

RESTRICTING GROUP RESULTS WITH HAVING CLAUSE

As we use the WHERE clause to restrict the rows that we select, we can use the HAVING clause to restrict groups.

For example: To find the maximum salary of each department, but show only the departments that have a maximum salary of more than Rs.2900, we need to do the following.

- Find the maximum salary for each department by grouping by department number.
- Restrict the groups to those departments with a maximum salary greater than Rs.2900.

Syntax of HAVING Clause

```
SELECT column, group_function
FROM      table
[WHERE    condition]
[GROUP BY group_by_expression]
[HAVING   group_condition]
[ORDER BY Column];
```

Here we use the HAVING clause to specify which groups are to be displayed. Therefore, we further restrict the groups on the basis of aggregate information.

In the syntax:

HAVING clause restricts the groups of rows returned to those groups for which the specified condition is TRUE

The Oracle Server performs the following steps when you use the HAVING clause:

- Rows are grouped
- The group function is applied to the group.
- The groups that match the criteria in the HAVING clause are displayed.

The HAVING clause can precede the GROUP By clause, but it is recommended that you place the GROUP By clause first because it is more logical. Groups are formed and group functions are calculated before the HAVING clause is applied to the groups in the SELECT list.

For example: To find the maximum salary of each department, but show only the departments that have a maximum salary of more than Rs.2900

```
SQL> SELECT deptno,max(sal) FROM emp
      GROUP BY deptno Having max(sal)>2900;
```

OUTPUT:

DEPTNO	MAX(SAL)
10	5000
20	3000

Use of WHERE clause with GROUP BY clause

List the total salary, maximum and minimum salary and the average salary of employees job wise, for department number 20 and display only those rows having average salary greater than 1000

```
SQL>SELECT job, SUM(sal), avg(sal), max(sal), min(sal) from emp
      WHERE deptno=20
      GROUP by job
      HAVING AVG(sal)>1000;
```

OUTPUT:

JOB	SUM(SAL)	AVG(SAL)	MAX(SAL)	MIN(SAL)
ANALYST	6000	3000	3000	3000
MANAGER	2975	2975	2975	2975

Here, the WHERE clause will retrieve the rows of department 20. The GROUP BY clause will group them job wise and apply the aggregate functions.

```
SQL> SELECT job, SUM(sal) PAYROLL  
FROM emp  
WHERE job NOT LIKE 'SALE%'  
GROUP BY job  
HAVING SUM(sal)>5000  
ORDER BY SUM(sal);
```

OUTPUT:

JOB	PAYROLL
ANALYST	6000
MANAGER	8275

The above query displays the job title and total monthly salary for each job title with a total payroll exceeding Rs.5000. The example excludes salespeople and sorts the list by the total monthly salary.

Some more examples:

Let us consider the following database of applicants of a six week summer training course at Thapar University.

Application_Number	Name	Class	City	Gender	Institute
A1000	Ram	BE-2nd	Patiala	Male	Thapar
A1001	Sham	BE-2nd	Amritsar	Male	Thapar
A1002	Rehmat	BE-2nd	Jalandhar	Female	DAV Engg College
A1003	Ruhi	MCA	Jalandhar	Female	DAV Engg College
A1004	Ashish	BE-2nd	Patiala	Male	NIT Jalandhar
A1005	Rahat	BE-2nd	Patiala	Male	NIT Delhi
A1006	Misha	MCA	Patiala	Female	Thapar
A1007	Jyoti	BE-2nd	Amritsar	Female	Thapar

The faculty in charge may be interested in following queries. The corresponding SQL statements for solution of these queries have been presented below.

Query 1: Find number of applicants from each institute.

Solution: *SELECT institute, count(*) from STUDENT group by Institute;*

Query 2: Find number of students from each class.

Solution: *SELECT class, count(*) from STUDENT group by class;*

Query 3: Find number of applicant from each institute excluding Thapar.

Solution: *SELECT institute, count(*) from STUDENT where institute != 'Thapar' group by Institute;*

Query 4: Find number of applicant from each institute excluding Thapar having more than 5 students.

Solution: *SELECT institute, count(*) from STUDENT where institute != 'Thapar' group by Institute HAVING count(*)>5;*

Query 5: Find number of applicant from each institute excluding Thapar having more than 5 students and arrange the output in descending order of total number of students.

Solution: *SELECT institute, count(*) from STUDENT where institute != 'Thapar' group by Institute HAVING count(*)>5 ORDER BY count(*) desc;*

Query 6: Find the maximum number of students applied from a single institute.

Solution: *SELECT max(count(*)) from STUDENT group by institute;*

Query 7: Find the name of institute having maximum students;

Solution: *SELECT institute from STUDENT group by institute having count(*)=(SELECT max(count(*)) from student group by institute;*

Note: User can summarize that final syntax of the SELECT statement is given below:

```
SELECT column, group_function
FROM      table
[WHERE    condition]
[GROUP BY group_by_expression]
[HAVING   group_condition]
[ORDER BY Column];
```

Learner can conclude that shows that syntax of SELECT is not only limited to SELECT column_names FROM table which is mandatory part of the statement, while it is also includes WHERE CLAUSE, which is optional and used to select the rows based on some condition, GROUP BY clause is also optional and it is used to make grouping on a particular column or set of columns, HAVING clause is also optional and it is used to select the group based on some condition and ORDER BY clause is also optional and it is used to arrange the records in ascending or descending order.