

Write a query to display movie names and number of times that movie is issued to customers. Incase movies are never issued to customers display number of times as 0.

Display the details in sorted order based on number of times (in descending order) and then by movie name (in ascending order).

The Alias name for the number of movies issued is ISSUE\_COUNT.

SELECT movie\_name, count(cid.movie\_id) ISSUE\_COUNT FROM movies\_master mm LEFT JOIN customer\_issue\_details cid ON mm.movie\_id=cid.movie\_id GROUP BY movie\_name ORDER BY Issue\_count DESC, movie\_name;

Write a query to display id,name,age,contact no of customers whose age is greater than 25 and and who have registered in the year 2012. Display contact no in the below format +91-XXX-XXX-XXXX example +91-987-678-3434 and use the alias name as "CONTACT\_ISD". If the contact no is null then display as 'N/A' Sort all the records in ascending order based on age and then by name.

select customer\_id,customer\_name, age, coalesce(concat('+91-',substring(contact\_no,1,3),'-',substring(contact\_no,4,3),'-',substring(contact\_no,7)),'N/A') CONTACT\_ISD from customer\_master where age>25 and substring(date\_of\_registration,1,4)=2012 order by age asc, customer\_name;

Write a query to display the movie category and number of movies in that category. Display records based on number of movies from higher to lower order and then by movie category in ascending order.

<br/>Hint: Use NO\_OF\_MOVIES as alias name for number of movies.

SELECT movie\_category, count(movie\_id) NO\_OF\_MOVIES FROM movies\_master GROUP BY movie\_category order by no\_of\_movies desc, movie\_category asc;

Write a query to display the number of customers having card with description “Gold card”. <br/>Hint: Use CUSTOMER\_COUNT as alias name for number of customers

SELECT count(customer\_id) CUSTOMER\_COUNT FROM library\_card\_master lcm INNER JOIN customer\_card\_details ccd ON lcm.card\_id=ccd.card\_id WHERE description='Gold Card';

Write a query to display the customer id, customer name, year of registration,library card id, card issue date of all the customers who hold library card. Display the records sorted by customer name in descending order.

<br> Use REGISTERED\_YEAR as alias name for year of registration.

SELECT c.customer\_id, c.customer\_name, extract(year from c.date\_of\_registration) REGISTERED\_YEAR,cd.card\_id,cd.issue\_date FROM customer\_master c join customer\_card\_details cd on c.customer\_id=cd.customer\_id ORDER BY c.customer\_name DESC;

Write a query to display issue id, customer id, customer name for the customers who have paid fine and whose name starts with 'R'. Fine is calculated based on return date and actual date of return. If the date of actual return is after date of return then fine need to be paid by the customer.

<br>

Display the records sorted in ascending order based on customer name.

SELECT issue\_id ,cid.customer\_id, customer\_name FROM customer\_issue\_details cid INNER JOIN customer\_master cm ON cid.customer\_id=cm.customer\_id WHERE actual\_date\_of\_return>return\_date and customer\_name like 'R%' order by customer\_name;

Write a query to display customer id, customer name, card id, card description and card amount in dollars of customers who have taken movie on the same day the library card is registered.

For Example Assume John registered a library card on 12th Jan 2013 and he took a movie on 12th Jan 2013 then display his details.

AMOUNT\_DOLLAR = amount/52.42 and round it to zero decimal places and display as $Amount. Example Assume 500 is the amount then dollar value will be $10.

Hint: Use AMOUNT\_DOLLAR as alias name for amount in dollar.

Display the records in ascending order based on customer name.

SELECT ccd.customer\_id, customer\_name, ccd.card\_id, description,concat('$',round(amount/52.42,0)) AMOUNT\_DOLLAR FROM customer\_master cm INNER JOIN customer\_card\_details ccd ON cm.customer\_id=ccd.customer\_id INNER JOIN library\_card\_master lcm ON ccd.card\_id=lcm.card\_id INNER JOIN customer\_issue\_details cid ON cid.customer\_id = cm.customer\_id WHERE cm.date\_of\_registration=cid.issue\_date order by customer\_name;

Write a query to display the customer id, customer name,contact number and address of customers who have taken movies from library without library card and whose address ends with 'Nagar'.

Display customer name in upper case. Hint: Use CUSTOMER\_NAME as alias name for customer name. Display the details sorted in ascending order based on customer name.

