

Write a query to display category and number of items in that category. Give the count an alias name of Count\_category. Display the details on the sorted order of count in descending order.

SELECT item\_category , count(item\_id) Count\_category

FROM item\_master

GROUP BY item\_category order by count\_category DESC

Write a query to display the number of employees in HR department. Give the alias name as No\_of\_Employees.

SELECT count(employee\_id) AS No\_of\_Employees

FROM employee\_master

WHERE department= 'HR'

Write a query to display employee id, employee name, designation and department for employees who have never been issued an item as a loan from the company. Display the records sorted in ascending order based on employee id.

SELECT employee\_id, employee\_name, designation, department

FROM employee\_master WHERE employee\_id

NOT IN ( SELECT employee\_id FROM employee\_issue\_details)

order by employee\_id;

Write a query to display the employee id, employee name who was issued an item of highest valuation.

<br> In case of multiple records, display the records sorted in ascending order based on employee id.

[Hint Suppose an item called dinning table is of 22000 and that is the highest price of the item that has been issued. So display the employee id and employee name who issued dinning table whose price is 22000.]

SELECT eid.employee\_id, employee\_name

FROM employee\_master em INNER JOIN employee\_issue\_details eid

ON em.employee\_id=eid.employee\_id

INNER JOIN item\_master im

ON eid.item\_id=im.item\_id

WHERE item\_valuation=(SELECT max(item\_valuation)

FROM employee\_issue\_details eid INNER JOIN item\_master im

ON eid.item\_id=im.item\_id) order by eid.employee\_id;

Write a query to display issue\_id, employee\_id, employee\_name.

Display the records sorted in ascending order based on issue id.

SELECT issue\_id, eid.employee\_id, employee\_name

FROM employee\_master em INNER JOIN employee\_issue\_details eid

ON em.employee\_id=eid.employee\_id order by issue\_id;

Write a query to display employee id, employee name who don’t have loan cards.

Display the records sorted in ascending order based on employee id.

SELECT employee\_id, employee\_name

FROM employee\_master

WHERE employee\_id NOT IN ( SELECT employee\_id FROM employee\_card\_details )

order by employee\_id;

Write a query to count the number of cards issued to an employee “Ram”. Give the count an alias name as No\_of\_Cards.

SELECT count(loan\_id) AS No\_of\_Cards

FROM employee\_card\_details c

JOIN employee\_master e

ON c.employee\_id = e.employee\_id

WHERE e.employee\_name= 'Ram'

GROUP BY c.employee\_id

Write a query to display the count of customers who have gone for loan type stationary. Give the count an alias name as Count\_stationary.

SELECT count(employee\_id) Count\_stationary

FROM employee\_card\_details ecd INNER JOIN loan\_card\_master lcd

ON ecd.loan\_id=lcd.loan\_id

WHERE loan\_type='stationary'

Write a query to display the employee id, employee name and number of items issued to them. Give the number of items an alias name as Count. Display the details in descending order of count and then by employee id in ascending order. Consider only employees who have been issued atleast 1 item.

SELECT eid.employee\_id, employee\_name, count(item\_id) Count

FROM employee\_master em INNER JOIN employee\_issue\_details eid

ON em.employee\_id=eid.employee\_id

GROUP BY employee\_id order by count DESC, eid.employee\_id;

Write a query to display the employee id, employee name who was issued an item of minimum valuation.

<br>

In case of multiple records, display them sorted in ascending order based on employee id.

[Hint Suppose an item called pen is of rupees 20 and that is the lowest price. So display the employee id and employee name who issued pen where the valuation is 20.]

SELECT eid.employee\_id, employee\_name

FROM employee\_master em INNER JOIN employee\_issue\_details eid

ON em.employee\_id=eid.employee\_id

INNER JOIN item\_master im

ON eid.item\_id=im.item\_id

WHERE item\_valuation=(SELECT min(item\_valuation)

FROM employee\_issue\_details eid INNER JOIN item\_master im

ON eid.item\_id=im.item\_id)

order by eid.employee\_id;

Write a query to display the employee id, employee name and total valuation of the product issued to each employee. Give the alias name as TOTAL\_VALUATION.

