



BATCH : 146 - 149

LESSON : SQL

DATE : 24.07.2023

SUBJECT : SQL - 1

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ



PostgreSQL

1. Ders

24.07.2023

B149 AWS & DevOps

B146 Cyber Security



Bugün ne yapıyoruz?

- ♦ Veritabanı Temel Kavramlar
- ♦ Veri Türleri ve Sınırlamaları
- ♦ PostgreSQL' de DB ve Tablo Oluşturma



PostgreSQL

SQL Nedir?

Structred

Query

Language

- ♦ **Y**apısal **S**orgulama **D**ili



Veritabanı (Database)

- ♦ **Data:** İşlenmemiş ham bilgilerdir. Database'ler belirli bir kurala göre bir araya getirilmiş verilerin kalıcı olarak (**persist**) depolanmasını sağlayan sistemlerdir.

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.

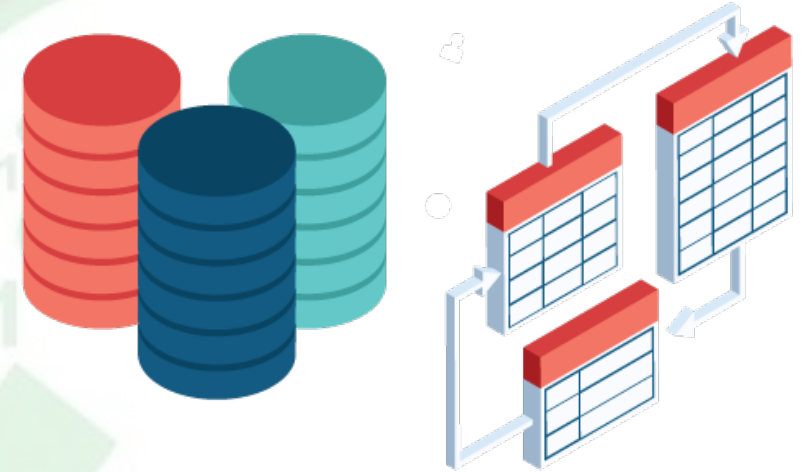
Kaydol

```
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();  
    }  
}
```



Database

- ♦ 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.

