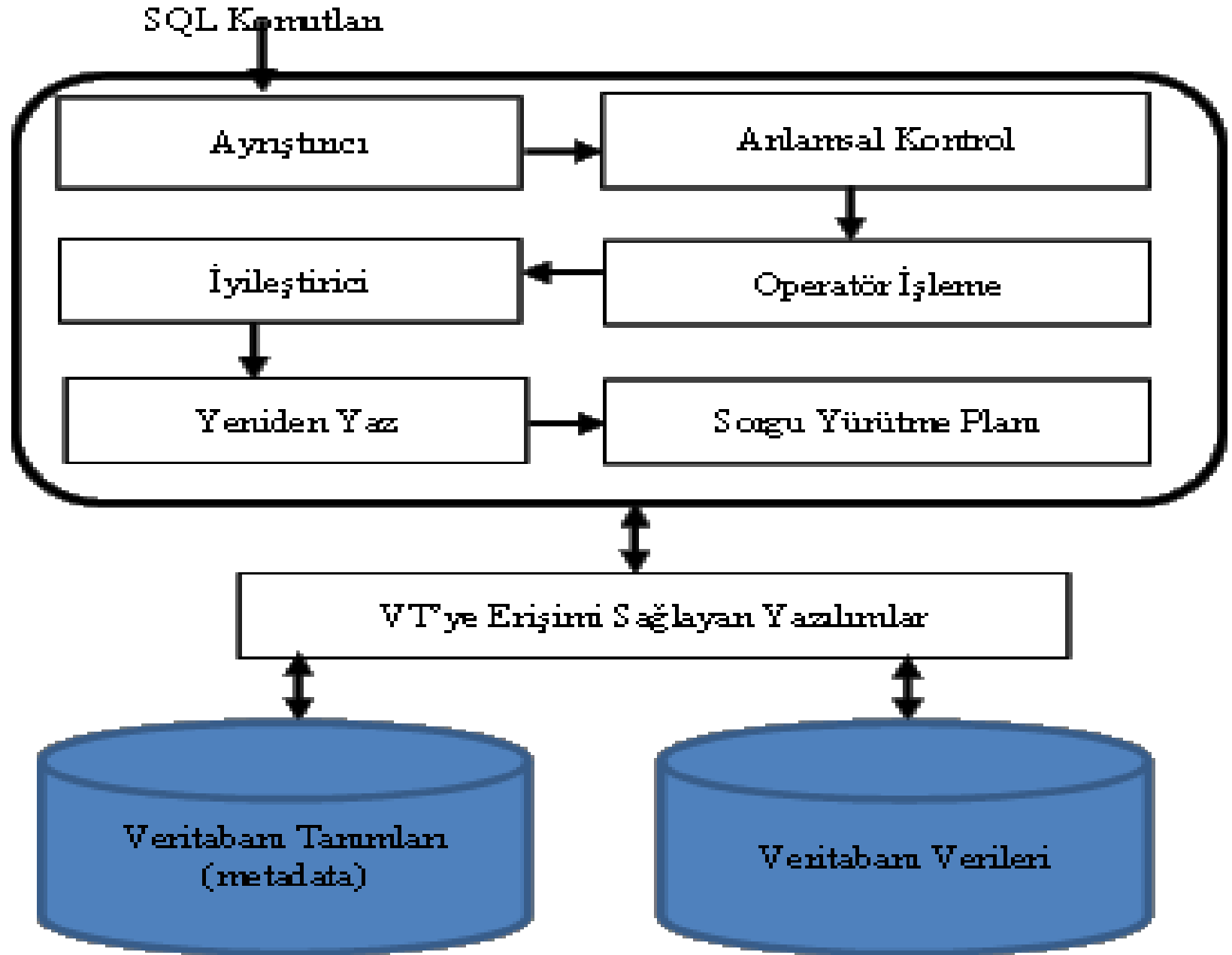
SQL Çalıştırma Kuralları

(Alp-Özdemir-Kilitçi, 2011:50 & Çağıltay-Tokdemir, 2010:242 & Coruh, 2000)

