FIAP

FIAP GRADUAÇÃO

FIAP

Tecnologia em Análise e Desenvolvimento de Sistemas

Profº Ms. Alexandre Barcelos
profalexandre.barcelos@fiap.com.br

2019

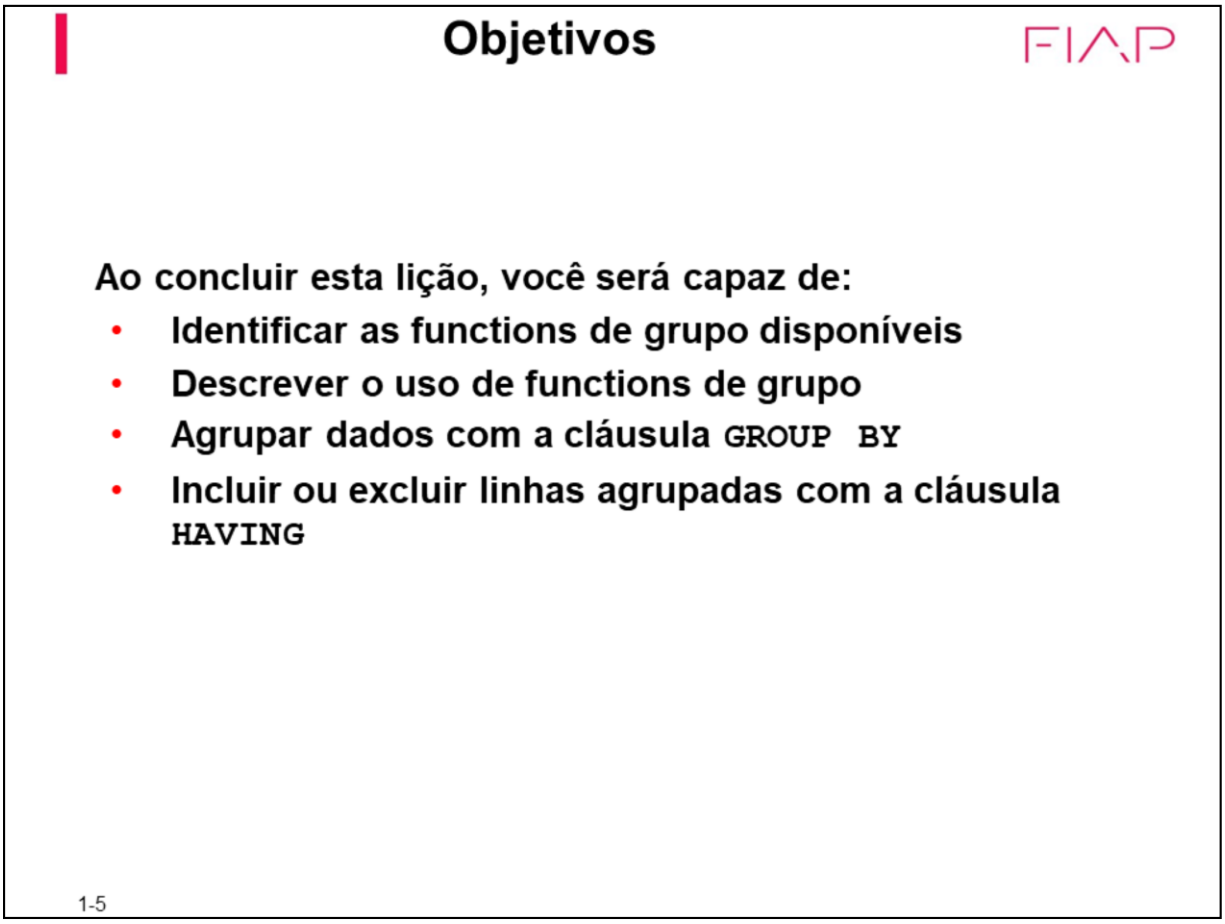
FIAP

Database Application Development

Profº Ms. Alexandre Barcelos
profalexandre.barcelos@fiap.com.br

2019

Gerando Relatórios de Dados Agregados com as Functions de Grupo



Objetivos

Esta lição aborda o uso de functions com mais detalhes. Ela concentra-se na obtenção de informações sumariadas (como médias) relativas a grupos de linhas. A lição mostra como agrupar as linhas de uma tabela em conjuntos menores e como especificar critérios de pesquisa para grupos de linhas.

FIAP

O Que São Functions de Grupo?

As functions de grupo operam em conjuntos de linhas para fornecer um resultado por grupo.

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
	7000
10	4400

...

20 rows selected.

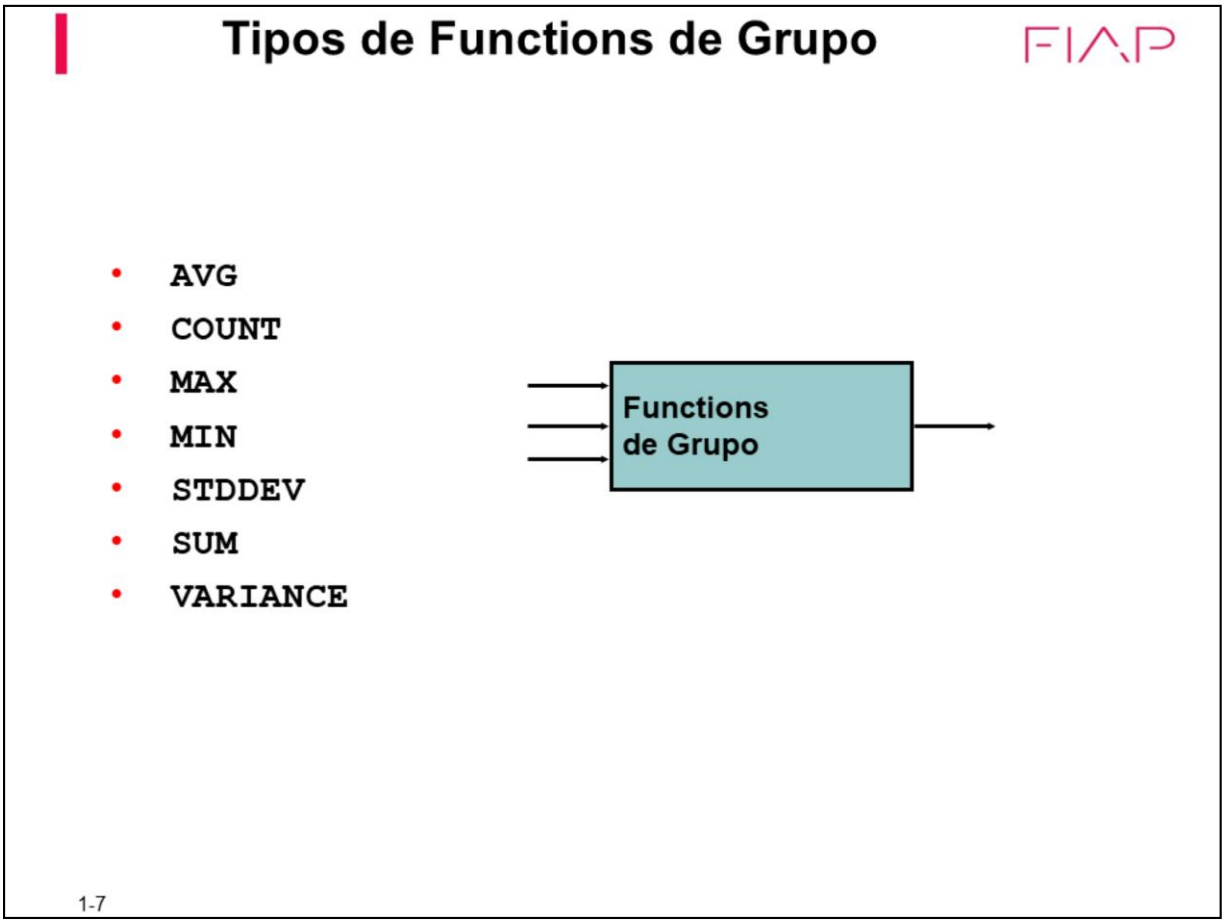
Salário máximo na tabela EMPLOYEES

MAX(SALARY)
24000

1-6

Functions de Grupo

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



Tipos de Functions de Grupo

Cada function aceita um argumento. Esta tabela identifica as opções que você pode usar na sintaxe:

Function	Descrição
AVG([DISTINCT|ALL]*n*)	Valor médio de *n*; ignora valores nulos
COUNT({*|[DISTINCT|ALL]*expr*})	Número de linhas, em que *expr* é avaliado como um valor diferente de nulo (conta todas as linhas selecionadas usando *, inclusive valores duplicados e linhas com valores nulos)
MAX([DISTINCT|ALL]*expr*)	Valor máximo de *expr*; ignora valores nulos
MIN([DISTINCT|ALL]*expr*)	Valor mínimo de *expr*; ignora valores nulos
STDDEV([DISTINCT|ALL]*x*)	Desvio padrão de *n*; ignora valores nulos
SUM([DISTINCT|ALL]*n*)	Valores somados de *n*; ignora valores nulos
VARIANCE([DISTINCT|ALL]*x*)	Variação de *n*; ignora valores nulos

