

# 4

## Funções de Agrupamento

## Objetivos deste Capítulo

- Ao concluir este capítulo, você poderá:
  - Identificar e descrever o uso das funções de agrupamento disponíveis no Oracle
  - Agrupar dados com a cláusula `GROUP BY`
  - Restringir as linhas agrupadas com a cláusula `HAVING`

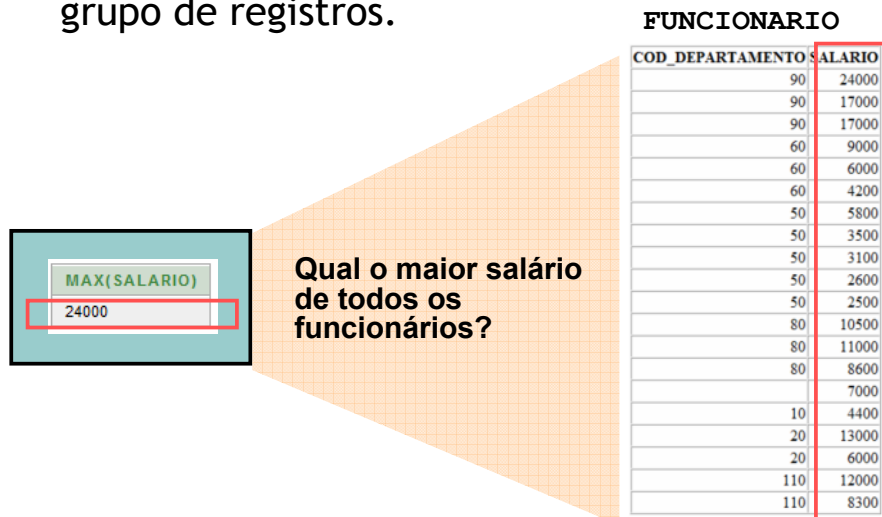
4-2

### Objetivos deste Capítulo

Este capítulo aborda o uso de funções de agrupamento do Oracle. Nos capítulos anteriores havia um relacionamento de um para um entre o retorno das consultas e os registros das tabelas. Porém, frequentemente você desejará agregar informações de vários registros e isso é conseguido através do uso de funções de agrupamento.

# Funções de Agrupamento

- As funções de agrupamento são funções que calculam um único resultado para um determinado grupo de registros.

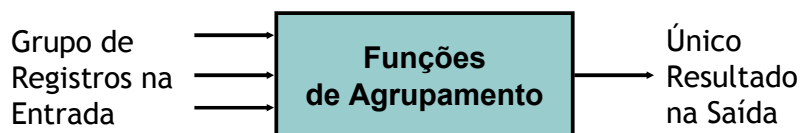


4-3

## Funções de Agrupamento

Diferentemente das funções de uma única linha, as funções de agrupamento operam em conjuntos de linhas para fornecer um resultado por grupo. Esses conjuntos podem abranger a tabela inteira ou a tabela dividida em grupos.

# Exemplos de Funções de Agrupamento



- COUNT ( )
- SUM ( )
- MIN ( )
- MAX ( )
- AVG ( )
- MEDIAN ( )
- STATS\_MODE ( )
- STDDEV ( )
- VARIANCE ( )

4-4

## Exemplos de Funções de Agrupamento

As funções de agrupamento só podem ser aplicadas a grupos de registros e retornam apenas um valor agregado derivado desse grupo de registros. A tabela a seguir lista as funções de agrupamento mais importantes no Oracle.

Função	Descrição	Aplicável a:
COUNT()	Quantidade de valores	Todos os tipos de dados
SUM()	Some de todos os valores	Dados numéricos
MIN()	Menor valor	Todos os tipos de dados
MAX()	Maior valor	Todos os tipos de dados
AVG()	Valor médio	Dados numéricos
MEDIAN()	Valor mediano	Dados numéricos ou data
STATS_MODE()	Valor mais frequente	Todos os tipos de dados
STDDEV()	Desvio padrão	Dados numéricos
VARIANCE()	Variância estatística	Dados numéricos

# Sintaxe das Funções de Agrupamento

```
SELECT      [coluna,] função_de_agrupamento(coluna), ...  
FROM        tabela  
[WHERE      condição]  
[GROUP BY   coluna]  
[ORDER BY   coluna];
```

4-5

## Sintaxe das Funções de Agrupamento

A sintaxe descrita no slide mostra como você deve utilizar as funções de agrupamento.

