

Course: MySQL

Difficulty: Intense

Instruction: Import the given **lco_films.sql** file in phpmyadmin and then answer the following questions with mySQL queries.

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

Ans: `SELECT category.name, category.category_id, film.release_year, film.film_id,
COUNT(category.category_id) AS number_of_films FROM category LEFT JOIN film_category ON
film_category.category_id = category.category_id RIGHT JOIN film ON film.film_id =
film_category.film_id WHERE film.release_year = "2018" GROUP BY category.category_id;`

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

Ans: `UPDATE `address` INNER JOIN actor ON actor.address_id=address.address_id SET
`address`="677 Jazz Street" WHERE actor.actor_id=36`

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

Ans: `INSERT INTO `film_actor`(`actor_id`, `film_id`) VALUES (115,41) , (85,41) ON DUPLICATE
KEY UPDATE film_id = VALUES(film_id) , actor_id = VALUES(actor_id);`

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

Ans: `SELECT DISTINCT film.title FROM film INNER JOIN film_actor ON film_actor.film_id =
film.film_id INNER JOIN actor ON actor.actor_id = film_actor.actor_id INNER JOIN address on
address.address_id = actor.address_id INNER JOIN city ON city.city_id = address.city_id
INNER JOIN country ON country.country_id=city.country_id WHERE country.country = "India";`

Q5) How many actors are from the United States?

Ans: `SELECT COUNT(*) FROM actor INNER JOIN address ON address.address_id = actor.address_id INNER JOIN city ON city.city_id = address.city_id INNER JOIN country ON country.country_id=city.country_id WHERE country.country = "United States";`

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

Ans: `SELECT language.name,film.release_year, COUNT(language.language_id) AS number_of_films FROM language LEFT JOIN film ON film.language_id = language.language_id WHERE film.release_year BETWEEN 2001 AND 2010 GROUP BY language.language_id;`

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

Ans: `UPDATE `film` SET language_id = (SELECT language.language_id FROM language WHERE language.name = "English") WHERE film.film_id = 17;`

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

Ans: `SELECT CONCAT(actor.first_name, " ", actor.last_name) AS actor_name,`

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

Ans: `SELECT CONCAT(actor.first_name, " ", actor.last_name) AS actor_name,
film.title,film.release_year,film.rating FROM actor INNER JOIN film_actor ON
film_actor.actor_id = actor.actor_id INNER JOIN film ON film.film_id = film_actor.film_id
WHERE film.release_year BETWEEN 2005 AND 2015 AND film.rating = "PG";`

Q9) In which year most films were released?

Ans: `SELECT film.title,film.release_year, COUNT(film.release_year) AS no_of_films FROM
`film` GROUP BY (film.release_year) ORDER BY COUNT(film.release_year) DESC LIMIT 1;`

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

Ans: `SELECT film.title,film.release_year, COUNT(film.release_year) AS no_of_films FROM
`film` GROUP BY (film.release_year) ORDER BY COUNT(film.release_year) ASC LIMIT 1;`

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

Ans: `SELECT *, language.name as language FROM `film` LEFT JOIN language ON
language.language_id = film.language_id WHERE film.release_year = "2014" AND film.length =
(SELECT MAX(film.length) FROM film);`

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

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

Ans: `SELECT film.film_id, film.title, film.description, film.release_year, film.length,
film.rating,category.name, language.name FROM film LEFT JOIN language ON
language.language_id=film.language_id RIGHT JOIN film_category ON film_category.film_id =
film.film_id LEFT JOIN category ON category.category_id=film_category.category_id WHERE
film.rating="NC-17" AND category.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.

Ans: Follow the steps:

S1) `INSERT INTO `city`(`city`, `country_id`) VALUES ("Florence",(SELECT country_id FROM
country WHERE country.country = "Italy"));`

S2) `UPDATE `address` INNER JOIN actor ON actor.address_id = address.address_id SET
address.address = "055, Piazzale Michelangelo", address.district = "Rifredi ", address.city_id
= (SELECT city_id FROM city WHERE city.city = "Florence") , address.postal_code = "50125"
WHERE actor.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

Org. 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.

Ans: INSERT INTO `film`(`title`, `description`, `release_year`, `language_id`,
`original_language_id`, `rental_duration`, `rental_rate`, `length`, `replacement_cost`, `rating`,
`special_features`) VALUES ("No Time to Die", "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.", 2020 , (SELECT language.language_id FROM

language WHERE language.name = "English"), (SELECT language.language_id FROM language
WHERE language.name = "English"), 6, 3.99, 100, 20.99, "PG-13", "Trailers,Deleted Scenes");

Q15) Assign the category Action, Classics, Drama to the movie "No Time to Die".

Ans: `INSERT INTO `film_category`(`category_id`, `film_id`) VALUES ((SELECT category_id FROM category WHERE category.name = "Action"), (SELECT film_id FROM film WHERE film.title = "No Time to Die")), ((SELECT category_id FROM category WHERE category.name = "Classics"), (SELECT film_id FROM film WHERE film.title = "No Time to Die")), ((SELECT category_id FROM category WHERE category.name = "Drama"), (SELECT film_id FROM film WHERE film.title = "No Time to Die"));`

Q16) Assign the cast: PENELOPE GUINNESS, NICK WAHLBERG, JOE SWANK to the movie "No Time to Die".

