

İLERİ VERİ TABANI UYGULAMALARI

VERİ TABANI YÖNETİM SİSTEMLERİ

VERİ TABANI NEDİR, NERELERDE KULLANILIR?

VERİ TABANI YAZILIMLARI VE İŞLEYİŞİ

VERİ TABANI MİMARİLERİ

VERİ TABANI PROBLEMLERİ



VERİ TABANI NEDİR?

VERİ TABANI HAKKINDA

Veri tabanları birbirleriyle ilişkili bilgilerin depolandığı alanlardır.

Bilgi artışıyla birlikte bilgisayarda bilgi depolama ve bilgiye erişim konularında yeni yöntemlere ihtiyaç duyulmuştur.

Veri tabanları; büyük miktardaki bilgileri depolamada geleneksel yöntem olan “**dosya-işlem sistemine**” alternatif olarak geliştirilmiştir.

Kısacası **bilgi depolarıdır**.



VERİ TABANI NEDİR, NERELEERDE KULLANILIR?

NEDEN VERİ TABANI KULLANILIR

- Verilerin tutulması, saklanması ve erişilmesinde geleneksel yaklaşım verilerin ayrı ayrı dosyalarda gruplanması yaklaşımını kullanmaktadır.
- Verilerin artması, verilere aynı anda erişme ve düzenlenme ihtiyacı ile geleneksel yaklaşım yetersiz kalmıştır.



VERİ TABANI NEDİR, NERELEERDE KULLANILIR?

KULLANILDIĞI ALANLAR

- Üniversite- Öğrenci İşleri Bilgi Sistemi
- Hastane-Hasta, doktor, tedavi, araç-gereç, mali bilgiler
- Ticari bir şirket- Müşteri, Ürün, Satış, Ödeme, Teslimat bilgileri
- Banka-Müşteri, mevduat, kredi kartı, kredi bilgileri
- Hava yolları, ulaşım firmaları,
- Kısacası her alanda...



VERİ TABANI NEDİR, NERELEERDE KULLANILIR?

VERİ TABANI KULLANMANIN AVANTAJLARI

- Veri Tabanları veriyi belirli şekillerde kategorize etmeyi sağlar
- Birbiriyle ilişkisi olan verileri bir araya getirerek bilgiye kolay bir şekilde ulaşmamızı sağlamaktadır.
- Veri Tabanının Amacı, veriyi en iyi şekilde saklayarak, veriye en hızlı ve en doğru şekilde erişmektir.
- Ortak verilerin tekrarını önlenmesi.



VERİ TABANI NEDİR, NERELEERDE KULLANILIR?

VERİ TABANI KULLANMANIN AVANTAJLARI

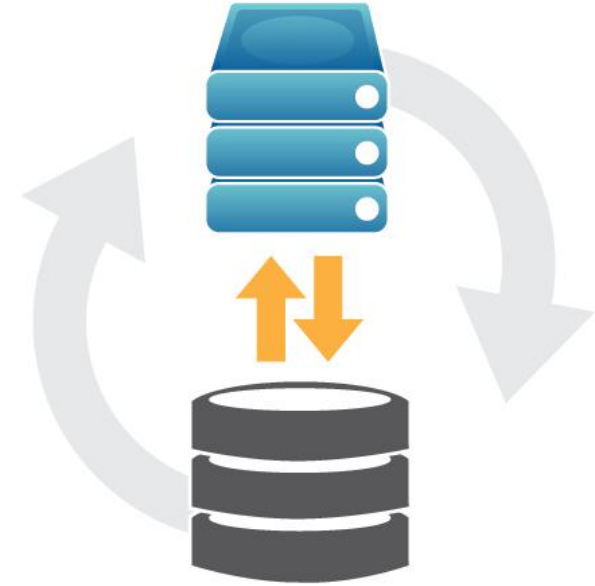
- Verilerin merkezi denetiminin ve tutarlılığının sağlanması
- Veri paylaşımının sağlanması
- Fiziksel yapı ve erişim yöntemi karmaşıklıklarının, çok katmanlı mimarilerle kullanıcıdan gizlenmesi,
- Her kullanıcıya yalnız ilgilendiği verilerin, alışık olduğu kolay, anlaşılır yapılarda sunulması.



