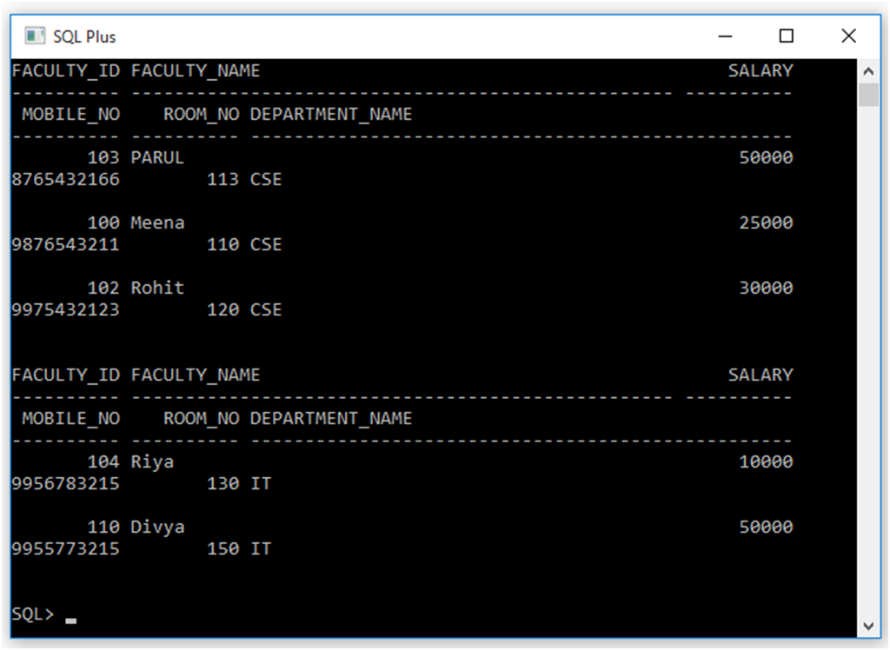
**To perform grouping, ordering and aggregation operations.**

Insert record



GROUPING

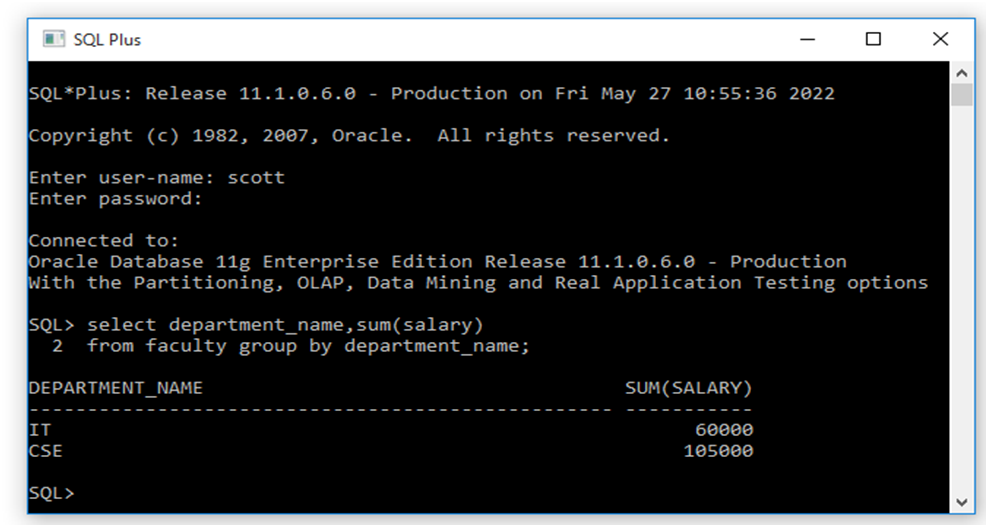
GROUP BY:- The GROUP BY statement groups rows that have the same values into summary rows, like "find the total salary of each department. The GROUP BY statement is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

Syntax:-

  SELECT column\_name FROM table\_name

  WHERE condition GROUP BY column\_name;

