

## SQL SORGULARI - Declarative programlama – Sonuç alma

**SELECT \* FROM** film; **Film** tablosunda ki bütün alanları çeker.

**SELECT title FROM** film; tablodan sadece **title** alanını çeker.

**SELECT title, description FROM** film; tablodan **title, description** alanlarını çeker.

**SELECT \* FROM** film **WHERE** replacement\_cost = 14.99; **WHERE** filtreleyerek koşul sağlar!

**SELECT \* FROM** actor **WHERE** first\_name = 'Penelope'; Penelope ismindeki actorleri getir.

### Karşılaştırma Operatörleri

**SELECT \* FROM** film **WHERE** lenght > 90; > 90'dan büyük olan filmleri çeker.

**SELECT \* FROM** film **WHERE** rental\_rate <= 1; rental\_rate tablosundan 1 den küçük ve eşit olanları çek.

**SELECT \* FROM** film **WHERE** rental\_rate <> 0.99; <> Eşit olmayanları gösterir.

### WHERE ve Mantıksal Operatörler

**SELECT** first\_name, last\_name **FROM** actor **WHERE** first\_name = 'Penelope'; isim ve soy isimler tablosunu göster. Actor tablosundan ismi penelope olanları çek.

**SELECT** first\_name, last\_name **FROM** actor **WHERE** first\_name = 'Penelope' **AND** last\_name = 'Monroe'; **AND** adı ve soy adı olanı getir!

**SELECT** first\_name, last\_name **FROM** actor **WHERE** first\_name = 'Penelope' **OR** first\_name = 'Bob'; isim ve soy isimler tablosunu göster. **OR** adı olanları getir!

**SELECT \* FROM** film **WHERE NOT** rental\_rate = 4.99; 4.99 olanları alma !

boolean **AND** boolean → boolean  
boolean **OR** boolean → boolean  
**NOT** boolean → boolean

**VE**  
**VEYA, YA DA**  
**DEĞİL**

<i>a</i>	<i>b</i>	<i>a AND b</i>	<i>a OR b</i>
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
TRUE	NULL	NULL	TRUE
FALSE	FALSE	FALSE	FALSE
FALSE	NULL	FALSE	NULL
NULL	NULL	NULL	NULL

<i>a</i>	<b>NOT a</b>
TRUE	FALSE
FALSE	TRUE
NULL	NULL

**SELECT \* FROM** film **NOT** Sorguların hepsi gelir fakat birbirleri  
**WHERE NOT** (rental\_rate = 4.99 **AND** replacement\_cost = 20.99); ile eşleşmezler!

**SELECT \* FROM** film **NOT NOT** birbirini götürür! Sorgu Doğru  
**WHERE NOT (NOT** (rental\_rate = 4.99 **AND** replacement\_cost = 20.99)); çalışır.

## LESSON 2

**BETWEEN** Belirli aralıktaki verileri yakalamak için between anahtar kelimesini kullanınız!

**SELECT** title, length **FROM** film 90 ve 120 dakika arasında olan verileri çek.  
**WHERE** length >= 90 **AND** length <= 120;

**SELECT** title, length **FROM** film 90 ve 120 dakika arasında olan  
**WHERE** length **BETWEEN** 90 **AND** 120; verileri **BETWEEN** ile çeker.

**SELECT** title, length **FROM** film **NOT** 90 ve 120 dakika arasında olmayaları  
**WHERE** length **NOT BETWEEN** 90 **AND** 120; çeker.

**SELECT** rental\_rate, replacement\_cost **FROM** film rate 2 ve 4 arasında cost 10 ve 20 arasında  
**WHERE** (rental\_rate **BETWEEN** 2 **AND** 4 ) **AND** (replacement\_cost **BETWEEN** 10 **AND** 20);

**IN** Belirli bir veri kümesi içerisinde olanları tek tek getirir.

**SELECT \* FROM** film **IN** sadece 40 50 60 var mı diye bakar varsa  
**WHERE** length **IN** (40 , 50, 60); çeker.

**SELECT \* FROM** film sadece 40 50 60 var mı diye bakar varsa  
**WHERE** length = 40 **OR** length = 50 **OR** length = 60; çeker. Uzun sorgu.

**SELECT \* FROM** film veriler varsa getir.  
**WHERE** replacement\_cost **IN** (10.99 , 12.99, 16.99);

**SELECT \* FROM** film verilerin dışında olanları getir.  
**WHERE** replacement\_cost **NOT IN** (10.99, 12.99, 16.99);

## LESSON 3

**LIKE** Benzer verileri belirli bir filtreden geçirmeyi sağlar.

**ILIKE** Case Sensitive ortadan kaldırır, bütün verileri çeker

**%** Ya herhangi bir karakter yoktur ya da birden fazla karakter için yer tutucu işlevi görevi yapar.

**SELECT \* FROM** customer Baş harfi M ile başlayan tüm verileri çeker.  
**WHERE** first\_name **LIKE** 'M%';

**SELECT \* FROM** customer Baş harfleri Ma ile başlayan tüm verileri çeker  
**WHERE** first\_name **LIKE** 'Ma%';

**SELECT \* FROM** customer Son karakter y ile bitenleri getirir.  
**WHERE** first\_name **LIKE** '%y';

**SELECT \* FROM** customer A harfi ile başlayıp Y harfi ile bitenleri getirir.  
**WHERE** first\_name **LIKE** 'A%y';

**SELECT \* FROM** customer  
**WHERE** first\_name **LIKE** 'A%' **AND** Last\_name **LIKE** 'A%';

**SELECT \* FROM** customer  
**WHERE** first\_name **ILIKE** 'a%';

Başına "I" -> ILIKE Case sensitive değil.

**SELECT \* FROM** customer  
**WHERE** first\_name **NOT LIKE** 'M%';

M ile başlamayan hepsini getirir.

**SELECT \* FROM** customer  
**WHERE** first\_name **LIKE** 'J\_an';

\_ -> Tek bir karakter için yer tutucu sağlar.

**SELECT \* FROM** customer  
**WHERE** first\_name **LIKE** 'J\_';

Jo, Ja, Jx, J?

**SELECT \* FROM** customer  
**WHERE** first\_name **LIKE** 'J\_n';

J\_n -> tek bir karakter tutar.

**SELECT \* FROM** customer  
**WHERE** first\_name **NOT LIKE** 'J%n';

Tüm verilerden J ile başlayıp n ile bitenlerin dışındakileri çeker.

**LIKE** yerine PostgreSQL'de ~~ işareti kullanılabilir!

**SELECT \* FROM** customer  
**WHERE** first\_name ~~ 'M%';

~~ -> **LIKE** Temsil eder M ile başlayan hepsini getirir.

**ILIKE** yerine PostgreSQL'de \* işareti kullanılabilir!

**SELECT \* FROM** customer  
**WHERE** first\_name ~~~\* 'm%';

\* -> **ILIKE** Temsil eder Case sensitive duyarlı değil.

**NOT** yerine PostgreSQL'de ! işareti kullanılabilir!

**SELECT \* FROM** customer  
**WHERE** first\_name !~~\* 'm%';

! -> **NOT** Temsil eder Değil anlamı katar.

## LESSON 4

**DISTINCT** Sütun içinde ki benzersiz (birbirinden farklı) verileri görmek için kullanılır.

**SELECT DISTINCT** rental\_rate **FROM** film;

**SELECT DISTINCT** replacement\_cost **FROM** film;

**SELECT DISTINCT** rental\_rate, replacement\_cost **FROM** film;

**COUNT** Sütun içinde ki verilerin sayısını bulmak için kullanılır.

**SELECT COUNT(\*) FROM** actor  
**WHERE** first\_name = 'Penelope';

**COUNT** Fonksiyon olduğu için () kullanılır.  
Aktör tablosun da 4 tane Penelope ismi vardır

**SELECT COUNT(\*) FROM** actor  
**WHERE** first\_name ~~ 'A%';

Aktör tablosun da 13 tane A ile başlayan isim vardır.

**SELECT COUNT(DISTINCT first\_name) FROM** actor;

Aktör tablosundan farklı kaç tane isim var.128

**SELECT COUNT(DISTINCT length) FROM** film;

toplam 140 tane veri olduğunu gösterir.