VERİ TABANI NEDİR, NERELEERDE KULLANILIR?

VERİ TABANI KULLANMANIN AVANTAJLARI

- Sunulan çözümleme, tasarım ve geliştirme araçları ile uygulama yazılımı geliştirmenin kolaylaşması.
- Veri bütünlüğünün gerekli olanakların sağlanması,
- Güvenlik ve gizliliğin istenilen düzeyde sağlanması
- Yedekleme, yeniden başlatma, onarma gibi işletim sorunlarına çözüm getirilmesi



VERİ TABANI NEDİR, NERELEERDE KULLANILIR?

VERİ TABANI ÖNEMİ!

- Veri Tabanındaki, veri değişikliklerinden verinin nasıl etkileneceğine doğru karar verebilmek!
- Veri Tabanındaki, verinin kimin erişebileceğine, nasıl erişebileceğine karar verebilmek. (Okuma, Yazma, Güncelleme Hakları)
- Verinin mümkün olan en verimli şekilde tutulması (**Klasik Veri Tabanı Uygulamaları için!**)



VERİ TABANI YAZILIMLARI VE İŞLEYİŞİ

VERİ TABANI YAZILIMLARI

Veri Tabanı Programları; verilere erişebilmemizi sağlayan programlardır. Bilgisayarın çeşitli donanımsal ihtiyaçlarını kullanırlar.

Data Management System de denilebilir. Micro bir işletim sistemi olarakta adlandırılabilirler.

Veri Tabanı programlarını daha verimli kullanabilmeniz için yaşanacak problemleri önceden ön görmemiz gereklidir.



VERİ TABANI YAZILIMLARI VE İŞLEYİŞİ

VERİ TABANI YAZILIMLARI

- **Oracle Database**
- **IBM DB/2**
- Adaptive Server Enterprise
- Informix
- Microsoft Access
- **Microsoft SQL Server**
- Microsoft Visual FoxPro
- **MySQL**
- PostgreSQL
- Progress
- SQLite
- Teradata
- CSQL
- OpenLink Virtuoso

