

# Ataşehir Adıgüzel MYO

## Bilgisayar Programcılığı

Veritabanı ve Yönetim Sistemleri II

<http://webders.mustafacoruh.com/Pdfs/Ders/Veritabanlari-Sunum.pdf>

Öğr. Görevlisi Mustafa ÇORUH

I. Hafta

# Veritabanı Yönetim Sistemlerine (VTYS) Giriş

Mustafa Çoruh

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

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

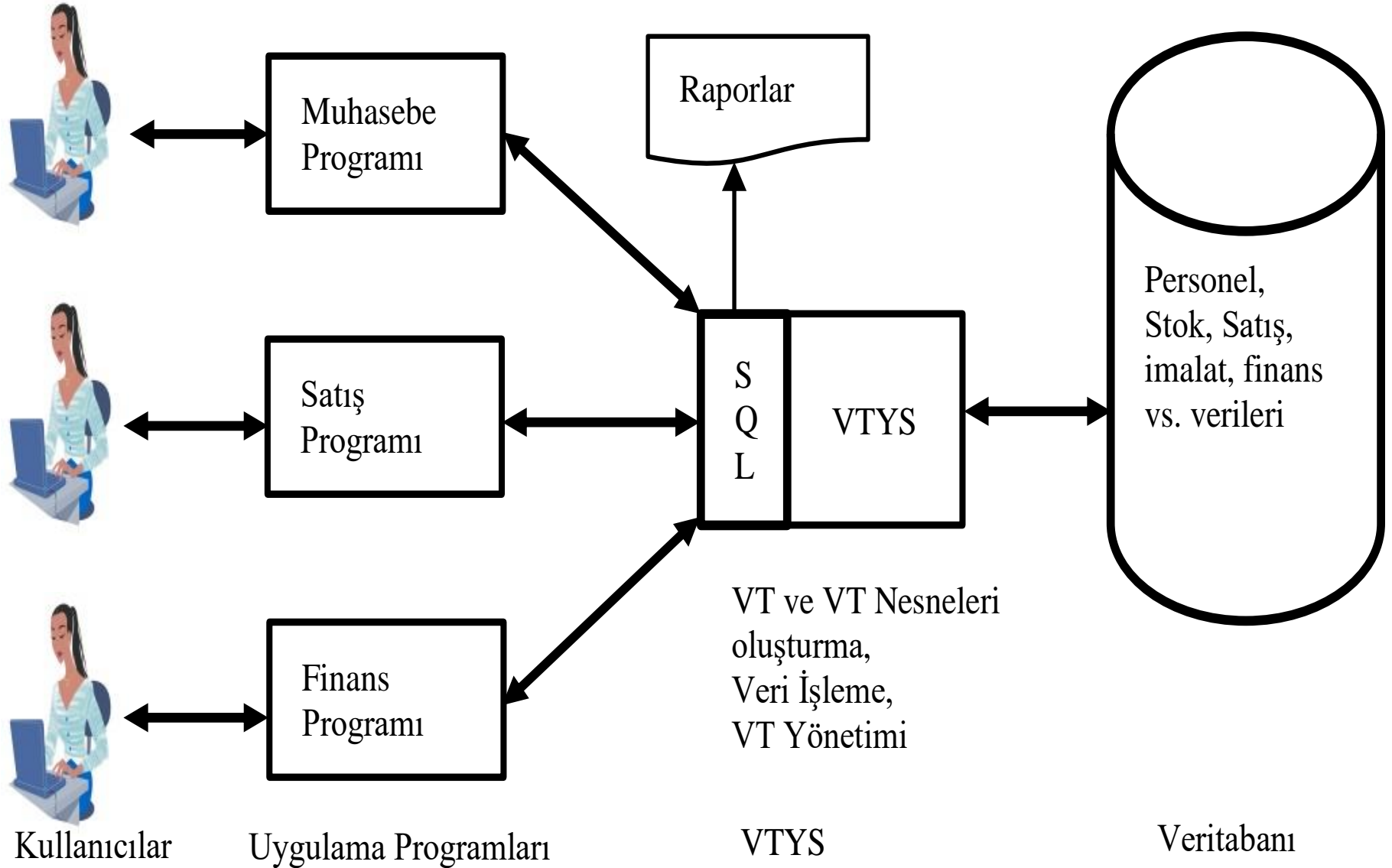
# Veritabanı Yönetim Sistemleri (VTYS)

- İşletmelerde toplanan sayısal verilerin bilgisayarlarda yönetilmesi, işlenmesi ve depolanması için geliştirilen veya kullanılan yazılımlara Veritabanı Yönetim Sistemleri (VTYS) denmektedir.
- VTYS, bilgisayarın fiziksel hafızasındaki verileri çeşitli özelliklerine göre gruplandırıp şekillendirdikten sonra saklayan, işleyen ve sorgulayan programlardır.
- VTYS kullanıcılar ve yazılımcılar ile VT arasında iletişimi sağlayan yazılımlardır. Kullanıcılar sorgulama yapabilir, rapor alabilir veya kayıtları güncelleyebilir.
- VTYS verilerin fiziksel yapılardan soyutlanmasını sağlayarak tasarımcılara yardım eder.
- VTYS'ler tek sorguyu birden fazla CPU'da çalıştıramaz. Ancak birden fazla sorguyu aynı anda farklı CPU'larda çalıştırılabilir.

