

```
-- tum kayitlar
select * from sakila.film;

-- film tablosundan film_id, title, description bilgilerini listeleyin
select film_id, title, description from sakila.film;

-- film tablosundan release_year 2006 olanlari listeleyin
select * from sakila.film where release_year=2006;

-- film tablosundan orifinal_language_id deęerlerinden null olmayanları getirin
select * from sakila.film where original_language_id is not null;

-- actor tablosunda rating bir kez gorulecek sekilde listeleyin
select distinct rating from sakila.film;

-- film tablosunda ratingi G olan rental_duration 5ten kucuk olanlari getirin
select * from sakila.film where rating="G" and rental_duration <5;

-- film tablosundan title bilgisinde baş harfi L ile baslayanlari listeleyin
select * from sakila.film where title like 'L%';

-- film tablosundan title bilgisinde baş harfi L ile baslayip son harfi S ile
bitenleri listeleyin
select * from sakila.film where title like 'L%S';

-- film tablosunda title bilgisinde ilk kelimenin 4 harf ve 2.kelimenin 4 harf
olanlari listeleyin
select * from sakila.film where title like '____ ____';

-- film tablosunda length bilgisinin 80 ve 120 arasındakileri sıralı bir sekilde
listeleyin
select length from sakila.film where length < 120 and length > 80 order by length;

-- uyari : koşullu ifadelerde eşitlik durumlarını kacirmayınız...

-- film tablosunda length bilgisinin 80 ve 120 arasındakileri sıralı bir sekilde
listeleyin
select length from sakila.film where length between 80 and 120 order by length;

-- film tablosunda length bilgisini azalan olarak listeleyin
select length from sakila.film order by length desc;

-- film tablosunda rating PG olan kayitlari film_id azalan olarak siralayin
select * from sakila.film where rating='PG' order by film_id desc;

-- film tablosundan ilk 10 kaydi listeleyin
select * from sakila.film limit 10;

-- film tablosunda title,film_id ilk 10 kaydi getirin
select title,film_id from sakila.film limit 10;
```

```

-- film tablosunda length degeri en büyük olanı getirin
select * from sakila.film order by length desc limit 1;

-- film title bilgisinin ilk kelimenin 2.harfi H olanlari getirin
select * from sakila.film where title like '_H%';

-- ----- 13_1
## Gruplandırarak Sorgulama

SELECT rating FROM film group by rating;
SELECT rating,title FROM film group by rating; #istenmeyen durum - ilgili ilk kayıt
SELECT rating,count(title) FROM film group by rating;
SELECT rating,count(*),avg(replacement_cost) FROM film group by rating;
SELECT rating,count(*),avg(replacement_cost) FROM film where replacement_cost<20
group by rating;

select rating,rental_duration from film group by rating,rental_duration;
select rating,rental_duration,avg(replacement_cost) from film group by
rating,rental_duration;
select rating,rental_duration,count(rental_duration) from film group by
rating,rental_duration;

select rating,rental_duration,count(rental_duration)
from film
where rental_duration<5
group by rating,rental_duration;

select rating,rental_duration,count(rental_duration)
from film
group by rating,rental_duration
having rental_duration<5; #üsttekiyle aynı

select rating,rental_duration,count(rental_duration) as ks
from film
group by rating,rental_duration
having ks<40 and rating in ('R','G')
order by ks;

-- ----- 13_2

## Birden Fazla Tablo Üzerinde Sorgulama

#kartezyen birleştirme
select city.city,country.country from city,country;

#ilişkisel veri sütunu ile anlamlı birleştirme
select city,country from city ci,country co
where ci.country_id=co.country_id;

```

```
#yukardakine ek olarak sadece turkiyeden olanlar
select ci.city,co.country from city as ci,country as co
where ci.country_id=co.country_id and co.country='Turkey';
```

```
#şehirlerin ülke adları
SELECT city, country
FROM city
INNER JOIN country
ON city.country_id = country.country_id
ORDER BY city;
```

```
#? adında a geçen türkiyedeki şehirler ve ülke adları
SELECT city, country
FROM city
INNER JOIN country ON
city.country_id = country.country_id
where country='Turkey' and city like '%a%';
```

```
#?actor tablosunda tekrar eden soyadlar ve sayıları
select last_name,count(*) from actor group by last_name having count(*) > 1;
```

```
#ülke adları ve şehir sayıları
SELECT country, COUNT(city)
FROM country a
INNER JOIN city b
ON a.country_id = b.country_id
GROUP BY country;
```

```
-- ----- 14
## Birden Fazla Tablo Üzerinde Sorgulama2
```

```
#customer tablosundaki kayıtle aynı soyadı taşıyan aktor varsa yanında göster
SELECT
```

```
    c.customer_id as c_id,
    c.first_name as c_isim,
    c.last_name as c_soyisim,
    a.actor_id,
    a.first_name,
    a.last_name
FROM sakila.customer c
LEFT JOIN sakila.actor a
ON c.last_name = a.last_name
ORDER BY c.last_name;
```

```
#where a.actor_id is null
```

```
#actor tablosundaki kayıtle aynı soyadı taşıyan customer varsa yanında göster
SELECT
```

```
    c.customer_id as c_id,
    c.first_name as c_isim,
```

```

        c.last_name as c_soyisim,
        a.actor_id,
        a.first_name,
        a.last_name
FROM sakila.customer c
RIGHT JOIN sakila.actor a
ON c.last_name = a.last_name
ORDER BY a.last_name;
-----
select customer.last_name as c_lastname,actor.last_name as a_lastname from
sakila.actor inner join sakila.customer on actor.last_name = customer.last_name;

