

# ORACLE

## Academy

# Database Programming with SQL

9-1

## Usando as Cláusulas Group By e Having

**ORACLE**  
Academy



# Objetivos

- Esta lição abrange os seguintes objetivos:
  - Construir e executar uma consulta SQL usando GROUP BY
  - Construir e executar uma consulta SQL usando GROUP BY ... HAVING
  - Construir e executar GROUP BY em mais de uma coluna
  - Aninhar funções de grupo

# Finalidade

- E se você quisesse saber a média de altura de todos os alunos?
- Você poderia gravar uma consulta como esta:

```
SELECT AVG(height) FROM students;
```



# Finalidade

- Mas e se você quisesse saber a média de altura dos alunos com base em seu ano na escola?
- Com o que sabe até agora, você precisaria gravar um número de instruções SQL diferentes para obter isso:

```
SELECT AVG(height) FROM students WHERE year_in_school = 10;
```

```
SELECT AVG(height) FROM students WHERE year_in_school = 11;
```

```
SELECT AVG(height) FROM students WHERE year_in_school = 12;
```

- E assim por diante!
- Para simplificar problemas como esse com apenas uma instrução, use as cláusulas GROUP BY e HAVING

# Uso de GROUP BY

- A cláusula GROUP BY é usada para dividir as linhas em uma tabela em grupos menores
- Em seguida, é possível usar as funções de grupo para retornar informações resumidas sobre cada grupo

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

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333333333333333
90	19333.3333333333333333
110	10150
-	7000

# Uso de GROUP BY

- Na instrução SELECT mostrada, as linhas foram agrupadas por id\_departamento
- Em seguida, a função AVG é aplicada a cada grupo

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

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333333333333333
90	19333.3333333333333333
110	10150
-	7000

# Exemplo de GROUP BY

- E se quiséssemos encontrar o salário máximo dos funcionários de cada departamento?
- Usamos a cláusula GROUP BY especificando qual coluna deve ser usada para agrupar as linhas

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

The diagram illustrates the grouping process. On the left, a large table shows individual salary entries. A red arrow points from the top row of this table to a smaller table on the right, which contains the maximum salary for each department. The large table has columns 'DEPT\_ID' and 'SALARY'. The small table has a single column 'MAX(SALARY)'.

DEPT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
...	...

MAX(SALARY)
7000
24000
13000
...

# Exemplo de GROUP BY

- Mas como podemos saber o salário máximo que pertence a cada departamento?

DEPT_ID	SALARY	MAX(SALARY)
90	24000	7000
90	17000	24000
90	17000	13000
60	9000	
60	6000	
60	4200	
50	5800	
50	3500	
50	3100	
50	2600	
50	2500	
...	...	...

# GROUP BY em SELECT

- Normalmente, incluímos a coluna GROUP BY na lista SELECT

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

DEPT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
...	...

DEPT_ID	MAX(SALARY)
-	7000
90	24000
20	13000
...	...

# Cláusula GROUP BY

- As funções de grupo requerem que qualquer coluna listada na cláusula SELECT que não faça parte de uma função de grupo seja listada em uma cláusula GROUP BY
- O que há de errado no exemplo a seguir?

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



ORA-00979: not a GROUP BY expression (não é uma expressão GROUP BY)

# COUNT

- Este exemplo mostra quantos países existem em cada região
- Lembre-se de que as funções de grupo ignoram os valores nulos, portanto, se um dos países não tiver um nome, ele não será incluído na contagem

```
SELECT COUNT(country_name) , region_id
FROM wf_countries
GROUP BY region_id
ORDER BY region_id;
```

COUNT(COUNTRY_NAME)	REGION_ID
15	5
28	9
21	11
8	13
7	14
8	15
5	17
17	18

# COUNT

- É claro que isso seria improvável, mas, quando se constrói instruções SQL, é preciso pensar em todas as possibilidades
- Seria melhor gravar a consulta usando COUNT(\*):

```
SELECT COUNT(*), region_id
FROM wf_countries
GROUP BY region_id
ORDER BY region_id;
```

- Isso contaria todas as linhas em cada grupo de regiões, sem a necessidade de verificar quais colunas contêm valores nulos

# Cláusula WHERE

- Também podemos usar a cláusula WHERE para excluir linhas antes que o restante seja agrupado

```
SELECT department_id, MAX(salary)
FROM employees
WHERE last_name != 'King'
GROUP BY department_id;
```

LAST_NAME	DEPT_ID	SALARY
King	90	24000
Kochhar	90	17000
De Haan	90	17000
Hunold	60	9000
Ernst	60	6000
Lorentz	60	4200
...	...	...

DEPT_ID	MAX(SALARY)
-	7000
90	17000
20	13000
...	...