As diretrizes para uso das funções de agrupamento são:

- Com `DISTINCT`, a função considera apenas valores não duplicados; com `ALL`, ela considera todos os valores, inclusive os duplicados. `ALL` é o padrão e não precisa ser especificado.
- Todas as funções de agrupamento ignoram valores nulos. Para substituir um valor por valores nulos, use as funções `NVL`, `NVL2` ou `COALESCE`.
- A única função de agrupamento que não ignora valores nulos é a função `COUNT` usada com o asterisco (`*`), por exemplo, `COUNT (*)`.

# Função COUNT

- COUNT (\*) retorna o número de linhas de uma tabela:

1

```
SELECT COUNT(*)  
FROM funcionario  
WHERE cod_departamento = 50;
```

COUNT(\*)

5

- COUNT (expressão) retorna o número de linhas com valores não nulos para expressão:

2

```
SELECT COUNT(comissao)  
FROM funcionario  
WHERE cod_departamento = 80;
```

COUNT(COMISSAO)

3

4-6

## Função COUNT

A função COUNT tem três formatos:

- COUNT (\*)
- COUNT (expressão)
- COUNT (DISTINCT expressão)

COUNT (\*) retorna o número de linhas que atendem aos critérios do comando SELECT, incluindo as linhas duplicadas e as linhas com valores nulos de qualquer uma das colunas. Se uma cláusula WHERE estiver incluída no comando SELECT, COUNT (\*) retornará o número de linhas que atendem à condição especificada nessa cláusula.

Por outro lado, COUNT (expressão) retorna o número de valores não nulos na coluna identificada por expressão.

COUNT (DISTINCT expressão) retorna o número de valores exclusivos e não nulos na coluna identificada por expressão.

### Exemplos

1. O exemplo do slide mostra o número de funcionários do departamento 50.
2. O exemplo do slide mostra o número de funcionários do departamento 80 que podem receber comissão.



## Palavra-Chave

- Para retornar os valores não nulos e distintos de uma expressão você pode usar o `COUNT (DISTINCT expressão)`.
- Por exemplo, para exibir os códigos distintos dos departamentos a partir da tabela `FUNCIONARIO`:

```
SELECT COUNT(DISTINCT cod_departamento)
FROM funcionario;
```

COUNT(DISTINCT COD_DEPARTAMENTO)
7

4-8

### Palavra-Chave `DISTINCT`

Você pode usar a palavra-chave `DISTINCT` para contar apenas os valores distintos de uma determinada coluna.

O exemplo do slide mostra os valores de códigos de departamentos distintos da tabela `FUNCIONARIO`.



# Valores Nulos em Funções de Agrupamento

- Os valores nulos de uma coluna são ignorados no cálculo das funções de agrupamento:

1

```
SELECT AVG(comissao)
FROM funcionario;
```

AVG(COMISSAO)

,2125

- Para que funções de agrupamento não ignorem valores nulos você pode usar a função NVL

2

```
SELECT AVG(NVL(comissao, 0))
FROM funcionario;
```

AVG(NVL(COMISSAO,0))

,0425

4-9

## Valores Nulos em Funções de Agrupamento

Todas as funções de agrupamento ignoram valores nulos em seu cálculo.

A função NVL pode ser usada para que funções de agrupamento incluam valores nulos, substituindo-os por um valor válido.

Os exemplos do slide mostram:

1. A média das comissões dos funcionários que recebem comissão. O cálculo é feito somando-se todas as comissões não nulas e dividindo-se o resultado pela quantidade de funcionários que recebem comissão.
2. A média das comissões de todos os funcionários, mesmo os que não recebem comissão, ou seja, possuem valor nulo para a comissão. No caso do funcionário não possuir comissão, a função de agrupamento utilizará o valor 0 (zero).

