



# **Informes de Datos Agregados con Funciones de Grupo**

# Objetivos

Al finalizar esta lección, debería estar capacitado para lo siguiente:

- Identificar las funciones de grupo disponibles
- Describir los usos de las funciones de grupo
- Agrupar datos con la cláusula `GROUP BY`
- Incluir o excluir filas agrupadas con la cláusula `HAVING`

# Agenda

- Funciones de grupo:
  - Tipos y sintaxis
  - Uso de AVG, SUM, MIN, MAX, COUNT
  - Uso de la palabra clave DISTINCT en funciones de grupo
  - Valores NULL en una función de grupo
- Agrupar filas:
  - Cláusula GROUP BY
  - Cláusula HAVING
- Anidamiento de funciones de grupo

# ¿Qué Son las Funciones de Grupo?

Las funciones de grupo funcionan en juegos de filas para proporcionar un resultado por grupo.

## EMPLOYEES

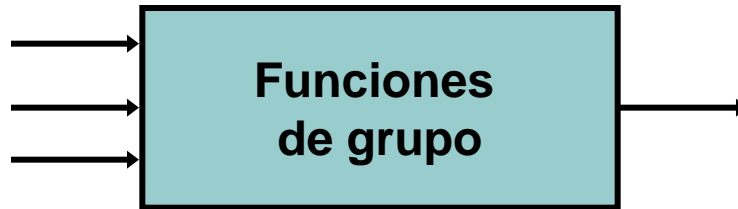
	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	110	12000
5	110	8300
6	90	24000
7	90	17000
8	90	17000
9	60	9000
10	60	6000
...		
18	80	11000
19	80	8600
20	(null)	7000

**Salario máximo en  
la tabla EMPLOYEES**

MAX(SALARY)
24000

# Tipos de Funciones de Grupo

- AVG
- COUNT
- MAX
- MIN
- STDDEV
- SUM
- VARIANCE



# Funciones de Grupo: Sintaxis

```
SELECT      group_function(column), ...  
FROM        table  
[WHERE      condition]  
[ORDER BY   column];
```

# Uso de las Funciones AVG y SUM

Puede utilizar AVG y SUM para datos numéricos.

```
SELECT AVG(salary), MAX(salary),  
       MIN(salary), SUM(salary)  
FROM   employees  
WHERE  job_id LIKE '%REP%';
```

	AVG(SALARY)	MAX(SALARY)	MIN(SALARY)	SUM(SALARY)
1	8150	11000	6000	32600

# Uso de Funciones MIN y MAX

Puede utilizar `MIN` y `MAX` para tipos de dato numéricos, de caracteres y de fecha.

```
SELECT MIN(hire_date), MAX(hire_date)  
FROM employees;
```

	MIN(HIRE_DATE)	MAX(HIRE_DATE)
1	17-JUN-87	29-JAN-00



# Uso de la Función COUNT

COUNT (\*) devuelve el número de filas en una tabla:

1

```
SELECT COUNT(*)  
FROM employees  
WHERE department_id = 50;
```

	COUNT(*)
1	5

COUNT(expr) devuelve el número de filas con valores no nulos para la expresión *expr*:

2

```
SELECT COUNT(commission_pct)  
FROM employees  
WHERE department_id = 80;
```

	COUNT(COMMISSION_PCT)
1	3

# Uso de la Palabra Clave DISTINCT

- `COUNT (DISTINCT expr)` devuelve el número con valores distintos no nulos de *expr*.
- Para mostrar el número de valores distintos de departamento en la tabla `EMPLOYEES`:

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

	COUNT(DISTINCTDEPARTMENT_ID)
1	7

# Funciones de Grupo y Valores Nulos

Las funciones de grupo ignoran los valores nulos de la columna:

1

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

	AVG(COMMISSION_PCT)
1	0.2125

La función `NVL` fuerza las funciones de grupo para que incluyan valores nulos.

2

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

	AVG(NVL(COMMISSION_PCT,0))
1	0.0425

# Agenda

- Funciones de grupo:
  - Tipos y sintaxis
  - Uso de AVG, SUM, MIN, MAX, COUNT
  - Uso de la palabra clave DISTINCT en funciones de grupo
  - Valores NULL en una función de grupo
- Agrupar filas:
  - Cláusula GROUP BY
  - Cláusula HAVING
- Anidamiento de funciones de grupo

# Creación de Grupos de Datos

## EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	50	2500
5	50	2600
6	50	3100
7	50	3500
8	50	5800
9	60	9000
10	60	6000
11	60	4200
12	80	11000
13	80	8600
...		
18	110	8300
19	110	12000
20	(null)	7000

4400

9500

3500

6400

10033

Salario medio en la  
tabla **EMPLOYEES** para  
cada departamento

	DEPARTMENT_ID	AVG(SALARY)
1	(null)	7000
2	20	9500
3	90	19333.333333333333...
4	110	10150
5	50	3500
6	80	10033.333333333333...
7	10	4400
8	60	6400

# Creación de Grupos de Datos: Sintaxis de la Cláusula GROUP BY

Puede dividir las filas de una tabla en grupos más pequeños utilizando la cláusula GROUP BY.

