

08118 Veri Tabanı I



Database Management System

Elbistan Meslek Yüksek Okulu
2013 – 2014 Bahar Yarıyılı

08118 Veri Tabanı I

NORMALİZASYON KURALLARI



- Bir tablo içerisinde yer alacak kaydın nelerden oluştuğuna karar vermeye yarayan normalizasyon kuralları başlı başına bir işlemdir.
- Normalizasyon; veritabanı tasarım aşamasında yapılması gereken bir işlemdir.

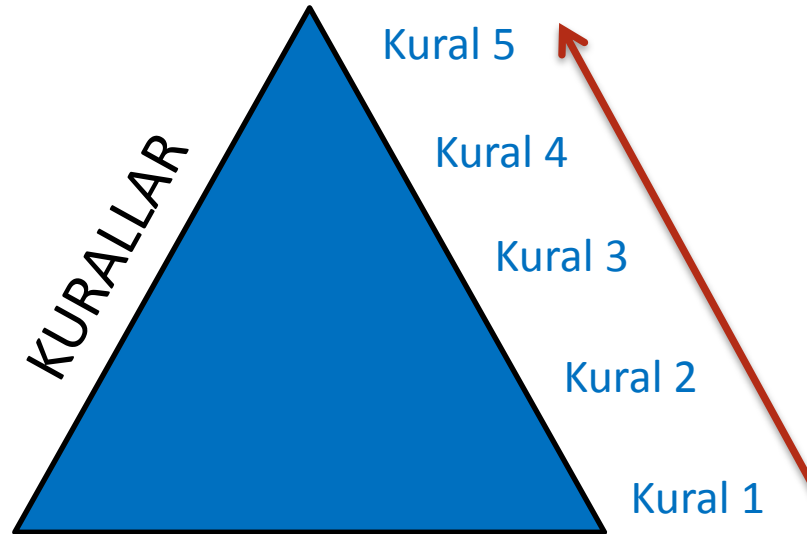
08118 Veri Tabanı I

VERİ TABANI NORMALİZASYONU



- Genel kabul görmüş 5 normalizasyon kuralı
- İlişkisel veritabanının tanımı ile birlikte ortaya konulmuştur.

Normalizasyon = Excellent DB



08118 Veri Tabanı I



I. Normalizasyon Kuralı :

- Bir satırdaki bir alan yalnızca bir tek bilgi içerebilir.

kitap

kitap_id	kitap_adi
K_1234	Veri Tabanı I

08118 Veri Tabanı I



I. Normalizasyon Kuralı :

- Birden fazla yazarı olan kitap için yazar1, yazar2 ve yazar3 diye alanların açılması ile bu kurala uyulmamış olunur.

Kitap_id	Kitap_adi	Yazar1	Yazar2	Yazar_n...
1234	Veri Tabanı I	Ahmet Ölmez	Selami Kelle
2345	Int. Programcılığı	Muratok	Selami Kelle	

- Veri tabanı tasarımında; verileri virgül veya bir başka karakter ile ayrılıp aynı alana girilmesi ve daha sonra program içerisinde split ile bu değerlerin ayrılması genellikle sık yapılan hatalardan birisidir.
- Bu durum ilişkisel veritabanının doğasına aykırıdır.

08118 Veri Tabanı I



I. Normalizasyon Kuralı :

- Bir satırdaki bir alan yalnızca bir tek bilgi içerebilir.

kitap

kitap_id	kitap_adi
K_1234	Veri Tabanı I

yazar

Yazar_id	YKitap_id	Y_adi	Yazar_soyadi
Y_001	K_1234	Selami	Kelle
Y_002	K_1234	Ahmet	Ölmez

Böyle bir durumda, ayrıca yazarlar tablosu da oluşturularak kural çiğnenmemiş olur.

08118 Veri Tabanı I



2. Normalizasyon Kuralı :

- Bir tablo için, anahtar olmayan her alan, birincil anahtar olarak tanımlı tüm alanlara bağlı olmalıdır.

Odunc [Tablosu]

odunc_id	okitap_id	odunc_adi	odunc_ver_tarihi	Kitap_adi
O_2731	K_1234	Selami	10.01.2013	Veri Tabanı I
O_1875	K_1234	Ahmet	15.02.2013	Veri Tabanı I
O_3456	K_1234	Mehmet	23.02.2013	Veri Tabanı I

Primary Key
Unique key

Foreign Key

Alan I

Alan II

?