```

```

##filmler ve kategorileri
select  f.title,c.name
from    film f
inner join film_category fc
on f.film_id=fc.film_id
inner join category c
on c.category_id=fc.category_id;

```

```

#filmlerde en çok oynayan aktör ve oynamaya sayısı
select actor.actor_id, actor.first_name, actor.last_name,count(actor_id) as
film_count
from actor
join film_actor using (actor_id)
group by actor_id
order by film_count desc
limit 1;

```

```

# kategori adi ve ortalama film uzunluğu
select category.name, avg(length)
from film join film_category using (film_id) join category using (category_id)
group by category.name
order by avg(length) desc;

```

```

# kategori adi ve ortalama film uzunluğu (ortalaması genel film ortalamasından
fazla olanlar)
select category.name, avg(length)
from film join film_category using (film_id) join category using (category_id)
group by category.name
having avg(length) > (select avg(length) from film)
order by avg(length) desc;

```

```

#kartezyen çarpım
select city,country from city cross join country;

```

```
#ortak sütuna göre (country_id) join
alter table city drop last_update; #öncelikle ikinci bir ortak sütun olan
last_update kaldırılıyor
select * from city natural join country;#inner join gibi davranır
```

```
#self join
create database ornek;
use ornek;
create table personel(id int,adi varchar(20),yon_id int);
insert into personel values
(1,'Ali',NULL),(2,'Veli',1),(3,'Ahmet',2),(4,'Mehmet',2);
#
select p1.id,p1.adi,p2.adi as yonetici
from personel p1
left join personel p2
on p1.yon_id=p2.id
order by p1.id;
```

```
#union
create table eski_personel(id int,adi varchar(20),yon_id int);
insert into eski_personel values
(101,'Hasan',NULL),(102,'Hüseyin',101),(4,'Mehmet',2);
```

```
select * from personel
union
select * from eski_personel;
```

```
select * from personel
union all
select * from eski_personel;
```

```
select id,adi from personel
union
select yon_id,adi from eski_personel;
```

```
-- ----- 15
```

```
##Alt Sorgular
```

```
##https://www.quackit.com/mysql/examples/mysql\_subquery.cfm
#film idsi 2 olan filmde oynayan aktörlerin tüm bilgileri
SELECT * FROM actor
WHERE actor_id IN
      (SELECT actor_id FROM film_actor
       WHERE film_id = 2);
```

```
#Ace Goldfinger filmindeki aktörlerin tüm bilgileri
SELECT * FROM actor
WHERE actor_id IN
```

```
(SELECT actor_id FROM film_actor
WHERE film_id =
      (SELECT film_id FROM film
      WHERE title = 'Ace Goldfinger')
);
```

#Insert örneği

```
CREATE TABLE ace_goldfinger_actors
(first_name VARCHAR(45), last_name VARCHAR(45));
```

```
INSERT INTO ace_goldfinger_actors (first_name, last_name)
SELECT first_name, last_name FROM actor a
      INNER JOIN film_actor fa ON
      a.actor_id = fa.actor_id
      WHERE fa.film_id =
      (SELECT film_id FROM film
      WHERE title = 'Ace Goldfinger');
```

```
SELECT * FROM ace_goldfinger_actors;
```

```
DROP TABLE IF EXISTS ace_goldfinger_actors;
```

#tablo yerine alt sorgu

```
SELECT AVG(a) FROM
      (SELECT
      customer_id,
      SUM(amount) a
      FROM payment
      GROUP BY customer_id) AS totals;
```

##<http://www.mysqltutorial.org/mysql-subquery/>

#tek seferde en fazla ödemeyi yapan müşteriler

```
SELECT
      customer_id, amount
FROM
      payment
WHERE
      amount = (SELECT
      MAX(amount)
      FROM
      payment);
```

#<https://dzone.com/articles/correlated-subqueries-are-evil-and-slow-or-are-the>

#oyuncuların oynadığı film sayısı

```
SELECT
      first_name, last_name,
      (SELECT count(*)
```

```
FROM film_actor fa
WHERE fa.actor_id = a.actor_id)
FROM actor a;
```

#yukarıdaki joinle yazılmışı

```
SELECT
  first_name, last_name, count(*)
FROM actor a
JOIN film_actor fa USING (actor_id)
GROUP BY actor_id;
```

##mg

#any-all

#ratingi pg olup ratingi g olan herhangi filmde daha kısa olan film

```
SELECT * FROM film
where length < any
(select length from film where rating='G')
and rating='PG';
```

#yukarıdaki joinle yazılmışı

```
SELECT * FROM film
where length <
(select max(length) from film where rating='G')
and rating='PG';
```

#ratingi pg olup ratingi g olan tüm filmlerden daha kısa olan film

```
SELECT * FROM film
where length < all
(select length from film where rating='G')
and rating='PG';
```

#yukarıdaki joinle yazılmışı

```
SELECT * FROM film
where length <
(select min(length) from film where rating='G')
and rating='PG';
```

#exists - not exists

#2 nolu filmde oynayan aktörler(ilk örneğin aynısı)

```
SELECT * FROM actor a
where exists
(select * from film_actor fa
where fa.actor_id=a.actor_id and fa.film_id=2
);
```

#müşteriler için hiç kullanılmayan adresler

```
select * from address a where not exists(select * from customer c where
c.address_id =a.address_id);
select * from address where address_id not in (select address_id from customer)
```