# Grupos de Dados

## FUNCIONARIO

COD_DEPARTAMENTO	SALARIO
10	4400
20	13000
20	6000
50	5800
50	2500
50	2600
50	3100
50	3500
60	4200
60	6000
60	9000
80	11000
80	10500
80	8600
90	17000
90	24000
90	17000
110	8300
110	12000
	7000

Qual a média salarial de cada departamento?

COD_DEPARTAMENTO	AVG(SALARIO)
	7000.00
90	19333.33
20	9500.00
110	10150.00
50	3500.00
80	10033.33
60	6400.00
10	4400.00

4-10

## Grupos de Dados

Até agora, todas as funções de agrupamento utilizadas tratavam todos os registros das tabelas como um único grupo e retornava um único resultado para ela.

Porém, frequentemente precisamos dividir os dados de uma tabela em grupos menores e aplicar as funções de agrupamento sob esses grupos. Como no exemplo do slide, onde se precisa saber qual a média salarial por departamento. Para resolver essa questão, você precisará criar subgrupos de funcionários utilizando o departamento como condição de agrupamento. Em seguida para cada subgrupo, aplicar a função de agrupamento que calcula a média dos salários.

A cláusula `GROUP BY` é utilizada para fazer a divisão dos dados de uma tabela em subgrupos de dados.

## Sintaxe da Cláusula GROUP BY

```
SELECT    coluna, função_de_agrupamento(coluna)
FROM      tabela
[WHERE    condição
[GROUP BY expressão_de_agrupamento]
[ORDER BY coluna];
```

- A cláusula GROUP BY é responsável por dividir os dados de uma tabela em subgrupos de dados.

4-11

### Sintaxe da Cláusula GROUP BY

A cláusula GROUP BY é responsável por dividir os dados de uma tabela em subgrupos de dados. Em seguida, você pode aplicar as funções de agrupamento nesses subgrupos de dados para retornar informações sumariadas de cada subgrupo.

Na sintaxe:

*expressão\_de\_agrupamento*      especifica colunas cujos valores determinam a base do agrupamento de linhas

As diretrizes para o uso do GROUP BY são as seguinte:

- A cláusula GROUP BY deve conter todas as colunas especificadas na cláusula SELECT que não sejam funções de agrupamento, senão será retornada uma mensagem de erro.
- A cláusula WHERE pode ser usada para excluir linhas dos agrupamentos antes delas serem divididas em grupos.
- Inclua as *colunas* na cláusula GROUP BY.
- Você não pode usar um apelido de coluna na cláusula GROUP BY.

Cláusula GROUP BY

- Todas as colunas especificadas na cláusula `SELECT` que não são funções de agrupamento devem ser incluídas na cláusula `GROUP BY`, senão será retornada uma mensagem de erro.

```
SELECT    cod_departamento, AVG(salario)
FROM      funcionario
GROUP BY  cod_departamento ;
```

[illegible]

4-12

### Cláusula GROUP BY

Verifique se todas as colunas da cláusula `SELECT` que não são funções de agrupamento estão incluídas na cláusula `GROUP BY`. O exemplo do slide mostra o código e a média salarial de cada departamento.

O comando `SELECT` com uma cláusula `GROUP BY` é avaliado da seguinte maneira:

- A cláusula `SELECT` especifica as colunas a serem recuperadas da seguinte forma:  

A coluna com código do departamento da tabela `FUNCIONARIO`

A média de todos os salários no subgrupo especificado na cláusula `GROUP BY`
- A cláusula `FROM` especifica as tabelas que o banco de dados deve acessar, no exemplo, a tabela `FUNCIONARIO`.
- A cláusula `WHERE` especifica as linhas a serem recuperadas. Como não há cláusula `WHERE`, todas as linhas são recuperadas por padrão.
- A cláusula `GROUP BY` especifica como as linhas devem ser agrupadas. As linhas são agrupadas por código do departamento, portanto, a função de agrupamento `AVG` aplicada à coluna `salário` calculará *a média salarial por departamento*.

## Cláusula GROUP BY

- A coluna usada na cláusula GROUP BY não precisa estar especificada na cláusula SELECT.

