

Q1) Which categories of movies released in 2018? Fetch with the number of movies.

Soln.

Part 1 :-

```
SELECT title
FROM film
WHERE release_year = '2018';
```

Part 2 :-

```
SELECT COUNT(film_id) AS total_movies
FROM film
WHERE release_year = '2018';
```

Q2) Update the address of actor id 36 to “677 Jazz Street”.

Soln.

```
UPDATE address
  LEFT JOIN actor ON address.address_id = actor.address_id
SET address.address = '677 Jazz Street'
WHERE actor.actor_id = 36;
```

Q3) Add the new actors (id : 105 , 95) in film ARSENIC INDEPENDENCE (id:41).

Soln.

```
INSERT INTO film_actor(actor_id, film_id)
VALUES (105, 41),
      (95, 41);
```

Q4) Get the name of films of the actors who belong to India.

Soln.

```
SELECT DISTINCT(f.title)
FROM film f
    INNER JOIN film_actor fa ON f.film_id = fa.film_id
    INNER JOIN actor a ON fa.actor_id = a.actor_id
    INNER JOIN address ad ON a.address_id = ad.address_id
    INNER JOIN city c ON ad.city_id = c.city_id
    INNER JOIN country ct ON c.country_id = ct.country_id
WHERE country = 'India';
```

Q5)How many actors are from the United States?

Soln.

```
SELECT COUNT(actor_id) AS total_count
FROM actor a
    INNER JOIN address ad ON a.address_id = ad.address_id
    INNER JOIN city c ON ad.city_id = c.city_id
    INNER JOIN country ct ON c.country_id = ct.country_id
WHERE country = 'United States';
```

Q6)Get all languages in which films are released in the year between 2001 and 2010.

Soln.

```
SELECT DISTINCT l.name
FROM language l
    INNER JOIN film f ON l.language_id = f.language_id
WHERE f.release_year BETWEEN '2001' AND '2010';
```

Q7)The film ALONE TRIP (id:17) was actually released in Mandarin, update the info.

Soln.

```
UPDATE film f
    INNER JOIN language l ON f.language_id = l.language_id
SET l.name = 'Mandarin'
WHERE title = 'ALONE TRIP'
    AND f.film_id = 17;
```

Q8)Fetch cast details of films released during 2005 and 2015 with PG rating.
Soln.

```
SELECT a.actor_id,
    CONCAT(a.first_name, ' ', a.last_name) AS full_name,
    f.rating
FROM actor a
    INNER JOIN film_actor fa ON a.actor_id = fa.actor_id
    INNER JOIN film f ON fa.film_id = f.film_id
WHERE f.release_year BETWEEN 2005 AND 2015
    AND rating = 'PG';
```

Q9)In which year most films were released?
Soln.

```
SELECT release_year,
    COUNT(release_year) AS total_count
FROM film
GROUP BY release_year
ORDER BY 2 DESC
LIMIT 1;
```

Q10)In which year least number of films were released?
Soln.

```
SELECT release_year,
       COUNT(release_year) AS total_count
FROM film
GROUP BY release_year
ORDER BY 2
LIMIT 1;
```

Q11)Get the details of the film with maximum length released in 2014 .
Soln.

```
SELECT film_id,
       title,
       release_year,
       length
FROM film
WHERE length = (
    SELECT MAX(length)
    FROM film
)
AND release_year = 2014;
```

Q12)Get all Sci-Fi movies with NC-17 ratings and language they are screened in.
Soln.

```
SELECT DISTINCT f.film_id,
       f.title,
       l.name,
       c.name,
       f.rating
FROM film f
    INNER JOIN language l ON f.language_id = l.language_id
    INNER JOIN film_category fg ON f.film_id = fg.film_id
    INNER JOIN category c ON fg.category_id = c.category_id
WHERE f.rating = 'NC-17'
    AND c.name = 'Sci-Fi';
```

Q13)The actor FRED COSTNER (id:16) shifted to a new address:055, Piazzale Michelangelo, Postal Code-50125 , District-Rifredi at Florence, Italy
Insert the new city and update the address of the actor.
Soln.

```
INSERT INTO city (city, country_id)
VALUES (
    'Florence',
    (
        SELECT country_id
        FROM country
        WHERE country = 'Italy'
    )
);
UPDATE address a
    INNER JOIN actor ac ON a.address_id = ac.address_id
SET a.address = '055, Piazzale Michelangelo',
    a.postal_code = '50125',
    a.district = 'Rifredi',
    a.city = (
        SELECT city_id
        FROM city
        WHERE city = 'Florence'
    )
WHERE ac.actor_id = 16;
```

Q14)A new film “No Time to Die” is releasing in 2020 whose details are :
Title :-No Time to Die
Description: Recruited to rescue a kidnapped scientist, globe-trotting spy James Bond finds himself hot on the trail of a mysterious villain, who's armed with a dangerous new technology.
Language: English
Length : 100

Rental duration : 6

Rental rate : 3.99

Rating : PG-13

Replacement cost : 20.99

Special Features = Trailers, Deleted Scenes

Insert the above data.

