Please follow instructions given below.

<br>

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
```

Please follow instructions given below.

<br>

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'
```

Please follow instructions given below.

<br><br>

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;
```

Please follow instructions given below.

<br><br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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
```

Please follow instructions given below.

<br>

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'
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

<br>

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;
```

Please follow instructions given below.

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;
```

Please follow instructions given below.

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;
```

Please follow instructions given below.

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;
```

Please follow instructions given below.

<br>

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;
```