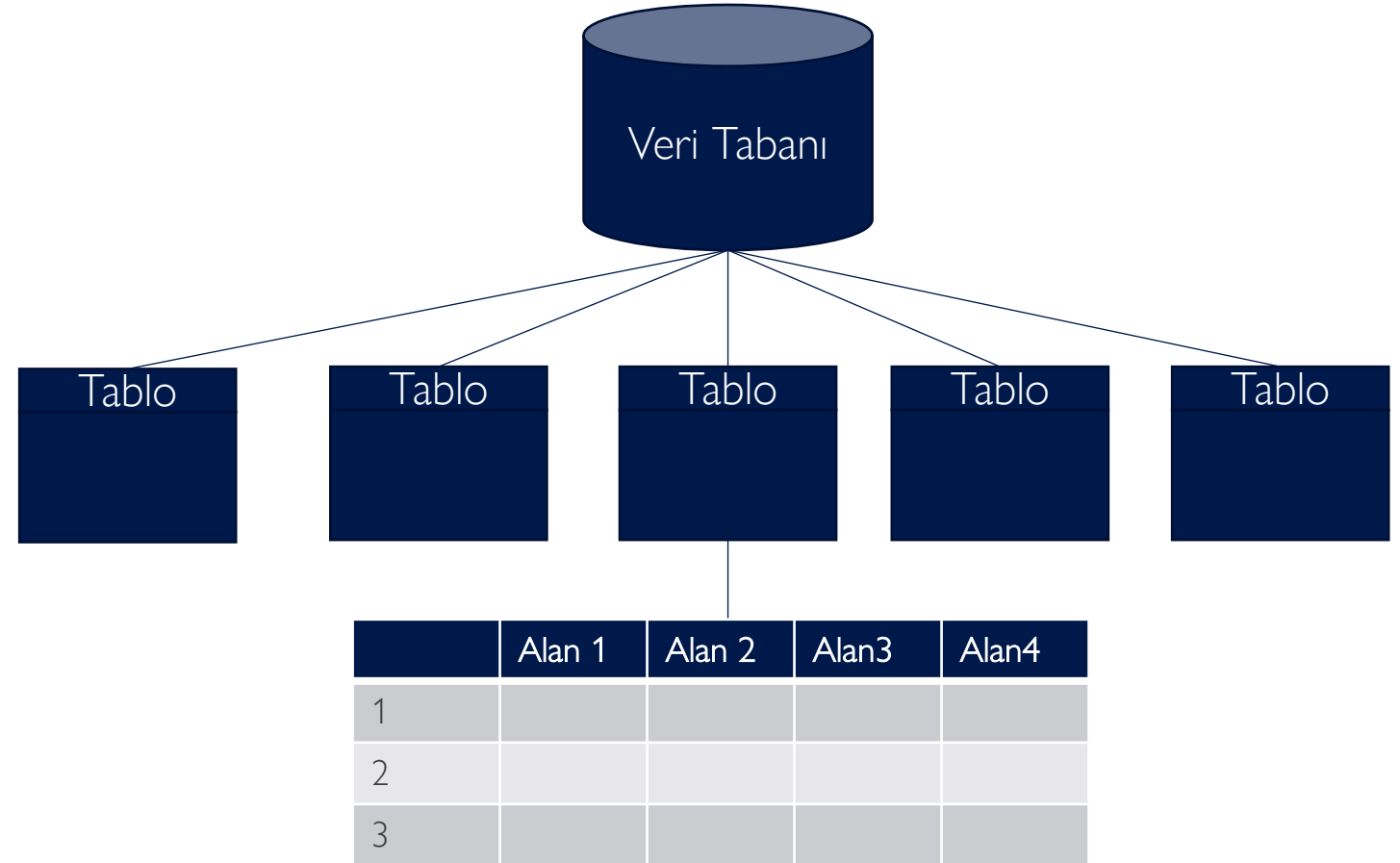
VERİ TABANI DİLLERİ

- PL/SQL
- SQL
- Tcl
- Transact-SQL (T-SQL)

VERİ TABANI YAZILIMLARI VE İŞLEYİŞİ

VERİ TABANI YAPISI

- Bir veri tabanı tablolarda saklanan verilerden oluşur.
- Tablolar verilerin satırlar ve sütunlar halinde düzenlenmesiyle oluşan veri grubudur.



VERİ TABANI MİMARİLERİ

GENEL VERİ TABANI YAPISI

- Tablo içindeki her bir bilgi kayıt,
- Sütunlar ise alan olarak isimlendirilir.

Örnek: Öğrenci Bilgileri Tablosu

Öğrenci Adı Soyadı, Doğum Tarihi,
Doğum Yeri, Mail Adresi gibi...

Alan

Ogr_no	Ad_soyad	d_tarih	d_yeri	e-mail
1	Alican HAZİR	20.12.1987	İzmir	alican@mail.com
2	Onur Önder	08.10.1988	İzmir	onur@mail.com
3	Mustafa Gencer	12.10.1988	İzmir	mustafa@mail.com
4	Onur Kasım	06.06.1988	İstanbul	kasim@mail.com

Kayıt

VERİ TABANI MİMARİLERİ

GENEL VERİ TABANI YAPISI

- Tablo içindeki her bir bilgi kayıttır.
- Sütunlar ise **alan** olarak isimlendirilmektedir.
- Veri tabanında tutulan kayıtların yapısı hakkında bilgi sahibi olmak için alanların bazı özelliklerinin önceden tanımlanması gerekir.

Örnek: Öğrenci Numarası mutlaka tam sayı, Ad ve Soyad harflerden oluşması gibi...



VERİ TABANI MİMARİLERİ

VERİ TABANI TASARLAMA

- Bir veri tabanı tasarlarken ilk olarak nesneleri tanımlamamız gereklidir

Örnek: Öğrenci Bilgi Sistemi için; Öğrenci No, Öğrenci Adı ve Soyadı, Fakülte, Bölümü, Doğum Yılı, Mail gibi...