- Örneğin, Ödünç tablosuna Kitap_Adi diye bir alan eklense idi, bu sadece ödünç verilen kitap ile ilgili bir bilgi olacaktı ve odunc_id 'ya bağlı bir nitelik olmayacaktı.

08118 Veri Tabanı I

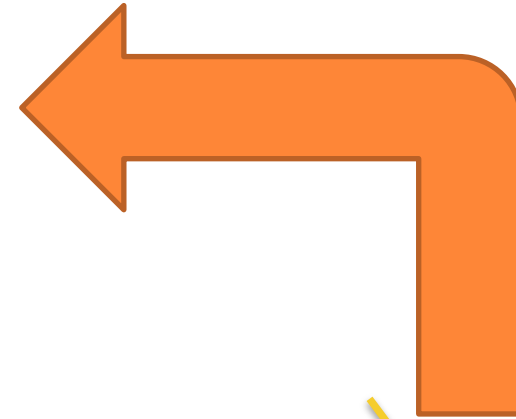


2. Normalizasyon Kuralı :

- Bunu durumu çözmek için, kitap adları ayrı bir tabloda tutulur.

Kitap [Tablosu]

kitap_id	kitap_adi
K_1234	Veri Tabanı I
K_5678	Ağ Temelleri



Odunc [Tablosu]

odunc_id	okitap_id	odunc_adi	odunc_ver_tarihi	Kitap_adi
O_2731	K_1234	Selami	10.01.2013	Veri Tabanı I
O_1875	K_1234	Ahmet	15.02.2013	Veri Tabanı I
O_3456	K_1234	Mehmet	23.02.2013	Veri Tabanı I



08118 Veri Tabanı I



3. Normalizasyon Kuralı :

- Anahtar alanın birden fazla alandan oluştuğu tablolarda, anahtar alanlardan sadece birine bağlı veriler tabloda yer almamalı, ayrı bir tabloya taşınmalıdır.
- Bunun tersi de geçerlidir. Yani iki ya da daha fazla tablonun birincil anahtarı aynı olamaz.
- Böyle bir durum söz konusu ise, bu iki tablo tek tabloya indirilmelidir.

08118 Veri Tabanı I



3. Normalizasyon Kuralı :

- Bir tablo için, anahtarı olmayan bir alan, anahtarı olmayan başka hiç bir alana bağlı olamaz.

Kitap [Tablosu]

kitap_id	kitap_no	kitap_adi	Cilt_tipi
K_1234	987	Veri Tabanı I	K (Karton)
K_5678	537	Ağ Temelleri	S (Spiral Tel)
K_8765	234	Bilişim Kuramları	D (Deri)

- Örneğin, kitaplar için cilt tipi adında bir alan eklenip burada da karton kapak için K, deri cilt için D, spiral tel için S yazılsaydı, bu kodlama, kitap tablosunun birincil anahtarı olan kitap_no alanına bağlı bir kodlama olamazdı.

08118 Veri Tabanı I



3. Normalizasyon Kuralı :

- Cilt tipi bilgisini kodlu olarak tutan alan aslında cilt tipi açıklaması olan başka bir alana bağlıdır.

Kitap [Tablosu]

kitap_id	kitap_no	kitap_adi	Cilt_tipi
K_1234	987	Veri Tabanı I	K (Karton)
K_5678	537	Ağ Temelleri	S (Spiral Tel)
K_8765	234	Bilişim Kuramları	D (Deri)

- Bu ilişki başka bir tabloda tutulmalıdır. Bu durumda, cilt şekillerini tutan bir tablo açılması gerekir. Bu tablonun alanları da ciltTipKodu ve ciltSekli olabilir.

08118 Veri Tabanı I



3. Normalizasyon Kuralı :

- Yeni bir tablo oluşturulur ve iki tablo arasında ilişki kurulur.