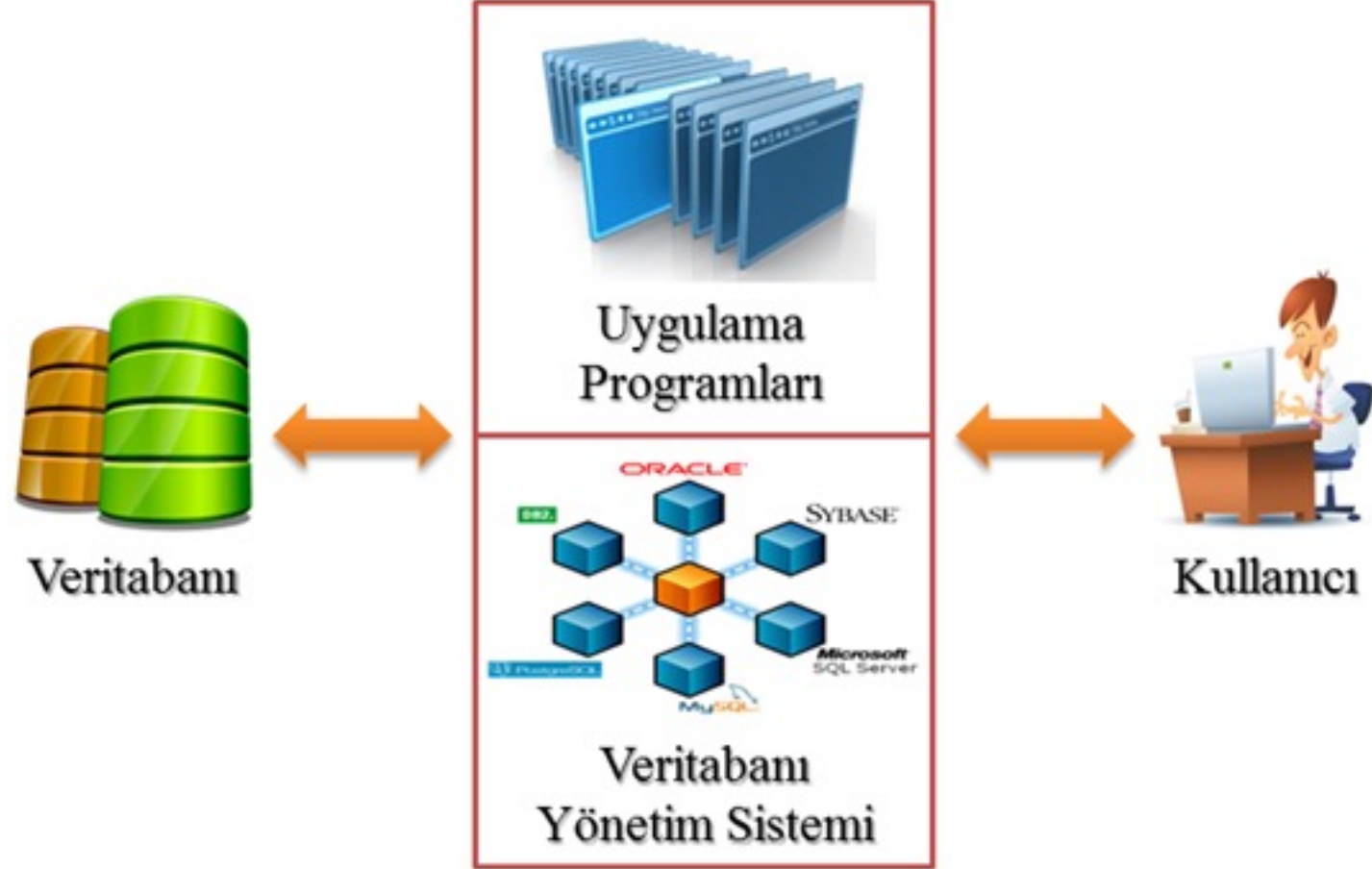


Database Faydaları Nelerdir?

- Yüksek miktarda bilgi depolanabilir
 - Oluşturma, Okuma, Değiştirme ve Silme kolaylığı
Create, Read, Update, Delete (CRUD)
- Veri girişin kolay ve kontrollü olması
- Dataya ulaşım kolaylığı
- Güvenlik

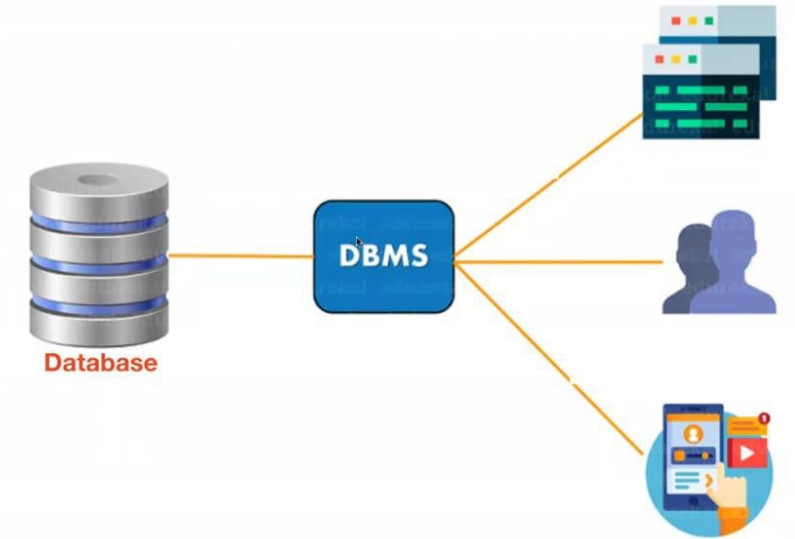


Veritabanı Yönetim Sistemi



DataBase Management System (DBMS)

- Veritabanlarını yönetmek, kullanmak, geliştirmek ve bakımını yapmak için kullanılan yazılımlardır.
- Database' e erişimi düzenler.
- Create, Read, Update, Delete işlemlerini düzenler.
- Data güvenliğini sağlar.
- Formlar oluşturur ve formları işler.



