

GROUPING BY MORE THAN ONE COLUMN

Sometimes there is a need to see results for groups within groups. For example if we have to find the total salary being paid to each job title, within each department. Then there is a need to having grouping on department number and within each department number grouping on the basis of job or in other words there is a need for grouping within a group.

Thus, the EMP table is grouped first by department number, and within that grouping, it is grouped by job title. For example, the two clerks in department 20 are grouped together and a single result (total salary) is produced for all clerks people within that group.

SQL> SELECT deptno,job,sum(sal) FROM emp GROUP BY deptno, job;

OUTPUT:

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600

9 rows selected.

The below figure shows the execution of query:

DEPT NO	JOB	SAL			DEPT NO	JOB	SUM(SAL)
10	CLERK	1300			10	CLERK	1300
10	MANAGER	2450			10	MANAGER	2450
10	PRESIDENT	5000			10	PRESIDENT	5000
20	ANALYST	3000			20	ANALYST	6000
20	ANALYST	3000			20	CLERK	1900
20	CLERK	800			20	MANAGER	2975
20	CLERK	1100			30	CLERK	950
20	MANAGER	2975			30	MANAGER	2850
30	CLERK	950			30	SALESMAN	1600
30	MANAGER	2850			30	SALESMAN	1250
30	SALESMAN	1600			30	SALESMAN	1500
30	SALESMAN	1250			30	SALESMAN	5600
30	SALESMAN	1500					
30	SALESMAN	1250					

By above example it is clear that we can return summary results for groups and subgroups by listing more than one GROUP BY column. We can determine the default sort order of the results by the order of the columns in the GROUP BY clause.

The SELECT clause specifies the column to be retrieved:

- Department number in the EMP table
- Job title in the EMP table
- The sum of all the salaries in the group that you specified in the GROUP BY clause
- The FROM clause specifies the tables that the database must access the EMP table
- The GROUP BY clause specifies how we must group the rows

First, department number groups the rows. Second, within the department number groups, the rows are grouped by job title. So, the SUM function is being applied to the salary column for all job titles within each department number group.

Illegal Queries Using Group Functions

Whenever you use a mixture of individual items (DEPTNO) and group functions (COUNT) in the same select statement, you must include a Group By clause that specifies the individual items (in this case, DEPTNO). If the GROUP By clause is missing, then the error message “not a single-group group function” appears and an asterisk (*) points to the offending column. You can correct the error by adding the GROUP BY clause.

For example, following is the illegal query:

```
SQL> SELECT deptno, COUNT(ename) FROM emp;
```

OUTPUT:

Column missing in the Group By clause

```
SELECT deptno,count(ename)
```

*

ERROR at line 1:

ORA-00937: not a single-group group function

In above select statement individual items DEPTNO and group function COUNT appears in the same SELECT statement without GROUP BY clause which results error, it can be corrected by adding the GROUP BY clause as shown below:

```
SQL> SELECT deptno,COUNT(ename) FROM emp Group By deptno;
```

OUTPUT:

DEPTNO	COUNT(ENAME)
10	3
20	5
30	6

Note: Any column or expression in the SELECT list that is not an aggregate function must be in the GROUP By clause.