SQL Examples

Tags	
Last edited time	@February 22, 2023 9:04 AM
Multi-select	
Status	Not started

TEMELLER
select * from database;
select * from …. where </>/=/! =;(filtreleme)

```
select first, last, city
   from empinfo
   where first LIKE 'Er%';
```

```
select * from film
where release_year=2006
and rental_duration >=7;
```

select * from … whre … and/ …;
and, her şey true
or ,bir şey true olsa yeter
and ve or koşulu, bir den fazla yazılabilir
select * from … where NOT ;

```
SELECT *
FROM film
WHERE NOT (rental_rate = 4.99 OR rental_rate = 2.99)
```

where… between x1 and x2;

```
SELECT length
FROM film
WHERE length BETWEEN 100 AND 140;
```

where… IN (x1,x2,…);

```
SELECT length
FROM film
```

```
WHERE length IN (30,60,90,120);
```

columns NOT IN (…);

select * from film where Like/ilike ‘harflerle başlama%/%harflerle bitme’;

```
SELECT *
FROM actor
WHERE first_name LIKE 'P%';

SELECT *
FROM actor
WHERE first_name not ilike 'p%e';

SELECT *
FROM actor
WHERE first_name ilike 'j__';
```

select Count/distinct (sayıları sayma,farklı sayıları sayma) from …;

```
select distinct film_id
from film;

select count(film_id)
from film;

select count
(distinct first_name )
from actor;
```

select ….. from ORDER BY ….. ASC/DESC ;

```
SELECT *
FROM film
WHERE title LIKE 'A%'
ORDER BY title ASC length DESC;
```

Ofset and limit (kırma ve belli bir sayı kadar sıralma)

```
SELECT title,replacement_cost,rating
FROM film
WHERE title LIKE '%r'
ORDER BY replacement_cost
OFFSET 4
LIMIT 6;
```

TEMELLER 2

AGGRegate Fonksiyonlar (**MIN, MAX, SUM, AVG,COUNT)**

```
select ROUND(avg(length),3) from film;

select min(length),max(length),sum(length),avg(length) from film;

select min(length),max(length),sum(length),avg(length) from film
where rental_rate=2.99;

--ödevler

select round(avg(rental_rate),2) from film;

select count(title) from film
where title like 'C%';

select max(length) from film
where rental_rate=0.99;

select count(distinct(replacement_cost)) from film
where length>150;
```

select x1,x2 from… GROUP by x1,x2; (gruplama)

```
select replacement_cost,rental_rate,min(length) from film
group by replacement_cost,rental_rate
order by min(length)
limit 8
offset 2;
```

1. veriyi al (agrregate et),
2. grupla
3. sırala
4. limitle,
5. ayır
6. ;

select x1,x2( )from… group by x1,x2 HAVING …>/</=/=! ..; (grupladığımız verilere filtreleme)

```
SELECT rental_rate, COUNT(**)
FROM film
GROUP BY rental_rate
HAVING COUNT(**) > 325;

--ödev 7
select rating from film group by rating;
select replacement_cost,count(*) from film group by replacement_cost having count(*)>50;
select store_id,count(*) from customer group by store_id;
select country_id as id,count(city) as maxcity from city group by country_id order by
count(city) desc limit 1 ;
```

ALIAS(takma isim vermek)
select x1 AS “text” from data;

```
select country_id as id,count(city) as maxcity from city
```

Tablolar Çalışmak
tablo açmak
(authora ilk tablo,author ikinci kopya tablo)

```
--create table <table_name> (
--<column_name> <data_type> <constraint>,
-- <column_name> <data_type> <constraint>,
--)
```

```
create table authora (
id serial primary key,
first_name varchar(50) not null,
last_name varchar(50) not null,
email varchar(100),
birthday date
);
```

tabloya veri eklemek

```
insert into authora (first_name,last_name,email,birthday)
values
(' mert','dil ','mertdil@ ','03.01.1999'),
('gizem','dil','izemdil@','09.01.1999');
```

tablo kopyalama

```
create table author (like authora);
```

satır kopyalama

```
insert into author
select * from authora
where first_name='gizem';
```

bütün tabloyu kopyalama

```
create table author3 as
select * from authora;
```

tablo silme

```
drop table if exists author10;
```

tabloya randam veriler eklemek istersek
https://www.mockaroo.com
tabloya update yapmak istersek

```
update author_3
set
first_name='mert',
last_name='dil',
email='mert@dil.com',
birthday='03.01.1899'
where id=2;
```

sona ekler ama!
koşullu günceleme

```
update author_3
set
first_name='xxxx',
last_name='yyyy'
where first_name like 'm%';
-----değiştirilen veriyi görme
update author_3
set
first_name='xxxx',
last_name='yyyy'
where first_name like '%a'
Returning * ;
```

veri silme

```
delete from author_3
where id>15
returning *;
```

ödev
ödev 8 (2)
Primary key ve foreign key

```
create table book (
id serial primary key,
title varchar(100),
page_number integer,
author_id integer references authora(id)
)
;
```

primary author(id) tek unique değeriki tablo arası foreign key arası ile references ile bağlantılantılır

Veri tipleri

Data Types (2)

KISITLAMALAR (constraints)

