

---

# SQL

Structured Query Language  
Yapılandırılmış Sorgu Dili

11 Ocak 2021

DERS 01

SQL'E GIRIS

Mehmet Bulutluoz  
Elektronik muh.

TECHPROED

---

---

# BUGUNKU KONUMUZ

- 1) Database Nedir ?
- 2) Genel Database Kavramlari
- 3) SQL'e giris

TECHPROED

---

# DATABASE

**Kaydol** ×

Hızlı ve kolaydır.

Doğum Tarihi ?

Cinsiyet ?

☐ Kadın ☐ Erkek ☐ Özel

Kaydol düğmesine tıklayarak, Koşullarımızı, Veri İlkemizi ve Çerezler İlkemizi kabul etmiş olursun. Bizden SMS Bildirimleri alabilir ve bu bildirimleri istediğin zaman durdurabilirsin.

**Kaydol**

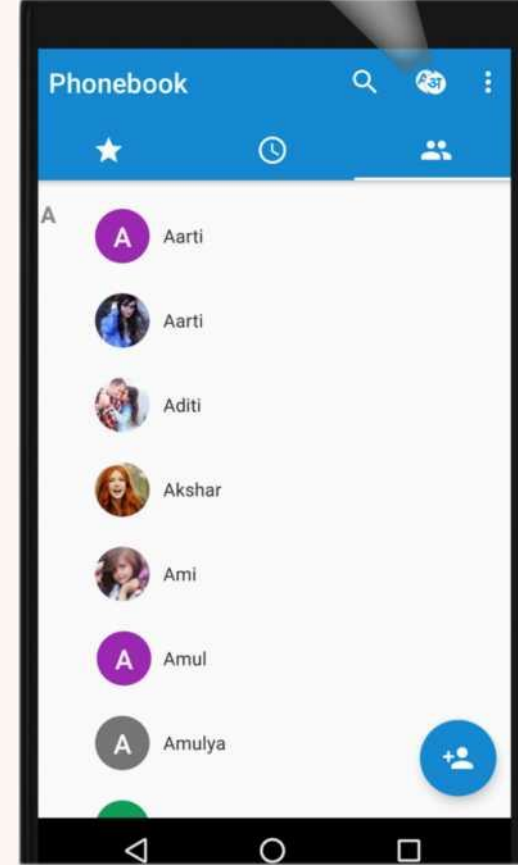
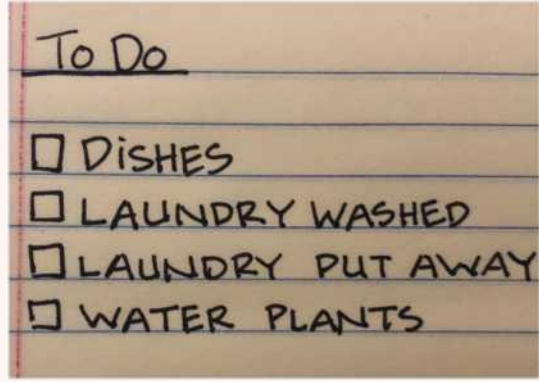
```
public class facebook {  
  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        System.out.println("Enter your name");  
        String name = scan.nextLine();  
  
        System.out.println("Enter your surname");  
        String surname = scan.nextLine();  
  
        System.out.println("Enter your surname");  
        String email = scan.nextLine();  
  
        System.out.println("Enter your pasword");  
        String password = scan.nextLine();  
  
        scan.close();  
    }  
}
```



TECHPROED



# DATABASE (VERITABANI) NEDİR?



Veritabanı genellikle elektronik olarak bir bilgisayar sisteminde depolanan yapılandırılmış(**Structured**) bilgi veya veriden oluşan düzenli bir koleksiyondur.

Veritabanı genellikle bir Veritabanı Yönetim Sistemi DBMS (**DataBaseManagementSystem**) ile kontrol edilir.

Çoğu veritabanında veri yazma ve sorgulama için yapılandırılmış sorgu dili SQL (**Structured Query Language**) kullanılır.

# DATABASE'IN FAYDALARI NELERDİR

- 1) Yüksek miktarda bilgi depolanabilir
- 2) Oluşturma, Okuma, Değiştirme ve Silme kolaylığı  
**Create, Read, Update, Delete (CRUD)**
- 3) Girişin kolay ve kontrollü olması
- 4) Dataya ulaşım kolaylığı
- 5) Güvenlik

ono	adi	soyadi	dyeri	bid
1	Ali	Turan	İstanbul	1
2	Ahmet	Büyük	Ankara	1
3	Leyla	Şahin	İzmir	1
4	Can	Türkoğlu	Manisa	2
5	Aziz	Keskin	İstanbul	2
6	Talat	Şanlı	İzmir	3
7	Kamuran	Kece	Adana	3
8	Turgut	Cemal	Bursa	4



TECH

CODE



# DATABASE VALIDATION(DOGRULAMA) TESTI

**Kaydol** ×

Hızlı ve kolaydır.

Doğum Tarihi ?

Cinsiyet ?  
☐ Kadın ☐ Erkek ☐ Özel

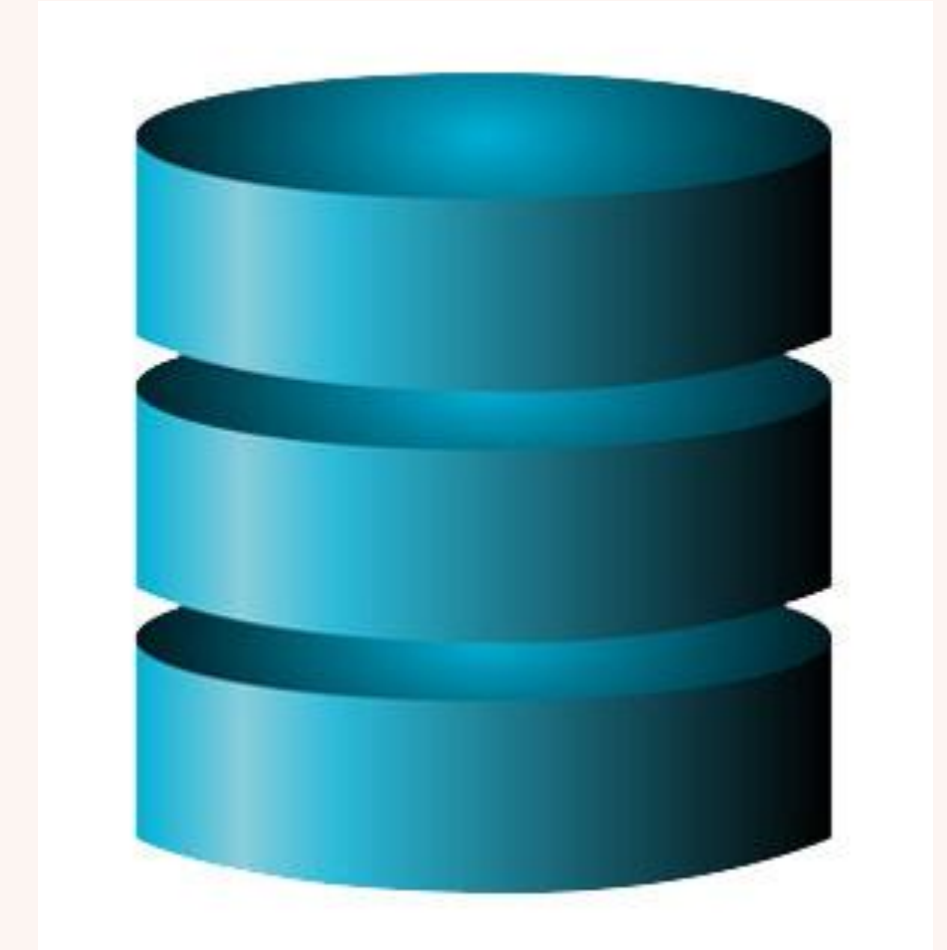
Kaydol düğmesine tıklayarak, Koşullarımızı, Veri İlkemizi ve Çerezler İlkemizi kabul etmiş olursun. Bizden SMS Bildirimleri alabilir ve bu bildirimleri istediğin zaman durdurabilirsin.

