

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6

**Institute of computer technology**

**Ganpat university**

**B.tech. CSE (CBA/BDA/CS)**

**(2CSE301) DATABASE MANAGEMENT SYSTEM**

### **PRACTICAL 6**

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**

**Name : Yash Lakhtariya**  
**Enrollement number : 21162101012**  
**Branch : CBA , Batch – 31**  
**DBMS Practical 6**

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 –

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6

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

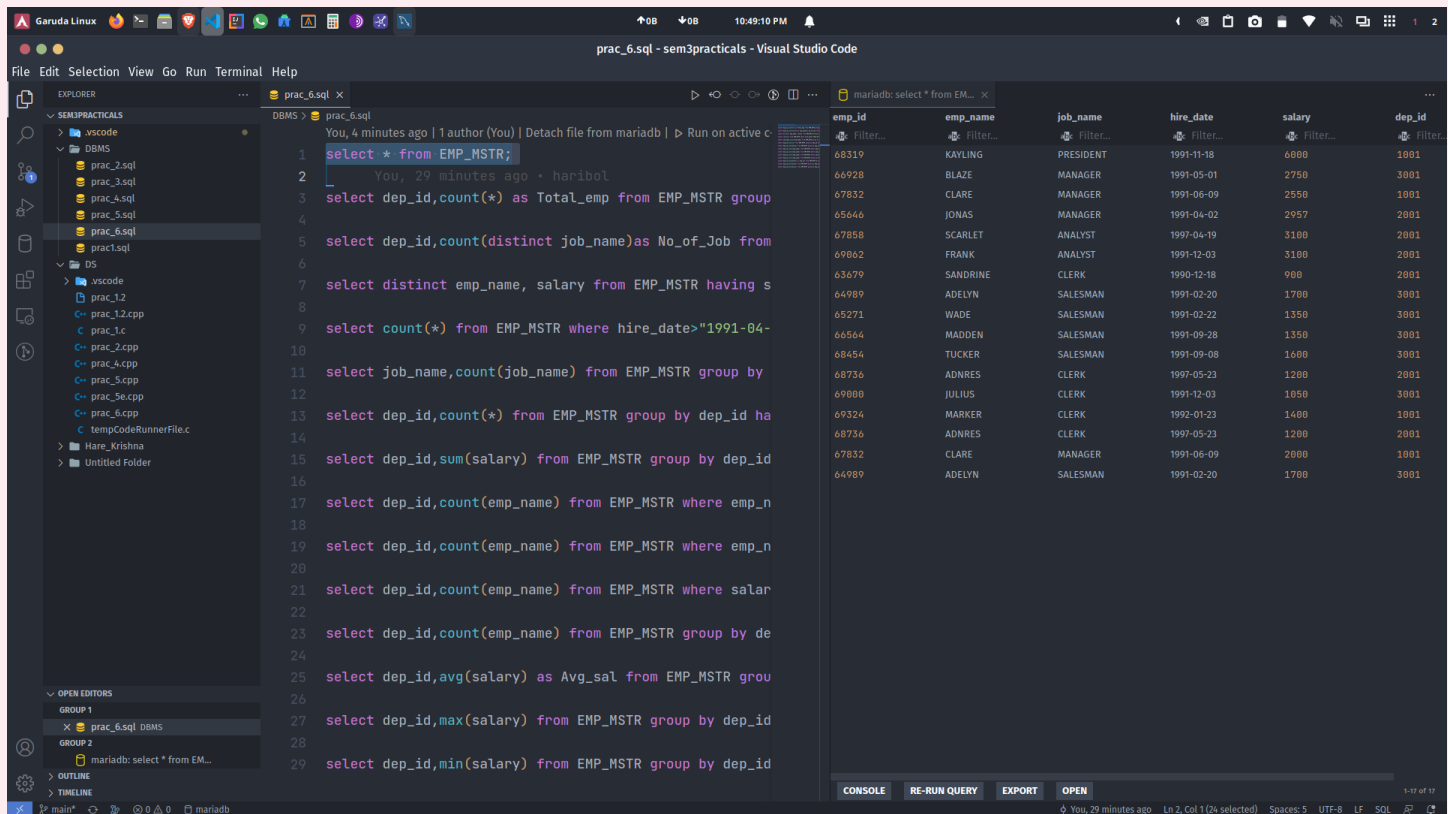
## The SQL HAVING Clause

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6

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

## Queries and Outputs :

```
select * from EMP_MSTR;
```



The screenshot shows a Visual Studio Code editor with a file explorer on the left, a central editor window with SQL queries, and a right-hand pane displaying the output of a query. The queries are as follows:

```
1 select * from EMP_MSTR;  
2  
3 You, 29 minutes ago * haribol  
4  
5 select dep_id,count(*) as Total_emp from EMP_MSTR group  
6  
7 select dep_id,count(distinct job_name)as No_of_Job from  
8  
9 select distinct emp_name, salary from EMP_MSTR having s  
10  
11 select count(*) from EMP_MSTR where hire_date>"1991-04-  
12  
13 select job_name,count(job_name) from EMP_MSTR group by  
14  
15 select dep_id,count(*) from EMP_MSTR group by dep_id ha  
16  
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n  
18  
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n  
20  
21 select dep_id,count(emp_name) from EMP_MSTR where salar  
22  
23 select dep_id,count(emp_name) from EMP_MSTR group by de  
24  
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou  
26  
27 select dep_id,max(salary) from EMP_MSTR group by dep_id  
28  
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

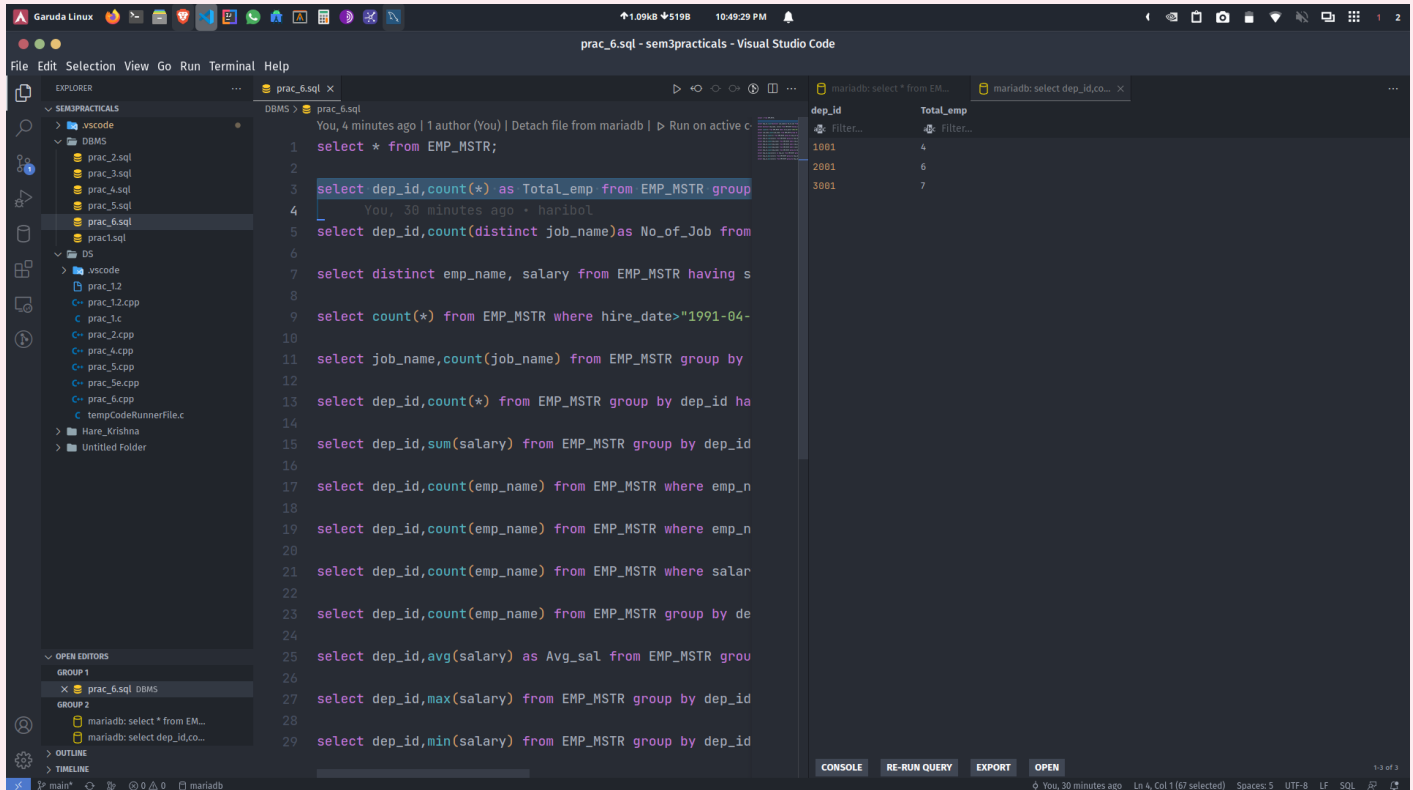
The output table shows the following data:

emp_id	emp_name	job_name	hire_date	salary	dep_id
68319	KAYLING	PRESIDENT	1991-11-18	6000	1001
66928	BLAZE	MANAGER	1991-05-01	2750	3001
67832	CLARE	MANAGER	1991-06-09	2550	1001
65646	JONAS	MANAGER	1991-04-02	2957	2001
67858	SCARLET	ANALYST	1997-04-19	3100	2001
69862	FRANK	ANALYST	1991-12-03	3100	2001
63679	SANDRINE	CLERK	1990-12-18	980	2001
64989	ADELYN	SALESMAN	1991-02-20	1700	3001
65271	WADE	SALESMAN	1991-02-22	1350	3001
66564	MADDEN	SALESMAN	1991-09-28	1350	3001
68454	TUCKER	SALESMAN	1991-09-08	1600	3001
68736	ADNRES	CLERK	1997-05-23	1200	2001
69800	JULIUS	CLERK	1991-12-03	1050	3001
69324	MARKER	CLERK	1992-01-23	1400	1001
68736	ADNRES	CLERK	1997-05-23	1200	2001
67832	CLARE	MANAGER	1991-06-09	2000	1001
64989	ADELYN	SALESMAN	1991-02-20	1700	3001

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6

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