Functions de Grupo: Sintaxe

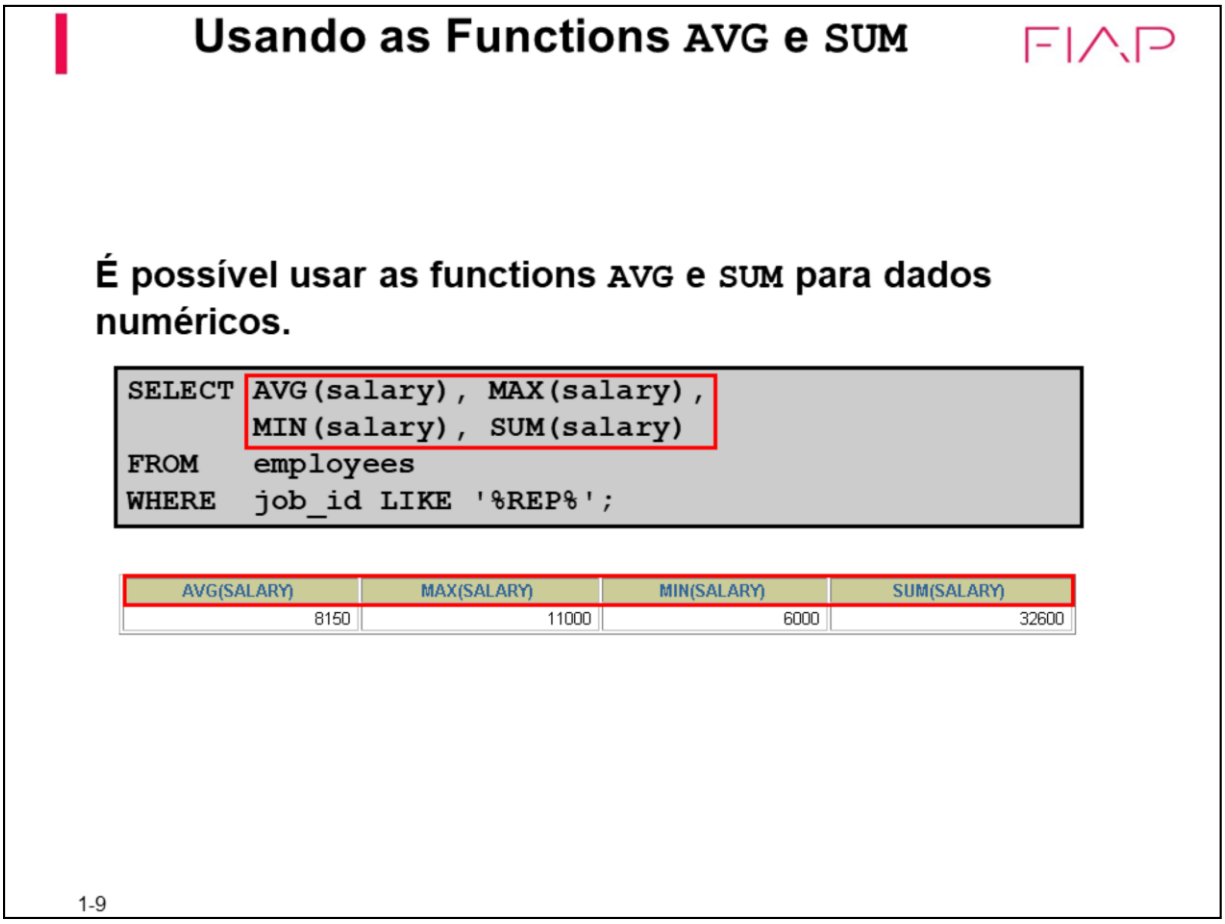
FIAP

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

1-8

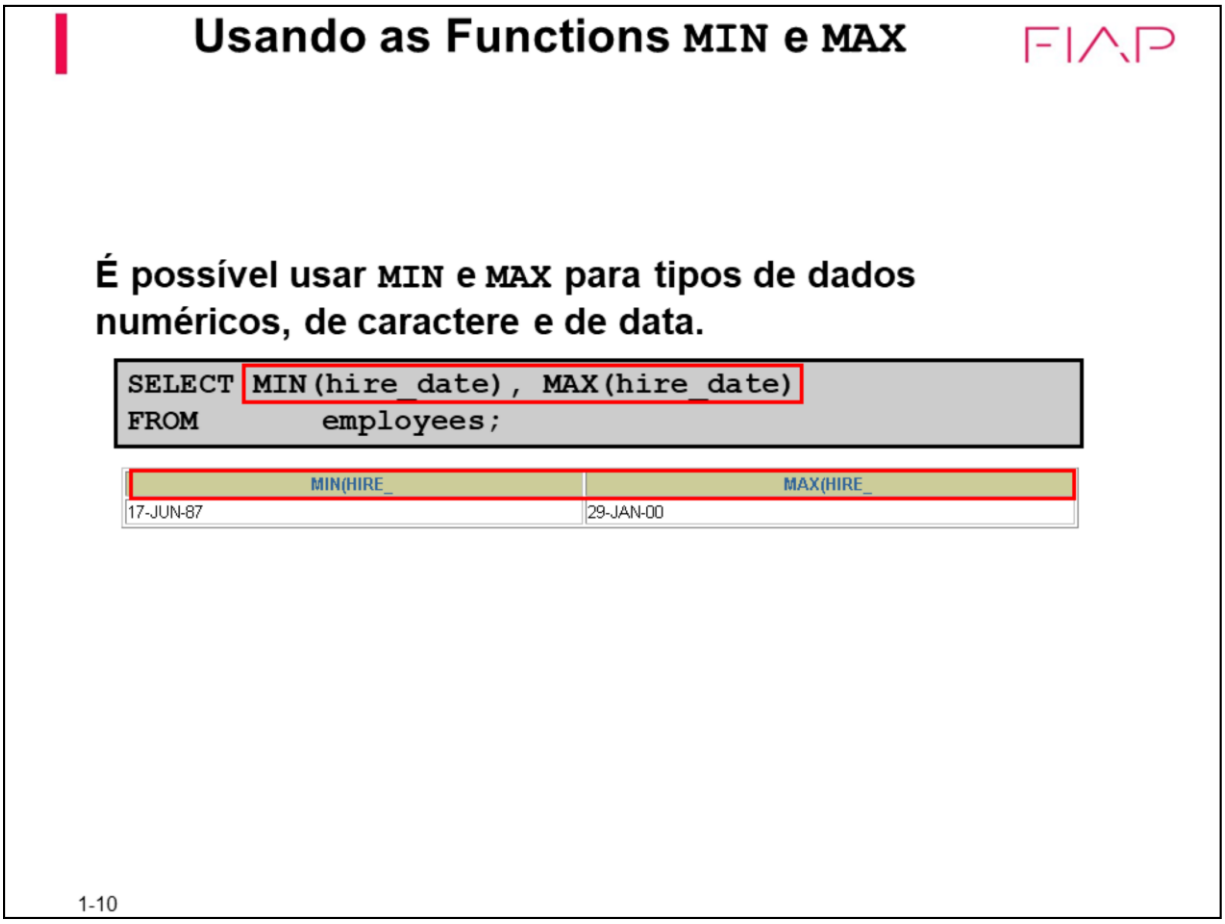
Diretrizes para a Utilização de Functions de Grupo

- Com `DISTINCT`, a function considera apenas valores não duplicados; com `ALL`, ela considera todos os valores, inclusive os duplicados. Como é o default, `ALL` não precisa ser especificado.
- Os tipos de dados das functions com um argumento `expr` podem ser `CHAR`, `VARCHAR2`, `NUMBER` ou `DATE`.
- Todas as functions de grupo ignoram valores nulos. Para substituir um valor por valores nulos, use as functions `NVL`, `NVL2` ou `COALESCE`.



Usando as Functions de Grupo

É possível usar as functions `AVG`, `SUM`, `MIN` e `MAX` em colunas que podem armazenar dados numéricos. O exemplo do slide mostra os salários médio, máximo e mínimo, bem como a soma dos salários mensais, de todos os representantes de vendas.



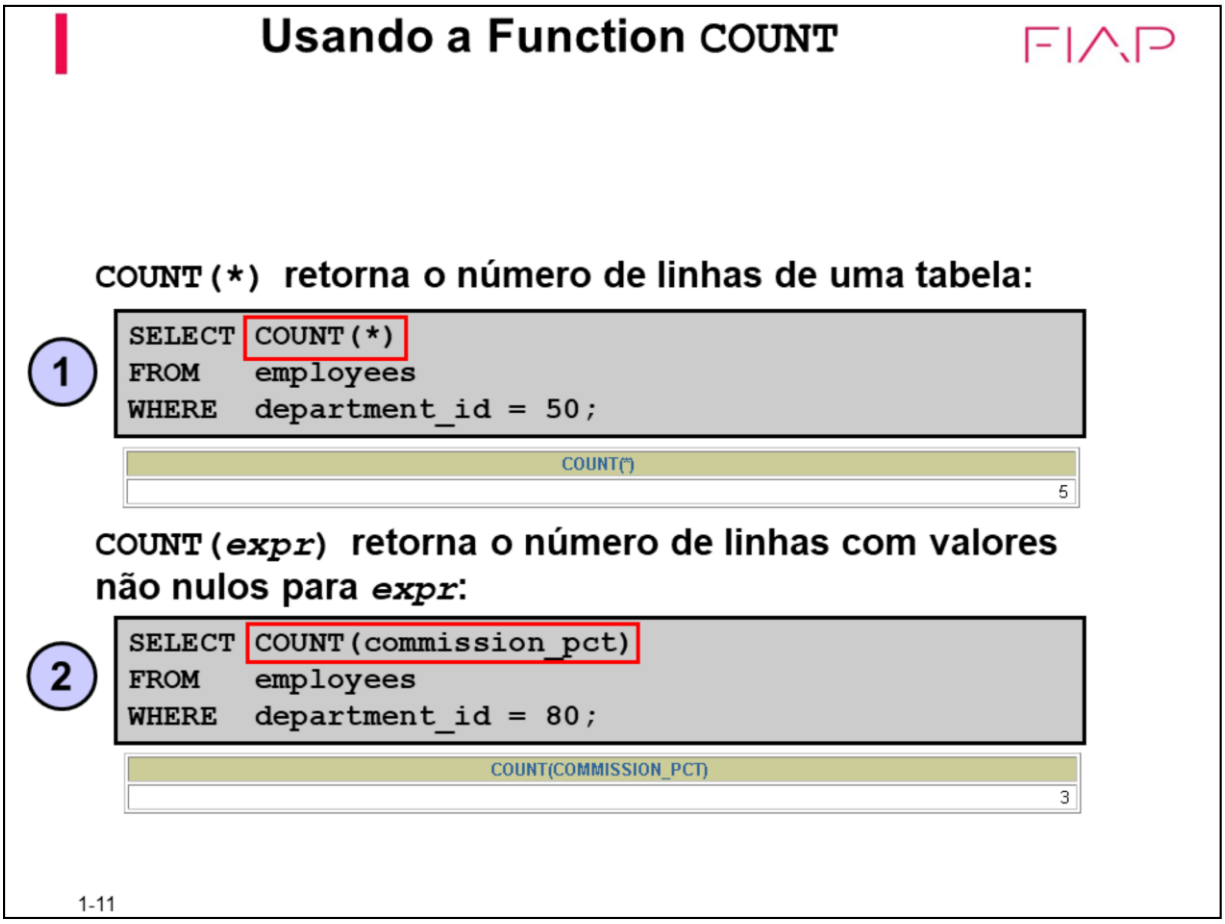
Usando as Functions de Grupo (continuação)

É possível usar as functions `MAX` e `MIN` para os tipos de dados numéricos, de caractere e de data. O exemplo do slide mostra o funcionário mais recente e o mais antigo.