**Kaydol**

User Interface



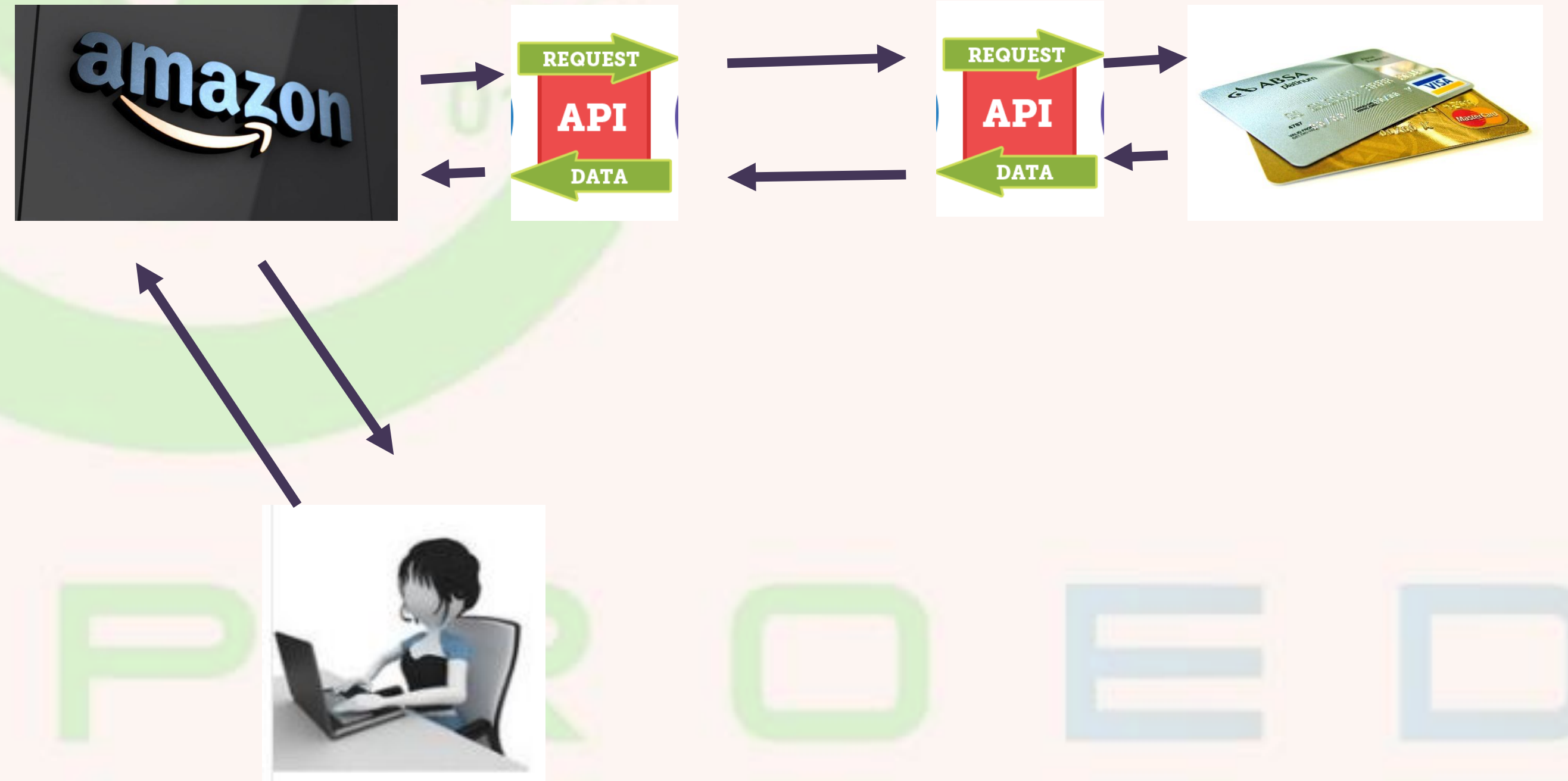
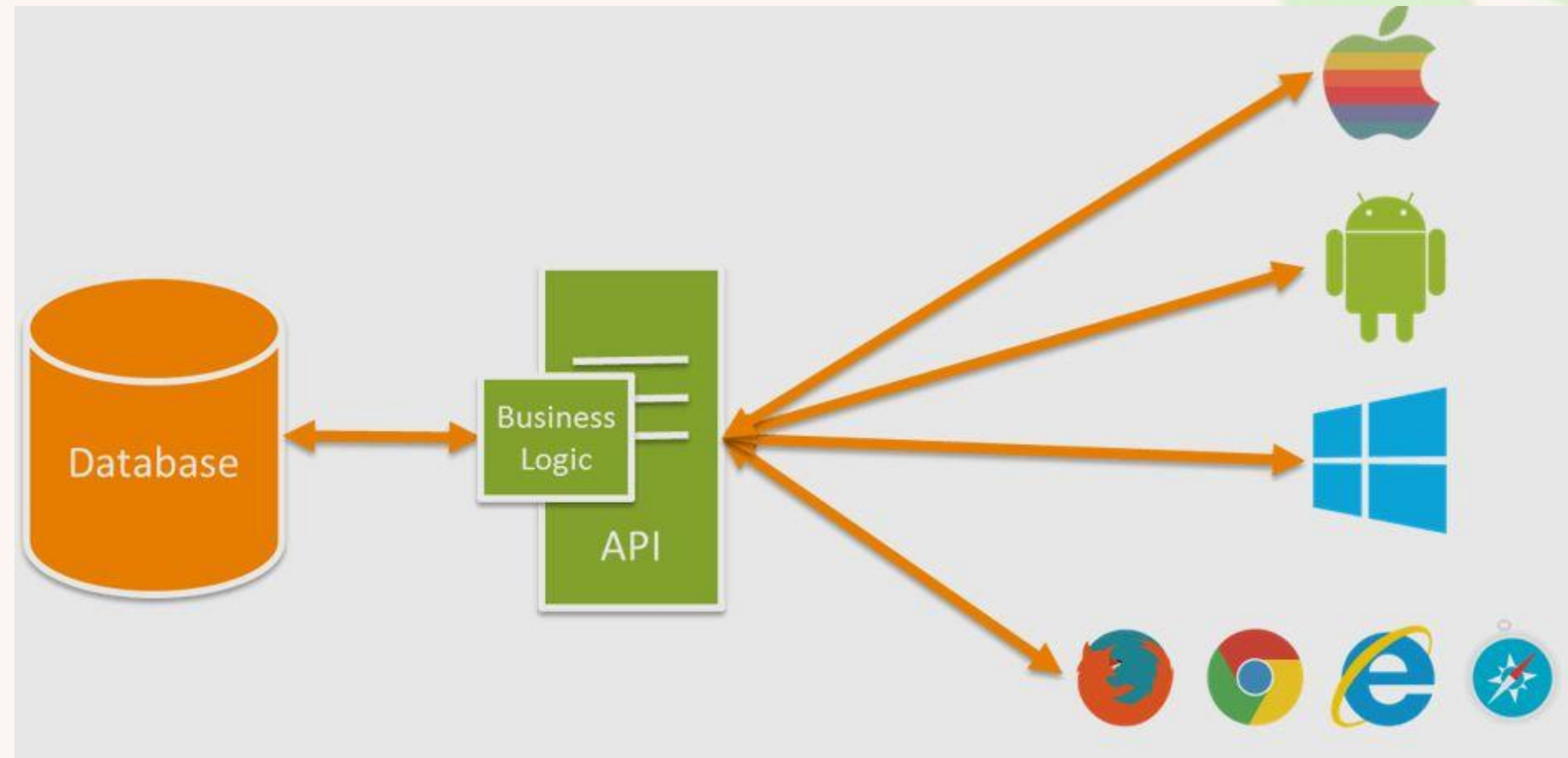
API



Database

# API

*Application Programming Interface*, bir uygulamaya ait yeteneklerin, başka bir uygulamada da kullanılabilmesi için, yeteneklerini paylaşan uygulamanın sağladığı arayüzdür.





