**Ganpat University**

**Faculty of Engineering & Technology**

**Computer Science & Engineering**

**Practical\_5**

***Name:- Dwij Vatsal Desai***

***Sem:- 3***

***Sub: - DBMS­­***

***Enrollment No.:- 23162121027***

**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.

# Solution Query:

# Transfered table:

# create database PRAC\_5;

# use PRAC\_5;

# Select \* from emp\_mstr;

# Image:

**-- 1) How many employees are there in each department?**

SELECT emp\_name, COUNT(salary) AS emp\_count

FROM emp\_mstr

GROUP BY emp\_name;

# Image:

# 

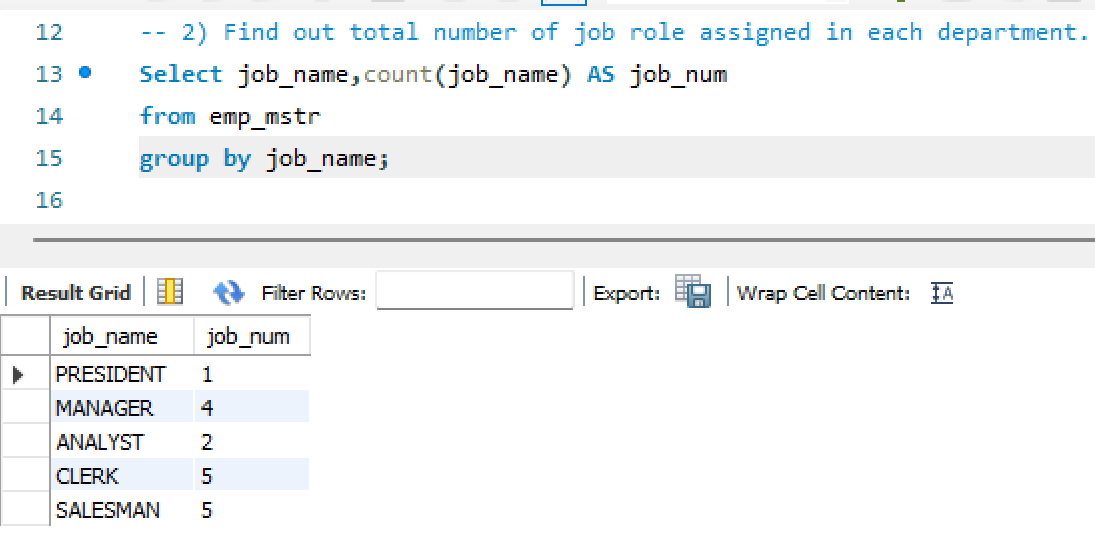
**-- 2) Find out total number of job role assigned in each department.**

Select job\_name,count(job\_name) AS job\_num

from emp\_mstr

group by job\_name;

# Image:



**-- 3) Find out employee’s names and salary whose having salary more than 2000.(Duplication in employee name should be removed)**

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

# Image:

**-- 4) Find out number of employees hired after 03rd April 1991.**

Select emp\_name,hire\_date,count(emp\_name) AS emp\_num

from emp\_mstr where hire\_date>"1991-06-03"

group by emp\_name,hire\_date;