- [illegible]

[illegible]

### Cláusula GROUP BY (continuação)

Você pode usar a função de agrupamento na cláusula ORDER BY:

[illegible]

## Agrupamento com Mais de Uma Coluna

FUNCIONARIO		
COD_DEPARTAMENTO	COD_CARGO	SALARIO
10	AD_ASST	4400
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
110	CT_GER	12000
110	CTPUB GER	8300
50	ES_AUX	2500
50	ES_AUX	3500
50	ES_AUX	3100
50	ES_AUX	2600
50	ES_GER	5800
60	IT_PROG	9000
60	IT_PROG	4200
60	IT_PROG	6000
20	MK_ANA	6000
20	MK_GER	13000
80	VE_GER	10500
80	VE_REP	8600
80	VE_REP	11000
	VE_REP	7000

Qual a soma dos salários dos funcionários agrupados por cargo e por departamento?

COD_DEPARTAMENTO	COD_CARGO	AVG(SALARIO)
10	AD_ASST	4400
20	MK_ANA	6000
20	MK_GER	13000
50	ES_AUX	2925
50	ES_GER	5800
60	IT_PROG	6400
80	VE_GER	10500
80	VE_REP	9800
90	AD_PRES	24000
90	AD_VP	17000
110	CT_GER	12000
110	CTPUB_GER	8300
	VE_REP	7000

4-14

### Agrupamento com Mais de Uma Coluna

Muitas vezes é necessário o agrupamento de dados dentro um outro agrupamento. O slide mostra a necessidade de criação de um relatório que exiba a soma dos salários pago a cada cargo e em cada departamento.

A tabela `FUNCIONARIO` é agrupada primeiro por código do departamento e, dentro desse agrupamento, por cargo.

# Cláusula GROUP BY com Várias Colunas

```
SELECT  cod_departamento cod_dept, cod_cargo, SUM(salario)
FROM    funcionario
GROUP BY cod_departamento, cod_cargo;
```

COD_DEPT	COD_CARGO	SUM(SALARIO)
50	ES_GER	5800
90	AD_VP	34000
80	VE_GER	10500
-	VE_REP	7000
20	MK_GER	13000
20	MK_ANA	6000
110	CTPUB_GER	8300
90	AD_PRES	24000
60	IT_PROG	19200
50	ES_AUX	11700
80	VE_REP	19600
10	AD_ASST	4400
110	CT_GER	12000

4-15

## Cláusula GROUP BY com Várias Colunas

Para criar agrupamentos de dados dentro de outros agrupamentos você deve listar mais de uma coluna na cláusula GROUP BY. Você pode determinar a ordem de ordenação padrão dos resultados pela ordem das colunas na cláusula GROUP BY. No exemplo do slide, o comando SELECT com uma cláusula GROUP BY é avaliado da seguinte maneira:

- A cláusula SELECT especifica as colunas a serem recuperadas:
  - O código do departamento na tabela FUNCIONARIO
  - O código do cargo na tabela FUNCIONARIO
  - A soma dos salários no grupo especificado na cláusula GROUP BY
- A cláusula FROM especifica as tabelas que o banco de dados deve acessar, no exemplo, a tabela FUNCIONARIO.
- A cláusula GROUP BY especifica como as linhas devem ser agrupadas, primeiro pelo código do departamento e depois pelo código do cargo dentro dos grupos de códigos do departamento.
- A função SUM é aplicada à coluna de salário para todos os códigos de cargo em cada grupo de códigos de departamento.

# Uso Incorreto de Funções de Agrupamento

- Todas as colunas ou expressões da cláusula `SELECT` que não sejam funções de agrupamento devem estar na cláusula `GROUP BY`:

```
SELECT cod_departamento, COUNT(sobrenome)
FROM funcionario;
```

```
SELECT cod_departamento, COUNT(sobrenome)
      *
ERRO na linha 1:
ORA-00937: não é uma função de grupo de grupo simples
```

4-16

## Funções de Agrupamento: Consultas Inválidas

Todas as vezes que você usar uma combinação de colunas individuais (`COD_DEPARTAMENTO`) e funções de agrupamento (`COUNT`) no mesmo comando `SELECT`, inclua uma cláusula `GROUP BY` que especifique as colunas individuais (no exemplo, `COD_DEPARTAMENTO`). Se a cláusula `GROUP BY` não for incluída, a mensagem de erro "não é uma função de grupo de grupo simples" será exibida e um asterisco (\*) indicará a coluna afetada. Para corrigir o erro no slide, adicione a cláusula `GROUP BY`:

```
SELECT cod_departamento, count(sobrenome)
FROM funcionario
GROUP BY cod_departamento;
```