---

## END To END (E2E) Testing

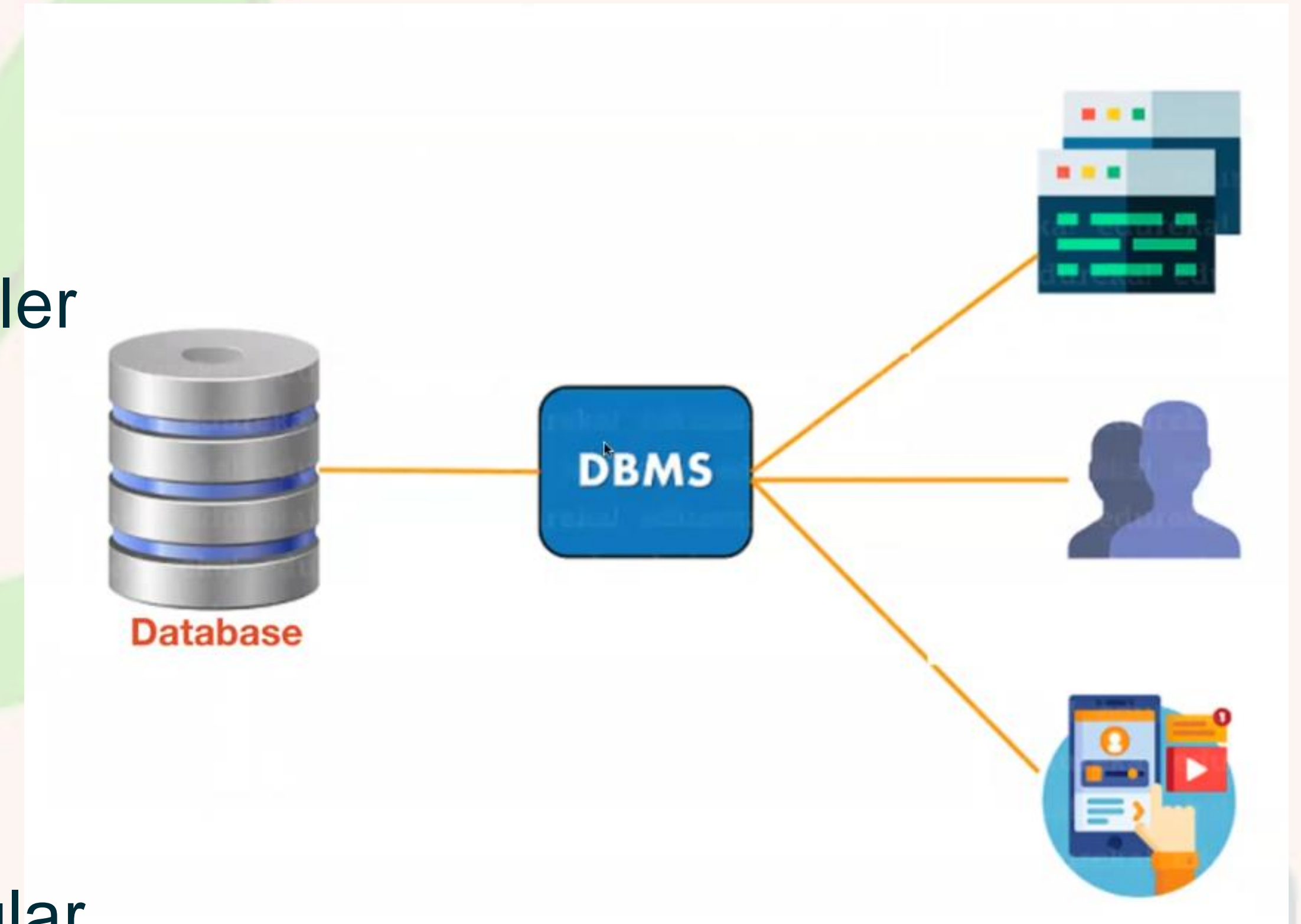
- 1 Eger datayi User Interface (UI) kullanarak yolladiysaniz
    - A) Datayi UI dan arama fonksiyonunu kullanarak dogrula (Selenium)
    - B) Datayi SQL kodlarini kullanarak dogrula (SQL + Selenium)
    - C) Datayi API kodlarini kullanarak dogrula (API + Selenium)
  - 2) Eger datayi SQL kodlarini kullanarak yolladiysaniz
    - A) Datayi UI dan arama fonksiyonunu kullanarak dogrula (Selenium)
    - B) Datayi SQL kodlarini kullanarak dogrula (SQL + Selenium)
    - C) Datayi API kodlarini kullanarak dogrula (API + Selenium)
  - 3) Eger datayi API kodlarini kullanarak yolladiysaniz
    - A) Datayi UI dan arama fonksiyonunu kullanarak dogrula (Selenium)
    - B) Datayi SQL kodlarini kullanarak dogrula (SQL + Selenium)
    - C) Datayi API kodlarini kullanarak dogrula (API + Selenium)
-



## Data Base Management System (DBMS)

Veritabanlarını yönetmek, kullanmak, geliştirmek ve bakımını yapmak için kullanılan yazılımlara denir.

- Database'e erişimi düzenler
- **C**reate, **R**ead, **U**ppdate, **D**eleete işlemlerini düzenler
- Data güvenliğini sağlar
- Formlar oluşturur ve formları işler,
- Sorgular oluşturur ve sorgular iletilir,
- Raporlar oluşturur ve raporları işletir,
- Uygulamayı kontrol eder
- Diğer uygulamalarla (Application) iletişimi sağlar.



# TABLULAR (TABLES)

Headers====>

Row (Record)====>

Row (Record)====>

Row (Record)====>

Row (Record)====>

contactID	name	company	email
1	Bill Gates	Microsoft	bill@XBoxOneRocks.com
2	Steve Jobs	Apple	steve@rememberNewton.com
3	Linus Torvalds	Linux Foundation	linus@gnuWho.org
4	Andy Harris	Wiley Press	andy@aharrisBooks.net