<br>

Display the records sorted in ascending order based on employee id.

<br>

Consider only employees who have been issued atleast 1 item.

SELECT em.employee\_id, employee\_name, sum(item\_valuation) TOTAL\_VALUATION

FROM employee\_master em INNER JOIN employee\_issue\_details eid

ON em.employee\_id=eid.employee\_id

INNER JOIN item\_master im

ON eid.item\_id=im.item\_id

GROUP BY eid.employee\_id

ORDER BY eid.employee\_id;

Write a query to display distinct employee id, employee name who kept the item issued for more than a year. Hint: Use Date time function to calculate the difference between item issue and return date. Display the records only if it is more than 365 Days.

<br>

Display the records sorted in ascending order based on employee id.

SELECT DISTINCT eid.employee\_id,employee\_name

FROM employee\_issue\_details eid JOIN employee\_master em

ON eid.employee\_id=em.employee\_id

WHERE datediff(return\_date,issue\_date) > 365

order by eid.employee\_id;

Write a query to display employee id, employee name and count of items of those who asked for more than 1 furniture. Give the alias name for count of items as COUNT\_ITEMS.

<br>

Display the records sorted in ascending order on employee id.

SELECT eid.employee\_id, employee\_name, count(eid.item\_id) COUNT\_ITEMS

FROM employee\_issue\_details eid JOIN item\_master im

ON eid.item\_id=im.item\_id

JOIN employee\_master em

ON eid.employee\_id=em.employee\_id

WHERE item\_category='furniture'

GROUP BY employee\_id

HAVING COUNT\_ITEMS>1

order by eid.employee\_id;

Write a query to display the number of men & women Employees. The query should display the gender and number of Employees as No\_of\_Employees. Display the records sorted in ascending order based on gender.

SELECT gender , count(gender) No\_of\_Employees

FROM employee\_master

GROUP BY gender ORDER BY gender

Write a query to display employee id, employee name who joined the company after 2005. Display the records sorted in ascending order based on employee id.

SELECT employee\_id, employee\_name

FROM employee\_master

WHERE extract(year from date\_of\_joining)>2005

order by employee\_id;

Write a query to get the number of items of the furniture category issued and not issued. The query should display issue status and the number of furniture as No\_of\_Furnitures.

Display the records sorted in ascending order based on issue\_status.

SELECT issue\_status, count(item\_id) No\_of\_Furnitures

FROM item\_master

WHERE item\_category='Furniture'

GROUP BY issue\_status

ORDER BY issue\_status

Write a query to find the number of items in each category, make and description. The Query should display Item Category, Make, description and the number of items as No\_of\_Items. Display the records in ascending order based on Item Category, then by item make and then by item description.

SELECT item\_category, item\_make,item\_description, count(item\_description) No\_of\_Items FROM item\_master

GROUP BY item\_category, item\_make, item\_description ORDER BY item\_category, item\_make,item\_description;

Write a query to display employee id, employee name, item id and item description of employees who were issued item(s) in the month of January 2013. Display the records sorted in order based on employee id and then by item id in ascending order.

SELECT eid.employee\_id, employee\_name, eid.item\_id, item\_description

FROM employee\_issue\_details eid JOIN employee\_master em

ON eid.employee\_id=em.employee\_id

JOIN item\_master im

ON eid.item\_id=im.item\_id

WHERE extract(month from issue\_date)=1 and extract(year from issue\_date)=2013

ORDER BY eid.employee\_id, eid.item\_id;

Write a query to display the employee id, employee name and count of item category of the employees who have been issued items in at least 2 different categories.

Give the alias name for category count as COUNT\_CATEGORY.

Display the records sorted in ascending order based on employee id.

SELECT em.employee\_id,employee\_name, count(distinct item\_category) COUNT\_CATEGORY

FROM employee\_issue\_details eid JOIN item\_master im

ON eid.item\_id=im.item\_id

JOIN employee\_master em

ON eid.employee\_id=em.employee\_id

GROUP BY employee\_id

HAVING COUNT\_CATEGORY>=2

ORDER BY employee\_id;

Write a query to display the item id , item description which was never issued to any employee. Display the records sorted in ascending order based on item id.