O exemplo a seguir mostra o primeiro e o último sobrenome de funcionário em uma lista em ordem alfabética de todos os funcionários:

MIN(LAST_NAME)	MAX(LAST_NAME)
Abel	Zlotkey

Observação: Só é possível usar as functions `AVG`, `SUM`, `VARIANCE` e `STDDEV` com tipos de dados numéricos. `MAX` e `MIN` não podem ser usadas com o tipo de dados `LOB` ou `LONG`.



Function COUNT

A function `COUNT` tem três formatos:

- `COUNT(*)`
- `COUNT(expr)`
- `COUNT(DISTINCT expr)`

`COUNT(*)` retorna o número de linhas de uma tabela que atendem aos critérios da instrução `SELECT`, incluindo as linhas duplicadas e as linhas com valores nulos de qualquer uma das colunas. Se uma cláusula `WHERE` estiver incluída na instrução `SELECT`, `COUNT(*)` retornará o número de linhas que atendem à condição especificada nessa cláusula.

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

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

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.

Usando a Palavra-Chave DISTINCT

- COUNT(DISTINCT expr) retorna o número de valores não nulos e distintos de *expr*.
- Para exibir os valores de números de departamentos distintos da tabela EMPLOYEES:

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

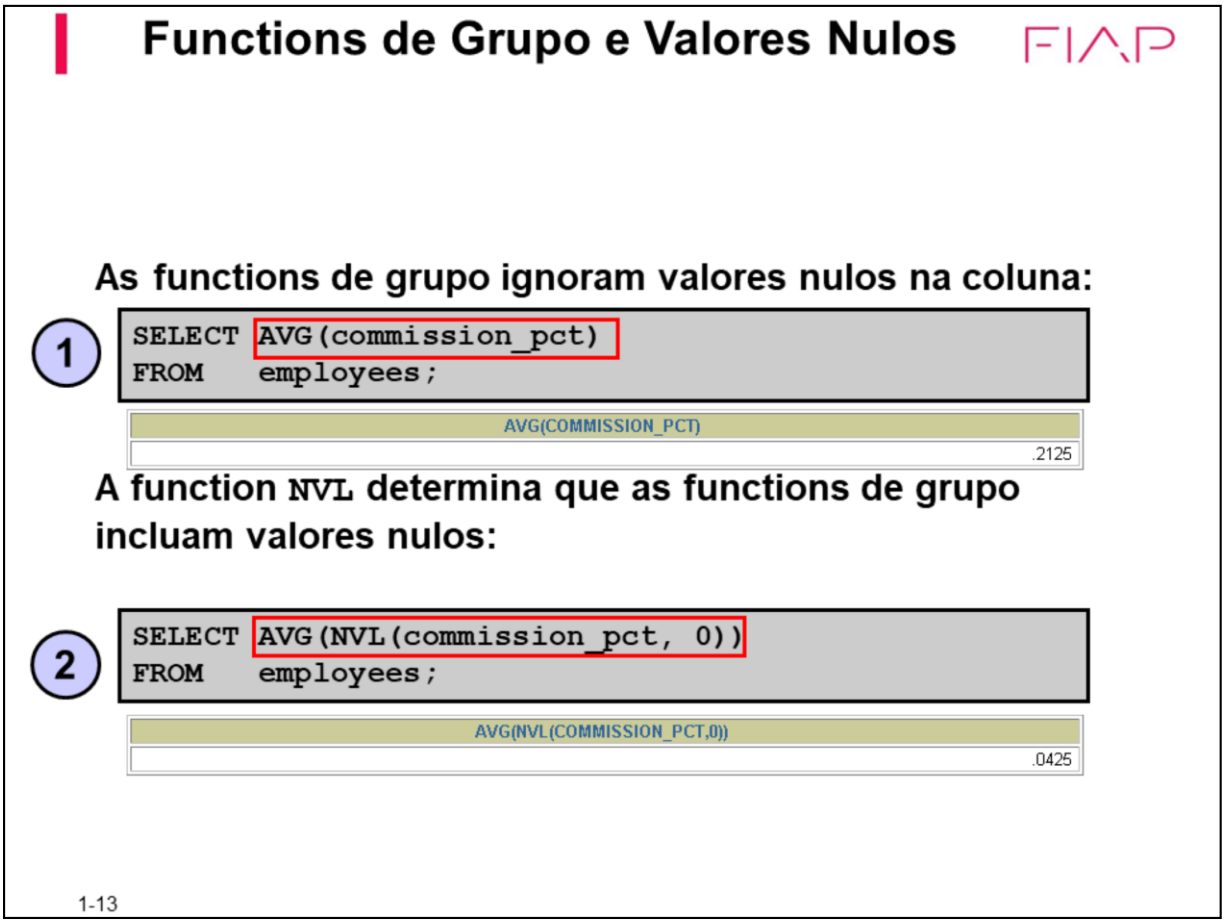
COUNT(DISTINCTDEPARTMENT_ID)
7

1-12

Palavra-chave DISTINCT

Use a palavra-chave DISTINCT para suprimir a contagem de valores duplicados em uma coluna.

O exemplo do slide mostra os valores de números de departamentos distintos da tabela EMPLOYEES.



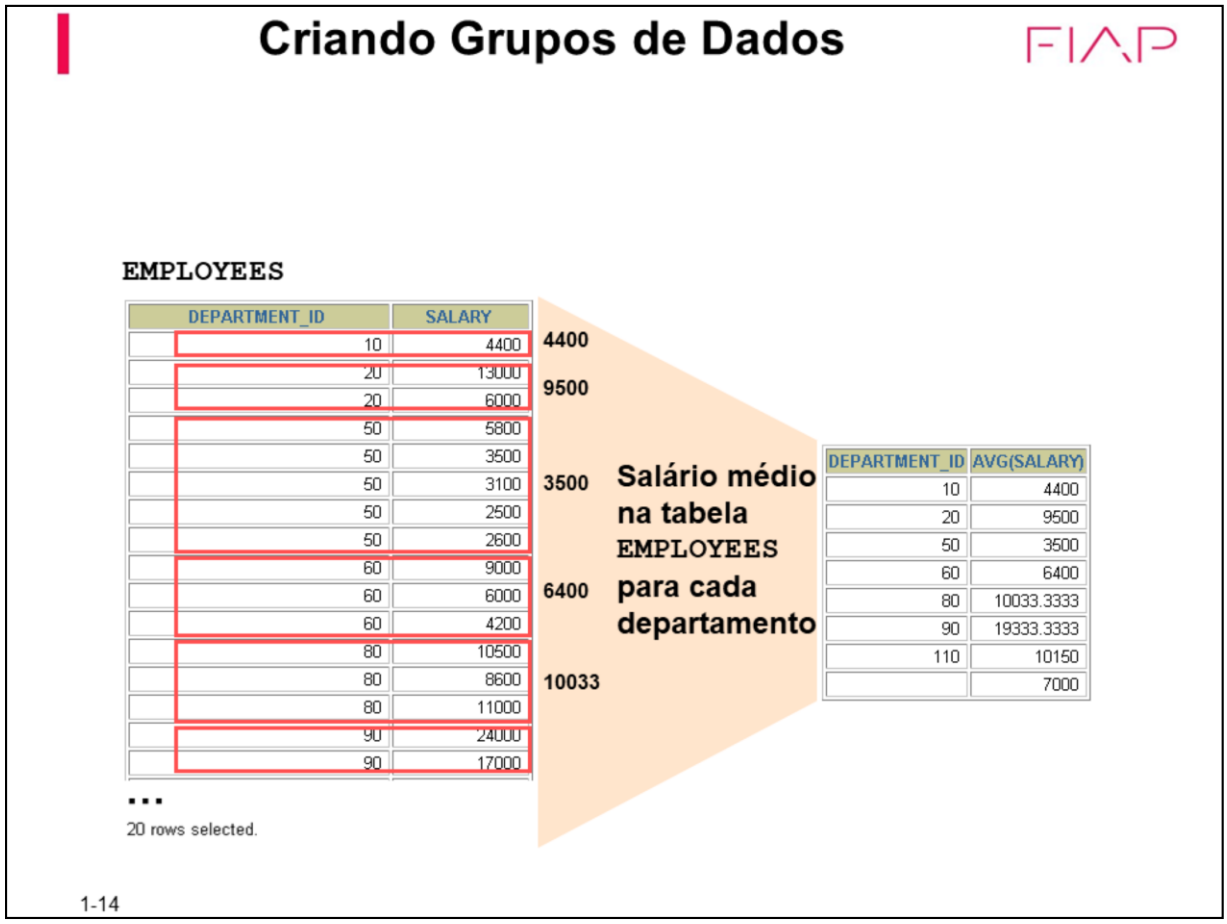
Functions de Grupo e Valores Nulos

Todas as functions de grupo ignoram valores nulos na coluna.