Column (Field) ^====

Column (Field) ^====

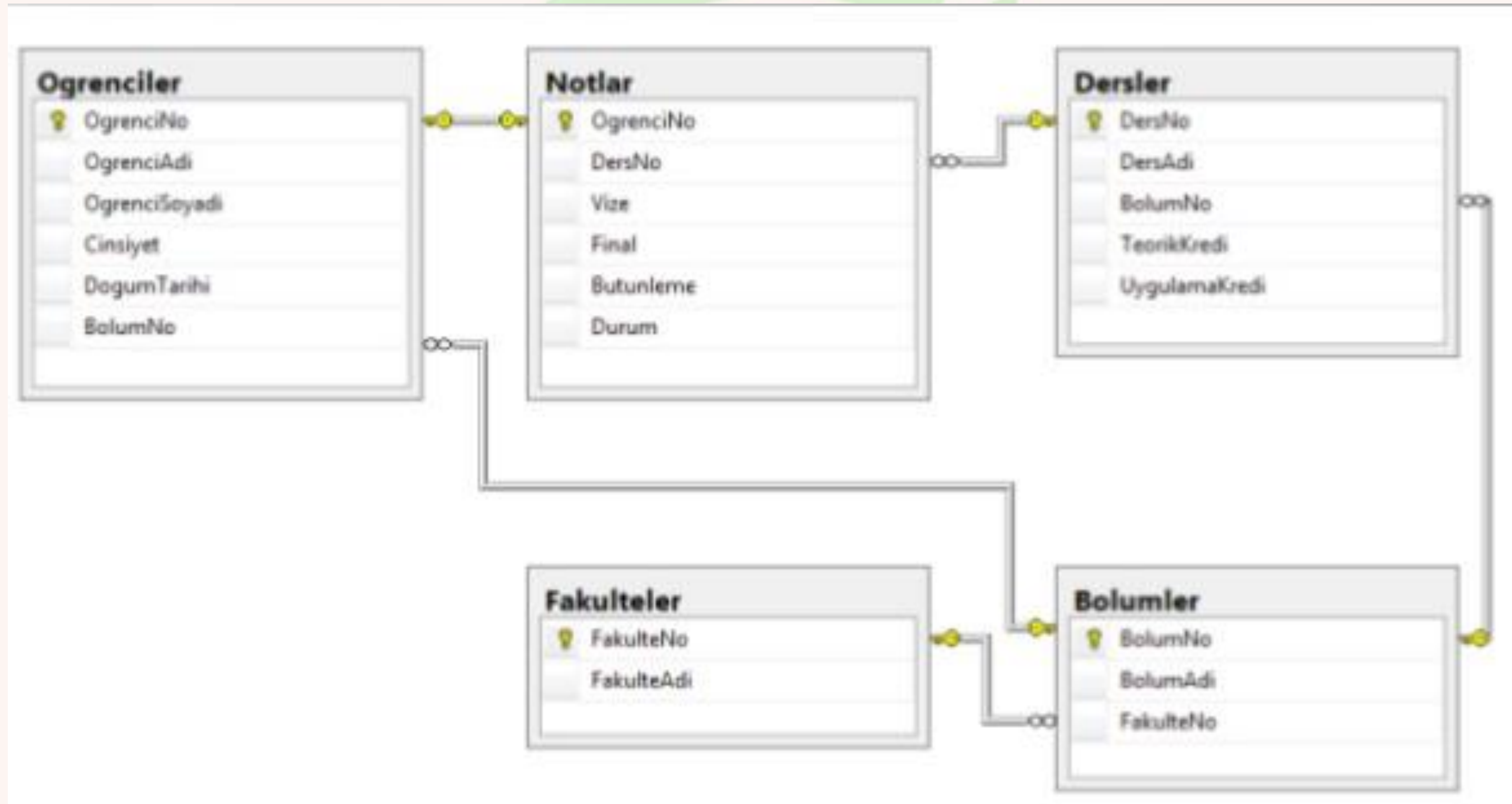
Column (Field) ^====

Column (Field) ^====

TECHPROED



# RELATIONANAL DATABASES (ILISKILI TABLOLAR)





# RELATIONAL DATABASES (İLİSKİLİ TABLOLAR)

➤ **SQL tablolar** dataları ilişkili tablolarda depolar.

➤ Tablolar arası ilişkiler net olmalıdır.

➤ Tablolar arası geçiş kolay olmalıdır

➤ Tabloların ve ilişkilerin butunüne SCHEMA denir

id	ogrenci_adi	ogrenci_soyadi
1	Elif	Türkmen
2	Ayşe	Sarı
3	Ender	Kaya
4	Ali	Demir
5	Adem	Salih

id	ogrenci_id	ders_id
1	1	3
2	1	5
3	2	1
4	3	4
5	4	2
6	4	3

id	ders_adi
1	Matematik
2	Tarih
3	Edebiyat
4	Yazılım
5	İstatistik

➤ Relational Databases, **SQL Databases** (Structured Query Language) olarak da adlandırılır

# Cok Kullanilan Relational Databases(SQL Database)



**SQL Server** : Microsoft tarafından geliştirilmiştir

**Negatif:** Pahalı – Kurumsal Kullanıcılar için binlerce dolar ödenmesi gereklidir

**Pozitif** : Zengin bir **user interface**'e sahip ve çok büyük verilerin kullanılmasında sorunsuz çalışır.



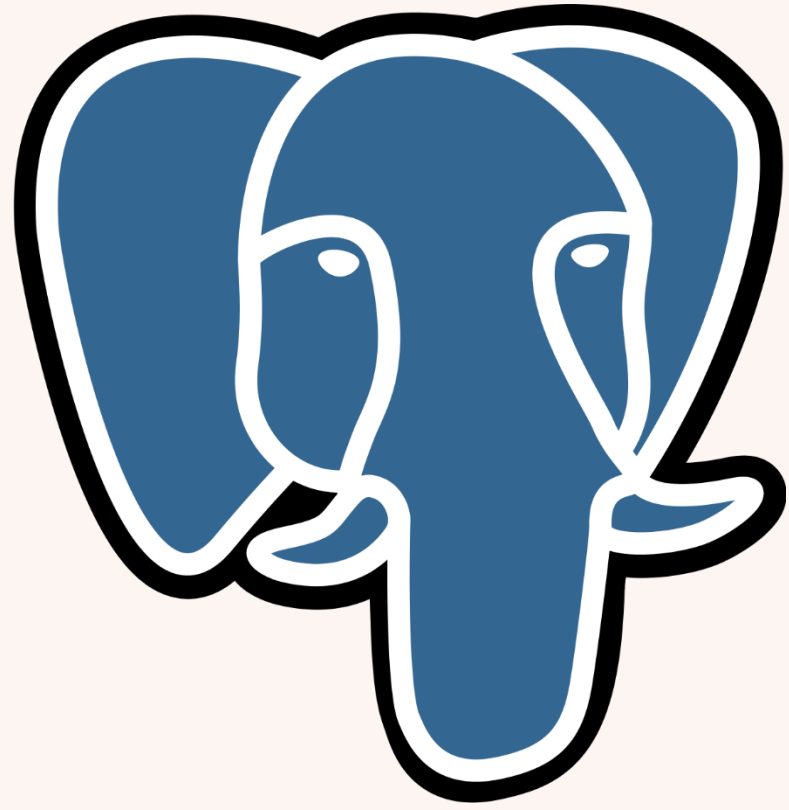
**MySQL Server** : İsveçli MySQL firması tarafından geliştirildi.  
2010'da Oracle satın aldı