Soln.

```
INSERT INTO film (
    title,
    release_year,
    description,
    language_id,
    original_language_id,
    length,
    rental_duration,
    rental_rate,
    rating,
    replacement_cost,
    special_features
)
VALUES (
    'No Time to Die',
    2020,
    "Recruited to rescue a kidnapped scientist, globe-trotting spy James Bond finds himself hot on the trail of a mysterious villain, who's armed with a dangerous new technology.",
    (
        SELECT language_id
        FROM language
        WHERE name = 'English'
    ),
    (
        SELECT language_id
        FROM language
        WHERE name = 'English'
    ),
    100,
    6,
    3.99,
    'PG-13',
```

```
20.99,  
    'Trailers,Deleted Scenes'  
);
```

Q15)Assign the category Action, Classics, Drama to the movie “No Time to Die” .
Soln.

```
INSERT INTO film_category (film_id, category_id)  
VALUES (  
    (  
        SELECT film_id  
        FROM film  
        WHERE title = 'No Time to Die'  
    ),  
    (  
        SELECT category_id  
        FROM category  
        WHERE name = 'Action'  
    )  
),  
(  
    (  
        SELECT film_id  
        FROM film  
        WHERE title = 'No Time to Die'  
    ),  
    (  
        SELECT category_id  
        FROM category  
        WHERE name = 'Classics'  
    )  
),  
(  
    (  
        SELECT film_id  
        FROM film  
        WHERE title = 'No Time to Die'  
    ),  
    (  
        SELECT category_id
```

```
        FROM category
        WHERE name = 'Drama'
    )
)
```

Q16)Assign the cast: PENELOPE GUINNESS, NICKWAHLBERG, JOE SWANK to the movie “NoTime to Die” .

Soln.

```
INSERT INTO `film_actor`(`actor_id`,`film_id`)
VALUES (
    (
        SELECT actor_id
        FROM actor
        WHERE actor.first_name = "PENELOPE"
            AND actor.last_name = "GUINNESS"
    ),
    (
        SELECT film_id
        FROM film
        WHERE film.title = "No Time to Die"
    )
),
(
    (
        SELECT actor_id
        FROM actor
        WHERE actor.first_name = "NICK"
            AND actor.last_name = "WAHLBERG"
    ),
    (
        SELECT film_id
        FROM film
        WHERE film.title = "No Time to Die"
    )
),
(
    (
        SELECT actor_id
```



```

        FROM actor
        WHERE actor.first_name = "JOE"
              AND actor.last_name = "SWANK"
    ),
    (
        SELECT film_id
        FROM film
        WHERE film.title = "No Time to Die"
    )
);

```

Q17) Assign a new category Thriller to the movie ANGELS LIFE.
Soln.

```

INSERT INTO category (name)
VALUES ('Thriller');
INSERT INTO film_category (film_id, category_id)
VALUES (
    (
        SELECT film_id
        FROM film
        WHERE title = 'ANGELS LIFE'
    ),
    (
        SELECT category_id
        FROM category
        WHERE name = 'Thriller'
    )
);

```

Q18) Which actor acted in most movies?
Soln.

```

SELECT a.actor_id,
       a.first_name,
       a.last_name,
       COUNT(a.actor_id) AS total_movies_acted
FROM actor a

```

```

    INNER JOIN film_actor fa ON a.actor_id = fa.actor_id
GROUP BY 1,
    2,
    3
ORDER BY 4 DESC
LIMIT 1;

```

Q19)The actor JOHNNY LOLLOBRIGIDA was removed from the movie GRAIL FRANKENSTEIN.How would you update that record?
Soln.

```

DELETE FROM film_actor
WHERE actor_id = (
    SELECT actor_id
    FROM actor
    WHERE first_name = 'JOHNNY'
        AND last_name = 'LOLLOBRIGIDA'
)
AND film_id = (
    SELECT film_id
    FROM film
    WHERE title = 'GRAIL FRANKENSTEIN'
);

```

Q20)The HARPER DYING movie is an animated movie with Drama and Comedy.
Assign these categories to the movie.
Soln.

```

INSERT INTO film_category(film_id, category_id)
VALUES (
    (
        SELECT film_id
        FROM film
        WHERE title = 'HARPER DYING'
    ),
    (

```

```

        SELECT category_id
        FROM category
        WHERE name = 'Drama'
    )
),
(
    (
        SELECT film_id
        FROM film
        WHERE title = 'HARPER DYING'
    ),
    (
        SELECT category_id
        FROM category
        WHERE name = 'Comedy'
    )
);

```

Q21) The entire cast of the movie WEST LION has changed. The new actors are DAN TORN, MAE HOFFMAN, SCARLETT DAMON. How would you update the record in the safest way?