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

# Uso de la Cláusula GROUP BY

Todas las columnas de la lista `SELECT` que no están incluidas en las funciones de grupo deben estar en la cláusula `GROUP BY`.

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id;
```

	DEPARTMENT_ID	AVG(SALARY)
1	(null)	7000
2	20	9500
3	90	19333.333333333333...
4	110	10150
5	50	3500
6	80	10033.333333333333...
7	10	4400
8	60	6400

# Uso de la Cláusula GROUP BY

No es necesario que la columna GROUP BY esté en la lista SELECT.

```
SELECT    AVG(salary)
FROM      employees
GROUP BY  department_id ;
```

	AVG(SALARY)
1	7000
2	9500
3	19333.333333333333333333...
4	10150
5	3500
6	10033.333333333333333333...
7	4400
8	6400



# Agrupación de Más de Una Columna

## EMPLOYEES

	DEPARTMENT_ID	JOB_ID	SALARY
1	10	AD_ASST	4400
2	20	MK_MAN	13000
3	20	MK_REP	6000
4	50	ST_CLERK	2500
5	50	ST_CLERK	2600
6	50	ST_CLERK	3100
7	50	ST_CLERK	3500
8	50	ST_MAN	5800
9	60	IT_PROG	9000
10	60	IT_PROG	6000
11	60	IT_PROG	4200
12	80	SA_REP	11000
13	80	SA_REP	8600
14	80	SA_MAN	10500
...			
19	110	AC_MGR	12000
20	(null)	SA_REP	7000

Agregar los salarios en la tabla  
EMPLOYEES para cada cargo,  
agrupado por departamento.

	DEPARTMENT_ID	JOB_ID	SUM(SALARY)
1	110	AC_ACCOUNT	8300
2	110	AC_MGR	12000
3	10	AD_ASST	4400
4	90	AD_PRES	24000
5	90	AD_VP	34000
6	60	IT_PROG	19200
7	20	MK_MAN	13000
8	20	MK_REP	6000
9	80	SA_MAN	10500
10	80	SA_REP	19600
11	(null)	SA_REP	7000
12	50	ST_CLERK	11700
13	50	ST_MAN	5800

# Uso de la Cláusula GROUP BY en Varias Columnas

```
SELECT    department_id, job_id, SUM(salary)
FROM      employees
WHERE     department_id > 40
GROUP BY  department_id, job_id
ORDER BY  department_id;
```

	DEPARTMENT_ID	JOB_ID	SUM(SALARY)
1	50	ST_CLERK	11700
2	50	ST_MAN	5800
3	60	IT_PROG	19200
4	80	SA_MAN	10500
5	80	SA_REP	19600
6	90	AD_PRES	24000
7	90	AD_VP	34000
8	110	AC_ACCOUNT	8300
9	110	AC_MGR	12000

# Consultas No Válidas Realizadas con las Funciones de Grupo

Cualquier columna o expresión de la lista `SELECT` que no sea una función de agregación debe estar en la cláusula `GROUP BY`:

```
SELECT department_id, COUNT(last_name)
FROM employees;
```

ORA-00937: not a single-group group function  
00937. 00000 - "not a single-group group function"

**Una cláusula `GROUP BY` se debe agregar para el recuento de los apellidos para cada `department_id`.**

