



Structured Query Language
Yapılandırılmış Sorgu Dili

SQL (Structured Query Language) Nedir?

Yapılandırılmış sorgu dili anlamına gelir. Veritabanları üzerinde işlem yapmak, veri sorgulamak, veri eklemek, güncellemek, silmek ve veritabanı yapısını yönetmek için kullanılan standart bir programlama dilidir.

SQL'in Tarihçesi

SQL, 1970'lerde IBM tarafından geliştirilmiştir.

İlk olarak "SEQUEL" (Structured English Query Language) adıyla ortaya çıkmış, daha sonra SQL olarak adlandırılmıştır.

1986 yılında ANSI (American National Standards Institute) tarafından standart hale getirilmiştir.

SQL'in Temel Özellikleri

Veri Sorgulama: Veritabanındaki verileri çekmek ve filtrelemek için kullanılır (SELECT).

Veri Ekleme: Veritabanına yeni kayıtlar eklemek için kullanılır (INSERT).

Veri Güncelleme: Var olan kayıtları değiştirmek için kullanılır (UPDATE).

Veri Silme: Veritabanından kayıtları silmek için kullanılır (DELETE).

Veritabanı Yönetimi: Tablo oluşturma, silme, indeksleme ve kullanıcı yetkilendirme gibi işlemler için kullanılır (CREATE, DROP, ALTER).

SQL'in Kullanım Alanları

Veri Analizi: Büyük veri kümeleri üzerinde analiz yapmak.

Web Uygulamaları: Kullanıcı bilgilerini, ürün bilgilerini ve diğer verileri depolamak ve yönetmek.

İş Zekası: Raporlama ve veri görselleştirme için veri çekmek.

Veri Entegrasyonu: Farklı sistemler arasında veri alışverişi sağlamak.

SQL'in Avantajları

Standart Dil: SQL, neredeyse tüm ilişkisel veritabanı sistemleri tarafından desteklenir (MySQL, PostgreSQL, Oracle, SQL Server vb.).

Esnek Yapı: Veritabanı üzerinde çeşitli işlemler yapmaya olanak tanır.

Kolay Öğrenilebilir: Temel SQL komutları kolayca öğrenilebilir ve uygulanabilir.

SQL Komutları

1.SELECT (Seçme, Listeleme)

Tanım: Veritabanından veri çekmek için kullanılır.

Açıklama: SELECT ifadesi ile belirtilen tablodan veriler seçilir. WHERE, ORDER BY, GROUP BY gibi ifadelerle filtreleme ve sıralama yapılabilir.

Örnek:

Sql kodu:

```
Select * from Ogresci
```

SQL Komutları

* (yıldız) Kullanımı

- * Yıldız tablo içindeki tüm alanları seçmek anlamına gelir.
- Eğer özel bir alan adı olacaksa mutlaka belirtmek gereklidir.

Örnek:

Sql kodu: **Select * from Ogrenci** -> tüm tabloyu listeler

Sql kodu: **Select adsoyad,numara from Ogrenci** -> tablodaki adsoyad ve numara alanlarını gösterir.

SQL Komutları

WHERE (Koşul,Şart)

Tanım: Sorgularda belirli bir koşula göre filtreleme yapmak için kullanılır.

Açıklama: WHERE ifadesi ile belirtilen koşulu sağlayan kayıtlar seçilir, güncellenir veya silinir.

Örnek:

Sql kodu

```
SELECT * FROM Ogrenciler WHERE sinif = '10A'
```

SQL Komutları

ORDER BY (Sıralama, A'dan Z'ye veya Z'den A'ya, küçükten büyüğe veya tersi)

Tanım: Sorgu sonuçlarını belirli bir sütuna göre sıralamak için kullanılır.

Açıklama: ORDER BY ifadesi ile sonuçlar artan (ASC) veya azalan (DESC) sırada sıralanır.

Örnek:

Sql kodu

```
SELECT * FROM Ogrenciler ORDER BY ad ASC
```

```
SELECT * FROM Ogrenciler ORDER BY numara DESC
```

SQL Komutları

AND ve OR (ve,veya / Mantıksal Operatörler)

Tanım: Birden fazla koşulu birleştirmek için kullanılır.

Açıklama: AND tüm koşulların doğru olmasını gerektirirken, OR en az bir koşulun doğru olmasını gerektirir.

Örnek:

Sql kodu

```
SELECT * FROM Ogrenciler WHERE sinif = '10A' AND numara > 100
```

```
SELECT * FROM Ogrenciler WHERE sinif = '10A' OR sinif = '10B'
```

SQL Komutları

MIN, MAX, AVG (En büyük, En küçük, Ortalama)

Tanım: Sütunlar üzerinde hesaplama yapmak için kullanılır.

Açıklama:

MIN: Bir sütundaki en küçük değeri bulur.

MAX: Bir sütundaki en büyük değeri bulur.

AVG: Bir sütundaki değerlerin ortalamasını hesaplar.

Örnek:

Sql kodu

```
SELECT MIN(not) FROM Notlar
```

```
SELECT MAX(not) FROM Notlar
```

```
SELECT AVG(not) FROM Notlar
```

SQL Komutları

Like (içeren, içinde yer alan)

Tanım: SQL'de metin veriler üzerinde belirli bir desen ile eşleşen kayıtları bulmak için kullanılan bir operatördür. Genellikle SELECT sorgularında WHERE koşulu ile birlikte kullanılır. LIKE, özellikle tam eşleşme yerine kısmi eşleşme arandığı durumlarda kullanışlıdır.