**Negatif:** Eszamanlı çok fazla işlem girildiğinde çalışmayı durdurabilir.

**Pozitif:** Açık kaynak. Online destek ve ücretsiz çok fazla doküman var

TECHPROED

# Cok Kullanilan Relational Databases(SQL Database)



**PostgreSQL Server** : Created by a computer science professor Michael Stonebraker.

**Negatif**: Kurulum ve ayarlar zor. Yeni başlayanlar için kullanımı zor

**Pozitif**: Yeni nesil olarak ortaya çıktı. Kisiselleştirme mümkündür, zor görevler için ideal olabilir.

**ORACLE®**  
DATABASE



**PL/SQL** Oracle database sunucuları içinde depolanır

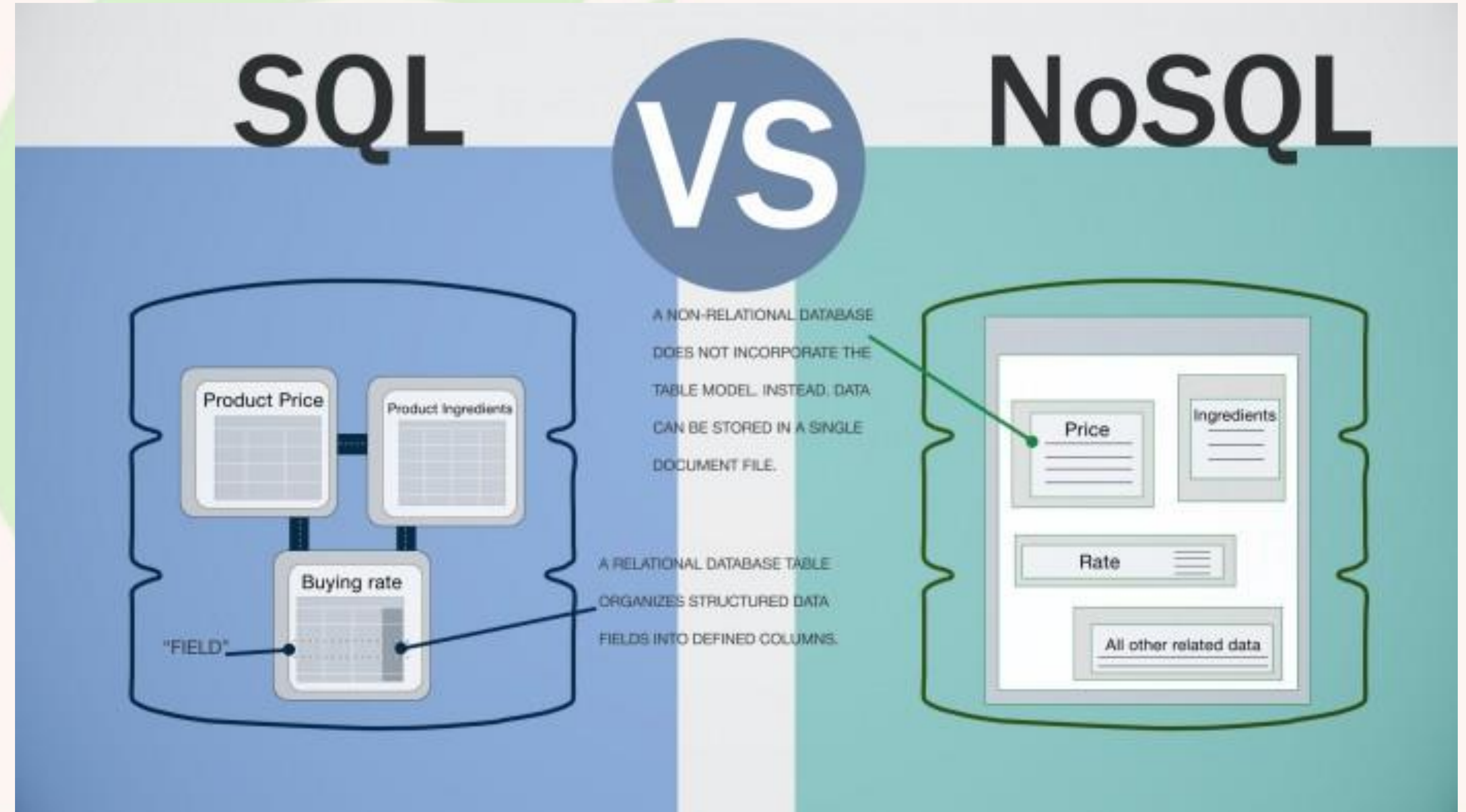
**PL/SQL** SQL komutlarını özellikle karşılamak üzere dizayn edilmiştir.

**Pros**: PL/SQL yüksek güvenlik seviyesi sağlar ve Object-Oriented Programming'e uyumludur



# Non Relational Databases(non-SQL Database)

**SQL** veritabanı verilerle ilgilenirken Yapısal Sorgu Dili kullanır. Veri yapısını belirlemek için önceden tanımlanmış şemalar gerektirir.



**NoSQL** veritabanı verilerle çalışırken Yapılandırılmamış Sorgu Dili kullanır.

---

# SQL Komutlari

## SQL komutlari 4 ana gruba ayrilir:

### 1. Veri Sorgulama Dili (Data Query Language - DQL)

DQL içindeki SELECT komutu ile veritabanında yer alan **mevcut kayıtların** bir kısmını veya tamamını tanımlanan koşullara bağlı olarak alır.

**SELECT** : Veritabanındaki verileri alır.

### 2. Veri Kullanma Dili (Data Manipulation Language - DML)

DML komutlari ile veritabanlarında bulunan verilere işlem yapılır. DML ile veritabanına yeni kayıt ekleme, mevcut kayıtları güncelleme ve silme işlemleri yapılır.

**INSERT** : Veritabanına yeni veri ekler.

**UPDATE** : Veritabanındaki verileri günceller.

**DELETE** : Veritabanındaki verileri siler.

---

---

# SQL Komutlari

## 3. Veri Tanımlama Dili (Data Definition Language - DDL)

DDL komutları ile veritabanı ve tabloları oluşturma, değiştirme ve silme işlemleri yapılır:

**CREATE** : Bir veritabanı veya veritabanı içinde tablo oluşturur.

**ALTER** : Bir veritabanı veya veritabanı içindeki tabloyu günceller.

**DROP** : Bir veritabanını veya veritabanı içindeki tabloyu siler.

## 4. Veri Kontrol Dili (Data Control Language - DCL)

DCL komutları ile kullanıcılara veritabanı ve tablolar için yetki verilir veya geri alınır:

**GRANT** : Bir kullanıcıya yetki vermek için kullanılır.

**REVOKE** : Bir kullanıcıya verilen yetkiyi geri almak için kullanılır.

---



# Primary Key

**Primary Key** (birincil anahtar), bir veri tablosunda yer alan her satır için bir vekil / tanımlayıcı (identify) görevi görür, kısıtlamadır (constraint) ve eşsizdir (Unique).

