

USING GROUP FUNCTIONS

Reporting Aggregate data using the Group functions

SQL has numerous predefined aggregate functions that can be used to write queries to produce exactly this kind of information. The GROUP BY clause specifies how to group rows from a data table when aggregating information, while the HAVING clause filters out rows that do not belong in specified groups.

Aggregate functions perform a variety of actions such as counting all the rows in a table, averaging a column's data, and summing numeric data. Aggregates can also search a table to find the highest "MAX" or lowest "MIN" values in a column. As with other types of queries, you can restrict, or filter out the rows these functions act on with the WHERE clause. For example, if a manager needs to know how many employees work in an organization, the aggregate function named COUNT * can be used to produce this information. The COUNT * function shown in the below SELECT statement counts all rows in a table.

```
SELECT COUNT(*)  
FROM employees;  
  
COUNT(*)  
-----  
24
```

The result table for the COUNT * function is a single column from a single row known as a scalar result or value. Notice that the result table has a column heading that corresponds to the name of the aggregate function specified in the SELECT clause.

Some of the commonly used aggregate functions are as below -

```
SUM( [ALL | DISTINCT] expression )  
AVG( [ALL | DISTINCT] expression )  
COUNT( [ALL | DISTINCT] expression )  
COUNT(*)  
MAX(expression)  
MIN(expression)
```

The ALL and DISTINCT keywords are optional, and perform as they do with the SELECT clauses that you have learned to write. The ALL keyword is the default where the option is allowed. The expression listed in the syntax can be a constant, a function, or any combination of column names, constants, and functions connected by arithmetic operators. However, aggregate functions are most often used with a column name. Except COUNT function, all the aggregate functions consider NULL values.

There are two rules that you must understand and follow when using aggregates:

- Aggregate functions can be used in both the SELECT and HAVING clauses
the HAVING clause is covered later in this chapter.
- Aggregate functions cannot be used in a WHERE clause. Its violation will produce the Oracle ORA-00934 group function is not allowed here error message.

Illustrations

The below SELECT query counts the number of employees in the organization.

```
SELECT COUNT(*) Count
```

```
FROM employees;
```

```
COUNT
```

```
24
```

The below SELECT query returns the average of the salaries of employees in the organization.

```
SELECT AVG(Salary) average_sal
FROM employees;
```

```
AVERAGE_SAL
```

```
15694
```

The below SELECT query returns the sum of the salaries of employees in the organization.

```
SELECT SUM(Salary) total_sal
FROM employees;
```

```
TOTAL_SAL
```

```
87472
```

The below SELECT query returns the oldest and latest hired dates of employees in the organization.

```
SELECT MIN (hire_date) oldest, MAX (hire_date) latest
FROM employees;
```

```
OLDEST LATEST
```

```
16-JAN-83 01-JUL-2012
```

GROUP BY

Aggregate functions are normally used in conjunction with a GROUP BY clause. The GROUP BY clause enables you to use aggregate functions to answer more complex managerial questions such as:

What is the average salary of employees in each department?

How many employees work in each department?

How many employees are working on a particular project?

Group by function establishes data groups based on columns and aggregates the information within a group only. The grouping criterion is defined by the columns specified in GROUP BY clause. Following this hierarchy, data is first organized in the groups and then WHERE clause restricts the rows in each group.

Guidelines of using GROUP BY clause

1 All the dependent columns or columns used in GROUP BY function must form the basis of grouping, hence must be included in GROUP BY clause also.

```
SELECT DEPARTMENT_ID, SUM(SALARY)
FROM employees;
```

```
DEPARTMENT_ID,
```

```
*
```

```
ERROR at line 2:
```

```
ORA-00937: not a single-group function
```

2 GROUP BY clause does not support the use of column alias, but the actual names.

3 GROUP BY clause can only be used with aggregate functions like SUM, AVG, COUNT, MAX, and MIN. If it is used with single row functions, Oracle throws an exception as "ORA-00979: not a GROUP BY expression".

4 Aggregate functions cannot be used in a GROUP BY clause. Oracle will return the "ORA-00934: group function not allowed" here error message.

Below query lists the count of employees working in each department.

```
SELECT DEPARTMENT_ID, COUNT (*)
FROM employees
GROUP BY DEPARTMENT_ID;
```

Similarly, below query to find sum of salaries for respective job ids in each department. Note the group is established based on Department and Job id. So they appear in GROUP BY clause.

```
SELECT DEPARTMENT_ID, JOB_ID, SUM (SAL)
FROM employees
GROUP BY DEPARTMENT_ID, JOB_ID;
```

The below query also produces the same result. Please note that grouping is based on the department id and job id columns but not used for display purpose.

```
SELECT SUM (SALARY)
FROM employees
GROUP BY DEPARTMENT_ID, JOB_ID;
```

Use of DISTINCT, ALL keywords with Aggregate functions

By specifying DISTINCT keyword with the input parameter, group by function considers only the unique value of the column for aggregation. By specifying ALL keyword with the input parameter, group by function considers all the values of the column for aggregation, including nulls and duplicates. ALL is the default specification.

The HAVING clause

The HAVING clause is used for aggregate functions in the same way that a WHERE clause is used for column names and expressions. Essentially, the HAVING and WHERE clauses do the same thing, that is filter rows from inclusion in a result table based on a condition. While it may appear that a HAVING clause filters out groups, it does not. Rather, a HAVING clause filters rows.

When all rows for a group are eliminated so is the group. To summarize, the important differences between the WHERE and HAVING clauses are:

A WHERE clause is used to filter rows BEFORE the GROUPING action
i. e. , before the calculation of the aggregate functions.

A HAVING clause filters rows AFTER the GROUPING action *i. e. , after the calculation of the aggregate functions.*

```
SELECT JOB_ID, SUM (SALARY)
FROM employees
GROUP BY JOB_ID
HAVING SUM (SALARY) > 10000;
```

The HAVING clause is a conditional option that is directly related to the GROUP BY clause option because a HAVING clause eliminates rows from a result table based on the result of a GROUP BY clause.

```
SELECT department_id, AVG(Salary)
FROM employees
HAVING AVG(Salary) > 33000;
ERROR at line 1: ORA-00937: not a single-group group function
Loading [MathJax]/jax/output/HTML-CSS/jax.js
```