Kitap [Tablosu]

kitap_id	kitap_no	kitap_adi	Cilt_tipi
K_1234	987	Veri Tabanı I	K (Karton)
K_5678	537	Ağ Temelleri	S (Spiral Tel)
K_8765	234	Bilişim Kuramları	D (Deri)

Cilt [Tablosu]

cilt_id	cilttpino	kitap_no	Cilt_tipi
c100	01	987	K (Karton)
c200	02	537	S (Spiral Tel)
c300	03	537	D (Deri)

4. Normalizasyon Kuralı :

- Birincil anahtar alanlar ile anahtarı olmayan alanlar arasında, birden fazla bağımsız bire-çok ilişkisine izin verilmez. Örneğin, tabloda yer alan bir kitap, hem hikaye kitabı hem de kişisel gelişim kitabı olabilir.

Kitap [Tablosu]

kitap_id	kitap_no	kitap_adi	Kitap_turu
K_1234	987	Atilla	Hikaye, Kişisel Gelişim
K_5678	537	Ağ Temelleri	Ders Kitabı
K_8765	234	12. Gezegen	Bilim Kurgu, Roman

- 4.Normal formu sağlamak için, her bağımsız bire çok ilişki için ayrı bir tablo oluşturulması gerekir.

4. Normalizasyon Kuralı :

- Bu örnekte, türler için yeni bir tablo açılması gerekir. Tablonun adına türler denilebilir.

Kitap [Tablosu]

kitap_id	kitap_no	kitap_adi	Kitap_turu
K_1234	987	Atilla	Hikaye, Kişisel Gelişim
K_5678	537	Ağ Temelleri	Ders Kitabı
K_8765	234	12. Gezegen	Bilim Kurgu, Roman

Turu [Tablosu]

kitapTur_id	kitap_no	Kitap_turu
A01	987	Hikaye
B05	537	Kişisel Gelişim
E14	234	Bilim Kurgu
C12	234	Roman

08118 Veri Tabanı I



5. Normalizasyon Kuralı :

- Tekrarlamaları ortadan kaldırmak için her bir tablonun mümkün olduğunca küçük parçalara bölünmesi gerekir.
- Aslında ilk 4 kural sonuçta bu işe yarar ancak, bu kurallar kapsamında olmayan tekrarlamalar da 5 normalizasyon kuralı ile giderilebilir.

08118 Veri Tabanı I



5. Normalizasyon Kuralı :

- Örneğin, kitaplar için bir edinme şekli bilgisi girilecek sütun eklenmek istenebilir: Bu bölüme girilebilecek bilgiler bellidir: Bağış - satın alma.

Kitap [Tablosu]

kitap_id	Kedinme_id	kitap_no	kitap_adi	Kitap_edinme_sekli
K_1234	E101	987	Atilla	Bağış
K_5678	E105	537	Ağ Temelleri	Satın Alma
K_8765	E101	234	12. Gezegen	Bağış

kedinme_id	kitap_no	Kitap_edinme_sekli
E101	987	Bağış
E105	537	Satın Alma
E101	234	Bağış

08118 Veri Tabanı I



5. Normalizasyon Kuralı :

- Böylelikle, kullanıcıların kitap edinme şekli alanına gelişi güzel bilgiler girmesi engellenmiş olur.
- Bu da sorgulama esnasında veriler arasında bir tutarlılık sağlar.
- Bu işlem sonucunda, tutarsızlıklara neden olabilecek ve sık tekrarlayan veriler başka bir tabloya taşınmış olur.
- Bu tablo için, veritabanı programlamada 'look-up table' terimi kullanılır.

08118 Veri Tabanı I

Normalizasyon Kuralları :



Veritabanı normalizasyon kuralları;

- Bir ilişkisel veritabanınının tasarlanma aşamalarını değil de ilişkisel veritabanında yer alacak kayıtların ilişkisel veritabanı ile uyumlu olup olmadığını denetlemeye yöneliktir.

08118 Veri Tabanı I



Normalizasyon Kuralları :

ilişkisel bir veritabanı tasarımı dört ögeyi barındırmalıdır;

1. Veri tekrarı yapılmamalıdır.
2. Boş yer mümkün olduğunca az olmalıdır.
3. Veri bütünlüğü sağlanmalıdır.
4. Veriler, aralarında bir ilişki tanımlanmaya müsait olmalıdır.