SELECT customer\_id , upper(customer\_name) CUSTOMER\_NAME,contact\_no,contact\_address FROM customer\_master WHERE customer\_id NOT IN ( select customer\_id from customer\_card\_details ) AND customer\_id IN ( SELECT customer\_id from customer\_issue\_details ) and contact\_address like '%Nagar' order by customer\_name ;

Write a query to display the movie id, movie name,release year,director name of movies acted by the leadactor1 who acted maximum number of movies .Display the records sorted in ascending order based on movie name.

select movie\_id,movie\_name,release\_year,director\_name from movies\_master where lead\_actor\_name1 in(select lead\_actor\_name1 from (select lead\_actor\_name1,count(movie\_id)ct from movies\_master group by lead\_actor\_name1)t where t.ct>=all(select count(movie\_id) from movies\_master group by lead\_actor\_name1)) order by movie\_name;

Write a query to display the customer name and number of movies issued to that customer sorted by customer name in ascending order. If a customer has not been issued with any movie then display 0. <br>Hint: Use MOVIE\_COUNT as alias name for number of movies issued.

SELECT customer\_name, count(movie\_id) MOVIE\_COUNT FROM customer\_master cm LEFT OUTER JOIN customer\_issue\_details cid ON cm.customer\_id=cid.customer\_id GROUP BY customer\_name ORDER BY customer\_name;

Write a query to display serial number,issue id, customer id, customer name, movie id and movie name of all the videos that are issued and display in ascending order based on serial number.

<br/>Serial number can be generated from the issue id , that is last two characters of issue id is the serial number.

<br/>For Example Assume the issue id is I00005 then the serial number is 05

<br/>Hint: Alias name for serial number is 'SERIAL\_NO'<br/><br/><br>

SELECT substring(issue\_id,5,2) SERIAL\_NO,issue\_id, cid.customer\_id,customer\_name, mm.movie\_id, movie\_name FROM customer\_issue\_details cid INNER JOIN customer\_master cm ON cid.customer\_id=cm.customer\_id INNER JOIN movies\_master mm ON cid.movie\_id=mm.movie\_id order by SERIAL\_NO ASC;

Write a query to display the issue id,issue date, customer id, customer name and contact number for videos that are issued in the year 2013.Display the records in decending order based on issue date of the video.

SELECT issue\_id, issue\_date, cid.customer\_id, customer\_name,contact\_no FROM customer\_issue\_details cid INNER JOIN customer\_master cm ON cid.customer\_id=cm.customer\_id where extract(year from issue\_date)=2013 order by issue\_date DESC

Write a query to display movie id ,movie name and actor names of movies which are not issued to any customers. <br> Actors Name to be displayed in the below format.LEAD\_ACTOR\_ONE space ambersant space LEAD\_ACTOR\_TWO.

Example: Assume lead actor one's name is "Jack Tomson" and Lead actor two's name is "Maria" then Actors name will be "Jack Tomsom & Maria"Hint:Use ACTORS as alias name for actors name. <br> Display the records in ascending order based on movie name.

SELECT movie\_id, movie\_name,concat(lead\_actor\_name1,' & ',lead\_actor\_name2) ACTORS FROM movies\_master WHERE movie\_id NOT IN ( SELECT movie\_id from customer\_issue\_details ) order by movie\_name asc;

Write a query to display the director's name, movie name and lead\_actor\_name1 of all the movies directed by the director who directed more than one movie. Display the directors name in capital letters. Use DIRECTOR\_NAME as alias name for director name column Display the records sorted in ascending order based on director\_name and then by movie\_name in descending order.

SELECT upper(director\_name) DIRECTOR\_NAME,movie\_name,lead\_actor\_name1 FROM movies\_master WHERE director\_name in (SELECT director\_name FROM movies\_master GROUP BY director\_name HAVING count(movie\_id)>1) order by director\_name, movie\_name desc;

Write a query to display number of customers who have registered in the library in the year 2012 and who have given/provided contact number. <br> Hint:Use NO\_OF\_CUSTOMERS as alias name for number of customers.

SELECT count(customer\_id) NO\_OF\_CUSTOMERS FROM customer\_master WHERE extract(year from date\_of\_registration)=2012 and contact\_no is not null

Write a query to display the customer's name, contact number,library card id and library card description of all the customers irrespective of customers holding a library card. If customer contact number is not available then display his address. Display the records sorted in ascending order based on customer name. Hint: Use CONTACT\_DETAILS as alias name for customer contact.