**Tablo-1.7: Veritabanı Yönetim Sistem Mimari Türleri  
(Alp-Özdemir-Kilitçi, 2011:14)**

<b>Yerel-Lokal (Single Tier)</b>	Eğer bir VTYS tek bilgisayarda kullanılıyorsa yerel veya masaüstü (desktop) bir sistemdir. Paradox, dBase, Access yerel VTYS'e örnek olarak gösterilebilir.
<b>İstemci/Sunucu (Client/Server)</b>	Veritabanı Sunucusu (server) bir bilgisayarda bulunuyor ve ağa bağlı diğer istemci bilgisayarlar tarafından ortak olarak kullanılıyorsa buna Sunucu/İstemci VTYS denir. Interbase, Oracle, Sybase, MS SQL Server ve MySQL bu tip VTYS'dir.
<b>Dağıtık</b>	Birçok medyadan veri alan ve farklı yerlerde sunucuları bulunan VTYS'lerdir. Şekil-1.17'de ulusal bir şirketin dağıtık VTYS görülmektedir.

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



**Tablo-1.10: VTYS Örnekleri  
(Coruh, 2000 & Alp-Özdemir-Kilitçi, 2011:15)**

<b>MS-Access</b>	<p>Microsoft firması tarafından Microsoft Windows İşletim Sisteminde çalışmak üzere yazılan ve dağılımı yapılan bir üründür. MS Access veritabanında bulunan nesnelerin birçoğu sihirbazlar (wizards) yardımıyla kolayca hazırlanabilir. Çoğu zaman tasarım ortamı kullanılmaksızın, sadece sihirbazlar kullanılarak VT nesneleri hazırlanabilir. Çoklu kullanıcı desteği bulunmamaktadır. İşletim sisteminin sağladığı güvenlik olanaklarını kullanmaktadır. Kayıt sayısı yaklaşık olarak en çok 1.000.000 olabilmektedir. Yine yaklaşık 500 MB boyuta kadar da sorunsuz çalışabilmektedir. Bölüm-VII'de tanıtılmıştır.</p>
<b>MS SQL Server</b>	<p>Sybase VTYS yazılım kodlarından sadece Windows İşletim Sistemi üzerinde çalıştırılmak için Microsoft tarafından geliştirilen bir veritabanıdır. Buna karşın MS SQL Server, zamanla kullanımı kolay, güvenilir ve işlem gücü yüksek bir veritabanı haline gelmiştir. Satın alma ve işletme maliyeti diğer çok kullanıcılı ilişkisel VT'lere göre düşüktür. Tablo başına, 4 TB veri depolayabilmektedir. Transaction Locking, Trigger, Functions ve Stored Procedure oluşturma özelliklerine sahiptir. Bölüm-VI'da detaylarıyla anlatılmıştır.</p>
<b>MySQL</b>	<p>Windows, Linux, OS/2, Solaris, AIX vb. işletim sistemleri üzerinde çalışabilen bir VTYS'dir. Trigger ve Stored Procedure özelliklerine sahiptir, ancak Transaction Locking özelliği bulunmamaktadır. Tablo başına 2 GB veri depolayabilir. Python'dan Java'ya kadar birçok programlama dili ile MySQL'e erişilebilir. Apache ve PHP ile beraber genellikle Unix-Linux temelli web uygulamalarında tercih edilmektedir. Açık kaynak kodlu, küçük ve orta ölçekli bir İVT yazılımıdır. Oracle tarafından desteklenmektedir.</p>
<b>IBM DB2</b>	<p>IBM firmasının çok kullanıcılı işletim sistemlerine yönelik geliştirdiği büyük ölçekli bir veritabanı yönetim sistemidir. Performansı, Access ve MySQL'e göre daha yüksek olmasına karşın daha yüksek maliyete sahiptir. Unix-Linux ve Windows işletim sistemleri üzerinde çalışabilir. Transaction Locking, Trigger ve Stored Procedure oluşturma özelliklerine sahiptir.</p>

<b>Informix</b>	Ücretli ve güçlü bir veritabanıdır. Orta ölçekli işletmelerin yükünü kaldırabilecek kapasitededir. 1994' deki Postgres kodundan geliştirilmiştir.
<b>PostgreSQL</b>	Linux, Unix, BSD, Windows, AIX vb. işletim sistemleri üzerinde çalışabilir. Ücretsiz ve akademik olarak hizmet veren bir veritabanıdır. Çok güçlü bir işlem yapısı vardır. Veri güvenliği ön plandadır. Tablo başına 64 TB veri tutabilme özelliği vardır. Transaction Locking, Trigger ve Stored Procedure özelliklerine sahiptir. PostgreSQL, açık kaynak kodlu bir VTYS'dir.
<b>Oracle</b>	Özellikle yüksek ölçekli uygulamalarda tercih edilen bir VTYS'dir. Çok yüksek güvenlik ve yönetim gerektiren durumlarda tercih edilen bir sistemdir. Maliyeti yüksektir. Windows, Unix, Linux işletim sistemleri üzerinde çalışabilir. Oracle, sınırsız sayıda tablo oluşturmayı ve kullanmayı desteklemektedir. Oracle VTYS genellikle sadece Oracle olarak anılır. Tüm İVTYS gibi, büyük miktarda verinin çok-kullanıcılı ortamda depolanmasını ve güvenli bir şekilde veriye erişimini sağlar. ORACLE çok sayıda araçtan oluşur ve uygulama geliştiricilerinin kolay ve esnek uygulamalar geliştirmesini sağlar. Oracle'da diğer VTYS'ler gibi, Stored Procedure'ler, paketler, trigger'lar gibi bileşenler yer alır.
<b>Sybase</b>	Orta ve büyük ölçekli veritabanı yönetim sistemidir. Çoğunlukla bankacılık ve kamusal alanlarda tercih edilen bir VTYS'dir.