DBMS – Varoluş Koşulu

X varlığının bulunması Y varlığına bağlı ise;

- X'in Y'ye bağlı olduğundan sözedilir.
- Diğer bir deyiş ile Y silinirse, X'in bir anlamı kalmayacaktır.
- Bu durumda “o halde X silinmelidir” sonucuna ulaşılır.
- Böyle bir durumda Y'ye **Baskın varlık (dominant Entity)** X'e, **Bağımlı varlık (subordinate Entity)** adı verilir.

DBMS – Varoluş Koşulu

Örneğimizde açıklayalım;

- Bir işletmede, “ürün” ve “stok hareket” mevcut olduğunu varsayalım.

Ürün={ürün isimleri}

Stok hareket={Günlük ürün giriş ve çıkış miktarları}

- Burada varlıklar arasındaki ilişki birden-çoğa doğrudur.

DBMS – Varoluş Koşulu

Bu durumda;

- “Stok Hareket” varlık kümesinden bir varlığın silinmesi durumunda, “ürün” varlık kümesinden bir varlığın (ürün) silinmesi söz konusu olamaz.

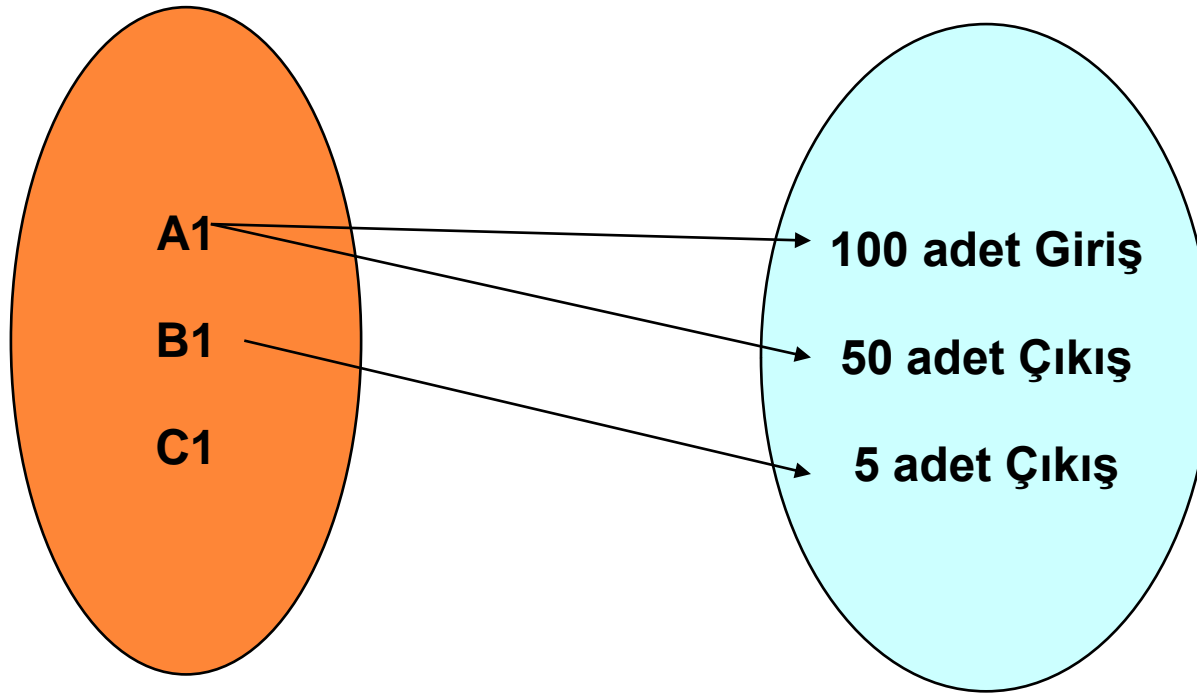
Ürün={ürün isimleri}

Stok hareket={Günlük ürün giriş ve çıkış miktarları}

- Bu varlık, “Stok Hareket” kümesindeki başka varlıklarlada ilişkilidir.