Primary Keys

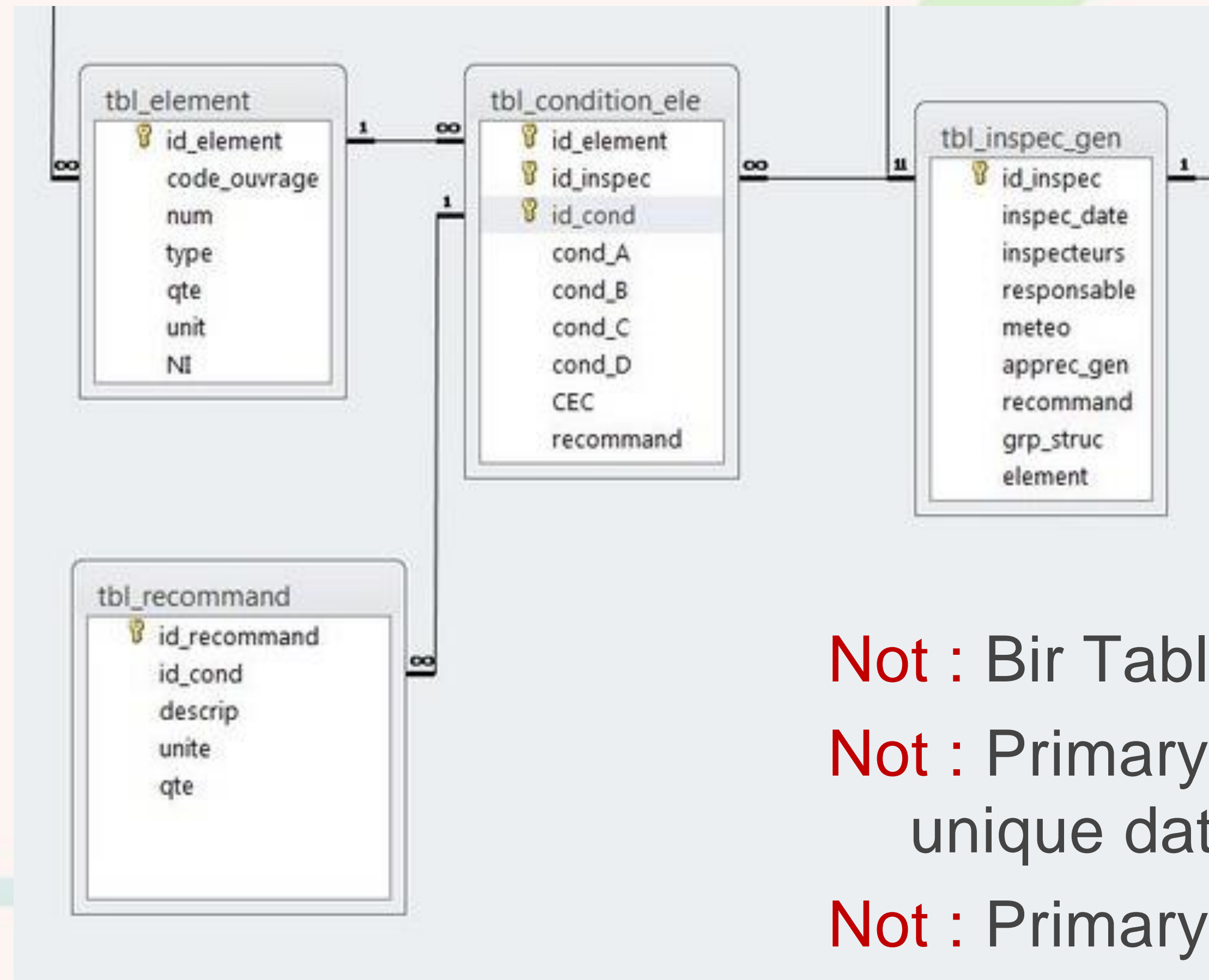


<u>StudentId</u>	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042
L0023487	Peter	Murray	P301
L0018453	Anne	Norris	S042

Satırlara ait değerlerin karışmaması adına bu alana ait bilginin tekrarlanmaması gerekir.

Çoğunlukla tek bir alan (id, user\_id, e\_mail, username, national\_identification\_number vb.) olarak kullanılsa da birden fazla alanın birleşimiyle de oluşturulabilir

# Primary Key



Primary Key değeri boş geçilemez ve NULL değeri alamaz.

Relational veri tabanlarında (relational database management system) mutlaka birincil anahtar olmalıdır.

**Not :** Bir Tabloda en fazla 1 tane primary Key olabilir.

**Not :** Primary Key benzersiz (Unique) olmalıdır ama her unique data Primary Key değildir

**Not :** Primary key her türlü datayı içerebilir. Sayı, String..

**Not :** Her tabloda Primary Key olması zorunlu değildir



# Primary Key

Primary Key, dış dünyadaki gerçek verileri temsil ediyorsa, orneğin; TC kimlik numarası, bir kitabın ISBN numarası, bir ürünün ismi, email hesabi gibi buna **Natural key** denir

StudentID	FirstName	LastName
10 ←	John	Walker
11	Tom	Hanks
12	Kevin	Star
13 ←	Carl	Wall
14	Andrei	Apazniak
15	Mark	High
16	Clara	Star
17	John	Ocean
18 ←	John	Walker
19	Pamela	Star
20 ←	Carl	Wall

Genel olarak kayıt eklenmeden önce üretilen sıra numarası gibi sayısal değerlere **Surrogate Key** denir

Email	FirstName	LastName
JWalker@gmail.com	John	Walker
THanks@gmail.com	Tom	Hanks
KStar@gmail.com	Kevin	Star
CWall@gmail.com	Carl	Wall
AApazniak@gmail.com	Andrei	Apazniak
MHigh@gmail.com	Mark	High
CStar@gmail.com	Clara	Star
JOcean@gmail.com	John	Ocean
JWalker01@gmail.com	John	Walker
PStar@gmail.com	Pamela	Star
CWall01@gmail.com	Carl	Wall



# Foreign Key

**Foreign Key** iki tablo arasında relation oluşturmak için kullanılır

**Foreign Key** başka bir tablodaki Primary Key ile ilişkilendirilmiş olmalıdır

**Foreign Key**

StudentID	FirstName	LastName	CourseID
10	John	Walker	200
11	Tom	Hanks	400
12	Kevin	Star	400
13	Carl	Wall	200
14	Andrei	Apazniak	300
15	Mark	High	400
16	Clara	Star	100
17	John	Ocean	100
18	John	Walker	200
19	Pamela	Star	300
20	Carl	Wall	NULL

**Child Table**

**Primary Key**

CourseID	CourseName	CourseCredit	CourseFee
100	Biology	3	1200
200	Math	3	1200
300	English	2	600
400	Selective	1	200

**Parent Table**

Bir Tabloda birden fazla Foreign Key olabilir

Foreign Key **NULL** değeri kabul eder

Foreign Key olarak tanımlanan field'da **tekrarlar** olabilir

**Foreign Key**, değerleri farklı bir tablodaki Primary Key ile eşleşen bir sütun veya sütunların birleşimidir.

# Foreign and Primary Key

**Note:** Foreign key Tablonun kendi icinde bir relation olusturabilir.

Emp_ID	first_name	last_name	birth_date	Gender	salary	Job_ID	Manager_ID
100	Jan	Levinson	1961-05-11	F	110,000	1	NULL
101	Michael	Scott	1964-03-15	M	75,000	2	100
102	Josh	Porter	1969-09-05	M	78,000	3	100
103	Angela	Martin	1971-06-25	F	63,000	2	101
104	Andy	Bernard	1973-07-22	M	65,000	3	101

Job_ID	Job_Name
2	SDET
3	Manual Tester
1	QE Lead

- 1) Michael Scott'un yoneticisi kimdir?
- 2) Angela Martin'in Job\_Name'i nedir ?
- 3) Manual Tester'lerin ortalama Salary'si ne kadardir ?
- 4) En yuksek Salary'yi alan kisinin Job\_Name'i nedir?

# SQL Composite Key

Job_ID	Job_Name
2	SDET
3	Manual Tester
1	QE Lead

Job Table

Recruiter	NumberOfClient
Mark Eye	121
John Ted	283
Cory AI	67
Angela Star	301

Recruiter Table

JobJD	Recruiter 1	
2	Mark Eye	RCG
2	John Ted	RCG
1	Mark Eye	Signature
1	John Ted	Info Log
1	Cory AI	Info Log
2	Angela Star	Signature

Company Table

**Composite Key** birden fazla field(kolon)'in kombinasyonu ile oluşturulur.

Tek basına bir kolon **Primary Key** olma özelliklerini taşıyamıyorsa, bu özellikleri elde etmek için birden fazla kolon birleştirilerek Primary oluşturulur



