Name: Ayush Patel Enrollment No: 22162171038 Class B Batch 35

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 |
+---------+-------------+
```

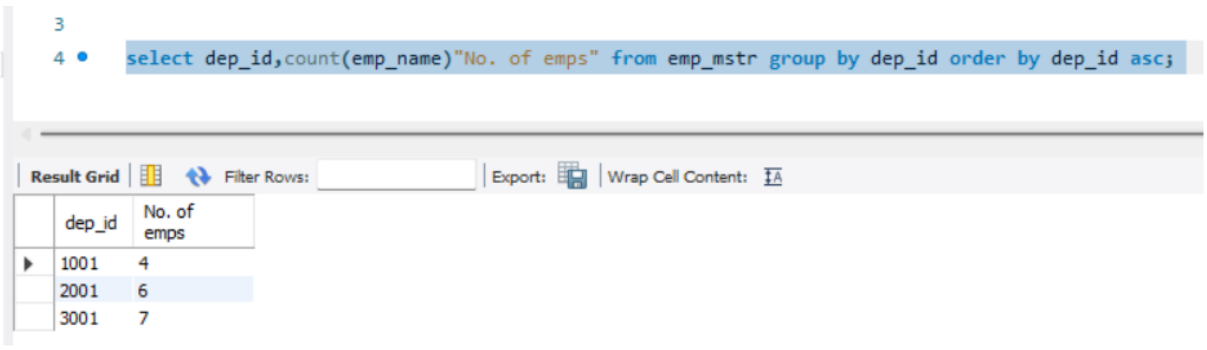
The SQL HAVING Clause

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;

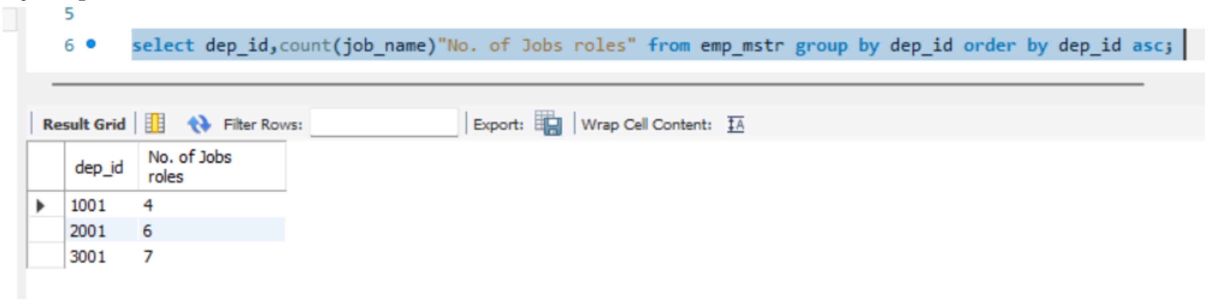


