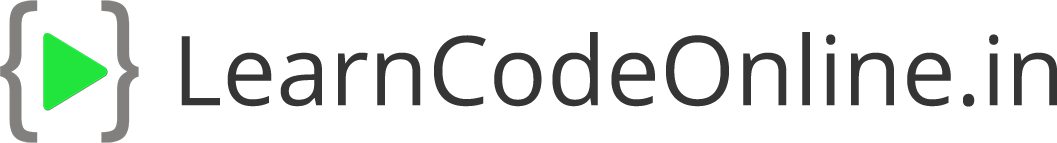
Visit our website at [https://web.learncodeonline.in](https://web.learncodeonline.in/) for more exciting courses.

**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, 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.

**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 GUINESS, 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 = "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.

**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?

**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 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 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 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;**