Açıklama:

Joker Karakterler

* (yüzde işareti): Sıfır veya daha fazla karakteri temsil eder.

_ (alt çizgi) : Tek bir karakteri temsil eder.

Örnek:

Sql kodu

```
SELECT * FROM Ogrenciler WHERE ad LIKE 'Ah*'  
SELECT * FROM Ogrenciler WHERE ad LIKE '_a*'
```

SQL Komutları

2.INSERT (Kayıt Ekleme)

Tanım: Veritabanına yeni kayıt eklemek için kullanılır.

Açıklama: INSERT INTO ifadesi ile belirtilen tabloya yeni bir kayıt eklenir.

Örnek:

Sql kodu

```
INSERT INTO Ogrenci (ad, soyad, numara) VALUES ('Ahmet', 'Yılmaz', 123)
```

SQL Komutları

3. UPDATE (Bilgileri Güncelleme)

Tanım: Var olan kayıtları güncellemek için kullanılır.

Açıklama: UPDATE ifadesi ile belirtilen tablodaki kayıtlar güncellenir. WHERE koşulu ile hangi kayıtların güncelleneceği belirlenir

Örnek:

Sql kodu

```
UPDATE Ogrenciler SET ad = 'Mehmet' WHERE numara = 123
```

SQL Komutları

4.DELETE (Kayıt Silme)

Tanım: Veritabanından kayıt silmek için kullanılır.

Açıklama: DELETE ifadesi ile belirtilen tablodan kayıtlar silinir. WHERE koşulu ile hangi kayıtların silineceği belirlenir.

Örnek:

Sql kodu

```
DELETE FROM Ogrenciler WHERE numara = 123
```


Veritabanı Yönetim Komutları

1.CREATE (oluşturma)

Tanım: CREATE komutu, yeni bir veritabanı nesnesi (tablo, veritabanı, indeks, görünüm vb.) oluşturmak için kullanılır.

3 farklı kullanım senaryosu vardır.

a.Veritabanı Oluşturma:

Sql kodu

```
CREATE DATABASE OkulDB
```

-> Bu sorgu, "**OkulDB**" adında yeni bir veritabanı oluşturur.

Veritabanı Yönetim Komutları

b. Tablo Oluşturma:

Sql kodu

```
CREATE TABLE Ogrenciler (  
  id INT PRIMARY KEY,  
  ad VARCHAR(50),  
  soyad VARCHAR(50),  
  sinif VARCHAR(10)  
);
```

-> Bu sorgu, "Ogrenciler" adında bir tablo oluşturur. Tablo, id, ad, soyad ve sinif sütunlarından oluşur.

Veritabanı Yönetim Komutları

c. İndeks Oluşturma:

Sql kodu

```
CREATE INDEX idx_ogrenci_ad ON Ogrenciler (ad);
```

-> Bu sorgu, "Ogrenciler" tablosundaki ad sütünü üzerinde bir indeks oluşturur.

Veritabanı Yönetim Komutları

2.ALTER (değiştirme)

Tanım: ALTER komutu, mevcut bir veritabanı nesnesinin yapısını değiştirmek için kullanılır. Örneğin, tabloya yeni bir sütun eklemek veya var olan bir sütunu silmek için kullanılır.

2 farklı kullanım örneği vardır.

Veritabanı Yönetim Komutları

a. Tabloya Yeni Sütun Ekleme:

Sql kodu

```
ALTER TABLE Ogrenciler ADD COLUMN dogum_tarihi DATE
```

-> Bu sorgu, "**Ogrenciler**" tablosuna **dogum_tarihi** adında yeni bir sütun ekler.

Veritabanı Yönetim Komutları

b. Sütun Silme:

Sql kodu

```
ALTER TABLE Ogrenciler DROP COLUMN dogum_tarihi
```

-> Bu sorgu, "Ogrenciler" tablosundan **dogum_tarihi** sütununu siler.

Veritabanı Yönetim Komutları

3. DROP (silme,atmak)

Tanım: DROP komutu, mevcut bir veritabanı nesnesini (tablo, veritabanı, indeks vb.) tamamen silmek için kullanılır. Bu işlem geri alınamaz, bu nedenle dikkatli kullanılmalıdır.

3 farklı kullanım örneği vardır.

Veritabanı Yönetim Komutları

a. Tablo Silme:

Sql kodu

```
DROP TABLE Ogrenciler;
```

-> Bu sorgu, "Ogrenciler" tablosunu içindeki verilerle birlikte tamamen siler.

Veritabanı Yönetim Komutları

b. Veritabanı Silme:

Sql kodu

```
DROP DATABASE OkulDB;
```

-> Bu sorgu, "OkulDB" veritabanını içindeki tablo ve verilerle birlikte tamamen siler.

Veritabanı Yönetim Komutları

c. İndeks Silme:

Sql kodu

```
DROP INDEX idx_ogrenci_ad ON Ogrenciler
```

-> Bu sorgu, "Ogrenciler" tablosundaki idx_ogrenci_ad indeksini siler.