```
select dep_id,count(emp_name)"No. of emps" from emp_mstr group by dep_id order by dep_id asc;
```

dep_id	No. of emps
1001	4
2001	6
3001	7

2) 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;



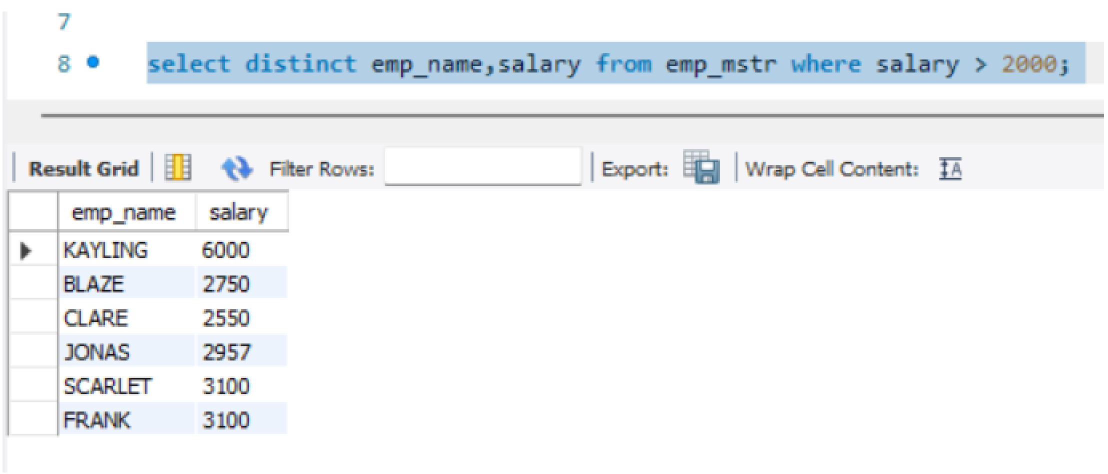
```
select dep_id,count(job_name)"No. of Jobs roles" from emp_mstr group by dep_id order by dep_id asc;
```

dep_id	No. of Jobs roles
1001	4
2001	6
3001	7

3) 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;

Name: Ayush Patel Enrollment No: 22162171038 Class B Batch 35



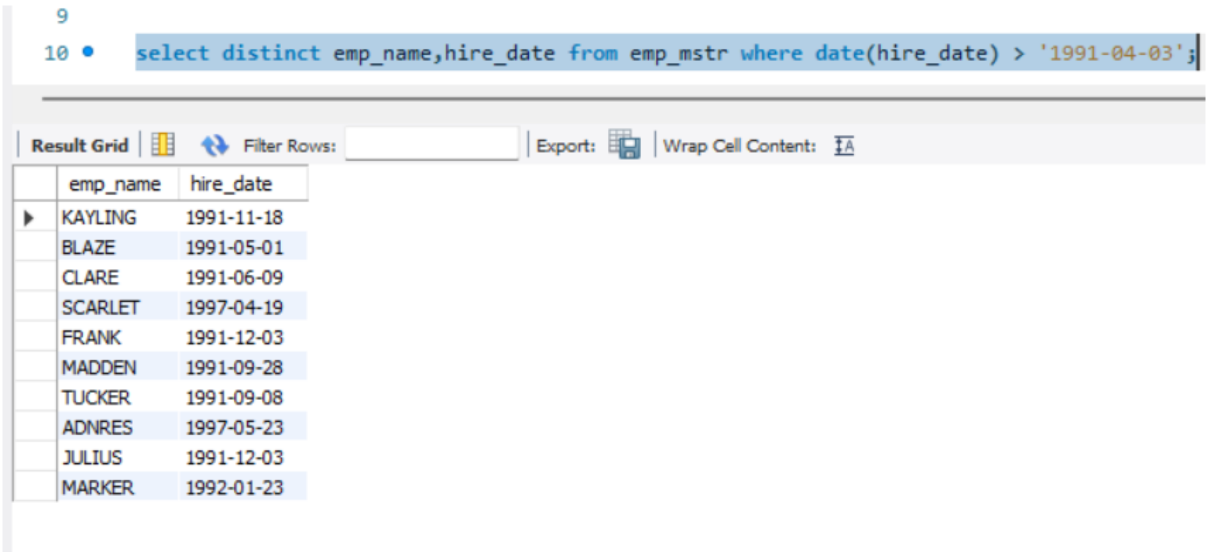
```
select distinct emp_name,salary from emp_mstr where salary > 2000;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_name	salary
KAYLING	6000
BLAZE	2750
CLARE	2550
JONAS	2957
SCARLET	3100
FRANK	3100

4) 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';