Soln.

```

DELETE FROM film_actor
WHERE film_id = (
    SELECT film_id
    FROM film
    WHERE title = 'WEST LION'
);

INSERT INTO film_actor(actor_id, film_id)
VALUES (
    (
        SELECT actor_id
        FROM actor
        WHERE first_name = 'DAN'
            AND last_name = 'TORN'
    ),
    (
        SELECT film_id

```

```

        FROM film
        WHERE title = 'WEST LION'
    )
),
(
    (
        SELECT actor_id
        FROM actor
        WHERE first_name = 'MAE'
            AND last_name = 'HOFFMAN'
    ),
    (
        SELECT film_id
        FROM film
        WHERE title = 'WEST LION'
    )
),
(
    (
        SELECT actor_id
        FROM actor
        WHERE first_name = 'SCARLETT'
            AND last_name = 'DAMON'
    ),
    (
        SELECT film_id
        FROM film
        WHERE title = 'WEST LION'
    )
);

```

Q22) The entire category of the movie WEST LION was wrongly inserted. The correct categories are Classics, Family, Children. How would you update the record ensuring no wrong data is left?

Soln.

```

DELETE FROM film_category
WHERE film_id = (
    SELECT film_id
    FROM film

```

```
WHERE title = 'WEST LION'

);

INSERT INTO film_category (film_id, category_id)
VALUES (
    (
        SELECT film_id
        FROM film
        WHERE title = 'WEST LION'
    ),
    (
        SELECT category_id
        FROM category
        WHERE name = 'Classics'
    )
),
(
    (
        SELECT film_id
        FROM film
        WHERE title = 'WEST LION'
    ),
    (
        SELECT category_id
        FROM category
        WHERE name = 'Family'
    )
),
(
    (
        SELECT film_id
        FROM film
        WHERE title = 'WEST LION'
    ),
    (
        SELECT category_id
        FROM category
        WHERE name = 'Children'
    )
);
```

Q23)How many actors acted in films released in 2017?

Soln.

```
SELECT COUNT(fa.actor_id) AS total_count
FROM film f
    INNER JOIN film_actor fa ON f.film_id = fa.film_id
WHERE release_year = 2017;
```

Q24)How many Sci-Fi films released between the year 2007 to 2017?

Soln.

```
SELECT COUNT(f.film_id) AS total_movies_released
FROM film f
    INNER JOIN film_category fg ON f.film_id = fg.film_id
    INNER JOIN category c ON fg.category_id = c.category_id
WHERE c.name = 'Sci-Fi'
    AND f.release_year BETWEEN 2007 AND 2017;
```

Q25)Fetch the actors of the movie WESTWARD SEABISCUIT with the city they live in.

Soln.

```
SELECT f.title,
    CONCAT(a.first_name, ' ', a.last_name) AS full_name,
    c.city
FROM film f
    INNER JOIN film_actor fa ON f.film_id = fa.film_id
    INNER JOIN actor a ON fa.actor_id = a.actor_id
    INNER JOIN address ad ON a.address_id = ad.address_id
    INNER JOIN city c ON ad.city_id = c.city_id
WHERE f.title = 'WESTWARD SEABISCUIT';
```

Q26)What is the total length of all movies played in 2008?

Soln.

```
SELECT SUM(length) AS total_length
FROM film
WHERE release_year = 2008;
```

Q27) Which film has the shortest length? In which language and year was it released?

Soln.

```
SELECT f.title,
       l.name,
       f.release_year,
       f.length
FROM film f
     INNER JOIN language l ON f.language_id = l.language_id
WHERE f.length = (
    SELECT MIN(length)
    FROM film
);
```

Q28) How many movies were released each year?

Soln.

```
SELECT release_year,
       COUNT(film_id) AS movies_released_yearly
FROM film
GROUP BY 1
ORDER BY 1;
```

Q29) How many languages of movies were released each year?

Soln.

```
SELECT release_year,
       language.name,
       COUNT(film.language_id) AS number_of_films
```

```
FROM `film`  
    INNER JOIN language ON language.language_id = film.language_id  
GROUP BY 1,  
    film.language_id  
ORDER BY 1;
```

Q30) Which actor did least movies?

Soln.

```
SELECT a.actor_id,  
    a.first_name,  
    a.last_name,  
    COUNT(a.actor_id) AS total_movies_acted  
FROM actor a  
    INNER JOIN film_actor fa ON a.actor_id = fa.actor_id  
GROUP BY 1,  
    2,  
    3  
ORDER BY 4  
LIMIT 1;
```