# Image:

# 

**-- 5) lists the number of employees in each job role, sorted high to low.**

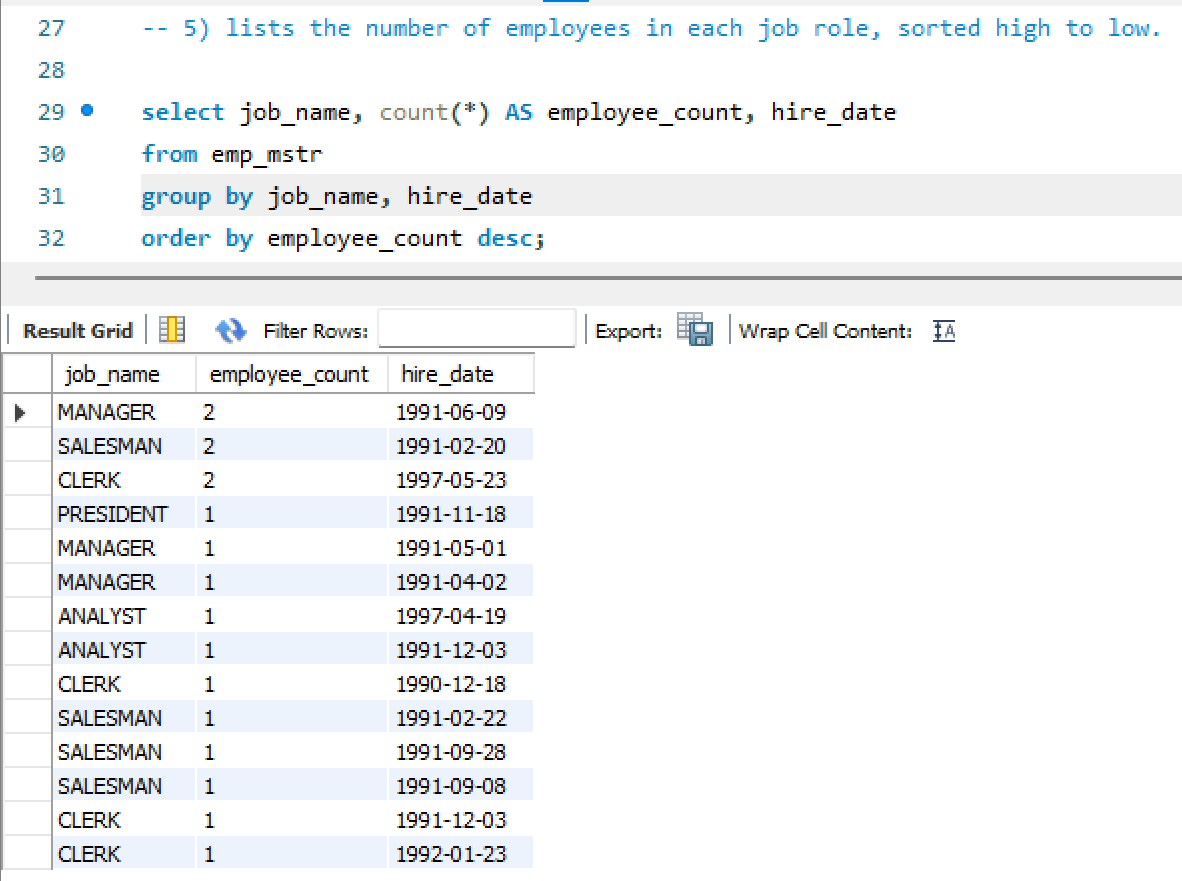
select job\_name, count(\*) AS employee\_count, hire\_date

from emp\_mstr

group by job\_name, hire\_date

order by employee\_count desc;

# Image:



**-- 6) lists the number of employees in each department. Only include department with more than 3 employees in each.**

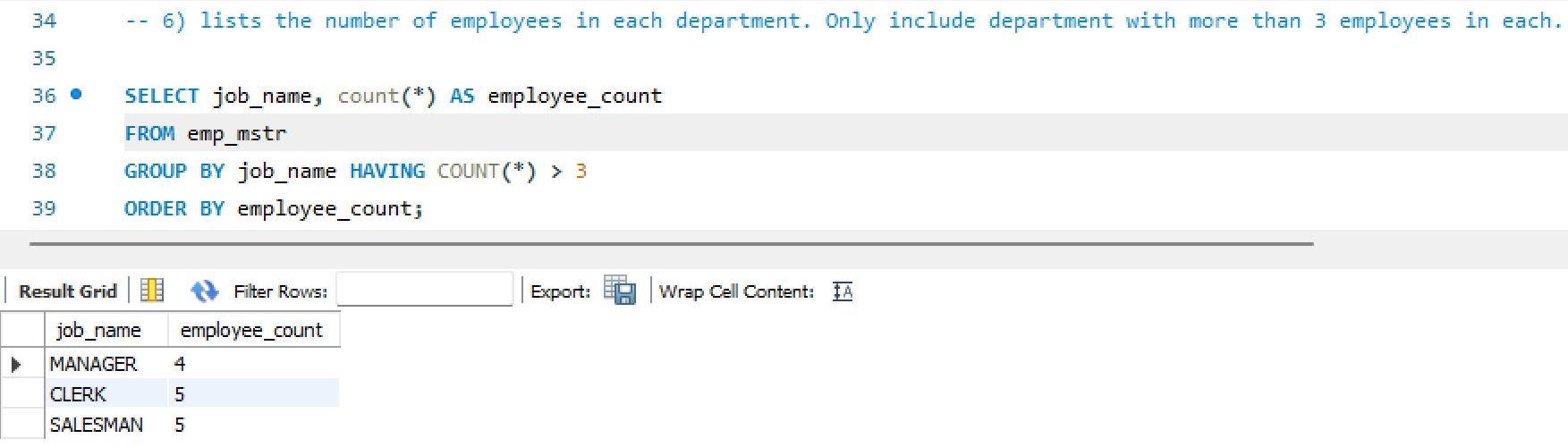
SELECT job\_name, count(\*) AS employee\_count