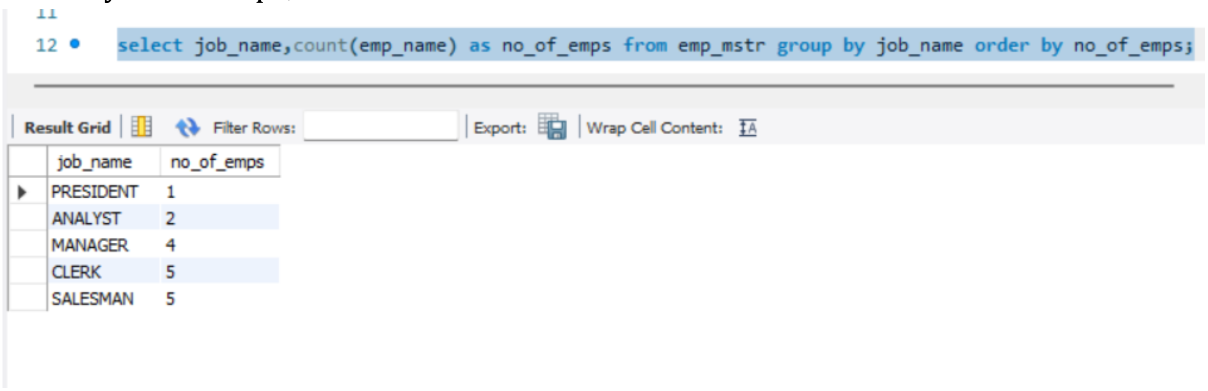
```
select distinct emp_name,hire_date from emp_mstr where date(hire_date) > '1991-04-03';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

emp_name	hire_date
KAYLING	1991-11-18
BLAZE	1991-05-01
CLARE	1991-06-09
SCARLET	1997-04-19
FRANK	1991-12-03
MADDEN	1991-09-28
TUCKER	1991-09-08
ADNRES	1997-05-23
JULIUS	1991-12-03
MARKER	1992-01-23

5) 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;



```
select job_name,count(emp_name) as no_of_emps from emp_mstr group by job_name order by no_of_emps;
```

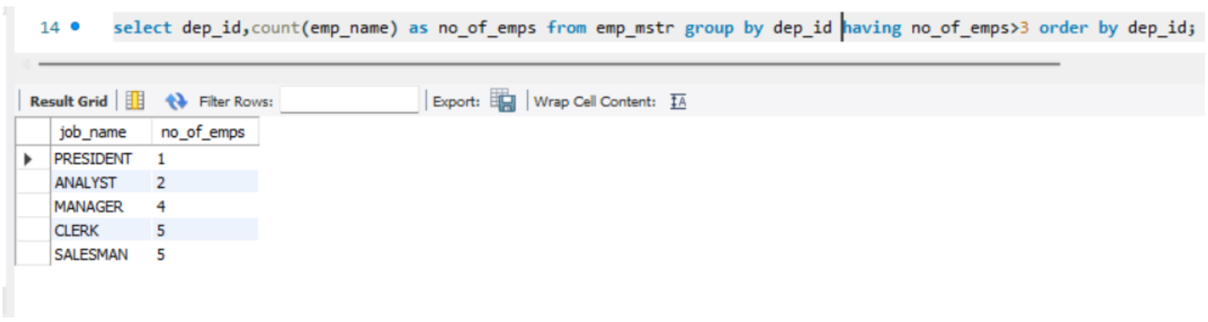
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

job_name	no_of_emps
PRESIDENT	1
ANALYST	2
MANAGER	4
CLERK	5
SALESMAN	5

6) 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;

Name: Ayush Patel Enrollment No: 22162171038 Class B Batch 35

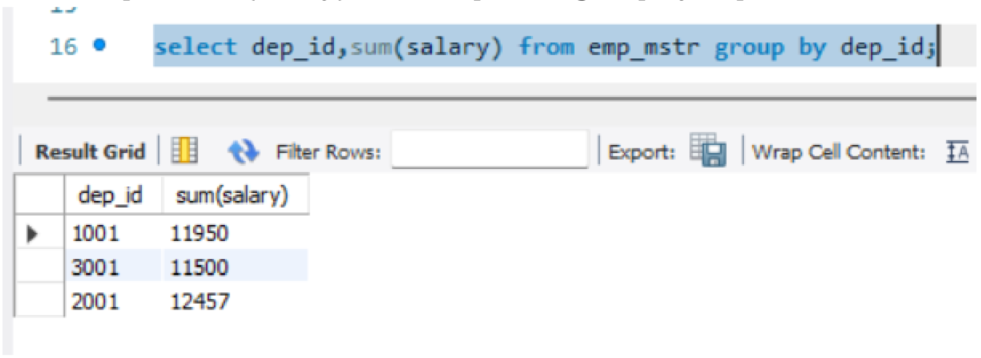