COD_DEPARTAMENTO	COUNT(SOBRENOME)
-	1
90	3
20	2
110	2
50	5
80	3
60	3
10	1





# Restringindo Resultados de Agrupamentos

## FUNCIONARIO

COD_DEPARTAMENTO	SALARIO
10	4400
20	13000
20	6000
50	5800
50	2500
50	2600
50	3100
50	3500
60	4200
60	6000
60	9000
80	11000
80	10500
80	8600
90	17000
90	24000
90	17000
110	8300
110	12000
	7000

Qual o maior  
salário por  
departamento  
desde que seja  
maior que  
R\$ 10.000,00.

COD_DEPARTAMENTO	MAX(SALARIO)
90	24000
20	13000
110	12000
80	11000

4-18

## Restringindo Resultados de Agrupamentos

Assim como você utiliza a cláusula `WHERE` para restringir as linhas selecionadas, você deve usar a cláusula `HAVING` para restringir grupos. Para obter o maior salário de cada departamento desde que ele seja superior a R\$ 10.000,00, você precisa fazer o seguinte:

1. Obter o maior salário de cada departamento agrupando por código do departamento.
2. Restringir os grupos aos departamentos cujo maior salário é superior a R\$10.000,00.

# Cláusula HAVING

```
SELECT    coluna, função_de_agrupamento
FROM      tabela
[WHERE    condição]
[GROUP BY expressão_de_agrupamento]
[HAVING   condição_de_agrupamento]
[ORDER BY coluna];
```

- A cláusula `HAVING` é usada para restringir os grupos que serão retornados pela consulta.

4-19

## Cláusula HAVING

Você deve usar a cláusula `HAVING` para especificar quais grupos devem ser exibidos com base nas informações agregadas pelas funções de agrupamento.

Na sintaxe, `condição_de_agrupamento` restringe o resultado retornado aos grupos cuja condição especificada é verificada.

Quando você usa a cláusula `HAVING`, o servidor Oracle executa as seguintes etapas:

1. Os registros são agrupados.
2. A função de agrupamento é aplicada ao grupo.
3. Os grupos que satisfazem aos critérios na cláusula `HAVING` são exibidos.

A cláusula `HAVING` pode vir antes da cláusula `GROUP BY`, mas é recomendável usar a cláusula `GROUP BY` primeiro para manter a lógica de entendimento da consulta. Os grupos são formados e as funções de agrupamento são calculadas antes da cláusula `HAVING` ser aplicada aos grupos.

## Cláusula HAVING

```
SELECT      cod_departamento, MAX(salario)
FROM        funcionario
GROUP BY    cod_departamento
HAVING      MAX(salario)>10000 ;
```

COD_DEPARTAMENTO	MAX(SALARIO)
90	24000
20	13000
110	12000
80	11000

4-20

## Cláusula HAVING (continuação)

O exemplo do slide mostra os códigos e os maiores salários dos departamentos desde que esse maior salário seja superior a R\$ 10.000,00.

Pode-se usar a cláusula `GROUP BY` sem uma função de agrupamento na cláusula `SELECT`.

Para restringir os registros com base no resultado de uma função de agrupamento especifique as cláusulas `GROUP BY` e `HAVING`.

O exemplo abaixo mostra os códigos e a média salarial dos departamentos cujo maior salário é superior a R\$ 10.000,00:

```
SELECT cod_departamento, AVG(salario)
FROM funcionario
GROUP BY cod_departamento
HAVING max(salario)>10000;
```

COD_DEPARTAMENTO	AVG(SALARIO)
90	19333,3333333333333333333333333333
20	9500
110	10150
80	10033,3333333333333333333333333333

# Cláusula HAVING

```
SELECT  cod_cargo, SUM(salario) FOLHA
FROM funcionario
WHERE   cod_cargo NOT LIKE '%REP%'
GROUP BY cod_cargo
HAVING  SUM(salario) > 13000
ORDER BY SUM(salario);
```

COD_CARGO	FOLHA
IT_PROG	19200
AD PRES	24000
AD_VP	34000

4-21

## Cláusula HAVING (continuação)

O exemplo do slide mostra o código do cargo e a soma dos salários de cada cargo desde que a soma dos salários daquele cargo seja superior R\$ 13.000,00. O exemplo exclui representantes de vendas e ordena a lista pela soma do salário.