FROM emp\_mstr

GROUP BY job\_name HAVING COUNT(\*) > 3

ORDER BY employee\_count;

# Image:



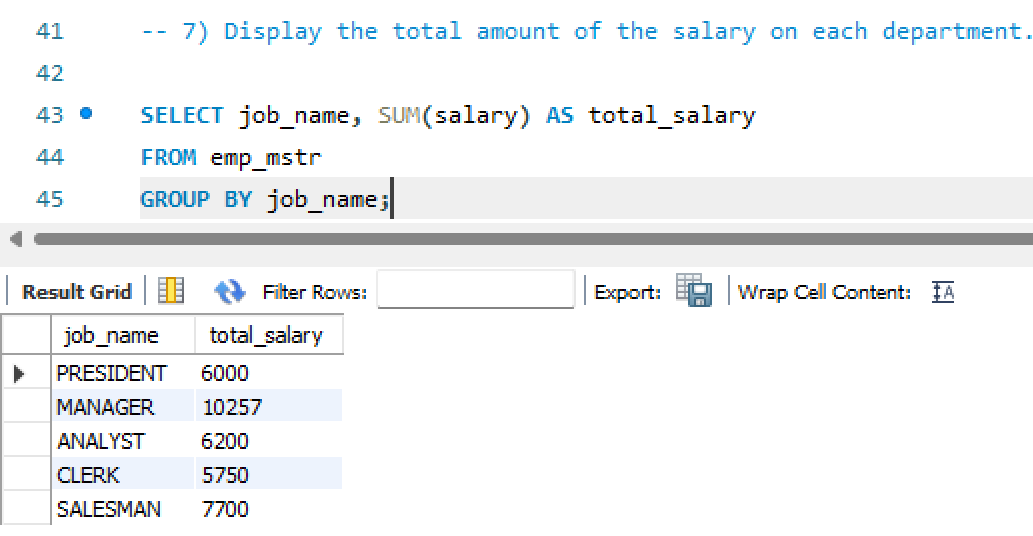
**-- 7) Display the total amount of the salary on each department.**

SELECT job\_name, SUM(salary) AS total\_salary

FROM emp\_mstr

GROUP BY job\_name;

# Image:



**-- 8) Count total number of employees assigned in each department whose**

**name end with “n”.**

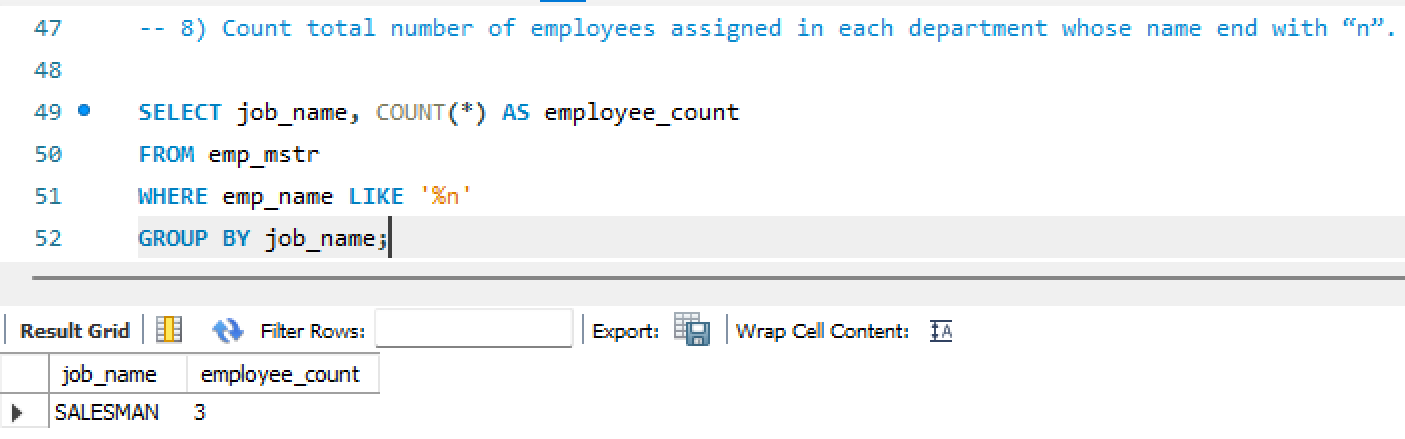
SELECT job\_name, COUNT(\*) AS employee\_count