```sql
select dep_id,count(emp_name) as no_of_emps from emp_mstr group by dep_id having no_of_emps>3 order by dep_id;
```

job_name	no_of_emps
PRESIDENT	1
ANALYST	2
MANAGER	4
CLERK	5
SALESMAN	5

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

select dep_id,sum(salary) from emp_mstr group by dep_id;

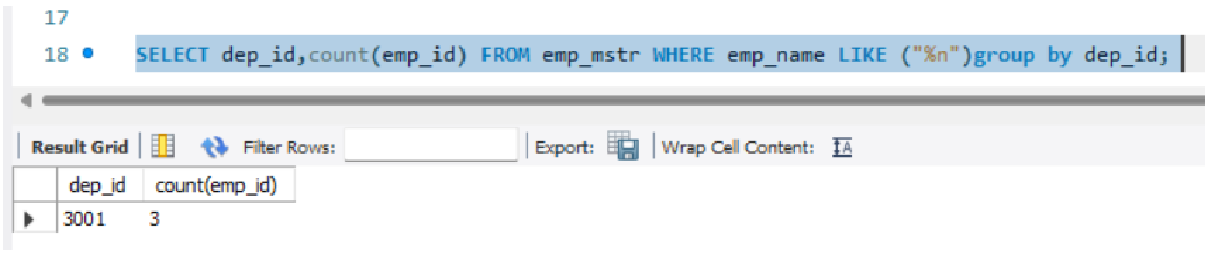


```sql
select dep_id,sum(salary) from emp_mstr group by dep_id;
```

dep_id	sum(salary)
1001	11950
3001	11500
2001	12457

8) 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;

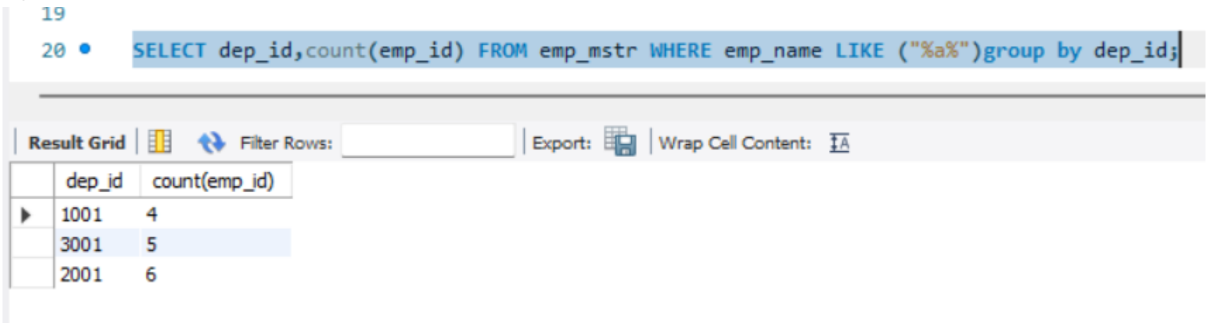


