CHAPTER 4

GROUP functions and Grouping

We have 5 GROUP functions,

- 1) Sum
- 2) **Max**
- 3) Min
- 4) Avg
- 5) Count

Sum – returns total value

Max - returns maximum value Min

- returns minimum value Avg returns average value Count -

returns number of records

Ex – 1) display the maximum salary, minimum salary and total salary from employee

SQL> select max(sal), min(sal), sum(sal) from emp;

To give aliases for the columns:-

SQL> select max(sal) "high",

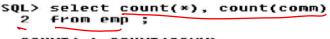
- 2 min(sal) "low", 3 sum(sal) "total"
- from emp ;

high	low	total
5000	800	29025

3) The below query gives the total number of employees

2 from emp;

4) The below query gives the <u>number of employees</u> who have commission



COUNT(*) COUNT(COMM)

null won't be counted

5) List the number of employees in department 30

```
SQL> select count(*) from emp

2 where deptno = 30;

COUNT(*)

------6
```

ASSIGMENT

1) Display the total salary in department 30

```
SQL> select sum(sal) "total" from emp
2 where deptno = 30;
     total
-----
9400
```

2) List the number of clerks in department 20

```
SQL> select count(*) from emp
2  where deptno = 20
3  and job = 'CLERK';

COUNT(*)
------
```

3) List the highest and lowest salary earned by salesmen

GROUPING

It is the process of computing the aggregates by segregating based on one or more columns. Grouping is done by using "group by" clause.

For ex – 1) <u>Display the total salary of all departments</u>

```
SQL> select deptno sum(sal)
2 from emp
3 group by deptno;

DEPTNO SUM(SAL)

30 9400
20 10875
10 8750
```

2) Display the maximum of each job

HAVING

"Having" is used to filter the grouped data. "Where" is used to filter the non grouped data.

"Having" should be used after group by clause "Where" should be used before group by clause

For ex – 1) Display job-wise highest salary only if the highest salary is more than Rs1500

SQL> sele	ct job, max(sal)	
2 from	from emp	
3 grou	p by job	
4 havi	ng max(sal) > 1500 ;	
JOB	MAX(SAL)	
SALESMAN	1600	
PRESIDENT	5000	
MANAGER	2975	
ANALYST	3000	

2) <u>Display job-wise highest salary only if the highest salary is more than 1500 excluding department</u> 30. Sort the data based on highest salary in the ascending order.

SQL> select job, max(sal)
2 from emp
3 where deptno (>30)
4 group by job
5 having max(sal) >1500

17 mit equal to un influenci offry

JOB MAX(SAL)

MANAGER 2975
ANALYST 3000
PRESIDENT 5000

order by 2;

RESTRICTIONS ON GROUPING

- we can select only the columns that are part of "group by" statement If we try selecting other columns, we will get an error as shown below,

SQL> select deptno, job sum(sal), sum(comm)
2 from emp
3 group by deptno;
select deptno, job, sum(sal), sum(comm)

ERROR at line 1: ORA-00979: not a GROUP BY expression

The above query is an error because "job" is there in the select query but not in the group by query. If it is enclosed in any of the group functions like sum(sal) etc – then it is not an error. But whatever table is included in the select query must also be included in the group by query.

The above problem can be overcome with the following query as shown below,

SQL> select deptno, job, sum(sal), sum(comm)

2 from emp 3 group by deptno, job muchy g

DEPTHO	JOB	SUM(SAL)	SUM(COMM)
20	CLERK	1900	
30	SALESMAN	5600	2200
20	MANAGER	2975	
30	CLERK	950	
10	PRESIDENT	5000	
30	MANAGER	2850	
10	CLERK	1300	
10	MANAGER	2450	
20	ANAL YST	6000	

9 rows selected.

The below query is also correct to rectify the above error,

- select deptno, sum(sal), sum(comm)
- from emp
- group by deptno, job
- 4* order by deptno SQL> /

Sort corruly

DEPTH0	SUM(SAL)	SUM(COMM)
10	1300	
10	2450	
19	5000	
20	6000	
20	1900	
20	2975	
30	950	
30	2850	
30	5600	2200

9 rows selected.

Whatever is there in the select statement must be there in the group by statement. But, whatever is there in the group by statement need not be present in the select statement. This is shown in the above two corrected queries.

ASSIGNMENT

1) Display the department numbers along with the number of employees in it

SQL> select deptno, count(*) 2 from emp

- 3 group by deptno
- 4 order by deptno;

DEPTHO	COUNT(*)
10	3
20	5
30	6

2) Display the department numbers which are having more than 4 employees in them

SQL> select deptno from emp

- 2 group by deptno 3 having count(*) >4
- 4 order by deptno;

	DEPTNO
-	
	20
	30

3) <u>Display the maximum salary for each of the job excluding all the employees whose name ends with</u> **"S**"

	Max Max
SQL>	select ename, job, min(sal)
2	from emp
3	where ename not like '%S'
4	group by ename, job
5	order by 3

1	0 0	Lhas-
enanez	=6,000	P

JOB	MIN(SAL)
CLERK	800
SALESMAN	1250
SALESMAN	1250
CLERK	1300
SALESMAN	1500
SALESMAN	1600
MANAGER	2450
MANAGER	2850
ANALYST	3000
ANALYST	3000
PRESIDENT	5000
	CLERK SALESMAN SALESMAN CLERK SALESMAN SALESMAN MANAGER MANAGER ANALYST ANALYST

11 rows selected.

4) <u>Display the department numbers which are having more than 9000 as their departmental total salary</u>

ador for much

NOTE:

30

To clear the screen, the command used is, cl scr;

9400

if it is a large query and we cannot type it repeatedly, then type in - SQL > ed; when we type ed; - we get the notepad – after making the necessary changes – then click on the "**x**" i.e, the close button at the top right corner – then click on yes when a dialog box asking whether to overwrite the file comes – after this it comes to the oracle screen – in the next line, enter "/ " and hit on enter button – another way of ending the query is by typing " / " in the next line of the query – this indicates the end of the query.