SELECT c.customer\_name,coalesce(c.contact\_no,c.contact\_address) CONTACT\_DETAILS,cd.card\_id,cd.description FROM customer\_master c left join customer\_card\_details ccd on ccd.customer\_id=c.customer\_id left join library\_card\_master cd on cd.card\_id=ccd.card\_id order by customer\_name;

Write a query to display the customer id, customer name and number of times the same movie is issued to the same customers who have taken same movie more than once. Display the records sorted by customer name in decending order For Example: Assume customer John has taken Titanic three times and customer Ram has taken Die hard only once then display the details of john. Hint: Use NO\_OF\_TIMES as alias name for number of times

SELECT cid.customer\_id, customer\_name,count(movie\_id) NO\_OF\_TIMES FROM customer\_master cm INNER JOIN customer\_issue\_details cid ON cm.customer\_id=cid.customer\_id group by customer\_id, customer\_name,movie\_id having count(movie\_id)>1 order by customer\_name desc;

Write a query to display customer id, customer name,contact number, movie category and number of movies issued to each customer based on movie category who has been issued with more than one movie in that category. Example: Display contact number as "+91-876-456-2345" format.&nbsp; <br>Hint:Use NO\_OF\_MOVIES as alias name for number of movies column.

<br>Hint:Use CONTACT\_ISD as alias name for contact number.

<br> Display the records sorted in ascending order based on customer name and then by movie category.

SELECT cid.customer\_id,customer\_name,concat('+91-',substring(contact\_no,1,3),'-',substring(contact\_no,4,3),'-',substring(contact\_no,7)) CONTACT\_ISD,movie\_category,count(movie\_category) NO\_OF\_MOVIES FROM customer\_issue\_details cid INNER JOIN movies\_master mm ON cid.movie\_id=mm.movie\_id INNER JOIN customer\_master cm ON cm.customer\_id=cid.customer\_id group by customer\_id,customer\_name,CONTACT\_ISD,movie\_category having count(movie\_category)>1 order by customer\_name, movie\_category;

Write a query to display customer id and customer name of customers who has been issued with maximum number of movies and customer who has been issued with minimum no of movies.

For example Assume customer John has been issued 5 movies, Ram has been issued 10 movies and Kumar has been issued 2 movies. The name and id of Ram should be displayed for issuing maximum movies and Kumar should be displayed for issuing minimum movies. Consider only the customers who have been issued with atleast 1 movie Customer(s) who has/have been issued the maximum number of movies must be displayed first followed by the customer(s) who has/have been issued with the minimum number of movies. In case of multiple customers who have been displayed with the maximum or minimum number of movies, display the records sorted in ascending order based on customer name.

(select cid.customer\_id , customer\_name FROM customer\_master cm INNER JOIN customer\_issue\_details cid

ON cm.customer\_id=cid.customer\_id

group by customer\_id , customer\_name

having count(movie\_id)>=ALL(select count(movie\_id)

FROM customer\_issue\_details

group by customer\_id) order by customer\_name)

UNION ALL

(select cid.customer\_id , customer\_name FROM customer\_master cm INNER JOIN customer\_issue\_details cid

ON cm.customer\_id=cid.customer\_id

group by customer\_id , customer\_name

having count(movie\_id)<=ALL(select count(movie\_id)

FROM customer\_issue\_details

group by customer\_id) order by customer\_name) ;

Write a query to display the customer id , customer name and number of times movies have been issued from Comedy category. Display only for customers who has taken more than once.

Hint: Use NO\_OF\_TIMES as alias name

Display the records in ascending order based on customer name.

SELECT cid.customer\_id,customer\_name,count(cid.movie\_id) NO\_OF\_TIMES FROM customer\_master cm INNER JOIN customer\_issue\_details cid ON cm.customer\_id=cid.customer\_id INNER JOIN movies\_master mm

ON cid.movie\_id=mm.movie\_id WHERE movie\_category='Comedy' GROUP BY customer\_name HAVING count(cid.customer\_id)>1 order by customer\_name;

Write a query to display customer id and total rent paid by the customers who are issued with the videos. Need not display the customers who has not taken / issued with any videos. Hint: Alias Name for total rent paid is TOTAL\_COST. Display the records sorted in ascending order based on customer id

SELECT cid.customer\_id, sum(rental\_cost) TOTAL\_COST FROM customer\_issue\_details cid INNER JOIN movies\_master mm ON cid.movie\_id=mm.movie\_id GROUP BY customer\_id order by customer\_id;