```sql
SELECT dep_id,count(emp_id) FROM emp_mstr WHERE emp_name LIKE ("%n")group by dep_id;
```

dep_id	count(emp_id)
3001	3

9) 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;

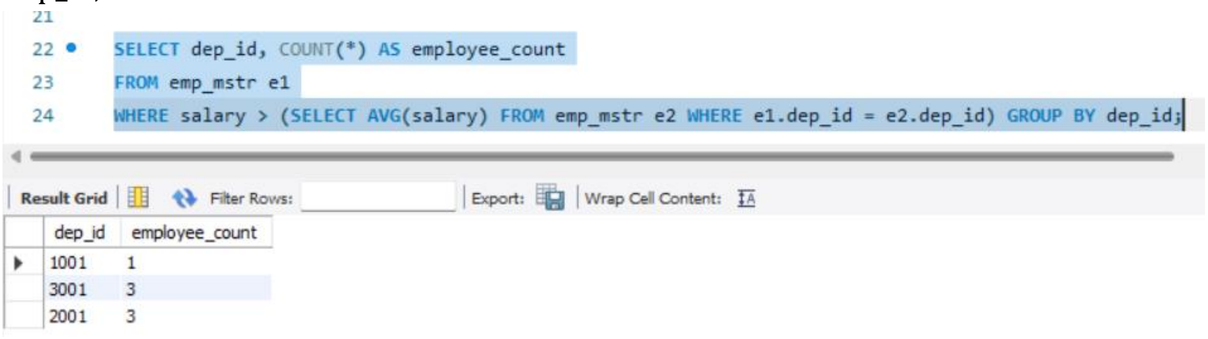


```sql
SELECT dep_id,count(emp_id) FROM emp_mstr WHERE emp_name LIKE ("%a%")group by dep_id;
```

dep_id	count(emp_id)
1001	4
3001	5
2001	6

10) 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;

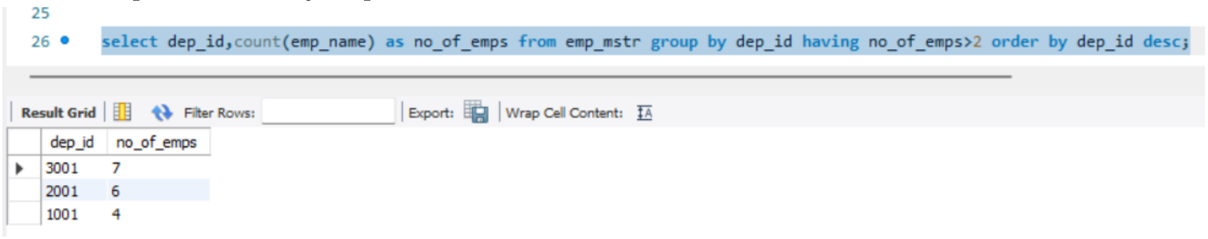


```
SELECT dep_id, COUNT(*) AS employee_count
FROM emp_mstr e1
WHERE salary > (SELECT AVG(salary) FROM emp_mstr e2 WHERE e1.dep_id = e2.dep_id) GROUP BY dep_id;
```

dep_id	employee_count
1001	1
3001	3
2001	3

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(emp_name) as no_of_emps from emp_mstr group by dep_id having no_of_emps>2 order by dep_id desc;

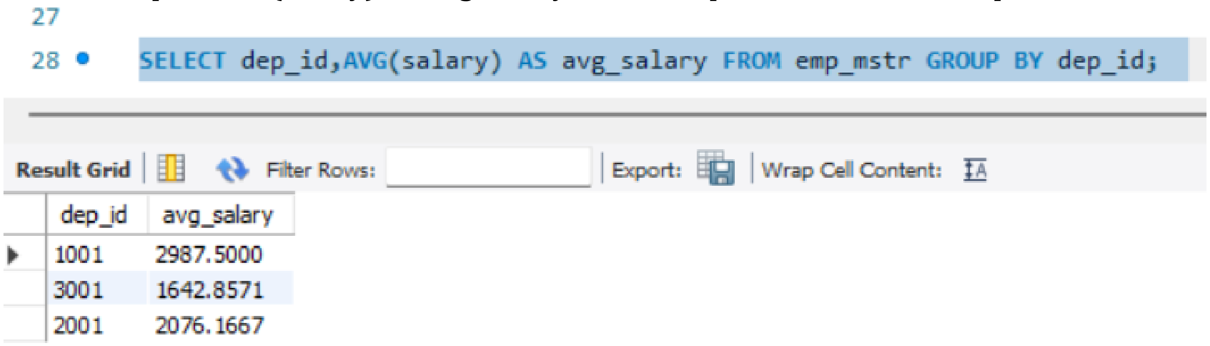


