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 GUINESS, 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 = "GUINESS"

),

(

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 movieWEST 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;