

Potika SQL kursu

Kurulum sonrası
Query tool

1-) Select

= Seçmek için

Select column1 From table1

-- = yorum satırı

örnek

→ Select title, description from film;

Select * FROM film; > hepsini seçmek için

2-) Where

= Koşullar arasında filtreleme yapmak için

Select * FROM film Where rental-rate = 0.99;

böyle de
yazılabilir

Select *
From film
Where rental-rate = 0.99;

Select * ③ From film ①
Where rental-rate = 0.99 ②

programın
bakma
sıralaması

Select * from actor
Where first-name = 'Penelope';

Select * From film
Where length <= 90;

<> = eşit değil
!= eşit değil

mantıksal operatörler

And, or, not

Select * from film
Where rental-rate = 4.99 AND rental-rate = 2.99;

And = Kendinden önceki koşulu sağlaması lazım

Or = yeni bir query oluşuyor gibi

3-) Between, IN

Between'de = olur.

(1 yapmazsan

IN'den sonra data veriyors

→ Belirli aralıktaki değerler için
where $x > 90$ and $x < 120$ yerie

IN → Pşittirin birden fazla
veri için olanı

Select title, length From film

Where length ≥ 90 AND length ≤ 120

= Where length BETWEEN 90 AND 120

Where length IN (40, 50, 60);

= Where length = 40 or
length = 50 or

Where (rental-rate BETWEEN 2 AND 4) AND (replacement-cost BETWEEN 4 AND 10)

Not = + Between'de parantez kullanırsan where'den sonra
+ IN'den sonra parantez koyarsın.

+ Between'de sayılar arası AND, OR
IN'de sayılar arası virgül

4-) LIKE, ILIKE

→ Belirli verileri almak için

*% → ya hiç bir karakter yok
yada birden fazla karakter var

* → tek karakter

→ Büyük küçük

hassasiyeti yok

Select * FROM customer

Where first-name LIKE 'm%';

= Adı M ile başlayanlar 'ma%';
ma ile başlayan

'o%y'

son harfi y olan

'A%y'

A ile başlayıp y ile biten

LIKE = büyük, küçük önemli

ILIKE = "" önemli

Where first-name LIKE 'A%' AND

last-name LIKE 'A%';

İsmi ve soyismi A ile başlayan

Like 'J-on';

↓
tek karakter demek

~ ~ = LIKE

~ ~ * = ILIKE

! ~ ~ = Not LIKE

! ~ ~ * = Not ILIKE

5-) Distinct, Count

Verilerin Benzersizlik
Sütunun tekrar etmemiş halini gösterir.

```
Select DISTINCT replacement_cost From film
```

Birden fazla sütun yazarsan
= Benzersizliği 2 sütun kombinasyonu şeklinde yapar

Kaç adet olduğunu gösteriyor

```
Select COUNT(*) FROM actor  
Where first_name = 'Penelope'
```

LIKE

actor tablosunda birbirinden farklı kaç first_name var

```
Select count(DISTINCT first_name) FROM actor;
```

Distinct count beraber kullanılırsa
Önce count sonra Distinct

-PSQL Uygulama-1

İ Zorunlu

Bitmes, Çalıştırmas

Kurulum yapıldı

1) Açmak için psql -U postgres

2) İstedğin veritabanını açmak için \connect dvdrental

3) tabloları görmek için \dt

4) tablo içi detayları görmek için \d actor

j → Çalıştırır

j koymazsan
alt satıra geçebilirsin

Ctrl+C = çıkma

cls = temizleme

SQL Temelleri 2

Verileri sıralama, gruplama
toplama fonksiyonları

1) ORDER BY → Sıralama

ASC → Ascending = Artan bir şekilde
DESC → Descending = Azalan bir şekilde

Koşullardan
sonra
Bulanılır

Select * From film
Order By title ASC; Order By length DESC;

Select title, rental-rate, length FROM film
ORDER BY rental-rate ASC, length DESC;

→ Where LIKE 'A%'
title

2) LIMIT, OFFSET

görmek istediğimiz
sayıda veri görüyoruz.

LIMIT 10;

En son yazıyoruz.

