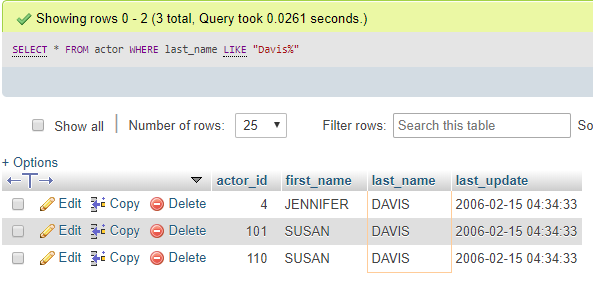
Part (a)

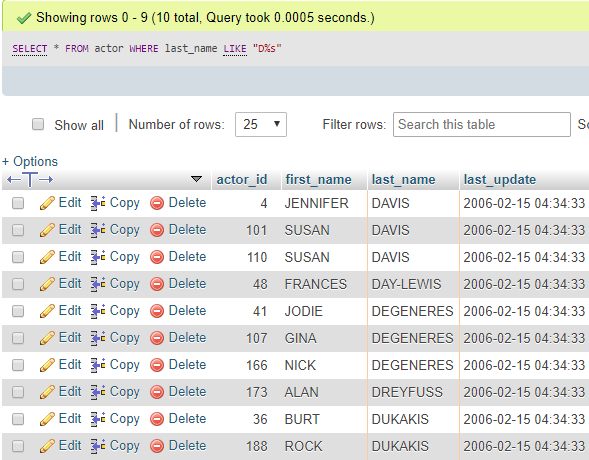
|  |  |  |
| --- | --- | --- |
|  | Table | Description & describe possible Use of table |
| 1 | film\_category | Contains the Category of films in dvd database, can be used to search the film which the user wants by category |
| 2 | rental | Contains rental information about dvd rentals such as date of issue, date of return. It can be used for renting dvd to user |
| 3 | store | Contains information about store such as address, staff, manager. It can be used as a store locator for customers |
| 4 | film\_actor | Contains information about film and actor who has acted in that film. Can be used to identify latest and best films of a particular actor. |
| 5 | inventory | Contains information about store,and films present in that store. Can be used to check remaining stock of films at any store. |

Part (b) Paste Screenshot here

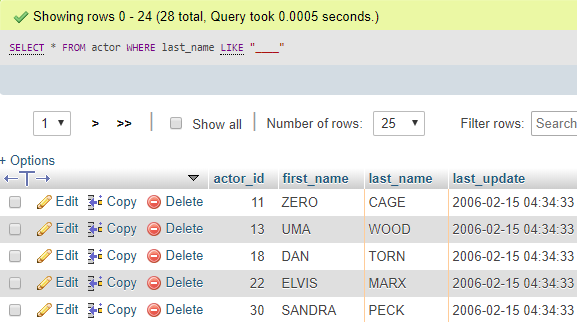
1. [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM actor WHERE last\_name [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) "Davis%"



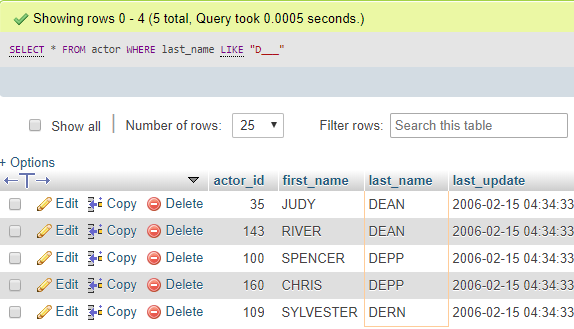
1. [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM actor WHERE last\_name [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) "D%s"



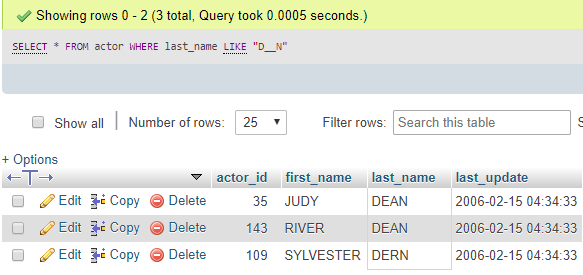
1. [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) \* FROM actor WHERE last\_name [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-comparison-functions.html#operator_like) "\_\_\_\_"



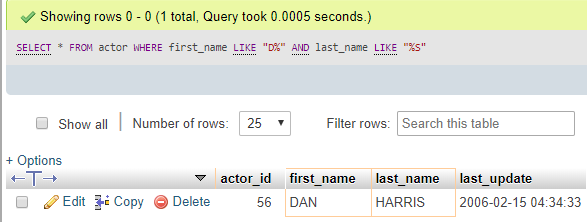
1. SELECT \* FROM actor WHERE last\_name LIKE "D\_\_\_"