A function `NVL` determina que as functions de grupo incluam valores nulos.

Exemplos

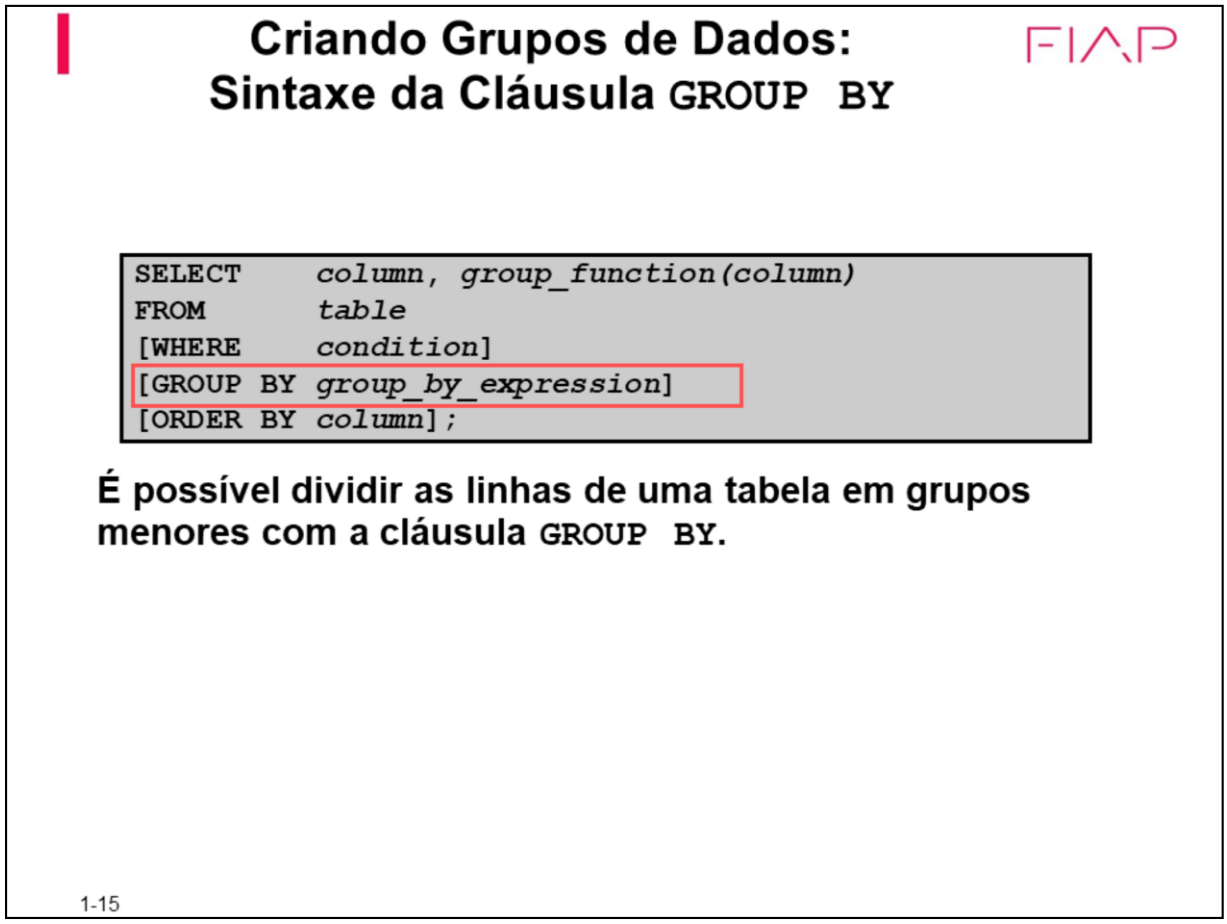
1. A média é calculada com base *apenas* nas linhas da tabela com valores válidos armazenados na coluna `COMMISSION_PCT`. A média é calculada como a comissão total paga a todos os funcionários dividida pelo número de funcionários que recebem comissão (quatro).
2. A média é calculada com base em *todas* as linhas da tabela, mesmo que valores nulos estejam armazenados na coluna `COMMISSION_PCT`. A média é calculada como a comissão total paga a todos os funcionários dividida pelo número total de funcionários da empresa (20).



Criando Grupos de Dados

Até esta etapa do nosso estudo, todas as functions de grupo trataram a tabela como um grande grupo de informações.

Entretanto, às vezes é necessário dividir a tabela de informações em grupos menores. É possível efetuar essa divisão com a cláusula `GROUP BY`.



Cláusula `GROUP BY`

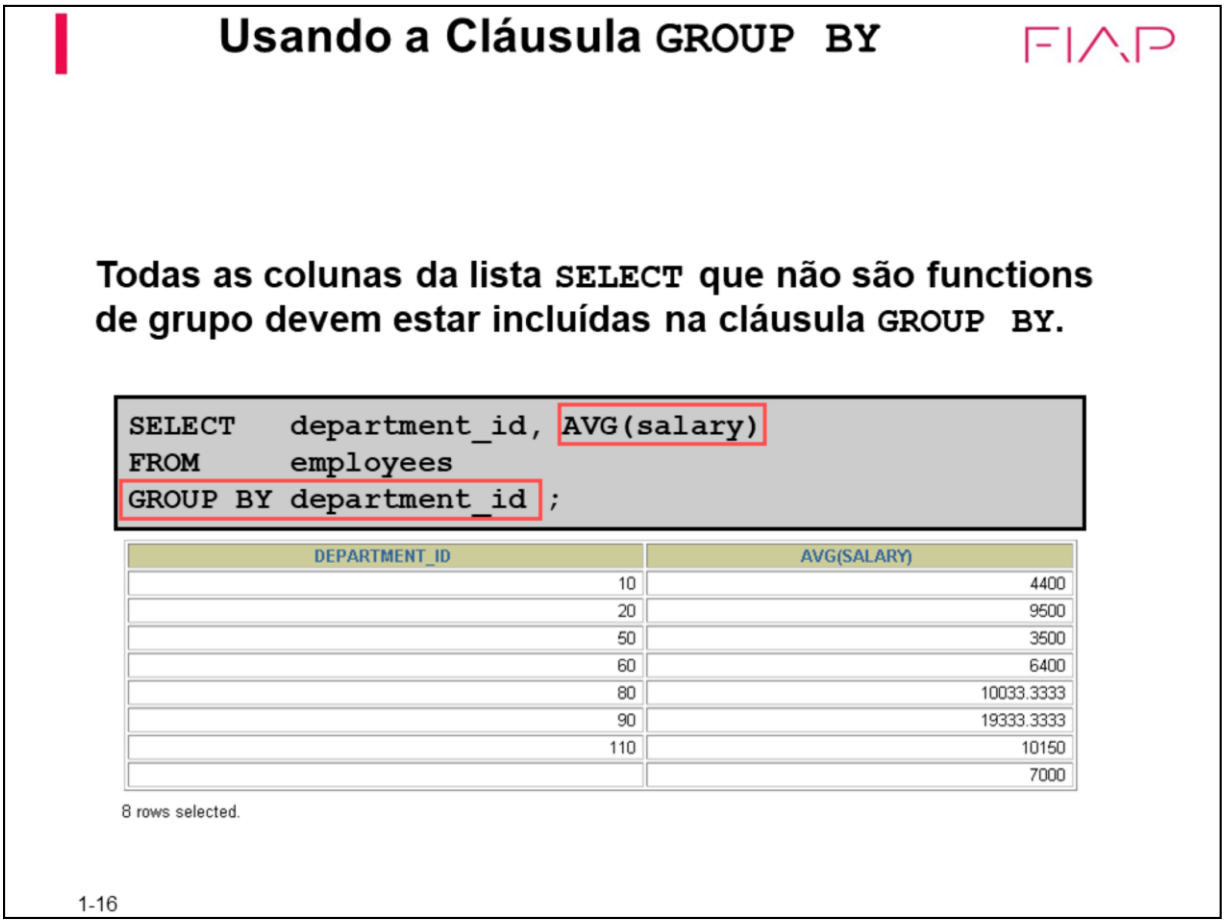
Você pode usar a cláusula `GROUP BY` para dividir as linhas de uma tabela em grupos. Depois, pode usar as functions de grupo para retornar informações sumariadas de cada grupo.

Na sintaxe:

`group_by_expression` especifica colunas cujos valores determinam a base do agrupamento de linhas

Diretrizes

- Se incluir uma function de grupo em uma cláusula `SELECT`, você não poderá selecionar resultados individuais, *a menos que* a coluna individual apareça na cláusula `GROUP BY`. Se não conseguir incluir a lista de colunas na cláusula `GROUP BY`, você receberá uma mensagem de erro.
- Com uma cláusula `WHERE`, você poderá excluir linhas antes de dividi-las em grupos.
- Inclua as *colunas* na cláusula `GROUP BY`.
- Não é possível usar um apelido de coluna na cláusula `GROUP BY`.