**Tablo-1.20: VTYS Performansını Etkileyen Bilgisayar Unsurları  
(Coruh, 2000 & Nizam, 2016:227)**

<b>Merkezi İşlem Birimi (MİB-CPU)</b>	Bilgisayarı bilgisayar yapan ve tüm işlemleri gerçekleştiren Merkezi İşlem Birimidir (MİB/CPU). MİB bir bilgisayarın beyni, ana veri işlemcisidir. MİB'nin temel görevi, verileri okumak, bu verileri yönetmek ve belleğe yazdırmaktır. MİB'in hızı Megahertz cinsinden ölçülür ve manası bir saniyede 1 milyon döngüdür. Bilgisayarın dikkat edilmesi gereken en önemli özelliği MİB'in kaç GHz'de hızda çalıştığı ve kaç çekirdekten oluştuğudur. MİB VT performansını doğrudan etkileyen ilk donanımdır.
<b>Veri yolu (bus)</b>	Bilgisayarın parçaları arasındaki veri akışını sağlayan sisteme veri yolu veya BUS denir. Görevi MİB ile Sabit Disk, RAM veya CD-ROM arasındaki veri akışını sağlamaktır. BUS hızı yavaş ise çok hızlı bir MİB veya yüksek kapasiteli bir RAM'a sahip olmak bir şey ifade etmez.
<b>Sabit Disk (SD)</b>	Verilerin kalıcı olarak saklandığı manyetik disklerden oluşan fiziksel bileşendir. Günümüzdeki VT Terabaytlarca veriyi saklayabilmektedir. Bu veriler ancak SD saklanabilir. Erişim hızı 10 ms mertebesinde. Sabit diskler manyetik disk veya flash disk şeklinde olabilir. Sabit Disk, rastgele erişimli (random access) saklama alanlarıdır. İstenilen herhangi bir noktasına erişilebilir.
<b>Ana Hafıza (RAM)</b>	Bilgisayardaki programların çalıştığı ve veri üzerindeki işlemlerin yapıldığı hafızadır. Hızı 10-100 nano saniye mertebesinde. Ana hafızadaki bilgiler sistem kapanırsa veya çökerse silinir. Ana hafıza erişim hızı yüksektir ve doğrudan adreslenebildiği için çok hızlıdır. RAM ise özel kartlar vasıtasıyla oluşturulur.
<b>İşlemci Tamponu (cache)</b>	Genellikle mikro işlemci içerisine entegre edilen ve çok hızlı çalışan bellektir. Programlar öncelikle ana hafızadan işlemci önbelleğine yüklenir. Birkaç nano saniye mertebesinde çalışır.