- SQL deyimleri büyük küçük harf ayırımından etkilenmez. Bunun anlamı SQL deyiminin bir kısmı büyük, diğer bir kısmı küçük harflerle yazılabilir.
- Bir SQL deyimi bir ya da daha fazla satırda yer alabilir.
- SQL anahtar kelimeleri kısaltılamaz ya da yarıda kesilerek ayrı ayrı satırlara bölünemez.
- Deyimin bittiği yeri belirlemek için en sonuna(;) işareti konulmalıdır.
- Anahtar kelimeler genellikle ayrı satırlara kaydedilerek daha okunaklı bir görünüm elde edilebilir.
- Bazı kayıtlarda eksik sütun-dikeç bilgileri olabilir veya SQL komutları boş sonuç verebilir. Bu değerlere NULL değerler adı verilir. NULL anahtar kelimesi SQL komutlarında kullanılabilir. NULL belirsiz, boş, bilinmeyen anlamına gelir.



Şekil-5.1: SQL Yaşam Döngüsü (Çağltay-Tokdemir, 2010:250)



Personel Maaş Takip Programı (PMTTP) Tabloları

| Pers_ID | Pers_Adi | Pers_Soyadi | Pers_Isim | Pers_DTarih | Pers_Ise_Gi | Pers_Isten | Pers_Adresi | Pers_Kenti | Pers_Ili | Pers_Il_Kod | Pers_Tel | Pers_... |
|---------|----------|-------------|---------------|-------------|-------------|------------|-----------------|-------------|-----------|-------------|------------|----------|
| 1 | Mustafa | Çoruh | Mustafa Çoruh | 01.01.1962 | 06.01.2016 | | Uzun Sok. No 1 | Kdz. Ereğli | Zonguldak | 67 | 3721112222 | 537111 |
| 2 | Buğra | Diniz | Buğra Diniz | 01.01.1998 | 07.01.2016 | | Kısa Sok. No 2 | Merkez | Bartın | 74 | 3721115555 | 535333 |
| 3 | Burak | Oğuz | Burak Oğuz | 01.01.1993 | 08.01.2016 | | Alt Sok. No 3 | Beylikdüzü | İstanbul | 34 | 3721114444 | 536222 |
| 4 | Ayşe | Meliha | Ayşe Meliha | 01.01.1968 | 10.01.2016 | | Yukarı Sok No 4 | Sincan | Ankara | 06 | 3721113333 | 536166 |

| Bolum_ID | Bolum_Adi | Bolum_Tel | Mudur_ID |
|----------|--------------------|------------|----------|
| 1 | Bilişim Sistemleri | 3721112222 | 1 |
| 2 | Pazarlama | 3722223333 | 2 |
| 3 | Satış | 3723334444 | 3 |
| 4 | Muhasebe | 3724445555 | 4 |
| 5 | Finans | 3725556666 | 4 |
| 6 | Yönetim | 3726667777 | 4 |
| * | Yeni | | |

| Maas_ID | Pers_ID | Maas_Oden | Maas_Tutari | Maas_Komi | Ay_ID | Maas_Topla | Maas_Yili | Ekleme |
|---------|---------|------------|-------------|------------|-------|------------|-----------|--------|
| 1 | 1 | 01.07.2020 | 3.540,00 ₺ | 0,00 ₺ | 7 | 3.540,00 ₺ | 2020 | |
| 2 | 1 | 01.08.2020 | 3.540,00 ₺ | 0,00 ₺ | 8 | 3.540,00 ₺ | 2020 | |
| 3 | 1 | 01.09.2020 | 3.540,00 ₺ | 0,00 ₺ | 9 | 3.540,00 ₺ | 2020 | |
| 4 | 1 | 01.10.2020 | 3.540,00 ₺ | 0,00 ₺ | 10 | 3.540,00 ₺ | 2020 | |
| 5 | 1 | 01.11.2020 | 3.540,00 ₺ | 0,00 ₺ | 11 | 3.540,00 ₺ | 2020 | |
| 6 | 1 | 01.12.2020 | 3.540,00 ₺ | 0,00 ₺ | 12 | 3.540,00 ₺ | 2020 | |
| 7 | 2 | 01.11.2020 | 2.500,00 ₺ | 1.200,00 ₺ | 11 | 3.700,00 ₺ | 2020 | |
| 8 | 2 | 01.12.2020 | 2.500,00 ₺ | 1.000,00 ₺ | 12 | 3.500,00 ₺ | 2020 | |
| 9 | 3 | 01.12.2020 | 3.000,00 ₺ | 1.000,00 ₺ | 12 | 4.000,00 ₺ | 2020 | |
| 10 | 4 | 01.12.2020 | 3.500,00 ₺ | 0,00 ₺ | 12 | 3.500,00 ₺ | 2020 | |
| * | Yeni | | | | | | | |