```
select dep_id,count(*) as Total_emp from EMP_MSTR group by  
dep_id;
```



The screenshot shows a Visual Studio Code window with a file explorer on the left, a code editor in the center, and a database client on the right. The code editor contains a SQL query to count employees by department. The database client shows the results of the query.

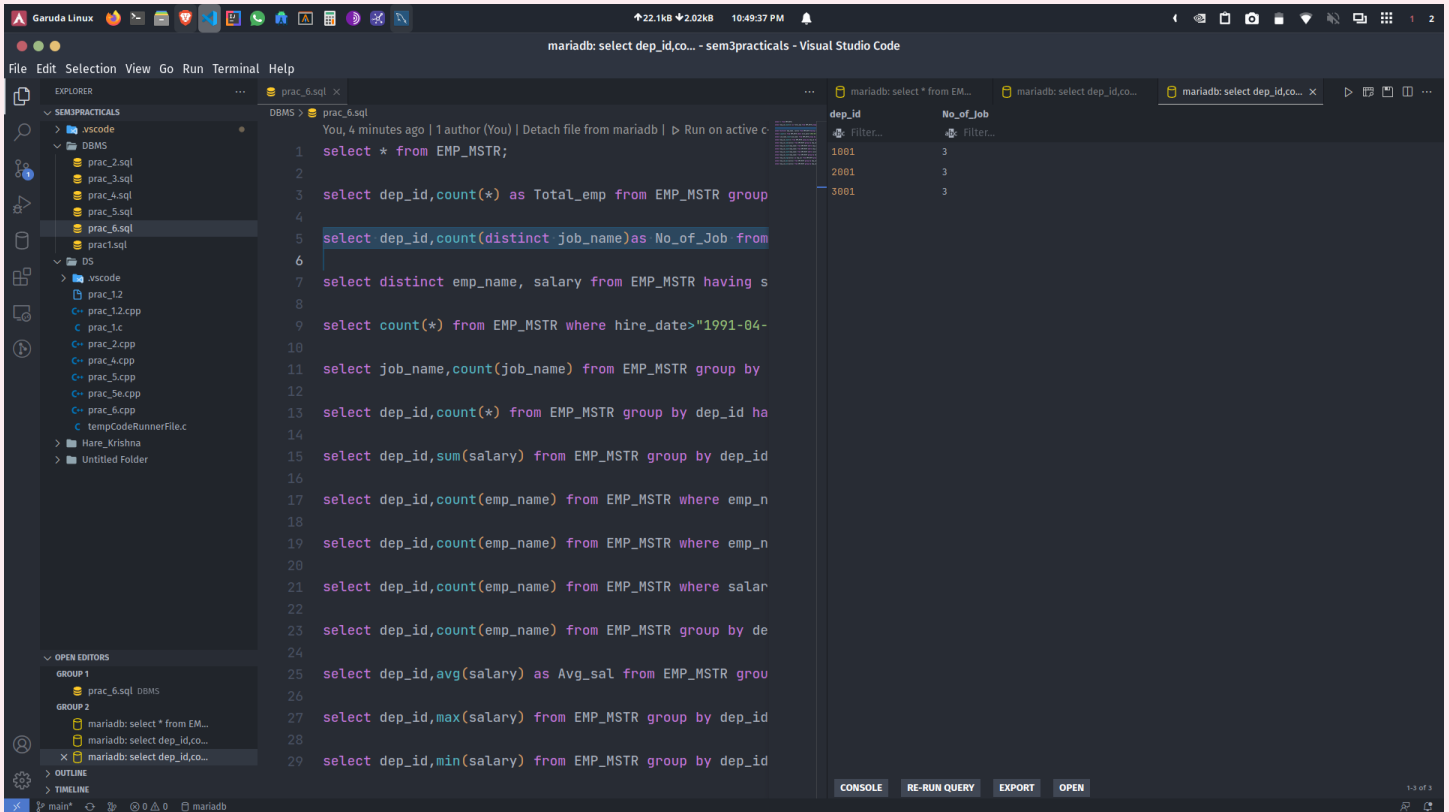
```
1 select * from EMP_MSTR;  
2  
3 select dep_id,count(*) as Total_emp from EMP_MSTR group  
4 by dep_id;  
5  
6 select dep_id,count(distinct job_name)as No_of_Job from  
7 EMP_MSTR group by dep_id;  
8  
9 select distinct emp_name, salary from EMP_MSTR having s  
10 alary > 10000;  
11  
12 select job_name,count(job_name) from EMP_MSTR group by  
13 job_name;  
14  
15 select dep_id,count(*) from EMP_MSTR group by dep_id ha  
16 ving s  
17 alary > 10000;  
18  
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n  
20 ame < 'A';  
21  
22 select dep_id,count(emp_name) from EMP_MSTR where salar  
23 y > 10000;  
24  
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou  
26 p by dep_id;  
27  
28 select dep_id,max(salary) from EMP_MSTR group by dep_id  
29  
30 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