# Funções de Agrupamento Aninhadas

- Exiba a maior média salarial considerando-se grupos por departamento:

```
SELECT    MAX (AVG (salario))  
FROM      funcionario  
GROUP BY  cod_departamento;
```

MAX(AVG(SALARIO))
19333,3333333333333333333333333333

4-22

- # Funções de Agrupamento Aninhadas
- Exiba a maior média salarial considerando-se grupos por departamento:
- ```
SELECT    MAX (AVG (salario))  
FROM      funcionario  
GROUP BY  cod_departamento;
```
- | MAX(AVG(SALARIO))                  |
|------------------------------------|
| 19333,3333333333333333333333333333 |
- 4-22

# Funções de Agrupamento Aninhadas

- Exiba a maior média salarial considerando-se grupos por departamento:

```
SELECT    MAX (AVG (salario))  
FROM      funcionario  
GROUP BY  cod_departamento;
```

| MAX(AVG(SALARIO))                  |
|------------------------------------|
| 19333,3333333333333333333333333333 |

4-22

# Funções de Agrupamento Aninhadas

- Exiba a maior média salarial considerando-se grupos por departamento:

```
SELECT    MAX (AVG (salario))  
FROM      funcionario  
GROUP BY  cod_departamento;
```

| MAX(AVG(SALARIO))                  |
|------------------------------------|
| 19333,3333333333333333333333333333 |

4-22

# Funções de Agrupamento Aninhadas

- Exiba a maior média salarial considerando-se grupos por departamento:

```
SELECT    MAX (AVG (salario) )  
FROM      funcionario  
GROUP BY  cod_departamento;
```

| MAX(AVG(SALARIO))                  |
|------------------------------------|
| 19333,3333333333333333333333333333 |

4-22

# Funções de Agrupamento Aninhadas

- Exiba a maior média salarial considerando-se grupos por departamento:

```
SELECT    MAX (AVG (salario) )  
FROM      funcionario  
GROUP BY  cod_departamento;
```

| MAX(AVG(SALARIO))                  |
|------------------------------------|
| 19333,3333333333333333333333333333 |

4-22

## Funções de Agrupamento Aninhadas

Você pode aninhar até duas funções de agrupamento. O exemplo do slide mostra a maior média salarial agrupando-se por departamento.

## Funções de Agrupamento Aninhadas

Você pode aninhar até duas funções de agrupamento. O exemplo do slide mostra a maior média salarial agrupando-se por departamento.

# Exercício 4

4-23

#### Exercício 4

Determine a validade das três instruções a seguir. Escolha Verdadeiro ou Falso.

1. As funções de agrupamento trabalham com várias linhas para produzir um resultado por grupo.  
Verdadeiro/Falso
2. As funções de agrupamento incluem valores nulos em cálculos.  
Verdadeiro/Falso
3. A cláusula `WHERE` restringe as linhas antes da inclusão em um cálculo de agrupamento.  
Verdadeiro/Falso
4. Obtenha o salário máximo, o salário mínimo, a soma dos salários e o salário médio de todos os funcionários. Atribua os nomes `Maximo`, `Minimo`, `Soma` e `Media`, respectivamente, às colunas. Arredonde os resultados para o número inteiro mais próximo. Salve o comando SQL no arquivo de texto `cap_04_04.sql`

| Maximo | Minimo | Soma   | Media |
|--------|--------|--------|-------|
| 24000  | 2500   | 175500 | 8775  |

5. Modifique a consulta em `cap_04_04.sql` para exibir o salário mínimo, o salário máximo, a soma dos salários e o salário médio de cada código de cargo. Salve como `lab_04_05.sql` e o execute.