## II. Hafta

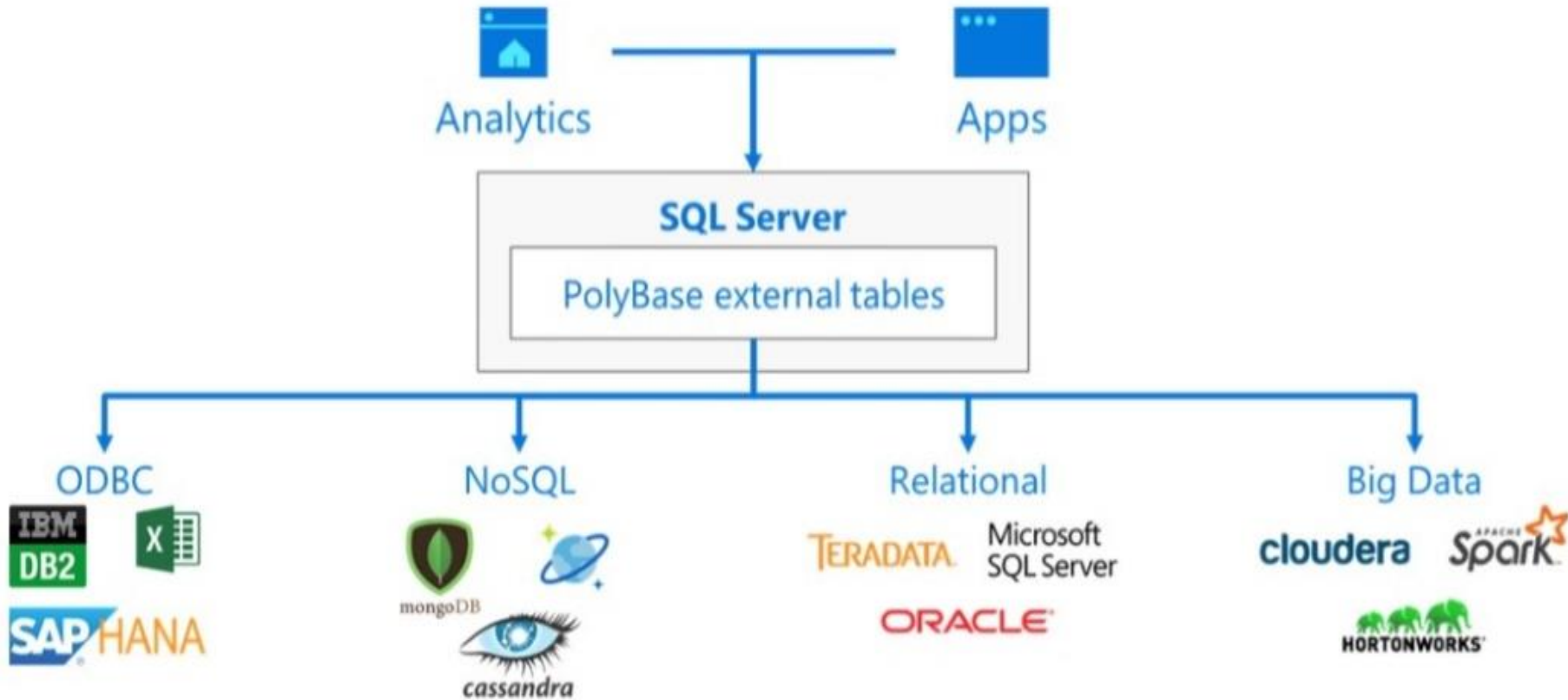
# MS-SQL Server VTYS Tanıtımı

Mustafa Çoruh

# MS SQL Server VTYS Özellikleri

- SQL Server, Microsoft tarafından geliştirilen ve pazarlanan ilişkisel Veritabanı Yönetim Sistemidir (İVTYS).
- Microsoft SQL Server'ın birçok farklı sürümleri vardır: SQL Server 2005, 2008, 2012, 2014, 2017 ve 2019.
- SQL Server 2019'un bu sürümleri Enterprise, Standard, Developer ve Express Edition'dır. Enterprise ve Standard sürümleri ücretli olarak satılırken Developer ve Express sürümleri ücretsiz olarak kullanıcılara sunulmaktadır.
- 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 çalışmaktadır.
- SQL Server sürekli olarak veriler üzerinde işlem yapan WEB siteleri, kurumsal uygulamalar ile masaüstü uygulamaları için zengin özelliklere sahip olan bir İVTYS'dir.
- SQL Server raporlama, yazılım geliştirme, veri depolama, veri yönetimi, Veri Ambarları ve Veri Analitiği için kullanılabilir.
- 2019 sürümünde ise Makine öğrenmesi, No SQL, Büyük Veri gibi daha önemli özelliklerin eklenmiştir.

# Şekil-6.1: MS SQL Server 2019 Sürüm Özellikleri (MSSQL, 2019)



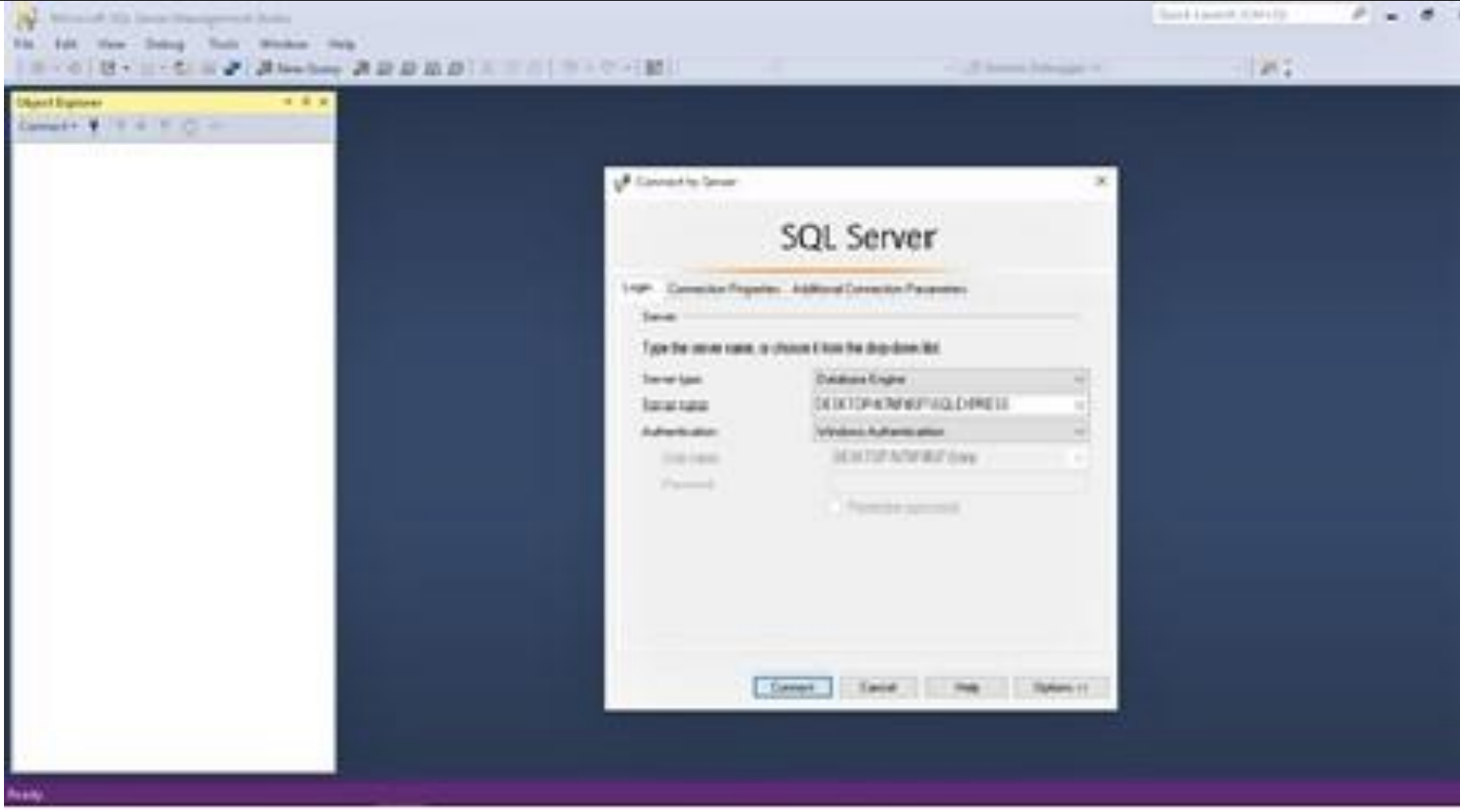
**Tablo-6.1: SQL Server Sistem Gereksinimleri**

<b>İşletim Sistemi</b>	Windows Server 2008 SP2 Windows Server 2008 R2 SP1 Windows Server 2012-2016 Windows 7 SP1 Windows 8 RTM Windows 10 Linux
<b>Bellek (RAM)</b>	1 GB Kurulumu için yeterlidir. 4 GB kullanım için önerilir.
<b>İşlemci</b>	x86 Processor: 1.0 GHz - Pentium III x64 Processor: 1.4 GHz -AMD Athlon, Pentium IV, Core2Duo
<b>Disk Alanı</b>	Kurulum için: 6 GB. Kullanım için: 100 GB

