**Institute of Computer Technology**

**B. Tech Computer Science and Engineering**

**Sub: Database Management System (2CSE301)**

**Practical: 5 Perform Queries using Group by and Having clause.**

The SQL **GROUP BY** clause is used in collaboration with the SELECT statement to arrange identical data into groups. This GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

Syntax

The basic syntax of a GROUP BY clause is shown in the following code block. The GROUP BY clause must follow the conditions in the WHERE clause and must precede the ORDER BY clause if one is used.

SELECT column1, column2

FROM table\_name

WHERE [ conditions ]

GROUP BY column1, column2

ORDER BY column1, column2

SELECT column1, column2

FROM table\_name

GROUP BY column1, column2

Having [CONDITION]

Example

Consider the CUSTOMERS table is having the following records −

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

If you want to know the total amount of the salary on each customer, then the GROUP BY query would be as follows.

SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS

GROUP BY NAME;

This would produce the following result −

+----------+-------------+

| NAME | SUM(SALARY) |

+----------+-------------+

| Chaitali | 6500.00 |

| Hardik | 8500.00 |

| kaushik | 2000.00 |

| Khilan | 1500.00 |

| Komal | 4500.00 |

| Muffy | 10000.00 |

| Ramesh | 2000.00 |

+----------+-------------+

Now, let us look at a table where the CUSTOMERS table has the following records with duplicate names −

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

# +----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Ramesh | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | kaushik | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

# +----+----------+-----+-----------+----------+

Now again, if you want to know the total amount of salary on each customer, then the GROUP BY query would be as follows −

SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS

GROUP BY NAME;

This would produce the following result −

+---------+-------------+

| NAME | SUM(SALARY) |

+---------+-------------+

| Hardik | 8500.00 |

| kaushik | 8500.00 |

| Komal | 4500.00 |

| Muffy | 10000.00 |

| Ramesh | 3500.00 |

+---------+-------------+

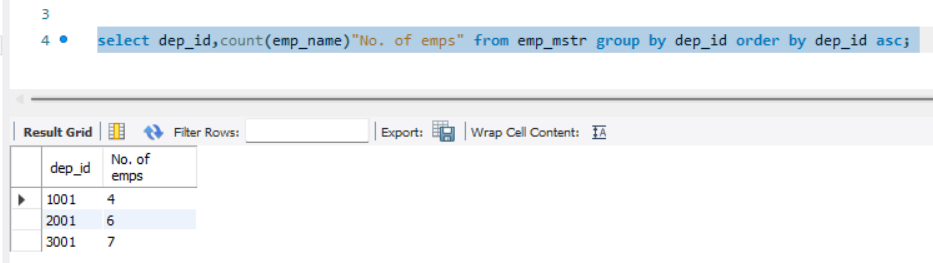
**The SQL HAVING Clause**

The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

**Queries:**

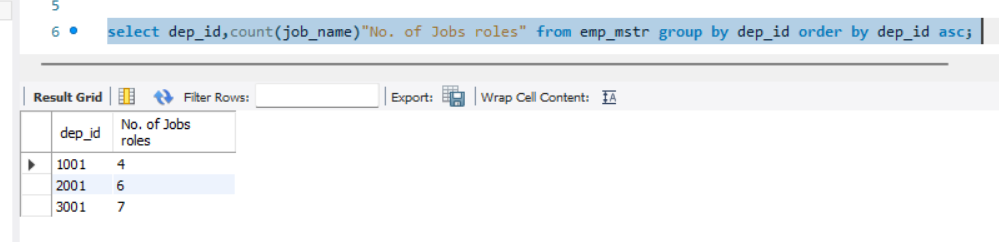
1. How many employees are there in each department?

select dep\_id,count(emp\_name)"No. of emps" from emp\_mstr group by dep\_id order by dep\_id asc;



1. Find out total number of job role assigned in each department.

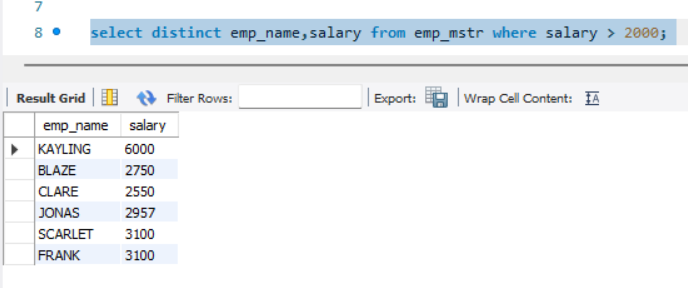
>select dep\_id,count(job\_name)"No. of Jobs roles" from emp\_mstr group by dep\_id order by dep\_id asc;