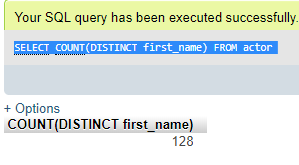
1. SELECT \* FROM actor WHERE last\_name LIKE "D\_\_N"



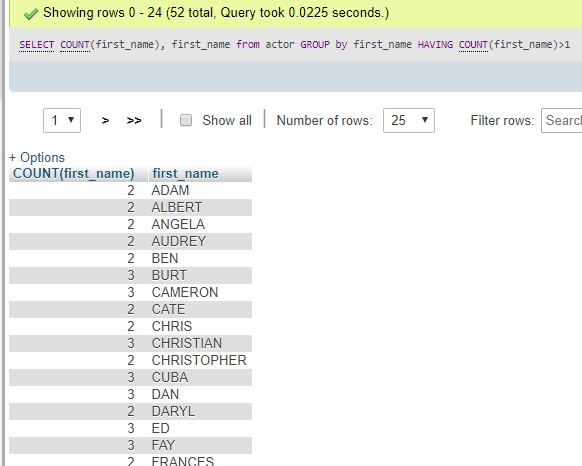
1. SELECT \* FROM actor WHERE first\_name LIKE "D%" AND last\_name LIKE "%S"



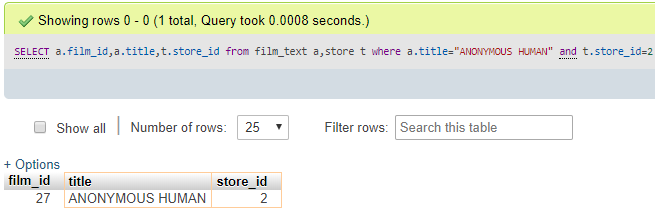
1. SELECT COUNT(DISTINCT first\_name) FROM actor



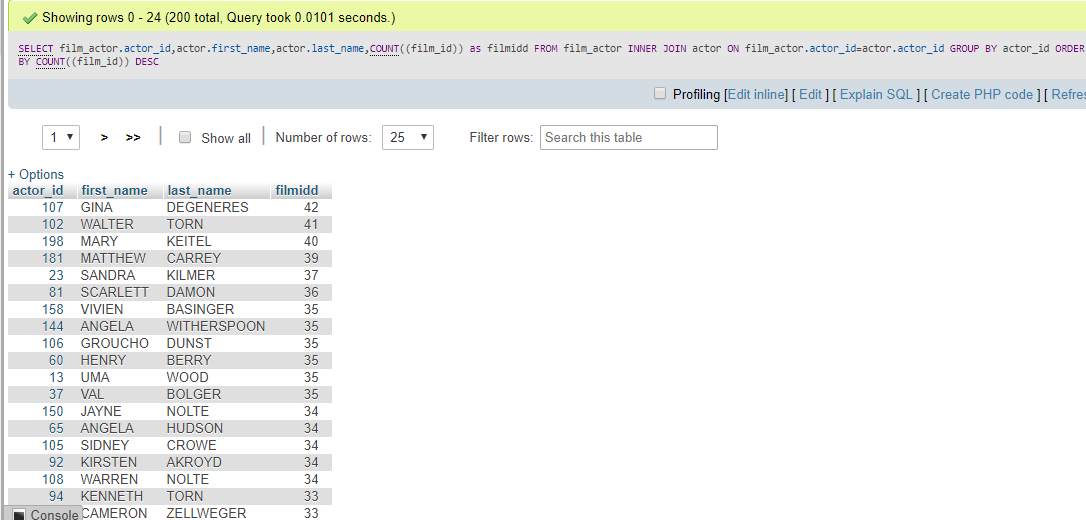
1. SELECT COUNT(first\_name), first\_name from actor GROUP by first\_name HAVING COUNT(first\_name)>1



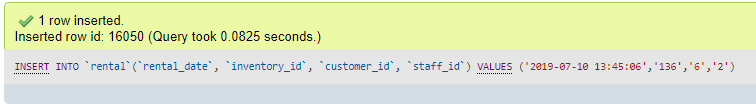
1. SELECT a.film\_id,a.title,t.store\_id from film\_text a,store t where a.title="ANONYMOUS HUMAN" and t.store\_id=2