**Tablo-6.2: SQL Server Sistem ve Örnek Veritabanları**

<b>Master</b>	SQL Server'ın ana veritabanıdır. SQL Server'ın doğru bir şekilde çalışmasını sağlayan veritabanıdır. Master veritabanında yapılan yanlışlar sonucunda SQL Server çalışmayabilir.
<b>Model</b>	Bir veritabanı model örneğidir. SQL Server içerisinde oluşturulan her yeni veritabanı bu model VT'nin standart değerlerini kullanır.
<b>Msdb</b>	SQL Server içerisinde yapılan zamanlanmış görevlerin tutulduğu veritabanıdır. SQL Server Agent'ın veritabanıdır.
<b>Tempdb</b>	Çalışılan veritabanlarınının geçici verilerinin tutulduğu veritabanıdır. SQL Server her açıldığında tekrar oluşturulur.
<b>Northwind</b>	Satış (Sales) işlemleriyle ilgili örnek veritabanıdır.
<b>Pubs</b>	Yayıncılık (Publication) işlemleriyle ilgili örnek veritabanıdır.
<b>AdventureWorks</b>	SQL Server'ın yeni özelliklerini kullanan örnek veritabanıdır.

**Şekil-6.2: SQL Server Management Studio Giriş Ekranı**



## Şekil-6.7: T-SQL Komutları ile Veritabanı Oluşturma

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' shows the server hierarchy. The central pane contains the following T-SQL script:

```
-- Veritabanı parametreleri  
USE [master]  
GO  
  
--CREATE DATABASE [db_vttestap]  
ON PRIMARY -- 1024 Megabyte  
NAME = N'db_vttestap', FILENAME = N'D:\SQLSERVER\db_vttestap.mdf' , SIZE = 1024MB, FILEGROWTH = 1024MB)  
LOG ON -- 1024 Megabyte  
NAME = N'db_vttestap_log', FILENAME = N'D:\SQLSERVER\log_vttestap_log.ldf' , SIZE = 1024MB, FILEGROWTH = 1024)  
GO  
  
-- SQL Server çalıştırma (çalıştırma) (SQL Server 2007)  
ALTER DATABASE [db_vttestap] SET COMPATIBILITY_LEVEL = 100  
GO
```

Below the query editor, a message pane displays: 'Database created successfully.' The status bar at the bottom indicates 'Query executed successfully.'

Five arrows point from labels below to specific elements in the interface:

- VT Listesi**: Points to the 'Databases' folder in the Object Explorer.
- SQL Çalıştırma Butonu**: Points to the 'Execute' button (a green play icon) in the query editor toolbar.
- SQL Parse (ayrıştırma) Butonu**: Points to the 'Parse' button (a blue 'P' icon) in the query editor toolbar.
- Execution (çalıştırma) Planı**: Points to the 'Execution Plan' button (a blue 'E' icon) in the query editor toolbar.
- SQL Query Menüsü**: Points to the 'Query' menu in the top application bar.



# III. Hafta

## Yapısal Sorgulama Dili -Structured Query Language (SQL) Giriş

Mustafa Çoruh

# SQL Özellikleri

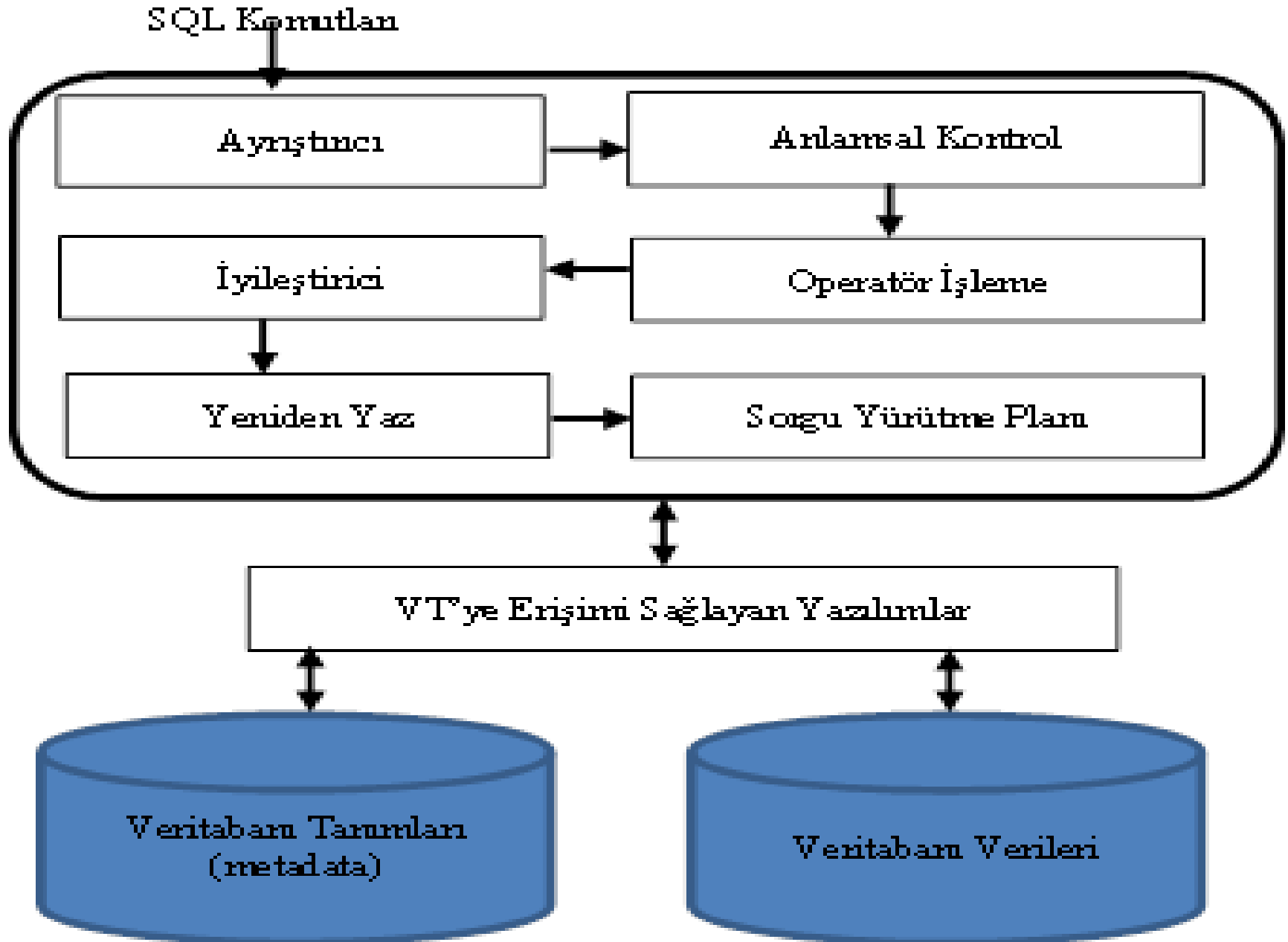
- SQL insanların ilişkisel VT sistemleri ile konuşmasını sağlayan özel bir dildir.
- Bu dil sayesinde, bir İVT'den kayıtlar çağrılabilir, değiştirilebilir, silinebilir ya da yeni kayıtlar eklenebilir.
- SQL bilgisayarda kullanılan bir dildir; ancak C++, C#, VB.NET gibi bir programlama dili değildir.
- 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 Komutları: ekleme (INSERT), silme (DELETE), değiştirme (UPDATE) ve sorgulama (SELECT).