| Kategori_ID | Cinsiyet | Unvan | Kent_Adi | Il_Kodu | Il_Adi | Ulke | Ay_Adi | Kitap_Turu |
|-------------|----------|-----------------|------------|---------|-----------|-----------|---------|------------|
| 1 | Erkek | VT Yöneticisi | Merkez | 01 | Adana | Türkiye | Ocak | Bilişim |
| 2 | Kadın | Satış Elemanı | Kdz.Ereğli | 67 | Zonguldak | ABD | Şubat | Kentleşme |
| 3 | | Pazarlamacı | Alaplı | 34 | İstanbul | İngiltere | Mart | Bilim |
| 4 | | Ofis Yöneticisi | Beylikdüzü | 81 | Düzce | Kanada | Nisan | Öğretim |
| 5 | | CIO | Kızılay | 05 | Amasya | Meksika | Mayıs | Yönetim |
| 6 | | CEO | Sincan | 06 | Ankara | Almanya | Haziran | Veritabanı |

Tablo-5.1: SQL Alt Dilleri ve Komutları

| | |
|-------------------------------------------------------------|-----------------------------|
| Veri Yapısı Tanımlama Dili (DDL - Data Definition Language) | CREATE, ALTER, DROP |
| Veri İşleme Dili (DML- Data Manipulation Language) | INSERT, UPDATE, DELETE |
| Veri Sorgulama Dili (DQL - Data Query Language) | SELECT |
| Veri Hareket Dili (TCC- Transactional Control Commands) | COMMIT, ROLLBACK, SAVEPOINT |
| Veri Kontrol Dili (DCL - Data Control Language) | GRANT, REVOKE, DENY |
| Veri Yönetim Dili (DAC - Data Administration Commands) | START AUDIT, STOP AUDIT |



Tablo-5.2: SQL Veri Yapısı Tanımlama Dil (VYTD) Komutları

| Komut | Açıklama | Örnek |
|---------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CREATE | İVT'lerde herhangi bir Nesneyi oluşturma için kullanılır. | <i>CREATE TABLE tbl_Personeller (Pers_ID int IDENTITY(1,1) NOT NULL, Pers_Adi Nvarchar(10), Pers_Soyadi Nvarchar (15), Pers_DTarihi date, Bolum_ID Int NOT NULL CONSTRAINT PK_tbl_Personeller_Pers_ID PRIMARY KEY CLUSTERED (Pers_ID ASC))</i> |
| ALTER | Veritabanındaki Nesnelerin üzerinde değişiklik yapılması için kullanılır. | <i>ALTER TABLE tbl_Personeller ADD Pers_ILi nvarchar (15), Pers_IL_Kodu char(2), Pers_Maas money, Pers_Komisyon_Yuzdesi Float, Pers_Tel char (10), Pers_Cep char (10), Pers_Email varchar(50), Pers_Cinsiyet char (5), Pers_Unvani nvarchar (50), Pers_Aktif_Mi bit, Pers_Resim image</i> |
| DROP | Nesnelerin VT'den silinmesi için DROP'la birlikte silinecek nesne adı kullanılır. | <i>DROP TABLE tbl_PersonelMaaslari</i> |



Veri İşleme Komutları

- Sorgulama: SELECT
- Ekleme: INSERT
- Silme: DELETE
- Değiştirme: UPDATE



Uygulama

**db_PMTP Veritabanında VYTD
Kullanımı.**



X. Hafta

**SQL İlişkisel Cebir Fonksiyonları,
SQL Veri Sorgulama Dili (VSD) ve
JOIN (Birleştirme) Komutları**



İlişkisel Cebir Fonksiyonları

- UNION (Birleşim): UNION komutu, iki SELECT sorgusunun sonucunu veya iki tabloyu tek bir sonuç halinde alabilmek için kullanılır.
- EXCEPT (Fark): Küme farkı işlemi yapar
- INTERSECT (Kesişim): küme kesişimi işlemi yapar.
- Kartezyen Çarpım (Product): İki tablonun Kartezyen çarpımını verir (NxM).
- Sütun Seçimi (Projection): Tablodan belirli sütunların seçilmesini sağlar.
- Veri Seçimi (Selection): Tablodan satırların seçilmesini sağlar.
- JOIN (Birleştirme): Tabloların aynı değerleri içeren alanları kullanılarak yapılır.
- PIVOT TABLE: Bir tablonun bir sütunundaki değerler sütun haline getirilerek diğer bir sütunun COUNT, SUM (toplam) alınabilir.



