4

Aggregating Data Using Group Functions

Objectives

At the end of this lesson, you should be able to:

- Identify the available group functions
- Describe the use of group functions
- Group data using the GROUP BY clause
- Include or exclude grouped rows by using the HAVING clause

What Are Group Functions?

Group functions operate on sets of rows to give one result per group.

EMP

```
    DEPTNO        SAL
---------- ----------
        10       2450
        10       5000
        10       1300
        20        800
        20       1100
        20       3000
        20       3000
        20       2975
        30       1600
        30       2850
        30       1250
        30        950
        30       1500
        30       1250
```

“maximum salary in the EMP table”

```
  MAX(SAL)
----------
      5000
```

ORACLE®

Types of Group Functions

- **AVG ([DISTINCT|ALL]*n*)**
- **COUNT ({ *|[DISTINCT|ALL]*expr*})**
- **MAX ([DISTINCT|ALL]*expr*)**
- **MIN ([DISTINCT|ALL]*expr*)**
- **STDDEV ([DISTINCT|ALL]*x*)**
- **SUM ([DISTINCT|ALL]*n*)**
- **VARIANCE ([DISTINCT|ALL]*x*)**

Using AVG and SUM Functions

You can use AVG and SUM for numeric data.

```
SQL> SELECT   AVG(sal), MAX(sal),
  2           MIN(sal), SUM(sal)
  3  FROM     emp
  4  WHERE    job LIKE 'SALES%';
```

```
AVG(SAL)  MAX(SAL)  MIN(SAL)  SUM(SAL)
--------  --------- --------- ---------
    1400      1600      1250      5600
```

Using MIN and MAX Functions

You can use MIN and MAX for any datatype.

```
SQL> SELECT   MIN(hiredate), MAX(hiredate)
  2  FROM     emp;
```

```
MIN(HIRED MAX(HIRED
--------- ---------
17-DEC-80 12-JAN-83
```

Using the COUNT Function

COUNT(*) returns the number of rows in a table.

```
SQL> SELECT    COUNT(*)
  2  FROM      emp
  3  WHERE     deptno = 30;
```

```
 COUNT(*)
---------
        6
```

Using the COUNT Function

COUNT(*expr*) returns the number of nonnull rows.

```
SQL> SELECT    COUNT(comm)
  2  FROM      emp
  3  WHERE     deptno = 30;
```

```
COUNT(COMM)
-----------
          4
```

Group Functions and Null Values

Group functions ignore null values in the column.

```
SQL> SELECT AVG(comm)
  2  FROM   emp;
```

```
 AVG(COMM)
---------
      550
```

Using the NVL Function with Group Functions

The NVL function forces group functions to include null values.

```
SQL> SELECT AVG(NVL(comm,0))
  2  FROM   emp;
```

```
AVG(NVL(COMM,0))
----------------
       157.14286
```

Creating Groups of Data

EMP

```
    DEPTNO        SAL
---------- ----------
        10       2450
        10       5000
        10       1300
        20        800
        20       1100
        20       3000
        20       3000
        20       2975
        30       1600
        30       2850
        30       1250
        30        950
        30       1500
        30       1250
```

2916.6667

2175

1566.6667

"average salary in EMP table for each department"

```
 DEPTNO   AVG(SAL)
------- ----------
     10  2916.6667
     20       2175
     30  1566.6667
```

ORACLE®

Creating Groups of Data: GROUP BY Clause

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

Divide rows in a table into smaller groups by using the GROUP BY clause.

ORACLE®

Using the GROUP BY Clause

All columns in the SELECT list that are not in group functions must be in the GROUP BY clause.

```
SQL> SELECT    deptno, AVG(sal)
  2  FROM      emp
  3  GROUP BY deptno;
```

```
   DEPTNO  AVG(SAL)
--------- ---------
       10 2916.6667
       20      2175
       30 1566.6667
```

Using the GROUP BY Clause

The GROUP BY column does not have to be in the SELECT list.

```
SQL> SELECT    AVG(sal)
  2  FROM      emp
  3  GROUP BY deptno;
```

```
 AVG(SAL)
---------
2916.6667
     2175
1566.6667
```

ORACLE®

Grouping by More Than One Column

EMP

```
   DEPTNO JOB              SAL
--------- --------- ---------
       10 MANAGER        2450
       10 PRESIDENT      5000
       10 CLERK          1300
       20 CLERK           800
       20 CLERK          1100
       20 ANALYST        3000
       20 ANALYST        3000
       20 MANAGER        2975
       30 SALESMAN       1600
       30 MANAGER        2850
       30 SALESMAN       1250
       30 CLERK           950
       30 SALESMAN       1500
       30 SALESMAN       1250
```

"sum salaries in the EMP table for each job, grouped by department"

```
  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
```

ORACLE®

Using the GROUP BY Clause on Multiple Columns

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

```
   DEPTNO JOB        SUM(SAL)
--------- ---------- ---------
       10 CLERK           1300
       10 MANAGER         2450
       10 PRESIDENT       5000
       20 ANALYST         6000
       20 CLERK           1900
...
9 rows selected.
```

ORACLE®

Illegal Queries Using Group Functions

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

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

```
SELECT deptno, COUNT(ename)
       *
ERROR at line 1:
ORA-00937: not a single-group group function
```

Column missing in the GROUP BY clause

ORACLE®

Illegal Queries Using Group Functions

- You cannot use the WHERE clause to restrict groups.
- You use the HAVING clause to restrict groups.