DBMS – Varoluş Koşulu

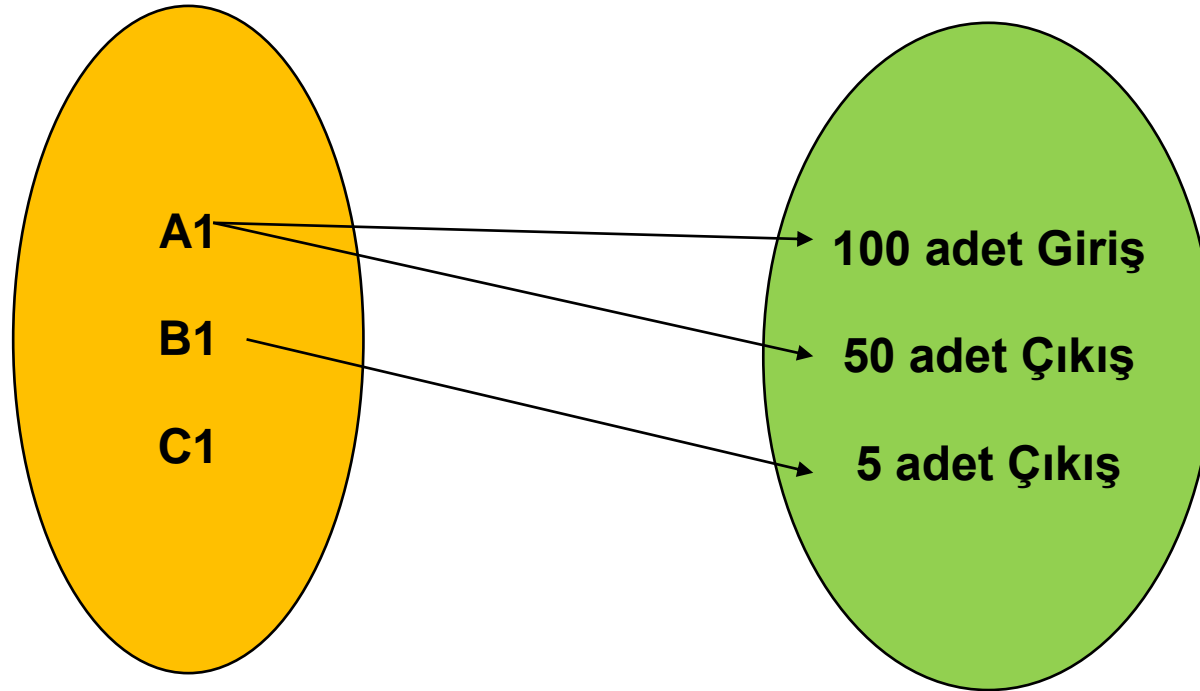
Bu durumu şematik küme olarak görelim;



- Ürün kümesinden bir varlık silinirse, o ürünün stok hareket kümesindeki verilerinin bir anlamı kalmayacaktır.

DBMS – Varoluş Koşulu

Baskın Varlık – Bağımlı Varlık;



- Ürün varlığı baskın, Stok hareketleri ise bağımlı varlık olacaktır.

DBMS – Varoluş Koşulu

Bir varlık veya ilişki kümesi içerisinde yer alan varlıklar birbirinden nasıl ayırt edilirler;

- Varlık Kümesi içindeki varlıkları birbirinden ayırt etmede anahtarlar kullanırız

Öğrenci={ogr_no, ogr_adi, ogr_soyadi}

- İki tür anahtar vardır; Süper Anahtar (superkey, Aday anahtar (candidate key)

DBMS – Anahtarlar

Superkey;

- Bir varlığı kesin olarak tanımlamaya yarayan anahtara “Süper Anahtar” denir.
- Bir nitelikte oluşabileceği gibi, birden fazla niteliğin birleşiminden de oluşabilir.
- Süper anahtarlar daima {süper küme} oluşturur.

DBMS – Anahtarlar

Superkey (Süper Anahtar);

- Örneğin, SSK numarası bir personeli diğerinden ayırt etmek için yeterlidir.
- Çünkü ülkede tüm çalışanların sadece birer tane SSK numarası vardır.
- İki farklı kişinin aynı SSK numarasına sahip olması imkansızdır.

DBMS – Anahtarlar

Superkey (Süper Anahtar);

- Personelin adı süper anahtar olarak tanımlanamaz. Aynı ismi taşıyan çok sayıda personel olabilir.