Ans: `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.

Ans: Create a new category and then insert data into pivot table:-

S1) `INSERT INTO `category`(`name`) VALUES ("Thriller");`

S2) `INSERT INTO `film_category`(`film_id`, `category_id`) VALUES ((SELECT film_id FROM film WHERE film.title= "ANGELS LIFE"), (SELECT category_id FROM category WHERE category.name = "Thriller"));`

Q18) Which actor acted in most movies?

Q18) Which actor acted in most movies?

Ans: `SELECT actor_id, COUNT(actor_id) FROM `film_actor` GROUP BY actor_id ORDER BY COUNT(actor_id) DESC LIMIT 1;`

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

Ans: `DELETE FROM `film_actor` WHERE film_id = (SELECT film_id FROM film WHERE film.title = "GRAIL FRANKENSTEIN") AND actor_id = (SELECT actor_id FROM actor WHERE actor.first_name = "JOHNNY" AND actor.last_name="LOLLOBRIGIDA");`

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

Ans: `INSERT INTO `film_category`(`category_id`, `film_id`) VALUES ((SELECT category_id FROM category WHERE category.name="Drama"), (SELECT film_id FROM film WHERE film.title ="HARPER DYING")), ((SELECT category_id FROM category WHERE category.name="Comedy"), (SELECT film_id FROM film WHERE film.title ="HARPER DYING"))
ON DUPLICATE KEY UPDATE film_id = VALUES(film_id) , category_id = VALUES(category_id);`

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?

Ans: Delete all and insert the new data to ensure no previous data is present.

S1) DELETE FROM `film_actor` WHERE film_id = (SELECT film_id FROM film WHERE film.title = "WEST LION");

S2) INSERT INTO `film_actor`(`actor_id`, `film_id`) VALUES ((SELECT actor_id FROM actor WHERE actor.first_name = "DAN" AND actor.last_name="TORN"), (SELECT film_id FROM film WHERE film.title="WEST LION")), ((SELECT actor_id FROM actor WHERE actor.first_name = "MAE" AND actor.last_name="HOFFMAN"), (SELECT film_id FROM film WHERE film.title="WEST LION")), ((SELECT actor_id FROM actor WHERE actor.first_name = "SCARLETT" AND actor.last_name="DAMON"), (SELECT film_id FROM film WHERE film.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?

Ans: Delete all and insert the new data to ensure no previous data is present.

S1) DELETE FROM `film_category` WHERE film_id = (SELECT film_id FROM film WHERE film.title = "WEST LION");

S2) INSERT INTO `film_category`(`category_id`, `film_id`) VALUES ((SELECT category_id

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?

Ans: Delete all and insert the new data to ensure no previous data is present.

```
S1) DELETE FROM `film_category` WHERE film_id = (SELECT film_id FROM film WHERE
film.title = "WEST LION");
S2) INSERT INTO `film_category`(`category_id`, `film_id`) VALUES ((SELECT category_id
FROM category WHERE category.name = "Classics"), (SELECT film_id FROM film WHERE
film.title="WEST LION")), ((SELECT category_id FROM category WHERE category.name =
"Family"), (SELECT film_id FROM film WHERE film.title="WEST LION")), ((SELECT category_id
FROM category WHERE category.name = "Children"), (SELECT film_id FROM film WHERE
film.title="WEST LION"));
```

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

Ans: SELECT COUNT(*) FROM `film_actor` INNER JOIN film ON film.film_id=film_actor.film_id
WHERE film.release_year = 2017;

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

Ans: SELECT COUNT(*) FROM `film_category` INNER JOIN film ON

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

Ans: `SELECT COUNT(*) FROM `film_category` INNER JOIN film ON
film.film_id=film_category.film_id INNER JOIN category ON
category.category_id=film_category.category_id WHERE film.release_year BETWEEN 2007
AND 2017 AND category.name="Sci-Fi";`

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

Ans: `SELECT actor.first_name ,actor.last_name , film.title , city.city FROM film_actor INNER
JOIN actor ON actor.actor_id = film_actor.actor_id INNER JOIN address ON
address.address_id = actor.address_id INNER JOIN city ON city.city_id = address.address_id
INNER JOIN film ON film.film_id = film_actor.film_id WHERE film.title="WESTWARD
SEABISCUIT";`

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

Ans: `SELECT SUM(film.length) FROM `film` WHERE film.release_year = "2008"`

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

Ans: `SELECT language.name, film.title, film.release_year, film.length FROM `film` LEFT JOIN
language ON language.language_id=film.language_id WHERE film.length = (SELECT
MIN(film.length) FROM film);`

Q28) How many movies were released each year?

Ans: `SELECT film.release_year, COUNT(film.release_year) AS number_of_films FROM `film`
GROUP BY film.release_year;`

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

Ans: `SELECT language.name, COUNT(film.language_id) AS number_of_films FROM `film`
INNER JOIN language ON language.language_id = film.language_id GROUP BY
film.language_id;`

Q30) Which actor did least movies?

Ans: `SELECT actor_id, COUNT(actor_id) FROM `film_actor` GROUP BY actor_id ORDER BY
COUNT(actor_id) ASC LIMIT 1;`