Tablo (Table)

Headers====>

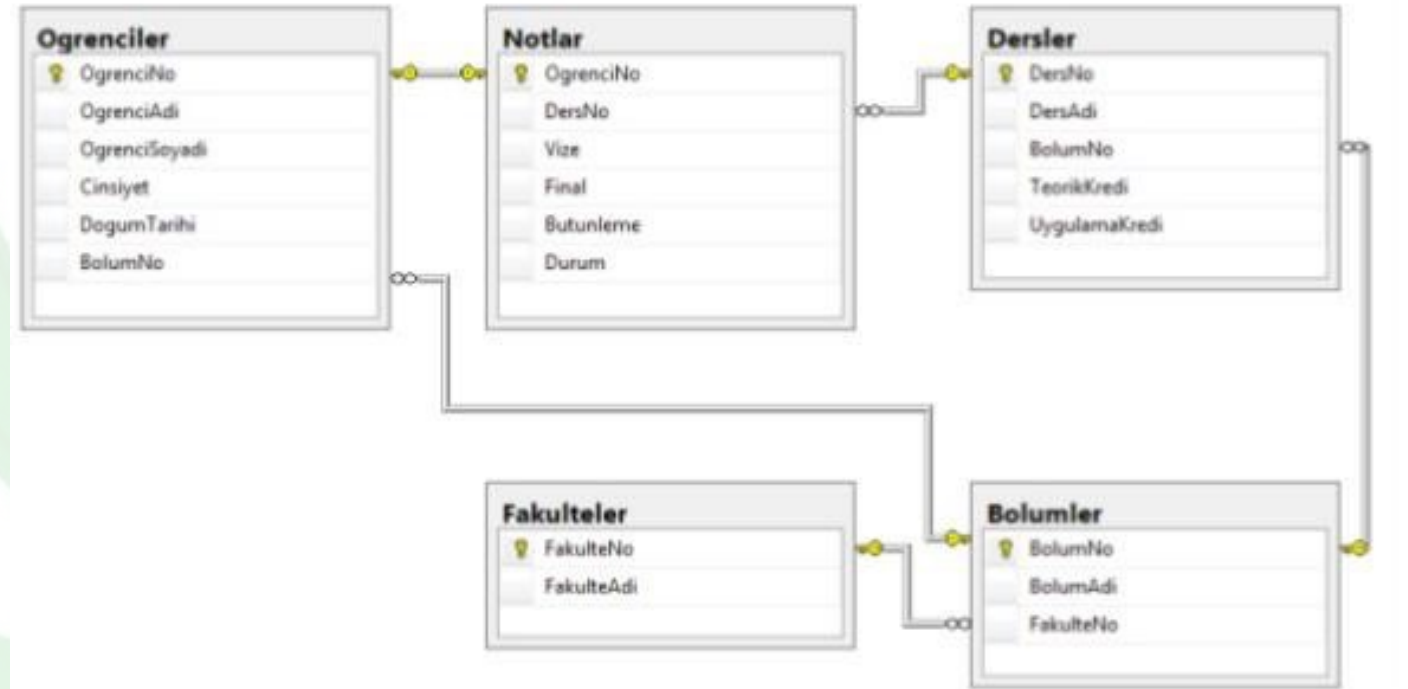
Row (Record)====>

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)====>

İlişkisel DB Yönetim Sistemi (RDBMS)

- Tekrar eden verileri tekilleştirmek amacıyla yapılandırılan veritabanı sistemleridir.
- **Örneğin;** sürekli gelen müşterilerin bilgilerinin kayıt altına alınması
 - İş gücü kaybı
 - Kullanıcı hataları sorunu