**[SQL\_23]** SQL>SELECT department\_name, sum(salary) FROM faculty GROUP BY department\_name;

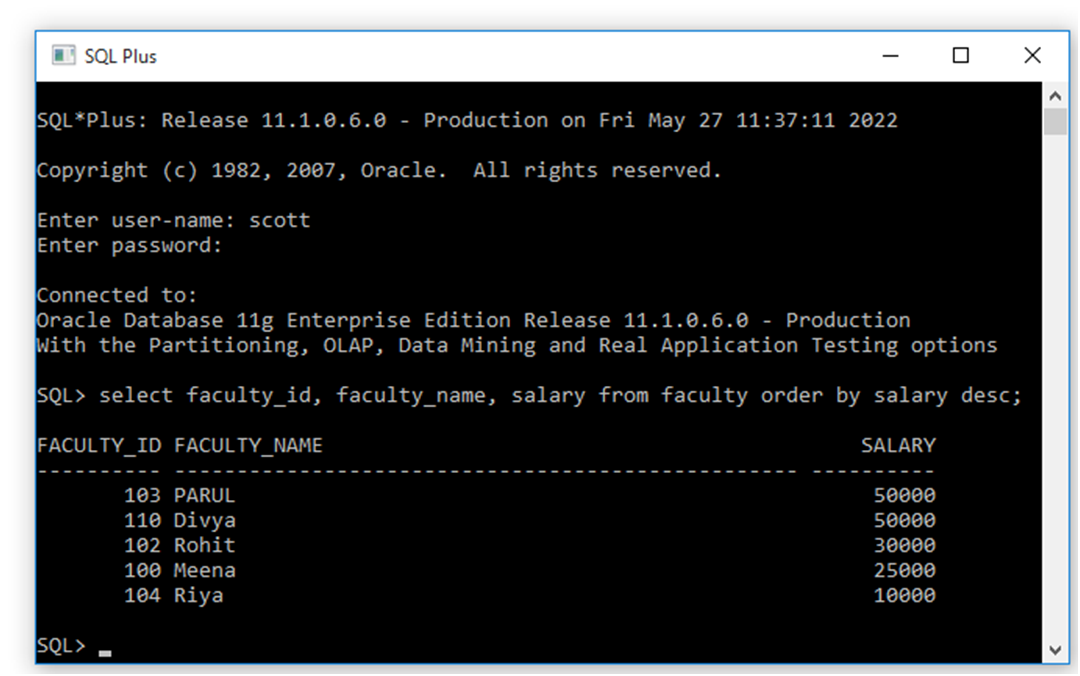


Q. Write a query with using where condition in GROUP BY.

ORDERING

ORDER BY: The ORDER BY command is used to sort the result set in ascending or descending order. The ORDER BY command sorts the result set in ascending order by default. To sort the records in descending order, use the DESC keyword.

**[SQL\_24]** SQL>SELECT faculty\_id, faculty\_name, salary FROM faculty ORDER BY salary DESC;



AGGREGATE FUNCTIONS

   Aggregate functions return a single result row based on groups of rows, rather than on single rows. Aggregate  functions can appear in select lists and in ORDER BY and HAVING clauses. They are commonly used with the GROUP BY clause in a SELECT statement, where Oracle Database divides the rows of a queried table or view into groups.

  Aggregate functions are AVG( ), COUNT( ), MIN( ), MAX( ), SUM( )