En uzun (length) 7 film

Select * FROM film → veriyi al

Where replacement-cost = 14.99 AND

rental-rate = 0.99 → koşul

ORDER BY length DESC → sırala

LIMIT 7; → sınırla

→ Belirli sayıda veri geçmek için

Genelde ilk sayfada 5 veri kullanmak için 6. 2. sayfada olsun diye kullanılır

OFFSET 6

Select ...
Where ...
ORDER BY ...
OFFSET ...
LIMIT ...

→ en son

3-) ^{toplanmış} Aggregate Fonksiyonlar

Şu ana kadar Aggregate fonksiyon sadece → Count gördük

→ Max, Min, AVG, ROUND, SUM, COUNT

Select MAX(replacement-cost) FROM film;

min

AVG → Average → duyarlılığı 3 karaktere yuvarlamak için

SUM

Select ROUND(AVG(length), 3) FROM film;

↓
yuvarlama

↓
Average

↓
kaç sayı yuvarlanacak

Select MAX(length), MIN(length), SUM(replacement-cost) FROM film;

Select MAX(length), rental-rate FROM film;

tek değer

sütun

eksikmiş hata verir
group'lar eksikmiyor

Select MAX(length) FROM film

Where rental-rate = 0.99;

→ rental-rate 0.99 olan length max
olan değer

rental-rate 0.99 max değerleri
2.99 olan bulmamız için
4.99 24 Group by
kullanıcısı

4-) Group by

Bir grupta London'da olan sayısını
count where ile bulursun

Ama

London → sayılarını
Toronto → Group by
Turkey → ile

Select rental-rate, MAX(length) FROM film
Group by rental-rate

her bir replacement-cost karşısına karşılık gelen en kısa film

Select, replacement-cost, MIN(length) FROM film

Group by replacement-cost

→ Buraya gruptadığımız sütun ve aggregate fonk.
dışında birşey yazamazsın.

5-) Having

↳ Gruplanmış verilere bir koşul eklemek istersen

```
Select rental-rate, COUNT(*) FROM film
Group by rental-rate
Having COUNT(*) > 325
```

↓
Group bazlı filtreleme

```
"
= Where rental-rate != 2.99
Group by rental-rate
```

↓
Sıra bazlı filtreleme

```
Select customer-id, SUM(amount) FROM payment
Group By customer-id
Having SUM(amount) > 100
ORDER BY SUM(amount) DESC
LIMIT 10;
```

6-) Alias (takma ad)

↓
As
Geçici isim verme sütunlara

çok karmaşık
yapılarda kullanılır

count → ? sayısı yapabilirsiniz

```
Select first-name As isim From actor;
```

```
Select first-name isim From actor;
```

*As yazmasanda oluyor.

birden fazla kelimede " " al.

```
Select first-name AS "isim + est"
```

Concat → 2 sütunu birleştirme

```
Select CONCAT(first-name, ' ', last-name) AS "isim ve soyisim" FROM actor;
```


3. Bölüm Tablolarla Çalışmak

1-) Tablo oluşturmak - Silmek

CREATE TABLE <table-name> (
 <column-name> <data-type> <constraint>,
 ...
 <column-name> <data-type> <constraint>)

Kullanıcı mı? sütunlar → *veri tipi (numeric, string, tarih...)*
→ *kısıtlama verinin özellikleri*

Create table author (

id SERIAL PRIMARY KEY

First-name VARCHAR(50) NOT NULL,

Last-name VARCHAR(50) NOT NULL,

email VARCHAR(100),

birthday DATE

);

⇒ Select * From author

INSERT INTO author (id, first-name, last-name, email, birthday)

VALUES

('Haruki', 'Murakami', '...', '1948-11-07'),

...

⇒ Select * From author

→ CREATE Table author2 (LIKE author);

INSERT INTO author2
SELECT * FROM author
Where first-name = 'Sabahattin';

→ hazır tablodan veri kopyalama

tabloyu verileriyle kopyalama

Create table author3 AS
Select * from author

tabloyu yedeği için

tabloyu silme

→ DROP table author2;

DROP table IF EXIST author2;

→ böyle bir tablo olmasa bir hata vermez.

2-) Verileri Güncellemek - Silmek

Verileri manuel olarak tek tek girmek çok zor

→ generate dummy data sql → mockaroo.com

SQL yap, PREVIEW yap, kopyala yapıştır.