dep_id	Total_emp
1001	4
2001	6
3001	7

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6

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

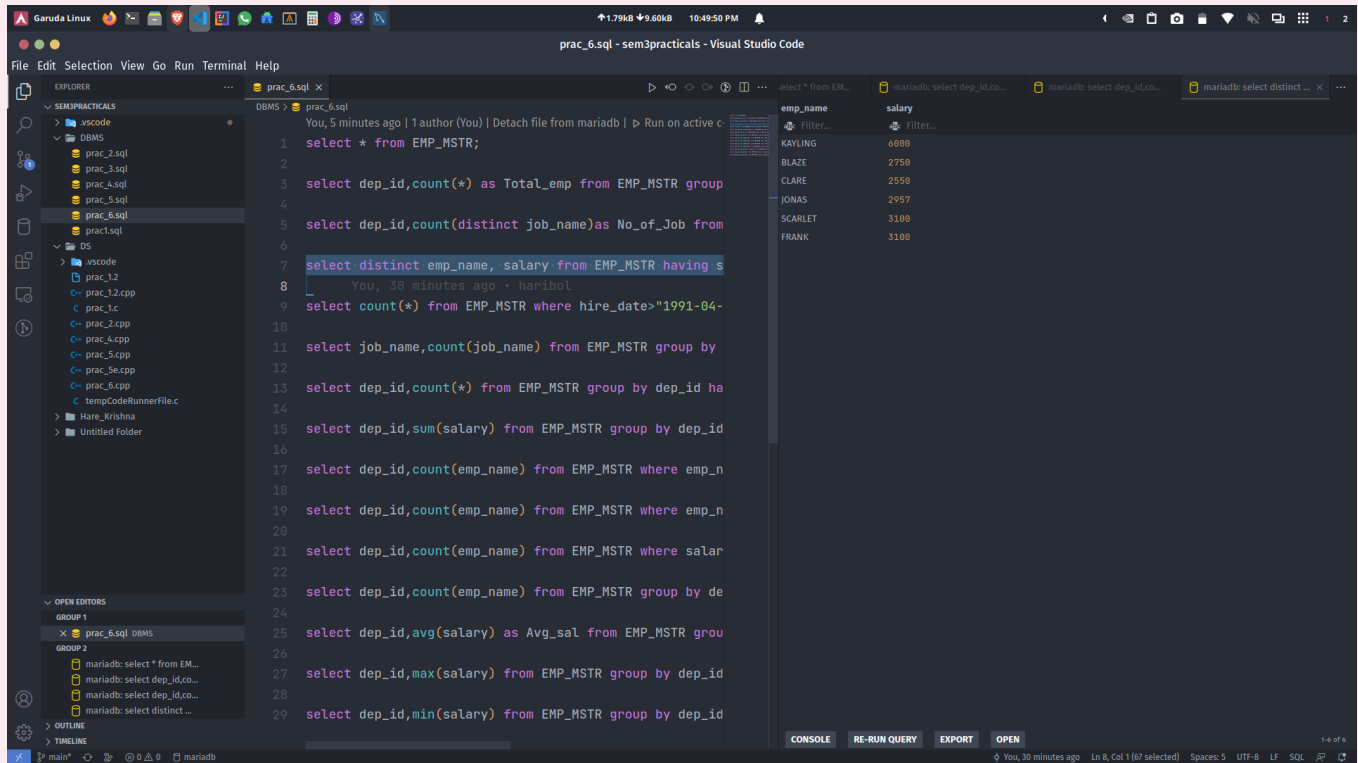
```
select dep_id,count(distinct job_name)as No_of_Job from EMP_MSTR  
group by dep_id;
```



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 having  
salary>2000;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



The screenshot shows the Visual Studio Code interface with a file named `prac_6.sql` open. The file contains a series of SQL queries. The Output window on the right displays the results of the query `select distinct emp_name, salary from EMP_MSTR having salary > 6000`, showing a list of employees and their salaries.

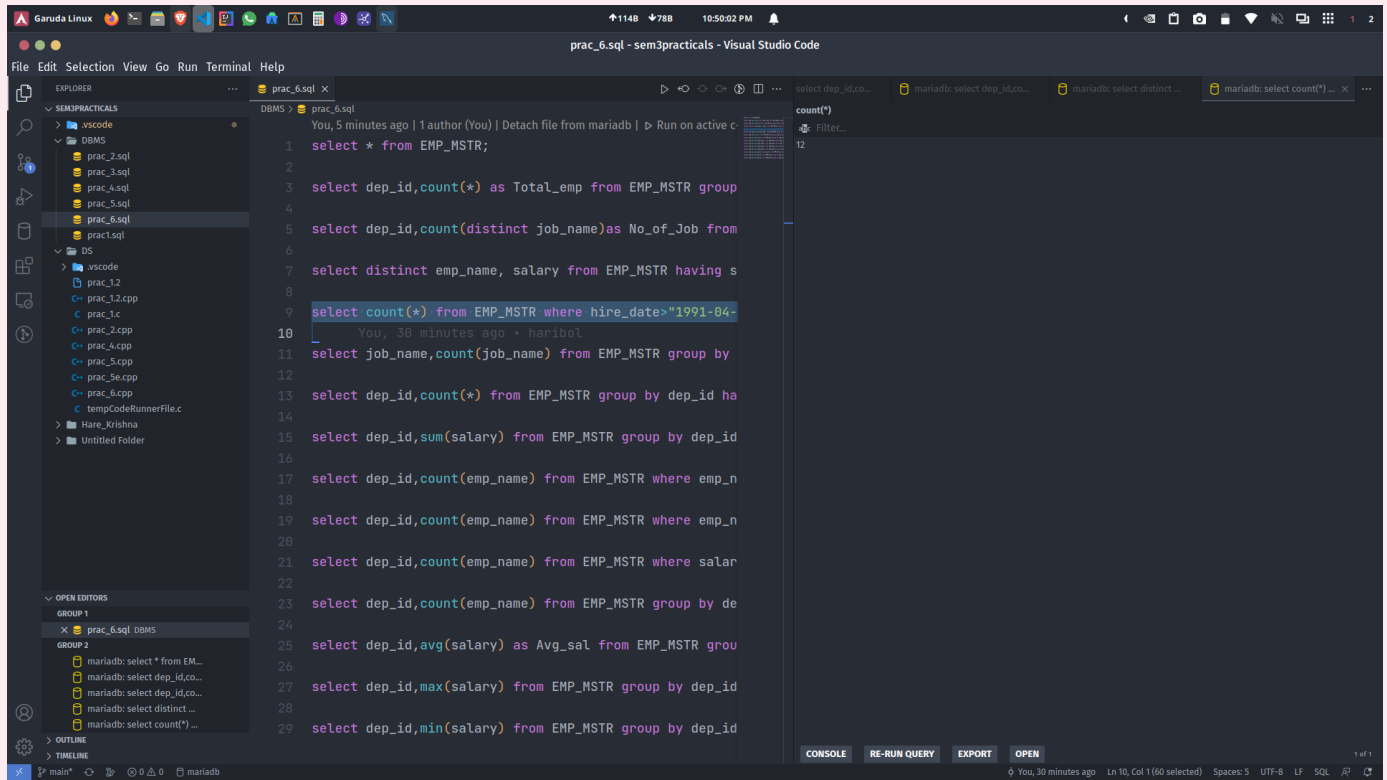
```
1 select * from EMP_MSTR;
2
3 select dep_id, count(*) as Total_emp from EMP_MSTR group by dep_id;
4
5 select dep_id, count(distinct job_name) as No_of_Job from EMP_MSTR group by dep_id;
6
7 select distinct emp_name, salary from EMP_MSTR having salary > 6000;
8
9 select count(*) from EMP_MSTR where hire_date > "1991-04-03";
10
11 select job_name, count(job_name) from EMP_MSTR group by job_name;
12
13 select dep_id, count(*) from EMP_MSTR group by dep_id having count(*) > 1;
14
15 select dep_id, sum(salary) from EMP_MSTR group by dep_id;
16
17 select dep_id, count(emp_name) from EMP_MSTR where emp_name < "A";
18
19 select dep_id, count(emp_name) from EMP_MSTR where emp_name < "B";
20
21 select dep_id, count(emp_name) from EMP_MSTR where salary < 5000;
22
23 select dep_id, count(emp_name) from EMP_MSTR group by dep_id;
24
25 select dep_id, avg(salary) as Avg_sal from EMP_MSTR group by dep_id;
26
27 select dep_id, max(salary) from EMP_MSTR group by dep_id;
28
29 select dep_id, min(salary) from EMP_MSTR group by dep_id;
```

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 count(*) from EMP_MSTR where hire_date > "1991-04-03";
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



The screenshot shows the Visual Studio Code interface with a DBMS window open. The DBMS window displays a list of SQL queries. The query at line 10 is highlighted, which is: `select count(*) from EMP_MSTR where hire_date>"1991-04-`. The query is part of a larger block of SQL code. The Explorer panel on the left shows a file tree with folders for 'SEM3PRACTICALS', 'DBMS', 'DS', and 'Hare\_Krishna'. The DBMS folder is expanded, showing a list of SQL files. The DBMS window also shows a list of queries, with the highlighted query being the 10th one. The status bar at the bottom indicates the current file is 'main\*' and the cursor is at line 10, column 1.