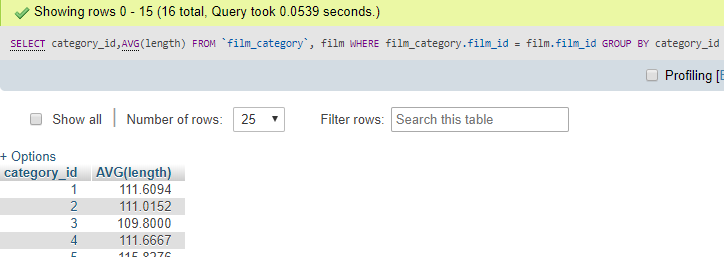
1. SELECT film\_actor.actor\_id,actor.first\_name,actor.last\_name,COUNT((film\_id)) as filmidd FROM film\_actor INNER JOIN actor ON film\_actor.actor\_id=actor.actor\_id GROUP BY actor\_id ORDER BY COUNT((film\_id)) DESC



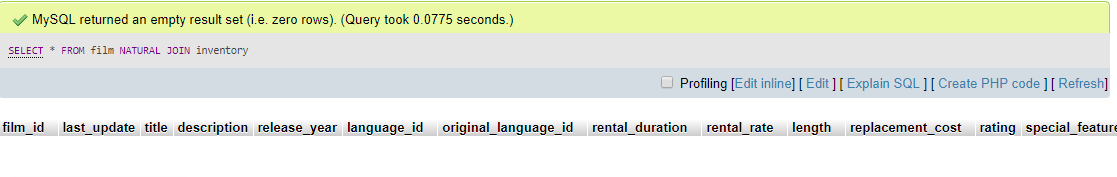
11. INSERT INTO `rental`(`rental\_date`, `inventory\_id`, `customer\_id`, `staff\_id`) VALUES ('2019-07-10 13:45:06','136','6','2')



12. SELECT category\_id,AVG(length) FROM `film\_category`, film WHERE film\_category.film\_id = film.film\_id GROUP BY category\_id

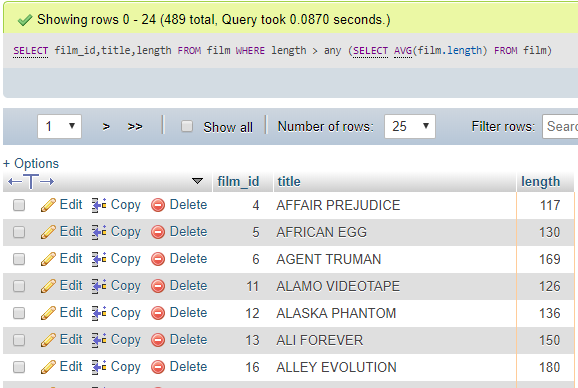


13. SELECT \* FROM film NATURAL JOIN inventory

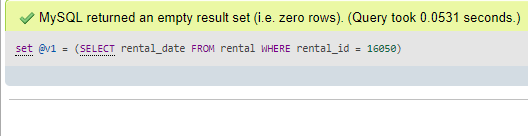


Here natural join does not give any output as compared to other types of joins, so it cannot be used in all cases and that is drawback of natural join.

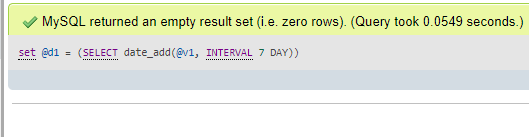
14. SELECT film\_id,title,length FROM film WHERE length > any (SELECT AVG(film.length) FROM film)



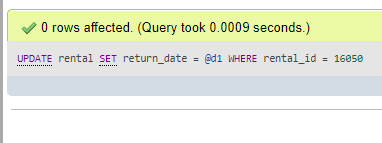
15. set @v1 = (SELECT rental\_date FROM rental WHERE rental\_id = 16050)



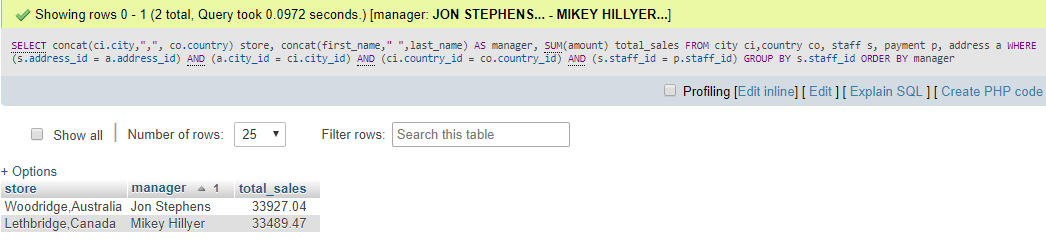
set @d1 = (SELECT date\_add(@v1, INTERVAL 7 DAY))