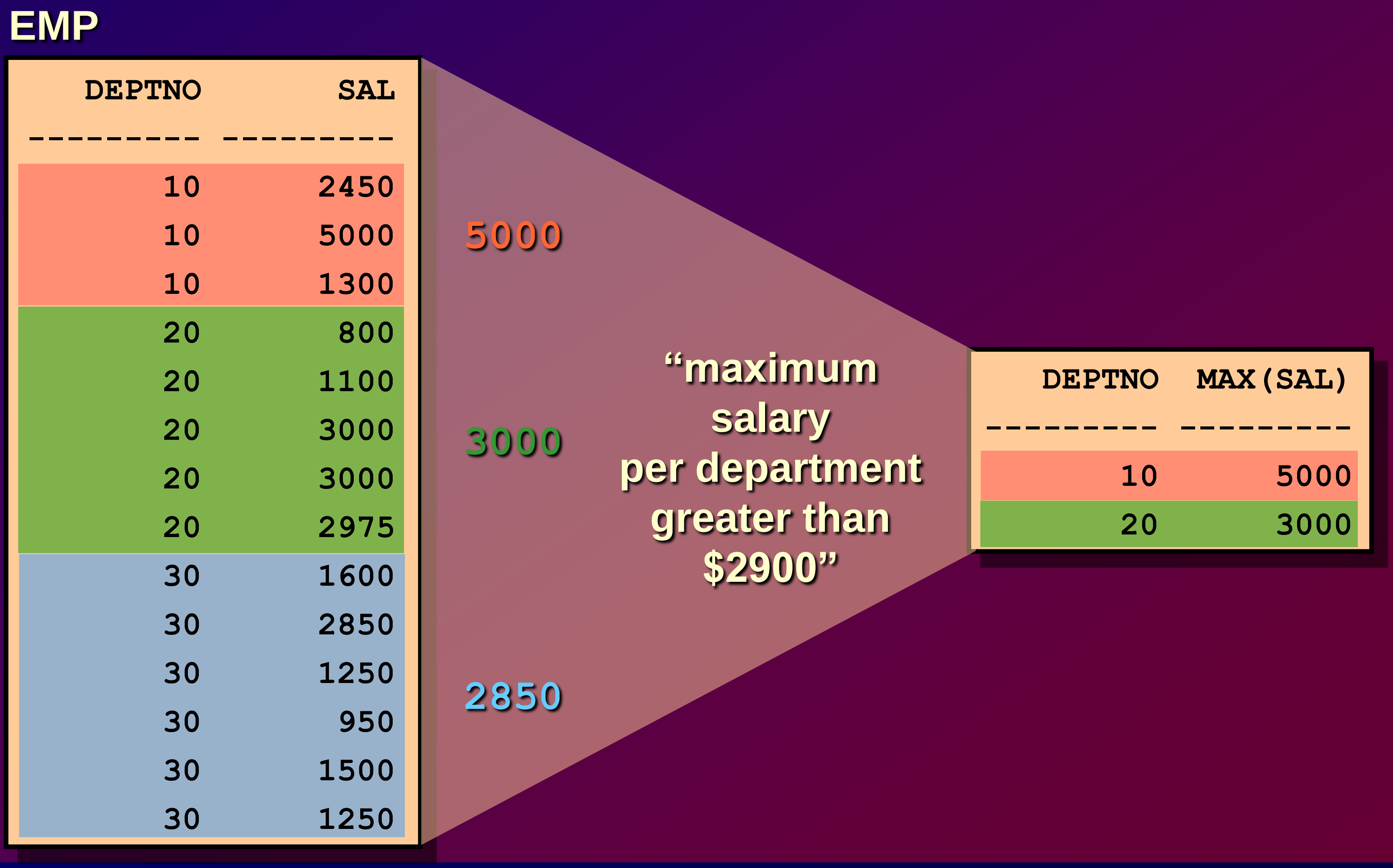
```
SQL> SELECT    deptno, AVG(sal)
  2  FROM      emp
  3  WHERE     AVG(sal) > 2000
  4  GROUP BY  deptno;
```

```
WHERE AVG(sal) > 2000
      *
ERROR at line 3:
ORA-00934: group function is not allowed here
```

Cannot use the WHERE clause to restrict groups

ORACLE®

Excluding Group Results



EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

"maximum salary per department greater than $2900"

DEPTNO	MAX(SAL)
10	5000
20	3000

ORACLE®

Excluding Group Results: HAVING Clause

Use the HAVING clause to restrict groups

- Rows are grouped.
- The group function is applied.
- Groups matching the HAVING clause are displayed.

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

Using the HAVING Clause

```
SQL> SELECT    deptno, max(sal)
  2  FROM      emp
  3  GROUP BY deptno
  4  HAVING    max(sal)>2900;
```

```
   DEPTNO  MAX(SAL)
--------- ---------
       10      5000
       20      3000
```

Using the HAVING Clause

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

```
JOB          PAYROLL
--------- ---------
ANALYST         6000
MANAGER         8275
```

ORACLE®

Nesting Group Functions

Display the maximum average salary.

```
SQL> SELECT    max(avg(sal))
  2  FROM      emp
  3  GROUP BY deptno;
```

```
MAX(AVG(SAL))
-------------
    2916.6667
```

Summary

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

Practice Overview

- Showing different queries that use group functions
- Grouping by rows to achieve more than one result
- Excluding groups by using the HAVING clause