**Tablo-5.11: JOIN (Birleřtirme) Komut Türleri
(Çaęıltay-Tokdemir, 2010:302-316 & Coruh, 2000)**

| | | |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INNER JOIN | İki tablodaki verilerin tek bir tabloda (View) birleřtirmesini saęlar. Bu birleřtirme Referans Alan deęerleri üzerinden yapılıır. | <i>SELECT Pers_ID, Pers_Adi, Pers_Soyadi, Bolum_Adi, Pers_Maas FROM tbl_Personeller INNER JOIN tbl_Bolumler ON tbl_Personeller.Bolum_ID = tbl_Bolumler.Bolum_ID</i> |
| OUTER (LEFT/ RIGHT) JOIN | OUTER JOIN, birleřtirilen tablolarda ortak olmayan kayıtların listelenmesini saęlar. Bu iřlem için veritabanına baęlı olarak farklı yapılar mevcuttur. | <i>SELECT Pers_ID, Pers_Adi, Pers_Soyadi, Bolum_Adi, Pers_Maas FROM tbl_Personeller LEFT JOIN tbl_Bolumler ON tbl_Personeller.Bolum_ID = tbl_Bolumler.Bolum_ID</i> |
| SELF JOIN (Özyineli Birleřtirme) | Bu durum, İVT tasarımındaki özyineli baęıntı (self relationship)'den doęan yapılarda kullanılır. Özyineli baęıntılar kurulurken, yapının daha iyi anlaşılabilmesi için, tabloya ait veri kümelerinden iki farklı veri kümesinin oluşturulduęu düşünülebilir. | <i>SELECT P1.Pers_ID, P1.Pers_Adi, P2.Pers_Soyadi FROM tbl_Personeller AS P1, tbl_Personeller AS P2</i> |
| NON EQUI JOIN | Bazı durumlarda, birleřme kořulunda eřitlik deęil eřitsizlik aranır. Bu tür kořullara eřit olmayan birleřme kořulu (non equi-join) denilir. Bu amaçla <, >, <> gibi farklı kořul ifadeleri kullanılabilir. | <i>SELECT P1.Pers_ID, P1.Pers_Adi, P2.Pers_Soyadi FROM tbl_Personeller AS P1, tbl_Personeller AS P2 WHERE P1.Pers_Maas > 1000 AND P2.Pers_Maas < 5000</i> |



Uygulama

**db_PMTP Veritabanından
İlişkisel Cebir Fonksiyonlarını
Kullanma.**



SQL Veri Sorgulama Dili (VSD)

- SELECT [DISTINCT], (*), [(gösterilecek sütun listesi)]
- FROM (seçim yapılacak tablo listesi, alt sorgular)
- WHERE (bileşke (join) şartları ve sorgu şartları)
- GROUP BY (gruclamada kullanılacak sütun listesi)
- HAVING (gruplar üzerinde çalışacak sorgu koşulları)
- ORDER BY (sıralamada kullanılacak sütunlar, ifadeler)