FROM emp\_mstr

WHERE emp\_name LIKE '%n'

GROUP BY job\_name;

# Image:



**-- 9) Find out total number of employees having "a" as a character in their**

**name in each department.**

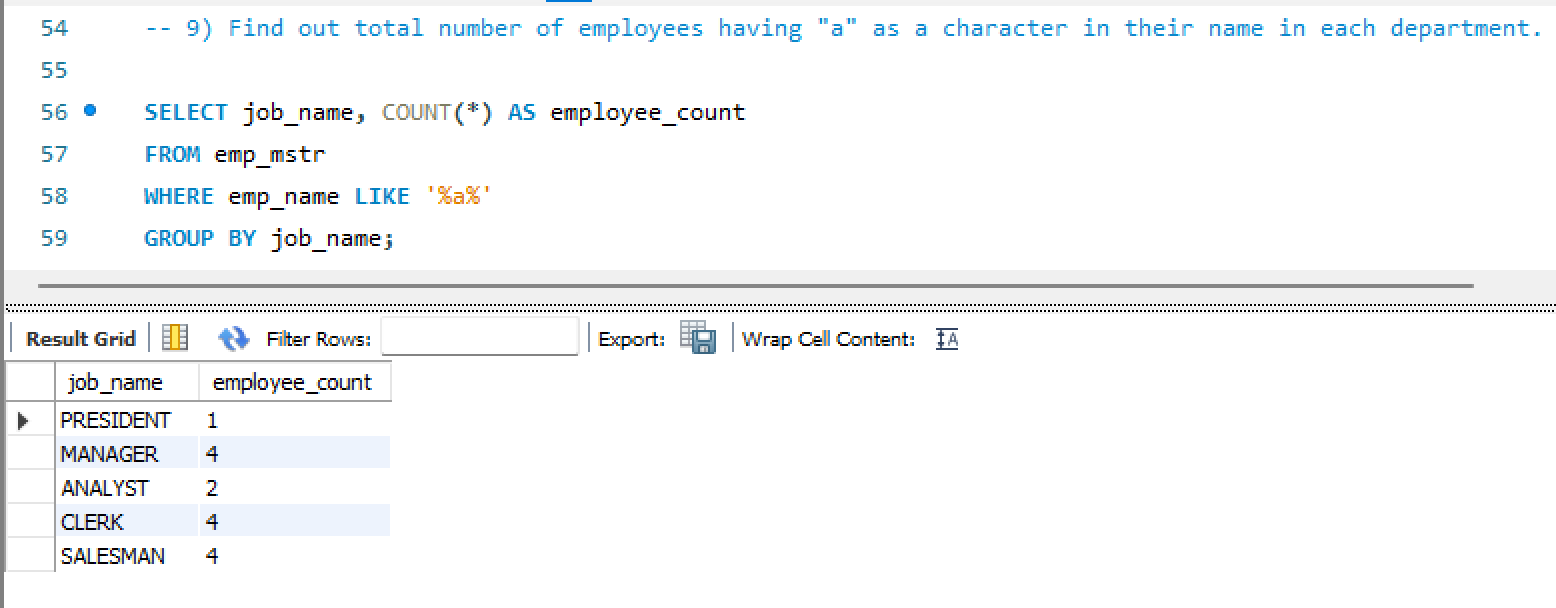
SELECT job\_name, COUNT(\*) AS employee\_count

FROM emp\_mstr

WHERE emp\_name LIKE '%a%'

GROUP BY job\_name;

# Image:



**-- 10) Find out total number of employees having salary more than average salary of all the employee in each department.**