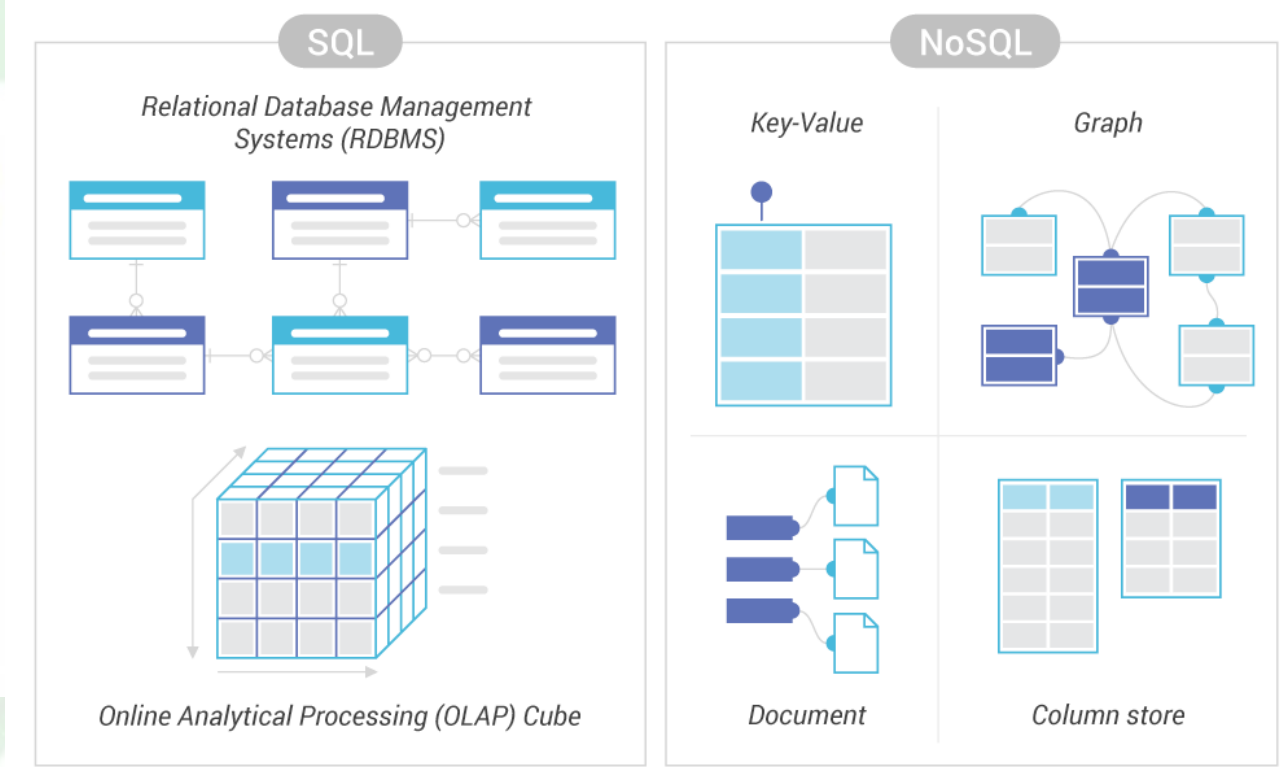
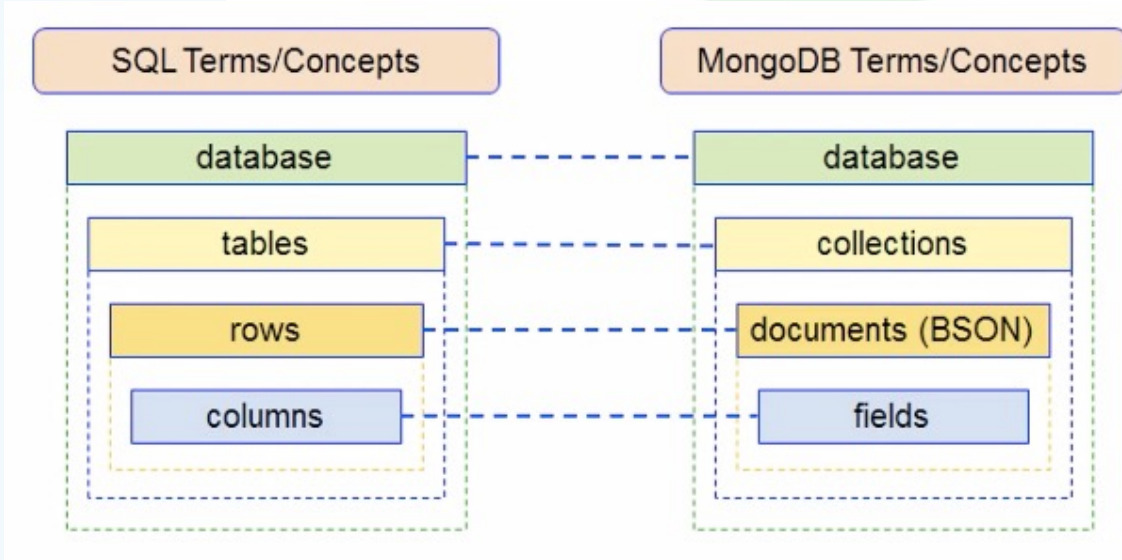
İlişkisel DB Yönetim Sistemi (RDBMS)

- SQL tablolar dataları ilişkili tablolarda depolar.
- Tablolar arası ilişkiler net olmalıdır.
- Tablolar arası geçiş kolay olmalıdır.
- Tabloların ve ilişkilerin bütününe **SCHEMA** denir.



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

SQL vs NoSQL



- SQL veritabanları yapılandırılmış veri modeli ve şema tabanlı yaklaşım sunarken,
- NoSQL veritabanları daha esnek veri modelleri ve şema özgürlüğü sunarak, büyük ölçekli ve dağıtık uygulamalara uygun alternatifler sunar.
- Veritabanı seçimi, uygulamanın gereksinimlerine, veri yapısına ve ölçeklenebilirlik ihtiyacına bağlı olarak belirlenir.

Summary

SQL Komutları

- ♦ 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.
- ♦ Veri Kullanma Dili (Data Manipulation Language - **DML**)
 - ♦ DML komutları 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 Komutları

- ♦ 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 veritabanının veya veritabanı içindeki tabloyu siler.
- ♦ 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**).
- ♦ 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.

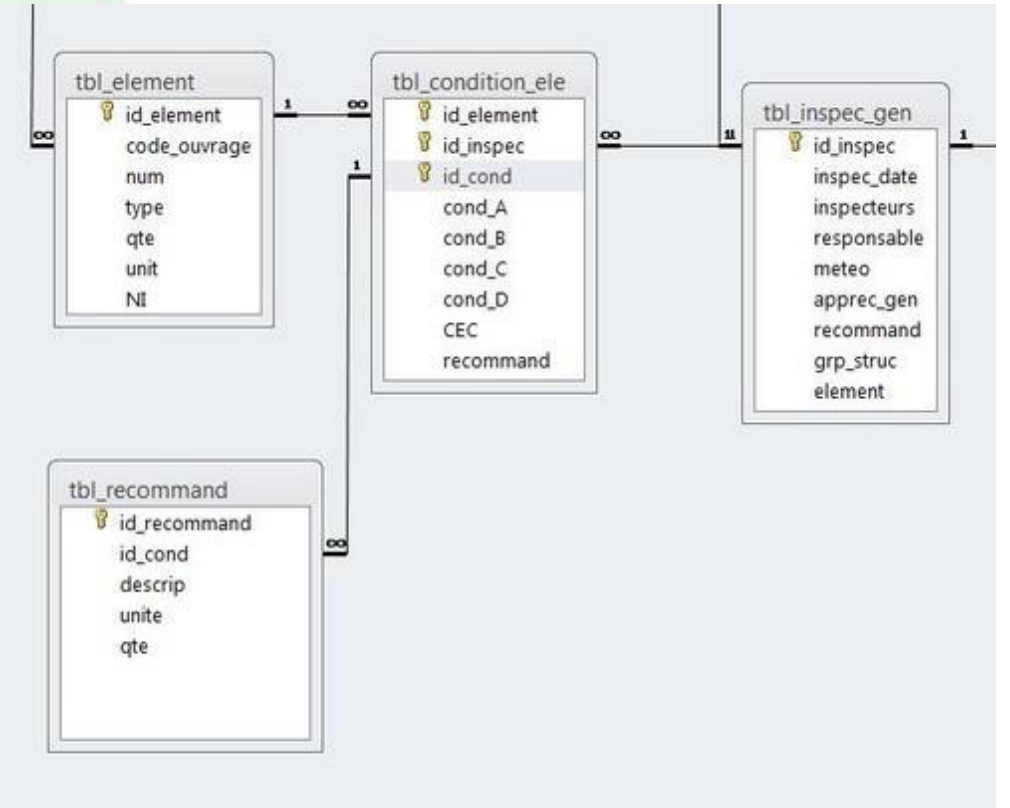
Primary Keys



<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

Primary Key

- ♦ Primary Key değeri boş geçilemez ve **NULL** değer 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**.



Foreign Key

- Foreign Key iki tablo arasinda relation olusturmak icin kullanilir
- Foreign Key baska bir tablodaki Primary Key ile iliskilendirilmis olmalidir.

Child Table

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

Parent Table

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

Bir Tabloda birden fazla Foreign Key olabilir Foreign Key NULL degeri kabul eder. Foreign Key olarak tanimlanan field' da tekrarlar olabilir.

Foreign Key, deęerleri farklı bir tablodaki Primary Key ile eęleşen bir sütun veya sütunların birleşimidir.

Foreign Key

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

- ♦ Michael Scott'un yoneticisi kimdir?
- ♦ Angela Martin'in Job_Name'i nedir ?
- ♦ Manual Tester'lerin ortalama Salary'si ne kadardir?
- ♦ En yuksek Salary' yi alan kisinin Job_Name'i nedir?

Composite Key

Job_ID	Job_Name
2	SDET
3	Manuel Tester
1	QA Led
Job Table	

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

Job_Id	Name	Company
2	Mark Eye	RCG
3	John Ted	RCG
1	Mark Eye	Signature
1	John Ted	Info Log
1	Cory Al	Info Log
2	Angela Star	Signature
Company Table		

- Composite Key birden fazla field (kolon)'in kombinasyonu ile olusturulur.
- Tek basina bir kolon Primary Key olma ozelliklerini tasimiyorsa, bu ozellikleri elde etmek icin birden fazla kolon birlestirilerek Primary olusturulur.

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

- ♦ Tom Hanks'in adresi nedir?
- ♦ John Walker'in eyaleti nedir?
- ♦ ID'si 15 olan kisinin sehri nedir?

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

- ♦ Biology dersi alan ogrenciler kimler?
- ♦ Selective ders alan ogrencilerin isimleri ?
- ♦ CourseFee 600 olan ogrencilerin isimleri ?

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

- ♦ Ogretmeni Mark Adam olan ogrencilerin isimleri nedir?
- ♦ Kevin Star'in ogretmenlerinin isimleri nedir?
- ♦ Pamela Star'in ogretmenlerinin isimleri nedir?

SQL Data Types (String)

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

SQL Data Types (Numeric)

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

SQL Data Types (Numeric)

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)

SQL Data Types (**Date**)

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

Summary

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

GRANT : Bir kullanıcıya yetki
vermek için kullanılır.

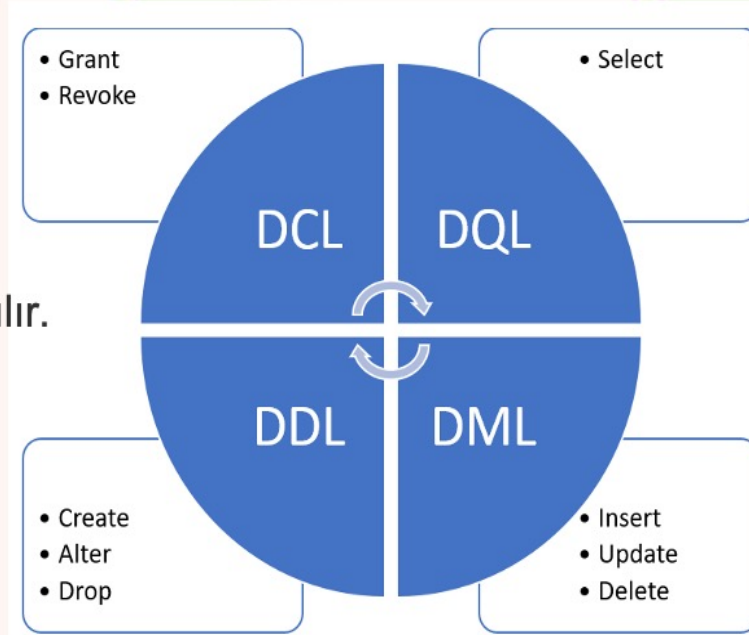
REVOKE : Bir kullanıcıya verilen
yetkiyi geri almak için kullanılır.

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.

En Popüler RDBMS

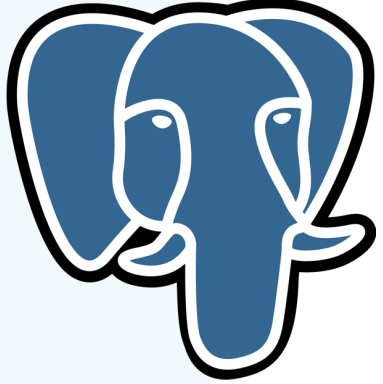


- **SQL Server** : Microsoft tarafından geliştirilmiştir.
- **Negatif**: Lisans Ücreti – Kurumsal Kullanıcılar için binlerce dolar ödenmesi gereklidir.
- **Pozitif**: Zengin bir user interface' e sahip ve çok büyük dataların kullanılmasında sorunsuz çalışır.



- **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.

En Popüler RDBMS

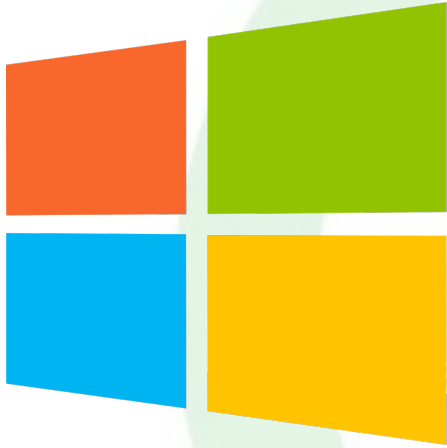


- **PostgreSQL Server** : Created by a computer science professor Michael Stonebraker.
- **Negatif**: İleri seviye yönetimi diğerlerine göre zayıf.
- **Pozitif**: Yeni nesil olarak ortaya çıktı. Kisiselleştirme mümkündür, zor görevler için ideal olabilir.



- **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. Komplike ve yüksek miktarda veri içeren yapılar için en çok tercih edilen Relational DB' dir.