Tablo-5.4: SELECT'te Kullanılan Komutlar, Karakterler ve Operatörler

| Komut | Açıklama | Örnek |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SELECT ... FROM | SELECT Sorguyu başlatan komuttur. FROM veri okunacak tablo veya kayıt kümelerini belirler. | <i>SELECT Pers_ID, Pers_Adi, Pers_Cep FROM tbl_Personeller</i> <i>SELECT Pers_Adi+' ' +Pers_Soyadi AS Isim FROM tbl_Personeller</i> |
| DISTINCT | Tekrarlı kayıtların gösterilmemesini yani mükerrer kayıtların tek kayıt olarak gösterilmesini sağlar. | <i>SELECT DISTINCT Pers_Adi FROM tbl_Personeller</i> |
| * | Bu karakter sorgudaki tüm sütunların (Hepsi) seçilmesi için kullanılır. | <i>SELECT * FROM tbl_Personeller</i> |
| TOP | Kayıtların kaç tanesinin veya % kaçının döndürüleceğini belirtir. | <i>SELECT TOP 3 * FROM tbl_Personeller ORDER BY Pers_Soyadi DESC</i> |
| WHERE | Tabloda özel değerlere sahip kayıtlara erişmek için WHERE kısmında yazılan koşullar kullanılır. WHERE kısmında Alt sorgu ifadeleriyle de koşul oluşturulabilir. Koşullar birbirine AND, OR gibi mantıksal operatörlerle de bağlanabilir. | <i>SELECT * FROM tbl_Personeller WHERE Pers_Adi = 'Mustafa' AND Pers_ID > 2</i> |



| | | |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>GROUP BY</p> | <p>Sayısal verilerin gruplandırılması için kullanılır. GROUP BY kısmında Grup fonksiyonları (AVG, SUM, COUNT vs.) NULL değerlerini hesaba katmazlar. Sırayı ters yönde değiştirmek söz konusu ise ORDER BY kullanılır.</p> | <pre>SELECT Pers_ID, SUM(Maas_Tutari) AS ToplamMaas, MIN(Maas_Tutari) AS EnDusukMaas, MAX(Maas_Tutari) AS EnYukseMaas, AVG(Maas_Tutari) AS OrtalamaMaas FROM tbl_PersonelMaaslari GROUP BY Pers_ID SELECT Bolum_ID, COUNT(*) AS Sayi FROM tbl_Personeller WHERE Pers_Cinsiyet='Erkek' GROUP BY Bolum_ID</pre> |
| <p>HAVING</p> | <p>HAVING grup fonksiyonlarını sonuçlarına göre kısıtlama ve kümeleme şartları tanımlamak için kullanılır. GROUP BY kısmından sonra yazılır. Grup fonksiyonları WHERE kısmındaki şartlar içerisinde kullanılmadığı için HAVING komutu kullanılır. Bu sayede WHERE şartında listelenecek kayıtlar belirlenirken HAVING şartında da gruplar belirlenir.</p> | <pre>SELECT Pers_Ili, count (*) FROM tbl_Personeller GROUP BY Pers_ILi HAVING COUNT(*)>0</pre> |
| <p>ORDER BY</p> | <p>Sorgulama sonucunu belirli bir sütuna göre sıralamayı sağlar</p> | <pre>SELECT Pers_ID, Pers_Adi, Pers_Soyadi, Pers_Cep FROM tbl_Personeller ORDER BY Pers_Soyadi DESC, Pers_Adi</pre> |



Uygulama

**db_PMTP Veritabanından
SELECT komutunu kullanarak
verilerin çağrılması.**



XI. Hafta

SQL Veri Sorgulama Dilinde (VSD) Fonksiyonlar ve Operantlar



T-SQL Tek Satır Fonksiyonları

- **Karakter Fonksiyonları: Lower, Upper**
- **Matematiksel Fonksiyonlar: Power, Rand, Sin**
- **Tarih Fonksiyonları: Getdate()**
- **Dönüştürme Fonksiyonları: Convert, Cast**
- **Genel Fonksiyonlar: Isnull, Isdate, Rank**



Tablo-5.9: Çok Satır Gruplandırma Fonksiyonları

| Fonksiyon | Yapılan İşlem |
|-----------------------|-------------------------------------------------|
| AVR (ORTALAMA) | Değerlerin ortalamasını hesaplar |
| COUNT(SAYMA) | Satır sayısını (kaç adet satır olduğunu) verir. |
| MIN (EN KÜÇÜK) | En küçük değeri döndürür. |
| MAX (EN BÜYÜK) | En büyük değeri döndürür. |
| SUM (TOPLAM) | Sütundaki Verilerin toplam değerini döndürür. |