SSK NO	Adı	Bölümü
101	Nilay	Muhasebe
110	Burak	Muhasebe
215	Selin	Satış
134	Sezin	Satış
143	Dilay	Ambar

- SSK numarası, adı ve bölümü nitelikleri ile de superkey kabul edilebilir.

DBMS – Anahtarlar

Candidate Key;

- Varlığı kesin olarak tanımlamaya yarayan anahtar süper anahtar değil ise “yabancı anahtar” yada kısaca “anahtar” denir.
- Diğer bir ifade ile “süper anahtar” ın alt kümesi aynı zamanda bir süper anahtar değilse “aday anahtar” denir.
- Aday anahtar fazla nitelik içermeme özelliğine sahiptir.

DBMS – Değişken Notasyonları

Değişken isimleri tanımlaması

- SQL'de tablo adları, alan(field) , veritabanı dosyası, indeks vb. isimler değişken isimleridir.
- Genel geçer değişken isimlendirme kurallarına burada da dikkat etmek, sağlıklı uygulamalar meydana getirebilmek için çok önemlidir.
- Bazen oldukça kritik hatalar ortaya çıkabilmektedir.



DBMS – Değişken Kuralları

Değişken isimlendirme Kuralları

1. Değişken isimleri, harf ile başlamak zorundadır.
2. Değişken isimleri, harf, rakamlar ve ‘_’ dan oluşmak zorundadır.
3. Değişken isimlerinde Türkçe’de ki noktalı harfler (İ,ı,Ğ,ğ,Ü,ü,Ş,ş,Ç,ç,Ö,ö,) yer alamaz.
4. Değişken isimlerinde ayrılmış kelimeler yer alamaz. (select, like, not, or, delete, update ...vs)
5. SQL büyük-küçük harf duyarlı değildir.
6. Değişken isimlerinde boşluk yer alamaz.



DBMS – Değişken Notasyonları

Değişken İsimlendirme Notasyonları

Deve Notasyonu:

DegiskenAdi şeklinde yazılır.

Alt çizgi notasyonu:

degisken_adi şeklinde yazılır.



DBMS – Değişken Notasyonları

Değişken İsimlendirme Notasyonları

- Veritabanı programlamada, büyük-küçük harf duyarlılığı olmadığından genellikle **alt çizgi notasyonu** kullanılır.
- Değişken adları küçük harf olarak verilir.
- Ancak bu bir kural olmayıp sadece okunurluğu artırmak için programcılarının bir çoğu tarafından tercih edilen bir yoldur.

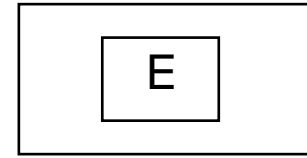


DBMS – Varlık- ilişki Şemaları

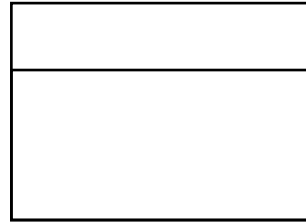
Varlık – İlişki şemalarında ilişkileri tanımlamak için kullanılır.