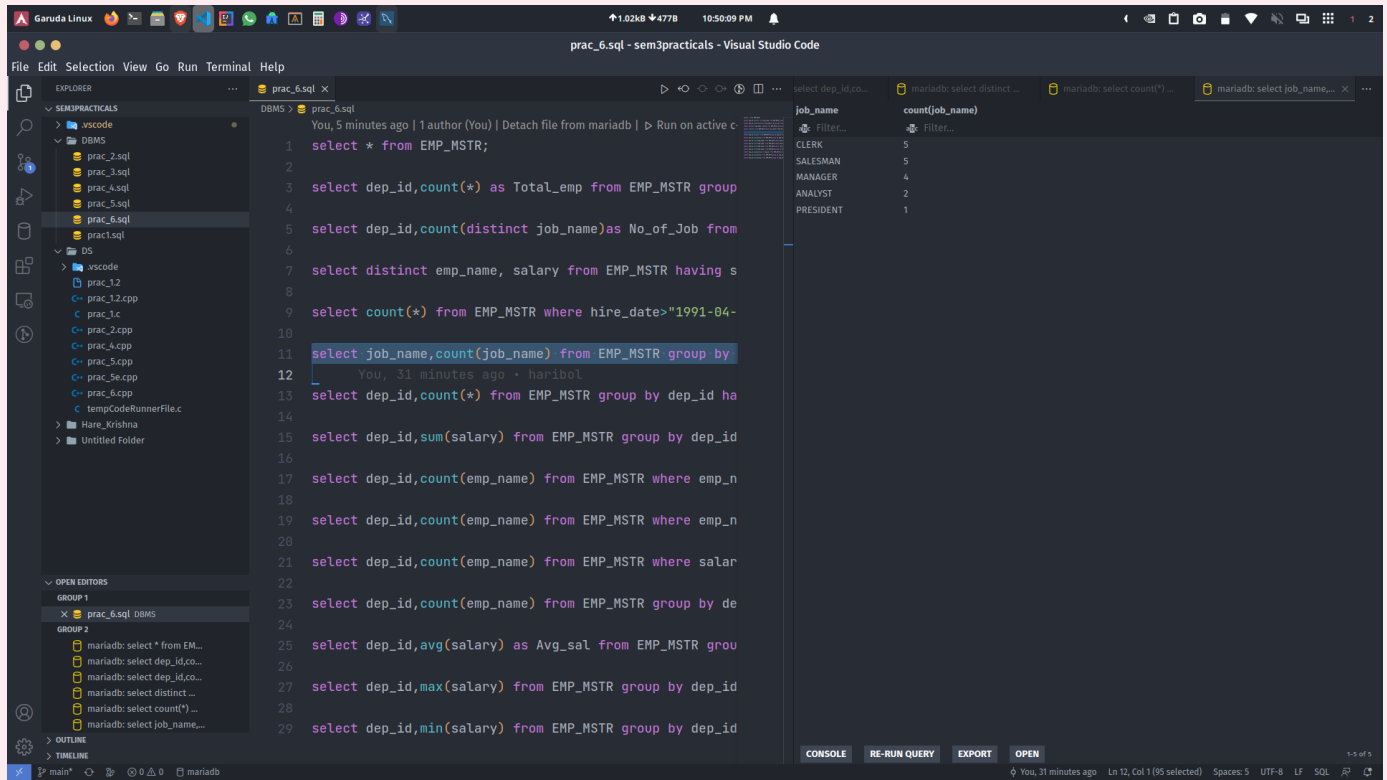
```
1 select * from EMP_MSTR;
2
3 select dep_id,count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id,count(distinct job_name)as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date>"1991-04-
10 You, 30 minutes ago • haribol
11 select job_name,count(job_name) from EMP_MSTR group by
12
13 select dep_id,count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id,sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n
20
21 select dep_id,count(emp_name) from EMP_MSTR where salar
22
23 select dep_id,count(emp_name) from EMP_MSTR group by de
24
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id,max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

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

```
select job_name,count(job_name) from EMP_MSTR group by job_name order
by count(job_name) desc;
```



Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



The screenshot shows the Visual Studio Code interface with a SQL script in 'prac\_6.sql' and its execution results in the 'CONSOLE' panel. The script contains 29 lines of SQL queries. The console shows the results of the 11th query, which is a grouped count of job names by department ID.

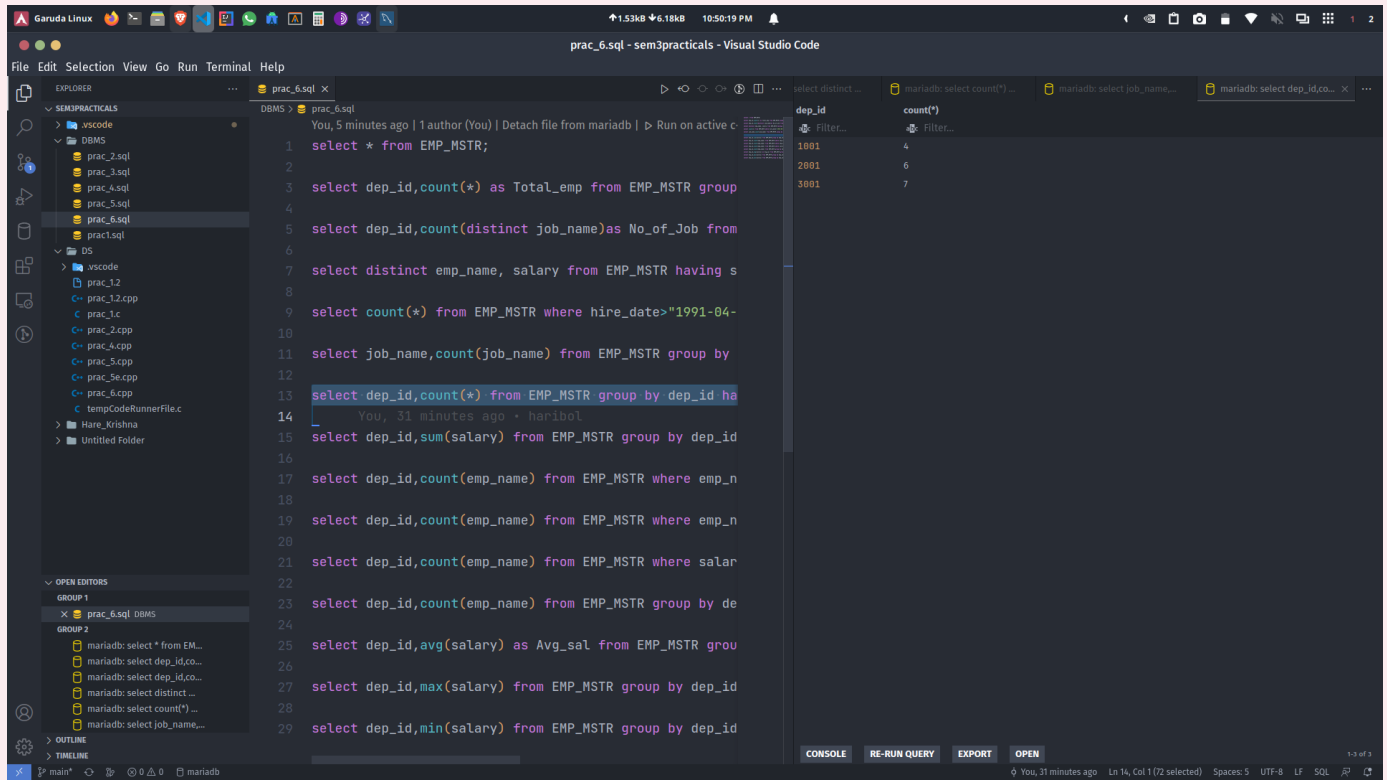
```
1 select * from EMP_MSTR;
2
3 select dep_id,count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id,count(distinct job_name)as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date>"1991-04-
10
11 select job_name,count(job_name) from EMP_MSTR group by
12
13 select dep_id,count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id,sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n
20
21 select dep_id,count(emp_name) from EMP_MSTR where salar
22
23 select dep_id,count(emp_name) from EMP_MSTR group by de
24
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id,max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