SELECT item\_id, item\_description

FROM item\_master

WHERE item\_id

NOT IN ( SELECT item\_id from employee\_issue\_details)

ORDER BY item\_id;

Write a query to display the employee id, employee name and&nbsp;&nbsp;total valuation&nbsp;for the employees who has issued minimum total valuation of the product. Give the alias name for total valuation as TOTAL\_VALUATION.

[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000 and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name of E00020 should be displayed. ]

select employee\_issue\_details.employee\_id,employee\_master.employee\_name,sum(item\_master.item\_valuation)as TOTAL\_VALUATION from

employee\_issue\_details inner join item\_master

on item\_master.item\_id = employee\_issue\_details.item\_id

inner join employee\_master

on employee\_master.employee\_id=employee\_issue\_details.employee\_id

group by employee\_issue\_details.employee\_id

order by TOTAL\_VALUATION asc limit 1;

Write a query to display the employee id, employee name, card issue date and card valid date.

Order by employee name and then by card valid date. Give the alias name to display the card valid date as CARD\_VALID\_DATE.

[Hint: Validity in years for the loan card is given in loan\_card\_master table. Validity date is calculated by adding number of years in the loan card issue date. If the duration of year is zero then display AS 'No Validity Date'. ]

SELECT ecd.employee\_id,employee\_name,

card\_issue\_date, CASE duration\_in\_years

WHEN 0 THEN 'No Validity Date'

ELSE DATE\_ADD(card\_issue\_date, INTERVAL duration\_in\_years YEAR)

END CARD\_VALID\_DATE

FROM employee\_master em INNER JOIN

employee\_card\_details ecd

ON em.employee\_id=ecd.employee\_id

INNER JOIN loan\_card\_master lcd

ON ecd.loan\_id=lcd.loan\_id

order by employee\_name, CARD\_VALID\_DATE;

Write a query to display the employee id, employee name who have not issued with any item in the year 2013. Hint: Exclude those employees who was never issued with any of the items in all the years. Display the records sorted in ascending order based on employee id.

SELECT DISTINCT a.employee\_id,b.employee\_name FROM employee\_issue\_details a JOIN

employee\_master b ON a.employee\_id=b.employee\_id WHERE a.employee\_id NOT IN(SELECT employee\_id

FROM employee\_issue\_details WHERE (EXTRACT(YEAR FROM issue\_date)=2013))

ORDER BY a.employee\_id;

Write a query to display issue id, employee id, employee name, item id, item description and issue date. Display the data in descending order of date and then by issue id in ascending order.

SELECT issue\_id, eid.employee\_id, employee\_name, im.item\_id, item\_description,issue\_date

FROM employee\_issue\_details eid INNER JOIN employee\_master em

ON eid.employee\_id=em.employee\_id

INNER JOIN item\_master im

ON eid.item\_id=im.item\_id

ORDER BY issue\_date desc, issue\_id asc;

Write a query to display the employee id, employee name and total valuation for employee who has issued maximum total valuation of the product.&nbsp; Give the alias name for total valuation as TOTAL\_VALUATION.&nbsp;

<br>[Hint: Suppose an employee E00019 issued item of price 5000, 10000, 12000 and E00020 issue item of price 2000, 7000, and 1000. So the valuation of items taken by E00019 is 27000 and for E00020 it is 10000. So the employee id, employee name and total valuation of E00019 should display. ]

select employee\_issue\_details.employee\_id,employee\_master.employee\_name,sum(item\_master.item\_valuation)as TOTAL\_VALUATION from

employee\_issue\_details inner join item\_master

on item\_master.item\_id = employee\_issue\_details.item\_id

inner join employee\_master

on employee\_master.employee\_id=employee\_issue\_details.employee\_id

group by employee\_issue\_details.employee\_id

order by TOTAL\_VALUATION desc limit 1;