1. Find out employee’s names and salary whose having salary more than 2000.

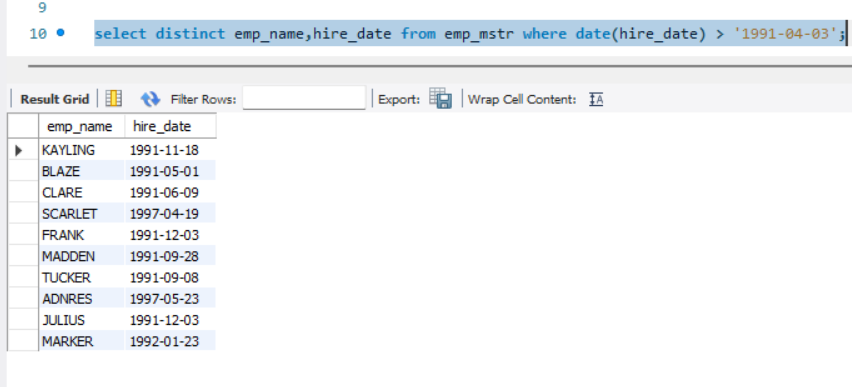
(Duplication in employee name should be removed)

select distinct emp\_name,salary from emp\_mstr where salary > 2000;



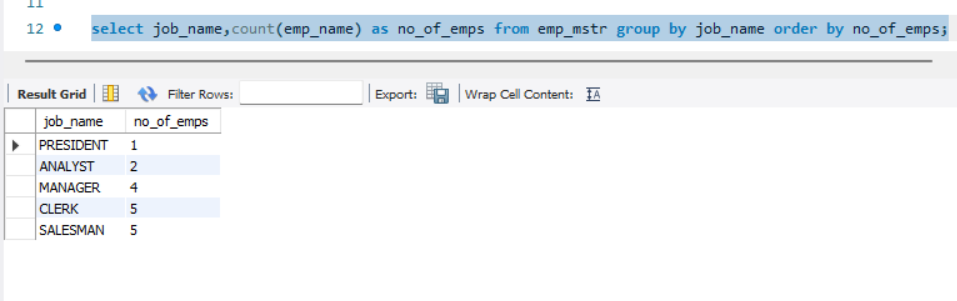
1. Find out number of employees hired after 03rd April 1991.

select distinct emp\_name,hire\_date from emp\_mstr where date(hire\_date) > '1991- 04-03';



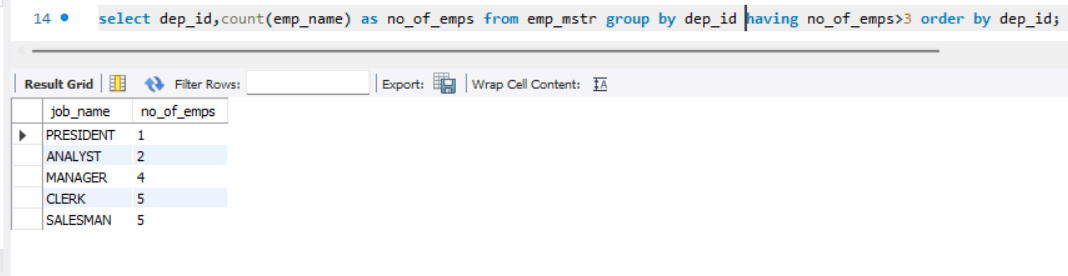
1. lists the number of employees in each job role, sorted high to low.

select job\_name,count(emp\_name) as no\_of\_emps from emp\_mstr group by job\_name order by no\_of\_emps;



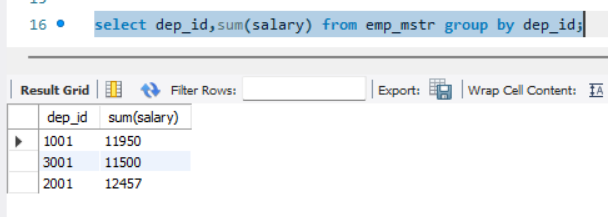
1. lists the number of employees in each department. Only include department with more than 3 employees in each.

select dep\_id,count(emp\_name) as no\_of\_emps from emp\_mstr group by dep\_id having no\_of\_emps>3 order by dep\_id;



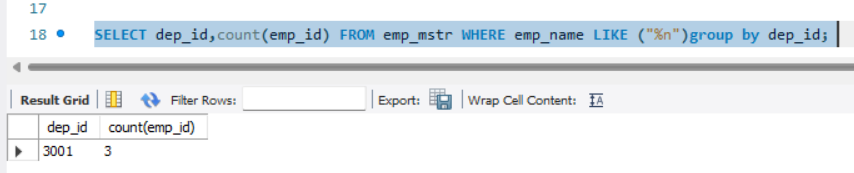
1. Display the total amount of the salary on each department.

select dep\_id,sum(salary) from emp\_mstr group by dep\_id;



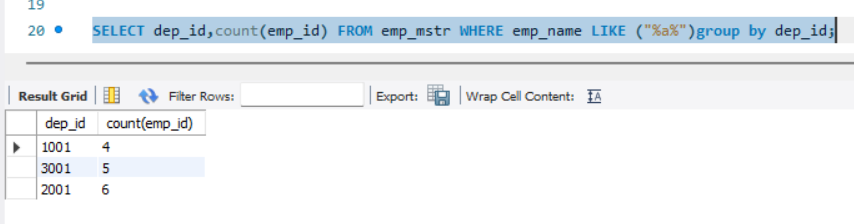
1. Count total number of employees assigned in each department whose name end with “n”.