- Tanımladığımız her nesne için tabloda bir sütun oluştururuz. Ayrıca oluşturduğumuz her sütun için bir alan belirlememiz gerekir.

Örnek: Öğrenci Numarasında; sadece sayısal(numaric) bir alan belirlenmesi gibi...



VERİ TABANI MİMARİLERİ

VERİ TABANI TASARLAMA

- Oluşturulan veri tabanında her tablo için bir anahtar belirlenir.

Örnek: Öğrenciler tablosunda anahtar olarak öğrenci numarası alanının belirlenmesi gibi.

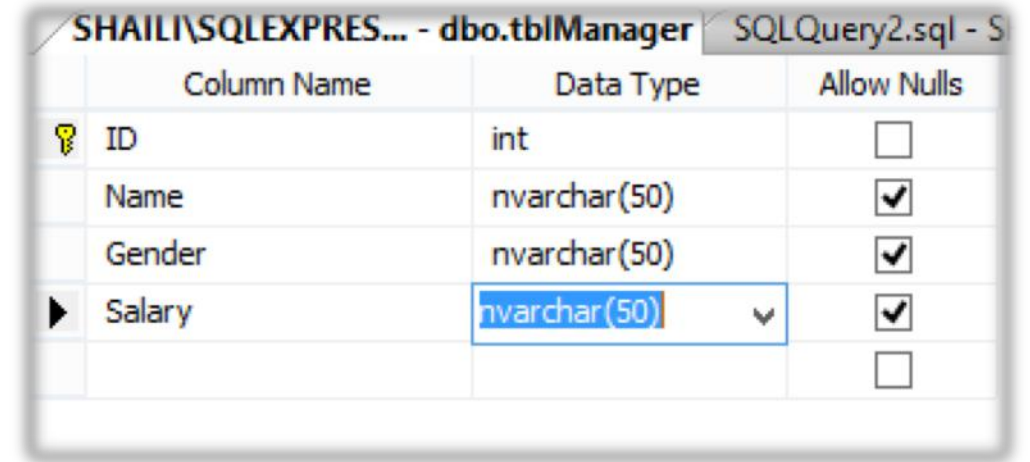
- Tekrarlanan nesne özellikleri için ek tablolar oluşturulmaktadır.
- Tablo ile doğrudan ilişkili olmayan alanlar belirlenir. (Inner-Join kullanımı)
- Tablolar arasındaki ilişkiler tanımlanmalıdır. Tanımlanan tablodaki alanların birbiri ile ilişkisi tanımlanır.



VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ

- Veri tabanı içerisinde tablolar tasarlanırken, saklanacak her veri alanı için ihtiyacı karşılayan en uygun veri tipinin belirlenmesi gerekir.
- Özellikle çok fazla verinin tutulduğu veri tabanlarında performans kaybını azaltmak, veri tabanı ve yedeklerin kapladığı disk alanını en aza indirmek için veri tipleri dikkatle seçilmelidir.



The screenshot shows the 'tblManager' table in the 'dbo' schema of a SQL Server instance. The table has four columns: ID (int, primary key), Name (nvarchar(50)), Gender (nvarchar(50)), and Salary (nvarchar(50)). The 'Allow Nulls' column indicates that ID is not nullable, while Name, Gender, and Salary are nullable.

Column Name	Data Type	Allow Nulls
ID	int	<input type="checkbox"/>
Name	nvarchar(50)	<input checked="" type="checkbox"/>
Gender	nvarchar(50)	<input checked="" type="checkbox"/>
Salary	nvarchar(50)	<input checked="" type="checkbox"/>

VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Metinsel

- **char:** Unicode olmayan sabit uzunlukta karakter verisi saklamak için kullanılır. Boyutu değişken olmakla birlikte en fazla 8000 karakter veri saklayabilir.
- **varchar:** Unicode olmayan değişken uzunlukta karakter verisi saklamak için kullanılır. Belirlenmiş veri kapasitesi 8000 karakter olmasına rağmen (max) parametresi ile bu değer 2^{31} byte' a kadar genişletilebilir.
- **text:** Daha önceki sql versiyonlarını desteklemek amacıyla Sql Server 2012 içerisinde hala yer almaktadır. Bu veri tipi yerine varchar(max) kullanın.