job_name	count(job_name)
CLERK	5
SALESMAN	5
MANAGER	4
ANALYST	2
PRESIDENT	1

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

```
select dep_id,count(*) from EMP_MSTR group by dep_id having
count(*)>3;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



The screenshot shows a Visual Studio Code editor window titled "prac\_6.sql - sem3practicals - Visual Studio Code". The editor is open to a file named "prac\_6.sql" which contains a series of SQL queries. The queries are as follows:

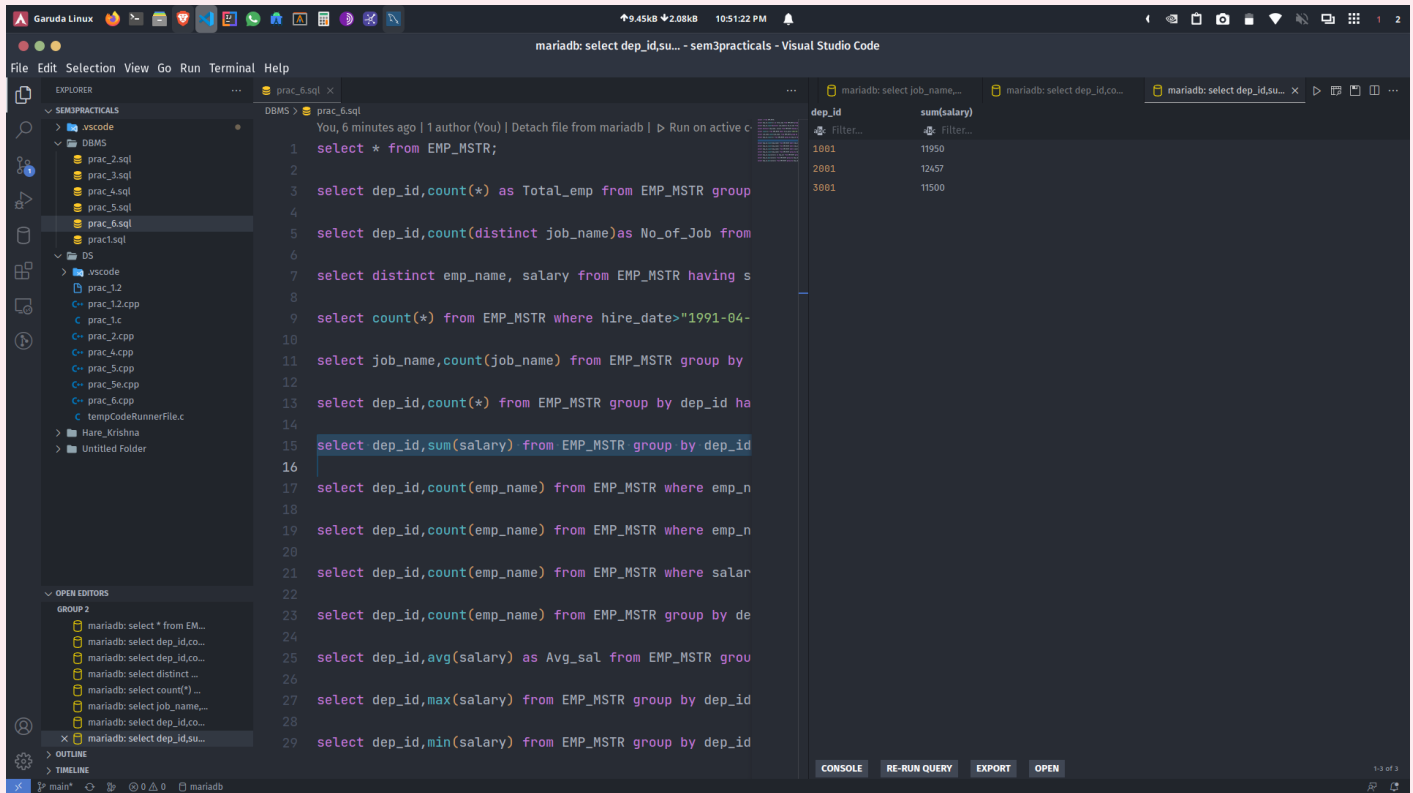
```
1 select * from EMP_MSTR;  
2  
3 select dep_id,count(*) as Total_emp from EMP_MSTR group  
4  
5 select dep_id,count(distinct job_name)as No_of_Job from  
6  
7 select distinct emp_name, salary from EMP_MSTR having s  
8  
9 select count(*) from EMP_MSTR where hire_date>"1991-04-  
10  
11 select job_name,count(job_name) from EMP_MSTR group by  
12  
13 select dep_id,count(*) from EMP_MSTR group by dep_id ha  
14 You, 31 minutes ago · haribol  
15 select dep_id,sum(salary) from EMP_MSTR group by dep_id  
16  
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n  
18  
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n  
20  
21 select dep_id,count(emp_name) from EMP_MSTR where salar  
22  
23 select dep_id,count(emp_name) from EMP_MSTR group by de  
24  
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou  
26  
27 select dep_id,max(salary) from EMP_MSTR group by dep_id  
28  
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

The console output shows the results of the queries. The first query returns all rows from the EMP\_MSTR table. The second query returns the count of distinct job names for each department. The third query returns the count of employees for each department. The fourth query returns the count of employees for each department, grouped by department ID. The fifth query returns the count of employees for each department, grouped by department ID. The sixth query returns the count of employees for each department, grouped by department ID. The seventh query returns the count of employees for each department, grouped by department ID. The eighth query returns the count of employees for each department, grouped by department ID. The ninth query returns the count of employees for each department, grouped by department ID. The tenth query returns the count of employees for each department, grouped by department ID. The eleventh query returns the count of employees for each department, grouped by department ID. The twelfth query returns the count of employees for each department, grouped by department ID. The thirteenth query returns the count of employees for each department, grouped by department ID. The fourteenth query returns the count of employees for each department, grouped by department ID. The fifteenth query returns the count of employees for each department, grouped by department ID. The sixteenth query returns the count of employees for each department, grouped by department ID. The seventeenth query returns the count of employees for each department, grouped by department ID. The eighteenth query returns the count of employees for each department, grouped by department ID. The nineteenth query returns the count of employees for each department, grouped by department ID. The twentieth query returns the count of employees for each department, grouped by department ID. The twenty-first query returns the count of employees for each department, grouped by department ID. The twenty-second query returns the count of employees for each department, grouped by department ID. The twenty-third query returns the count of employees for each department, grouped by department ID. The twenty-fourth query returns the count of employees for each department, grouped by department ID. The twenty-fifth query returns the count of employees for each department, grouped by department ID. The twenty-sixth query returns the count of employees for each department, grouped by department ID. The twenty-seventh query returns the count of employees for each department, grouped by department ID. The twenty-eighth query returns the count of employees for each department, grouped by department ID. The twenty-ninth query returns the count of employees for each department, grouped by department ID. The thirtieth query returns the count of employees for each department, grouped by department ID.