SELECT dep\_id,count(emp\_id) FROM emp\_mstr WHERE emp\_name LIKE ("%n")group by dep\_id;



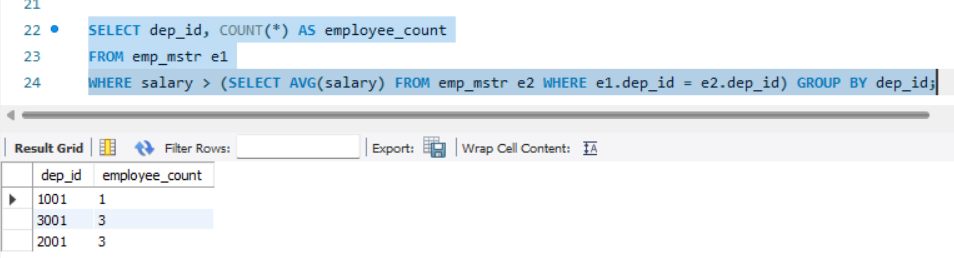
1. Find out total number of employees having "a" as a character in their name in each department.

SELECT dep\_id,count(emp\_id) FROM emp\_mstr WHERE emp\_name LIKE ("%a%")group by dep\_id;



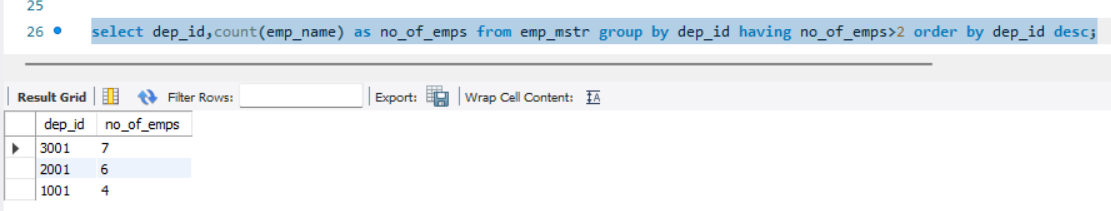
1. Find out total number of employees having salary more than average salary of all the employee in each department.

SELECT dep\_id, COUNT(\*) AS employee\_count FROM emp\_mstr e1 WHERE salary > (SELECT AVG(salary) FROM emp\_mstr e2 WHERE e1.dep\_id = e2.dep\_id) GROUP BY dep\_id;



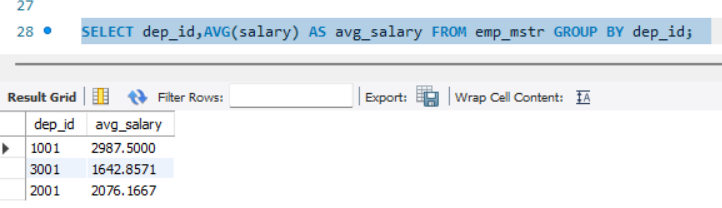
1. Display total number of employees in each department whose department having more than 2 employees also display department id in descending order.

select dep\_id,count(emp\_name) as no\_of\_emps from emp\_mstr group by dep\_id having no\_of\_emps>2 order by dep\_id desc;



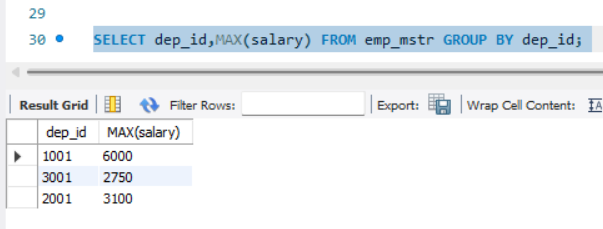
1. Display department wise average salary of employee.

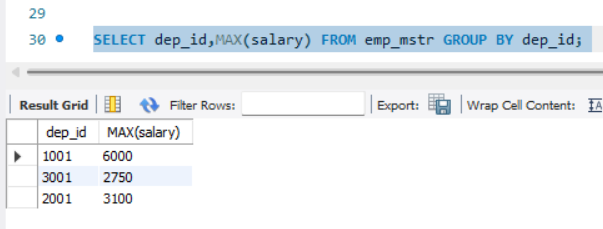
SELECT dep\_id,AVG(salary) AS avg\_salary FROM emp\_mstr GROUP BY dep\_id;



1. Display department id of the employee along with salary whose salary is maximum in respective department.

SELECT dep\_id,MAX(salary) FROM emp\_mstr GROUP BY dep\_id;





1. Display department id of the employee along with salary whose salary is minimum in respective department.

SELECT dep\_id,MIN(salary) FROM emp\_mstr GROUP BY dep\_id