•

VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Metinsel

- **nchar:** Sabit uzunlukta 4000 karakter unicode karakter verisi saklamak için kullanılır. Boyutu değişken olmakla birlikte kısa olan değerler atanan uzunluğa tamamlanır.
- **nvarchar:** Değişken uzunlukta unicode karakter verisi saklamak için kullanılır. Belirlenmiş maksimum uzunluk 4000 karakter olmasına rağmen (max) parametresiyle 2^{31} byte'a kadar veri depolanabilir.
- **ntext:** text veri tipi gibi, bu veri tipide sadece eski versiyonları desteklemek amacıyla mevcuttur. Text tipinden tek farkı unicode veri saklayabilir.

VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Sayısal(Numeric)

- **bit:** Bir byte uzunluğunda tamsayı veri tipidir. Boyutu aklınızı karıştırmayın. Tablodaki ilk bit, bir byte büyüklüğünde yer kaplar, sonraki yedi bit aynı byte'ı kullanabilir.
- **tinyint:** 1 byte büyüklüğünde 0 ile 255 arasında değer alabilen tamsayı veri tipidir.
- **smallint:** 2 byte büyüklüğünde -32768 ile 32767 arasında değer alabilen tamsayı veri tipidir.
- **int:** 4 byte büyüklüğünde yaklaşık -2 milyar ile +2 milyar arasında değer alabilen tamsayı veri tipidir.
- **bigint:** 8 byte büyüklüğünde -2^{63} ile $+2^{63}$ arasında değer alabilen tamsayı veri tipidir.

VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Sayısal(Numeric)

- **decimal ve numeric:** Decimal ve nümeric veri tipleri adları farklı olmasına rağmen kullanımları aynıdır. Boyutu değişken olmakla birlikte -10^{38} ile $+10^{38}$ arasında ondalık ve tamsayı türünde verileri saklayabilir.
- **float:** Boyutu ve doğruluğu (ondalık kısım duyarlılığı) aldığı parametreye göre değişen kayan noktalı sayılar için kullanılır.
- **real:** Float veri tipinin parametre olarak 24 değerini aldığı anda ortaya çıkan halidir. $-3.40E+38$ ile $3.40E+38$ arasında değer alır.

VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ - Parasal

- **smallmoney:** 4 byte uzunluğunda yaklaşık -214 000 ile 214 000 arasında parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır.
- **money:** 8 byte uzunluğunda yaklaşık -922 milyar ile 922 milyar arasındaki parasal değerleri tutmak için kullanılır. Ondalık kısım duyarlılığı 4 basamaktır.



VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Tarih ve Zaman

- **date:** Tarihleri YYYY-MM-DD şeklinde saklayan 3 byte uzunluğunda veri tipidir. 0001-01-01 ile 9999-12-31 tarihleri arasındaki tüm değerleri tutabilir.
- **smalldatetime:** Tarih ve zaman verilerini YYYY-MM-DD hh:mm:ss şeklinde saklayan 4 byte uzunluğunda veri tipidir. 1900-01-01 00:00:00 ile 2079-06-06 23:59:59 arasındaki tarih ve zaman değerlerini saklar.
- **datetime:** YYYY-MM-DD hh:mm:ss:mmm şeklinde tarih ve zaman verilerini tutan 8 byte uzunluğunda veri tipidir. 1753-01-01 00:00:00.000 ile 9999-12-31 23:59:59.999 arası değerlerini saklar.

VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Tarih ve Zaman

- **datetime2:** datetime ile arasındaki fark salise hassasiyetinin daha yüksek olmasıdır. 7 basamağa kadar salise hassasiyeti kullanıcı tarafından belirlenebilir. Kapladığı alan salise hassasiyetine göre değişir. (6-8 byte)
- **time:** Sadece saat verilerini hh:mm:ss:nnnnnnnn şeklinde saklayan veri tipidir. Datetime2 gibi salise hassasiyeti maksimum 7 basamaktır ve kullanıcı tarafından değiştirilebilir. (3-5 byte)
- **datetimeoffset:** Ülkelere göre değişen zaman farkını tutmak için kullanılır.

VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ - Binary(İkilik)

- **binary:** Maksimum 8000 byte boyutunda, sabit uzunlukta binary veri saklamak için kullanılır.
- **varbinary:** Değişken uzunlukta 8000 byte boyutunda binary değer saklamak için kullanılır. (max) parametresiyle tutacağı maksimum değer 2^{31} byte'a kadar yükseltilebilir.
- **image:** Önceki sql versiyonlarını desteklemek amacıyla kullanılmaktadır. Bu tip yerine varbinary (max) veri tipi tercih edilmelidir.



VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Diğer

- **timestamp ve rowversion:** Tabloya kayıt eklendiğinde ve güncellendiğinde binary türünde özel bir değer alan veri tipidir. Tarih veri tipleriyle alakası yoktur. Kullanıcı bu alanı değiştiremez veya kendisi değer atayamaz.
- **uniqueidentifier:** 16 byte uzunluğunda eşsiz bir GUID değeri saklamak için kullanılır.
- **xml:** XML türünde hiyerarşik verileri saklamak için kullanılır. Bellekteki boyutu saklanan xml verisine göre değişir.



VERİ TABANI MİMARİLERİ

VERİ TÜRLERİ – Diğer

- **geography:** Coğrafi koordinatları ve gps verilerini tutmak için kullanılır.
- **geometry:** Öklid koordinat sistemine ait verileri tutmak için kullanılır. Geometrik şekillerin en, boy, yükseklik verilerini saklar.
- **hierarchyid:** Ağaç veri modeli veya hiyerarşik olarak sınıflandırılmış verileri saklamak için kullanılır.



VERİ TABANI MİMARİLERİ

ANAHTARLAR

- Anahtar bir veya birden fazla alanın bir satır için niteleyici olarak girilmesi için zorlanan bir çeşit zorlayıcıdır.
- 2 çeşit anahtar vardır:
 - Birincil Anahtar (Primary Key)
 - Yabancı Anahtar (Foreign Key)



VERİ TABANI MİMARİLERİ

BİRİNCİL ANAHTAR

- Bir kayıta ulaşmayı sağlayacak anahtar veridir.

Örnek: Öğrenciler arasında iki Ahmet var. Arama yaparken istediğimiz Ahmet'i bulmak için her bir öğrenciye özel bir numara olmalıdır.

Örnek: Öğrenci Numarası

- Birden fazla alanda birlikte birincil anahtar olabilir



VERİ TABANI MİMARİLERİ

YABANCI ANAHTAR

- Bir tabloya girilebilecek kayıtları başka bir tablonun belli alanındaki verilerle
 - Sınırlandırmaya ve
 - İlişkilendirmeye yarar.

Örnek: Öğrencilerin not verilerinin girildikleri tablodaki her satıra öğrenci bilgileri tablosundaki öğrenci numarası ile eşleşen bir değer girilmesi gibi.



VERİ TABANI PROBLEMLERİ

GENEL VERİ TABANI PROBLEMLERİ!


- Redundancy (Gereksiz) / Inconsistency (Verinin Özelliği)
- Accessing (Veriye erişebilmek için gerekli yazılım)
- Verinin İzole edilmesi (Çok Sayıda Formatta Olmaması)
- Integrity (Koşulların Bağlanması)
- Atomicity (Banka Para Transferi A->B Hesabı)
- Concurrency (Birden fazla süreç aynı anda çalışırsa bir biriyle senkonize olması gereklidir.)
- Security (Kimin veriye erişeceği ve nasıl erişeceği?)



TEŞEKKÜRLER

[HTTP://WWW.ALICANHAZIR.COM](http://www.alicanhazir.com)

 **Alican HAZIR**

 +90 (850) 304-1-456

 info@alicanhazir.com

