

1. **Hoeveel verschillende landen zijn er?**

```
SELECT count(country.country_id)
FROM country
```
2. **Ontdek de top 5 landen met het grootste aantal klanten.**

```
SELECT country.country, count(customer.customer_id) as
aantal
FROM country LEFT JOIN city on country.country_id =
city.country_id
LEFT JOIN address on city.city_id = address.city_id
LEFT JOIN customer on customer.address_id =
address.address_id
GROUP BY country.country
ORDER BY aantal DESC
LIMIT 5
```
3. **Wat zijn de namen van alle talen in de database (alfabetisch gesorteerd)?**

```
SELECT name FROM language
ORDER by name ASC
```
4. **Retourneer de volledige namen (eerste en laatste) van acteurs met "SON" in hun achternaam, gesorteerd op hun voornaam.**

```
SELECT distinct `actor`.`last_name`, `actor`.`first_name`
FROM `language`, `actor` where `actor`.`last_name` LIKE
'%SON' ORDER BY `actor`.`first_name`
```
5. **Maak een lijst met films en de bijbehorende categorieën.**

```
SELECT film.title, category.name
FROM film LEFT JOIN film_category ON film.film_id =
film_category.film_id
LEFT JOIN category ON category.category_id =
film_category.category_id
```
6. **Maak een lijst met categorieën en het aantal films voor elke categorie.**

```
SELECT category.name, count(film.film_id) as aantalFilms
FROM category
LEFT JOIN film_category on film_category.category_id =
category.category_id
LEFT JOIN film on film.film_id =
film_category.film_id
GROUP BY category.name
```
7. **Maak een lijst met acteurs en het aantal films van elke acteur.**

```
SELECT actor.actor_id, CONCAT(actor.first_name, '
',actor.last_name) as naam,
count(film.film_id)
FROM actor
LEFT JOIN film_actor on actor.actor_id =
film_actor.actor_id
LEFT JOIN film on film.film_id = film_actor.film_id
GROUP BY actor.actor_id
```
8. **Maak een lijst van de film_id en de titels van die films die niet in de inventaris staan.**

```
SELECT film.film_id, film.title, inventory.film_id
FROM film
LEFT JOIN inventory USING (film_id)
WHERE inventory.film_id IS NULL
```

9. Zoek een lijst met klanten die nog geen film hebben gehuurd.
`SELECT customer.first_name, customer.last_name,
rental.customer_id
FROM customer
LEFT JOIN rental USING(customer_id)
GROUP BY customer.first_name, customer.last_name`
10. Zoek het aantal Engelse films in de categorie 'Documentaire'.
`SELECT language.name, category.name, count(film.film_id)
FROM language
LEFT JOIN film on film.language_id = language.language_id
LEFT join film_category on film_category.film_id =
film.film_id
LEFT JOIN category on category.category_id =
film_category.category_id
WHERE category.name='Documentary'
GROUP BY language.name, category.name`
11. Welke acteurs hebben de voornaam 'Scarlett'
`select * from actor where first_name = 'Scarlett';`
12. Welke acteurs hebben de achternaam 'Johansson'
`select * from actor where last_name = 'Johansson';`
13. Hoeveel verschillende acteurs zijn er enkel rekening houdend met de
familiennaam?
`select count(distinct last_name) from actor`
14. Welke namen worden niet herhaald?
`select last_name
from actor
group by last_name
having count(*) = 1`
15. Welke namen verschijnen meer dan eens?
`select last_name
from actor
group by last_name
having count(*) > 1`
16. Welke acteur is in de meeste films verschenen?
`select actor.actor_id, actor.first_name, actor.last_name,
count(actor_id) as film_count
from actor
inner join film_actor using (actor_id)
group by actor_id
order by film_count desc
limit 1`
17. Is 'Academy Dinosaur' te huur bij Store 1?
JA

18. Voeg een record in om Mary Smith te vertegenwoordigen bij het huren van 'Academy Dinosaur' van Mike Hillyer in Store 1 vandaag.

```
insert into rental (rental_date, inventory_id,
customer_id, staff_id) values (NOW(), 1, 1, 1);
```

19. Wanneer is 'Academy Dinosaur' verschuldigd (=teruggave)? Geef 2 kolommen weer, nl. rental date en due date

Stap 1: wat is de duur van de huur van deze film?

```
select rental_duration from film where film_id = 1;
```

Stap 2: Kies de laatst uitgeleende rental_id.

```
select rental_id from rental order by rental_id desc
limit 1;
```

Stap 3: tel de duur van de film op bij de verhuurdatum

```
select rental_date, rental_date + interval(select
rental_duration from film where film_id = 1) day
as due_date from rental
where rental_id = (select rental_id from rental order by
rental_id desc limit 1);
```

20. Wat is de gemiddelde looptijd (duur) van alle films in de sakila DB?

```
select avg(length) from film
```

21. Wat is de gemiddelde looptijd (=duur) van films per categorie?

```
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;
```