SELECT job\_name, COUNT(\*) AS employee\_count

FROM emp\_mstr

WHERE salary > (SELECT AVG(salary) FROM emp\_mstr)

GROUP BY job\_name;

# Image:

# 

**-- 11)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(\*) AS employee\_count

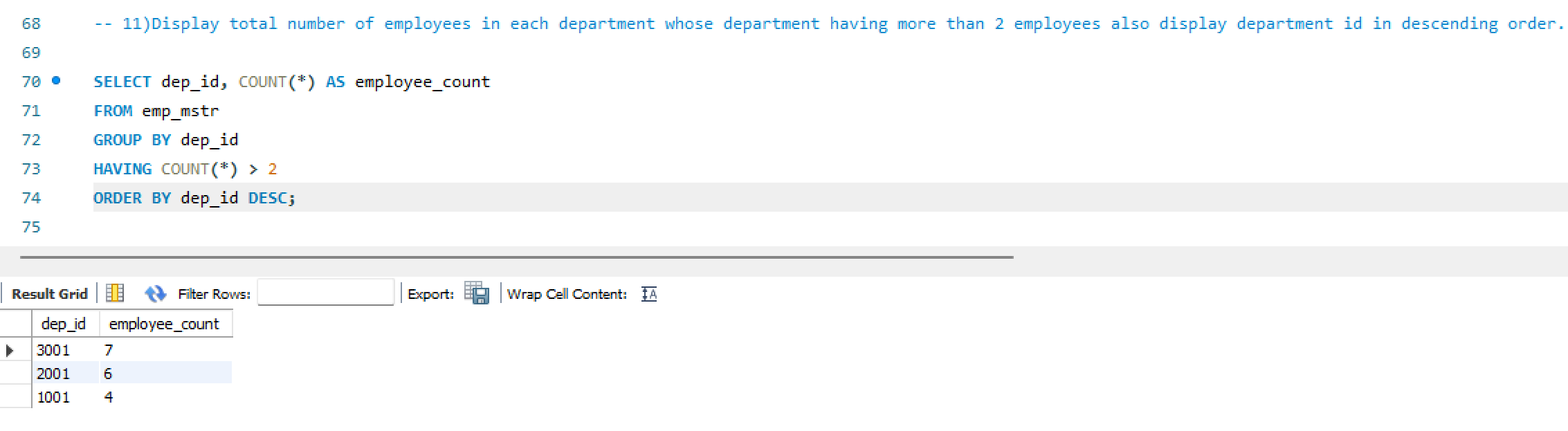
FROM emp\_mstr

GROUP BY dep\_id

HAVING COUNT(\*) > 2

ORDER BY dep\_id DESC;

# Image:



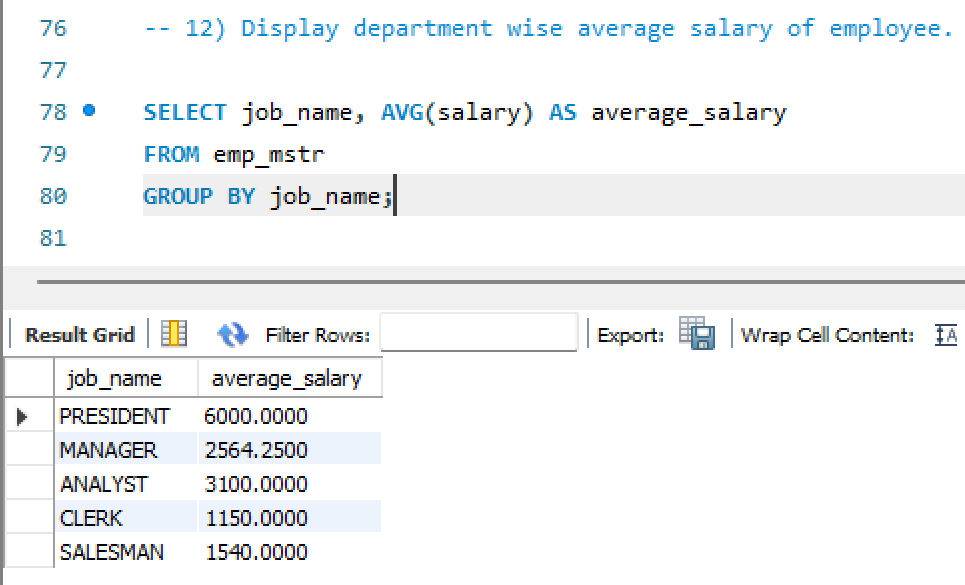
**-- 12) Display department wise average salary of employee.**