- Update Güncellemek

güncellemek
istediğimiz
veri

```
UPDATE <table-name>  
SET column1 = Value1,  
    column2 = Value2;  
Where condition;
```

```
UPDATE author  
SET first-name = 'Emrah',  
    last-name = '...',  
    email = '...';
```

Where id = 10;

→ Where first-name LIKE 'V%';

- Returning

Çalıştırınca → değeri döndürüyor
gösteriyor

Returning *
Returning first-name

Delete

→ From

```
Delete <table-name>  
Where condition
```

→ From
Delete company3
Where id = 6

Delete From company3
Where id > 11

Returning *;

Primary Key - Foreign Key

Satırda bulunan bir veriyi diğerleriyle ayırtırmak

Başka bir tablodaki satıra referans verir.

→ önce yeni bir tablo oluşturalım

Create table book (

id Serial Primary Key,

title VARCHAR(100) Not null,

Page-number INTEGER Not null,

author-id INTEGER REFERENCES author(id)

↳ referans vericeğin tablo tekrar edebilir diye integer

);

Not = Referans vericeği sayı, asıl tabloda yoksa hata verir

Primary Key = 0 satır için benzersiz olucak (Benzersiz tamamlayıcı)
1 tabloda yalnızca 1 primary key olur.
Değerleri değiştirilemez.

Foreign Key = tekrar edebilir (ör = hangi yazarların yazdığı)
yazılan sayı referans verilecek tabloda olmalı.
değiştirilebilir

Join → Birleştirme tabloları

Join author ON author.id = book.author.id;

Veri tipleri 1

Sütuna yazılacak verileri belirler.

Numeric types →
Real = 6 rakam
double precision = 15 rakam
numeric = çok rakam

CHARACTER TYPE

Select ('Lorem' :: Char(10)); → Lorem → 10 Karakter Sabit

Select ('Lorem' :: Varchar(10)); → Lorem → 5 Karakter ^{02 varsa olduğu kadar}

Select ('... ' :: Varchar); → Kaç Karakterse
TEXT

Boolean type

→ TRUE, FALSE, NULL
" " "
yes, on, 1 no, off, 0

Select ('no' :: Boolean) = False

Date time types

Select ('1980-12-03' :: DATE);

Select ('02:16' :: TIME);

TIMESTAMP → yıl ay gün saat dk snr

Not Null ve Alter

Null = Bilinmeyen bir veri, eksik veri

→ Not null 'i tabloyu oluştururken yapmadıysan

Alter TABLE users

Alter COLUMN username

SET NOT NULL;

Delete from users

Where username IS NULL

Returning *;

= kullanılmaz.

Bos veri eklemek

INSERT INTO users (username, email, age)
VALUES

(' ', 'ali@gmail.com', 35);

Unique Benzersiz

Verilerin benzersiz olmasını istediği zaman.

```
Insert into users (username, email, age)
Values
(      ),
(      );
```

1- tabloyu oluştururken
email varchar(50) UNIQUE

2- oluşturduktan sonra

~~ALTER TABLE users~~
ADD UNIQUE(email);

Check

Veri eklerken belirli şartlar koymak için

ör= Yaş 18'den az olamaz

1- tabloyu oluştururken

2- tabloda değişiklik yaparken

2-) ALTER TABLE users
ADD CHECK (age > 18);

→ uymayanlar var diye hata varsa

```
Delete From users
Where id = 11;
```

1-) Yeni tabloda

```
CREATE TABLE users (
age numeric CHECK (age > 18),
second age numeric check (second age > 20),
CHECK (second age > age)
);
```

Tablolar PSQL

→ Komut ekranı

psql -U postgres > Giriş yapma

\1 > Veri tabanlarına bakma

CREATE DATABASE testdb; > oluşturma

\1

psql -h localhost -p 5432 -U postgres testdb → direkt girme

\c dvdrental > başka veritabanına bağlanmak istersen

```
CREATE TABLE users (
id SERIAL PRIMARY KEY,
```

);

\dt users

Sütun ismi değiştirme

```
ALTER TABLE users COLUMN
birthday to birthday
```


Önemli Not

Tabloyu oluşturdun aynı değerler var.
Bir sütunu UNIQUE yapmak istiyosun.
Yapınca → aynı değerler var hatası alırsın.

① Önce

güncelleme

UPDATE users SET username = 'Ali'

Where id = 51;

aynı değerlerden birini değiştirdin

② Sonra

değiştirme

ALTER TABLE users ADD CONSTRAINT

unique_username UNIQUE(username)

ekleme

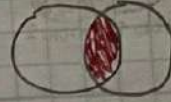
ALTER TABLE users
ADD UNIQUE(email)

silme

ALTER TABLE users
DROP CONSTRAINT email;

Join Yapıları Tabloları Birleştirmek

1. tablo 2. tablo



1-) INNER JOIN (2 kürenin ortak kesişimi)

2. tablo → *author
id name last-name
email birthday

*book
id title
page-number
author-id

İstenilen = book → title author → name, last-name

```
Select title, first-name, last-name FROM book
INNER JOIN author ON book.author-id = author.id;
```

birleştirilmesini 2. tablo *1. tablo* *author id* *book* *tablo* *sütun* *tablo* *sütun*

→ Join yazınca = INNER JOIN olur. *INNER JOIN → önce author sonra book yazmakta olur. Simetrik*

```
Select book.title, author.first-name, author.last-name FROM book
```

2-) Left Join

Önce ilk tabloyu alıyo, sonra 2. tabloda eşleşenlerle eşleştiriyor.



```
Select author.first-name, author.last-name,
book.title FROM author
```

1. tablo

```
LEFT JOIN book ON author.id = book.author-id;
```

2. tablo

⇒ Önce yazarların hepsini alıyosun, sonra yazarlara göre kitapları eşliyorsun.

Sadece kitabı olanlar

```
Where book.id IS NOT NULL
ORDER BY book.title
```

↓
bişey yoksa
= 0 değil
IS NOT NULL

3-) Right join

eşleşen
null var



Önce 2'yi alıyor
Sonra eşleşen 1'leri
Eşleşmeyen NULL

Where ayarı

Select * from author

FULL join book ON author.id = book.author.id

Where (book.id IS NULL OR author.id IS NULL)



4-) Full join

eşleşen
var null
null var



Önce 1'i alıyosun
Sonra 1 ile eşleşen 2'leri
Eşleşmeyene NULL
En son eşleşmeyen 2 ile 1'leri

5-) Union

Birden fazla select'li tek sonuç olarak çıkarılır

```
(  
  Select * FROM book  
  ORDER BY page-number DESC  
  LIMIT 5  
)  
  
UNION  
  
(  
  Select * FROM book  
  ORDER BY title  
  LIMIT 5  
)  
;
```

9 sonuç çıkıyor

demek ki 1 tane kesişim var

UNION ALL → 10 sonucu çıkarır

Üstte fazla sütun **Hata**

altta farklı

edit olmalı

Sütun sayıları aynı olmalı
Sütunlar aynı cins olmalı → varchar
integer

6-) Intersect ve Except

↓
Kesişimi alır
2 select fonk.

↓
İlk sorguda bulunan
ikinci sorguda olmayan

```
(  
Select * FROM book  
ORDER BY Page-number DESC  
LIMIT 10
```

sütun sayıları uymak zorunda
sütun veri tipleri

```
)  
INTERSECT/Except
```

```
(  
Select * FROM book  
ORDER BY title  
LIMIT 10
```

```
);
```

Alt Sorgular (Subqueries)

- Alt Sorgular Nedir?

```
Subqueries  
Select title, page-number, (Select MAX(page-number) FROM book),  
( (Select MAX(page-number) FROM book) - Page-number) AS differ  
FROM book
```

Where page-number >

```
(  
Select page-number FROM book  
Where title = 'Gülün adı'
```

Subqueries

```
);
```


Any ve All Operatörleri

↓
herhangi bir koşul doğru olursa oluyor.

↓
hepsi doğru olmalı

Birden fazla gelen değeri bir sütundaki değeri karşılaştırmamızı sağlıyor

```
Select first-name, last-name FROM author  
Where id = any
```

```
(  
  Select id FROM book  
  Where title = 'Ali' OR title = 'veli'  
)
```

Subquery and join