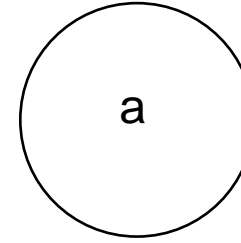
Varlık



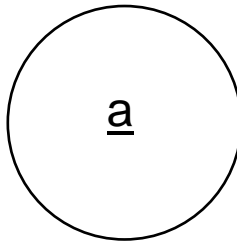
Zayıf Varlık



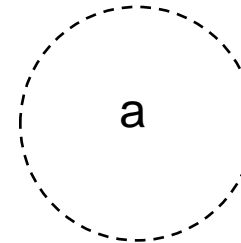
Bağımlı Varlık



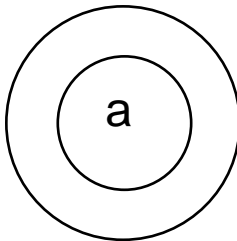
Nitelik



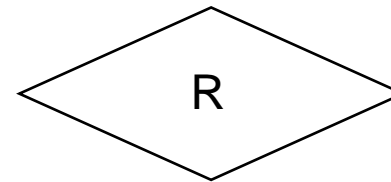
Anahtar Nitelik



Türetilen Nitelik



Çok Değerli Nitelik

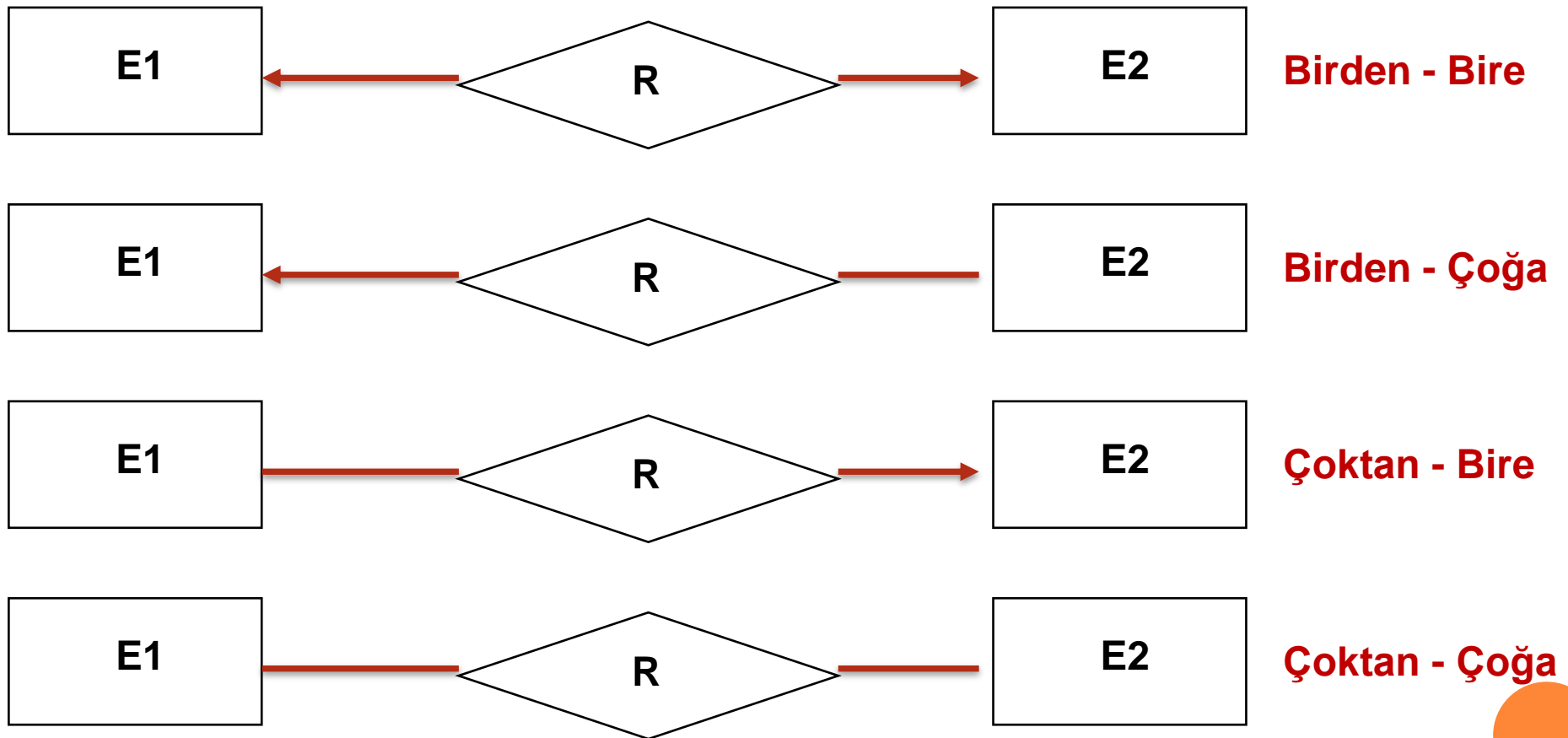


İlişki



DBMS – Varlık- ilişki Şemaları

Varlık – İlişki şemalarında ER Şemaları



SQL NEDİR ?

