

Database Programming

COUNT, DISTINCT, NVL



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 (YEAR)
FROM d_cds
WHERE year < 2001;
```

COUNT (YEAR)
5

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

```
SELECT COUNT
(DISTINCT year)
FROM d_cds
WHERE year < 2001;
```

COUNT (DISTINCT YEAR)
4

COUNT and NULL Values

Six rows of comments are listed in the table, including the empty (null) comment, but the count function returned only five. Why?

Because COUNT specifically counts the comments column but ignores the null value in the column.

```
SELECT comments  
FROM d_play_list_items;
```

COMMENTS
Play late
(null)
Play early
Play after cake cutting
Play first
Play for the father

```
SELECT COUNT(comments)  
FROM d_play_list_items;
```

COUNT (COMMENTS)
5

COUNT All Rows

COUNT(*) returns the number of rows in a table that satisfy the criteria of the SELECT statement. It does not specify a specific 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 of DJs on Demand CDs were produced before 2001, COUNT can be used in the SELECT statement.

```
SELECT COUNT ( * )  
FROM d_cds  
WHERE year < 2001;
```

COUNT (*)
5

COUNT All Rows (cont.)

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

```
SELECT COUNT (*)  
FROM d_cds  
WHERE year < 2001;
```

COUNT (*)
5

DISTINCT

The keyword **DISTINCT** is used to return only non-duplicate values or combinations of non-duplicate values in a query.

Examine the query on the right. Without using the keyword **DISTINCT**, the query returned all of the year values from the DJs on Demand D_CDS table, including the duplicate values.

```
SELECT year as 'CD Year'  
FROM d_cds;
```

CD Year
1997
2000
2002
1999
2000
2001
1998
2004

DISTINCT Example

To eliminate duplicate rows, use the **DISTINCT** keyword as shown here.

Using the **DISTINCT** keyword returned all of the CD years exactly once, with no duplicate values.

```
SELECT DISTINCT year AS 'CD  
Year'  
FROM d_cds;
```

CD Years
1997
1998
1999
2000
2001
2002
2004

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 the year and the title even though duplicates of the year do exist.

YEAR	TITLE
1997	The Celebrants Live in Concert
1998	Graduation Songbook
1999	Songs from My Childhood
2000	Cape Diem
2000	Party Music for All Occasions
2001	Here Comes the Bride
2002	Back to the Shire
2004	Whirled Peas

```
SELECT DISTINCT year, title  
FROM d_cds;
```

Using DISTINCT

The keyword **DISTINCT** can be used with all group functions. Using **DISTINCT** makes the function consider only non-duplicate values.

Why do the two statements on the right produce different results?

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

SALARY	SUM(SALARY)
24000	58000
17000	
17000	

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

SALARY	SUM(DISTINCT SALARY)
24000	41000
17000	
17000	

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.

How many different jobs are assigned to employees? How many different salary amounts are paid to employees?

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

COUNT (DISTINCT job_id)
12

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

COUNT (DISTINCT salary)
18

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 (cont.)

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

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

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