Usando a Cláusula `GROUP BY`

Ao usar a cláusula `GROUP BY`, verifique se todas as colunas da lista `SELECT` que não são functions de grupo estão incluídas nessa cláusula. O exemplo do slide mostra o número e o salário médio de cada departamento. A instrução `SELECT` com uma cláusula `GROUP BY` é avaliada da seguinte maneira:

- A cláusula `SELECT` especifica as colunas a serem recuperadas da seguinte forma:
  - A coluna de número de departamento da tabela `EMPLOYEES`
  - A média de todos os salários no grupo especificado na cláusula `GROUP BY`
- A cláusula `FROM` especifica as tabelas que o banco de dados deve acessar: a tabela `EMPLOYEES`.
- A cláusula `WHERE` especifica as linhas a serem recuperadas. Como não há cláusula `WHERE`, todas as linhas são recuperadas por default.
- A cláusula `GROUP BY` especifica como as linhas devem ser agrupadas. As linhas são agrupadas por número de departamento, portanto, a function `AVG` aplicada à coluna de salário calculará o *salário médio de cada departamento*.

Usando a Cláusula GROUP BY

FIAP

A cláusula GROUP BY seguida pelo nome da coluna não precisa estar na lista SELECT.

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

AVG(SALARY)
4400
9500
3500
6400
10033.3333
19333.3333
10150
7000

1-17

Usando a Cláusula GROUP BY (continuação)

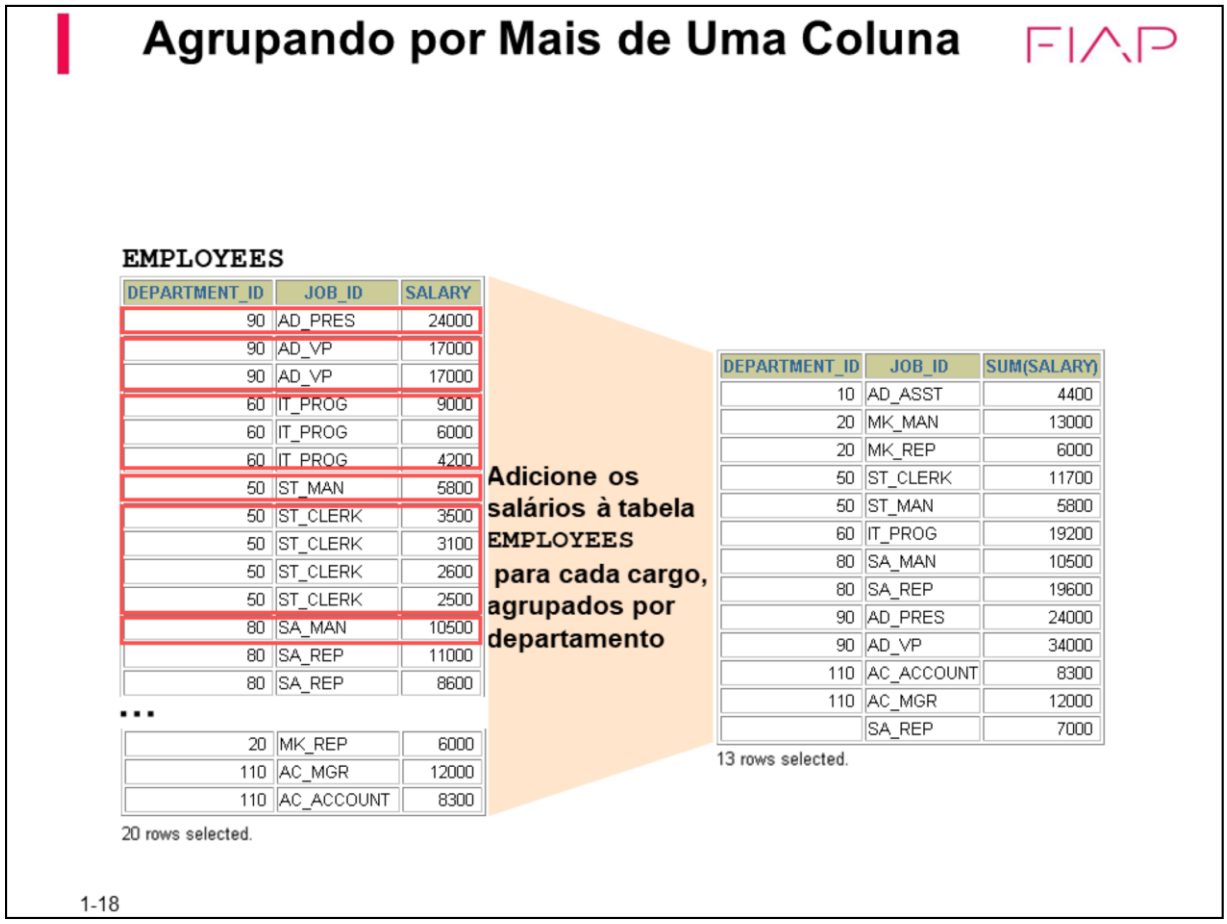
A cláusula GROUP BY seguida pelo nome da coluna não precisa estar na cláusula SELECT. Por exemplo, a instrução SELECT do slide mostra os salários médios de cada departamento sem exibir os respectivos números de departamento. No entanto, sem os números de departamento, os resultados parecem não fazer sentido.

É possível usar a function de grupo na cláusula ORDER BY:

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

DEPARTMENT_ID	AVG(SALARY)
50	3500
10	4400
60	6400
...	
90	19333.3333

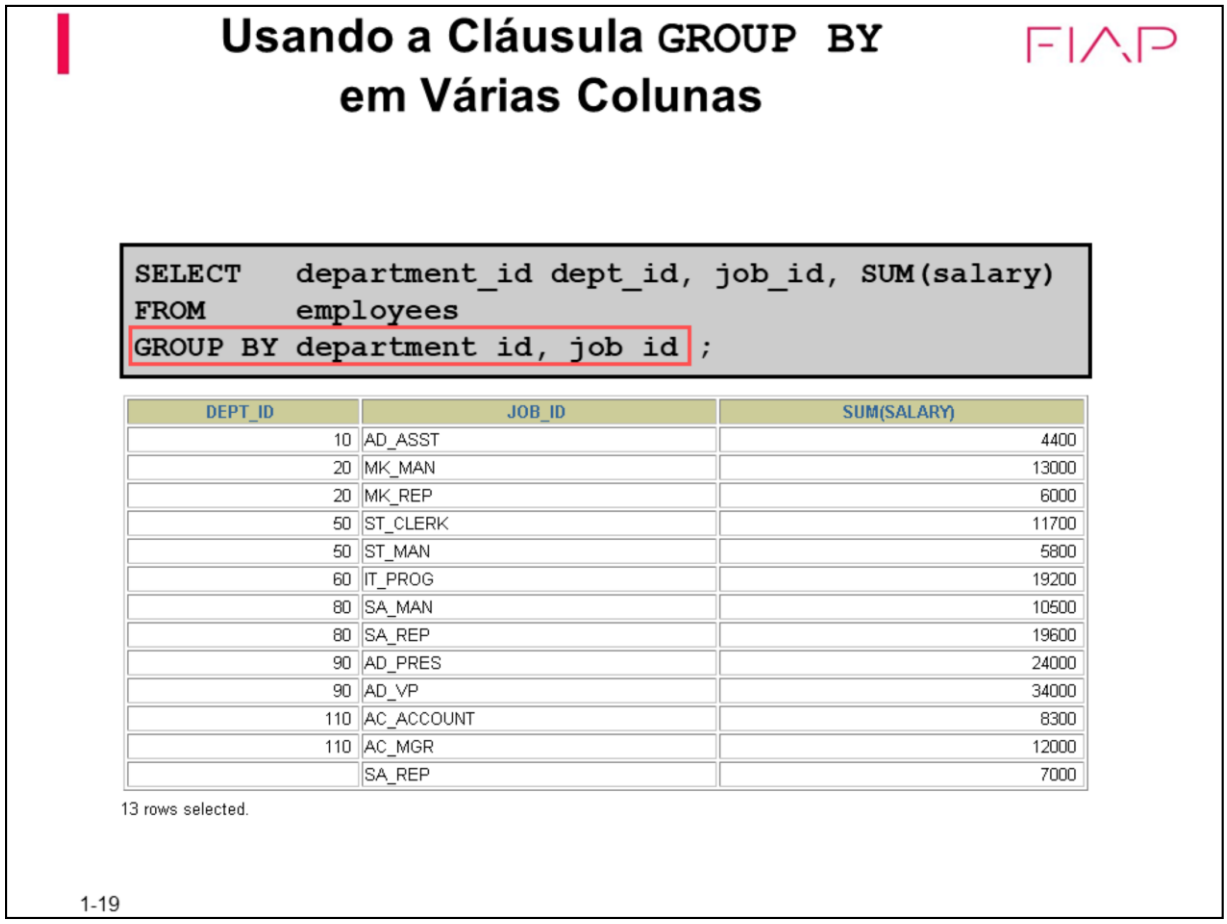
8 rows selected.



Grupos Contidos em Outros Grupos

Às vezes, você precisa ver os resultados de grupos contidos em outros grupos. O slide mostra um relatório que exibe o salário total pago a cada cargo, em cada departamento.

A tabela `EMPLOYEES` é agrupada primeiro por número de departamento e, nesse agrupamento, por cargo. Por exemplo, os quatro estoquistas do departamento 50 são agrupados e um único resultado (salário total) é fornecido para todos os estoquistas do grupo.



Grupos Contidos em Outros Grupos (continuação)

É possível obter resultados sumariados de grupos e subgrupos listando mais de uma coluna na cláusula `GROUP BY`. Você pode determinar a ordem de classificação default dos resultados pela ordem das colunas na cláusula `GROUP BY`. No exemplo do slide, a instrução `SELECT` com uma cláusula `GROUP BY` é avaliada da seguinte maneira:

- A cláusula `SELECT` especifica a coluna a ser recuperada:
  - O número do departamento na tabela `EMPLOYEES`
  - O ID do cargo na tabela `EMPLOYEES`
  - A soma de todos os salários no grupo especificado na cláusula `GROUP BY`
- A cláusula `FROM` especifica as tabelas que o banco de dados deve acessar: a tabela `EMPLOYEES`.
- A cláusula `GROUP BY` especifica como você deve agrupar as linhas:
  - Primeiro, as linhas são agrupadas por número de departamento.
  - Depois, as linhas são agrupadas por ID de cargo nos grupos de números de departamento.