# 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)



**Tablo-5.1: SQL Alt Dilleri ve Komutları**

Veri Yapısı Tanımlama Dili (DDL - Data Definition Language)	CREATE, ALTER, DROP
Veri İşleme Dili (DML- Data Manuplation 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

# Şekil-5.2: Örnek Tablolar

tbl\_Personeller

Pers_ID	Pers_Adi	Pers_Soyadi	Pers_DTarih	Pers_Adresi	Pers_Kenti	Pers_Ili	Pers_Cep	Bolum_IC	Pers_Unvan	Pers_Il_Kod	Pers_Tel	Per
1	Mustafa	Çoruh	1962-01-01	Uzun Sok. No 1	Kdz. Ereğli	Zonguldak	5371111222	1	VT Yöneticisi	67	3724444555	mo
2	Buğra	Diniz	1998-01-01	Kısa Sok. No 2	Merkez	Bartın	5353333444	3	Satış Elemanı	74	3726666777	bdc
3	Burak	Oğuz	1993-01-01	Alt Sok. No 3	Beylikdüzü	İstanbul	5371234567	2	Pazarlamacı	34	3723333888	bo
4	Meliha	Ayşe	1968-01-01	Yukarı Sok No 4	Sincan	Ankara	5361234567	6	Ofis Yöneticisi	06	3728888111	aco

Record: 5 of 5

tbl\_Bolumler

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

Record: 1 of 6

tbl\_Aylar

Ay_ID	Ay_Adi	Ay_Kisa_Ad
1	Ocak	OCK
2	Şubat	ŞUB
3	Mart	MAR
4	Nisan	NİS

Record: 1 of 12

tbl\_PersonelMaaslari

Maas_ID	Pers_ID	Maas_Odeme	Maas_Tutari	Maas_Komisyon	Maas_Toplam	Ay_ID	Maas_Yili
1	1	2016-08-10	3.540,00 ₺	0,00 ₺	3.540,00 ₺	8	2016
2	1	2016-07-10	3.540,00 ₺	0,00 ₺	3.540,00 ₺	7	2016
3	1	2016-09-10	3.540,00 ₺	0,00 ₺	3.540,00 ₺	9	2016
4	1	2016-06-10	3.540,00 ₺	0,00 ₺	3.540,00 ₺	6	2016
5	1	2016-05-10	3.540,00 ₺	0,00 ₺	3.540,00 ₺	5	2016
6	2	2016-10-04	2.750,00 ₺	0,00 ₺	2.750,00 ₺	9	2016
7	3	2016-10-04	2.360,00 ₺	0,00 ₺	2.360,00 ₺	9	2016
8	4	2016-10-04	1.770,00 ₺	0,00 ₺	1.770,00 ₺	9	2016
9	3	2016-10-04	2.360,00 ₺	0,00 ₺	2.360,00 ₺	10	2016
10	3	2016-10-04	2.360,00 ₺	0,00 ₺	2.360,00 ₺	11	2016

Record: 3 of 10

## IV. Hafta

# SQL Veri Yapısı Tanımlama Dili (VYTD)

Mustafa Çoruh

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

<b>Komut</b>	<b>Açıklama</b>	<b>Örnek</b>
<b>CREATE</b>	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>
<b>ALTER</b>	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>
<b>DROP</b>	Nesnelerin VT'den silinmesi için DROP'la birlikte silinecek nesne adı kullanılır.	<i>DROP TABLE tbl_PersonelMaaslari</i>



```
/***/ 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
```

# Personel Maaşları Takip Sistemi Veritabanını Oluşturma

SQL Script Yazımı

V. Hafta

SQL Veri Sorgulama Dili (VSD)

Mustafa Çoruh

# 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 (gruplamada 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**

<b>Komut</b>	<b>Açıklama</b>	<b>Örnek</b>
<b>SELECT</b> ... <b>FROM</b>	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 Isım FROM tbl_Personeller</i>
<b>DISTINCT</b>	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 *</i> <i>FROM tbl_Personeller</i>
<b>TOP</b>	Kayıtların kaç tanesinin veya % kaçının döndürüleceğini belirtir.	<i>SELECT TOP 3 * FROM tbl_Personeller</i> <i>ORDER BY Pers_Soyadi DESC</i>
<b>WHERE</b>	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 *</i> <i>FROM tbl_Personeller</i> <i>WHERE Pers_Adi = ‘Mustafa’ AND Pers_ID &gt;2</i>