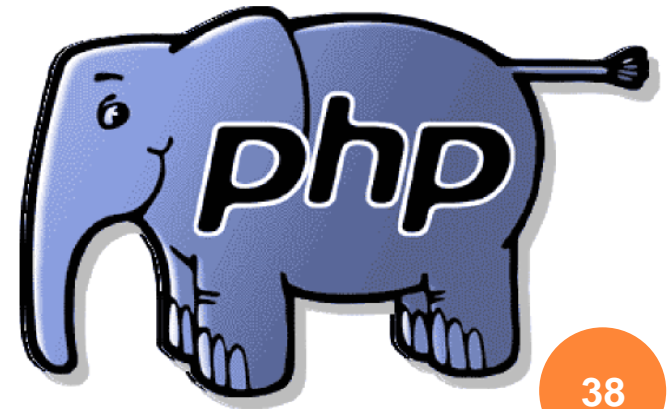
- SQL insanların veritabanı sistemleri ile konuşmasını sağlayan popüler bir dildir.
- Bu dil tüm veritabanı programlarında kullanılabilir.
- Bu dil sayesinde, bir veritabanından kayıtlar alınabilir, değiştirilebilir ya da yeni kayıtlar eklenebilir.

SQL NEDİR ?

- SQL bir dildir; ancak bir programlama dili değildir.
- Program geliştirme aşamasında SQL 'den faydalanılır, ancak tek başına bu iş için yeterli değildir.
- PHP, Asp, Visual Basic, Delphi, C, C++ gibi birçok teknoloji/programlama dili SQL komutlarını desteklemektedir.

SQL NEDİR ?

- Yani bir program geliştirme aşamasında; SQL komutlarını bilmek gerekmektedir.
- SQL tek başına bir programlama dili olmadığı için ayrıca bir programlama diline (Teknolojisine) de ihtiyaç duyulmaktadır.



SQL NEDİR ?

- 1983 'lü yıllarda IBM laboratuvarlarında yapılan çalışmalarda SQL (Structural Query Language) standartları tanımlanmıştır.
- 1987 de ISO ardından da ANSI tarafından bir standart olarak kabul edilmiştir.



SQL NEDİR ?

- Daha sonra, bu standartlar çerçevesinde bir çok veritabanı yönetim sistemleri geliştirilmiştir.
- Bunlardan belli başlıları, Oracle, Sybase, MS SQL Server, Informix ve MySQL 'dir.



SQL NEDİR ?

- Bu VTYS 'lerin işlerin daha kolay yürümesi için kendi adlarına standart dilden uzaklaşan tarafları vardır.
- Ancak genel işlemlerde kullanılan dil tüm veri tabanı programları için de ortaktır ve bu SQL 'dir.
- VTYS'de saklanan veriler SQL komutları ile insanların istekleri çerçevesinde işler ve yeniden şekillendirilir.

SQL NEDİR ?

- Açılımı “Structured Query Language” yani “Yapısal Sorgulama Dili” olan SQL, veritabanı işlemleri ile ilgili komutlardan oluşan bir dildir.

```
Welcome to the WP MySQL Console.  
Connected to server version 5.0.41-community-nt. Commands end with ;  
Type 'help;' for help.  
mysql>
```

CONSOL ←

(MySQL)

SQL NEDİR ?

- Bu dil ile veri tabanı üzerinde;
 - veritabanının kendisini oluşturmak,
 - tablo,
 - indeks,
 - kullanıcı

oluşturmak gibi komutlar ve kayıt ekleme, silme, düzeltme gibi işlemler yapılabilir.

SQL NEDİR ?

- SQL dilindeki komutlar Pascal, C, Visual Basic, Delphi ve benzeri dillerdeki fonksiyon ve prosedür oluşturarak bir program yazmaktan biraz farklıdır.
- Yani kullanıcı SQL kullanırken fonksiyon ve prosedür yazamaz.

SQL NEDİR ?

- Yine SQL kullanımında şartlı ifadeler ve dallanmalar bulunmaz.
- Yani kullanıcı diğer programlama dillerindeki if, Case, next, do gibi ifadeler kullanamaz.
- Şartlı ifadeler, döngüler, karşılaştırmalar SQL'de bulunmaz.

SQL NEDİR ?

- SQL'de kullanılamayan procedür, fonksiyon, şartlı ifadeler, döngüler ve karşılaştırmaların eksikliğini giderebilmek için;
- Oracle PL/SQL (Programming Language/SQL), MS SQLServer ve Sybase 'de T-SQL dili geliştirilmiştir.

08118 Veri Tabanı I



... Devamı Haftaya