Portanto, a function `SUM` é aplicada à coluna de salário para todos os IDs de cargo em cada grupo de números de departamento.

Consultas Inválidas Usando Functions de Grupo

FIAP

Qualquer coluna ou expressão da lista SELECT que não seja uma function agregada deverá estar na cláusula GROUP BY:

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

```
SELECT department_id, COUNT(last_name)
       *
ERROR at line 1:
ORA-00937: not a single-group group function
```

Coluna ausente na cláusula GROUP BY

1-20

Consultas Inválidas Usando Functions de Grupo

Sempre que usar uma combinação de itens individuais (DEPARTMENT_ID) e functions de grupo (COUNT) na mesma instrução SELECT, inclua uma cláusula GROUP BY que especifique os itens individuais (neste caso, DEPARTMENT_ID). Se a cláusula GROUP BY não for incluída, a mensagem de erro "not a single-group group function" será exibida e um asterisco (*) indicará a coluna afetada. Para corrigir o erro no slide, adicione a cláusula GROUP BY:

```
SELECT   department_id, count(last_name)
```

DEPARTMENT_ID	COUNT(LAST_NAME)
10	1
20	2
...	
	1

8 rows selected.

Qualquer coluna ou expressão da lista SELECT que não seja uma function agregada deverá

estar na cláusula `GROUP BY`.

Consultas Inválidas Usando Functions de Grupo

FIAP

- Não é possível usar a cláusula `WHERE` para restringir grupos.
- Use a cláusula `HAVING` para restringir grupos.
- Não é possível usar functions de grupo na cláusula `WHERE`.

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

```
WHERE  AVG(salary) > 8000
       *
ERROR at line 3:
ORA-00934: group function is not allowed here
```

Não é possível usar a cláusula `WHERE` para restringir grupos

1-21

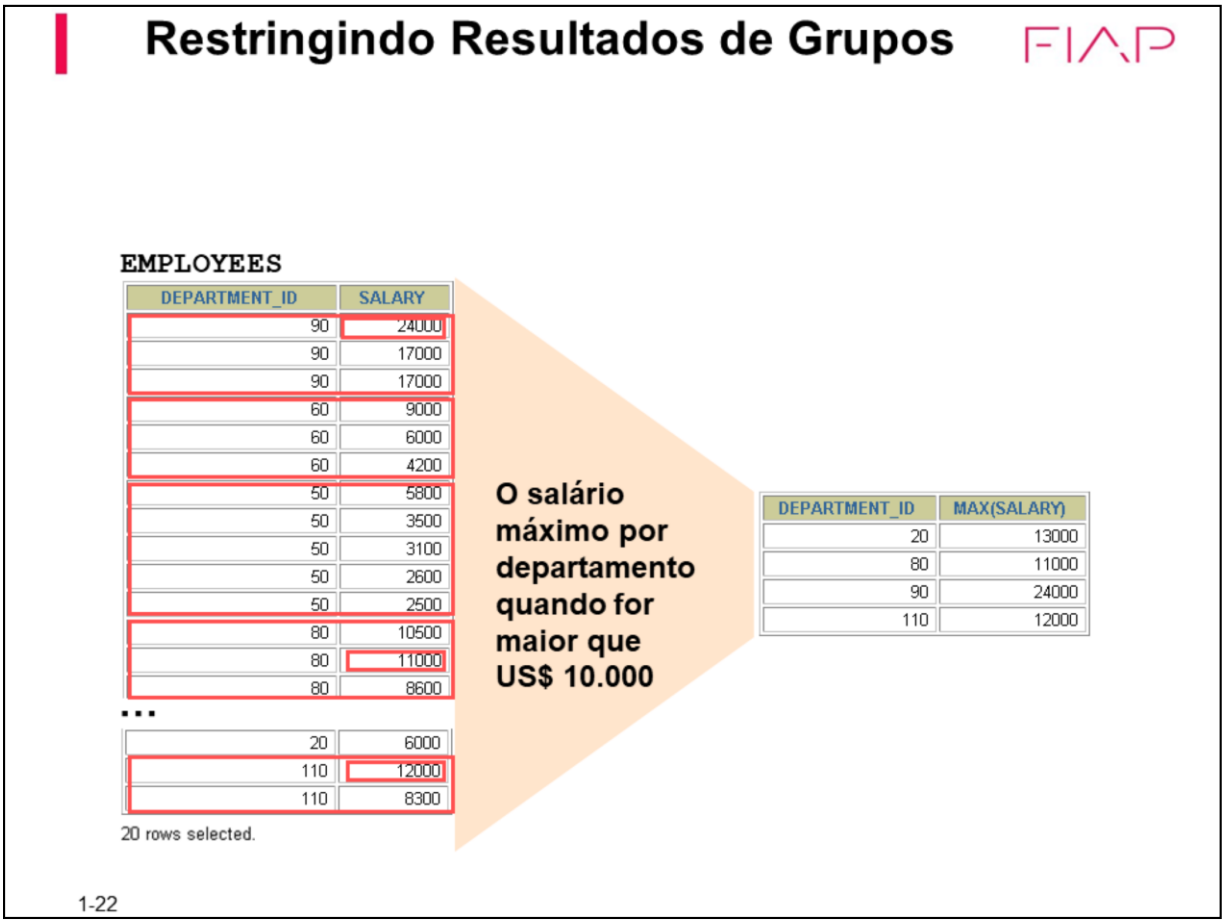
Consultas Inválidas Usando Functions de Grupo (continuação)

Não é possível usar a cláusula `WHERE` para restringir grupos. A instrução `SELECT` do exemplo do slide resulta em erro, pois utiliza a cláusula `WHERE` para restringir a exibição dos salários médios dos departamentos cujo salário médio é superior a US$ 8.000.

Para corrigir o erro do exemplo, use a cláusula `HAVING` para restringir grupos:

```
SELECT department_id, AVG(salary)
FROM employees
HAVING   AVG(salary) > 8000
```

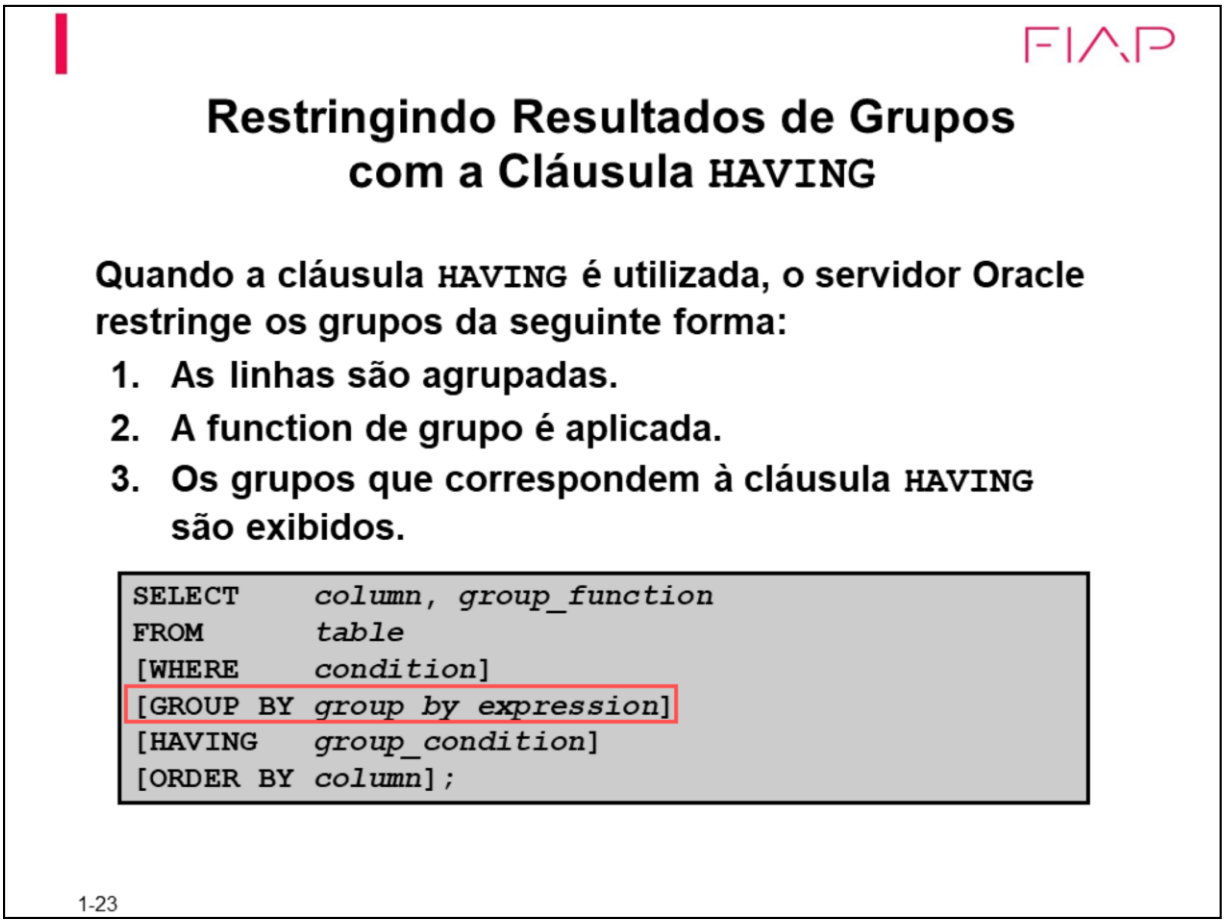
DEPARTMENT_ID	AVG(SALARY)
20	9500
80	10033.3333
90	19333.3333
110	10150



Restringindo Resultados de Grupos

Da mesma maneira que você utiliza a cláusula `WHERE` para restringir as linhas selecionadas, use a cláusula `HAVING` para restringir grupos. Para obter o salário máximo de cada departamento cujo salário máximo é superior a US$ 10.000, você precisa fazer o seguinte:

1. Obtenha o salário médio de cada departamento agrupando por número de departamento.
2. Restrinja os grupos aos departamentos com um salário máximo maior que US$10.000.