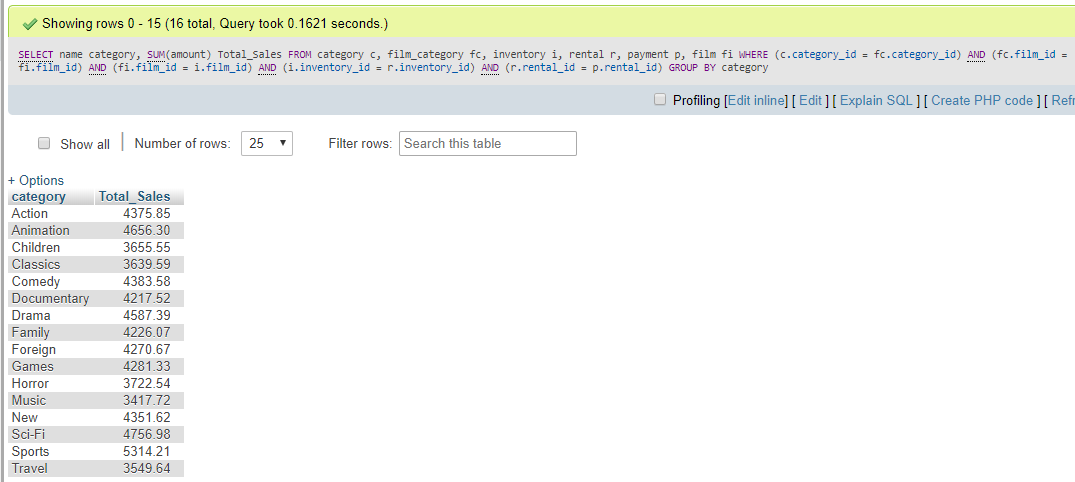
UPDATE rental SET return\_date = @d1 WHERE rental\_id = 16050



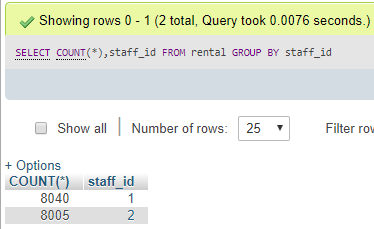
16. SELECT concat(ci.city,",", co.country) store, concat(first\_name," ",last\_name) AS manager, SUM(amount) total\_sales FROM city ci,country co, staff s, payment p, address a WHERE (s.address\_id = a.address\_id) AND (a.city\_id = ci.city\_id) AND (ci.country\_id = co.country\_id) AND (s.staff\_id = p.staff\_id) GROUP BY s.staff\_id ORDER BY manager



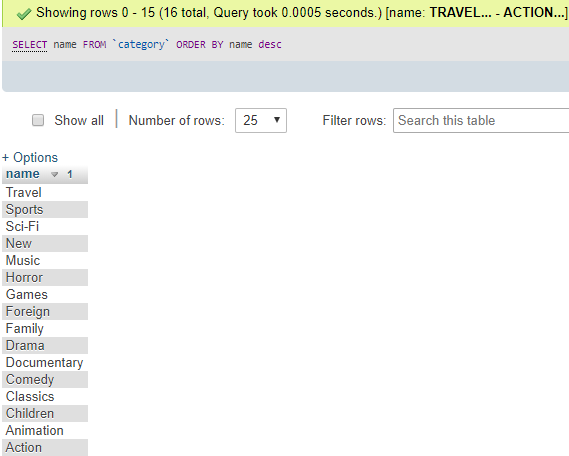
17. SELECT name category, SUM(amount) Total\_Sales FROM category c, film\_category fc, inventory i, rental r, payment p, film fi WHERE (c.category\_id = fc.category\_id) AND (fc.film\_id = fi.film\_id) AND (fi.film\_id = i.film\_id) AND (i.inventory\_id = r.inventory\_id) AND (r.rental\_id = p.rental\_id) GROUP BY category



18. SELECT COUNT(\*),staff\_id FROM rental GROUP BY staff\_id



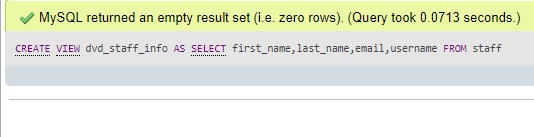
19. SELECT name FROM `category` ORDER BY name desc



20. select s.staff\_id id,concat(s.first\_name,' ',s.last\_name)as name,a.address address, a.postal\_code zipcode,a.phone phone,b.city city,c.country country,s.staff\_id sid FROM address a,city b,country c, staff s WHERE a.city\_id=b.city\_id and b.country\_id=c.country\_id and a.address\_id=s.address\_id



21. CREATE VIEW dvd\_staff\_info AS SELECT first\_name,last\_name,email,username FROM staff



22. UPDATE `dvd\_staff\_info` SET `first\_name`='Mikey' WHERE last\_name = 'Hillyer'

