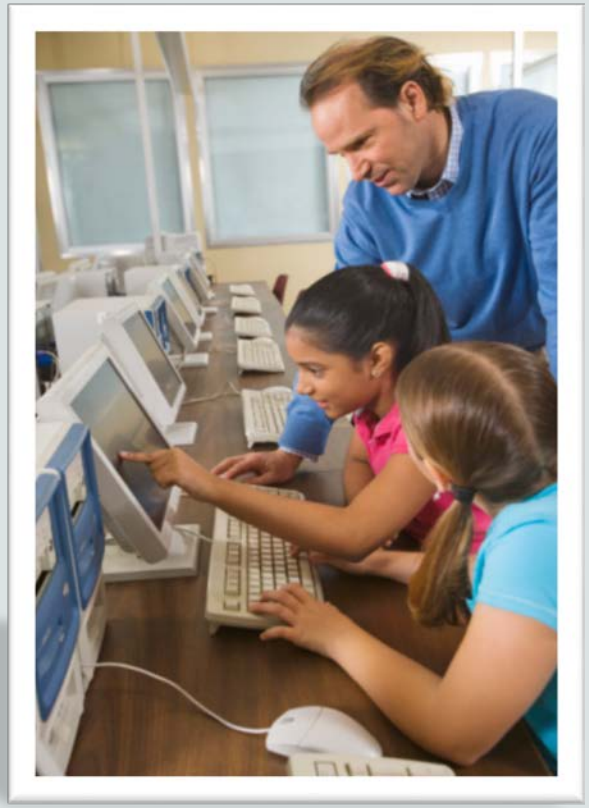

ORACLE
ACADEMY

Database Programming with SQL

8-2
COUNT, DISTINCT, NVL



ORACLE ACADEMY

Objectives

This lesson covers the following objectives:

- Construct and execute a SQL query using the COUNT group function
- Use DISTINCT and the NVL function with group functions

Purpose

- Being able to aggregate (group together) data using SQL functions enables businesses to do calculations that would otherwise have to be done by hand.
- Remember the example of having to count all of the students in your school? A daunting task!
- There just aren't enough hands to accomplish it manually.
- Fortunately, the SQL group functions can easily process these types of requests.

COUNT

- COUNT(expression) returns the number of non-null values in the expression column.

```
SELECT COUNT(job_id)
FROM employees;
```

COUNT(JOB_ID)
20

COUNT and NULL Values

- Twenty rows of employees are listed in the employees table, and if you select commission_pct, twenty rows are returned.
- Adding a count function to the query COUNT returned only four.
- COUNT specifically counts the commission_pct column but ignores the null values in the column.

```
SELECT commission_pct
FROM employees;
```

20 rows returned in 0.01 seconds

```
SELECT COUNT(commission_pct)
FROM employees;
```

COUNT(COMMISSION_PCT)
4

COUNT All Rows

- COUNT(*) returns the number of rows in a table.
- It does not specify a column (which may or may not contain nulls) to count; it counts the number of rows returned in the result set.
- For example, to find out how many employees were hired before 01/Jan/1996, COUNT can be used in the SELECT statement.

```
SELECT COUNT(*)
FROM employees
WHERE hire_date < '01-Jan-1996';
```

COUNT (*)
9

COUNT All Rows

- We use COUNT(*) when we want to make sure that we count all the rows (including dupicates), as well as those that may have nulls in one or more columns.

```
SELECT COUNT(*)
FROM employees
WHERE hire_date < '01-Jan-1996';
```

COUNT (*)
9

DISTINCT

- The keyword DISTINCT is used to return only non-duplicate values or combinations of non-duplicate values in a query.
- Examine the query below.
- Without using the keyword DISTINCT, the query returned all of the job_id values from the employees table, including the duplicate values.

```
SELECT job_id
FROM employees;
```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
AD_VP
IT_PROG
…

20 rows returned in 0.01 seconds

DISTINCT Example

- To eliminate duplicate rows, use the DISTINCT keyword as shown here.
- Using the DISTINCT keyword returned all of the job IDs exactly once, with no duplicate values.

```
SELECT DISTINCT job_id
FROM employees;
```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
IT_PROG
MK_MAN
…

12 rows returned in 0.01 seconds

DISTINCT Non-duplicate

- The keyword DISTINCT, when used in a query selecting more than one column, will return non-duplicate combinations of the selected columns.
- Examine the result set shown here.
- Notice that no duplicates exist of the combination of job_id and department_id even though duplicates exist in both columns.

```
SELECT DISTINCT job_id,
  department_id
FROM employees;
```

JOB_ID	DEPARTMENT_ID
IT_PROG	60
SA_REP	80
ST_MAN	50
AD_VP	90
AD_ASST	10
MK_MAN	20
MK_REP	20
SA_MAN	80
SA_REP	-
…	…

13 rows returned in 0.01 seconds

Using DISTINCT

- The keyword DISTINCT can be used with all group functions.
- Using DISTINCT makes the function consider only non-duplicate values.
- The two statements on the right produce different results because the second only considers one occurrence of 17000

```
SELECT SUM(salary)
FROM employees
WHERE department_id = 90;
```

SALARY
24000
17000
17000
....

SUM(SALARY)
58000

```
SELECT SUM(DISTINCT salary)
FROM employees
WHERE department_id = 90;
```

SALARY
24000
17000
17000
....

SUM(DISTINCT SALARY)
41000

DISTINCT and COUNT

- When using DISTINCT with a group function such as COUNT, the result set will return the number of non-duplicate column values.

```
SELECT COUNT (DISTINCT job_id)
FROM employees;
```

COUNT (DISTINCT job_id)
12

How many different jobs are assigned to employees?

```
SELECT COUNT (DISTINCT salary)
FROM employees;
```

COUNT (DISTINCT salary)
18

How many different salary amounts are paid to employees?

NVL

- Sometimes it is desirable to include null values in group functions.
- For example, knowing the average number of customer orders served each day could be used to judge how much food to order each month.
- Some days the restaurant is closed and no customers are served, but the owner has found that computing the average by including the days he is closed is a better indicator than just counting the days with customers.

NVL

- The SELECT statement to include null values could be written starting with:

```
SELECT AVG(NVL(customer_orders, 0))
```

- Another example on employees table:

```
SELECT AVG(commission_pct)
FROM employees;
```

AVG(COMMISSION_PCT)
.2125

```
SELECT AVG(NVL(commission_pct, 0))
FROM employees;
```

AVG(NVL(COMMISSION_PCT,0))
.0425

NVL

- Compare the results of the following two queries.

```
SELECT AVG(commission_pct)
FROM employees;
```

AVG(COMMISSION_PCT)
.2125

```
SELECT AVG(NVL(commission_pct, 0))
FROM employees;
```

AVG(NVL(COMMISSION_PCT,0))
.0425

Terminology

Key terms used in this lesson included:

- Aggregate
- COUNT (expression)
- COUNT (DISTINCT expression)
- DISTINCT

Summary

In this lesson, you should have learned how to:

- Construct and execute a SQL query using the COUNT group function
- Use DISTINCT and the NVL function with group functions


ORACLE
ACADEMY