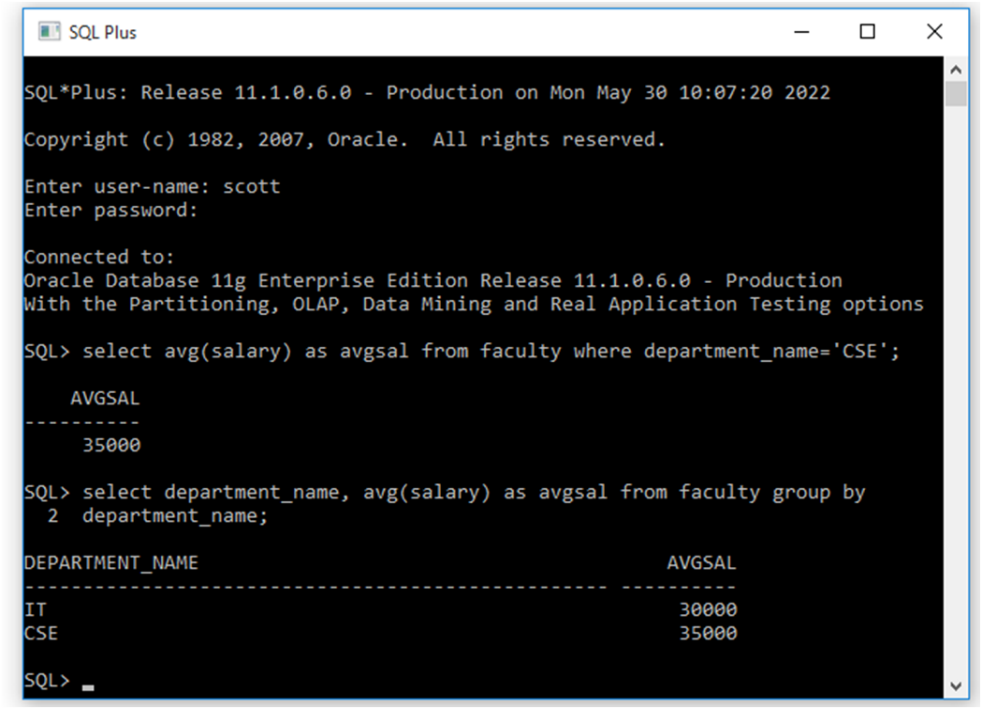
a) AVG():-AVG is an aggregate function that evaluates the average of an expression over a set of rows. AVG is allowed only on expressions that evaluate to numeric data types.

 Syntax:-

 AVG(Column\_name)

**[SQL\_25]** SQL>SELECT avg(salary) as avgsal FROM faculty WHERE department\_name=’CSE’;

**[SQL\_26]** SQL>SELECT department\_name, avg(salary) as avg(sal) FROM faculty GROUP BY department\_name;



**a)Count():-**This function returns the number of rows returned by the query. If you specify expr, then COUNT returns the number of rows where expr is not null. You can count either all rows, or only distinct values of expr. If you specify the asterisk (\*), then this function returns all rows, including duplicates and nulls. COUNT never returns null.