PostgreSQL Kurulum



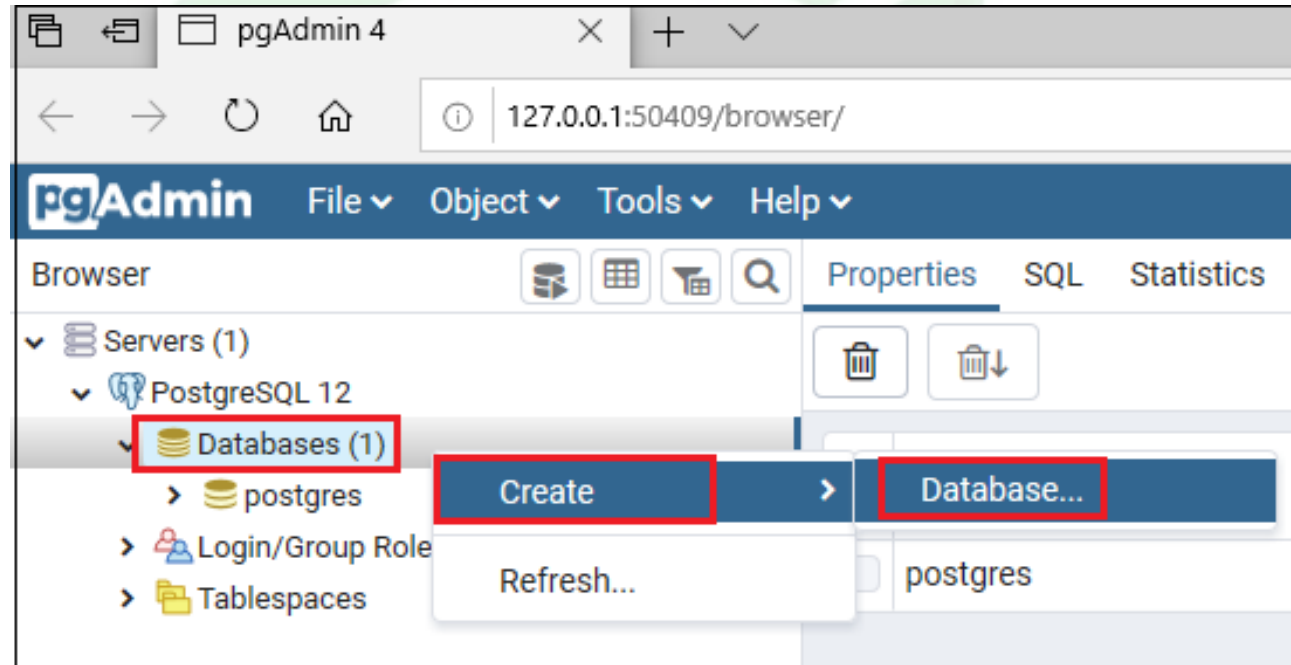
Kurulum Linki

<https://www.postgresql.org/download/>

Veritabanı İşlemleri

-- 1- Yorum Satırı

Veritabanı İşlemleri



-- 2- Database Oluşturma - DDL

CREATE DATABASE ms;

-- SQL komutlarında case sensitive yoktur ancak değer atamalarında vardır.

Veritabanı İşlemleri

-- 3- Database Silme – DDL

DROP DATABASE ms;

-- Silme işleminin yapılabilmesi için Database' den Disconnect yapılması gerekmektedir.

Veritabanı İşlemleri

-- 4- Tablo Oluşturma – DDL

```
CREATE TABLE students(  
  id CHAR(4),  
  name VARCHAR(50),  
  grade REAL,  
  register_date DATE  
);
```

-- ; kullanmak zorunda değiliz. Birden fazla komutu çalıştırmak istediğinizde sorgunun bittiği
-- yeri belirtmek için kullanılır. Daha readable bir yapı kavuşturmak istediğiniz zamanda bunu kullanabilirsiniz.

Veritabanı İşlemleri

-- 5- Var olan bir tablodan yeni bir tablo oluşturma

```
CREATE TABLE grades as SELECT name, grade FROM students;
```

-- Sadece isim ve notları tutan bir tablo yapıldı.

Veritabanı İşlemleri

-- 6- Tablo içine veri ekleme – DML

```
INSERT INTO students VALUES ('1000','Mehmet', 92.8, '2021-01-01');  
INSERT INTO students VALUES ('1001','Sungur', 81, '2022-10-08');  
INSERT INTO students VALUES ('1002','Kerem', 100, now());
```

-- Komutları çok seçerekde Execute edebilirsiniz. Bunun için ; kullanılması gerekmektedir.

Veritabanı İşlemleri

-- 7- Tablonun bazı sütunlarına veri ekleme – DML

```
INSERT INTO students(name,grade) VALUES ('Aslı', 87);
```


Veritabanı İşlemleri

-- 8- Tablonun tüm kayıtlarını getirme

```
SELECT * FROM students;
```

Veritabanı İşlemleri

-- 9- Tablonun bazı kayıtlarını getirme

```
SELECT name,grade FROM students;
```

Veritabanı İşlemleri

/*----- ÖDEV -----

1) Tablo Oluşturma

"tedarikciler" isminde bir tablo oluşturun,
"tedarikci_id", "tedarikci_ismi", "tedarikci_adres"
ve "ulasim_tarihi" field'lari olsun.

2) Var olan tablodan yeni tablo oluşturmak

"tedarikci_info" isminde bir tabloyu "tedarikciler" tablosundan oluşturun.
İçinde "tedarikci_ismi", "ulasim_tarihi" field'lari olsun

3) Data ekleme

"ogretmenler" isminde tablo oluşturun.

İçinde "kimlik_no", "isim", "brans" ve "cinsiyet" field'lari olsun.

"ogretmenler" tablosuna bilgileri aşağıdaki gibi olan kişileri ekleyin.

kimlik_no: 234431223, isim: Nana TechWorld, brans : DevOps, cinsiyet: kadın.

kimlik_no: 234431224, isim: Mehmet Ince, brans : Security, cinsiyet: erkek.

4) Bazı fieldları olan kayıt ekleme

"ogretmenler" tablosuna bilgileri aşağıdaki gibi olan bir kişi ekleyin.

kimlik_no: 567597624, isim: Stephane Maarek

★/