---

## “UNIQUE KEY” & “PRIMARY KEY”

“UNIQUE KEY” ve “PRIMARY KEY” arasındaki farklar

### Primary Key

Bir Tabloda sadece 1 tane olur  
NULL değer Kabul etmez

### Unique Key

Bir tabloda birden fazla olabilir  
NULL değeri Kabul eder

“UNIQUE KEY” ve “PRIMARY KEY” ortak özellikleri

Dublication(Cift Kullanım)’a izin vermez

TECHPROED

---

# Ornek Okul Tablosunun Bir Parçasi

sinif tablosu		
sinif id	sinif	sube adi
9a	9 a	
9b	9 b	
9c	9 c	
9d	9 d	
10a	10 a	
10b	10 b	
10c	10 c	

ders tablosu	
ders id	ders adi
k10	10.sinif kimya
k11	11.sinif kimya
k12	12.sinif kimya
b10	10.sinif biyoloji
k9	9.sinif kimya
b9	9.sinif biyoloji

ogrenci tablosu				
ogrenci no	adi	soyadi	giris yili	sinif id
111	ali	velioglu	2020	9a
112	ayse	atakul	2018	9a
113	hasan	delioglan	2019	9a
114	hulya	kar	2019	9b
115	ali	yasa	2019	9b
116	ayse	atakul	2020	9b
117	kemal	velioglu	2018	10a
118	hatice	gulsen	2019	10b
119	hasan	delioglan	2019	10c
120	kemal	kar	2018	10c

ogretmen tablosu			
adi	soyadi	ders	ogr id
ahmet	baba	kimya	k101
mehmet	kilim	fizik	f102
ayse	gulcu	tarih	t101
ayse	gulmez	biyoloji	b102
kemal	yasa	biyoloji	b105
fatma	yasa	kimya	k103

ogrenci sahsi bilgileri					
ogrenci no	tel	boyu	kilosu	saglik raporu	fotografi
111	12124435	160	50	var	var

veli bilgileri						
ogrenci no	veli adi	veli soy	veli yak.	veli tel	veli tel 2	adres
111	hasan	velioglu	babasi	64654613	31646	

yazili tablosu			
ogrenci no	ders	ogretmen	not
111	k9	k101	85
112	b9	b102	80
116	b9	b105	65
118	k10	k103	90

# Related Tablolarla Calisma

## One to One Relation

StudentID	FirstName	LastName
10	John	Walker
11	Tom	Hanks
12	Kevin	Star
13	Carl	Wall
14	Andrei	Apazniak
15	Mark	High
16	Clara	Star
17	John	Ocean
18	John	Walker
19	Pamela	Star
20	Carl	Wall

StudentID	Street	ZipCode	City	State
10	1234 W 23th Street	33018	Hialeah	Florida
11	1235 N 3th Street	22145	Austwell	Texas
12	1236 SE 12th Street	54234	Orange	California
13	1237 N 5th Street	33018	Hialeah	Florida
14	1238 SW 53th Street	33026	Miami	Florida
15	1239 S 123th Street	22314	Avery	Texas
16	1240 N 1 st Street	12345	Arlington	Virginia
17	1241 NW 2nd Street	65432	Pittsburgh	Pensylvania
18	1242 W 5th Street	22133	Baytown	Texas
19	1243 SE 55th Street	74352	Beachwood	Ohio
20	1244 SW 17th Street	22314	Avery	Texas

- 1) Tom Hanks'in adresi nedir?
- 2) John Walker'in eyaleti nedir?
- 3) ID'si 17 olan kisinin sehri nedir?



# Related Tablolarla Calisma

## One to Many Relation

CourseID	CourseName	CourseCredit	CourseFee	InstructorID
100	Biology	3	1200	1
200	Math	3	1200	2
300	English	2	600	3
400	Selective	1	200	1

StudentID	FirstName	LastName	CourseID
10	John	Walker	200
11	Tom	Hanks	400
12	Kevin	Star	400
13	Carl	Wall	200
14	Andrei	Apazniak	300
15	Mark	High	400
16	Clara	Star	100
17	John	Ocean	100
18	John	Walker	200
19	Pamela	Star	300
20	Carl	Wall	400

- 1) Biology dersi alan ogrenciler kimler?
- 2) Selective ders alan ogrencilerin isimleri ?
- 3) CourseFee 600 olan ogrencilerin isimleri ?

# Related Tablolarla Calisma

## Many to Many Relation

StudentID	FirstName	LastName
10	John	Walker
11	Tom	Hanks
12	Kevin	Star
13	Carl	Wall
14	Andrei	Apazniak
15	Mark	High
16	Clara	Star
17	John	Ocean
18	John	Walker
19	Pamela	Star
20	Carl	Wall

StudentID	InstructorID
12	1
11	2
12	2
13	1
15	1
17	3
15	4

InstructorID	FirstName	LastName	Phone	Department
1	Mark	Adam	1234567891	Science
2	Eve	Sky	1239876543	Engineering
3	Leo	Ocean	1237845691	Language
4	Andy	Mark	1232134567	Health

- 1) Ogretmeni Mark Adam olan ogrencilerin isimleri nedir?
- 2) Kevin Star'in ogretmenlerinin isimleri nedir?
- 3) Pamela Star'in ogretmenlerinin isimleri nedir?

# SQL Data Types

## String Data Types

Data Type	Aciklama
<b>char(size)</b>	Maximum boyutu <b>2000 byte</b> olur. 1 karakter 1 byte kullanir. “ <b>size</b> ” database’e eklenecek karakter sayisidir. “char” data tipinden <b>uzunlugu sabit</b> datalari depolar. (Strings) “char” SSN, zip kodu gibi uzunlugu sabit datalari depolamak icin idealdir.
<b>nchar(size)</b>	Maximum boyutu <b>2000 byte</b> olur. 1 karakter 2 byte kullanir “ <b>size</b> ” depolanacak <b>karakter sayisi</b> ’dir. “ <b>nchar</b> ” <b>Unicode</b> datalari depolamak icin kullanilir. Genellikle <b>farkli dillerdeki karakterler</b> icin kullanilir <b>Uzunlugu belli</b> Stringler icin kullanilir.
<b>varchar2(size)</b>	Maximum boyutu <b>4000 byte</b> olur. 1 karakter 1 byte kullanir. “ <b>size</b> ” database’e eklenecek <b>max.</b> karakter sayisidir. <b>Degisken uzunluktaki</b> stringler icin kullanilir.
<b>nvarchar2(size)</b>	Maximum boyutu <b>8000 byte</b> olur. 1 karakter 2 byte kullanir “ <b>size</b> ” depolanacak <b>karakter sayisi</b> ’dir. <b>Degisken uzunluktaki</b> stringlerin Unicode degerleri icin kullanilir.

Alphabets	CHAR ( 4 )	Data Size	VARCHAR ( 4 )	Data Size
' '	' '	4 bytes	' '	1 byte
'ab'	'ab '	4 bytes	'ab'	3 bytes
'abcd'	'abcd'	4 bytes	'abcd'	5 bytes
'abcdefgh'	'abcd'	4 bytes	'abcd'	5 bytes



# SQL Data Types

## Numeric Data Types

Data Type	Aciklama
<b>number(p, s)</b>	<p>“<b>Precision</b>” (p) sayidaki rakam sayisidir “<b>Scale</b>” (s) virgulden sonar kac rakam oldugunu belirler Ornegin: <b>1234,56</b> ==&gt; <b>Precision : 6, Scale : 2.</b></p> <p>Precision ( <b>p</b> ) can range from <b>1 to 38</b> Scale ( <b>s</b> ) can range from <b>-84 to 127</b></p> <p>1) “<b>number(5, 2)</b>” virgulden once 3,virgulden sonra 2 rakam olan sayi ==&gt; <b>123,45</b></p> <p>2) “<b>number(4, 2)</b>” ==&gt; <b>123,45</b> ==&gt; <b>error verir</b></p> <p>3) “<b>number(7)</b>” ondalik kısmi olmayan 7 basamakli sayi demektir ==&gt; <b>12345,67’i kabul eder ama 12345 olarak depolar</b> <b>Note:</b> “<b>number(7)</b>” ve “<b>number(7, 0)</b>” ayni seydir</p> <p>4) “<b>number(7, -2)</b>” rounds the numeric value to hundreds. ==&gt; <b>1234567,89</b> ==&gt; <b>1234600</b></p>