22. Toon de voor- en achternaam van alle acteurs van de tabel acteur.

```
SELECT first_name, last_name FROM actor;
```

23. Toon de voor- en achternaam van elke acteur in één kolom in hoofdletters.

Noem de kolom acteursnaam

```
SELECT
first_name, last_name FROM actor;
```

24. Je moet het ID-nummer, de voornaam en de achternaam van een acteur vinden, van wie je alleen de voornaam kent, 'Joe'. Schrijf de select hiervoor.

```
SELECT actor_id, first_name, last_name
FROM actor
WHERE first_name = 'Joe'
```

25. Zoek alle acteurs wiens achternaam de letters GEN bevat

```
SELECT *
FROM actor WHERE last_name LIKE '%GEN%';
```

26. Zoek alle acteurs wiens achternaam de letters LI bevat. Bestel deze keer de rijen op achternaam en voornaam, in die volgorde

```
SELECT first_name, last_name
FROM actor
WHERE last_name LIKE '%LI%'
ORDER BY last_name, first_name ASC
```

27. Gebruik IN om de land_id en landkolommen van de volgende landen weer te geven: Afghanistan, Bangladesh en China

```
SELECT country_id, country
FROM country
WHERE country IN ('Afghanistan', 'Bangladesh', 'China')
```

28. Hoeveel verschillende unieke namen zijn er (actors)?

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

29. Welke namen worden niet herhaald?

```
select last_name
from actor
group by last_name
having count(*) = 1;
```

30. Welke namen verschijnen meer dan eens?

```
select last_name
from actor
group by last_name having count(*) > 1;
```

31. Welke acteur is in de meeste films verschenen?

```
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
```

32. Selecteer de meest gehuurd films omgekeerd gesorteerd

```
SELECT i.film_id, f.title, COUNT(r.inventory_id)
FROM inventory i
INNER JOIN rental r
ON i.inventory_id = r.inventory_id
INNER JOIN film_text f
ON i.film_id = f.film_id
GROUP BY r.inventory_id
ORDER BY COUNT(r.inventory_id) DESC;
```

33. Hoeveel exemplaren van de film Hunchback Impossible bestaan er?

```
SELECT title, COUNT(inventory_id)
FROM film f
INNER JOIN inventory i
ON f.film_id = i.film_id
WHERE title = "Hunchback Impossible"
```

34. Geef het totaal bedrag dat elk staflid in augustus 2005 heeft ontvangen.
`SELECT first_name, last_name, SUM(amount)`
35. Lijst het totaal op die door elke klant betaald werd, alfabetisch gesorteerd.
`SELECT last_name, first_name, SUM(amount)
FROM payment p
INNER JOIN customer c
ON p.customer_id = c.customer_id
GROUP BY p.customer_id
ORDER BY last_name ASC;`
36. Selecteer title, beschrijving, rating, film lengte en toon alle films van langer dan 3 uur.
`SELECT title,description,rating, length
FROM film
WHERE length >= 180`
37. selecteer het betalings id, bedrag en betaaldatum uitgevoerd na 27/5/2005
`SELECT payment_id, amount,payment_date
FROM payment
WHERE
payment_date >= '2005-05-27';`
38. Selecteer alle klanten waarvan de familienamen beginnen met een s en de voornamen eindigen met een n.
`SELECT *
FROM customer
WHERE last_name LIKE 'S%' AND first_name LIKE '%N'`
39. Selecteer alle niet actieve klanten waarvan de familienaam begint met een m.
`SELECT *
FROM
customer
WHERE last_name LIKE 'M%' OR active = "false"`
40. Selecteer alle kolommen van de category tabel waar de primary key groter is dan 4 en de naam begint met C of S of T
`SELECT *
FROM category
WHERE category_id > 4 AND name LIKE 'C%' OR name LIKE 'S%' OR name LIKE 'T%'`

41. Selecteer alle kolommen zonder de kolom met het paswoord van de staff tabel voor rijen die een paswoord bevatten.

```
SELECT
staff_id,
first_name,
last_name,
address_id,
email,
picture,
store_id,
active,
username,
last_update
FROM staff
WHERE
password IS NOT NULL;
```

42. Toon alle namen van acteurs die eenzelfde familienaam delen door ten minste 2 acteurs.

```
SELECT actor.last_name as naam, count(actor.actor_id) as aantal
FROM actor
GROUP BY actor.last_name
HAVING aantal >=2
ORDER BY naam;
```

43. De acteur met de naam HARPO WILLIAMS is per ongeluk in de actor tabel terechtgekomen. Zorg ervoor dat zijn naam wordt aangepast naar GROUCHO WILLIAMS

```
UPDATE actor SET last_name='GROUCHO WILLIAMS'
WHERE last_name = 'WILLIAMS' and first_name='HARPO'
```