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

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

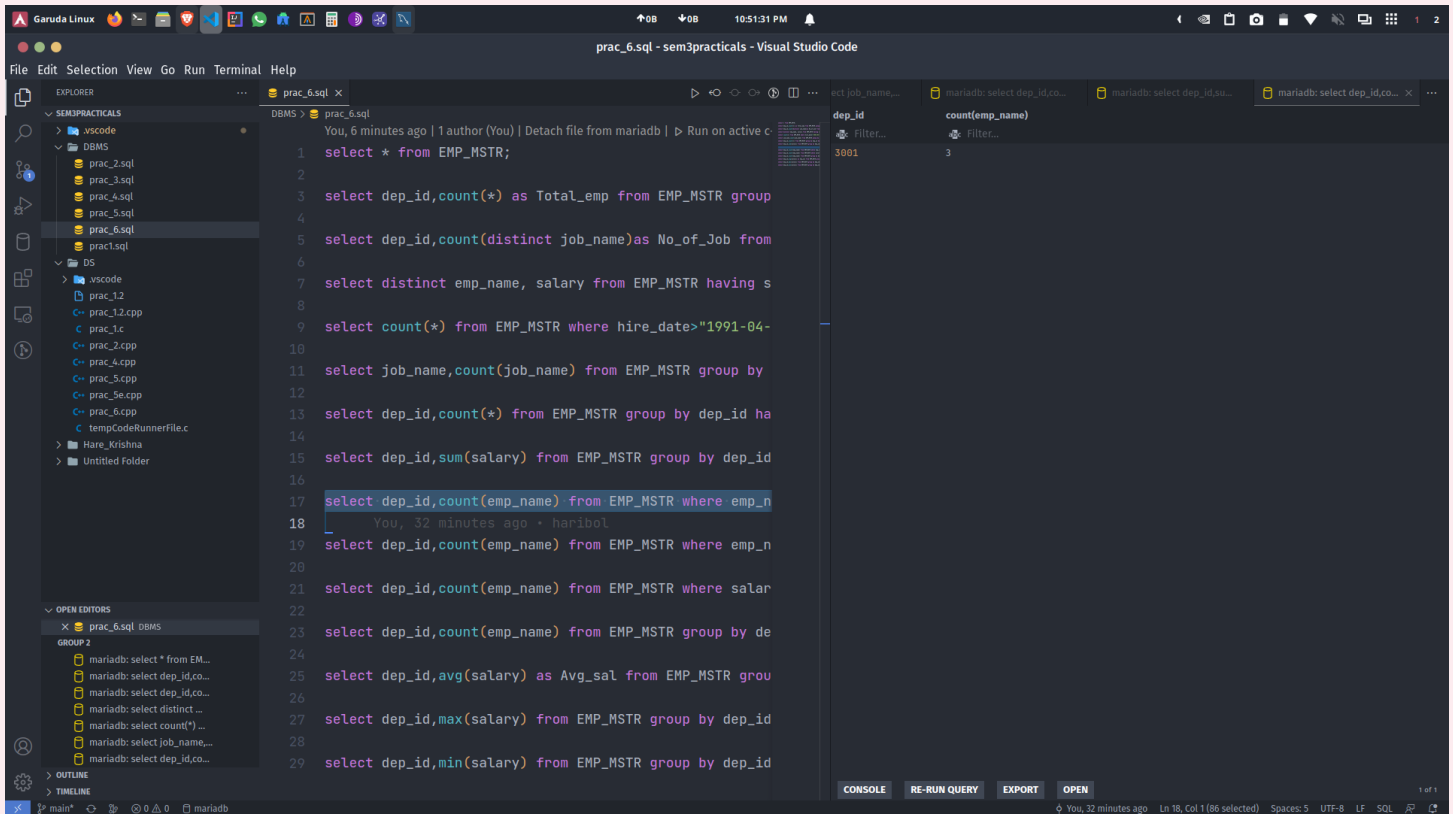
Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



8) Count total number of employees assigned in each department whose name end with "n".

```
select dep_id,count(emp_name) from EMP_MSTR where emp_name like  
'%n' group by dep_id;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



```
1 select * from EMP_MSTR;
2
3 select dep_id,count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id,count(distinct job_name)as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date>"1991-04-
10
11 select job_name,count(job_name) from EMP_MSTR group by
12
13 select dep_id,count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id,sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n
20
21 select dep_id,count(emp_name) from EMP_MSTR where salar
22
23 select dep_id,count(emp_name) from EMP_MSTR group by de
24
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id,max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

dep_id	count(emp_name)
3801	3

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

```
select dep_id,count(emp_name) from EMP_MSTR where emp_name like
'%a%' group by dep_id;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6

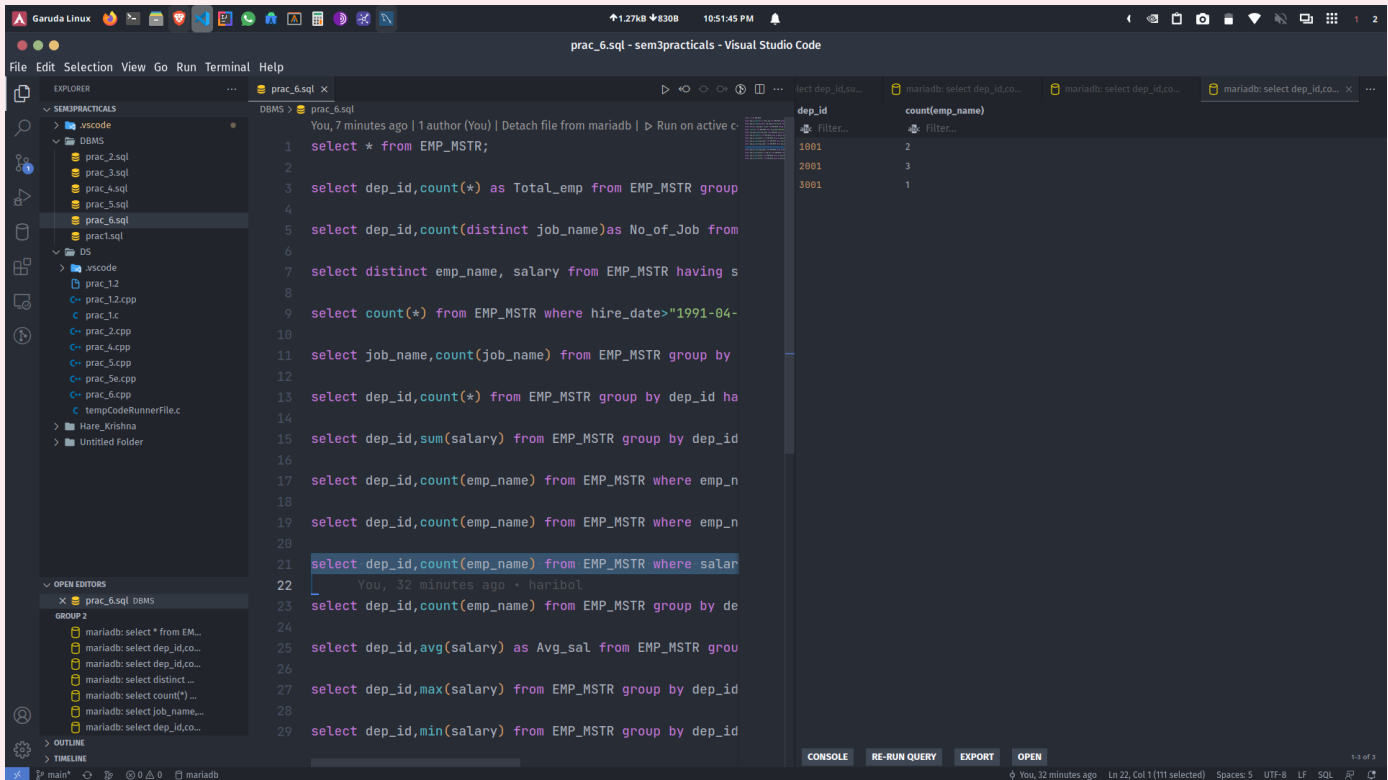
```
1 select * from EMP_MSTR;
2
3 select dep_id,count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id,count(distinct job_name)as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date>"1991-04-
10
11 select job_name,count(job_name) from EMP_MSTR group by
12
13 select dep_id,count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id,sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n
20
21 select dep_id,count(emp_name) from EMP_MSTR where salar
22
23 select dep_id,count(emp_name) from EMP_MSTR group by de
24
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id,max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