| COD_CARGO | Maximo | Minimo | Soma  | Media |
|-----------|--------|--------|-------|-------|
| AD_ASST   | 4400   | 4400   | 4400  | 4400  |
| AD_PRES   | 24000  | 24000  | 24000 | 24000 |
| AD_VP     | 17000  | 17000  | 34000 | 17000 |
| CTPUB_GER | 8300   | 8300   | 8300  | 8300  |
| CT_GER    | 12000  | 12000  | 12000 | 12000 |
| ES_AUX    | 3500   | 2500   | 11700 | 2925  |
| ES_GER    | 5800   | 5800   | 5800  | 5800  |
| IT_PROG   | 9000   | 4200   | 19200 | 6400  |
| MK_ANA    | 6000   | 6000   | 6000  | 6000  |
| MK_GER    | 13000  | 13000  | 13000 | 13000 |
| VE_GER    | 10500  | 10500  | 10500 | 10500 |
| VE_REP    | 11000  | 7000   | 26600 | 8867  |



#### Exercício 4 (continuação)

6. a) Crie uma consulta para exibir o número de pessoas com o mesmo cargo.

| COD_CARGO | COUNT(*) |
|-----------|----------|
| AD_ASST   | 1        |
| AD_PRES   | 1        |
| AD_VP     | 2        |
| CTPUB_GER | 1        |
| CT_GER    | 1        |
| ES_AUX    | 4        |
| ES_GER    | 1        |
| IT_PROG   | 3        |
| MK_ANA    | 1        |
| MK_GER    | 1        |
| VE_GER    | 1        |
| VE_REP    | 3        |

- b) Generalize a consulta para que o usuário solicitado a informar um código de cargo. Salve o script no arquivo `cap_04_06.sql`.

7. Determine o número de gerentes sem listá-los. Atribua o nome `Numero de Gerentes` à coluna. *Dica: Use a coluna `COD_GERENTE` para determinar o número de gerentes.*

| Numero De Gerentes |
|--------------------|
| 8                  |

8. Descubra a diferença entre o salário mais alto e o mais baixo. Atribua o label `DIFFERENCE` à coluna.

| DIFERENÇA |
|-----------|
| 21500     |

9. Crie um relatório para exibir o número do gerente e o salário do funcionário com menor remuneração desse gerente. Exclua todas as pessoas cujo gerente seja desconhecido. Exclua todos os grupos em que o menor salário seja R\$ 6.000 ou inferior. Ordene a saída em ordem decrescente de salário.

| COD_GERENTE | MIN(SALARIO) |
|-------------|--------------|
| 102         | 9000         |
| 205         | 8300         |
| 149         | 7000         |

#### Exercício 4 (continuação)

10. Crie uma consulta que exiba o número total de funcionários e, desse total, mostre o número de funcionários admitidos em 2005, 2006, 2007 e 2008. Crie cabeçalhos de colunas apropriados.

| TOTAL | 2005 | 2006 | 2007 | 2008 |
|-------|------|------|------|------|
| 20    | 1    | 2    | 2    | 3    |

11. Crie uma consulta matriz que exiba o cargo, o salário relativo a esse cargo com base no número do departamento e o salário total desse cargo para os departamentos 20, 50, 80 e 90, atribuindo um cabeçalho apropriado a cada coluna.

| Cargo     | Dept 20 | Dept 50 | Dept 80 | Dept 90 | Total |
|-----------|---------|---------|---------|---------|-------|
| AD_ASST   | -       | -       | -       | -       | 4400  |
| AD PRES   | -       | -       | -       | 24000   | 24000 |
| AD_VP     | -       | -       | -       | 34000   | 34000 |
| CTPUB_GER | -       | -       | -       | -       | 8300  |
| CT_GER    | -       | -       | -       | -       | 12000 |
| ES_AUX    | -       | 11700   | -       | -       | 11700 |
| ES_GER    | -       | 5800    | -       | -       | 5800  |
| IT_PROG   | -       | -       | -       | -       | 19200 |
| MK_ANA    | 6000    | -       | -       | -       | 6000  |
| MK_GER    | 13000   | -       | -       | -       | 13000 |
| VE_GER    | -       | -       | 10500   | -       | 10500 |
| VE_REP    | -       | -       | 19600   | -       | 26600 |