---

# SQL Data Types

## Numeric Data Types

### DBMS Numeric Types:

<i>DBMS and version</i>	<i>Types</i>
MySQL 5.7	INTEGER(TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT, INTEGER) FIXED-POINT(DECIMAL, NUMERIC) FLOATING-POINT(FLOAT, DOUBLE) BIT-VALUE(BIT),
PostgreSQL 9.5.3	SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE PRECISION, SMALLSERIAL, SERIAL, BIGSERIAL
SQL Server 2014	EXACT NUMERICS(BIGINT, BIT, DECIMAL, INT, MONEY, NUMERIC, SMALLINT, SMALLMONEY, TINYINT) APPROXIMATE NUMERICS(FLOAT, REAL )
Oracle 11g	NUMBER FLOATING-POINT(BINARY_FLOAT, BINARY_DOUBLE)

TECHPRO

---

# SQL Data Types

## Date Data Types

<i>Data Type</i>	<i>Aciklama</i>
<b>DATE</b>	<p>“<b>DATE</b>” data tipi tarih ve zamani depolamak icin kullanilir. Saniyenin virgullu kismini da alir.</p> <p>“<b>DATE</b>” <b>yil</b>, ay, gun, saat, dakika, ve saniye icerir.</p> <p>Standart “<b>Date Format</b>” , “<b>dd - MMM - yy</b>”. Ornegin 13 - Apr - 20</p> <p>Tarih formatini “<b>ALTER SESSION SET NLS_DATE_FORMAT = “YYYY-MM-DD</b>” kodu kullanilarak degistirilebilir. Koddan sonra tarih 2020 - 04 – 13 olur.</p>



---

# SQL Data Types

## BLOB Data Types

<i>Data Type</i>	<i>Aciklama</i>
<b>BLOB</b>	<p>“<b>BLOB</b>” , “<b>B</b>inary <b>L</b>arge <b>OB</b>jects” demektir</p> <p>“<b>BLOB</b>” resim,video,ses gibi datalari binary formatina cevирerek depolar.</p>

TECHPROED

---

# SQL Komutlari

4. Veri Kontrol Dili (Data Control Language - DCL)  
veritabanı ve tablolar için yetki verilir  
veya geri alınır

**GRANT** : Bir kullanıcıya yetki  
vermek için kullanılır.

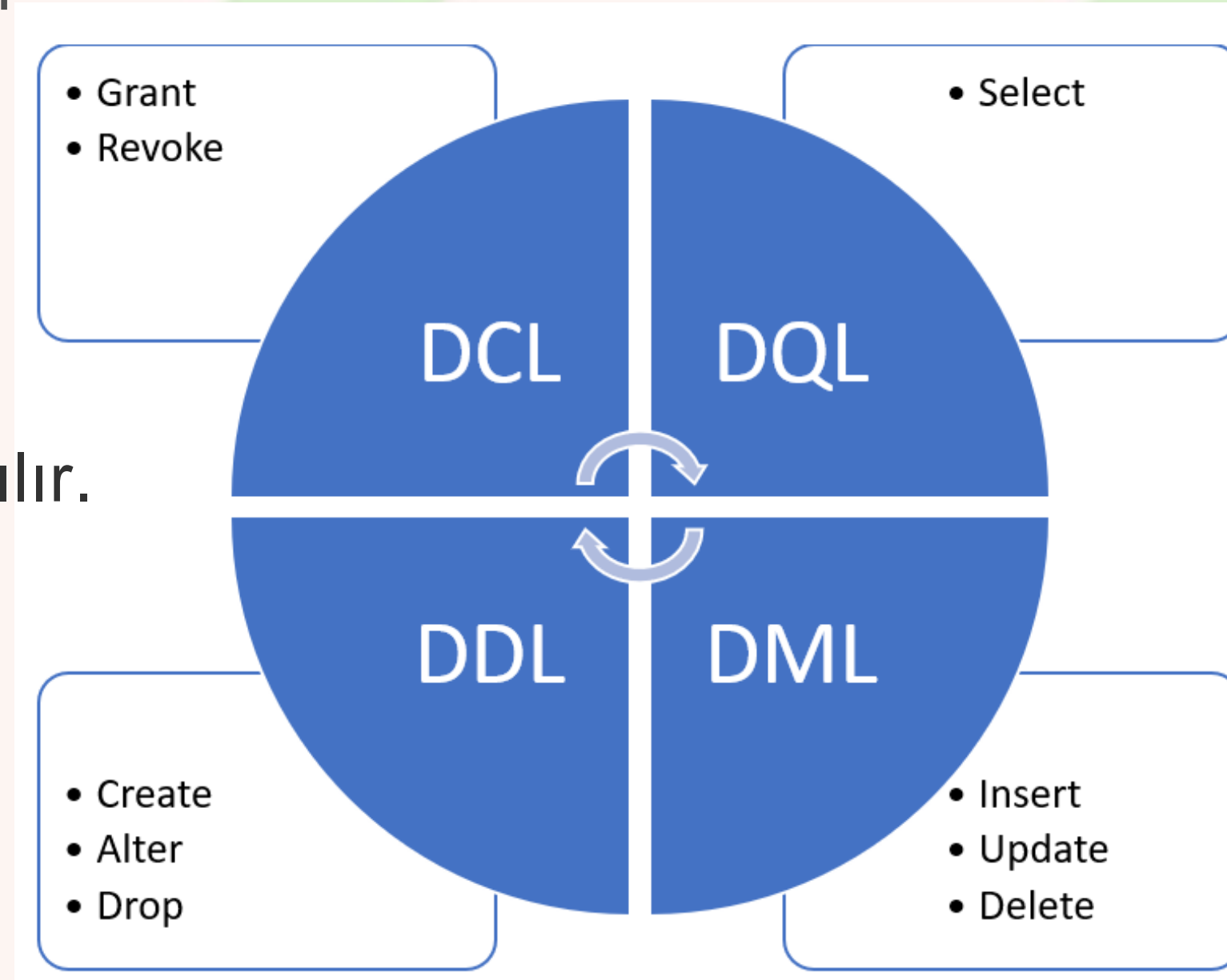
**REVOKE** : Bir kullanıcıya verilen  
yetkiyi geri almak için kullanılır.

3. Veri Tanımlama Dili  
(Data Definition Language - DDL)  
veritabanı ve tabloları oluşturma,  
değiştirme ve silme işlemleri yapılır

**CREATE** : Bir veritabanı veya tablo oluşturur.

**ALTER** : Bir veritabanı veya tabloyu günceller.

**DROP** : Bir veritabanını veya tabloyu siler.



1. Veri Sorgulama Dili (Data Query Language - DQL)  
**mevcut** kayıtların bir kısmını veya tamamını  
tanımlanan koşullara bağlı olarak alır.

**SELECT** : Veritabanındaki verileri alır.

2. Veri Degistirme Dili (Data Manipulation  
Language - DML)  
veritabanına yeni kayıt ekleme, mevcut  
kayıtları güncelleme ve silme işlemleri  
yapılır.

**INSERT** : Veritabanına yeni veri ekler.

**UPDATE** : Veritabanındaki verileri günceller.

**DELETE** : Veritabanındaki verileri siler.

---

# Table Nasıl Olusturulur?

## 1) Create from Scratch

```
CREATE TABLE ogrenciler  
(  
id char(11),  
isim varchar2(50),  
not_ort number(3),  
adres varchar2(100),  
kayit_tarihi date  
);
```

## 2) Var olan tablodan yeni tablo olusturmak

```
CREATE TABLE ogrenci_ortalamalar  
AS SELECT isim, not_ort  
FROM ogrenciler;
```

TECHPROED

---