dep_id	count(emp_name)
1001	4
2001	6
3001	5

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

```
select dep_id,count(emp_name) from EMP_MSTR where salary >
(select avg(salary) from EMP_MSTR) group by dep_id;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



```
1 select * from EMP_MSTR;
2
3 select dep_id, count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id, count(distinct job_name) as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date > "1991-04-
10
11 select job_name, count(job_name) from EMP_MSTR group by
12
13 select dep_id, count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id, sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id, count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id, count(emp_name) from EMP_MSTR where emp_n
20
21 select dep_id, count(emp_name) from EMP_MSTR where salar
22
23 select dep_id, count(emp_name) from EMP_MSTR group by de
24
25 select dep_id, avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id, max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id, min(salary) from EMP_MSTR group by dep_id
```

dep_id	count(emp_name)
1001	2
2001	3
3001	1

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) from EMP_MSTR group by dep_id
having count(emp_name) > 2 order by dep_id desc;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6

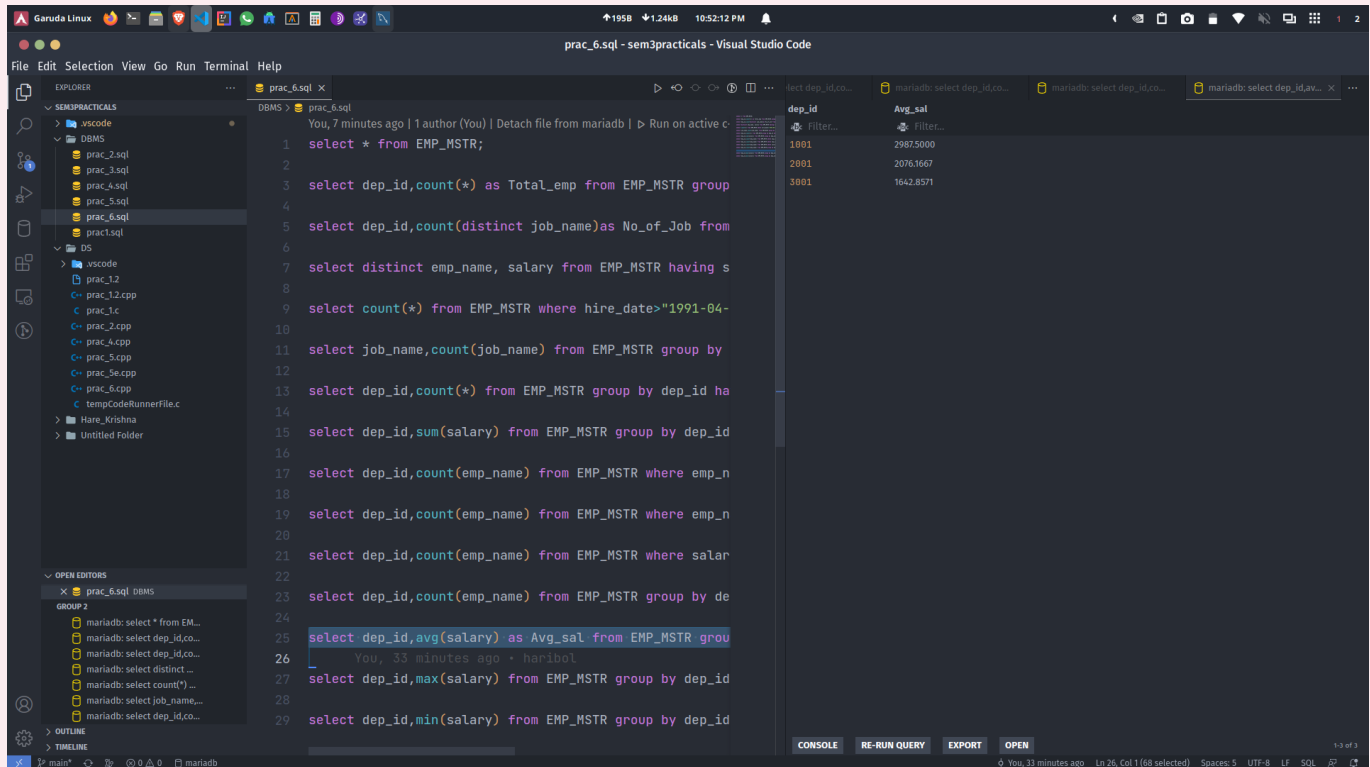
```
1 select * from EMP_MSTR;
2
3 select dep_id, count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id, count(distinct job_name) as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date > "1991-04-
10
11 select job_name, count(job_name) from EMP_MSTR group by
12
13 select dep_id, count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id, sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id, count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id, count(emp_name) from EMP_MSTR where emp_n
20
21 select dep_id, count(emp_name) from EMP_MSTR where salar
22
23 select dep_id, count(emp_name) from EMP_MSTR group by de
24
25 select dep_id, avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id, max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id, min(salary) from EMP_MSTR group by dep_id
```

dep_id	count(emp_name)
3001	7
2001	6
1001	4

12) Display department wise average salary of employee.

```
select dep_id, avg(salary) as Avg_sal from EMP_MSTR group by
dep_id;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



The screenshot shows the Visual Studio Code interface with a SQL query in the editor and its results in a table. The query is:

```
select dep_id,avg(salary) as Avg_sal from EMP_MSTR group by dep_id;
```

The results table shows the following data:

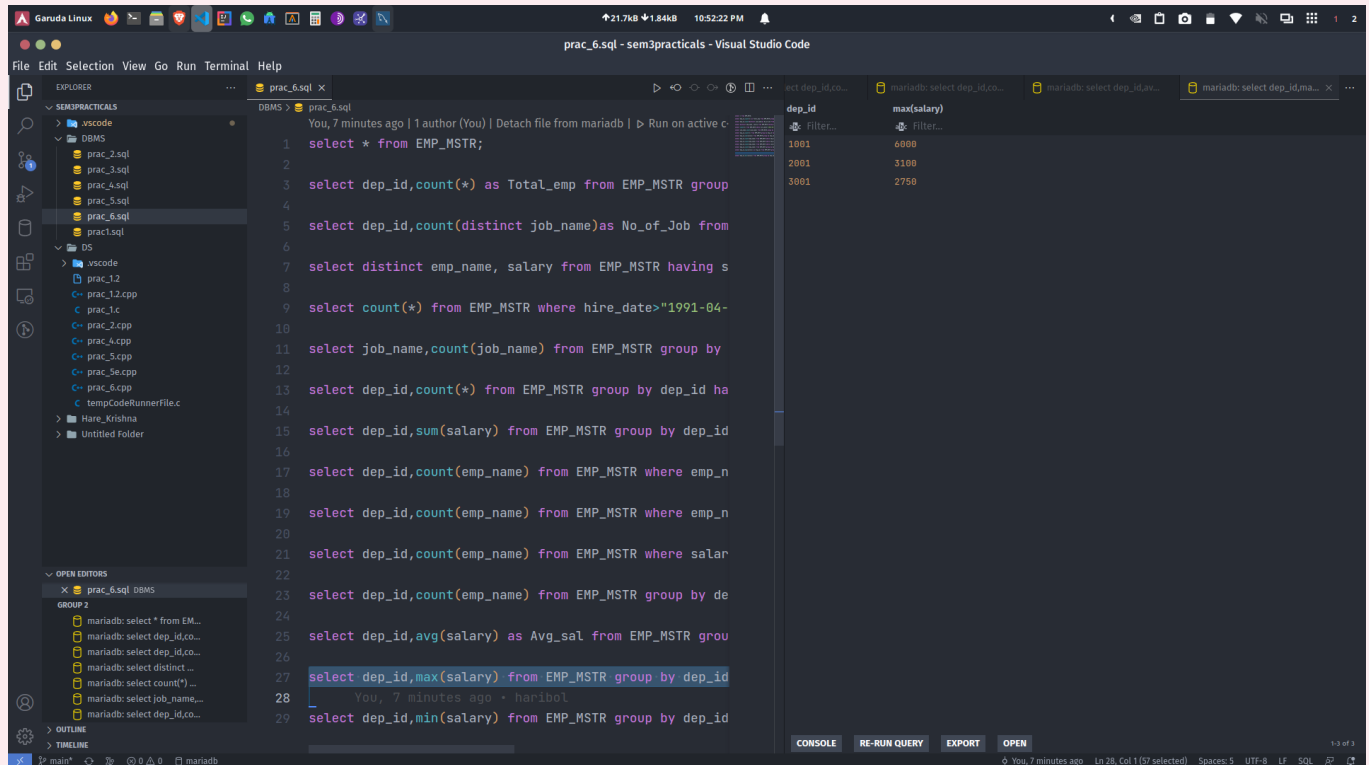
dep_id	Avg_sal
1001	2987.5000
2001	2076.1667
3001	1642.8571

13) Display name 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;
```



Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



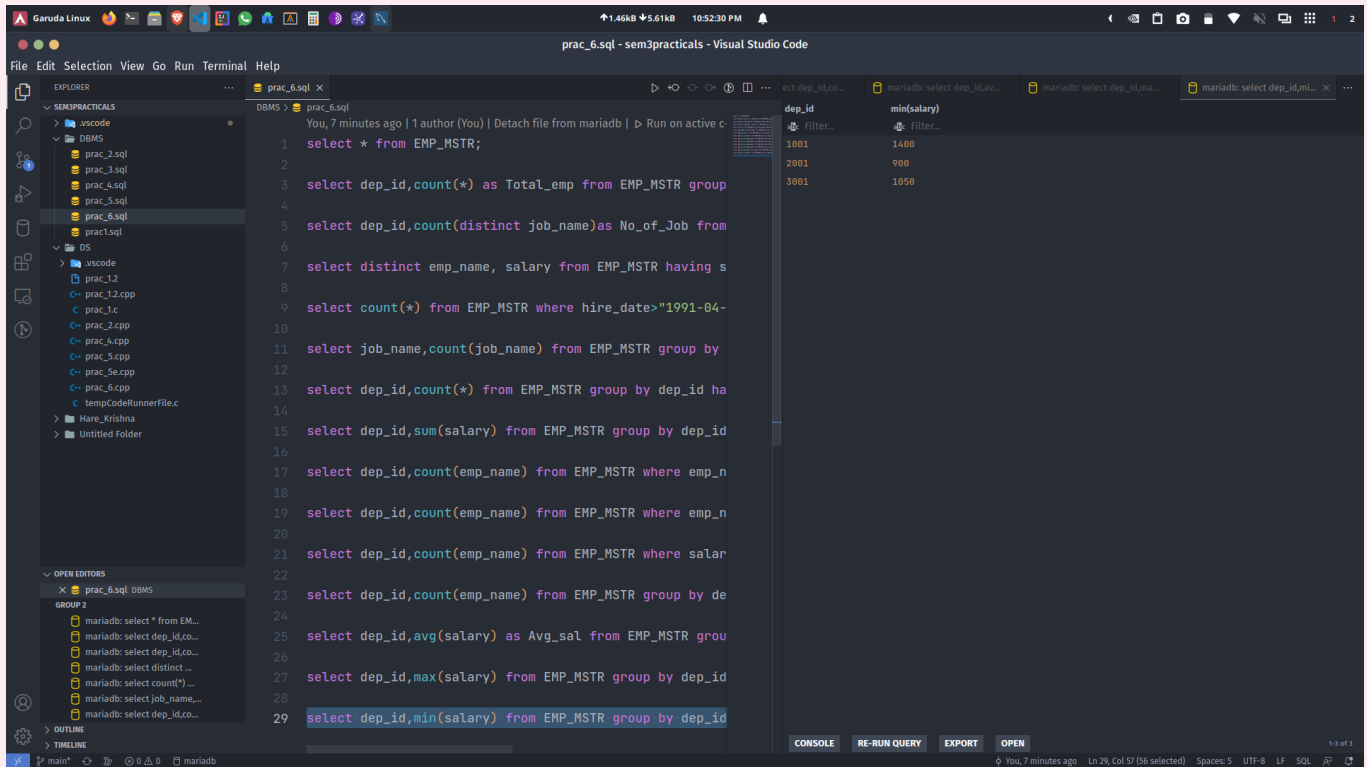
```
1 select * from EMP_MSTR;
2
3 select dep_id,count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id,count(distinct job_name)as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date>"1991-04-
10
11 select job_name,count(job_name) from EMP_MSTR group by
12
13 select dep_id,count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id,sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id,count(emp_name) from EMP_MSTR where emp_n
20
21 select dep_id,count(emp_name) from EMP_MSTR where salar
22
23 select dep_id,count(emp_name) from EMP_MSTR group by de
24
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id,max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

dep_id	max(salary)
1001	6000
2001	3100
3001	2750

14) Display name 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;
```

Name : Yash Lakhtariya  
Enrollement number : 21162101012  
Branch : CBA , Batch – 31  
DBMS Practical 6



The screenshot shows the Visual Studio Code interface with the DBMS extension. The Explorer sidebar on the left shows a project named 'SEM3PRACTICALS' with a folder 'DBMS' containing several SQL files. The 'prac\_6.sql' file is open in the editor, displaying 29 lines of SQL queries. The DBMS extension's Results pane on the right shows the output of the 29th query, which is a SELECT statement filtering for 'dep\_id' 1001, 2001, and 3001, and displaying 'min(salary)'. The results table shows three rows of data.

```
1 select * from EMP_MSTR;
2
3 select dep_id,count(*) as Total_emp from EMP_MSTR group
4
5 select dep_id,count(distinct job_name)as No_of_Job from
6
7 select distinct emp_name, salary from EMP_MSTR having s
8
9 select count(*) from EMP_MSTR where hire_date>"1991-04-
10
11 select job_name,count(job_name) from EMP_MSTR group by
12
13 select dep_id,count(*) from EMP_MSTR group by dep_id ha
14
15 select dep_id,sum(salary) from EMP_MSTR group by dep_id
16
17 select dep_id,count(emp_name) from EMP_MSTR where emp_n
18
19 select dep_id,count(emp_name) from EMP_MSTR where salar
20
21 select dep_id,count(emp_name) from EMP_MSTR where salar
22
23 select dep_id,count(emp_name) from EMP_MSTR group by de
24
25 select dep_id,avg(salary) as Avg_sal from EMP_MSTR grou
26
27 select dep_id,max(salary) from EMP_MSTR group by dep_id
28
29 select dep_id,min(salary) from EMP_MSTR group by dep_id
```

dep_id	min(salary)
1001	1400
2001	900
3001	1050