# Mais Exemplos de GROUP BY

- Mostre a população média de todos os países em cada região
- Arredonde a média para um nome inteiro

```
SELECT region_id, ROUND(AVG(population)) AS population
FROM wf_countries
GROUP BY region_id
ORDER BY region_id;
```

- Conte o número de idiomas falados em todos os países

```
SELECT country_id, COUNT(language_id) AS "Number of
languages"
FROM wf_spoken_languages
GROUP BY country_id;
```

# Diretrizes de GROUP BY

- As diretrizes importantes a serem lembradas quando se usa uma cláusula GROUP BY são:
  - Se você incluir uma função de grupo (AVG, SUM, COUNT, MAX, MIN, STDDEV, VARIANCE) em uma cláusula SELECT junto com quaisquer outras colunas individuais, cada coluna individual também deverá aparecer na cláusula GROUP BY
  - Você não pode usar um alias de coluna na cláusula GROUP BY
  - A cláusula WHERE exclui linhas antes que sejam divididas em grupos

# Grupos dentro de GRUPOS

- Às vezes, você precisa dividir grupos em grupos menores
- Por exemplo, você pode querer agrupar todos os funcionários por departamento; depois, dentro de cada departamento, agrupá-los por cargo

```
SELECT department_id, job_id,  
count(*)  
FROM employees  
WHERE department_id > 40  
GROUP BY department_id, job_id;
```

DEPT_ID	JOB_ID	COUNT(*)
110	AC_ACCOUNT	1
50	ST_CLERK	4
80	SA_REP	2
90	AD_VP	2
50	ST_MAN	1
...	...	...

# Grupos dentro de GRUPOS

- Este exemplo mostra quantos funcionários estão ocupando cada cargo dentro de cada departamento.

```
SELECT department_id, job_id,  
count(*)  
FROM employees  
WHERE department_id > 40  
GROUP BY department_id, job_id;
```

DEPT_ID	JOB_ID	COUNT(*)
110	AC_ACCOUNT	1
50	ST_CLERK	4
80	SA_REP	2
90	AD_VP	2
50	ST_MAN	1
...	...	...

# Agrupando Funções de Grupo

- As função de grupo podem ser aninhadas até uma profundidade de dois quando GROUP BY for usado

```
SELECT max(avg(salary))  
FROM employees  
GROUP by department_id;
```

- Quantos valores serão retornados por essa consulta?
- A resposta é um. A consulta encontrará a média salarial de cada departamento e, nessa lista, selecionar o valor mais alto

# HAVING

- Vamos supor que queremos encontrar o salário máximo somente dos departamentos com mais de um funcionário
- O que há de errado no exemplo a seguir?

```
SELECT department_id, MAX(salary)
FROM employees
WHERE COUNT(*) > 1
GROUP BY department_id;
```



ORA-00934: group function is not allowed here (a função de grupo não é permitida aqui)

# HAVING

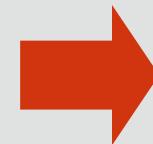
- Da mesma forma que usa a cláusula WHERE para restringir as linhas selecionadas, você pode usar a cláusula HAVING para restringir grupos
- Em uma consulta que utiliza as cláusulas GROUP BY e HAVING, as linhas são primeiro agrupadas; as funções de grupo são aplicadas; e, em seguida, são exibidos somente os grupos correspondentes à cláusula HAVING



# HAVING

- A cláusula WHERE é usada para restringir linhas; a cláusula HAVING é usada para restringir os grupos retornados por uma cláusula GROUP BY

```
SELECT department_id,MAX(salary)
FROM employees
GROUP BY department_id
HAVING COUNT(*)>1
ORDER BY department_id;
```



DEPARTMENT_ID	MAX(SALARY)
20	13000
50	5800
60	9000
80	11000
90	24000
110	12000

# HAVING

- Esta consulta encontra a população média dos países em cada região
- Em seguida, retorna somente os grupos de regiões com uma população mínima superior a 300000

```
SELECT region_id,  
       ROUND (AVG (population) )  
FROM wf_countries  
GROUP BY region_id  
HAVING MIN (population) >300000  
ORDER BY region_id;
```



REGION_ID	ROUND(AVG(POPULATION))
14	27037687
17	18729285
30	193332379
34	173268273
143	12023602
145	8522790
151	28343051

# HAVING

- Embora a cláusula HAVING possa vir antes da cláusula GROUP BY em uma instrução SELECT, é recomendado que você as posicione na ordem mostrada
- A cláusula ORDER BY (se for usada) sempre vem por último!

```
SELECT column, group_function  
FROM table  
WHERE  
GROUP BY  
HAVING  
ORDER BY
```

# Terminologia

- Estes são os principais termos usados nesta lição:
  - GROUP BY
  - HAVING

# Resumo

- Nesta lição, você deverá ter aprendido a:
  - Construir e executar uma consulta SQL usando GROUP BY
  - Construir e executar uma consulta SQL usando GROUP BY ... HAVING
  - Construir e executar GROUP BY em mais de uma coluna
  - Aninhar funções de grupo

# ORACLE

## Academy