Restringindo Resultados de Grupos com a Cláusula `HAVING`

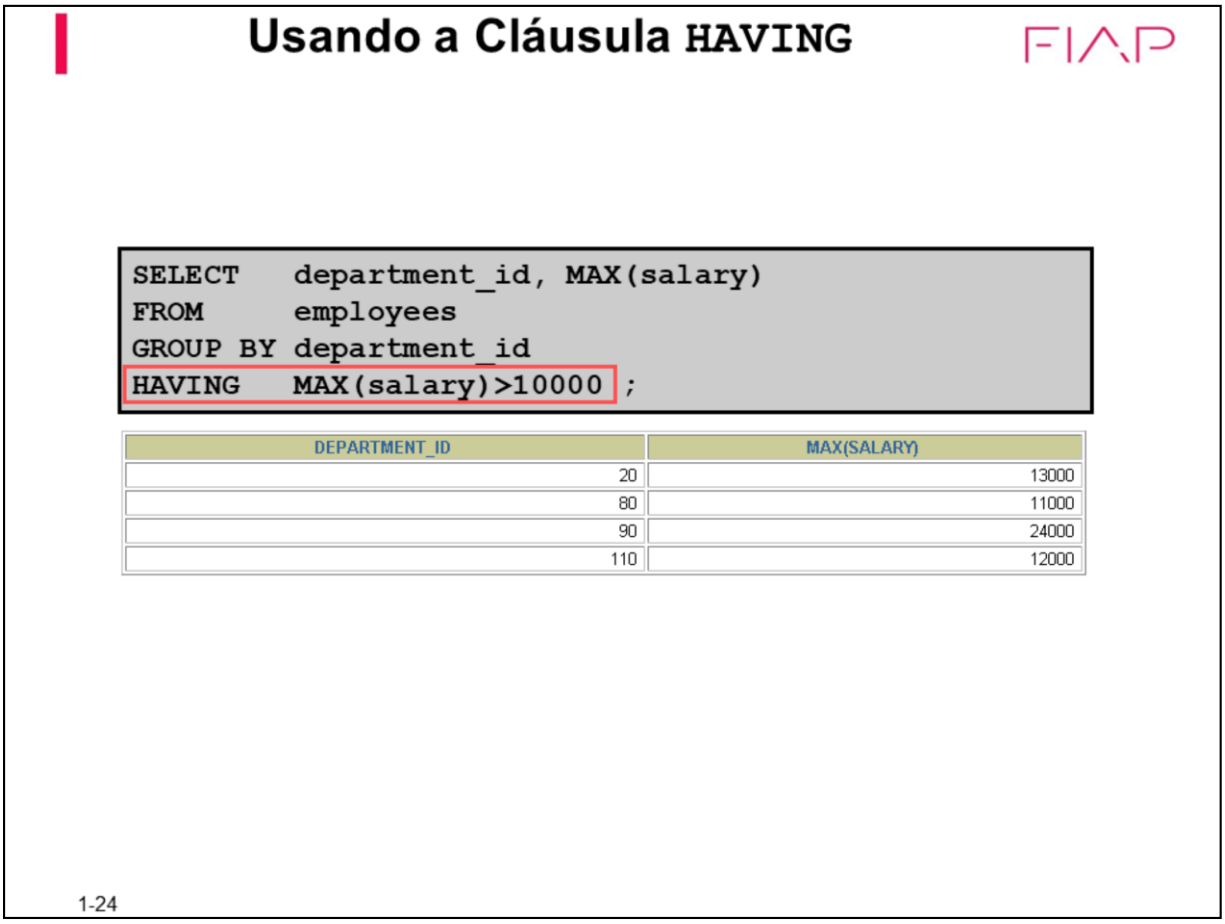
Use a cláusula `HAVING` para especificar quais grupos devem ser exibidos e, dessa forma, restringir ainda mais os grupos com base nas informações agregadas.

Na sintaxe, *`group_condition`* restringe os grupos de linhas retornados aos grupos cuja condição especificada é verdadeira.

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

1. As linhas são agrupadas.
2. A function de grupo é aplicada ao grupo.
3. Os grupos que correspondem aos critérios na cláusula `HAVING` são exibidos.

A cláusula `HAVING` pode anteceder a cláusula `GROUP BY`, mas é recomendável usar a cláusula `GROUP BY` primeiro por razões lógicas. Os grupos são formados e as functions de grupo são calculadas antes de a cláusula `HAVING` ser aplicada aos grupos na lista `SELECT`.



Usando a Cláusula `HAVING`

O exemplo do slide mostra os números e os salários máximos dos departamentos cujo salário máximo é maior que US$ 10.000.

É possível usar a cláusula `GROUP BY` sem uma function de grupo na lista `SELECT`.

Se você restringir as linhas com base no resultado de uma function de grupo, especifique as cláusulas `GROUP BY` e `HAVING`.

Este exemplo mostra os números e os salários médios dos departamentos cujo salário máximo ultrapassa US$ 10.000:

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

DEPARTMENT_ID	AVG(SALARY)
20	9500
80	10033.3333
90	19333.3333
110	10150

Usando a 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
IT_PROG	19200
AD_PRES	24000
AD_VP	34000

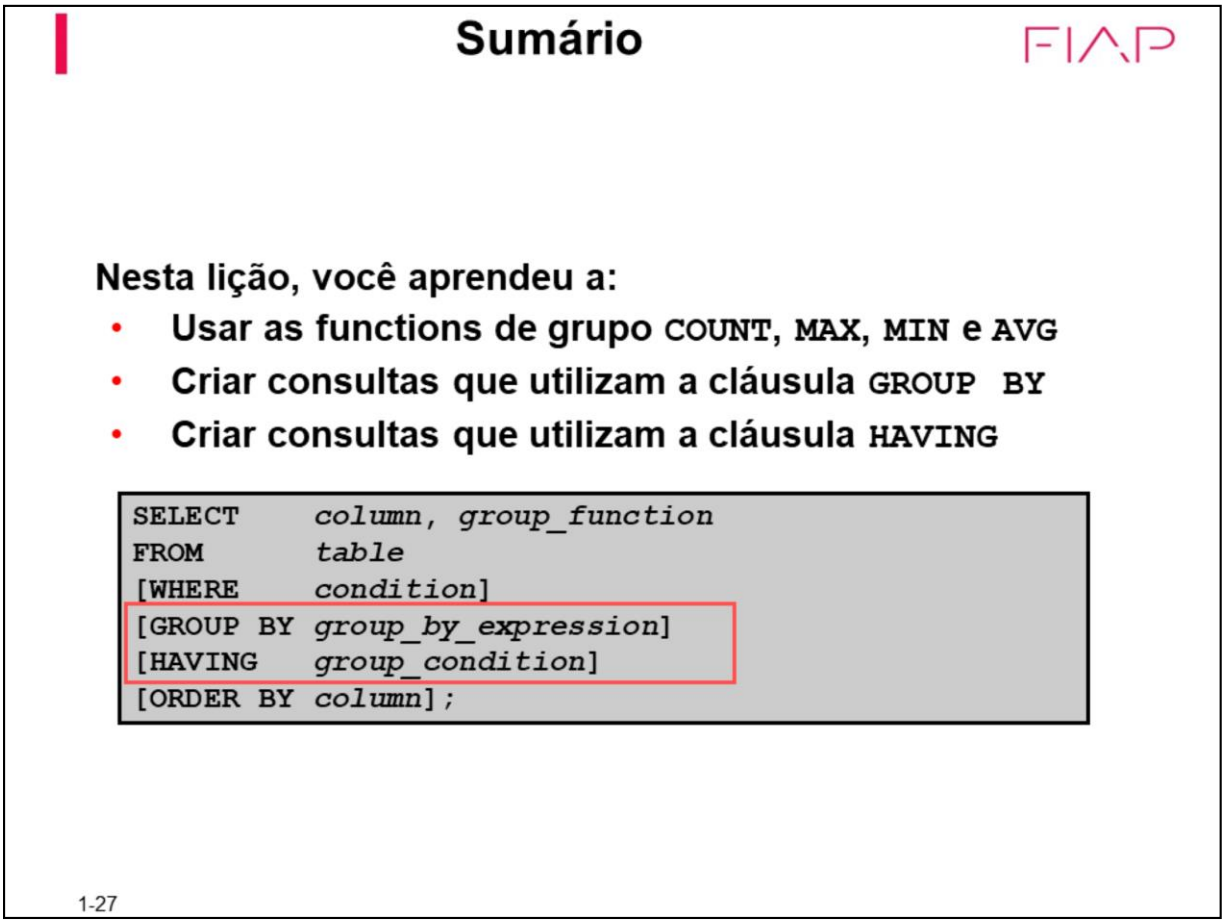
1-25

Usando a Cláusula HAVING (continuação)

O exemplo do slide mostra o ID do cargo e o salário mensal total de cada cargo com uma folha de pagamento total que ultrapassa US$ 13.000. O exemplo exclui representantes de vendas e classifica a lista pelo salário mensal total.

Aninhando Functions de Grupo

É possível aninhar até duas functions de grupo. O exemplo do slide mostra a média de salário máximo.