SELECT job\_name, AVG(salary) AS average\_salary

FROM emp\_mstr

GROUP BY job\_name;

# Image:



**-- 13)Display department id of the employee along with salary whose salary is maximum in respective department.**

SELECT dep\_id, emp\_name, salary

FROM emp\_mstr e

WHERE salary = (

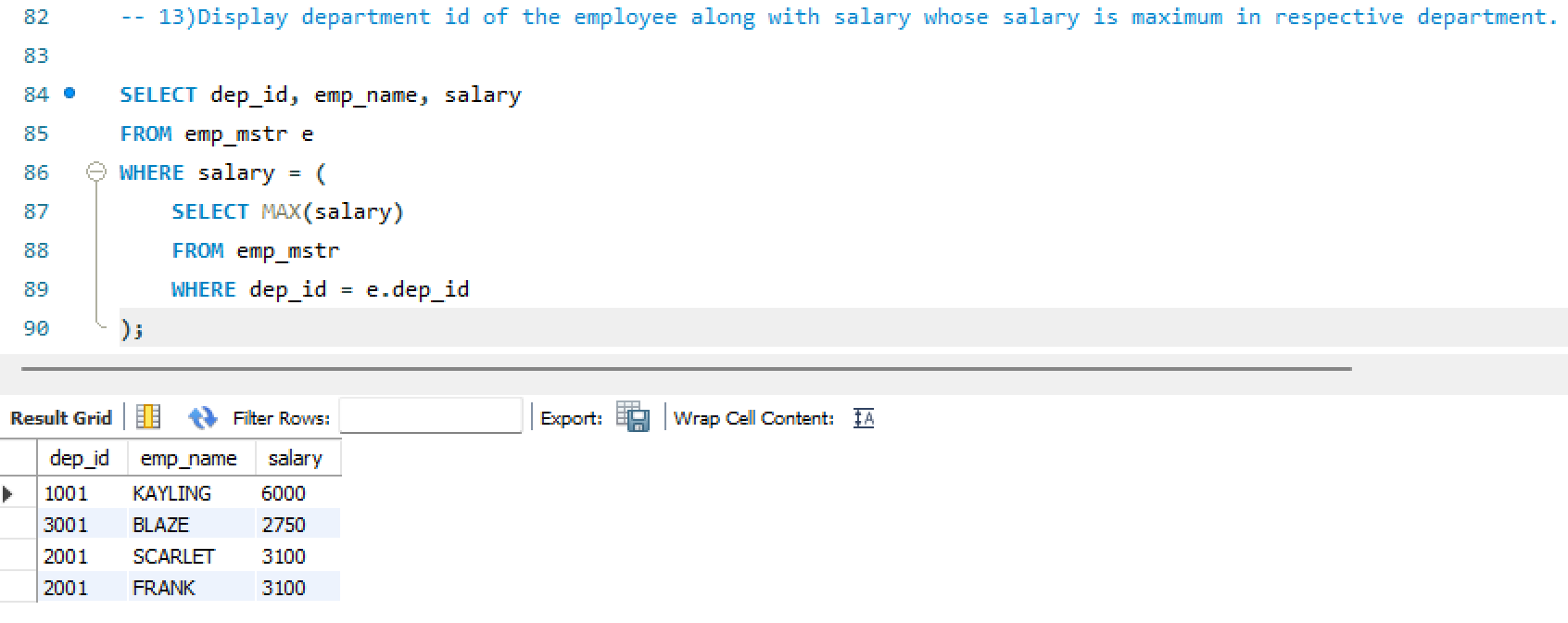
SELECT MAX(salary)

FROM emp\_mstr

WHERE dep\_id = e.dep\_id

);

# Image:



**-- 14)Display department id of the employee along with salary whose salary is minimum in respective department.**

SELECT dep\_id, emp\_name, salary

FROM emp\_mstr e

WHERE salary = (

SELECT MAX(salary)

FROM emp\_mstr

WHERE dep\_id = e.dep\_id

);

# Image:

