Combining aggregate and non-aggregate values in SQL using Joins and Over clause

ishaan bhatnagar

Prerequisite – Aggregate functions in SQL, Joins in SQL

Aggregate functions perform a calculation on a set of values and return a single value. Now, consider an employee table EMP and a department table DEPT with following structure:

Table – EMPLOYEE TABLE

Name	Null	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7, 2)
COMM		NUMBER(7, 2)
DEPTNO		NUMBER(2)

Table – DEPARTMENT TABLE

Name	Null	Type
DEPTNO		NUMBER(2)

Name	Null	Type
DNAME		VARCHAR2(14)
LOC		VARCHAR2(13)

And the following results are needed:

1. DISPLAY NAME, SAL, JOB OF EMP ALONG WITH MAX, MIN, AVG, TOTAL SAL OF THE EMPS DOING THE SAME JOB.
2. DISPLAY DEPTNAME WITH NUMBER OF EMP WORKING IN IT.

The aggregated values can't be directly used with non-aggregated values to obtain a result. Thus one can use the following concepts:

1. Using Joins –

1. Create a sub-table containing the result of aggregated values.
2. Using Join, use the results from the sub-table to display them with non-aggregated values.

Solutions for problem 1 using JOIN:

```
SELECT ENAME, SAL, EMP.JOB,
            SUBTABLE.MAXSAL, SUBTABLE.MINSAL,
            SUBTABLE.AVGSAL, SUBTABLE.SUMSAL
FROM EMP
INNER JOIN
        (SELECT JOB, MAX(SAL) MAXSAL, MIN(SAL)
                MINSAL, AVG(SAL) AVGSAL, SUM(SAL) SUMSAL
         FROM EMP
          GROUP BY JOB) SUBTABLE
                  ON EMP.JOB = SUBTABLE.JOB;
```

Output for sample data:

Ename	Sal	Job	MaxSal	MinSal	AvgSal	SumSal
SCOTT	3300	ANALYST	3300	1925	2841.67	8525
HENRY	1925	ANALYST	3300	1925	2841.67	8525
FORD	3300	ANALYST	3300	1925	2841.67	8525
SMITH	3300	CLERK	3300	1045	1746.25	6985
MILLER	1430	CLERK	3300	1045	1746.25	6985

2. Using 'Over' clause –

1. OVER CLAUSE ALONG WITH PARTITION BY IS USED TO BRAKE UP DATA INTO PARTITIONS.
2. THE SPECIFIED FUNCTION OPERATES FOR EACH PARTITION.

Solutions for problem 2 using OVER Clause:

```
SELECT DISTINCT(DNAME),
COUNT(ENAME) OVER (PARTITION BY EMP.DEPTNO) EMP
FROM EMP
RIGHT OUTER JOIN DEPT
ON EMP.DEPTNO=DEPT.DEPTNO
ORDER BY EMP DESC;
```

Dname	Emp
SALES	6
RESEARCH	5
ACCOUNTING	3
OPERATIONS	0
OTHERS	0