```
select dep_id,count(emp_name) as no_of_emps from emp_mstr group by dep_id having no_of_emps>2 order by dep_id desc;
```

dep_id	no_of_emps
3001	7
2001	6
1001	4

12) Display department wise average salary of employee.

SELECT dep_id,AVG(salary) AS avg_salary FROM emp_mstr GROUP BY dep_id;

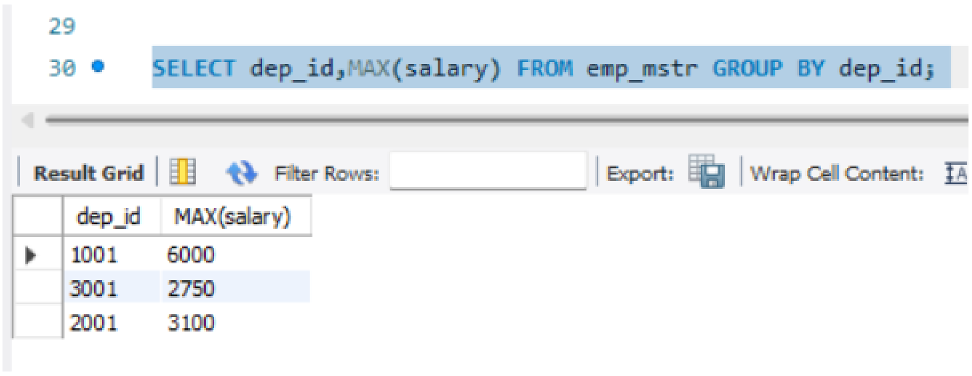


```
SELECT dep_id,AVG(salary) AS avg_salary FROM emp_mstr GROUP BY dep_id;
```

dep_id	avg_salary
1001	2987.5000
3001	1642.8571
2001	2076.1667

13) 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;

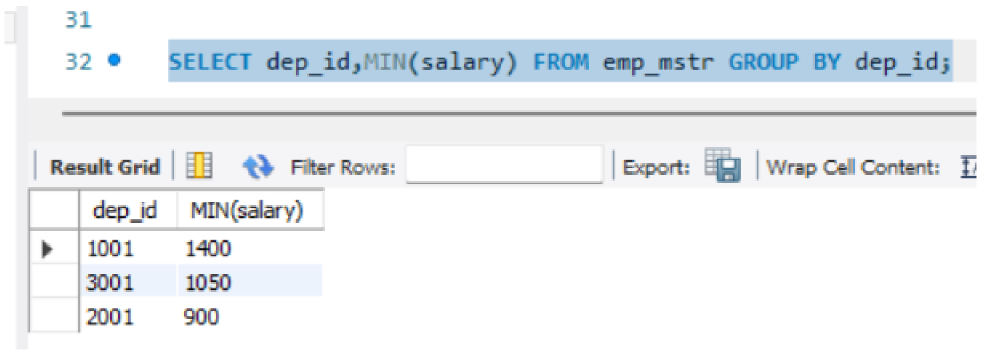


```
SELECT dep_id,MAX(salary) FROM emp_mstr GROUP BY dep_id;
```

dep_id	MAX(salary)
1001	6000
3001	2750
2001	3100

Name: Ayush Patel Enrollment No: 22162171038 Class B Batch 35

14) 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



```
SELECT dep_id,MIN(salary) FROM emp_mstr GROUP BY dep_id;
```

dep_id	MIN(salary)
1001	1400
3001	1050
2001	900