# VI. Hafta

## SQL Veri Sorgulama Dili (VSD)

Mustafa oruh

<p><b>GROUP BY</b></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><b>HAVING</b></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(*)&gt;0</pre>
<p><b>ORDER BY</b></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>

# VII. Hafta

## SQL İlişkisel Cebir Fonksiyonları

Mustafa Çoruh



**Tablo-6.13: T-SQL İlişkisel Cebir Fonksiyonları ve Özellikleri**

<b>Fonksiyon</b>	<b>Açıklama</b>	<b>Örnek</b>
<b>UNION (Birleşim)</b>	UNION komutu, iki SELECT sorgusunun sonucunu veya iki tabloyu tek bir sonuç halinde alabilmek için kullanılır. Birleştirme sırasında her iki kümede de ortak olan değerler teke indirilir, ancak teke indirilmesi istenmiyorsa ve tekrar etmesi amaçlanıyorsa bu durumda UNION ALL sözcüğü kullanılır. SELECT komutlarının işlenmesi sonucu oluşacak tablolar aynı sayıda alan (sütun) içermelidir. Sonuç tablolarındaki karşılıklı sütunlar aynı veri tipi ve aynı genişlikte olmalıdır.	<pre>SELECT Bolum_ID, MAX(Pers_Maas) AS Maas FROM tbl_Personeller WHERE Bolum_ID=1 GROUP BY Bolum_ID  UNION  SELECT Bolum_ID, MIN(Pers_Maas) AS Maas FROM tbl_Personeller WHERE Bolum_ID=1 GROUP BY Bolum_ID</pre>
<b>EXCEPT (Fark)</b>	Küme farkı işlemi yapar. İki ayrı SELECT komutu sonucunda oluşan tabloların arasındaki küme farkını listeler (A-B). Bu işlem için de birçok yöntem kullanılabilir. NOT IN, NOT EXISTS bunlardan ikisidir.	<pre>SELECT Pers_ID, Pers_Adi FROM tbl_Personeller EXCEPT SELECT Pers_ID, Pers_Adi FROM tbl_Personeller WHERE Bolum_ID=2</pre>
<b>INTERSECT (Kesişim)</b>	INTERSECT sözcüğü küme kesişimi işlemi yapar. İki ayrı SELECT komutu sonucunda oluşan tabloların içindeki ortak kayıtları listeler. Ve (AND) operatörü gibidir.	<pre>SELECT Bolum_ID FROM tbl_Personeller WHERE Pers_ILi LIKE '%İzmir%' INTERSECT SELECT Bolum_ID FROM tbl_Personeller WHERE Pers_ILi LIKE '%Zonguldak%'</pre>
	İki tablonun Kartezyen çarpımını verir (NxM).	

## **PIVOT TABLE**

T-SQL'le eklenen yeni bir fonksiyondur. Bir tablonun bir sütunundaki değerler sütun haline getirilerek diğer bir sütunun COUNT, SUM (toplam) alınabilir. Excel'deki Pivot tabloyu oluşturmayı sağlar. Yandaki sorgunun sonucu Şekil-6.14'te görülmektedir.

```
SELECT * FROM  
(SELECT Pers_Adi, Pers_ILi,  
COUNT(Pers_ID) AS  
ElemanSayisi  
FROM tbl_Personeller  
GROUP BY Pers_Adi,  
Pers_ILi  
) AS ElemanSayisi  
PIVOT  
(COUNT(ElemanSayisi) FOR  
Pers_ILi IN ([Ankara],  
[İstanbul],[İzmir],[Zonguldak]  
) AS P
```

# VIII. Hafta

## SQL Veri Sorgulama Dili Fonksiyonlar ve JOIN İşlemleri

Mustafa Çoruh

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

<b>INNER JOIN</b>	İki tablodaki verilerin tek bir tabloda (View) birleřtirmesini sađlar. Bu birleřtirme Referans Alan deđerleri üzerinden yapılır.	<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>
<b>OUTER (LEFT/RIGHT) JOIN</b>	OUTER JOIN, birleřtirilen tablolarda ortak olmayan kayıtların listelenmesini sađlar. Bu iřlem iin veritabanına bađlı 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>
<b>SELF JOIN (Özyineli Birleřtirme)</b>	Bu durum, İVT tasarımındaki özyineli bađıntı (self relationship)'den dođan yapılarda kullanılır. Özyineli bađıntılar kurulurken, yapının daha iyi anlaşılabilmesi iin, tabloya ait veri kümelerinden iki farklı veri kümesinin oluřturulduđu düşünülebilir.	<i>SELECT P1.Pers_ID, P1.Pers_Adi, P2.Pers_Soyadi FROM tbl_Personeller AS P1, tbl_Personeller AS P2</i>
<b>NON EQUI JOIN</b>	Bazı durumlarda, birleřme kořulunda eřitlik deđil eřitsizlik aranır. Bu tür kořullara eřit olmayan birleřme kořulu (non equi-join) denilir. Bu amala <, >, <> 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 &gt; 1000 AND P2.Pers_Maas &lt; 5000</i>

# IX. Hafta

## SQL SQL Veri Sorgulama Dilinde (VSD) Fonksiyonlar ve Operantlar

Mustafa Çoruh

**Tablo-5.9: Gruplandırma Fonksiyonları**

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

X. Hafta

# SQL Veri İşleme Dil (VİD)

Mustafa Çoruh

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

<b>Komut</b>	<b>Açıklama</b>	<b>Örnek</b>
<b>INSERT INTO Tablo İsmi</b> <b>(Alan İsimleri)</b> <b>VALUES</b> <b>(Veriler)</b>	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>
<b>INSERT INTO Tablo İsmi</b> <b>(Alan İsimleri) SELECT</b> ...	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>
<b>SELECT (Alan İsimleri)</b> <b>INTO Tablo İsmi</b> <b>FROM Kaynak Tablo</b> <b>İsmi</b>	Kaynak tablodan verilerin hedef tabloya kopyalanmasını sağlar.	<i>SELECT Pers_Adi, Pers_Soyadi, Pers_DTarihi, Bolum_ID INTO tbl_PersonellerTest FROM tbl_PersonellerPaz</i>



