

1) Sakira

2) SELECT * FROM actor;

SELECT last_name FROM actor WHERE first_name = 'John';

3) SELECT * FROM actor WHERE last_name = 'Neeson';

4) SELECT * FROM actor WHERE actor_id LIKE '%0';

5. What is the description of the movie with an ID of 100?

SELECT description FROM film WHERE film_id = 100;

6. Find every R-rated movie.

SELECT * FROM film WHERE rating = 'R';

7. Find every non-R-rated movie.

SELECT * FROM film WHERE rating != 'R';

8. Find the ten shortest movies.

SELECT title FROM film ORDER BY length ASC LIMIT 10;

9. Find the movies with the longest runtime, without using LIMIT

SELECT title FROM film WHERE length = (SELECT MAX(length) FROM film);

10. Find all movies that have deleted scenes.

SELECT title FROM film WHERE special_features LIKE '%deleted scenes%';

11. Using HAVING , reverse-alphabetically list the last names that are not repeated.

```
SELECT first_name, last_name FROM actor GROUP BY last_name HAVING COUNT(actor_id)=1 ORDER BY last_name DESC;
```

12. Using HAVING , list the last names that appear more than once, from highest to lowest frequency.

```
SELECT first_name, last_name FROM actor GROUP BY last_name HAVING COUNT(actor_id)>1 ORDER BY COUNT(actor_id) DESC;
```

13. Which actor has appeared in the most films?

```
SELECT * FROM actor WHERE actor_id = (  
SELECT actor_id FROM film_actor GROUP BY actor_id ORDER BY COUNT(actor_id) DESC LIMIT 1);
```

14. When is 'Academy Dinosaur' due?

```
SELECT title, release_year FROM film WHERE title = 'Academy Dinosaur';
```

15. What is the average runtime of all films?

```
SELECT AVG(length) FROM film;
```

16. List the average runtime for every film category.

```
SELECT category.name AS category, AVG(length) AS avg_runtime  
FROM film  
JOIN  
film_category ON film.film_id = film_category.film_id  
JOIN category ON category.category_id = film_category.category_id  
GROUP BY film_category.category_id;
```

17. List all movies featuring a robot.

```
SELECT title, description WHERE film.title LIKE '%robot%;
```

18. How many movies were released in 2010?

```
SELECT COUNT(film_id) AS FilmsReleasedIn2010 FROM film WHERE film.release_year = '2010';
```

19. Find the titles of all the horror movies.

```
SELECT title FROM film
```

```
JOIN film_category ON film_category.film_id = film.film_id
```

```
JOIN category ON film_category.category_id = category.category_id
```

```
WHERE category.name = 'Horror';
```

20. List the full name of the staff member with the ID of 2.

```
SELECT staff_id, first_name, last_name FROM staff WHERE staff.staff_id = '2';
```

21. List all the movies that Fred Costner has appeared in.

```
SELECT title FROM film WHERE film_id IN (
```

```
SELECT film_id FROM film_actor WHERE actor_id = (
```

```
SELECT actor_id FROM actor
```

```
WHERE first_name = 'Fred' AND last_name = 'Costner' ));
```

22. How many distinct countries are there?

```
SELECT COUNT(DISTINCT country) FROM country;
```

23. List the name of every language in reverse-alphabetical order.

```
SELECT name FROM language ORDER BY name DESC;
```

24. List the full names of every actor whose surname ends with '-son' in alphabetical order by their forename.

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

25. Which category contains the most films?

```
SELECT category.name, COUNT(film_category.film_id)  
AS numberOfFilms FROM film_category JOIN category  
ON film_category.category_id = category.category_id  
GROUP BY film_category.category_id  
ORDER BY COUNT(film_id) DESC LIMIT 1;
```

```
*****  
*****
```

World.

1. Using COUNT , get the number of cities in the USA.

```
SELECT COUNT(*) FROM city WHERE CountryCode = 'USA';
```

2. Find out the population and life expectancy for people in Argentina.

SELECT Name, population AND LifeExpectancy FROM country WHERE Name = 'Argentina';

3. Using IS NOT NULL , ORDER BY , and LIMIT , which country has the highest life expectancy?

SELECT Name, LifeExpectancy FROM country WHERE LifeExpectancy IS NOT NULL ORDER BY Population DESC LIMIT 1;

4. Using JOIN ... ON , find the capital city of Spain.

SELECT country.Name AS Country, city.Name AS Capital
FROM city JOIN country ON country.capital = city.ID WHERE country.Name = 'Spain';

5. Using JOIN ... ON , list all the languages spoken in the Southeast Asia region.

SELECT COUNT(countryLanguage.Language) AS LanguagesNumber FROM countryLanguage JOIN
country ON country.code = countryLanguage.CountryCode WHERE Region = 'Southeast Asia';

6. Using a single query, list 25 cities around the world that start with the letter F.

SELECT Name FROM city WHERE Name LIKE 'F%' LIMIT 25;