Uygulama

**db_PMTP Veritabanında
View'larda Operant ve
Fonksiyonların kullanılması.**



XII. Hafta

SQL Veri İşleme Dili (ViD) Komutları



Tablo-5.3: SQL Veri İşleme Dil (VID) Komutları

| Komut | Açıklama | Örnek |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| INSERT INTO Tablo İsmi (Alan İsimleri) VALUES (Veriler) | Veritabanına yeni verilerin eklenmesini sağlar. INSERT komutu kullanılarak tabloya aynı anda yalnızca tek bir kayıt (satır) veya birden fazla kayıt ekleme yapılabilir. | <i>INSERT INTO tbl_Personeller (Pers_Adi, Pers_Soyadi, Pers_DTarihi, Bolum_ID, Pers_ILi, Pers_IL_Kodu, Pers_Maas, Pers_Komisyon_Yuzdesi, Pers_Cep, Pers_Email, Pers_Cinsiyet, Pers_Unvani, Pers_Aktif_Mi) VALUES ('Mustafa', 'Çoruh', '1.1.1962', 1, 'Zonguldak', '67', 4500, 0, '5371111222', 'mcoruh@mustafacoruh.com', 'Erkek', 'VTY', 1)</i> |
| INSERT INTO Tablo İsmi (Alan İsimleri) SELECT ... | Başka bir tablodan SELECT ile seçilen kayıtların belirtilen tabloya eklenmesini sağlar. Bu durumda VALUES sözcüğü kullanılmaz. (Her bölümde aynı sayıda Alan ismi olmalıdır.) | <i>INSERT INTO tbl_Personeller (Pers_Adi, Pers_Soyadi, Bolum_ID, Pers_Aktif_Mi) SELECT (Pers_Adi, Pers_Soyadi, Pers_DTarihi, Bolum_ID, 1) FROM tbl_PersonellerPaz</i> |
| SELECT (Alan İsimleri) INTO Tablo İsmi FROM Kaynak Tablo İsmi | Kaynak tablodan verilerin hedef tabloya kopyalanmasını sağlar. | <i>SELECT Pers_Adi, Pers_Soyadi, Pers_DTarihi, Bolum_ID INTO tbl_PersonellerTest FROM tbl_PersonellerPaz</i> |



Uygulama

**db_PMTP Veritabanında
Veri işleme Dili
komutlarının kullanılması.**



XIII. Hafta

SQL Veri İşleme Dil (ViD) Komutları



Tablo-5.3: SQL Veri İşleme Dil (VİD) Komutları

| Komut | Açıklama | Örnek |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UPDATE Tablo İsmi SET ... | Veriler üzerinde değişiklik (güncelleme) yapılmasını sağlar. UPDATE özellikle birden fazla veri değiştirmek için kullanılan bir yapıdır. | <i>UPDATE tbl_Personeller SET Pers_Maas = Pers_Maas *1.18</i> <i>UPDATE tbl_Personeller SET Pers_IL_Kodu = 67 WHERE Pers_ID=5</i> <i>UPDATE tbl_Personeller SET Pers_Resim = (SELECT * FROM OpenRowSet (BULK 'D:\Mustafa-Resim.jpg', SINGLE_BLOB) Pers_Resim) WHERE Pers_ID=1</i> |
| DELETE FROM Tablo İsmi | Veritabanından verilerin silinmesini sağlar. | <i>DELETE FROM tbl_Personeller WHERE Pers_Soyadi LIKE '%Ser%'</i> |
| TRUNCATE TABLE Tablo İsmi | Bir tablonun tüm satırlarını yok etmek için kullanılır. | <i>TRUNCATE TABLE tbl_Personeller</i> |
| MERGE INTO | INSERT ve UPDATE komutlarının bir birleşimidir. MERGE komutu veritabanına INSERT işlemi yaparken kayıt varsa UPDATE, kayıt yoksa INSERT etmeye yarayan çok faydalı bir komuttur. | <i>MERGE INTO tbl_PersonelMaaslari PM USING tbl_Personeller P ON P.Pers_ID = PM.Pers_ID WHEN MATCHED THEN UPDATE SET PM.Maas_Tutari = P.Pers_Maas WHEN NOT MATCHED THEN INSERT (Pers_ID, Maas_Odeme_Tarihi, Maas_Tutari) VALUES (P.Pers_ID, GETDATE(), P.Pers_Maas);</i> |



Uygulama

**db_PMTP Veritabanında Veri
işleme Dili komutlarının
kullanılması.**



XIV. Hafta

Sınava Hazırlık Konuların Tekrar Edilmesi