Sumário

Várias functions de grupo estão disponíveis em SQL, como estas:
`AVG`, `COUNT`, `MAX`, `MIN`, `SUM`, `STDDEV` e `VARIANCE`

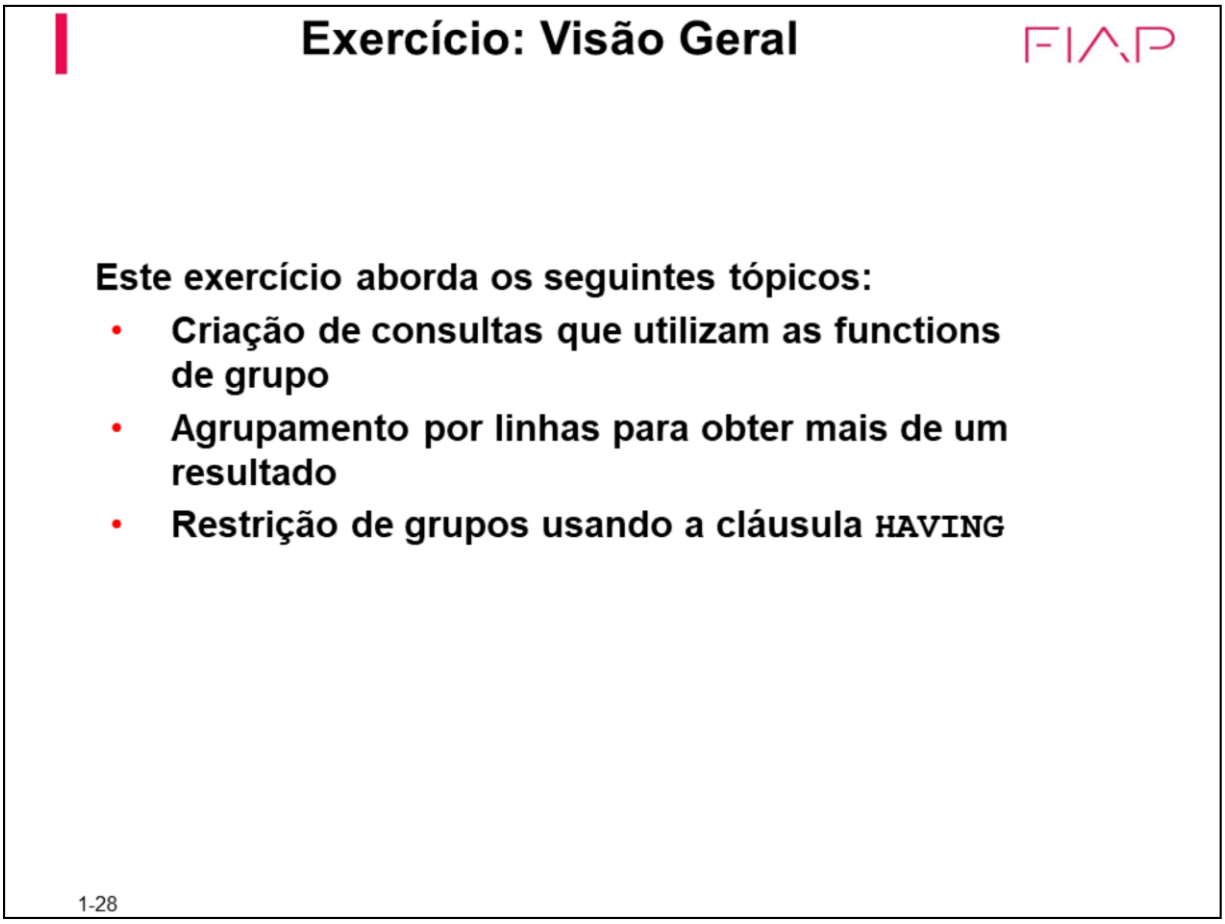
Para criar subgrupos, use a cláusula `GROUP BY`. É possível restringir grupos com a cláusula `HAVING`.

Especifique as cláusulas `HAVING` e `GROUP BY` após a cláusula `WHERE` em uma instrução. A ordem na qual você especifica essas cláusulas após a cláusula `WHERE` não é importante. Informe a cláusula `ORDER BY` por último.

O servidor Oracle avalia as cláusulas na seguinte ordem:

1. Se a instrução contiver uma cláusula `WHERE`, o servidor estabelecerá as linhas candidatas.
2. O servidor identifica os grupos especificados na cláusula `GROUP BY`.
3. A cláusula `HAVING` restringe os grupos de resultados que não atendem aos critérios de grupo especificados na cláusula `HAVING`.

Observação: Para obter uma lista completa das functions de grupo, consulte o manual *Oracle SQL Reference*.



Exercício: Visão Geral

No final deste exercício, você deverá estar familiarizado com a utilização de functions de grupo e a seleção de grupos de dados.

Exercício

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

1. As functions de grupo trabalham com várias linhas para produzir um resultado por grupo.
   Verdadeiro/Falso
2. As functions de grupo incluem valores nulos em cálculos.
   Verdadeiro/Falso
3. A cláusula `WHERE` restringe as linhas antes da inclusão em um cálculo de grupo.
   Verdadeiro/Falso

O departamento de RH necessita dos seguintes relatórios:

4. Obtenha o salário máximo, o salário mínimo, a soma dos salários e o salário médio de todos os funcionários. Atribua os labels `Maximum`, `Minimum`, `Sum` e `Average`, respectivamente, às colunas. Arredonde os resultados para o número inteiro mais próximo. Inclua a instrução SQL no arquivo de texto `lab_04_04.sql`.

	Maximum	Minimum	Sum	Average
1	24000	2100	691400	6462

5. Modifique a consulta em `lab_04_04.sql` para exibir o salário mínimo, o salário máximo, a soma dos salários e o salário médio de cada tipo de cargo. Salve novamente `lab_04_04.sql` como `lab_04_05.sql`. Execute a instrução em `lab_04_05.sql`.

	JOB_ID	Maximum	Minimum	Sum	Average
1	IT_PROG	9000	4200	19200	6400
2	AC_MGR	12000	12000	12000	12000
3	AC_ACCOUNT	8300	8300	8300	8300
4	ST_MAN	8200	5800	36400	7280
5	PU_MAN	11000	11000	11000	11000
6	AD_ASST	4400	4400	4400	4400
7	AD_VP	17000	4800	43600	10900
8	SH_CLERK	4200	2500	64300	3215
9	FI_ACCOUNT	9000	6900	39600	7920
10	FI_MGR	12000	12000	12000	12000
11	PU_CLERK	3100	2500	13900	2780
12	SA_MAN	14000	10500	61000	12200
13	MK_MAN	13000	13000	13000	13000
14	PR_REP	10000	10000	10000	10000
15	AD_PRES	24000	24000	24000	24000
16	SA_REP	11500	6100	250500	8350
17	MK_REP	6000	6000	6000	6000
18	ST_CLERK	3600	2100	55700	2785

Exercício

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

	JOB_ID	COUNT(*)
1	IT_PROG	3
2	AC_MGR	1
3	AC_ACCOUNT	1
4	ST_MAN	5
5	PU_MAN	1
6	AD_ASST	1
7	AD_VP	4
8	SH_CLERK	20
9	FI_ACCOUNT	5
10	FI_MGR	1
11	PU_CLERK	5
12	SA_MAN	5
13	MK_MAN	1

   Generalize a consulta para que o usuário do departamento de RH seja solicitado a informar um cargo. Salve o script no arquivo `lab_04_06.sql`.

7. Determine o número de gerentes sem listá-los. Atribua o label `Number of Managers` à coluna. *Dica: Use a coluna `MANAGER_ID` para determinar o número de gerentes*.

	Number of Managers
1	18

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

	DIFFERENCE
1	21900

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 salário mínimo seja US$ 6.000 ou inferior. Classifique a saída em ordem decrescente de salário.

	MANAGER_ID	MIN(SALARY)
1	102	9000
2	205	8300
3	145	7000
4	146	7000
5	108	6900
6	147	6200
7	149	6200
8	148	6100

Exercício

Se quiser tomar parte em mais um desafio, faça estes exercícios:

10. Crie uma consulta que exiba o número total de funcionários e, desse total, mostre o número de funcionários admitidos em 1995, 1996, 1997 e 1998. Crie cabeçalhos de colunas apropriados.

	TOTAL	1995	1996	1997	1998
1	107	4	10	28	23

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.

	Job	Dept 20	Dept 50	Dept 80	Dept 90	Total
1	IT_PROG	(null)	(null)	(null)	(null)	19200
2	AC_MGR	(null)	(null)	(null)	(null)	12000
3	AC_ACCOUNT	(null)	(null)	(null)	(null)	8300
4	ST_MAN	(null)	36400	(null)	(null)	36400
5	PU_MAN	(null)	(null)	(null)	(null)	11000
6	AD_ASST	(null)	(null)	(null)	(null)	4400
7	AD_VP	(null)	(null)	(null)	34000	43600
8	SH_CLERK	(null)	64300	(null)	(null)	64300
9	FI_ACCOUNT	(null)	(null)	(null)	(null)	39600
10	FI_MGR	(null)	(null)	(null)	(null)	12000
11	PU_CLERK	(null)	(null)	(null)	(null)	13900
12	SA_MAN	(null)	(null)	61000	(null)	61000
13	MK_MAN	13000	(null)	(null)	(null)	13000
14	PR_REP	(null)	(null)	(null)	(null)	10000
15	AD_PRES	(null)	(null)	(null)	24000	24000
16	SA_REP	(null)	(null)	243500	(null)	250500
17	MK_REP	6000	(null)	(null)	(null)	6000
18	ST_CLERK	(null)	55700	(null)	(null)	55700
19	HR_REP	(null)	(null)	(null)	(null)	6500

Bibliografia Utilizada

FIAP

Database SQL Language Reference: http://docs.oracle.com/database/121/SQLRF/toc.htm

Manuais Oracle – Oracle Database 12c: SQL Workshop I/II