



# ORACLE

## Academy



# Database Programming with SQL

8-2

COUNT, DISTINCT, NVL

**ORACLE**  
Academy



# Objetivos

- En esta lección se abordan los siguientes objetivos:
  - Crear y ejecutar una consulta SQL utilizando la función de grupo COUNT
  - Utilizar DISTINCT y la función NVL con funciones de grupo



# Objetivo

- Ser capaz de agregar (agrupar) los datos mediante las funciones SQL permite a las empresas realizar cálculos que, de lo contrario, se tendrían que ejecutar manualmente
- ¿Recuerda el ejemplo en el que tenía que contar a todos los alumnos de su centro educativo? Una tarea descomunal
- No hay suficientes manos para realizarla manualmente
- Afortunadamente, las funciones de grupo SQL pueden procesar fácilmente estos tipos de solicitudes

# COUNT

- COUNT (expresión) devuelve el número de valores no nulos de la columna de expresión

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

COUNT(JOB_ID)
20

# Valores COUNT y NULL

- En la tabla employees se muestran veinte filas de empleados y, si selecciona commission\_pct, se devuelven veinte filas
- Al agregar una función count a la consulta, COUNT solo ha devuelto cuatro
- COUNT cuenta específicamente la columna commission\_pct, pero
- ignora los valores nulos en la columna

```
SELECT commission_pct  
FROM employees;
```

20 filas devueltas en 0,01 segundos

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

COUNT(COMMISSION_PCT)
4

# COUNT All Rows

- COUNT(\*) devuelve el número de filas de una tabla
- No especifica la columna (que puede o no incluir nulos) que contar; cuenta el número de filas devueltas en el juego de resultados
- Por ejemplo, para averiguar cuántos empleados fueron contratados antes del 01/Ene/1996, se puede utilizar COUNT en la sentencia SELECT

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

COUNT (*)
9



# COUNT All Rows

- Utilizamos COUNT(\*) cuando queremos asegurarnos de contar todas las filas (incluidos los duplicados), así como aquellos que pueden tener valores nulos en una o más columnas

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

COUNT (*)
9



# DISTINCT

- La palabra clave DISTINCT se utiliza para devolver solo valores no duplicados o combinaciones de valores no duplicados en una consulta
- Examine la consulta siguiente
- Sin utilizar la palabra clave DISTINCT, la consulta ha devuelto todos los valores job\_id de la tabla employees, incluidos los valores duplicados

```
SELECT job_id  
FROM employees;
```

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
AD_VP
IT_PROG
...

20 filas devueltas en 0,01 segundos

# Ejemplo de DISTINCT

- Para eliminar las filas duplicadas, utilice la palabra clave DISTINCT, tal y como se muestra aquí
- Utilizando la palabra clave DISTINCT se han devuelto todos los ID de trabajo exactamente una vez, sin valores duplicados

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

JOB_ID
AC_ACCOUNT
AC_MGR
AD_ASST
AD PRES
AD VP
IT_PROG
MK_MAN
...

12 filas devueltas en 0,01 segundos

# Valores DISTINCT no duplicados

- La palabra clave DISTINCT, cuando se utiliza en una consulta en que se seleccione más de una columna, devolverá combinaciones no duplicadas de las columnas seleccionadas
- Examine el juego de resultados que se muestra aquí

```
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 filas devueltas en 0,01 segundos

# Valores DISTINCT no duplicados

- Tenga en cuenta que no existen duplicados de la combinación de job\_id y department\_id, incluso aunque existan duplicados en ambas columnas

```
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 filas devueltas en 0,01 segundos

# Uso de DISTINCT

- La palabra clave DISTINCT se puede utilizar con todas las funciones de grupo
- Con DISTINCT, la función solo tiene en cuenta los valores no duplicados
- Con las dos sentencias de la derecha se producen resultados diferentes, ya que la segunda solo tiene en cuenta una incidencia de 17000

```
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 y COUNT

- Al utilizar DISTINCT con una función de grupo como COUNT, el juego de resultados devolverá el número de valores de columna no duplicados

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

COUNT (DISTINCT job_id)
12

¿Cuántos trabajos diferentes tienen asignados los empleados?

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

COUNT (DISTINCT salary)
18

¿Cuántos importes de salarios diferentes se pagan a los empleados?

# NVL

- A veces es preferible incluir valores nulos en funciones de grupo
- Por ejemplo, saber el número medio de pedidos de clientes servidos cada día se podría utilizar para determinar cuánta comida pedir cada mes
- Algunos días el restaurante está cerrado y no se sirve a los clientes, pero el propietario ha descubierto que calcular la media mediante la inclusión de los días que está cerrado es un indicador más fiable que solamente contar los días con clientes



# NVL

- La sentencia SELECT para incluir los valores nulos se podría escribir empezando por:

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

- Otro ejemplo de la tabla employees:

```
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 los resultados de las dos consultas siguientes

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

AVG(COMMISSION_PCT)
.2125

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

AVG(NVL(COMMISSION_PCT,0))
.0425

# Terminología

- Entre los términos clave utilizados en esta lección se incluyen:
  - Aggregate
  - COUNT (expression)
  - COUNT (DISTINCT expression)
  - DISTINCT

# Resumen

- En esta lección, debe haber aprendido lo siguiente:
  - Crear y ejecutar una consulta SQL utilizando la función de grupo COUNT
  - Utilizar DISTINCT y la función NVL con funciones de grupo





# ORACLE

## Academy