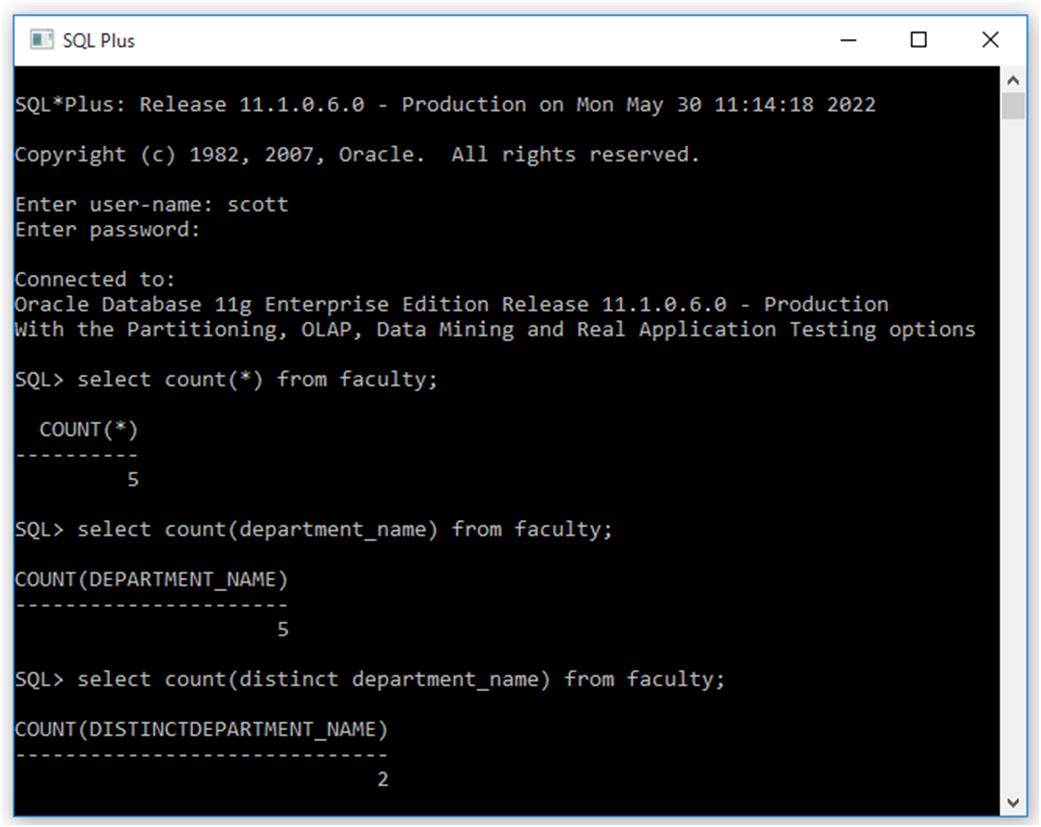
Syntax:-

COUNT({\*| DISTINCT| ALL|expr})

**[SQL\_27]** SQL>SELECT COUNT(\*) FROM faculty;

**[SQL\_28]** SQL>SELECT COUNT(department\_name) FROM faculty;

**[SQL\_29]** SQL>SELECT COUNT(distinct department\_name) FROM faculty;



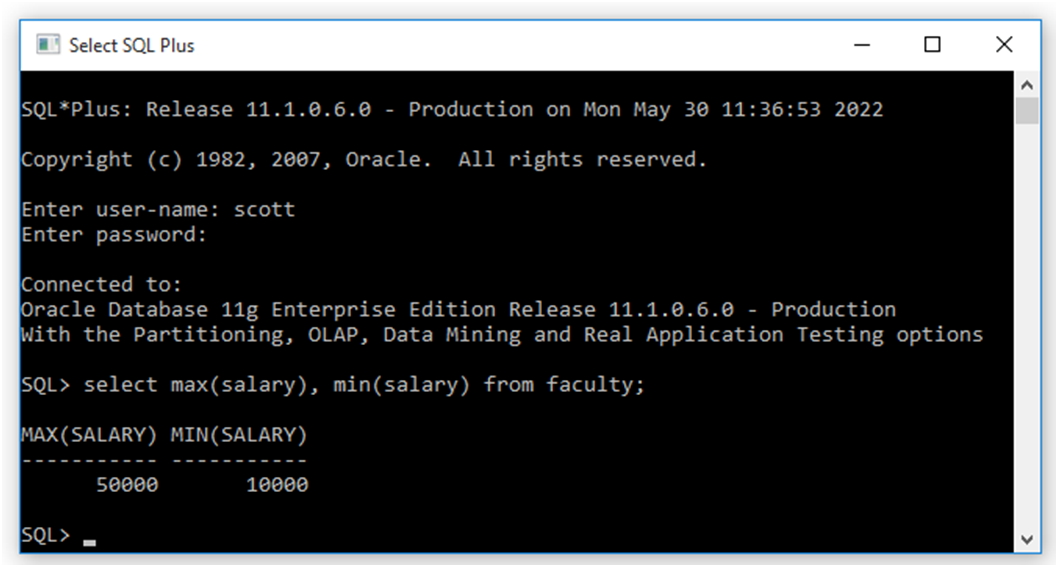
**b)MAX() & MIN():-** MAX is an aggregate function that evaluates the maximum of an expression over a set of rows. This function also ignores NULL values.

Syntax:- MAX(expression)

**MIN()**:-MIN is an aggregate function that evaluates the minimum of an expression over a set of rows. This function also ignores NULL values.

Syntax:- MIN(expression)

**[SQL\_30]** SQL>SELECT MAX(salary), MIN(salary)  FROM  faculty;



**c)SUM():-** This function is an aggregate function that returns the sum of all or distinct values in a set of values. It ignores NULL values.

Syntax:- SUM( [ALL | DISTINCT ] expression)

**[SQL\_31]** SQL>SELECT SUM(salary)  FROM  faculty;

**[SQL\_32]** SQL>SELECT department\_name, SUM(salary)  FROM  faculty GROUP BY department\_name;

**[SQL\_33]** SQL>SELECT SUM(distinct salary)  as salary FROM  faculty;

s