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

<b>Komut</b>	<b>Açıklama</b>	<b>Örnek</b>
<b>UPDATE Tablo İsmi SET ...</b>	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>
<b>DELETE FROM Tablo İsmi</b>	Veritabanından verilerin silinmesini sağlar.	<i>DELETE FROM tbl_Personeller WHERE Pers_Soyadi LIKE '%Ser%'</i>
<b>TRUNCATE TABLE Tablo İsmi</b>	Bir tablonun tüm satırlarını yok etmek için kullanılır.	<i>TRUNCATE TABLE tbl_Personeller</i>
<b>MERGE INTO</b>	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>

XI. Hafta

**SQL Veri Hareket Dili (VHD) ve  
Veri Güvenlik ve Kontrol Dili  
(VGKD)**

Mustafa Çoruh

**Tablo-5.12: SQL Veri Hareket Dili Komutları**

<b>Komut</b>	<b>Açıklama</b>	<b>Örnek</b>
<b>COMMIT</b>	SQL komutlarının içerdiği işlemleri kalıcı şekilde kaydetmek için kullanılır. Hareketi değişiklikleri kaydederek sonlandırır.	<i>BEGIN TRANSACTION</i> <i>INSERT INTO tbl_Personeller (Pers_Adi, Pers_Soyadi, Bolum_ID, Pers_Aktif_Mi) VALUES ('Durdu', 'Çoruh', 6, 0)</i> <i>COMMIT TRANSACTION</i>
<b>ROLLBACK</b>	Hareketin içerdiği işlemleri tamamını kalıcı şekilde geri almak için kullanılır. Hareketi değişiklikleri geri alarak sonlandırır. Yandaki örnekte Fatma Çoruh kaydı girilmez.	<i>BEGIN TRANSACTION</i> <i>INSERT INTO tbl_Personeller (Pers_Adi, Pers_Soyadi, Bolum_ID, Pers_Aktif_Mi) VALUES ('Fatma', 'Çoruh', 6, 0)</i> <i>ROLLBACK TRANSACTION</i>
<b>SAVE</b>	İşlemleri tanımlı bir noktaya kadar geri almak için kullanılır. Hareket içerisinde kısmi geri dönüş noktaları tanımlar	<i>BEGIN TRANSACTION</i> <i>UPDATE tbl_Personeller SET Pers_Adi = 'Mustafa2'</i> <i>WHERE Pers_ID = 1;</i> <i>SAVE TRANSACTION save_point_Adi;</i> <i>UPDATE tbl_Personeller SET Pers_Adi = 'Ayşe2'</i> <i>WHERE Pers_ID = 4;</i> <i>ROLLBACK TRANSACTION save_point_Adi;</i> <i>COMMIT TRANSACTION</i>
<b>SET TRANSACTION</b>	İşlemlerdeki hareketlerin nasıl okunacağını belirler.	<i>SET TRANSACTION ISOLATION LEVEL READ COMMITTED</i> <i>SET TRANSACTION ISOLATION LEVEL SERIALIZABLE</i> <i>SET TRANSACTION ISOLATION LEVEL SNAPSHOT</i>

**Tablo-5.13: SQL Veri Güvenlik ve Kontrol Dil Komutları**

<b>Komut</b>	<b>Açıklama</b>	<b>Örnek</b>
<b>GRANT</b>	Bir kullanıcının verileri kullanmasına ve T-SQL komutlarını çalıştırmasına izin verir. GRANT komutu, veritabanı kullanıcılarına, veritabanı rolüne ya da uygulama rolüne izinler vermek için kullanılır. Yandaki örneklerde Emre ve Burak adlı kullanıcıların daha önce yaratılmış olması gerekir.	<i>CREATE USER Emre WITHOUT LOGIN</i> <i>CREATE USER Burak WITHOUT LOGIN</i>  <i>GRANT CREATE TABLE TO Emre, Burak</i> <i>GRANT SELECT ON OBJECT::tbl_Personeller TO Burak</i>
<b>DENY</b>	Bir kullanıcının verileri kullanmasını kısıtlar. Kullanıcıların haklarını kısıtlamak için kullanılan komuttur.	<i>DENY CREATE TABLE ON Emre</i> <i>DENY SELECT ON tbl_Personeller TO Burak</i>
<b>REVOKE</b>	Daha önce yapılmış olan tüm GRANT ve DENY komutlarını iptal eder. Ancak bir nesneyi oluşturan kullanıcının, nesne üzerindeki yetkilendirme ve kullanma hakkı REVOKE komutu ile iptal edilemez.	<i>REVOKE SELECT ON tbl_Personeller TO Public</i> <i>REVOKE SELECT ON OBJECT::tbl_Personeller FROM Emre</i>

# XII. Hafta

## SQL Serverda Programlama Nesneleri Oluřturma

Mustafa oruh

# SQL Server Nesne Konuları

- Trigger Oluşturma
- Stored Procedure Oluşturma
- Function Oluşturma
- Döngü Komutları (IF ELSE, WHILE, CASE)

# XIII. Hafta

## SQL Serverda XML Uygulamaları

Mustafa Çoruh

# XML Konuları

- XML Komutları
- XML Dosyası Oluşturma
- XML Dosyası Okuma



# XIV. Hafta

## Veritabanı Proje Sunumları

Mustafa Çoruh