7. Using COUNT and JOIN ... ON , get the number of cities in China.

SELECT COUNT(ID) FROM city JOIN country ON city.CountryCode = country.Code WHERE
country.Name = 'China';

8. Using IS NOT NULL , ORDER BY , and LIMIT , which country has the lowest population? Discard non-zero populations.

```
SELECT Name FROM country WHERE Population IS NOT NULL AND Population != 0 ORDER BY Population ASC LIMIT 1;
```

9. Using aggregate functions, return the number of countries the database contains.

```
SELECT COUNT(*) FROM country AS NumberOfCountries;
```

10. What are the top ten largest countries by area?

```
SELECT Name, SurfaceArea FROM country ORDER BY SurfaceArea DESC LIMIT 10;
```

11. List the five largest cities by population in Japan

```
SELECT NAME, Population FROM city WHERE Countrycode = (SELECT code FROM country WHERE Name = 'Japan') ORDER BY Population DESC LIMIT 5;
```

2. List the names and country codes of every country with Elizabeth II as its Head of State. You will need to fix the mistake first!

```
UPDATE country SET HeadOfState = 'Elizabeth II' WHERE HeadOfState = 'Elisabeth II';
```

```
SELECT Name, Code FROM country WHERE HeadOfState = 'Elizabeth II';
```

13. List the top ten countries with the smallest population-to-area ratio. Discard any countries with a ratio of 0.

```
SELECT Name, Population/SurfaceArea FROM country WHERE Population!= 0 ORDER BY Population/SurfaceArea ASC LIMIT 10;
```

14. List every unique world language.

```
SELECT DISTINCT language FROM countrylanguage;
```

15. List the names and GNP of the world's top 10 richest countries.

```
SELECT Name, GNP FROM country ORDER BY GNP DESC LIMIT 10;
```

16. List the names of, and number of languages spoken by, the top ten most multilingual countries.

```
SELECT country.Name, COUNT(countrylanguage.Language) FROM country
JOIN countrylanguage ON country.code = countrylanguage.CountryCode
GROUP BY country.Name ORDER BY COUNT(countrylanguage.Language) DESC LIMIT 10;
```

17. List every country where over 50% of its population can speak German.

```
SELECT country.Name FROM country
JOIN countrylanguage ON country.Code = countrylanguage.Countrycode
WHERE Language= 'German' AND percentage>50;
```

18. Which country has the worst life expectancy? Discard zero or null values.

```
SELECT Name, LifeExpectancy FROM country WHERE LifeExpectancy IS NOT NULL AND
LifeExpectancy != 0
ORDER BY LifeExpectancy ASC LIMIT 1;
```

19. List the top three most common government forms.

```
SELECT GovernmentForm, COUNT(Code) AS GlobalFrequency FROM country
GROUP BY GovernmentForm
ORDER BY GlobalFrequency DESC LIMIT 3;
```

20. How many countries have gained independence since records began?

```
SELECT COUNT(IndepYear) FROM country;
```

```
*****
*****
```

1. List the titles and release_dates of movies released between 1983-1993 in reverse chronological order.

```
SELECT title, release_date FROM movies WHERE release_date BETWEEN '1983-01-01' AND '1993-12-31'
```

ORDER BY release_date ASC;

2. Without using LIMIT , list the titles of the movies with the lowest average rating.

```
SELECT Q.name, Q.avg FROM
(SELECT movies.title AS name, AVG(ratings.rating) AS avg FROM ratings
JOIN movies ON movies.id = ratings.movie_id
GROUP BY ratings.movie_id) AS Q
WHERE Q.avg = (SELECT MIN(y.avg) FROM (SELECT AVG(ratings.rating) AS avg FROM ratings
GROUP BY ratings.movie_id) AS y);
```

3. List the unique records for Sci-Fi movies where male 24-year-old students have given 5-star ratings.

```
SELECT title FROM movies
WHERE id IN (
    SELECT movie_id FROM ratings
    WHERE user_id IN (
        SELECT id FROM users
        WHERE occupation_id = (
            SELECT id FROM occupations
            WHERE name = 'student')
        AND age = 24
        AND gender = 'M')
    AND rating = 5)
AND id IN (
    SELECT movie_id FROM genres_movies WHERE genre_id = (SELECT id FROM genres WHERE name
= 'Sci-Fi'));
```

4. List the unique titles of each of the movies released on the most popular release day.


```
SELECT DISTINCT title FROM movies WHERE release_date = (  
SELECT release_date FROM movies  
GROUP BY release_date ORDER BY COUNT(id) DESC LIMIT 1);
```

5. Find the total number of movies in each genre; list the results in ascending numeric order.

```
SELECT genres.name AS genre, COUNT(genres_movies.movie_id) FROM genres_movies
```

```
JOIN genres ON genres.id = genres_movies.genre_id
```

```
GROUP BY genres_movies.genre_id
```

```
ORDER BY COUNT(genres_movies.movie_id) ASC;
```

```
SELECT genre_id, COUNT(movies_id) FROM genres_movies GROUP BY genre_id;
```