```
SELECT department_id, job_id, COUNT(last_name)
FROM employees
GROUP BY department_id ;
```

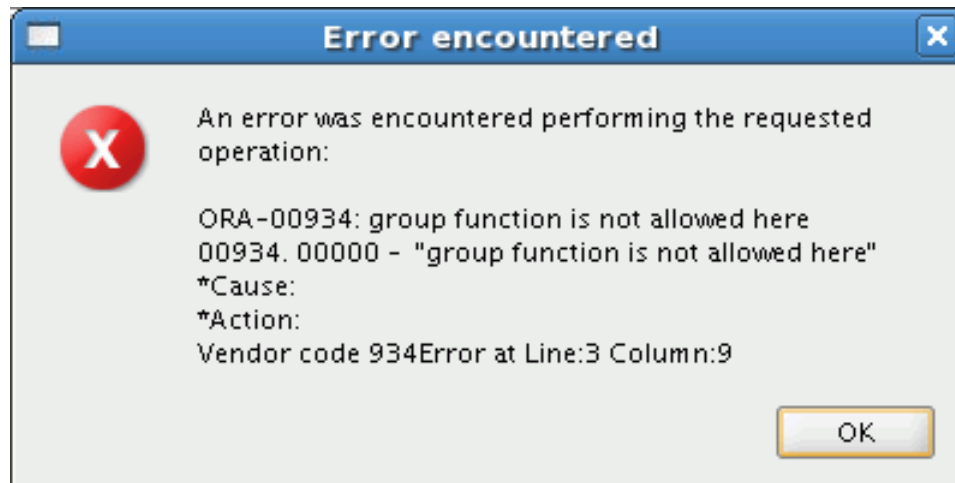
ORA-00979: not a GROUP BY expression  
00979. 00000 - "not a GROUP BY expression"

**Agregar `job_id` en `GROUP BY` o eliminar la columna `job_id` de la lista `SELECT`.**

# Consultas No Válidas Realizadas con las Funciones de Grupo

- No puede utilizar la cláusula `WHERE` para restringir grupos.
- Debe utilizar la cláusula `HAVING` para restringir grupos.
- No puede utilizar las funciones de grupo de la cláusula `WHERE`.

```
SELECT    department_id, AVG(salary)
FROM      employees
WHERE     AVG(salary) > 8000
GROUP BY  department_id;
```



**No puede utilizar la cláusula `WHERE` para restringir grupos**

# Restricción de Resultados de Grupo

## EMPLOYEES

	DEPARTMENT_ID	SALARY
1	10	4400
2	20	13000
3	20	6000
4	50	2500
5	50	2600
6	50	3100
7	50	3500
8	50	5800
9	60	9000
10	60	6000
11	60	4200
12	80	11000
13	80	8600
...		
18	110	8300
19	110	12000
20	(null)	7000

**Salario máximo por departamento cuando es superior a 10.000 dólares**

	DEPARTMENT_ID	MAX(SALARY)
1	20	13000
2	90	24000
3	110	12000
4	80	11000

# Restricción de Resultados de Grupo con la Cláusula HAVING

Al utilizar la cláusula `HAVING`, el servidor de Oracle restringe los grupos de la siguiente forma:

1. Agrupa las filas.
2. Aplica la función de grupo.
3. Muestra los grupos que coinciden con la cláusula `HAVING`.

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



# Uso de la Cláusula HAVING

```
SELECT    department_id, MAX(salary)
FROM      employees
GROUP BY  department_id
HAVING    MAX(salary)>10000 ;
```

	DEPARTMENT_ID	MAX(SALARY)
1	20	13000
2	90	24000
3	110	12000
4	80	11000

# Uso de la Cláusula HAVING

```
SELECT    job_id, SUM(salary) PAYROLL
FROM      employees
WHERE     job_id NOT LIKE '%REP%'
GROUP BY  job_id
HAVING    SUM(salary) > 13000
ORDER BY  SUM(salary);
```

	 JOB_ID	 PAYROLL
1	IT_PROG	19200
2	AD_PRES	24000
3	AD_VP	34000



# Agenda

- Funciones de grupo:
  - Tipos y sintaxis
  - Uso de AVG, SUM, MIN, MAX, COUNT
  - Uso de la palabra clave DISTINCT en funciones de grupo
  - Valores NULL en una función de grupo
- Agrupar filas:
  - Cláusula GROUP BY
  - Cláusula HAVING
- Anidamiento de funciones de grupo

# Anidamiento de Funciones de Grupo

## Mostrar el salario máximo medio:

```
SELECT MAX (AVG (salary))
FROM employees
GROUP BY department id;
```

[illegible]

# Prueba

Identificar las instrucciones para las funciones de grupo y la cláusula `GROUP BY`.

1. No puede utilizar un alias de columna en la cláusula `GROUP BY`.
2. La columna `GROUP BY` debe estar en la cláusula `SELECT`.
3. Al utilizar la cláusula `WHERE`, puede excluir las filas antes de dividir las en grupos.
4. La cláusula `GROUP BY` agrupa filas y asegura el orden del juego de resultados.
5. Si incluye una función de grupo en una cláusula `SELECT`, no puede seleccionar también resultados individuales.

# Resumen

En esta lección, debe haber aprendido lo siguiente:

- Utilizar funciones de grupo COUNT, MAX, MIN, SUM y AVG
- Escribir consultas que utilicen la cláusula GROUP BY
- Escribir consultas que utilicen la cláusula HAVING

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

# Práctica 5: Visión General

En esta práctica se abordan los siguientes temas:

- Escritura de consultas que utilizan las funciones de grupo
- Agrupación por filas para obtener más de un resultado
- Restricción de grupos utilizando la cláusula `HAVING`