1-NOT NULL ve Alter

null değer üzerine koşul yapma

```
where username is null
```

ALTER anahtar kelimesini varolan bir tabloda değişiklik yapmak için kullanılır. Aşağıdaki senaryoda bir sütuna **NOT NULL** kısıtlaması vermek için aşağıdaki söz dizimi yapısı kullanılır.

```
ALTER TABLE <tablo_adı>
ALTER COLUMN <sütun_adı>
SET NOT NULL;
```

2-Unique

```
ALTER TABLE <tablo_adı>
ADD UNIQUE <sütun_adı>
```

3-CHECK

```
ALTER TABLE <tablo_adı>
ADD CHECK (age>=18)
```

sorularikiye bölünebilen sayıları listeler

```
where mod(ID,2) = 0
```

JOIN YAPILARI

Inner Join (kesişim alır) ikili tablo gibi düşün

```
SELECT <sütun_adı>, <sütun_adı> ...
FROM <tablo1_adı>
INNER JOIN <tablo2_adı>
ON <tablo1_adı>.<sütun_adı> = <tablo2_adı>.<sütun_adı>;
```

```
select * from book
join author on book.author_id=author_id;
```

```
select title,last_name,page_number from book
inner join author on book.author_id=author_id;
```

Left Join (ilk tabloyu alır sonra kesişimleri alır)
null değerleri olur

```
SELECT <sütun_adı>, <sütun_adı> ...
FROM <tablo1_adı>
LEFT JOIN <tablo2_adı>
ON <tablo1_adı>.<sütun_adı> = <tablo2_adı>.<sütun_adı>;
```

Right Join

```
SELECT <sütun_adı>, <sütun_adı> ...
FROM <tablo1_adı>
RIGHT JOIN <tablo2_adı>
ON <tablo1_adı>.<sütun_adı> = <tablo2_adı>.<sütun_adı>;
```

FULL Join
<aside>
💡 çok yüksek satırlar çıkarabilir dikkat edilmesi gerekir
</aside>

```
SELECT <sütun_adı>, <sütun_adı> ...
FROM <tablo1_adı>
FULL JOIN <tablo2_adı>
ON <tablo1_adı>.<sütun_adı> = <tablo2_adı>.<sütun_adı>;
```

FULL Outer Join(ortak olmayan verileri)
Union (birden fazla sıralama yapmak)
union all

```
(
select * from book
order by title
limit 5
)
Union all
(
select * from book
order by page_number desc
limit 5
);
```

Unioniki farklı tablo ise (union)sütunlar aynı sayıda ve aynı type ta olucak

```
(
select id,title from book
)
Union
(
select id,email from author
);
```

INTERSECT ve EXCEPT

sorguda kesişenleri almak istersek intersect

ilk sorguda bulunan ikinci sorguda bulunmayan verileri almak istersek except

```
(
SELECT *
FROM book
ORDER BY title
LIMIT 5
)
except
(
SELECT *
FROM book
ORDER BY page_number DESC
LIMIT 5
);
```

Subquery

birden fazla sorgunun iç içe bulunması

```
SELECT *
FROM book
WHERE page_number >
(
SELECT page_number
FROM book
WHERE title = 'Gülün Adı'
);
```

Any

Alt sorgudan gelen herhangi bir değer koşulu sağlaması durumunda `TRUE` olarak ilgili değerin koşu sağlamasını sağlar.

küçük eşittir ve büyük eşittir anlamına da gelir. (or gibi)

```
SELECT first_name, last_name,id FROM author WHERE id <Any ( SELECT id FROM book WHERE
title = 'Flying Tigers' OR title = 'Conqueror, The')
```

All

and gibi kullanılır

```
select * from book;
SELECT first_name, last_name,id FROM author WHERE id <ALL ( SELECT id FROM book WHERE
title = 'Flying Tigers' OR title = 'Conqueror, The')
```

Join and Subquery

```
SELECT author.first_name, author.last_name, book.title
FROM author
INNER JOIN book ON book.author_id = author.id
WHERE page_number >
(
  SELECT AVG(page_number) FROM book
);
```

```
SELECT actor.first_name, actor.last_name, film.title
FROM actor
JOIN film_actor ON film_actor.actor_id = actor.actor_id
JOIN film ON film.film_id = film_actor.film_id
WHERE film.length =
(
  SELECT MAX(length)  FROM film
)
```

GENEL TEKRAR

```
select count(title) from film where title ilike '%e%e%e%e';
--4 tane e harfi bulunan filmlerin sayısı

select category.name,count(*) from category
join film_category on film_category.category_id=category.category_id
join film on film.film_id=category.category_id
group by category.name;
--category isimleri ve categorylere düşen film sayısı

select rating, count(*)from film
group by rating
order by count(*) desc
limit 1;
-- en çok rating alan film sayısı

select title,length,replacement_cost from film
where title like 'K%'

order by length desc, replacement_cost asc
limit 10;
---k harfi ile başlayan en uzun ve en az rep_cost ile 10 film

select sum(amount),customer.first_name,customer.last_name from customer
```

```
join payment on payment.customer_id=customer.customer_id
group by payment.customer_id,customer.first_name,customer.last_name
order by sum(amount) desc
limit 3;
--en çok alışveriş yapan müşteri sayısı
```